



Design and evaluation of a time adaptive multimodal virtual keyboard

Yogesh Kumar Meena¹ · Hubert Cecotti² · KongFatt Wong-Lin³ · Girijesh Prasad³

Received: 23 November 2017 / Accepted: 11 January 2019 / Published online: 8 February 2019
© The Author(s) 2019

Abstract

The usability of virtual keyboard based eye-typing systems is currently limited due to the lack of adaptive and user-centered approaches leading to low text entry rate and the need for frequent recalibration. In this work, we propose a set of methods for the dwell time adaptation in asynchronous mode and trial period in synchronous mode for gaze based virtual keyboards. The rules take into account commands that allow corrections in the application, and it has been tested on a newly developed virtual keyboard for a structurally complex language by using a two-stage tree-based character selection arrangement. We propose several dwell-based and dwell-free mechanisms with the multimodal access facility wherein the search of a target item is achieved through gaze detection and the selection can happen via the use of a dwell time, soft-switch, or gesture detection using surface electromyography in asynchronous mode; while in the synchronous mode, both the search and selection may be performed with just the eye-tracker. The system performance is evaluated in terms of text entry rate and information transfer rate with 20 different experimental conditions. The proposed strategy for adapting the parameters over time has shown a significant improvement (more than 40%) over non-adaptive approaches for new users. The multimodal dwell-free mechanism using a combination of eye-tracking and soft-switch provides better performance than adaptive methods with eye-tracking only. The overall system receives an excellent grade on adjective rating scale using the system usability scale and a low weighted rating on the NASA task load index, demonstrating the user-centered focus of the system.

Keywords Gaze-based access control · Adaptive control · Multimodal dwell-free control · Graphical user interface · Virtual keyboard · Eye-typing · Human-computer interaction

1 Introduction

Several types of modalities have been recently evaluated on natural user interface design for intuitive interaction with computers. For example, electroencephalogram (EEG) based brain-computer interface (BCI), eye-tracking based human-computer interface (HCI), electromyography (EMG) based gesture recognition, speech recognition, and different input access switches have been adopted for natural user interface methods [1–4]. Among these approaches, eye-tracking considers the position of the eye relative to the head, and the orientation of the eyes in space, or the point

of regard. Eye-tracking has many applications to communicate and control devices such as eye-typing interfaces, robotics control, for facilitating human–computer interactions, assessing web page viewing behavior, entertainment (e.g., video games), switching control, and virtual automobile control [5–9].

In eye-tracking research, broadly two methods have been used to measure eye movements. First, a wearable-camera-based method wherein a high-resolution image for calculating the gaze point can be obtained from the wearable camera at a close distance. However, the user may experience discomfort during eye-tracking interactions because the camera equipment must be worn [10]. Second, a remote-camera-based method wherein the gaze position is captured through non-contacting fixed cameras without any additional equipment or support. In this case, because the image resolution for the eye is relatively low, pupil tremors cause severe vibrations of the calculated gaze point. Furthermore, time-varying characteristics of the remote-camera-based method can lead to a low accuracy and the need for frequent calibration [11, 12].

✉ Yogesh Kumar Meena
y.k.meena@swansea.ac.uk

¹ Future Interaction Technology Lab, Swansea University, Swansea, UK

² Department of Computer Science, College of Science and Mathematics, Fresno State University, Fresno, CA, USA

³ School of Computing, Engineering and Intelligent Systems, Ulster University, Londonderry, UK

Similar to EEG-based BCI, gaze-based control can be accessed in eye tracking based HCI in both synchronous (*cue-paced*) and asynchronous (*self-paced*) modes [13]. In synchronous mode, a user action (e.g., click events) is performed after a fixed interval (trial period) whereas in asynchronous mode the click events are performed through dwell time. In synchronous mode, an item is selected when the user focuses on the target item most of the time during a predefined trial duration. At the end of the trial, the target item is selected, if it has the *maximum* duration of the focus (dos) compared to the estimated dos on other items. In such case, the user has to spend a maximum amount of time on the desired item. In asynchronous mode, an item is selected when the user is focusing his/her attention by fixating the target item for a specific predefined period of time *continuously*. These two methods effectively reflect user intention, and often are time-consuming when there are many selections to be made [1,14].

The issues related to the high number of commands that can be accessed at any moment, the Midas touch problem [15–18], and the requirement of adapting parameters need to be taken into account to design a user interface meeting these constraints. The goal of this study is to propose several time-adaptive, dwell-based, and dwell-free methods evaluated using multimodal access facility with beginner users. In this work, we address these issues with the following novel major contributions: (1) a set of methods for the adaptation over time of the dwell time in asynchronous mode, (2) a set of methods for the adaptation of the trial period in synchronous mode, and (3) a benchmark with beginner users of several dwell-based and dwell-free mechanisms with the multimodal access facility wherein the search of a target item is achieved through gaze detection and the selection can happen via the use of a dwell time, soft-switch, or gesture detection using surface electromyography (sEMG) in asynchronous mode; and the search and selection may be performed with eye-tracker in synchronous mode.

This paper is organized as follows: Sect. 2 presents a critical literature review. Section 3 proposes new gaze-based control methods for both synchronous and asynchronous operations of HCI. It includes a benchmark of several dwell-free mechanisms to overcome the Midas touch problem of HCI including proposed models of multimodal system. Section 4 describes the development of the multimodal virtual keyboard system. Particularly, it takes into account design challenges related to the management of a complex structure and a large set of characters in the Hindi language. Section 5 provides the design, experimental procedure, and the performance evaluation methods. The results are presented in Sect. 6. The subjective evaluation of the system is provided in Sect. 7. The contributions of this paper and their impacts are discussed in Sect. 8. Finally, Sect. 9 concludes the paper.

2 Background

Generally, most of the eye-tracking methods are developed in asynchronous mode as it lets some freedom to the user to follow his/her own pace. Interestingly, in such mode the dwell time should be sufficiently long enough for the correct selection of the intended item, otherwise high false selections (Midas touch problem) may happen, leading to increased frustration for the user and thus delaying the overall process [16,17]. The choice of an effective dwell time has encouraged some researchers to propose adaptive strategies for the choice of the dwell time [19,20]. Indeed, with such enhancements, users can select desired items easily, increasing the overall systems performance. In one of these studies, the dwell time was adjusted based on the exit time [21]. This online adjustment, however, suffers from delayed feedback and uncontrolled variations in the exit time. In a different work, dwell time was tuned by controlling the speed of the control keys [22]. One of the key drawbacks of this method is the requirement of extra selection time.

A recent study proposed a probabilistic model for gaze based selection, which adjusts the dwell time based on the probability of each letter based on the past selection [23]. A different work suggested an approach that dynamically adjusts the dwell time of keys by using selection and location of the keys on the keyboard [24]. However, one of the limitations of these studies is the manual selection of the hyperparameter values (e.g., thresholds) and user's variability, which may not be suitable for other applications. Therefore, adjustment of dwell time largely depends on the application type, and the parameter selections. The outcome of these systems depends upon the typing errors/correction command but not much attention has been paid to these parameters while designing the automation of dwell time choice.

On the other hand, online adjustment of a fixed interval time in synchronous mode has been largely ignored in eye typing studies. Such an approach can be valuable for people who are not able to maintain their gaze on a desired location for a sufficient continuous period, e.g., people suffering from nystagmus, but can still keep their gaze on the desired location most of the time compared to other undesired items. Another advantage of the synchronous mode is for users to follow a tempo during the typing task. However, this mode does not require the complete user attention while performing the typing task. Thus, this mode can be useful for special kinds of users, e.g., with attention deficit hyperactivity disorder.

Dwell-free techniques have been implemented with user interfaces of virtual keyboard applications wherein the dwell-free eye-typing systems provide moderately higher text entry rate than dwell based eye-typing systems [25–27]. The user interfaces of virtual keyboard systems have been designed

based on various keyboard approaches such as the Dvorak, FITALY, OPTI, Cirrin, Lewis, Hookes, Chubon, Metropolis, and ATOMIK [28]. However, it is challenging to control these keyboards through gaze detection due to the underlying gaze detection procedure where the accuracy decreases in relation to the proximity of the commands. In particular, dwell-free gaze controlled typing system such as EyeWrite [29], dwell-free eye-typing [26], Dasher [30], Eyeboard [31], Eyeboard++ [32], EyePoint [33], EyeSwipe [34], Filtered-eye-typing [35], StarGazer [36], openEyes [37], and Gazing with pEyes [38] have been effectively implemented for both assistive and mainstream uses.

Moreover, the hand and eye motion have been utilized to control the virtual keyboard for disabled people [39]. The eye-tracking-based communication system has been developed for patients having major neuro locomotor disabilities wherein they can verbally communicate through signs or in writing [40]. Another concern is that above approaches incorporate a large number of commands on the user interface leading to a lower text entry rate [41]. Other dwell-free techniques include multimodal and hybrid interfaces. These techniques address issues highlighted in previous studies [18, 42–48]. In particular, these studies have introduced a dwell-free technique for an eye-typing system, which focused on a combination of different modalities such as eye-tracking, smiling movements, input switches, and speech recognition.

The multimodal interfaces can be operated in two distinct modes. The first mode uses eye gaze as a cursor-positioning tool, and either smiling movements, input switches, or voice commands are used to perform mouse click events. For example, a multimodal application involving the combination of eye gaze and speech has been developed for selecting differently sized, shaped, and colored figures [49]. A multimodal interface involving eye gaze, speech, and gesture has been proposed for object manipulation in virtual space [50]. However, a user study shows that a gaze and speech recognition based multimodal interaction is not as fast as using mouse and keyboard for correction; but a gaze enhanced correction significantly outperforms voice alone correction and is preferred by the users, offering a truly hands-free means of interaction [51]. A previous study has introduced a dwell-free technique for an eye-typing system that focused on a combination of different modalities such as eye-tracking and input switches [43]. The dwell-free techniques provide an effective solution to overcome the Midas touch problem with gaze only and/or in combination with several input modalities. However, the choice of input modalities depends on the individual users, their needs, and the type of applications.

The usability of virtual keyboard systems with gaze-based access controls is currently impaired by the difficulty to set optimal values to the key parameters of the system, such as the dwell time, as they can depend on the user (e.g., fatigue,

knowledge of the system) [28]. In addition, the fluctuation of attention, the degree of fatigue, and the users' head motion while controlling the application represent obstacles for efficient gaze-based access controls as they can lead to low performance [52]. These continuous variations can be overcome by recalibrating the system at regular intervals or when a significant drop in performance is observed. However, this procedure is time consuming and may not be user-friendly.

A solution proposed in this work is to adapt the system over time based on its current performance by considering key features of the application (e.g., correction commands) in both synchronous and asynchronous modes. The proposed adaptation methods are based on users' typing performance whereas existing systems for the adaption of the dwell time require a significant number of hyperparameters and thresholds that are set manually, which prevent fair comparisons with a different virtual keyboard layout. Furthermore, we propose dwell-free techniques with the multimodal access facility to overcome the conventional issues associated with individual input modalities. In particular, the addition of a switch or the regular mouse that have no thresholds can give a clear performance baseline. Moreover, switch mechanisms can provide a baseline performance that allows to better appreciate the performance that is obtained with the dwell time, and from the adaptive dwell time.

In this study, we provide multiple levels of comparisons to better appreciate the performance of the proposed approaches of beginner users. A synergistic fusion of these modalities can be used for communication and control purposes as per user's particular preferences. Such an approach is particularly relevant for stroke rehabilitation where a user may desire to keep a single graphical layout and seamlessly progress from a gaze only modality to the mouse or touch screen throughout the rehabilitation process.

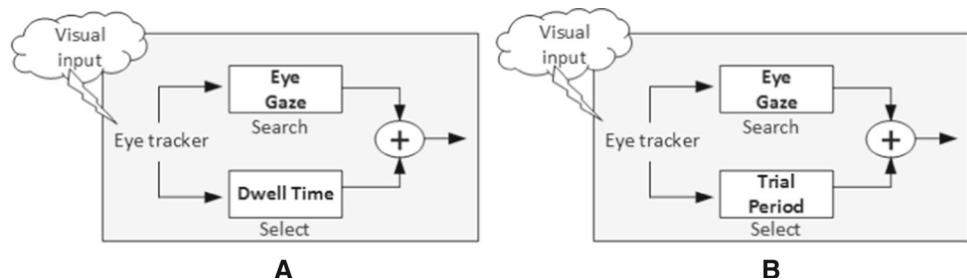
3 Proposed methods

In this study, two methods for the adaptation (over time) of the dwell time in asynchronous mode and the trial period in synchronous mode are proposed for gaze-based access control and compared with non-adaptive methods. We have set a benchmark for several dwell-free mechanisms including several portable, non-invasive, and low-cost input devices. A multimodal dwell-free approach is presented to overcome the Midas touch problem of the eye-tracking system.

3.1 Gaze-based access control

A gaze based control can be accessed in two different modes (see Fig. 1). The eye-tracking can be used for both search and selection purposes with synchronous and asynchronous (i.e., self-paced) modes. First, the asynchronous mode offers

Fig. 1 Proposed models of gaze-based access control modes. The search and selection of the items are performed by **a** eye-tracker only in asynchronous mode and **b** eye-tracker only in synchronous mode



a natural mode of interaction without waiting for an external cue. The command selection is managed through the dwell time concept. During this mode, the users focus their attention by fixating the target item for a specific period of time (i.e., dwell time in seconds) which results in the selection of that particular item (see Fig. 1a). Second, the way of interaction in synchronous mode is mainly based on an external cue. This mode can be used to avoid artifacts such as involuntary eye movements of users as the command is selected at the end of the trial duration/trial period. During this mode, the users focus their attention by fixating an item during a single trial of a particular length (i.e., the trial length (in seconds)), and the item is selected at the end of the trial based on the maximum duration of focus (see Fig. 1b).

We denote the total number of commands that are available at any time in the system by M . Each command c_i is defined by the coordinates corresponding to the center of its box (x_c^i, y_c^i) , where $i \in \{1 \dots M\}$. We denote the gaze coordinates at time t by (x_t, y_t) , then the distance between a command box and the current gaze position, d_t^i is defined by its Euclidean distance as:

$$d_t^i = \sqrt{(x_c^i - x_t)^2 + (y_c^i - y_t)^2} \quad (1)$$

We denote the selected command at time t by select_t , where $1 \leq \text{select}_t \leq M$. For the asynchronous and synchronous modes, we defined the dwell time and the trial period as Δt_0 and Δt_1 , respectively. Δt_0 represents the minimum time that is required to select a command i.e., when a subject continuously keeps his/her gaze on a command. If the user looks outside the screen, no item will be selected and the timer is restarted when user next looks back at the targeted item on the screen. In synchronous mode, Δt_1 represents the time after which a command has been selected based on the maximum duration of focus, i.e., the selected item is the one at which the user was looking during the trial period for maximum duration. If the user is shifting his/her attention by fixating on an item outside the screen after some time then an item can be selected because the timer is still in progress.

The approach to select a command in asynchronous mode is detailed in the Algorithm 1. δ represents a counter for the selection of each command. The method to select a command

after each trial, in synchronous mode, is presented in the Algorithm 2. The vector w represents the weight of each command during a trial and α_1 represents a threshold used for the selection. Besides, each time point is weighted by \sqrt{t} in order to emphasize the gaze positions towards the end of the trial. select_s represents the command that is selected after each trial, $\text{select}_s \in \{-1, 1 \dots M\}$, if the value is -1 then no command is selected, otherwise one of the M commands is obtained.

Algorithm 1 Command selection - asynchronous mode

```

1:  $t \leftarrow 0, \delta \leftarrow 0$ 
2:  $\text{select}_0 \leftarrow -1$ 
3: while (true) do
4:    $\text{select}_t \leftarrow \text{argmin}_{1 \leq i \leq M}(d_t^i)$ 
5:   if ( $\text{select}_t == \text{select}_{t-1}$ ) then
6:      $\delta \leftarrow \delta + 1$ 
7:    $v \leftarrow 255 * (\Delta t_0 - \delta) / \Delta t_0$  {update color}
8:   else
9:      $\delta \leftarrow 0$ 
10:  if ( $\delta \geq \Delta t_0$ ) then
11:    Run command ( $\text{select}_t$ )
12:     $\delta \leftarrow 0$ 
13:   $t \leftarrow t + 1$ 

```

Algorithm 2 Command selection - Synchronous mode

```

1:  $\alpha_1 \leftarrow 0.5$ 
2: while (true) do
3:    $w(i) \leftarrow 0, \forall i \in \{1..M\}$ 
4:   for  $t \leftarrow 1$  to  $\Delta t_1$  do
5:      $\text{select}_t \leftarrow \text{argmin}_{1 \leq i \leq M}(d_t^i)$ 
6:      $w(\text{select}_t) \leftarrow w(\text{select}_t) + \sqrt{t}$ 
7:    $\text{select}_s \leftarrow \text{argmax}_{1 \leq i \leq M}(w)$ 
8:    $P(\text{select}_s) \leftarrow \frac{\max(w)}{\sum_{j=1}^M w(j)}$ 
9:   if ( $P(\text{select}_s) \geq \alpha_1$ ) then
10:    Run command ( $\text{select}_s$ )
11:   else
12:     $\text{select}_s \leftarrow -1$ 

```

However, the performance of both synchronous and asynchronous modes depends on time dependent characteristics of the users when using the predefined time parameters to select an item on the screen. Therefore, the adaptation over

time is essential for designing a more natural mode of interaction. The adaptive algorithms are explained in the next subsection.

3.1.1 Eye-tracker with adaptive dwell time in asynchronous mode

For the adaptive dwell time in asynchronous mode, we consider two rules where Δt_0 can change between Δmin_0 and Δmax_0 . In this study, Δmin_0 and Δmax_0 correspond to 1 s and 5 s, respectively [43,53]. Initially, Δt_0 is set to 2000 ms. Both rules are included in Algorithm 3 where, β_1 represents a particular dwell time increment and decrement in ms. The ϵ_1 and ϵ_2 indicate a threshold of dwell time increment and decrement, respectively. In the first rule, if the number of commands, N_{cor} , corresponding to a “delete” or “undo” represents more than half of the commands in the history of N_h commands (i.e., $2N_{cor} \geq N_h$), then we assume that there exists some difficulties for the user, and the dwell time has to be increased. The second rule is based on the assumption that if the average time between two consecutive commands during N_h commands is close to the dwell time, then the current dwell time acts as a bottleneck and it can be reduced. We denote the variable that contains the difference of time between two consecutive commands by Δt_c in which $\Delta t_c(k)$ corresponds to the time interval between the command k and $k-1$. The current average of Δt_c over the past N_h commands is defined by:

$$\overline{\Delta t_c}(k) = \frac{1}{N_h} \sum_{k_0=1}^{N_h} \Delta t_c(k - k_0) \quad (2)$$

Algorithm 3 Adaptive dwell time

```

1:  $\beta_1 \equiv 500 \text{ ms}$  ,  $\epsilon_1 \equiv 500 \text{ ms}$  ,  $\epsilon_2 \leftarrow 0.5$ 
2: if ( $N_{cor}/N_h > \epsilon_2$ ) then {rule #1}
3:    $\Delta t_0 \leftarrow \Delta t_0 + \beta_1$ 
4: if ( $|\Delta t_c(k) - \Delta t_0| \leq \epsilon_1$ ) then {rule #2}
5:    $\Delta t_0 \leftarrow \Delta t_0 - \beta_1$ 
```

3.1.2 Eye-tracker with adaptive trial period in synchronous mode

With the adaptive trial period (i.e., trial duration Δt_1) in synchronous mode, we consider three rules, where Δt_1 can change between Δmin_1 and Δmax_1 . In this study, Δmin_1 and Δmax_1 correspond to 1 s and 5 s, respectively [43,53]. Initially, Δt_1 is set to 2000 ms. The three rules are summarized in Algorithm 4 where β_2 represents a particular trial period increment and decrement in ms. The ϵ_2 indicates a threshold of trial period to select an item and ϵ_3 represents

the mean probability of a particular command deletion. In the first rule, we define by $\overline{P(\text{select}_s)_k}$ the average probability to detect a command in the k^{th} trial by considering the last N_h previous trials. If this probability is high, then it indicates that the commands are selected in a reliable manner and the trial period can be decreased.

$$\overline{P(\text{select}_s)_k} = \frac{1}{N_h} \sum_{k_0=1}^{N_h} P(\text{select}_s)_{k-k_0} \quad (3)$$

Algorithm 4 Adaptive trial period

```

1:  $\beta_2 \equiv 500 \text{ ms}$  ,  $\epsilon_2 \leftarrow 0.5$  ,  $\epsilon_3 \leftarrow 0.9$ 
2: if ( $\overline{P(\text{select}_s)_k} > \epsilon_3$ ) then {rule #1}
3:    $\Delta t_1 \leftarrow \Delta t_1 - \beta_2$ 
4: if ( $N_r/N_h \geq \epsilon_2$ ) then {rule #2}
5:    $\Delta t_1 \leftarrow \Delta t_1 + \beta_2$ 
6: if ( $N_{cor}/N_h \geq \epsilon_2$ ) then {rule #3}
7:    $\Delta t_1 \leftarrow \Delta t_1 + \beta_2$ 
```

The second rule deals with the trials with no command selection. In this case, we assume that if a command was not selected during the interval Δt_1 , it means that Δt_1 was too short to allow the user to select an item. In such a case, the trial period is increased where the number of rejected commands are N_r in the history of the last N_h commands ($N_r \leq N_h$). In the third rule, if the number of commands related to corrections, N_{cor} , corresponding to a “delete” or “undo” represents more than half of the commands in the history of N_h commands included, then we assume that there exist some difficulties for the user, and the trial period has to be increased.

3.2 Dwell-free mechanisms

A benchmark of several dwell-free mechanisms using several portable, non-invasive, and low-cost input devices (e.g., a surface electromyography; and an access soft-switch) is proposed. There were five different combinations of the input modalities which provided four different dwell-free models (see Fig. 2) to control a virtual keyboard system. First, the search and selection of the target item were performed by the user’s eyes without eye-tracking and a normal computer mouse, respectively (see Fig. 2a). Second, the search of the target item was performed by the user’s eyes without eye-tracking and the participant used the touch screen to finally select an item (see Fig. 2b). Third, the eye-tracker along with the soft-switch were used in a hybrid mode wherein the user focused their attention by fixating their gaze onto the target item, and the selection happens via a soft-switch (see Fig. 2c). Fourth, the eye-tracker was used in combination with five different sEMG-based hand gestures wherein eye-gaze was

Fig. 2 Proposed models of multimodal system based on various input modalities. The search and selection of the items are performed by **a** naked eyes without eye-tracker and computer mouse, **b** naked eyes without eye-tracker and touch screen, **c** eye-tracker and soft-switch, and **d** eye-tracker and sEMG based hand gesture

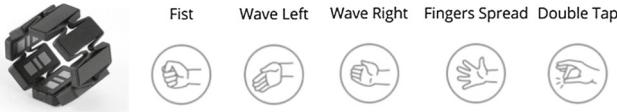
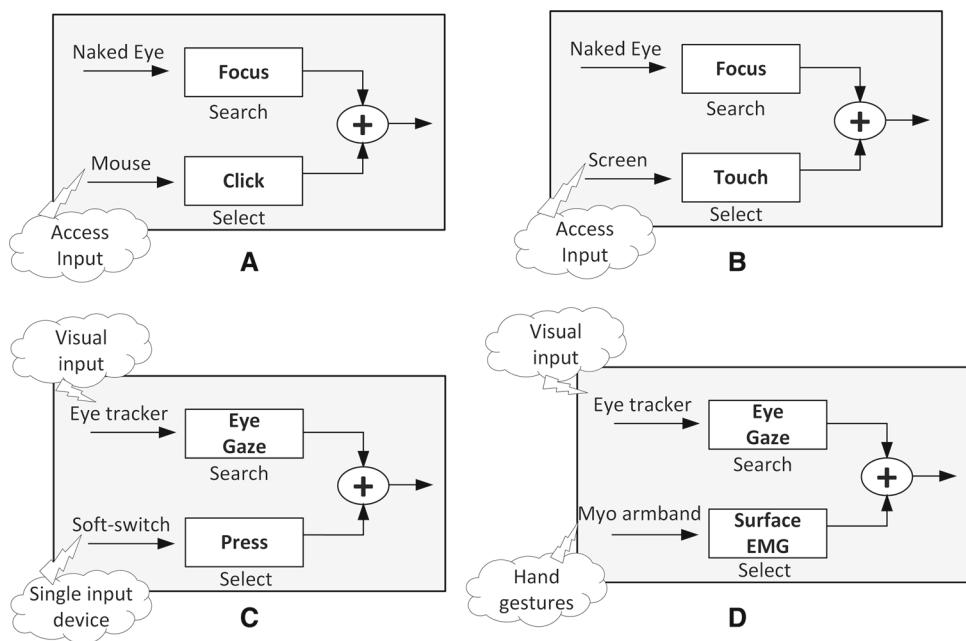


Fig. 3 Myo gesture control armband with the five hand gestures:fist (hand close), wave left (wrist flexion), wave right (wrist extension), finger spread (hand open), and double tap

used for search purpose and each gesture acted as an input modality to select the item (see Fig. 2d). This combination of input modality used five different hand gestures (see Fig. 3) to select a command on screen.

3.2.1 Command selection with single modality

The single input devices such as mouse and touch screen are well known methods (that is, very familiar to users as opposed to eye-tracking) to access the computing devices. Therefore, these devices are integrated as a baseline measure of performance, while operating the virtual keyboard system. Two basic models of dwell-free mechanisms for search and selection of the command are presented in Fig. 2a, b). With both single input modalities (mouse and touch-screen), the user only needs to hit at the target item for selection via the mouse or the touch-screen. Once the item is selected, the user receives an auditory feedback, i.e., an acoustic beep.

3.2.2 Command selection with multimodality

Two models of the dwell-free multimodal system are proposed in Fig. 2c, d) wherein a command can be selected

without using dwell time. In particular, an eye-tracker is used with a soft-switch and/or sEMG hand gestures.

(A) Eye-tracker with soft-switch: The addition of the soft-switch has helped to overcome the Midas touch problem, as the user needs only point to the target item through the eye-tracker, and the selection happens via the soft-switch. In this study, the soft-switch was pressed by the user's dominant hand. The searching of the target items is implemented by Equation 1. The color-based visual feedback is provided to the user during the searching of an item (see Sect. 4). The visual feedback allows the user to continuously adjust and adapt his/her gaze to the intended region on the screen. Once the item is selected, the auditory feedback is given to the user.

(B) Eye-tracker with sEMG hand gestures: The sEMG hand gestures combined with an eye-tracker in a hybrid mode can provide extra input modalities to the users. The eye-tracker is used to point to a command on the screen using Equation 1. Then, the command is selected through a hand gesture by using predefined functions from the Myo SDK. Five conditions were evaluated related to gesture control with the Myo: fist (hand close), wave left (wrist flexion), wave right (wrist extension), finger spread (hand open), and double tap (see Fig. 3). The color-based visual feedback is provided to the user during the searching of an item (see Sect. 4). After the selection of each item, the user gets the auditory feedback as well. Thus, the hybrid system helps to overcome the Midas touch problem of gaze controlled HCI system.

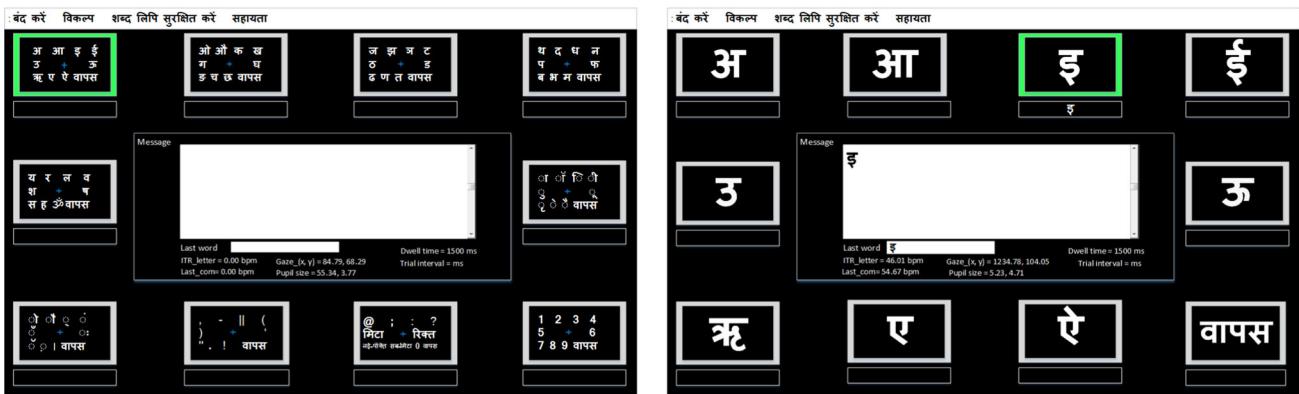


Fig. 4 Layout of proposed Hindi virtual keyboard application in level one when c1 is selected (left) and level two after the selection of c1 (right), with the ten commands (from left to right, top to bottom)

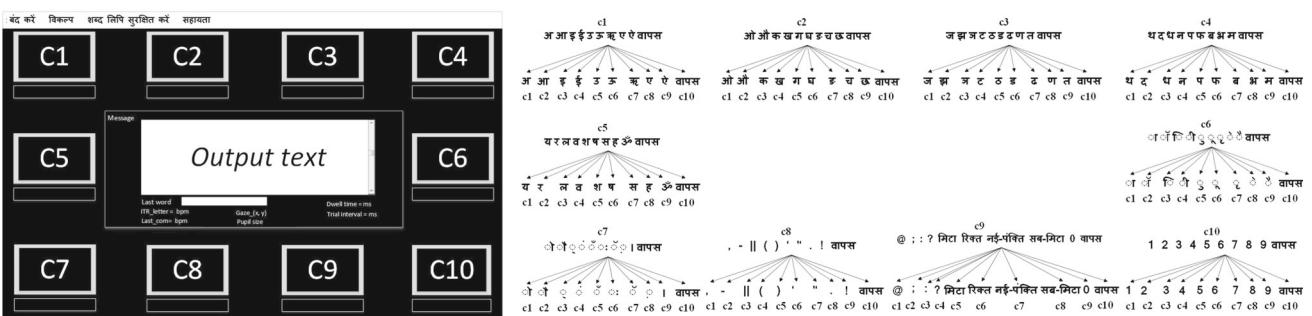


Fig. 5 Positions of the ten commands in the Hindi virtual keyboard application (left), the tree structure depicting the command tags used for letter selection (right)

4 System overview

The developed graphical user interface (GUI) consists of two main components, which are depicted in Fig. 4. The first component is a command display wherein a total of ten commands are presented and the command currently being pointed to, is highlighted in a different color. The second component is an output text display where the user can see the typed text in real-time. The position and tree structure of the ten commands (i.e., c1 to c10) are depicted in Fig. 5. An alphabetical organization with script specific arrangement layout is developed as the alphabetic arrangement is easier to learn and remember, specially for complex structured language [54]. The size of each rectangular command button is approximately 14% of the GUI window. All command buttons are placed on the periphery of the screen while the output text box is placed at the center of the screen (see Fig. 4).

The GUI of the virtual keyboard is based on a multi-level menu selection method comprised of ten commands at each level [55,56]. This approach can be beneficial when the screen size is limited and it takes into account potential confusions that may arise with gaze detection if two commands are too close from each other [57,58]. The proposed hierarchical layout is organized as a rectangle, and not as a

circle, but it follows the same spirit as a crude pie menu at each level [59]. The tree-based structure of the GUI provides the ability to type 45 Hindi language letters, 17 different matras (i.e., diacritics) and halants (i.e., killer strokes), 14 punctuation marks and special characters, and 10 numbers (from 0 to 9). Other functionalities such as delete, delete all, new line, space, and go back commands for corrections are included.

The first level of the GUI consists of 10 command boxes; each represents a set of language characters (i.e., 10 characters). 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 (i.e., at first level of GUI) where the desired character is located. The successful selection of 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 at the previous level. In the second step, the user can see the desired character and finally select it for writing to the text-box. After the selection of a particular character at the second level, the GUI goes back to the initial stage (i.e., at first level) to start further iterations. The placement and size of the command boxes are identical at both levels of GUI.

Table 1 Participants' demographics in Group A

Variables	Participant ID											
	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12
Age (years)	31	30	30	30	29	28	32	27	29	21	29	25
Gender	M	M	M	M	M	M	M	M	F	M	F	F
Dominant side	R	R	R	R	R	R	R	R	R	R	R	R
Vision correction	No	No	Yes	No	No	No	Yes	Yes	Yes	Yes	No	No

Table 2 Participants' demographics in Group B

Variables	Participant ID											
	B01	B02	B03	B04	B05	B06	B07	B08	B09	B10	B11	B12
Age (years)	30	28	32	25	28	26	25	23	23	28	24	27
Gender	M	M	M	M	M	M	M	F	M	M	M	F
Dominant side	R	R	R	R	R	R	R	R	R	R	L	L
Vision correction	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	No	Yes

In addition, this system can be utilized to overcome the shortcomings of previous study [43] by adding multiple modalities and extra command features to write all the Hindi language letters including half letter scripts and required punctuation marks. The halant is commonly used to write half letters. It is represented by . For instance, can be written as + + + . Thus, a halant-based approach is also considered in this study, wherein can be written as + + . A similar process can be applied to three character words (e.g., character 1 + halant + character 2 + halant + character 3). Another special matra is known as nukta. It is represented by . For instance, can be written as + . Therefore, while designing a virtual keyboard application for the Hindi language these nukta and halant based approaches must be considered. A demonstrative video of the system is available online with eye-tracking only in asynchronous mode.¹

On a virtual keyboard using eye-tracking, it is necessary that the user is given an efficient feedback that the intended command box/character was selected to avoid mistakes and increase efficiency. Hence, a visual feedback is provided to the user by a change in the color of the button border while looking at it. Initially, the color of the button border is silver (RGB: 192,192,192). When the user fixates and maintains his/her gaze to a particular button for a duration of time t , the color of the border changes linearly in relation to the dwell time Δt_0 or the trial period (i.e., trial duration) Δt_1 and the border becomes greener with time. The RGB color is defined as ($R = v$, $G = 255$, $B = v$), where $v = 255 * (\Delta t_0 - t) / \Delta t_0$.

The visual feedback allows the user to continuously adjust and adapt his/her gaze to the intended region on the screen. An audio feedback is provided to the user through an acoustic beep after successful execution of each command. This

sound makes them proactive so that they can prepare for the next character. Moreover, to improve the system performance by using minimal eye movements, the last five used characters are displayed in the GUI at the bottom of each command box, helping the user to see the previously written characters without shifting significantly their gaze from the desired command box to the output display box. Here, the goal is to avoid visual attention shifts between the message box that contains the full text and the boxes that contain the commands [2].

5 Experimental protocol

5.1 Participants

A total of twenty-four healthy volunteers (5 females) in the age range of 21–32 years (27.05 ± 2.96) participated in this study. Fifteen participants performed the experiments with vision correction. These participants were divided equally into two groups i.e., Group A (see in Table 1) and Group B (see in Table 2) for different experiments. The participants' demographics were kept similar in both groups. Experiments 1 and 2 were performed with Group A, whereas experiments 3 and 4 were completed with Group B. No participant had prior experience of using an eye-tracker, soft-switch and/or sEMG with the application. Participants were informed about the experimental procedure, purpose, and nature of the study in advance. There was no financial reward provided to the participants. The Helsinki Declaration of 2000 was followed while conducting the experiments.

5.2 Multimodal input devices

Three different input devices were used in this study (see in Fig. 6). First, a portable eye-tracker (The Eye Tribe Aps,

¹ <https://youtu.be/e4DILEsa7fw>.



Fig. 6 Commercially available input devices. These devices are used for searching and selection of the items on virtual keyboard application. These devices can be utilized separately and/or in combination with each other to meet the particular needs of the user

Denmark) was used for pursuing the eye gaze of the participants [60]. Second, gesture recognition was obtained with the Myo armband (Thalmic Labs Inc., Canada) for recording sEMG. This non-invasive device includes a 9 degree-of-freedom (DoF) Inertial Measurement Unit (IMU), and 8 dry sEMG sensors. The Myo can be slipped directly on the arm to read sEMG signals with no preparation needed for the participant (no shaving of hair or skin-cleaning) [61]. Third, a soft-switch (The QuizWorks Company, USA) is used as a single-input device [62].

5.3 Data acquisition

The eye-tracker data was recorded at 30 Hz sampling rate. It involves binocular infrared illumination with spatial resolution (0.1 root mean square (RMS)), which records *x* and *y* coordinates of gaze and pupil diameter for both eyes in mm. The Myo armband provides sEMG signals with a sampling frequency of 200 Hz per channel. Electrode placement was set empirically in relation to the size of the participant's forearm because the Myo armband's minimum circumference size is about 20 cm. An additional short calibration was performed for each participant with the Myo (about 1 min). The soft-switch was used as a single-input device to select a command on a computer screen. Participants were seated in a comfortable chair in front of the computer screen. The distance between a participant and the computer screen (PHILIPS, 23 inches, 60 Hz refresh rate, optimum resolution: 1920 × 1080, 300 cd/m², touch-screen) was about 80 cm. The vertical and horizontal visual angles were measured at approximately 21 and 36 degrees, respectively.

5.4 Design and operational procedure

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 SDK [63] was used to acquire the gaze data. Prior to each experiment, a calibration session lasting about 20 s, using a 9-point calibration scheme was conducted for each participant. The rating control provides a quantifiable measure of the current accuracy of user's calibration. The five-star ratings and the corresponding messages are coupled in the following manner: Re-Calibrate (*), Poor (**), Moderate (***)*, Good (****), and Excellent (*****). After completing the calibration process, the UI will always show the latest calibration rating in the bottom-part of the track box in EyeTribe UI. The participant can only start the experiment after achieving good/excellent calibration rating. Prior to each experiment, participants were advised to avoid moving their body and head positions during the tests as far as possible. However, users can manage their body position and adjust their head position if needed easily after few minutes of using the system. No pre-training session was performed for the predefined sentence, as a goal of this study is to determine the performance of beginner users.

There were four different combinations of the input modalities i.e., a mouse, a touch screen, an eye-tracker, a soft-switch, and a Myo armband which provided twenty different conditions of experimental design. The working functionalists of input modalities are explained in the proposed method section. First, the user's eyes without eye-tracking and a regular computer mouse were used for search and selection purpose (see Fig. 2a). Second, the user's eyes without eye-tracking and the touch screen were used (see Fig. 2b). Third, the eye-tracker along with the soft-switch were used in a hybrid mode (see Fig. 2c). Fourth, the eye-tracker was used in combination with five different sEMG-based hand gestures (see Fig. 2d). This combination of input modalities covered five different experimental conditions. Fifth, the eye-tracker was used for both search and selection purposes in synchronous and asynchronous modes (see Fig. 1a, b). We implemented asynchronous and synchronous modes with five different dwell time and trial period values, respectively, resulting in ten different experimental conditions. In addition, there were two more experimental conditions, which incorporated asynchronous and synchronous modes with adaptive dwell time and adaptive interval time, respectively.

The sequence of the experimental conditions was randomized for each participant. The total duration of the experiment was about 3–4 h, making the task difficult and tedious for the participants. Therefore, we organized the experimental conditions and the 24 participants into separate groups.

The twenty different conditions of experimental design were divided into four experiments to evaluate the performance of virtual keyboard across the input modalities.

5.4.1 Experiment 1: mouse versus touch screen

This experiment corresponds to the comparison between the mouse and the touch screen to find and select the characters. With the mouse, the user must click on the target item, whereas the user must touch on the target item with the touch screen only. The mouse only condition was incorporated to find out the performance with GUI without a touch screen.

5.4.2 Experiment 2: eye-tracker with soft-switch versus eye-tracker with sEMG based hand gestures

This experiment was conducted under six different conditions: soft-switch and five sEMG based hand gestures (i.e., fist, wave left, wave right, fingers spread, and double tap) along with eye-tracker (see Fig. 3). These five different hand gestures conditions were included to validate the usability of all available hand gestures of Myo Gesture Control Armband device with VK application to select the items. In these experiments, the eye-tracker was used in a hybrid mode, where the user should gaze at the target item, and the selection happens via switch/-sEMG signals. During the experiments, the participants use these input modalities once they received the visual feedback (i.e., the color of the gazed item begins to change).

5.4.3 Experiment 3: fixed versus adaptive dwell time with eye-tracker asynchronous mode

In this experiment, only the eye-tracker in an asynchronous mode was used by the participants under six different conditions (i.e., Dwell time = 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and adaptive dwell time), where the item is determined through gazing, and the item selection is made by dwell time/adaptive dwell time. These different conditions were included to find out the optimal dwell time. These predefined five dwell time conditions were chosen as the initial threshold for dwell time is set to 2 s. Therefore, we have considered dwell time values with upper bound (2.5 s, 3 s) and lower bound (1.5 s, 1 s).

5.4.4 Experiment 4: fixed versus adaptive trial period with eye-tracker synchronous mode

In this experiment, only the eye-tracker in a synchronous mode was used by the participants for pointing and the selection of items, where pointing to the items is achieved through gaze fixation and the selection is enabled by one of the five different trial periods (i.e., 1 s, 1.5 s, 2 s, 2.5 s, or 3 s) or with an adaptive trial period. These different trial periods were

considered to find out the optimal trial period. To the best of our knowledge, no adaptive method is currently available for gaze-based interaction in synchronous mode. These pre-defined five trial period conditions were chosen as the initial threshold for the trial period was set to 2 s. Therefore, similarly to the asynchronous mode, we have considered trial period values with upper bound (2.5 s, 3 s) and lower bound (1.5 s, 1 s).

5.5 Performance evaluation

Several performance indexes such as text entry rate (the number of letters spelled out per minute, without any error in the desired text), the information transfer rate (ITR) at the basic letter level ITR_{letter} and command level ITR_{com} [43], and the mean and standard deviation (mean \pm SD) of the time to produce a command were used to evaluate the performance of the virtual keyboard in different conditions. The ITR at the letter level is called the ITR_{letter} because it is based on the produced letters on the screen, and at the command level it is called the ITR_{com} because it is based on the produced commands in the GUI. In our case, the number of possible commands is 10 ($M_{com} = 10$), these commands correspond to selected item through eye-tracker. The number of commands at the letter level is 88 ($M_{letter} = 88$), which includes the Hindi letters, matras (i.e., diacritics), halants (i.e., killer strokes), basic punctuation, and space button. The delete, clear-all, and go-back buttons were used as a special command to correct the errors. The ITR is calculated based on the total number of actions (i.e., basic commands and letters) and the duration that is required to perform these commands. To define the ITR, all these different commands and letters were assumed as equally probable and without misspelling. The ITR is defined as follows:

$$ITR_{com} = \log_2(M_{com}) \cdot \frac{N_{com}}{T} \quad (4)$$

$$ITR_{letter} = \log_2(M_{letter}) \cdot \frac{N_{letter}}{T} \quad (5)$$

where N_{com} is the total number of commands produced by the user to type N_{letter} characters. T is the total time to produce N_{com} or type all N_{letter} .

6 Results

The overall performance evaluation of the virtual keyboard was undertaken based on the results collected from a typing experiment. The corrected error rate was measured for each condition without considering the special commands as an error. The corrected errors are errors that are committed but then corrected during text entry [24]. The different

Table 3 Typing performance (mean and standard deviation (SD) across participants) for the mouse and the touch screen alone in experiment 1

Text entry rate (letters/min)	<i>ITR</i> _{com} (bits/min)	<i>ITR</i> _{letter} (bits/min)	Average time (ms)										
			c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	All
Mouse													
Mean	15.68	105.60	101.28	1485	2239	2665	2277	3240	1968	1983	1732	2042	1647
Std.	5.79	37.01	37.39	421	1299	1307	881	1879	1020	961	486	939	439
Touch screen													
Mean	18.00	122.67	116.26	1403	2128	2152	1887	2652	1349	1745	1796	1922	1376
Std.	6.89	45.24	44.52	327	1207	1133	987	2461	630	715	541	812	432
Bold values indicate the best results among the other conditions within a table													

experimental conditions were categorized into four experiments. For computing statistical significance, the Wilcoxon signed-rank test was applied using false discovery rate (FDR) correction method for multiple comparisons on performance indexes across the conditions in each experiment. A Friedman test was conducted to see whether the method was significant for the dependent variable. Furthermore, Wilcoxon rank sum test and two-sample *t*-test were conducted to compare the different groups' performances.

6.1 Experiment 1: mouse versus touch screen

The typing performance for both mouse and touch screen conditions are presented in Table 3. The average text entry rate with touch screen (18.00 ± 6.8 letters/min) is significantly higher ($p < 0.05$) than the mouse (15.68 ± 5.79 letters/min). The best performance was achieved by the participant A09 (30.06 letters/min). A similar pattern of performance is measured in terms of *ITR*_{com} and *ITR*_{letter} for each condition. The *ITR*_{com} and *ITR*_{letter} with touch screen (122.67 ± 45.24 bits/min) and (116.26 ± 44.52 bits/min) were greater than the mouse (105.60 ± 37.01 bits/min and 101.28 ± 37.39 bits/min) ($p < 0.05$), respectively. The average corrected error rate for mouse and touch screen conditions was 0.42% and 0.65%, respectively.

6.2 Experiment 2: eye-tracker with soft-switch versus eye-tracker with sEMG based hand gestures

The eye-tracker was used under six different input conditions. The average typing performance is shown in Table 4 across the conditions. The text entry rate, *ITR*_{com}, and *ITR*_{letter} with soft-switch were found 21.83 ± 6.58 letters/min, 144.00 ± 45.89 bits/min, and 141.01 ± 42.48 bits/min, respectively. For text entry rate, a Friedman test of differences among repeated measures (six different input conditions) confirmed that there is a significant effect of the type of soft-switch in this experiment ($\chi^2 = 20.72$, $p < 10e-3$). The performance

with the soft-switch in terms of text entry rate and *ITR* was found superior to all other conditions ($p < 0.05$, FDR corrected). However, when the eye-tracker was used in a hybrid mode with the five hand gestures and the best text entry rates, *ITR*_{com}, and *ITR*_{letter} were achieved by the wave right (16.17 ± 5.39 letters/min), (96.13 ± 31.03 bits/min), and (89.41 ± 27.74 bits/min), respectively. With hand gestures, we found that the wave right leads to significantly superior performance in terms of text entry rate and *ITR* compared to the fist ($p < 0.05$, FDR corrected). The average corrected error rate for soft-switch, fist, wave left, wave right, fingers spread, and double tap conditions was 1.31%, 2.30%, 3.28%, 1.97%, 3.15%, and 2.63%, respectively.

6.3 Experiment 3: fixed versus adaptive dwell time with eye-tracker asynchronous mode

The eye-tracker was used in an asynchronous mode to perform the typing task. The average typing performance is shown in Table 5. For text entry rate, a Friedman test of differences among repeated measures (six different conditions (5 with fixed and 1 with adaptive dwell time)) revealed a significant effect of the dwell time ($\chi^2 = 48.91$, $p < 10e-6$). The text entry rate, *ITR*_{com}, and *ITR*_{letter} with 1 s dwell time condition were found 13.41 ± 5.21 letters/min, 101.82 ± 15.26 bits/min, and 90.53 ± 18.66 bits/min, respectively. This condition provides highest performance of all the other four conditions. However, using 1 s dwell time condition, the participant B06 was unable to complete the task as it requires fast eye movements. The text entry rate with 1.5 s dwell time condition (11.32 ± 1.84 letters/min) was higher than that with 2 s (8.30 ± 1.75 letters/min), 2.5 s (7.67 ± 1.45 letters/min), and 3 s (6.44 ± 1.21 letters/min) dwell time conditions ($p < 0.05$, FDR corrected).

The dwell time adaptive algorithm was explored to improve the text entry rate and accuracy of the system. The initial value for Δt_0 was set to 2 s. The text entry rate with the adaptive asynchronous condition (16.10 ± 3.36 letters/min) was found greatest of all the dwell time conditions. Subse-

Table 4 Typing performance (mean and standard deviation (SD) across participants) for the soft-switch and each hand gesture: fist, waveLeft, waveRight, fingers spread, and double tap with eye-tracker in experiment 2

Gesture	Text entry rate (letters/min)	<i>ITR_{com}</i> (bits/min)	<i>ITR_{letter}</i> (bits/min)	Average time (ms)										
				c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	All
Soft-switch														
Mean	21.83	144.00	141.01	1236	1376	1395	1342	1490	1068	1422	1293	1330	1496	1347 ± 859
Std.	6.58	45.89	42.48	284	304	465	549	498	423	626	368	449	533	357 ± 376
Fist														
Mean	13.61	91.97	87.91	1878	2417	2694	2523	2346	2175	2396	2064	2151	2304	2362 ± 1989
Std.	5.45	35.95	35.21	887	1018	1424	1406	904	1138	1320	971	879	942	1046 ± 1372
Wave left														
Mean	13.84	96.13	89.41	1717	2985	2289	2171	2245	1858	1965	2188	2079	2191	2166 ± 1710
Std.	4.29	31.03	27.74	617	1978	1183	878	939	1024	762	1046	676	902	818 ± 1169
Wave right														
Mean	16.17	110.33	104.45	2006	1741	1853	1939	1991	1709	1825	2062	1841	1997	1911 ± 1097
Std.	5.39	40.33	34.80	949	642	662	935	773	690	559	1078	796	933	727 ± 604
Fingers spread														
Mean	15.51	109.95	100.16	1815	2566	2306	1891	2595	1990	2159	1681	2325	1918	2189 ± 1777
Std.	7.07	50.51	45.64	1302	1752	1799	908	1992	1068	891	540	1393	1009	1215 ± 1753
Double tap														
Mean	10.25	73.40	66.18	2462	2470	2482	2509	2390	2078	2669	2413	2754	2823	2538 ± 1494
Std.	2.92	22.63	18.83	958	518	498	823	554	597	999	607	656	671	533 ± 840

Bold values indicate the best results among the other conditions within a table

Table 5 Typing performance (mean and standard deviation (SD) across participants) for each dwell time (DT): 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and adaptive DT with eye-tracker asynchronous mode in experiment 3

DT	Text entry rate (letters/min)	<i>ITR_{com}</i> (bits/min)	<i>ITR_{letter}</i> (bits/min)	Average time (ms)										
				c1	c2	c3	c4	c5	c6	c7	c8	c9	c10	All
1 s														
Mean	13.41	101.82	90.53	1800	1810	2159	1875	1872	1792	1756	1812	1979	1830	1876 ± 788
Std.	5.21	15.26	18.66	339	319	625	357	357	336	387	384	429	325	280 ± 266
1.5 s														
Mean	11.32	78.88	70.85	2291	2344	2599	2496	2510	2312	2364	2490	2576	2632	2483 ± 935
Std.	1.84	8.68	11.04	311	319	394	364	412	414	324	497	436	366	287 ± 330
2 s														
Mean	8.30	58.00	50.29	2946	3173	3478	3302	3518	3420	3062	3336	3400	3359	3312 ± 1329
Std.	1.75	7.64	8.92	373	590	812	530	902	901	676	777	647	758	565 ± 595
2.5 s														
Mean	7.67	50.80	47.76	3668	3624	3873	4007	4212	3603	3895	3891	3909	3857	3855 ± 1336
Std.	1.45	7.40	9.42	609	392	954	711	904	662	1026	702	802	712	575 ± 522
3 s														
Mean	6.44	43.97	40.35	4215	4182	4613	4579	4470	4476	4546	4259	4869	4620	4490 ± 1570
Std.	1.21	6.13	7.35	718	552	768	649	807	972	1115	615	1132	1077	671 ± 878
Adaptive														
Mean	16.10	105.19	98.05	1675	1949	2090	1913	1979	1668	1846	1731	2001	1607	1846 ± 683
Std.	3.36	17.00	19.09	329	455	536	440	621	337	382	336	501	238	359 ± 273

Bold values indicate the best results among the other conditions within a table

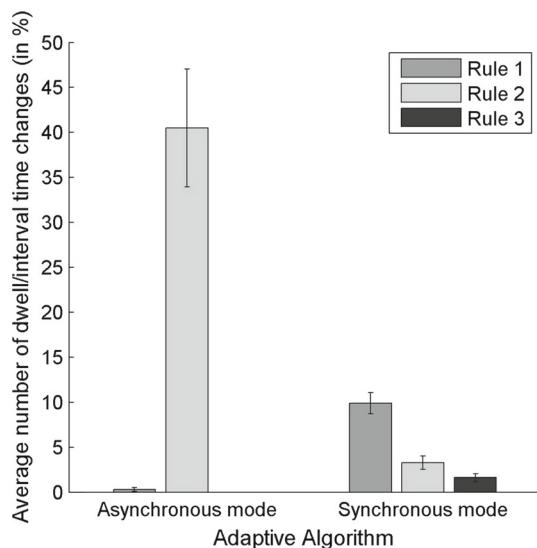


Fig. 7 The average dwell time in asynchronous mode and trial period in synchronous mode changes (in%) across rules (2 rules in asynchronous mode and 3 rules in synchronous mode) of adaptive time parameters algorithm. The error bars represent standard errors across trials

quently, we found that the adaptive asynchronous condition leads to a better performance in terms of text entry rate and ITR than any of the other five dwell time conditions ($p < 0.05$, FDR corrected). Figure 7 depicts the dwell time changes in percentage across group B for the two rules of adaptive

dwell time algorithm. Rule #2 of decreasing dwell time ($40.5 \pm 20.73\%$) was used more often than Rule #1 of increasing dwell time ($0.3 \pm 0.67\%$). It shows that Rule #2 was used the maximum number of times by the participants in order to achieve higher performance ($p < 0.05$). In particular, the text entry rate of 20.20 letters/min was achieved by the participant B10 wherein Rule #2 is used about 70% of the times. The average corrected error rate for fixed dwell time of 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and adaptive dwell time was 3.05%, 2.84%, 1.31%, 0.65%, 0.42%, and 1.07%, respectively.

6.4 Experiment 4: fixed versus adaptive trial period with eye-tracker in synchronous mode

The eye-tracker was used in the synchronous mode that included five conditions of trial periods and one condition with adaptive trial period algorithm. The average typing performance is shown in Table 6. For text entry rate, a Friedman test of differences among repeated measures (six different conditions (5 with fixed and 1 with adaptive trial period)) confirmed that there is a significant effect of the trial duration ($\chi^2 = 45.81, p < 10e-6$). The text entry rate, ITR_{com} , and ITR_{letter} with 1.5 s trial period condition were found 14.89 ± 2.17 letters/min, 123.89 ± 4.49 bits/min, and 91.36 ± 14.27 bits/min, respectively. The text entry rate and ITR with 1.5 s trial period condition were found higher than

Table 6 Typing performance (mean and standard deviation (SD) across participants) for each trial period (TP): 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and adaptive TP with eye-tracker synchronous mode in experiment 4

TP	Text entry rate (letters/min)	ITR_{com} (bits/min)	ITR_{letter} (bits/min)	Average time (ms)									
				c1	c2	c3	c4	c5	c6	c7	c8	c9	c10
1 s													
Mean	11.46	182.34	94.54	–	–	–	–	–	–	–	–	–	–
Std.	6.79	9.21	37.88	–	–	–	–	–	–	–	–	–	–
1.5 s													
Mean	14.89	123.89	91.36	–	–	–	–	–	–	–	–	–	–
Std.	2.17	4.49	14.27	–	–	–	–	–	–	–	–	–	–
2 s													
Mean	11.65	93.20	71.57	–	–	–	–	–	–	–	–	–	–
Std.	1.91	2.15	12.17	–	–	–	–	–	–	–	–	–	–
2.5 s													
Mean	9.32	75.20	57.46	–	–	–	–	–	–	–	–	–	–
Std.	1.64	2.14	9.39	–	–	–	–	–	–	–	–	–	–
3 s													
Mean	8.45	63.01	52.47	–	–	–	–	–	–	–	–	–	–
Std.	0.87	0.62	5.34	–	–	–	–	–	–	–	–	–	–
Adaptive													
Mean	17.06	145.48	107.36	1374	1643	1446	1395	1380	1348	1413	1271	1377	1210
Std.	3.06	19.71	18.78	278	236	277	247	255	323	249	299	317	167

Bold values indicate the best results among the other conditions within a table

the all the other trial period conditions ($p < 0.05$, FDR corrected). However, the participant B03 achieved highest text entry rate of 25.27 letters/min with 1 s trial period condition but two participants (i.e., B08, B10) were unable to complete the task as it required higher attention and faster eye movement for selection of the items.

The text entry rate, ITR_{com} , and ITR_{letter} with adaptive trial period condition were computed 17.06 ± 3.06 letters/min, 145.48 ± 19.71 bits/min, and 107.36 ± 18.78 bits/min, respectively. The initial value for Δt_1 is set to 2 s. It has been shown that the adaptive trial period algorithm provides the best performance ($p < 0.05$, FDR corrected) in experiment 4. The Fig. 7 represents the average trial period changes (in %) across the rules of adaptive trial period algorithm in the synchronous mode. It has been found that Rule #1 of decreasing trial period ($9.9 \pm 3.67\%$) (mean \pm SD) was used more often than Rule #2 of increasing trial period ($3.3 \pm 2.36\%$) and Rule #3 of increasing dwell interval ($1.6 \pm 1.36\%$) ($p < 0.05$, FDR corrected). It shows that Rule #1 was used the maximum number of times by the participants in order to achieve higher performance. The average corrected error rate for fixed trial period of 1 s, 1.5 s, 2 s, 2.5 s, 3 s, and adaptive trial period was 9.20%, 5.13%, 3.10%, 2.61%, 1.31%, and 2.91%, respectively.

6.5 Time-adaptive synchronous versus asynchronous mode

The average text entry rate, ITR_{com} , and ITR_{letter} with time-adaptive algorithm in synchronous mode were calculated as: 17.06 ± 3.06 letters/min, 145.48 ± 19.71 bits/min, and 107.36 ± 18.78 bits/min, respectively, whereas the average text entry rate, ITR_{com} , and ITR_{letter} with time-adaptive algorithm in asynchronous mode were found to be 16.10 ± 3.36 letters/min, 105.19 ± 17.00 bits/min, and 98.05 ± 19.09 bits/min, respectively. The adaptive synchronous mode leads to a greater ITR_{com} than the adaptive asynchronous ($p < 0.05$). However, no significant difference was found for the text entry rate and ITR_{letter} between the two conditions.

6.6 Dwell-free versus Time-adaptive modes

The touch-screen and eye-tracking with soft-switch methods/modalities of dwell-free provide the higher average text entry rate, ITR_{com} , and ITR_{letter} in experiment 1 and experiment 2, respectively within group A participants. Similarly, the time-adaptive methods of asynchronous and synchronous mode produce the best typing performance in experiment 3 and experiment 4, respectively within group B participants. As these two groups of participants are independent (same type of participants in terms of age, gender, and education), we have compared paired group performance of touch-screen method of experiment 1 with the time-adaptive

asynchronous method of experiment 3 and time-adaptive synchronous method of experiment 4. Likewise, we have compared paired group performance of eye-tracking with the soft-switch method of experiment 2 with the time-adaptive asynchronous method of experiment 3 and time-adaptive synchronous method of experiment 4. No significant difference in performance in terms of typing speed was found between methods.

7 Subjective evaluation

7.1 NASA task load index

NASA Task Load Index (NASA-TLX) is a widely used, subjective, multidimensional assessment tool that rates perceived workload in order to assess the effectiveness and/or other aspects of performance of a task, system, or team. It is a well-established method for analyzing user's workload [64, 65]. Final scores for the NASA-TLX ranges from 0 to 100, where a low score indicates a better performance. The workload experienced by the users during the interaction with the virtual keyboard application was measured using this index, wherein mental demand, physical demand, temporal demand, performance, effort, and frustration aspects were included.

Separate NASA-TLX tests were conducted with each group of participants. First, the NASA-TLX test was evaluated with group A (17.08 ± 3.05) for experiments 1 and 2. Second, the NASA-TLX test was evaluated with group B (17.45 ± 4.45) for experiments 3 and 4. The average score for each item across two groups of participants is depicted in Fig. 8. The system achieved the average NASA-TLX score below 18% with both groups, showing a low workload (see in Fig. 9) [64].

7.2 System usability scale

The system usability scale (SUS) is a ten-item attitude Likert-type scale giving a global view of subjective assessments of usability [66]. It is composed of 10 items that are scored on a 5-point scale of the strength of agreement. Each item score ranges from 0 to 4. Final scores for the SUS ranges from 0 to 100, where a high score indicates better usability. The usability of a system can be measured by taking into account the context of use of the system (e.g., who is using the system, what they are using it for, and the environment in which they are using it). Therefore, this scale is used to evaluate a system based on three major aspects of the usability: effectiveness, efficiency, and satisfaction. This scale was used to determine the level of usability, and to receive a feedback from the participants to transfer the system into an effective and commercial augmentative and alternative communica-

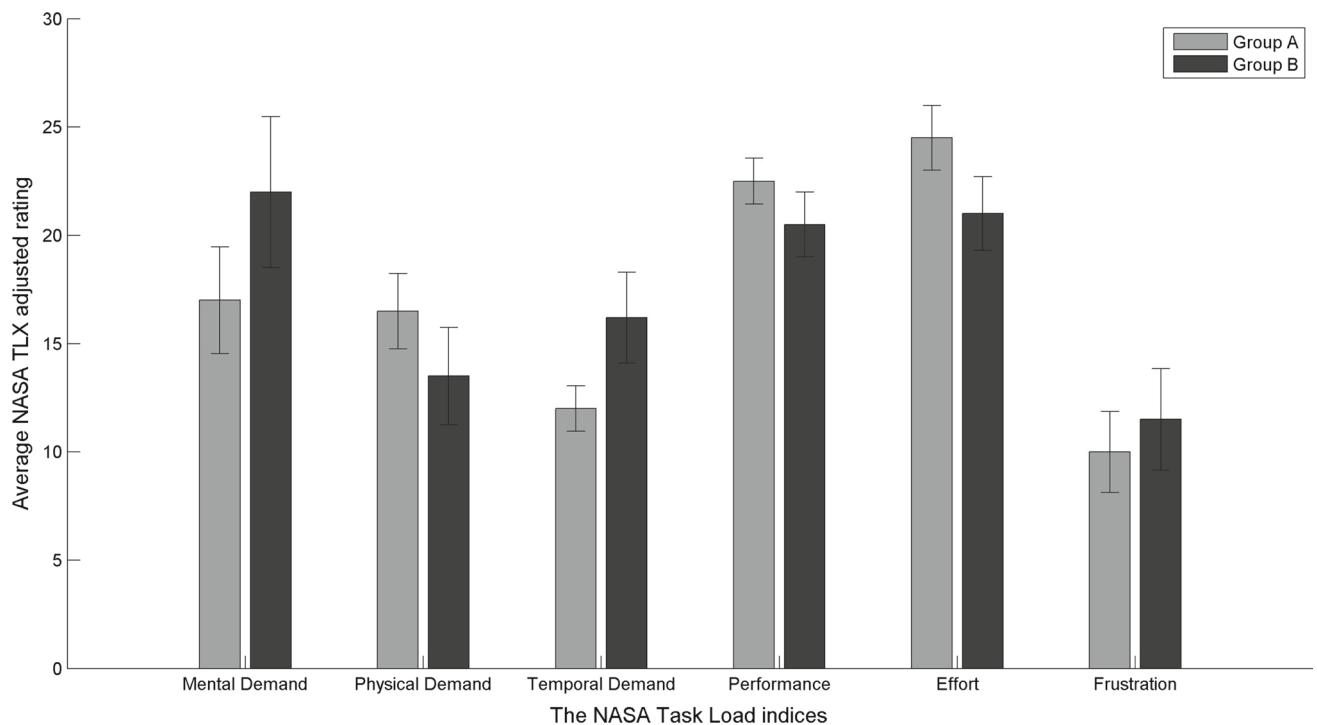


Fig. 8 The global view of subjective assessments of workload: The average NASA TLX adjusted rating score across a group of participants. The error bars represent standard errors across participants

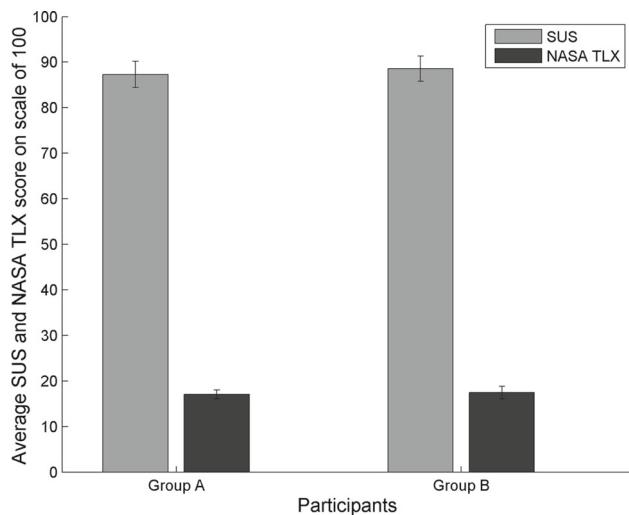


Fig. 9 The average system usability scale (SUS) and NASA work load index (NASA-TLX) score across a group of participants. The error bars represent standard errors across participants. A higher SUS score indicates better usability whereas a low NASA-TLX score indicates a better performance

tion (AAC) device. One SUS test was conducted with each group of participants. First, the SUS test was evaluated with group A (87.29 ± 9.07) for experiments 1 and 2. Second, the SUS test was evaluated with group B (88.54 ± 8.69) for experiments 3 and 4. The system was validated by SUS score, and achieved an average SUS score above 87% with

both groups, indicating an excellent grade on the adjective rating scale (see in Fig. 9) [67].

8 Discussion

This study includes comprehensive and multiple levels of comparisons to better appreciate the performance of the proposed approaches of beginner users. The proposed time-adaptive methods provide higher average text entry rate in both synchronous (17.06 ± 3.06 letters/min) and asynchronous (16.10 ± 3.36 letters/min) modes with new users. Furthermore, the multimodal dwell-free mechanism using a combination of eye-tracking and soft-switch (21.83 ± 6.58 letters/min) provides better performance than eye-tracker with sEMG based hand gestures and adaptive methods with eye-tracking only. The methods related to the adaptation of the system over time that are proposed in this paper, were applied to a gaze-based virtual keyboard, which can be operated using a portable non-invasive eye-tracker, sEMG based hand gesture recognition device, and/or a soft-switch. This study focuses on users' initial adaptation of a new system, instead of learning over a longer timescale. The proposed algorithms suggest the beneficial impact of an adaptive approach in both synchronous and asynchronous modes, which needs to be confirmed over long sessions while performance is typically expected to improve over time [68].

It is known that use cases can vary a lot across participants [52]. For instance, some users may have some disabilities or other issues related to attention that can prevent them from using the system for prolonged durations. For this reason, the parameters of the system must evolve over time to match the current performance of the user. Multimodal interfaces should adapt to the needs, abilities of different users, and different contexts of use [69]. The proposed system provides a single GUI that offers different modalities, which can be selected in relation to the preference of the user. The mode of action using the eye-tracker (synchronous or asynchronous) can be selected in relation to the frequency of use. On the one hand, the synchronous mode can be a relevant choice if the user is focused and desires to write text during a long session. On the other hand, if the user alternates between the typing task and other side tasks, then the asynchronous mode will be a more relevant choice as the system will be self-paced.

This study has four main outcomes. First, we proposed a set of methods for both adaptive synchronous and asynchronous modes to improve the text entry rate and detection accuracy. Second, we presented a benchmark of several dwell-free mechanisms with a novel robust virtual keyboard for a complex structured language (the Hindi language) that can make use of the mouse, touch screen, eye-gaze detection, gesture recognition, and a single input switch, either alone as a single modality, or in combination as a multimodal device. Third, we evaluated the performance of the virtual keyboard in 20 different conditions to assess the effect of different types of input controls on the system performance (e.g., text entry rate). Fourth, we demonstrated an excellent grade usability of the system based on the SUS questionnaires and low workload of the system based on the NASA TLX scale.

The GUI was implemented to build a complete and robust solution on top of previous pilot study [43] with an increased number of commands to include 88 characters along with half letter, go-back, and delete facility to correct errors. In addition, the system incorporated time-adaptive methods and more input modalities such as a touch screen and gesture recognition wherein users can employ any of them according to their comfort and/or need. In general, the performance of virtual scanning keyboards is evaluated by its text entry rate and accuracy [2,43,70]. While a set of rules have been proposed for both synchronous and asynchronous modes, a set of thresholds were empirically chosen to validate the method. The maximum and minimum values for the thresholds and the steps that were set could be determined via additional experiments to determine the extent to which these values could be determined as well. The addition of other inputs related to the cognitive state of the user may provide additional information about the choice of the values for the parameters of the system.

The proposed virtual keyboard provided an average text entry rate of 22 letters/min with the use of eye-tracking and a soft-switch. Although a variation in performance was expected across conditions, the average performance with the use of only eye-tracking in a synchronous and asynchronous mode with a set of rules still remains high enough (i.e., 17 letters/min) to be used efficiently. The major confounding factor to achieve high accuracy and text entry rate in an eye-tracker based system is the number of commands, which is further constrained by the quality of calibration method. We have therefore taken into account the size of the command boxes and the distance between them for increasing the robustness of the system to involuntary head and body movements. Furthermore, the calibration issue of gaze tracking could be handled by implementing an additional threshold adjustment if the calibration problem happens multiple times. It is worth noting that the proposed adaptive methods are script independent and can be applied to other scripts (e.g., the Latin script). The proposed system can be directly used for the Marathi/Konkani language users (70 million speakers) by including one additional letter (i.e., ऋ). Therefore, the present research findings have potential application for a large user population (560 million).

The performance evaluation of a virtual keyboard depends on several factors such as the nature of the typing task, its length, the type of users, and their experience and motivation during the typing task. On the one hand, for effectively accounting for all these factors, it becomes challenging to evaluate the performance of a virtual keyboard. Moreover, typing rate is affected by the word completion and word prediction methods [71]. On the other hand, the concept of AugKey is to improve throughput by augmenting keys with a prefix, to allow continuous text inspection, and suffixes to speed up typing with word prediction [72]. Thus, to avoid performance variations, we 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. As this virtual keyboard provided a high text entry rate of 18 letters/min with a touch screen, it can be employed as an AAC system with or without eye-tracking for physically disabled people to interact with currently available personal information technology (IT) systems.

In terms of performance comparison, virtual keyboards based on brain activity detection, such as the P300 and SSVEP speller, offer significantly lower performance than the proposed system. Studies reported an average ITR of 25 bits/min with P300 speller [73] and 37.62 bits/min (average text entry rate of 5.51 letters/min) with SSVEP speller [74]. In addition, an EOG based typing system and an eye-tracker based virtual keyboard system reported average text entry rate of 15 letters/min [70], 9.3 letters/min [2], and 11.39 letters/min [75] respectively. Thus, the proposed system outperforms these solutions with an average ITR and average

text entry rate of 145.48 bits/min and 17 letters/min, respectively. Finally, the system achieved an excellent grade on the adjective rating scale to the SUS (87%) and low workload (NASA TLX with 17 scores). Despite good performance obtained with 24 healthy participants, the system should be further evaluated with speech and motor impaired people, wherein target selection can be performed with other modalities (e.g., brain-wave responses) [44,46,76,77].

While the present study was evaluated with healthy people, the end user targets include people with severe disabilities who are unable to write messages with a regular interface. As the goal was to assess the improvement that can be obtained with an adaptive system in synchronous or asynchronous mode, the degree of physical disability was not relevant for the evaluation of the algorithms but it may have an impact on the usability and workload evaluation. However, the usability and workload tests provided excellent results, showing that people with no physical impairment were still able to appreciate the value of the system. Furthermore, the system evaluation for a particular type of disability is limited by the number of available participants with this disability. Within the context of rehabilitation, a patient may start with a particular mode of control and modality, and this user may recover over time and change his/her favorite type of control and modality, while keeping the same GUI throughout the rehabilitation period. The proposed system may therefore allow a smooth transition between different modes of control and modalities for a patient throughout the rehabilitation stages.

9 Conclusion

This paper presented an efficient set of methods and rules for the adaptation over time of gaze-controlled multimodal virtual keyboards in synchronous and asynchronous modes. We demonstrated the effectiveness of the proposed methods with the Hindi language, which is a language with complex structure. However, these results are preliminary with beginner users, and show the potential of the proposed methods during their first encounter with the system. Despite the above facts, the adaptive approaches outperform non-adaptive methods, and we presented a benchmark of several dwell-free mechanisms of beginner users. Future longitudinal studies should confirm the advantages of the adaptive methods on the fixed dwell times. Future works will include the system evaluation with more complex sentences, with an improved GUI design, and with the participation of users with disabilities.

Kenney, EJ (1975) Ovid, Metamorphoses-Ovid: Metamorphoses, Book xi. Edited with an Introduction and Commentary by Murphy GMH. Pp.[vi]+ 137. London: Oxford University Press, 1972. Paper, €1–50 net. The Classical Review, 25(1), 35–36

Acknowledgements Y.K.M. was supported by Govt. of India (Education-11016152013). G.P., K.W.-L., and H.C. were supported by the Northern Ireland Functional Brain Mapping Facility (1303/101154803). G.P. was also supported by UKIERI DST Thematic Partnership Project: DST-UKIERI-2016-17-0128.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Wolpaw JR, Birbaumer N, Mcfarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain–computer interfaces for communication and control. *Clin Neurophysiol* 113:767–91
- Cecotti H (2016) A multimodal gaze-controlled virtual keyboard. *IEEE Trans Hum Mach Syst* 46(4):601–606
- Wheeler K R, Chang M H, Knuth K H (2006) Gesture-based control and emg decomposition. *IEEE Trans Syst Man Cybern C Appl Rev* 36(4):503–514
- Bhattacharya S, Basu A, Samana D (2008) Performance models for automatic evaluation of virtual scanning keyboards. *IEEE Trans Neural Syst Rehabil Eng* 16(5):510–519
- MacKenzie IS, Zhang X (2008) Eye typing using word and letter prediction and a fixation algorithm. In: Proceedings of the 2008 symposium on eye tracking research and applications, pp 55–58
- Zhu Z, Ji O (2004) Eye and gaze tracking for interactive graphic display. *Mach Vis Appl* 15(3):139–148
- Cutrell E, Guan Z, Cutrell E (2007) What are you looking for? an eye-tracking study of information usage in web search. In: CHI ’07: proceedings of the SIGCHI conference on human factors in computing systems, pp 407–416
- Pan B, Hembrooke HA, Gay GK, Granka LA, Feusner MK, Newman JK (2004) The determinants of web page viewing behavior: an eye-tracking study. In: Proceedings of the 2004 symposium on eye tracking research and applications, pp 147–154
- Meena YK, Chowdhury A, Cecotti H, Wong-Lin K, Nishad SS, Dutta A, Prasad G (2016) Emohex: an eye tracker based mobility and hand exoskeleton device for assisting disabled people. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC). IEEE, pp 002122–002127
- Lee EC, Park KR (2007) A study on eye gaze estimation method based on cornea model of human eye. Springer, Cham, pp 307–317
- Jacob Robert JK (1995) Eye tracking in advanced interface design. In: Virtual environments and advanced interface design, pp 258–288
- Katarzyna H, Pawel K, Mateusz S (2014) Towards accurate eye tracker calibration methods and procedures. *Procedia Comput Sci* 35:1073–1081
- Nicolas-Alonso LF, Gomez-Gil J (2012) Brain computer interfaces, a review. *Sensors* 12(2):1211–1279
- Huckauf A, Urbina MH (2011) Object selection in gaze controlled systems: what you don’t look at is what you get. *ACM Trans Appl Percept* 8(2):13:1–13:14
- Kenney, EJ (1975) Ovid, Metamorphoses-Ovid: Metamorphoses, Book xi. Edited with an Introduction and Commentary by Murphy GMH. Pp.[vi]+ 137. London: Oxford University Press, 1972. Paper, €1–50 net. The Classical Review, 25(1), 35–36
- Jacob RJK, Karn KS (2003) Eye tracking in human–computer interaction and usability research: ready to deliver the promises. *Mind* 2(3):4

17. Jacob RJK (1990) What you look at is what you get: eye movement-based interaction techniques. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 11–18
18. Meena YK, Cecotti H, Wong-lin K, Prasad G (2017) A multimodal interface to resolve the midas-touch problem in gaze controlled wheelchair. In: Proceedings of the IEEE engineering in medicine and biology, pp 905–908
19. Majaranta P, MacKenzie IS, Aula A, Raiha KJ (2006) Effects of feedback and dwell time on eye typing speed and accuracy. *Univ Access Inf Soc* 5:119–208
20. Räihä KJ, Ovaska S (2012) An exploratory study of eye typing fundamentals: dwell time, text entry rate, errors, and workload. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 3001–3010
21. Špakov O, Miniotas D (2004) On-line adjustment of dwell time for target selection by gaze. In: Proceedings of the 3rd Nordic conference on human-computer interaction. ACM pp 203–206
22. Majaranta P, Aula A, Spakov O (2009) Fast gaze typing with an adjustable dwell time. In: Proceedings of the CHI, pp 1–4
23. Pi J, Shi BE (2017) Probabilistic adjustment of dwell time for eye typing. In: 10th international conference on Human system interactions (HSI), 2017. IEEE, pp 251–257
24. Mott ME, Williams S, Wobbrock JO, Morris MR (2017) Improving dwell-based gaze typing with dynamic, cascading dwell times. In: Proceedings of the 2017 CHI conference on human factors in computing systems. ACM, pp 2558–2570
25. Chakraborty T, Sarcar S, Samanta D (2014) Design and evaluation of a dwell-free eye typing technique. In: Proceedings of the extended abstracts of the 32nd annual acm conference on human factors in computing systems, pp 1573–1578
26. Kristensson PO, Vertanen K (2012) The potential of dwell-free eye-typing for fast assistive gaze communication. In: Proceedings of the symposium on eye tracking research and applications, pp 241–244
27. Nayyar A, Dwivedi U, Ahuja K, Rajput N, Nagar S, Dey K (2017) Optidwell: intelligent adjustment of dwell click time. In: Proceedings of the 22nd international conference on intelligent user interfaces, pp 193–204
28. Gomide RDS et al (2016) A new concept of assistive virtual keyboards based on a systematic review of text entry optimization techniques. *Res Biomed Eng* 32(2):176–198
29. Wobbrock JO, Rubinstein J, Sawyer MW, Duchowski AT (2008) Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In: Proceedings of the 2008 symposium on eye tracking research & applications. ACM, pp 11–18
30. Ward DJ, MacKay DJC (2002) Artificial intelligence: fast hands-free writing by gaze direction. *Nature* 418(6900):838
31. Panwar P, Sarcar S, Samanta D (2012) Eyeboard: a fast and accurate eye gaze-based text entry system. In: 4th international conference on intelligent human computer interaction (IHCI), 2012. IEEE, pp 1–8
32. Sarcar S, Panwar P (2013) Eyeboard++: an enhanced eye gaze-based text entry system in hindi. In: Proceedings of the 11th Asia Pacific conference on computer human interaction. ACM, pp 354–363
33. Kumar M, Paepcke A, Winograd T (2007) Eyepoint: practical pointing and selection using gaze and keyboard. In: Proceedings of the SIGCHI conference on human factors in computing systems. ACM, pp 421–430
34. Kurauchi A, Feng W, Joshi A, Morimoto C, Betke M (2016) Eyeswipe: dwell-free text entry using gaze paths. In: Proceedings of the 2016 CHI conference on human factors in computing systems. ACM, pp 1952–1956
35. Pedrosa D, Pimentel MDG, Wright A, Truong KN (2015) Filteryedping: design challenges and user performance of dwell-free eye typing. *ACM Trans Access Comput (TACCESS)* 6(1):3
36. Hansen DW, Skovsgaard HHT, Hansen JP, Møllenbach E (2008) Noise tolerant selection by gaze-controlled pan and zoom in 3D. In: Proceedings of the 2008 symposium on eye tracking research and applications. ACM, pp 205–212
37. Li D, Babcock J, Parkhurst DJ (2006) Openeyes: a low-cost head-mounted eye-tracking solution. In: Proceedings of the 2006 symposium on eye tracking research and applications. ACM, pp 95–100
38. Huckauf A, Urbina MH (2008) Gazing with peyes: towards a universal input for various applications. In: Proceedings of the 2008 symposium on eye tracking research and applications. ACM, pp 51–54
39. Krejcar O (2011) Human computer interface for handicapped people using virtual keyboard by head motion detection. In: Semantic methods for knowledge management and communication. Springer, pp 289–300
40. Lupa RG, Bozomitu RG, Ungureanu F, Cehan V (2011) Eye tracking based communication system for patient with major neuro-locomotor disabilities. In: 15th international conference on system theory, control, and computing (ICSTCC), 2011. IEEE, pp 1–5
41. Samanta D, Sarcar S, Ghosh S (2013) An approach to design virtual keyboards for text composition in indian languages. *Int J Hum Comput Interact* 29(8):516–540
42. Oviatt S, Schuller B, Cohen P, Sonntag D, Potamianos G (2017) The handbook of multimodal-multisensor interfaces: foundations, user modeling, and common modality combinations. Morgan & Claypool, San Rafael
43. Meena YK, Cecotti H, Wong-Lin K, Prasad G (2016) A novel multimodal gaze-controlled hindi virtual keyboard for disabled users. In: Proceedings of IEEE international conference on systems, man, and cybernetics, pp 1–6
44. Meena YK, Cecotti H, Wong-lin K, Prasad G (2015) Towards increasing the number of commands in a hybrid brain–computer interface with combination of gaze and motor imagery. In: Proceedings of the IEEE engineering in medicine and biology, pp 506–509
45. Meena YK, Cecotti H, Wong-Lin K, Prasad G (2015) Powered wheelchair control with a multimodal interface using eye-tracking and soft-switch. In: Proceedings of translational medicine conference p 1
46. Doherty DO, Meena YK, Raza H, Cecotti H, Prasad G (2014) Exploring gaze-motor imagery hybrid brain-computer interface design. In: Proceedings of the IEEE international conference on bioinformatics and biomedicine, pp 335–339
47. Meena YK, Chowdhury A, Sharma U, Cecotti H, Bhushan B, Dutta A, Prasad G (2018) A hindi virtual keyboard interface with multimodal feedback: a case study with a dyslexic child. In: 2018 32nd British human computer interaction conference (BHCI). ACM, pp 1–5
48. Cecotti H, Meena YK, Prasad G (2018) A multimodal virtual keyboard using eye-tracking and hand gesture detection. In: 2018 40th Annual international conference of the IEEE engineering in medicine and biology society (EMBC), pp 3330–3333. IEEE
49. Zhang Q, Imamiya A, Go K, Mao X (2004) Overriding errors in a speech and gaze multimodal architecture. In: Proceedings of the 9th international conference on intelligent user interfaces. ACM, pp 346–348
50. Sharma R, Pavlović VI, Huang TS (2002) Toward multimodal human–computer interface. In: Advances in image processing and understanding: a festschrift for Thomas S Huang. World Scientific, pp 349–365
51. Portela MV, Rozado D (2014) Gaze enhanced speech recognition for truly hands-free and efficient text input during hci. In: Proceedings of the 26th Australian computer–human interaction conference on designing futures: the future of design. ACM, pp 426–429

52. Kar A, Corcoran P (2017) A review and analysis of eye-gaze estimation systems, algorithms and performance evaluation methods in consumer platforms. *IEEE Access* 5:16495–16519
53. Prabhu V, Prasad G (2011) Designing a virtual keyboard with multimodal access for people with disabilities. In: Proceedings of world congress on information and communication technologies (WICT), pp 1133–1138
54. Bhattacharya S, Laha S (2013) Bengali text input interface design for mobile devices. *Univ Access Inf Soc* 12(4):441–451
55. Gaede V, Gnther O (1998) Multidimensional access methods. *ACM Comput Surv* 30(2):170–231
56. Isokoski P (2004) Performance of menu-augmented soft keyboards. In: Proceedings of international ACM conference on human factors in computing systems, pp 423–430
57. Bonner MN, Brudvik JT, Abowd GD, Edwards WK (2010) No-look notes: accessible eyes-free multi-touch text entry. In: Proceedings of the 8th international conference on pervasive computing, pp 409–426
58. Neiberg F, Venolia G (1994) T-cube: a fast, self-disclosing pen-based alphabet. In: Proceedings of the SIGCHI conference on human factors in computing systems, pp 265–270
59. Callahan J, Hopkins D, Weiser M, Shneiderman B (1988) An empirical comparison of pie vs. linear menus. In: Proceedings of the SIGCHI conference on Human factors in computing systems. ACM, pp 95–100
60. Dalmaijer D (2014) Is the low-cost eyetribe eye tracker any good for research?. *PeerJ Preprints*, San Diego, pp 1–35
61. Nymoen K, Haugen MR, Jensenius AR (2015) Mumyo - evaluating and exploring the myo armband for musical interaction. In: Proceedings of internatonal conference on new interfaces for musical expression, pp 506–509
62. Singh JV, Prasad G (2015) Enhancing an eye-tracker based human-computer interface with multi-modal accessibility applied for text entry. *Int J Comput Appl* 130(16):16–22
63. (2015) The eye tribe, copenhagen, denmark. <https://theeyetribe.com/>. Accessed 01 June 2015
64. Whittington P, Dogan H (2017) Smartpowerchair: characterization and usability of a pervasive system of systems. *IEEE Trans Human Mach Syst* 47(4):500–510
65. Räihä KJ, Ovaska S (2012) An exploratory study of eye typing fundamentals: Dwell time, text entry rate, errors, and workload. In: Proceedings of internatinal acm conference on human factors in computing systems, pp 3001–3010
66. Brooke J (1996) Sus: a “quick and dirty” usability scale. Taylor and Francis, London
67. Bangor A, Kortum P, Miller J (2009) Determining what individual sus scores mean: adding an adjective rating scale. *J Usability Stud* 4(3):114–123
68. Tuisku O, Rantanen V, Surakka V (2016) Longitudinal study on text entry by gazing and smiling. In: Proceedings of the 9th biennial ACM symposium on eye tracking research & applications. ACM, pp 253–256
69. Reeves LM et al (2004) Guidelines for multimodal user interface design. *Commun ACM* 47(1):57–59
70. Nathan DS, Vinod AP, Thomas KP (2012) An electrooculogram based assistive communication system with improved speed and accuracy using multi-directional eye movements. In: Proceedings of international conference on telecommunications and signal processing, pp 554–558
71. Anson D et al (2006) The effects of word completion and word prediction on typing rates using on-screen keyboards. *Assist Technol* 18(2):146–154
72. Diaz-Tula A, Morimoto CH (2016) Augkey: increasing foveal throughput in eye typing with augmented keys. In: Proceedings of the 2016 CHI conference on human factors in computing systems. ACM, pp 3533–3544
73. Townsend G et al (2010) A novel P300-based brain–computer interface stimulus presentation paradigm: moving beyond rows and columns. *Clin Neurophysiol* 121(7):1109–20
74. Cecotti H (2010) A self-paced and calibration-less SSVEP-based brain–computer interface speller. *IEEE Trans Neural Syst Rehabil Eng* 18(2):127–133
75. Meena YK, Cecotti H, Wong-Lin K, Dutta A, Prasad G (2018) Toward optimization of gaze-controlled human–computer interaction: application to hindi virtual keyboard for stroke patients. *IEEE Trans Neural Syst Rehabil Eng* 26(4):911–922
76. Meena YK, Cecotti H, Wong-lin K, Prasad G (2015) Simultaneous gaze and motor imagery bci increases single-trial detection performance: a compatible incompatible study. In: Proceedings of 9th IEEE-EMBS international summer school on biomedical signal processing, p 1
77. Meena YK, Wong-Lin K, Cecotti H, Prasad G (2017) Hybrid brain–computer interface for effective communication of decisions. In: 36th Meeting of the European group of process tracing studies in judgment and decision making (EGPROC), p 1

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.