2015-03

Filteryedping: design challenges and user performance of dwell-free eye typing

ACM Transactions on Accessible Computing, New York, v. 6, n. 1, p. 3:1-3:37, Mar. 2015
http://www.producao.usp.br/handle/BDPI/50948

Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing

DIOGO PEDROSA* and MARIA DA GRAÇA PIMENTEL, ICMC–University of Sao Paulo
AMY WRIGHT, Carolinas Neuromuscular/ALS-MDA Center
KHAI N. TRUONG, University of North Carolina at Charlotte

3

The ability to use the movements of the eyes to write is extremely important for individuals with a severe motor disability. With eye typing, a virtual keyboard is shown on the screen and the user enters text by gazing at the intended keys one at a time. With dwell-based eye typing, a key is selected by continuously gazing at it for a specific amount of time. However, this approach has two possible drawbacks: unwanted selections and slow typing rates. In this study, we propose a dwell-free eye typing technique that filters out unintentionally selected letters from the sequence of letters looked at by the user. It ranks possible words based on their length and frequency of use and suggests them to the user. We evaluated Filteryedping with a series of experiments. First, we recruited participants without disabilities to compare it with another potential dwell-free technique and with a dwell-based eye typing interface. The results indicate it is a fast technique that allows an average of 15.95 words per minute after 100min of typing. Then, we improved the technique through iterative design and evaluation with individuals who have severe motor disabilities. This phase helped to identify and create parameters that allow the technique to be adapted to different users.

Categories and Subject Descriptors: H.5.2. [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces—*Input devices and strategies (e.g., mouse, touchscreen)*; H.5.2. [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces—*Interaction styles (e.g., commands, menus, forms, direct manipulation)*; H.5.2. [**Information Interfaces and Presentation (e.g., HCI)**]: User Interfaces—*Evaluation / methodology*

General Terms: Design, Experimentation, Human Factors, Measurement

Additional Key Words and Phrases: Gaze, eye typing, dwell-free, motor disability, ALS, DMD, Filteryedping, AltTyping

ACM Reference Format:
Diogo Pedrosa, Maria da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design challenges and user performance of dwell-free eye typing. ACM Trans. Access. Comput. 6, 1, Article 3 (March 2015), 37 pages.
DOI: http://dx.doi.org/10.1145/2724728

*Work done while visiting University of North Carolina at Charlotte.
This work is supported by grant #2012/01510-0, São Paulo Research Foundation (FAPESP). Pimentel also acknowledges the support received from CAPES and CNPq.
Authors' addresses: D. Pedrosa and M. da G. Pimentel, Departamento de Ciências de Computação, Instituto de Ciências Matemáticas e de Computação, University of Sao Paulo, Avenida Trabalhador São-carlense, 400 - Centro, São Carlos, SP, Brazil, CEP: 13566-590; emails: {diogo, mgp}@icmc.usp.br; A. Wright, Carolinas Neuromuscular/ALS-MDA Center, 1010 Edgehill Rd N, Charlotte, NC, USA, 28207; email: Amy.Wright@carolinashealthcare.org; K. N. Truong, Department of Software and Information Systems, College of Computing and Informatics, University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC, USA, 28223; email: ktruong8@uncc.edu.

DOI: http://dx.doi.org/10.1145/2724728

1. INTRODUCTION

People affected by motor neuron diseases and disorders that cause muscle degeneration face communication struggles as their condition progresses over time and their ability to type or speak declines. Eye trackers—devices that determine where on the screen the user looks—are often used to help people with these conditions communicate. While interacting with a computer using an eye tracker, eye movements may be used to control the position of the pointer. Selections of targets on the screen can then be performed by blinking an eye (e.g., Ashtiani and MacKenzie [2010] and Tangsuksant et al. [2012]), pushing a physical button (e.g., MacKenzie and Zhang [2008]), or moving muscles that can still be controlled (e.g., Zhao et al. [2012]). However, these solutions for performing target selections do not suit all users and may cease to be viable options for many users because of their declining physical capabilities. These solutions may also be considered inconvenient for tasks that demand frequent selections, which is true in the case of typing. With eye typing, a virtual keyboard is shown on the screen and the user gazes at the keys that they want to type in sequence.

With dwell-based eye typing, the user selects a key by continuously gazing at it for an amount of time, which can be less than 400ms [Majaranta et al. 2009; Räihä and Ovaska 2012]. Although it is currently the most common method of eye typing, the dwell-based approach has two main drawbacks: when a short dwell time is used, it may result in unwanted selections of keys—which is known as the Midas touch problem; alternatively, it can be a relatively slow text input method when a long dwell time is used. Thus, the dwell time has to be carefully adjusted in order to find an optimal duration that minimizes both of these problems.

A dwell-free text entry technique does not require a dwell time to detect the user's intention for inputting a letter. Some of the possible approaches include the use of eye gestures for writing individual letters [Isokoski 2000; Sarcar et al. 2013; Chakraborty et al. 2014; Bee and André 2008; Wobbrock et al. 2008], eye typing via context switching [Morimoto and Amir 2010], and visual navigation of nested boxes of letters [Ward et al. 2000; Rough et al. 2014]. Although these works demonstrate the possibility of dwell-free eye typing, the user must learn a new way to write a letter rather than simply looking at where the intended keys would be found on a QWERTY keyboard layout. Kristensson and Vertanen [2012] demonstrated that dwell-free eye typing with a QWERTY-based keyboard layout can be theoretically much faster than existing eye-based text entry techniques. In their study, however, the software knows the text that the user wants to type and only accepts input when the user looks at a key that corresponds to the next letter in the word; it ignores a key if it is not the next letter in the word. Words can thereby be written even when the user looks at extra keys. In practice, such errors in eye typing must be handled by an actual dwell-free technique.

In this study, we first implemented two QWERTY-based dwell-free eye typing techniques—one of them proposed by us—and measured users' input performance with each. This helped us identify a strong candidate for dwell-free eye typing that we then evaluated alongside AltTyping[1] [Majaranta et al. 2009; Räihä and Ovaska 2012], currently one of the fastest dwell-based eye typing tools. Our aim with this work was to provide an understanding of how dwell-free eye typing compares to dwell-based approaches in actual practice.

In the preliminary study, we compared two approaches for supporting dwell-free eye typing. The first is a *shape-based* approach that implements the algorithm described by Kristensson and Zhai [2004] to recognize the intended word by comparing the shape of the path covered by the eye gaze with shapes stored in a word list. The

[1]Downloadable at http://www.sis.uta.fi/~csolsp/downloads.php, accessed January 24, 2014.

second technique is a *key filtering–based* approach that recognizes the intended word by applying a weighting to the length and frequency of all possible words formed by filtering extra letters from the sequence of letters gazed at by the user. For short, we refer to it as *filtering*. Based on results from this preliminary study, we adopted a *filtering-based* approach that incorporates visual feedback as the dwell-free eye typing candidate to compare further against a dwell-based eye typing method—AltTyping.

We divided our main evaluation into two phases. First, we compared the *filtering* using a visual feedback dwell-free eye typing method with AltTyping in an experiment involving participants without motor disabilities. Then, we recruited participants with Amyotrophic Lateral Sclerosis (ALS) and Duchenne Muscular Dystrophy (DMD) and conducted an iterative design and evaluation of the *filtering* method to enhance its design for those with motor disabilities. ALS is a disease that causes the degeneration of the upper and lower motor neurons, which in advanced stages causes the loss of the ability to initiate and control all voluntary movement [Medical News Today 2009; National Institute of Neurological Disorders and Stroke (NINDS) 2012]. DMD is a form of muscular dystrophy caused by a defective gene, which usually affects boys. Individuals with this condition have progressive loss of muscle function and weakness, which begins in the lower limbs. The ability to walk may be lost by age 12, and breathing difficulties and heart disease usually start by age 20 [U.S. National Library of Medicine 2012; National Human Genome Research Institute 2013; Patient.co.uk 2013].

Our evaluation results show that participants without motor disabilities were able to reach an average of 15.95 words per minute (WPM) with the proposed *key filtering–based* approach and 11.71WPM with AltTyping after 100min of typing with each (in the 6th session). Participants affected by ALS or DMD were able to reach an average of 7.60WPM with the *filtering* method and 6.36WPM with AltTyping after using each for 60min, even though many of these participants currently use or have previous experience with dwell-based eye typing systems. By the end of their participation in the study, 11 out of 12 participants preferred dwell-free eye typing over dwell-based eye typing. Subjective workload assessment scores reveal that the workload for typing with a filtering-based technique is lower than or equivalent to the workload for typing with a dwell-based technique. In addition to these results, we learned through the course of our evaluation that preference for alternate keyboard layouts, variability in the precision and accuracy of eye tracking, and saccades with longer duration and slower velocity are key challenges that participants with ALS and DMD had with our dwell-free eye typing. We introduced and evaluated two key features—a short focus dwell time and a slow movement threshold—to address these issues. These features helped overcome the problem of selecting wrong words from the candidate list and allowed the system to differentiate slow eye movements from when the user's eye gaze has reached a target.

This article is organized as follows. The next section discusses related work. Section 3 explains the Filteryedping technique and presents the preliminary study comparing two dwell-free techniques. Section 4 details the two phases of the comparative study between a dwell-free and a dwell-based technique. Discussions and conclusions are presented in Sections 5 and 6, respectively.

2. RELATED WORK

Research regarding eye typing techniques has increased in the last 7 years, perhaps due to the popularization of eye trackers. One of the first works aimed to support dwell-free text input was proposed by Isokoski [2000]. It used off-screen targets as a way to avoid unwanted selections caused by unintended dwells when the user gazes at a target and tries to recognize what she is looking at—the Midas touch problem. This approach also helped to conserve the display area required by the keyboard. Isokoski discussed adaptations of different schemes—Morse code, MDITIM, Quikwriting, and Cirrin—for decoding target hit sequences into text, but did not validate the technique with controlled experiments. Quikwriting, which was originally created for pen-based computers, was also evaluated as a potential eye typing method in the work by Bee and André [2008]. In a controlled experiment with three participants, Quikwriting supported a typing rate of 5.0WPM, which was slower than the 7.8WPM rate achieved with their implementation of a dwell-based keyboard that used a dwell duration of 750ms.

Some other gesture-based techniques for typing letters have also been proposed. In EyeK, the dwell time is replaced by a gesture that moves the pointer from inside the key area for a letter to outside that area and then back inside it again; with this approach, users reached rates between 5.6 and 8.8WPM [Sarcar et al. 2013; Chakraborty et al. 2014]. With EyeWrite [Wobbrock et al. 2008], a method that is based on EdgeWrite's letter-like unistroke alphabet [Wobbrock et al. 2003], users type by moving the gaze point between the four corners of the gesture area. In a longitudinal experiment, participants typed at an average rate of 4.87WPM. pEYEdit, Iwrite, and StarWrite are three techniques proposed by Urbina and Huckauf [2007] that require the user to confirm the selection of a letter by looking at a specific area of the screen. These three techniques had comparable rates, ranging from 5.9 to 7.6WPM with novice users and from 8.4 to 11.4WPM with advanced users. The idea of looking at a specific region to confirm the selection of a letter was also explored by Morimoto and Amir [2010]. Their work introduces "context switching" through the duplication of the keyboard. In this way, the region used to confirm the selection of a letter from one keyboard is the other keyboard, which is then used to select the following letter. Trading screen space for speed, this approach led to an input rate of about 12WPM.

One of the best performing gaze-based text entry techniques is Dasher. It is a predictive text entry technique in which nested boxes of letters and symbols move across the screen from right to left and the user writes text by directing her gaze toward the box containing the desired letter or symbol. The size of each box is proportional to the probability under a language model of the corresponding letter or symbol being selected next [Ward et al. 2000; Rough et al. 2014]. In an experiment comparing Dasher with a baseline eye typing method, Dasher supported significantly faster entry rates (14.2WPM vs. 7.0WPM) [Rough et al. 2014]. Rough et al. noted that "*different experimental setups sample different participants, use different apparatus and stimuli, and use slightly different procedures*," however, the rates obtained with the baseline eye typing method were comparable with several works that they cited. A disadvantage with using Dasher is its interface, which requires a lot of concentration to learn and use.

The fastest eye typing tool reported to date is AltTyping [Majaranta et al. 2009], with reported entry rates reaching 20–24WPM [Räihä and Ovaska 2012]. It is a dwell-based method that allows the user to adjust the dwell time directly on the keyboard interface. AltTyping's relatively high input rate might be due to the method that was used to analyze the data, which focused on expert and error-free performance. In particular, they "*decided to analyze the character-level text entry rate only for characters entered correctly after another correctly entered character. [...] Furthermore, if the participant glanced at either the model line or the result line in between two key presses, the latter key press was again omitted from analysis*" [Räihä and Ovaska 2012].

Our Filteryedping prototype is an implementation of a dwell-free eye typing technique for the QWERTY keyboard layout. Kristensson and Vertanen [2012] previously showed that such a method could be "potentially much faster" than current eye typing implementations. In their experiment, users reached a mean entry rate of 46WPM using a system that simulates a perfect recognizer for dwell-free eye typing. That is, their study software knows in advance what the user wants to type. Each time the user looks at the next letter in the sequence, that letter is selected. As a result, there was no way for the user to commit an insertion error even when she may have looked at additional letters while typing.

3. DWELL-FREE EYE TYPING

Before comparing the dwell-free and dwell-based eye typing techniques, we first explored two possible approaches for dwell-free eye typing. We compare a *shape-based* approach against a *key filtering–based* approach. In a position paper, Hoppe et al. [2013] have suggested previously that perhaps a dwell-free technique could increase the entry rate of eye typing systems. They propose an approach, called Eype, which turns eye gaze data over a QWERTY keyboard into a trace that joins the characters looked at by the user. Eype removes repetitions within the trace and then compares it against the optimal traces of each word in a corpus to identify the closest match. No performance evaluation of Eype has been reported. However, this approach is similar in concept to SHARK[2] [Kristensson and Zhai 2004], an established touch-based word-level gesture keyboard technique that has inspired many subsequent systems, such as Word Flow,[2] and has been well evaluated. Thus, for our shape-based approach, we implemented an adaptation of SHARK[2], to work with eye trackers. As an alternative approach, we developed Filteryedping, a key filtering technique that addresses the problem that some of the letters that the user looks at are accidental and may not be part of the desired word.

In this section, we first describe our implementation of the two dwell-free techniques in detail. Then, we describe an experiment comparing them. The obtained results indicate that Filteryedping is a suitable dwell-free eye input candidate to evaluate against AltTyping—a fast dwell-based tool.

3.1. The Filteryedping Technique

The Filteryedping technique recognizes the intended word by looking in a word frequency list for all the words that can be formed when discarding none or some of the letters that the user has looked at. The possible words are sorted based on the length and frequency and then presented to the user as a ranked list. The name of the technique is an example of a stream of letters that can generate the words "filtered," "eye," and "typing." Thus, it succinctly describes and demonstrates the idea of the technique: filtered eye typing.

3.1.1. Interface. Figure 1 shows the interface of the Filteryedping prototype. The user writes a word using this technique by looking at each letter of that word, in the same way as she would while using a dwell-based input technique, except that she does not have to dwell over a letter to select it. Visual feedback is provided to show the user where the system recognizes her current gaze position to be on the screen. Filteryedping displays the key looked at by the user in a different color (see letter "a" in Figure 1). The time it takes for the system to recognize the location and highlight a key is almost imperceptible (<33ms).

After typing the last letter of a word, the user must look at the bottom part of the interface that will then be populated with a candidate list of words. A target button helps the user to look at the position where the top-ranked suggested word will appear. Then, the user can traverse the candidate list to look for the intended word. Arrow buttons in the extremities support paginating for more candidates. Again, no dwell time is involved. If the user does not find the word she wants, she can try to type

[2]http://research.microsoft.com/apps/video/default.aspx?id=211650, accessed July 10, 2014.

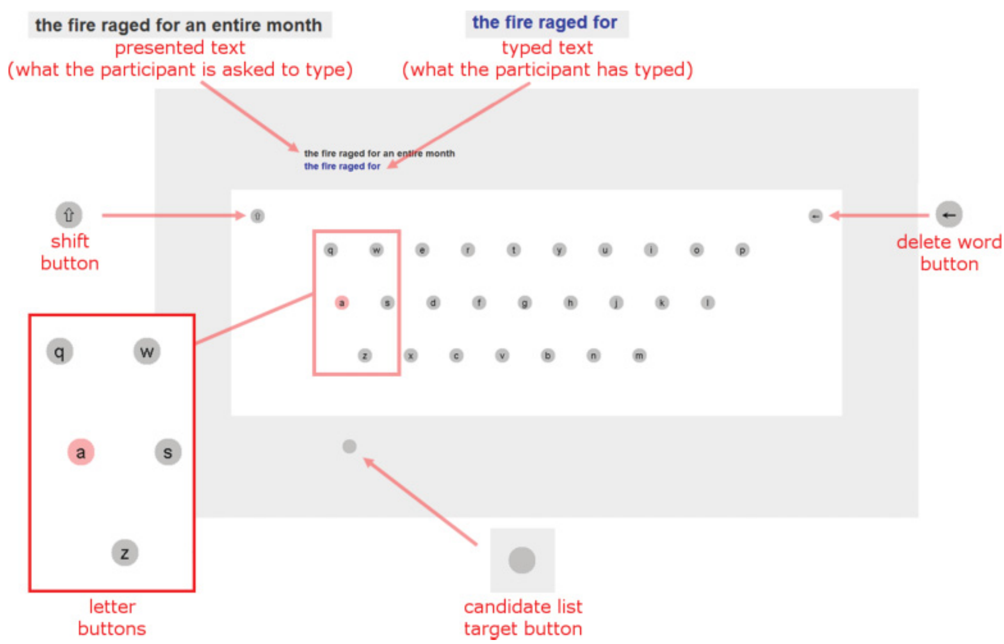
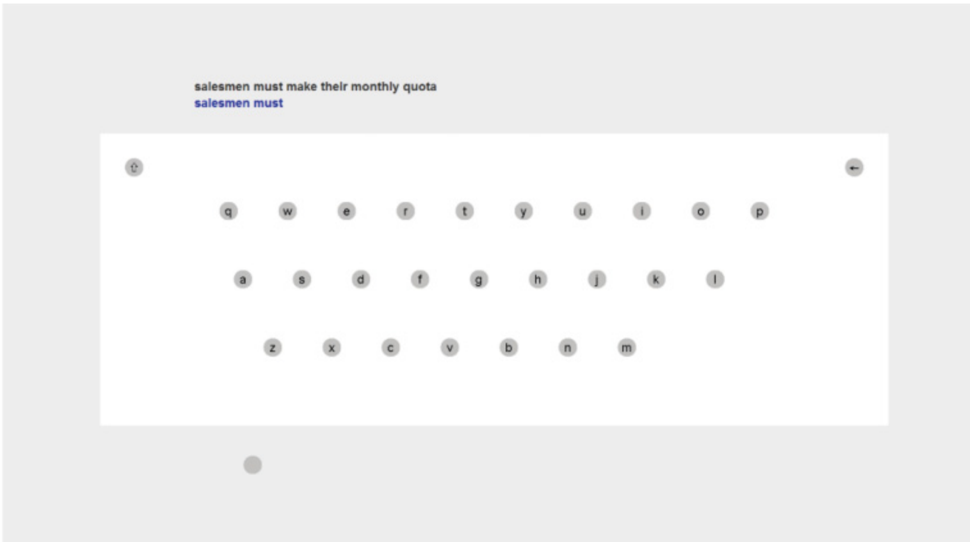


Fig. 1. Screenshot of the study software with the Filteryedping interface while user looks at the “a” key.

that word again by looking back to the keyboard while one of the two arrow buttons is selected. By doing this, no word is added as typed text. Alternatively, to accept the highlighted candidate word, the user must look to the typed text area or back to the keyboard to type the next word. In this way, the user can confirm that the correct word has been entered or simply continue to type if she does not need to check.

A storyboard illustrating this process is shown in Figure 2. Note that when looking at the screen, the user may perform a *fixation* to rest her eye gaze on a particular area of the display and a *saccade* to perform a ballistic eye movement between two fixations. High saccadic velocity (>300deg/s) and the specific frequency at which eye trackers sample gaze points mean that the eye tracker typically does not report many points along the trajectory between two distant fixations. Thus, in this example, when the user looks at one key after another, the eye tracker does not report keys in between them. However, the user may overshoot or undershoot a target key and hit another key by mistake; we illustrate this issue in the example by demonstrating the accidental selection of the “s” key when the user performs a saccade from the “m” key to the “a” key.

We intentionally designed the interface to show a small visible area for each key to help direct the user’s gaze toward the center of the detection area. However, the detection area for a key is not the same as its visible area. Figure 3 shows in green the actual detection area for each key. Note that the keys overlap horizontally. Not considering the overlap area, the aspect ratio of a key matches the one typically used in physical QWERTY keyboards, which is 0.867. However, the small width of a key may cause some horizontal recognition errors—that is, the recognizer falsely detects the gaze point over an adjacent key to the one that the user actually wants to select. To overcome this issue, we made the detection area wider, introducing an overlap area. If the system recognizes the gaze inside an overlap area, such as between “e” and “r,” it includes in the character stream not only one letter, but the concatenation of the



Screenshot of how the study software with the Filteryedping interface looks before the user types "make."

Step 1	The user has successfully typed "must" and sees "make" is the next word	**Step 2**	The user looks at the "m" key	**Step 3**	The user accidentally looks at the "s" key while trying to look the "a" key
Step 4	The user looks at the "a" key	**Step 5**	The user looks at the "k" key	**Step 6**	The user looks at the "e" key
Step 7	The user looks at the candidate list target button				
Step 8	The system shows the candidate words that best match the sequence of letters looked at. The word "make" is the top ranked candidate. Thus, it shows up first and is automatically selected.				
Step 9	*(Option 1)* The user looks at the output text area to accept the selected candidate word and add "make" to the written text				
	(Option 2) The user looks at the "t" key to begin typing the next word ("their"). This adds the selected candidate word ("make") to the written text				

Fig. 2. A storyboard illustrating the user typing "make." The top image shows how the study software looks before the user begins to type "make." Below it are the steps taken by the user to type "make." For each step, a cropped image is included to show where the user looks and the how the system responds.

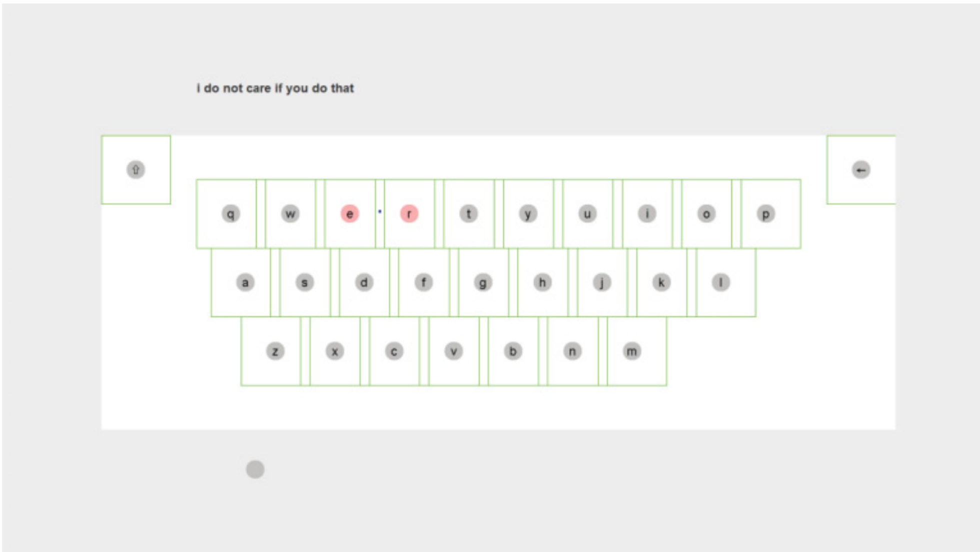


Fig. 3. Filteryedping interface: Actual detection area for each key (not seen by the user).

first key with the second key and then the first one again. In the preceding example, it would concatenate "ere" in the stream. That way, the stream is useful not only for words that include one of the letters, but also for words that include "er" or "re." The user will see both letters highlighted in the interface at the same time and can safely proceed to type the next letter in the word.

If the user looks at the top right key ("←", which we refer to as the "delete word button"), the interface actually shows a menu with four options. The first three options are Delete word, Backspace, and Enter. The option menu works exactly like the candidate list: no dwell is required and the functionality corresponding to the selected button is activated only after the user looks back to the keyboard or to the text area. The fourth button works as a dismiss button, for cases in which the menu was opened by mistake (Figure 4, right). Similarly, the top left key ("⇧," which we refer to as the "shift button") offers options for Shift, Caps Lock, and switching to a numbers and punctuation layout (Figure 4, left).

In order to also provide auditory feedback, we used the FreeTTS[3] speech synthesizer library. The prototype speaks each written word or command name (i.e., "deleted," "backspaced," "shift," "caps lock," and "lower case") immediately after confirmation/selection.

3.1.2. Word Frequency List. We created a word frequency list by starting with words from the Corpus of Contemporary American English (COCA) [Davies 2008], and reducing it to omit words that contained nonalphabetical characters and words that are not included in *dictionary.com*. Then, we added the British spelling for two words that were in the Mackenzie and Soukoreff [2003] phrase set used in the experiments (see Section 3.3). The result was a list of 133,223 words with their associated frequencies of occurrence. Additionally, we included several common misspellings (see Section 3.1.6).

3.1.3. Support for Entering Out-of-List Words. The technique explained so far allows for input of words contained in the word frequency list. To allow users to type words out

[3]http://freetts.sourceforge.net, accessed July 7, 2014.

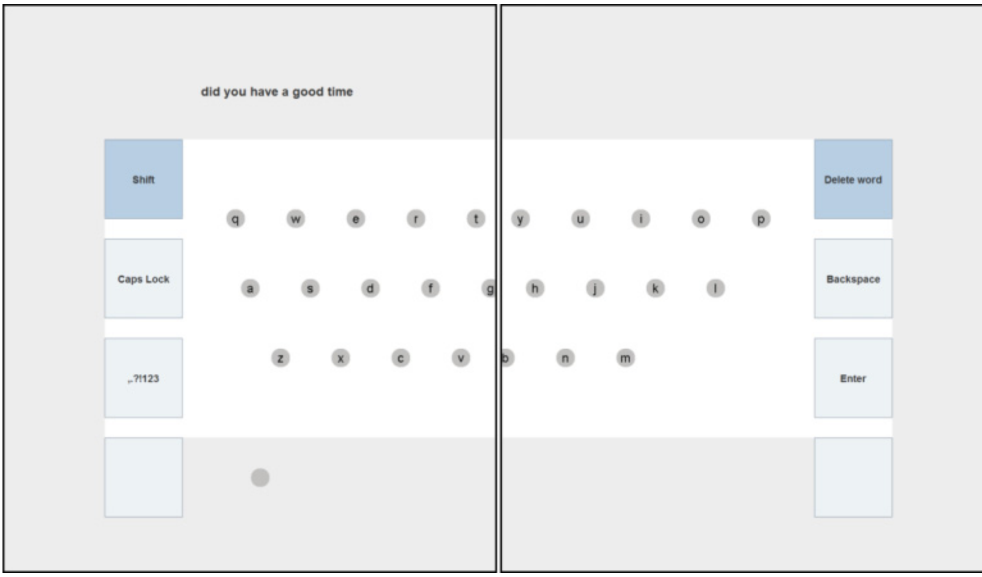


Fig. 4. Filteryedping interface: Left view while user looks at "Shift" button (left) and right view while user looks at "Delete word" button (right).

of the list, such as passwords or less common first and last names, the user can dwell about 1s (though this default value is adjustable) over each desired key and then look at the leftmost button in the candidate list after typing all keys. It will show the word composed by the dwelled keys. That position is typically where the "previous page" button is shown after paginating at least once. If the user has not dwelled over any letter, the first page of the candidate list does not show anything at that position. If the user has dwelled by mistake over one or more letters, all the user has to do is to ignore the word suggested in the leftmost button. In this way, we also support dwell-based eye typing without requiring an explicit mode change.

The dwell time is a configurable parameter. After half of this duration, the visual feedback changes to indicate that the letter is about to be included as a dwelled letter. The key color smoothly transitions from pink to red. After dwelling for the complete duration, an abrupt transition (or blink) from red to pink is used to indicate that the letter has been appended to the sequence of dwelled characters that will be shown in the first candidate word button.

3.1.4. Ranking Algorithm. As commented before, words in the candidate list are sorted based on the length and frequency. We weight those two factors using the following formula:

$$\text{score}\,(\text{word}) = \log_{10}(\text{freq}\,(\text{word})) + w * \text{length}\,(\text{word})\,, \tag{1}$$

where freq(word) is the number of times the word appears in the corpus, length(word) is the number of characters in the word, and w is a weight. The higher the score of a word, the closer to the beginning of the candidate list it will be. To define the value of w, we compare the average position that all words in the word frequency list would have in the candidate list (weighted by the frequency of the word) in the case of perfect gaze input from the user, for different values of w. Figure 5 shows the average position for different values of w. Jumps in the graph happen when two words that would have the same input (e.g., "to" and "too") change positions. By choosing the weight that leads to the minimum average position, we would rely too heavily on the user's ability to

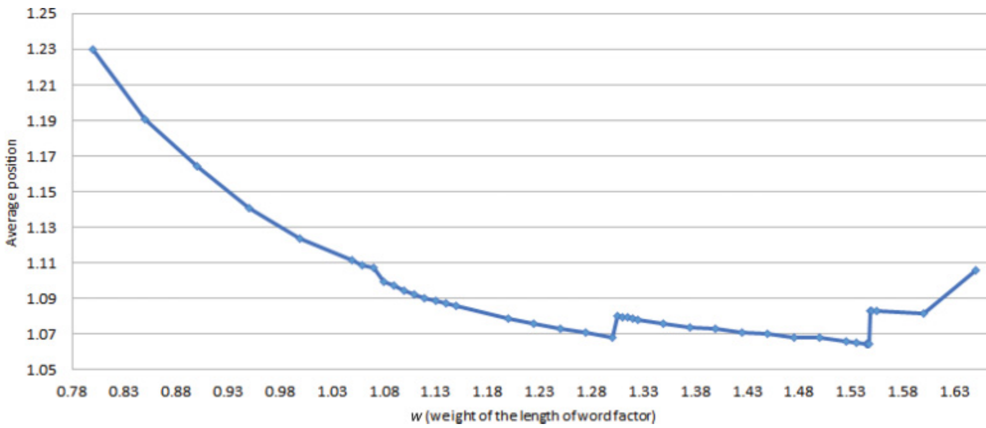


Fig. 5. Average position of words in the candidate list for different values of w (weight of the length of word factor).

ALGORITHM 1: Creating the list of suggestions

```
For each word in the frequency list
    If input stream contains it (match)
        Add to the suggestions list
Sort the list by the rank
```

ALGORITHM 2: Testing if an input stream contains a word

```
boolean match (String input, String word){
        valid = false;
        i = 0;
        for (w = 0; w<word.length(); w++){
                for (; i < input.length(); i++){
                        if (word.charAt(w) == input.charAt(i)){
                                valid = true;
                                break;
                        }
                }
                if (valid){
                        break;
                }
        }
        return valid;
}
```

perform perfect input. Instead, we decided to use the value of 1.08, which is small enough to not favor the length of a word in detriment to its frequency, but still leads to a small average position of 1.0996.

The basic algorithm for creating the candidate list is shown in Algorithm 1 and the pseudocode for the method that tests if an input stream contains a word is shown in Algorithm 2.

3.1.5. Processing Eye Tracker Data. Because of the way that human eyes work and the limited accuracy and precision of eye trackers, the data provided by eye trackers via the Application Programming Interfaces (APIs) are noisy. To address this issue, we implemented the saccade detection and fixation smoothing algorithm proposed by

Kumar [2007]. We developed a module for processing the data generated by the eye tracker to be completely independent from the prototype. The module first acquires the point on the screen corresponding to the eye gaze, then it applies the smoothing algorithm, and finally it issues an operating system command to move the mouse pointer to that position.

In each sampling cycle, the application is notified if neither one, nor both eyes were detected by the eye tracker. When both eyes are detected, the application receives separate positions for each eye. We calculate and use the average position. If only one eye is detected, we simply use the provided position. We do not do anything in the cycles when neither eye is detected by the eye tracker.

The first step of the saccade detection and fixation smoothing algorithm is to determine whether the most recent data point is the beginning of a saccade or a continuation of the current fixation. We use a saccade threshold of 90 pixels, which is a bit smaller than the 115 pixels horizontal separation between the keys in our keyboard. The basic idea of the algorithm is to determine the current fixation as a weighted mean of the points that are less than the saccade threshold apart and occurred within a specific duration. We use a window of 15 points at most, which for an eye tracker that works at 30Hz is about 500ms.

3.1.6. Spell Correction. Misspelled words are often highlighted or automatically corrected in many text editors, which handle this problem immediately after the user has finished typing the word. In Filteryedping, misspellings caused by letter insertions are not possible, because of the filtering approach used. However, if the user skips a letter or switches the order of letters, neither the misspelled word nor the desired word would be in the candidate list. To help users in these cases, we created a list of misspelled words by merging two lists available online.[4] We then added this list of misspelled words to our word frequency list. Thus, when the user misspells a word, the algorithm is able to detect the user's intention to write a word. Instead of adding the misspelling to the candidate list, the system adds the corresponding correctly spelled word. We compute the score for these words using the length of the misspelled word but the frequency of the corrected one.

3.2. Shape-Based Eye Typing

The shape-based eye typing technique uses an algorithm that recognizes the intended word by comparing the shape of the gaze path with the precomputed shape for each word in a corpus [Kristensson and Zhai 2004]. We followed the algorithm described by the authors as closely as possible. We used 20 points as the total number of sampling points in the Proportional Shape Matching algorithm. We used 1,000 pixels as the bounding box length L while normalizing the shapes in scale and location. For the integration of the location and shape channel, we used 44 and 100 for σ in the shape Gaussian probability density function for location and shape, respectively.

As the authors suggested, we also prune all word candidates that have a shape or location distance larger than 2σ. Then, as a second pruning step, we process only the first 48 words (6 pages worth of suggestions) from the list. Finally, we sort the candidates by distance, showing first the ones that better match the input.

When eye typing, there is no explicit delimiter for an input path—contrary to what happens in touch-based typing, in which the contact of the finger with the screen is the delimiter. As a result, the beginning and end of a gaze path must be trimmed to remove

[4]"Wikipedia:Lists of common misspellings/For machines" (http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines, accessed March 6, 2014) and "Common misspellings" (http://www.oxforddictionaries.com/words/common-misspellings, accessed March 6, 2014).

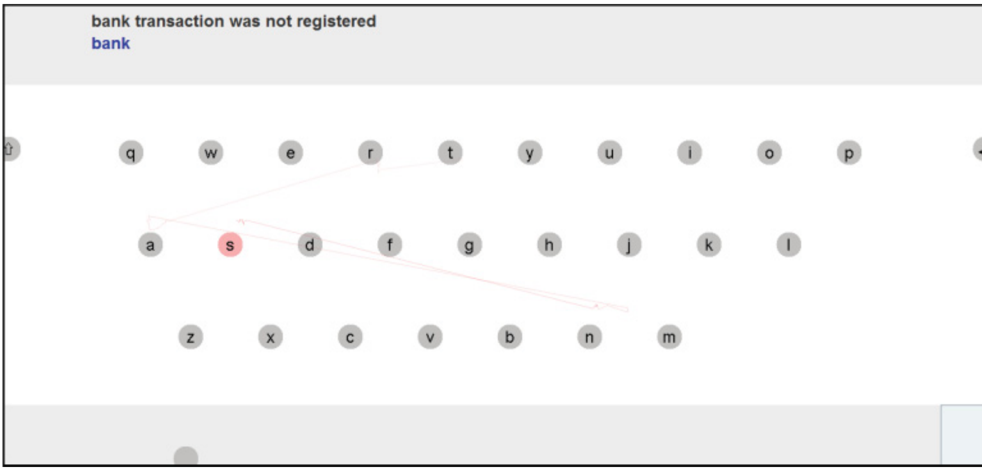


Fig. 6. Visual feedback for the shape-based eye typing technique.

the additional points that the user's eye gaze crosses through as it enters and leaves the keyboard. Our implementation discards points in the beginning of the gaze path until the first point is within 60 pixels of the following two points. Similarly, it discards points from the end of the path until it reaches a point that is within 60 pixels of the previous two points. Although we adopted the described approach, other strategies could be tested and employed, such as using a dwell time to unequivocally identify the first letter of the word [Hoppe et al. 2013].

Except for the visual feedback, the interface for our implementation of shape-based eye typing is the same as the interface for Filteryedping. Besides highlighting in pink the current key that the user is looking at, the system also draws a fading line connecting the last 60 gazed points (Figure 6). The eye tracker data processing method and spell correction described for Filteryedping were also used in the shape-based eye typing technique.

3.3. Comparison of Approaches

To determine which dwell-free eye typing method to evaluate further against dwell-based eye typing, we first conducted an experiment in which half of the participants were asked to perform typing tasks with and without visual feedback using the shape-based technique and the other half did the same using the Filteryedping technique. In the typing without feedback conditions, the gaze position recognized by the system is not indicated to the user. The technique and the order of feedback used were randomly selected. The last participant(s) were assigned a technique and order of feedback type so that we had the same number of users for each condition. Participants were asked to type as quickly and accurately as possible; additionally they were instructed not to use the dwell functionality.

The version of Filteryedping used in this preliminary study was the same as the one described in Section 3.1, except for four differences: (1) confirmed words were not spoken by the prototype; (2) the candidate list bar had eight suggestions instead of six, and did not have any space between the suggestions; (3) there was no space between options in the vertical menus; and (4) "Enter" was the first option in the right vertical menu.

We recruited 12 participants (five females) via a university mailing list containing the e-mails of students and staff, and via word of mouth with personal contacts in our social circle who are or have access to potential participants. Participants were

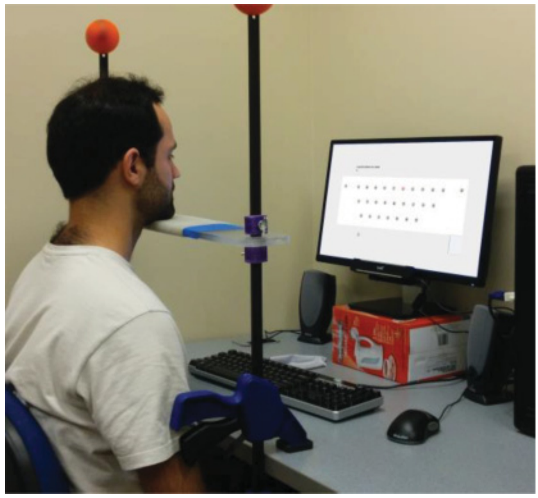


Fig. 7. Study setup for the trials: A chin rest, speakers, a monitor showing a virtual keyboard, and an eye tracker attached to its bottom part.

25.2 years old on average (std = 7.6, min = 19, max = 44). All have extensive experience with QWERTY keyboards and were fluent in English. None of the participants have used an eye tracker before, except one, who had used it for less than 1h. None of the participants have any motor disorder. Two participants use glasses, two use contact lenses, and eight performed the tasks without any corrective lenses.

The experiment consisted of three sessions for each participant. No more than 72h elapsed between sessions, and no more than two sessions occurred on the same day. If two sessions were performed on the same day, at least 2h elapsed between them. Each session was divided into four blocks. In each block, participants were asked to type, as quickly and accurately as possible, randomly selected phrases from the set of 500 phrases created by MacKenzie and Soukoreff [2003], for 5min (time between phrases was not considered). Participants were not interrupted in the middle of a phrase when the 5min had elapsed; they were allowed to finish typing that last phrase. The number of phrases seen by a participant in each block depended entirely on how fast she typed. After pressing “Enter” to indicate the end of a phrase, the interface showed the participant the adjusted typing rate (typing rate adjusted to take into consideration the uncorrected error rate) near the typed phrase. This was done to motivate the participants to try to improve upon their performance.

Halfway through each session, a 5-min break was taken and the feedback type was changed. To allow the participants to become familiar with the technique and to warm up, they started each half-session by practicing for 5min. In all sessions, participants were asked to sit still in front of the monitor, resting their chin over a chin rest (Figure 7), to calibrate the eye tracker and perform the typing tasks. We used a Tobii REX eye tracker (which works at a sampling rate of $\sim$30Hz) attached to a 21.5″ Dell monitor (1,920$\times$1,024 pixels). Nine-point calibration was used in the study. Although the chin rest was not necessary, it was used to help participants understand that they only needed to move their eyes. Before each block the participants were allowed a chance to recalibrate until they were comfortable with the accuracy of the eye tracker.

The first session began with an informed consent agreement and a demographic survey. Also in the first session, the participants were briefed on the procedures of the experiment and introduced to the dwell-free eye typing technique that they were about to use. Additionally, after finishing the second block of the first session only, the participants were given a page to read containing instructions on how to rate scales of subjective workload assessments with the NASA Task Load Index (NASA TLX)[5] and a page containing the instructions on how to specify the sources of workload. In all sessions, after blocks 2 and 4, the participants were asked to fill in a NASA TLX rating sheet and the sources-of-workload evaluation. At the end of each session, the participants were asked to complete a form regarding their experience, thanked for their time, and compensated US$10. At the end of the final session, each participant also received a US$25 bonus for completing all sessions. An additional bonus of US$10 was given to the two participants (one per technique) who obtained the fastest adjusted typing rate in each session. This compensation schedule was chosen to encourage continued participation in the experiment and effort to type quickly and accurately.

3.4. Results

To analyze the study data, we used StreamAnalyzer for computing the typing rate and the rate of errors that were left in the transcribed text. It is a publicly available tool[6] that analyzes the text input stream logs and produces text file output containing several statistics [Wobbrock and Myers 2006]. Results from each block were averaged to form a single measure per block. Then, results from blocks 1 and 2 and results from blocks 3 and 4 were averaged to form a single measure per participant per session per condition. All statistical significance tests used a significance level of $\alpha = 0.05$.

3.4.1. Metrics. Text entry rate (WPM) is defined in words per minute, where a word is five characters, including spaces. It is obtained by dividing the number of typed words by the number of minutes measured from the moment the user's gaze enters the keyboard for the first time after reading the target phrase to the moment the last word is written. The time taken to hit the Enter key was not included.

As a metric for the amount of errors left in the transcribed text, we use

$$\text{MSD error rate} = \frac{\text{MSD}\,(P, T)}{\overline{S_A}} \times 100\%,$$

where MSD(P,T) is the minimum string distance between the presented and transcribed strings, and $\overline{S_A}$ is the mean length of the alignment strings [MacKenzie and Soukoreff 2002].

To better understand the quality of the ranking algorithm, we are also interested in the average position of the selected words in the candidate list (avgPos). An average of 1 would mean that all the written words were found in the first position. Recall from Section 3.1.4 that with perfect user input, our Filteryedping algorithm results in an avgPos of 1.0996 for all words in the dictionary.

The quality of the ranking algorithm is not the only factor that affects the text entry rate. When the user needs to retype a word for any reason, this decreases her input performance. One reason why the user might need to retype a word is because she did not select a word from the candidate list after a typing attempt. For Filteryedping, if the user does not correctly gaze at all of the letters in the word, then the intended word will not be suggested. For the shape-based technique, if the shape of the path covered by the gaze does not closely match with the shape stored for the intended word, then it will not be suggested. Because the word that the user wants to type is not in the

[5]NASA TLX–Paper/Pencil Version. http://humansystems.arc.nasa.gov/groups/tlx/paperpencil.html, accessed July 17, 2014.
[6]http://depts.washington.edu/ewrite/eval.html, accessed July 14, 2014.

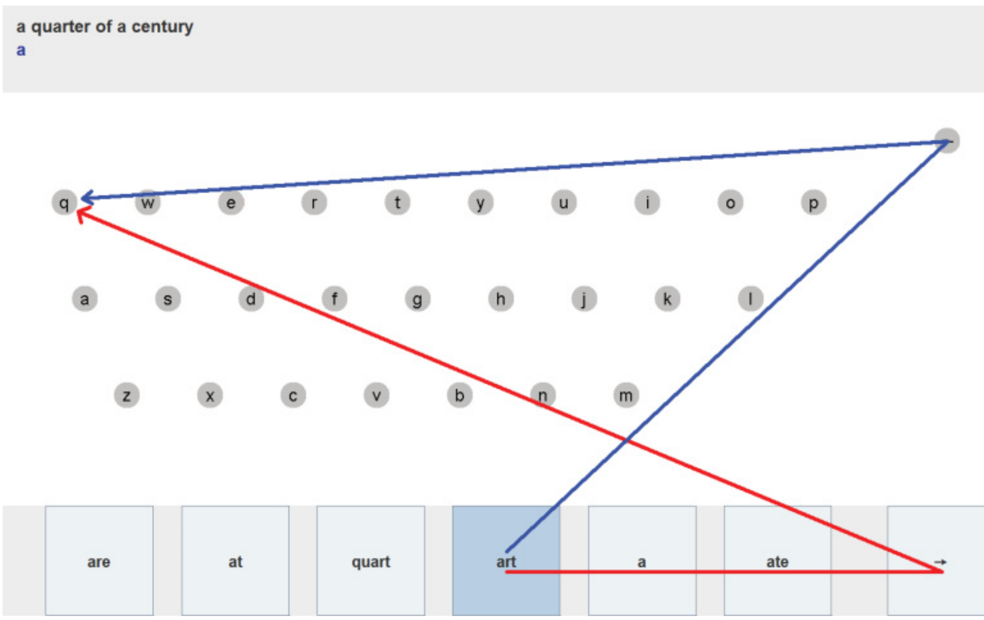


Fig. 8. Paths covered by the gaze in two equivalent strategies for the user to handle the case in which the intended word is not found.

candidate list, the user would not select any for input. The user also might not select a candidate word because the ranking algorithm failed to suggest the intended word at the beginning of the list or the user did not see it and missed it. We use a no word selected rate as a metric that indicates the percentage of writing attempts that resulted in no word being selected from the candidate list and therefore the intended word must be retyped:

$$\text{No word selected rate} = \frac{\text{\# of times no word was selected}}{\text{\# of times a word was selected} + \text{\# of times no word was selected}} \times 100\%.$$

A second reason why the user might need to retype a word is because a selected word is not what the user wants. This could happen because the user accidentally selected the wrong word from the candidate list. Alternatively, when the candidate list does not contain the intended word, instead of selecting one of the arrow buttons (to indicate to the system that no word is selected for input) before returning to the keyboard to retype the word, the user simply exits the candidate list with a word selected for input and then deletes it immediately afterward. Figure 8 shows two possible paths covered by the user's eye gaze for retyping the word "quarter" after not finding it in the middle of the candidate list. Note that they both require the user to gaze at either the right arrow button or the delete key before looking at the "q" key to begin retyping. Because the effort required in both actions are similar, either strategy might be adopted by the user.

The percentage of words typed that were deleted is calculated as follows:

$$\text{Deleted word rate} = \frac{\text{\#of times a word was deleted}}{\text{\#of times a word was selected}} \times 100\%.$$

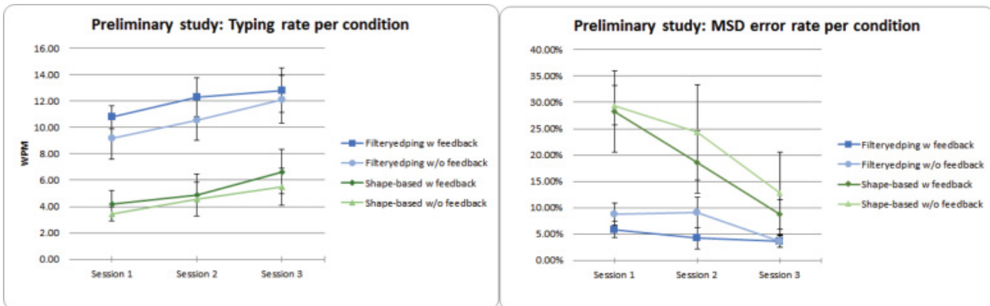


Fig. 9. Results of the preliminary study comparing two dwell-free eye typing techniques: Text entry rate (left) and MSD error rate (right). Bars convey the standard error of the mean.

Finally, to evaluate the task workload of each condition, we used the NASA TLX assessment tool. The overall workload score is the average of the ratings of the six subscales, weighted by the contribution of each factor.

3.4.2. Text Entry Rate (WPM). Figure 9 (left) shows the average entry rate obtained per condition per session. The entry rate with Filteryedping was on average 2.4 times faster than with a shape-based approach.

We conducted two repeated measures analyses of variance, one for each technique, to analyze the impact of Feedback and Session on the typing rate. For Filteryedping, there was no significant main effect for Feedback ($F_{1,30} = 1.17$, $\eta^2 = 0.034$, $p = 0.29$); no significant main effect for Session ($F_{2,30} = 1.36$, $\eta^2 = 0.080$, $p = 0.27$); and no significant interaction between Feedback and Session ($F_{2,30} = 0.07$, $\eta^2 = 0.004$, $p = 0.93$). For the shape-based method, there was no significant main effect for Feedback ($F_{1,30} = 0.50$, $\eta^2 = 0.015$, $p = 0.49$); no significant main effect for Session ($F_{2,30} = 1.48$, $\eta^2 = 0.088$, $p = 0.24$); and no significant interaction between Feedback and Session ($F_{2,30} = 0.05$, $\eta^2 = 0.003$, $p = 0.95$). Although we expected that participants would improve with practice and that the inclusion of feedback would help them to achieve better performance, we found no evidence to support this.

Next, we aggregated participants' typing rate in each session by averaging the values obtained with and without feedback. A new repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the typing rate. There was a significant main effect for Technique ($F_{1,30} = 34.79$, $\eta^2 = 0.512$, $p < 0.001$); but no significant main effect for Session ($F_{2,30} = 1.59$, $\eta^2 = 0.047$, $p = 0.22$); and no significant interaction between Technique and Session ($F_{2,30} = 0.02$, $\eta^2 = 0.001$, $p = 0.98$).

3.4.3. MSD Error Rate. Figure 9 (right) shows the average MSD error rate obtained per condition per session. The shape-based approach resulted in an average of 3.5 times more errors than Filteryedping.

To analyze the impact of different factors on the MSD error rate, we conducted the same sequence of analysis as we had done for typing rate. For Filteryedping, there was no significant main effect for Feedback ($F_{1,30} = 2.75$, $\eta^2 = 0.071$, $p = 0.11$); no significant main effect for Session ($F_{2,30} = 2.11$, $\eta^2 = 0.110$, $p = 0.14$); and no significant interaction between Feedback and Session ($F_{2,30} = 0.82$, $\eta^2 = 0.042$, $p = 0.45$). For the shape-based approach, there was no significant main effect for Feedback ($F_{1,30} = 0.45$, $\eta^2 = 0.012$, $p = 0.51$); a significant main effect for Session ($F_{2,30} = 3.83$, $\eta^2 = 0.200$, $p = 0.03$), which means participants committed less errors with practice; and no significant interaction between Feedback and Session ($F_{2,30} = 0.06$, $\eta^2 = 0.003$, $p = 0.94$).

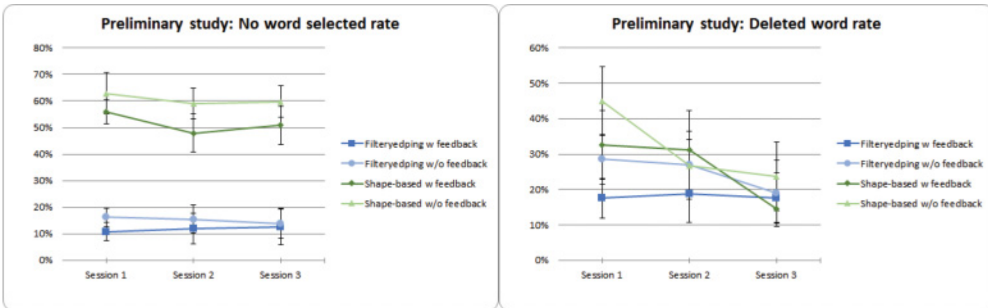


Fig. 10. Results of the preliminary study comparing two dwell-free eye typing techniques: No word selected rate (left) and deleted word rate (right). Bars convey the standard error of the mean.

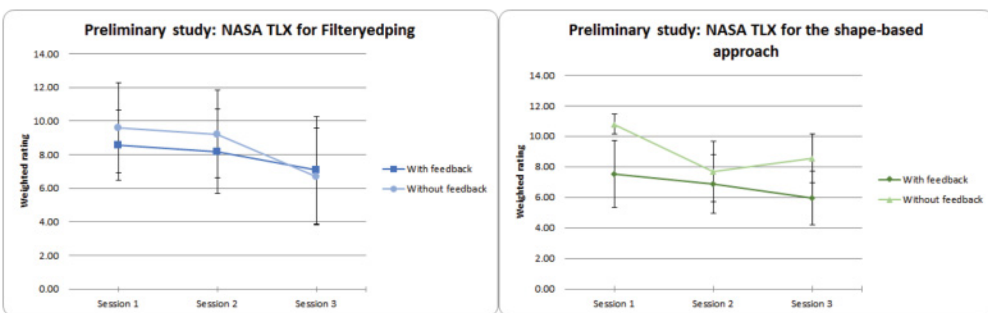


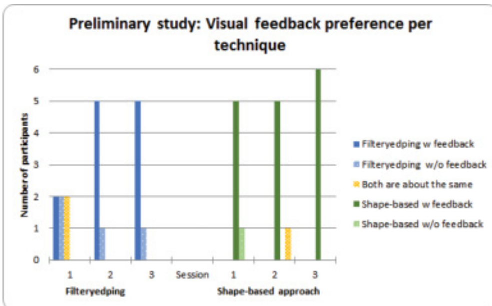
Fig. 11. NASA TLX weighted ratings of the preliminary study comparing two dwell-free eye typing techniques: Filteryedping (left) and shape-based approach (right). Bars convey the standard error of the mean.

After averaging values obtained with and without feedback, there was a significant main effect for Technique ($F_{1,30} = 21.51$, $\eta^2 = 0.340$, $p < 0.001$); and a significant main effect for Session ($F_{2,30} = 4.14$, $\eta^2 = 0.131$, $p = 0.03$); but no significant interaction between Technique and Session ($F_{2,30} = 1.77$, $\eta^2 = 0.056$, $p = 0.19$).

3.4.4. No Word Selected and Deleted Word Rates. Figure 10 shows the average obtained per condition per session for the two metrics on how often participants retyped a word because no word was selected from the candidate list or they needed to delete a word that was entered.

The results indicate that participants needed to retype words frequently. For example, over a quarter of times that participants tried to type a word using Filteryedping with visual feedback in Session 3, they ultimately needed to retype it because no word was selected 13% of the time and they needed to delete a word 18% of the time. In Section 3.5, we discuss reasons that lead to the high no word selected rate and deleted word rate that required users to retype text.

3.4.5. NASA TLX. For the task workload, we consider only the responses from eight participants (four per technique), because we do not have separate responses for each feedback condition from the first four participants. Figure 11 shows the average NASA TLX rating obtained per condition per session. It is hard to compare ratings given to Filteryedping with ratings given to the shape-based approach, because it is a subjective evaluation and users tend to give ratings that differentiate between the two conditions that they have used. Thus, in this study, we compare ratings for each method with and without feedback. For both approaches, the results suggest that the workload when



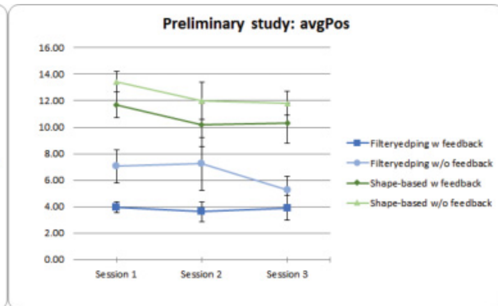


Fig. 12. Results of the preliminary study comparing two dwell-free eye typing techniques: Answers of the participants to the question: "Which of the two methods allows easier input?" after each session (left) and average position of the selected words in the candidate list—including bars conveying the standard error of the mean—(right).

using visual feedback is lower than or equivalent to the workload when typing without feedback.

3.4.6. Feedback Conditions. In the post-test questionnaire, we asked the participants: "Which of the two feedback conditions (with feedback or without feedback) allows easier input?" Figure 12 (left) shows the participants' answers separated by technique. Except for the first session with Filteryedping, participants believed that the use of feedback facilitates input.

3.4.7. avgPos. Figure 12 (right) shows the average position of the selected words in the candidate list (avgPos) per condition per session. We aggregated avgPos in each session by averaging the values obtained with and without feedback. A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the avgPos. There was a significant main effect for Technique ($F_{1,30} = 56.27$, $\eta^2 = 0.639$, $p < 0.001$); but no significant main effect for Session ($F_{2,30} = 0.68$, $\eta^2 = 0.016$, $p = 0.51$); and no significant interaction between Technique and Session ($F_{2,30} = 0.22$, $\eta^2 = 0.005$, $p = 0.80$). This suggests that the key filtering–based approach identified the words typed by users in dwell-free mode better than a shape-based approach.

3.5. Discussion

The objective metrics (WPM, MSD error rate, and avgPos) and subjective metric (NASA TLX) indicate that the key filtering–based approach is a satisfactory dwell-free eye typing method. Participants also indicated that the use of visual feedback was a positive functionality to include. Thus, we decided to make some improvements and continue our investigation using the Filteryedping with visual feedback prototype only.

One of the main problems that we learned about the prototypes was related to users selecting words in the candidate list and options in the vertical menus accidentally. This was a common problem because the gaze position reported by the eye tracker is subject to tiny, rapid, and unstable movements associated with visual fixations, and lack of accuracy and precision in the eye tracking mechanism itself. Because the selection of a candidate word or a menu option requires only the detection of one single point inside that button followed by the detection of a single point in the keyboard or in the typed text area, unstable fixations or noise in the eye tracking data can easily cause this error. This problem decreased the participants' input performance in two ways. First, it often caused the candidate list to be accidentally invoked before the user has finished typing a word. This can result in the selection of a random word from the candidate list by mistake, which the user must then delete. Second, when this same problem happens while the user is traversing the candidate list, it can cause the selection of words that are not intended by the user. The user must also delete these words and retype the intended word. These problems contribute to the high no word selected rate and deleted word rate discussed in Section 3.4.4.

Related to this, four participants mentioned in the post-test questionnaire that they preferred to have the “Delete word” and “Enter” buttons further apart from each other. Because both options were in the same vertical menu, users sometimes selected one instead of the other by error. We also observed that participants wanted to use “Delete word” more frequently than “Enter,” thus, it should be easier to access.

To overcome these problems, we introduced two changes to the prototype: (1) we reduced the number of options in the candidate list from eight to six to create some space between them, and (2) we changed the order of the options in the right vertical menu from “Enter,” “Delete word,” and “Backspace” to “Delete word,” “Backspace,” and “Enter.”

This preliminary study investigates two potential ways to support dwell-free eye typing. Hoppe et al. [2013] have previously suggested and implemented a *shape-based* approach, but have not yet conducted or reported a performance evaluation of that method. Thus, the results of this experiment provide an understanding about the effectiveness of *shape-based* and *key filtering–based* eye typing methods. Furthermore, it provides some initial guidance on how to design a dwell-free eye typing keyboard interface.

4. DWELL-FREE VERSUS DWELL-BASED EVALUATION

We next performed a study comparing dwell-free eye typing with dwell-based eye typing. For this purpose, we used Filteryedping and AltTyping [Räihä and Ovaska 2012], the fastest dwell-based eye typing tool reported in the literature. We divided this study into two phases:

—Phase 1: a performance evaluation with participants without physical disabilities.
—Phase 2: an iterative design and evaluation of Filteryedping with participants with ALS and DMD.

In this section, we first describe how AltTyping was used in the study. Then, we describe the experiment conducted in Phase 1 and its results. We finish by presenting the methodology and results for Phase 2.

4.1. AltTyping

AltTyping allows the user to type a letter by fixating her gaze over that key for a certain amount of time. Räihä and Ovaska’s evaluation of AltTyping [2012] was divided into two phases: (1) a Learning Phase, during which participants used AltTyping for 10 sessions of about 15min each and could adjust the dwell time as they wished, and (2) an Advanced Phase, during which the same participants used AltTyping for five 15-min sessions. In the Advanced Phase, the dwell time started at 410ms in the first session and was decreased by 40ms each session.

In our study, we used the first session to allow participants to become familiar with the interface, similar to the Learning Phase from Räihä and Ovaska’s study. The dwell time started at 450ms and could be adjusted by the participant at any time. From the second to the sixth session, we reproduced the dwell times used in the five sessions from the Advanced Phase in Räihä and Ovaska’s study.

In Räihä and Ovaska’s study [2012], a 17-in. monitor with $1,280 \times 1,024$ resolution was used. AltTyping was displayed as a full-screen interface. To mimic their conditions as much as possible, we set the AltTyping window to be 17in. diagonally (using specifically a $1,350 \times 1,080$ window size to keep the same aspect ratio as their study),

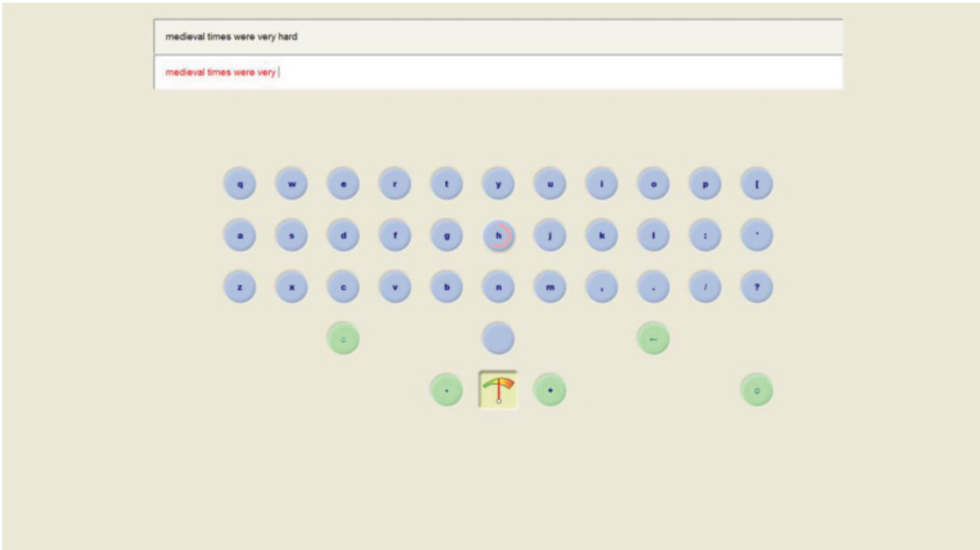


Fig. 13. A screenshot of the AltTyping interface during first session of the experiment. The following five sessions did not show the "−" and "+" buttons or the dwell time indicator.

centered it, and placed a software frame around it to hide all other elements of the interface (Figure 13).

Because Tobii REX is not one of the eye trackers supported by AltTyping, we configured the software to work with the mouse mode and used the same module for processing eye tracker data also used with Filteryedping. Additionally, we configured the operating system to hide the mouse cursor, to minimize visual distractions for the participant.

4.2. Phase 1

In Phase 1, we recruited participants without physical disabilities. The experiment followed a within-subject design, in which all the participants performed typing tasks using Filteryedping with visual feedback and using AltTyping. The order of use of the techniques was randomly selected at the beginning of the first session for each user, and was kept constant for each participant's remaining sessions. The last participant(s) were assigned the order of use of the techniques so that we had the same amount of users starting with each condition.

The version of Filteryedping used in Phase 1 was the same as described in Section 3.1, except for three differences: (1) confirmed words were not spoken by the prototype, (2) there was no space between the options in the vertical menus, and (3) the stream of gazed at letters was cleared every time the user looked outside the keyboard (for example, to check the typed or presented text). Differing from what we had in the preliminary study, this version did not show the adjusted typing rate after each phrase. We removed that feedback in order to ensure that consistent information was offered to the participants while using AltTyping and Filteryedping. Participants were again instructed not to use the dwell functionality.

We recruited six participants (two females) using the same recruiting methods as described for the preliminary study comparing the dwell-free approaches. Participants were 22.8 years old in average (std = 1.3, min = 21, max = 24). All had extensive experience with QWERTY keyboards and were fluent in English. None of them had

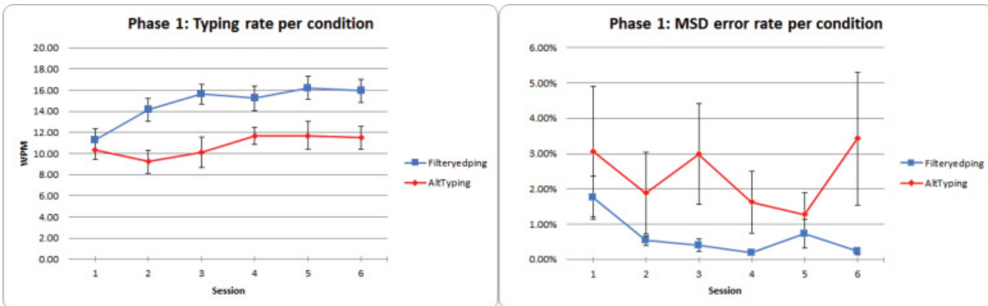


Fig. 14. Results of the comparison between a dwell-free and a dwell-based eye typing technique: (a) Text entry rate and (b) MSD error rate. Bars convey the standard error of the mean.

used an eye tracker before or had any motor disorder. Three participants used glasses and three performed the tasks without any corrective lenses.

This phase consisted of six typing sessions for each participant. Each session was divided into two blocks of 20min each. Before each block, participants were given 2min to practice and become familiar with the typing technique they would use next. Again, we used the NASA TLX to evaluate subjective workload. However, we adjusted the procedure so that each participant specified the sources of workload only once per technique, after using it in the last session. The phrase set, the session schedule, payment schedule and amounts, and equipment were the same as those used in the preliminary study of the dwell-free methods (see Section 3.3).

4.3. Results of Phase 1

4.3.1. Text Entry Rate (WPM). Figure 14 (left) shows the average entry rate obtained per condition per session. Filteryedping was on average 37% faster than AltTyping across the six sessions.

A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the typing rate. There was a significant main effect for Technique ($F_{1,60} = 39.49$, $\eta^2 = 0.335$, $p < 0.001$); and a significant main effect for Session ($F_{5,60} = 2.57$, $\eta^2 = 0.109$, $p = 0.04$); but no significant interaction between Technique and Session ($F_{5,60} = 1.10$, $\eta^2 = 0.047$, $p = 0.37$). These results suggest that Filteryedping is faster than AltTyping, and that users become faster with practice.

Note that the AltTyping text entry rates measured in this study are lower than the ones reported by Räihä and Ovaska [2012]. We believe the difference lies in the way that the data was analyzed, where Räihä and Ovaska's analysis focused on providing an understanding of potential expert and error-free performance.

4.3.2. MSD Error Rate. Figure 14 (right) shows the average MSD error rate obtained per condition per session. AltTyping led to an average of five times more errors than Filteryedping across the six sessions.

A repeated measures analysis of variance was conducted to analyze the impact of Technique and Session on the MSD error rate. There was a significant main effect for Technique ($F_{1,60} = 8.95$, $\eta^2 = 0.120$, $p = 0.004$); but no significant main effect for Session ($F_{5,60} = 0.65$, $\eta^2 = 0.044$, $p = 0.66$); and no significant interaction between Technique and Session ($F_{5,60} = 0.47$, $\eta^2 = 0.031$, $p = 0.80$).

4.3.3. No Word Selected and Deleted Word Rates.. Figure 15 compares the no word selected rate and the deleted word rate obtained with Filteryedping using visual feedback in the preliminary study and in Phase 1. AltTyping supports writing and deleting one

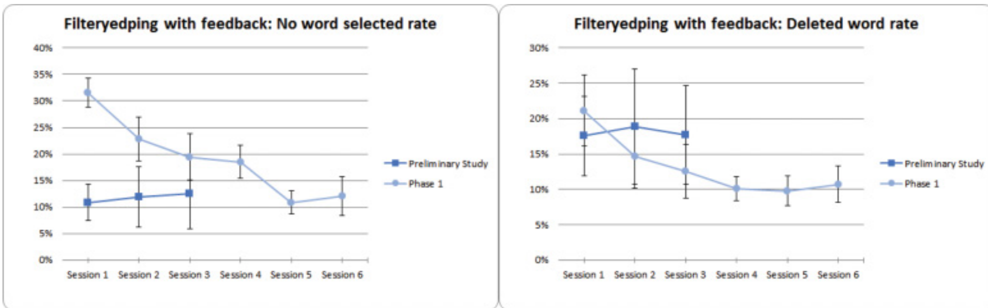


Fig. 15. No word selected rate (left) and deleted word rate (right) for Filteryedping with visual feedback for the preliminary study and Phase 1. Bars convey the standard error of the mean.

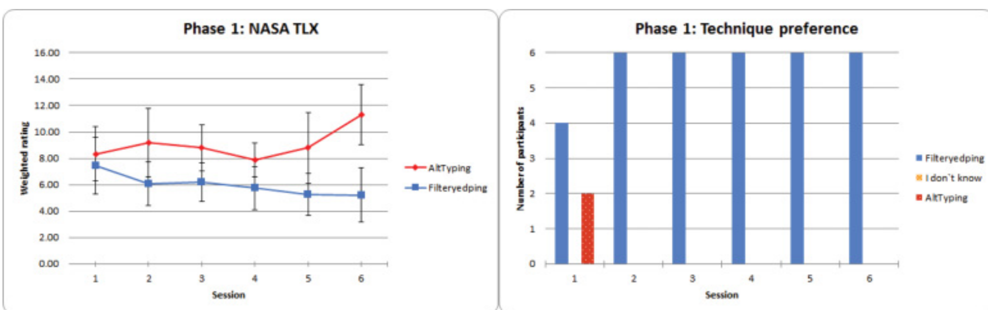


Fig. 16. Subjective results collected about AltTyping and Filteryedping: NASA TLX weighted ratings—including bars conveying the standard error of the mean (left) and responses to the question: "Which of the two techniques would you prefer to use?" after each session (right).

character at a time. Thus, these metrics cannot be applied to the AltTyping data in the same way.

Although changes made to the prototype between the preliminary study and Phase 1 may have improved the participants' input performance in general, the no word selected rate actually increased. In Phase 1, we modified the prototype to clear the stream of letters when the user looked outside the keyboard. We introduced this change so that if the user paused while typing a word—for example, because of an interruption—she may restart the word rather than try to remember which letters she has looked at already. However, we later noticed that participants often looked outside the keyboard while in the middle of a word to check whether they were typing the word correctly, and then continued typing. When they looked at the candidate list, the options provided were computed based only on the second half of the word and not on the whole word. This forced the users to select none of the suggested words and then retype the whole word again. As a result, we stopped clearing the stream of gazed at letters in the subsequent version of the prototype.

4.3.4. NASA TLX. Figure 16 (left) shows the average NASA TLX rating obtained per condition per session. For Session 1, the averages for Filteryedping and AltTyping were 7.57 and 10.70, respectively. For Session 6, the averages were 4.87 and 11.80. The results suggest that the workload of typing with a filtering-based technique is lower than the workload of typing with a dwell-based technique.

4.3.5. Technique Preference. In the post-test questionnaire, we asked the participants: "Which of the two techniques would you prefer to use?" Figure 16 (right) shows the participants' answers per session. Except in the first session, preference for Filteryedping was unanimous.

4.4. Phase 2

It has been previously demonstrated that evaluating eye gaze interaction techniques intended for people with physical disabilities using only participants without disabilities compromises the ecological validity of the research [Istance et al. 2012]. Thus, in Phase 2, we performed an iterative design and evaluation of the Filteryedping prototype with participants with ALS and DMD.

4.4.1. Procedure. In Phase 2, we aimed to understand how well user performance with our dwell-free eye typing approach generalizes when it is used by participants with motor impairments in natural settings. We conducted this phase of the study in each participant's home. Additionally, we wanted to learn about challenges that participants face when using a dwell-free eye typing approach and to explore ways of addressing those issues. Thus, we divided Phase 2 into two subphases and two case studies. In the two subphases, we asked these participants to type phrases using both Filteryedping and AltTyping. However, we expect that most, if not all, participants would have some experience with eye trackers and dwell-based text entry methods. As a result, because it would not be possible to collect data about how Phase 2 participants would learn to use and improve with both dwell-based and dwell-free eye typing methods over time, we included AltTyping only as a reference technique to help us understand how fast participants can currently type with a dwell-based method. Thus, these subphases did not consist of a fixed number of sessions as we had asked of Phase 1 participants. Still, this approach allowed us to understand how participants performed with and felt about dwell-free eye typing, and provided us with an opportunity to identify major challenges that they had when using Filteryedping. We then conducted two case studies, exploring how to address these challenges and directly testing those solutions with the participants.

Thus, Phase 2 consisted of the following subphases and case studies:

—*Subphase A*: Phase 2 began with three participants (P1–P3) using the same version of Filteryedping software that was used in Phase 1.

—*Subphase B*: Based on results learned in *subphase A*, we improved the system by adding two key features: (1) a *focus dwell* time that allows the system to determine when the user has changed focus between the keyboard and the candidate list, and (2) a *slow movement threshold* time that allows the system to determine if the user is performing a saccade with slow eye movements or not. We also improved the prototype by adding auditory feedback and space separating the options in the vertical menus, and by not clearing the stream of gazed at letters when the user looked outside the keyboard. We evaluated these changes to the system with participants P3–P6 in this subphase.

—*Focus Dwell Case Study*: In this case study, we explored the effect of different values for the *focus dwell* parameter. We wanted to gain an understanding of the relationship between this value and the different calibration quality that users might have with an eye tracker. The *Focus Dwell Case Study* was conducted with P5 and P6.

—*Slow Movement Threshold Case Study*: In this case study, we examined the effect of different values for the *slow movement threshold* parameter. We wanted to gain an understanding of how a well-adjusted threshold affects the input performance of users with slow eye movements. The *Slow Movement Threshold Case Study* was conducted with P3.

AltTyping was used in the two subphases the same way as in Phase 1. However, we used the dwell time of the end of the first session as reference. In *subphase A*, a reduction of 40ms after each session was applied. In *subphase B*, we used a reduction rate of 10% per session. This change was employed when we realized that a reduction by 40ms was too small when applied to the large dwell values used by the motor-disabled participants.

4.4.2. Participants. In total, we recruited six participants (two females), four of them with ALS and two with DMD, in different stages of the disease. All were fluent in English. In the following, we provide a brief description of each participant based on the observation of the researchers during trials. For each participant, we also present in parentheses their rating for the "Speech" subscale of the revised ALS Functional Rating Scale (ALSFRS-R) [Cedarbaum et al. 1999]. Possible values for the subscale are as follows: 4, normal speech processes; 3, detectable speech disturbance; 2, intelligible with repeating; 1, speech combined with nonvocal communication; and 0, loss of useful speech.

P1 is a woman in her 60s, living with ALS for decades. She can still communicate with her daughter by muttering (speech rating: 0). She also uses an Augmentative and Alternative Communication (AAC) device controlled by her foot. She has had about 1–3h of experience with eye trackers before this study. She uses glasses and had some difficulties calibrating the eye tracker. The layout of her own keyboard was organized by the frequency of use of the letters. As a result, she complained about having to look for the letters in a QWERTY layout. Because of her low familiarity with the QWERTY layout, she took part only in one session in s*ubphase A*.

P2 is a 62-yr-old female whose ALS onset occurred in 2003. She can still communicate verbally with difficulty (speech rating: 2), but could not move anything below the neck. She was very familiar with eye trackers. She did not use glasses and was able to successfully calibrate the eye tracker easily. She took part in four complete sessions in *subphase A*.

P3 is a 45-year-old man whose ALS onset occurred in 2005. Among the first three participants, he was the participant who showed the most physical debility in general. He currently uses an AAC device controlled by an eye tracker. During *subphase A*, he communicated with us most of the time by giving yes or no answers with his eyebrows (speech rating: 0). He uses glasses and had some difficulties calibrating the eye tracker. He completed six sessions in *subphase A*, six sessions in *subphase B*, and eight sessions in the *Slow Movement Threshold Case Study*. Forty-eight days elapsed between his last session in *subphase A* and his first session in *subphase B*. The disease had progressed noticeably during this period. An indication of the progression was that he had stopped being able to use his eyebrows to answer yes or no questions and started using discrete cheek movements.

P4 is a 76-year-old man whose ALS onset occurred in 2010. He could still produce sounds, but had lost the ability to produce useful speech (speech rating: 0). He still maintained good movements of his hands, which allowed him to control his wheelchair and use a tablet. He uses an AAC software system on his tablet to communicate. He has had about 1–3h of experience with eye trackers before this study. He uses bifocal glasses and had great difficulty calibrating the eye tracker. In fact, he successfully completed only one session in *subphase B* because of this problem. Two other session attempts did not materialize because the eye tracker could not be successfully calibrated to track his gaze.

P5 is a 33-year-old man, with DMD. He can still talk; however, his voice is weaker than that of most people and fails sometimes (speech rating: 2). He does not use corrective lenses and was able to achieve very good calibration results with the eye

Table I. Summary of the Description of Participants of Phase 2

Participant	Age	Gender	Disease	Speech rating	Previous experience with eye trackers	Quality of calibration with eye tracker	Corrective vision	Participation
P1	60s	F	ALS	0	1–3h	Some difficulties	Glasses	One session in *subphase A*
P2	62	F	ALS	2	Very familiar	Good results	None	Four sessions in *subphase A*
P3	45	M	ALS	0	Very familiar	Some difficulties	Glasses	Six sessions in *subphase A* Six sessions in *subphase B* Eight sessions in the *Slow Movement Threshold Case Study*
P4	76	M	ALS	0	1–3h	Great difficulty	Glasses	One session in *subphase B*
P5	33	M	DMD	2	Never used	Very good results	None	Eight sessions in *subphase B* Two sessions in the *Focus Dwell Case Study*
P6	37	M	DMD	2	Never used	Some difficulties	Glasses	Six sessions in *subphase B* One session in the *Focus Dwell Case Study*

tracker. He had never used an eye tracker before the study. He uses a head mouse every day, which enables general computer use, such as sending e-mails, navigating the web, and playing games. He completed eight sessions in *subphase B* and two sessions in the *Focus Dwell Case Study*.

P6 is a 37-year-old male, with DMD. He is still able to talk, but must often repeat what he wants to say (speech rating: 2). He uses glasses and had some difficulties calibrating the eye tracker. He can still issue some commands to his wheelchair using his right hand, such as leaning it forward and backward without any help. He had never used an eye tracker before the study. He uses a head mouse every day, primarily to play computer games. He completed six sessions in *subphase B* and one session in the *Focus Dwell Case Study*.

A summary of the preceding descriptionsis presented in Table I.

4.4.3. Findings from Subphase A. P1 achieved an entry rate of 3.26WPM with AltTyping and 0.95WPM with Filteryedping, and a MSD error rate of 1.67% with AltTyping and 7.90% with Filteryedping. Normally, P1 uses a keyboard layout based on the frequency of use of the letters. As a result, she had difficulty using a QWERTY-based layout. Unfortunately, we were not prepared to quickly adjust the keyboard layout in our prototype and so she only participated in one session. Despite her performance results, among these two techniques, she felt that she preferred Filteryedping over AltTyping.

In the four sessions that P2 took part in, she achieved an average entry rate of 6.50WPM with AltTyping and 7.90WPM with Filteryedping. Her average MSD error rate was 1.57% with AltTyping and 1.60% with Filteryedping. Figure 17 shows her-WPM and MSD error rate. In the posttest questionnaire, she showed a preference for Filteryedping in the first two sessions and AltTyping in the last two sessions.

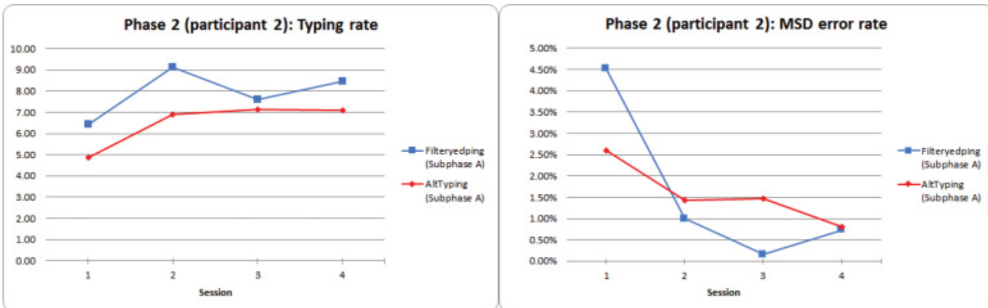


Fig. 17. Results of participant 2 of Phase 2: Text entry rate (left) and MSD error rate (right).

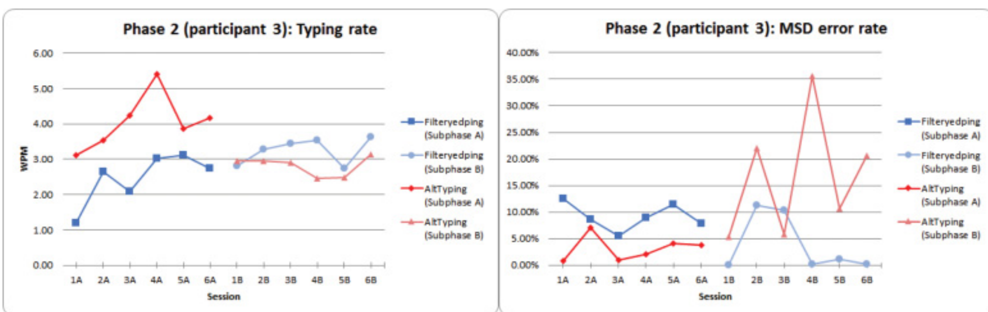


Fig. 18. Results of participant 3 of Phase 2: Text entry rate (left) and MSD error rate (right).

The advanced stage of P3's disease impacted the movements of his eyes. He has saccades with longer duration and slower velocity, which is a reported symptom for some ALS patients [Leveille et al. 1982]. With a dwell-based eye typing technique, this problem can be overcome by increasing the required dwell time. As a result, he finished his first session with AltTyping configured to use a dwell time of 1,120ms so that he could type with the system. The impact of his condition on Filteryedping proved to be a bigger challenge. While typing the word "appointment," for instance, during the saccade from the "a" to the "p," the letters "s," "d," "f," "t," "y," "h," "j," "u," "i," and "o" were also included in the stream of gazed at letters. A user without disabilities would be capable of "jumping" from the "a" to the "p" without selecting all or most of those letters. Naturally, because of the longer stream to be filtered, more candidate words might be suggested and the quality of the results would be negatively affected. The average avgPos for his Filteryedping sessions was 6.66, almost twice the average of avgPos for users from Phase 1. In all six sessions, he had an average of 2.47WPM using Filteryedping and 4.05WPM using AltTyping, and an average MSD error rate of 9.12% using Filteryedping and 3.11% using AltTyping. The NASA TLX weighted rate was 8.82 for Filteryedping and 6.77 for AltTyping, indicating that he considered the workload of the task of typing with Filteryedping higher. However, contradicting these metrics, his preference showed a trend toward Filteryedping. He preferred AltTyping in the first session, had no preference in the second and third sessions, and preferred Filteryedping in the last three sessions. Figure 18 shows hisWPM and MSD error rate with the initial prototype.

From this subphase, we learned about three aspects of the prototype that needed improvements:

(1) Some users may be familiar with or have a preference for alternate keyboard layouts. Thus, it is important for the prototype to allow the user to customize the layout. Because only a small number of people use alternate layouts, this change to the application was not a high priority. After the conclusion of Phase 2, we evolved the prototype to include the possibility of using the same keyboard layout that P1 uses. We did this as a proof of concept, to assure that different keyboard layouts could also be used with the Filteryedping technique. After about 100min of typing, in blocks of 5min, the first author reached 10.04WPM with the new layout, which was completely new to him.
(2) One clear issue with eye trackers is that the calibration quality varies a lot from user to user (see Table I). Low calibration quality results in excessive noise in the tracker data for some users and is an important source of errors when using the prototype. With Filteryedping, an error often occurs while participants type a letter located in the third line of the keyboard. The tracker sometimes mistakenly detects a single gaze point in the candidate list area, causing the system to mistake that the user has selected a candidate word, as briefly discussed in Section 3.5. Another problem often occurs while participants select a word in the candidate list. If the tracker detects a single gaze point in the keyboard area, the currently selected word is written and the candidate list disappears. Because it is not the intended word, the participant must then delete the word and type the intended word again. Many of these errors can be detected in the log files. If the participant's gaze stays in the candidate list for only a very short period of time, it suggests that an involuntary activation of the candidate list had occurred. We use a time threshold of 200ms because the average reading speed is 200–240ms/word [Just and Carpenter 1987]. Thus, 200ms represents the shortest amount of time possible for the user to intentionally activate the candidate list, select the intended word, and then continue to type again. The candidate list was open for less than 200ms 38.0% of the time. Noisy eye tracker data inside the keyboard did not cause a problem because extra letters included in the stream are simply filtered away by our algorithm later. Similarly, noisy eye tracker data inside the candidate list is not a strong problem either because of the spaces separating the suggestions. As a result, we introduce a short *focus dwell* time to help the system determine a switch in focus from the keyboard to the candidate list and vice versa.
(3) The saccades with longer duration and slower velocity that we observed from P3 motivated us to introduce a *slow movement threshold*. In analyzing our logs, we learned that while P3's gaze moved from one desired letter to the next, his slow eye movements caused the eye tracker to regularly detect gaze points between them. This could be verified by analyzing the three participants' data; we compared the average number of letters added to the stream of letters that is filtered for every typed word. P3 gazed on average at 44.4 letters per word, while P1—who is not familiar with the QWERTY layout—and P2 gazed 34.6 and 16.6 letters per word, respectively. The logs also showed that the number of gaze points at the desired keys is higher than the gaze points at keys that are crossed while the eye moves toward the target. This means that a small threshold value can be used to distinguish a letter that should be added to the stream of letters from those gazed at because of slow eye movements. This slow movement threshold differs from a regular dwell in two ways: (1) the slow movement threshold is an implicit mechanism to differentiate gaze points detected because of slow eye movements between fixation points, while a regular dwell is an explicit user action to perform a selection; (2) it can be set to be much shorter than a regular dwell (and imperceptibly fast) because our Filteryedping algorithm mitigates the penalty for falsely adding letters to the stream of letters.

4.4.4. Findings from Subphase B. Based on findings from *subphase A*, we modified the prototype to improve it. We used the version described in Section 3.1, adding the focus dwell and slow movement threshold features described previously. To calculate how long the slow movement threshold should be for each user, we measure the amount of time the user takes to alternate her gaze between two points located about 25cm apart from each other three times (to travel about 75cm). We performed this measurement before the first practice block of the first session for each user. It resulted in a slow movement threshold of 167ms (five sample points reported by the eye tracker) for P3, 67ms (two points) for P4, and no slow movement threshold time (i.e., a single point over any key adds that letter to the stream) for P5 and P6. The focus dwell time that we used for all participants was 100ms, which is the time it takes for our 30-Hz eye tracker to report three points.

It is hard to directly compare the results from *subphase B* with results from *subphase A* because of the small number of participants involved and the large variability among them. However, we believe that, based on what we observed, the inclusion of the focus dwell, the slow movement threshold, and other changes to the prototype improved the user experience.

One indication that the Filteryedping prototype actually improved comes from the results achieved by P3. He was the only participant that took part in both *subphases A and B* (six sessions each). In *subphase B*, the entry rate that he achieved with AltTyping was 2.81WPM, which was about 70% of the rate he achieved with AltTyping in *subphase A*. His MSD error rate was 16.69%—more than five times his error rate in *subphase A*. The decline in the results likely was caused by the disease progression in the 48 days separating his last session in *subphase A* and his first session in *subphase B*, as he suggested to us. Even with a slower eye movement, he was able to improve his typing rate and reduce his error rate using Filteryedping (from 2.47WPM and 9.12% to 3.24WPM and 3.86%, respectively). He finished his first session with AltTyping using a regular dwell of 1,240ms and followed the scheduled reduction of 10% each session, leading to a dwell of 910ms in the fourth session. That was already too fast for him and we decided for the last two sessions to repeat the dwell times with which he achieved better performance in *subphase A*: 1,040 and 1,000ms. Figure 17 shows a per session comparison of those two metrics. In this subphase, he preferred Filteryedping in all the sessions. Analysis of his responses to the NASA TLX form also indicates a lower workload when using Filteryedping (6.91) compared to AltTyping (9.87).

In the one session that P4 took part, he achieved an entry rate of 0.82WPM with AltTyping and 0.63WPM with Filteryedping, and a MSD error rate of 4.03% with AltTyping and 36.38% with Filteryedping. Despite this, he declared that, among these two techniques, he would prefer to use Filteryedping. He finished the AltTyping session with the dwell time set at 1,120ms. An issue that he faced was the small font size used in both interfaces. He sometimes adjusted the position of his head in order to be able to read what was on the screen when using both prototypes. Larger font sizes would have been welcomed.

For P5, he was most comfortable with his head position tilted a little to the left. Surprisingly, the difference in the height of his two eye positions did not prevent the eye tracker from being able to track his eye gaze well. In eight sessions, he achieved an average entry rate of 10.61WPM with AltTyping and 11.40WPM with Filteryedping. He finished his first session with AltTyping using a dwell time of 620ms and followed the scheduled reduction of 10% each session, leading to a dwell time of 300ms in the eighth session. Figure 19 (left) shows hisWPM in *subphase B*. His performance in the fifth session might have been affected by low concentration due to conversations with a personal visitor during the session. His average MSD error rate was 0.30% for AltTyping and 0.45% for Filteryedping. He preferred AltTyping in Sessions 1, 2, 5, and

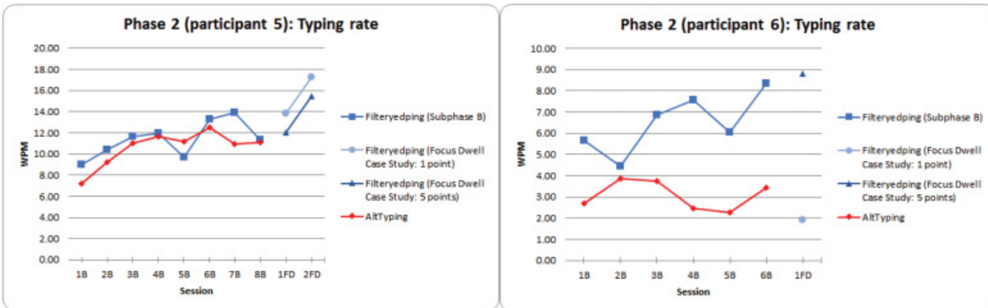


Fig. 19. Text entry rate of participants 5 (left) and 6 (right) of Phase 2. *Subphase B* used a value of about 100ms for the focus dwell parameter, which is the time it takes for the eye tracker to report three points. *Focus Dwell Case Study* used a single point or five points (167ms).

Table II. No Word Selected Rate (left) and Deleted Word Rate (right) for Filteryedping in Subphase A and Subphase B

No word selected rate							
	Participants						
	Subphase A			*Subphase B*			
Sessions	P1	P2	P3	P3	P4	P5	P6
1	76%	52%	41%		5%	1%	14%
2		42%	39%	0%		1%	14%
3		33%	46%	0%		0%	5%
4		38%	22%	0%		1%	6%
5			62%			3%	8%
6			36%	0%		0%	7%
7						0%	
8						2%	
AVG	76%	41%	41%	0%	5%	1%	9%

Deleted word rate							
	Participants						
	Subphase A			*Subphase B*			
Sessions	P1	P2	P3	P3	P4	P5	P6
1	93%	50%	85%		74%	22%	26%
2		23%	26%	6%		16%	42%
3		22%	63%	5%		7%	33%
4		28%	29%	13%		10%	17%
5			45%			11%	24%
6			59%	11%		8%	14%
7						8%	
8						7%	
AVG	93%	31%	51%	9%	74%	11%	26%

6, and Filteryedping in Sessions 7 and 8. He had no preference in Session 3 and did not provide a response to this question in Session 4. Analysis of his responses to the NASA TLX form indicates an equivalent workload when using Filteryedping (4.03) and AltTyping (4.25).

In six sessions, P6 achieved an average entry rate of 3.08WPM with AltTyping and 6.49WPM with Filteryedping, and a MSD error rate of 0.23% with AltTyping and 1.76% with Filteryedping. He finished his first session with AltTyping using a regular dwell time of 840ms and followed the scheduled reduction of 10% each session, leading to a dwell time of 500ms in the sixth session. Figure 19 (right) shows his WPM in *subphase B*. From sessions 2 to 6, he expressed a preference for using Filteryedping over AltTyping. He had no preference in the first session. Also for P6, the NASA TLX results indicated an equivalent workload; both techniques received a weighted rating of 3.81.

Improvements made to the Filteryedping prototype between *subphase A* and *subphase B* also led to decreases in both the no word selected rate and the deleted word rate, as shown in Table II.

From this subphase, we learned the following:

(1) Including the focus dwell reduced the occurrence of problems introduced by noisy eye tracker data. In particular, the candidate list was opened for less than 200ms only 0.3% of the time—it was 38.0% in *subphase A*. Although it helped participants who had trouble calibrating the eye tracker, it may not have benefited those who

Fig. 20. Cropped screenshots of the calibration check screen of participants 5 (left) and 6 (right) of Phase 2. The scattered points obtained by P6 indicate a lack of precision of the eye tracker after his calibration. Displacement of the points regarding the target also indicates low accuracy, in this case.

already had good calibration results. The dwell time required to change focus could have reduced their input speed.

(2) Including the slow movement threshold parameter clearly helped P3—the only participant with slow eye movements. His input performance continued to improve with Filteryedping, but declined with AltTyping as his ALS condition progressed. An analysis of our log showed that with the introduction of the slow movement threshold parameter, the average number of gazed at letters per word was now lowered to 19.5 for P3. This is smaller than the 44.4 letters per word for P3 from *subphase A* and the 34.6 letters per word for P1 from *subphase A*—who is unfamiliar with the QWERTY layout. However, this value is still higher than the 16.6 letters per word for P2 from *subphase A* and 16.4 letters per word for the other participants from *subphase B*. This suggests that perhaps the threshold value used for P3 could still be adjusted to improve his input performance even more.

4.4.5. Findings from the Focus Dwell Case Study. Results of *subphase B* provided good indication that the focus dwell helps to minimize the impact of poor eye tracker calibration in the Filteryedping interface. At the same time, however, this feature seemed to stifle the input performance for those who obtained good eye tracker calibration. To further validate this, we performed a case study with P5 and P6 using Filteryedping with different focus dwell values. While P5 usually had good calibration results, P6 experienced poor calibration, as shown in Figure 20. The use of different focus dwell values with participants who had different calibration quality would allow us to evaluate the relationship between the parameter and calibration quality.

In *subphase B*, a focus dwell was performed when the eye tracker reported three consecutive gaze points in either the keyboard area or the candidate list. In this case study, we compared the use of a single gaze point reported by the eye tracker to perform a focus dwell (which is the configuration used by participants in *subphase A*) against the use of five gaze points. Participants used both configurations in each session. We expected that P5 would have better results with the single-point configuration, because he would be able to avoid the delay required for changing focus between the keyboard and the candidate list. On the other hand, we expected that P6 would have better results with the five-point configuration, because it would reduce the amount of errors caused by the poor eye tracker calibration he typically had in comparison to P5.

P5 took part in two sessions. In the first session, he started with the one-point configuration. The opposite ordering was used in the second session. After the first session, he said he could not notice the difference between the two configurations. After the second session he said that the difference was perceptible and that he preferred

Fig. 21. Optional animation indicating the timing of the gaze over a letter in relation to the slow movement threshold.

using the one gaze point focus dwell configuration. In both sessions, he was faster with the one-point than the five-point configuration, as we had expected. He reached an average of 15.54WPM using the one-point configuration and an average of 13.77WPM using the five-point configuration (Figure 19, left). In *subphase B,* P5 reached a max average of 13.92WPM (session 7) using the three-point configuration. Thus, for P5, when the focus dwell time is reduced, he is able to input text faster.

P6 took part in only one session, but the difference in the results was even more evident. He started with the one-point configuration and reached 1.91WPM using it and 8.82WPM using the five-point configuration (Figure 19, right), and indicated a clear preference for the five-point configuration. In *subphase B,* P6 reached a max average of 8.36WPM (session 5) using the three-point configuration. Thus, his results were also in agreement with what we had expected.

From *subphase B* and this case study, we learned that dwell-free text input methods which have an input area and a candidate list must take into consideration the effect of the different levels of precision and accuracy at which an eye tracker can detect the user's gaze. When the eye tracker detects the user's gaze with low precision and accuracy, gaze points near the border between the input area and the candidate list can falsely change the focus between these two components. One way to address this issue is to increase the distance between these components; however, this means the interface for the text input method would need to consume more screen space. A different approach is to introduce a focus dwell. Findings from *subphase B* demonstrate that this can reduce false changes in the focus between these two components. Findings from this case study further validate this approach. However, it also highlights that the mechanisms for addressing this problem should be tailored to the level of precision and accuracy at which an eye tracker detects the user's gaze. When the eye tracker can detect a user's gaze with high precision and accuracy, adding a large distance between these two components or introducing a long focus dwell time can lower the user's input performance. Thus, future research should explore ways to customize the interface of dwell-free text input methods based on the level of precision and accuracy at which the eye tracker is detecting a user's gaze.

4.4.6. Findings from the Slow Movement Threshold Case Study. The introduction of a slow movement threshold seemed to allow P3 to improve his input performance during *subphase B*. However, while observing him type, we noticed that the dwell time calculated by our simple method of measuring the amount of time the participant takes to alternate his gaze between two points located about 25cm apart from each other for three times may not have been a large enough value. For P3, this method returned a value of five gaze points. At that threshold value, there were still many letters included in the stream whenever he performed a saccade to a distant letter. Thus, we performed a case study with P3 exploring the effect of different slow movement threshold values. P3 was the only participant with a slow eye movement, and thus was the sole participant in this case study.

After testing the use of large values as the slow movement threshold on ourselves, we learned that perhaps additional visual feedback would be useful. Thus, we implemented an animation, very similar to the one used in AltTyping, to indicate the timing of the gaze over a letter as it relates to the slow movement threshold (Figure 21). When the animation completes a 360-degree arc, the whole key becomes pink, as before, and the letter is included in the stream. Nothing changed with the animation for the regular dwell.

In this case study, we adjusted the threshold value based on his feedback after each session. Each session consisted of a single block and the block duration was reduced from 20 to 12min to avoid fatigue. A break of at least 30min was taken between the sessions. The case study occurred over two consecutive days, with four sessions completed per day. Table III presents all the configurations (threshold values and presence of animated feedback or not) that P3 tested, the typing rates obtained, and the options for the next configuration that were presented to him after each session.

Following this iterative evaluation process, we learned that for P3 a slow movement threshold of about 300–400ms (9–12 points) without additional visual feedback allowed him to achieve a comparable input rate to what he obtained with a larger slow movement threshold value. In the last three sessions, the average number of gazed letters per word was 14.6. His preference was for a shorter threshold value without visual feedback. After the last session, he used the prototype again to give us a final comment. He typed: “I really liked the software.” Additional research effort is needed to develop methods to automatically characterize eye movement velocity and determine the appropriate slow movement threshold value.

5. DISCUSSION

By running experiments with people without disabilities and motor-disabled individuals, we showed that Filteryedping is a promising eye typing technique, both in terms of objective metrics and subjective metrics. Studies with participants without disabilities helped us collect an amount of data that would be difficult to obtain with only the target population. Motor-impaired participants are harder to recruit and may not be able to participate very easily due to schedule and location constraints. At the same time, a small study with motor-impaired participants was important to include because it helped us understand how the results with participants without disabilities generalizes to the target user population. Furthermore, it helped us learn about key issues that must be addressed in our prototype and explore solutions to those problems.

Along the study, we identified several points that could be improved. An important observation was the difficulty participants faced in detecting an error in the typed text. As they use their eyes to type, constantly checking the results could result in a slower typing rate. We perceived that the inclusion of auditory feedback brought more confidence to users, letting them know that an error was committed without requiring constant checking of the text area.

One of the biggest findings was that the calibration quality varies a lot and is extremely user dependent. In general, participants who did not use corrective lenses seemed to achieve a better calibration than those with corrective lenses. In the preliminary study, two potential participants could not be included in the experiment because the eye tracker could not be calibrated to work with those individuals. One of those participants uses glasses and the other uses contact lenses. In the second phase, another participant (a glasses wearer) had similar problems.

In the early experiments, we noticed that a good strategy for avoiding the accidental selection of a wrong word would be to include some space to separate the options in the candidate list, even though this reduces the number of candidates shown per page. Later, we verified that the user was writing words accidentally, even when she did not want to write a word, because excessive noise in the tracker data was causing the system to mistake that the user wanted to open the candidate list.

The variability in the precision and accuracy of the calibration and tracking must be taken into consideration in the development of any gaze-based interface. We included

Table III. Configurations (Slow Movement Threshold Values and Presence of Animated Feedback or Not) Tested and Typing Rates Obtained by P3 During the *Slow Movement Threshold Case Study*

Session	Configuration	WPM	Letters added to filtered text stream per word	Options presented to the participant for the next session with P3's preference highlighted in bold
1	700ms (21 gaze points) with feedback	3.23	8.6	(1) 700ms without feedback (2) A slower threshold with feedback **(3) A faster threshold with feedback** And then: (1) 600ms with feedback **(2) 500ms with feedback** (3) 400ms with feedback
2	50ms (15 gaze points) with feedback	3.79	12.94	(1) 500ms without feedback **(2) 600ms with feedback** (3) 400ms with feedback
3	600ms (18 gaze points) with feedback	2.96	12.42	(1) 600ms or slower threshold **(2) 500ms or faster threshold** And then: **(1) 500ms without feedback** (2)400ms with feedback (3) 400ms without feedback
4	500ms (15 gaze points) without feedback	3.65	10.09	**(1) 400ms without feedback** (2) 400ms with feedback
5	(The wrong configuration was tested) One gaze point without feedback	1.89	31.52	
6	400ms (12 gaze points) without feedback	3.39	12.11	(1) 500ms or slower threshold **(2) 400ms or faster threshold** And then: (1) 400ms with feedback **(2) 300ms without feedback** (3) 433ms without feedback (4) 466ms without feedback
7	300ms (Nine gaze points) without feedback	3.72	17.09	(1) Faster than 300 ms **(2) Slower than 300ms (but faster than 400ms)** (3) Slower than 400ms And then: **(1) 333ms without feedback** (2) 367ms without feedback
8	333ms (10 gaze points) without feedback	2.04	14.92	

a focus dwell parameter in our prototype to help overcome the problem of accidentally selecting wrong words from the candidate list. A case study on the effect of the focus dwell parameter corroborates our belief that users with poor eye tracker calibration require longer focus dwell values. An automatic metric that indicates the quality of the calibration should be developed and included to automatically configure the interface. Besides the focus dwell, a parameter that determines how close the keys on the keyboard should be to each other has the potential to improve users' performance. Users with good eye tracker calibration would benefit from the extra screen space enabled by a smaller virtual keyboard.

The precision and accuracy of gaze points reported by the eye tracker was not the only source of variation among users. One of the participants with ALS demonstrated impairments in the velocity of his saccades. This type of problem may affect not only ALS patients [Leveille et al. 1982], but also DMD patients [Lui et al. 2001]. Ashtiani and MacKenzie [2010] previously mentioned this limitation while advocating the development of a text entry technique based on blinking instead of eye movements for the severely motor impaired. However, the introduction of the slow movement threshold parameter seems to help with this problem. One of the advantages of the slow movement threshold solution is that it may be adjusted as the disease progresses, requiring no abrupt changes in the interface to which the user is already familiarized. Furthermore, this parameter allows the system to differentiate slow eye movements from when the user's eye gaze has reached a target. Although from an external observer's perspective, it may seem similar to a normal dwell, from the perspective of a user with slow eye movement, this parameter is still smaller than what a normal dwell threshold value is for them. The mechanism itself does not require any explicit user action. As result, some letters along the eye movement toward the target letter might still be added to the text stream, but our key filtering–based approach minimizes the effect of such errors and does not require the user to delete unintended letters. Thus, from the user's perspective, this input approach differs from a dwell-based approach. P3's preference for Filteryedping over AltTyping lies in the fact that he did not need to perform full dwells over keys that he wanted to enter. This difference also enabled him to be faster with Filteryedping.

We discussed with P5 and P6, the two participants who use a head mouse, whether they think that Filteryedping could be used with a head mouse instead of with an eye tracker. Both of them said that they think it would be possible. Additionally, in a discussion with P5, we concluded that the dwell-based eye typing could be even more integrated to Filteryedping. If a high threshold value is used for the normal dwell—2s, for example—we would practically eliminate the possibility of selecting a letter by mistake. Then, we could enable the real time dwell-based eye typing; that is, without having to search for the dwelled word in the first position of the candidate list.

Finally, the difference in performance between participants with and without disabilities needs to be examined. Perhaps one reason is the presence and progression of ALS or DMD. However, there were several other factors that could have contributed to the performance difference. One possible reason is the difference in the study conditions. Individuals without disabilities took part in a controlled laboratory study, while participants with ALS and DMD typed in their homes. This facilitated their participation, but at the same time hindered us from controlling for other factors, such as lighting, noise, and other distractions. Another possible reason might be the difference in age. Motor-disabled participants were older (avg = 53.0 years old) than participants without disabilities (avg = 22.8 years old). However, despite the difference in the rate of typing, the results were consistent in showing that both participants with and without motor disabilities liked dwell-free eye typing and performed better with a dwell-free method than with a dwell-based approach.

6. CONCLUSIONS

In an effort to increase the communication power of severely motor-impaired individuals, we have introduced and evaluated a dwell-free eye typing technique that allows the user to enter text without requiring a long fixation over a key to input that letter. With Filteryedping, the user types simply by gazing sequentially at all of the letters in a word. It overcomes the Midas touch problem by filtering out letters that were gazed at accidentally by the user. Filteryedping then creates a ranked list of candidates based on the frequency and length of the words in a corpus.

Our first evaluation step was to compare it with another plausible way that dwell-free eye typing could be implemented—a shape-based approach that has been shown as an effective typing method for touch-based interfaces. The shape-based technique identifies candidate words by comparing the shape of the path covered by the gaze with the optimal shape of each word in a dictionary. It has been suggested and investigated as a good candidate for eye typing [Kristensson and Vertanen 2012; Hoppe et al. 2013]. Objective and subjective results from a study with 12 users without disabilities testing our implementation of the two dwell-free techniques indicated that Filteryedping is a satisfactory method to support eye typing.

We then evaluated the Filteryedping technique along with AltTyping, the fastest dwell-based eye typing tool reported in the literature. The evaluation was divided into two phases, the first with 10 participants without disabilities and the second with six severely motor-disabled individuals. Results of the first phase showed that Filteryedping enabled participants to reach an average of 14.75WPM in six sessions (about 2h of typing per user) and an average of 15.95WPM in the last session, and 10.77WPM with AltTyping in six sessions and an average of 11.71WPM in the fastest session (session 5, using a dwell time of 290ms). The fastest participant typed at 19.28WPM with Filteryedping in the sixth session. The fast typing rate with Filteryedping was not reached at the expense of accuracy. The average MSD error rate in six sessions was 0.64%. Even before the last improvements, Filteryedping fared better than AltTyping not only in these objective metrics, but also in terms of user's preference and workload. The second phase iteratively evolved the Filteryedping technique to address problems that we learned by evaluating the method with participants with ALS and DMD in natural settings. We implemented and conducted case studies of two important parameters (focus dwell and slow movement threshold), which allow the technique to be adapted to different user needs.

Our goal here was to examine how well a dwell-free approach would work in practice against a dwell-based one. Our preliminary study was conducted to identify which of our two implementations of possible dwell-free techniques could be used to test against AltTyping. The results of the preliminary study provided an understanding about the effectiveness of *shape-based* and *key filtering–based* eye typing methods. Although in the main study we tested the key filtering–based approach against AltTyping, the shape-based approach can also be employed to support dwell-free eye typing. Future work should further explore how to adapt SHARK[2] for eye typing and evaluate it. The results of our main study show that dwell-free typing is an approach that could potentially be faster than dwell-based and furthermore is lower in workload and preferred by participants. Our main study also uncovered challenges that participants with severe motor impairments face when using a dwell-free eye typing approach and explored ways of addressing those issues. The *slow eye movement threshold* was a functionality introduced to overcome one of these challenges. We believe that this functionality is also useful for Filteryedping users without slow eye movements who use high-frequency eye trackers; this should be investigated in future work. To improve the external validity of the technique, future work should also evaluate Filteryedping using composition tasks instead of transcription tasks by following procedures such as the ones suggested by Vertanen and Kristensson [2014].

ACKNOWLEDGMENTS

We would like to thank all the participants of the experiment, the family members and nurses that provided support, and Juliana Cobre and Dorival Leão for helping with statistical analysis. This study was covered under UNCC's Institutional Review Board Protocol #14-01-36.

REFERENCES

Behrooz Ashtiani and I. Scott MacKenzie. 2010. BlinkWrite2: An improved text entry method using eye blinks. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications (ETRA'10)*. ACM, New York, NY, 339–345. DOI:http://doi.acm.org/10.1145/1743666.1743742

Nikolaus Bee and Elisabeth André. 2008. Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *Perception in Multimodal Dialogue* Systems. Lecture Notes in Computer Science 5078. Springer, Berlin, 111–122. DOI:http://link.springer.com/chapter/10.1007/978-3-540-69369-7_13

Jesse M. Cedarbaum, Nancy Stambler, Errol Malta, Cynthia Fuller, Dana Hilt, Barbara Thurmond, and Arline Nakanishi. 1999. The ALSFRS-R: A revised ALS functional rating scale that incorporates assessments of respiratory function. BDNF ALS Study Group (Phase III). *J. Neurol. Sci.* 169, 1–2 (October 1999), 13–21.

Tuhin Chakraborty, Sayan Sarcar, and Debasis Samanta. 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 (CHI EA'14)*. ACM, New York, NY, 1573–1578. DOI:http://doi.acm.org/10.1145/2559206.2581265

Mark Davies. 2008. The corpus of contemporary American English (COCA): 450 million words, 1990–2012. Retrieved from http://www.americancorpus.org.

Sabrina Hoppe, Markus Löchtefeld, and Florian Daiber. 2013. Eype—Using eye-traces for eye-typing. In *Proceedings of the Workshop on Grand Challenges in Text Entry (CHI'13)*.

Poika Isokoski. 2000. Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research and Applications (ETRA'00)*. ACM, New York, NY, 15–21. DOI:http://doi.acm.org/10.1145/355017.355020

Howell Istance, Stephen Vickers, and Aulikki Hyrskykari. 2012. The validity of using non-representative users in gaze communication research. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA'12)*, Stephen N. Spencer (Ed.). ACM Press, New York, NY, 233–236. DOI:http://doi.acm.org/10.1145/2168556.2168603

Marcel Adam Just and Patricia Anderson Carpenter. *The Psychology of Reading and Language Comprehension*. Allyn & Bacon, Boston, 1987.

Per Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA'12)*, Stephen N. Spencer (Ed.). ACM, New York, NY, 241–244. DOI:http://doi.acm.org/10.1145/2168556.2168605

Per-Ola Kristensson and Shumin Zhai. 2004. SHARK^2: A large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology (UIST'04)*. ACM, New York, NY, 43–52. DOI:http://doi.acm.org/10.1145/1029632.1029640

Manu Kumar. 2007. Gaze-Enhanced User Interface Design. Stanford University. Retrieved from http://hci.stanford.edu/research/GUIDe/publications.html.

Albert Leveille, Joseph Kiernan, James A. Goodwin, and Jack Antel. 1982. Eye movements in amyotrophic lateral sclerosis. *Arch. Neurol.* 39, 11 (November 1982), 684–86.

Fausta Lui, Sergio Fonda, Luciano Merlini, and Ruggero Corazza. 2001. Saccadic eye movements are impaired in Duchenne muscular dystrophy. *Doc. Ophthal. Adv. Ophthal.* 103, 3 (November 2001), 219–228.

Ian Scott MacKenzie and Robert William Soukoref. 2002. A character-level error analysis technique for evaluating text entry methods. In *Proceedings of the 2nd Nordic Conference on Human-Computer Interaction (NordiCHI'02)*. ACM, New York, NY, 243–246. DOI:http://doi.acm.org/10.1145/572020.572056

Ian Scott MacKenzie and Robert William Soukoref. 2003. Phrase sets for evaluating text entry techniques. In CHI'03 Extended Abstracts on Human Factors in Computing Systems *(CHI EA'03)*. ACM, New York, NY, 754–755. DOI:http://doi.acm.org/10.1145/765891.765971

Ian Scott MacKenzie and Xuang Zhang. 2008. Eye typing using word and letter prediction and a fixation algorithm. In *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications (ETRA'08)*. ACM Press, New York, NY, 55–58. DOI:http://doi.acm.org/10.1145/1344471.1344484

Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'09)*. ACM, New York, NY, 357–360. DOI:http://doi.acm.org/10.1145/1518701.1518758

Carlos H. Morimoto and Arnon Amir. 2010. Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications (ETRA'10)*. ACM, New York, NY, 271–274. DOI:http://doi.acm.org/10.1145/1743666.1743730

Medical News Today. 2009. What is motor neuron disease? What is amyotrophic lateral sclerosis (ALS), or Lou Gehrig's disease? Medical News Today. DOI:http://www.medicalnewstoday.com/articles/164342.php

National Human Genome Research Institute. 2013. *Learning About Duchenne Muscular Dystrophy*. Retrieved from Genome.gov. DOI:http://www.genome.gov/19518854

National Institute of Neurological Disorders and Stroke (NINDS). 2012. *Motor Neuron Diseases Fact Sheet. National Institute of Neurological Disorders and Stroke (NINDS).* Retrieved from http://www.ninds.nih.gov/disorders/motor_neuron_diseases/detail_motor_neuron_diseases.htm.

Patient.co.uk. 2013. Duchenne Muscular Dystrophy. Patient.co.uk, http://www.patient.co.uk/health/duchenne-muscular-dystrophy-leaflet (accessed July 17, 2014).

Kari-Jouko Räihä and Saila Ovaska. 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 (CHI'12)*. ACM, New York, NY, 3001–3010. DOI:http://doi.acm.org/10.1145/2207676.2208711

Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An evaluation of Dasher with a high-performance language model as a gaze communication method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces (AVI'14)*. ACM, New York, NY, 169–176. DOI:http://doi.acm.org/10.1145/2598153.2598157

Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An efficient dwell-free eye gaze-based text entry system. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction (APCHI'13)*. ACM, New York, NY, 215–220. DOI:http://doi.acm.org/10.1145/2525194.2525288

Watcharin Tangsuksant, Chittaphon Aekmunkhongpaisal, Patthiya Cambua, Theekapun Charoenpong, and Theerasak Chanwimalueang. 2012. Directional eye movement detection system for virtual keyboard controller. In *Proceedings of the IEEE Biomedical Engineering International Conference (BMEiCON'12)*. 1–5. DOI:10.1109/BMEiCon.2012.6465432

Mario H. Urbina and Anke Huckauf. 2007. Dwell time free eye typing approaches. In *Proceedings of the 3rd Conference on Communication by Gaze Interaction (COGAIN'07)*. Leicester, UK, 65–70.

U.S. National Library of Medicine. 2012. Duchenne Muscular Dystrophy. MedlinePlus. http://www.nlm.nih.gov/medlineplus/ency/article/000705.htm. Accessed 17/Jul/2014.

Keith Vertanen and Per Ola Kristensson. 2014. Complementing text entry evaluations with a composition task. *ACM Trans. Comput.-Hum. Interact.* 21, 2, Article 8 (February 2014), 33 pages. DOI:http://doi.acm.org/10.1145/2555691

David J. Ward, Alan F. Blackwell, and David J. C. MacKay. 2000. Dasher—A data entry interface using continuous gestures and language models. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology (UIST'00)*. ACM, New York, NY, 129–137. DOI:http://doi.acm.org/10.1145/354401.354427

Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact.* 13, 4 (December 2006), 458–489. DOI:http://doi.acm.org/10.1145/1188816.1188819

Jacob O. Wobbrock, Brad A. Myers, and John A. Kembel. 2003. EdgeWrite: A stylus-based text entry method designed for high accuracy and stability of motion. In *Proceedings of the 16th Annual ACM Symposium on User Interface Software and Technology (UIST'03)*. ACM, New York, NY, 61–70. DOI:http://doi.acm.org/10.1145/964696.964703

Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. 2008. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research and Applications (ETRA'08)*. ACM, New York, NY, 11–18. DOI:http://doi.acm.org/10.1145/1344471.1344475

Xiaoyu (Amy) Zhao, Elias D. Guestrin, Dimitry Sayenko, Tyler Simpson, Michel Gauthier, and Milos R. Popovic. 2012. Typing with eye-gaze and tooth-clicks. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA'12)*, Stephen N. Spencer (Ed.). ACM, New York, NY, 341–344. DOI:http://doi.acm.org/10.1145/2168556.2168632

Received September 2014; revised November 2014; accepted January 2015