RESEARCH-ARTICLE

Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times

MARTEZ E. MOTT, University of Washington, Seattle, WA, United States

SHANE WILLIAMS, Microsoft Research, Redmond, WA, United States

JACOB O. WOBBROCK, University of Washington, Seattle, WA, United States

MEREDITH RINGEL MORRIS, Microsoft Research, Redmond, WA, United States

Published: 02 May 2017

Citation in BibTeX format

CHI '17: CHI Conference on Human Factors in Computing Systems
May 6 - 11, 2017
Colorado, Denver, USA

Conference Sponsors:
SIGCHI

CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (May 2017)
https://doi.org/10.1145/3025453.3025517
ISBN: 9781450346559

Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times

Martez E. Mott[1,2], Shane Williams[1], Jacob O. Wobbrock[2] and Meredith Ringel Morris[1]
[1]Microsoft Research, Redmond, WA, USA
[2]The Information School | DUB Group, University of Washington, Seattle, WA, USA
memott@uw.edu, shanewil@microsoft.com, wobbrock@uw.edu, merrie@microsoft.com

ABSTRACT
We present *cascading dwell gaze typing*, a novel approach to dwell-based eye typing that dynamically adjusts the dwell time of keys in an on-screen keyboard based on the likelihood that a key will be selected next, and the location of the key on the keyboard. Our approach makes unlikely keys more difficult to select and likely keys easier to select by increasing and decreasing their required dwell times, respectively. To maintain a smooth typing rhythm for the user, we *cascade* the dwell time of likely keys, slowly decreasing the minimum allowable dwell time as a user enters text. Cascading the dwell time affords users the benefits of faster dwell times while causing little disruption to users' typing cadence. Results from a longitudinal study with 17 non-disabled participants show that our dynamic cascading dwell technique was significantly faster than a static dwell approach. Participants were able to achieve typing speeds of 12.39 WPM on average with our cascading technique, whereas participants were able to achieve typing speeds of 10.62 WPM on average with a static dwell time approach. In a small evaluation conducted with five people with ALS, participants achieved average typing speeds of 9.51 WPM with our cascading dwell approach. These results show that our dynamic cascading dwell technique has the potential to improve gaze typing for users with and without disabilities.

Author Keywords
Gaze typing; eye typing; text entry; accessibility.

ACM Classification Keywords
H.5.2. [Information interfaces and presentation]: User interfaces – *input devices and strategies*. K.4.2. [Computers and society]: Social issues – *assistive technologies for persons with disabilities*.

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025517

Figure 1. Participants completed 8 sessions of gaze typing tasks using our software. The device shown above is a Microsoft Surface Pro 3 with an SMI REDn eye-tracker mounted at the base.

INTRODUCTION
For people with neurodegenerative diseases such as amyotrophic lateral sclerosis (ALS) or severe motor disabilities such as cerebral palsy or quadriplegia, interaction with computing devices may be limited to eye gaze, as their mobility restrictions can make a mouse, keyboard, or touch screen difficult or impossible to operate [21]. As a result, text entry for interpersonal communication [3] and other tasks is often accomplished through *gaze typing* (also referred to as *eye typing*). Gaze typing is achieved through the use of an eye tracking device and an on-screen keyboard. Although several novel approaches to gaze typing exist [22,24,25,31,35,41], the most common method is static dwell-based gaze typing, where the user is required to look at a key they would like to select, and then fixate on it for a certain, predetermined duration. The fixation duration needed to select the target is called the *dwell time*. Although the use of delimiters, such as blinking or other muscle movements, could enable more efficient interaction than dwell-based activation [1,45], many users with severe motor limitations (such as advanced ALS) do not have sufficient motor control for such interactions, so dwell-based gaze typing is the standard interaction for these populations [21].

The duration of dwell times can be user- or system-specific. In static dwell-based gaze typing systems, the dwell time is held constant for all keys on the keyboard. Longer dwell times help prevent accidental key activations, but force the user to take more time to input text. Shorter dwell times may afford faster text entry rates, but the risk of errors due to false activations is higher, as every key is more susceptible to the user's gaze, commonly referred to as the "Midas Touch problem" [11]. As a result, users and evaluators must negotiate the speed-accuracy tradeoff associated with selecting a dwell time. Due to the difficulty of negotiating this tradeoff, many gaze typing evaluations are conducted with medium to long dwell times (typically between 450 – 1000 ms [19]). Although gaze-based systems with constant, static dwell times are the most common, such dwell times remain the primary deterrent to faster text entry rates; regardless of how fast a user can gaze from one key to the next, he or she must remain fixated on the key until the dwell time is reached [12,28].

The text entry rates for dwell-based techniques are typically between 5 to 10 WPM [18], considerably slower than manual typing speeds of 40 to 60 WPM [4]. Improvements to gaze typing, even small ones, can be life-changing for people who rely on gaze typing for their everyday communication needs.

To alleviate the burden that static dwell times place on gaze-based text entry systems, and to improve text entry rates for users of these systems, we created a novel dwell-based gaze typing technique called *dynamic cascading dwell gaze typing*. Unlike systems with static dwell approaches, where all keyboard keys possess the same dwell time, our technique dynamically adjusts the dwell time of each key based on the likelihood that the key will be selected next, and based on the location of the of the key on the keyboard relative to other likely keys. Our approach makes less likely keys more difficult to select by increasing their required dwell time, reducing text entry errors caused by accidental activations. Conversely, our approach decreases the required dwell times of likely keys, allowing users to acquire their intended key faster. Our approach also takes into account keyboard layout, slowing clusters of likely keys to prevent false activations, and accelerating likely keys surrounded by unlikely keys.

To maintain a smooth and pleasant typing rhythm for the user, we *cascade* the dwell times of likely keys. As a user inputs text, the minimum allowable dwell time of likely keys is lowered slightly. By cascading the dwell time of likely keys, we make the selection of these keys easier while simultaneously preventing any jarring disruptions in the user's typing rhythm.

In a longitudinal study where 17 participants interacted with our system over 8 sessions, we found that with our dynamic cascading dwell approach, users' text entry speeds were significantly faster compared to the static approach, improving text entry speeds by 16.67% with no significant difference in errors remaining in the produced text. Our results also show that our cascading dwell technique significantly reduced the number of errors participants corrected while entering text, reducing the corrected error rate by 35.28%.

We also conducted a small evaluation with 5 participants with ALS. In our evaluation, participants achieved an average typing speed of 9.51 WPM with our cascading dwell technique. These results indicate that our dynamic cascading dwell approach is usable by people with ALS and can produce results similar to our non-disabled users.

The contributions of this work are twofold: (1) our novel dynamic cascading dwell gaze typing technique; and (2) empirical results from a study comparing our dynamic cascading dwell technique to a typical static dwell approach. We believe this work takes a significant step forward in improving dwell-based gaze typing systems.

RELATED WORK

This work was inspired by previous research to enhance dwell-based gaze typing, dwell-free gaze typing, novel gaze typing techniques, and studies on gaze typing behavior.

Improvements to Dwell-Based Gaze Typing

Researchers have investigated numerous ways to improve dwell-based gaze typing systems. Majaranta *et al*. [19,20] discovered that feedback that may work for longer dwell times may actually decreases typing speed when dwell times are short. We followed the feedback design guidelines provided by Majaranta *et al*. [20] in our dynamic cascading dwell and static dwell implementations by providing both auditory and visual feedback during key selection.

In their system, MacKenzie and Zhang [17] used a language model to predict which characters are likely to be selected next and a fixation algorithm that drifts a user's current fixation point away from unlikely keys and toward likely keys, reducing error rates. Our dynamic cascading dwell approach also uses next-letter prediction, but instead of drifting the gaze point, our technique increases the required dwell time of unlikely keys, making them more difficult to accidently activate, while decreasing the required dwell time of likely keys to speed entry.

Other notable improvements to dwell-based gaze typing include an on-line adjustment algorithm developed by Špakov and Miniotas [33], where the static dwell time is user-specific and dependent on how quickly a user can exit a key after it has been selected. Nantais *et al*. [23] proposed a prediction selection technique for dwell-based text entry where likely keys are selected once the user's gaze crosses the key boundary. A simulation of their technique's performance predicted that the technique could significantly reduce key dwell time, but no user evaluation of their technique was conducted. *EyeBoard* and *EyeBoard++* [24,30] are systems that rearrange the keys of an on-screen keyboard so that it is optimized for eye movement. The systems also dynamically change the dwell time of the entire keyboard based on user performance (although the dwell time might change, all of the keys share the same dwell time). In *Eye-K* by Sarcar *et al.* [31], a user enters text by gazing at the key they would like to select, looking outside the key, and back inside again.

Dwell-Free Gaze Typing

An alternative to dwell-based gaze typing is dwell-free typing. Instead of fixating on a key for a predetermined length of time, dwell-free systems allow users to gaze briefly at their intended key before moving to the next. The system is then responsible for disambiguating the user's input.

In a simulation of dwell-free gaze typing, Kristensson and Vertanen [12] found that dwell-free techniques have the potential to significantly improve text entry rates. However, since their results are based on simulations, it remains to be seen what performance gains might be had in practice. Furthermore, these simulations are based on eye gaze data collected in 10 minute intervals with long breaks, as fatigue caused performance to deteriorate over 10 minutes of gaze typing. Additionally, these simulations are based on error-free performance, further limiting the proposed theoretical limit of dwell-free eye typing.[1]

Filteryedping [25] is a gaze-based approach where the user quickly looks at the keys they would like to select on an on-screen keyboard. An algorithm disambiguates the collection of keys where the user's gaze was present and suggests likely words based on the user's gaze pattern. Non-disabled participants achieved average text entry speeds of 14.75 WPM with an average minimum string distance (MSD) error rate of 0.64% using Filteryedping. The average performance of participants with disabilities using Filteryedping varied greatly. Text entry rates ranged from 0.82 WPM to 15.54 WPM and MSD error rates ranged from 0.45% to 9.12%.

EyeSwipe [14] is a dwell-free system which aims to replicate the popular shape writing systems [13,42–44] found on touch-enabled mobile devices. Users select the first and last characters of the word using a reverse crossing technique, while middle characters are selected as the user gazes at them in order. Words are selected from a *n*-best list constructed from the user's gaze path. The authors reported that their non-disabled participants achieved average text entry rates of 11.7 wpm with an average MSD error rate of 1.31%.

Commercial eye-tracking company Tobii recently introduced a dwell-free gaze typing system which claims to get text entry rates up to 30 WPM[2], but this appears to represent peak performance by fully-abled, expert users without error correction. To our knowledge no scientific studies have been conducted with the Tobii dwell-free system. Thus, we do not know what the average text entry performance will be for users with and without disabilities.

Although dwell-free systems have the potential to improve gaze typing, they are not without their drawbacks. First, dwell-free systems have not yet surpassed the performance of dwell-based systems (no dwell-free technique has reported average typing speeds higher than the dwell-based 19.9 WPM reported by Majaranta *et al*. [18]). Second, dwell-free gaze typing may pose challenges for people with disabilities such as ALS, as fatigue may cause users to take a break between entering characters. Users with disabilities may also pause their typing to acknowledge or make eye contact with people in their environment. Dwell-based techniques support these intra-word typing behaviors while dwell-free approaches do not.

Novel Gaze-Based Text Entry Techniques

In addition to traditional dwell-based text entry systems, researchers have built and evaluated many novel gaze-based text entry techniques. Dasher by Ward and MacKay [35] has a zooming interface where users select characters by fixating on a particular dynamically-sized key until that key crosses a boundary point. Hansen *et al*. [6] created *StarGazer*, a 3D interface that uses continuous pan and zoom to select characters in 3D space. *pEYEwrite* by Huckauf and Urbina [9] is a hierarchical pie menu system that allows users to enter text by gazing at regions of the menu that contain their intended letter. *Context Switching* by Morimoto [22] is a two keyboard layout where users gaze between two keyboards to select keys.

The Minimal Device Independent Test Input Method (MDITIM) by Isokoski [10] is a gesture-based technique where users gaze at off-screen targets placed at the screen edges to form letters based on the MDITIM stroke alphabet. Wobbrock *et al*. [41] created *EyeWrite*, a stroke gesture-based system where users enter text by making letter-like gaze-strokes—based on the EdgeWrite alphabet [40]—among the four corners of an on-screen square. *Eye-S* by Porta and Turina [26] is another gesture-based system where the user fixates on a sequence of hotspots to enter a character.

These techniques and systems are useful contributions toward improving gaze-based text entry. However, many of these systems are difficult to implement in practice due to constraints such as interactive areas that require too much screen space [22], and steep learning curves [34,35] that may dissuade users from interacting with the system. As a result, dwell-based keyboard systems remain the most widely used gaze typing systems in practice. Our dynamic cascading dwell approach improves dwell-based gaze typing without changing the interface and without placing any additional learning burden on the user.

Studies of Gaze Typing Behavior

Researchers have conducted studies to better understand users' gaze typing behaviors. Majaranta *et al*. [18] conducted a longitudinal study to see how quickly users could type with a dwell-based keyboard if they were able to self-adjust the keyboard's dwell time. The authors discovered that the average self-selected dwell time of their participants decreased substantially from the first session to the tenth session, significantly improving participants' text entry performance. Räihä and Ovaska [28] performed an analysis

[1] See https://www.youtube.com/watch?v=Hvg45-IpBi0 for a research talk given by the first author of [12].

[2] http://www.tobiidynavox.com/communicator5/dwell-free/

of dwell-based gaze typing similar to the study conducted by Majaranta *et al.* [18]. The authors discovered that with training, participants were able to type effectively with short dwell times. In addition to dwell time, the authors found that *slack*, the time required to locate and fixate on a key, is fairly consistent across various dwell times and represents a small amount of key selection time. These results imply that dwell time is the biggest hindrance to faster text entry rates.

Results from these studies show that gaze typing with short dwell times can improve text entry rates, as the time required to select a key is greatly reduced. Our dynamic cascading dwell approach improves text entry rates by reducing the dwell time of likely keys, allowing users to select their intended keys quickly.

Majaranta *et al*. [20] discussed how feedback mechanisms should not disrupt the typing rhythm of the user. Dynamically changing key dwell times has the potential to interrupt users' typing rhythm, as the user may not be able to predict how long he or she must dwell on a key before it is selected. Our cascading dwell approach preserves users' typing rhythm by cascading the dwell time of likely keys, preventing jarring changes in key dwell times.

DYNAMIC CASCADING DWELL TIME DESIGN

Our dynamic cascading dwell time technique was designed to improve the performance of dwell-based gaze text entry by decreasing the dwell time of likely keys, making them easier to select. In addition, our technique increases the dwell time of unlikely keys, making them more difficult to select, reducing errors caused by false activations. In this section, we describe each step of our technique.

Step 1: Assigning the Baseline Dwell Time

Before the user begins to type a word, each key is assigned a *baseline dwell time*. The baseline dwell time represents the dwell time at which a user is comfortable entering text with the keyboard, and is analogous to the static dwell time applied uniformly to all keys in *status quo* dwell keyboards. The baseline dwell time can be user- or system-specific.

Step 2: Letter Prediction

Word prediction has been used in numerous gaze typing systems in an effort to improve performance [17,25]. Typically, systems that use word prediction suggest a small number of possible word outcomes based on the text entered by the user. The list of suggested words is displayed on the screen, typically above or to the side of the keyboard. The user then fixates on their desired word to select it.

Our technique utilizes word prediction to determine, for each key on the keyboard, what the likelihood is that the key will be selected next. When a character is input by the user, we generate an *n*-best list of possible word outcomes using a proprietary word prediction engine. (Any prediction engine can work with our technique, as long as individual likelihoods can be assigned to each key.) For every predicted word w, we determine which letter is located at position w_{i+1}, where i is the index of the last letter in the word currently being entered by the user. Next, we sum the occurrences of each letter present at w_{i+1}, and, for each letter, divide the sum by the total number of predicted words, resulting in every letter receiving a likelihood score between 0.00 and 1.00. The sum of all key likelihoods is equal to 1.00.

Step 3: Determine the Minimum Cascading Dwell Time

Once a character has been entered, we determine the *minimum cascading dwell time*. The minimum cascading dwell time is the *minimum* dwell time that can be assigned to a key. (Before the first character of a word is entered, the minimum cascading dwell time is equal to the baseline dwell time.) The minimum cascading dwell time is reduced by 10% every time a character is entered. Thus, after the first character in a word is entered, the minimum cascading dwell time is 10% shorter than the baseline dwell time.

The value of the cascading minimum dwell time will continue to drop as the user enters text until it reaches the *minimum allowable dwell time*, which is the shortest dwell time allowed by the system.

The minimum cascading dwell time was inspired by the psychological principle of the *just-noticeable difference* [2,36], which describes how much a stimulus must be changed before a difference can be noticed. The purpose of the minimum cascading dwell time is to allow for faster dwell times while providing little to no disruptions to the user's typing cadence. Large differences in dwell time from one key to the next can disrupt the user's typing rhythm, as the user would not be able to predict how long they must dwell on the key before it is selected. By *cascading* the dwell time—slightly lowering the dwell time after each character entry—our technique provides a series of successive small dwell time decreases that, over time, make the selection of keys faster without jarring the user.

Step 4: Adjusting Character Key Dwell Times

At this point, each letter has received a score between 0.00 and 1.00 representing how likely the key is to be selected next. (Letters with higher scores are more likely.) Now, we adjust the dwell time of each letter key based on this likelihood.

We iterate through every character key. For characters with likelihoods less than 0.01, or 1%, we increase their key dwell times to be equal to the *maximum allowable dwell time*. The maximum allowable dwell time serves two purposes. First, and most importantly, it increases the dwell time of unlikely keys, making their selection more difficult, which reduces the chance that an unintended character will be entered because of a false activation. Second, it allows users to enter text for words that are outside the vernacular, such as names or email addresses. Therefore, the maximum allowable dwell time *should not* be a value that makes entering out of vocabulary text impossible.

For characters with likelihoods greater than 0.01, we assign two dwell time values. The first is the *density dwell time*. We calculate the density dwell time using Equation 1:

$$DT_d = \left(\left({}^{a_l}/_{a_t}\right) * (dt_{bl} - dt_{min})\right) + dt_{min} \qquad (1)$$

Here, DT_d is the density dwell time, a_l is the number of likely adjacent keys, a_t is the total number of adjacent keys, dt_{bl} is the baseline dwell time, and dt_{min} is the minimum cascading dwell time.

The density dwell time is the dwell time of a likely key based on the number of other likely keys adjacent to it. In cases where all keys adjacent to a likely key are also likely (*i.e.*, a_l and a_t are the same), the density dwell time for that key will be equal to the baseline dwell time. Conversely, if a likely key is surrounded by no other likely keys, that key's density dwell time will be equal to the minimum cascading dwell time. The purpose of the density dwell time is to *slow* clusters of likely targets. If a user attempts to acquire a key surrounded by other likely keys, we do not want to shorten the dwell times of keys in that cluster too much, as the user may accidently activate an adjacent key. (For example, if 'I' and 'O' are both likely keys, we do not want the user to accidently select 'I' when 'O' was the intended key.) When a likely key is not surrounded by other likely keys, however, the dwell time of that key can be set to the minimum cascading dwell time. In this case, we are not concerned that the user will accidently activate the key while attempting to select another key, since all of the adjacent keys will have relatively long dwell times, and are therefore unlikely to be accidentally selected.

If a likely key has no adjacent keys that are also likely, the dwell time of the key is set to the value of DT_d, which is the minimum cascading dwell time. If a likely key is surrounded by other likely keys (*i.e.*, a_l does not equal 0), we compute a second dwell time value. The second dwell time is the likelihood dwell time. The likelihood dwell time is a key dwell time value where the dwell time is weighted by the key's likelihood. The more likely a key is to be selected, the lower the key's likelihood dwell time value. We calculate the likelihood dwell time using Equation 2:

$$DT_l = \left((1 - k_l) * (dt_{bl} - dt_{min})\right) + dt_{min} \qquad (2)$$

Here, DT_l is the likelihood dwell time, k_l is the likelihood of a given key (a value between 0.01 and 1.00), dt_{bl} is the baseline dwell time, and dt_{min} is minimum cascading dwell time.

After both dwell time scores have been computed, the maximum dwell time value between DT_d and DT_l is chosen and used for the key's dwell time. It is possible that the word prediction engine returns zero predicted words, indicating that the word currently being entered is outside the vocabulary. If no predicted words are available, we cannot compute the key likelihoods. In this case, the dwell time of all character keys is set equal to 160% of the baseline dwell time, and the dwell time of the BACKSPACE key is set equal to the minimum allowable dwell time. This penalty to the character key is to allow the user to enter out of vocabulary words at a reasonable pace, and also to signal to the user that they may have committed an error while typing their current word. Reducing the dwell time of the BACKSPACE key allows users to quickly correct errors when they occur.

Step 5: Adjusting the Dwell Time of the SPACE Key

In addition to updating the dwell time of letter keys, we also adjust the dwell time of the SPACE key. When a character is entered, we perform a dictionary lookup of the word currently being typed by the user. If the current word matches a complete word in our dictionary, the dwell time of the SPACE key is set equal to two-thirds of the baseline dwell time. If the current word is not found in the dictionary, the SPACE key is set to the baseline dwell time. Reducing the dwell time of the SPACE key allows users to enter text more efficiently by quickly enter a space after typing.

We only want to cascade key dwell times while users are entering a word. Therefore, after a word has been entered, signaled by the presence of a space, the aforementioned process is reset back to Step 1.

LONGITUDINAL STUDY

To discover whether our cascading dwell technique could improve user performance compared to the traditional static dwell approach, we conducted an experiment spanning eight sessions. Conducting the experiment over eight sessions allowed us to see how user performance with the two techniques changed over time, as we expected the learning curves of the two techniques to be different.

Participants

We recruited 20 people without physical disabilities to participate in our study. Participants were recruited from the employee base of a large software company in North America, and included people with diverse job roles (*e.g.*, administrative assistant, IT support specialist, software developer, etc.). Three female participants did not complete all experiment sessions, resulting in a total of 17 participants (2 female, 15 male, average age of 36.8, SD=10.0). One participant wore glasses and one participant wore contact lenses. None of the participants had previous experience using gaze-based interfaces or gaze-typing systems.

Participants were paid a total of $200 for participating in the study according to the following schedule: $25 for session 1, $10 each for sessions 2 through 7, and $115 for session 8.

Apparatus

We conducted our study on a Microsoft Surface Pro 3, with a 12-inch screen and a 2160×1440 pixel resolution. An SMI REDn eye tracker with a sampling rate of approximately 60 Hz was used to collect eye-tracking data.

We collected text entry data using a custom-built testbed written in C# .NET 4.5. The testbed housed an on-screen keyboard that resembles a QWERTY keyboard with all 26 English letter keys, SHIFT, SPACE, BACKSPACE and ENTER (see Figure 2). The testbed randomly presented a target phrase above the keyboard from the MacKenzie and Soukoreff phrase corpus [16]. The transcribed text entered by participants was shown directly below the target phrase.

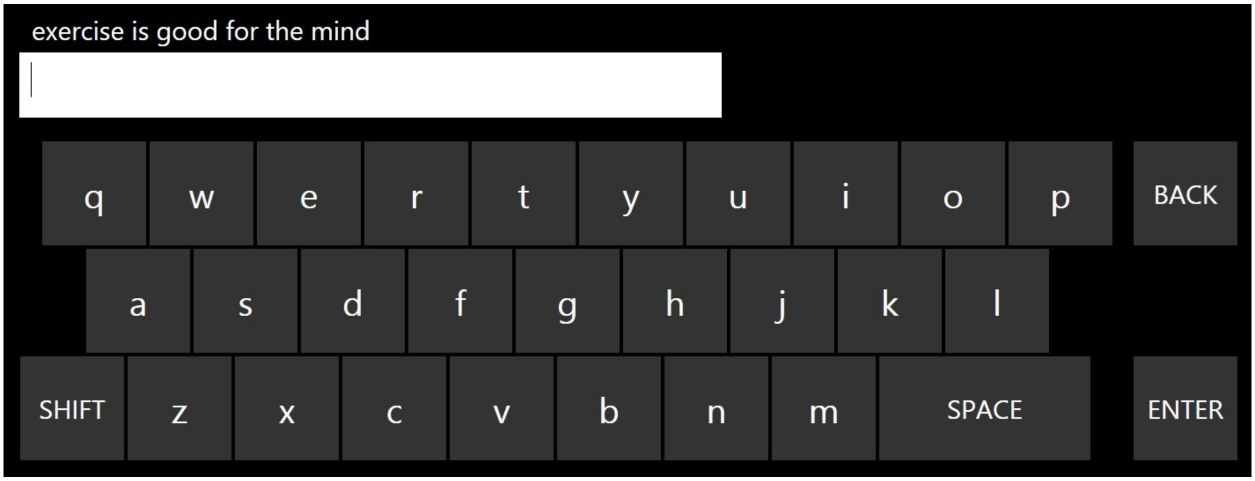


Figure 2. The keyboard used in our longitudinal study, including a target phrase and output area.

We provided a user control to allow participants to adjust their dwell time before they began a trial (Figure 3). Allowing participants to self-adjust their dwell time was first implemented and studied by Majaranta *et al.* [18]. Participants could gaze at the "-50" or "+50" buttons to decrease or increase the current dwell time by 50 ms. The participants' current dwell time was displayed between the time adjustment buttons. For the static approach, the dwell time of all the keys was set to the participants' selected dwell time. For our dynamic cascading technique, the selected dwell time served as the baseline dwell time.

Figure 3. Participant were given an opportunity to memorize the target phrase and to adjust their dwell time using the time adjustment controls located at the right of the UI.

The keyboard had a two-step key selection feedback mechanism. First, when a user fixated on a target, the background border color of the key changed color (Figure 4A). For the key border to change color, the user had to fixate on the key for at least 30% of the key's dwell time. Second, after the key's dwell time was reached, the background of the key changed color, and an audible "click" sounds was played, indicating to the user that a key selection had been made (Figure 4B).

To distinguish between the cascading and static approaches, the two approaches were given separate feedback colors. Feedback for the cascading approach was light blue, and feedback for the static condition was purple. Participants were blind to the differences between the keyboards (*i.e.*, use of static *vs.* cascading dwell times), and simply knew that there were two keyboards, distinguished visually only by the color of the selection feedback.

For our study implementation of the dynamic cascading dwell approach, the minimum allowable dwell time was set to 100 ms. The maximum allowable dwell time was set to 1000 ms. Prior research has shown that a 1000 ms dwell time

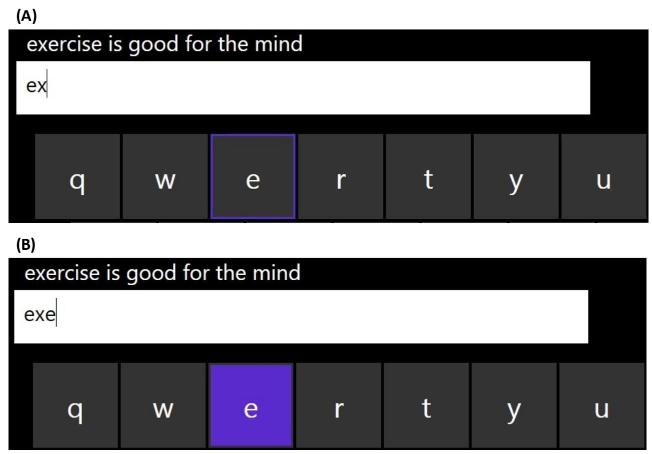


Figure 4. (A) When a user fixates on a key for at least 30% of its dwell time the border of the key changes color. (B) Once a key's dwell timer is reached its background color is changed and the user hears an audible click.

is long enough to prevent accidental activations [20]. The starting dwell time for all participants was set to 600 ms.

Procedure

Each participant completed eight experiment sessions. Sessions 1 and 8 lasted one hour and sessions 2 through 7 lasted 30 minutes. In the first session the experimenter explained that the purpose of the study was to measure the performance of two gaze keyboards, but the specifics of the purpose (what the keyboards did, which one was our innovation, etc.) were hidden from the participant to avoid bias. In session 1 the experimenter familiarized participants with the testbed and explained how to adjust the dwell time using the on-screen controls and how to select a key.

In each session, the user was seated comfortably in front of the tablet. Participants were instructed to move as little as possible. Before the start of the text entry task, the eye-tracker was calibrated. Recalibration was not allowed once the text entry task began. At the beginning of a trial, participants were presented with a target phrase from the

MacKenzie and Soukoreff phrase corpus [16] and were asked to memorize it. To enter the phrase, participants had to select the "Start" key using gaze, which would bring up the on-screen keyboard. Participants would then transcribe the target phrase and select the "Enter" key to end the trial. After selecting the "Enter" key, the keyboard would disappear, a new phrase would be presented, and the dwell time adjustment controls would reappear. (Although the Start and Enter keys were used as described, the timing information used to calculate speeds was from the entry of the first character to the entry of the last.) Participants were instructed to enter text as quickly and as accurately as possible. Participants were also instructed to correct errors that may occur, but only if the errors had occurred in the current word.

Participants were instructed to adjust their dwell time whenever they wanted. The dwell time could only be adjusted at the beginning of a trial (before the keyboard was shown), using the on-screen controls shown in Figure 2. Participants were able to set two separate dwell times, one for each technique. Participants knew which technique they were using based on the color of the key selection feedback (purple or blue). The last selected dwell time for each approach was saved at the end of the session, and was set as the default dwell time at the beginning of the next session.

When switching from one approach to the next, and before the end of the session, participants were presented with a set of subjective questions (primarily derived from NASA TLX [7]) asking them to rate their most recently used approach on several dimensions. Participants made their subjective question selections using gaze by selecting responses the same way they selected keys during the text entry trials. In the 8th session, participants also answered a series of comparative questions about the two approaches.

Design and Analysis

Our study was a 2 × 8 within-subjects design with the following factors and levels:

- *Technique*: Cascading and Static
- *Session*: 1, 2, 3, 4, 5, 6, 7, 8

The presentation order of *Technique* was counterbalanced across participants and sessions to account for order effects. Participants completed 24 text entry trials with each technique in sessions 1 and 8, with the first four trials in each technique serving as practice. Participants completed 12 text entry trials with each technique in sessions 2 through 7, with the first two trials in each technique serving as practice. Due to time constraints 73 trials were not completed by our participants, resulting in a total of 3327 collected trials out of a possible 3400.

The dependent measures for our study were words per minute (WPM), uncorrected error rate, and corrected error rate [32]. For a given participant, we averaged the trial data for our dependent measures in each *Session* for each *Technique*. Words per minute were analyzed with a mixed-effects model analysis of variance [15]. Our model used fixed effects for *Technique* and *Session*; *Subject* was modeled as a random effect to accommodate for repeated measures. Due to wider confidence intervals, mixed-effects models do not make detection of significance any easier compared to traditional fixed effects models.

Uncorrected and corrected error rates, as error rates often go, were not amenable to parametric models like analysis of variance due to the violation of model assumptions. We therefore used the nonparametric Aligned Rank Transform (ART) procedure [8,29,38], which aligns and ranks the data prior to analysis. The ART procedure preserves the integrity of both main and interaction effects. Subjective questions with ordinal Likert scale responses were also analyzed using the ART procedure.

RESULTS

This section presents the results from our longitudinal study to determine the effectiveness of our cascading dwell typing technique.

Speed

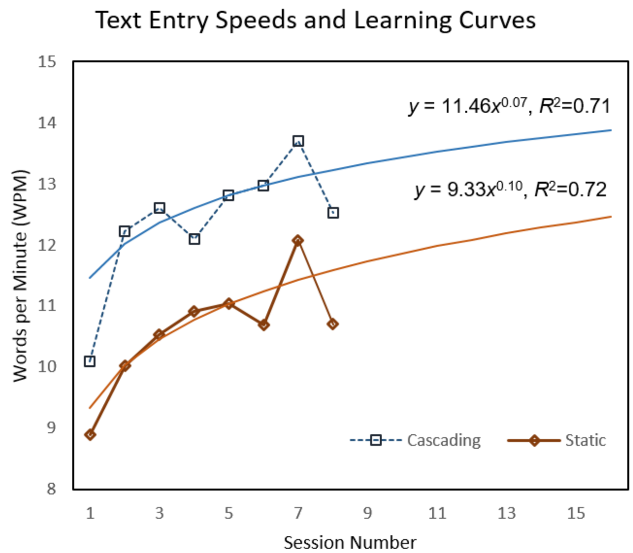
Speed was measured as words per minute (WPM). Over all sessions (1-8), participants' average typing speed was 12.39 WPM (σ=3.23) with the dynamic cascading dwell keyboard and 10.62 WPM (σ=2.89) with the static dwell keyboard (Table 1). There was a significant effect of *Technique* on speed ($F_{1,235}$=57.12, p<.0001). Unsurprisingly, *Session* also exhibited a significant effect on typing speed ($F_{7,235}$=9.10, p<.0001), as participants got faster over sessions. Mean typing speeds for the two techniques over the eight sessions are shown in Figure 5. There was not a significant *Session* × *Technique* interaction ($F_{7,235}$=0.26, *n.s.*), however, indicating that additional training did not provide an advantage for one technique over the other. Cascading dwell remained the faster technique across all eight sessions. The maximum average session speeds were 13.70 WPM for dynamic cascading dwell (session 7) and 12.08 WPM for static dwell (also session 7).

Learning Curves

We fit our speed data to learning curves of the form $y=ax^b$, where a and b are empirically determined constants, y is speed (in WPM), and x is session. The learning curves for both techniques are shown in Figure 5. The R^2 values were 0.71 for cascading dwell and 0.72 for static dwell. The learning curves show that our cascading technique started with higher typing rates and remained higher over sessions.

Uncorrected Error Rate

Uncorrected errors are errors that remain in the final text [32]. Speed and uncorrected errors are at odds, as a user can enter text faster by leaving errors. Conversely, a user's typing speed will be reduced by correcting errors. Over all eight sessions, the average uncorrected error rate for cascading dwell was 1.45% (σ=2.50), and for static dwell it was 1.95% (σ=3.60). Although cascading dwell was more accurate on average, this difference was not statistically significant ($F_{1,235.1}$=0.95, *n.s.*). There was a significant effect of *Session* on uncorrected error rate ($F_{7,235.1}$=2.21, p<.05), as



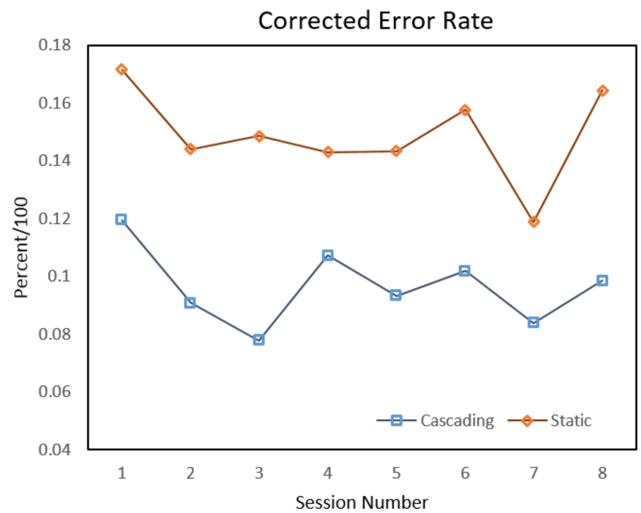


Figure 5. (Left) Text entry speeds (in WPM) for each technique over all eight sessions and learning curves fit to the speed data modeled by the function $y=ax^b$. (Right) Corrected error rates for each technique over all sessions.

participants made fewer uncorrected errors in later sessions. There was no significant *Session* × *Technique* interaction ($F_{7,235.2}$=0.81, *n.s.*).

Corrected Error Rate

Corrected errors are errors that are committed but then corrected during entry [32]. Corrected errors are a measure of how error-prone a text entry method is, regardless of the correctness of the text produced at the end [37]. Over all eight sessions, the average corrected error rate for cascading dwell was 9.63% (σ=0.50), and for static dwell it was 14.88% (σ=0.77) (see Table 1). The mean corrected error rates for the two techniques over the eight sessions are shown in Figure 5. There was a statistically significant effect of *Technique* on corrected error rate ($F_{1,235}$=73.34, p<.0001). There was no statistically significant effect of *Session* on corrected error rate ($F_{7,235}$=2.02, p=.0532), though the trend indicates that participants were somewhat more accurate in later sessions. There was also no significant *Session* × *Technique* interaction ($F_{7,235}$=0.80, *n.s.*).

Error-Free Performance

To discover how cascading the dwell time of likely keys impacted performance, we looked at text entry speed for trials where no corrected or uncorrected errors were present. By excluding trials with error, we can see how error-free performance compares for the two techniques. Thus, the difference in speed for the two techniques can be attributed to the dynamic cascading dwell time of likely keys.

There were a total of 368 error-free trials (228 for cascading and 140 for static). The average typing speed for error-free trials was 16.16 WPM (σ=3.54) with cascading and 15.11 WPM (σ=3.24) for static. The average baseline dwell time of the error-free trials for cascading was 373.03 ms (σ=108.69) and 388.21 ms (σ=101.54) for static. The difference between two keyboards represents a 7% increase in typing speed for cascading dwell over static dwell.

Technique	WPM*	Uncorrected Error Rate*	Corrected Error Rate
Cascading	12.39 (3.23)	1.45% (2.50)	9.63% (0.50)
Static	10.62 (2.89)	1.95% (3.60)	14.88% (0.77)

Table 1. Overall means for all eight sessions for typing speed (in WPM), corrected error rate, and uncorrected error rate. Standard deviations are shown in parenthesis. Items with an asterisk (*) showed statistically significant differences (p<.05).

Dwell Time

The mean baseline dwell time at the end of session 1 was 395 ms (σ=107.07) for cascading and 408 ms (σ=97.12) for static. At the end of session 8, the mean baseline dwell time for cascading was 334 ms (σ=92.30) and 327 ms (σ=60.87) for static. Only two participants did not change their dwell time in the first session, and no participants increased their dwell time over the initial value of 600 ms. Although participants selected slightly higher dwell times for our cascading technique, it was still able to outperform the static approach in terms of text entry speed and accuracy.

Physical Keyboard Typing Performance

At the beginning of session 1 all participants completed a short typing task where they transcribed five phrases using a physical keyboard. We used the *StreamAnalyzer* and *TextTest* software by Wobbrock and Myers [39] to conduct the assessments and analyze the data. We wanted to know if familiarity with a QWERTY keyboard layout would benefit participants in their gaze typing performance. There was no significant correlation between participants' physical keyboard and gaze typing speeds (using the static approach) in the first session (r=-0.25, n=17, *n.s.*).

Subjective Preferences and Workload Ratings

After completing a set of text entry trials with each technique in each session, participants were asked to answer a set of subjective questions regarding their most recently used technique. Workload was assessed using the NASA TLX questionnaire [7], which rates workload on the following dimensions: mental demand; physical demand; temporal demand; performance; effort; and frustration. Instead of the 21-point scales used in the traditional TLX questionnaire, we used a 7-point scale that eliminated the three within-point gradations present in the original TLX. We also asked participants to rate the techniques using a 5-point Likert scale on additional qualities which were: perceived speed, perceived accuracy, perceived adaptation to typing ability, pleasantness of typing rhythm, and ease of error correction.

Participants exhibited significant preferences for our cascading technique over the static approach. Participants felt that our cascading technique allowed them to enter text faster ($F_{1,235}$=27.41, p<.0001) and more accurately ($F_{1,235}$=30.49, p<.0001). Participants also felt that our cascading approach adapted to their typing ability ($F_{1,235}$=18.98, p<.0001), had a more pleasant typing rhythm ($F_{1,235}$=36.54, p<.0001), and made it easier to correct errors ($F_{1,235}$=17.10, p<.0001) compared to the static technique. Mean subjective preference scores are shown in Table 2.

Technique	Q	AC	AD	R	C
Cascading	3.62 (1.00)	3.41 (1.05)	2.98 (0.93)	3.65 (0.73)	3.26 (1.13)
Static	3.07 (1.10)	2.84 (1.06)	2.59 (0.88)	3.23 (0.86)	2.87 (1.16)

Table 2. Mean subjective preference scores (1-5) across all sessions for perceived quickness (Q), accuracy (AC), adaptation to typing ability (AD), pleasantness of typing rhythm (R), and ease of correction (C). Higher is better for all measures. Standard deviations are shown in parenthesis.

Mean response scores for the NASA TLX are shown in Table 3. Results for the effect of *Technique* on the NASA TLX workload dimensions show that our cascading approach required significantly less mental demand ($F_{1,235}$=12.39, p<.001), required significantly less physical demand ($F_{1,235}$=7.28, p<.01), provided better perceived performance ($F_{1,235}$=26.75, p<.0001), required significantly less effort ($F_{1,235}$=19.35, p<.0001), and was significantly less frustrating to use ($F_{1,235}$=15.48, p<.0001). There was no significant difference between the techniques for temporal demand ($F_{1,235}$=0.02, *n.s.*). With the exception of temporal demand, our cascading technique received better workload ratings compared to the static dwell approach.

In the final session, participants were asked to directly compare which technique they felt was more accurate, which was faster, which had the more comfortable typing rhythm, and which technique they would prefer to use on a regular basis. (Recall that participants were kept from knowing the underlying differences between techniques; the techniques were distinguished merely as the keyboard with blue- or purple-colored feedback.) One-sample Pearson Chi-square tests of proportions show that participants felt our cascading technique was more accurate (14 of 17, $\chi^2_{(1,N=17)}$ = 7.12, p<.01), faster (13 of 17, $\chi^2_{(1,N=17)}$ = 4.76, p<.05), provided a more comfortable typing rhythm (15 of 17, $\chi^2_{(1,N=17)}$ = 9.94, p<.01), and would be the technique they would prefer to use on a regular basis (15 of 17, $\chi^2_{(1,N=17)}$ = 9.94, p<.01).

Technique	MD	PD	TD	P	E	F
Cascading	3.14 (1.54)	2.80 (1.43)	2.67 (1.15)	2.84 (1.57)	3.05 (1.54)	2.41 (1.43)
Static	3.53 (1.53)	3.13 (1.51)	2.66 (1.14)	3.54 (1.63)	3.62 (1.64)	3.05 (1.71)

Table 3. Mean NASA TLX scores (1-7) across all sessions for mental demand (MD), physical demand (PD), temporal demand (TD), performance (P), effort (E), and frustration (F). Lower is better for all measures. Standard deviations are shown in parenthesis.

Evaluation with People with ALS

To gain a better understanding of how well people with disabilities might use our cascading dwell approach, we conducted a small evaluation with five people with ALS (1 female, average age of 50.4, SD=6.95). Each participant completed a total of 10 phrases from the same phrase set as our non-disabled participants. Participants completed 10 phrases (rather than a longer, longitudinal study) and only in one condition (using the dynamic cascading dwell keyboard). We chose to use all 10 of these trials with our new keyboard so as to have more data to get a more reliable estimate of ALS patients' performance with and acceptance of this technique. Our goal was to verify that people with disabilities could successfully use the system and could achieve performance in keeping with our able-bodied study participants.

Six trials were discarded due to participants becoming distracted during the trial or prematurely progressing to the next trial, resulting in 44 total trials. Each participant used our cascading dwell technique with a baseline dwell time of 400 ms. The average typing speed was 9.51 WPM (σ=2.70) with average uncorrected error and corrected error rates of 1.64% (σ=0.03) and 8.80% (σ=0.07), respectively. This speed and these error rates are in keeping with those observed in our larger study of able-bodied participants.

DISCUSSION

We wanted to discover if our dynamic cascading dwell technique could improve text entry performance over the traditional static dwell approach. Our results show that our cascading dwell technique was significantly faster than static dwell. Our results also show that participants made significantly fewer errors while entering text with cascading dwell compared to static dwell. In addition, final text transcriptions were comparably accurate with both techniques, avoiding a speed-accuracy tradeoff so common in human performance studies [5]. Our results indicate that our design decisions to decrease the dwell time of likely keys and increase the dwell time of unlikely keys were effective at improving text entry rates. By increasing the dwell time of unlikely keys, our technique significantly reduced the amount of corrected errors by making it more difficult to accidently select an incorrect key. Fewer errors made during entry allowed participants to enter text more quickly, since less time was spent correcting errors. Furthermore, by examining trials with no errors, we see that our dynamic cascading technique was 7% faster than the static approach. This result shows that decreasing the dwell time of likely keys does provide an improvement in text entry speed.

The corrected error rates for both cascading and static dwell are higher than error rates reported in previous studies [18,28]. We believe the difference in error rates can be attributed to the screen size used in our study. Previous studies conducted experiments on 15- to 17-inch monitors. As a result, the keys on their on-screen keyboards were much larger, making them easier to acquire using eye gaze. Acquiring small on-screen keys is more error-prone due to poor eye tracker calibrations [27]. If our study were conducted on a larger screen, we believe the error rates for the two techniques would be lowered substantially. We felt it was important, however, to conduct our study on a device that resembles those used by people who rely on gaze typing for their everyday communication needs; tablet computers are increasingly used for AAC because they are relatively low-cost, are lightweight enough to be easily mounted on a wheelchair or carried about, and offer a reasonable tradeoff between being large enough to enable interaction but small enough so as not to completely occlude a user's field of view when mounted on an arm in front of their wheelchair.

It is possible that the decrease in gaze typing performance observed in session 8 is due to fatigue caused by the increase in gaze typing trials in that session. (Recall that participants completed a total of 48 trials each in sessions 1 and 8 and 24 total trials each in sessions 2 through 7.) Responses to the NASA TLX questionnaire support this claim, as the responses for physical demand (an approximation of fatigue) were slightly higher for sessions 1 and 8 (a combined average score of 3.37) compared to sessions 2 through 7 (a combined average score of 2.84).

The average baseline dwell times for both techniques were quite similar throughout the study, with the static technique having a slightly lower average baseline dwell time in session 8 compared to the cascading technique (327 ms compared to 334 ms). Even with a slightly shorter average dwell time, our dynamic cascading technique was able to outperform the static approach. The short average baseline dwell times for both techniques is similar to the results found in previous studies where users were allowed to self-adjust their dwell times [18,28].

Subjective results show that participants felt that our dynamic cascading technique imposed significantly less workload. Subjective results also show that our cascading technique had a more pleasant typing rhythm compared to the static approach. This result is encouraging, as it demonstrates that cascading the dwell time of likely keys did not disrupt the participants' typing rhythm. Instead, participants actually *preferred* the cascading dwell typing rhythm compared to the static approach.

The results from our short typing assessment with a physical keyboard demonstrated that touch typing proficiency on a QWERTY keyboard does not guarantee initial success with gaze typing using an on-screen keyboard with a QWERTY layout. This result suggests that non-QWERTY layouts designed to support eye gaze interaction styles may offer further benefits, perhaps with cascading dwell. This serves as a potential direction for future work.

Limitations

A limitation of this work is that our longitudinal study was not conducted with users with disabilities. Our participants' performance with the two techniques may not match the longitudinal performance of users with motor disabilities. We expect that the relative differences between the two keyboards (*i.e.*, better performance with cascading dwell versus static dwell) would generalize to participants with disabilities, though the specifics of WPM and error rates would likely vary depending on the nature of a user's disability (*e.g.*, susceptibility to fatigue, degree of control of eye muscles, etc.). Our initial results with people with ALS give us confidence in making this generalization; they were able to successfully use our technique with no training and achieve good typing speeds; although their typing speed was slightly lower than that of our longitudinal study participants, this is to be expected both due to the fatigue and motor control issues accompanying ALS, and since the participants with ALS did not have the benefit of practice over many trials that the users in our longitudinal studies had. Evaluating our technique in controlled, longitudinal sessions with users with motor impairments is an important area for future work.

CONCLUSION

In this paper, we have presented a new technique for gaze typing called cascading dwell gaze typing. Our dynamic cascading dwell technique improves dwell-based gaze typing by dynamically reducing the dwell time of likely keys, making them easier to select, while preserving a pleasant typing rhythm. Conversely, our technique increases the dwell time of unlikely keys, making them more difficult to accidently activate. In a controlled user study, we found that our cascading dwell technique significantly improved text entry speed, and significantly reduced the number of errors committed while entering text compared to a static dwell approach. This work takes a significant step toward improving gaze base text entry, a life-changing form of interaction for many people with motor disabilities.

ACKNOWLEDGEMENTS

We thank all of our participants for their time and feedback. We would also like to thank the members of Microsoft Research's Enable team for their assistance with this work, and the members of the PALS program for the inspiration they provided. We would especially like to thank Ann Paradiso, Mira Shah, Harish Kulkarni, Pete Ansell, Jon Campbell, Jay Beavers, and Rico Malvar for their contributions.

REFERENCES

1. Behrooz Ashtiani and I. Scott MacKenzie. 2010. BlinkWrite2: An improved text entry method using eye blinks. *Proceedings of the ACM Symposium on Eye-Tracking Research & Applications* (ETRA ’10), 339–345. https://doi.org/10.1145/1743666.1743742
2. Gosta Ekman. 1959. Weber’s law and related functions. *The Journal of Psychology: Interdisciplinary and Applied* 47: 343–351. https://doi.org/10.1080/00223980.1959.9916336
3. Melanie Fried-Oken, Lynn Fox, Marie T. Rau, Jill Tullman, Glory Baker, Mary Hindal, Nancy Wile, and Jau-Shin Lou. 2006. Purposes of AAC device use for persons with ALS as reported by caregivers. *Augmentative and Alternative Communication* 22, 3: 209–221. https://doi.org/10.1080/07434610600650276
4. Jonathan T. Grudin. 1983. Error patterns in novice and skilled transcription typing. *Cognitive Aspects of Skilled Typewriting*, William E. Cooper (ed.). Springer New York, 121–143. http://link.springer.com/chapter/10.1007/978-1-4612-5470-6_6
5. P. A. Hancock and K. M. Newell. 1985. The movement speed-accuracy relationship in space-time. In *Motor Behavior: Programming, Control, and Acquisition.*, Professor Dr Herbert Heuer, Professor Dr Uwe Kleinbeck and Dipl-Psych Klaus-Helmut Schmidt (eds.). Springer Berlin Heidelberg, 153–188. http://link.springer.com/chapter/10.1007/978-3-642-69749-4_5
6. Dan Witzner Hansen, Henrik H. T. Skovsgaard, John Paulin Hansen, and Emilie Møllenbach. 2008. Noise tolerant selection by gaze-controlled pan and zoom in 3D. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’08), 205–212. https://doi.org/10.1145/1344471.1344521
7. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in Psychology*, Peter A. Hancock and Najmedin Meshkati (ed.). North-Holland, 139–183. http://www.sciencedirect.com/science/article/pii/S0166411508623869
8. James J. Higgins and S Tashtoush. 1994. An aligned rank transform test for interaction. *Nonlinear World* 1, 2: 201–211.
9. Anke Huckauf and Mario H. Urbina. 2008. Gazing with pEYEs: Towards a universal input for various applications. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’08), 51–54. https://doi.org/10.1145/1344471.1344483
10. Poika Isokoski. 2000. Text input methods for eye trackers using off-screen targets. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’00), 15–21. https://doi.org/10.1145/355017.355020
11. Robert J. K. Jacob. 1991. The use of eye movements in human-computer interaction techniques: What you look at is what you get. *ACM Transactions on Information Systems* 9, 2: 152–169. https://doi.org/10.1145/123078.128728
12. Per Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. *Proceedings of the Symposium on Eye Tracking Research and Applications* (ETRA ’12), 241–244. https://doi.org/10.1145/2168556.2168605
13. Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: A large vocabulary shorthand writing system for pen-based computers. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST ’04), 43–52. https://doi.org/10.1145/1029632.1029640
14. Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI ’16), 1952–1956. https://doi.org/10.1145/2858036.2858335
15. R. C. Littell, P. R. Henry, and C. B. Ammerman. 1998. Statistical analysis of repeated measures data using SAS procedures. *Journal of Animal Science* 76, 4: 1216–1231.
16. I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (CHI EA ’03), 754–755. https://doi.org/10.1145/765891.765971
17. I. Scott MacKenzie and Xuang Zhang. 2008. Eye typing using word and letter prediction and a fixation algorithm. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’08), 55–58. https://doi.org/10.1145/1344471.1344484
18. Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI ’09), 357–360. https://doi.org/10.1145/1518701.1518758
19. Päivi Majaranta, Anne Aula, and Kari-Jouko Räihä. 2004. Effects of feedback on eye typing with a short dwell time. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’04), 139–146. https://doi.org/10.1145/968363.968390
20. Päivi Majaranta, I. Scott MacKenzie, Anne Aula, and Kari-Jouko Räihä. 2006. Effects of feedback and dwell time on eye typing speed and accuracy. *Universal Access in the Information Society* 5, 2: 199–208. https://doi.org/10.1007/s10209-006-0034-z
21. Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty years of eye typing: Systems and design issues. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA ’02), 15–22. https://doi.org/10.1145/507072.507076

22. Carlos H. Morimoto and Arnon Amir. 2010. Context switching for fast key selection in text entry applications. *Proceedings of the ACM Symposium on Eye-Tracking Research & Applications* (ETRA '10), 271–274. https://doi.org/10.1145/1743666.1743730
23. T. Nantais, F. Shein, and J. Treviranus. 1994. A predictive selection technique for single-digit typing with a visual keyboard. *IEEE Transactions on Rehabilitation Engineering* 2, 3: 130–136. https://doi.org/10.1109/86.331562
24. P. Panwar, S. Sarcar, and D. Samanta. 2012. EyeBoard: A fast and accurate eye gaze-based text entry system. *International Conference on Intelligent Human Computer Interaction (IHCI)*, 1–8. https://doi.org/10.1109/IHCI.2012.6481793
25. 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 Transactions on Accessible Computing* 6, 1: 3:1–3:37. https://doi.org/10.1145/2724728
26. Marco Porta and Matteo Turina. 2008. Eye-S: A full-screen input modality for pure eye-based communication. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA '08), 27–34. https://doi.org/10.1145/1344471.1344477
27. Kari-Jouko Räihä. 2015. Life in the fast lane: Effect of language and calibration accuracy on the speed of text entry by gaze. *Human-Computer Interaction – INTERACT 2015*, 402–417. http://link.springer.com/chapter/10.1007/978-3-319-22701-6_30
28. Kari-Jouko Räihä and Saila Ovaska. 2012. An exploratory study of eye typing fundamentals: Dwell time, text entry rate, errors, and workload. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '12), 3001–3010. https://doi.org/10.1145/2207676.2208711
29. K. C. Salter and R. F. Fawcett. 1993. The art test of interaction: a robust and powerful rank test of interaction in factorial models. *Communications in Statistics - Simulation and Computation* 22, 1: 137–153. https://doi.org/10.1080/03610919308813085
30. Sayan Sarcar and Prateek Panwar. 2013. Eyeboard++: An enhanced eye gaze-based text entry system in hindi. *Proceedings of the Asia Pacific Conference on Computer Human Interaction* (APCHI '13), 354–363. https://doi.org/10.1145/2525194.2525304
31. Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An efficient dwell-free eye gaze-based text entry system. *Proceedings of the Asia Pacific Conference on Computer Human Interaction* (APCHI '13), 215–220. https://doi.org/10.1145/2525194.2525288
32. R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for Text Entry Research: An Evaluation of MSD and KSPC, and a New Unified Error Metric. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI '03), 113–120. https://doi.org/10.1145/642611.642632
33. Oleg Špakov and Darius Miniotas. 2004. On-line adjustment of dwell time for target selection by gaze. *Proceedings of the Nordic Conference on Human-computer Interaction* (NordiCHI '04), 203–206. https://doi.org/10.1145/1028014.1028045
34. Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash Away!: Longitudinal study of fast text entry by eye gaze. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA '08), 19–26. https://doi.org/10.1145/1344471.1344476
35. David J. Ward, Alan F. Blackwell, and David J. C. MacKay. 2000. Dasher—A data entry interface using continuous gestures and language models. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST '00), 129–137. https://doi.org/10.1145/354401.354427
36. Ernst H. Weber. 1834. *De pulsu, resorptione, auditu et Tactu. Annotationes, Anatomicae et Physiologicae.* Koehler, Leipzig, Germany.
37. Jacob O. Wobbrock. 2007. Measures of text entry performance. *Text Entry Systems: Mobility, Accessibility, Universality*. Morgan Kaufmann, 47–74.
38. Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for nonparametric factorial analyses using only ANOVA procedures. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '11), 143–146. https://doi.org/10.1145/1978942.1978963
39. Jacob O. Wobbrock and Brad A. Myers. 2006. Analyzing the input stream for character- level errors in unconstrained text entry evaluations. *ACM Transactions on Computer-Human Interaction* 13, 4: 458–489. https://doi.org/10.1145/1188816.1188819
40. 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. *Proceedings of the ACM Symposium on User Interface Software and Technology* (UIST '03), 61–70. https://doi.org/10.1145/964696.964703
41. Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. 2008. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. *Proceedings of the ACM Symposium on Eye Tracking Research & Applications* (ETRA '08), 11–18. https://doi.org/10.1145/1344471.1344475
42. Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand writing on stylus keyboard. *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '03), 97–104. https://doi.org/10.1145/642611.642630
43. Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: Reimagining keyboard interaction.

Communications of the ACM 55, 9: 91–101. https://doi.org/10.1145/2330667.2330689

44. Shumin Zhai, Per Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the Iphone: From the laboratory to the real world. In *Extended Abstracts of the ACM Conference on Human Factors in Computing Systems* (CHI EA '09), 2667–2670. https://doi.org/10.1145/1520340.1520380
45. 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 ACM Symposium on Eye Tracking Research and Applications* (ETRA '12), 341–344. https://doi.org/10.1145/2168556.2168632