2018

# Dwell-free input methods for people with motor impairments

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

Dissertation

# DWELL-FREE INPUT METHODS FOR PEOPLE WITH MOTOR IMPAIRMENTS

by

**WENXIN FENG**

B.E., Renmin University of China, 2012

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2018

Approved by

First Reader

Margrit Betke, Ph.D.
Professor of Computer Science

Second Reader

Stan Sclaroff, Ph.D.
Professor of Computer Science

Third Reader

Carlos H. Morimoto, Ph.D.
Assistant Professor of Computer Science
University of São Paulo

## Acknowledgments

First and foremost, I would like to thank my advisor Margrit Betke for her guidance and support over the years. She gives me the freedom to work on different research topics and projects, and always believe in my decisions. She dedicated her time to provide me with valuable inputs and help me become an independent researcher. I truly appreciate her reliability and trust in me so that I can pursue my passions without hesitations.

I am very grateful to Stan Sclaroff, Carlos Morimoto, Robert Jacob, and Kate Saenko for offering their time to being on my dissertation committee. Their advice helped me better frame my research to a broader audience. I would also like to thank each of them for their insightful comments, inspiring discussions, and enthusiasm in my research.

I want to thank the National Science Foundation (grants 0713229, 0855065, 0910908, 1337866, 1421943, and 1551572) for supporting this research. I would also like to thank all the participants involved in the user studies.

My next thank you goes to my research collaborators, Ming Chen, Andrew Kurauchi, Ajjen Joshi, Mehrnoosh Sameki, and Jiangnan Zou, for the enjoyable and productive time together. I thank each of them for the shared experiences and fruitful collaborations.

Thanks to all the members of image and video computing group at Boston University, including Qinxun Bai, Sarah Bargal, Mikhail Breslav, Fatih Cakir, Danna Gurari, Kun He, Shugao Ma, John Magee, Xingchao Peng, Liliana Lo Presti, Vasili Ramanishka, Elham Saraee, Diane Theriault, Ben Usman, Huijuan Xu, Jianming Zhang, and many others for the great discussions, valuable feedback of practice talks, and conversations about random topics.

I would like to thank my internship managers and advisors for broadening the boundaries of my research. I appreciate Jehan Hamedi and Dexter Ang for having me to turn research ideas into products. I am very grateful to Philip Quinn and Shumin Zhai for fostering my professional and personal growth.

I am deeply grateful to my parents for their unconditional care and support from the very beginning of my life.

Finally, a special thank to Yue Pu, my husband, who is always there for me. There are ups and downs in this long journey, and his company and love make me determined in every step forward.

# DWELL-FREE INPUT METHODS FOR PEOPLE WITH MOTOR IMPAIRMENTS

**WENXIN FENG**

Boston University, Graduate School of Arts and Sciences, 2018

Major Professor: Margrit Betke, Ph.D., Professor of Computer Science

ABSTRACT

Millions of individuals affected by disorders or injuries that cause severe motor impairments have difficulty performing compound manipulations using traditional input devices. This thesis first explores how effective various assistive technologies are for people with motor impairments. The following questions are studied: (1) What activities are performed? (2) What tools are used to support these activities? (3) What are the advantages and limitations of these tools? (4) How do users learn about and choose assistive technologies? (5) Why do users adopt or abandon certain tools? A qualitative study of fifteen people with motor impairments indicates that users have strong needs for efficient text entry and communication tools that are not met by existing technologies.

To address these needs, this thesis proposes three dwell-free input methods, designed to improve the efficacy of target selection and text entry based on eye-tracking and head-tracking systems. They yield: (1) the Target Reverse Crossing selection mechanism, (2) the EyeSwipe eye-typing interface, and (3) the HGaze Typing interface. With Target Reverse Crossing, a user moves the cursor into a target and reverses over a goal to select it. This mechanism is significantly more efficient than dwell-time selection. Target Reverse Crossing is then adapted in EyeSwipe to delineate the start and end of a word that is eye-typed with a gaze path connecting the intermediate characters (as with traditional gesture typing). When compared with a dwell-based virtual keyboard, EyeSwipe affords higher text entry rates and a more comfortable interaction. Finally, HGaze Typing adds head gestures to gaze-path-based text entry to enable simple and explicit command activations. Results from a user study demonstrate that HGaze Typing has better performance and user satisfaction than a dwell-time method.

# Contents

# List of Tables

# List of Figures

# List of Abbreviations

| | | |
|---|---|---|
| ANOVA | ...... | Analysis of Variance |
| ALS | ...... | Amyotrophic Lateral Sclerosis |
| CPM | ...... | Characters Per Minute |
| DTW | ...... | Dynamic Time Warping |
| MD | ...... | Muscular Dystrophy |
| MS | ...... | Multiple Sclerosis |
| MSD | ...... | Minimum String Distance |
| NCC | ...... | Normalized Cross-Correlation |
| PALS | ...... | People with ALS |
| WPM | ...... | Words Per Minute |

# Chapter 1

# Introduction

Motor impairments can result from degenerative neurological diseases, such as Amyotrophic Lateral Sclerosis (ALS), Muscular Dystrophy (MD), or Multiple Sclerosis (MS); or from brain and spinal cord injuries, for example, due to motor vehicle accidents and stroke. Worldwide, millions of individuals are affected by disorders or injuries that cause severe motor impairments. MS affects more than 2.3 million people worldwide, and every hour someone new is diagnosed (National Multiple Sclerosis Society). The estimated numbers of individuals with spinal cord injuries range from 236 to 4,187 per million (Witiw and Fehlings, 2015).[1]

The lack of muscle control caused by these diseases makes it challenging for people with severe motor impairments to exert any force on objects. Individuals with partial mobility have difficulties performing compound manipulations, for example, grasping, holding, moving, and pressing buttons on a computer mouse. Various assistive technology tools, devices, and services have been developed to help people with motor impairments as a means of communication – connecting individuals to the computer and the Internet. This access provides people with motor impairments a much-needed opportunity to reveal their talents and creativity, and communicate with friends and family.

Investigations on how people with motor impairments use and evaluate the wide-range of assistive technologies are limited. This thesis explores this question, es-

[1] The estimation of paralysis prevalence varies widely due to different data collection and analysis methodologies.

pecially the evolving adoption of assistive technologies of people with degenerative neurological diseases. Then, with a focus on input techniques, this thesis develops techniques that improve the efficacy of the target selection and text-entry interfaces for people with motor impairments using eye-tracking and head-tracking systems.

## 1.1 Background

People with motor impairments use various assistive-technology tools, systems, and devices to assist them in communicating, taking control of their actions, and living life to the fullest. An essential part of assistive technology involves human-computer interaction – tools that assist individuals in accessing computers and the Internet, and take part in and contribute to activities that connect them to the wider world.

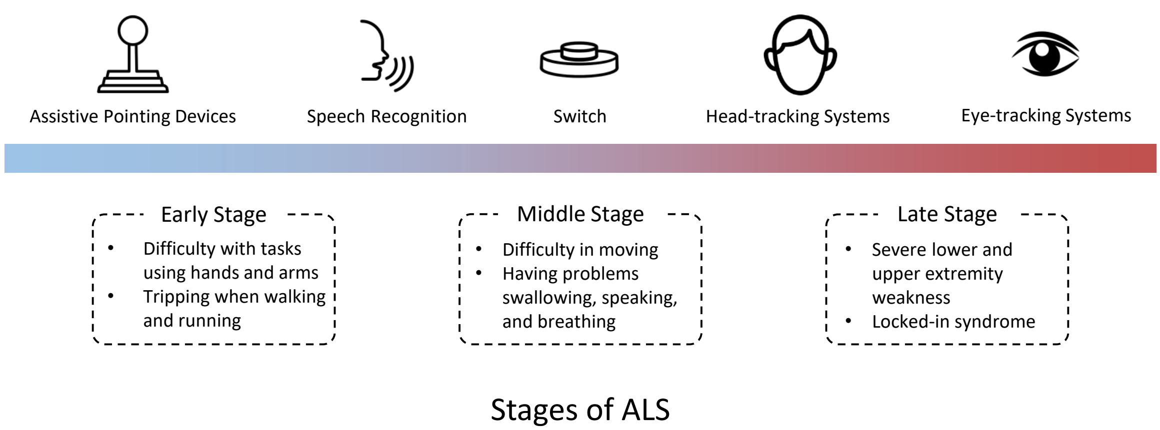



**Figure 1·1:** Assistive technologies used in different stages of ALS.

Individuals with progressive neurological diseases, such as ALS, choose different assistive technologies according to their progression stages (Lancastre; ALS Association). The examples of assistive technologies used in different stages of ALS are shown in Figure 1·1. In the early stage of ALS, tasks requiring control precision become hard due to muscle weakness of hands and arms. The computer mouse is replaced by a joystick or trackball to reduce the control effort. Speech recognition enables users to perform text-entry tasks and supports target selection at the same time. As the muscle atrophy spreads to other parts of the body, people with ALS encounter problems with moving, swallowing, forming sounds, and even breathing. In the middle and late stages, they choose systems requiring minimal muscle movement, such as switches, head-movement-based systems, and gaze-based systems.

Among various assistive technologies, eye-tracking and head-tracking systems involve few muscles to generate input and can therefore be used throughout different stages of degenerative neurological diseases. Gaze-based systems place the mouse pointer directly at the user's point of gaze on the screen (Majaranta et al., 2009). Head-movement-based systems, on the other hand, map the initial position of a facial feature, such as nose, mouth, or a reflective dot attached to the user's head, to the mouse pointer (Betke et al., 2002). Both types of systems follow the movements of the selected feature (gaze or facial features) to assist individuals in moving a cursor, writing text, or navigating the Web.

Eye-tracking and head-tracking systems can only provide single continuous input, which can cause the Midas Touch problem (Jacob, 1990). To address this issue, many target selection and text entry interfaces use a dwell-time method: maintaining the cursor in a button or key region for a given period of time to select a button or a key on a virtual keyboard. However, dwell-time selection for text entry is slow since the dwell period needs to be sufficiently long to prevent unintentional selections (Majaranta and Räihä, 2002), and is typically between 0.4–1 second for eye-tracking systems and about one second for head-tracking systems. Additionally, people with motor impairments may find dwell-time selection challenging if they experience spasm symptoms in their eye or neck muscles, causing them to have difficulty in holding the mouse cursor still.

Previous target selection research has focused mostly on the use of a regular computer mouse, and research on dwell-free selection methods for eye-tracking and head-tracking systems are rare. As for text entry, Kristensson and Vertanen (2012) showed the potential of dwell-free eye typing in a pilot experiment. They simulated a dwell-free virtual keyboard interface for which users only had to look at the vicinity of the letters in the words of a phrase. They showed that a considerable speed gain compared to traditional eye-typing systems can be achieved, assuming that the dwell-free text entry interface is well-implemented.

Combination gaze and head inputs can provide opportunities for efficient and accurate pointing and typing. In a gaze-based system, the user can quickly point to a target from the previous fixation by a saccade. However, gaze input can be considered unstable because it is difficult to accurately fixate on the desired position due to unconscious eye movements. These jittering movements, along with eye-tracking-system errors, make the corresponding input data noisy, which is problematic for fine pointing tasks. Previous work (Špakov et al., 2014; Kurauchi et al., 2015; Kytö et al., 2018) has shown that adding accurate and stable head movements to gaze-based systems allow faster and more precise target pointing and selection. How to best integrate head and gaze inputs for text entry is an unexplored research question.

## 1.2 Contributions

This thesis first contributes to the understanding of the usage and adoption of assistive technologies by people with motor impairments caused by progressive neurological diseases. The results of a qualitative study disclose how individuals with severe motor impairments choose and use assistive technologies, and how they assess their effectiveness. The results also suggest possible solutions to existing problems and future research directions (Feng et al., 2018).

Contributions are also made by the proposal of three dwell-free input methods:

- Target Reverse Crossing is an alternative selection mechanism that enables a user to control the mouse pointer by first entering the target region and then leaving it by crossing a certain edge. This method avoids the possibly difficult-to-fulfill requirement that people with motor impairments must hold the cursor in a target region using a head-tracking system (Feng et al., 2014).

- The EyeSwipe interface adapts the method of swipe-based text entry on touch screens to gaze-based systems. The interface does not require the user to look at each letter of a word to be typed. It is sufficient for the user to look at the vicinity of the key on the virtual keyboard screen. EyeSwipe uses reverse crossing to explicitly select the first and last letter of a word. While the user's gaze is swiping through the keyboard, EyeSwipe suggests candidate words dynamically with pop-ups above the key (Kurauchi et al., 2016).

- HGaze Typing is a novel bi-modal text entry system that combines gaze inputs and head gestures. The system effectively integrates head and gaze inputs so that the two modi do not interfere with each other. HGaze Typing uses simple head gestures to perform text entry tasks (e.g., selection and deletion) and associates navigation and depicts letter sequences (word paths) with gaze inputs. HGaze Typing takes advantage of the speed of eye movement and the accuracy of head movement. Furthermore, it enables a wider set of interactions than would be possible with either modes alone.

## 1.3 Thesis Overview

The remainder of this thesis is organized as follows. Chapter 2 provides an overview of the assistive technologies designed for or used by people with motor impairments and describes a qualitative study assessing the usage and adoption of these technologies.

The literature on input methods for gaze-based and head-movement-based systems is introduced in Chapter 3. This includes target selection methods and text entry interfaces.

Chapter 4 presents the proposed dwell-free selection mechanism Target Reverse Crossing. Chapter 5 describes EyeSwipe, the proposed text entry method that uses gaze paths. Chapter 6 presents HGaze Typing, the proposed text entry interface combining gaze inputs and head gestures. Finally, Chapter 7 summarizes the main findings and contributions of this thesis and proposes future research directions.

# Chapter 2

# Exploration of Assistive Technologies Used by People with Motor Impairments

When a technology is proposed for accessibility purposes, it is often tested by itself, using benchmarks or qualitative user studies, and not in comparison with other options. To provide a broad understanding of the evolving use of assistive technologies by people with quadriplegia, we conducted a qualitative study (Feng et al., 2018) with a focus on the following questions: (1) What are the activities that individuals with quadriplegia regularly perform on computing devices? (2) How do people with quadriplegia use assistive technologies to perform these activities? (3) What benefits and problems exist with their assistive technologies? (4) How do users learn about, choose, and adopt assistive technologies?

This chapter first gives an overview of assistive technologies designed for or used by people with severe motor impairments to access computers. Then, it presents a qualitative study on how users with severe motor impairments choose and use assistive technologies and how they assess their effectiveness. The results provide the following findings: (1) Users with severe motor impairments have limited opportunities to learn about assistive technology; (2) They strongly desire better mechanisms for text entry (text entry is a much more important task for them than any other computer activity); (3) The promise of personalization in assistive technologies has not come to fruition, although personalization would greatly benefit them; and (4) Adaptive assistive systems that address the changing needs of people with degenerative motor impairments also do not exist for the study participants but are critically needed.

## 2.1 Assistive Technologies for People with Motor Impairments

This section describes widely used devices, systems and interfaces, as well as research prototypes for assistive use. It is worth noting that many assistive technologies can be categorized into two or more types listed in this section.

### 2.1.1 Assistive Devices and Systems

***Alternative pointing devices.*** Users with moderate motor disabilities may not be able to use a traditional computer mouse and may benefit from alternative devices that control the mouse pointer on the screen, such as trackballs and joysticks (Fuhrer and Fridie, 2001). Users move their fingers, palm, or thumb to rotate the trackball and thus control the mouse pointer. Although joysticks were not initially designed for people with motor impairments, some users from this population benefit greatly from this device due to its short control distance. A stylus is also an effective pointing tool for touchscreen computing device users with motor impairments (National Multiple Sclerosis Society).

***Speech and sound recognition systems.*** Speech recognition software like Dragon Naturally Speaking (Nuance Communications) is widely used by individuals with motor impairments who have the ability to speak but have problems with controlling a traditional mouse or typing on a physical keyboard. It can significantly speed up their text input (Hawley, 2002) and be a means of navigating a computer (Sears et al., 2003). Sears et al. (2001) showed that people with spinal cord injuries had the same input efficiency as traditional users when using speech recognition software. Sound recognition systems have also been proposed for non-verbal individuals with severe motor impairments. Given a sound that the user makes, the systems generate binary inputs for computer access and communication. Some research projects have investigated the effect of non-speech vocal parameters including pitch and loudness (Igarashi and Hughes, 2001; Mihara et al., 2005; Harada et al., 2008).

***Camera-based interaction systems.*** Camera-based mouse-replacement systems provide inexpensive, non-intrusive computer access for individuals who can control movements of their head, facial features, or fingers, even if these motions are very small. Some use head-mounted cameras (Evans et al., 2000), or follow a reflective dot attached to the user's head like HeadMouse Nano (Origin Instruments Corporation) and SmartNav (NaturalPoint). The Camera Mouse (Betke et al., 2002) and Sina (Manresa-Yee et al., 2008) are examples of non-contact systems that track the user's movements with a video camera and translate them into the movements of the mouse pointer on the screen. Some camera-based systems also help locked-in users who can only blink, furrow or raise their eyebrows, or open their eyes fully, to perform tasks such as mouse control and text entry (Lombardi and Betke, 2002; Missimer and Betke, 2010; Krapic et al., 2015).

***Gaze-based interaction systems.*** Gaze-based interaction systems, or eye-tracking systems, follow eye movements and assist individuals in moving a cursor, writing text, or navigating through the Web. Gaze-based interaction systems commonly used by people with quadriplegia are infrared systems (Tobii Technology; The Eye Tribe) that illuminate the user's eyes with infrared lighting and track the pupil-corneal reflection (Duchowski, 2007). Emerging light-weight eye tracking systems based on an electrooculogram (EOG) can adapt to changes in ambient light and allow long-term data storage (Bulling et al., 2009). The accuracy of eye trackers depends on the calibration (Majaranta and Bulling, 2014), which usually requires the user to keep his or her head almost still and fixate on a sequence of calibration points. An eye tracker can be categorized as head-mounted or remote (non-contact).

Head-mounted eye trackers can adapt to users' head movements while remote ones may require more frequent recalibration.

***Switch-based systems.*** Switch-based systems, paired with scanning assistive software where buttons or commands are sequentially highlighted, are used by many non-verbal individuals with quadriplegia to initiate computer commands (AbleNet Switches; Don Johnston Sensor Switch). Users activate switches by performing actions such as touching, pressing, or squeezing. Two or more switches are sometimes used for more efficient and flexible computer control.

***Tongue and breath-based systems*** Some people with quadriplegia benefit from tongue-computer interfaces that use devices placed in or around the mouth (Vaidyanathan et al., 2007; Kim et al., 2013). Breath-based interfaces that use airflow sensing devices, such as sniff-detection interfaces, can also serve as alternative input tools (Surdilovic, 2006; Plotkin et al., 2010).

***EEG and EMG systems.*** Communication methods that are based on feature analysis in electroencephalograms (EEG) and cursor-control systems that use implantation of electrode arrays (Keirn and Aunon, 1990; Pregenzer and Pfurtscheller, 1999; Wolpaw et al., 2002; Wu et al., 2004) have shown promise in lab-setting experiments. Brain interfaces are appropriate choices for individuals with locked-in syndrome, who cannot even control their eye movement (Mappus IV et al., 2009). Visual and auditory brain-computer interfaces that show high speed and accuracy have been studied in recent years (Gao et al., 2014). Electromyography (EMG) signals are generated by muscle activity, which can be measured by intramuscular or surface electrodes. Surface EMG (sEMG) sensors do not require implantation and can be placed on multiple muscle sites, including the face, neck (Williams and Kirsch, 2008), and forearm (Choi et al., 2013).

### 2.1.2 Assistive Interfaces

***Assistive pointing interfaces.*** Many interfaces have been proposed to minimize unwanted errors caused by a user's motor impairments while he or she is making selections on a computer. Among them, Steady Clicks (Trewin et al., 2006) tries to minimize slipping and accidental mouse clicks by "freezing" the mouse pointer upon clicking. Other techniques or interfaces, like the bubble cursor (Grossman and Balakrishnan, 2005), goal crossing (Wobbrock and Gajos, 2008), the angle mouse (Wobbrock et al., 2009), enhanced area cursors (Findlater et al., 2010), PointAssist (Salivia and Hourcade, 2013), and, most recently, Target Reverse Crossing (Feng et al., 2014) have been proposed to help computer users with motor impairments focus on targets and avoid distractors that may influence the pointing task. These interfaces have the goal of enabling users to select a target on a computer screen efficiently and accurately.

***Assistive text-entry interfaces.*** Assistive devices and systems described in Section 2.1.1 usually have assistive text-entry interfaces and specialized keyboard layouts. CHANTI (Sporka et al., 2011), for example, is a text-entry application working with non-verbal vocal inputs. Its keyboard has four voice keys and a $4 \times 4$ scanning board. For scanning-based interfaces, an ambiguous keyboard, i.e., a keyboard where each key is related to many letters, can improve the text entry rate by reducing "key press" numbers (Mackenzie and Felzer, 2010).

A highly effective feature of assistive text-entry interfaces is word prediction. A user can select a recommended word based on his or her input of the initial segments of the word so that the number of keystrokes needed for composing a message is significantly decreased (Trnka et al., 2009). The Reactive Keyboard (Darragh et al., 1990) is an instance of word prediction research that uses previous words and contents in the message and a tree-structured language model to predict words. Dasher (Ward and MacKay, 2002), a popular dwell-free typing method, vertically arranges the letters on the side of the screen, devoting larger areas for more likely letters, based on the language model. As the user selects letters, the keyboard dynamically changes, moving selected letters horizontally and collecting them into words. We proposed a text-entry method that is also based on a language model but uses a fixed on-screen keyboard (Kurauchi et al., 2016). Words are predicted based on the trajectory of a user's gaze while the user scans through the keyboard.

***Specialized user interfaces***. Many user interfaces are designed by and developed for people without disabilities, so the resulting interfaces usually do not fit the situation of individuals with motor impairments. Few research projects have investigated the possibility of developing systems that can automatically generate user interfaces for users with motor impairments. Among them, Supple can generate a user-customized interface automatically, and Supple++ asks users to go through a one-time performance test so that the system can provide a personalized user interface (Gajos et al., 2008, 2010).

Self-adaptive interfaces have been explored in traditional user interfaces (Jameson, 2003) and used as accessibility aids (Sloan et al., 2010; Heron et al., 2013). The Dynamic Keyboard (Trewin, 2004), for example, could adjust fundamental keyboard accessibility configuration according to the user's inputs. Montague et al. (2014) proposed a touchscreen gesture recognizer based on in-the-wild user data to accommodate people with motor impairments. Since adaptive interfaces collect user data and accommodate individual needs automatically, user control and privacy of these interfaces need to be considered (Trewin, 2000).

## 2.2 Qualitative Studies of Assistive Technologies

Prior works explored the accessibility of commonly used computing devices through qualitative studies. Kane et al. (2009) interviewed 20 people with visual and motor impairments, followed by a diary study, to report the usage and adaptation strategies for mobile devices in everyday life. They suggested increasing the accessibility and configurability of existing mobile devices, and integrating assistive devices with mobile phones. Anthony et al. (2013) employed 187 non-commercial YouTube videos as a data source to investigate touchscreen devices used by people with motor impairments. The results showed that touchscreen devices are empowering, but that there are still accessibility concerns.

Only a few studies have investigated the use of a *broad* range of assistive technologies for people with motor impairments. Survey results from 227 people with various disabilities showed that 29.3% assistive devices were completely abandoned due to the lack of consideration of user opinion, easy procurement, poor performance, or change in user needs (Phillips and Zhao, 1993). A nationwide telephone survey in the United States (Carlson et al., 2001) showed that there was a general awareness of assistive technologies but that the amount of information on obtaining them was perceived to be limited. Shinohara and Wobbrock (2011) investigated the affects of social interactions on using assistive technologies and came to the conclusion that assistive devices should be integrated into mainstream technologies, taking social acceptability into consideration. Some of the findings of our study reinforce the conclusions of these prior works, as we will describe in Section 2.5.

## 2.3 Study Method

We conducted semi-structured interviews to explore how people with severe motor impairments caused by degenerative neurological diseases assess and adopt a broad range of assistive technologies.

### 2.3.1 Participants

A total of 15 individuals with quadriplegia (5 females and 10 males) with ages between 28-67 participated in our study (Table 2.1). The participants had various degenerative neurological diseases: Multiple Sclerosis (MS), Multiple System Atrophy (MSA), Krabbe's disease (KD), Machado-Joseph disease (MJD), and Amyotrophic Lateral Sclerosis (ALS). They had the cognitive ability to provide informed consent to this study. All of the participants were living in the United States, although some of them originally came from other countries.

Twelve interviewees (P1–P10, P12–P13) were residents at The Boston Home or

**Table 2.1:** Information on study participants and setting. Functional scores for mobility (lower extremity function); for being able to carry, move, and handle objects (CMHO or upper extremity function); and for voice use are given from 7 (no impairment) down through 1 (100% impaired) based on U.S. government codes (Medicare Learning Network). The scores for P1-P10 were measured by the care center and the scores for P11-P15 were self-reported.

| ID | Age | Gender | Health Condition | Mobility | CMHO | Voice Function | Family Members Present | Demo Given |
|---|---|---|---|---|---|---|---|---|
| 1 | 62 | F | Multiple sclerosis | 1 | 1 | 4 | 0 | Y |
| 2 | 60 | M | Multiple system atrophy | 1 | 3 | 2 | 4 | N |
| 3 | 43 | F | Multiple sclerosis | 1 | 3 | 2 | 0 | Y |
| 4 | 52 | M | Krabbes disease | 1 | 3 | 2 | 1 | N |
| 5 | 67 | F | Multiple sclerosis | 1 | 1 | 5 | 0 | N |
| 6 | 53 | M | Multiple sclerosis | 1 | 1 | 4 | 0 | Y |
| 7 | 54 | M | Multiple sclerosis | 1 | 3 | 3 | 1 | N |
| 8 | 55 | F | Multiple sclerosis | 1 | 1 | 4 | 0 | N |
| 9 | 54 | M | Multiple sclerosis | 1 | 3 | 4 | 0 | N |
| 10 | 35 | M | Machado Joseph disease | 2 | 3 | 1 | 0 | Y |
| 11 | 67 | M | Amyotrophic lateral sclerosis | 1 | 3 | 6 | 2 | N |
| 12 | 41 | M | Multiple sclerosis | 1 | 1 | 5 | 0 | Y |
| 13 | 46 | M | Amyotrophic lateral sclerosis | 1 | 1 | 1 | 0 | Y |
| 14 | 28 | F | Amyotrophic lateral sclerosis | 4 | 2 | 4 | 1 | N |
| 15 | 37 | M | Amyotrophic lateral sclerosis | 3 | 2 | 4 | 0 | Y |

the Leonard Florence Center for Living, and three interviewees (P11, P14, and P15) lived with their families. The two institutes are long-term care facilities for people with degenerative neurological diseases.

Participants were recruited through references, emailing, and snowball sampling. For example, the director of rehabilitation services at The Boston Home, who is a speech-language pathologist, suggested residents to participate in this study. Some of these residents then also recommended their friends at the institution to participate in the study.

In addition to the 15 study participants, the director of rehabilitation services at The Boston Home provided information about the assistive technologies used at the institute.

### 2.3.2 Interview Setting and Procedure

Semi-structured interview guidelines based on the following five themes were designed, with additional related sub-topics emerging during the interviews:

- What participants usually do with computing devices;
- The assistive technologies tried or used for a period, or currently in use, and the advantages and disadvantages of these technologies;
- How participants learned about and chose these assistive technologies;
- Why assistive technologies that had been tried or used in the past were abandoned;
- For assistive technologies in use, the participants' opinions about and wishes for their current assistive tools.

One 30–60 minute semi-structured interview was conducted with each study participant in person. There was only one shorter interview: one participant felt fatigue after 20 minutes of the interview, and the remaining part of the interview was conducted at another time. The interview locations were selected to be familiar to the participants (their home, dormitory, or residence) in order to create a natural environment for conversation.

Six of the participants had limited speech abilities (P2–P4, P7, P10, and P13). Two participants (P3, P10), who could not speak, responded to our questions by typing simple words on an iPad. One participant (P13) answered our questions by typing with the Head Mouse and the Dasher text-entry interface. The others (P2, P4, P7) formed single words or made soft sounds to respond to the interview questions. Three of them (P2, P4, P7) were accompanied by family members during the interview. When family members were present, we first asked the participants questions directly and waited for their response. Sometimes they answered with several single words. When family members stepped in and answered a question, we would wait for the participants' reaction. They either signaled their opinions by nodding or shaking their heads, or said "yes" or "no," sometimes followed by a short explanation. Two participants (P11, P14) with speech ability also involved their family members in the interviews. We emphasized that they should answer the questions first. The family members could add additional comments, which needed to be confirmed by the participants.

After the interview, seven participants (P1, P3, P6, P10, P12, P13, P15) chose to demonstrate their usage of computer and assistive technologies, while the other participants did not have time or energy to show their technologies.

### 2.3.3 Data Recording and Analysis

In all interview sessions with the study participants, handwritten or typed notes were taken. The interviews were audio-recorded and transcribed. During the voluntary demonstration sessions, at least one observer took detailed notes, including how participants used assistive technologies and computers, and how they performed daily tasks.

During the data analysis process, both the five pre-determined interview themes (Section 2.3.2) and sub-topics obtained by a data-driven method (Corbin and Strauss, 1998) were used to categorize, i.e., code, statements of interest from notes and transcriptions. Two researchers coded the statements independently, based on the relevance to the themes. If a new topic emerged in the data analysis process, the researcher would generate a new code. After the first round of coding, the researchers discussed the topics that emerged from them and combined similar topics into the same category. The researchers also reviewed coding disagreements until they arrived at a consensus. In the third round of coding, the researchers found that no new topics appeared. This indicates that the analysis process achieved theoretical saturation (Holton, 2007).

## 2.4 Results

In this section, for each topic that emerged from the data analysis described above, we start by providing a summary of the issues that our study participants brought up in their interviews. After we present a result, it is reinforced by related quotes. We label answers from participants with motor impairments with "P" followed by their ID, and quotes from a family member with "F," followed by the participant ID. The answers from the director of rehabilitation of The Boston Home are labeled with "D."

### 2.4.1 Activities Using Computing Devices

Study participants reported that they engage in or have engaged in various activities with computers.

### Use of Computing Devices

All of the study participants have experience using a computer, and fourteen out of fifteen currently use a computer or an iPad. Two of them (P3, P10) use iPads with a stylus while the others use computers. As for the frequency of usage, thirteen out of fifteen participants use a computer or iPad every day for at least one hour. On average, they spend about four hours with computing devices every day. One participants (P6) uses a computer only when he feels good.

*"My symptom does not allow me to use [a] computer [so I use an iPad instead]." - P10*

*"[I use a computer] only in the morning when I feel very good, for about half an hour." - P6*

One of the interviewees (P8) used to work with computers, but now, due to her rapidly-developing fatigue and preference for other activities, she does not use them anymore.

### Activities on Computing Devices

The interviews revealed that participants engaged in nine types of activities on computers or tablets, as summarized in Figure 2·1.

Receiving and responding to emails are the main use of computing devices. Only two of the participants (P11, P15) are still working and sometimes use email for work, while the others use email to contact friends and family. Besides email, two participants (P2, P7) use video-chat to connect with family members and friends regularly.

Seven participants are Facebook users, and none of them use other social networks. Two participants (P6, P7) do not use any social network since they do not want to show their current situation to others.

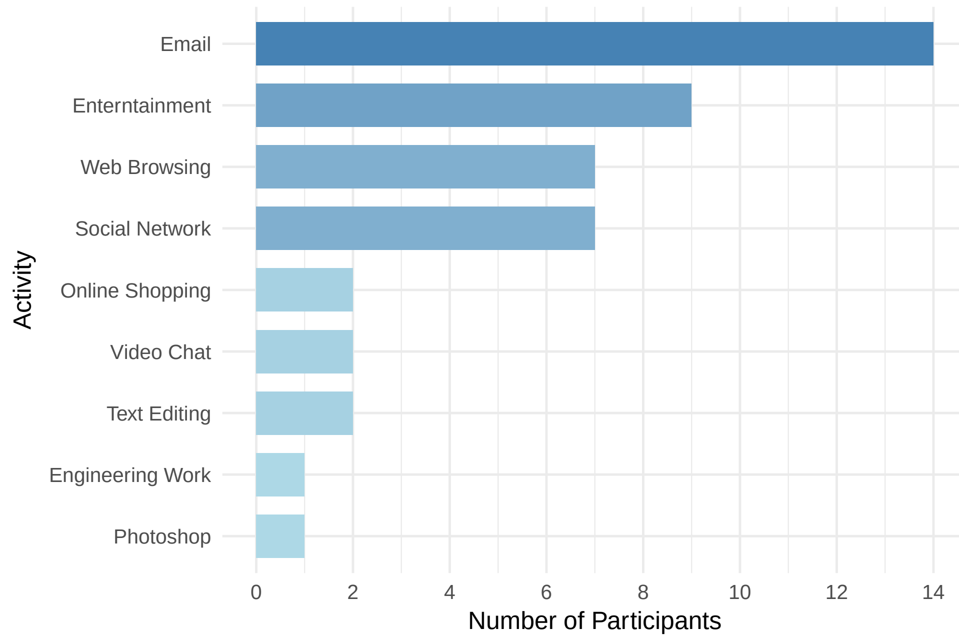



**Figure 2·1:** Number of participants that engage in activities on computing devices.

*"I am a very private person. I don't want to use Facebook now since I don't want people know where I am and feel sorry about me." - P6*

Computers and touchscreen devices are important channels for the study participants to access information. Seven participants surf the Internet for news, blogs, or other things they are interested in.

*"He can no longer hold things to read them, so everything he reads now is electronic, even the newspaper. He used to be able to read the daily newspaper, now that's all down." - F11*

Entertaining activities, such as reading books, listening to the music, watching videos, and playing computer games, are popular with nine out of fifteen study participants. Two participants mentioned that online shopping provided them means to choose their necessities on their own.

Two of the participants use assistive technologies to work. One participant (P11) is a landscape architect, who uses software programs including Microsoft Office and

Adobe Photoshop, paired with assistive pointing interfaces, to edit pictures. The other participant (P15) works as an engineer and relies on a joystick to control the computer mouse.

#### Abandoned Activities

Ten participants cannot continue at least one activity on the computer or iPad that they could do previously. Many of the abandoned activities, for example, playing computer games, require precise mouse control, or take too much effort, for example, entry of long texts. Participants wished they still had the ability to perform the abandoned activities. They hoped for new technologies that would enable them to do these activities again.

*“[If I were able to use the computer] I [would] use it to do emails and computer games.” - P8*

*“He was watching a lot of videos via YouTube in the past. However, he cannot type or click and cannot use it anymore.” - F2*

### 2.4.2 Usage, Advantages and Limitations of Assistive Technologies

Study participants reported using or having used various computing technologies and described encountered difficulties.

#### Assistive Technologies Used or in Use

The study participants who still use computers all rely on assistive technologies. Detailed information on their assistive technology use is given in Table 2.2. The table describes which technologies the participants tried out (T), currently use (U), and abandoned due to the progression of their disease (A).

Pointing devices (joysticks, trackballs, styluses) are the most popular assistive hardware for the study participants; they are used by seven participants. Assistive

**Table 2.2:** Assistive technologies tried (T), abandoned (A), and in use (U)

| Assistive Technologies | P1 | P2 | P3 | P4 | P5 | P6 | P7 | P8 | P9 | P10 | P11 | P12 | P13 | P14 | P15 | Tried/Abandoned | In Use |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alternative pointing devices | - | U | U | U | U | T | T | - | U | U | T | T | - | T | U | 5 | 7 |
| Speech recognition systems | U | - | - | T | U | U | - | - | T | - | A | T | - | U | - | 4 | 4 |
| Camera-based systems | - | - | - | - | - | A | - | - | - | - | A | U | U | - | - | 2 | 2 |
| Gaze-based systems | - | - | - | T | - | U | - | - | - | - | U | A | T | U | - | 3 | 3 |
| Assistive pointing interfaces | U | - | - | - | U | - | - | - | - | - | - | U | U | - | T | 1 | 4 |
| Assistive text-entry interfaces | - | - | U | - | - | U | - | - | - | U | U | U | U | - | T | 1 | 6 |
| Specialized graphical interfaces | - | - | - | - | - | U | - | - | - | - | U | - | - | - | - | 0 | 2 |
| Abandoned | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 0 | | |
| Tried | 0 | 0 | 0 | 2 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 2 | 1 | 1 | 2 | | |
| In Use | 2 | 1 | 2 | 1 | 3 | 4 | 0 | 0 | 1 | 2 | 3 | 3 | 3 | 2 | 1 | | |

text-entry interfaces are the most used assistive interfaces.

Nine interviewees tried camera-based, gaze-based, or speech-based systems, and seven continue to use them. Four participants (P1, P5, P6, P14) currently use speech recognition software for text entry. Two of them (P1, P5) use the Mouse Grid software for target selection when using speech recognition, which can reduce the difficulty of moving the mouse pointer to a specific point on the screen. The other two participants (P6, P14) use gaze-based systems for computer control. Head-tracking systems are used by two participants (P12, P13). The system can be used to support input to alternative pointing interfaces and text-entry interfaces, such as Point-N-Click (Polital Enterprises) and Dasher (Ward and MacKay, 2002).

### Advantages of Assistive Technologies in Use

***Reduced control efforts.*** Technology that does not require the use of hands or that reduces the use of hands is helpful for many users with quadriplegia. Three participants noted that trackballs and joysticks are easier to grip and operate than a regular computer mouse. Also, the distance that the hand has to move is shortened. Gaze-based, camera-based, and speech-based systems provide hands-free interaction environments for selection and text input.

*"Using [speech recognition] software, I don't need to move the mouse cursor at all.*

*I cannot use any pointer [devices] since it is not possible for me to control [them]. So I prefer a mouse-cursor free-approach." - P6*

*"He uses the [track] ball. It seems easy for him to use it instead of moving the mouse around. It's like a mouse, instead of moving the whole thing, he just needs to move the ball." - F4*

***Customizable interfaces*** were used by three out of fifteen participants. Two participants use specialized interfaces that have large fonts and buttons, and are satisfied with them. Such interfaces are not used by the majority of the participants since most participants could read regular fonts and select regularly sized buttons. One participant (P6) uses a customized assistive interface that allows him to circumvent the need to use a computer mouse, which significantly improved his communication efficiency.

***Dynamic interfaces*** like Dasher were praised by two study participants (P12, P13). Individuals with motor impairments usually take more time and effort to complete basic tasks like target selection. Because these tasks are performed repetitively, a dynamic interface may prevent users from becoming tired or bored. One participant (P13) noted that interaction models that required waiting, such as dwell-time clicking, are monotonous and tiring.

*"Dasher is also more interesting for others to watch and takes away some of the monotony of waiting so long. It is also kind of fun - like a video game." - P13*

### Difficulties with Assistive Technologies

The study participants reported encountering difficulties in using assistive technologies for various reasons.

***The efficiency and accuracy*** of assistive technologies are considered problematic by ten participants, and the issue of communication inefficiency is raised by eight participants. The speed of human-computer interaction is affected by the nature of the particular assistive device used. For example, gaze-based and camera-based systems provide single sequential input from the user, which constrains the speed of text entry. Five of the participants noted that dwell-time typing is inefficient even with word prediction. Making corrections for typing errors is especially frustrating for them. Also, some systems were not stable enough and occasionally froze during a demonstration (P6).

*"I have used head-tracking software, which is so slow. I have to type letter by letter. Eye-tracking is very slow [as well], letter by letter." - P6*

*"It can be annoying when the [speech recognition software] doesn't recognize the right word and comes with something that's totally off what I said. Sometimes it gives you some choices, which should be the first choice, you have to say choice one, which is frustrating." - P5*

*"Thats a problem if [I] coincidently look at [a]wrong key [using the gaze-based device]." - P11*

***Control precision*** is not satisfactory according to four participants. Interaction with assistive devices may not be as precise as desired due to the user's limited control of movements and speech. Two joystick users mentioned that they sometimes missed the desired targets because of muscle tremor. Gaze-based systems keep track of the eye movement, which includes rapid saccades and fixation jitter. Although the input signal is filtered by the systems, it may not be useful for tasks like photo editing, which require precise control, as reported by one participant (P13).

*"Eye-tracking is not as precise as I need for things like Photoshop; I like to be very accurate, so precision is important." - P13*

***Fatigue caused by using assistive devices or interfaces*** was reported by six participants. The participants wanted to save their strength as much as possible when using assistive technologies. Computer interaction with certain devices and interfaces can be very tiring. For example, one participant (P11) indicated that a keyboard with large keys was strenuous for him to use since he needed to move extra distances to reach the keys. Moreover, fatigue itself affects the usage of assistive technology. Two participants (P1, P6) reported their voice became weak and unclear after a few hours, and the speech-recognition systems would stop working.

*"When I get tired, my voice volume falls and gets slower. Thus, it gets difficult for the software to detect what I am saying." - P1*

*"He had trouble getting into the right place [with the head-tracking system]. He would have trouble holding [the cursor] long enough because his neck is weak." - F11*

*"I always feel fatigue by the end of the day [using the Head Mouse], my neck is tired." - P12*

***Dependency on others*** is a concern of five study participants. Many systems still rely on others to configure and start the assistive technology. Once a device has been set up, and the appropriate software has been started, most of the study participants do not need help from others to use the computer. Some assistive systems can automatically load the user's profile and preference. When problems appear during use, the participants will ask the professional caregivers or family members for help. Immediate help may not be available in time, as described by two participants.

*"[I need help to] turn on the machine and set up. I don't need help [during the use]. Of course I want to [be independent.]" - P11*

*"[They] may require intermittent help if there is a problem they are unable to fix. Or if they accidentally get into trouble, in the sense of opening multiple programs or freezing the program." - D*

### 2.4.3 Learn about, Choose, and Adopt Assistive Technology

The study participants reported how they were introduced to specific technologies and which technologies they eventually adopted for regular use (see also Table 2.2).

**Get to Know Assistive Technologies**

In total, thirteen out of fifteen study participants obtained assistive devices from their care facilities. The twelve participants that were residents of The Boston Home or the Leonard Florence Center for Living have been exposed to assistive technologies by the caregivers there. The caregivers encourage residents with quadriplegia to try different tools, which are usually commercial products, and give them customized suggestions based on their symptoms. A study participant with ALS (P11), who was not a resident of any care center, was introduced to assistive tools at the university-affiliated hospital that provides his treatments. He also received a speech-recognition system from a research university. Two other participants (P14, P15) were introduced to assistive technologies by assistive device companies and university research teams.

*"Almost all the hardware [is] something that hasn't been seen [by new residents] in the [Boston Home] community or heard about before. So we are offering [new tools] to them. And we don't always offer the entire array [of assistive technology] to everybody, we try to customize to what the person's needs are at that moment. ... Generally, they come up open-minded and say 'I'll try anything.' And then we don't necessarily just only do one thing but we may try if different things work. And then we will make the recommendation [asking] 'Are you comfortable with that?' So usually it's coming from us recommending a specific technology for them." - D*

*"We learned about most technologies from the hospital. We got Dragon from NYU." - F11*

Some participants mentioned that their families and friends introduced assistive technologies to them. One participant (P6) said a family member designed assistive software specifically for him.

*"My brother-in-law is a computer engineer. He has developed [a] speech recognition [system] for me so that I can read and send emails easily using my voice, and its*

*predefined phrases." - P6*

Interestingly, none of the participants actively searched for technologies by themselves. One study participant (P5) said she knew nothing about head tracking or eye tracking before the caregivers introduced them to her. One individual (P7) even showed a negative attitude towards exploring new assistive technologies.

*"I fear change." - P7*

Individuals with quadriplegia sometimes find out about new technologies in the mainstream media. Mainstream media, however, introduce a very limited selection of appropriate tools because they typically describe commercial products for all customers instead of specialized tools targeted to help people with motor impairments.

*"People might rarely see or only see on the special reports about somebody who has a severe brain injury [in the media]. Recently, with the increased awareness about the ALS ice bucket challenge all over the Internet. People might have seen some of the technologies that are used by people with advanced ALS. ... If anything else besides our recommendations, it's from the mainstream media, which generally does not talk a lot about the customized access devices." - D*

### Learning to Use Assistive Technologies

Instead of learning from instruction manuals or videos, participants of our study learned how to use assistive technologies with the assistance of their friends, family members, volunteers, or therapists. For individuals who live in a care center, peer instruction is a helpful way for them to learn about assistive technology. Experienced users are usually willing to demonstrate an assistive tool to a new user.

*"First of all, [my friend] knows quite [a lot] about [assistive technology], so I am learning from her. And people in the hospital [taught me to use assistive tools]." - P11*

*"Some of the folks use the software for a long time. They are often models for*

*other residents. They may coach them or at least say 'look at what I am doing' to show them some of the basic things of the program." - D*

**Reasons for Continued Use or Abandonment**

The study participants considered the advantages and disadvantages summarized in the previous section when they selected assistive technologies (see Table 2.2). The main reason for choosing or abandoning an assistive technology was the symptoms that the participants were experiencing, which varies among individuals. Since a disease like ALS is progressive, the movement abilities of the individual changes. A participant in our study with ALS (P11) gave up all the assistive technologies that he had used and changed to gaze-based interaction about six months before our study.

*"One of the challenges is this disease lets things change so fast, so we are always dealing with changing in technologies. And [I am] trying to be ahead enough so that you are not just learning it by the time you [are] already behind." - P11*

Individuals with quadriplegia caused by neurological disease typically have accompanying symptoms like tremors, spasms, and difficulties in gripping and holding. These symptoms limit their choices of assistive technolgies. For example, P2's tremors sometimes significantly impacted his use of a trackball, as was observed during the interview with P2. Also, people with neurological diseases may have other health conditions, in addition to paralysis of their limbs, that prevent them from using specific technologies. In our study, the participants (e.g., P4) chose assistive technologies that do not trigger discomfort, and abandoned those that do not accommodate their symptoms.

*"He tried eye movement [interaction], which didn't work. His eyes don't track at the same time; they track separately, and with bifocals. So he has problem with that." - F4*

Some participants have to use medical equipment that affects their usage of assistive technology. For instance, two study participants (P11, P12) wear nasal pillow masks to support their breathing. Both participants abandoned speech recognition systems because these systems were negatively affected by the breathing noise.

*"He no longer uses Dragon. It's hard for Dragon to recognize his voice because he has to wear the nose pillow all the time, so it's a lot of effort. It's not only because of the noise of the nose pillow, but also he is breathing, he breaks his sentences in different places than what's normal. And Dragon doesn't recognize the sentence. So it makes a lot of mistakes." - F11*

The financial status of the participants and the care center is another factor that influences the choice of assistive technology. The study participants themselves weigh the price of the assistive technology and its functions. Care centers try to balance the expenses of assistive technologies and the other needs.

*"The hospital did not provide the full version of [the] Tobii [gaze tracker], only the limited functions due to the funding issue." - F11*

## 2.5 Discussion

This study found that people with quadriplegia caused by neurological diseases have strong needs for efficient text-entry and communication that are often not met. Opportunities exist for assistive technology researchers and designers to improve the options for these individuals, for example, by developing personalized or adaptive systems that could provide better support for people with quadriplegia.

### 2.5.1 Let Them Know

There are many assistive technologies designed for people with severe motor impairments, as described in Section 2.1. However, most people with these impairments had little information about existing assistive devices, as was revealed by a nationwide survey (Carlson et al., 2001) and is corroborated by our study. Our study showed that individuals with quadriplegia only have limited means to discover assistive technologies that are suitable for them. They usually try those recommended by their friends, families, or caregivers, rather than searching for online products or consult assistive technology specialists.

Although the professionals in hospitals and care centers are aware of many assistive technologies, most of their recommendations are off-the-shelf hardware devices and commercialized software, such as Tobii and Dragon. Freeware interfaces and research prototypes are typically unknown to them, let alone to the individuals with quadriplegia and their families. It would be helpful if researchers and developers promote their interfaces or applications directly to local care centers. It is also important that users with quadriplegia and caregivers pay attention to a broader range of assistive technologies and actively consult assistive technology specialists.

Mainstream media provides a platform to inform about new technologies. Unfortunately, these technologies are designed for a general audience and are often not accessible to individuals with motor impairments. Developers should consider addressing this problem, possibly by integrating assistive features into mainstream technologies, as Shinohara and Wobbrock (2011) suggested.

### 2.5.2 Desire for Efficient Text-Entry and Communication

Our study participants contact their friends and families on a regular, sometimes daily basis via email, video-chat, or social networks. They use assistive technologies to facilitate this need for communication. Interestingly, they prefer traditional emails to video-chat and social networks. Researchers could put new efforts into exploring these communication behaviors and developing innovative assistive software accordingly.

All study participants desire efficiency in typing, as most of them use email to communicate with their friends and families. They encounter many problems and feel dissatisfied with their text-entry technologies. Their current range of solutions includes speech recognition software, eye-tracking interfaces, and on-screen keyboards. Not all of the participants with quadriplegia in our study have the ability to use a speech recognition system, since their diseases affect the muscles for forming speech (Table 2.1). Those who can use a voice recognition system noted that the system makes mistakes and that the editing process is difficult for them. This also applies to study participants who use a combination of an on-screen keyboard and eye tracking as an alternative interaction method. When a user accidentally looks at the wrong key and unintentionally selects it, it takes time and efforts to revise this selection. Researchers and developers may want to continue working on improving editing interfaces for speech recognition, eye-tracking, and head-tracking systems. Easy corrections are strongly desired by people with motor impairments using these systems.

### 2.5.3 Meet Specific and Changing Demands

Study participants are not satisfied with the currently available assistive technologies. Quadriplegia caused by diseases leads not only to the loss of mobility of four limbs, but it also has various accompanying symptoms that vary among users and should also be addressed by the assistive technology. However, often the technology is not sufficiently specific for a certain person. Earlier attempts on physical capability assessments (Price and Sears, 2009) and matching measurements with technologies (Scherer and Craddock, 2002) have been made to provide personalized assistive technology solutions. Researchers and developers should consider creating a recommendation system that proposes assistive technologies based on the specific symptoms and capacities of the user, and thus meet his or her needs and increase satisfaction. Such a system could also be a supplement to the caregivers' recommendations.

For a given assistive technology, integrating performance evaluation and automatic customization would help people with motor impairments personalize their devices or interfaces effectively. This idea has been illustrated almost a decade ago with the publicly available research prototype Supple (Gajos et al., 2008, 2010) but has not found its way into commercially available products. Our study findings reinforce the importance of the idea of personalization, and we encourage researchers and product developers to put more efforts into automatic customization systems.

It is important to note that degenerative neurological diseases like ALS result in constantly changing symptoms, which, as our study shows, results in changes in the demands on the assistive technology. Prior work (Phillips and Zhao, 1993) also reported that changes in a user's needs is a general issue among people with various disabilities. It would be beneficial to design an adaptive system that can monitor, collect, and analyze data on individuals' use of assistive tools, and then accordingly change the configuration for them over time. If the current choices of assistive technologies are no longer useful, the system could automatically change the interface and suggest new devices and software based on the user's performance.

## 2.6 Conclusions

This qualitative study provided an overview of assistive technologies for people with severe motor impairments and described a qualitative user study to better understand the experiences individuals with quadriplegia have with these technologies.

We recognize that interview-based research has the limitations of small sample size, which may affect the generality of the study results. To alleviate this effect, we used three methods to recruit study participants with various gender, age, health condition, and degree of impairments. We also note that our study participants live in the northeastern United States, and this lack of geographic diversity may lead to location-biased study results.

The qualitative study revealed how limited individuals' opportunities are to discover assistive technologies, in particular, non-commercial options that may be beneficial for them. Our study also revealed the importance of text entry for our user group. The analysis suggests that because people with quadriplegia have specific needs, personalization should be considered by researchers and developers of assistive technologies. The analysis also suggests that because people with degenerative muscular disease have changing symptoms, adaptive systems should be built for them. Although the tools, found in the literature or encountered in the user study, greatly improve users' lives, much work remains to develop technology that more effectively assists individuals with severe motor impairments.

# Chapter 3

# Literature on Input Methods for Eye-tracking and Head-tracking Systems

People with motor impairments have a strong desire for communication (Section 2.5) and therefore would like to rely on efficient and accurate input interfaces. In this chapter, we discuss the rich body of literature on target selection and text entry methods, with a focus on eye-tracking and camera-based head-tracking systems.

## 3.1 Target Selection Methods

Target selection consists of two steps: pointing and selection (or clicking). Studies have been conducted to improve the selection efficiency and accessibility for traditional computer mouse inputs. For gaze-based and head-based systems, users typically dwell on a target for a predefined period of time to make a selection. Previous research investigated pointing techniques for both systems. Facial expressions and gestures were explored as an alternative selection method for head-tracking systems.

### 3.1.1 Target Selection Using Traditional Mouse

Accot and Zhai (2002) proposed goal-crossing as a selection mechanism, a click-free alternative to traditional pointing-clicking selection. A user selects an object by crossing a line target with goal crossing. This method essentially combines the two target acquisition steps, pointing and clicking, into one, and thus reduces the target selection time. Wobbrock and Gajos (2007, 2008) compared goal crossing with area pointing from an accessibility perspective, and the found that goal crossing had higher throughput for people with motor impairments than area pointing, though it had higher crossing error rates.

Ages and motor impairments affect the speed and accuracy of target selection using mouse (Riviere and Thakor, 1996; Trewin and Pain, 1999). The area cursor has a larger activation area than a regular cursor and has been designed to help older adults perform selection tasks (Worden et al., 1997). Paired with the sticky icons, which reduce the cursor's gain when the cursor is on a target, the area cursor reduced selection time by 50% when selecting small targets. Trewin et al. (2006) developed steady clicks that freezes the cursor during a clicking to prevent wrong selections.

Variations of the mouse cursor were proposed to improve the selection of small targets with dense target arrangements. The bubble cursor, instead of dynamically changing the target size as the bubble target (Cockburn and Firth, 2004), resizes the activation area according to its surrounding targets (Grossman and Balakrishnan, 2005). User study results showed that the bubble cursor significantly increased the target acquisition time and reduced the error rate in selection tasks with distractors. Findlater et al. (2010) designed and compared four enhanced area cursors utilizing magnification design and goal crossing to traditional selection and the bubble cursor. In an experiment of small target selections with dense distractors, people with motor impairments had better performance and satisfaction with two enhanced area cursors, Visual-Motor-Magnifier and Click-and-Cross cursor, than with other selection methods.

### 3.1.2 Gaze-based and Head-based Target Selection

Dwell-time is the most common approach to address the Midas Touch problem in gaze-based and head-based target selection. In gaze-based systems, the user is required to look at a target for a period of time to make a selection (Jacob, 1990; Sibert and Jacob, 2000). Camera-based head-tracking systems, such as the Camera Mouse (Betke et al., 2002), also use the dwell-time method to select a target: the user holds their head still to maintain the pointer in a target region for a certain period.

Some head-based and gaze-based systems take advantage of the facial expressions and gestures to perform selections. Lombardi and Betke (2002); Grauman et al. (2003); Krapic et al. (2015) allowed users to select a target via voluntary eye blinks or by raising their eyebrows. Surakka et al. (2004) combined gazing and frowning (detected by electrodes) for target pointing and selection, respectively. The bi-modal selection mechanism showed comparable performance to traditional mouse selection.

Some studies proposed and examined head-assisted eye pointing (Špakov et al., 2014; Kurauchi et al., 2015; Kytö et al., 2018) since head-based pointing is more stable and easier to control while not as fast as gaze-based pointing (Bates and Istance, 2003; Kytö et al., 2018). The multi-modal pointing method allows using gaze to perform a coarse long-distance jump and using head movements to make fine adjustment. User experiments demonstrated that combining the accuracy of head movements and the speed of gaze increased the efficacy of pointing.

## 3.2 Text Entry Interfaces

Eye-tracking and head-tracking systems are often combined with virtual keyboards to provide a text entry method (Majaranta and Räihä, 2002; Missimer and Betke, 2010). Efforts have been made to improve the dwell-time method or to find an alternative interface to eliminate dwelling.

### 3.2.1 Gaze-based Text Entry

Gaze-based text entry has been explored for years, and various text entry methods have been proposed. We categorize these typing interfaces based on the how a user can form an input (a letter, a word, or text with any length), and in three types: dwell-time-based interfaces, gesture-based interfaces, and continuous writing interfaces.

### Dwell-time-based interface

Many gaze-based text-entry interfaces use dwell-time to select virtual keys letter by letter. Dwell-time-based interfaces are intuitive to learn but impose a waiting period to the user, which affects his or her typing speed and text entry rate.

Word prediction is an effective feature to accelerate dwell-time-based keyboard. A user can select a predicted word or phrase with entering the first few letters of the desired word or phrase (Trnka et al., 2009; Diaz-Tula et al., 2012). However, checking the word candidate list requires additional visual search so that the user has to move the gaze to a different place. Diaz-Tula and Morimoto (2016) designed AugKey, which places the previously entered text and the predicted word close to the fixated key. User studies showed that the interface outperformed a dwell-based keyboard with word prediction by 20% in speed.

Adjustable dwell-time keyboards allow the user to change the dwell period according to their typing rhythm and result in a higher typing rate. Majaranta et al. (2009) conducted a longitudinal experiment to examine the performance of a dwell-based virtual keyboard with an adjustable dwell-time feature. The dwell-time was initially set to 100 ms, and the average user-adjusted dwell-time in the last experiment session was 282 ms; the text entry rate increased from 6.9 wpm in the first session to 19.9 wpm in the last session. In another exploratory study (Räihä and Ovaska, 2012), participants achieved typing rates higher than 20 wpm with a dwell-time of approximately 300 ms. Mott et al. (2017) proposed a cascading dwell-time method, which achieved an average typing rate of 12.4 wpm (10.6 wpm for the static dwell-time method). The method utilizes a language model to compute the probability of each key being entered next, and adjusts the dwell-time accordingly. The minimum dwell-time set by the user was never lower than 250 ms.

**Gesture-based interface**

Previous work investigated different gaze gestures to replace dwell-time selections of a character (Isokoski, 2000; Wobbrock et al., 2008; Porta and Turina, 2008; Sarcar et al., 2013). Adapting the Morse code concept, Isokoski (2000) proposed Minimal Device Independent Text Input Method (MDITIM) that uses five off-screen targets to create prefix codes. A user enters a letter by performing off-screen gaze gestures based on the predefined code. Wobbrock et al. (2008) designed the EyeWrite interface that utilizes four corners and the center of a small square window to enter a character. A character is entered by "writing" it as a sequence of fixations on the corners of the window, and the gesture is finished by fixating the center of the window. EyeS used a similar gesture alphabet concept (Porta and Turina, 2008).

Besides drawing a gesture, gaze gestures associating with dynamic visual elements have also been proposed. Huckauf and Urbina (2007) designed pEYEWrite, an expandable pie menu with letter groups. As a user's gaze enters the border of a pie section, the section expands as a secondary pie menu with each letter in a section, and the user can type a letter by simply crossing the border of the corresponding section. Bee and André (2008) adapted QuikWriting (Perlin, 1998) to gaze typing. Although the letter-level gaze gesture typing methods have slower text entry rates compared with dwell-time methods, they can save screen real estate by only using small regions for gaze gestures.

Interfaces using a gaze gesture to type word by word show promising efficacy in eye typing. The idea of word-shaped-based typing was introduced in a broader text entry literature with Shorthand Aided Rapid Keyboarding (SHARK), which maps gestures on a touch-screen virtual keyboard to words in a lexicon (Zhai and Kristensson, 2003; Kristensson and Zhai, 2004; Zhai and Kristensson, 2012). Kristensson and

Vertanen (2012) showed the potential of dwell-free word-shape-based eye typing in a pilot experiment. Hoppe et al. (2013) introduced Eype that maps gaze samples to the closest key and uses resulting letter sequence to predict a word. Pedrosa et al. (2015) proposed Filteryedping, which asks the user to look at the keys that form a word and then look at a button to list word candidates. To type a word the user looks at the desired word candidate and then looks back at the keyboard or the text field. The word-level gaze gesture concept can be applied to a non-QWERTY keyboard for small screens (Zhang et al., 2017).

#### Continuous Writing

With continuous writing interfaces, a user can type a letter, a word, or even a phrase without any pause. Dasher (Ward and MacKay, 2002) is a popular dwell-free method realizing the continuous writing concept. The interface vertically arranges the letters on the side of the screen, devoting larger areas for more likely letters, based on a language model. As the user selects letters, the keyboard dynamically changes, moving selected letters horizontally and collecting them into words. One of the latest studies found significantly faster text entry rates for Dasher when compared to a dwell-keyboard (14.2 wpm versus 7.0 wpm) (Rough et al., 2014).

Another example is Context Switching, a saccade-based activation mechanism for gaze-controlled interfaces (Morimoto and Amir, 2010). The interface has two duplicated keyboard regions, or contexts. The last fixated key is selected when a user switches contexts. Diaz-Tula et al. (2012) proposed Dynamic Context Switching to reduce the screen area by minimizing the non-fixated context.

### 3.2.2 Head-based Text Entry

Most gaze-based text entry methods and interfaces can be directly applied to head-based text entry. However, few studies have explored the performance and user experience of head-based text entry. Hansen et al. (2004) compared mouse-, head- and gaze-based text entry methods. Using a dynamic Danish keyboard, the average text entry rate for mouse, head, and gaze inputs were 7.45 wpm, 6.1 wpm, and 6.26 wpm respectively. The study also showed that participants made fewer errors with the head-based method than the gaze-based method. Gizatdinova et al. (2012) conducted an experiment to examine the performance of gaze-based and head-based pointing in text entry. The participants used gaze or head movements, respectively, to point at a key on a virtual keyboard and confirm the selection with a physical space bar. The participants achieved an average text entry rate of 4.42 wpm with head-based pointing and 10.98 wpm with gaze-based pointing.

Using facial expressions or gestures as an alternative selection method (Lombardi and Betke, 2002; Missimer and Betke, 2010; Krapic et al., 2015) can be also applied to text entry tasks (Grauman et al., 2003; Gizatdinova et al., 2012). Grauman et al. (2003) implemented the BLINKLINK system to detect voluntary blinks, and study participants successfully entered text with BLINKLINK paired with a scanning keyboard. Another study examined mouth-open and brow-up selections with head-based pointing on a virtual QWERTY keyboard (Gizatdinova et al., 2012). The mean text entry speed was 3.07 wpm with mouth-open and 2.85 with brow-up. To the best of our knowledge, head gestures are not used in gaze-based or head-based text entry systems.

# Chapter 4

# Target Reverse Crossing: A Dwell-free Selection Mechanism for Camera-based Mouse-replacement Systems

This chapter presents Target Reverse Crossing, a dwell-free selection mechanism designed for users who control the mouse pointer with camera-based mouse-replacement systems using their head or gaze (Betke et al., 2002). In our work (Feng et al., 2014), we assessed target reverse crossing by comparing it to the dwell-time-based selection method that is widely used by camera-based mouse-replacement systems. Our results showed that target reverse crossing is more efficient than dwell-time clicking, although its one-time success rate is lower. Target directions have an effect on the accuracy of using target reverse crossing. We also show that increasing the target size improves the performance of target reverse crossing significantly, which provides interface design implications for this selection method.

## 4.1 Target Reverse Crossing

Target reverse crossing is inspired by the goal crossing selection mechanism, originally proposed by Accot and Zhai (2002). With goal crossing, the user selects a target by simply moving the mouse pointer to cross it. Wobbrock and Gajos (2007, 2008) showed that goal crossing allowed smoother target acquisition for people with motor impairments using the regular mouse than the common pointing-clicking selection method.

With target reverse crossing, the user selects a target by first moving the mouse cursor into the target region and then moving the cursor out in a *reverse* direction. For example, if the mouse pointer enters a target from the right, it must also leave the button from the right to select it.

The mechanism of target reverse crossing is illustrated in Figure 4·1. A semi-circular boundary appears when the cursor crosses the target edge. Then, the user controls the cursor to cross the boundary to finish the selection. This mechanism can be seen as a variation of goal crossing, where the boundary to be crossed is dynamically generated by a user's entering point of a target.

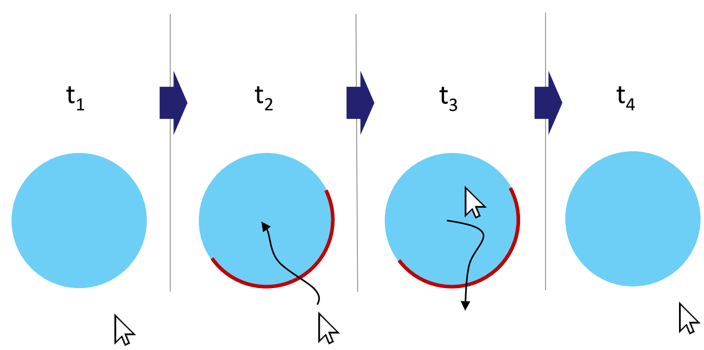



**Figure 4·1:** Target Reverse Crossing selection mechanism. To select a target, the user moves the cursor by head movements to the target ($t_1$). When the cursor crosses the edge of the target, a red semicircle, bisected by the entry point, appears ($t_2$). The user then moves the cursor out of the target through the red edge ($t_3$). The red edge disappears and the target is selected ($t_4$).

## 4.2 Experiment

We conducted an experiment to evaluate the selection efficiency and accuracy of target reverse crossing selection mechanism in comparison with the dwell-time selection method.

### 4.2.1 Participants

Ten adults without disabilities, four females and six males, with an average age of 26 years volunteered to participate in the experiments. All participants were allowed to move their head freely. Half of them had no or little previous experience using head-tracking systems.

### 4.2.2 Apparatus

The experiment was conducted on a PC with an Intel Xeon 2.67 GHz CPU, 12 GB RAM, running Windows 7, and connected to a 30-inch LCD monitor (2,560 $\times$ 1,600 pixels). We used the Camera Mouse 2013 software to collect head-movement data as mouse-replacement input. The real-time video input source for the Camera Mouse was captured by a Logitech QuickCam Pro 9000 webcam (8 megapixels).

### 4.2.3 Procedure

Every participant was asked to perform two experimental sessions in the same day, with a break in-between for 3-5 minutes. In each session, the participants completed 5 target reverse crossing blocks and 5 dwell-time blocks, resulting in a total of 10 blocks for each selection method. To reduce the learning range effect (Poulton, 1975), the order of the target reverse crossing and dwell-time blocks was counterbalanced. Participants could have a break between blocks when needed.

In each block, a blue circle with one of three sizes (radius = 35, 50, 65 pixels) in one of 8 possible positions was displayed on the screen. For each block, targets with all 24 (3 $\times$ 8) possible settings were displayed once in a random order.

The experimental interface is shown in Figure 4·2. The red button in the center of the interface is the start button for a trial. After the cursor dwells on it for one second, it disappears, and a blue disk (the target) appears with one of the 24 possible settings. The distance between the start button and the target is 140 pixels. In the target reverse crossing experimental block, the participant was asked to perform a target reverse crossing selection of the blue circle; while in the dwell-time block, the participant was asked to move the cursor to the target and stay within it for at least for one second, which is the pre-defined dwell-time.

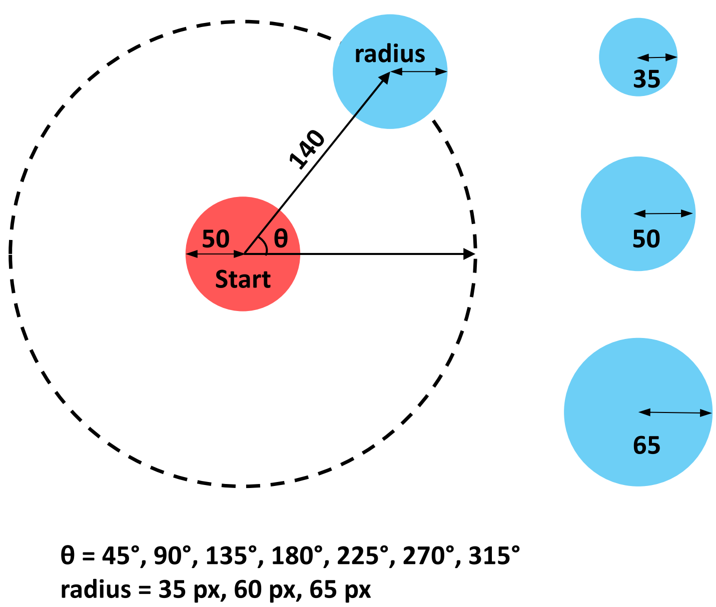



**Figure 4·2:** Experiment design and interface.

Regardless of whether a participant selected the target successfully or missed it, the experimental system stored the movement time and other statistics and began the next trial. The maximum time allowed for one trial was 10 seconds; after 10 seconds, the current trial was marked as a time-out and would be discounted in the analysis of the results.

## 4.3 Results

The participants generated 2,400 trails. We used a repeated measures ANOVA (analysis of variance) to analyze the data.

### 4.3.1 Movement Time

We measured the cursor movement time in both the ballistic and corrective phases (Meyer et al., 1990) of selection. The *ballistic phase movement time* was measured from the point in time when a trial was activated (the start button was selected) to the point in time when the cursor entered the target for the first time. The *corrective phase movement time* was measured as the period between the end of the ballistic phase and the selection of the target. The *total movement time* or *movement time*, the sum of the ballistic phase and corrective phase movement time, was also computed.

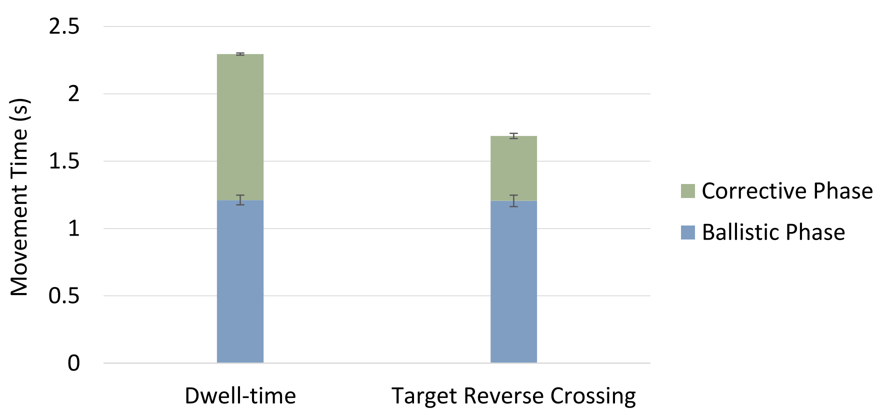



**Figure 4·3:** Mean and standard error of the movement time (seconds) in the ballistic and corrective phases for each selection method.

The mean movement time using the dwell-time selection method was 2.33 s, higher than the 1.69 s when using target reverse crossing mechanism (Figure 4·3). We found a significant effect of selection method in the movement time ($F_{1,18} = 111.06$, $p < 0.001$). The significance level ($F_{1,18} = 872.5$, $p < 0.001$) showed that the selection methods affected the corrective phase movement time. Target reverse crossing saved about 0.6 s in the corrective phase since it did not require the subject to keep the cursor in a target region for a certain time. The dwell-time in the experiment was set as one second, which could be reduced but not be avoided. The difference in the ballistic phase movement time between the two methods was not significant ($F_{1,18} = 0.01$, $p = 0.913$).

### 4.3.2 Accuracy

We use the *one-time success rate* of a selection to describe the accuracy. The definition of "missing a target" for the dwell-time method was "dwelling outside the target area for one second," and for the target reverse crossing method was "leaving the target by crossing over a non-highlighted edge" (Figure 4·4).

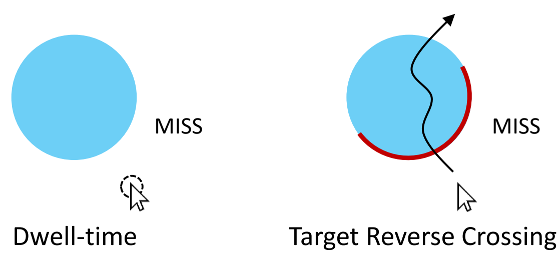



**Figure 4·4:** Dwell-time target miss (left) and target reverse crossing target miss (right).

The method had a significant effect on the one-time success rate ($F_{1,18} = 66.04$, $p < 0.001$). The mean rate of missing a target was 96.7% for the dwell time method and 79.3% for target reverse crossing. This can be explained by the fact that we counted crossing out of the non-highlighted part of the boundary as a missed selection when measuring target reverse crossing method. In an actual application of target reverse crossing, if the cursor slipped out of the non-highlighted boundary by mistake, the user can go back and perform target reverse crossing again.

### 4.3.3 Learning Effects

The learning effects were not significant for either methods. No significant difference between blocks was found for the dwell-time method in both movement time ($F_{4,45} = 0.78$, $p = 0.542$) nor one-time success rate ($F_{4,45} = 1.09$, $p = 0.375$), neither for target reverse crossing mechanism in movement time ($F_{4,45} = 0.71$, $p = 0.591$) nor accuracy ($F_{4,45} = 1.58$, $p = 0.196$).

Since none of the participants had used target reverse crossing, the results showed that this selection method could be learned quickly without training. Some of the participants had experience with the Camera Mouse and dwell-time selection, so that may affect the learning effect measured for the dwell-time method. Feng et al. (2013) showed that the Camera Mouse and the dwell-time selection had insubstantial learning effects.

### 4.3.4 Effects of Target Size

With target reverse crossing, the effect of target size was significant for movement time ($F_{2,27} = 4.04$, $p = 0.029$) and one-time success rate ($F_{2,27} = 28.04$, $p < 0.001$). A pair-wise comparison showed that the increase in size significantly improved the one-time success rate. Figure 4·5 demonstrates that the effect of target size on accuracy is reduced when the target radius is sufficiently large. For dwell-time selection, target size also affected ($F_{2,27} = 7.1$, $p = 0.003$) one-time success rate. No significant effect for target size was found on movement time with the dwell-time method ($F_{2,27} = 2.57$, $p = 0.096$).

### 4.3.5 Effects of Target Direction

The experimental results indicated that target direction could affect accuracy when using target reverse crossing. No such effect was found when using the dwell-time method. The straight up, down, right, and left directions had lower accuracy than

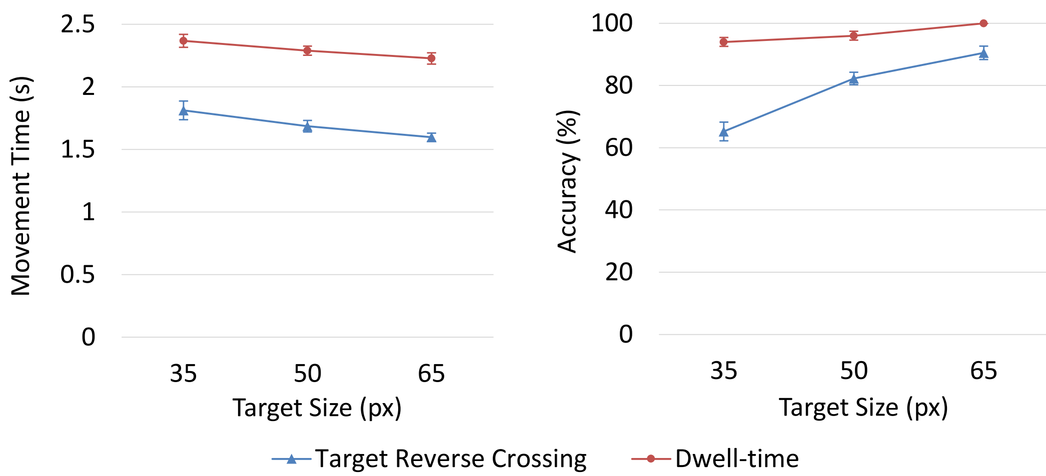



**Figure 4·5:** Effects of target size on each method. The mean and standard error of movement time (left) and one-time success rate (right) for each method are shown.

the four diagonal directions (Figure 4·6). Due to the limited number of trials of each direction, we did not perform a statistical analysis of the effects of target direction. A possible explanation for this phenomenon is that ergonomic factors can affect the performance of the head-movement-based interface. Specifically, controlling their diagonal head movements may have been easier for some participants.

### 4.3.6 Subjective Feedback

After the experiment, we asked the participants to choose the selection method they preferred and explain their choice. Half of the subjects preferred target reverse crossing, while the other half chose the dwell-time selection. Subjects who preferred target reverse crossing considered it was the more efficient selection method. The dwell-time selection was commented as a natural input method.

None of the participants felt fatigue after the experiments. However, two of the participants reported that keeping the cursor in a target region made them feel tired after several blocks of dwell-time selection. Three participants said they were afraid of missing targets when using target reverse crossing, and this made them feel a little

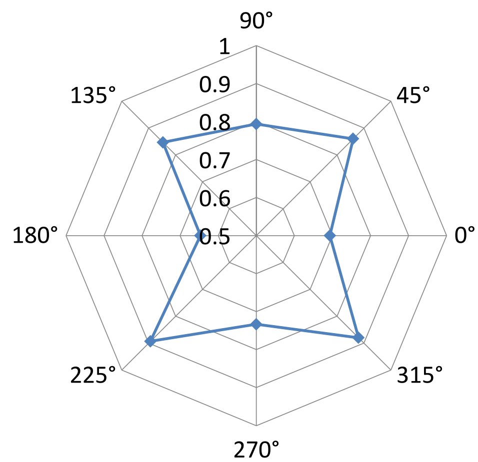



**Figure 4·6:** The effect of target direction on one-time success rate with target reverse crossing.

nervous.

## 4.4 Discussion

Target reverse crossing, the selection mechanism proposed and evaluated in this chapter, had a satisfactory performance when used with head-tracking systems. Both statistical analysis and subjective feedback showed that target reverse crossing was more efficient than a dwell-time selection method. However, we found that accuracy (one-time success rate) of target reverse crossing was lower than the dwell-time selection. A possible solution to this problem is to enlarge the target size or magnify the target using an enhanced cursor.

Dwell-time selection is commonly used not only by camera-based mouse-replacement systems but also by other assistive systems, for example, eye-tracking systems. Some participants felt dwell-time is a natural and simple selection mechanism. However, when considering the speed of the selection process, it must be noted that the dwelling time itself increases the movement time during the corrective phase of selecting a target. In addition, maintaining a cursor that is not as stable as a regular mouse pointer in the target region for the dwell period can cause fatigue.

We found that target size could significantly affect accuracy and movement time of reverse crossing, and influence the performance of the dwell-time method to some extent. Target direction was also a factor that affected the one-time success rate of target reverse crossing. Thus, when designing a graphic interface for a camera-based mouse-replacement system, target size and direction should be taken into consideration.

# Chapter 5

# EyeSwipe: Text Entry Using Gaze Paths

This chapter introduces EyeSwipe, a text entry method using gaze paths, analogous to how the user's finger trace is mapped to words on a swipe-based typing interface (Zhai and Kristensson, 2003). To segment the gaze path from continuous gaze inputs, the user indicates the first and last characters of a word using a deterministic selection method, "reverse crossing," adapted from target reverse crossing. The user only needs to glance at the vicinity of the respective keys to type the characters in the middle of the word. EyeSwipe provides candidate words dynamically with pop-up buttons above the keys (Figure 5·1) while the user's gaze is swiping through the keyboard. We compared the performance of EyeSwipe with that of a dwell-time-based keyboard. EyeSwipe shows significantly higher text entry rates and is perceived to be more comfortable and efficient by the users. This chapter has adapted and updated content from our published work (Kurauchi et al., 2016).

## 5.1 Interface Description

This section introduces how EyeSwipe uses reverse crossing to segment a gaze path from continuous gaze inputs and describes its text entry interface.

### 5.1.1 Reverse Crossing

With EyeSwipe, a word is typed based on the users gaze path, similar to swipe-based interfaces that trace a user's finger on a touch screen to determine the desired word from a lexicon. However, a users finger trace and gaze path differ in that a finger trace has clear start and end points, while a gaze path does not. With ambiguous start and end points, selecting a word from a lexicon is error prone.

To resolve this issue, we adapted target reverse crossing (Chapter 4) to deal with noisy gaze inputs and make explicit selection. We refer to the revised selection method simply as "reverse crossing". Instead of using the edge between the outer and inner regions of the button for the selection that target reverse crossing originally employed, two separate buttons are used. Once a user looks at a target (defined by a 100 ms fixation), another button will pop up above the target. The user looks at the pop-up button and then back at the target to perform the selection (Figure 5·1). All selections in EyeSwipe are activated by reverse crossing.

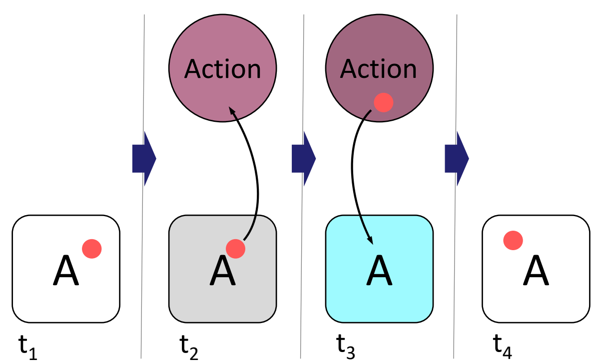



**Figure 5·1:** The reverse crossing mechanism. To select the "A" key by reverse crossing, the user moves their gaze (red circle) to the key ($t_1$) and the action button pops up ($t_2$). The user then looks at the action button ($t_3$) and then back at the key ($t_4$). The action button disappears and the action indicated is performed.

### 5.1.2 EyeSwipe

The EyeSwipe text entry interface is composed of a text box, a word-candidate list, a cancel/delete button, and a virtual keyboard. The selectable targets are the keys representing the letters, the buttons that show the candidate words computed from the gaze path, a punctuation key, and the cancel/backspace button (Figure 5·2).

To type a word with EyeSwipe, the user initially selects the first letter by looking at the virtual key and performs a reverse crossing on the “start” action button that appears. The gaze path then starts, and the interface will show the latest gaze samples with a blue line and indicate the current gaze point as a small blue circle until the end of the current gaze path. Next, the user glances through the vicinity of the letters in the middle of the word, stopping at the last letter. An action button will pop up, showing the most probable candidate in the candidate list computed from the gaze path starting at the first letter. This visual feedback allows the user to know what word will be typed without having to look at the text box, enabling him or her to continue typing the next word without shifting his or her attention to other places. As the user selects the action button to indicate that the gaze path is finished, the most probable word is entered into the text box. A space is automatically added after each selection.

After a word is selected, the top five candidates are then displayed above the keyboard, enabling the user to replace the most probable word automatically typed by EyeSwipe, when necessary, by reverse crossing. A backspace key is provided to delete the last typed word. The backspace key also serves as a cancel button when a user is performing a gaze path. If the user wants to cancel the current gaze path when the wrong first letter was selected, he or she can look at the backspace button and performing a reverse crossing.

Selections of the punctuation key can also be performed by reverse crossing, with

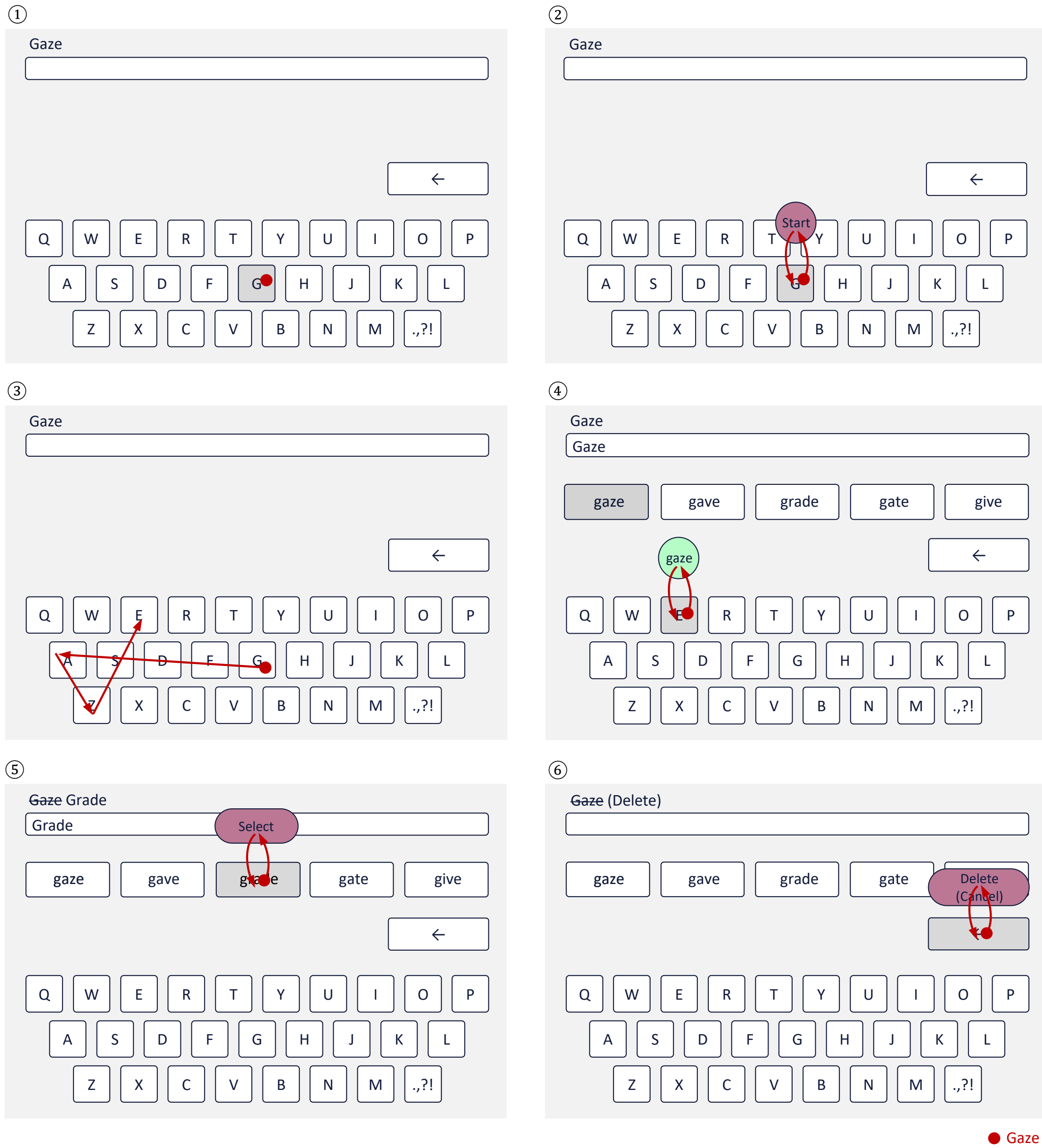



**Figure 5·2:** The EyeSwipe interface. A word can be typed by: (1) Fixate on the first letter of the desired word. (2) Select the key using reverse crossing to initiate a gaze path. (3) Glance through the intermediate letters. The corresponding gaze path is used to compute the candidate words. (4) When fixating on the last letter, the most probable candidate word will be displayed on the pop-up action button. As the user selects it, the word is typed. (5) The top 5 candidates are displayed when a word is typed. The user can change the typed word by selecting another candidate button. (6) Select the cancel/delete key to cancel a gaze path or delete a typed word.

pop-up buttons in different directions. An advantage of reverse crossing is that it allows for the use of multiple pop-up buttons for a single selection target (Figure 5·3). This feature allows users to type multiple punctuation characters with a single punctuation key. For EyeSwipe interface, as the user looks at the punctuation key, four pop-up buttons are displayed: above, below, to the left and to the right. Each pop-up button represents one punctuation mark: comma, exclamation mark, question mark, and period. As each pop-up button represents a different action that can be performed for that given selection target (e.g., type one of the punctuation marks), we refer to them as action buttons. Most keys have only one action button, which always pops up above it.

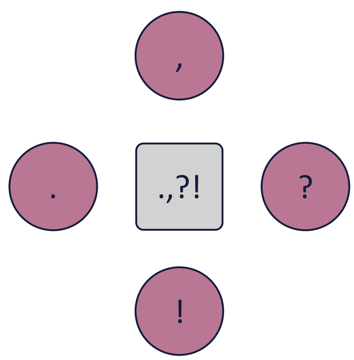



**Figure 5·3:** The punctuation key has multiple action buttons for different punctuation marks.

## 5.2 Candidate Selection

EyeSwipe uses a gaze path and a simple language model to compute probabilities of words in a lexicon stored in a trie data structure. As the user indicates the first and last letters of the desired word, a gaze path is segmented. The gaze path is often noisy and may contain points sampled while the gaze was shifting from one key to another. For this reason, we apply an average filter and then remove samples that are isolated from both their previous and next samples. The distance threshold used was set to half a key's height (approximately 45 px).

The lexicon is filtered by the first and last letter selected by a user. The resulting list of candidate words is sorted according to a score based on their likelihood of occurrence and their similarity to the user's gaze path. The similarity to the user's gaze path is computed according its distance to an "ideal path." The ideal path of a word comprises the sequence of coordinates between the centers of the keys that form the word. For example, the ideal path for the word "gaze" comprises the centers of "g", "a", "z", and "e" keys. Repeated letters, such as "l" in "hello", are included just once.

EyeSwipe uses Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978), a common technique to compare two time sequences that might be stretched or compressed, to compute a DTW score. DTW has also been used in swipe-based typing on touch screens (Zhai and Kristensson, 2012). The complexity of DTW is $O(\max(M, N))$ where $M$ and $N$ are the lengths of the two time sequences (Ratanamahatana and Keogh, 2004). Since repeated letters are counted only once, the upper limit for the length of the ideal path is less or equal to the length of the word. For an English lexicon, this means that the average length of the ideal path is close to 5. On the other hand, the length of the gaze path may consist of a few hundred coordinates when using the raw gaze data, depending on the sampling rate of the eye tracker.

For a given sampled gaze path $g$, the candidates are initially sorted according to their DTW match:

$$\mathrm{DTW}_{match}(g, p) = \frac{1}{1 + \mathrm{DTW}(g, p)} \tag{5.1}$$

where $p$ is the ideal path of a word, and $\mathrm{DTW}(g, p)$ is the output distance from the DTW algorithm.

EyeSwipe then computes the "score" for each of the top $n$ word candidates as a convex combination of the DTW match and the occurrence probability in a unigram model:

$$\text{Score}(g,w) = \alpha \frac{\text{DTW}_{match}(g, \text{Ideal}(w))}{\sum\limits_{v\in\mathcal{L}'} \text{DTW}_{match}(g, \text{Ideal}(v))} + (1-\alpha)\frac{|\text{Occurrences}(w)|}{\sum\limits_{v\in\mathcal{L}'} |\text{Occurrences}(v)|} \tag{5.2}$$

where $\alpha$ is a weighting factor, $\mathcal{L}'$ is the set of the top $n$ words selected from the lexicon $\mathcal{L}$, Ideal($w$) is the ideal path of the word $w$, and |Occurrences($w$)| is the number of occurrences of the word $w$ in a text corpus (Winwaed Software Technology LLC, 2012). Note that after the first sifting step (Equation 5.1), only the top $n$ candidates remain to be considered in the second step (Equation 5.2). This is because some words, such as "get", occur a few orders of magnitude more often than the others and would always have an extremely high score if the first step were omitted. These high-frequency words are possible candidates only if their DTW match is among the top $n$ words. We empirically set the values of $n$ as 10 and $\alpha$ as 0.95.

## 5.3 Experiment

We conducted a user experiment to further examine the performance and user experience of text entry using gaze paths by comparing EyeSwipe to a dwell-time-based virtual keyboard.

### 5.3.1 Participant

Ten university students without disabilities (5 males and 5 females, aged 18 to 21) participated in the experiment. All participants had normal or corrected-to-normal vision, and half of the participants wore glasses. They were all proficient in English (8 native speakers) and familiar with the QWERTY keyboard layout. They had no or little experience with eye-tracking systems.

The participants received compensation of $25 for participating in the study. Additionally, the participant with the best performance (measured by both speed and accuracy) would receive an additional $20 as an incentive.

### 5.3.2 Apparatus

We used a 19-inch LCD monitor (1024 $\times$ 768 pixels resolution) connected to a laptop (2.30 GHz CPU, 4GB RAM) running Windows 7. A Tobii EyeX eye tracker with a sampling rate of approximately 70 Hz was used for collecting gaze input data.

We built the EyeSwipe interface and the dwell-time-based keyboard in C++ using the Qt framework. The EyeSwipe typing interface is shown in Figure 5·2. We used Kaufman's lexicon (Kaufman, 2015), augmented with contractions and the words in MacKenzie and Soukoreff's phrase set (MacKenzie and Soukoreff, 2003), resulting in 10,219 words. We used the same interface for dwell time, with a shrinking gray box on the fixated key as visual feedback for the elapsed dwell period. The dwell period is set to 600 ms, following Hansen et al. (2003).

### 5.3.3 Procedure

We used a within-subject design with session and interface as the independent variables, and text entry rate and error rate as the dependent variables. Each participant performed eight 10-minute eye-typing sessions, four using EyeSwipe and four using the dwell-time keyboard, totaling 40 minutes with each interface, aside from a short practice section. The interfaces were counterbalanced using a Latin square. Each participant visited the lab on two different days (48 – 72 hours apart) and completed two sessions of each typing method (balanced order) per day.

The participants were seated comfortably in front of the screen with a distance of about 70 cm. They were asked to move their head as little as possible during the experimental sessions. The eye tracker was calibrated for each participant at the start of each day and re-calibrated as needed.

The experimenter began by introducing how eye-tracking systems work and the two typing methods. Before starting the formal sessions, participants practiced typing two sentences using each typing method. During the sessions, phrases were randomly sampled from the MacKenzie and Soukoreff's phrase set and presented at the top of the experimental window. The participants were encouraged to memorize the phrase and type as fast and accurately as possible. If the session timed out in the middle of a sentence, the participant had to finish typing it to end the session. Between sessions, participants were allowed to take a break for 3 – 5 minutes. At the end of the last session, the participants completed a questionnaire on their subjective feedback regarding the two typing methods, along with their demographic information.

## 5.4 Results

In this section, we analyze the experiment results to examine the efficacy of EyeSwipe comparing with dwell-time typing method.

### 5.4.1 Text Entry Rate

The text entry rate was measured in words per minute (WPM), where a word is any sequence of 5 characters, including spaces. Participants were able to type, on average, faster using EyeSwipe than using the dwell-time keyboard in all four sessions (Figure 5·4). In the last session, after 30 minutes of typing with each method, participants achieved a mean text entry rate of 11.7 wpm for EyeSwipe and 9.5 wpm for dwell-time typing ($F_{1,9} = 8.54$, $p = 0.017$).

There was a significant learning effect on typing rate for both methods (EyeSwipe: $F_{3,27} = 8.61$, $p < 0.001$, dwell-time: $F_{3,27} = 3$, $p = 0.048$). From the first session to the fourth, the mean typing rate increased from 8.6 wpm to 9.5 wpm using the dwell-time method, and from 9.4 wpm to 11.7 wpm using EyeSwipe.

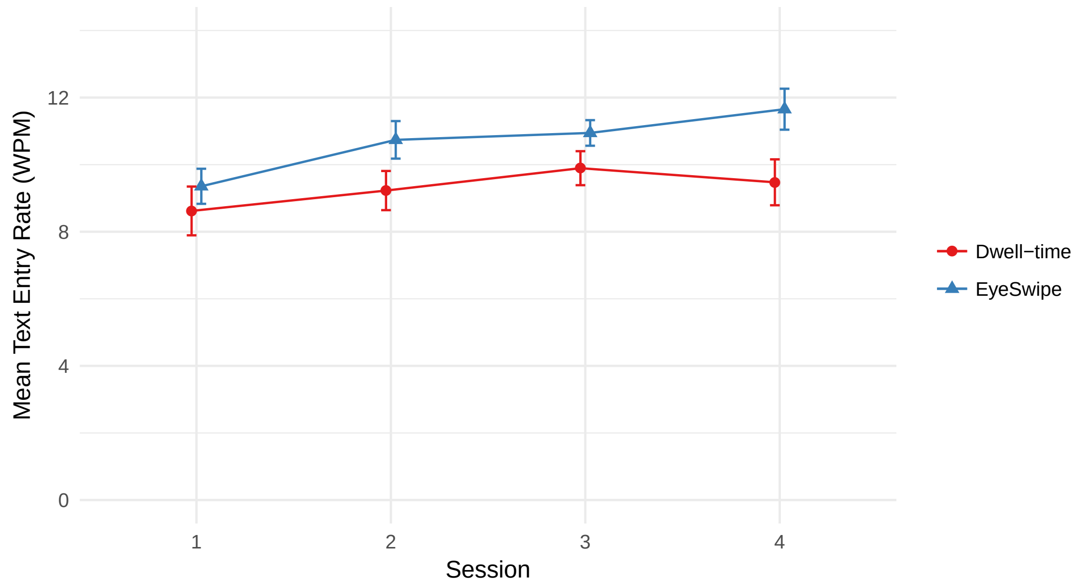



**Figure 5·4:** Mean and standard error of the text entry rate in words per minute (WPM) for each session with EyeSwipe and the dwell-time keyboard.

The mean maximum text entry rates (computed for each sentence) in each session are shown in Figure 5·5. Significant main effects were found for both session and typing interface ($F_{3,27} = 3.49$, $p = 0.029$ and $F_{1,9} = 26.81$, $p < 0.001$, respectively). The maximum typing rates for the dwell-keyboard are all below 15 wpm, while most of the maximum typing rates for EyeSwipe are above 15 wpm, with some above 20 wpm.

The overall increase in the maximum typing rate for EyeSwipe over the four sessions indicates that even higher rates may be achieved with practice. Based on Kristensson and Vertanen's model (Kristensson and Vertanen, 2012), for a user to achieve a typing rate of 15 wpm with dwell-time equal to 600 ms, the overhead must be 200 ms. With an overhead of 100 ms, the user would be typing at about 17 wpm, and in the ideal case of zero overhead, the user would type at most at 20 wpm. We thus conclude that some participants reached close to the theoretical maximum typing rate using the dwell-keyboard.

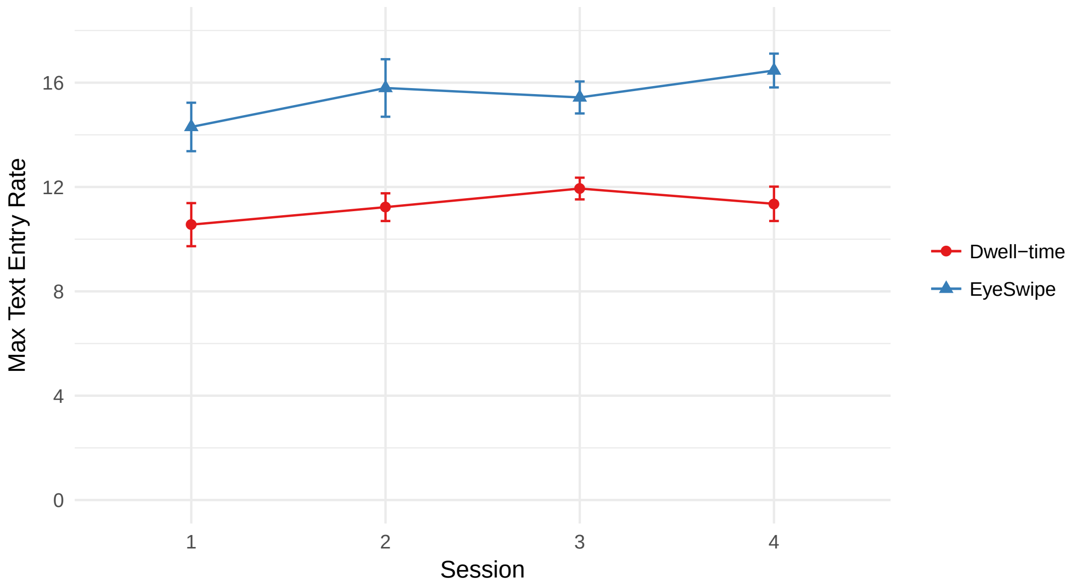



**Figure 5·5:** Mean and standard error of the maximum text entry rate in words per minute (WPM) for each session and interface.

Participants with glasses (5 out of 10) experienced calibration problems, which affected their usage of dwell-time typing, as expected according to Räihä (2015). The experimental results suggest that EyeSwipe may be more robust to noisy gaze data associated with people wearing glasses. In session 4, the average text entry rate for participants wearing glasses was 12.2 wpm using EyeSwipe and 8.7 wpm using dwell time. One participant with glasses achieved a 15.2 wpm average typing rate using EyeSwipe but only 7.3 wpm using dwell time due, in part, to gaze-input noise.

### 5.4.2 Gesture Entry Rate

The gesture entry rate measures characters entered by one gaze path, in characters per minute (CPM). The necessity of dwelling on individual keys for a set amount of time compromises the speed of dwell-time-based typing interfaces. EyeSwipe counteracts this disadvantage by requiring that only the first and last letters of a word to be selected explicitly. Also, the first and last selections are dwell-free, because of the reverse crossing technique, so the user does not have to wait for the interface. A consequence of this approach is that longer words can be typed faster with EyeSwipe than by using dwell-time typing (Figure 5·6).

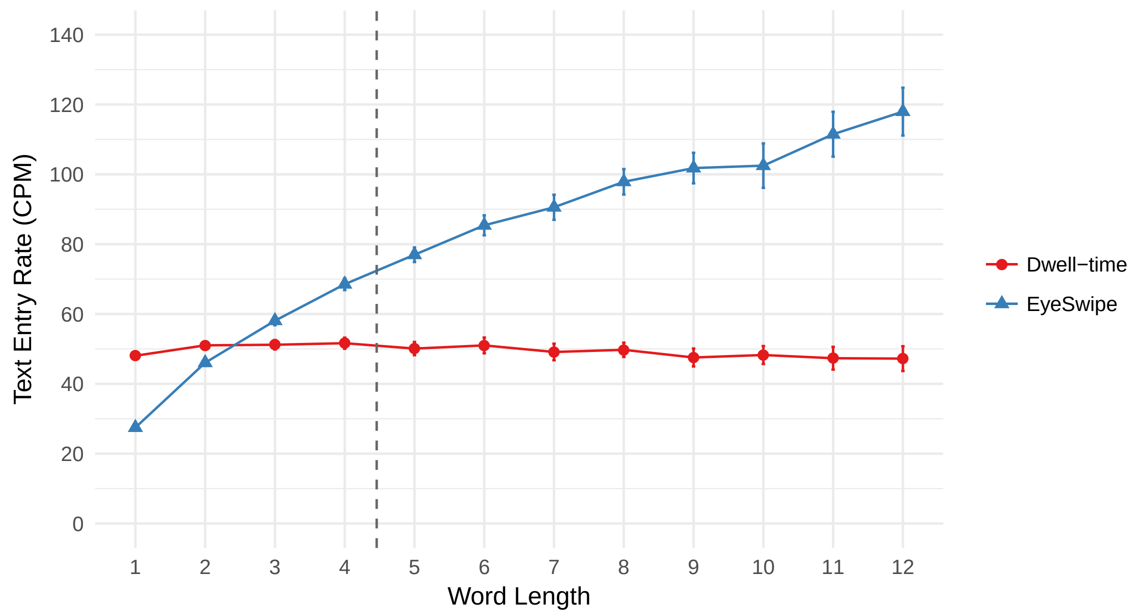



**Figure 5·6:** Mean and standard error of the text entry rate in characters per minute (CPM) per typed word length. The dashed line represents the average word length of expected words.

Specifically, for words of three or more characters, typing with EyeSwipe was found to be faster than dwell-time typing, with the difference in typing rate proportional to the length of the word. Words of a single character were expected to be slower with EyeSwipe because it requires two reverse-crossing gestures: one to indicate the first letter and the other to indicate the last letter, which, in this case, are the same. Words of two characters were expected to take a similar time for both interfaces because EyeSwipe requires two reverse-crossing gestures while the dwell-keyboard requires two selections by dwell. From this result, we can also infer that each reverse-crossing took approximately 600 ms because the typing rates are similar to the dwell-keyboard.

### 5.4.3 Accuracy

Error rate was measured using the Minimum String Distance (MSD) metric (Soukoreff and MacKenzie, 2003) for both methods (Figure 5·7). The low error rate (of less than 2%) for sessions 2, 3, and 4 implies that participants in our experiment were careful in typing the given phrases accurately using both methods.

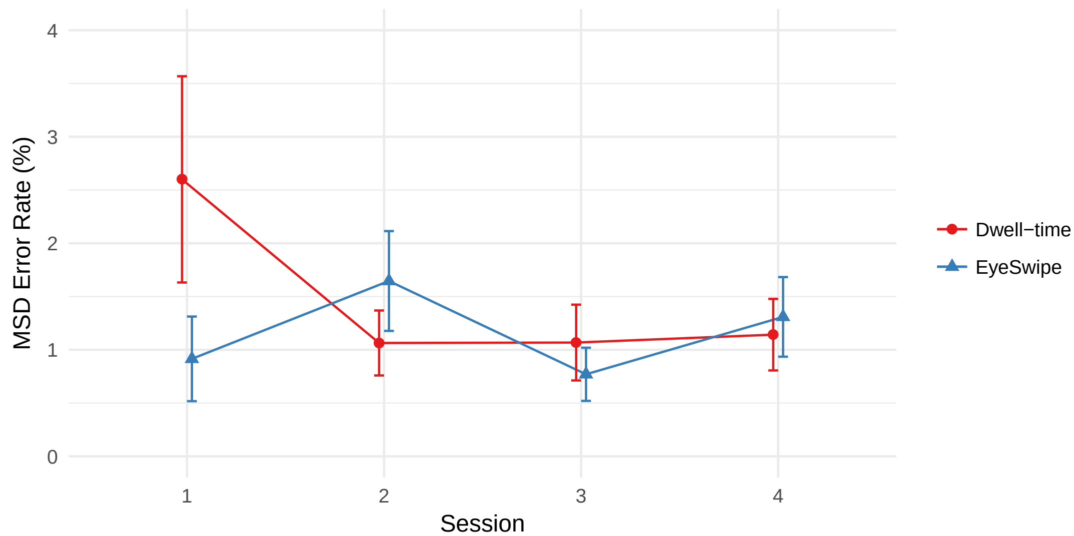



**Figure 5·7:** Mean and standard error of the minimum string distance rate (MSD rate) for each session and interface.

The correction rate is the ratio between deleted words and total typed words. It is a measure for EyeSwipe of the number of times words were deleted because the wrong last letter was unintentionally selected, or because no candidate matched the word being typed. On average, 10.68% of the entered words were deleted by participants. The correction rate improved from 13.25% in session 1 to 8.13% in session 4 ($F_{1,9} = 11.86$, $p = 0.007$), suggesting that increasing familiarity with EyeSwipe reduces instances of deletion.

The accuracy of selecting the first and last characters of a word by reverse crossing was 98.3%, which was calculated as the ratio between the number of correctly selected first and last letters and the total number of selections.

By pruning candidates that do not meet the criteria of the first and last letters before computing the scores, we improve both the computational cost and the accuracy of word prediction. The computational cost is improved because significantly less score computations are performed. The top-$k$ accuracy is the rate in which the correct word was among the first $k$ candidates. We computed the top-$k$ accuracy for the gesture classification with the whole lexicon and with the pruned lexicon (that included only the words starting and ending with the selected letters). The results are shown in Table 5.1. For all tested values of $k$ (1 through 5), we found that the top-$k$ accuracy was always at least 16 percentage points higher for the filtered lexicon than for the full lexicon.

**Table 5.1:** Top-$k$ word prediction accuracy (correct word was among 1st $k$ candidates) using (1) the lexicon filtered by the first and last letters, and (2) the whole lexicon

| | **top-1** | **top-2** | **top-3** | **top-4** | **top-5** |
|---|---|---|---|---|---|
| **Pruned Lexicon** | 82.7% | 92.1% | 95.5% | 97.1% | 98.3% |
| **Entire Lexicon** | 62.8% | 73.9% | 78.6% | 80.5% | 81.7% |

### 5.4.4 Subjective Feedback

Participants indicated their opinion on the performance of the two interfaces and their preference on a scale from 1 to 7 (low – high). On average, EyeSwipe was preferred to the dwell-time keyboard (5.8 vs. 4.1), and the performance of EyeSwipe was also rated better than the dwell-time keyboard (5.5 vs. 4.7).

The results for the subjective evaluation of the performance are shown in Figure 5·8. Compared to dwell-time typing, EyeSwipe was considered faster (6.1 vs. 3.2) and more comfortable (4.6 vs. 3.4), but also less accurate (5.1 vs. 6.0) and harder to learn (4.5 vs. 6.5). For their respective levels of eye and neck fatigue, EyeSwipe was rated to cause more neck fatigue (2.6 vs. 2.0, lower is better) and less eye fatigue (3.9 vs. 4.3) than dwell-time typing.

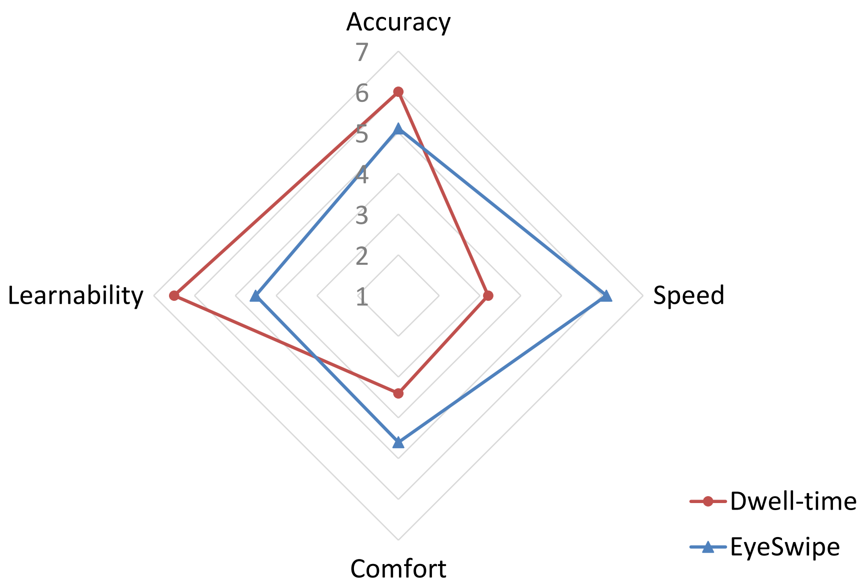



**Figure 5·8:** The average perceived performance on a 7-point Likert scale.

Participants were also asked to comment on each interface. Their responses are summarized in the following subsections.

### Feedback on the Dwell-time Keyboard

Participants in general considered the dwell-time keyboard "intuitive and simple" (P5), and "easy to learn" (P6). P3 reported feeling "more in control" of the interface. A possible explanation is that participants perceived high accuracy when using the dwell-time keyboard: "I was assured of its accuracy by the clicking and I did not feel the need to check what I had typed to see if it was correct" (P8).

P9 indicated that "[Dwell-time] method is very accurate but it takes a lot of time to write each individual words." As for speed, P6 considered "it is very slow. Often I would move my eyes to the next letter slightly too fast and then have to go back." Using the space bar also lowered the speed: "Took a lot longer, but [had] better accuracy. Using space bar is time consuming" (P7). The low speed also increased the cognitive load using the dwell-time keyboard: "It was painfully slow and I felt easy to get distracted or zone out while I was spelling each word. My brain was thinking faster than it was clicking, which sometimes led to mistakes" (P9).

P1 added that the dwell-keyboard is "a lot easier for shorter words" and "never very frustrating." However, regarding longer words, some participants added: "It was much more tedious to type longer words using this method" (P2) and "it is more time consuming especially if you want to type a longer word" (P3).

### Feedback on EyeSwipe

Overall, EyeSwipe was perceived as faster: "it is much faster than the [dwell-keyboard] because you are not selecting each letter individually" (P3). P8 considered EyeSwipe "faster and more natural."

Most participants indicated the need for practice to use EyeSwipe: "[EyeSwipe] takes a lot of practice to get used to. I think it would be overall faster to someone who got used to it" (P7). "At first this method was difficult to get used to, but once I did, I was pretty accurate and fast at typing" (P9). "The gaze-gesture typing is extremely fast and accurate but it has a high learning curve. I can see this method being the preferred method, especially after long periods of use" (P1).

P7 added that it was not only faster but also "even if [she] missed a few of the letters, the suggestions would come up."

As indicated in the comments about the dwell-keyboard, some participants noticed the advantages of typing long words using EyeSwipe: "The improved speed for longer words is very noticeable though, and it probably made up for all the time lost when mistyping small words" (P1). "Short words were more of a hassle to type, but were still manageable. It was much easier to type long words using this method" (P2).

## 5.5 Discussion

Participants achieved an average typing rate of 11.7 wpm after 30 minutes of typing experience using EyeSwipe, against only 9.5 wpm with the dwell-time keyboard. The results showed that the EyeSwipe interface was significantly faster than the dwell-time keyboard. Significant learning effects were observed for EyeSwipe typing, indicating the possibility of a higher text entry rate with longer practice. Users reported that EyeSwipe delivers speedier and more comfortable interaction than the dwell-time interface.

The results of our experiment also suggest that the EyeSwipe interface is useful when dealing with gaze-input noise. Reflections from glasses can make it difficult for an eye tracker to estimate gaze direction accurately. Gaze-input noise due to glasses was handled more robustly by EyeSwipe than by the dwell-time method in the experiment: participants with glasses experienced, on average, a larger difference in typing rates between EyeSwipe and dwell-time typing compared with participants without glasses.

The pop-up buttons make the on-screen keyboard in EyeSwipe easy to extend. An example is the punctuation key (Figure 5·3), where buttons are displayed in multiple directions. Function keys, numbers, and punctuation marks can also be added to the virtual keyboard with multi-direction pop-up buttons.

EyeSwipe shows only the first candidate during the reverse crossing that ends the gesture with the current pop-up button design. To select a different candidate, the first candidate must always be entered first. In such cases, the user must select the wrong word to finish the gesture, search for the desired word in the candidate list and, if present, select it. The concept of selecting the wrong word and then replacing it was not clear to some participants in the beginning, and sometimes it was necessary to explain it more than once during the training session. Another issue of the action button is that it partially occludes one or two keys on the keyboard.

Explicitly requiring that the first and last letters of the word be selected with reverse crossing is advantageous for several reasons. Pruning the lexicon with given first and last letter can reduce the computational cost and increase the accuracy of gaze path classification. It enables users to type with few input errors: typing with EyeSwipe yielded a correction rate of only 8.13% in the last experiment session. The information about the first and last characters can be beneficial to other dwell-free interfaces such as Filteryedping (Pedrosa et al., 2015) by reducing the number of possible candidates.

# Chapter 6

# HGaze Typing: Head-Gesture Assisted Gaze Typing

This chapter proposes a novel bi-modal text-entry interface, HGaze Typing, which combines the simplicity and accuracy of head gestures with the speed of gaze input to provide efficient and comfortable interaction. By adding a head-based input channel – with head pitch, yaw, and roll data – to gaze-based text entry, HGaze Typing allows explicit activation of common text entry commands, such as selection, deletion, and revision (by nodding, shaking, and tilting gestures, respectively). Similar to EyeSwipe, HGaze Typing uses gaze path information to compute candidate words. Nodding gestures clearly separate the gaze path from the continuous gaze input. HGaze Typing reduces the size of the screen regions for cancel and deletion buttons, and the word candidate list, which are required by most eye-typing interfaces. A lab-based user study shows that HGaze Typing outperforms a dwell-time-based virtual keyboard in efficacy and user satisfaction. The experimental results also demonstrate that our method of integrating gaze and head-movement inputs is effective and robust.

## 6.1 Interface Description

With HGaze Typing, a user can perform text entry tasks for gaze-path-based typing including starting, ending, and canceling a gaze path; selecting candidate words; and deleting words using simple head gestures (nodding, shaking, and tilting). The gaze path, segmented by two nods, is created by the user's fast saccades across the keyboard and collected by the HGaze Typing system for producing a word candidate list.

### 6.1.1 Design Principles

Head-movement-based systems can provide a more accurate and stable input than gaze-based systems (Hansen et al., 2004). When performing commands with head-tracking systems, users can simultaneously receive visual information or feedback with their eyes, and perform actions with their head movements. However, people with motor impairments feel fatigue when using head-tracking systems for long periods (Section 2.4.2). Gaze inputs, on the other hand, require minimal muscle movements. The gaze-based systems thus outperform head-based systems in speed and comfort.

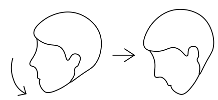
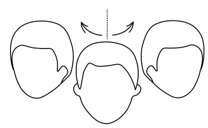
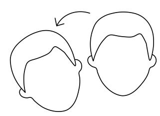
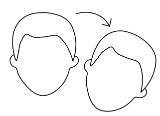
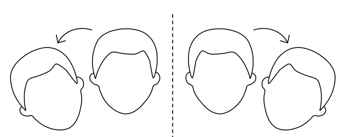
We designed the HGaze Typing interface to take advantage of the benefits of head gestures and eye movements as input mechanisms and limit the potential frustration that users may feel. Based on the features of head and gaze inputs, the main design principles of HGaze Typing are: (1) use head gestures to perform tasks requiring high accuracy, (2) use head gestures for commands associated with visual activities, and (3) use gaze inputs for tasks that can be accomplished with fast movements and a coarse level of accuracy.

### 6.1.2 Head Gestures and Text Entry Tasks

A text entry interface that uses a word path method must provide solutions for five text entry tasks: (a) initiating or finishing a word path by selecting the first letter or the last letter respectively, (b) connecting (by gaze inputs or head movements) the middle letters of the desired word to generate a path, (c) canceling a path when it is started involuntarily, (d) choosing a word from a candidate list, and (e) deleting an unwanted word.

According to the design principles, we map nodding gestures to selection tasks.

**Table 6.1:** Head gestures and the corresponding text entry tasks.

| Gesture | Illustration | Text Entry Task |
|---|---|---|
| Nod |  | Start or end a gaze path |
| Shake<br>(Rotate head to<br>the left or right) |  | Cancel a gaze path<br>Delete a typed word |
| Tilt left |  | Change the selected candidate<br>to its left candidate |
| Tilt right |  | Change the selected candidate<br>to its right candidate |
| Tilt and hold |  | Change the selected candidate to<br>a candidate $n$ words away from it<br>($n$ depends on the tilting direction<br>and the holding time of the head<br>position) |

Gaze path recognition depends on the correct selections of the first and last letter. Additionally, selecting the first and last letter of a word in word-path-based text entry happens more frequently than the other typing tasks, such as deleting a word. A nodding gesture is faster than other common head gestures and is considered as the best head gesture for making a selection (Špakov and Majaranta, 2012).

Deletion tasks are usually associated with visual activities (for example, check if the entered word is correct) and also require accuracy. HGaze Typing uses a head shake (rotate head to the left or right) for a word deletion. Word path canceling is a similar concept as deletion, and is also assigned to the shaking head gesture.

When there is a list of candidates, a user will need to scan the word list before choosing the desired word and check if an entered word needs to be replaced. A head gesture, especially tilting, will be more suitable than a gaze input for this task. When tilting the head, one can keep his or her gaze on the screen without missing any visual information. Besides, the direction of a tilt provides easy navigation on the word list: a left tilt initiates navigation to the left, and a right tilt to the right.

The definition and illustrations of these head gestures and their corresponding text entry tasks are shown in Table 6.1.

Using gaze saccades to depict the word path on a virtual keyboard produced accurate word prediction results (Chapter 5): more than 98% with a five-word candidate list. Given gaze movements are significantly faster than head movements, HGaze Typing collects gaze inputs to generate a word path, which is simply called a *gaze path*.

### 6.1.3 HGaze Typing

The HGaze Typing interface is composed of a text box, a cancel/delete indicator, a virtual keyboard, and a confirm key (Figure 6·1).

To enter a word with HGaze Typing, the user selects the first letter of the word by fixating on the virtual key and performing a nod. A red dot appears at the bottom of the key as visual feedback of fixation. After the user confirms the first letter, the borders of the virtual keys disappear as an indication of the start of a gaze path. Next, the user glances through the vicinity of the intermediate letters of the word until reaching the last letter. A red dot appears on the key showing a fixation is detected by the system, and the user nods to confirm the last letter. The gaze inputs segmented by the two nods, or the gaze path, is used by the HGaze Typing system for word prediction.

A word candidate list containing five possible words pops up above the last letter key. The most probable candidate is placed at the center of the list, with the second

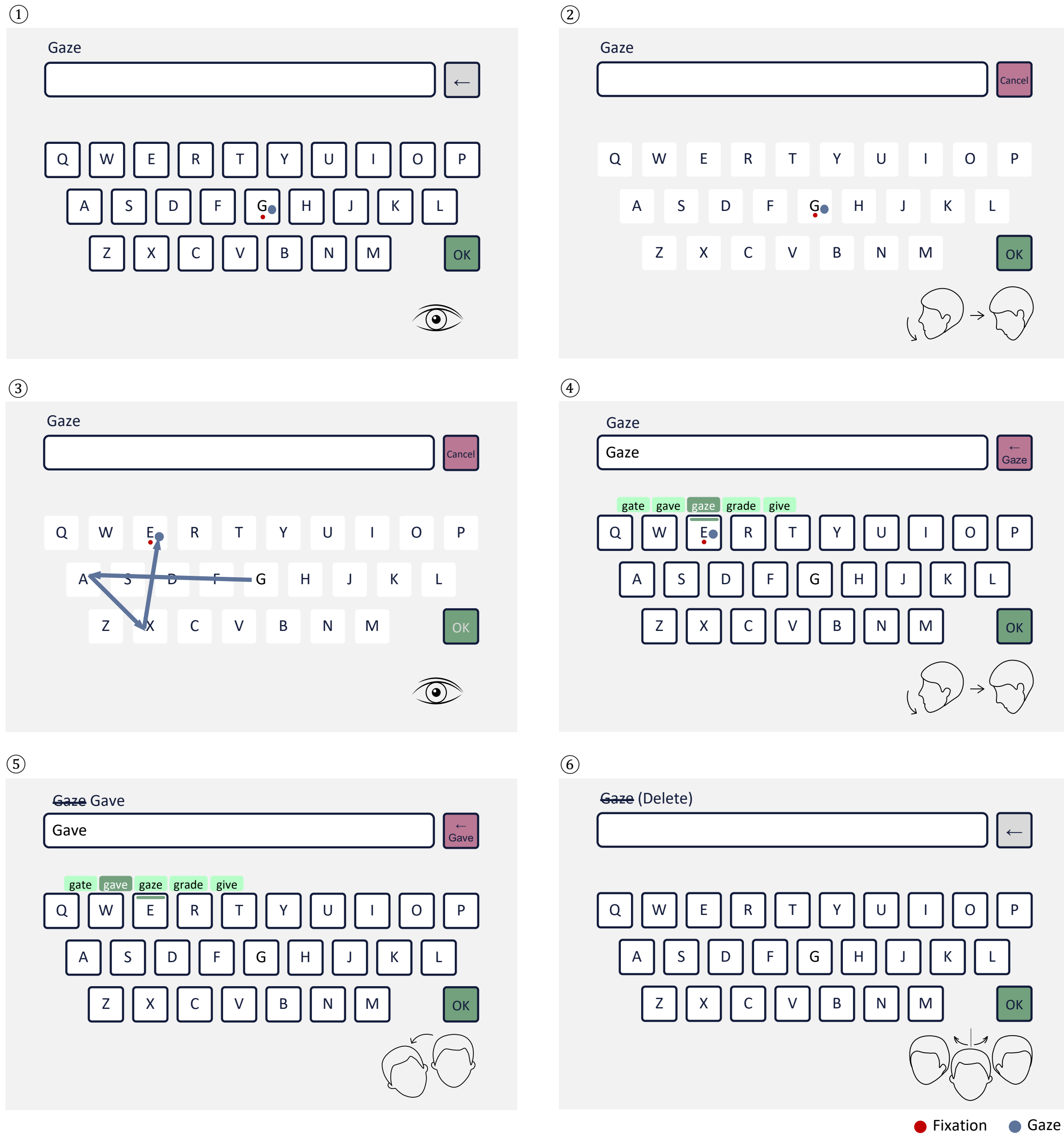



**Figure 6·1:** The HGaze Typing interface. The text to be entered is shown above the text box ("gaze"). The word can be typed by completing the following actions: (1) Fixate on the first letter of the desired word. The red dot on the key indicates a fixation. (2) Select the key by a nod. (3) Glance through the intermediate letters. The corresponding gaze path is used to compute the candidate words. (4) Fixate on the last letter and selected by a nod. The most probable word is typed, and a candidate list with five words shows above the key. (5) If desired, change the typed word with a head tilt (left or right). (6) Use a shaking gesture to delete a typed word (A shake can also be used to cancel a gaze path).

most probable to its left and the third to its right, and so forth. The most probable word is highlighted in the list and is automatically entered in the text box. The user can tilt his or her head to the left or right to replace the typed word by another candidate word. The selected candidate is highlighted in the list, and the word in the text box is replaced.

If the desired word is not in the candidate list, the user can delete the typed word by shaking his or her head. If the user accidentally triggers an unwanted gaze path, he or she can also cancel the current path with this gesture. A cancel/delete indicator is placed to the right of the text box, which serves as a reference for the user to check the cancel/delete status.

## 6.2 System Design and Implementation

The HGaze Typing system takes gaze and head data as inputs. We designed two algorithms, fixation estimation and head gesture recognition, to process the gaze and head movement inputs respectively. The text entry interface uses the fixation information (including the fixated key information) and head gestures to perform text entry tasks. The system design is illustrated in Figure 6·2.

The HGaze Typing interface has four states: an idle state (also the initial state), a gaze lock state, a gaze path state, and a candidate selection state. When the system has detected a downward head movement, or an intention to nod, a gaze lock is activated to prevent the gaze shift due to the potential nod. When the head stops moving down, the lock will be released. Once a nodding gesture is captured, the gaze position information is unlocked and restored for gaze path classification.

The following sections will introduce the fixation estimation and head gesture recognition algorithms. The gaze lock mechanism and the paired gaze path restoring algorithm will also be described. Finally, we will explain the Fréchet score used for

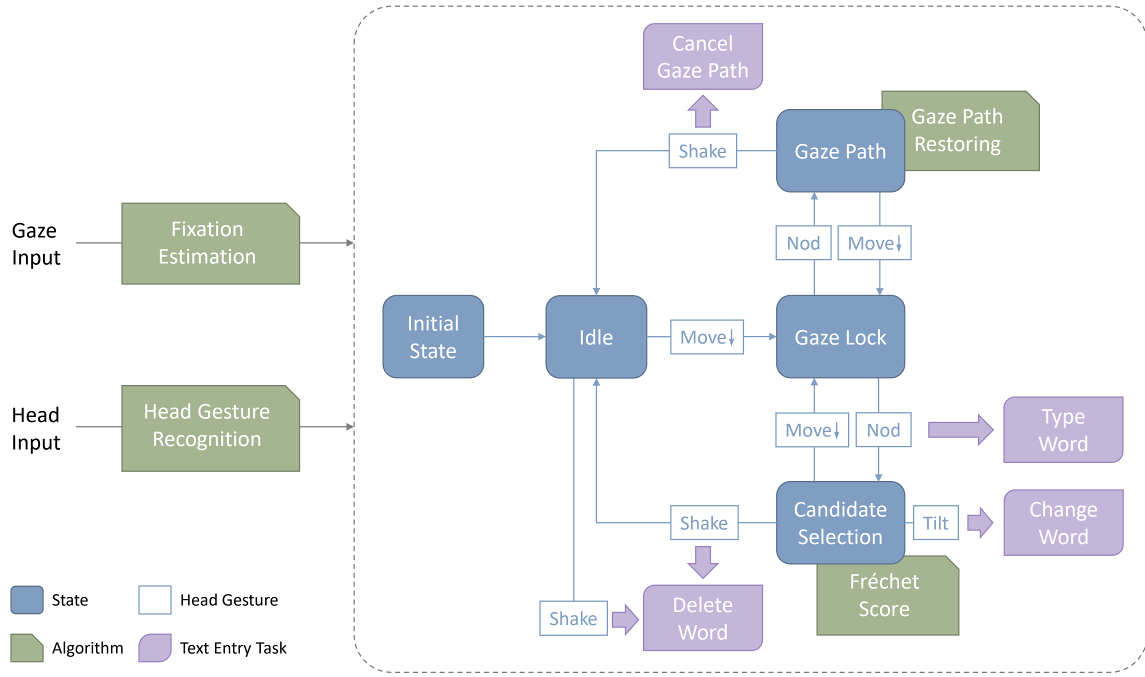



**Figure 6·2:** HGaze Typing takes eye and head movements as inputs. Algorithms are proposed for fixation estimation and head gesture recognition. The text entry interface uses the processed gaze and head information to perform text entry tasks. Detected head gestures can result in a transition between typing states or the execution of text entry tasks.

computing candidate words from a gaze path.

### 6.2.1 Fixation Estimation

The system first applies an average filter on the fixation data from the eye tracker to reduce the noise of the gaze inputs. This filter removes samples that are about 45 pixels (half of a key's height) away from their neighbors, which is the same as the EyeSwipe filter.

The filtered fixations are used for generating the gaze path and identifying each letter key the user intends to select. To estimate the intended key from the gaze input, the system computes the squared Euclidean distance between the center of every letter key and the filtered fixation position. The key closest to the gaze position is registered as the intended, currently focused key and marked with a red dot on the graphical interface.

The eye tracker sometimes fails to emit a fixation signal when a user looks at a key due to noise in the raw gaze input. This causes a delayed detection and can affect the text entry experience negatively. We used a time accumulation strategy to solve this problem and enhance the fixation estimation. When the user's gaze enters a virtual key, the system starts recording the time that the gaze stays inside the key (a temporary gaze shift outside the key of less than 50 ms does not count). Once the elapsed time exceeds a threshold, 80 ms in HGaze Typing, a fixation signal with the coordinates of the center of the key as the position information is generated.

### 6.2.2 Head Gesture Recognition

Three head gestures, nodding, shaking, and tilting, as well as the intention to nod (moving the head down), are detected based on the head pitch, yaw, and roll data from the eye tracker (the eye tracker we use provides the angle measurements at a frequency of 90 Hz, which corresponds to 11 ms per frame). The nodding and shaking gestures are mapped to text entry tasks performed only once in a short amount of time, which can be recognized by a simple classification method. Tilting, on the other hand, supports the holding operations, and is detected using a threshold-based method using both head rotation data and time information. The intention to nod can be easily recognized by the velocity of head movement in a downward direction.

#### Nod and Shake Recognition

The system first pre-processes the raw data stream of the (pitch, yaw, roll) angle vectors $\mathbf{u}_i \in \mathbb{R}^3$, where $i$ is the index in the data stream, as follows:

(a) Smooth the data stream by replacing each frame vector $\mathbf{u}_i \in \mathbb{R}^3$ with the average of the current vector values and the values of its two predecessors:

$$\mathbf{v}_1 = \mathbf{u}_1, \mathbf{v}_2 = \mathbf{u}_2, \quad \text{and} \quad \mathbf{v}_k = \frac{1}{3}(\mathbf{u}_{k-2} + \mathbf{u}_{k-1} + \mathbf{u}_k), \text{ for } k \geq 3, \tag{6.1}$$

and obtain vector $\mathbf{v}_i$.

(b) Then normalize the data stream by subtracting the mean of the previous K vector values:

$$\mathbf{w}_k = \mathbf{v}_k - \frac{1}{k}\sum_{i=1}^{k}\mathbf{v}_i \text{ for } k \leq K-1, \text{ and } \mathbf{w}_k = \mathbf{v}_k - \frac{1}{30}\sum_{i=k-K+1}^{k}\mathbf{v}_i \text{ for } k \geq K. \tag{6.2}$$

The result vector of preprocessing, $\mathbf{w}_i$, is smoothed and normalized so that each component of the data stream fluctuates around 0 degrees. The number $K$ of frames is empirically set to 30.

Then, the system takes templates from the user (using the same pre-processing method) and utilizes these templates to classify nodding and shaking gestures on each frame of the data stream. To collect a personalized template for each user, we asked the users to nod and shake their heads three times. The system pre-processes the data, finds the global extremum (e.g., the global minimum for pitch data for nodding), and saves 15 frames of the processed pitch, yaw, and roll data (before and with the global extremum) as templates. Each template can be seen as a $3 \times 15$ matrix. In total, there are six templates: three for each gesture.

Detections are performed on each frame of the data stream: $\mathbf{D} = (\mathbf{p}, \mathbf{y}, \mathbf{r})^T \in \mathbb{R}^{3\times 15}$ denotes the pre-processed data stream from the latest 15 frames, where $\mathbf{p}, \mathbf{y}, \mathbf{r} \in \mathbb{R}^{15}$ are vectors denoting the pitch, yaw, and roll data respectively. We denote the pre-processed temlate by $\overline{\mathbf{D}} = (\overline{\mathbf{p}}, \overline{\mathbf{y}}, \overline{\mathbf{r}})^T \in \mathbb{R}^{3\times 15}$.

The normalized cross-correlation (NCC) score that the system uses to compare the data stream angles with the template angles is computed as

$$\begin{aligned}\mathrm{NCC}(\mathbf{D},\overline{\mathbf{D}}) &\overset{\mathrm{def}}{=} \frac{\mathbf{D}:\overline{\mathbf{D}}}{\|\mathbf{D}\|_{L^2}\|\overline{\mathbf{D}}\|_{L^2}} \\ &= \frac{\mathbf{p}\cdot\overline{\mathbf{p}}+\mathbf{y}\cdot\overline{\mathbf{y}}+\mathbf{r}\cdot\overline{\mathbf{r}}}{\sqrt{\sum_{i=1}^{15}\left(\mathbf{p}_i^2+\mathbf{y}_i^2+\mathbf{r}_i^2\right)\sum_{i=1}^{15}\left(\overline{\mathbf{p}}_i^2+\overline{\mathbf{y}}_i^2+\overline{\mathbf{r}}_i^2\right)}},\end{aligned} \tag{6.3}$$

where $\mathbf{D}:\overline{\mathbf{D}}$ indicates the matrix inner product.

To decide whether there is a nod, we used the three user-nodding templates $\overline{\mathbf{D}}_{n,1}, \overline{\mathbf{D}}_{n,2}, \overline{\mathbf{D}}_{n,3}$ and computed the NCC "nodding score"

$$\mathrm{NCC}_n = \frac{1}{\sum_{i=1}^{3}\alpha_i}\sum_{i=1}^{3}\alpha_i\,\mathrm{NCC}(\mathbf{D},\overline{\mathbf{D}}_{n,i}) \tag{6.4}$$

as a weighted average of the NCC's with three user nodding templates. Empirically, the highest NCC is assigned the weight $\alpha_i = 2$ while the other two have weights of 1.

Analogously, for the three user shaking templates $\overline{\mathbf{D}}_{s,1}, \overline{\mathbf{D}}_{s,2}, \overline{\mathbf{D}}_{s,3}$, we define the NCC shaking score

$$\mathrm{NCC}_s = \frac{1}{\sum_{i=1}^{3}\alpha_i}\sum_{i=1}^{3}\alpha_i\,\mathrm{NCC}(\mathbf{D},\overline{\mathbf{D}}_{s,i}), \tag{6.5}$$

where the highest NCC has weight of $\alpha_i = 2$ as above and the other weights are 1.

The system classifies a nod or a shake if the NCC nodding score or the NCC shaking score is above a threshold. The default thresholds of nodding and shaking are set to 0.9 and 0.85, respectively, and can be adjusted during text entry. Note that the nodding and shaking gestures have distinct pitch, yaw, and roll data streams so that a gesture that has a high NCC nodding score will not have a high NCC shaking score (Figure 6·3).

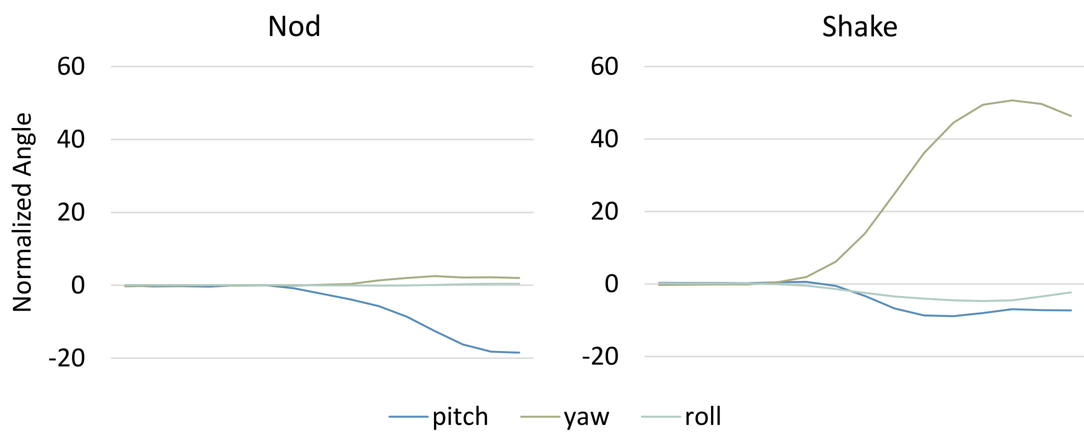



**Figure 6·3:** Example 15-frame templates of nodding and shaking

### Tilt Detection

We use a threshold-based algorithm to detect a left or right tilt as well as the possible holding action in the left or right tilt direction. When the roll value is larger than a predetermined threshold for $T$ frames, a right tilt is detected by the system. If the user keeps the right-tilting position, the system will change the selected word to its right candidate every 10 frames. The processes for the left side are similar. The number $T$ can be adjusted based on the user's performance and preference.

### Nodding Intention

A nodding intention is defined as a movement of the head downward. The system takes the pitch data and computes the velocity of pitch rotation. If the downward speed is above a threshold, a nodding intention is detected and registered in the system. The nodding intention will be canceled if the pitch rotation stops decreasing. In our implementation, a counter is added to handle the noise of the raw data. After a nodding intention is registered, the counter counts the number of frames in which the head is not moving down. If the count is more than $n$ frames, the nodding intention will be canceled. The velocity threshold is set to 1, and $n$ is set to 4 based on pilot tests. When a nodding gesture is classified, the nodding intention will also be canceled.

### 6.2.3 Gaze Lock

Head and eye movements can interfere with each other. Measurements of gaze direction will inevitably include a shift when the user nods for two reasons: (1) an remote eye-tracker cannot precisely estimate gaze position during a sudden head movement, and (2) users naturally move their gaze down a little when performing nods. In HGaze Typing, the gaze shift during a nodding gesture can cause an unwanted selection of a key. As a result, the accuracy of selecting the first and last letter of a word will decline and cause the prediction of a wrong word. To prevent this issue, we added a *gaze lock state* to the system. Once a nodding intention is detected, the gaze lock will be triggered. The system will stop receiving gaze inputs during this lock state, and the fixated key remains unchanged. The gaze lock will be released once a nod is detected or the nodding intention is canceled. Pilot studies showed that the gaze lock is not noticeable to participants and is an effective way of handling the nod-gaze interference.

### 6.2.4 Gaze Path Restoring

After a nod is detected, the gaze lock will release, and the gaze fixation inputs will be used for generating gaze path information. In this gaze path state, the gaze shift is likely to affect the first few fixations and therefore produce an inaccurate word candidate list. The system restores the gaze path by "dragging" the shifted gaze fixations back to the center of the first letter as we describe next.

We denote the point in time when the nod is detected by $t_0$ and the length of the time interval after the user selected the first letter of a word by $T$. The system computes a new fixation using a convex combination of the center position of the selected first letter and the current gaze fixation:

$$\text{Fixation}_{\text{new}}(t) = \frac{t - t_0}{T}\ \text{Fixation}(t) + \left(1 - \frac{t - t_0}{T}\right) \text{KeyCenter} \qquad (6.6)$$

for $t \in [t_0, t_0 + T]$. Here Fixation, $\text{Fixation}_{\text{new}}$, and KeyCenter are represented in 2D screen coordinates.

The above method will restore the first part of the gaze path to the user-desired path and alleviate the effects of a gaze shift. Once $T$ frames have been processed after a nod is detected, the system uses the actual gaze positions. The interval $T$ was empirically set to 500 ms.

### 6.2.5 Candidate Selection

HGaze Typing utilizes gaze paths to compute candidate words, similar to EyeSwipe. The nodding gestures specify the first and last letters of a word and segment the gaze path from the continuous gaze inputs. The first and last letters are used by the system to filter the lexicon when choosing the candidate words.

HGaze Typing interface utilizes the Fréchet score to select candidates. The Fréchet score is widely used in gesture and input pattern recognition (Sriraghavendra et al., 2007; Zhao et al., 2013; Despinoy et al., 2016). Kurauchi (2018) showed that the Fréchet score provided better gaze path classification than the DTW score. To compute the Fréchet score, the HGaze Typing system uses Bayes' theorem and the discrete Fréchet distance (Fréchet, 1906).

Intuitively, the problem of finding the Fréchet distance between two curves is that of a man walking his dog along a given curve while keeping his dog on the other curve. Both the man and the dog can vary their speeds but only walk forward. The Fréchet distance the length of the shortest leash that enables both the man and the dog to traverse their respective paths. Mathematically, a curve in $\mathbb{R}^2$ is a continuous map $G : [0,1] \to \mathbb{R}^2$. A prescribed walking speed is formulated as a reparametrization $\alpha : [0,1] \to [0,1]$ that is continuous, non-decreasing and surjective. With such notations, the continuous map $G \circ \alpha$ describes a curve in $\mathbb{R}^2$ with a prescribed speed. The Fréchet distance between two curves $G_1, G_2$ is defined as:

$$\begin{aligned} D_F(G_1, G_2) &\stackrel{\text{def}}{=} \inf_{\alpha_1,\alpha_2} \max_{t\in[0,1]} d_{\mathbb{R}^2}(G_1(\alpha_1(t)), G_2(\alpha_2(t))) \\ &= \inf_{\alpha_1,\alpha_2} \|G_1 \circ \alpha_1 - G_2 \circ \alpha_2\|_{L^\infty([0,1])}, \end{aligned} \tag{6.7}$$

where $d_{\mathbb{R}^2}$ is a distance function (or metric) on $\mathbb{R}^2$, and the infimum is taken over all reparametrizations $\alpha_1, \alpha_2$ on $[0,1]$. As a comparison, the DTW distance between two curves $A, B$ is:

$$\begin{aligned} D_{\text{DTW}}(G_1, G_2) &\stackrel{\text{def}}{=} \inf_{\alpha_1,\alpha_2} \int_0^1 d_{\mathbb{R}^2}(G_1(\alpha_1(t)), G_2(\alpha_2(t)))\, dt \\ &= \inf_{\alpha_1,\alpha_2} \|G_1 \circ \alpha_1 - G_2 \circ \alpha_2\|_{L^1([0,1])}. \end{aligned} \tag{6.8}$$

For two discrete curves (or time series data) with $p, q$ points respectively, both distances can be computed in $O(pq)$ using dynamic programming algorithms.

As pointed out by Kurauchi (2018), the discrete Fréchet distance inherits from the $L^\infty$ norm the property that it is extremely sensitive to the value at one point. This can be beneficial in our scenario. For example, with a gaze path resembling that of the word "apple," the discrete Fréchet distance favors the word "apple" much more than the word "apply" despite the fact that two words only differ by one letter. The downside is that when the user needs to search for a letter on the virtual keyboard (e.g., the user may not have memorized the QWERTY layout), points outside the ideal path will be added, which will significantly increase the discrete Fréchet distance.

The DTW score (or discrete DTW distance) does the contrary: it is accumulative and not sensitive to a single point. Consequently, the DTW score is less affected by users' word-searching but has lower word prediction accuracy than the Fréchet score.

The experiments by Kurauchi (2018) indicate that the Fréchet score was better overall for classifying gestures into words. This also indicates that, in these experiments, users' word-searching behaviors were rare.

The HGaze Typing system subsamples the gaze path and the ideal path to calculate the discrete Fréchet distance. For every pair of consecutive points $(a, b)$ in the path, points between $a$ and $b$ with a fixed step size are generated. After preprocessing, the numbers of points along each path have the same order of magnitude. According to Eiter and Mannila (1994), the computational cost of computing a discrete Fréchet distance is $O(pq)$, where $p$ and $q$ are the numbers of segments on the respective paths.

Next, the system computes the Fréchet score for words in the lexicon with the given first and last letters.

We next explain how to use the Fréchet score as a representation of the posterior distribution of the words in the lexicon, which combines the discrete Fréchet distance with the word probability of occurrence using Bayes' theorem.

The prior discrete distribution of the words in the lexicon $\mathcal{L}$ is calculated in the same way as in the DTW score:

$$\Pr(W = w_i) = \frac{|\text{Occurrences}(w_i)|}{\sum_{v \in \mathcal{L}} |\text{Occurrences}(v)|} \tag{6.9}$$

where $W$ is the random variable that represents the word entered by the user. Word occurrence data extracted from the Corpus of Contemporary American English (Davies, 2008) was used to compute the probability $\Pr(W = w_i)$. Based on Bayes' theorem, the system computes the probability

$$\Pr(W = w_i | G = g) = \frac{\Pr(G = g | W = w_i) \Pr(W = w_i)}{\Pr(G = g)} \tag{6.10}$$

where $G$ is the random variable representing the shape of the gaze path and $g$ is the gaze path from the user. As $\Pr(G = g)$ is independent of $w_i$, the systems searches for the $w_i$ that maximizes $\Pr(G = g|W = w_i) \cdot \Pr(W = w_i)$ and $\Pr(G = g)$ does not need to be calculated.

The probability $\Pr(G = g|W = w_i)$ is defined as:

$$\Pr(G = g|W = w_i) = \frac{S(g, w_i)}{\sum\limits_{w_j \in \mathcal{L}} S(g, w_j)} \tag{6.11}$$

where the gaze path score $S(g, w_i)$ is calculated as:

$$S(g, w_i) = \frac{|\,\mathrm{Ideal}(w_i)|}{1 + \mathrm{DFD}^k(g, \mathrm{Ideal}(w_i))}\,. \tag{6.12}$$

Here $\mathrm{DFD}(\cdot, \cdot)$ represents the discrete Fréchet distance and $k$ is a positive constant. The gaze path score is proportional to $|\,\mathrm{Ideal}(w_i)|$ in order to favor longer words and is inversely proportional to the distance $\mathrm{DFD}(g, \mathrm{Ideal}(w_i))$. A scaling constant $k$ is used to adjust the distribution of the distances, which is empirically set to 6.6.

## 6.3 Experiment

We conducted an experiment spanning nine sessions to evaluate the usability of HGaze Typing interface. In this experiment, we used a dwell-time keyboard as a baseline text entry method. The dwell-time method was selected as the baseline because it is widely used. Note that we could have chosen EyeSwipe as a comparison input interface but pilot tests showed that users sometimes performed the head movements they learned for HGaze Typing when they used EyeSwipe, which affected the performance of EyeSwipe negatively.

### 6.3.1 Participants

Eleven university students without physical impairments were recruited for the experiment. One participant met significant calibration issues with the eye tracker and did not complete the experimental sessions on the first day, resulting in a total of ten participants (4 males and 6 females, with an average age of 21.4). The participants were all native speakers of English and proficient in using the QWERTY physical keyboard layout. Two participants wore glasses, and one participant wore contact lenses. All participants had little or no experience with eye tracking or head tracking systems.

Each participant received a total of $50 compensation for participating in the study: $15 for a one-hour experiment on day one and day two, and $20 on day three.

### 6.3.2 Apparatus

We conducted the experiment on a laptop (3.70 GHz CPU, 16GB RAM) running Windows 10, connected to a 19-inch LCD monitor (1280 $\times$ 1024 px resolution). A Tobii Eye Tracker 4C with a sampling rate of 90 Hz was used to collect the gaze and head information.

The HGaze Typing interface and the dwell-time-based keyboard were built in C++ using the Qt framework. The HGaze Typing interface is shown in Figure 6·1, and the dwell-time keyboard used the same layout with an additional space key below the virtual letter keys. The lexicon was the same as the EyeSwipe interface: 10,219 words from the union of Kaufman's lexicon (Kaufman, 2015) and the words in MacKenzie and Soukoreff's phrase dataset (MacKenzie and Soukoreff, 2003). The dwell period was set to be the same as in the EyeSwipe experiment, 600 ms, following Hansen et al. (2003).

### 6.3.3 Procedure

We used a 9 × 2 within-subject design, with session (1–9) and interface (HGaze and dwell-time) as the independent variables, to evaluate the text entry speed, accuracy, performance, and user preference.

Each participant visited the lab on three different days (12–48 hours apart) and performed three sessions of typing with each interface per day, resulting in nine text entry sessions for each interface. The order of the interfaces was counterbalanced using a Latin square. Each session took 40 to 75 minutes.

Before the formal sessions of each interface on each day, there was a practice session in which participants typed 1–3 sentences. For both HGaze Typing and the dwell-time keyboard, every formal session on the first day contained 6 phrases. Participants typed 7 phrases in each session on the last two days (Figure 6·4).

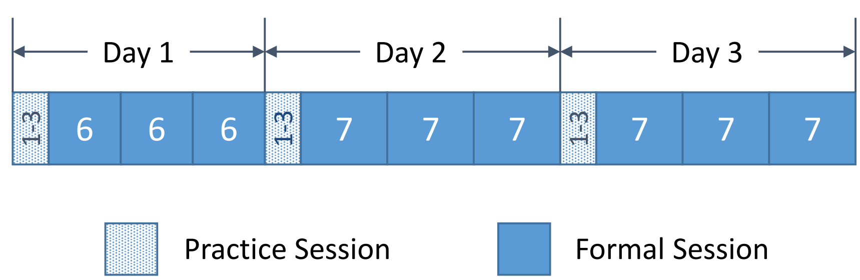



**Figure 6·4:** Experiment procedure for each interface. The number of phrases to be typed in each session is shown in the session box.

On the first day, the experimenter started the study with a brief introduction of eye-tracking and head-tracking systems and the two text entry interfaces. The eye tracker was calibrated for each participant at the beginning of each day and re-calibrated when necessary. The participants were instructed to seat comfortably in front of the screen with a distance of about 70 cm. Before the start of the HGaze Typing text entry sessions on each day, the users' head gestures were collected to generate personalized templates. The templates were generated automatically by the HGaze Typing system and adjusted when needed. The thresholds of head gesture recognition were modified during the practice session.

The participants were encouraged to memorize the phrase and type as fast and accurately as possible. The phrases given in the experimental sessions were randomly selected from MacKenzie and Soukoreff's dataset (MacKenzie and Soukoreff, 2003). For each typing trial, both interfaces compared the typed phrase with the given phrase and proceeded to the next trial automatically if there was a match. Alternatively, the participant could finish the current trial by selecting the "OK" key. Between sessions, participants could take a break of up to 5 minutes. At the end of the last session, the participants completed a short questionnaire on their demographics and their subjective feedback about the two text entry methods.

## 6.4 Results

In this section, we analyze the results and summarize the performance and subjective feedback of the two text entry interfaces. We collected 1,185 trials out of the possible 1,200 trials. Due to calibration errors or the user skipping a phrase by mistake, 15 trials had to be removed.

### 6.4.1 Text Entry Rate

The text entry rate was measured in words per minute, and a word is defined as a sequence of 5 characters. Overall, the mean text entry rate using HGaze Typing was higher than using the dwell-time keyboard (Figure 6·5). In the last three sessions, a significant effect of the interface was found ($F_{1,9} = 7.81$, $p = 0.021$), with an average typing speed of 11.22 wpm for HGaze Typing and 9.53 wpm for the dwell-time keyboard.

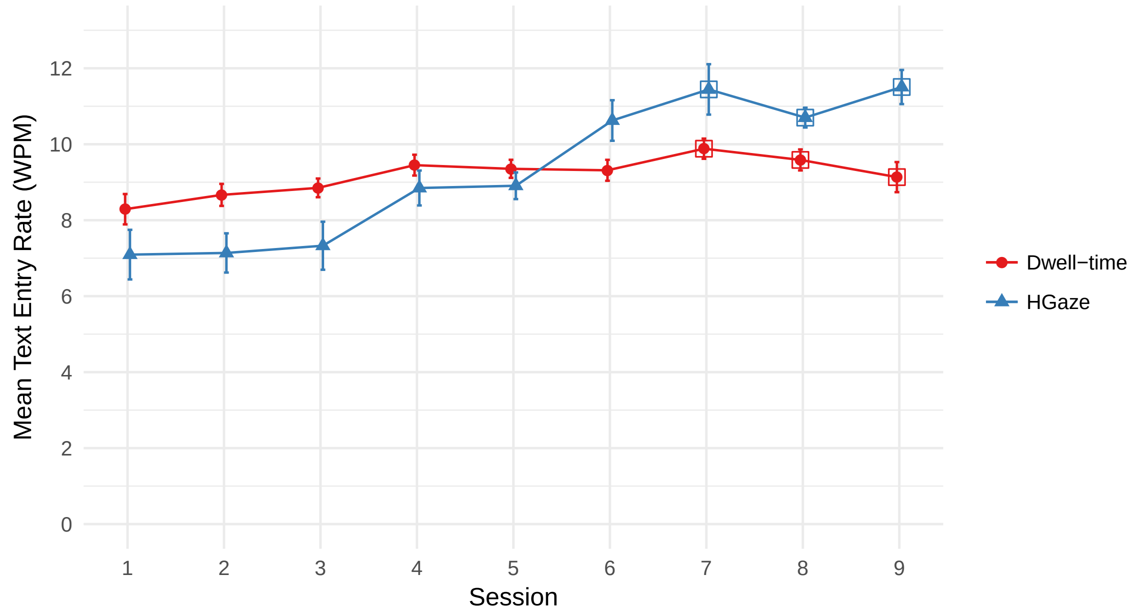



**Figure 6·5:** Mean and standard error of text entry rate in word per minute (wpm) for each session and interface. The text entry rates in the last three sessions are marked with squares.

For both methods, there was a significant effect of session on the text entry rate (HGaze: $F_{8,72} = 15.74$, $p < 0.001$; Dwell-time: $F_{8,72} = 5.42$, $p < 0.001$), which indicated a learning effect for both methods. The average text entry rate with HGaze Typing increased from 7.09 wpm in the first session to 11.5 wpm in the last session. Using the dwell-time keyboard, on the other hand, increased the typing speed from 8.29 wpm in the first session to 9.13 wpm in the last session. There was also a significant session $\times$ interface interaction on text entry rate ($F_{8,72} = 21.06$, $p < 0.001$). That is, extra training is likely to further expand the difference between HGaze Typing and dwell-time method in terms of typing speed.

The mean maximum text entry rate in each session with each typing method is shown in Figure 6·6. The maximum text entry rate with HGaze Typing was 16.21 wpm and 10.85 wpm with the dwell-time keyboard. Both interface and session had a significant effect on the maximum typing speed (interface: $F_{1,9} = 30.93$, $p < 0.001$;

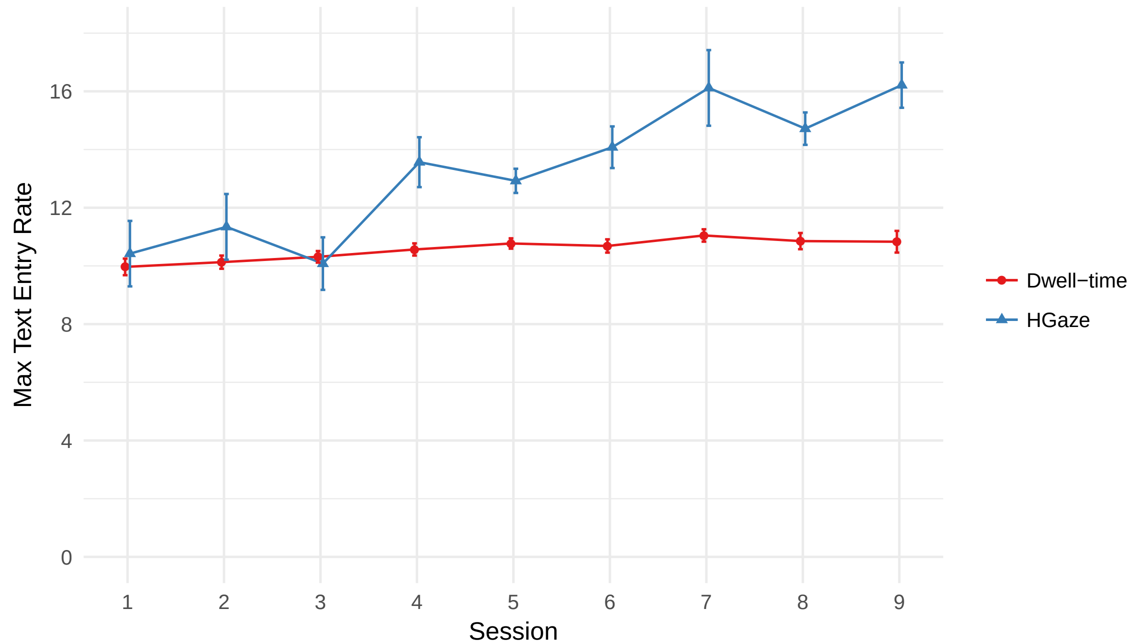



**Figure 6·6:** Mean and standard error of the maximum text entry rates in words per minute (WPM) for each session and interface.

session: $F_{8,72} = 10.04$, $p < 0.001$). The interaction between session and interface was also significant ($F_{8,72} = 5.22$, $p < 0.001$). The maximum text entry rate for HGaze Typing was 23 wpm. All participants achieved a text entry rate of at least 15 wpm using HGaze Typing and only 11 wpm using the dwell-time keyboard.

### 6.4.2 Accuracy

The average MSD error rate between the given phrase and the typed phrase in each session were less than 3.5% over the nine sessions, and less than 1.5% in the last six sessions for both HGaze Typing and the dwell-time keyboard. In the last session, the mean uncorrected error rate (Soukoreff and MacKenzie, 2003) using HGaze Typing was 0.37% and 0.21% using the dwell-time method. The low rate of uncorrected errors in the entered sentences indicated that participants were keeping the typed phrases accurate with both interfaces.

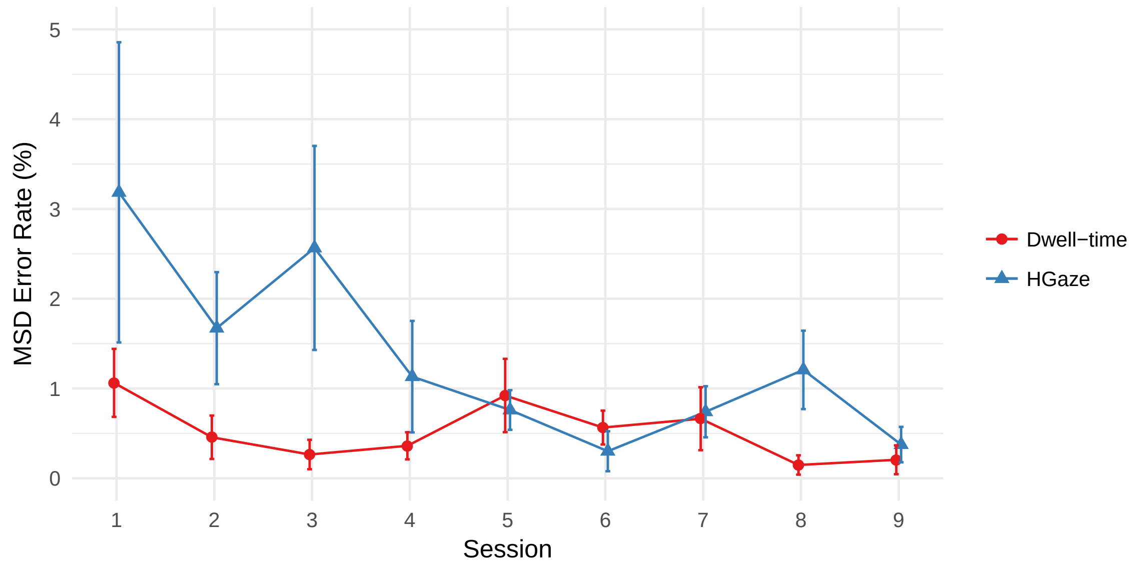



**Figure 6·7:** Mean and standard error of minimum string distance rate (MSD rate) for each session and interface.

The mean number of deletes and cancels per sentence in each session when typing with HGaze Typing interface is shown in Figure 6·8. The average number of deletes reduced from 1.75 in session 1 to 0.57 in session 9 ($F_{8,72} = 7.41$, $p < 0.001$), and the number of cancels reduced from 11.05 in session 1 to 0.6 in session 9 ($F_{8,72} = 2.69$, $p = 0.086$). The phrases provided in the experiment had an average length of 5.4 words. That is, for every ten words, the participants only performed about one cancel and one delete after practice. In the last session, the word deletion rate (the ratio between deleted words and typed words) and the cancel rate (the ratio between started gestures and typed words) were 10.8% and 11.4% respectively.

The deletion rate of the dwell-time keyboard was defined as the ratio between the number of backspaces and the total characters of the typed phrase (including spaces). The average deletion rate using the dwell-time keyboard was 3.8%, and there was no significant effect of session on deletion rate ($F_{8,72} = 2.04$, $p = 0.054$).

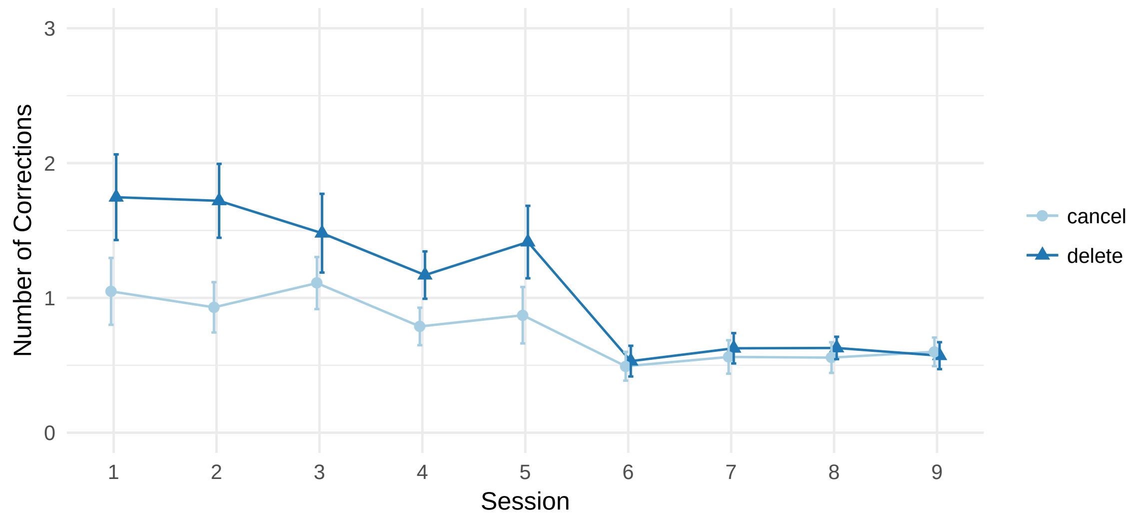



**Figure 6·8:** Mean and standard error of the number of deletes and cancels per sentence in each session with HGaze Typing.

### 6.4.3 Subjective Feedback

The post-experiment questionnaire asked participants to rate both interfaces on a 7-point scale of their performance and preference (Figure 6·9). Participants believed HGaze Typing had better performance (5.5) than the dwell-time keyboard (4.9). They also indicated a preference for HGaze Typing interface over the dwell-time method (5.8 vs. 4.9).

Speed, accuracy, comfort, and learnability were also evaluated on a scale of 1–7 (low-high) by participants, shown in Figure 6·9. HGaze Typing was rated higher on average for speed and general comfort, and the dwell-time keyboard was rated higher for its accuracy and learnability. Additionally, participants reported their eye-control effort and head-control effort for both interfaces. They reported the same eye-control effort using both interfaces – 4.6 on a scale of 1 (low effort) to 7 (high effort). However, some participants experienced eye strain only when using the dwell-time keyboard. HGaze Typing required a higher head-control effort of the participants (5.8) than the dwell-time keyboard (2.9). No participant reported neck fatigue using HGaze Typing.

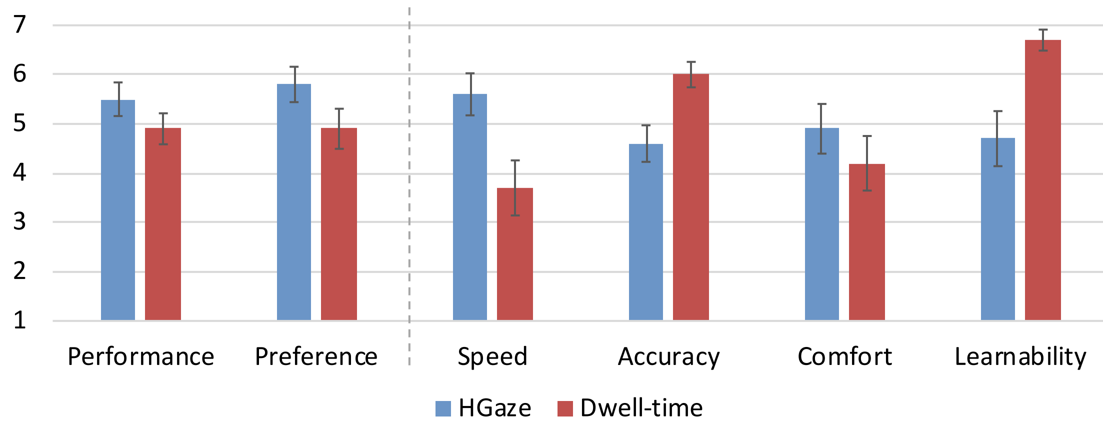



**Figure 6·9:** The average ratings of overall performance and preference, and perceived accuracy, speed, comfort and learnability, on a 7-point Likert scale.

### Feedback on the Dwell-time Keyboard

Similar to the feedback from the experiment in the previous chapter, the dwell-time keyboard was considered “very easy to understand and use” (P3) and a “predictable solution for typing” (P11). Participants perceived a slow typing speed but high accuracy when using the dwell-time keyboard: “It takes a long time to finish a sentence but it is accurate” (P4). “[The method is] slow, but accurate” (P8). They noted that typing long words were “tedious” (P11) and “take a lot more time” (P1).

P4 indicated that “the dwell-time keyboard required more energy and better eye precision for the best outcome.” Eye fatigue was a concern of using the dwell-time keyboard. Some participants reported eye strains after using the dwell-time keyboard for long: “I felt that it strained my eyes a lot more” (P1). “[The dwell-time method] is quick to fatigue the user” (P8).

### Feedback on HGaze Typing

The efficiency of HGaze Typing was highlighted by the study participants. They thought the interface was "much more time and speed efficient" (P3) and "was a lot nicer to be able to spell out longer words" (P1). The gaze path typing method adapted from EyeSwipe contributed to the speed and comfort: "As the program is able to easily recognize the words you are trying to type even with minimal eye precision" (P3).

As for the accuracy, which was not perceived to be as good as the dwell-time keyboard at the beginning, participants reported it improving after practice: "Accuracy increases over time with experience as does speed" (P8). HGaze Typing took a longer time for the participants to learn, compared with the dwell-time keyboard. P1 indicated that "Overall [I] liked this one (HGaze Typing) better after getting used to it".

Participants liked the concept of using various head gestures after practice. P9 reported that "[it] took a while to master the nod. Shake was fine, and so was the tilt. I really like the tilting concept to change words. I thought that was pretty unique." "The nod is easy to use and understand" (P8). P4 also showed favor toward using tilt gestures to choose from candidate list since "tilting was cool."

Participants had different opinions on the head gesture recognition model, especially nodding. The recognition was considered as "sensitive to even the slightest head movements" (P2) while sometimes "I had to nod my head several times" (P1). Collecting personalized head gesture templates enhanced the performance of gesture recognition: "The gestures are easy to use once calibrated well with the user's style of performing the gestures" (P3).

The "nodding - glancing through keys - nodding - (tilting)" pattern created a rhythm for this bi-modal text entry method. P11 indicated that "The [HGaze Typing] keyboard allowed me to establish a rhythm where I could imagine a sentence in my head and simultaneously type it out on the screen." He also considered the flow of using HGaze Typing was the same as using a swipe keyboard on a phone.

## 6.5 Feedback from PALS

To better understand how people with motor impairments assess gaze-path-based text entry and using head gestures in eye-typing, we conducted an informal study with five people with ALS and one caregiver to collect their qualitative feedback. In this study, we first explained the gaze-path-typing concept and demonstrated the EyeSwipe interface. We then presented and showed how the HGaze Typing interface works and is different from EyeSwipe. Two participants (A1, A3) with very limited head-movement range tried EyeSwipe only. Another participant (A2) tried both interfaces. The other two participants (A4, A5) who met calibration issues with the eye tracker, were present during the demo and the testing and provided feedback based on their observations.

All participants liked the idea of using gaze path typing. They thought that gaze-path-based typing was fun and very fast. A1 considered gaze-path-based text entry to be "decent and much easier with predictions and overall writing," but also believed that he would "need more time using it to see how good it really is." The caregiver also noted that the gaze path required familiarity with the QWERTY layout and would be efficient with practice.

A2 tried both interfaces and reported that the head gestures were easy to learn and perform. Other participants who can perform simple head gestures also showed a preference for HGaze Typing. A4 liked the concept but also wanted to reduce the frequency of head movements. Therefore, he suggested "this [interface] could be adapted to using a micro switch instead of head nodding."

## 6.6 Discussion

We designed and implemented HGaze Typing, a text entry interface combining simple head gestures and gaze paths. User study results showed that HGaze Typing had better overall performance and user satisfaction than a dwell-time keyboard. With HGaze Typing, participants achieved an average text entry rate of 11.5 wpm after 8 experimental sessions (about 56 phrases). The uncorrected error rate of using HGaze Typing is low, which indicates users will not need to balance speed and accuracy deliberately when using this system. Subjective results demonstrate that HGaze Typing is more comfortable and provides better-perceived performance than a dwell-time keyboard.

Adding head gestures to gaze-based text entry provides natural and efficient command activations. The study participants also liked the concept, and one participant noted that the nodding gestures created a pattern for text entry. Some text entry activities, such as choosing a word from a candidate list, require visually search before making a selection. With just gaze input, the two tasks - scanning and selecting - has to be done sequentially. The tilting gestures used in HGaze Typing allows simultaneous scanning visual activities and candidate selection.

Another advantage of using head gestures is that it can reduce the virtual keyboard area. By using shaking gestures, the delete/cancel button for gaze-based selection is no longer needed. Additionally, a word candidate list is necessary for gaze-path-based text entry interface as well as for features like auto-completion and word prediction. Gaze-based selection methods require a certain minimum button size, to handle gaze noise, and can consume a considerable area on the screen. Tilting gestures, on the other hand, provide navigation and use much less space to list words. The candidate list is small and is placed between the letter keys in the HGaze Typing interface.

HGaze Typing manages the processing of the gaze and head inputs in one system.

The average number of deletes and cancels is about one per ten words, indicating the robustness of the system. The gaze lock and gaze path restoring algorithms handled the head-gaze interference issue and the resulting gaze shift well. In the current system, head movement inputs are pitch, yaw, and roll data. In the future, we can use “richer” information from video input instead, which is likely to improve head gesture recognition. By utilizing additional head gestures in the gaze-based text entry systems, we can facilitate additional text entry and editing tasks.

The proposed design model that uses gaze path plus text entry commands can also extend to other devices and situations. Head and gaze inputs are prevalent in virtual reality and augmented reality devices, which this bi-modal text entry interface could be transfered to. People with motor impairments use various assistive technologies and sometimes have to stick to one kind. The text entry commands can not only be mapped to head gestures but also to the input from a switch or other assistive technology, as suggested by our study participants with ALS.

# Chapter 7

# Conclusions

This last chapter summarizes the main contributions of this thesis and discusses the strengths and limitations of our work. Important directions for future research in the areas of assistive technology and dwell-free input interface are pointed out at the end.

## 7.1 Summary of Contributions

This thesis reviewed a broad range of assistive technologies and investigated how people with motor impairments evaluated their effectiveness. This thesis also proposed three novel dwell-free input methods using eye-tracking and head-tracking systems that improved the efficacy of target selection and text entry.

In summary, the main contributions of this thesis are:

- A qualitative study with fifteen people with quadriplegia caused by degenerative neurological diseases that explores their usage, adaption, and assessment of various assistive technologies, as well as highlights the existing challenges and provides suggestions for further research.

- A dwell-free selection method, Target Reverse Crossing, which reduces target selection time.

- A text entry interface, EyeSwipe, using gaze paths that only requires the user to delineate the start and end of a word, with a gaze path connecting the intermediate characters.

- A bi-modal text entry interface, HGaze Typing, that effectively integrates head and gaze inputs, utilizing head gestures to perform text entry tasks and gaze paths to form words.

## 7.2 Strengths and Limitations

The qualitative study provided insights on the experience people with motor impairment have with a broad range of assistive technologies. The study revealed three existing obstacles to be solved: (1) limited opportunities to discover non-commercial assistive technologies, (2) inefficient text entry, and (3) lack of personalization and adaption of disease progression. Our proposed input methods improved the selection and text entry efficacy by using a dwell-free design in eye-tracking and head-tracking systems. We also added a personalization component to HGaze Typing – the interface collects the user's head gestures as templates to model their specific head movement patterns. However, our interfaces currently lack real-time user performance measurements, and therefore cannot automatically adapt to changing symptoms caused by degenerative neurological diseases.

User studies with the three input techniques showed that our proposed dwell-free input methods provided better performance and user experience than a static dwell-time method. The dwell-free input methods also allow users to look anywhere on the screen without time restrictions. A user can read the information on a button as long as needed before making a selection, or deliberately compose the next sentence to be typed without moving his or her gaze outside the virtual keyboard.

The main design question for dwell-free input methods is how we should separate commands from continuous gaze inputs. Target Reverse Crossing and EyeSwipe decompose "performing a command" into two steps: (1) showing an intention and (2) confirming the selection. By providing a dynamic visual element of a potential selection on a target or a key, Target Reverse Crossing and EyeSwipe allow a user to either confirm or cancel this selection with the element. The drawback of using dynamic visual elements is that they are not as intuitive to learn as traditional static dwell-time methods. To avoid additional learning efforts of adapting new keyboard layouts, we chose the QWERTY layout in our proposed text entry interfaces.

HGaze Typing uses head movement as an additional input for the user to perform commands, which allows more interactions and simple command activations, and free the eyes for viewing visual feedback from the interface. In selection tasks, gaze inputs provide a screen position, and the nodding gestures confirm the selections. This mechanism is analogous to the computer mouse selection method: navigating by moving the mouse and confirming selections by pressing the left button. As for the text entry, nodding gestures are used as a switch, toggling in and out of the gaze path mode. This is similar to the finger on and off a touch screen when typing with a swipe-based virtual keyboard. Other head gestures, such as shaking and tilting, were mapped to other text entry tasks to provide shortcuts for typing.

When designing a multi-modal interface with both head and gaze inputs, a major issue is to handle the commonly occurring head-gaze interference: (1) the eye tracker will provide inaccurate gaze position measurements during a fast head movement, and (2) the user's gaze is not stable when performing head gestures. In HGaze Typing, we designed the gaze lock state and gaze path restoring algorithm, which can handle the gaze shift during a nodding gesture. It should be noted that this solution is specifically designed for simple selection tasks and gaze-path-based text entry, and cannot recover the exact gaze position, which may be required by other applications (e.g., gaming or design).

## 7.3 Future Directions

As discussed in the previous section, we used dynamic visual elements to separate commands from continuous gaze inputs to enable dwell-free text entry. It is worth to explore how different dynamically changing visual elements may affect eye movements and user performance in gaze-based text entry. Another interesting research focus in dwell-free text entry is on-screen keyboards with a dynamic layout. The advantages of the dynamic layouts is that they only use a small screen area like pEYEWrite (Huckauf and Urbina, 2007) or enable high text entry rates like Dasher (Ward and MacKay, 2002). A drawback of such interfaces is that they require significant learning efforts.

Multi-modal text entry is a research direction that can benefit many research areas that include multiple input channels, such as assistive technology, augmented reality, and virtual reality. In our user studies, people with ALS also suggested mapping the inputs from other assistive technologies that they are comfortable or familiar with to certain text entry tasks. Note that different input channels may interfere with each other, like the gaze shift during a head gesture, which should be considered in designing multi-modal interfaces in the future.

The analysis of the interview results suggests that people with severe motor impairments usually have specific needs and therefore require personalization features of assistive technologies. Choosing the best settings of assistive interfaces and systems is a hurdle for most individuals with motor impairments and their caregivers. Additionally, people with degenerative neurological diseases have changing symptoms. That is, even if an appropriate set of assistive technologies was found, it may not be used for long. Exploring automatic measurements of and adaptation to users' performance is one of the most important and impactful directions in assistive technology research.

# References


AbleNet Switches (n.d.). Retrieved from `https://www.ablenetinc.com/technology/switches`.

Accot, J. and Zhai, S. (2002). More than dotting the i's – Foundations for crossing-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '02, pages 73–80, New York, NY, USA. ACM.

ALS Association (n.d.). Symptoms and diagnosis. Retrieved July 14, 2018, from `http://www.alsa.org/about-als/symptoms.html`.

Anthony, L., Kim, Y., and Findlater, L. (2013). Analyzing user-generated YouTube videos to understand touchscreen use by people with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1223–1232, New York, NY, USA. ACM.

Bates, R. and Istance, H. (2003). Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society*, 2(3):280–290.

Bee, N. and André, E. (2008). Writing with your eye: A dwell time free writing system adapted to the nature of human eye gaze. In *Proceedings of the 4th IEEE tutorial and research workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*, PIT '08, pages 111–122, Berlin, Heidelberg. Springer-Verlag.

Betke, M., Gips, J., and Fleming, P. (2002). The Camera Mouse: Visual tracking of body features to provide computer access for people with severe disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(1):1–10.

Bulling, A., Roggen, D., and Tröster, G. (2009). Wearable EOG goggles: Seamless sensing and context-awareness in everyday environments. *Journal of Ambient Intelligence and Smart Environments*, 1(2):157–171.

Carlson, D., Ehrlich, N., Berland, B. J., and Bailey, N. (2001). Assistive technology survey results: Continued benefits and needs reported by americans with disabilities. *Research Exchange*, 7(1):17 pages.

Choi, C., Na, Y., Rim, B., Kim, Y., Kang, S., and Kim, J. (2013). An SEMG computer interface using three myoelectric sites for proportional two-dimensional cursor motion control and clicking for individuals with spinal cord injuries. *Medical Engineering & Physics*, 35(6):777–783.

Cockburn, A. and Firth, A. (2004). Improving the acquisition of small targets. In O'Neill, E., Palanque, P., and Johnson, P., editors, *People and Computers XVII – Designing for Society*, pages 181–196, London. Springer London.

Corbin, J. and Strauss, A. (1998). *Basics of Qualitative Research: Procedures and Techniques for Developing Grounded Theory*. SAGE Publications.

Darragh, J. J., Witten, I. H., and James, M. L. (1990). The Reactive Keyboard: A predictive typing aid. *Computer*, 23(11):41–49.

Davies, M. (2008). The corpus of contemporary american english (COCA): 520 million words, 1990-present. `https://corpus.byu.edu/coca/`. Accessed: 2017-11-22.

Despinoy, F., Bouget, D., Forestier, G., Penet, C., Zemiti, N., Poignet, P., and Jannin, P. (2016). Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering*, 63(6):1280–1291.

Diaz-Tula, A., de Campos, F. M. S., and Morimoto, C. H. (2012). Dynamic context switching for gaze based interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 353–356, New York, NY, USA. ACM.

Diaz-Tula, A. and Morimoto, C. H. (2016). AugKey: Increasing foveal throughput in eye typing with augmented keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 3533–3544, New York, NY, USA. ACM.

Don Johnston Sensor Switch (n.d.). Retrieved from `http://donjohnston.com/sensor-switch-by-don-johnston/`.

Duchowski, A. (2007). *Eye tracking methodology: Theory and practice*, volume 373. Springer Science & Business Media.

Eiter, T. and Mannila, H. (1994). Computing discrete Fréchet distance. Technical report.

Evans, D. G., Drew, R., and Blenkhorn, P. (2000). Controlling mouse pointer position using an infrared head-operated joystick. *IEEE Transactions on Rehabilitation Engineering*, 8(1):107–117.

Feng, W., Chen, M., and Betke, M. (2013). Preliminary investigation of the impact of visual feedback on a camera-based mouse-replacement system. In *Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '13, pages 24:1–24:4, New York, NY, USA. ACM.

Feng, W., Chen, M., and Betke, M. (2014). Target Reverse Crossing: A selection method for camera-based mouse-replacement systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '14, pages 39:1–39:4, New York, NY, USA. ACM.

Feng, W., Sameki, M., and Betke, M. (2018). Exploration of assistive technologies used by people with quadriplegia caused by degenerative neurological diseases. *International Journal of Human–Computer Interaction*, 34(9):834–844.

Findlater, L., Jansen, A., Shinohara, K., Dixon, M., Kamb, P., Rakita, J., and Wobbrock, J. O. (2010). Enhanced Area Cursors: Reducing fine pointing demands for people with motor impairments. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*, UIST '10, pages 153–162, New York, NY, USA. ACM.

Fréchet, M. (1906). *Sur quelques points de calcul fonctionnel.* PhD thesis, Rendiconti del Circolo Matematico di Palermo. 22:1–74.

Fuhrer, C. and Fridie, S. (2001). There's a mouse out there for everyone. In *California State University, Extended Abstracts on Northridge's Annual International Conference on Technology and Persons with Disabilities.*

Gajos, K. Z., Weld, D. S., and Wobbrock, J. O. (2010). Automatically generating personalized user interfaces with Supple. *Artificial Intelligence*, 174(12-13):910–950.

Gajos, K. Z., Wobbrock, J. O., and Weld, D. S. (2008). Improving the performance of motor-impaired users with automatically-generated, ability-based interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 1257–1266, New York, NY, USA. ACM.

Gao, S., Wang, Y., Gao, X., and Hong, B. (2014). Visual and auditory brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 61(5):1436–1447.

Gizatdinova, Y., Špakov, O., and Surakka, V. (2012). Comparison of video-based pointing and selection techniques for hands-free text entry. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, AVI '12, pages 132–139, New York, NY, USA. ACM.

Grauman, K., Betke, M., Lombardi, J., Gips, J., and Bradski, G. (2003). Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society*, 2(4):359–373.

Grossman, T. and Balakrishnan, R. (2005). The Bubble Cursor: Enhancing target acquisition by dynamic resizing of the cursor's activation area. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 281–290, New York, NY, USA. ACM.

Hansen, J. P., Johansen, A. S., Hansen, D. W., Itoh, K., and Mashino, S. (2003). Command without a click: Dwell time typing by mouse and gaze selections. In et al., M. R., editor, *Human-Computer Interaction*, INTERACT '03, pages 121–128. IOS Press.

Hansen, J. P., Tørning, K., Johansen, A. S., Itoh, K., and Aoki, H. (2004). Gaze typing compared with input by head and hand. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications*, ETRA '04, pages 131–138, New York, NY, USA. ACM.

Harada, S., Landay, J. A., Malkin, J., Li, X., and Bilmes, J. A. (2008). The Vocal Joystick: Evaluation of voice-based cursor control techniques for assistive technology. *Disability and Rehabilitation: Assistive Technology*, 3(1-2):22–34.

Hawley, M. S. (2002). Speech recognition as an input to electronic assistive technology. *The British Journal of Occupational Therapy*, 65(1):15–20.

Heron, M., Hanson, V. L., and Ricketts, I. W. (2013). ACCESS: A technical framework for adaptive accessibility support. In *Proceedings of the 5th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, EICS '13, pages 33–42, New York, NY, USA. ACM.

Holton, J. A. (2007). The coding process and its challenges. In *The Sage Handbook of Grounded Theory*, pages 265–289. SAGE Publications.

Hoppe, S., Daiber, F., and Lchtefeld, M. (2013). Eype - Using eye-traces for eye-typing. In *Workshop on Grand Challenges in Text Entry. ACM International Conference on Human Factors in Computing Systems (CHI'13)*. ACM.

Huckauf, A. and Urbina, M. (2007). Gazing with pEYE: New concepts in eye typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization*, APGV '07, pages 141–141, New York, NY, USA. ACM.

Igarashi, T. and Hughes, J. F. (2001). Voice as sound: Using non-verbal voice input for interactive control. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 155–156, New York, NY, USA. ACM.

Isokoski, P. (2000). Text input methods for eye trackers using off-screen targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, ETRA '00, pages 15–21, New York, NY, USA. ACM.

Jacob, R. J. K. (1990). What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '90, pages 11–18, New York, NY, USA. ACM.

Jameson, A. (2003). The human-computer interaction handbook. chapter Adaptive Interfaces and Agents, pages 305–330. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.

Kane, S. K., Jayant, C., Wobbrock, J. O., and Ladner, R. E. (2009). Freedom to roam: A study of mobile device adoption and accessibility for people with visual and motor disabilities. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '09, pages 115–122, New York, NY, USA. ACM.

Kaufman, J. (2015). Google 10000 english. `https://github.com/first20hours/google-10000-english`. Accessed: 2015-09-22.

Keirn, Z. A. and Aunon, J. I. (1990). Man-machine communications through brainwave processing. *IEEE Engineering in Medicine and Biology Magazine*, 9(1):55–57.

Kim, J., Park, H., Bruce, J., Sutton, E., Rowles, D., Pucci, D., Holbrook, J., Minocha, J., Nardone, B., West, D., Laumann, A., Roth, E., Jones, M., Veledar, E., and Ghovanloo, M. (2013). The tongue enables computer and wheelchair control for people with spinal cord injury. *Science Translational Medicine*, 5(213):213ra166.

Krapic, L., Lenac, K., and Ljubic, S. (2015). Integrating blink click interaction into a head tracking system: Implementation and usability issues. *Universal Access in the Information Society*, 14(2):247–264.

Kristensson, P. O. and Vertanen, K. (2012). The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '12, pages 241–244, New York, NY, USA. ACM.

Kristensson, P.-O. and Zhai, S. (2004). SHARK2: 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, pages 43–52, New York, NY, USA. ACM.

Kurauchi, A., Feng, W., Joshi, A., Morimoto, C., and Betke, M. (2016). EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference* *on Human Factors in Computing Systems*, CHI '16, pages 1952–1956, New York, NY, USA. ACM.

Kurauchi, A., Feng, W., Morimoto, C., and Betke, M. (2015). HMAGIC: Head movement and gaze input cascaded pointing. In *Proceedings of the 8th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '15, pages 47:1–47:4, New York, NY, USA. ACM.

Kurauchi, A. T. N. (2018). *EyeSwipe: Text Entry Using Gaze Paths*. PhD thesis, University of São Paulo.

Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., and Billinghurst, M. (2018). Pinpointing: Precise head- and eye-based target selection for augmented reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 81:1–81:14, New York, NY, USA. ACM.

Lancastre, J. (n.d.). Stages of ALS. Retrieved July 14, 2018, from `https://alsnewstoday.com/stages-of-als/`.

Lombardi, J. and Betke, M. (2002). A camera-based eyebrow tracker for hands-free computer control via a binary switch. In *Proceedings of the 7th ERCIM Workshop, User Interfaces For All*, pages 199–200.

Mackenzie, I. S. and Felzer, T. (2010). SAK: Scanning ambiguous keyboard for efficient one-key text entry. *ACM Transactions on Computer-Human Interaction*, 17(3):11:1–11:39.

MacKenzie, I. S. and Soukoreff, R. W. (2003). Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 754–755, New York, NY, USA. ACM.

Majaranta, P., Ahola, U.-K., and Špakov, O. (2009). Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 357–360, New York, NY, USA. ACM.

Majaranta, P. and Bulling, A. (2014). Eye tracking and eye-based human–computer interaction. In Fairclough, Stephen H.and Gilleade, K., editor, *Advances in Physiological Computing*, pages 39–65. Springer, London.

Majaranta, P. and Räihä, K.-J. (2002). Twenty years of eye typing: Systems and design issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications*, ETRA '02, pages 15–22, New York, NY, USA. ACM.

Manresa-Yee, C., Varona, J., Perales, F. J., Negre, F., and Muntaner, J. J. (2008). Experiences using a hands-free interface. In *Proceedings of the 10th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '08, pages 261–262, New York, NY, USA. ACM.

Mappus IV, R. L., Venkatesh, G. R., Shastry, C., Israeli, A., and Jackson, M. M. (2009). An fNIR based BMI for letter construction using continuous control. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '09, pages 3571–3576, New York, NY, USA. ACM.

Medicare Learning Network (n.d.). G-codes for functional reporting. Retrieved June 13, 2018, from https://www.cms.gov/Outreach-and-Education/Medicare-Learning-Network-MLN/MLNProducts/Downloads/G-Codes-Chart-908924.pdf.

Meyer, D. E., Keith-Smith, J., Kornblum, S., Abrams, R. A., and Wright, C. E. (1990). Speed-accuracy tradeoffs in aimed movements: Toward a theory of rapid voluntary action. *Attention and Performance XIII*, pages 173–226.

Mihara, Y., Shibayama, E., and Takahashi, S. (2005). The Migratory Cursor: Accurate speech-based cursor movement by moving multiple ghost cursors using non-verbal vocalizations. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '05, pages 76–83, New York, NY, USA. ACM.

Missimer, E. and Betke, M. (2010). Blink and wink detection for mouse pointer control. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '10, pages 23:1–23:8, New York, NY, USA. ACM.

Montague, K., Nicolau, H., and Hanson, V. L. (2014). Motor-impaired touchscreen interactions in the wild. In *Proceedings of the 16th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '14, pages 123–130, New York, NY, USA. ACM.

Morimoto, C. H. and Amir, A. (2010). Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*, ETRA '10, pages 271–274, New York, NY, USA. ACM.

Mott, M. E., Williams, S., Wobbrock, J. O., and Morris, M. R. (2017). Improving dwell-based gaze typing with dynamic, cascading dwell times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2558–2570, New York, NY, USA. ACM.

National Multiple Sclerosis Society (n.d.). Who gets MS? Retrieved July 14, 2018, from http://www.nationalmssociety.org/What-is-MS/Who-Gets-MS.

NaturalPoint (n.d.). SmartNave. Retrieved from http://www.naturalpoint.com/smartnav/.

Nuance Communications (n.d.). Dragon Naturally Speaking. Retrieved from http://www.nuance.com/dragon/index.htm.

Origin Instruments Corporation (n.d.). HeadMouse Nano. Retrieved from `http://www.orin.com/access/headmouse/`.

Pedrosa, D., Pimentel, M. D. G., Wright, A., and Truong, K. N. (2015). Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing*, 6(1):3:1–3:37.

Perlin, K. (1998). Quikwriting: Continuous stylus-based text entry. In *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology*, UIST '98, pages 215–216, New York, NY, USA. ACM.

Phillips, B. and Zhao, H. (1993). Predictors of assistive technology abandonment. *Assistive Technology*, 5(1):36–45.

Plotkin, A., Sela, L., Weissbrod, A., Kahana, R., Haviv, L., Yeshurun, Y., Soroker, N., and Sobel, N. (2010). Sniffing enables communication and environmental control for the severely disabled. *Proceedings of the National Academy of Sciences*, 107(32):14413–14418.

Polital Enterprises (n.d.). Point-N-Click. Retrieved from `http://www.polital.com/pnc/`.

Porta, M. and Turina, M. (2008). Eye-S: A full-screen input modality for pure eye-based communication. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, pages 27–34, New York, NY, USA. ACM.

Poulton, E. (1975). Range effects in experiments on people. *The American Journal of Psychology*, 88(1):3–32.

Pregenzer, M. and Pfurtscheller, G. (1999). Frequency component selection for an EEG-based brain to computer interface. *IEEE Transactions on Rehabilitation Engineering*, 7(4):413–419.

Price, K. J. and Sears, A. (2009). The development and evaluation of performance-based functional assessment: A methodology for the measurement of physical capabilities. *ACM Transactions on Accessible Computing*, 2(2):10:1–10:31.

Räihä, K.-J. (2015). Life in the fast lane: Effect of language and calibration accuracy on the speed of text entry by gaze. In Abascal, J., Barbosa, S., Fetter, M., Gross, T., Palanque, P., and Winckler, M., editors, *Human-Computer Interaction INTERACT 2015*, volume 9296 of *Lecture Notes in Computer Science*, pages 402–417. Springer International Publishing.

Räihä, K.-J. and Ovaska, S. (2012). An exploratory study of eye typing fundamentals: Dwell time, text entry rate, errors, and workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pages 3001–3010, New York, NY, USA. ACM.

Ratanamahatana, C. and Keogh, E. J. (2004). Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*.

Riviere, C. and Thakor, N. V. (1996). Effects of age and disability on tracking tasks with a computer mouse: accuracy and linearity. *Journal of Rehabilitation Research and Development*, 33(1).

Rough, D., Vertanen, K., and Kristensson, P. O. (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, pages 169–176, New York, NY, USA. ACM.

Sakoe, H. and Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Salivia, G. and Hourcade, J. P. (2013). PointAssist: Assisting individuals with motor impairments. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1213–1222, New York, NY, USA. ACM.

Sarcar, S., Panwar, P., and Chakraborty, T. (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, pages 215–220, New York, NY, USA. ACM.

Scherer, M. J. and Craddock, G. (2002). Matching person & technology (MPT) assessment process. *Technology and Disability*, 14(3):125–131.

Sears, A., Feng, J., Oseitutu, K., and Karat, C.-M. (2003). Hands-free, speech-based navigation during dictation: Difficulties, consequences, and solutions. *Human-Computer Interaction*, 18(3):229–257.

Sears, A., Karat, C.-M., Oseitutu, K., Karimullah, A., and Feng, J. (2001). Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society*, 1(1):4–15.

Shinohara, K. and Wobbrock, J. O. (2011). In the shadow of misperception: Assistive technology use and social interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 705–714, New York, NY, USA. ACM.

Sibert, L. E. and Jacob, R. J. K. (2000). Evaluation of eye gaze interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '00, pages 281–288, New York, NY, USA. ACM.

Sloan, D., Atkinson, M. T., Machin, C., and Li, Y. (2010). The potential of adaptive interfaces as an accessibility aid for older web users. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility*, W4A '10, pages 35:1–35:10, New York, NY, USA. ACM.

Soukoreff, R. W. and MacKenzie, I. S. (2003). Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 113–120, New York, NY, USA. ACM.

Sporka, A. J., Felzer, T., Kurniawan, S. H., Poláček, O., Haiduk, P., and MacKenzie, I. S. (2011). CHANTI: Predictive text entry using non-verbal vocal input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 2463–2472, New York, NY, USA. ACM.

Sriraghavendra, E., Karthik, K., and Bhattacharyya, C. (2007). Fréchet distance based approach for searching online handwritten documents. In *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 461–465. IEEE.

Surakka, V., Illi, M., and Isokoski, P. (2004). Gazing and frowning as a new human–computer interaction technique. *ACM Transactions on Applied Perception*, 1(1):40–56.

Surdilovic, T. (2006). Fuzzy mouse cursor control system for computer users with spinal cord injuries. Master's thesis, Georgia State University.

The Eye Tribe (n.d.). Retrieved from `https://theeyetribe.com/`.

Tobii Technology (n.d.). Retrieved from `http://www.tobii.com/`.

Trewin, S. (2000). Configuration agents, control and privacy. In *Proceedings on the 2000 Conference on Universal Usability*, CUU '00, pages 9–16, New York, NY, USA. ACM.

Trewin, S. (2004). Automating accessibility: the Dynamic Keyboard. In *Proceedings of the 6th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '04, pages 71–78, New York, NY, USA. ACM.

Trewin, S., Keates, S., and Moffatt, K. (2006). Developing Steady Clicks: A method of cursor assistance for people with motor impairments. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '06, pages 26–33, New York, NY, USA. ACM.

Trewin, S. and Pain, H. (1999). Keyboard and mouse errors due to motor disabilities. *International Journal of Human-Computer Studies*, 50(2):109 – 144.

Trnka, K., McCaw, J., Yarrington, D., McCoy, K. F., and Pennington, C. (2009). User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing*, 1(3):17:1–17:34.

Vaidyanathan, R., Chung, B., Gupta, L., Kook, H., Kota, S., and West, J. D. (2007). Tongue-movement communication and control concept for hands-free human–machine interfaces. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 37(4):533–546.

Špakov, O., Isokoski, P., and Majaranta, P. (2014). Look and lean: Accurate head-assisted eye pointing. In *Proceedings of the Symposium on Eye Tracking Research and Applications*, ETRA '14, pages 35–42, New York, NY, USA. ACM.

Špakov, O. and Majaranta, P. (2012). Enhanced gaze interaction using simple head gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp '12, pages 705–710, New York, NY, USA. ACM.

Ward, D. J. and MacKay, D. J. (2002). Fast hands-free writing by gaze direction. *Nature*, 418:838.

Williams, M. R. and Kirsch, R. F. (2008). Evaluation of head orientation and neck muscle EMG signals as command inputs to a human–computer interface for individuals with high tetraplegia. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 16(5):485–496.

Winwaed Software Technology LLC (2012). Calculating word and n-gram statistics from a wikipedia corpora. `http://www.monlp.com/2012/04/16/calculating-word-and-n-gram-statistics-from-a-wikipedia-corpora/`. Accessed: 2015-09-22.

Witiw, C. D. and Fehlings, M. G. (2015). Acute spinal cord injury. *Journal of Spinal Disorders & Techniques*, 28(6):202–210.

Wobbrock, J. O., Fogarty, J., Liu, S.-Y. S., Kimuro, S., and Harada, S. (2009). The Angle Mouse: Target-agnostic dynamic gain adjustment based on angular deviation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 1401–1410, New York, NY, USA. ACM.

Wobbrock, J. O. and Gajos, K. Z. (2007). A comparison of area pointing and goal crossing for people with and without motor impairments. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07, pages 3–10, New York, NY, USA. ACM.

Wobbrock, J. O. and Gajos, K. Z. (2008). Goal Crossing with mice and trackballs for people with motor impairments: Performance, submovements, and design directions. *ACM Transactions on Accessible Computing*, 1(1):4:1–4:37.

Wobbrock, J. O., Rubinstein, J., Sawyer, M. W., and Duchowski, A. T. (2008). Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications*, ETRA '08, pages 11–18, New York, NY, USA. ACM.

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., and Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical neurophysiology*, 113(6):767–791.

Worden, A., Walker, N., Bharat, K., and Hudson, S. (1997). Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, CHI '97, pages 266–271, New York, NY, USA. ACM.

Wu, W., Shaikhouni, A., Donoghue, J., and Black, M. (2004). Closed-loop neural control of cursor motion using a Kalman filter. In *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 4126–4129.

Zhai, S. and Kristensson, P.-O. (2003). Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, pages 97–104, New York, NY, USA. ACM.

Zhai, S. and Kristensson, P. O. (2012). The word-gesture keyboard: Reimagining keyboard interaction. *Commun. ACM*, 55(9):91–101.

Zhang, X., Kulkarni, H., and Morris, M. R. (2017). Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 2878–2889, New York, NY, USA. ACM.

Zhao, X., Feng, T., and Shi, W. (2013). Continuous mobile authentication using a novel graphic touch gesture feature. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, pages 1–6. IEEE.

# CURRICULUM VITAE

[redacted]