



# HGaze Typing: Head-Gesture Assisted Gaze Typing

Wenxin Feng  
Google  
Mountain View, CA, United States  
wenxinfeng@google.com

Jiangnan Zou  
Google  
Mountain View, CA, United States  
jalonzou@google.com

Andrew Kurauchi  
Insper Institute of Education and  
Research  
São Paulo, Brazil  
andrewtnk@insper.edu.br

Carlos Morimoto  
University of São Paulo  
São Paulo, Brazil  
hitoshi@ime.usp.br

Margrit Betke  
Boston University  
Boston, MA, United States  
betke@bu.edu

## ABSTRACT

This paper introduces a bi-modal typing interface, HGaze Typing, which combines the simplicity of head gestures with the speed of gaze inputs to provide efficient and comfortable dwell-free text entry. HGaze Typing uses gaze path information to compute candidate words and allows explicit activation of common text entry commands, such as selection, deletion, and revision, by using head gestures (nodding, shaking, and tilting). By adding a head-based input channel, HGaze Typing reduces the size of the screen regions for cancel/deletion buttons and the word candidate list, which are required by most eye-typing interfaces. A user study finds HGaze Typing outperforms a dwell-time-based keyboard in efficacy and user satisfaction. The results demonstrate that the proposed method of integrating gaze and head-movement inputs can serve as an effective interface for text entry and is robust to unintended selections.

## CCS CONCEPTS

• Human-centered computing → Interaction techniques.

## KEYWORDS

Text entry, dwell-free typing, multi-modal text entry, eye tracking, head gestures

### ACM Reference Format:

Wenxin Feng, Jiangnan Zou, Andrew Kurauchi, Carlos Morimoto, and Margrit Betke. 2021. HGaze Typing: Head-Gesture Assisted Gaze Typing. In *2021 Symposium on Eye Tracking Research and Applications (ETRA '21 Full Papers)*, May 25–27, 2021, Virtual Event, Germany. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/3448017.3457379>

## 1 INTRODUCTION

People with motor impairments use various assistive technologies to access and interact with computers. Among them, eye-tracking and head-tracking systems involve muscles that are unaffected in most conditions and therefore can be used by people with different

levels of motor impairments. 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 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 and writing text.

Eye-tracking and head-tracking systems can cause unintended selections, also known as the Midas Touch problem [Jacob 1990]. To address this issue, many gaze and head-based input interfaces use selection by dwell-time, i.e., to select a button or a key on a virtual keyboard the user has to maintain the cursor within the key region for a given period of time. However, this method is slow because the dwell period needs to be sufficiently long to prevent unintentional selections. 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. Recent studies [Kristensson and Vertanen 2012; Kurauchi et al. 2016] found that text entry by eye-swiping can be faster and more natural than traditional dwell-time-based eye typing.

The combination of gaze and head movements provides new opportunities for efficient and accurate multimodal interaction. In gaze-based systems, user saccades allow to quickly point to targets. However, unconscious small eye movements along with eye-tracking-system errors make the corresponding input data noisy, which is problematic for fine pointing tasks. Previous work [Kurauchi et al. 2015; Kytö et al. 2018; Špakov et al. 2014] has shown that adding accurate and stable head movements to gaze-based systems allows faster and more precise target pointing and selection. How to integrate head and gaze inputs for text entry, however, is still under explored.

We propose HGaze Typing, a novel dwell-free text entry system that combines gaze paths with head gestures. The main contributions of this system are: (1) HGaze Typing takes advantage of the speed of eye saccades and the accuracy of head movements by associating most common text entry tasks (e.g., selection and deletion) with simple head gestures, and letter-by-letter word path generation with gaze input; (2) The system reduces the screen region size used for the cancel/deletion button and the word candidate list; (3) HGaze Typing effectively integrates head and gaze inputs so that



This work is licensed under a Creative Commons Attribution International 4.0 License.

ETRA '21 Full Papers, May 25–27, 2021, Virtual Event, Germany

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8344-8/21/05.

<https://doi.org/10.1145/3448017.3457379>

the two modes do not interfere with each other; (4) This bi-modal system enables a wider set of interactions than would be possible with either mode alone.

## 2 RELATED WORK

In this section, we summarize gaze-based and head-based text entry methods, and discuss methods of combining gaze and head inputs.

### 2.1 Gaze-based Text Entry

We categorize gaze-based text entry interfaces based on two features. The first feature considers the input element such as a letter, a word, or a text of any length, and the second considers three interaction modes: dwell-time, gesture, and continuous writing interfaces.

**2.1.1 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. Word prediction is an effective feature to accelerate dwell-time-based keyboards [Diaz-Tula et al. 2012; Diaz-Tula and Morimoto 2016; Trnka et al. 2009]. Adjustable or cascading dwell-time keyboards allow the user to change the dwell period according to their typing rhythm and result in a higher typing rate (about 20 wpm) [Majaranta et al. 2009; Rähä and Ovaska 2012]. Mott et al. [2017] proposed a cascading dwell-time method, which achieved an average typing rate of 12.4 wpm by utilizing a language model to compute the probability of each key being entered next and adjusts the dwell-time accordingly.

**2.1.2 Gesture-based interface.** Previous work investigated different gaze gestures to replace dwell-time selections of a single character [Isokoski 2000; Porta and Turina 2008; Sarcar et al. 2013; Wobbrock et al. 2008]. Gaze gestures associated with dynamic visual elements have also been proposed. Huckauf and Urbina [2007] designed pEYEWite, an expandable pie menu with letter groups, in which a user can type a letter by simply crossing the borders of the corresponding sections. 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. Shorthand Aided Rapid Keyboarding (SHARK) introduced word-shaped-based typing in a broader text entry literature, which maps gestures on a touch-screen virtual keyboard to words in a lexicon [Kristensson and Zhai 2004; Zhai and Kristensson 2003, 2012]. Kristensson and Vertanen [2012] showed the potential of dwell-free word-shape-based eye typing in a pilot experiment. The Filtered typing interface asks the user to look at the keys that form a word and then look at a button to list word candidates [Pedrosa et al. 2015]. Kurauchi et al. [2016] proposed a word-path-based text entry interface, EyeSwipe, that predicted words based on the trajectory of a user's gaze while the user scans over the keyboard. The word-level gaze gesture concept can be applied to a non-QWERTY keyboard for small screens [Zhang et al. 2017].

**2.1.3 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. 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 [Diaz-Tula et al. 2012; Morimoto and Amir 2010]. The interface has two duplicated keyboard regions, and the last fixated key is selected when a user's gaze switches between the keyboard regions.

### 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. The study showed that participants made fewer errors with the head-based method than with 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 the gaze- and head-based pointing method, and achieved an average text entry rate of 4.42 wpm and 10.98 wpm respectively. They further explored the effects of a key size reduction on the accuracy and speed performance of text entry with video-based eye-tracking and head-tracking systems, and showed that head-tracking systems supported more accurate and faster text entry than eye-tracking systems for the smallest key size [Gizatdinova et al. 2018].

Facial expressions or gestures, as an alternative selection method [Krapic et al. 2015; Lombardi and Betke 2002; Missimer and Betke 2010], can also be applied to text entry tasks [Gizatdinova et al. 2012; Grauman et al. 2003]. 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 wpm with brow-up. Our work is the first to use head gestures for editing commands in gaze-based or head-based text entry systems.

### 2.3 Combination of Gaze and Head Input

As head-based pointing is more stable and easier to control while not as fast as gaze-based pointing [Bates and Istance 2003], some studies proposed and examined head-assisted eye pointing [Kurauchi et al. 2015; Kytö et al. 2018; Sidenmark et al. 2020; Špakov et al. 2014]. The multi-modal pointing method allows using gaze to perform a coarse long-distance jump and using head movements to make fine adjustments. User experiments demonstrated that combining the accuracy of head movements and the speed of gaze increased the efficacy of pointing. Mardanbegi et al. [2012] used the vestibulo-ocular reflex to detect head movements and allowed users to maintain their focus on a target while issuing a command.

Instead of using both modes to improve target pointing and selection, in this paper, we propose the use of head gestures for editing commands and eye gaze for fast and hands-free text entry.

### 3 INTERFACE DESIGN

HGaze Typing offers an enhanced typing experience with richer text editing features. The interface can achieve efficient text entry by using a gaze-path-based interface combined with head gestures (nodding, shaking, and tilting) to start, end, and cancel a gaze path; select candidate words; and delete words.

#### 3.1 Design Principles

Head-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 [Feng et al. 2018]. Gaze can be used to reduce head movements and help to create a faster and more comfortable interaction.

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) using head gestures to perform tasks requiring high accuracy, (2) using head gestures for commands associated with visual activities, (3) using gaze inputs for tasks that can be accomplished with fast movements and a coarse level of accuracy.

#### 3.2 Head Gestures and Text Entry Tasks

A gaze-based text entry interface that uses a word path method (e.g., EyeSwipe [Kurauchi et al. 2016]) must provide solutions for five text entry tasks: (1) initiating or finishing a word path by selecting the first letter or the last letter respectively, (2) connecting (by gaze inputs or head movements) the middle letters of the desired word to generate a path, (3) canceling a path when it is started involuntarily, (4) choosing a word from a candidate list, and (5) deleting an unwanted word.

According to the design principles, we map nodding gestures to selection tasks. Gaze path recognition depends on the correct selection 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 head gestures and their corresponding text entry tasks are shown in Table 1.

#### 3.3 HGaze Typing Interface

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

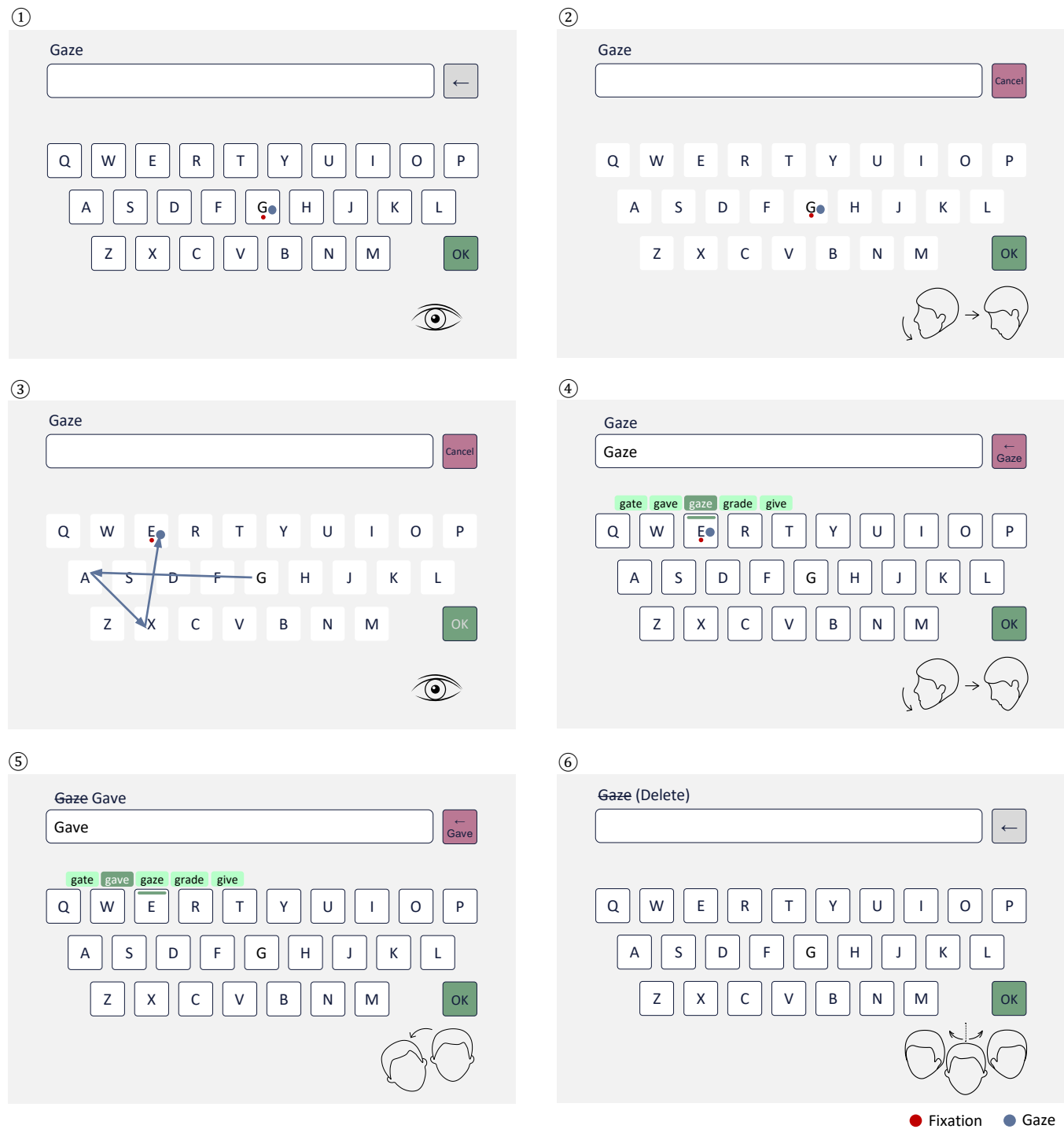
If the desired word is not in the candidate list, the user can delete the typed word by shaking his or her head. If the user accidentally triggers an unwanted gaze path, he or she can also cancel the current path with this gesture. A cancel/delete indicator is placed to the right of the text box, which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

### 4 SYSTEM DESIGN AND IMPLEMENTATION

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 (Figure 2). 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. Once a nodding gesture is completed, the gaze position information is unlocked and restored for gaze path classification. We designed two algorithms, fixation estimation and head gesture recognition, to process the gaze and head movement data required to operate the HGaze Typing interface. We also proposed the gaze lock mechanism and a paired gaze path restoring algorithm to minimize the effects of gaze shift from head movements.

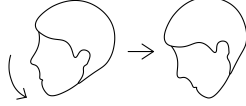
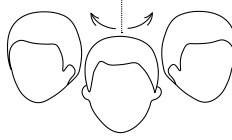
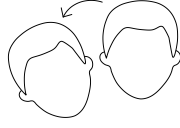
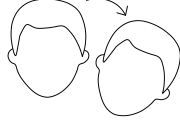
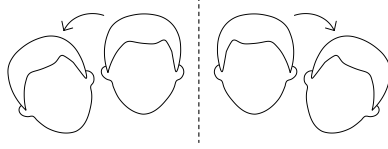
We designed two algorithms, fixation estimation and head gesture recognition, to process the gaze and head movement data required to operate the HGaze Typing interface.

The following sections introduce the fixation estimation and head gesture recognition algorithms. The gaze lock mechanism and the paired gaze path restoring algorithm are also described. Finally,



**Figure 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 with 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 is shown 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).

**Table 1: Head gestures and the corresponding text entry tasks.**

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

we explain the Fréchet score used for computing candidate words from a gaze path.

#### 4.1 Fixation Estimation

The system first applies an average filter on the fixation data from the eye tracker to reduce the noise from the gaze input. This filter removes samples that are about 45 pixels (half of a key's height) away from their neighbors. The system uses the filtered fixations to compute the gaze path and identify 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.

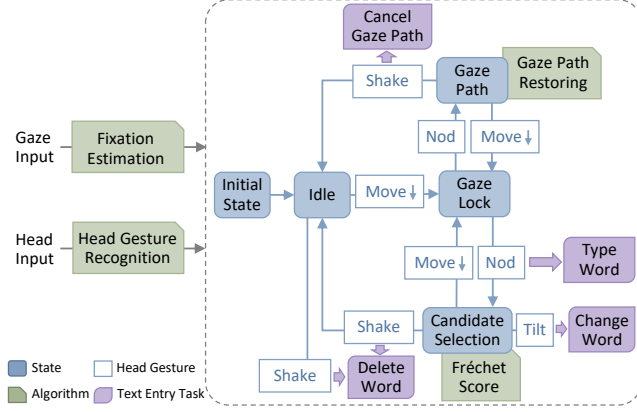
The eye tracker sometimes fails to emit a fixation signal when a user looks at a key due to noise in the raw gaze input. This causes a delayed detection and can affect the text entry experience negatively. We used a time accumulation strategy to solve this problem and enhance the fixation estimation. When the user's gaze enters a virtual key, the system starts recording the time that the

gaze stays inside the key (temporary gaze shifts outside the key shorter than 50 ms are ignored). 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.

#### 4.2 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 (at a frequency of 90 Hz). 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 template matching method. Tilting 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.

**4.2.1 Nod and Shake Recognition.** The system takes templates from the user and uses these templates to classify nodding and shaking



**Figure 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.**

gestures. To collect a personalized template for each user, we asked the users to nod and shake their heads three times. The system finds the global extremum (e.g., the global minimum for pitch data for nodding), and saves 15 frames before the global extremum of the pitch, yaw, and roll data as templates. Each template can be seen as a  $3 \times 15$  matrix.

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 template  $\bar{\mathbf{D}} = (\bar{\mathbf{p}}, \bar{\mathbf{y}}, \bar{\mathbf{r}})^T \in \mathbb{R}^{3 \times 15}$ .

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

$$\begin{aligned} \text{NCC}(\mathbf{D}, \bar{\mathbf{D}}) &\stackrel{\text{def}}{=} \frac{\mathbf{D} : \bar{\mathbf{D}}}{\|\mathbf{D}\|_{L^2} \|\bar{\mathbf{D}}\|_{L^2}} \\ &= \frac{\mathbf{p} \cdot \bar{\mathbf{p}} + \mathbf{y} \cdot \bar{\mathbf{y}} + \mathbf{r} \cdot \bar{\mathbf{r}}}{\sqrt{\sum_{i=1}^{15} (\mathbf{p}_i^2 + \mathbf{y}_i^2 + \mathbf{r}_i^2) \sum_{i=1}^{15} (\bar{\mathbf{p}}_i^2 + \bar{\mathbf{y}}_i^2 + \bar{\mathbf{r}}_i^2)}} \end{aligned} \quad (1)$$

To decide whether there is a nod, we use the three user-nodding templates  $\bar{\mathbf{D}}_{n,1}, \bar{\mathbf{D}}_{n,2}, \bar{\mathbf{D}}_{n,3}$  and computed the NCC “nodding score”,  $\text{NCC}_n = \frac{1}{\sum_{i=1}^3 \alpha_i} \sum_{i=1}^3 \alpha_i \text{NCC}(\mathbf{D}, \bar{\mathbf{D}}_{n,i})$ , as a weighted average of the NCC’s with three user nodding templates. We computed the NCC “shaking score” analogously. The highest NCC is empirically assigned the weight  $\alpha_i = 2$  while the other two have weights of 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 typing. 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.

**4.2.2 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  $n$  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  $n$  can be adjusted based on the user’s performance and preference.

**4.2.3 Nodding Intention.** A nodding intention is defined as a downward head movement. 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 value 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 is canceled. The velocity threshold is empirically set to 1, and  $n$  is set to 4. When a nodding gesture is classified, the nodding intention is also canceled.

### 4.3 Gaze Lock

Head and eye movements can interfere with each other. Measurements of gaze direction inevitably include a shift when the user nods for two reasons: (1) a 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 an earlier version of HGAze Typing, the gaze shift during a nodding gesture that we observed 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 is triggered. The system stops 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.

### 4.4 Gaze Path Restoring

The system releases the gaze lock after a nod is detected, and use the gaze fixation inputs to generate 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:

$$\text{Fixation}_{\text{new}}(t) = \frac{t - t_0}{T} \text{Fixation}(t) + \left(1 - \frac{t - t_0}{T}\right) \text{KeyCenter} \quad (2)$$

for  $t \in [t_0, t_0 + T]$ . Here Fixation, Fixation<sub>new</sub>, and KeyCenter are represented in 2D screen coordinates.

The above method restores the first part of the gaze path to the user-desired path and alleviates 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.

#### 4.5 Candidate Selection

HGaze Typing uses gaze paths to compute candidate words. The nodding gestures specify the first and last letters of a word and segment the gaze path from the continuous gaze inputs. The system uses the first and last letters to filter the lexicon when choosing the candidate words.

HGaze Typing computes a probability for the gaze path using the Fréchet distance [Fréchet 1906], which is widely used in gesture and input pattern recognition [Despinoy et al. 2016; Sriraghavendra et al. 2007; Zhao et al. 2013]. The final probability of each word candidate is obtained by applying Bayes' theorem. Only words in the lexicon with the given first and last letters are considered as word candidates. We use the Fréchet score as a representation of the posterior distribution of the words in the lexicon.

The prior discrete distribution of the words in the lexicon  $\mathcal{L}$  is a simple word frequency count normalized by the total number of words in a text corpus. Word frequency data was extracted from the Corpus of Contemporary American English [Davies 2008]. The posterior distribution of the word candidates is given by:

$$\Pr(W = w_i | G = g) = \frac{\Pr(G = g | W = w_i) \Pr(W = w_i)}{\Pr(G = g)} \quad (3)$$

where  $W$  is a random variable that represents the word entered by the user,  $G$  is a 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:

$$\Pr(G = g | W = w_i) = \frac{S(g, w_i)}{\sum_{w_j \in \mathcal{L}} S(g, w_j)} \quad (4)$$

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

$$S(g, w_i) = \frac{|\text{Ideal}(w_i)|}{1 + \text{DFD}^k(g, \text{Ideal}(w_i))}. \quad (5)$$

Here  $\text{DFD}(\cdot, \cdot)$  represents the discrete Fréchet distance,  $\text{Ideal}(\cdot)$  is the sequence of key center coordinates of the characters that form the word, and  $k$  is a positive scaling constant, empirically set to 6.6. The gaze path score is proportional to  $|\text{Ideal}(w_i)|$  to favor longer words.

## 5 EXPERIMENT

We conducted an experiment spanning nine sessions to evaluate the usability of the HGaze Typing interface. In this experiment, we used a dwell-time keyboard as a baseline text entry method, following previous research [Hansen et al. 2004; Kurauchi et al. 2016; Rough et al. 2014; Wobbrock et al. 2008].

### 5.1 Participants

Eleven university students without physical impairments were recruited for the experiment. One participant met significant calibration issues with the eye tracker due to glasses reflections 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 English speakers and proficient in using a QWERTY keyboard. Three 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.

### 5.2 Apparatus

We conducted the experiment on a laptop (3.70 GHz CPU, 16 GB RAM) running Windows 10, connected to a 19-inch LCD monitor (1280 × 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 1, and the dwell-time keyboard used the same layout with an additional space key below the virtual letter keys. The lexicon was the union of Kaufman's lexicon [Kaufman 2015] and the words in MacKenzie and Soukoreff's phrase dataset [2003]. The dwell period was set to be 600 ms, following Hansen et al. [2003].

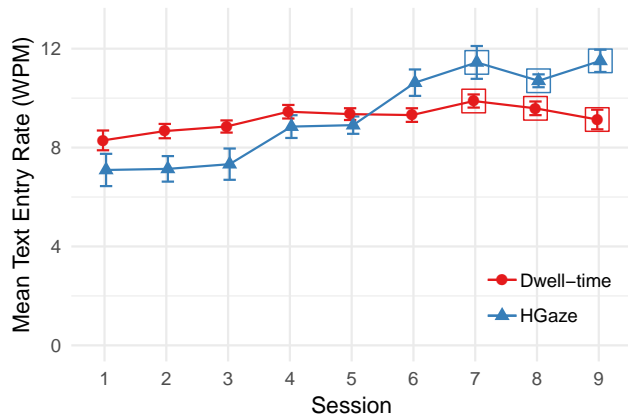
### 5.3 Procedure

Each participant visited the lab on three different days (12–48 hours apart) and performed 3 sessions of typing with each interface per day, resulting in 9 text entry sessions for each interface. The order of the interfaces was counterbalanced using a Latin square. Participants spent 40 to 75 minutes on each day. 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.

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 sit 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 by the HGaze Typing system as templates. 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 the dataset by MacKenzie and Soukoreff [2003]. For each typing trial, both typing 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





**Figure 3: 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.**

completed a short questionnaire on their demographics and their subjective feedback about the two text entry methods.

## 6 RESULTS

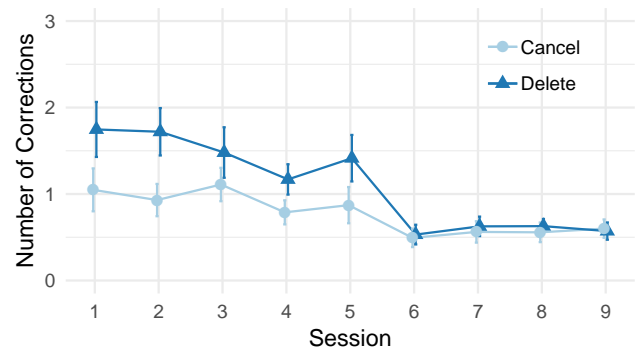
In this section, we analyze the results and summarize the subjective feedback of the two text entry interfaces.

### 6.1 Text Entry Rate

The text entry rate was measured in words per minute. The average length of a word was defined as 5 characters (excluding space). Overall, the mean text entry rate using HGaze Typing was higher than using the dwell-time keyboard (Figure 3). 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.

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 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 the HGaze Typing and dwell-time method in terms of typing speed.

The average maximum text entry rate for each session and participant with HGaze Typing was 16.21 wpm and 10.85 wpm with the dwell-time keyboard. Interface, session, and interaction between session and interface had a significant effect on the maximum typing speed (interface:  $F_{1,9} = 30.93$ ,  $p < 0.001$ ; session:  $F_{8,72} = 10.04$ ,  $p < 0.001$ ; session  $\times$  interface:  $F_{8,72} = 5.22$ ,  $p < 0.001$ ). One participant achieved 23 wpm using HGaze Typing. All participants achieved a text entry rate of at least 15 wpm using HGaze Typing and only 11 wpm using the dwell-time keyboard.



**Figure 4: Mean and standard error of the number of deletes and cancels per sentence in each session with HGaze Typing.**

### 6.2 Accuracy

The average Minimum String Distance (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 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.

The mean number of deletes and cancels per sentence in each session when typing with HGaze Typing interface is shown in Figure 4. 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 1.05 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.

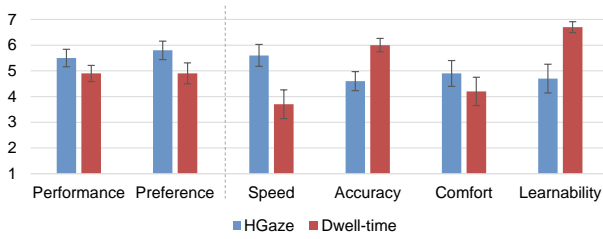
### 6.3 Subjective Feedback

The post-experiment questionnaire asked participants to rate both interfaces on a 7-point Likert scale of their performance and preference (Figure 5). Participants believed HGaze Typing had better performance (5.5) than the dwell-time keyboard (4.9). They also indicated a preference for the HGaze Typing interface over the dwell-time method (5.8 vs. 4.9).

Speed, accuracy, comfort, and learnability were evaluated by participants (Figure 5). 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. The participants also evaluated eye-control and head-control efforts. Participants reported the same eye-control effort using both interfaces (4.6). HGaze Typing required a higher head-control effort of the participants (5.8) than the dwell-time keyboard (2.9) but no participant reported neck fatigue using 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). As for the accuracy, which was not perceived to





**Figure 5: The average ratings of overall performance and preference, and perceived speed, accuracy, comfort and learnability, on a 7-point Likert scale.**

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). 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). 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.”

## 7 DISCUSSION AND CONCLUSIONS

We designed and implemented HGaze Typing, a text entry interface combining head gestures and gaze paths. After five experimental sessions, HGaze showed better efficacy than a dwell-time keyboard. Subjective feedback showed that HGaze Typing had better 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.

HGaze Typing achieved a text entry rate comparable to gaze-path-based interfaces (e.g., EyeSwipe), and is considerably more efficient than head-based text entry. Comparing to interfaces that use gaze input only, HGaze Typing uses head movement as an additional input to perform commands, which allows 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 without moving his or her gaze outside the virtual 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 visual 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. By utilizing additional head gestures in the gaze-based text entry systems, we can facilitate additional text entry and editing tasks.

Another advantage of using head gestures is that it can reduce the virtual keyboard area. By using shaking gestures, the cancel/deletion button for gaze-based selection is no longer needed. Additionally, a word candidate list is necessary for gaze-path-based text entry interfaces 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 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. 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 provides inaccurate gaze position measurements during a fast head movement, and (2) the user’s gaze is not stable when performing head gestures. To handle the gaze shift during a nodding gesture, in HGaze Typing, we designed the gaze lock state and the gaze path restoring algorithm. It should be noted that this solution is specifically designed for simple selection tasks and gaze-path-based text entry, and cannot recover the gaze position to a precision that may be required by other applications (e.g., gaming or design). When designing other multi-modal interfaces in the future, designers should consider the potential interference between different input channels.

Besides hands-free text entry and accessibility applications, the proposed method can be extended to other devices. For example, head motion and gaze tracking sensors are prevalent on virtual reality and augmented reality (VR/AR) devices. When using these devices in public, the HGaze Typing interface can be an alternative to manual and voice input.

The current implementation of HGaze Typing system requires calibration for head inputs. In the future, we plan to use machine learning algorithms to perform head gesture classification and allow calibration-free text entry commands. Another limitation of this work is the lack of usability evaluation for people with motor impairments. People with motor impairments can have different gaze and head movement patterns, so that a longitudinal user studies with people with different mobility levels can provide insights into head gesture design in HGaze Typing.

## REFERENCES

- R. Bates and H.O. Istance. 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 (01 Oct 2003), 280–290. <https://doi.org/10.1007/s10209-003-0053-y>
- Margrit Betke, James Gips, and Peter Fleming. 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 (2002), 1–10. <https://doi.org/10.1109/TNSRE.2002.1021581>
- Mark Davies. 2008. The Corpus of Contemporary American English (COCA): 520 million words, 1990–present. <https://corpus.byu.edu/coca/>. Accessed: 2017-11-22.
- Fabien Despinoy, David Bouget, Germain Forestier, Cédric Penet, Nabil Zemiti, Philippe Poignet, and Pierre Jannin. 2016. Unsupervised trajectory segmentation for surgical gesture recognition in robotic training. *IEEE Transactions on Biomedical Engineering* 63, 6 (2016), 1280–1291.

- Antonio Diaz-Tula, Filipe M. S. de Campos, and Carlos H. Morimoto. 2012. Dynamic Context Switching for Gaze Based Interaction. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) (ETRA '12). ACM, New York, NY, USA, 353–356. <https://doi.org/10.1145/2168556.2168635>
- Antonio Diaz-Tula and Carlos H. Morimoto. 2016. AugKey: Increasing Foveal Throughput in Eye Typing with Augmented Keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). ACM, New York, NY, USA, 3533–3544. <https://doi.org/10.1145/2858036.2858517>
- Wenxin Feng, Mehrnoosh Sameki, and Margrit Betke. 2018. Exploration of Assistive Technologies Used by People with Quadriplegia Caused by Degenerative Neurological Diseases. *International Journal of Human-Computer Interaction* 34, 9 (2018), 834–844. <https://doi.org/10.1080/10447318.2017.1395572> arXiv:<https://doi.org/10.1080/10447318.2017.1395572>
- Maurice Fréchet. 1906. *Sur quelques points de calcul fonctionnel*. Ph.D. Dissertation. Rendiconti del Circolo Matematico di Palermo. 22:1–74.
- Yulia Gizatdinova, Oleg Špakov, and Veikko Surakka. 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* (Capri Island, Italy) (AVI '12). ACM, New York, NY, USA, 132–139. <https://doi.org/10.1145/2254556.2254582>
- Yulia Gizatdinova, Oleg Špakov, Outi Tuisku, Matthew Turk, and Veikko Surakka. 2018. Gaze and Head Pointing for Hands-free Text Entry: Applicability to Ultra-small Virtual Keyboards. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications* (Warsaw, Poland) (ETRA '18). ACM, New York, NY, USA, Article 14, 9 pages. <https://doi.org/10.1145/3204493.3204539>
- K. Grauman, M. Betke, J. Lombardi, J. Gips, and G.R. Bradski. 2003. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society* 2, 4 (01 Nov 2003), 359–373. <https://doi.org/10.1007/s10209-003-0062-x>
- John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Itoh, and Satoru Mashino. 2003. Command Without a Click: Dwell Time Typing by Mouse and Gaze Selections. In *Human-Computer Interaction (INTERACT '03)*, M. Rauterberg et al. (Ed.), IOS Press, 121–128.
- John Paulin Hansen, Kristian Tørning, Anders Sewerin Johansen, Kenji Itoh, and Hirotaka Aoki. 2004. Gaze Typing Compared with Input by Head and Hand. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications* (San Antonio, Texas) (ETRA '04). ACM, New York, NY, USA, 131–138. <https://doi.org/10.1145/968363.968389>
- Anke Huckauf and Mario Urbina. 2007. Gazing with pEYE: New Concepts in Eye Typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization* (Tubingen, Germany) (APGV '07). ACM, New York, NY, USA, 141–141. <https://doi.org/10.1145/1272582.1272618>
- Poika Isokoski. 2000. Text Input Methods for Eye Trackers Using Off-screen Targets. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications* (Palm Beach Gardens, Florida, USA) (ETRA '00). ACM, New York, NY, USA, 15–21. <https://doi.org/10.1145/355017.355020>
- Robert J. K. Jacob. 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* (Seattle, Washington, USA) (CHI '90). ACM, New York, NY, USA, 11–18. <https://doi.org/10.1145/97243.97246>
- Josh Kaufman. 2015. Google 10000 English. <https://github.com/first20hours/google-10000-english>. Accessed: 2015-09-22.
- Luka Krapic, Kristijan Lenac, and Sandi Ljubic. 2015. Integrating Blink Click interaction into a head tracking system: Implementation and usability issues. *Universal Access in the Information Society* 14, 2 (2015), 247–264. <https://doi.org/10.1007/s10209-013-0343-y>
- Per Ola Kristensson and Keith Vertanen. 2012. The Potential of Dwell-free Eye-typing for Fast Assistive Gaze Communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) (ETRA '12). ACM, New York, NY, USA, 241–244. <https://doi.org/10.1145/2168556.2168605>
- Per-Ola Kristensson and Shumin Zhai. 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* (Santa Fe, NM, USA) (UIST '04). ACM, New York, NY, USA, 43–52. <https://doi.org/10.1145/1029632.1029640>
- Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) (CHI '16). ACM, New York, NY, USA, 1952–1956. <https://doi.org/10.1145/2858036.2858335>
- Andrew Kurauchi, Wenxin Feng, Carlos Morimoto, and Margrit Betke. 2015. HMAGIC: Head Movement and Gaze Input Cascaded Pointing. In *Proceedings of the 8th ACM International Conference on Pervasive Technologies Related to Assistive Environments* (Corfu, Greece) (PETRA '15). ACM, New York, NY, USA, Article 47, 4 pages. <https://doi.org/10.1145/2769493.2769550>
- Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 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* (Montreal, QC, Canada) (CHI '18). ACM, New York, NY, USA, Article 81, 14 pages. <https://doi.org/10.1145/3173574.3173655>
- J Lombardi and M Betke. 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*, 199–200.
- I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems* ( Ft. Lauderdale, Florida, USA) (CHI EA '03). ACM, New York, NY, USA, 754–755.
- Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) (CHI '09). ACM, New York, NY, USA, 357–360. <https://doi.org/10.1145/1518701.1518758>
- Diako Mardanbegi, Dan Witzner Hansen, and Thomas Pederson. 2012. Eye-Based Head Gestures. In *Proceedings of ETRA'2012* (Santa Barbara, CA). ACM, New York, NY, USA, 139–146. <https://doi.org/10.1145/2168556.2168578>
- Eric Missimer and Margrit Betke. 2010. Blink and wink detection for mouse pointer control. In *Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments* (Samos, Greece) (PETRA '10). ACM, New York, NY, USA, Article 23, 8 pages. <https://doi.org/10.1145/1839294.1839322>
- Carlos H. Morimoto and Arnon Amir. 2010. Context Switching for Fast Key Selection in Text Entry Applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications* (Austin, Texas) (ETRA '10). ACM, New York, NY, USA, 271–274. <https://doi.org/10.1145/1743666.1743730>
- Martez E. Mott, Shane Williams, Jacob O. Wobbrock, and Meredith Ringel Morris. 2017. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (CHI '17). ACM, New York, NY, USA, 2558–2570. <https://doi.org/10.1145/3025453.3025517>
- Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filterypeding: Design Challenges and User Performance of Dwell-Free Eye Typing. *ACM Transactions on Accessible Computing* 6, 1, Article 3 (March 2015), 37 pages. <https://doi.org/10.1145/2724728>
- Marco Porta and Matteo Turina. 2008. Eye-S: A Full-screen Input Modality for Pure Eye-based Communication. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia) (ETRA '08). ACM, New York, NY, USA, 27–34. <https://doi.org/10.1145/1344471.1344477>
- Kari-Jouko Räihä and Salla Ovasa. 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* (Austin, Texas, USA) (CHI '12). ACM, New York, NY, USA, 3001–3010. <https://doi.org/10.1145/2207676.2208711>
- Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An Evaluation of Dasher with a High-performance Language Model As a Gaze Communication Method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces* (Como, Italy) (AVI '14). ACM, New York, NY, USA, 169–176. <https://doi.org/10.1145/2598153.2598157>
- Sayan Sarcar, Prateek Panwar, and Tuhin Chakraborty. 2013. EyeK: An Efficient Dwell-free Eye Gaze-based Text Entry System. In *Proceedings of the 11th Asia Pacific Conference on Computer Human Interaction* (Bangalore, India) (APCHI '13). ACM, New York, NY, USA, 215–220. <https://doi.org/10.1145/2525194.2525288>
- Ludwig Sidenmark, Diako Mardanbegi, Argenis Ramirez Gomez, Christopher Clarke, and Hans Gellersen. 2020. BimodalGaze: Seamlessly Refined Pointing with Gaze and Filtered Gestural Head Movement (ETRA '20 Full Papers). Association for Computing Machinery, New York, NY, USA, Article 8, 9 pages. <https://doi.org/10.1145/3379155.3391312>
- E. Sriraghavendra, K. Karthik, and Chiranjib Bhattacharyya. 2007. Fréchet distance based approach for searching online handwritten documents. In *Ninth International Conference on Document Analysis and Recognition (ICDAR)*, Vol. 1. IEEE, 461–465.
- Keith Trnka, John McCaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2009. User interaction with word prediction: The effects of prediction quality. *ACM Transactions on Accessible Computing* 1, 3, Article 17 (Feb. 2009), 34 pages. <https://doi.org/10.1145/1497302.1497307>
- Oleg Špakov, Poika Isokoski, and Päivi Majaranta. 2014. Look and Lean: Accurate Head-assisted Eye Pointing. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Safety Harbor, Florida) (ETRA '14). ACM, New York, NY, USA, 35–42. <https://doi.org/10.1145/2578153.2578157>
- Oleg Špakov and Päivi Majaranta. 2012. Enhanced Gaze Interaction Using Simple Head Gestures. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing* (Pittsburgh, Pennsylvania) (UbiComp '12). ACM, New York, NY, USA, 705–710. <https://doi.org/10.1145/2370216.2370369>
- David J Ward and David JC MacKay. 2002. Fast hands-free writing by gaze direction. *Nature* 418 (22 Aug 2002), 838. <https://doi.org/10.1038/418838a>
- Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. 2008. Longitudinal Evaluation of Discrete Consecutive Gaze Gestures for Text Entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (Savannah, Georgia) (ETRA '08). ACM, New York, NY, USA, 11–18. <https://doi.org/10.1145/1344471.1344475>

- Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand Writing on Stylus Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) (*CHI '03*). ACM, New York, NY, USA, 97–104. <https://doi.org/10.1145/642611.642630>
- Shumin Zhai and Per Ola Kristensson. 2012. The Word-gesture Keyboard: Reimagining Keyboard Interaction. *Commun. ACM* 55, 9 (Sept. 2012), 91–101.
- Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-Based Gaze Gesture Communication for People with Motor Disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) (*CHI '17*). ACM, New York, NY, USA, 2878–2889. <https://doi.org/10.1145/3025453.3025790>
- Xi Zhao, Tao Feng, and Weidong Shi. 2013. Continuous mobile authentication using a novel graphic touch gesture feature. In *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, 1–6. <https://doi.org/10.1109/BTAS.2013.6712747>