Hummer: Text Entry by Gaze and Hum

Ramin Hedeshy
Universität Stuttgart
Stuttgart, Germany
Ramin.Hedeshy@ipvs.uni-stuttgart.de

Chandan Kumar
Universität Stuttgart
Stuttgart, Germany
Chandan.Kumar@ipvs.uni-stuttgart.de

Raphael Menges
Universität Koblenz-Landau
Koblenz, Germany
raphaelmenges@uni-koblenz.de

Steffen Staab
Universität Stuttgart
Stuttgart, Germany
University of Southampton, UK
s.r.staab@soton.ac.uk

ABSTRACT

Text entry by gaze is a useful means of hands-free interaction that is applicable in settings where dictation suffers from poor voice recognition or where spoken words and sentences jeopardize privacy or confidentiality. However, text entry by gaze still shows inferior performance and it quickly exhausts its users. We introduce text entry by gaze and hum as a novel hands-free text entry. We review related literature to converge to word-level text entry by analysis of gaze paths that are temporally constrained by humming. We develop and evaluate two design choices: "HumHum" and "Hummer." The first method requires short hums to indicate the start and end of a word. The second method interprets one continuous humming as an indication of the start and end of a word. In an experiment with 12 participants, Hummer achieved a commendable text entry rate of 20.45 words per minute, and outperformed HumHum and the gaze-only method EyeSwipe in both quantitative and qualitative measures.

CCS CONCEPTS

• **Human-centered computing** → **Text input**; *Accessibility technologies*; *Interaction design theory, concepts and paradigms*; Interaction devices.

KEYWORDS

eye typing, hands-free interaction, eye tracking, humming, swipe

ACM Reference Format:
Ramin Hedeshy, Chandan Kumar, Raphael Menges, and Steffen Staab. 2021. Hummer: Text Entry by Gaze and Hum. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3411764.3445501

CHI '21, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445501

1 INTRODUCTION

A core aspect of human-computer interaction is the entry of text, which can be used to query a search engine, to issue a command to the system, or to compose a text message. The predominant method of text entry is manual typing on physical or virtual keyboards. However, people who have motor impairments or whose hands are busy with other tasks require hands-free means of text entry. Therefore, eye gaze [27] and voice input [56] have been studied as viable alternatives.

Initial methods for gaze-based text entry let users *enter characters one by one*. A character is selected on a virtual on-screen keyboard either by dwelling on a key [28, 33] or by specific eye gestures [10, 55]. More recent methods let users *enter complete words* to reduce delays between keystrokes. These methods use fixation sequences [35] or the shape of the gaze path across the keyboard [22]. They have been shown to improve text entry rates. However, these methods still require explicit gaze gestures for selection which slow down the interaction and tiring users' eyes. It has been argued that adding a second modality can make gaze-based text entry appear to be a more natural and faster means of human-computer interaction. For instance, TAGSwipe [21] lets users gaze on virtual keys, but the users issue the actual selection of a word by the press of a button on the touchscreen. The disadvantage of TAGSwipe is that the process of text entry no longer remains a hands-free interaction.

Speech-based text entry constitutes an alternative that is often used for text entry on smartphones and smart home applications. Many people with physical disabilities, however, also suffer from speech impairments due to conditions such as dysarthria [40]. Furthermore, the accuracy and robustness of speech recognition depend on the mother tongue and accent of the user as well as on interfering noises from the background. Even if these issues can be overcome, text entry by voice might reveal words or sentences to — possibly inadvertent — eavesdroppers conflicting with concerns of confidentiality or privacy.

Another alternative, *non-verbal voice interaction* (NVVI), involves humming or whistling [17]. Humming reveals less semantic content to third parties, and it is even possible with a closed mouth [48]. Particularly, humming is a universal habit and is commonly used for communication between humans, even across cultural borders [15]. However, its application to human-computer interaction, and more specifically to text entry, has been rare so far. Humming has been used as unimodal input for character-level text

entry [39, 46], where distinct humming patterns were used to select from n-grams and characters. This rendered the interaction erroneous, slow, cognitively demanding, and tiring for users.

We propose a spatio-temporal integration of gaze and hum for an effective hands-free bimodal approach to text entry. Considering our review of related work up to here, let us briefly review design choices for such an objective. To enter single characters, one might replace dwelling with a hum to select a key. Analogous to the case of character by character text entry by gaze alone, the sequential focusing (by gaze) and selection (by hum) may compound individual delays. Therefore, we evolve this approach similar to the gaze-only research and propose and compare two text entry methods that integrate gaze and hum to select complete words instead of single letters, i.e., word-level text entry. In both methods, a user skims the keys that compose the characters of a word in the corresponding order. In the first method, which we call "HumHum," the user issues a short hum both at the start and at the end of each word. In the second method, the user hums continually while skimming the keys; we call it "Hummer."

We have conducted a study with 12 participants to assess humming methods efficacy compared to EyeSwipe, which constitutes the state-of-art in gaze-based word-level text entry. The results exhibit superior performance with Hummer (15.48 words per minute, WPM) compared to HumHum (12.55 WPM) and EyeSwipe (10.35 WPM). Most importantly, the participants achieved a commendable text entry rate of 20.45 WPM after four sessions with Hummer. Furthermore, the qualitative responses and explicit feedback show a clear subjective preference for Hummer as being fast, comfortable, and easy to learn. Study participants liked the playful experience of continuous humming providing a rhythmic, singing-like, and overall enjoyable feeling while swiping with their gaze over the virtual keyboard to compose words.

To the best of our knowledge, this is the first research to combine eye tracking with humming for interaction. We believe the combination of gaze and hum will open doors for several useful HCI applications, as gaze with its natural spatial orientation, and humming with its temporal characteristic can be unified to enable efficient hands-free human-computer interaction in scenarios such as information browsing, gaming, and virtual or augmented reality.

2 RELATED WORK

Our proposed methods combine two modalities for hands-free human-computer interaction, gaze and hum. Therefore, we discuss research related to eye tracking and non-verbal voice interaction.

2.1 Eye Tracking for Hands-free Text Entry

Researchers have evaluated the eye gaze of users as a human-computer interaction method for several decades [12, 53]. Usually, a remote eye-tracking system encompassing a camera and near-infrared illumination is employed to track the eyes and pupils of a user. Estimated gaze can be exploited as a signal that is interpreted as an input modality in several HCI domains [29], such as gaming [2], Web browsing [30], or text entry [27]. We survey gaze-based text entry methods in the following, distinguishing between approaches that allow users to either enter single characters or complete words.

2.1.1 Character-level Text Entry Methods. Selecting characters one by one on a virtual keyboard is a well-established gaze-based text entry approach. The most common method makes users fixate a key for longer than a threshold duration – called *dwell time* [27, 44] – in order to select the corresponding character. Higher thresholds slow the text entry process, while lower thresholds increase the error rate. There are approaches to automatically adjust the dwell time according to the user's typing rhythm [28] or based on language model probability to adjust the dwell time of the most likely next character [33]. Alternative methods replace dwell time by eye gestures to indicate the selection of individual characters. EyeK [43] is a method where the user moves her gaze in an in-out-in fashion to select a key. The pEYEWrite keyboard [10, 51] consists of an expandable pie menu with the grouping of characters and each character is selected by looking through the borders of the pie sections. In a context-switching method [31, 32], the keyboard layout is duplicated in two separate regions and the last fixated character is selected by an eye gesture crossing from the current region to the other region. In Dasher [50] a character is selected by fixating on a dynamically-sized key until the key crosses a boundary point in a zooming interface. There are also multimodal approaches that suggest replacing the dwell time by an additional modality like touch [37], mouse [9], or tooth-clicks [58].

2.1.2 Word-level Text Entry Methods. Kristensson and Vertanen [19] show the potential performance gain of word-level entry on virtual keyboards, where users just skim over characters contained in a word and no minimal dwell time is required. Pedrosa et al. [35] introduce Filteryedping, where a user looks at all the characters that compose the word to be typed. The short key fixations produce a stream with characters that belong to the word, including several character clusters that do not belong to the word but are generated during eye movements over the keyboard. The stream of characters is filtered to generate a list of words based on a dictionary. The user can then select a word from that list to be entered. While Liu et al. [25] further improve on the idea of filtering character streams, these approaches cannot recover from errors when the user fails to gaze over one or two characters of the intended word.

More recent gaze-path based text entry methods have been inspired by manual or pen-based swiping on virtual keyboards [20, 57]. These methods map gaze paths to words in a lexicon and do not rely on perfect recall of character occurrences. In EyeSwipe [22], users select the first and last character of the word using an eye gesture. The user who fixates a key is shown a popup button above the fixated key. By fixating the button before returning the fixation to the targeted key, the user executes this gesture, which is called "reverse-crossing." Having marked the start key by reverse-crossing, the user swipes his gaze over the keys corresponding to the middle characters until she reaches the end key, which she also indicates by reverse-crossing. Candidate target words are selected from an *n*-best list constructed from the gaze path between the reverse-crossings. Practicing EyeSwipe markedly improves users' text entry rates, though using the reverse-crossing mechanism is tedious and the dynamic pop-up button is a source of fatigue and confusion, aggravated by irreducible eye-tracking inaccuracies. Furthermore, for one- or two-letter words, multiple reverse-crossing selections lead to an interaction overload.

TAGSwipe [21] adapts gaze-path methods to bi-modal interaction. It combines fast gaze swiping over the keys corresponding to the intended character sequence with a button or touch interface that, through being pressed and later released, indicates the start- and endpoints of the gaze path. Thus, it eliminates the need for eye gestures, but its application is limited to users who can perform at least coarse interaction with their hands.

2.2 Non-verbal Voice Interaction

Non-verbal voice interaction (NVVI) refers to a set of techniques that exploit sounds other than speech, such as humming, whistling, blowing, or hissing, as their modality of interaction.

2.2.1 Applications of NVVI. Usually, the sound signals (pitch, volume, timbre, etc.) are measured over time to interpret them as input signals [38]. One of the most common NVVI applications uses humming as a means to query a musical database. Multiple works [14, 18, 34] compare the humming rhythm to songs in a database using sub-sequence matching methods. Watts and Robinson [54] propose a system where whistling triggered OS UNIX commands. Igarashi and Hughes [11] use non-verbal sounds to specify numeric parameters, e. g., a user can utter *'aaah'* and the system increases the volume as long as the user emits *'aaah.'* Non-verbal input (blowing) can be used for clicking [59]. Bilmes et al. [1] use non-verbal input to emulate joystick device controls for uses as an assistive tool. Peixoto et al. [36] demonstrate an application of humming to control a wheelchair, where they have measured humming as vibration on the neck. NVVI has also been employed as an input technique in a Tetris game [47]. More recently, Funk et al. [8] have used non-verbal input for interacting with smart assistants while driving. They evaluated binary input like clapping or snapping fingers against continuous input humming and found humming as the preferred option while driving.

2.2.2 Humming-based Text Entry Methods. Sporka et. al. [47] presented one of the first approaches of humming for text entry, where each humming pattern is assigned a specific key on the keyboard. When the user emits a sound, the corresponding character is entered. On mobile phones with just a dozen keys, QANTI [5] overloads each key with multiple characters to be able to offer all possible characters as input. CHANTI [46] transfers the principles underlying QANTI [5] to humming-based input. Humsher [39] combines humming as an input signal and with a language model. It uses a dynamic layout (like Dasher [50]) in which n-grams of characters are presented to the user to choose from, according to their probability in the given context. All these text entry methods were found to be slow and reported a text entry rate in the range of 10-20 characters per minute, i. e., approximately 2-4 words per minute.

All the above-mentioned NVVI approaches require classifiers to distinguish different humming patterns. Also, users need to remember and emit hums of appropriate length and pitch in order to activate the intended commands. In contrast, we propose to use gazing for spatially selecting which keys to consider for activation, humming for temporally selecting which keys to consider for activation, and the intersection of both to modalities to derive the appropriate activation of keys.

3 DESIGNING TWO TEXT ENTRY METHODS BASED ON EYE TRACKING AND HUMMING

We base our research on the results from the related work. Therefore, we skip designs that would require locating and selecting individual keys by gazing and humming, respectively, as we expect that they would exhibit the same disadvantages of compound delays observed in character-level gaze-only approaches to text entry. Rather, we design two methods based on hum and gaze, HumHum and Hummer, that both provide for word-level text entry. We base our design choices on the observations that users, (i), naturally use their eyes for looking and observing [13] and, (ii), naturally hum to voice agreement and confirmation [17]. In our design, we target to exploit such natural tendencies to, (i), let users gaze over the characters of the word that they intend to write and, (ii), let them confirm the start and end of a word entry activity.

Having constrained the design space in this way, the specific operation to signal the start and end of a word entry activity is generally open to at least two design options, exploiting humming as a point-based or an interval-based signal. The first option leads us to the design of HumHum, in which the start and end of the word entry activity are indicated by a short hum each. The second option leads us to the design of Hummer, in which the user keeps humming from the start to end of a complete word. Both design choices may come with advantages or disadvantages.

A high-level flow chart for both methods is depicted in Figure 1. The system comprises three modes: *idle*, *swipe*, and *candidate selection*. The system goes to the idle mode after startup. When the system detects humming while the gaze of the user is on a key, the system enters the swipe mode. In swipe mode, the system records the gaze path of the user. Interleaved computation continuously updates the list of candidate words which considers the most probable candidates given the current gaze path. When the *word-end-event* – which flags the end of the word entry activity – is caught, the swipe mode ends. In HumHum, the *word-end-event* is raised by a second short hum, assuming some clear delineation from the first hum that started the swipe mode. In Hummer, the *word-end-event* is raised when the – usually long – hum ends. Upon receiving the *word-end-event*, the swipe mode is exited and the list of top-ranked candidate words is output for further use in the user interface. At this point, the top-ranked candidate word is presented as the next word in the text entry box, while the other candidate words appear as further possible choices in the user interface. Then there is a non-deterministic choice that will be realized with the next hum. If the user gazes on a key, the system will enter idle mode again for choosing the next words. If the user gazes at one of the candidate words, the system will assume that the top-ranked candidate word has been mistakenly chosen by the candidate computation and requires correction. A short hum will then change the selected word in the text entry box. Special characters and backspace can also be selected by gazing at a special key and voicing a short hum in idle mode.

3.1 Interface Design

The interface comprises a virtual QWERTY-style keyboard and a text box for displaying the entered words. See Figure 2 for a screenshot with annotations. In Hummer, a word can be typed

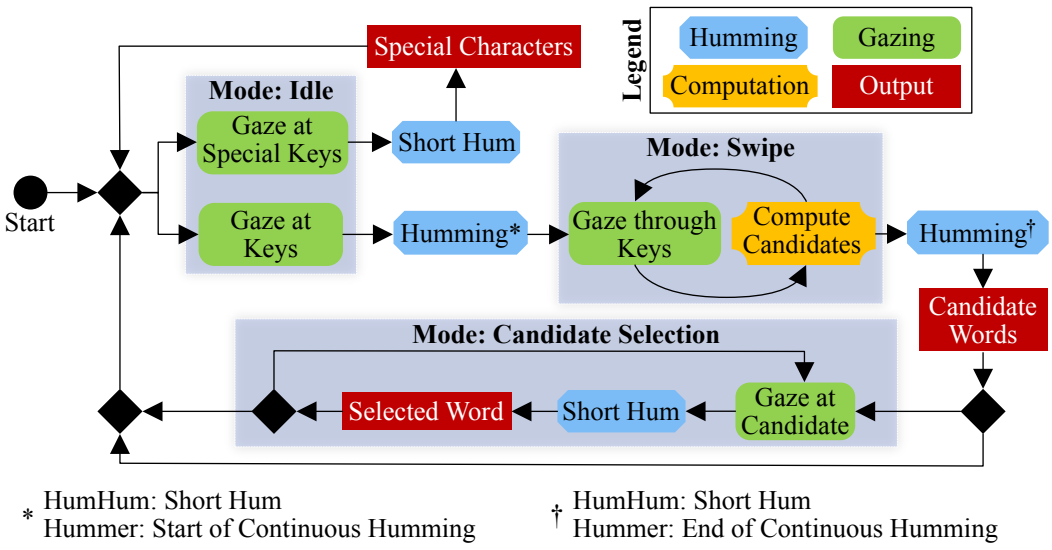

Figure 1: High-level flow chart for HumHum and Hummer text entry.

by completing the following actions: First, the user gazes at the first letter of the desired word and starts humming. The system gives visual feedback by changing the background color of the key currently under focus. Then, while humming, the user glances through the further letters of the word in the sequence the letters appear in the word. When gazing at the last letter, the user stops humming. To type, a word with HumHum user follows the same interaction sequence, with one exception: The user signals the start and end of the swipe mode with short hums instead of a continuous humming. The first and last letters are used by the system to filter the lexicon for potential candidates. The recorded gaze path is used to compute the candidate words. See the next section for details about the algorithm for the candidate selection. The algorithm is designed in a way that even if the gaze path misses intermediate letters, there is a chance that the system comes up with the intended word as candidate. The top candidate word appears on top of the key of the last letter. This provides instantaneous system feedback to the user about the current top candidate word while swiping. If the top candidate word differs from the word desired by the user, the user can check the list of alternative candidate words appearing below the text box. The list contains the six most probable candidate words sorted by their probability from the candidate selection algorithm from left to right.

3.2 Candidate Selection

We use the gaze path to compute candidate words. Following EyeSwipe [23], we initially filter the lexicon $\mathcal{L}$ for candidates w, with $w \in \mathcal{L}$, that confirm with the first and the last letter on the gaze path g.

The resulting word list is sorted according to a score based on the similarity of the user's gaze path to the word's ideal path, which is the sequence of the center coordinates of the keys. Repeated letters, such as "o" in "cool," are counted just once. So the ideal path of the word "cool" is the sequence of the center coordinates of the keys "C," "O," and "L."

For calculating the distance between a gaze path and an ideal path, the original EyeSwipe implementation uses *dynamic time warping* (DTW) [42], which has been widely used in gesture and input pattern recognition for comparing two time sequences [52]. However, studies suggest that the Fréchet distance [6] is a more appropriate solution for comparing sequences with widely varying sampling rates like gaze path [24, 49]. Therefore, we rank the differences between the given gaze path and the ideal path of each candidate word using a score that is based on the Fréchet distance between two polygonal curves. We define the ideal path $Ideal(w)$ of a word as the sequence of center coordinates of the letters on the virtual keyboard that form the word. We can interpret the gaze path and the ideal path of each candidate word as discrete polygonal curves [24]. In detail, we use the discrete Fréchet distance $DFD(a, b)$ by Eiter and Mannila [4], which computes the difference between two discrete polygonal curves a and b.

The score $S(g, w)$ for each candidate word is thus calculated as:

$$S(g, w) = \frac{1}{1 + DFD(g, Ideal(w))}$$

4 EVALUATION

It has been shown that word-level text entry is more efficient than character-level text entry for gaze-only methods [21, 22]. Before designing the experiment, we wanted to validate that combining gaze and hum for word-level entry in HumHum and Hummer is more effective than character-level entry, too. Hence, we conducted a pilot study with four participants (five sessions, no additional practice session) comparing Hummer and HumHum with character-level text entry: look and a short hum to select each letter. The results indicated that character-level text entry was 52% and 77% slower than HumHum and Hummer, respectively. Furthermore, participants

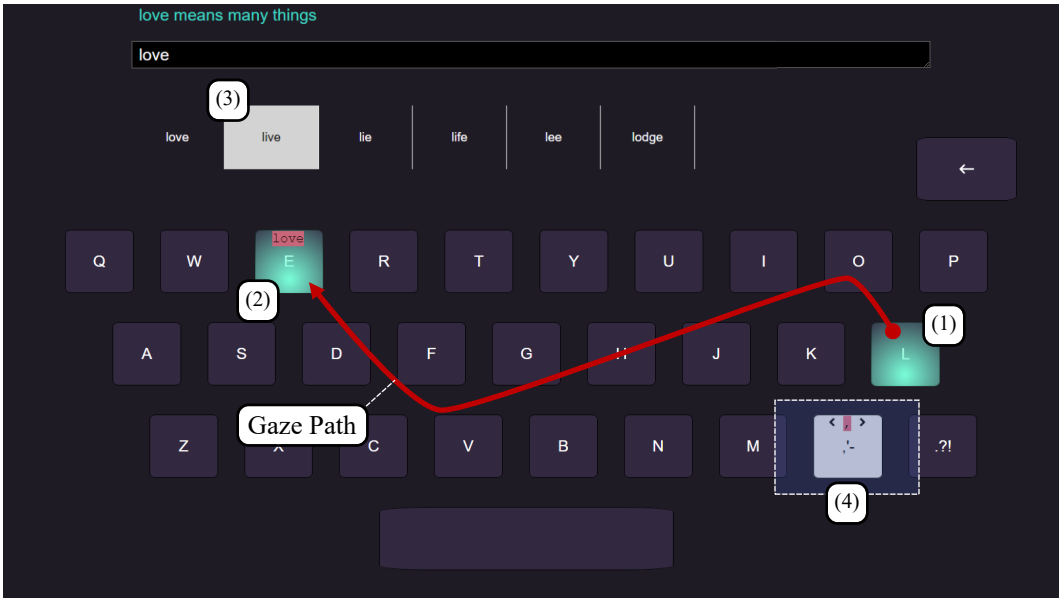

Figure 2: The interface of HumHum and Hummer. To type the word "love," the user fixates (1) on the first character 'L,' hums, and glances over the intermediate characters (red line). While the user glances over characters, the candidate word that would come out on top if this was the last letter is indicated on the key (cf. (2)). Reaching the character 'E,' the user may want to signal the ending of the word. In HumHum, the user issues a second hum and, in Hummer, the user simply stops humming to do so. The top-ranked candidate word is displayed in the text entry box. All candidate words are displayed below (3). Special characters and backspace can be selected by glancing at them and issuing a single short hum.

were frustrated and tired after using gaze and hum for character-level entry. This verified our assumption, and thus in the main experiment design, we compare Hummer and HumHum against a state-of-art word-level text entry as a baseline, i. e., EyeSwipe [22]. All the humming-based approaches so far (see Section 2.2) are character-level text entry methods incorporating an unfamiliar design and layout. Hence, they are not comparable to word-level text entry with a QWERTY layout used in our methods.

Three input methods were compared in our study:

(1) EyeSwipe (baseline) — the conventional word-level text entry method with gaze input [22].
(2) HumHum — short hum at the first and the last letter of the word.
(3) Hummer — continuous humming from first to the last letter of the word.

4.1 Participants

We recruited twelve participants (7 males and 5 females; aged 22 to 28, mean = 24.75, *SD* = 1.81). Ten participants were university students and two participants were employees. The vision was normal (uncorrected) for seven participants, while four participants wore glasses and one participant used contact lenses. Two participants had previously participated in studies with eye tracking, but these studies were not related to text entry. The other ten participants had never operated an eye tracker. All participants were familiar with the QWERTY layout (mean = 5.91, *SD* = 1.31, on the Likert scale from 1 = not familiar to 7 = very familiar) and were proficient in English (mean = 5.83, *SD* = 1.46 from 1 = very bad to 7 = very good) according to self-reported measures. The participants were paid 30 Euro for participating in the study. To motivate participants, we awarded the participant with the best performance – measured by both speed and accuracy of all three methods together – with an additional 50 Euro.

4.2 Apparatus

We conducted the experiment on a laptop (3.70 GHz CPU, 32 GB RAM) running Windows 10 on a 17" LCD monitor (1600 × 900 pixels). We used a RØDE NT-USB microphone to capture the humming. Gaze was recorded using a Tobii 4C eye tracker with a tracking frequency of 90 Hz. No chin rest was used. The eye tracker was placed at the lower edge of the screen. See Figure 3 for a picture of the setup. The eye-tracker tracking-box dimensions as reported by the manufacturer were 16" × 12" / 40 cm × 30 cm at a distance of the head of 29.5" / 75 cm.

The Hummer interface, HumHum, and the EyeSwipe were implemented in JavaScript, NodeJS,[1] using the Express framework.[2] All methods shared the same virtual keyboard layout and color theme. The EyeSwipe method was implemented using the techniques described by Kurauchi et al. [22]. We used the discrete Fréchet distance for word-gesture recognition in all the methods EyeSwipe, HumHum, and Hummer, as implemented in the Django REST framework.[3] The interface was rendered in the Google Chrome Web

[1] https://nodejs.org
[2] https://expressjs.com
[3] https://www.django-rest-framework.org

Figure 3: Experimental setup: A participant performing the experiment using Hummer on a laptop computer equipped with an eye tracker and an external microphone.

browser.[4] There was no gaze cursor displayed on the screen. The lexicon consisted of the union of Kaufman's lexicon [16] and the words from the MacKenzie and Soukoreff phrase set [26].

4.3 Procedure

The study was conducted in a university lab with no direct sunlight. During the experiment, the room noise level was approximately around 30-40 decibels (dB). Participants were able to operate the humming method at approximately 40-50 dB, which is comparable to whispering — a normal conversation is about 60 dB. This means the participants were able to hum very quietly and still activate the swipe mode. Each participant visited the lab for about three hours. Initially, each participant filled a questionnaire regarding their demographics, prior experience with eye tracking, and familiarity with the QWERTY layout. The eye tracker was calibrated for each participant before each method. For each method, participants were asked to first transcribe five phrases as a practice session to freely explore the interface, then transcribe in five sessions each five phrases, resulting in a total of 25 phrases per method. During the practice session of the Hummer method and the HumHum method, the threshold of the humming detection was individually adjusted for each participant. The participants were instructed to type as fast and accurately as they can. To minimize learning or fatigue bias, a minimum of ten minutes break was given to the participants between the three methods. The participants could optionally take a short break between the sessions. The phrases were randomly chosen from the phrase set and shown above the text box in each interface. After transcribing a phrase, the participants pressed the physical "Space" key to go to the next phrase to enter. Time measures began when the participant enters the swipe mode for the first time for each phrase.

[4]https://www.google.com/chrome

After completing all the five sessions of one method, participants completed a questionnaire to provide subjective feedback. The questions were crafted as seven-point Likert scales in which we asked the participants to rate each method in terms of speed, accuracy, comfort, learnability, and overall preference. We have also asked for their opinion and their suggestion on how to improve each method.

We randomized the test order of the three methods for the different participants using the Latin Square technique. This avoided potential cross-learning effects between the three methods. Thus, "Group" is a between-subjects independent variable with three levels. Four participants were assigned to each group. The total number of trials is 900 (= 12 participants × 3 input methods × 5 sessions × 5 phrases). The experiment itself is a 3 × 5 within-subjects design with the following independent variables and levels:

- Method (EyeSwipe, HumHum, Hummer)
- Sessions (1, 2, 3, 4, 5)

The variable "Session" is included to capture the participants' improvement with practice. The dependent variables are the following. The entry rate, measured as WPM. The error rate, measured using the *minimum string distance* (MSD) [45] in percentage. The correction rate, and the selection accuracy of the first and last letters of the entered words.

4.4 Results

The results are provided organized per dependent variable. For all the dependent variables, the group effect on entry speed is not statistically significant ($F_{(2,9)} = 0.209, ns$), indicating that counterbalancing had the desired result of offsetting order effects between the three methods.

4.4.1 Entry Rate. The entry rate is measured in WPM, while the length of a word is defined as five characters for normalization purposes. Overall, the mean entry rate using Hummer is higher than using the two other methods. Figure 4 illustrates the mean entry rate and its standard deviation for each session and method. The mean entry rate of EyeSwipe over all the sessions is 10.35 WPM. In contrast, using HumHum and Hummer, participants achieved on average 12.55 WPM, and 15.48 WPM, respectively.

We have conducted a within-subjects repeated measures ANOVA on the entry rate with the independent variables "Method" (EyeSwipe, HumHum, and Hummer) and "Session" (1-5). There are significant main effects of "Method" with ($F_{2,18)} = 19.862, p < .0001$) and "Session" with ($F_{(4,36)} = 33.431, p < .0001$).

There is a notable effect of the variable "Session" on the entry rate, which indicates a learning effect for all three methods. The mean entry rate with Hummer increases from 11.70 WPM in the first session to 20.45 WPM in the last session. Using the HumHum method, the typing speed increases from 10.35 WPM in the first session to 16.71 WPM in the last session. The mean entry rate with EyeSwipe keyboard also increases from 8.60 WPM in the first session to 12.03 WPM in the last session.

Hummer remains noticeably faster than other methods across all sessions. Every participant reached an average entry rate of at least 16 WPM using Hummer. The highest mean entry rate during a session achieved by participants was 30.64 WPM in the fifth session using Hummer. The highest mean entry rate with HumHum was

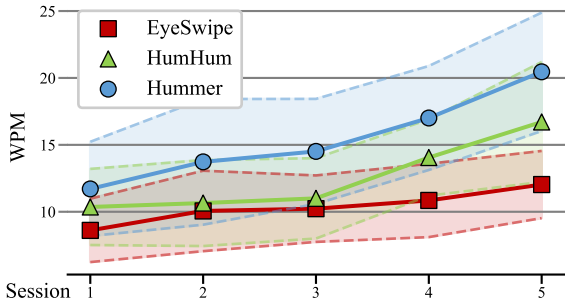

Figure 4: Mean entry rate (WPM) for each method and session, plotted as lines. Error bars indicate the standard deviation, plotted as areas with dotted lines as border.

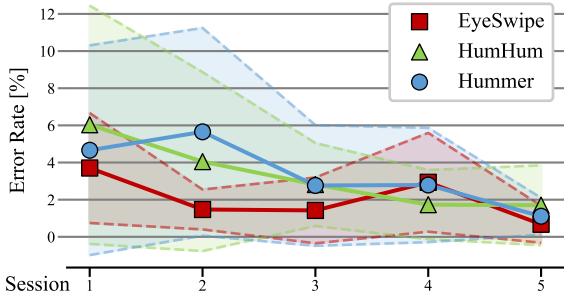

Figure 5: Error rate in percentage by method and session, plotted as lines. Error margins indicate the standard deviation, plotted as areas with dotted lines as border.

23.22 WPM also in the last session and the highest mean entry rate using EyeSwipe was 14.92 WPM achieved in the fourth session.

4.4.2 Error Rate. The error rates for the three methods over the five sessions are shown in Figure 5. The average MSD error rate is 2.04% for EyeSwipe and 3.39% and 3.26% for Hummer and HumHum, respectively. There is an evident reduction of errors over the five sessions, as the effect of "Session" on error rate is statistically significant with ($F_{(4,36)} = 4.541, p < .005$). This indicates that participants made less uncorrected errors with practice. The effect of method on error rate was significant (F(2, 18) = 4.516, p < .05). However, the difference was noteworthy only in the initial sessions, in the last three sessions error rate remains below 3% for all methods. There is no significant "Method × Session" effect with ($F_{8,72} = 1.510, p > .05$).

4.4.3 Correction Rate and Selection Accuracy. The correction rate reflects the number of times words have been deleted [23]. Situations in which this occurred include either an incorrect last letter or no candidate words matching the gaze path. The *correction rate* is calculated for each phrase as the number of deleted words divided by the number of entered words. The average correction rate is 16.82%, 20.98%, and 17.84% using EyeSwipe, HumHum, and Hummer, respectively. However, the correction rate in the last session of typing using Hummer is only 7.82%, suggesting that once users' familiarity with the Hummer method increases the instances of deletion reduce.

The selection of the first and the last letter is a critical factor in the word-level entry as the lexicon is filtered based on the first and last letter. Hence, we compute *selection accuracy* as the number of correctly selected first and last letters divided by the total number of selections required for the desired phrase. The overall selection accuracy for the first and last letter in a word by reverse crossing with EyeSwipe is 85.92%. For HumHum and Hummer, that use humming for selection, the corresponding selection accuracy is 82.93% and 82.70%, respectively.

The correction rates and selection accuracy over the five sessions are shown in Table 1. It is evident that selection accuracy has a direct impact on the correction rate, which also affects the performance (text entry and error rate) of all three methods. It also signifies the learning required to get accustomed to humming in synchronizing with the start and end of a word. Notably, in the initial sessions, despite having low selection accuracy and high correction rate, Hummer and HumHum show significantly better performance compared to EyeSwipe, indicating that humming based interaction is fast enough to recover from errors.

4.4.4 Subjective Feedback. We asked the participants for feedback about their overall performance, comfort, speed, accuracy, and ease of use. Questionnaire responses were on a seven-point Likert scale. Higher scores are better. See Figure 6 for a radar chart of the results.

The participants indicate that Hummer had a better overall performance (M = 5.75) than the HumHum (M = 5.0) and EyeSwipe method (M = 4.41). As it is shown in the radar chart, Hummer is also considered to be easier to learn, faster, more accurate, and more comfortable than HumHum and EyeSwipe. We have also asked participants about their overall preference on which method they would like to use for hands-free text entry. Seven participants selected Hummer as their preferred method, four opted for HumHum, and one picked EyeSwipe.

Moreover, we have asked the participants for comments about the methods in our evaluation. They were overall positive with Hummer, yet, sometimes they struggled with entering long words. This is reflected in feedback like *"its difficult with long words but works really good."* and *"for really long words it can be difficult to use Hummer."* In contrast, EyeSwipe was criticized for the reverse-crossing technique with feedback like *"the start button sometimes covers the letters I needed. It could be moved in a different section of the layout, then it might be more comfortable to use it," "going up and down to start, end the word is difficult. Instead it can be blinking,"* or one participant remarked that *"Eyeswipe is very stressful for [the] eyes."*

Table 1: Means over five sessions for correction rate and selection accuracy of the first and last letters of words.

	Session	1	2	3	4	5
Correction Rate [%]	EyeSwipe	20.9	16.5	18.7	15.5	12.5
	HumHum	25.5	26.7	25.8	18.0	9.4
	Hummer	26.2	22.3	18.8	14.1	7.8
Selection Accuracy [%]	EyeSwipe	78.9	87.0	86.2	86.8	90.7
	HumHum	79.7	74.0	80.7	86.1	94.1
	Hummer	77.6	77.9	82.5	82.9	92.6

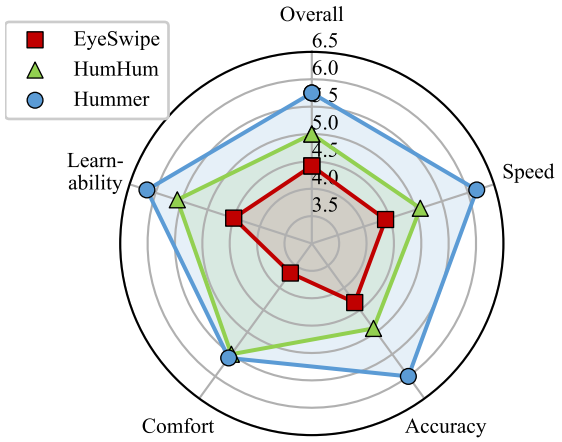

Figure 6: Subjective response (1 to 7) by method and questionnaire item. Higher scores are better.

5 DISCUSSION

We have designed and implemented Hummer and HumHum, two text entry methods combining hum and gaze, and have validated them by comparing them with the sophisticated baseline for hands-free text entry, EyeSwipe. Hummer and HumHum show excellent performance with an average entry rate of 15.48 WPM and 12.55 WPM, respectively. Using Hummer, participants could type 49% faster than with EyeSwipe (10.35 WPM) on average. Figure 4 depicts the preeminence of Hummer overall typing sessions. Besides, every participant reached an average entry rate of at least 16 WPM using Hummer, indicating the efficiency of the novel method. We would like to point out that the performance of EyeSwipe in our experiment is aligned with the reported results in the original EyeSwipe paper [23]. Although subjective to experimental settings and participant characteristics, the experiment provides a solid validation to reported comparisons.

It is interesting to note that eye tracking and humming both are novel input paradigms for most users (also stated by our participants), and hence it is evident that learning is essential to showcase the potential of the proposed methods. More specifically, in eye-tracking literature the typing speed in the last sessions and the maximum typing speed has been considered as a major baseline to assess the potential of interaction methods. In this regard, Hummer achieved an average entry rate of 20.45 WPM in the last session, i. e, after typing 25 phrases; and the maximum average text entry rate achieved in a session was 30.64 WPM, showcasing the potential of Hummer.

Our experiment design was focused on quantifying the humming performance as a hands-free method to assist gaze-based text entry for people with a long-term impairment, rather than testing the feasibility of humming against other interaction modalities. Hence, a similar hands-free and gaze-based method EyeSwipe was used for comparison. Different designs and modalities cater to different target groups and situations, comparing them all in a controlled study would be infeasible. Nevertheless, WPM as a universal benchmark provides a basis of comparison with other approaches. Extending this discussion, Table 2 summarizes the most prominent approaches of gaze-based text entry, providing a comparison based on the text entry rate in the last session and the overall maximum text entry rate.

The reported performance measures from our experiment were also aligned with subjective feedback. All participants rated Hummer significantly faster, accurate, and comfortable compared to EyeSwipe. Participants were often tired after using EyeSwipe and found the reverse-crossing uncomfortable. Participants felt engaged with the continuous humming in Hummer providing a rhythmic and playful feeling while bracketing the swipe action with a hum. However, a few participants found it difficult to produce a long continuous hum to enter very long words. Nonetheless, long words are rare, i. e., 80% of English words are between two and seven letters long[5], hence it affects Hummer's performance sparingly. The long-word-issue was a post-experiment finding and therefore not foreseen. In the future, to enable long continuous hum, we could imagine a possibility of pausing, e. g., if the hum stops but the gaze continues, humming detection might continue as well. For HumHum, synchronizing a short hum at the start and end has been a notable issue, e. g., participants hummed before their gaze reach the last character while swiping. Moreover, we have observed participants forgetting to finish a swipe action with a short hum at the end, which affected their performance.

The Hummer method as described so far is limited to words that are contained in the lexicon. However, for real-world deployment, for entering words that are not contained in the lexicon, e. g., names or uncommon words, the user can look at any letter in the virtual keyboard and perform a short hum for entering it character-by-character. After entering the unknown word, it would be added to the lexicon, to be found as a candidate word in the future. Numbers and special characters can also be entered by performing a short hum.

Another issue with the real-world deployment of the proposed approach would be triggering of selection events with external audio events, e. g., talking, coughing, sneezing, or ambient noise. The experiment was conducted in a controlled environment, however, a computational approach to classify humming would make the approach more robust. Moreover, the microphone threshold could be dynamically adjusted depending on the ambient noise level. In quiet surroundings, the humming threshold could be adjusted similarly as in conversations, e. g., people whisper to not disturb others (lowering threshold works because of less noise in surroundings). In fact, one of the authors typed with Hummer during development by just blowing into the microphone, without disturbing another colleague sharing the same office. Alternative methods like vocal cord vibration [36] could also improve the humming detection against external noise, and allow quieter hums to avoid distracting others. As a substitute, a similar method to the SilentVoice [7], a tooth-click that requires physical input to sense selection, could also be used in situations that a user has or wants to be quiet.

6 CONCLUSION

Hands-free text entry is a vivid field of research that is tackled with various interaction mechanisms, whether they are used as unimodal input or combined to multimodal systems. We introduce

[5] http://norvig.com/mayzner.html

Authors	Method	Average WPM	Last WPM	Maximum WPM	Practice Time
Rough et al. [41]	Dasher with adjustable dwell time	-	14.2	~19.5	~7.5 hours
Mott et al. [33]	Cascading dwell time	12.39	~10	13.79	20 phrases + ~150min
Urbina and Huckauf [51]	pEYEWrite with bigrams and word prediction	13.47 (last 3 session)	17.26	-	15 min + ~150 phrases
Majaranta et al. [28]	Adjustable dwell time	~15	19.9	-	10 days (~150min)
Diaz-Tula and Morimoto [3]	AugKey	15.31 (last 3 session)	~17	-	3-5 phrases + 84 min
Morimoto and Amir [31]	Context switching	12	-	20	5 min + 70 min
Morimoto et al. [32]	Context switching with dynamic targets	13.1	13.42	-	5 min + ~35 phrases
Pedrosa et al. [35]	Filteryedping	-	15.95	19.25	100 min
Kurauchi et al. [23]	EyeSwipe	9.92	11.7	20.6 (by an author)	2 phrases + 30 min
Kumar et al. [21]	TAGSwipe	15.46	~16	20.5	25 phrases
Hedeshy et al.	Hummer	15.48	20.45	30.64	25 phrases

Table 2: Summary of text entry rates from recent eye-typing literature. Some entries are inferred from the figures and other data available in the original papers, which is indicated by the '~' symbol. Some entries are not available, which is indicated by the '-' symbol. Some papers only reported the mean of the last session(s). The count of sessions is therefore mentioned in parentheses. A few papers report the WPM only from the last session, which we display in the column "Last WPM." Practice time correlates with the approximate training effort required by participants to achieve the entry rate in the last session.

Hummer as a novel bi-modal method for hands-free text entry combining eye-tracking and humming. The average text entry speed with Hummer is significantly faster than the gaze-based hands-free method EyeSwipe. After four sessions of typing with Hummer, the participants achieve a commendable speed of 20.45 WPM. The qualitative responses and explicit feedback also indicate a clear preference for Hummer as a fast, comfortable, and easy to learn text entry method. Most interestingly, participants enjoyed the experience of swiping and humming in a rhythm, which they found very persuasive.

We envision further methods to optimize hands-free text entry using humming. A differentiation among hums that indicate agreement, disagreement, or questioning could enable more complex interactions. For text entry, this could be useful in switching between different modes or layouts of the virtual keyboard. Different tones of humming could be useful in text entry for a user to provide feedback about the current swiping process. The system could then react instantaneously and correct the gaze path on-the-fly, according to the non-verbal feedback of the user. Beyond text entry, the intersections of eye tracking and humming could lead to several exciting applications in the domain of communication, gaming, and virtual or augmented reality.

ACKNOWLEDGMENTS

This work was made possible thanks to funding provided by ZIM,[6] a funding program of the German Federal Ministry for Economic Affairs and Energy under grant no. KK5057901LF0.[7] We would like to thank all the participants of the experiment, the IPVS institute of Stuttgart University that provided support, and Andreea Crăciun for helping with the demo video.

REFERENCES

[1] Jeff Bilmes, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, James Landay, Patricia Dowden, et al. 2005. The Vocal Joystick: A voice-based human-computer interface for individuals with motor impairments. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*. 995–1002.

[2] Peter M. Corcoran, Florin Nanu, Stefan Petrescu, and Petronel Bigioi. 2012. Real-time eye gaze tracking for gaming design and consumer electronics systems. *IEEE Transactions on Consumer Electronics* 58, 2 (2012), 347–355.

[3] Antonio Diaz-Tula and Carlos H. Morimoto. 2016. AugKey: Increasing foveal throughput in eye typing with augmented keys. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, 3533–3544. https://doi.org/10.1145/2858036.2858517

[4] Thomas Eiter and Heikki Mannila. 1994. *Computing discrete Fréchet distance*. Technical Report. Citeseer.

[5] Torsten Felzer, I. Scott MacKenzie, Philipp Beckerle, and Stephan Rinderknecht. 2010. Qanti: a software tool for quick ambiguous non-standard text input. In

[6] https://zim.de
[7] https://bmwi.de

International Conference on Computers for Handicapped Persons. Springer, 128–135.

[6] Maurice René Fréchet. 1906. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)* 22, 1 (1906), 1–72.

[7] Masaaki Fukumoto. 2018. SilentVoice: Unnoticeable Voice Input by Ingressive Speech. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology* (Berlin, Germany) *(UIST '18)*. Association for Computing Machinery, New York, NY, USA, 237–246. https://doi.org/10.1145/3242587.3242603

[8] Markus Funk, Vanessa Tobisch, and Adam Emfield. 2020. Non-Verbal Auditory Input for Controlling Binary, Discrete, and Continuous Input in Automotive User Interfaces. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–13.

[9] 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 *Proceedings of Human-Computer Interaction–INTERACT*. Springer, Berlin, 121–128.

[10] 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* (Tübingen, Germany) *(APGV '07)*. ACM, New York, 141–141. https://doi.org/10.1145/1272582.1272618

[11] Takeo Igarashi and John F. Hughes. 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*. 155–156.

[12] Robert J. K. Jacob. 1990. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Seattle, WA) *(CHI '90)*. ACM, New York, 11–18. https://doi.org/10.1145/97243.97246

[13] Robert J. K. Jacob. 1990. What you look at is what you get: Eye movement-based interaction techniques. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90)*. ACM, New York, 11–18. https://doi.org/10.1145/97243.97246

[14] Jyh-Shing Jang, Chao-Ling Hsu, and Hong-Ru Lee. 2005. Continuous HMM and Its Enhancement for Singing/Humming Query Retrieval. *Proc. of ISMIR*, 546–551.

[15] Joseph Jordania. 2009. Times to fight and times to relax: Singing and humming at the beginnings of human evolutionary history. *Kadmos* 1, 2009 (2009), 272–277.

[16] Josh Kaufman. 2015. Google 10000 english.

[17] Suk-Jun Kim. 2018. *Humming*. Bloomsbury Publishing USA.

[18] Alexios Kotsifakos, Panagiotis Papapetrou, Jaakko Hollmén, Dimitrios Gunopulos, and Vassilis Athitsos. 2012. A Survey of Query-by-Humming Similarity Methods. In *Proceedings of the 5th International Conference on Pervasive Technologies Related to Assistive Environments* (Heraklion, Crete, Greece) *(PETRA '12)*. Association for Computing Machinery, New York, NY, USA, Article 5, 4 pages. https://doi.org/10.1145/2413097.2413104

[19] Per-Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the ACM Symposium on Eye Tracking Research and Applications (ETRA '12)*. ACM, New York, 241–244.

[20] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK 2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '04)*. ACM, New York, 43–52.

[21] Chandan Kumar, Ramin Hedeshy, I. Scott MacKenzie, and Steffen Staab. 2020. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–12. https://doi.org/10.1145/3313831.3376317

[22] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, 1952–1956. https://doi.org/10.1145/2858036.2858335

[23] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (San Jose, CA) *(CHI '16)*. ACM, New York, 1952–1956. https://doi.org/10.1145/2858036.2858335

[24] Andrew T. N. Kurauchi. 2018. EyeSwipe: text entry using gaze paths.

[25] Yi Liu, Chi Zhang, Chonho Lee, Bu-Sung Lee, and Alex Qiang Chen. 2015. Gazetry: Swipe text typing using gaze. In *Proceedings of the annual meeting of the australian special interest group for computer human interaction*. ACM, 192–196.

[26] I. Scott MacKenzie and R. William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, 754–755.

[27] Päivi Majaranta. 2012. Communication and text entry by gaze. In *Gaze interaction and applications of eye tracking: Advances in assistive technologies*. IGI Global, 63–77.

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

[29] Päivi Majaranta and Andreas Bulling. 2014. Eye tracking and eye-based human–computer interaction. In *Advances in physiological computing*. Springer, 39–65.

[30] Raphael Menges, Chandan Kumar, and Steffen Staab. 2019. Improving user experience of eye tracking-based interaction: Introspecting and adapting interfaces. *ACM Transactions on Computer-Human Interaction* 26, 6, Article 37 (Nov. 2019), 46 pages. https://doi.org/10.1145/3338844

[31] Carlos H. Morimoto and Arnon Amir. 2010. Context Switching for Fast Key Selection in Text Entry Applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, NY, USA, 271–274. https://doi.org/10.1145/1743666.1743730

[32] Carlos H. Morimoto, Jose A. T. Leyva, and Antonio Diaz-Tula. 2018. Context switching eye typing using dynamic expanding targets. In *Proceedings of the Workshop on Communication by Gaze Interaction* (Warsaw, Poland) *(COGAIN '18)*. ACM, New York, Article 6, 9 pages. https://doi.org/10.1145/3206343.3206347

[33] 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 ACM CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, 2558–2570.

[34] Steffen Pauws. 2002. CubyHum: A Fully Operational Query by Humming System. In *ISMIR 2002 Conference Proceedings*. 187–196.

[35] Diogo Pedrosa, Maria da Graça Pimentel, and Khai N. Truong. 2015. Filteryedping: A dwell-free eye typing technique. In *Extended Abstracts of the ACM SIGCHI Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, 303–306. https://doi.org/10.1145/2702613.2725458

[36] Nathalia Peixoto, Hossein Ghaffari Nik, and Hamid Charkhkar. 2013. Voice Controlled Wheelchairs. *Comput. Methods Prog. Biomed.* 112, 1 (Oct. 2013), 156–165. https://doi.org/10.1016/j.cmpb.2013.06.009

[37] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and touch interaction on tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. ACM, New York, 301–311. https://doi.org/10.1145/2984511.2984514

[38] Ondřej Poláček, Zdeněk Míkovec, Adam J. Sporka, and Pavel Slavík. 2010. New way of vocal interface design: Formal description of non-verbal vocal gestures. *Proceedings of the CWUAAT* (2010), 137–144.

[39] Ondrej Polacek, Zdenek Mikovec, Adam J. Sporka, and Pavel Slavik. 2011. Humsher: a predictive keyboard operated by humming. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*. 75–82.

[40] Ondrej Polacek, Adam J. Sporka, and Pavel Slavik. 2017. Text input for motor-impaired people. *Universal Access in the Information Society* 16, 1 (2017), 51–72.

[41] 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*. ACM, New York, 169–176.

[42] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26, 1 (February 1978), 43–49. https://doi.org/10.1109/TASSP.1978.1163055

[43] 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*. ACM, New York, 215–220.

[44] Korok Sengupta, Raphael Menges, Chandan Kumar, and Steffen Staab. 2019. Impact of variable positioning of text prediction in gaze-based text entry. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications* (Denver, Colorado) *(ETRA '19)*. ACM, New York, Article 74, 9 pages. https://doi.org/10.1145/3317956.3318152

[45] R. William Soukoreff and I. Scott MacKenzie. 2003. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error Mmtric. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Ft. Lauderdale, Florida, USA) *(CHI '03)*. ACM, New York, 113–120. https://doi.org/10.1145/642611.642632

[46] Adam J. Sporka, Torsten Felzer, Sri H. Kurniawan, Ondřej Poláček, Paul Haiduk, and I. Scott MacKenzie. 2011. CHANTI: Predictive Text Entry Using Non-Verbal Vocal Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11)*. Association for Computing Machinery, New York, NY, USA, 2463–2472. https://doi.org/10.1145/1978942.1979302

[47] Adam J. Sporka, Sri H. Kurniawan, Murni Mahmud, and Pavel Slavík. 2006. Non-speech input and speech recognition for real-time control of computer games. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. 213–220.

[48] Melisa Stevanovic. 2013. Managing participation in interaction: The case of humming. *Text and Talk* 33 (01 2013), 113–137. https://doi.org/10.1515/text-2013-0006

[49] Kevin Toohey and Matt Duckham. 2015. Trajectory similarity measures. *Sigspatial Special* 7, 1 (2015), 43–50.

[50] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash away!: longitudinal study of fast text entry by eye gaze. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications (ETRA '08)*. ACM, New York, 19–26.

[51] Mario H. Urbina and Anke Huckauf. 2010. Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research and Applications* (Austin, Texas) *(ETRA '10)*. Association for Computing Machinery, New York, 315–322. https://doi.org/10.1145/1743666.1743738

[52] Alex Waibel and Kai-Fu Lee (Eds.). 1990. *Readings in Speech Recognition*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[53] Colin Ware and Harutune H. Mikaelian. 1987. An evaluation of an eye tracker as a device for computer input. In *Proceedings of the ACM SIGCHI/GI Conference on Human Factors in Computing Systems and Graphics Interface* (Toronto, Ontario, Canada) *(CHI+GI '87)*. ACM, New York, 183–188. https://doi.org/10.1145/29933.275627

[54] Richard Watts and Peter Robinson. 1999. Controlling computers by whistling. In *Proceedings of Eurographics UK*. Cambridge University Press.

[55] 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 ACM Symposium on Eye Tracking Research & Applications (ETRA '08)*. ACM, New York, 11–18.

[56] Nicole Yankelovich, Gina-Anne Levow, and Matt Marx. 1995. Designing SpeechActs: Issues in Speech User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '95)*. ACM Press/Addison-Wesley Publishing Co., USA, 369–376. https://doi.org/10.1145/223904.223952

[57] Shumin Zhai and Per-Ola Kristensson. 2012. The word-gesture keyboard: Reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101.

[58] Xiaoyu (Amy) Zhao, Elias D. Guestrin, Dimitry Sayenko, Tyler Simpson, Michel Gauthier, and Milos R. Popovic. 2012. Typing with Eye-Gaze and Tooth-Clicks. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 341–344. https://doi.org/10.1145/2168556.2168632

[59] Daniel Zielasko, Neha Neha, Benjamin Weyers, and Torsten W. Kuhlen. 2017. A reliable non-verbal vocal input metaphor for clicking. In *2017 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, 40–49.