Smartphone-Based Gaze Gesture Communication for People with Motor Disabilities

Xiaoyi Zhang
University of Washington
Seattle, WA, USA
xiaoyiz@cs.washington.edu

Harish Kulkarni
Microsoft Research
Redmond, WA, USA
harish.kulkarni@microsoft.com

Meredith Ringel Morris
Microsoft Research
Redmond, WA, USA
merrie@microsoft.com

ABSTRACT
Current eye-tracking input systems for people with ALS or other motor impairments are expensive, not robust under sunlight, and require frequent re-calibration and substantial, relatively immobile setups. Eye-gaze transfer (e-tran) boards, a low-tech alternative, are challenging to master and offer slow communication rates. To mitigate the drawbacks of these two status quo approaches, we created GazeSpeak, an eye gesture communication system that runs on a smartphone, and is designed to be low-cost, robust, portable, and easy-to-learn, with a higher communication bandwidth than an e-tran board. GazeSpeak can interpret eye gestures in real time, decode these gestures into predicted utterances, and facilitate communication, with different user interfaces for speakers and interpreters. Our evaluations demonstrate that GazeSpeak is robust, has good user satisfaction, and provides a speed improvement with respect to an e-tran board; we also identify avenues for further improvement to low-cost, low-effort gaze-based communication technologies.

Author Keywords
Eye gesture; accessibility; augmentative and alternative communication (AAC); Amyotrophic lateral sclerosis (ALS).

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g., HCI); K.4.2. Assistive Technologies for Persons with Disabilities.

INTRODUCTION
Eye gaze keyboards [21,28] are a common communication solution for people with Amyotrophic Lateral Sclerosis (ALS) and other motor impairments. ALS is a neurodegenerative disease that leads to loss of muscle control, including the ability to speak or type; because eye muscle movement is typically retained, people with late-stage ALS usually rely on eye tracking input for communication. Unfortunately, the hardware for commercial gaze-operated keyboards is expensive. For example, the poplar Tobii Dyanvox [28] eye gaze hardware and software package costs between $5,000 and $10,000, depending on the specific model and configuration. Additionally, eye trackers do not work in certain conditions that interfere with infrared light (IR) (such as outdoors), and require a stand to keep the apparatus relatively static with respect to the user, which makes it difficult to use in certain situations such as in a car or in bed.

Figure 1. The communication partner of a person with motor disabilities can use the GazeSpeak smartphone app to translate eye gestures into words. (a) The interpreter interface is displayed on the smartphone's screen. (b) The speaker sees a sticker depicting the four letter groupings affixed to the phone's case.

Eye-gaze transfer (e-tran) boards [3,17] are an alternative, low-tech communication solution, where clusters of letters are printed on a transparent plastic board. The communication partner holds the board, and observes the gaze pattern of the person with ALS (PALS), who selects a letter by making two coarse eye gestures: one to select which of several letter groupings contain the target letter, and a second to indicate the position within the group. Unfortunately, e-tran boards have several drawbacks: their cost is relatively low compared to gaze-tracking systems, but is not negligible (~$100); the large plastic board (e.g., one popular model [17] measures 14″ x 18″) is not easily portable; patients need to perform two eye gestures to enter one letter, which may take more than 8 seconds [25] including correcting mistakes; our survey of PALS' caregivers indicated they found e-tran to have a high learning curve, as they have to decode and remember entered characters and predict words.

In this work, we investigate how to provide low-cost, portable, robust gaze-based communication for PALS that is easy for patients and caregivers to use. Our solution uses

CHI 2017, May 06 - 11, 2017, Denver, CO, USA

ACM 978-1-4503-4655-9/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3025453.3025790

a smartphone to capture eye gestures and interpret them using computer vision techniques. Our system, GazeSpeak, consists of computer-vision-based eye gaze recognition, a text prediction engine, and text entry interfaces that provide feedback to the speaker (PALS) and interpreter (caregiver or communication partner). GazeSpeak is as portable as a mobile phone, patients perform only one eye gesture per character, and the mobile phone records the entered characters and predicts words automatically. Further, GazeSpeak does not require re-calibration under similar lighting conditions, and it does not require a bulky stand, as caregivers can hold the phone. GazeSpeak has no additional cost other than a smartphone, which most people in the U.S. (68% in 2015) [2] already own.

We evaluate the error rate of our eye gesture recognition, as well as satisfaction and usability for both the speaker and interpreter. We also compare the communication speed of GazeSpeak to that of an e-tran board. We find GazeSpeak is robust, has good user satisfaction, and provides a speed improvement with respect to e-tran. GazeSpeak offers a viable low-cost, portable, lighting-robust alternative for situations in which eye-tracking systems are unaffordable or impractical.

The specific contributions of this work include:

- The GazeSpeak system, including algorithms to robustly recognize eye gestures in real time on a hand-held smartphone and decode these gestures into predicted utterances, and user interfaces to facilitate the speaker and interpreter communication roles.
- User study results demonstrating GazeSpeak's error rate and communication speed, as well as feedback from PALS and their communication partners on usability, utility, and directions for further work.

RELATED WORK

Low-Tech Gaze Input Solutions

As shown in Figure 2, an e-tran board [3,17] is a low-tech AAC (augmentative and alternative communication) solution that comprises a transparent board containing groups of symbols, such as letters. An interpreter holds the board and observes and decodes the eye gestures of the speaker; the speaker gazes in the direction of a group to select a cluster of symbols and then again to disambiguate the location of a specific symbol within the cluster. Another low-tech solution, EyeLink [25], is also a transparent board printed with letters. To use EyeLink, the speaker keeps staring at the desired letter, while the interpreter moves the board

Figure 2. Low-tech gaze input solutions: a) E-tran board; b) EyeLink; c) Laser pointer on communication board.

Figure 3. High-tech gaze recognition solutions: (a) Tobii Dynavox eye-tracking computer [28]**; (b) EyeSpeak eye-tracking glasses** [18]**.**

until she can "link" her own eye gaze with that of the speaker, and note the letter on the board where their eyes meet. Families and occupational/speech therapists may also develop myriad custom low-tech communication solutions, such as attaching a laser pointer to a patient's head (if they have head movement control) that can be used to point at letters printed out on a poster or board. While relatively cheap (costing tens or low hundreds of dollars for materials), low-tech solutions provide low communication bandwidth (entering a letter takes 8-12 seconds [25]) and place a high learning/cognitive burden on the interpreter. Low tech solutions, while not optimal, are nonetheless important in offering a communication option in situations where other options are unavailable.

High-Tech Gaze Recognition Solutions

As shown in Figure 3, commercial gaze-operated keyboards allow PALS to type characters to complete a sentence [21,29], or select symbols to build sentences word by word [30]. Typically, systems are dwell-based [15,20] (i.e., the user dwells their gaze on a key for a period, typically several hundred milliseconds, in order to select that key), though dwell-free gaze systems [13,37] are an emerging area of research and commercial development that may offer further speed improvements. Other, less common, gaze input interfaces may include techniques like scanning [4] or zooming [34].

Specialized hardware and setups are used in eye gaze tracking systems. Head-mounted eye trackers [1,18] keep the eyes and the camera close and relatively static during head movement. However, such systems are expensive, and their bulk/weight is not typically comfortable for constant use as required by someone with a motor disability; head-worn systems may also interfere with eye contact and the ability to observe the environment, making them even more impractical for constant use.

Other eye tracker solutions mount a camera on a computer monitor or table, and find pupil locations based on reflections. Because of the longer distance between eye and camera, commercial systems [27,28] often emit IR to increase light reflection from the eyes. IR makes eye movement more detectable, but limits outdoor usage due to interference from the strong IR in sunlight. Eye trackers range in price from hundreds to thousands of dollars depending on quality; some relatively low-cost commercial eye trackers are available (e.g., the Tobii EyeX costs around $150 USD); however, people who rely on eye gaze for

AAC must also purchase a computer to connect to the eye tracker and proprietary software that allows the eye gaze to control a keyboard and other computer programs; such bundles [28] typically cost between $5,000 - $10,000. In the United States, government health insurance (Medicare) has only just begun to help pay for AAC devices [32]; some insurers do not consider access to AAC as a medical necessity.

There are also attempts at using low-cost webcams or phone cameras to recognize eye gaze location, and use direct gaze pointing as an input method. WebGazer [24] uses a webcam to infer the gaze locations of web visitors on a page in real time, and self-calibrates while visitors interact with content on the screen with their mouse cursor. Any movement of the camera or head requires additional interactions to re-calibrate. However, PALS or other motor impairments cannot move the mouse to interact, which would break WebGazer's self-calibration algorithm. iTracker [12] uses an iPhone's front camera to estimate gaze location on the screen. Calibration can increase accuracy, but is not required, as it is pre-trained on a large-scale eye tracking database. However, extending this method to additional mobile devices may require collecting large eye tracking datasets for each device type. In addition, its prediction error on the iPhone 5 is almost 30% of the screen width.

Besides gaze location, eye-switches and eye gestures can also be used as an input method. Eye-switches [7] use voluntary eye blinks as binary signals, which helps in scanning input methods. In a 2D letter grid, the system moves the focus line by line, and the first eye-switch can select the line that contains the desired letter; then the system moves the focus letter by letter on that line, and the second eye-switch selects the desired letter. EyeWrite [37] is the first letter-like gestural text entry system for the eyes, and uses a Tobii IR eye tracker to capture gaze input. Its letter-drawing interface is a square with four corners. The user has to move gaze to the corners to map out a letter. EyeWrite does not require the dwell time to select a letter, except needing slight dwell time to signal character segmentation. Testing found EyeWrite's text entry speed was around 5 wpm. Vaitukaitis and Bulling [31] presented a prototype that could recognize different continuous eye gesture patterns on a laptop and mobile phone. For example, the user can move his gaze left, then up, then right, and finally down to draw a diamond pattern. However, their evaluation was conducted in an indoor setting with controlled lighting, requiring fixed device position and distance to participants. Even with these carefully controlled conditions, the performance of this prototype was less than 5 frames per second on the phone with only 60% accuracy. In our work, we do not use compound eye gestures to draw letters or shapes; rather, we use simple, single-direction gestures (e.g., look left) to select among groups of letters.

DESIGN GOALS

Before we started to build GazeSpeak, we conducted an online survey targeted at communication partners (spouses, caregivers, etc.) of PALS, and advertised our survey via an email list about ALS in the Seattle metropolitan area. We received 22 responses; this low number is not surprising given the low incidence rate of ALS, about 1 in 50,000 people [26].

All of the respondents indicated owning either an iPhone or Android phone. 36% of them said their companion with ALS did not own an eye-tracking system. For those whose companion did have an eye-tracker, 28% of them reported that the PALS was unable to use the eye-tracking system during more than half of the waking hours, due to issues such as system crashes, positioning at angles or locations where mounting the system is impractical (being in bed, inclined in a chair, or using the bathroom), being in situations with limited space (such as traveling in a car), or outdoors due to interference from sunlight. They also noted that when their companions with ALS only want to convey a quick communication, the start-up costs of such systems (which respondents indicated required frequent re-calibration in practice) seemed too long in proportion to the length of the communication.

64% of respondents indicated having used e-tran boards as an alternative when eye tracking was not available. Some who had not tried e-tran boards indicated they had not done so because their companion was in earlier stages of ALS's progression and still retained some speech capabilities. The self-reported learning curve for e-tran was varied; 36% of respondents reported it took a few hours to master, 14% spent a day, 29% spent a week, and 21% indicated it took more than a month. Respondents reported challenges in using e-tran boards: the interpreter may misread the eye gesture, forget the sequence of previously specified letters, and finds the board heavy/uncomfortable to hold; for the speaker, it is difficult to correct a mistake in gesture or interpretation. Respondents indicated that one or two words is the typical length of an utterance specified via e-tran board.

These survey responses added to our knowledge of the concerns facing end-users of gaze-based AAC; based on these responses and the other cost and practicality issues discussed in the *Introduction* and *Related Work* sections, we articulated several design goals for GazeSpeak:

1) *Create a low-cost/high-tech alternative in an eco-system that currently offers only high-cost/high-tech and low-cost/low-tech solutions*: GazeSpeak is meant to be affordable for people who may not be able to purchase an expensive, multi-thousand-dollar eye tracking solution. Since smartphone ownership is common in the U.S. [2] and was ubiquitous among our survey respondents, a smartphone app seems a reasonable way to reach a large audience at no additional cost to end-users. We do not expect that our smartphone app should exceed the

performance (in terms of text entry rates) of expensive, commercial eye tracker setups; however, we do expect that GazeSpeak will offer performance and usability enhancements as compared to the current low-cost alternative. In addition to supporting a low-cost solution, the mobile phone form-factor preserves (and even exceeds some of) the advantages of current low-tech solutions in being smaller, lighter, more portable, and more robust to varied lighting sources than high-cost commercial eye-tracking setups.

2) *Simplify the e-tran process through automation*: While cheap and flexible for many scenarios, e-tran still has several drawbacks that we aim to mitigate with GazeSpeak, particularly: (1) slow speed of text entry, (2) difficulty of error correction, and (3) high cognitive burden for the interpreter.

SYSTEM DESIGN AND IMPLEMENTATION

GazeSpeak application currently supports iOS devices released after 2012. It uses the phone's built-in camera, touch screen, and speaker, and does not require extra hardware. To improve learnability and usability, a simple guide representing the four keyboard groupings and associated gesture directions can be printed and taped to the back of the phone's case (Figure 1b). To use this system, the interpreter holds the phone and points the back camera toward the speaker (PALS) (Figure 1a and Figure 8).

The app consists of three major components: 1) eye gesture recognition, 2) a predictive text engine, and 3) a text entry interface.

Eye Gesture Recognition

GazeSpeak can robustly recognize six eye gestures for both eyes: look up, look down, look left, look right, look center, and close eyes. If the speaker can wink (closing only one eye at a time), it also recognizes winking the left eye and winking the right eye. The recognition code is written in C++ and depends on open-source libraries: Dlib [11] and OpenCV [5]. To save development time, we only implemented eye gesture recognition on iOS; however, GazeSpeak could be easily extended to other platforms that support these dependencies, such as Android, Windows, Mac and Linux.

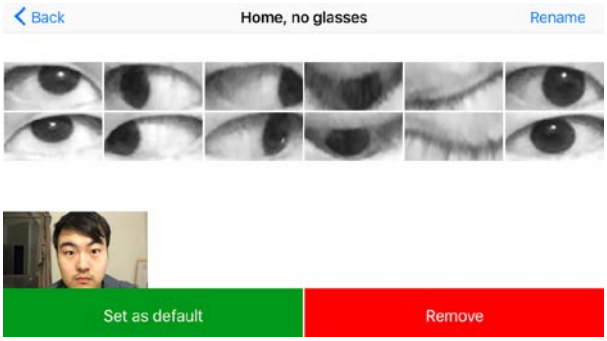

Figure 4. Calibration review screen. A photo at the bottom left serves as a reminder of context (e.g., indoors, glasses off) that can also be added to the calibration name (top). This screen also shows the calibration templates captured for the *up*, *left*, *right*, *down*, *closed* and *center* gestures for each eye.

Calibration

GazeSpeak collects a set of eye gestures from the speaker as calibration templates. When the interpreter holds the phone with the rear camera facing the speaker, they can press the "calibrate" button. The app then plays audio instructions that tell the speaker to prepare to calibrate, and then instructs him to look up, down, left, right, center, and to close both eyes. Template matching will be more robust if these six gestures are distinct from each other, so for best performance, the speaker should make eye gestures that are exaggerated (far up, to the rightmost, etc.) to the extent possible while not being uncomfortable (Figure 4). In addition, while looking down, eyelashes may cover eyes naturally, which makes it appear to be similar to closed eyes. Thus, for the *look down* gesture, we suggest speakers try to keep their eyes open as wide as they are able while looking down, to improve performance (Figure 4). This calibration sequence takes ten seconds. Calibration is only required for the first time using the app, or if lighting conditions vary drastically (having a separate outdoor and indoor calibration may improve performance). Calibrations taken under different circumstances can be stored, labeled, loaded as needed, and transferred between different iOS devices.

To obtain calibration templates for eye gaze recognition, GazeSpeak performs the following four steps:

1) *Detect face and align landmarks*: We use iOS's built-in face detector to obtain a rectangle containing the speaker's face. (OpenCV's face detector could be used when extending GazeSpeak to other platforms that do not have built-in face detection support.) Then, we use dlib's implementation of fast face alignment [10] to extract landmarks on the face.

2) *Extract an image of each eye*: Once we get face landmarks, we calculate the bounding rectangle of eye landmarks. Then we extract images of each eye and process them separately.

3) *Normalize eye images*: We resize each eye image to 80x40 pixels. Then we convert the image to the HSV color space and only keep its V channel (value, i.e., brightness).

4) *Store eye gesture template*: We save the normalized eye images on the phone, using the filenames to indicate the eye (left/right) and gesture (up/down/left/right/center/closed).

Recognition Algorithm

Once there is a new or existing calibration for the speaker, he may perform eye gestures to communicate while the interpreter aims the rear phone camera toward the speaker (Figure 1a). The interpreter can see the camera view on the screen and ensure the speaker's face is in the view (Figure 5x). For each camera frame, GazeSpeak performs the following steps as shown in Figure 5. It detects the face and

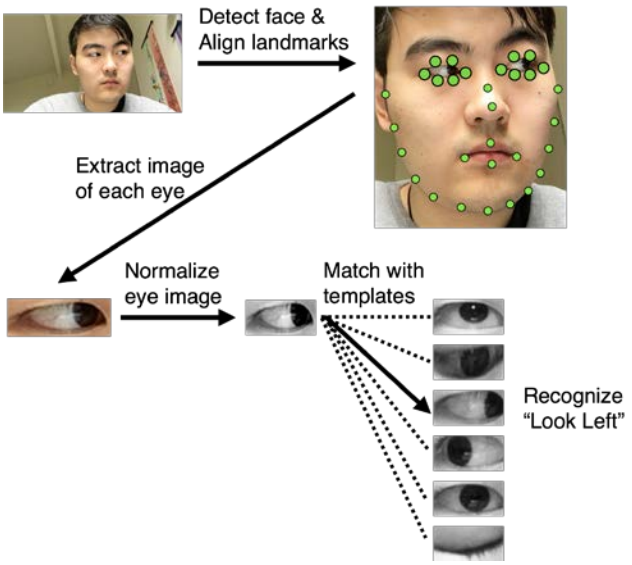

Figure 5. Flowchart of eye gesture recognition algorithm.

aligns landmarks, extracts an image of each eye, and normalizes the eye images (the same first three steps as during calibration). Then GazeSpeak classifies eye gestures by matching the normalized eye images extracted from the current video frame with the calibration templates.

Mean squared error (MSE) [16] measures the difference between two images. We use MSE to find the closest match for the normalized eye images among the six eye gesture templates obtained during calibration. Structural similarity [33] and the sum of absolute differences [35] are also candidates for measuring the difference between two images; however, MSE achieved the best recognition rate in our iterative testing and development of the system.

Performance

We tested our eye gesture recognition on recent models of the iPhone and iPad. For the iPhone, the recognition speed ranges from 27 frames per second (fps) (iPhone 6s Plus) to 17 fps (iPhone 5s). For the iPad, it ranges from 34 fps (iPad Pro 9.7) to 16 fps (iPad mini 2). The slowest device still processes enough frames to confirm gestures for text entry.

Robustness

Our algorithm works in a variety of lighting conditions, including indoors and outdoors. Since it uses an RGB rather than IR camera, its performance is unlikely to be degraded under sunlight. In low-light conditions, GazeSpeak can turn on the phone's flashlight to make the speaker's face visible. To avoid flash burn to the eyes, we need to apply a diffuser-like tape such as 3M tape (less than $1) over the flashlight. Our algorithm tolerates two major transformations: *Scaling* (e.g., if the interpreter moves the phone closer to the speaker), *Translation* (e.g., if the speaker moves his head a bit, or the interpreter slightly moves the phone while holding it), and *Scaling* + *Translation* (e.g., the interpreter puts down the phone, and later holds it in a slightly different position). *Rotation* of speaker's head or the phone significantly changes perceived face shape and thus reduces recognition accuracy. Our app shows the face image and recognition results to visually assist the interpreter in self-correcting the positioning.

Accuracy

To assess the accuracy of our gesture recognition system, we recruited 12 participants through email lists within our organization, and paid each participant $5 for a 30-minute session. They reported a mean of 29 years old (min 20, max 44); five were male and seven were female. Six had a normal (uncorrected) vision, one wore contact lenses, and five wore glasses. Participants had varied skin and eye colors. During the study, each participant performed each of up/down/left/right/closed eye gestures 30 times; after each gesture, the participant looked back to center. In total, we recorded 300 eye gestures for each participant (30 x 5 + an additional 150 gazes toward the center).

	Up	Down	Left	Right	Close	Center
Mean	88.6%	75.3%	87.8%	86.9%	77.5%	98.6%
Stdev	5.2%	17.5%	4.8%	7.4%	13.5%	1.8%

Table 1. GazeSpeak's recognition rate of each eye gesture.

For our 12 participants, the algorithm correctly recognized an average of 86% of all eye gestures (min = 68%, max = 92%, med = 89%, stdev = 6.9%). Table 1 reports the recognition accuracy for each of the six eye gestures. GazeSpeak's algorithm can recognize the center gesture with near perfect accuracy; looking up, left, or right also had good recognition rates. Looking down and closing the eyes are harder to recognize due to their sometimes-similar appearance, as explained in the *Calibration* section.

We did not see any effect from age, gender, eye color, or skin color on the accuracy. For participants without corrective lenses, our system achieved 89.7% accuracy. Wearing contact lenses (only one participant) did not have a noticeable effect on accuracy (89%). However, an independent-samples t-test indicates wearing glasses significantly lowers the recognition accuracy to 80.4% ($t(9) = 2.714$, $p = 0.024$). Although glasses do not the affect the face detector, we suspect they may interrupt facial landmark alignment, sometimes resulting in mislabeling the eye area. Because we analyzed the recorded gesture videos offline, when we record the video, we did not have the recognition results as feedback to warn us to move the phone when the facial landmark alignment went wrong; however, when using GazeSpeak in real time, the interpreter can adjust the phone angle until she hears the correct recognition audio feedback, thus mitigating error for users with glasses. In fact, during our *Usability Study* (next section), we encountered very few recognition errors, even for speakers with glasses, likely because interpreters were able to use GazeSpeak's video and audio feedback to mitigate error.

Predictive Text Engine

To avoid fatigue from making complex eye gestures, improve learnability, and improve recognition rates,

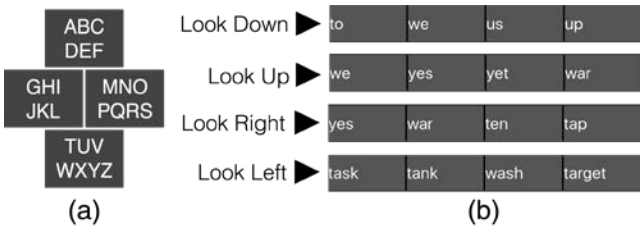

Figure 6. Word predictions update after each gesture in this example four-gesture sequence to spell the word "task".

GazeSpeak uses a small number of simple eye gestures (up/down/left/right) to refer to all 26 letters of the English alphabet, using only one gesture per character entered (to reduce fatigue and increase throughput). This design leads to an ambiguous keyboard (Figure 6a) in which the letters of the alphabet are clustered into four groups that can each be indicated with one of the up, down, left, or right gestures. GazeSpeak implements a predictive text engine [22,23] to find all possible words that could be created with the letters in the groups indicated by a gesture sequence. Our implementation uses a trie [14] data structure to store common words and their frequencies in oral English: we select the most common 5,000 words (frequency min=4,875, max=22,038,615, mean=12,125) from this wordlist [6]. This trie can also be extended by an interpreter, who can add the speaker's frequently-used words (including out-of-dictionary words, such as names). For a series of eye gestures with length *n*, our trie structure allows us to rapidly look up its matching words and word frequencies in O(n) time. GazeSpeak can also look up high-frequency words whose initial characters match the gesture sequence thus far.

We display a list of likely words based on the current gesture sequence on the phone's screen for the interpreter (Figure 6b); we currently show the top four predictions by default, though additional predictions can be cycled through by horizontal scrolling. The prediction and auto-complete features help the interpreter predict possible words while the speaker is still entering text, enabling guessing ahead to improve throughput, and easing the interpretation burden that is present in using a low-tech e-tran board.

For instance, in order to enter the word *TASK*, the speaker first looks down for T, then looks up for A, looks right for S, and looks left for K. The predictions after each eye gesture in this sequence are shown in Figure 6.

To make it easier for the speaker to learn the gesture direction associated with a given letter, our four groups are simply clusters in alphabetical order (Figure 6a). An alternative letter grouping may reduce the conflict rate for word prediction; however, learnability was a higher design priority for this audience, as learnability is one of the challenges of using e-tran boards that motivated GazeSpeak's creation. The word prediction conflict rate is reasonably low for this intuitive letter grouping. The 5,000 most common words in our dictionary can be represented by 3248 unique eye gesture sequences. 83.5% of sequences match a unique word, 92.6% match two or fewer words, 97.7% match four or fewer words, 99.2% match six or fewer words, and 99.6% match 8 or fewer words.

If a word is not in the trie, the speaker has to use a scan-based method to type the word letter by letter. To type a letter, he stares at the direction that contains that letter; GazeSpeak would read each letter in that key, and the speaker looks back to center once he hears the desired letter.

Text Entry Interface

The speaker and interpreter use different interfaces. The speaker sees the back of the phone, and thus the screen-less text entry interface consists of two major components: 1) a sticker displaying the four-key keyboard, and 2) audio feedback. The interpreter sees the phone screen, which shows four major components as in Figure 7: 1) the four-key keyboard, 2) the input box and word predictions, 3) the sentence box and 4) the camera preview.

Speaker Interface

On the back of the phone, a four-key sticker (figure 1b) reminds the speaker of the letter groupings associated with each of the four gesture directions. To enter one character, the speaker moves his eyes in the direction associated with that letter's group. Once GazeSpeak detects that the eyes have settled in one direction, it speaks aloud the direction it detected (e.g., "Up"). The speaker can then move his eyes to enter the next character. When the speaker mistypes or hears feedback indicating an incorrect gesture recognition, he can wink his left eye (if the speaker cannot wink, an alternative is to close both eyes for at least two seconds); this gesture removes the last character from the current sequence. When the speaker finishes a sequence for an entire word, he can wink his right eye (or alternatively look center for at least two seconds) to indicate the end of the word; then the system will speak aloud the first word prediction based on the entire series of eye gestures. The speaker can wink his right eye again to confirm this prediction, or perform a *look right* gesture to hear the next prediction. After a word has been confirmed, it is added to the sentence being constructed (Figure 7d). After the speaker confirms the last word of the sentence, he can wink his right eye again to confirm the end of the sentence, and the system will play the whole sentence aloud.

When a tripod or phone stand is available, speakers could also choose to use GazeSpeak in the front-facing mode with minimum help from the caregivers. The speaker sees the screen, and the interface is similar to the interpreter interface in Figure 7. The major difference happens when the speaker indicates the end of the word. In addition to speaking aloud the first word, the system turns the first prediction red in order to visually indicate the current focus. The speaker can still perform a *look right* gesture to hear the next prediction, and see the visual focus switch to that prediction.

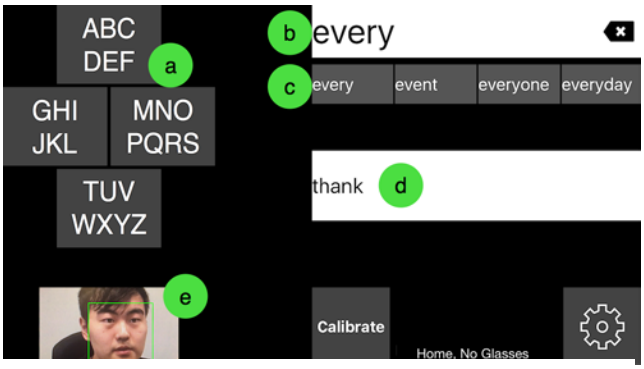

Figure 7. Interpreter interface: (a) four-key keyboard; (b) the word input box shows top prediction of characters entered in the current sequence; (c) top four word predictions; (d) the sentence box shows prior words in the communication; (e) camera preview with a green face detection box.

Interpreter Interface

With information displayed on the phone screen, the interpreter can speed up the communication process. The input box (Figure 7b) shows the most likely word based on the current gesture sequence, and its length also indicates the number of eye gestures performed so far in the current sequence. When using low-tech e-tran boards, the ability to make guesses about likely words based on partial information is an important part of speeding up communication throughput; we designed GazeSpeak to support this existing communication practice. Even before the speaker finishes a word, if the interpreter sees a likely word prediction based on her knowledge of context, she can either say the word aloud or tap the prediction box to let GazeSpeak play the word aloud. The speaker can confirm or reject the prediction, either using gestures recognized by GazeSpeak (right-winking to confirm or left-winking to reject) or using mutually-agreed upon conventions common in e-tran communication (e.g., using other motions depending on the speaker's range of mobility, such as nods, eyebrow raises, etc.). With a confirmation, the interpreter can then long-press the word to add it to the sentence box, and the speaker can proceed to begin a gesture sequence for the next word in the sentence. With a rejection of the prediction, the interpreter should let the speaker continue the eye gestures to complete the word. Once the speaker indicates the end of the sentence, the interpreter should confirm the sentence with the speaker by saying the sentence aloud or touching the sentence box to let GazeSpeak play it.

There is also a manual mode that can be used when GazeSpeak has a hard time recognizing the speaker's eyes; for example, if an oxygen mask or other medical equipment covers the speaker's face, when the eye movement capability is very limited, or when the speaker wears highly reflective glasses. In this case, the interpreter can note the eye gestures of the speaker, and touch the corresponding keys on the four-key keyboard. This manual mode still allows the interpreter to take advantage of GazeSpeak's predictive text engine to quickly decode what the speaker would like to say and to keep track of the words the speaker said, and still reduces the number of eye gestures required per character as compared to using an e-tran board.

USABILITY STUDY

We conducted a lab-based usability study to understand the relative performance, usability, and user preference for different modes of gesture decoding, comparing GazeSpeak to an e-tran board. Because ALS is a low-incidence disease (impacting 1 in 50,000 people [26]), recruiting a sufficient number of PALS to participate in a controlled study was impractical; additionally, the fatigue associated with ALS would make providing data for a controlled experiment such as this (which lasted nearly an hour in total) prohibitive for many potential participants. While we recognize that using able-bodied participants reduces the ecological validity of our study, it was a necessary tradeoff for being able to gather a sufficient amount of systematic data to evaluate GazeTalk's baseline performance characteristics. We also conducted shorter, less formal testing sessions of GazeTalk with PALS and their caregivers, which we discuss in the next section.

Participants

For this evaluation, we recruited pairs of able-bodied participants from our organization; participants did not have prior experience with e-tran boards. Performing a lab study with able-bodied users allowed us to supplement the data possible to obtain from PALS with a larger and more controlled experiment; further, these participants provide a realistic insight into the learnability and usability of the system to first-time, novice adopters of the technology in either the interpreter or speaker role. Partners in our study were not necessarily acquainted with each other prior to the study; a close relationship might facilitate better guessing and speed input. However, in some situations e-tran may be used by a visitor or other medical professional who would be less familiar with the speaker's context; our study better resembles that scenario.

We recruited participants through email lists within our organization, and paid each participant $15 for a one-hour session. Interested participants completed a brief questionnaire about basic demographics and screening for e-tran board experience (which would be a disqualifier). From the eligible pool, we randomly selected twenty-four participants for a total of twelve pairs; two pairs did not show up for their assigned timeslot. The remaining twenty participants completed the study and reported an average age of 32 years (min 20, max 52); fourteen were male and six were female. When pairs arrived at the study session, they were randomly assigned to either the speaker or interpreter role. Among the ten speakers, five had a normal (uncorrected) vision, two wore contact lenses, and three wore glasses.

Procedure

We employed a within-subjects design to examine the input speed, usability, and user preference among three input

methods: an e-tran board, GazeSpeak's default operation style, and GazeSpeak's front-facing mode. We used a Latin Square design to counterbalance the ordering of the three input methods across participant pairs. For sentences to be entered during testing, we randomly picked a set of eighteen five-word-long sentences (29-31 characters each) from the Mackenzie and Sourkeroff phrase sets commonly used for evaluating text entry techniques [19] to use as our testing corpus.

Participation began with a brief introduction of the purpose of the study. We then randomly assigned the speaker and interpreter roles to the members of a pair (these roles were held constant for all three input methods). Participants sat face to face (in chairs set 18-inches apart).

For each of the three conditions, we presented a tutorial on how to use the communication method, and let the pair practice until they felt comfortable using the method to communicate a two-word example utterance (e.g., "hello world"). We then privately showed the speaker a sentence from the testing corpus, and instructed them to communicate that sentence as quickly and accurately as possible to their partner without speaking, using only the current communication method. We then started a timer and stopped the timer when the interpreter correctly decoded the sentence. This procedure continued until either six sentences had been successfully communicated or ten minutes had elapsed, at which point we stopped the session in order to avoid excessive fatigue. Participants then completed a short questionnaire providing feedback about their experience using that communication method, and took a short break if they felt fatigued. After repeating this procedure for all three communication methods, participants completed a final questionnaire ranking their preferences among all three methods.

Results

For each 10-minute session using a given interface, we prepared six sentences for the speaker to communicate to the interpreter. In the two GazeSpeak conditions, all pairs successfully communicated all six phrases. However, in the e-tran condition, pairs completed an average of 4 phrases.

	Mean	Median	Min	Max	Stdev
E-tran	143.4	122.5	72	298	57.4
GazeSpeak	80.9	77.5	51	132	18.8
GazeSpeak Front-Facing Mode	77.1	76	56	120	11.5

Table 2. The time (in seconds) spent to complete a sentence using each input method.

Participants, on average, spent 137.8 seconds to complete a sentence using the e-tran board, 80.9 seconds using GazeSpeak's default mode and 77.1 seconds using GazeSpeak's front-facing mode. A one-way repeated measures ANOVA indicates that mean input time differed significantly between input methods ($F(1.067, 9.602) = 21.032$, $p = 0.002$). Follow-up pairwise paired-samples t-tests show that both modes of GazeSpeak bring a statistically significant reduction in input time as compared to the e-tran board (default mode vs. e-tran: $t(9) = 4.136$, $p = 0.003$, and front-facing mode vs. e-tran: $t(9) = 3.983$, $p = 0.003$).

	E-tran		GazeSpeak		Front-Facing	
	S	I	S	I	S	I
It is unnecessarily complex	2.9	2.4	1.7	1.7	1.6	2.1
I feel confident to use it	3.0	3.3	4.3	4.3	4.3	3.8

Table 3. Speaker(S) and Interpreter(I)'s average agreement level (1=strongly disagree, 5=strongly agree) on each statement.

The questionnaires completed after each condition asked participants to indicate their level of agreement with several statements about that condition's input method on a 5-point Likert scale (Table 3). The Friedman test indicates there were statistically significant differences in perceived complexity between input methods ($\chi^2 = 8.5$, $p = 0.014$). A Wilcoxon signed-rank test shows that GazeSpeak's default mode has a statistically significant reduction in perceived complexity over the e-tran board ($Z = -2.834$, $p = 0.005$). There are no significant differences between GazeSpeak's default and front-facing mode ($Z = -0.612$, $p = 0.541$), or between e-tran board and GazeSpeak front-facing mode ($Z = -1.794$, $p = 0.073$). Part of the e-tran board's complexity may be due to its diagonal directions; for instance, P21 (speaker) and other speakers made comments such as "*looking diagonal was difficult.*" A Friedman test also indicates there were statistically significant differences in perceived confidence in correctly communicating using each input methods ($\chi^2 = 7.4$, $p = 0.024$). Participants reported feeling more confident using any mode of GazeSpeak than e-tran (default mode vs. e-tran: $Z = -2.871$, $p = 0.004$, and front-facing mode vs. e-tran: $Z = -1.970$, $p = 0.049$).

	E-tran		GazeSpeak		Front-Facing	
	S	I	S	I	S	I
Mentally Demanding	4.8	4.5	3.3	2.5	3.4	2.6
Task Difficulty	3.9	3.5	2.4	2.4	3.1	1.9
Feel Stressed/Discouraged	3.9	2.2	2.1	1.9	2.4	1.8

Table 4. Speakers' (S) and Interpreters' (I) average ratings (1=Very low, 7=Very high) for NASA TLX items.

Our study questionnaires also included items from the NASA TLX scale [8], reported on a 7-point scale (Table 4). Participants rated the e-tran board as more mentally demanding than both modes of GazeSpeak (Friedman test: $\chi^2 = 13.5$, $p = 0.001$, default mode vs. e-tran: $Z = -3.471$, $p = 0.001$, and front-facing mode vs. e-tran: $Z = -2.774$, $p = 0.006$). P12 (interpreter) described aspects of using the e-tran board that were challenging, such as "*to keep track of what had been spelled out so far, as well as which eye movements were to select a letter vs looking at the board.*" P5 (speaker) noted that using GazeSpeak makes it "*very*

simple to memorize the gestures, resulting in much less confusion and mistakes [than with e-tran]." Compare to the e-tran board, both modes of GazeSpeak were considered less difficult to use (Friedman test: $\chi^2 = 7.0$, $p = 0.03$, default mode vs. e-tran: $Z = -2.560$, $p = 0.01$, and front-facing mode vs. e-tran: $Z = -1.930$, $p = 0.05$). P18 (interpreter) explained this preference, "*I liked that the technology on the phone was doing the hard work whereas I was only responsible for keeping track of when the speaker was done with spelling a word. It was much easier for me as an interpreter with the technology than without the technology.*" Using GazeSpeak in front-facing mode made participants feel less stressed or discouraged than using the e-tran board ($Z = -1.951$, $p = 0.05$). For speakers, using GazeSpeak's default mode significantly reduced feelings of stress and discouragement compared to using the e-tran board ($Z = -2.328$, $p = 0.02$). As P13 (speaker) described, "*It [GazeSpeak] was way simpler than the first one [e-tran]. Only having to do one of four movements for a letter was much easier to indicate and I didn't feel like I had to emphasize the movements as much. The instant [audio] feedback on which direction I went was also helpful.*"

	E-tran		GazeSpeak		Front-Facing	
	S	**I**	**S**	**I**	**S**	**I**
Average Rank	2.9	2.8	1.6	2	1.5	1.2

Table 5. Speaker(S) and Interpreter(I)'s average rank (1=Best, 3=Worst) of each input method.

After completing all sessions, participants ranked all three input methods based on their experience (Table 5). Overall, participants preferred GazeSpeak to the e-tran board; the e-tran board was not selected by any participants as their favorite input method. A one-sample chi-square test showed a significant difference in the rankings of preferences for each system, with the e-tran board most likely to be ranked least favorite by all participants, regardless of role ($\chi^2(1, N = 20) = 9.80$, $p = 0.002$). P8 (interpreter) articulated the reasoning behind this preference: "*The smartphone app was better at interpreting the eye gestures and recounting the words.*"

Interpreters favored GazeSpeak's front-facing mode as their top-ranked interface (likely because this mode resulted in less active interpreter involvement), $\chi^2(1, N = 10) = 6.40$, $p = 0.01$, whereas participants in the speaker role were not significantly more likely to have a clear preference between front-facing or rear-facing GazeSpeak, $\chi^2(1, N = 10) = 0.40$, $p = 0.53$. However, preferences for front and rear-facing use will likely vary with in-context use from the target user groups, as issues such as communication urgency, speed, autonomy, context, and others are likely quite different for PALS and their caregivers than for able-bodied users testing GazeSpeak in a lab setting (for example, while interpreters in our lab study enjoyed their reduced responsibilities with the front-facing UI, real-world communication partners of PALS may prefer a more active role in helping their partner communicate more quickly).

The key learnings from this study are that GazeSpeak, in both configurations, was quick for participants to learn to use, and resulted in faster communication and greater user preference than an e-tran board.

FEEDBACK FROM PALS AND THEIR CAREGIVERS

We included PALS and their communication partners throughout our design process. Before we started the development, we did a formative survey to identify end-user needs. During different stages of app development, we invited PALS and partners to our lab and did iterative demos/informal testing. When the GazeSpeak prototype was mature, we visited seven PALS and their caregivers to demonstrate GazeSpeak and obtain their feedback. In addition to demonstrating the system and giving a tutorial, we let them try GazeSpeak. One of the PALS, using GazeSpeak's default mode with his caregiver, completed one of our 5-word (30-character) testing sentences in 62 seconds, after only a brief tutorial. He also believed that communicating with his caregiver in his real life will be even faster than typing testing sentences: "*I will say the same thing over and over, so my caregiver can predict the words really fast.*" Caregivers liked the general concept: "*I love the phone technology; I just think that would be so slick.*" Caregivers envisioned GazeSpeak could be useful outdoors because they would not have to use "*the big Tobii*"; they also noted it would be useful in the car, as Tobii and its mount are so bulky that one PALS's spouse noted she does not allow her husband to use it in the car due to concerns that it could be a safety hazard in the event of a crash.

These informal tests with PALS also identified important opportunities for improvement, revealing usability issues not uncovered in our lab study with able-bodied participants. Two of the seven PALS wore oxygen masks, which confused GazeSpeak's face detector, resulting in failed eye gesture recognition. The use of manual mode, mentioned in the end of *Interpreter Interface* section, is one possible solution that could be used in this circumstance; however, a

Figure 8. A PALS using GazeSpeak with his caregiver.

more desirable solution would be to train a detector for faces with an oxygen mask, or train a detector for eyes in the absence of other facial landmarks. Also, testing with PALS revealed variation in the speed with which they could move their eyes; four of the PALS with more advanced disease progression could not move their eyes as quickly as our automated calibration sequence demanded. Consequently, we improved GazeSpeak by (1) slowing down the calibration sequence to allow more time to perform each gesture and (2) adding a "manual calibration" option in which the caregiver can control the pacing of capturing each calibration image before moving on to the next one. Caregivers requested that the screen show more predictions if an eye gesture sequence represents more than four words, especially when the caregivers extend the dictionary to include names and other custom vocabulary. One PALS commented: "*Having more options could slow the caregiver down, but the fact that my caregiver and I work independently could mean it will be faster [overall]*." Automatic feedback on the quality of an acquired calibration would also give caregivers extra confidence in using the system.

DISCUSSION

We created GazeSpeak, a low-cost and high-tech AAC solution for people who rely on eye gaze communication, as an alternative to current high-cost/high-tech and low-cost/low-tech solutions. Compared to a low-tech e-tran board, GazeSpeak is smaller (a phone is about 6% of a typical board's size) and lighter (about 30% of a typical board's weight). Compared to a high-end IR-based eye-tracking system, GazeSpeak is also smaller and lighter, it does not require a fixed mounting and is therefore more portable, and it is more robust to varied lighting sources, since IR light such as bright sunlight does not interfere with a smartphone's camera. To reduce the difficulty of error correction, GazeSpeak does not only allow interpreters to manually recognize mutually-agreed upon conventions common in e-tran communication, but also allows the speaker to correct errors by recognizing a left wink or closing both eyes for a period of time; GazeSpeak also reduces the cognitive complexity of communication as compared to e-tran board use: for the speaker, letter entry is simplified from two gestures per character in e-tran to one in GazeSpeak, and for the interpreter, GazeSpeak improves upon e-tran by showing predictions of likely words and by keeping track of prior words spoken in the current utterance. Our usability study showed that GazeSpeak significantly improves communication speed compared to an e-tran board.

To make GazeSpeak even more robust and suitable to a wider range of end-user needs and abilities, we include a manual input mode for use when eye gestures cannot be recognized, and a front-facing mode that can be used when a phone stand is available. The front-facing mode allows GazeSpeak to be used without the help of an interpreter, offering the possibility of a lower-cost, but lower-fidelity and less general-purpose, alternative to commercial eye-tracking solutions if someone cannot afford one. In following principles of ability-based design [36], GazeSpeak gracefully adds/degrades functionality depending on a user's capabilities, such as allowing backspacing if left-winking is possible, offering dual-eye closing as an alternative to winking, and still functioning without the backspace capability at all if the speaker can perform neither of those gestures. We have also implemented a head-tracking mode that can be used as an alternative to eye gestures depending on the capabilities and preferences of the speaker.

Our feedback sessions with PALS and their communication partners found areas for further improvement of GazeSpeak. To improve eye gesture recognition availability, we can improve the face detection and eye detection algorithms to have robustness in cases of face-worn medical equipment. We can also further improve the word prediction by automatically updating the weight of prediction from daily usage, and making word prediction based on already-typed words and other contextual information (e.g., location information from the phone's GPS [9]), and offering a larger n-best list for interpreters to select from.

While our evaluations gave insight into usability and performance for novice users, additional types of evaluation can offer additional value. For example, longer-term studies of use would give insight into the relative learning curves of GazeSpeak versus other AAC methods, and would allow characterization of expert-level performance. More formal, long-term data collection from PALS and their communication partners, while logistically complex, is particularly important for understanding in situ use of this new type of AAC. GazeSpeak may also hold value for other audiences besides PALS (e.g. people with cerebral palsy, spinal cord injury, stroke, traumatic brain injury, etc.). Because PALS can have different eye movement pattern than other motor-impaired users, additional testing may be necessary to modify GazeSpeak to support such users.

CONCLUSION

In this paper, we introduce GazeSpeak, a smartphone app to facilitate communication for people with motor disabilities like ALS. In real time, GazeSpeak robustly recognizes eye gestures and decodes them to words. Our user studies show that GazeSpeak surpasses e-tran boards (a commonly-used low-tech solution) in both communication speed and usability, with a low rate of wrong recognition. Feedback from PALS and their communication partners affirmed the value of offering a portable, low-cost technology to supplement IR-based eye tracking systems in situations where they are impractical to use (or are unaffordable to purchase).

ACKNOWLEDGEMENTS

We thank the MSR Enable team for their support and insight and all of the PALS and their communication partners who participated in our survey and testing.

REFERENCES

1. Javier S. Agustin, Henrik Skovsgaard, John P. Hansen, and Dan W. Hansen. 2009. Low-cost Gaze Interaction: Ready to Deliver the Promises. In *Conference on Human Factors in Computing (CHI)*, 4453–4458. http://doi.org/10.1145/1520340.1520682
2. Monica Anderson. 2015. Technology Device Ownership: 2015. Retrieved September 15, 2016 from http://www.pewinternet.org/2015/10/29/technology-device-ownership-2015/
3. Gary Becker. 1996. Vocal Eyes Becker Communication System. Retrieved September 15, 2016 from http://jasonbecker.com/eye_communication.html
4. Pradipta Biswas and Pat Langdon. 2011. A new input system for disabled users involving eye gaze tracker and scanning interface. *Journal of Assistive Technologies* 5, 2: 58–66. http://doi.org/10.1108/17549451111149269
5. Gary Bradski. 2000. The OpenCV Library. *Doctor Dobbs Journal* 25: 120–126.
6. Mark Davies. 2008. The Corpus of Contemporary American English: 520 million words, 1990-present. Retrieved from http://corpus.byu.edu/coca/
7. Kristen Grauman, Margrit Betke, Jonathan Lombardi, James Gips, and Gary R. Bradski. 2003. Communication via eye blinks and eyebrow raises: Video-based human-computer interfaces. *Universal Access in the Information Society* 2, 4: 359–373. http://doi.org/10.1007/s10209-003-0062-x
8. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology* 52, C: 139–183. http://doi.org/10.1016/S0166-4115(08)62386-9
9. Shaun K. Kane, Barbara Linam-Church, Kyle Althoff, and Denise McCall. 2012. What We Talk About: Designing a Context-Aware Communication Tool for People with Aphasia. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility - ASSETS '12*, 49. http://doi.org/10.1145/2384916.2384926
10. Vahid Kazemi and Josephine Sullivan. 2014. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1867–1874. http://doi.org/10.1109/CVPR.2014.241
11. Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10: 1755–1758.
12. Kyle Krafka, Aditya Khosla, Petr Kellnhofer, et al. 2016. Eye tracking for Everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
13. 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*, 241–244. http://doi.org/10.1145/2168556.2168605
14. Rene De La Briandais. 1959. File searching using variable length keys. In *Papers presented at the western joint computer conference*, 295–298.
15. Chris Lankford. 2000. Effective eye-gaze input into windows. In *Proceedings of the 2000 Symposium on Eye Tracking Research & Applications*, 23–27. http://doi.org/10.1145/355017.355021
16. Erich Leo Lehmann and George Casella. 2006. *Theory of point estimation*. Springer Science & Business Media.
17. Low Tech Solutions. 2016. E-tran Board. Retrieved September 15, 2016 from http://store.lowtechsolutions.org/e-tran-board/
18. LusoVu. 2016. EyeSpeak. Retrieved September 15, 2016 from http://www.myeyespeak.com/eyespeak/
19. Scott MacKenzie and William Soukoreff. 2003. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, 754–755. http://doi.org/10.1145/765968.765971
20. Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 357. http://doi.org/10.1145/1518701.1518758
21. Microsoft Research. 2016. Hands-Free Keyboard. Retrieved September 15, 2016 from https://www.microsoft.com/en-us/research/project/hands-free-keyboard/
22. Steven Nowlan, Ali Ebrahimi, David Richard Whaley, Pierre Demartines, Sreeram Balakrishnan, and Sheridan Rawlins. 2001. Data entry apparatus having a limited number of character keys and method. Retrieved from https://www.google.com/patents/US6204848
23. Nuance. 2016. T9 Text Input. Retrieved September 15, 2016 from http://www.nuance.com/for-business/by-product/t9/index.htm
24. Alexandra Papoutsaki, Patsorn Sangkloy, James Laskey, Nediyana Daskalova, Jeff Huang, and James Hays. 2016. WebGazer: Scalable Webcam Eye Tracking Using User Interactions. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 3839–3845.

25. Sarah Marie Swift. 2012. Low-Tech, Eye-Movement-Accessible AAC and Typical Adults.

26. The ALS Association. 2016. Facts You Should Know. Retrieved September 15, 2016 from http://www.alsa.org/about-als/facts-you-should-know.html

27. TheEyeTribe. 2016. The Eye Tribe. Retrieved September 15, 2016 from https://theeyetribe.com/

28. Tobii. 2016. Tobii Dynavox Webshop. Retrieved September 15, 2016 from https://www.tobiiati-webshop.com/

29. Tobii. 2016. Communicator 5 - Tobii Dynavox. Retrieved September 15, 2016 from http://www.tobiidynavox.com/communicator5/

30. Tobii. 2016. Sono Lexis - Tobii Dynavox. Retrieved September 15, 2016 from http://www.tobiidynavox.com/sono-lexis/

31. Vytautas Vaitukaitis and Andreas Bulling. 2012. Eye gesture recognition on portable devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, 711. http://doi.org/10.1145/2370216.2370370

32. David Vitter. 2015. S.768 - Steve Gleason Act of 2015. Retrieved September 15, 2016 from https://www.congress.gov/bill/114th-congress/senate-bill/768

33. Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4: 600–612. http://doi.org/10.1109/TIP.2003.819861

34. David J. Ward, Alan F. Blackwell, and David J. C. MacKay. 2002. Dasher: A Gesture-Driven Data Entry Interface for Mobile Computing. *Human-Computer Interaction* 17, 2/3: 199–228.

35. Craig Watman, David Austin, Nick Barnes, Gary Overett, and Simon Thompson. 2004. Fast sum of absolute differences visual landmark detector. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, 4827–4832 Vol.5. http://doi.org/10.1109/ROBOT.2004.1302482

36. Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Transactions on Accessible Computing* 3, 3: 1–27. http://doi.org/10.1145/1952383.1952384

37. 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*, 11. http://doi.org/10.1145/1344471.1344475