Eyelid Gestures on Mobile Devices for People with Motor Impairments

MINGMING FAN*, School of Information, Rochester Institute of Technology, USA
ZHEN LI*, Department of Computer Science, University of Toronto, Canada
FRANKLIN MINGZHE LI*, Department of Computer Science, University of Toronto, Canada

Eye-based interactions for people with motor impairments have often used clunky or specialized equipment (e.g., eye-trackers with non-mobile computers) and primarily focused on gaze and blinks. However, two eyelids can open and close for different duration in different orders to form various eyelid gestures. We take a first step to design, detect, and evaluate a set of eyelid gestures for people with motor impairments on mobile devices. We present an algorithm to detect nine eyelid gestures on smartphones in real-time and evaluate it with twelve able-bodied people and four people with severe motor impairments in two studies. The results of the study with people with motor-impairments show that the algorithm can detect the gestures with .76 and .69 overall accuracy in user-dependent and user-independent evaluations. Moreover, we design and evaluate a gesture mapping scheme allowing for navigating mobile applications only using eyelid gestures. Finally, we present recommendations for designing and using eyelid gestures for people with motor impairments.

CCS Concepts: • **Human-centered computing → Interaction techniques**; **Accessibility technologies**.

Additional Key Words and Phrases: eyelid gestures, people with motor impairments, mobile interaction

ACM Reference Format:
Mingming Fan, Zhen Li, and Franklin Mingzhe Li. 2020. Eyelid Gestures on Mobile Devices for People with Motor Impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20), October 26–28, 2020, Virtual Event, Greece.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3373625.3416987

1 INTRODUCTION

Fifteen percent of people in the US have difficulties with their physical functioning, among whom almost half find it very difficult or impossible to walk unassisted for a quarter-mile [11]. Although specialized devices, such as eye-trackers [21, 29, 31, 33], brain-computer interfaces [3], and mechanical devices (e.g., joysticks [35, 39, 40], trackballs [38], mouse pieces [9, 34]), have been investigated to assist people with motor impairments, such devices are often clunky, intrusive, expensive, and limited in accuracy and functions (e.g., text entry). In contrast, smartphones become increasingly ubiquitous, powerful, and can be beneficial to people with motor impairments [24].

Rich sensors on smartphones have enabled new opportunities to assist people with motor impairments. For example, motion sensors and touch screens have been used to recognize physical activities [2] and diagnose and quantify motor ability [1, 5, 14, 28]; microphones allow for using speech to enter texts [32] and issue commands [10, 27]. Although another sensor—camera—has been explored to assist people with motor impairments to enter text [6, 26, 41], issue gesture

*Authors contributed equally.

ASSETS '20, October 26–28, 2020, Virtual Event, Greece

ACM ISBN 978-1-4503-7103-2/20/10…$15.00
https://doi.org/10.1145/3373625.3416987

commands [13, 17, 29], and navigate a wheelchair [4, 8, 33, 42] , such research has primarily focused on utilizing gaze (e.g., eyeball movements) [13, 17, 26, 29, 33, 41, 43] and blinks [12, 16, 22, 43] for interactions. However, human's two eyelids can be in open and close *states* for short and long *periods* in concurrent and sequential *orders* to form a rich set of eyelid gestures, which could extend and enrich existing eye-based interactions. In this work, we make an initial exploration into the design space of eyelid gestures on mobile devices for people with motor impairments.

We first introduce a taxonomy to describe and construct potential eyelid gestures based on four primitive eyelid states. Although some eyelid gestures, such as winks, were proposed for hands-free interaction [15], our work explores a richer set of eyelid gestures and is the first to present an algorithm to recognize them on a smartphone in real-time. Moreover, we evaluated the performance of the algorithm in two user studies with people without and with motor impairments. In the first study, twelve able-bodied participants performed the nine eyelid gestures in two indoor environments and different postures. The overall accuracy of user-dependent and user-independent models was .76 and .68 respectively, which shows that the algorithm was robust to differences in environments and postures. We then conducted the second study in which four participants with severe motor impairments performed the same set of gestures. The overall accuracy of user-dependent and user-independent models was .76 and .69 respectively. Furthermore, we designed a mapping scheme to allow users to navigate mobile applications only using eyelid gestures. We asked the participants with severe motor impairments to complete a set of navigation tasks only using eyelid gestures. Results show that they perceived the eyelid gestures were easy to learn and the mapping was intuitive. They further reported how the eyelid gestures and the mapping scheme can be further improved. Finally, we present design recommendations for using eyelid gestures for people with motor impairments and discuss the limitations and future research directions.

2 EYELID GESTURE DESIGN AND RECOGNITION

2.1 Design

Eyelid states refer to the states in which two eyelids can be and have four possible values: *both eyelids open*, *both eyelids close*, *only the right eyelid close*, and *only the left eyelid close*. Technically, an eyelid can also be in a half-closed state (e.g., squinting). However, sustaining eyelids in a half-close state can cause them to twitch or cramp [15]. Moreover, our investigation found that it is still challenging to robustly recognize half-close states with current technology. Thus, as an initial exploration into this design space, we focus on the four states when constructing eyelid gestures. Because the "both eyelids open" state is the most common state when humans are awake, we use it as the *gesture delimiter* to label the start and end of an eyelid gesture.

In addition to the four eyelid states, humans can control the *duration* of an eyelid state [15]. As it can hard to memorize the exact duration of a state, we discretize duration into two levels—*short* and *long*. *Short* duration refers to the time that it takes to *intentionally* close an eyelid (e.g., longer than a spontaneous blink (50 - 145 ms) [36]) and open it immediately afterward. *Long* duration is closing an eyelid, sustaining it for some time, and then opening it. As users may have different preferences for holding the eyelids in a state, it is ideal to allow them to decide on their preferred holding duration as long as they keep it consistent. For simplicity, in this work, users are instructed to count a fixed number of numbers (e.g., three) by heart while holding eyelids in a state.

By controlling the eyelid states and their duration, we could construct an infinite number of eyelid gestures with one or more eyelid states between the gesture delimiter. As an initial step toward exploring this vast design space, we focused on recognizing nine relatively simple eyelid gestures, which consist of only one or two eyelid states between the gesture delimiter. Fig. 1 shows these nine eyelid gestures and their abbreviations.

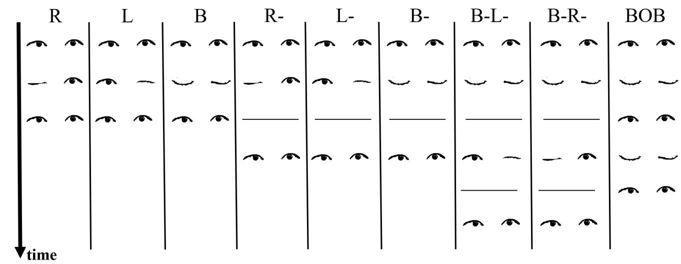

Fig. 1. The descriptions and abbreviations of the nine eyelid gestures that our algorithm detects in real-time. Each *letter* in a gesture *abbr.* depicts the gesture's key eyelid states between the common start and end states (i.e., "both eyelids open"). The *dash line* indicates holding the eyelid(s) in the state that it follows. For example, *'B-R-'* represents the gesture that starts from "both eyelids open", transitions to "Both eyelids close", sustains in the state for some time (-), transitions to "only the Right eyelid close", sustains in the state for some time (-), and ends at "both eyelids open". Similarly, the "double blink" gesture *'BOB'* includes "Both eyelids close", "Both eyelids Open", and "Both eyelids close" between the common start and end states (i.e., delimiter).

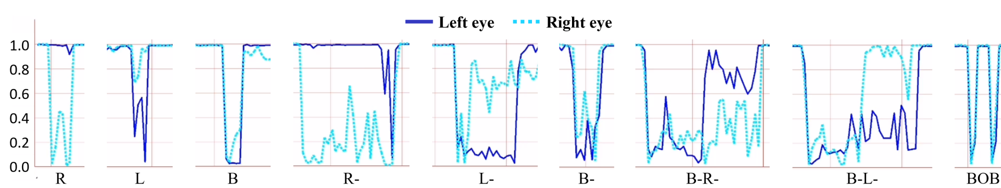

Fig. 2. The probabilities of two eyes being open when a user performs each of the nine eyelid gestures. The blue (solid) and cyan (dashed) lines represent the probabilities of the left and right eye being open respectively.

2.2 Recognition Algorithm

Our algorithm is implemented on Samsung S7 running Android OS 8.0. It first obtains images from the front-camera (30 frames per second) with 640 x 480 resolution and leverages Google Mobile Vision API to generate a stream of probability pairs of each eye being open (P_L, P_R) [20]. The details of how the API estimates probability can be found in [20]. Fig. 2 shows some examples of the probabilities of two eyes being open in the nine eyelid gestures performed by a user. Notice that when the user closes the right or left eye, the probability of this eye being open is not necessarily the same, and the probability of the other eye being open might also drop at the same time. It suggests that the probability estimation of the API [20] is noisy, and there are variations in the probability estimations even when the same user performs the same gesture.

To cope with the variations in probability estimations, our algorithm incorporates an *eyelid-state* Support Vector Machine (SVM) classifier to classify an input pair (P_L, P_R) into two states: *open* (O) if both eyes are open and *close* (C) if any eye is closed. Because the "both eyes open" (O) state is used as the gesture delimiter, the algorithm then segments the stream of probability pairs between the delimiter. The algorithm then computes the duration of an segment and filters it out if its duration is too short because extremely short segments are likely caused by spontaneous blinks (50 - 145 ms [36]) or noises in probability estimations. We tested different thresholds for duration from 150 to 300 ms and adopted 220 ms for its best performance. Next, the duration of the segment is fed into another SVM classifier, which further distinguishes if it is a *short-duration* or *long-duration* gesture (Fig. 1). The algorithm then re-samples the sequence of probability pairs (P_L, P_R) in the segment to ensure all segments contain the same number of probability pairs (50 and 100 samples for short and long gestures respectively). Next, the re-sampled same-length vector is fed into the corresponding *short-duration SVM classifier* or a *long-duration SVM classifier*. Finally, the short-duration classifier detects whether the segment is *R*, *L*, *B* or *BOB*; and the long-duration classifier detects whether the segment is *R-*, *L-*, *B-*, *B-R-*, or *B-L-*. All SVM classifiers are implemented using scikit-learn library with the Radial Basis Function kernel and default parameters [25]. For greater reproducibility, we make our code available here[1].

3 STUDY WITH PEOPLE WITHOUT MOTOR IMPAIRMENTS

We conducted the first study to understand how well our algorithm recognizes eyelid-gestures on a mobile device for able-bodied people before moving to people with motor impairments.

3.1 Participants

We recruited 12 able-bodied participants aged between 23 and 35 ($M = 26$, $SD = 4$, five males and seven females) to participate in the study. Their eye colors include brown (11) and amber (1). Seven wore glasses, one wore contact lenses, and four did not wear glasses or contact lenses. No one worn false eyelashes. The study lasted half an hour, and participants were compensated with $15.

3.2 Procedure

We used a Samsung S7 Android phone as the testing device to run the eyelid gesture recognition evaluation app (Fig. 3) in real-time. To increase evaluation validity, we collected training and testing data in two different offices. We first collected training data by asking participants to keep their eyelids in each of the four eyelid states and then perform each of the nine eyelid gestures five times following the instructions in the app while *sitting* at a desk and holding the phone in their preferred hand in one office. We then collected testing data by asking them to perform each eyelid gesture another five times while *standing* in *another* office room and holding the phone in their preferred hand. The differences in physical environments and postures increased variations between training and testing data. Similarly, the variations in ways how they held the phone in their preferred hands also introduced variations between training and testing data.

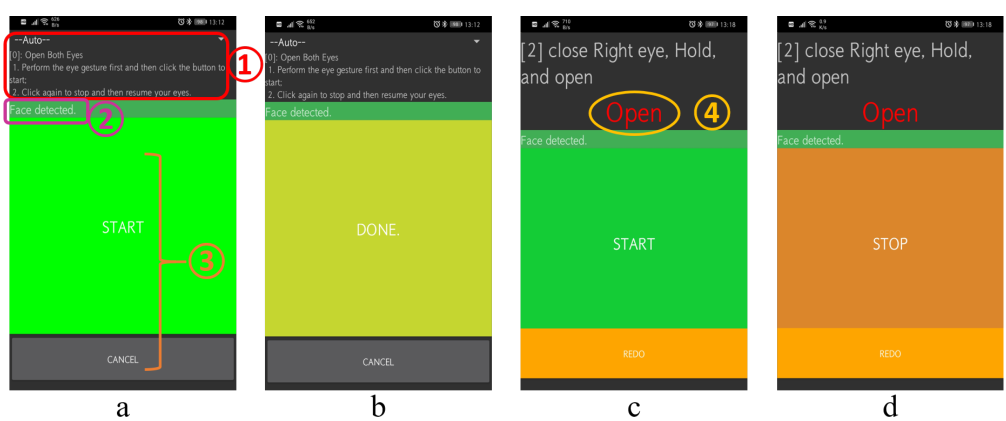

Fig. 3. (a)-(d) present the data collection UIs for eyelid states (a, b) and for eyelid gestures (c, d). ① shows the name of eyelid states or eyelid gestures, ② shows the face detection result, and ③ are control buttons, such as "start", "cancel", and "redo". During eyelid gesture evaluation, detected eyelid state is shown in ④.

[1] https://github.com/mingming-fan/EyelidGesturesDetection

	R	L	B	R-	L-	B-	B-R-	B-L-	BOB
R	47	0	7	8	0	0	0	1	6
L	4	56	7	1	5	2	2	0	8
B	0	0	46	0	0	0	0	0	0
R-	3	0	0	44	0	5	3	1	2
L-	0	0	0	0	46	4	1	7	0
B-	0	0	0	0	1	45	4	2	0
B-R-	0	0	0	6	0	1	47	1	1
B-L-	0	0	0	1	6	2	1	47	0
BOB	0	1	0	0	0	0	0	0	34
N/A	6	3	0	0	2	1	2	1	9

a

	R	L	B	R-	L-	B-	B-R-	B-L-	BOB
R	47	2	7	15	0	3	3	1	7
L	2	53	6	3	13	2	0	0	10
B	0	0	45	0	0	0	0	0	0
R-	3	0	1	36	0	9	4	1	1
L-	0	1	0	0	42	4	0	6	0
B-	0	0	1	4	0	34	10	1	0
B-R-	0	0	0	0	0	2	38	1	0
B-L-	0	0	0	0	3	0	1	46	0
BOB	0	0	0	0	0	0	0	0	28
N/A	8	4	0	2	2	6	4	4	14

b

Fig. 4. Study One: The confusion matrix of user-dependent (a) and user-independent (b) evaluations respectively (columns: ground truth; rows: predictions; N/A means not recognized).

To collect data samples for each eyelid state, the evaluation app first presented a target eyelid state on the top side of the screen (Fig. 3a, b) in a random order. Participants were asked to first prepare their eyes in the state and then press the green 'START' button to start data collection at a speed of 30 frames per second. The app beeped after collecting 200 frames, and the button turned to yellow to indicate that the data collection for this eyelid state was done. The app presented another eyelid state and repeated the procedure until data samples for all four eyelid states were collected. These data were used to perform 10-cross cross-validation of the eyelid state classifier on the phone in real-time. The training process took on average 558 milliseconds.

To collect the training data for each of the nine eyelid gestures, the evaluation app presented a target gesture on the top side of the screen (Fig. 3c, d). Participants were asked to press the green 'START' button and then perform the target gesture. Upon finishing, participants pressed the 'STOP' button. The app recorded and stored the stream of eyelid states during this period. The app presented each eyelid gesture five times randomly. Thus, the app collected five samples per gesture for each participant, which was used to train the eyelid gesture classifier on the phone in real-time. The training process took on average 102 milliseconds.

To collect testing data, participants were asked to perform each eyelid gesture five more times while standing in another office room using the same app and aforementioned procedure.

3.3 Results

To evaluate the eyelid state classifier, we performed 10-fold cross validations; to evaluate the eyelid gesture classifier, we performed both *user-dependent* and *user-independent* evaluations.

3.3.1 Eyelid State Evaluation. We performed a 10-fold cross-validation on each participant's data and averaged the performance across all participants. The overall accuracy was .92 (SD = .09). The accuracy for each eyelid state was as follows: *both eyelids open* (.98), *right eyelid close* (.89), *left eyelid close* (.85), and *both eyelids close* (.96). Because *both eyes open* was the *gesture delimiter* to separate eyelid gestures, we further trained a classifier to recognize only two eyelid states by grouping all the other three states together. The average accuracy was 0.98 (SD = .02).

3.3.2 User-dependent Eyelid Gesture Evaluation. For each participant, we trained a user-dependent classifier with five samples for each gesture and tested it with another five samples. We then averaged the performance of the classifier for each gesture across all participants. The average accuracy of all gestures was .76 (SD = .19) and the average accuracy for each gesture was as follows: *L* (.93), *R* (.78), *B-R-* (.78), *B-L-* (.78), *B* (.77), *L-* (.77), *B-* (.75), *R-* (.73), and *BOB* (.57). This result suggests that user-dependent gesture classifiers were able to detect eyelid gestures when users were in different indoor environments and postures. We further computed the confusion matrix to show how gestures were misclassified in Fig. 4a. In addition, the average time it took for participants to complete each gesture was as follows: *R* (745 ms), *L* (648 ms), *B* (668 ms), *R-* (2258 ms), *L-* (2010 ms), *B-* (2432 ms), *B-L-* (4169 ms), *B-R-* (4369 ms), *BOB* (2198 ms). It shows that more complex gestures took longer to complete overall.

3.3.3 User-independent Eyelid Gesture Evaluation. To assess how well a pre-trained *user-independent* eyelid gesture classifier would work for a *new* user whose data the classifier is not trained on, we adopted a leave-one-participant-out scheme by keeping one participant's data for testing and the rest participants' data for training. The average accuracy of all gestures is .68 (SD = .17), and the average accuracy for each gesture was as follows: *L* (.88), *R* (.78), *B-L-* (.77), *B* (.75), *L-* (.7), *B-R-* (.63), *R-* (.6), *B-* (.57), and *BOB* (.47). We also computed the confusion matrix to show how gestures were misclassified in Fig. 4b. This result suggests that a pre-trained user-independent eyelid gesture classifier could be used "out-of-box" with reasonable accuracy for a user, but the performance could be improved if the classifier is trained with the user's data samples (i.e., user-dependent classifier).

4 STUDY WITH PEOPLE WITH SEVERE MOTOR IMPAIRMENTS

4.1 Participants

Although people with motor impairments are relatively small population [7, 37], we were able to recruit four people with severe motor impairments (PMI) for the study with the help of a local organization of people with disabilities. Table 1 shows participants' demographic information. One participant wore contact lens, and the rest did not wear glasses or contact lens. The study lasted roughly an hour, and each participant was compensated with $15.

Table 1. The demographic information of the people with motor impairments.

ID	Gender	Age	Motor Impairments	Hand Function	Note
P1	F	29	cervical spinal cord injury (C5)	having difficulty holding and grasping; using a ring holder stand for her phone	car accident in 2012; using a wheelchair
P2	F	32	cervical spinal cord injury (C6)	having difficulty extending and strengthening fingers; using an index finger's knuckle to touch her phone	acute myelitis in 2003; using a wheelchair
P3	M	53	cervical spinal cord injury (C5)	having no control over individual fingers; moving forearms to move hands and using a ring fingertip to touch his phone	car accident in 2004; using a wheelchair
P4	M	63	two forearms amputation	using his prosthetic arms to hold and interact with his phone	electric shock during high-voltage work in 1989

4.2 Procedure

The studies were conducted in participants' homes. Fig 5 shows the study setup. We asked participants to sit in their daily wheelchair or a chair. We positioned an Android phone (Huawei P20) on the top of a tripod and placed the tripod on their wheelchair tables or desks so that the phone was roughly 30-50 cm away from their faces and its front camera was roughly at their eye level.

We slightly modified the evaluation app (Fig. 3) to accommodate the participants' motor impairments. Instead of asking them to press 'START' and 'STOP' buttons, the app used a 10-second countdown timer to automatically trigger the start and end of each task. In cases where participants needed a pause, they simply asked the moderator to pause the task for them. The participants followed the instructions of the evaluation app to keep their eyelids in instructed eyelid states so that 200 frames were collected for each eyelid state. These data were used to evaluate the

Fig. 5. During the study, P1, P2, and P3 sat in their daily wheelchairs. P4 did not use a wheelchair and sat in a chair in front of a desk. The evaluation smartphone was mounted on the top of a tripod, which was placed on the wheelchair trays or the desk with the phone's front camera roughly at their eye levels.

	R	L	B	R-	L-	B-	B-R-	B-L-	BOB
R	15	1	0	2	0	1	0	0	4
L	0	16	0	1	1	0	0	0	0
B	0	0	19	0	0	0	0	0	0
R-	0	0	0	12	0	0	0	0	0
L-	0	0	0	0	17	0	0	2	0
B-	0	0	0	1	1	19	0	4	0
B-R-	1	0	0	2	0	0	20	2	0
B-L-	0	0	0	0	1	0	0	11	0
BOB	0	0	1	0	0	0	0	0	7
N/A	4	3	0	2	0	0	0	1	9

a

	R	L	B	R-	L-	B-	B-R-	B-L-	BOB
R	10	0	0	3	0	1	0	0	1
L	2	15	0	2	3	0	0	0	0
B	0	0	17	0	0	0	0	0	0
R-	0	0	0	11	1	0	0	0	0
L-	0	2	0	0	13	0	0	1	0
B-	3	0	2	1	1	19	2	6	1
B-R-	2	0	0	2	0	0	18	0	0
B-L-	0	2	0	0	1	0	0	11	0
BOB	0	0	1	0	0	0	0	1	11
N/A	3	1	0	1	1	0	0	1	7

b

Fig. 6. Study Two: The confusion matrix of user-dependent (a) and user-independent (b) evaluations respectively (columns: ground truth; rows: predictions; N/A means not recognized).

eyelid state classifier in a 10-fold cross-validation. Next, the participants followed the instructions of the evaluation app to perform each gesture five times, which were used as training data for user-dependent evaluation. After a break, the participants followed the same procedure to perform each gesture five times again, which were used as testing data for the user-dependent evaluation.

4.3 Results

4.3.1 Eyelid State Evaluation. We performed a 10-fold cross-validation on each participant's data and averaged the performance across all participants. The overall accuracy was .85 (SD = .15), and the accuracy for each eyelid state was as follows: *both eyelids open* (.99), *right eyelid close* (.65), *left eyelid close* (.79), and *both eyelids close* (.99). We noticed that individual differences exist. For example, P2 had trouble controlling her right eyelid and consequently had much lower accuracy for closing the right eyelid: *both eyelids open* (.997), *right eyelids close* (.02), *left eyelids close* (.57), and *both eyelids close* (1.00). When the last three eyelid states were grouped into one *close* state, the accuracy of the two-state classifier was more robust: .997 (SD = .004).

4.3.2 User-dependent Eyelid Gesture Evaluation. We performed the same user-dependent evaluation as Section 3.3.2, and the overall accuracy of all gestures was .76 (SD = .15). The accuracy for each gesture was as follows: *B-R-* (1.00), *B-* (.95), *B* (.95), *L-* (.85), *L* (.80), *R* (.75), *R-* (.60), *B-L-* (.55), and *BOB* (.35). We computed the confusion matrix (Fig. 6a) to show how gestures were misclassified. Similarly, we also computed the average time to complete each gesture: *R* (699 ms), *L* (889 ms), *B* (850 ms), *R-* (3592 ms), *L-* (3151 ms), *B-* (3722 ms), *B-L-* (6915 ms), *B-R-* (6443 ms), *BOB* (3002 ms).

4.3.3 User-independent Eyelid Gesture Evaluation. We performed the same user-independent evaluation as Section 3.3.3, and the overall accuracy was .69 (SD = .20). The accuracy of each gesture was as follows: *B-* (.95), *B-R-* (.90), *B* (.85), *L* (.75), *L-* (.65), *R-* (.55), *B-L-* (.55), *BOB* (.55), and *R* (.50). We also computed the confusion matrix (Fig. 6b) to show where the misclassifications happened.

4.4 Interacting with Mobile Apps with Eyelid Gestures

Navigating between and within mobile apps is a common task that is typically accomplished by a series of touch actions on the screen. App navigation happens at three levels: between *apps*, between *tabs/screens* in an app, and between *containers* in a tab/screen of an app. *Tab* is a common way of organizing content in an app. *Screen* is another way of organizing content, usually in the launcher. Within a *tab*, content is further organized by *containers*, often visually presented as cards.

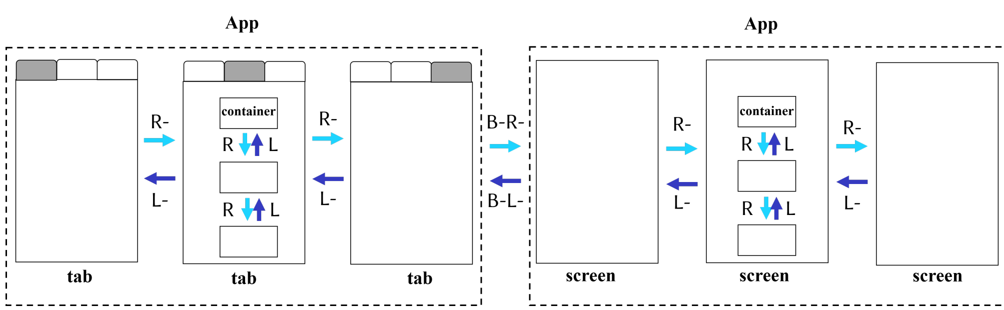

Fig. 7. The mapping scheme for navigating *apps* (*B-R-*, *B-L-*), *tabs/screens* (*R-*, *L-*), and *containers* (*R*, *L*).

To allow people with motor impairments to accomplish the three types of navigation using eyelid gestures only, we iteratively designed a mapping scheme between the gestures and the types of navigation (Fig. 7) by following two design guidelines: 1) navigation directions should be mapped consistently with the eyelid being closed (e.g., closing the *right*/*left* eyelid navigates *forward*/*backward* to the next opened app); and 2) the *complexity* of the eyelid gestures for the *lowest*-level to the *highest*-level navigation should increase. Because navigating between *apps* has the most significant overhead [18], we assign the eyelid gestures with two eyelid states (e.g., B-R-, B-L-) to this level of navigation. In addition to navigation, *BOB* is used for selecting an item.

4.4.1 Evaluation. We designed app navigation tasks to measure how well participants would be able to learn the mappings and use the eyelid gestures to accomplish various navigation tasks. The evaluation app simulated three mobile apps (APP1, APP2, APP3), which were color-coded (Fig. 8). Each app contains three tabs (TAB1, TAB2, TAB3). Each tab contains four containers numbered from 1 to 4. The outline of the container in focus is highlighted in red. The focus of attention was on the first container in TAB1 of APP1 when the evaluation started. Each participant was given a practice session which contained five navigation tasks, and the target item for each navigation was randomly generated. The app spoke out a target location using Android's text-to-speech API and also showed it on the bottom left of the UI. Each participant was asked to use eyelid gestures to navigate the focus of attention to the target item. Once the target location was reached, the next navigation task was delivered in the same manner. The practice session took on average less than 5 minutes to complete. Afterwards, the evaluation app generated another five randomized navigation tasks for participants to work on. Upon completion, participants were asked whether each gesture was a good match for completing the corresponding task (i.e., *"would that gesture be a good way to complete the navigation?"*), and whether each gesture was easy to perform (i.e. *"rate the difficulty of*

Fig. 8. (a)-(d) present the app navigation UIs. ① shows the containers, ② shows the tabs, ③ shows the current app name, and ④ shows the target item of current trial. Three types of navigation are illustrated: between containers in a tab (a, b), between tabs within an app (a, c), and between apps (a, d).

carrying out the gesture's physical action") using 7-point Likert-scale questions, which were used to elicit feedback on gesture commands (e.g., [23, 30]).

4.4.2 Subjective feedback. The average ratings of the *physical difficulty of carrying out the eyelid gestures* were as follows (the higher the value, the easier the gesture): *BOB* (7),*B-* (7), *R* (6.8), *L* (6.5), *L-* (5.5), *R-* (5.5), *B-R-* (5.5) and *B-L-* (5.5). Three out of the four PMI participants felt eyelid gestures were easy to learn and they were getting better after a brief practice. "*It was hard for me to perform some gestures because I had barely trained for these gestures other than blinking. For example, I had difficulty closing both eyelids first and then opening the left eyelid alone. I think the reason was that I had better control over the right eyelid than the left, and I had not practiced this gesture before. However, I did find it became more natural after I practiced for a couple of times.-P3*"

The rest PMI participant felt that the gestures requiring to open one eyelid at first and then both (i.e., B-L- and B-R-) are fatiguing. Instead, they proposed new eyelid gestures in the opposite direction, such *closing* one eye first and then closing the other one (e.g., L-B-, R-B-).

For those long eyelid gestures, our method required users to sustain their eyelids in a state (i.e., open or close) for a period (i.e., counting three numbers by heart). P1 expressed that she would like to be able to customize the duration, such as shortening it: "*I noticed that a long holding time did help the system distinguish my 'long' gestures from 'short' ones well. But I was a bit frustrated about the long holding time because I felt somehow it wasted time. The system could allow me to define the duration for 'short', 'long', or perhaps even 'long-long'. For example, it could ask me to perform these gestures and then learns my preferred duration for short and long gestures.*"

The average ratings of the *mappings between eyelid gestures and the levels of navigation* were as follows (the higher the value, the better the mapping): *R* (6.08), *L* (6.08), *R-* (5.83), *L-* (5.83), *B-* (5.67), *B-R-* (5.33) and *B-L-* (5.33). All four PMI participants felt the mappings were natural. In particular, participants appreciated that more complex eyelid gestures were assigned to less-frequent and high-cost commands (e.g., switching apps) while simpler eyelid gestures were assigned to relatively more-frequent and low-cost commands (e.g., switching between containers or tabs within an app).

"*As a person with a cervical spine injury, it is common for me to commit false inputs. Making apps-switching harder can prevent me from entering other apps by accident. Since I use in-app functionalities more often than switching between apps, I prefer having simple eyelid gestures associate with frequent in-app inputs, such as scrolling up to view new updates in a social media app.-P1*"

In addition, P4 felt that it would be even better to allow a user to define their own mappings in cases where the user is unable to open and close both of their eyes at the same level of ease. Furthermore, P2 and P4 wished to have an even harder-to-perform gesture as the "trigger" to activate the recognition. "*I have difficulty holding my phone stable and might have falsely triggered the recognition more often than others. For example, I may need more time to place the phone at a comfortable position before using it. During this time, I may accidentally trigger false commands to the phone. Therefore, a harder-to-perform gesture, perhaps triple winking, might be a good one for me to trigger the recognition.-P4 (with prosthetic arms)*"

We further asked participants about the *usage scenarios* of the eyelid gestures. Participants felt that eyelid gestures are handy when it is inconvenient to use their hands or fingers. "*Eyelid gestures are useful when I lie down on my stomach and rest. I have better control over my eyelids than my fingers. In fact, I can barely control my fingers. Similarly, I would like to use it when I cook or take a bathroom. Also, because it is extremely difficult for me to press buttons on a TV remote, I'd love to use the eyelid gestures to switch TV channels.-P2*"

Overall, we found that participants would like to apply eyelid gestures on various types of electronic devices (e.g., TVs, PCs, smartphones, tablets) in daily activities. Moreover, we found that participants preferred the eyelid gesture system to allow them to 1) customize the eyelid gesture holding time and the mappings between gestures and the triggered commands; 2) use a hard-to-perform gesture to activate the recognition to reduce false positives; and 3) interact with computing devices in scenarios when fingers or hands are inconvenient or unavailable to use.

5 DISCUSSION

Our user studies with people without and with motor impairments have shown that our algorithm was able to recognize their eyelid gestures on mobile devices in real-time with reasonable accuracy. This result is encouraging because they only had less than five minutes to practice the gestures. Thus, we believe our algorithm opens up a new opportunity for people with motor impairments to interact with mobile devices using eyelid gestures.

We present five *recommendations* for designing and using eyelid gestures for people with motor impairments: 1) because not all users could open and close two eyelids with the same level of ease, it is important to estimate how well a user can control each eyelid and then only use the gestures the user can comfortably perform; 2) because a pre-defined duration for holding an eyelid in a state may not work the best for everyone, it is desirable to allow for customizing the duration. Indeed, participants suggested that the system could learn their preferred duration from their gestures; 3) use the eyelid gestures with two or more eye-states (e.g., *B-R-*, *B-L-*) to trigger rare or high error-cost actions because users perceive such gestures more demanding and less likely to be falsely triggered; 4) allow users to define a "trigger" gesture to activate the gesture detection to avoid false recognition; 5) allow users to define their own gestures to enrich their interaction vocabulary.

6 LIMITATIONS AND FUTURE WORK

Although our studied included participants of different ages and motor abilities, the number of participants was still small. Future work should conduct larger scale studies with more participants who have a more diverse set of motor-impairments to better understand practices and challenges associated with using eyelid gestures.

We explored a subset of possible eyelid gestures with one or two eyelid states between the gesture delimiter. There are other gestures with two or more eyelid states, such as "winking three times consecutively." Although such gestures seem to be more complex, they might be more expressive and thus easier to remember. Future work should explore the trade-offs between the complexity and expressiveness of eyelid gestures.

We divided the duration of an eyelid state into two levels: short and long. However, more levels are possible. Indeed, a participant in Study 2 suggested "long-long" duration. Future work should study the levels of duration that users could reasonably distinguish to uncover more gestures.

We focused on two eyelid states (i.e., open and close) when constructing eyelid gestures. As is described in Section 2.1, our eyes could also be in half-closed states (e.g., squinting). Future work should explore a more diverse set of eyelid gestures including open, close, and half-closed states.

We used "both eyelids open" as the gesture delimiter as it is the default state when we are awake. However, other delimiters might enable new eyelid gestures, such as "blinking the right/left eye twice (while closing the other eye)." Thus, future work should explore other reasonable delimiters.

Our study showed that people with motor impairments preferred customizing eyelid gestures to use in different contexts and to avoid false activation of recognition. Thus, it is valuable to understand what eyelid gestures people with motor impairments would want to create and use, such as via a co-design workshop with them.

Lastly, people with motor impairments have already used gaze for text entry [43], drawing on computer screens [13], and navigating their wheelchairs [4, 8, 19, 33, 42]. Thus, future work could explore ways to combine eyelid gestures with gaze to enrich their interaction bandwidth.

7 CONCLUSION

We have presented a taxonomy to describe and construct eyelid gestures and an algorithm to detect nine eyelid gestures on smartphones in real-time. We have demonstrated that the algorithm could recognize nine eyelid gestures for both able-bodied users in different indoor environments and postures (i.e., sitting and standing) and for people with motor impairments with only five training samples per gesture. Moreover, we have designed a gesture mapping scheme for people with motor impairments to navigate apps only using eyelid gestures. Our study also shows that they were able to learn and use the mapping scheme with only a few minutes practice. Based on participants' feedback and our observations, we proposed five recommendations for designing and using eyelid gestures. Our work took the first step to explore the potential of a subset of possible eyelid gestures for people with motor impairments. Future work includes conducting larger scale studies with more people with a diverse set of motor ability in different environments, exploring a richer set of eyelid gestures by allowing for customization and using different gesture delimiters, and combining eyelid gestures with other input modalities, such as gaze.

ACKNOWLEDGMENTS

We would like to thank DGP members for their help in the pilot study and Hebei Disabled Persons' Federation and Shijiazhuang Shi Disabled Persons' Federation, in particular the following people, for their help in recruitment: Hongtao Zhen, Jian Gao, Huijie Feng, Yukun Ding, Qiuping Ning, Yuanyuan Si, Suling Yang, Shengmei Guo, Wenge Zheng, Yongzhe Chang, Shiwei Cui, Xiaolian Chen, Dongwei Shi, Jun Wang and Lina Zhang.

REFERENCES

[1] Mark V Albert, Cliodhna McCarthy, Juliana Valentin, Megan Herrmann, Konrad Kording, and Arun Jayaraman. 2013. Monitoring functional capability of individuals with lower limb amputations using mobile phones. *PLoS One* 8, 6 (2013).

[2] Mark V Albert, Santiago Toledo, Mark Shapiro, and Konrad Koerding. 2012. Using mobile phones for activity recognition in Parkinson's patients. *Frontiers in neurology* 3 (2012), 158.

[3] Fabio Aloise, Francesca Schettini, Pietro Arico, Serenella Salinari, Fabio Babiloni, and Febo Cincotti. 2012. A comparison of classification techniques for a gaze-independent P300-based brain–computer interface. *Journal of neural engineering* 9, 4 (2012), 045012.

[4] Jacopo M Araujo, Guangtao Zhang, John Paulin Paulin Hansen, and Sadasivan Puthusserypady. 2020. Exploring Eye-Gaze Wheelchair Control. In *ACM Symposium on Eye Tracking Research and Applications*. 1–8.

[5] T. Arroyo-Gallego, M. J. Ledesma-Carbayo, Á. Sánchez-Ferro, I. Butterworth, C. S. Mendoza, M. Matarazzo, P. Montero, R. López-Blanco, V. Puertas-Martín, R. Trincado, and L. Giancardo. 2017. Detection of Motor Impairment in Parkinson's Disease Via Mobile Touchscreen Typing. *IEEE Transactions on Biomedical Engineering* 64, 9 (2017), 1994–2002.

[6] Behrooz Ashtiani and I Scott MacKenzie. 2010. BlinkWrite2: an improved text entry method using eye blinks. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. 339–345.

[7] ALS Association. 2020. *Who Gets ALS?* http://www.alsa.org/about-als/facts-you-should-know.html

[8] Rafael Barea, Luciano Boquete, Luis Miguel Bergasa, Elena López, and Manuel Mazo. 2003. Electro-oculographic guidance of a wheelchair using eye movements codification. *The International Journal of Robotics Research* 22, 7-8 (2003), 641–652.

[9] Veronika Maria Berger, Gerhard Nussbaum, Carina Emminger, and Zoltan Major. 2018. 3D-Printing of Personalized Assistive Technology. In *International Conference on Computers Helping People with Special Needs*. Springer, 135–142.

[10] Eric Corbett and Astrid Weber. 2016. What can I say? addressing user experience challenges of a mobile voice user interface for accessibility. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 72–82.

[11] National Center for Health Statistics et al. 2018. Summary health statistics: National Health Interview Survey.

[12] 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 (2003), 359–373.

[13] Anthony J Hornof and Anna Cavender. 2005. EyeDraw: enabling children with severe motor impairments to draw with their eyes. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 161–170.

[14] Shuai Huang, Chun Luo, Shiwei Ye, Fei Liu, Bin Xie, Caifeng Wang, Li Yang, Zhen Huang, and Jiankang Wu. 2012. Motor impairment evaluation for upper limb in stroke patients on the basis of a microsensor. *International journal of rehabilitation research* 35, 2 (2012), 161–169.

[15] Ricardo Jota and Daniel Wigdor. 2015. Palpebrae Superioris: Exploring the Design Space of Eyelid Gestures. In *Proceedings of the 41st Graphics Interface Conference*. Canadian Human-Computer Communications Society, Toronto, Ontario, Canada, 3–5. https://doi.org/10.20380/GI2015.35

[16] Aleksandra Królak and Paweł Strumiłło. 2012. Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society* 11, 4 (2012), 409–419.

[17] Chris Lankford. 2000. Effective eye-gaze input into windows. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. 23–27.

[18] Luis Leiva, Matthias Böhmer, Sven Gehring, and Antonio Krüger. 2012. Back to the app: the costs of mobile application interruptions. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services - MobileHCI '12*. ACM Press, New York, New York, USA, 291. https://doi.org/10.1145/2371574.2371617

[19] Dongheng Li, David Winfield, and Derrick J Parkhurst. 2005. Starburst: A hybrid algorithm for video-based eye tracking combining feature-based and model-based approaches. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*. IEEE, 79–79.

[20] Google LLC. 2019. Mobile Vision | Google Developers. https://developers.google.com/vision/

[21] Robert Gabriel Lupu, Florina Ungureanu, and Valentin Siriteanu. 2013. Eye tracking mouse for human computer interaction. In *2013 E-Health and Bioengineering Conference (EHB)*. IEEE, 1–4.

[22] I Scott MacKenzie and Behrooz Ashtiani. 2011. BlinkWrite: efficient text entry using eye blinks. *Universal Access in the Information Society* 10, 1 (2011), 69–80.

[23] Meredith Ringel Morris, Jacob O. Wobbrock, and Andrew D. Wilson. 2010. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface*. Canadian Information Processing Society, Toronto, Ontario, Canada, 261–268. https://doi.org/10.1016/j.actamat.2009.07.058

[24] Maia Naftali and Leah Findlater. 2014. Accessibility in context: understanding the truly mobile experience of smartphone users with motor impairments. In *Proceedings of the 16th international ACM SIGACCESS conference on Computers & accessibility*. 209–216.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[26] Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N Truong. 2015. Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing (TACCESS)* 6, 1 (2015), 1–37.

[27] Alisha Pradhan, Kanika Mehta, and Leah Findlater. 2018. " Accessibility Came by Accident" Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–13.

[28] B. P. Printy, L. M. Renken, J. P. Herrmann, I. Lee, B. Johnson, E. Knight, G. Varga, and D. Whitmer. 2014. Smartphone application for classification of motor impairment severity in Parkinson's disease. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. 2686–2689.

[29] David Rozado, Jason Niu, and Martin Lochner. 2017. Fast human-computer interaction by combining gaze pointing and face gestures. *ACM Transactions on Accessible Computing (TACCESS)* 10, 3 (2017), 1–18.

[30] Jaime Ruiz, Yang Li, and Edward Lank. [n.d.]. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA. https://doi.org/10.1145/1978942.1978971

[31] Javier San Agustin, Henrik Skovsgaard, Emilie Mollenbach, Maria Barret, Martin Tall, Dan Witzner Hansen, and John Paulin Hansen. 2010. Evaluation of a low-cost open-source gaze tracker. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications*. 77–80.

[32] Andrew Sears, Clare-Marie Karat, Kwesi Oseitutu, Azfar Karimullah, and Jinjuan Feng. 2001. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the information Society* 1, 1 (2001), 4–15.

[33] Corten Clemente Singer and Björn Hartmann. 2019. See-Thru: Towards Minimally Obstructive Eye-Controlled Wheelchair Interfaces. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 459–469.

[34] Ruth Smith. 1989. Mouth stick design for the client with spinal cord injury. *American Journal of Occupational Therapy* 43, 4 (1989), 251–255.

[35] Young Chol Song. 2010. Joystick text entry with word prediction for people with motor impairments. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility*. 321–322.

[36] John A Stern, Larry C Walrath, and Robert Goldstein. 1984. The endogenous eyeblink. *Psychophysiology* 21, 1 (1984), 22–33.

[37] Non-Hispanic White and Non-Hispanic Black. 2016. Spinal cord injury (SCI) facts and figures at a glance. *Birmingham: National Spinal Cord Injury Statistical Center, Facts and Figures at a Glance* (2016).

[38] Jacob Wobbrock and Brad Myers. 2006. Trackball text entry for people with motor impairments. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 479–488.

[39] Jacob O Wobbrock, Brad A Myers, and Htet Htet Aung. 2004. Joystick text entry with date stamp, selection keyboard, and EdgeWrite. In *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. 1550–1550.

[40] Jacob O Wobbrock, Brad A Myers, Htet Htet Aung, and Edmund F LoPresti. 2003. Text entry from power wheelchairs: EdgeWrite for joysticks and touchpads. In *Proceedings of the 6th international ACM SIGACCESS conference on Computers and accessibility*. 110–117.

[41] 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*. 11–18.

[42] Guangtao Zhang and John Paulin Hansen. 2020. People with Motor Disabilities Using Gaze to Control Telerobots. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–9.

[43] Xiaoyi Zhang, Harish Kulkarni, and Meredith Ringel Morris. 2017. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2878–2889.