

Eyelid Gestures for People with Motor Impairments

By Mingming Fan, Zhen Li, and Franklin Mingzhe Li

Abstract

Although eye-based interactions can be beneficial for people with motor impairments, they often rely on clunky or specialized equipment (e.g., stationary eye-trackers) and focus primarily on gaze and blinks. However, two eyelids can open and close in different orders and for different duration to form rich 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 12 able-bodied people and 4 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. Furthermore, we design and evaluate a gesture mapping scheme for people with motor impairments to navigate mobile applications only using eyelid gestures. Finally, we discuss considerations for designing and using eyelid gestures for people with motor impairments.

1. INTRODUCTION

Fifteen percent of people in the U.S. have difficulties with their physical functioning, among whom almost half find it very difficult or impossible to walk unassisted for a quarter-mile.⁵ Although specialized devices, such as eye-trackers, brain-computer interfaces, and mechanical devices (e.g., joysticks, trackballs) can be helpful, they are often clunky, intrusive, expensive, or limited in functions supported (e.g., text entry). By contrast, as a general-purpose device, smartphones can benefit people with motor impairments with rich on-board sensors.¹² For example, motion sensors and the touch screen have been used to recognize users' physical activities (e.g., see Albert et al.¹) and assess their motor ability (e.g., see Printy et al.¹⁶). Microphone allows for using speech to enter texts (e.g., see Sears et al.¹⁹) or issue commands (e.g., see Pradhan et al.¹⁵). Camera enables eye-based interactions for people with motor impairments to enter text (e.g., see Pedrosa et al.¹⁴), issue gesture commands (e.g., see Rozado et al.¹⁷), and navigate a wheelchair (e.g., see Araujo et al.²).

Although helpful, eye-based interactions have primarily focused on gaze (i.e., eyeball movement^{14,17,22,23} or blinks^{10,23}). However, human's two eyelids can be in open or close states for short or long periods and in concurrent or sequential orders to form a rich set of eyelid gestures, which could 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 construct potential eyelid gestures. Although some eyelid gestures, such as winks, were proposed for hands-free interaction,⁶ 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, 12 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 state refers to the state of two eyelids and has 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 eyelids to twitch or cramp.⁶ Moreover, our investigation found that it is still challenging to robustly recognize half-closed states with current technology. Thus, we focus on the four states when constructing eyelid gestures. Future work could explore the potentials of "half-closed" eyelid states. As the "both eyelids open" state is the most common state when humans are awake, we use it as the *gesture delimiter* to

The original version of this paper is entitled "Eyelid Gestures on Mobile Devices for People with Motor Impairments" and was published in *Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, 2020.

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.⁶ As it can be hard to memorize the exact duration of a state, we discretize duration into two levels—*short* and *long*. *Short* duration refers to the time it takes to *intentionally* close an eyelid (e.g., longer than a spontaneous blink (50–145 ms)²⁰) 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. Figure 1 shows these nine eyelid gestures and their abbreviations.

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 640×480 resolution and leverages Google Mobile Vision API to generate a stream

of probability pairs of each eye being open (P_L, P_R).⁹ The details of how the API estimates probability can be found in the work of G. LLC.⁹ Figure 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 API⁹ 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 a segment and filters it out if its duration is too short because extremely short segments are likely caused by spontaneous blinks (50–145 ms²⁰) 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 (see Figure 1). The algorithm then resamples the sequence of probability pairs (P_L, P_R) in the segment to ensure all

Figure 1. The illustrations and abbreviations of the nine eyelid gestures that our algorithm detects. Each letter in a gesture *abbr.* depicts its 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 this state for some time (-), transitions to “only the Right eyelid close,” sustains in this 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.

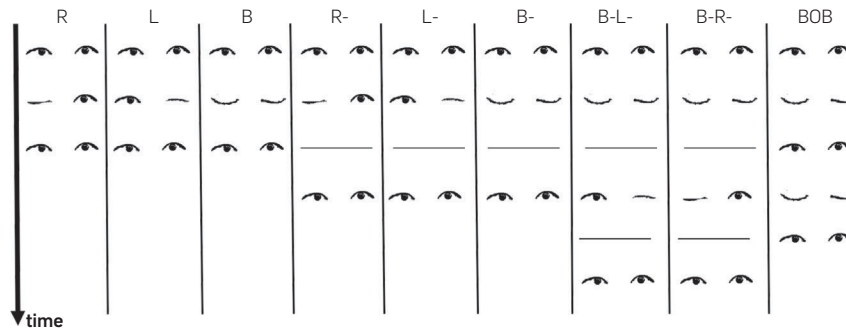
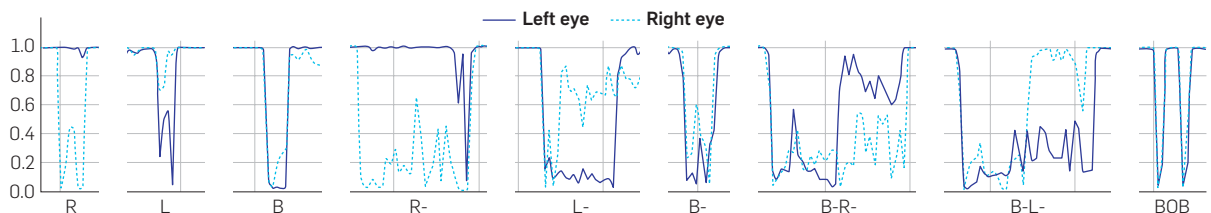


Figure 2. The probabilities of two eyes being open when a user performs each of the nine eyelid gestures. The blue (solid) and cyan (dashed) lines represent the probabilities of the left and right eye being open respectively.



segments contain the same number of probability pairs (50 and 100 samples for short and long gestures, respectively). Next, the resampled 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.¹³ More details can be found in original articles.^{4,8} Our source code is available at: <https://github.com/mingming-fan/EyelidGesturesDetection>.

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 people without motor impairments before testing with people with motor impairments.

3.1. Participants

We recruited 12 able-bodied participants aged between 23 and 35 ($M = 26$, $SD = 4$, 5 males, and 7 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 wore 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 (see Figure 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

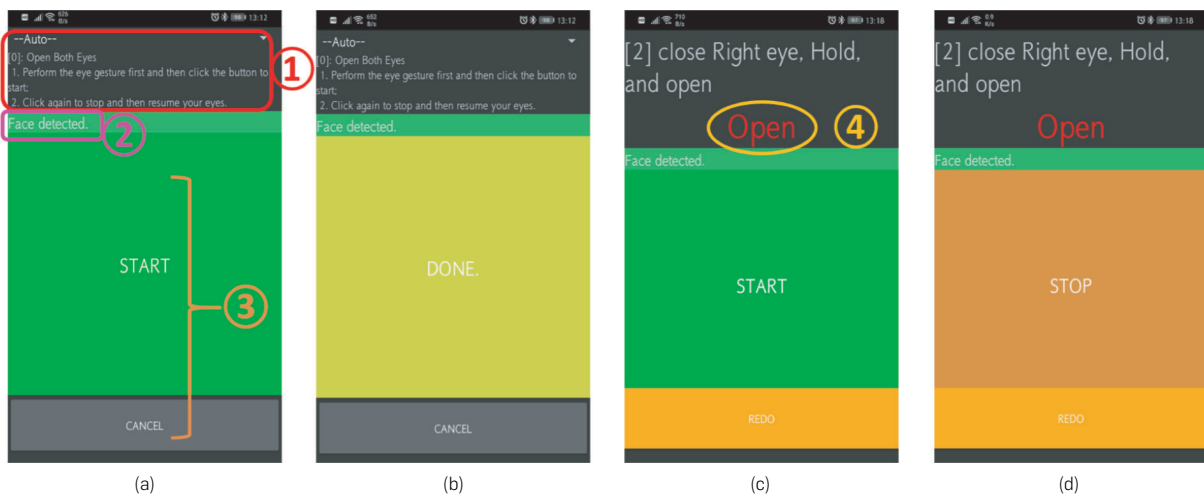
instructions in the app although *sitting* at a desk and holding the phone in their preferred hand in one office. We then collected testing data by asking them to perform each eyelid gesture another five times although *standing* in another office room and holding the phone in their preferred hand. The 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.

To collect data samples for each eyelid state, the evaluation app first presented a target eyelid state on the top side of the screen (see Figure 3a and 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-fold cross-validation¹ of the eyelid state classifier on the phone in real time. The training process took on average 558 ms.

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 (see Figure 3c and 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 ms.

To collect testing data, participants performed each eyelid gesture five more times although standing in another office

Figure 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 shows the control buttons, such as “start,” “cancel,” and “redo.” During eyelid gesture evaluation, detected eyelid state is shown in 4.



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 *user-dependent* and *user-independent* evaluations.

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* were the *gesture delimiter* to separate eyelid gestures, we further trained a classifier to recognize only two eyelid states by grouping the last three states (with an eyelid close) together. The average accuracy was 0.98 ($SD = .02$).

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 Figure 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), *B* (2258 ms), *L-* (2010 ms), *B-* (2432 ms), *B-L-* (4169 ms), *B-R-* (4369 ms), and *BOB* (2198 ms). It shows that more complex gestures took longer to complete overall.

User-independent eyelid gesture evaluation. To assess how well a pretrained 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 Figure 4b. This result suggests that a pretrained 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,^{3,21} 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 approximately an hour, and each participant was compensated with \$15.

4.2. Procedure

The studies were conducted in participants' homes. Figure 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 (see Figure 3) to accommodate the participants' motor impairments. Instead of asking them to press “START” and “STOP” buttons, the

Figure 4. Study 1: The confusion matrix of user-dependent (a) and user-independent (b) evaluations, respectively (columns: ground truth; rows: predictions; and N/A means not recognized).

	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)

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

Figure 5. 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 smartphone to be evaluated was mounted on the top of a tripod, which was placed on the wheelchair trays or the desk with its front camera roughly at their eye levels.



app used a 10-s 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 eye lid state. These data were used to evaluate the 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

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 (with at least one eyelid close) were grouped into one *close* state, the accuracy of the two-state classifier was more robust: .997 ($SD = .004$).

User-dependent eyelid gesture evaluation. We performed the same user-dependent evaluation as Section "User-dependent eyelid gesture evaluation", 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 (see Figure 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), and *BOB* (3002 ms).

User-independent eyelid gesture evaluation. We performed the same user-independent evaluation as Section "User-independent Eyelid Gesture Evaluation", and the

Figure 6. Study 2: The confusion matrix of user-dependent (a) and user-independent (b) evaluations, respectively (columns: ground truth; rows: predictions; and N/A means not recognized).

	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	2	0
B-R-	1	0	0	2	0	0	20	2	0
B-L-	0	0	0	0	1	0	1	11	0
BOB	0	0	0	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	0	0	0	0	0
L-	0	2	0	0	13	1	0	1	0
B-	3	0	2	1	1	19	2	6	1
B-R-	2	0	0	2	0	0	18	1	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)

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 (see Figure 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.

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 (see Figure 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,⁷ 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.

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 (see Figure 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 that contained five navigation tasks, and the target item for each navigation was randomly generated. The app

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

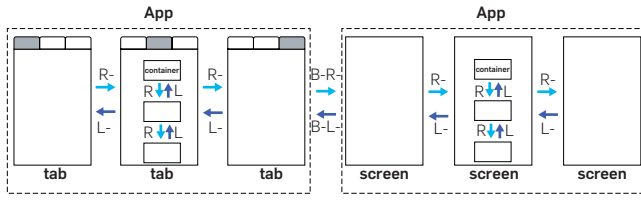
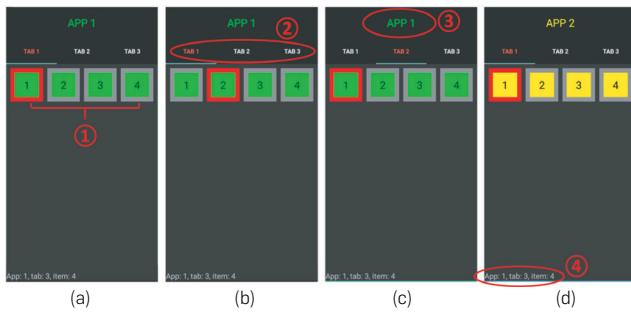


Figure 8. (a)–(d) are the app navigation UIs. 1 shows the containers, 2 shows the tabs, 3 shows the current app name, and 4 shows the target item of current trial. Three types of navigation are illustrated: between containers in a tab (a, b), between tabs within an app (a, c), and between apps (a, d).



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 min to complete. Afterward, 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 carrying out the gesture’s physical action”) using 7-point Likert-scale questions, which were used to elicit feedback on gesture commands (e.g., see Morris et al.¹¹ and Naftali and Findlater¹⁸).

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 but high-cost commands (e.g., switching apps) although simpler eyelid gestures were assigned to relatively more-frequent but 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 switching to 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 or close both eyelids 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. 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 5 min 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 predefined 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; and 5) allow users to define their own gestures to enrich their interaction vocabulary.

6. LIMITATIONS AND FUTURE WORK

Although our participants did not complain about fatigue due to short study duration, performing eyelid gestures for a long time might be fatiguing. Furthermore, our study only included a small number of people with motor impairments. Future work should conduct larger scale studies with more participants who have a more diverse set of motor impairments for longer periods to better understand practices and challenges associated with using eyelid gestures.

As a first step toward designing eyelid gestures for people with motor impairments, our work opens up promising future research directions.

Eyelid states. Our eyelid gesture design space is based on four eyelid states with eyelids either open or close. As is described in Section 2, eyelids could also be in half-closed states (e.g., squinting). Including half-closed states into the design space would result in more eyelid gestures.

Duration of eyelid states. We divided the duration of an eyelid state into two discrete 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 eyelid gestures.

Gesture delimiter. We used “both eyelids open” as the gesture delimiter as it is the default state of the eyes when

people are awake. However, other delimiters might enable new eyelid gestures, such as “blinking the right/left eye twice (although keeping the other eye closed).”

User-defined eyelid gesture design. Our study showed that people with motor impairments preferred customizing eyelid gestures to use in different contexts and to avoid false activation of recognition based on their ability to control their eyelids. Thus, it is imperative to understand what eyelid gestures people with motor impairments would want to create and use. Co-design workshops with people with motor impairment and gesture elicitation studies are viable approach to uncover user-defined eyelid gestures.

Eyelid gestures and hybrid eye gestures. We explored a subset of possible eyelid gestures with one or two eyelid states between the gesture delimiter (i.e., both eyelids open). 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.


Furthermore, the literature has explored gaze gestures for people with motor impairments to entry texts, draw on computer screens, and navigate wheelchairs. Thus, it is worth exploring ways to combine eyelid gestures with gaze to create hybrid eye gestures to enrich touch-free interactions for both people with and without motor impairments.

7. CONCLUSION

We have taken a first step toward designing eyelid gestures for people with motor impairments to interact with mobile devices without needing to touch the devices. We have presented an algorithm to detect nine eyelid gestures on smartphones in real time and demonstrated that it 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 only scratches the surface of the potential of eyelid gestures for people with motor impairments and for the general public for hands-busy scenarios. 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 and facial gestures.

Acknowledgment

We would like to thank our participants and Hebei Disabled Persons’ Federation and Shijiazhuang Shi Disabled Persons’ Federation for their help in recruitment. 

References

1. Albert, M.V., Toledo, S., Shapiro, M., Koering, K. Using mobile phones for activity recognition in Parkinson's patients. *Front. Neural.* 3, 158 (2012), 7.
2. Araujo, J.M., Zhang, G., Hansen, J.P.P., Puthusserypady, S. Exploring eye-gaze wheelchair control. In *ACM Symposium on Eye Tracking Research and Applications* (2020), ACM Press, New York, USA, 1–8.
3. A. Association. Who gets als? ALS Association, 2020. <https://www.als.org/understanding-als/who-gets-als>.
4. Fan, M., Li, Z., Li, F.M. Eyelid gestures on mobile devices for people with motor impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, (New York, NY, USA, 2020), Association for Computing Machinery, NY, USA.
5. N. C. for Health Statistics et al. Summary Health Statistics: National Health Interview Survey, 2018.
6. Jota, R., Wigdor, D. Palpebrae superioris: Exploring the design space of eyelid gestures. In *Proceedings of the 41st Graphics Interface Conference* (Toronto, Ontario, Canada, 2015), Canadian Human-Computer Communications Society, ACM Press, New York, USA, 3–5.
7. Leiva, L., Böhmer, M., Gehring, S., Krüger, A. 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* (New York, New York, USA, 2012), ACM Press, New York, USA, 291.
8. Li, Z., Fan, M., Han, Y., Truong, K.N. iWink: Exploring eyelid gestures on mobile devices. In *Proceedings of the 1st International Workshop on Human-Centric Multimedia Analysis*, (New York, NY, USA, 2020), Association for Computing Machinery, New York, USA, 83–89.
9. G. LLC. Mobile Vision | Google Developers, 2019.
10. MacKenzie, I.S., Ashtiani, B. Blinkwrite: Efficient text entry using eye blinks. *Universal Access Information Soc.* 10, 1 (2011), 69–80.
11. Morris, M.R., Wobbrock, J.O., Wilson, A.D. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface* (Toronto, Ontario, Canada, 2010), Canadian Information Processing Society, ACM Press, New York, USA, 261–268.
12. Naftali, M., Findlater, L. 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* (2014), ACM Press, New York, USA, 209–216.
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* 12, (2011), 2825–2830.
14. Pedrosa, D., Pimentel, M.D.G., Wright, A., Truong, K.N. Filteryping: Design challenges and user performance of dwell-free eye typing. *ACM Trans. Accessible Comput.* 6, 1 (2015), 1–37.
15. Pradhan, A., Mehta, K., Findlater, L. "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* (2018), ACM Press, New York, USA, 1–13.
16. Printy, B.P., Renken, L.M., Herrmann, J.P., Lee, I., Johnson, B., Knight, E., Varga, G., Whitmer, D. 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* (2014), IEEE, 2686–2689.
17. Rozado, D., Niu, J., Lochner, M.. Fast human-computer interaction by combining gaze pointing and face gestures. *ACM Trans. Accessible Comput.* 10, 3 (2017), 1–18.
18. Ruiz, J., Li, Y., Lank, E. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (New York, New York, USA, 2011), ACM Press, New York, USA.
19. Sears, A., Karat, C.-M., Oseitutu, K., Karimullah, A., Feng, J. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access Information Soc.* 1, 1 (2001), 4–15.
20. Stern, J.A., Walrath, L.C., Goldstein, R. The endogenous eyeblink. *Psychophysiology* 21, 1 (1984), 22–33.
21. White, N.-H., Black, N.-H. Spinal cord injury (sci) facts and figures at a glance. In *National Spinal Cord Injury Statistical Center, Facts and Figures at a Glance*, Birmingham (2016).
22. Wobbrock, J.O., Rubinstein, J., Sawyer, M.W., Duchowski, A.T. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 Symposium on Eye Tracking Research & Applications* (2008), ACM Press, New York, USA, 11–18.
23. Zhang, X., Kulikarni, H., Morris, M.R. Smartphone-based gaze gesture communication for people with motor disabilities. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), ACM Press, New York, USA, 2878–2889.

Mingming Fan ([mingmingfan]@ust.hk), Computational Media and Arts, Hong Kong University of Science and Technology, Guangzhou, China. He served as corresponding author.

Zhen Li ([zhen]@dgp.toronto.edu), Department of Computer Science, University of Toronto, Toronto, ON, Canada.

Franklin Mingzhe Li ([mingzhe2]@cs.cmu.edu), Human-Computer Interaction, Carnegie Mellon University, Pittsburgh, PA, USA.

© 2022 ACM 0001-0782/22/1 \$15.00

The Handbook on Socially Interactive Agents

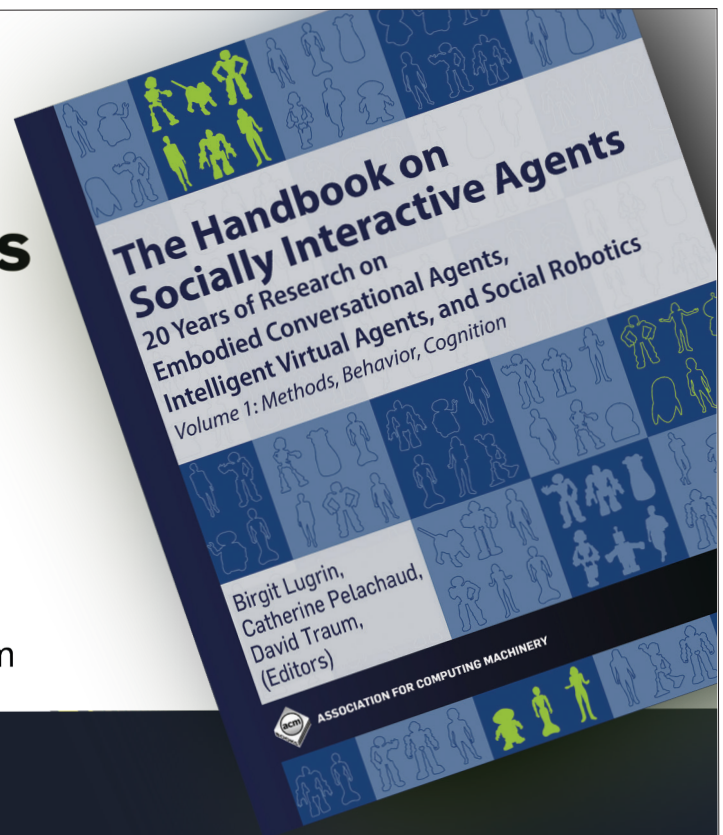
Birgit Lugrin
Catherine Pelachaud
David Traum (Eds.)

ISBN: 978-1-4503-8721-7

DOI: 10.1145/3477322

<http://books.acm.org>

<http://store.morganclaypool.com/acm>



ACM BOOKS
Collection II