

FittsFace: Exploring Navigation and Selection Methods for Facial Tracking

Justin Cuaresma and I. Scott MacKenzie^(✉)

Department of Electrical Engineering and Computer Science, York University,
Toronto, ON, Canada
justincuaresma@gmail.com, mack@cse.yorku.ca

Abstract. An experimental application called *FittsFace* was designed according to ISO 9241-9 to compare and evaluate facial tracking and camera-based input on mobile devices for accessible computing. A user study with 12 participants employed a Google *Nexus 7* tablet to test two facial navigation methods (positional, rotational) and three selection methods (dwell, smile, blink). Positional navigation was superior, with a mean throughput of 0.58 bits/second (bps), roughly 1.5× the value observed for rotational navigation. Blink selection was the least accurate selection method, with a 28.7% error rate. The optimal combination was positional+smile, with a mean throughput of 0.60 bps and the lowest tracking drop rate.

Keywords: Camera input · Facial tracking · Blink selection · Smile selection · Fitts' law · ISO 9241-9 · Qualcomm Snapdragon · Accessible input techniques

1 Background

Smartphones and tablets are now an integral component of our daily lives, as new technologies emerging regularly. Current devices include a media player, a camera, and sophisticated communications electronics. More specifically, buttons have given way to smooth touchscreen surfaces. Not surprisingly, this impacts user interaction. Touchscreen surfaces allow input via touch gestures such as taps and swipes. Additional sensors, such as gyroscopes and accelerometers, enable other forms of interaction (e.g., tilt) to enhance the user experience.

One common sensor on smartphones and tablets that is rarely employed for user input is the front-facing camera. Although mostly used as a novelty, it is possible to adopt the front-facing camera to perform facial tracking. The front-facing camera is sometimes used in picture-taking applications to enhance or alter one's facial features. An app called *Reframe* on the App Store, shown in Fig. 1 illustrates a typical example. The app places a pair of user-selected glasses on the user's face, allowing him/her to virtually try on a pair to see how they look.

Besides novelty apps, there are additional ways to use facial tracking. For games, facial tracking is presently used to maneuver in-game elements [1]. Accessible computing is another candidate. Android and iOS already integrate accessible functionalities via voice I/O (e.g., *Siri*); however, there is still no comprehensive mobile solution

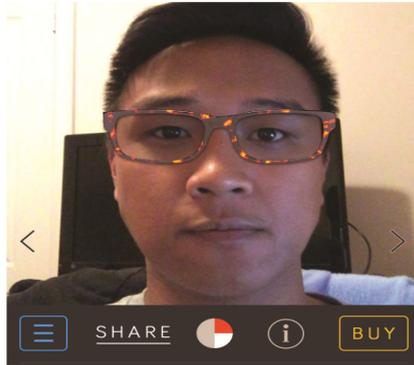


Fig. 1. Using the front-facing camera, *Reframe* lets a user try on eyeglasses and sunglasses.

for people with motor disabilities. Thus, a potential application for facial tracking is for users with deficient motor control in the hands or arms but with full motor control of the head, neck, and face. The starting point, however, is to determine if facial tracking can be used for simple interactions such as selecting on-screen targets.

FittsFace was designed as an ISO-conforming Fitts' law tool to study facial input methods for traditional point-select operations on mobile devices. Before detailing the operation of *FittsFace*, we review related work using facial tracking in a desktop environment.

2 Related Work

Most work on facial tracking is directed at desktop or laptop systems with a webcam. The literature generally focuses on computer vision algorithms rather than on usability. The following review is limited to the use of facial tracking for target selection and empirical evaluations of such.

Perini et al. used computer vision techniques and a webcam to develop *FaceMouse* [9]. The application was text-entry for tetraplegic users; thus, the targets were keys on a soft keyboard. Dwell-time selection was combined with a letter-prediction scheme. The text entry rate was 13 chars/min.

Varona et al. built a desktop-webcam facial tracking system that used nose position to detect face gestures [13]. A grid of 25 circular targets (radius = 15 pixels) was positioned on the display. Users selected the targets by moving their head (i.e., nose) and selected targets by dwelling or winking. Although selection time was not reported, accuracy was 85.9% for untrained users and 97.3% for trained users.

Javanovic and MacKenzie developed *MarkerMouse* using a laptop webcam and a head-mounted tracker [4]. Using a Fitts-type ISO 9241-9 experiment, they compared a conventional mouse with a head-mounted tracker using position control and velocity control. Position control was superior and considered more intuitive, yielding a throughput of 1.61 bits/second (bps). Not surprisingly, the mouse performed best with a throughput of 4.42 bps, which is consistent with other ISO 9241-9 evaluations [12].

Gizatdinova et al. presented a hands-free text entry system using facial tracking, gesture detection, and an on-screen keyboard [2]. They reported that single-row layouts achieved better results, especially when consecutive targets are far apart. They used eyebrow movement for selection, reporting that brow-up movements outperformed brow-down movements due to asymmetry in muscle control.

3 Performance Evaluation

To evaluate facial tracking as an input method in the mobile context, we used the methodology in the ISO 9241-9 standard for non-keyboard input devices [3]. *FittsFace* was modelled after *FittsTouch*, a touch-based Android application following the ISO standard [5]. The main difference is that *FittsFace* receives input from a facial tracking subsystem rather than from touch events. Details are provided below in the Method section.

The most common ISO 9241-9 evaluation procedure uses a two-dimensional task with targets of width W arranged in a circle. Selections proceed in a sequence moving across and around the circle (Fig. 2). Each movement covers an amplitude A – the diameter of the layout circle. The movement time (MT , in seconds) is recorded for each trial and averaged over the sequence.

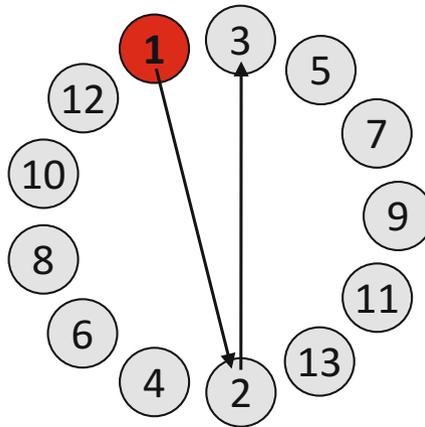


Fig. 2. Two-dimensional target selection task in ISO 9241-9 [3].

The difficulty of each trial is quantified using an index of difficulty (ID , in bits) and is calculated from A and W as follows:

$$ID = \log_2 \left(\frac{A}{W} + 1 \right) \quad (1)$$

The main performance measure in ISO 9241-9 is throughput (TP , in bits/second or bps) which is calculated over a sequence of trials as the ID - MT ratio:

$$TP = \frac{ID_e}{MT} \quad (2)$$

The standard specifies calculating throughput using the *effective index of difficulty* (ID_e). The calculation includes an adjustment for accuracy to reflect what participants actually did, rather than what they were asked to do:

$$ID_e = \log_2 \left(\frac{A_e}{W_e} + 1 \right) \quad (3)$$

with

$$W_e = 4.133 \times SD_x \quad (4)$$

The term SD_x is the standard deviation in the selection coordinates computed over a sequence of trials. For the two-dimensional task, selections are projected onto the task axis, yielding a single normalized x -coordinate for each trial. For $x = 0$, the selection was on a line orthogonal to the task axis that intersects the center of the target. x is $-ve$ for selections on the near side of the target centre and $+ve$ for selections on the far side. The factor 4.133 adjusts the target width for a nominal error rate of 4% under the assumption that the selection coordinates are normally distributed. The effective amplitude (A_e) is the actual distance traveled along the task axis. The use of A_e instead of A is only necessary if there is an overall tendency for selections to overshoot or undershoot the target (see [5] for additional details).

4 Method

Our user study compared two facial tracking navigational methods (rotational, positional) and three selection methods (blink, smile, dwell) using the performance measures of throughput, movement time, error rate, and number of tracking drops. The goal was to find a viable configuration for hands-free input in a mobile context.

4.1 Participants

Twelve non-disabled participants (7 female) were recruited from the local university campus. Ages ranged from 19 to 29 years. All participants were casual smartphone users. Participants receive a nominal fee (\$10).

4.2 Apparatus (Hardware and Software)

The hardware was a Google *Nexus 7* tablet running Android 5.0.2 (Fig. 3a). The device had a 7" display (1920×1200 px, 323 px/inch). *FittsFace* was developed in Java using the Android SDK, with special focus on the Qualcomm Snapdragon Facial Recognition API (Fig. 3b) [10].

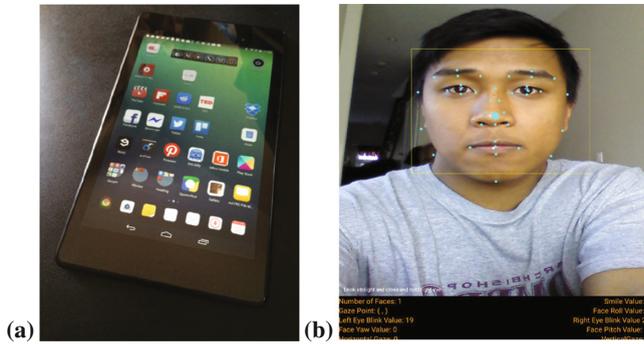


Fig. 3. (a) Nexus 7 (b) Facial recognition SDK demonstration.

FitsFace implements the two-dimensional ISO task described earlier (Fig. 2). After entering setup parameters (Fig. 4a) a test condition appears with 9 targets (see Fig. 4b and c for examples). The participant navigates a white dot (cursor) around the display by moving his/her head. To avoid inadvertent selections, a sequence of trials started only after the participant selected a green circle in the center of the display. Data collection began when the first target circle (red) was selected. The goal was to place the cursor in the target circle via the current navigation method and select it via the current selection method. If a selection occurred with the cursor outside the target, an error was logged and the next target was highlighted.

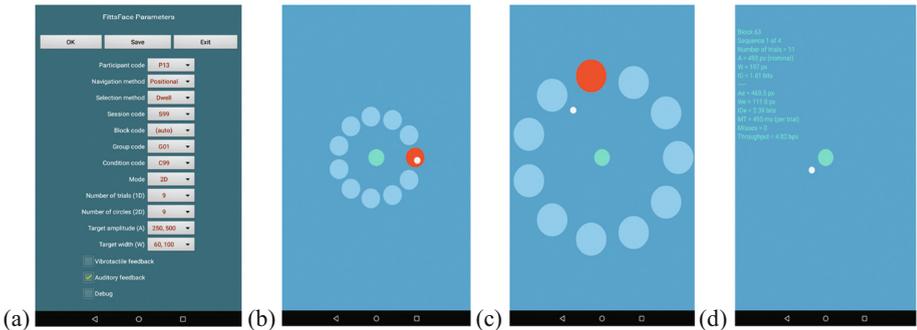


Fig. 4. FittsFace (a) setup activity (b) $A = 250, W = 60$ (c) $A = 500, W = 100$ (d) results after completing a sequence. (Color figure online)

A “pop” was heard on a correct selection, a “beep” for an error. After all 9 targets were selected, a results dialog appeared (see Fig. 4d).

Six configurations were tested to accommodate two navigation methods (positional, rotational) and three selection methods (smile, blink, dwell), as now described.

Positional navigation tracked the nose and its offset from the center of the display. The offset from the center was multiplied by a scaling factor to achieve efficient head movement. Through pilot testing, we settled on a scaling factor of 3.8.

Rotational navigation determines the pitch and yaw of one's face. More specifically, a z -plane point coordinate is derived with $x = \arcsin(\text{pitch})/(\text{scaling factor})$ and $y = \arcsin(\text{yaw})/(\text{scaling factor})$. We used a scaling factor of 1500.

Smile selection utilized the Qualcomm Snapdragon's smile detection API. A "select" was triggered when the smile value exceeded 65. Although the best way to select was a wide open smile, a big closed-mouth smile also worked.

Blink selection utilized the blink detection API with selection occurring when the blink value exceed 55 for 550 ms.

Dwell selection was triggered by maintaining the cursor within the target for 2500 ms. This rather high value was deemed necessary through pilot testing due to the overall slow pace of movements and selections with the facial processing API. If the cursor entered the target but the participant failed to complete a selection within a 60-second timeout, an error was registered and the next target was highlighted.

4.3 Procedure

Participants were tested in a controlled environment that provided adequate lighting for the front-facing camera. The test device was placed on a table using a laptop as a stand (Fig. 5). Before starting, participants were introduced to the objectives of the experiment and to the operation of *FittsFace*.

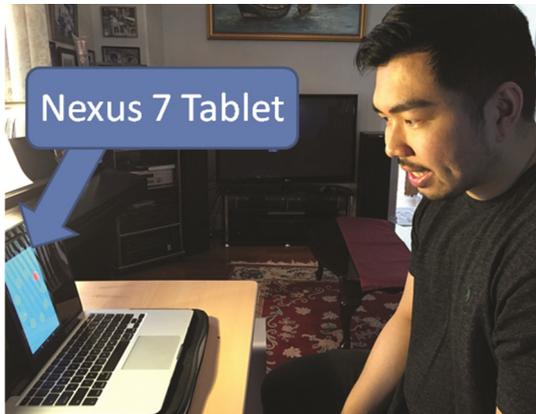


Fig. 5. Participant performing trials using smile as a selection method.

Each navigation and selection mode was briefly demonstrated. No practice trials were given. Participants took about 45 min to complete the experiment. After testing, participants were given a questionnaire to solicit qualitative feedback.

4.4 Design

The experiment was a 2×3 within-subjects design. The main independent variables and levels were

- Navigation method (positional, rotational)
- Selection method (smile, blink, dwell)

Two additional independent variables were target amplitude (250, 500) and target width (60, 100). These were included to ensure the tasks covered a range of difficulties. Using Eq. 1, the difficulties ranged from $ID = \log_2(250/100 + 1) = 1.81$ bits to $ID = \log_2(500/60 + 1) = 3.22$ bits. For each *A-W* condition, participants performed 9 target selections in a sequence.

The dependent variables were throughput, movement time, error rate, number of tracking drops, and timeouts (dwell only).

Participants were divided into six groups to counterbalance the six input conditions and thereby offset learning effects.

The total number of trials was 2592 (12 participants \times 2 navigation methods \times 3 selection methods \times 2 widths \times 2 amplitudes \times 9 selections/sequence).

5 Results and Discussion

5.1 Throughput

The grand mean for throughput was 0.47 bps. Here, we see empirical evidence of the challenges in facial tracking. For a similar task in a desktop-mouse environment, throughput is typically 4–5 bps – about 10 \times greater [12]. However, other pointing devices, such as a touchpad or joystick, often yield low throughputs – under 2 bps. Given that an application for facial tracking is accessible computing, achieving high values for throughput is perhaps not so important. Furthermore, this user study was mostly concerned with examining various interaction possibilities for facial tracking, rather than directly comparing facial tracking to other input methods.

Throughput by condition is seen in Fig. 6. Rotational navigation produced a mean throughput of 0.36 bps. Positional, on the other hand, yielded a higher mean throughput of 0.58 bps, or 59% higher. The effect of navigation method on throughput was statistically significant ($F_{1,11} = 47.1, p < .0001$).

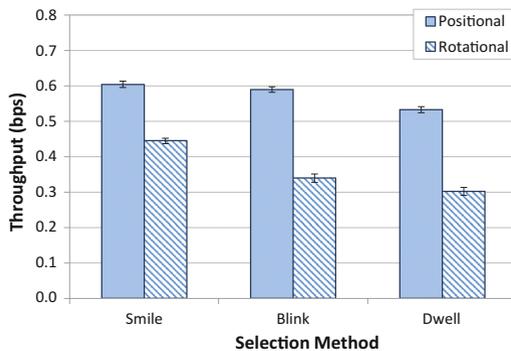


Fig. 6. Throughput (bps) by navigation method and selection method. Error bars show ± 1 SE.

By selection method, throughput was 0.47 bps (blink), 0.42 bps (dwell), and 0.53 bps (smile). Comparing the best to worst, throughput was 26% higher for smile selection than for dwell selection. The effect of selection method on throughput was statistically significant ($F_{2,22} = 8.83, p < .01$).

We examined different combinations of navigation and selection methods and their effects on throughput. Positional+smile performed best with a throughput of 0.60 bps. Positional+blink was similar with a throughput of 0.59 bps. At 0.53 bps, positional+dwell produced the lowest throughput among the positional combinations.

Rotational combinations resulted in lower throughputs. The highest among rotational combinations was rotational+smile, with a mean throughput of 0.45 bps. Rotational+blink produced a mean of 0.34 bps. The combination with the least throughput overall was the rotational+dwell combination, with a mean throughput of only 0.30 bps, a value 50% lower than for the positional+smile combination.

As observed, positional navigation was superior, producing higher throughputs than rotational navigation. Key factors that impact the throughput of both navigation methods were the stability and precision in the tracking system. Participants noted that positional tracking was more stable than rotational tracking. The values derived from the rotational data (pitch and yaw) were noisier than the values gathered from positional tracking of the nose. Although a running average algorithm was implemented to smooth the values, the effect of noise still impacted the stability of the cursor movement when using rotational tracking. More complex filter algorithms may achieve better results [7, 8, 14]. A common observation was a cursor drift or jump which caused additional movement time and, therefore, a lower throughput.

As observed, smile selection yielded the highest throughput. Our original assumption hypothesized that blink would have the highest throughput. The result can be justified by analyzing the implementation of the selection types. Smile selection did not use a timer, unlike dwell and blink. Hence, participants triggered selection as soon as they flashed their smile. With the blink and dwell selection methods, the timer-based trigger added to the overall movement time (and, hence, lowered throughput).

It follows from our analysis that positional+smile is the optimal combination due to its stability, more precise tracking, and the apparently instant triggering mechanism.

5.2 Movement Time

The grand mean for movement time over all 2592 trials was 7139 ms. This result is yet another sign of the challenge in using facial tracking on mobile devices. For a desktop-mouse experiment, movement times would be under one second per trial using similar index of difficulties (*IDs*). Controlling the white dot (cursor) using facial tracking required the participants to make rather slow and deliberate actions – quite unlike the rapid, facile movement of a mouse on a desktop.

The means for movement time by condition are shown in Fig. 7. Looking first at navigation method, the mean movement time using the positional method was 5180 ms. In contrast, the mean movement time for the rotational method was 9097 ms, or about

1.8× longer. Not surprisingly, the difference between the two input methods was statistically significant ($F_{1,11} = 8.56, p < .05$). The results suggest that rotational navigation was inferior to positional navigation when it comes to movement time.

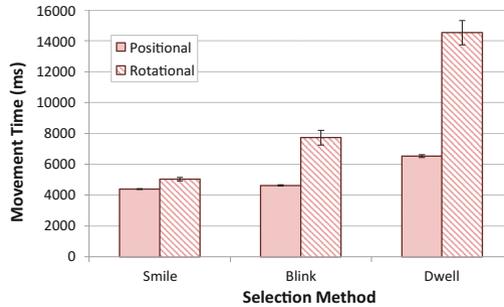


Fig. 7. Movement time (ms) by navigation method and selection method. Errors bars show $\pm 1 SE$.

In terms of selection method, smile yielded the best movement time resulting in a mean of 4708 ms. Blink came second with a mean of 6171 ms, 1.3× more than smile. Lastly, dwell resulted in a mean of 10538 ms, roughly 2.2× more than smile. Hence, it is evident that smile is the superior selection method when considering movement time. The effect of selection method on movement time was statically significant ($F_{2,22} = 14.5, p < .0005$).

We examined different combinations of navigation methods paired with selection methods. Rotational+dwell produced the longest mean movement time, 14539 ms (see Fig. 7). The combination with the least movement time was positional+smile, having a mean movement time of 4383 ms, roughly one third the value for rotational+dwell. The next fastest combination was positional+blink, yielding a mean of 4621 ms. Considering all selection methods, those using rotational navigation produced longer mean movement times than those using positional navigation.

5.3 Error Rate

Error rate was observed as a percentage of missed selections. See Fig. 8. The grand mean for error rate was 16.6%. Positional navigation produced a mean of 7.6% compared to rotational navigation at 25.7%. Rotational navigation was about 3.3× more error prone and was, on average, missing a quarter of the targets. This result, however, is directly related to the issue of stability during rotational tracking, as discussed in the previous section. The effect of navigation method on error rate was statistically significant ($F_{1,11} = 52.6, p < .0001$).

Blink selection produced a mean error rate of 28.7% and was the most error prone among the selection methods. Smile had a mean error rate of 19.9%. Lastly, dwell had the least error rate at 1.4%. The effect of selection method on error rate was statistically significant ($F_{2,22} = 38.9, p < .0001$).

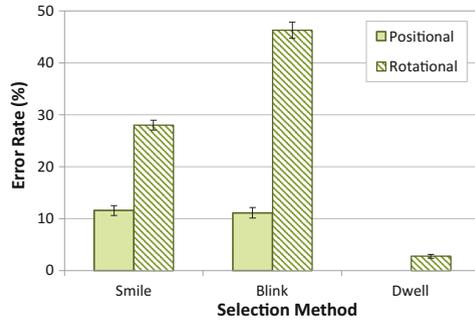


Fig. 8. Error rate (%) by navigation method and selection method. Errors bars show $\pm 1 SE$.

Dwell-time selection is inherently an error-free selection method; however an error was logged if a timeout occurred (60 s without staying within the target for the prescribed dwell-time interval). No such events were logged for the positional+dwell condition. For the rotational+dwell condition, 2.8% of the trials were errors due to a timeout.

The blink timer had an impact on error rate. It was observed that participants sometimes shifted the cursor out of the target before the timeout. This resulted in a missed selection.

The results of navigation and selection combinations, however, indicate a slightly different result. Other than dwell combinations, positional+blink produced the lowest error rate, 11.1%. Positional+smile had a slightly higher error rate at 11.6%. In some cases, the instantaneous triggering of smile selection was disadvantageous as natural reactions such as the opening of the mouth and speaking triggered a selection. For example, there were a few instances where a participant left his/her mouth slightly opened after flashing a smile, and this would inadvertently trigger a premature selection on the next target [9].

The rotational+blink combination was the least accurate with a mean error rate of 46.3%. This result is significant as this implies participants missed almost half the targets. Further investigation revealed that the rotational tracking system may have tracked the rotation of individual facial features as well [9]. This is evident in the error rates produced by the rotational+blink and rotational+smile.

Comparing blink and smile combined with rotational navigation suggests that the tracker is much more dependent on the eyes of the participant. This may be due to the large surface area occupied by one’s mouth which makes it easier to track. Not surprisingly, the interaction effect of navigation method by selection method was statistically significant ($F_{2,22} = 13.5, p < .001$).

5.4 Tracking Drops

Another dependent variable was the number of times tracking was lost or dropped in a sequence of 9 target selections. This dependent variable is important as it bears on the efficiency and precision of a facial tracking system.

The grand mean for tracking drops was 1.4 per sequence (Fig. 9). Positional navigation averaged 0.4 drops/sequence. On the other hand, rotational navigation averaged 2.3 drops/sequence, roughly 500% more than with positional tracking. Our results are consistent with previous studies, which showed that the nose was the optimal facial feature to track [13, 14]. Blink selection had the highest mean number of drops in the context of selection methods, averaging 1.5 drops/sequence. Smile, lowest, averaged 1.2 drops/sequence, consistent with our observations on error rate: Smile is easier to track.

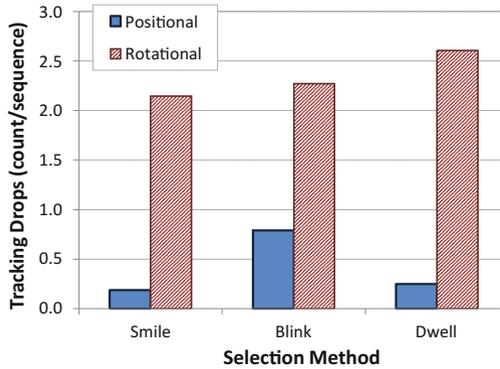


Fig. 9. Tracking drops per sequence by navigation method and selection method.

The optimal combination which yielded the least tracking drops was, again, positional+smile, with 0.19 drops/sequence. Positional+blink had 0.79 drops/sequence which further enforces that facial tracking is highly dependent on one's facial structure. However, rotational+dwell was highest at 2.27 drops/sequence.

It was observed that with rotational navigation participants rotated past a threshold where tracking holds (i.e., some facial features were occluded). Participants would tilt their head up to grab a target near the top of the display. Over tilting caused tracking drops due to the distorted perspective of the facial image. Although positional navigation suffers from a similar issue, tracking a single facial feature does not cause as many tracking drops since moving (versus just rotating) one's head adds to the movement of the cursor. The same effect was noted in previous studies [2, 4, 6, 11], where selection problems increased as target distance from the origin increased.

5.5 Participant Feedback

Based on the questionnaire given at the end of testing, participants praised the idea of integrating facial tracking as an assistive feature on mobile devices. Ten of the twelve participants preferred the positional navigation method to rotational. One participant expressed a preference thus:

The positional tracking was significantly easier to control and target the cursor accuracy compared to rotational. The direction and distance travelled by the cursor was predictable and expected.

One participant described a preference of rotational over positional:

I did not have to move my upper body to get the cursor to go to my desired direction. I stayed more relaxed and focused instead of figuring out how to position my shoulders, etc.

Overall, participants praised positional navigation for stability, smoothness, and controllability. The rotational navigation method was criticized for its occasional jitters, unexpected tracking drops, and extra effort to move the cursor.

When asked to rate the difficulty of each method on a Likert scale from 1 (not difficult) to 10 (extremely difficult), the positional method received an average rating of 2.9 while the rotational method received an average rating of 7.2.

Seven of the 12 participants preferred the dwell selection method; four preferred smile; one preferred blink. Although smile selection produced the highest throughput, it wasn't necessarily the crowd favorite. Many participants preferred dwell selection as there was no added effort required. Participants criticized the effort required for blinking and smiling. It seems the initial novelty of interacting with facial gestures wore off: After the experiment, they described the process of using facial gestures as more error prone and fatiguing.

6 Conclusion

Facial tracking radiates with potential to become an accessibility feature on modern mobile devices, allowing access to people with deficient motor control in the hands and arms but with full motor control of the head, neck, and face. Our study showed that facial tracking, given supporting hardware, has potential as an input method relying on position, rotation, and facial gestures to simulate touch gestures. The combination of positional tracking and smile selection performed best on the dependent variables throughput (0.60 bps) and movement time (4383 ms). The positional+smile combination also yielded the lowest number of tracking drops (0.19 per sequence of 9 trials). For error rates, positional+smile and positional+blink were similar (11.6% and 11.1%, respectively).

Participants overall expressed as preference for positional tracking over rotational tracking. Seven of 12 participants preferred dwell selection. Despite taking longer, dwell selecting was considered less fatiguing than smile or blink selection.

FittsFace was developed as an ISO-conforming tool to study the multidimensionality of facial tracking, observing different attributes and parameters that make an efficient and user-friendly experience. The potential to use facial tracking as a form of assistive technology using pre-installed front facing cameras is not far off. *FittsFace* allows us to step back and examine the different elements and characteristics of what constitutes a facial tracking input method that works well with users.

6.1 Future Work

Our research is open to future work. One idea is to use facial gestures as conventional touch-based events. As an example, a scroll event, normally triggered by a swipe touch gesture, could be implemented using facial tracking by closing one eye and flicking the head downward or upward. We aim to further extend our research by running the *Fitts-Face* experiment on users with motor disabilities. The project will be open sourced to encourage other similar HCI research and projects using facial tracking on mobile devices.

References

1. Cuaresma, J., MacKenzie, I.S.: A comparison between tilt-input and facial tracking as input methods for mobile games. In: Proceedings of the IEEE-GEM 2014, pp. 70–76. IEEE, New York (2014)
2. Gizatdinova, Y., Špakov, O., Surakka, V.: Face typing: vision-based perceptual interface for hands-free text entry with a scrollable virtual keyboard. In: IEEE WACV 2012, pp. 81–87. IEEE, New York (2012)
3. ISO, Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) - Part 9: Requirements for Non-keyboard Input Devices (ISO 9241-9), International Organisation for Standardisation Report Number ISO/TC 159/SC4/WG3 N147 (2000)
4. Javanovic, R., MacKenzie, I.S.: MarkerMouse: mouse cursor control using a head-mounted marker. In: Miesenberger, K., Klaus, J., Zagler, W., Karshmer, A. (eds.) ICCHP 2010. LNCS, vol. 6180, pp. 49–56. Springer, Heidelberg (2010). doi:[10.1007/978-3-642-14100-3_9](https://doi.org/10.1007/978-3-642-14100-3_9)
5. Scott MacKenzie, I.: Fitts' throughput and the remarkable case of touch-based target selection. In: Kurosu, M. (ed.) HCI 2015. LNCS, vol. 9170, pp. 238–249. Springer, Cham (2015). doi:[10.1007/978-3-319-20916-6_23](https://doi.org/10.1007/978-3-319-20916-6_23)
6. Magee, J.J., Wu, Z., Chennamaneni, H., Epstein, S., Theriault, D.H., Betke, M.: Towards a multi-camera mouse-replacement interface. In: PRIS 2010, 10 p. (2010)
7. Manresa-Yee, C., Ponsa, P., Varona, J., Perales, F.J.: User experience to improve the usability of a vision-based interface. *Interact. Comput.* **22**, 594–605 (2010)
8. Manresa-Yee, C., Varona, J., Perales, F.J., Salinas, I.: Design recommendations for camera-based head-controlled interfaces that replace the mouse for motion-impaired users. *UAIS* **13**, 471–482 (2014)
9. Perini, E., Soria, S., Prati, A., Cucchiara, R.: FaceMouse: a human-computer interface for tetraplegic people. In: Huang, T.S., Sebe, N., Lew, M.S., Pavlović, V., Kölsch, M., Galata, A., Kisačanin, B. (eds.) *Computer Vision in Human-Computer Interaction*, pp. 99–108. Springer, Berlin (2006)
10. Qualcomm, Snapdragon SDK for Android. <https://developer.qualcomm.com/mobile-development/add-advanced-features/snapdragon-sdk-android>. Accessed 28 Mar 2016
11. Roig-Maimó, M.F., Manresa-Yee, C., Varona, J., MacKenzie, I.Scott: Evaluation of a mobile head-tracker interface for accessibility. In: Miesenberger, K., Bühler, C., Penaz, P. (eds.) ICCHP 2016. LNCS, vol. 9759, pp. 449–456. Springer, Cham (2016). doi:[10.1007/978-3-319-41267-2_63](https://doi.org/10.1007/978-3-319-41267-2_63)
12. Soukoreff, R.W., MacKenzie, I.S.: Towards a standard for pointing device evaluation: perspectives on 27 years of Fitts' law research in HCI. *Int. J. Hum. Comput. Stud.* **61**, 751–789 (2004)

13. Varona, J., Manresa-Yee, C., Perales, F.J.: Hands-free vision-based interface for computer accessibility. *J. Net. Comp. Appl.* **31**, 357–374 (2008)
14. Villaroman, N.H., Rowe, D.C.: Improving accuracy in face tracking user interfaces using consumer devices. In: *Proceedings of the RIIT 2012*, pp. 57–62. ACM, New York (2012)