Squint-to-Drag: Exploring Hands-free Eye-based Drag-and-Drop Interaction

Anonymous Author(s)*

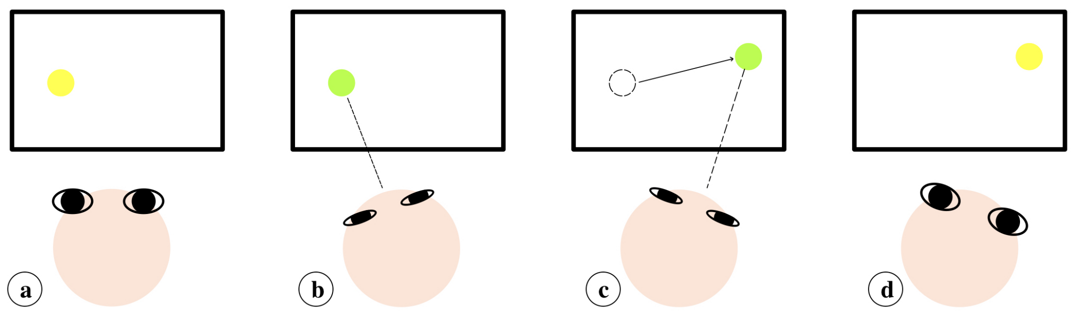

Figure 1: Squint-to-Drag. (b) squint on the target to select; (c) turn the head while squinting to move the target; (d) relax the eyes to drop the target.

ABSTRACT

Drag-and-drop is a fundamental interaction in desktop and mobile user interfaces. However, due to limited expressiveness of hands-free interaction techniques, there is currently no solutions other than explicitly selecting a button or menu to activate and deactivate drag and drop. This paper explores hands-free drag-and-drop interaction techniques, and proposes squinting as a viable technique. We evaluated squinting, voluntary blink, and dwell-time selected menu, and showed that while menu had the lowest error rate, squinting had significantly faster throughput and was most preferred by users.

CCS CONCEPTS

• **Human-centered computing → Gestural input**.

KEYWORDS

drag and drop, hands-free, blink interaction

Conference'17, July 2017, Washington, DC, USA
© 2022 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Anonymous Author(s). 2022. Squint-to-Drag: Exploring Hands-free Eye-based Drag-and-Drop Interaction. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Drag-and-drop is a fundamental interaction in modern desktop and mobile user interfaces. On the desktop, the dragging is performed by holding down a button and moving the pointer, then releasing the button to drop the object. On Android and iOS, it is by long pressing the touchscreen and moving the finger to drag, then lifting the finger to drop.

However, due to limited expressiveness of hands-free interaction techniques, there is no equivalent gesture for "holding down" nor "long pressing". Thus, current hands-free solution to dragging uses a menu or button [1] with selection gesture for mode switching, which require extra movement and efforts.

In this paper, we explored relevant works on eye-based interaction along with facial gesture, looking for a suitable gesture. Based on prior works, we'd like to explore squinting as a drag-and-drop technique for the following two reason:

- It provides semantic mappings to the drag-and-drop action.
- It's a gesture high success rate.

, conducting user study to compare it with voluntarily blink and dwell menu selection.

As pointing interaction performance is usually evaluated in terms of speed, accuracy, and throughput [5, 14], where throughput is a composite measure based on both speed and accuracy, the same criteria can be also applied to dragging interaction [21]. Our user study implement ISO 9241-9 multi-directional tapping test [41] with dragging target to evaluate the drag-and-drop performance of each input method, as well as workload and user preference.

Also, since the drag-and-drop action includes 2 state, which is "pointing and selecting" and "dragging and dropping". That is, the result will be calculated separately, while the comparison will be discussed.

2 RELATED WORK

We discuss relevant prior works in the area of eye-based interaction and facial gestures.

Eye-based Interaction

Previous works on eye-based interaction have been proposed in the eighties for cursor movement and target selection to improve the usability of computer systems for users with limited motor control [3, 9, 12].

Researchers have explored applications of cursor movement using gaze input, such as in-game AR controls [36] and basic computing tasks [24]. More specific research efforts have focused on subtle or low-effort selection of on-screen objects via smooth pursuits of eye movements [6, 24, 29, 38], adapting motion-path-based gesturing techniques into gaze interaction space [4, 23, 24, 37], and monitoring gaze patterns for informational purposes [7, 26].

As for target selection, intentional blink [27] have been widely studied and employed as alternative to keyboard [2, 16] and mouse input [22, 31]. Researchers have also explored using eye tracking and gaze pointing with dwell time (or fixation) to trigger selection [10, 12, 33]. Komogortsev et al. [14] proposed saccade selection, which showed 57% faster and 1.9 times greater throughput, but 3 times higher error rate than dwell time selection.

Facial Gestures

Many prior works regarding facial gestures as input modality focused on selection mechanism. Silva et al. [30] proposed vision-based algorithm allowing users to enter a click by opening their mouths, and evaluated it with a text input system. Huang et al. [8] employed facial EMG signal to perform mouse clicking and movement. Surakka et al. [32] and Rantanen et al. [28] have studied voluntary frowning and smiling as selection mechanisms combined with gaze pointing. Tuisku et al. [35] analyzed throughput of three facial activities, frowning, raising eyebrow, and smiling, and further used smiling for text entry input [34].

Some works considered the contraction of muscle when facial actions were performed, and explored its mapping of input command. Lyons [20] reviewed using the area of open mouths to control sound distortion in music performance and brush parameters in digital painting. Ku et al. evaluated users' preference and ability to perform 12 eye expressions [17], and further investigated the semantic mapping of eye expressions [18].

While these works have shown the potential of using eye-based interaction and facial gesture as input mechanism, drag-and-drop technique are often treated as combination of techniques for selection and movement, and designed in the context of mapping mouse function onto hands-free interaction [8, 15, 39]. For example, Tu et al. [13] showed that users selected a card by opening their mouths, and dragged a card by moving their heads with mouths opened when playing Solitaire and Minesweeper.

3 USER STUDY

We compare three methods for gaze-based hands-free drag and drop. Following previous works, we evaluate intentional blink and dwell for selection with using head pointing for movement as head movement was more deliberate and accurate then gaze pointing [19]. Dwell provides higher accuracy than saccade selection [14] and is the default drag-and-drop implementation in Tobii eye tracker [1], so we also implement dwell in first selecting a toggle in a menu to turn dwell from triggering selection to triggering drag-and-drop and then hold or drop objects with selection as Tobii eye tracker does.

Besides, according to [18], squint could be associated with decrease and focus, we propose using squint as a novel hands-free drag-and-drop technique. The detailed procedure of three methods are described below:

- Squint: partially close the eyes to select the movable object at the cursor position, keep squinting as to hold the object which follows the cursor, and relax the eyes to drop the object at the desired position.
- Blink: voluntarily blink the eyes to select the movable object at the cursor position, then the object follows the cursor, and voluntarily blink again to drop the object at the desired position.
- Menu: choose the drag-and-drop toggle in the menu using 300 ms uniform dwell time, select the movable object using 500 ms uniform dwell time, then the target will follow the cursor, and then drop the target at the desired position using 500 ms uniform dwell time again.

To evaluate the performance and workload of drag and drop with squint, blink and menu, we conducted a user study modified from ISO 9241-9 multi-directional tapping test [41].

Participants

We recruited 17 participants (8 female) aged from 20 to 25 (average 22.0), with 12 wearing glasses. All the participants are capable of performing all 3 methods without previous experience on the system, and no participants had eye-related disabilities.

System Design and Implementation

The study was implemented in an iOS app on a 12.9-inch iPad Pro. We used iPad's front-facing TrueDepth camera for face tracking and facial action detection with built-in library ARKit [11]. Its screen was projected to a 65-inch 4K television to increase head rotation, but participants are prohibited from looking at the display of iPad Pro.

Figure 2 illustrates the whole system setup. Users' eyes are 57-cm away from the center of the television and 20-cm away from the front camera of iPad.

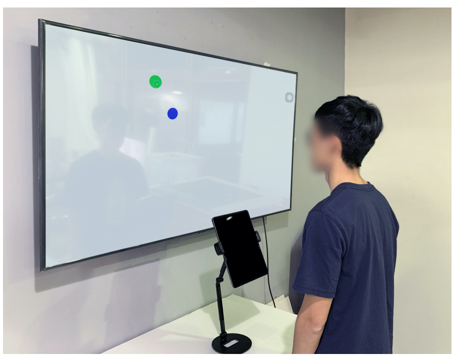

Figure 2: User Study Setup.

Procedure

We modified ISO 9241-9 test to from selection to evaluate drag and drop. The test was composed of 96 trials, with 3 methods (squint, blink and menu) randomly decided for every participant, 2 target distances (10° and 20°, or 180 px and 360 px), 2 object sizes (3.5° and 7°, or 90 px and 120 px), and 8 targets equally positioned away from the center forming a circle. In each trial, users were asked to drag a yellow circular object from the center of the screen to the destination where a blue circular target was placed. The order of next destinations to be targeted followed the original test.

Participants practiced 8 trials at the beginning of every method. As squint detection is greatly affected by users' eye sizes and preference, we customized the squint threshold with decision stump from data collected through practice. For menu and blink, we also adjusted dwell time and blink threshold according to users' request.

We recorded timestamp and position of each triggered selection and drop, as well as the position of the cursor every frame. All participants were also asked to self-report any incorrect system respond during the study.

A post-experiment questionnaire was asked for some quantitative data, which includes NASA Task Load Index (NASA-TLX) survey on the workload for each method, user preference ratings on a 7-point Likert scale regarding "I would like to use this method for hands-free drag and drop", and some open questions to collect participants' views and preferences about the different input methods. After all three methods, participants are asked to rank the methods by their preference.

4 DRAG AND DROP PERFORMANCE EVALUATION

In prior study[14, 25], the performance of hands-free pointing interaction under ISO9241-9 is usually shown in terms of accuracy, speed, and throughput [5, 40], while accuracy can be evaluated by the position of each action, speed can be evaluated by the movement time and distance, and throughput is a ISO dependent measure in "bits per second" based on speed and accuracy, along with Fitts' Index of Performance as Equation 1 shows.

$$Throughput = ID_e/MT \tag{1}$$

where MT is the mean movement duration time in seconds and

$$ID_e = log_2(D/W_e + 1) \tag{2}$$

ID_e is the effective index of difficulty measured in "bits," while D is the distance to the target and W_e is the effective width of the target calculated by

$$W_e = 4.1333 \times SD \tag{3}$$

in which SD is the standard deviation of the distances between each selection position and the center of each target.

As for evaluating drag-and-drop interactions with ISO9241-9, since prior study[21] has clearly pointed out there's a difference in performance regarding the state of drag-and-drop (pointing and selecting target vs dragging and dropping target), in this paper, throughput and accuracy in the above two states will be calculated separately.

5 RESULT

Movement Time

Table 1 shows the average movement time of drag-and-drops action performed by the participants, which is the duration since the time the movable target is selected to the time it's dropped with different combinations of input method, target size, and target distance, (notice that the time for dwelling on the drag-and-drop toggle in the Menu method task is excluded here). It's clear that the movement time for Squint, 2.67 secs, is slightly shorter than the other two input methods by about 17%, while the other two methods share similar average movement time of 3.20 sec for Blink, and 3.14 sec for Menu. Pairwise comparison shows statistically significant differences between each of them with $p < 0.005$ by Wilcoxon test.

Table 1: Average duration time (in secs) of drag-and-drops (with standard deviation).

Target		Input Method		
Size	Distance	Squint	Blink	Menu
120px	180px	2.14 (0.97)	2.67 (1.48)	1.55 (2.80)
120px	360px	2.64 (1.48)	3.16 (1.57)	3.66 (1.39)
90px	180px	2.79 (1.65)	3.20 (1.80)	3.30 (1.21)
90px	360px	3.11 (1.56)	3.78 (1.92)	4.05 (1.84)
Average		**2.67s**	**3.20s**	**3.14s**

Accuracy

Table 2 shows the accuracy of the three input methods in terms of error rate by different state of drag-and-drop action, which is the total number of false select (selection outside the movable target) and false drop (dropping outside the required target) divided by the number of required trials respectively. As the error rate shows, Menu provides the best accuracy among the three input techniques, while Blink is slightly better than Squint. Pairwise comparison shows statistically significant differences between each of them with $p < 0.05$ by Wilcoxon test.

The reason that Squint and Blink encounter more errors are also reported by user in the open questions that 4 of the participants have mentioned that "The selection of squint is easily triggered" and nearly have of the participants have stated that "when blinking it's likely to lost the curson position."

Table 2: Error rate of three interaction methods in different state of drag-and-drop (with standard deviation).

	Error Rate	
Input Method	Selecting	Dropping
Squint	18% (3.66)	17% (4.8)
Blink	10% (2.00)	16% (3.53)
Menu	8% (3.50)	5% (1.19)

Throughput

Table 3 shows the performance of the three input methods in terms of throughput by different state of drag-and-drop action according to Equation 1. Squint input method provide an average throughput of 1.34 bps (SD=0.44) for selecting the target and 1.71 bps (SD=0.5) for dropping the target, which is 26% better than the performance of Menu method of 1.17 bps (SD=0.37) and 1.35 (SD=0.23) bps in selecting and dropping respectively, while the blink method is 1.04 bps (SD = 0.37) and 1.28 bps (SD=0.40). Pairwise comparison shows statistically significant differences between each of them with $p < 0.005$ by Wilcoxon test.

Table 3: Throughput of three interaction methods in different state of drag-and-drop (with standard deviation).

	Throughput (bps)	
Input Method	Selecting	Dropping
Squint	1.34 (0.44)	1.71 (0.50)
Blink	1.04 (0.37)	1.28 (0.40)
Menu	1.17 (0.37)	1.35 (0.23)

However, comparing the selecting and dropping of the same input method, only Squint provides significance difference with $p < 0.0001$, which we believe is because of the difference eye expression of Squint's selecting (squinting) and dropping (relaxing), while others have the same expressions in both state. This result is consistent with Mackenzie's et. al findings [21]

Task Load

The result from the NASA-TLX questionnaire showed differences for perceived task load between methods. As no time pressure was given in the task, *Temporal Demand* was eliminated from the questionnaire. The result of pairwise comparisons between methods is shown in Figure 3.

Among the 3 methods, Squint had the lowest task load for all attributes. However, comparing Menu, it can be noted that there were no significant differences between conditions. Blink had a significant differences ($p < 0.05$) to the other two methods in *Effort* and the overall mean.

User Response

Likert Scale Points. Participants' response to "I would like to use this input method for hands free drag-and-drop tasks" in terms of average points of 7-point likert scale, is 5.06 for Squint, 4.71 for Blink, and 3.94 Menu.

Preference Rankings. Half of the participants rank the squint input method as their first choice, nearly another half rank menu as their first choice, and only one participant rank blink as her first choice. Only 3 participants rank Squint as their third choice, while there are 9 participants rank Blink as their third choice.

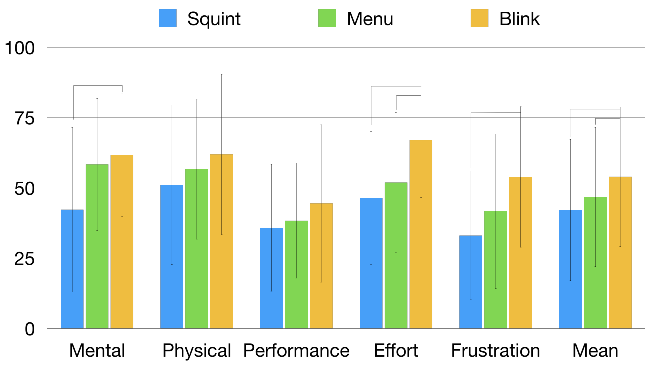

Figure 3: The mean responses for the attributes of NASA-TLX questionnaire. Error bars represent the standard deviations. Statistical significant differences are marked as connecting lines.

Table 4: System error rate, which is the number of user self-report system errors during the experiments for each input method divided by the number of required drag-and-drop trials.

Input Method	Errors Reported	System Error Rate
Squint	7	1.3%
Blink	25	4.8%
Menu	8	1.5%

System Error Rate

Table 4 shows the system error rate from the user-report errors. The error rate for the squint method and menu method is about 1/3 the error rate of the blink input method. As there are 8 trials for each round in our study, this error rate shows that for every 4 rounds (32 drag-and-drop trials), user may encounter 1.5 system errors with the blink method, and 0.5 with squint and menu.

6 DISCUSSION

Performance Comparison

Our outcome of the throughput is relatively lower than the prior study [14, 40**?**] while our movement time is relatively longer. We think this result is because of the sensitivity of the cursor which is reported by some participants that it require more effort to aim. However, under the same device settings, the performance is still comparable. In terms of accuracy, Menu is obviously a better method than Blink and Squint. However, throughput, the composite measure based on both speed and accuracy shows that Squint provide a better bps, which is consistent with our user response. We believe thus result is due to the effect of the movement time in Equation 1. Also, the extra movement time for user to trigger the drag-and-drop toggle require extra effort and workload for Menu method, which is excluded from our performance evaluation and mentioned in our user response. As a result, according to throughput, workload, and user ratings, Squint is a possible suitable interaction for hands-free drag-and-drop.

Evaluating drag and drop interaction

Compared with pointing interaction, drag-and-drop interaction is a 2-phased action and thus the comparison between different states should be explored. In this paper, the selecting and dropping state shows no significant difference for Menu and Blink methods, which is corresponding to the fact that both of the methods use the same gesture (dwell time and voluntarily blink, respectively). Also, in this paper, both selecting and dropping state for Squint method provide a consistent comparison result with other methods, which is easier for us to analyze and compare.

Future Work

In our study, squint provide a faster method for drag-and-drop in hands-free interaction compare to the conventional commercial dwell time Menu. Also, as one of our user mentioned, the squint method provides a striking resemblance with the mouse-based drag-and-drop since they are both continuous physical holding action. However, the Squint method may come with a trade-off, lower accuracy. That is, Squint may be a better solution for drag-and-drop in the faster condition that is more error tolerant, such as gaming.

We would like to further study more on hands-free interaction for drag-and-drop. Discover other continuous gesture, which might contain the usage of mouse or other facial expression, as it is a more intuitive approach and better provide a method for drag-and-drop interaction with lower error rate and user preference.

7 CONCLUSION

In this paper, we explore possible gaze-based interaction and proposed squint for hands-free drag and drop. Based on prior work, blink method and the conventional commercial dwell time menu method are chosen to be compared. We use the ISO 9241-9 test and evaluate each drag-and-drop method as a 2-stated interaction [21], which enable us to evaluate drag-and-drop performance with Fitts' Index of Performance through speed, accuracy, throughput, and user workload. The results indicate that the squint method is 26% better than the conventional menu method in terms of throughput and provides a 17% faster movement time along with most user agreement points.

By providing a significant decrease in movement time and increase in throughput, we expect squint to be a new possible mapping for hands-free drag-and-drop.

REFERENCES

[1] Tobii AB. 2020. Tobii Eye Tracker. https://www.tobii.com

[2] Behrooz Ashtiani and I. MacKenzie. 2010. BlinkWrite2: An improved text entry method using eye blinks. *Proc. ETRA 2010*, 339–345. https://doi.org/10.1145/1743666.1743742

[3] Richard A. Bolt. 1982. Eyes at the Interface. In *Proceedings of the 1982 Conference on Human Factors in Computing Systems (CHI '82)*. Association for Computing Machinery, New York, NY, USA, 360–362. https://doi.org/10.1145/800049.801811

[4] Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2009. Wearable EOG Goggles: Eye-Based Interaction in Everyday Environments. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 3259–3264. https://doi.org/10.1145/1520340.1520468

[5] Sarah A. Douglas, Arthur E. Kirkpatrick, and I. Scott MacKenzie. 1999. Testing Pointing Device Performance and User Assessment with the ISO 9241, Part 9 Standard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '99)*. Association for Computing Machinery, New York, NY, USA, 215–222. https://doi.org/10.1145/302979.303042

[6] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: Gaze Interaction for Smart Watches Using Smooth Pursuit Eye Movements. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software Technology (UIST '15)*. Association for Computing Machinery, New York, NY, USA, 457–466. https://doi.org/10.1145/2807442.2807499

[7] Seongwon Han, Sungwon Yang, Jihyoung Kim, and Mario Gerla. 2012. EyeGuardian: A Framework of Eye Tracking and Blink Detection for Mobile Device Users. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems Applications (HotMobile '12)*. Association for Computing Machinery, New York, NY, USA, Article Article 6, 6 pages. https://doi.org/10.1145/2162081.2162090

[8] Cheng-Ning Huang, Chun-Han Chen, and Hung-Yuan Chung. 2006. Application of facial electromyography in computer mouse access for people with disabilities. *Disability and Rehabilitation* 28, 4 (2006), 231–237. https://doi.org/10.1080/09638280500158349 PMID: 16467058.

[9] T. E. Hutchinson, K. P. White, W. N. Martin, K. C. Reichert, and L. A. Frey. 1989. Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics* 19, 6 (Nov 1989), 1527–1534. https://doi.org/10.1109/21.44068

[10] Aulikki Hyrskykari, Howell Istance, and Stephen Vickers. 2012. Gaze Gestures or Dwell-Based Interaction?. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 229–232. https://doi.org/10.1145/2168556.2168602

[11] Apple Inc. 2020. ARKit Developer Documentation. https://developer.apple.com/documentation/arkit

[12] Robert J. K. Jacob. 1990. What You Look at is What You Get: Eye Movement-Based Interaction Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '90)*. Association for Computing Machinery, New York, NY, USA, 11–18. https://doi.org/10.1145/97243.97246

[13] Jilin Tu, T. Huang, and Hai Tao. 2005. Face as mouse through visual face tracking. In *The 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*. 339–346. https://doi.org/10.1109/CRV.2005.39

[14] Oleg V. Komogortsev, Young Sam Ryu, Do Hyong Koh, and Sandeep M. Gowda. 2009. Instantaneous Saccade Driven Eye Gaze Interaction. In *Proceedings of the International Conference on Advances in Computer Enterntainment Technology (ACE '09)*. Association for Computing Machinery, New York, NY, USA, 140–147. https://doi.org/10.1145/1690388.1690412

[15] Ilkka Kosunen, Antti Jylha, Imtiaj Ahmed, Chao An, Luca Chech, Luciano Gamberini, Marc Cavazza, and Giulio Jacucci. 2013. Comparing Eye and Gesture Pointing to Drag Items on Large Screens. In *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces (ITS '13)*. Association for Computing Machinery, New York, NY, USA, 425–428. https://doi.org/10.1145/2512349.2514920

[16] Aleksandra Królak and Paweł Strumiłło. 2012. Eye-blink detection system for human–computer interaction. *Universal Access in the Information Society* 11, 4 (01 Nov 2012), 409–419. https://doi.org/10.1007/s10209-011-0256-6

[17] Pin-Sung Ku, Te-Yan Wu, and Mike Y. Chen. 2017. EyeExpression: Exploring the Use of Eye Expressions as Hands-Free Input for Virtual and Augmented Reality Devices. In *Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology (VRST '17)*. Association for Computing Machinery, New York, NY, USA, Article Article 60, 2 pages. https://doi.org/10.1145/3139131.3141206

[18] Pin-Sung Ku, Te-Yen Wu, and Mike Y. Chen. 2018. EyeExpress: Expanding Hands-Free Input Vocabulary Using Eye Expressions. In *The 31st Annual ACM Symposium on User Interface Software and Technology Adjunct Proceedings (UIST '18 Adjunct)*. Association for Computing Machinery, New York, NY, USA, 126–127. https://doi.org/10.1145/3266037.3266123

[19] Mikko Kytö, Barrett Ens, Thammathip Piumsomboon, Gun A. Lee, and Mark Billinghurst. 2018. Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, Article Paper 81, 14 pages. https://doi.org/10.1145/3173574.3173655

[20] M. J. Lyons. 2004. Facial gesture interfaces for expression and communication. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583)*, Vol. 1. 598–603 vol.1. https://doi.org/10.1109/ICSMC.2004.1398365

[21] I. Scott MacKenzie, Abigail Sellen, and William A. S. Buxton. 1991. A Comparison of Input Devices in Element Pointing and Dragging Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '91)*. Association for Computing Machinery, New York, NY, USA, 161–166. https://doi.org/10.1145/108844.108868

[22] Eric Missimer and Margrit Betke. 2010. Blink and Wink Detection for Mouse Pointer Control. In *Proceedings of the 3rd International Conference on PErvasive Technologies Related to Assistive Environments (PETRA '10)*. Association for Computing Machinery, New York, NY, USA, Article Article 23, 8 pages. https://doi.org/10.1145/1839294.1839322

[23] Emilie Mollenbach, John Paulin Hansen, Martin Lillholm, and Alastair G. Gale. 2009. Single Stroke Gaze Gestures. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. Association for Computing Machinery, New York, NY, USA, 4555–4560. https://doi.org/10.1145/1520340.1520699

[24] Emilie Møllenbach, John Paulin Hansen, and Martin Lillholm. 2013. Eye Movements in Gaze Interaction. *Journal of Eye Movement Research* 6, 2 (Aug. 2013). https://doi.org/10.16910/jemr.6.2.1

[25] Daniel Natapov, Steven J. Castellucci, and I. Scott MacKenzie. 2009. ISO 9241-9 Evaluation of Video Game Controllers. In *Proceedings of Graphics Interface 2009 (GI '09)*. Canadian Information Processing Society, CAN, 223–230.

[26] Anh Nguyen, Raghda Alqurashi, Zohreh Raghebi, Farnoush Banaei-kashani, Ann C. Halbower, and Tam Vu. 2016. A Lightweight and Inexpensive In-Ear Sensing System For Automatic Whole-Night Sleep Stage Monitoring. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM (SenSys '16)*. Association for Computing Machinery, New York, NY, USA, 230–244. https://doi.org/10.1145/2994551.2994562

[27] Kyung S. Park and Kyung T. Lee. 1996. Eye-controlled human/computer interface using the line-of-sight and the intentional blink. *Computers Industrial Engineering* 30, 3 (1996), 463 – 473. https://doi.org/10.1016/0360-8352(96)00018-6 IE in Korea.

[28] Ville Rantanen, Jarmo Verho, Jukka Lekkala, Outi Tuisku, Veikko Surakka, and Toni Vanhala. 2012. The Effect of Clicking by Smiling on the Accuracy of Head-Mounted Gaze Tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications (ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 345–348. https://doi.org/10.1145/2168556.2168633

[29] Simon Schenk, Marc Dreiser, Gerhard Rigoll, and Michael Dorr. 2017. GazeEverywhere: Enabling Gaze-Only User Interaction on an Unmodified Desktop PC in Everyday Scenarios. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. Association for Computing Machinery, New York, NY, USA, 3034–3044. https://doi.org/10.1145/3025453.3025455

[30] G. C. D. Silva, M. J. Lyons, S. Kawato, and N. Tetsutani. 2003. Human Factors Evaluation of a Vision-Based Facial Gesture Interface. In *2003 Conference on Computer Vision and Pattern Recognition Workshop*, Vol. 5. 52–52. https://doi.org/10.1109/CVPRW.2003.10055

[31] W. Siriluck, S. Kamolphiwong, T. Kamolphiwong, and S. Sae-Whong. 2007. Blink and Click. In *Proceedings of the 1st International Convention on Rehabilitation Engineering Assistive Technology: In Conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting (i-CREATe '07)*. Association for Computing Machinery, New York, NY, USA, 43–46. https://doi.org/10.1145/1328491.1328503

[32] Veikko Surakka, Marko Illi, and Poika Isokoski. 2004. Gazing and Frowning as a New Human–Computer Interaction Technique. *ACM Trans. Appl. Percept.* 1, 1 (July 2004), 40–56. https://doi.org/10.1145/1008722.1008726

[33] Geoffrey Tien and M. Stella Atkins. 2008. Improving Hands-Free Menu Selection Using Eyegaze Glances and Fixations. In *Proceedings of the 2008 Symposium on Eye Tracking Research Applications (ETRA '08)*. Association for Computing Machinery, New York, NY, USA, 47–50. https://doi.org/10.1145/1344471.1344482

[34] Outi Tuisku, Ville Rantanen, and Veikko Surakka. 2016. Longitudinal Study on Text Entry by Gazing and Smiling. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research Applications (ETRA '16)*. Association for Computing Machinery, New York, NY, USA, 253–256. https://doi.org/10.1145/2857491.2857501

[35] Outi Tuisku, Ville Rantanen, Oleg Špakov, Veikko Surakka, and Jukka Lekkala. 2014. Pointing and Selecting with Facial Activity. *Interacting with Computers* 28, 1 (07 2014), 1–12. https://doi.org/10.1093/iwc/iwu026 arXiv:https://academic.oup.com/iwc/article-pdf/28/1/1/6991708/iwu026.pdf

[36] Ying-Chao Tung, Chun-Yen Hsu, Han-Yu Wang, Silvia Chyou, Jhe-Wei Lin, Pei-Jung Wu, Andries Valstar, and Mike Y. Chen. 2015. User-Defined Game Input for Smart Glasses in Public Space. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3327–3336. https://doi.org/10.1145/2702123.2702214

[37] Vytautas Vaitukaitis and Andreas Bulling. 2012. Eye Gesture Recognition on Portable Devices. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp '12)*. Association for Computing Machinery, New York, NY, USA, 711–714. https://doi.org/10.1145/2370216.2370370

[38] Mélodie Vidal, Andreas Bulling, and Hans Gellersen. 2013. Pursuits: Spontaneous Interaction with Displays Based on Smooth Pursuit Eye Movement and Moving Targets. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '13)*. Association for Computing Machinery, New York, NY, USA, 439–448. https://doi.org/10.1145/2493432.2493477

[39] Yu-Luen Chen. 2001. Application of tilt sensors in human-computer mouse interface for people with disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 9, 3 (Sep. 2001), 289–294. https://doi.org/10.1109/7333.948457

[40] Xuan Zhang and I. Scott MacKenzie. 2007. Evaluating Eye Tracking with ISO 9241 - Part 9. In *Proceedings of the 12th International Conference on Human-Computer Interaction: Intelligent Multimodal Interaction Environments (HCI'07)*. Springer-Verlag, Berlin, Heidelberg, 779–788.

[41] J. E. Zucco, B. H. Thomas, and K. Grimmer. 2006. Evaluation of Four Wearable Computer Pointing Devices for Drag and Drop Tasks when Stationary and Walking. In *2006 10th IEEE International Symposium on Wearable Computers*. 29–36. https://doi.org/10.1109/ISWC.2006.286339