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

Ching-Yi Tsai*
Hsin-Ying Lee*
Mike Y. Chen
National Taiwan University

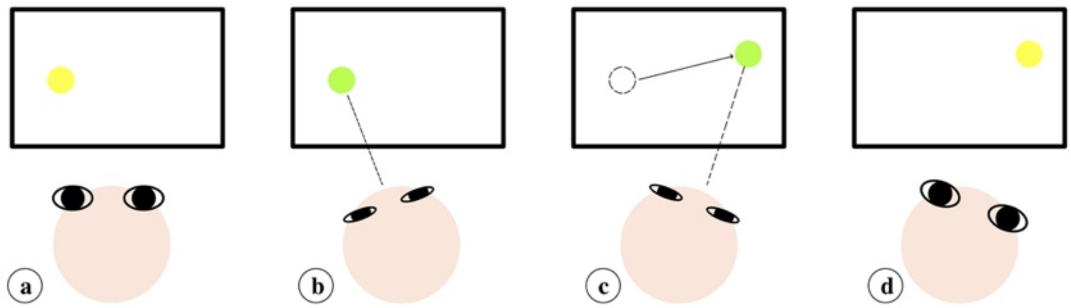

Figure 1: Squint-to-Drag. (b) squint on the target to select; (c) turn the head while squinting to move the target; (d) relax the eyes to drop the target.

ABSTRACT

Drag-and-drop is a fundamental interaction in desktop and mobile user interfaces. However, due to the limited expressiveness of hands-free interaction techniques, there is currently no solution 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 menus and showed that while the menus 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

1 INTRODUCTION

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

However, due to the limited expressiveness of hands-free interaction techniques, there is no equivalent gesture for "holding down" or "long pressing". The existing hands-free solution uses a menu or button [1] with a selection gesture for mode switching, which requires additional movement and effort.

Thus, we explore eye-based interaction and facial gestures in this work. By conducting a user study analyzing intentional blink, dwell menu selection, and squinting, we find squinting superior to the other two gestures. It provides semantic mappings to the drag-and-drop action and achieves a high success rate.

For the user study, we implement ISO 9241-9 multi-directional tapping test [41] with a dragging target to evaluate candidate techniques. We collect the performance, workload, and user preferences of each method. The performance is estimated with the same criteria as assessing pointing interaction, including speed, accuracy, and throughput [5, 14], which is a composite measure based on both speed and accuracy. Besides, since the drag-and-drop action comprises two states: "pointing and selecting" and "dragging and dropping", the results are calculated separately.

*Both authors contributed equally to this research.

2 RELATED WORK

We discuss relevant prior works on 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] has been widely studied and employed as an 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 three times higher error rate than dwell time selection.

Facial Gestures

Many prior works regarding facial gestures as input modalities focused on selection mechanisms. Silva et al. [30] proposed a vision-based algorithm allowing users to enter a click by opening their mouths and evaluating it with a text input system. Huang et al. [8] employed the 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 the throughput of three facial activities, frowning, raising eyebrows, and smiling, and further used smiling for text entry input [34].

Some works considered the contraction of muscles when facial actions were performed and explored its mapping of input commands. 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 user preferences and ability to perform 12 eye expressions [17]. They 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 mechanisms, drag-and-drop techniques are often treated as a 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 it by moving their heads with mouths open when playing Solitaire and Minesweeper.

3 USER STUDY

We propose using squint as a novel hands-free drag-and-drop technique for its association with decrease and focus [18], and its performance is evaluated by comparing against intentional blink and dwell in a user study. Following previous work, we consider moving by head pointing for blink and dwell since head movements are more deliberate and accurate than gaze pointing [19].

Since dwell is the default drag-and-drop implementation in the Tobii eye tracker [1], we implement dwell as the Tobii eye tracker: users first select a toggle in a menu to turn the effect of dwell from triggering selection into triggering drag-and-drop. Then they dwell on an object to hold or drop the object. The detailed procedures of the three methods are described below:

- Squint: partially close the eyes to select the movable object at the cursor position, keep squinting to hold the object following the cursor, and relax the eyes to drop the object at the desired position.
- Blink: intentionally blink the eyes to select the movable object at the cursor position; the object follows the cursor; 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, and select the movable object using 500 ms uniform dwell time; the target follows the cursor; 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 could perform all three methods without previous experience with 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 the iPad's front-facing TrueDepth camera for face tracking and facial action detection with the built-in library ARKit [11]. Its screen was projected to a 65-inch 4K television to increase head rotation, but participants were prohibited from looking at the display of the 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 the iPad.

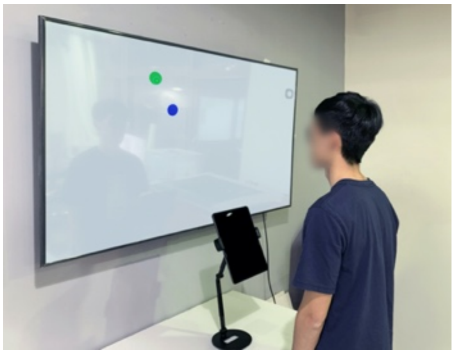

Figure 2: User Study Setup.

Procedure

We modified the ISO 9241-9 test from selection to evaluate drag and drop. The test was composed of 96 trials, with three methods (squint, blink, and menu) randomly decided for every participant, two target distances (10° and 20°, or 180 px and 360 px), two object sizes (3.5° and 7°, or 90 px and 120 px), and eight 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 eight trials at the beginning of every method. As squint detection is greatly affected by users' eye sizes and preferences, we customized the squint threshold with decision stump from data collected through practice. For the menu and blink, we also adjusted the dwell time and blink threshold according to users' requests.

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

A post-experiment questionnaire was asked for quantitative evaluation, including a 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 several 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]. Accuracy can be evaluated by the position of each action, speed can be evaluated by the movement time and distance, and throughput is an ISO-dependent measure in "bits per second" based on speed and accuracy, along with Fitts' Index of Performance:

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

where MT is the mean movement duration time in seconds. ID_e is the effective index of difficulty measured in "bits" and obtained via:

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

where 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 pointed out there is 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 from the time the movable target is selected to the time it is dropped, collected from 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). The movement time for squint, 2.67 secs, is slightly shorter than the other two input methods by about 17%, while the average movement times of the other two methods are similar: 3.20 sec for blink and 3.14 sec for menu. The pairwise comparison shows statistically significant differences between them with $p < 0.005$ by the Wilcoxon test.

Accuracy

Table 2 shows the accuracy of the three input methods in terms of error rate in different states of drag-and-drop action, which is the total number of false selects (selection outside the movable target) and false drops (dropping outside the required target) divided by the number of trials needed respectively. As the error rate shows, Menu provides

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

Target Size	Target 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**

the best accuracy among the three input techniques, while blink is slightly better than squint. The pairwise comparisons show statistically significant differences between them with $p < 0.05$ by the Wilcoxon test.

The higher error rates of squint and blink echo the response of the users in the open questions. Four of the participants mentioned that the selection of squint was easily triggered. Nearly half of the participants stated that the cursor positions were likely lost when they blinked.

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

Input Method	Error Rate: Selecting	Error Rate: 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: Throughput of three interaction methods in different states of drag-and-drop (with standard deviation).

Input Method	Throughput (bps): Selecting	Throughput (bps): 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)

Table 3 presents the performance of the three input methods in terms of throughput in different states of drag-and-drop action obtained via Equation 1. Squint method provides 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, 26% better than the performance of menu with 1.17 bps (SD=0.37) and 1.35 (SD=0.23) bps. The blink method shows 1.04 bps (SD = 0.37) and 1.28 bps (SD=0.40). The pairwise comparisons show statistically significant differences between each of them with $p < 0.005$ by the Wilcoxon test.

However, comparing the selecting and dropping of the same input method, only squint shows a significant difference with $p < 0.0001$, which we speculate is due to the difference in eye gestures of selecting (squinting) and dropping (relaxing). In contrast, others have the same gesture in both states. This result is consistent with the findings of Mackenzie et al. [21]

Task Load

The result from the NASA-TLX questionnaire showed the differences in 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 three 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 difference ($p < 0.05$) to the other two methods in *Effort* and the overall mean.

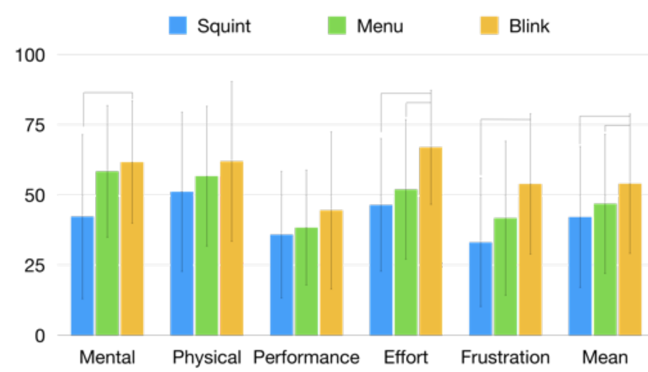

Figure 3: The mean responses for the attributes of the NASA-TLX questionnaire. Error bars represent the standard deviations. Statistical significant differences are marked as connecting lines.

User Response

Likert Scale Points. Participants' responses to "I would like to use this input method for hands-free drag-and-drop tasks" in terms of average points of a 7-point Likert scale is 5.06 for squint, 4.71 for blink, and 3.94 for 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 three participants ranked squint as their third choice, while nine participants rank blink as their third choice.

System Error Rate

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%

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 eight trials for each round in our study, this error rate shows that for every four rounds (32 drag-and-drop trials), a user may encounter 1.5 system errors with the blink method and 0.5 with squint and menu.

6 DISCUSSION

Performance Comparison

The estimated throughput is relatively lower than the prior study [14, 40], while the movement time is somewhat longer. This could be attributed to the sensitivity of the cursor, which requires more effort to control, as reported by some participants. However, under the same device settings, the performance is still comparable. In terms of accuracy, menu is a better method than blink and squint. However, throughput, the composite measure based on both speed and accuracy, shows that squint provides better bps, which is consistent with our user response. We believe this result is due to the effect of the movement time in Equation 1. Also, the extra movement time for a user to trigger the drag-and-drop toggle requires extra effort and workload for the 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 the menu and blink methods, corresponding to the fact that both methods use the same gesture (dwell time and intentional blink, respectively). Also, in this paper, both selecting and dropping states for the 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 provides a faster method for drag-and-drop in hands-free interaction compared to the conventional commercial dwell time menu. Also, as one of our users mentioned, the squint method resembles the mouse-based drag-and-drop since they are both continuous physical holding actions. However, the squint method may come with a trade-off of lower accuracy. We suggest squint can be a better solution for drag-and-drop in faster, more error-tolerant conditions, such as gaming.

We would like to study more on hands-free interaction for drag-and-drop and discover other continuous gestures, probably with the usage of a mouse or other facial expressions, as they are more intuitive and have the potential to provide lower error rates.

7 CONCLUSION

In this paper, we explore possible gaze-based interaction and propose squint for hands-free drag and drop. Based on prior work, the blink method and the conventional commercial dwell time menu method are compared. We use the ISO 9241-9 test and evaluate each drag-and-drop method as a 2-stated interaction [21], which enables 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. In *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