GazeBar: Exploiting the Midas Touch in Gaze Interaction

Carlos Elmadjian
University of São Paulo
São Paulo, Brazil
elmad@ime.usp.br

Carlos H. Morimoto
University of São Paulo
São Paulo, Brazil
hitoshi@ime.usp.br

ABSTRACT

Imagine an application that requires constant configuration changes, such as modifying the brush type in a drawing application. Typically, options are hierarchically organized in menu bars that the user must navigate, sometimes through several levels, to select the desired mode. An alternative to reduce hand motion is the use of multimodal techniques such as *gaze-touch*, that combines gaze pointing with mechanical selection. In this paper, we introduce GazeBar, a novel multimodal gaze interaction technique that uses gaze paths as a combined pointing and selection mechanism. The idea behind GazeBar is to maximize the interaction flow by reducing "safety" mechanisms (such as clicking) under certain circumstances. We present GazeBar's design and demonstrate it using a digital drawing application prototype. Advantages and disadvantages of GazeBar are discussed based on a user performance model.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; *Interface design prototyping*.

KEYWORDS

gaze interaction, multimodal interaction, interface modes

ACM Reference Format:
Carlos Elmadjian and Carlos H. Morimoto. 2021. GazeBar: Exploiting the Midas Touch in Gaze Interaction. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts (CHI '21 Extended Abstracts), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3411763.3451703

1 INTRODUCTION

The concept of flow [27] regards a highly focus mental state where users are fully immersed in the primary task they are performing. It is highly desirable to create interaction designs that allow users to "flow". In typical computer applications, such as text editors, users can fully focus on writing (the primary task) after they become comfortable using the interface. Typical graphical user interfaces (GUIs) use hierarchical menus for controlling and configuring an application and frequently used resources are placed in menu bars for quick access. Easy transition between resources or interaction modes provided by the interface facilitates the user to achieve a state of flow.

The transition between modes can disturb productivity though, as users have to temporarily relinquish their primary task and navigate through the interface to select the desired mode (for example, change the font) through a sequence of mouse motions and clicks or touches on a touch-sensitive screen. This results in an interruption of task flow, something that expert users learn to mitigate using, for example, shortcuts such as CTRL-c for "copy" and CTRL-v for "paste". Unfortunately, shortcuts must be limited and cannot be applied to all occasions. Even though expert users can customize shortcuts, interfaces must provide other forms of acceleration whenever possible, for example, through multimodal interaction, such as touch and speech.

Another alternative is the use of eye gaze, captured by an eye tracking device, to accelerate pointing and menu navigation. Though pointing with our eyes can be very natural and easily integrated in a computer interface, target selection still poses many challenges due to the risk of performing an activation just by looking at a target, which is known as the Midas touch problem (MTP) [12]. Dwelling at a target until it gets selected is the most common method used to circumvent the MTP. It is not ideal though: short dwell times might still cause involuntary selections and long dwell times slow the interaction. Using gaze for pointing and some other mechanism for selection, such as touching or clicking a button [6], has been suggested as a natural multimodal method robust to the MTP.

In this paper we argue that mechanisms such as dwelling and touch, created as safety measures to avoid the MTP, are like breaks that can be removed from the interaction process under certain circumstances so that the user can achieve higher flow states. To demonstrate this idea we propose GazeBar, a hierarchical menu interface where the options are triggered by "just looking" at them. GazeBar options and the bar itself are activated by context: depending on where the user is looking at, a different menu is shown or simply not shown if there are no actions to be selected by gaze, therefore reducing the chances of unintended selections.

We have designed GazeBar to keep manual and gaze inputs as independent and concurrent interaction modes. Manual input (from keyboard, mouse, or stylus) serves the primary task, while gaze controls menu selections by just looking. Because the user is not required to dwell or click, GazeBar provides an easier interaction with fewer steps than other gaze interaction methods, facilitating the user to achieve a higher flow state.

2 BACKGROUND AND RELATED WORK

Eye trackers are devices that capture eye movements typically using video cameras and they usually require a calibration procedure to estimate the user's point of gaze on the computer screen [24].

CHI '21 Extended Abstracts, May 8–13, 2021, Yokohama, Japan

ACM ISBN 978-1-4503-8095-9/21/05…$15.00
https://doi.org/10.1145/3411763.3451703

dwell-time [5, 19, 25, 43]	look and wait until the target is selected
context-switch/goal-crossing [11, 23, 37]	look and saccade in a certain direction to select the focused key
eye-swipe [18, 28]	look at every key and then select using a short dwell
eye pursuit [7, 14, 42]	look and follow until the target is selected
swipe and touch [16, 17]	look at every key and then select with a manual input
gaze-touch [4, 29, 31, 32]	look and select with a manual input
gaze-shifting [30, 31]	look within or outside the manual input area then manual select

Table 1: Selection mechanisms applied in gaze interaction. White rows show gaze-only techniques while gray ones are multi-modal.

Modern low-cost commercial eye trackers are now commonly available [38], though some early challenges still remain. Jacob [12] coined the term "Midas touch problem" when developing early gaze interaction methods. He pointed out that it might feel empowering to have something activated just by looking at it, but it eventually gets impossible look anywhere without issuing a command. To solve this issue, he suggested the use of short dwell times (about 250 ms) over a button for target selection.

Eye typing (or text-entry by eye gaze) seems like a natural application to understand the evolution of gaze interaction since it demonstrates and compares the performance of very different gaze selection mechanisms [26]. Basic key selection using a fixed dwell time (typically between 500 and 1000 ms) has been used as a standard technique due to its simplicity to implement and use [20]. Some extensions to dwell-time interfaces [5, 19, 25, 43] focused more on increasing the performance of eye typing, while others suggested adaptive dwell times [19, 43], improving both user experience and text entry rate.

2.1 Gesture-based selection

Many extensions of dwell time show how users can type faster with short dwells and still feel under control. So why not set the dwell time to zero? Kristensson and Vertanen [15] showed the potential speed gain of an ideal dwell-free virtual keyboard. They presented a model that decomposes the text entry rate into dwell time and overhead time. The overhead time is defined as the time needed to transition between keys and to perform error corrections. This has inspired eye swipe gestures as proposed in [18, 28].

A hierarchical alternative to swipe gestures is pEYEwrite [11], that uses circular pie menus to display letter groups in each pie. As the user enters an outer pie border region, called *selection border*, a second level pie menu is displayed with the letters redistributed in different pies. A letter is selected by crossing the outermost border of a leaf node, containing a single character. Alternatively, Context Switching [23, 37] suggests the use of single saccades (quick eye movements) for selection. The idea is to use duplicated contexts, and the focused key in one context is selected by quickly switching the gaze to the other context. Thus the user can freely explore the keyboard before deciding to enter a character.

More recently, the use of eye pursuits for gaze interaction have gained great interest because of its potential to use non-calibrated eye trackers, though early pursuit methods, such as Dasher [44], requires calibration. The idea of non-calibrated pursuits is to display moving objects along known trajectories and perform a particular selection when the eyes follow a corresponding target for a certain time [14, 42].

2.2 Multimodal Gaze Interaction

Multimodal interaction combines gaze pointing with some non-gaze selection mechanism, such as touch [29], click [4], or even brain interfaces [45]. One of the earliest works that showed benefits of combining mouse and gaze was MAGIC Pointing [46]. This idea heavily inspired other designs, such as the one proposed by Stellmach and Dachselt for target acquisition, following the principle of *gaze suggests, touch confirms* [36].

With the popularization of touch interfaces, Pfeuffer et al. investigated how to seamlessly combine gaze and touch interaction [29, 31]. This was further explored in the context of tablets, in which the thumb used for holding the device is used in combination with gaze to provide whole-screen reachability with minimum hand movement [32].

Another multimodal application is to modulate interface behavior with gaze. Some examples are *gaze-shifting* [30], in which manual input is processed differently according to direct or indirect gaze, and GazeButton [33], a technique that enhances the expressiveness of a single button element based on gaze context.

More recently Kumar et al. proposed TouchGazePath for PIN entry [16] and TAGSwipe [17], leveraging eye swipe plus manual input for confirmation, while Creed et al. introduced the *Sakura* application for creative design, in which gaze is used for pointing and selection is performed by a mechanical device [4]. Table 1 summarizes the basic actions explored by these methods.

In the next section we describe the design of GazeBar, both a novel interface and gaze interaction technique that exploits the original concept of "just-look" for target selection, i.e., no dwell, gesture, touch, or click is required when combined with a manual input for the primary task.

3 GAZEBAR DESIGN

GazeBar interaction design was inspired by MAGIC Pointing [46] and takes advantage of the spontaneous and characteristic gaze paths made by the user when switching modes in GUIs. Figure 1 illustrates four steps required to switch modes using GazeBar.

First, we assume there is an active mode (blue button in the GazeBar shown in the leftmost picture) and the user is focusing on his or her primary task, which is located within the interface central area. When a change of mode is desired, the user directs his or her eyes towards a bar on the bottom edge of the screen, looking

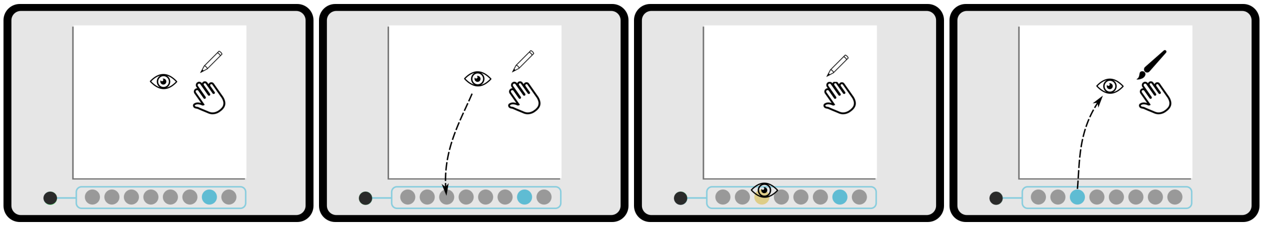

Figure 1: Steps required to switch modes using GazeBar. When the user decides to switch modes, she looks at the GazeBar and hovers over different options on the menu (in yellow). The last gazed option is selected when her gaze leaves the GazeBar.

for the appropriate mode option. While gaze is within the bar area, the bar visually indicates which mode is being targeted (yellow button). Upon locating the desired mode, the user looks back at the central area, resuming the primary task. The current active mode is defined by the last gazed option, and a short visual feedback is shown to the user. Different than gaze-and-touch techniques, no manual confirmation is required.

To give the impression to the user that safe selections are done by simply looking at a button, we use a trigger mechanism similar to reverse-crossing [8]: once the gaze is captured in the GazeBar area, a selection is only confirmed after leaving it, allowing the user to freely navigate in the bar area. To determine which mode option is to be set, we resort to the minimum distance between the estimated gaze point and a button, so no dwell time is required.

Modes in GazeBar are sorted hierarchically, so that the state of secondary modes only become available if the parent's mode is selected first, as shown in Figure 2. With respect to the interactive logic, this is somehow similar to the idea of hierarchical pie menus, or pEYEs [39], though our design presents stark differences in visualization and functionality.

The content and appearance of GazeBar is also context-sensitive, meaning that if the user is gazing at an application for which there are mode options available, a corresponding GazeBar will pop-up on screen. But no GazeBar is shown if the gaze context contains no known applications.

Lastly, we do not enforce any coordination between gaze and manual input. Unlike other multimodal gaze-based approaches, we decouple the primary task from the mode-switching task, making manual input solely responsible for the former, and gaze for the latter, promoting a higher state of flow.

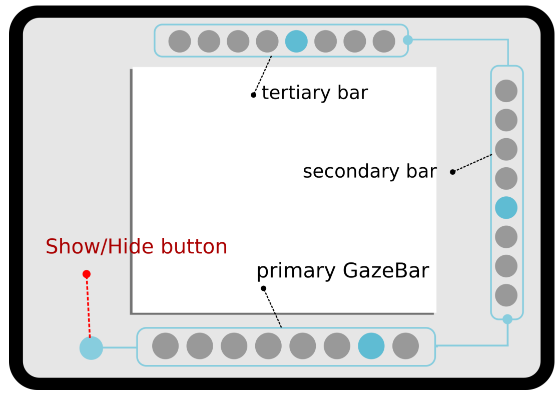

Figure 2: GazeBar options are sorted hierarchically. Root mode options are always at the bottom. One option can trigger a secondary bar, which, by its turn, can trigger another secondary bar. Gazebar interface design supports at most four levels of submenus.

3.1 Interface design

The use of gaze on GUIs imposes several constraints. In particular, gaze-based interfaces have to at least account for eye tracking accuracy and eye jittery. Mode buttons in GazeBar were created to roughly span 2^o on screen, which is above the higher bound gaze estimation error found in most commercial eye trackers. Due to eye jittery, we also place buttons relatively apart from each other.

Gaze focus over a mode button is determined by spatial hysteresis [10]. We say that the user is gazing at a button if the Euclidean distance between the estimated gaze point G and the button center M is less than a empirically determined threshold d. But we say the user stopped gazing if this distance is greater than $2 \times d$, as indicated by Figure 3. This is used to avoid involuntary switches due to eye tracking instability.

Also, while gazing a mode button, GazeBar highlights its color and the target is expanded. Expanding a target is a way to secure gaze focus detection, since the target spans over a larger screen area and its neighbours are pushed away, lessening selection ambiguity.

Involuntary selections, although still possible, are mitigated by interface design. The bar's one-dimensional format and its placement on screen edges make it more unlikely for accidental gaze incursions, as the user's primary task is located at the central part of the screen. Also, GazeBar and its secondary bars can always be deactivated at user's will (Figure 2). As for involuntary changes, they can always be avoided by fixating over the current active mode before leaving the bar area.

To minimize the use of screen space, secondary menus are always presented as another bar, and these bars are also placed near the edges of the screen. The hierarchical structure of menus and submenus is demarcated by lines connecting the parent mode option with the child bar. By default, the current path chosen by the user in the option tree is always visible, which means that multiple bars might be shown on screen at the same time. This is done in order to speed up activation of secondary options without impairing the visibility of the primary task.

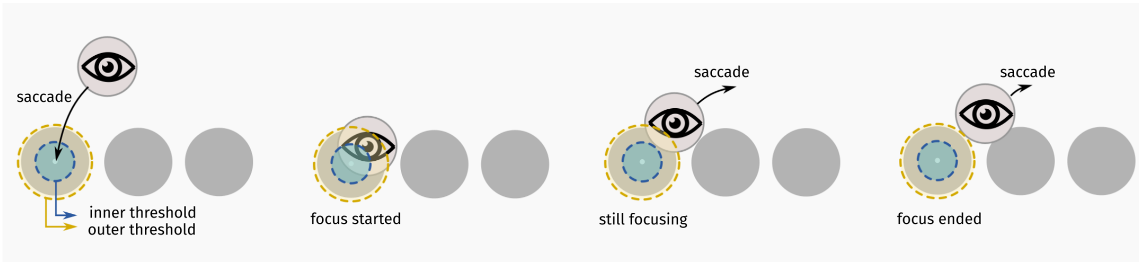

Figure 3: GazeBar uses an implementation of spatial hysteresis [10] based on two criteria to tell when gaze focus over a target starts and ends. Once a gaze estimate is trapped inside a target, GazeBar only considers it out with a larger threshold. This avoids involuntary selections due to eye tracker fidelity or eye jittery.

4 GAZEBAR PROTOTYPE

We have built a prototype designed to be used on top of the open-source application for digital painting Krita [9]. In this section we address some implementation details, the devices used in our setup, and how GazeBar can be adapted as an overlay to any other application intensive on mode switching. The code for this proof of concept (PoC) is available at `https://github.com/elmadjian/GazeBar`.

The PoC was designed to run on laptops and desktop PCs. We used the Tobii 4C eye tracker, which operates at 90 Hz, providing a constant stream of eye-gaze points for a single 24" monitor at 75 Hz (see Figure 4). Data is collected through a mixed C#/Python application in the backend, which is also responsible for managing the communication between GazeBar and Krita.

The GazeBar interface was completely written in QML. The interface controls the actual user interaction with mode buttons, secondary bars, and eventual switches. Changes in the bar state that affects Krita are signaled to the backend module using the PySide library for Qt 5. Mode switches are mapped to Krita through hotkeys, which means that some fine-grained input changes, such as a color picking, are simply not possible with GazeBar's PoC. However, Krita allows us to create custom hotkeys for any discrete mode available in the GUI, thus enabling GazeBar to capture most of the mode-switching workflow. Expert users working with digital painting or image editing often make simultaneous use of keyboard and graphics tablets. Non-experts, however, have a very limited knowledge of keyboard hotkeys or key + pen combos for mode switching. These users rely on mode options available on the GUI and, therefore, can benefit the most from our PoC.

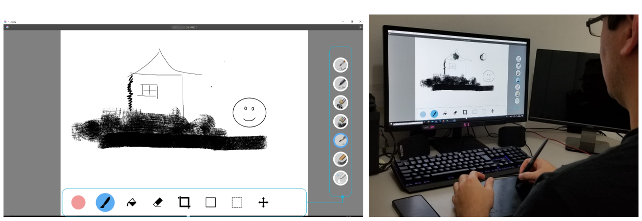

Figure 4: On the left, a screenshot of our prototype, while on the right we demonstrate its use case with a digital pen and a graphics tablet.

While using our PoC, the user does not need to lift up the pen and move it to change modes, thus saving manual work. Let us say that the user wants to select a different brush. The brush options can be found navigating through the hierarchical options tree. Remember that there is no timeout or specific gestures for a selection. Upon fixating over a bar, the user can explore the brush options freely. The last one gazed before moving the eyes back to the painting canvas will indicate to the system that an input change was made. If a wrong brush is selected, recovering from the error is as simple as gazing the correct brush and focusing back on the painting.

Since GazeBar does not expect a hand-gaze coordination event, our PoC can be easily adapted to any other application where mode switching is frequent. The only requirement is that the application provides the possibility of changing modes via shortcut keys. The expected adaptation effort would be in terms of changing mode icons, defining a new mode hierarchy and mapping new shortcuts.

5 USER EXPERIENCE AND PERFORMANCE ISSUES

The literature has mixed results regarding the efficiency of gaze input in contrast with traditional manual input. Some studies revealed that gaze can be significantly faster than mouse input for target acquisition [35, 41], though there are also some diverging results [21]. As Schuetz et al. pointed out, most of these comparative works are backed up by Fitts' Law, which may show some conflicting results depending on how gaze interaction is assessed [34].

Saccades are ballistic movements (up to 900^o/s) [2], and therefore cannot benefit from online and controlled adjustments during its course. Thus, an efficient use of gaze primarily depends on the size of a saccade with respect to a stimulus and the target size [34]. Additionally, eye tracker fidelity has been found to significantly affect this task [3], which brings high variability to the constants in Fitt's Law formula.

To claim a theoretical efficiency of our design over manual mode-switching in GUIs, we assume the following conditions: the user is seated at 50 cm from a 24" display, our eye tracker accuracy lies between 0.5-1^o, and targets have an approximate size of 2^o. Assuming also that the user's primary task is located at the center of the screen, this results in an average of 27^o of horizontal and 16^o of vertical saccadic span to reach one of the bars.

Based on a quadratic approximation of an empirical model [1], this saccade length demands at most 140 ms. Since overshoots are very likely in this span, two extra short corrective saccades of 30 ms are usually necessary for target acquisition [34]. With a variable fixation time on the target of at most 200 ms, a non-optimistic time estimate for mode-switching with GazeBar would be 500 ms. Note that a fixation dwell is assumed only as a consequence of human cognitive processing [13], but it is not required by our technique.

Manual input has been demonstrated to take roughly double the time to close the distance to targets, albeit being much more precise than saccades [41]. This suggests that using the GUI for switching modes using a mouse, for example, would require 600 ms just for the complete manual movement, not counting the time for selection and visualization feedback.

A more fundamental argument in favor of our technique is to realize that GazeBar leverages the natural scanpaths made when switching modes manually. Thus, manual response is bounded by the user visual perceptive task and should improve flow. Unless the flow is broken by unintended selections.

5.1 Comparative design analysis

To avoid unintended selections, GazeBar graphical design has considered the low accuracy of gaze trackers, eye jittery, and other factors as described in Section 3.1. Yet, if unintended selections occur often, maybe due to lack of experience of a novice user, user experience will be damaged. Therefore, GazeBar is probably not appropriate for tasks such as typing, but can benefit tasks that require not so frequent selections, and where involuntary selections have a low interaction cost since GazeBar allows very fast recovery.

The GazeBar selection mechanism is designed to be experienced as a “Midas touch”, but activation is triggered in practice in a similar fashion to context-switching [23] or reverse-crossing [8]. These gaze-only techniques, as well as other methods such as dwell-time selection [20], eye gestures [22], or motion correlation [40], create additional preventive steps to avoid the MTP, while GazeBar takes advantage of the expected gaze path to eliminate this need.

Though GazeBar might resemble other multimodal gaze-based interaction techniques, such as gaze-touch [29] and gaze-shifting [30], it is important to notice that gaze and manual inputs are choreographed in their case, while gaze is independent from manual control with GazeBar. We compare our own experience of using GazeBar to driving an automatic car: while other methods requires a "clutch" to change gears, GazeBar improves flow by allowing the user to just look at the desired option.

For traversing hierarchical menus, the design of pEYEs [39] shares many similarities to GazeBar as well. With pEYEs, however, selections are performed by navigating through expanding sub-menus, until only one option is available. For m options per menu and N items total, this leads to a minimum time complexity of $\log_m N$ for each selection. Options search in GazeBar is also bounded by $\log_m N$, but on average faster because, besides the last branch of the tree being always visible, the user does not necessarily have to reach a leaf to make a selection.

6 LESSONS FROM THE GAZEBAR PROTOTYPE

In terms of mode-switching efficiency, we showed that GazeBar's selection technique can be interpreted as an approximate theoretical upper bound for manual input performance, since manual input depends on the visual perceptive channel to coordinate selections. By modeling GazeBar's selection on mode-switching scanpaths, we discard the preventive measures found in gaze-based techniques, effectively embracing a Midas touch-like selection mechanism that does necessarily imposes involuntary mistakes.

Our prototype mitigates involuntary mode switches by design, such as the one-dimensional bar format, its positioning on screen, and the two-criteria threshold for activation. We are aware though that an empirical study still remains necessary to verify the effect of these choices and establish an average expected number of wrong selections. Another sensitive aspect that we shall address in a future experiment is the impact of GazeBar on user experience. Some objective aspects, such as saving user's manual effort compared to other methods, are easy to acknowledge, but others, such as a measurable state of flow, can only be assessed in a user study.

Compared to other gaze-only techniques, our prototype shows that GazeBar has a low error recovery cost. That is because a GazeBar selection is fast and requires minimal eye movement. Dwell-time selection, for instance, not only requires the same amount of movement (in the application workflow), but also additional fixation times. Some users might prefer more low-cost mistakes than costly errors that impose a greater penalty on user performance.

The GazeBar design, though, is not as multipurpose as other methods. However, it is fairly safe to say that complex and multimodal scenarios are more likely to be benefited by this technique, since the idea of GazeBar could be easily adapted to other intensive manual application, such as word processing, video editing, or 3D editing. And the principle of modeling gaze-based selections on expected eye movements to maximize the state of flow during interaction could be further applied to tasks other than mode-switching.

7 CONCLUSION

Is the Midas touch always a problem for gaze interaction? While many previous techniques have exploited mechanisms such as dwell time or eye gestures, and combined gaze with other mechanism such as speech or finger touch to trigger selections, we propose a just-look-to-select mechanism that can improve gaze interaction efficiency.

We argue that interaction flow can be improved by eliminating safety interaction steps (such as dwell-time or mouse click) suggested in the literature to avoid the Midas' touch problem. Of course this improvement can only be achieved while keeping the number of errors small or compatible with other techniques. To demonstrate this concept we have developed GazeBar to enhance a digital painting application, that seamlessly integrates a just-look-to-select mechanism with the primary manual task for switching between application modes. This is still a work in progress though. Future work will investigate user experience aspects of GazeBar and evaluate its performance against similar techniques such as gaze-touch and gaze-shifting.

ACKNOWLEDGMENTS

This work was supported by the São Paulo Research Foundation, grants 2015/26802-1 and 2016/10148-3.

REFERENCES

[1] Richard A Abrams, David E Meyer, and Sylvan Kornblum. 1989. Speed and accuracy of saccadic eye movements: characteristics of impulse variability in the oculomotor system. *Journal of Experimental Psychology: Human Perception and Performance* 15, 3 (1989), 529.

[2] A. Terry Bahill, Michael R. Clark, and Lawrence Stark. 1975. The main sequence, a tool for studying human eye movements. *Mathematical biosciences* 24, 3-4 (1975), 191–204.

[3] Michael Barz, Florian Daiber, Daniel Sonntag, and Andreas Bulling. 2018. Error-aware gaze-based interfaces for robust mobile gaze interaction. In *Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, ETRA 2018, Warsaw, Poland, June 14-17, 2018*, Bonita Sharif and Krzysztof Krejtz (Eds.). ACM, 24:1–24:10. https://doi.org/10.1145/3204493.3204536

[4] Chris Creed, Maite Frutos Pascual, and Ian Williams. 2020. Multimodal Gaze Interaction for Creative Design. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–13. https://doi.org/10.1145/3313831.3376196

[5] Antonio Diaz-Tula and Carlos H. Morimoto. 2016. AugKey: Increasing Foveal Throughput in Eye Typing with Augmented Keys. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 3533–3544. https://doi.org/10.1145/2858036.2858517

[6] Heiko Drewes and Albrecht Schmidt. 2009. The MAGIC Touch: Combining MAGIC-Pointing with a Touch-Sensitive Mouse. 415–428. https://doi.org/10.1007/978-3-642-03658-3_46

[7] Augusto Esteves, Eduardo Velloso, Andreas Bulling, and Hans Gellersen. 2015. Orbits: enabling gaze interaction in smart watches using moving targets. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC Adjunct 2015, Osaka, Japan, September 7-11, 2015*, Kenji Mase, Marc Langheinrich, Daniel Gatica-Perez, Hans Gellersen, Tanzeem Choudhury, and Koji Yatani (Eds.). ACM, 419–422. https://doi.org/10.1145/2800835.2800942

[8] Wenxin Feng, Ming Chen, and Margrit Betke. 2014. Target reverse crossing: a selection method for camera-based mouse-replacement systems. In *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, PETRA 2014, Island of Rhodes, Greece, May 27 - 30, 2014*, Fillia Makedon, Mark Clements, Catherine Pelachaud, Vana Kalogeraki, and Ilias Maglogiannis (Eds.). ACM, 39:1–39:4. https://doi.org/10.1145/2674396.2674443

[9] Krita Foundation. 2020. http://krita.org/

[10] John Paulin Hansen, Vijay Rajanna, I. Scott MacKenzie, and Per Bækgaard. 2018. A Fitts' law study of click and dwell interaction by gaze, head and mouse with a head-mounted display. In *Proceedings of the Workshop on Communication by Gaze Interaction, COGAIN@ETRA 2018, Warsaw, Poland, June 15, 2018*, Carlos Morimoto and Thies Pfeiffer (Eds.). ACM, 7:1–7:5. https://doi.org/10.1145/3206343.3206344

[11] Anke Huckauf and Mario Urbina. 2007. Gazing with pEYE: New Concepts in Eye Typing. In *Proceedings of the 4th Symposium on Applied Perception in Graphics and Visualization* (Tubingen, Germany) *(APGV '07)*. ACM, New York, NY, USA, 141–141. https://doi.org/10.1145/1272582.1272618

[12] Robert J. K. Jacob. 1991. The Use of Eye Movements in Human-Computer Interaction Techniques: What You Look At is What You Get. *ACM Trans. Inf. Syst.* 9, 2 (1991), 152–169. https://doi.org/10.1145/123078.128728

[13] Marcel Adam Just and Patricia A Carpenter. 1976. The role of eye-fixation research in cognitive psychology. *Behavior Research Methods & Instrumentation* 8, 2 (1976), 139–143.

[14] Mohamed Khamis, Ozan Saltuk, Alina Hang, Katharina Stolz, Andreas Bulling, and Florian Alt. 2016. TextPursuits: Using Text for Pursuits-based Interaction and Calibration on Public Displays. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 274–285. https://doi.org/10.1145/2971648.2971679

[15] Per Ola Kristensson and Keith Vertanen. 2012. The Potential of Dwell-free Eye-typing for Fast Assistive Gaze Communication. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. ACM, New York, NY, USA, 241–244. https://doi.org/10.1145/2168556.2168605

[16] Chandan Kumar, Daniyal Akbari, Raphael Menges, I. Scott MacKenzie, and Steffen Staab. 2019. TouchGazePath: Multimodal Interaction with Touch and Gaze Path for Secure Yet Efficient PIN Entry. In *International Conference on Multimodal Interaction, ICMI 2019, Suzhou, China, October 14-18, 2019*, Wen Gao, Helen Mei-Ling Meng, Matthew A. Turk, Susan R. Fussell, Björn W. Schuller, Yale Song, and Kai Yu (Eds.). ACM, 329–338. https://doi.org/10.1145/3340555.3353734

[17] Chandan Kumar, Ramin Hedeshy, I. Scott MacKenzie, and Steffen Staab. 2020. TAGSwipe: Touch Assisted Gaze Swipe for Text Entry. In *CHI '20: CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, April 25-30, 2020*, Regina Bernhaupt, Florian 'Floyd' Mueller, David Verweij, Josh Andres, Joanna McGrenere, Andy Cockburn, Ignacio Avellino, Alix Goguey, Pernille Bjøn, Shengdong Zhao, Briane Paul Samson, and Rafal Kocielnik (Eds.). ACM, 1–12. https://doi.org/10.1145/3313831.3376317

[18] Andrew T. N. Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos H. Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free Text Entry Using Gaze Paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (Santa Clara, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 1952–1956. https://doi.org/10.1145/2858036.2858335

[19] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast Gaze Typing with an Adjustable Dwell Time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 357–360. https://doi.org/10.1145/1518701.1518758

[20] Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty Years of Eye Typing: Systems and Design Issues. In *Proceedings of the 2002 Symposium on Eye Tracking Research & Applications* (New Orleans, Louisiana) *(ETRA '02)*. ACM, New York, NY, USA, 15–22. https://doi.org/10.1145/507072.507076

[21] Darius Miniotas. 2000. Application of Fitts' law to eye gaze interaction. In *CHI '00 Extended Abstracts on Human Factors in Computing Systems, CHI Extended Abstracts '00, The Hague, The Netherlands, April 1-6, 2000*, Marilyn Tremaine (Ed.). ACM, 339–340. https://doi.org/10.1145/633292.633496

[22] Emilie Møllenbach, Martin Lillholm, Alastair G. Gail, and John Paulin Hansen. 2010. Single gaze gestures. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari, and Qiang Ji (Eds.). ACM, 177–180. https://doi.org/10.1145/1743666.1743710

[23] Carlos Hitoshi Morimoto and Arnon Amir. 2010. Context switching for fast key selection in text entry applications. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari, and Qiang Ji (Eds.). ACM, 271–274. https://doi.org/10.1145/1743666.1743730

[24] Carlos H. Morimoto and Marcio R.M. Mimica. 2005. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding* 98, 1 (2005), 4 – 24. https://doi.org/10.1016/j.cviu.2004.07.010 Special Issue on Eye Detection and Tracking.

[25] Martez E. Mott, Shane Williams, Jacob O. Wobbrock, and Meredith Ringel Morris. 2017. Improving Dwell-Based Gaze Typing with Dynamic, Cascading Dwell Times. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 2558–2570. https://doi.org/10.1145/3025453.3025517

[26] Emilie Møllenbach, John Paulin Hansen, and Martin Lillholm. 2013. Eye Movements in Gaze Interaction. *Journal of Eye Movement Research* 6, 2 (2013). https://bop.unibe.ch/index.php/JEMR/article/view/2354

[27] J. Nakamura and M. Csikszentmihalyi. 2014. *Flow and the Foundations of Positive Psychology*. Springer, Chapter The Concept of Flow. https://doi.org/10.1007/978-94-017-9088-8_16

[28] Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N. Truong. 2015. Filteryedping: Design Challenges and User Performance of Dwell-Free Eye Typing. *ACM Trans. Access. Comput.* 6, 1, Article 3 (March 2015), 37 pages. https://doi.org/10.1145/2724728

[29] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, and Hans Gellersen. 2014. Gaze-touch: combining gaze with multi-touch for interaction on the same surface. In *The 27th Annual ACM Symposium on User Interface Software and Technology, UIST '14, Honolulu, HI, USA, October 5-8, 2014*, Hrvoje Benko, Mira Dontcheva, and Daniel Wigdor (Eds.). ACM, 509–518. https://doi.org/10.1145/2642918.2647397

[30] Ken Pfeuffer, Jason Alexander, Ming Ki Chong, Yanxia Zhang, and Hans Gellersen. 2015. Gaze-Shifting: Direct-Indirect Input with Pen and Touch Modulated by Gaze. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology* (Charlotte, NC, USA) *(UIST '15)*. Association for Computing Machinery, New York, NY, USA, 373–383. https://doi.org/10.1145/2807442.2807460

[31] Ken Pfeuffer, Jason Alexander, and Hans Gellersen. 2016. Partially-indirect Bimanual Input with Gaze, Pen, and Touch for Pan, Zoom, and Ink Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, May 7-12, 2016*, Jofish Kaye, Allison Druin, Cliff Lampe, Dan Morris, and Juan Pablo Hourcade (Eds.). ACM, 2845–2856. https://doi.org/10.1145/2858036.2858201

[32] Ken Pfeuffer and Hans Gellersen. 2016. Gaze and Touch Interaction on Tablets. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology, UIST 2016, Tokyo, Japan, October 16-19, 2016*, Jun Rekimoto, Takeo Igarashi, Jacob O. Wobbrock, and Daniel Avrahami (Eds.). ACM, 301–311. https://doi.org/10.1145/2984511.2984514

[33] Sheikh Rivu, Yasmeen Abdrabou, Thomas Mayer, Ken Pfeuffer, and Florian Alt. 2019. GazeButton: enhancing buttons with eye gaze interactions. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, ETRA 2019, Denver , CO, USA, June 25-28, 2019*, Krzysztof Krejtz and Bonita Sharif (Eds.). ACM, 73:1–73:7. https://doi.org/10.1145/3317956.3318154

[34] Immo Schuetz, T. Scott Murdison, Kevin J. MacKenzie, and Marina Zannoli. 2019. An Explanation of Fitts' Law-like Performance in Gaze-Based Selection Tasks Using a Psychophysics Approach. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, CHI 2019, Glasgow, Scotland, UK, May 04-09, 2019*, Stephen A. Brewster, Geraldine Fitzpatrick, Anna L. Cox, and Vassilis Kostakos (Eds.). ACM, 535. https://doi.org/10.1145/3290605.3300765

[35] Linda E. Sibert and Robert J. K. Jacob. 2000. Evaluation of eye gaze interaction. In *Proceedings of the CHI 2000 Conference on Human factors in computing systems, The Hague, The Netherlands, April 1-6, 2000*, Thea Turner and Gerd Szwillus (Eds.). ACM, 281–288. https://doi.org/10.1145/332040.332445

[36] Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *CHI Conference on Human Factors in Computing Systems, CHI '12, Austin, TX, USA - May 05 - 10, 2012*, Joseph A. Konstan, Ed H. Chi, and Kristina Höök (Eds.). ACM, 2981–2990. https://doi.org/10.1145/2207676.2208709

[37] Antonio Diaz Tula, Filipe Morgado Simoes de Campos, and Carlos H. Morimoto. 2012. Dynamic context switching for gaze based interaction. In *Proceedings of the 2012 Symposium on Eye-Tracking Research and Applications, ETRA 2012, Santa Barbara, CA, USA, March 28-30, 2012*, Carlos Hitoshi Morimoto, Howell O. Istance, Stephen N. Spencer, Jeffrey B. Mulligan, and Pernilla Qvarfordt (Eds.). ACM, 353–356. https://doi.org/10.1145/2168556.2168635

[38] Jayson Turner, Eduardo Velloso, Hans Gellersen, and Veronica Sundstedt. 2014. EyePlay: Applications for Gaze in Games. In *Proceedings of the First ACM SIGCHI Annual Symposium on Computer-Human Interaction in Play* (Toronto, Ontario, Canada) *(CHI PLAY '14)*. Association for Computing Machinery, New York, NY, USA, 465–468. https://doi.org/10.1145/2658537.2659016

[39] Mario H. Urbina and Anke Huckauf. 2010. Alternatives to single character entry and dwell time selection on eye typing. In *Proceedings of the 2010 Symposium on Eye-Tracking Research & Applications, ETRA 2010, Austin, Texas, USA, March 22-24, 2010*, Carlos Hitoshi Morimoto, Howell O. Istance, Aulikki Hyrskykari, and Qiang Ji (Eds.). ACM, 315–322. https://doi.org/10.1145/1743666.1743738

[40] Eduardo Velloso, Marcus Carter, Joshua Newn, Augusto Esteves, Christopher Clarke, and Hans Gellersen. 2017. Motion Correlation: Selecting Objects by Matching Their Movement. *ACM Trans. Comput. Hum. Interact.* 24, 3 (2017), 22:1–22:35. https://doi.org/10.1145/3064937

[41] Roel Vertegaal. 2008. A Fitts Law comparison of eye tracking and manual input in the selection of visual targets. In *Proceedings of the 10th International Conference on Multimodal Interfaces, ICMI 2008, Chania, Crete, Greece, October 20-22, 2008*, Vassilios Digalakis, Alexandros Potamianos, Matthew A. Turk, Roberto Pieraccini, and Yuri Ivanov (Eds.). ACM, 241–248. https://doi.org/10.1145/1452392.1452443

[42] 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* (Zurich, Switzerland) *(UbiComp '13)*. ACM, New York, NY, USA, 439–448. https://doi.org/10.1145/2493432.2493477

[43] Oleg Špakov and Darius Miniotas. 2004. On-line Adjustment of Dwell Time for Target Selection by Gaze. In *Proceedings of the Third Nordic Conference on Human-computer Interaction* (Tampere, Finland) *(NordiCHI '04)*. ACM, New York, NY, USA, 203–206. https://doi.org/10.1145/1028014.1028045

[44] David J. Ward and David J. C. MacKay. 2002. Fast Hands-free writing by Gaze Direction. *Nature* 418, 6900 (2002), 838. http://www.inference.phy.cam.ac.uk/mackay/abstracts/eyeshortpaper.html

[45] Thorsten O. Zander, Matti Gaertner, Christian Kothe, and Roman Vilimek. 2011. Combining Eye Gaze Input With a Brain-Computer Interface for Touchless Human-Computer Interaction. *Int. J. Hum. Comput. Interact.* 27, 1 (2011), 38–51. https://doi.org/10.1080/10447318.2011.535752

[46] Shumin Zhai, Carlos Morimoto, and Steven Ihde. 1999. Manual and Gaze Input Cascaded (MAGIC) Pointing. In *Proceeding of the CHI '99 Conference on Human Factors in Computing Systems: The CHI is the Limit, Pittsburgh, PA, USA, May 15-20, 1999*, Marian G. Williams and Mark W. Altom (Eds.). ACM, 246–253. https://doi.org/10.1145/302979.303053