Tension&Gaze: Gaze-Responsive UI Gated by Finger Tension

Taejun Kim
HCI Lab, KAIST
Daejeon, Republic of Korea
taejun.kim@kaist.ac.kr

Ludwig Sidenmark
University of Toronto
Toronto, Ontario, Canada
lsidenmark@dgp.toronto.edu

Parastoo Abtahi
Princeton University
Princeton, New Jersey, USA
parastoo@princeton.edu

Jisu Yim
HCI Lab, KAIST
Daejeon, Republic of Korea
yimjisu99@kaist.ac.kr

YoungIn Kim
HCI Lab, KAIST
Daejeon, Republic of Korea
youngin@kaist.ac.kr

Geehyuk Lee
HCI Lab, KAIST
Daejeon, Republic of Korea
geehyuk@kaist.ac.kr

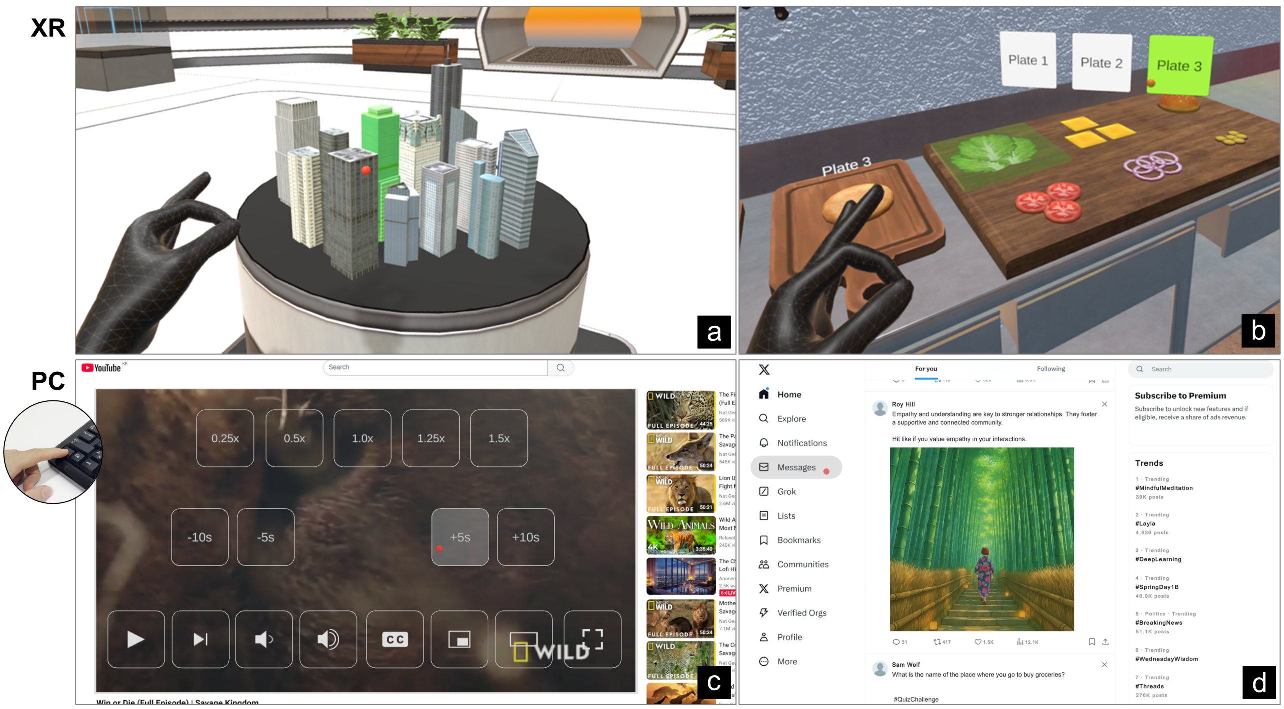

Figure 1: Demonstrating interaction scenarios of *Tension&Gaze.* (a) XR museum: a pinch-and-hold activates gaze-responsiveness on miniaturized building models, with its release triggering selection. Users can also perform occluded target selection with hand-based adjustment while pinching. (b) XR Cooking Room: a pinch-and-hold activates gaze-responsiveness of various targets for preparing a burger order. (c, d) PC-based Youtube and Social Media: holding a key activates gaze-responsive UIs, and releasing it triggers selection on the gazed target. A red dot, invisible to users, indicates the current gaze position.

Abstract

Gaze-interactive applications involve UI responses triggered by users' gaze, which is often involuntary and reflexive. While it may not cause false input 'registration' in eye+hand contexts, it can still produce unintended chains of visual UI responses. Although designers strive to minimize potential distraction and visual irritation, gaze input inherently carries noise; our eyes are *always-on*, often unconsciously scanning the visual field as active sensors, making this a persistent and challenging issue.

We argue that *always gaze-responsive* applications may not be favored by consumers, particularly when they risk causing visual discomfort. As a simple yet potentially effective alternative, this work introduces *Tension&Gaze* interaction techniques, in which UI

UIST Adjunct '25, Busan, Republic of Korea

ACM ISBN 979-8-4007-2036-9/25/09
https://doi.org/10.1145/3746058.3759018

elements become gaze-responsive only during moments of finger-induced tension, e.g., pinching in XR or key pressing on PC. This allows users to engage with gaze-adaptive UI changes only when desired, maintaining a comfortable default viewing experience. We demonstrate four application scenarios: XR Museum and Cooking Room using Meta Quest Pro, and PC Youtube and Social Media using a Tobii eye tracker-equipped laptop.

ACM Reference Format:
Taejun Kim, Ludwig Sidenmark, Parastoo Abtahi, Jisu Yim, YoungIn Kim, and Geehyuk Lee. 2025. Tension&Gaze: Gaze-Responsive UI Gated by Finger Tension. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST Adjunct '25), September 28–October 01, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3746058.3759018

1 Introduction

Integration of eye tracking into digital environments, including desktops and laptops [11, 13, 20], smartphones [1, 7, 36], tablets [6, 12], and Extended Reality (XR) headsets [8–10], has gained growing interests in gaze-interactive applications. Meanwhile, gaze-based input inherently involves UI response aids, primarily due to the limited eye tracking accuracy [4]. For example, basic *gaze-hover* feedback of color [32], opacity [8], or outline [28] changes are vital for confident pointing. A more intrusive form involves displaying a gaze cursor, allowing users to perceive tracking offsets in real time [8, 16, 33]. More dynamic UI adaptations, such as adjusting the spacing of elements or enlarging targets [22, 26], also help accommodate limited tracking resolution.

However, eye movements are largely involuntary and reflexive, often unconsciously scanning the visual field as active receptors of information [24]. Suppressing this natural motor behavior is known to impose significant cognitive strain [15]. While such eye movements may not directly cause false input registration in gaze+hand contexts, it can still trigger unintended chains of visual UI responses. Despite designers' efforts to minimize potential distraction and visual irritation, this inherent noise remains a persistent and challenging issue in the design of gaze-interactive applications.

We argue that *always gaze-responsive* applications may not be favored for consumer adoption, especially when they risk causing visual irritation. As a simple yet potentially effective alternative, we suggest the *Tension&Gaze* technique (Figure 2), in which UI elements remain unresponsive to gaze by default and become gaze-responsive only during a user-maintained period of finger-induced tension. This design approach gates *gaze-interactivity* through a tension medium (e.g., holding finger pinch in XR for 0.4 seconds), which is carefully crafted to avoid interfering with existing interaction vocabularies.

2 Tension&Gaze Idea

While the *quasimode*, a mode activated and sustained through kinesthetic tension [2, 30, 31], has been combine with gaze-input techniques, most prior work has not addressed the unintended UI response problem we target [5, 14, 21, 29, 34]. For example, techniques of Tagswipe [21], Hummer [5], Tobii Window Interaction Feature [14], Gaze+Hold [29], and Gaze & Touch [34] all continued to rely on gaze-responsive visual aids, such as gaze hovers or gaze cursor displays, throughout the interaction phase, i.e., even before a user initiates the tension.

To our knowledge, only a few documented instances align with the *Tension&Gaze* concept as we define it: the Look-Press-Look-Release technique in [23], and the Offset Menu and Ray Selection techniques in [37], all developed for desktop environments. Building on these foundations, we extend the idea to 3D XR environments, which feature a significantly larger visual field and utilize a different form of tension (i.e., finger pinching in freehand settings and button pressing in controller-based settings). We also broaden the scope by incorporating further dynamic UI feedback and adaptation scenarios, such as gaze-triggered dynamic spacing adjustment [26], document scrolling [35], information unfolding [27], and XR occluded object selection [3].

3 New Interaction Benefits

3.1 XR: Gated with Pinch Hold

Most existing XR systems rely predominantly on manual input methods like controllers and freehand gestures [17, 18]. Even in newer platforms like the Apple Vision Pro [8], which features gaze+pinch as one of its default input mechanism, *gaze interactivity* seems to remain largely limited, possibly due to the aforementioned issue of false visual responses and resulting discomfort.

Adopting a *Tension&Gaze* technique, interfaces could support unrestricted gaze-responsiveness without this concern. Users can perform gaze-based selection of any XR objects with dynamically adapting UIs when needed, i.e., beyond basic minimal gaze-hovers. In addition to facilitating object selection, new interactive opportunities like glanceable information unfolding [25] or gaze-driven menus [16] can be provided exclusively during the tension period. This is particularly beneficial in contexts like museums, where both uninterrupted viewing and interactivity are important.

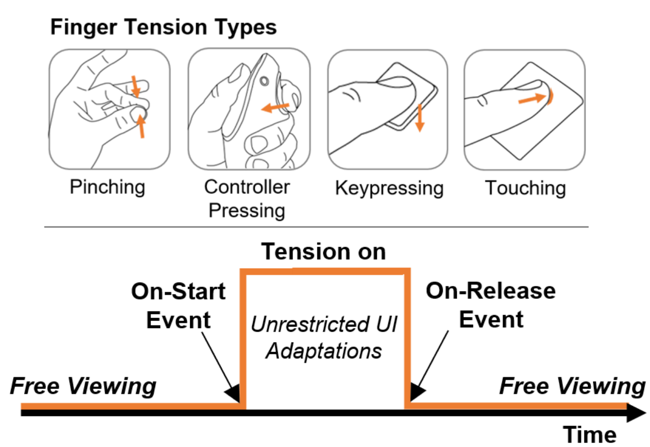

Figure 2: A conceptual illustration of the *Tension&Gaze* workflow. *On-Start Event* is a one-time event coupled with the tension onset, distinct from gaze-responses during the tension period. *On-Release Event* is another one-time event that occurs upon the release of tension; e.g., the triggering of selection on a gazed-at target.

3.2 PC: Gated with Press Hold

Contrasting XR, PC systems rely on a mouse or touchpad for pointing. In this context, *Tension&Gaze* can offer a faster and more effortless alternative. During a keypress hold, the screen UI can also dynamically adjust element spacing or enlarge hitboxes to account for eye tracking accuracy. This enables users to easily make selections with a simple press-look-release gesture (possibly forming a glued "chunk" [2]). This approach may reduce the need for continuous reliance on physical pointing devices while preserving a comfortable, non-responsive viewing state by default.

4 Demo Walkthrough

4.1 XR Demo Using Meta Quest Pro

4.1.1 XR Museum: free viewing, normal gaze+pinch object selection, occluded object selection. A user at metropolitan architecture museum are freely observing miniature building models. For more detailed presentation, the user makes a finger pinch for longer than 0.4 seconds. Holding the pinch and looking at the target building, which might be obscured by others, adjacent buildings respond together with gaze-hover feedback. Moving their wrist while maintaining the pinch activates fine-adjustments. The user then releases the pinch when the desired one is highlighted, getting detailed information including the building's established year, name, height, and more with an enlarged view.

4.1.2 XR Cooking Room: gaze+pinch object selection, menu selection, and gaze-triggered information unfolding. A user is in a burger restaurant kitchen and will prepare an order. Holding the pinch and looking at the wanted vegetable on the worktop triggers a gaze-hover response (e.g. the lettuce in Figure 1b). Releasing the pinch selects the vegetable and reveals buttons for plate placement. The user then pinches again, looks at a button, and release to place the lettuce on a plate. On the other side, patties and bacon are cooking on the stove. To check the remaining cook time before flipping, the user holds a pinch while gazing at the stove area, prompting time labels to appear (e.g., "0:23"). To flip a patty, the user gazes at the spatula hanging at the front while holding a pinch again.

4.1.3 PC Youtube & Social Media: gaze+keypress dynamic spacing adjustment, scrolling, button selection, menu triggering and selection. A user can control the Youtube video player with keypress-and-gaze, such as play/pause, rewind, change speed, adjust volume, or enable captions. Holding the "*gaze key*" on the keyboard and looking at the video player area, a gaze menu with large buttons appear, leveraging the tactile signal of clear state feedback [19]. Releasing the key selects the gazed button on screen and trigger the command. For close screen buttons challenging the gaze tracking offset, e.g., "like", "dislike", and "share" below the player, they dynamically adjusts the spacing upon gazed at, making gaze-based selection easier. To scroll, the user looks above or below the screen while holding the gaze key. The PC Social Media (Figure 1d) offers a similar keypress-and-gaze experience, allowing users to browse a social media without the need to constantly switch to a mouse or touchpad.

Acknowledgments

This work was supported by the IITP(Institute of Information & Communications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2024-00436398).

References

[1] Riku Arakawa, Mayank Goel, Chris Harrison, and Karan Ahuja. 2022. Rgbdgaze: Gaze tracking on smartphones with RGB and depth data. In *Proceedings of the 2022 International Conference on Multimodal Interaction*. 329–336. https://doi.org/10.1145/3536221.3556568

[2] William Buxton. 1995. Chunking and phrasing and the design of human-computer dialogues. In *Readings in human–computer interaction*. Elsevier, 494–499. https://dl.acm.org/doi/10.5555/212925.212970

[3] Di Laura Chen, Marcello Giordano, Hrvoje Benko, Tovi Grossman, and Stephanie Santosa. 2023. Gazeraycursor: Facilitating virtual reality target selection by blending gaze and controller raycasting. In *Proceedings of the 29th ACM Symposium on Virtual Reality Software and Technology*. 1–11.

[4] Anna Maria Feit, Shane Williams, Arturo Toledo, Ann Paradiso, Harish Kulkarni, Shaun Kane, and Meredith Ringel Morris. 2017. Toward everyday gaze input: Accuracy and precision of eye tracking and implications for design. In *Proceedings of the 2017 Chi conference on human factors in computing systems*. 1118–1130. https://doi.org/10.1145/3025453.3025599

[5] Ramin Hedeshy, Chandan Kumar, Raphael Menges, and Steffen Staab. 2021. Hummer: Text entry by gaze and hum. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3411764.3445501

[6] Jiarui Hou, James F Lee, and Stephen Doherty. 2025. State-of-the-Art of Eye Tracking in Mobile-Assisted Learning Studies: A Review of Twenty Years of Empirical Studies. *Journal of Computer Assisted Learning* 41, 1 (2025), e13118.

[7] Apple inc. 2024. Apple announces new accessibility feature including eye tracking. https://www.apple.com/newsroom/2024/05/apple-announces-new-accessibility-features-including-eye-tracking/, last visited Jul. 2025.

[8] Apple inc. 2024. Apple Vision Pro. https://www.apple.com/apple-vision-pro/, last visited Sep. 2024.

[9] Meta inc. 2024. Meta Quest Pro. https://www.meta.com/kr/en/quest/quest-pro/, last visited Sep. 2024.

[10] Magic Leap inc. 2024. Magic Leadp AR glasses. https://www.magicleap.io/magic-leap-2, last visited Sep. 2024.

[11] Tobii inc. 2024. Alienware, eye tracking laptop. https://gaming.tobii.com/product/alienware-17/, last visited Sep. 2024.

[12] Tobii inc. 2024. Tobii Dynavox, assistive technology for communication. https://us.tobiidynavox.com/, last visited Feb. 2024.

[13] Tobii inc. 2024. Tobii Eye Tracker 5. https://gaming.tobii.com/product/eye-tracker-5/, last visited Sep. 2024.

[14] Tobii inc. 2024. Windows Interaction Features, 'Mouse warp on key" feature with the "perform a click when releasing key" option. https://help.tobii.com/hc/en-us/articles/209525449-Windows-Interaction-features, last visited Sep. 2024.

[15] Robert JK 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*. 11–18. https://doi.org/10.1145/97243.97246

[16] Taejun Kim, Auejin Ham, Sunggeun Ahn, and Geehyuk Lee. 2022. Lattice menu: A low-error gaze-based marking menu utilizing target-assisted gaze gestures on a lattice of visual anchors. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3491102.3501977

[17] Taejun Kim, Amy Karlson, Aakar Gupta, Tovi Grossman, Jason Wu, Parastoo Abtahi, Christopher Collins, Michael Glueck, and Hemant Bhaskar Surale. 2023. Star: Smartphone-analogous typing in augmented reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–13. http://doi.org/10.1145/3586183.3606803

[18] Taejun Kim, Youngbo Aram Shim, Youngin Kim, Sunbum Kim, Jaeyeon Lee, and Geehyuk Lee. 2024. QuadStretcher: a forearm-worn skin stretch display for bare-hand interaction in AR/VR. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–15. http://doi.org/10.1145/3613904.3642067

[19] Taejun Kim, Youngbo Aram Shim, and Geehyuk Lee. 2021. Heterogeneous stroke: Using unique vibration cues to improve the wrist-worn spatiotemporal tactile display. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–12. http://doi.org/10.1145/3411764.3445448

[20] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye tracking for everyone. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2176–2184.

[21] Chandan Kumar, Ramin Hedeshy, I Scott MacKenzie, and Steffen Staab. 2020. Tagswipe: Touch assisted gaze swipe for text entry. In *Proceedings of the 2020 chi conference on human factors in computing systems*. 1–12. https://doi.org/10.1145/3313831.3376317

[22] Chandan Kumar, Raphael Menges, Daniel Müller, and Steffen Staab. 2017. Chromium based framework to include gaze interaction in web browser. In *Proceedings of the 26th international conference on world wide web companion*. 219–223. https://doi.org/10.1145/3041021.3054730

[23] Manu Kumar, Andreas Paepcke, and Terry Winograd. 2007. Eyepoint: practical pointing and selection using gaze and keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 421–430. https://doi.org/10.1145/1240624.124069

[24] Simon P Liversedge. 2011. *The Oxford Handbook of Eye Movements*. Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199539789.001.0001

[25] Feiyu Lu, Shakiba Davari, and Doug Bowman. 2021. Exploration of techniques for rapid activation of glanceable information in head-worn augmented reality. In *Proceedings of the 2021 ACM Symposium on Spatial User Interaction*. 1–11.

[26] Carlos H Morimoto, Jose AT Leyva, and Antonio Diaz-Tula. 2018. Context switching eye typing using dynamic expanding targets. In *Proceedings of the Workshop on Communication by Gaze Interaction*. 1–9. https://doi.org/10.1145/3206343.3206347

[27] Robin Piening, Ken Pfeuffer, Augusto Esteves, Tim Mittermeier, Sarah Prange, Philippe Schröder, and Florian Alt. 2021. Looking for info: Evaluation of gaze based information retrieval in augmented reality. In *Human-Computer Interaction–INTERACT 2021: 18th IFIP TC 13 International Conference, Bari, Italy, August 30–September 3, 2021, Proceedings, Part I 18*. Springer, 544–565.

[28] Vijay Rajanna, Murat Russel, Jeffrey Zhao, and Tracy Hammond. 2022. PressTapFlick: Exploring a gaze and foot-based multimodal approach to gaze typing. *International Journal of Human-Computer Studies* 161 (2022), 102787. https://doi.org/10.1016/j.ijhcs.2022.102787

[29] Argenis Ramirez Ramirez Gomez, Christopher Clarke, Ludwig Sidenmark, and Hans Gellersen. 2021. Gaze+ Hold: Eyes-only direct manipulation with continuous gaze modulated by closure of one eye. In *ACM symposium on eye tracking research and applications*. 1–12. https://doi.org/10.1145/3448017.3457381

[30] Jeff Raskin. 2000. *The humane interface: New directions for designing interactive systems*. Addison-Wesley. https://dl.acm.org/doi/book/10.5555/333103

[31] Abigail J Sellen, Gordon P Kurtenbach, and William AS Buxton. 1992. The prevention of mode errors through sensory feedback. *Human-computer interaction* 7, 2 (1992), 141–164. https://doi.org/10.1207/s15327051hci0702_1

[32] Asma Shakil, Christof Lutteroth, and Gerald Weber. 2019. Codegazer: Making code navigation easy and natural with gaze input. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–12. https://doi.org/10.1145/3290605.3300306

[33] Ludwig Sidenmark, Dominic Potts, Bill Bapisch, and Hans Gellersen. 2021. Radi-eye: Hands-free radial interfaces for 3d interaction using gaze-activated head-crossing. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–11. https://doi.org/10.1145/3411764.3445697

[34] Sophie Stellmach and Raimund Dachselt. 2012. Look & touch: gaze-supported target acquisition. In *Proceedings of the SIGCHI conference on human factors in computing systems*. 2981–2990.

[35] Jayson Turner, Shamsi Iqbal, and Susan Dumais. 2015. Understanding gaze and scrolling strategies in text consumption tasks. In *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. 829–838.

[36] Nachiappan Valliappan, Na Dai, Ethan Steinberg, Junfeng He, Kantwon Rogers, Venky Ramachandran, Pingmei Xu, Mina Shojaeizadeh, Li Guo, Kai Kohlhoff, et al. 2020. Accelerating eye movement research via accurate and affordable smartphone eye tracking. *Nature communications* 11, 1 (2020), 4553.

[37] Ken Neth Yeoh, Christof Lutteroth, and Gerald Weber. 2015. Eyes and keys: An evaluation of click alternatives combining gaze and keyboard. In *Human-Computer Interaction–INTERACT 2015: 15th IFIP TC 13 International Conference, Bamberg, Germany, September 14-18, 2015, Proceedings, Part I 15*. Springer, 367–383. https://doi.org/10.1007/978-3-319-22701-6_28