A Gaze Gesture-Based Paradigm for Situational Impairments, Accessibility, and Rich Interactions

Vijay Rajanna
Department of Computer Science and Engineering
Texas A&M University, College Station, Texas
vijay.drajanna@gmail.com

Tracy Hammond
Department of Computer Science and Engineering
Texas A&M University, College Station, Texas
thammond@gmail.com

ABSTRACT

Gaze gesture-based interactions on a computer are promising, but the existing systems are limited by the number of supported gestures, recognition accuracy, need to remember the stroke order, lack of extensibility, and so on. We present a gaze gesture-based interaction framework where a user can design gestures and associate them to appropriate commands like minimize, maximize, scroll, and so on. This allows the user to interact with a wide range of applications using a common set of gestures. Furthermore, our gesture recognition algorithm is independent of the screen size, resolution, and the user can draw the gesture anywhere on the target application. Results from a user study involving seven participants showed that the system recognizes a set of nine gestures with an accuracy of 93% and a F-measure of 0.96. We envision, this framework can be leveraged in developing solutions for situational impairments, accessibility, and also for implementing rich a interaction paradigm.

CCS CONCEPTS

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

KEYWORDS

Gaze gestures; accessibility; situational impairment; eye tracking.

ACM Reference Format:
Vijay Rajanna and Tracy Hammond. 2018. A Gaze Gesture-Based Paradigm for Situational Impairments, Accessibility, and Rich Interactions. In *ETRA '18: 2018 Symposium on Eye Tracking Research and Applications, June 14–17, 2018, Warsaw, Poland*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3204493.3208344

1 INTRODUCTION

Real-time information of the eye movements can be used for direct manipulation of the interface elements; this forms the basis of gaze-assisted interaction [Majaranta 2011]. Gaze-assisted interaction is crucial in scenarios of situational impairments, i.e., the inability to work on a computer due to busy hands. Also, individuals with physical impairments or disabilities use eye movements for pointing, selecting, and typing tasks on a computer [Hansen et al. 1995, 2003]. With gaze-assisted interaction, an on-screen cursor navigation is achieved by mapping the eye movements to screen co-ordinates [Duchowski 2007]. However, simply using gaze positions for target selection leads to inadvertent activations which is also known as the "Midas Touch" issue [Jacob 1991]. Multiple solutions like fixating on the target for a predefined dwell time (150-200 milliseconds) [Jacob 1991], blinking, or using a supplemental input [Rajanna and Hammond 2016] have been proposed for target selection. However, these solutions are limited by the issues like accuracy, speed, need for an additional device, repeated calibrations, or fatigue due to prolonged usage [Majaranta 2011].

In our approach, we re-contextualize gaze input as gestural input such that each gesture represents a user command (action) that can be executed on an application. For example, in the case of a situational impairment or disability, a gaze gesture can minimize, maximize, restore, or close an active window, or it can create a new tab, scroll, refresh, etc., on an application like a browser. Also, an individual with speech impairment can use gaze gestures to make the computer speak specific phrases without switching to an assistive application. Furthermore, as an example of a rich interaction paradigm, a user can create a set of gestures to execute code, debug, format, comment, etc., that can work across different code editors. This relieves the user from remembering different shortcuts on different code editors, or a time consuming option of using the mouse to select the action from the menu. Therefore, an accurate gaze gesture recognition framework would allow for improved and extensible gaze-assisted interactions.

2 PRIOR WORK

The feasibility of gaze-assisted interaction was first demonstrated by Jacob [Jacob 1991]. Since then, gaze-assisted interactions have been used for point-and-click [Kumar et al. 2007; Lankford 2000; Rajanna and Hammond 2016], typing [Hansen et al. 2004; Rajanna 2016], authentication [Bulling et al. 2012; Rajanna et al. 2018, 2017], biometrics [Hammond et al. 2017], performing secondary actions like zooming and panning [Göbel et al. 2013], etc. Focusing specifically on some of the major research in gaze gesture-based interaction, Drewes et al. [2007], presented a framework to interact with computers using gaze gestures. The authors implemented a gaze gesture algorithm based on mouse gestures, where users move their gaze in a combination out of eight directions to draw a gesture and execute an associated action. Wobbrock et al. [2008], presented EyeWrite: a gaze typing system, where characters are entered by performing predefined gestures for each character. EyeWrite achieves an average typing speed of five words per minute, and the participants felt it was easier to use EyeWrite than on-screen keyboard. Bulling et al. [2009], presented a wearable EOG goggles using which gaze gestures can be performed, as presented by Drewes et al. [2007], to interact with computers. Similarly, the usability of gaze

ETRA '18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5706-7/18/06.
https://doi.org/10.1145/3204493.3208344

gestures is demonstrated for gaming. All the prior systems translate gaze gesture into a series of directional movements which limit the recognition accuracy because of jittery eye movements, and also, remembering a gesture as a sequence of directional changes is hard [Drewes and Schmidt 2007]. To address these limitations, our approach considers a gesture as a sketch stroke with a series of points, and the gesture is compared against a set of templates for recognition. This approach results in high accuracy, the gestures are independent of the screen size and resolution, the user is not required to remember the exact gestures, and the gestures can be executed anywhere on the screen.

3 SYSTEM ARCHITECTURE

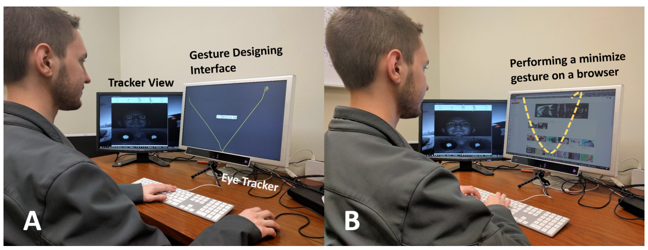

Figure 1: A) Design gestures, and B) Perform gestures

The system consists of a Gaze Tracking Module, and Gesture Recognition Engine (Figure 1.B). The gaze tracking module uses a table-mounted “Gaze Point" eye tracker that provides (X,Y) gaze coordinates at 150 Hz. The gesture recognition engine constantly receives gaze points from the eye tracker, and is responsible for recognizing the gesture performed and executing the associated action on the target application. The beginning of the gesture is indicated by either pressing a hot-key (e.g., F2), or fixating for nearly 200 ms on the top left corner of the screen. Figure 1.A shows the gesture design interface where the user creates gestures with eye movements and associates an action to each gesture. Figure 1.B shows the gaze gesture-based interaction framework in action.

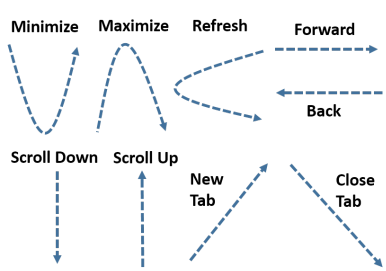

Figure 2: A set of gaze gestures designed to interact with a browser.

4 TEMPLATE MATCHING ALGORITHM

The gesture recognition engine performs template matching to match the gesture performed by the user (candidate gesture) to one of the various template gestures [Wobbrock et al. 2007]. This is a multistage process (Figure 3), where the candidate gesture (3.A) is first sampled to N = 220 points (3.B). After sampling, the centroid of the gesture is calculated, and the centroid is moved to (0,0) co-ordinate, and also, all other points are moved to new points relative to the centroid (3.C). Finally, the transformed candidate gesture is compared with a set of template gestures by computing the Euclidean distance (3.D) between corresponding points. The template gesture that is at a least root-mean-square distance from the candidate gesture is chosen as the gesture performed.

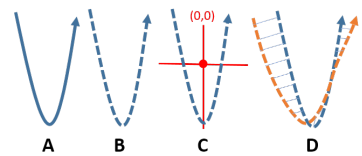

Figure 3: Pattern matching algorithm: A - Candidate gesture, B - Sampling, C - Centroid moved to origin (0,0), D - Computing Euclidean distance to a template gesture.

5 SYSTEM EVALUATION AND RESULTS

We evaluated the gesture recognition accuracy and usability of the system through a preliminary study by involving seven users (μ_{age} = 26.28). During the study, each user interacted with the browser by performing nine gestures, shown in Figure 2, to execute the associated actions like minimize, maximize, etc., on a browser as shown in Figure 1.B. The gaze interaction framework achieved a recognition accuracy of **93%** and a F-measure of **0.96**. The confusion matrix of the gestures performed is shown in Table 1. Also, the users shared that with practice, it was easy and quicker to perform gaze gestures than switching to a mouse and selecting the command from the menu, and the system was found to be responsive.

Table 1: Confusion Matrix - Template Matching. Key: A - Minimize, B - Maximize, C - Forward, D - Back, E - Scroll Down, F- Scroll Up, G - Refresh, H - New Tab, I - Close Tab

	A	B	C	D	E	F	G	H	I
A	0.88		0.04	0.04			0.04		
B		1.0							
C			1.0						
D				0.84	0.16				
E					1.0				
F	0.04			0.04		0.88		0.04	
G				0.12			0.88		
H				0.04				0.96	
I		0.04		0.04					0.92

6 CONCLUSION AND FUTURE WORK

We presented a gaze gesture-based interaction framework for situational impairments, accessibility, and rich interactions. The gesture recognition algorithm used addresses the limitations of the existing gaze gesture recognition technique, and also supports high accuracy. As the future work, more gestures will be added and the framework will be leveraged in various application contexts.

REFERENCES

Andreas Bulling, Florian Alt, and Albrecht Schmidt. 2012. Increasing the Security of Gaze-based Cued-recall Graphical Passwords Using Saliency Masks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 3011–3020. https://doi.org/10.1145/2207676.2208712

Andreas Bulling, Daniel Roggen, and Gerhard Tröster. 2009. Wearable EOG Goggles: Seamless Sensing and Context-awareness in Everyday Environments. *J. Ambient Intell. Smart Environ.* 1, 2 (April 2009), 157–171. http://dl.acm.org/citation.cfm?id=1735835.1735842

Heiko Drewes and Albrecht Schmidt. 2007. Interacting with the Computer Using Gaze Gestures. In *Human-Computer Interaction, INTERACT 2007*. Springer Berlin Heidelberg, Berlin, Heidelberg, 475–488. https://doi.org/10.1007/978-3-540-74800-7_43

Andrew T Duchowski. 2007. Eye tracking methodology. *Theory and practice* 328 (2007).

Fabian Göbel, Konstantin Klamka, Andreas Siegel, Stefan Vogt, Sophie Stellmach, and Raimund Dachselt. 2013. Gaze-supported Foot Interaction in Zoomable Information Spaces (Interactivity). In *Proceedings of the Conference on Human Factors in Computing Systems - Extended Abstracts*. ACM, 4.

Tracy Hammond, Georgia Tourassi, Hong-Jun Yoon, and Folami T. Alamudun. 2017. Geometry and Gesture-Based Features from Saccadic Eye-Movement as a Biometric in Radiology. (7 2017).

John Paulin Hansen, Allan W. Andersen, and Peter Roed. 1995. Eye-gaze control of multimedia systems. In *Symbiosis of Human and Artifact*, Yuichiro Anzai, Katsuhiko Ogawa, and Hirohiko Mori (Eds.). Advances in Human Factors/Ergonomics, Vol. 20. Elsevier, 37 – 42. https://doi.org/10.1016/S0921-2647(06)80008-0

John Paulin Hansen, Anders Sewerin Johansen, Dan Witzner Hansen, Kenji Itoh, and Satoru Mashino. 2003. Command without a click: Dwell time typing by mouse and gaze selections. In *Proceedings of Human-Computer Interaction–INTERACT*. 121–128.

John Paulin Hansen, Kristian Tørning, Anders Sewerin Johansen, Kenji Itoh, and Hirotaka Aoki. 2004. Gaze Typing Compared with Input by Head and Hand. In *Proceedings of the 2004 Symposium on Eye Tracking Research & Applications (ETRA '04)*. ACM, New York, NY, USA, 131–138. https://doi.org/10.1145/968363.968389

Robert JK Jacob. 1991. The use of eye movements in human-computer interaction techniques: what you look at is what you get. *ACM Transactions on Information Systems (TOIS)* 9, 2 (1991), 152–169.

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*. ACM, 421–430.

Chris Lankford. 2000. Effective eye-gaze input into windows. In *Proceedings of the 2000 symposium on Eye tracking research & applications*. ACM, 23–27.

Päivi Majaranta. 2011. *Gaze Interaction and Applications of Eye Tracking: Advances in Assistive Technologies: Advances in Assistive Technologies.* IGI Global.

Vijay Rajanna. 2016. Gaze Typing Through Foot-Operated Wearable Device. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '16)*. ACM, New York, NY, USA, 345–346. https://doi.org/10.1145/2982142.2982145

Vijay Rajanna and Tracy Hammond. 2016. GAWSCHI: Gaze-augmented, Wearable-supplemented Computer-human Interaction. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications (ETRA '16)*. ACM, New York, NY, USA, 233–236. https://doi.org/10.1145/2857491.2857499

V. Rajanna, A. H. Malla, R. A. Bhagat, and T. Hammond. 2018. DyGazePass: A gaze gesture-based dynamic authentication system to counter shoulder surfing and video analysis attacks. In *2018 IEEE 4th International Conference on Identity, Security, and Behavior Analysis (ISBA)*. 1–8. https://doi.org/10.1109/ISBA.2018.8311458

Vijay Rajanna, Seth Polsley, Paul Taele, and Tracy Hammond. 2017. A Gaze Gesture-Based User Authentication System to Counter Shoulder-Surfing Attacks. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems (CHI EA '17)*. ACM, New York, NY, USA, 1978–1986. https://doi.org/10.1145/3027063.3053070

Jacob O. Wobbrock, James Rubinstein, Michael W. Sawyer, and Andrew T. Duchowski. 2008. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *ETRA '08*. ACM Press, New York, New York, USA, 11. https://doi.org/10.1145/1344471.1344475

Jacob O. Wobbrock, Andrew D. Wilson, and Yang Li. 2007. Gestures Without Libraries, Toolkits or Training: A $1 Recognizer for User Interface Prototypes. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. ACM, New York, NY, USA, 159–168. https://doi.org/10.1145/1294211.1294238