Using Eye Tracking to Control Video Call Functions

A proposal on eye-tracking and natural user interfaces.

Robson Basha
100750867

***Abstract*—Eye-tracking can be a valuable tool for improving the quality of life for those unable to use traditional input methods, and can also make video calling a more natural and intuitive experience for all users. Despite this, eye-tracking has very minimal mainstream uses in the consumer world. This report explores the potential of using eye-tracking in video calling applications and identifies the key challenges and opportunities in creating a natural eye-tracking interaction framework. By understanding the benefits of eye-tracking in user interactions, it is possible to create a more efficient and user-friendly system that can be easily integrated into our own video-conferencing platform. The ultimate aim of this proposal is to make video calling a more natural and intuitive experience for all users through eye-tracking, as well as make eye-tracking a vital tool for enhancing the quality of life for people unable to use conventional input techniques.**

***Keywords*—** eye-tracking, control, interface

I. Introduction

Eye-tracking technology has come a long way throughout the last century. From the days of using invasive contact on the cornea, to the head-mounted solutions of the 1940s, to today's sleeker externally-mounted systems. [1] However, the use of eye-tracking technology has been very sparse in commercial products. One of the main reasons for this is privacy concerns. Gaze data gathered from eye-tracking can reveal a lot of information deemed sensitive by most. [2] This includes gender, age, ethnicity, and personality traits among others. [2] And in an age where data is being shuffled from third-party advertiser to third-party advertiser by companies who gathered it from their user base, it makes sense for users to be wary of technology that can siphon data so easily that they have little control over. However, this lies less on the hardware and more on those creating the software and handling the data gathered from the hardware. With enough belief behind the technology, eye-tracking has the potential to innovate many human-computer interfaces. One of these is how we interact with video chat applications. Software like Zoom, Google Meet, Discord, and Skype became not only popular but integral to keeping day-to-day activities both in and out of the workspace up and running during COVID lockdowns and work-from-home mandates. In fact, roughly 80% of Americans used video calling to communicate with others during the pandemic, with 20% doing video-conferencing at least once per day. [3] But how are some people able to use video calling interfaces if they aren't able to use traditional input methods? The previously mentioned pieces of software rely heavily on either a keyboard and mouse or a touchscreen to navigate different functions such as muting your microphone, typing in the chat box, or raising your hand. How would you use these features if you did not have fine motor skills in your hands? Or if you do not have hands at all? What if your hands are occupied with another task, such as caring for a child, or writing out notes in a journal? Eye-tracking can allow you to use these video calling apps as easily as you could with a mouse or touchscreen while taking these issues into consideration. The question is how do you design this to naturally conform to where your eyes will glance?

II. Methods

The main software we are going to use to develop this concept is Unity Version 2021.3.13f1 and the Tobii Unity SDK. This will be what we use to interface the video chat software with the Tobii Eye Tracker 5 hardware that we will be using to conduct the actual eye-tracking. This will be a strictly Windows PC-focused project, as Mac and Linux driver support from Tobii's end seems to be non-existent and would require looking into third-party solutions. The main focus of research is on how we will format the user interface to naturally lean toward where the user's eyes will fall when wanting certain actions done. One idea in mind right now is to substitute the push-to-talk concept — which has the user hold a button for as long as they want to talk — with a ”look-to-talk” function. This would involve the user looking directly at the main speaker to automatically unmute their microphone and have it immediately re-mute when not looking at them. See Figure 1 for a visual example of this concept. Another idea is to switch the spotlighted speaker (i.e. the main speaker) position to a different member of the call. Rather than right-clicking on the user to open a context menu, glancing at them and using some sort of confirm action (blinking, staring for two seconds, etc.) can switch their position in the call. The confirmation action will be dependent on what Tobii's API is capable of doing with the eye-tracking data. Using a button to confirm is also possible, but we want to avoid that as it gets in the way of a completely hands-free video call experience. See Figure 2 for a visual example of this concept. We also want to use Agora software in Unity to create an actual video call solution with this eye-tracking technology baked in. Please refer to Figure 3 for our Gantt chart showing our estimated development timeline.

III. Results

Current video-conferencing software has used eye-tracking via the user's webcam in different ways. For example, Zoom has used eye-tracking to see where attendees of a call are looking and how much attention they are giving to the speaker(s). [4] There have also been many uses of eye-tracking to make the user always look like they are looking directly into the camera, such as NVIDIA's recent solution. [5] None of these solve the problem of having an interface that can be interacted with using solely eye-tracking, which is where our solution will fill the gap.

IV. Conclusion

In conclusion, we find a significant lack of killer demonstrations of what eye-tracking can do in mainstream technology. We think it can revolutionize the way we interact with all types of computing devices, including the traditional desktop setup. This can be especially apparent with video calls, where traditional input methods such as a keyboard and mouse or a touchscreen can be considered overkill and cumbersome. Our aim is to create an interface that allows for natural navigation of video call features such as muting your microphone, raising your hand, and leaving a call, using nothing but tracking the movement of a user's eyes. Eye-tracking can be considered a privacy nightmare when used by certain untrustworthy software companies, but that is not a concern with our solution as all the eye-tracking data will be local to the user's device and destroyed on the exit of the application.

References

[1] R. J. K. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and Usability Research," The Mind's Eye, pp. 573–605, 2003.

[2] J. L. Kröger, O. H.-M. Lutz, and F. Müller, "What does your gaze reveal about you? on the privacy implications of Eye Tracking," Privacy and Identity Management. Data for Better Living: AI and Privacy, pp. 226–241, 2020.

[3] S. Atske, "1. how the internet and technology shaped Americans' personal experiences amid covid-19," Pew Research Center: Internet, Science amp; Tech, 28-Apr-2022. [Online]. Available: https://www.pewresearch.org/internet/2021/09/01/how-the-internet-and-technology-shaped-americans-personal-experiences-amid-covid-19/. [Accessed: 23-Jan-2023].

[4] "Eye tracking reveals where people look during Zoom, Webex," RSS. [Online]. Available: https://www.news.iastate.edu/news/2022/02/07/zoom-eye-tracking. [Accessed: 23-Jan-2023].

[5] S. Ahmed, "Nvidia adds uncanny eye-tracking features to broadcast software," Dexerto, 13-Jan-2023. [Online]. Available: https://www.dexerto.com/tech/nvidia-eye-tracking-broadcast-software-2031892/. [Accessed: 24-Jan-2023].

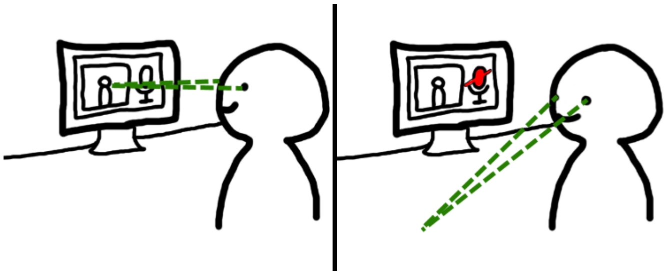


Fig. 1. Concept artwork of unmuting the microphone when looking directly at the other person.

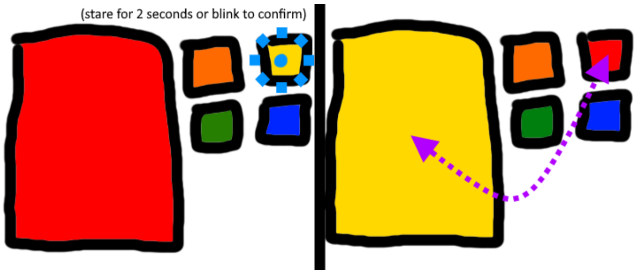
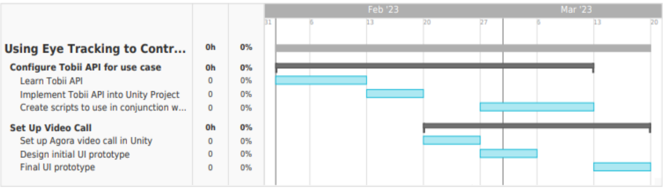


Fig. 2. Concept artwork of making a member of the call the main speaker by looking at their window and confirming the switch.



Using Eye Tracking to Contr...	**0h**	**0%**
Configure Tobii API for use case	**0h**	**0%**
Learn Tobii API	0	0%
Implement Tobii API into Unity Project	0	0%
Create scripts to use in conjunction w...	0	0%
Set Up Video Call	**0h**	**0%**
Set up Agora video call in Unity	0	0%
Design initial UI prototype	0	0%
Final UI prototype	0	0%

Fig. 3. Gantt chart for the project's main development, going from Feburary to late March of 2023.