Characterizing Eye Gaze for Assistive Device Control

Larisa Y.C. Loke[1,2], Demiana R. Barsoum[1,2], Todd D. Murphey[1], and Brenna D. Argall[1,2]

***Abstract*— Eye gaze tracking is increasingly popular due to improved technology and availability. However, in assistive device control, eye gaze tracking is often limited to discrete control inputs. In this paper, we present a method for collecting both reactionary and control eye gaze signals to build an individualized characterization for eye gaze interface use. Results from a study conducted with motor-impaired participants are presented, offering insights into maximizing the potential of eye gaze for assistive device control. These findings can inform the development of continuous control paradigms using eye gaze.**

I. INTRODUCTION

Humans interact with assistive technology systems through accessible control interfaces. For individuals with motor impairments, these assistive technologies can help to mitigate limitations on their independence and ability to participate in society—such as using a computer for communication or a powered wheelchair for independent mobility.

An individual's level of motor impairment determines the types of control interfaces that they can effectively use. In general, the interfaces accessible to users with greater severity of motor impairment become increasingly limited in the richness of control commands that can be issued through these interfaces—including the number of simultaneous degrees of freedom that can be controlled and the resolution of the signal in terms of direction and magnitude.

For example, individuals with higher levels of spinal cord injury may be unable to use a joystick, which is the most common interface for controlling powered wheelchairs. For these users, one standard alternative interface is the sip-and-puff device, which however introduces the following challenges in comparison to joystick control: It only allows for the operation of one (of the two) wheelchair control dimensions at a time, and its signals are interpreted in a discrete manner. As a result, the physical motion of the powered wheelchair can be choppy and imprecise, and the control can be difficult to learn and be mentally (and often physically) taxing [1].

Eye gaze is one alternative input mechanism for assistive device interfacing currently gaining traction within the field. Eye gaze trackers are especially useful for individuals with neurodegenerative diseases such as amyotrophic lateral sclerosis (ALS), as voluntary eye movement is retained until the disease has progressed to terminal stages [2]. Using eye gaze trackers for communication has been shown to restore agency to ALS patients even as the motor impairment progresses—leading to improvements in quality of life [3]. A natural extension in the use of eye gaze trackers is to help restore independent mobility. However, due to reasons of safety and ease of use, current commercial eye gaze systems for controlling powered wheelchairs only give users access to discrete switch control [4], [5]. This limits controllability for users, which may lead to a perceived loss of user agency, in turn resulting in frustration, disuse, or even injury [6].

In the field of eye-based human-computer and human-robot interaction, there has been extensive work done in the areas of eye movement event detection, human intent inference from eye motion, and usability studies of user interfaces. However, in the domain of eye-based control, the field has largely converged on eye gaze as a complementary input mode to other modalities (e.g., used together with a joystick or buttons, rather than the sole input mode [7], [8]), and there is limited work investigating the (sole) use of eye gaze inputs for *explicit* and *continuous* assistive device or robot control. In order to design an eye-based user interface for continuous robot control, a better understanding of the characteristics of eye gaze signals when operating an eye-based input system, as well as limitations in the eye gaze signal, is necessary.

This work presents the following contributions:

1) A suite of open-source assessment and data gathering tasks for use with an eye gaze tracking interface.[1]
2) An open-source system for interfacing an eye gaze tracker for real-time control,[1] integrated within the Robot Operating System (ROS) [9] software suite.
3) An end-user study that employs these tools to collect data for an individualized characterization of eye gaze for control, to provide further insights into the design of interfaces for systems that use eye gaze as input.

In Section II, we provide background on the related literature. In Section III, we present the contributed virtual tasks. In Section IV, we detail the data gathering pipeline, as well as the experimental setup and protocol. In Section V, we present and discuss the results from the end user study, followed by conclusions and avenues for future work in Section VI.

II. BACKGROUND

In this section, we provide background on related literature on eye gaze for control and interface customization.

A. Eye Gaze for Control

Alternative control interfaces such as eye gaze trackers are gaining popularity with their increasing commercial availability and technological improvements. In addition to its use as a clinical control interface, eye gaze tracking has been increasingly adopted in the fields of human-computer inter-

*Funding for this work was provided by UL Research Institutes through the Center for Advancing Safety of Machine Intelligence.

[1]Northwestern University, Evanston, IL., USA
[2]Shirley Ryan AbilityLab, Chicago, IL., USA

[1]https://github.com/argallab/eyegaze_characterization_tasks

action (HCI) and human-robot interaction (HRI) to evaluate the usability of interfaces [10], as well as for control input in robot teleoperation and shared control contexts [11]. Work in this latter area includes improving the interpretation of these control signals, as well as to improve the control of assistive devices in conjunction with robot autonomy [12].

In the field of robotics, systems use eye gaze as an *explicit control input*—where users use voluntary eye movements to issue control commands and interact with the input space, usually a screen—to teleoperate robotic manipulators [13], mobile robots [14], and powered wheelchairs [15]. Eye gaze can also be used as an *implicit control input* to robotic systems, where user intention or goals are inferred from eye gaze data and used to provide control assistance [16]. Thus, eye gaze is a multi-interpretable interface, with both explicit and implicit ways to interpret the control input.

Using eye gaze for input presents challenges due to the dual role of vision in both perceiving the scene and issuing explicit control commands. This can lead to the 'Midas touch' problem, wherein there is a lack of differentiation between viewing of the scene for perception versus intentional control. Proposed solutions include combining eye gaze with another input modality, which may not be viable for individuals with limited motor function or implementing a dwell time requirement for control selection [7]. However, the latter approach may result in a perceived decrease in system responsiveness [8]. Hence, the improvement of paradigms using pure eye gaze input are of interest.

When used for powered wheelchair driving, many interfaces implement *zones* on the screen, where looking at a certain area of the screen activates a virtual button that issues a discrete control command for the wheelchair [5], [17]. Like all discrete control interfaces, this presents challenges in controllability for the user, which can be further confounded by the 'Midas touch' problem if simultaneously scanning the environment while issuing control commands results in unintended or jerky motion. In this work, we identify and analyse eye gaze characteristics during eye-based control towards the aim of uncovering alternative ways of using the gaze signal, such as for simultaneous explicit and implicit control, as well as continuous control.

B. The Importance of Customization

Input signals measured by interfaces are subject to interpretation, and discrepancies between a user's intended control commands and those received by the machine can exist. User-specific calibration (software or physical) can help to address these discrepancies. Variability in physiology, behavior, and skill over time may result in a need for regular recalibration [18]. In the domain of eye gaze control, accuracy and usability of the system is highly dependent on good calibration of the system to the user [8]. However, calibration of gaze tracking systems is often described as tedious and difficult [19]. In this work, we look at eye gaze characteristics on an individual basis in order to better inform alternative avenues for customization in eye-based control systems, in addition to standard calibration procedures.

III. EYE GAZE CHARACTERIZATION PIPELINE

Previous work that evaluates eye-gaze based input systems for assistive devices has been application-oriented, with evaluations based on success in completing tasks *performed by the assistive device* (such as an application-specific GUI, or powered wheelchair) which is controlled via eye gaze [15], [20]. For the purpose of *characterizing* eye-based control input, tasks that are independent of the application domain (specifically, of the assistive device being controlled) are needed. In an attempt to isolate the features of interest being measured with each task, we design a suite of characterization tasks to collect eye gaze data for specific types of gaze movement patterns and metrics during eye-based control. Gaze patterns and metrics of interest include saccades, fixations, and smooth pursuits, as these types of eye movement events are commonly used in gaze-based control interfaces [7], [21], and also expected to occur in the domain that we are interested in—assistive device control—as it involves observation of a dynamic environment [20].

The eye gaze data collection pipeline is implemented within the Robot Operating System (ROS) [9], an open-source software suite used by millions across the world. The pipeline consists of a series of ROS2 software nodes.

- A *streamer node* publishes data from the eye gaze tracker to ROS2.
- A *receiver node* operates on the gaze data streams to handle real-time gaze position transforms and moving-window average smoothing of the signal.
- *Task controller nodes* handle calculations relating to task progress and broadcast relevant data streams for task control to Unity®.

The eye gaze characterization tasks are screen-based tasks implemented in the Unity® Engine (Unity Editor Version 2021.3.10f1) [22], described as follows.

A. Painting Task

<u>Features of interest</u>: Distribution of eye gaze, distribution of fixations, areas of visual neglect.

<u>Task description</u>: Participants' eye gaze position controls a virtual paintbrush. They are asked to 'paint' the entire screen with their eyes within 5 minutes.

B. Focus Task

<u>Features of interest</u>: Saccades, fixations, dwell time.

<u>Task description</u>: Participants viewed and fixated on 60 randomly appearing circular targets of varying sizes on the screen and were asked to maintain their gaze for 2 seconds. If 2 seconds of continuous fixation was not maintained by the end of 10 seconds, the target timed out.

<u>Visual feedback</u>: Participants are given feedback on how much longer they need to fixate on the target via variations in opacity. A target becomes more transparent the longer participants dwell on it, up to 2 seconds when the target fades completely and the next target appears. Two variants of the task exist: (1) Participants are not given any visual feedback of where their measured gaze is. (2) Participants are provided with this feedback in the form of a moving dot on the screen.

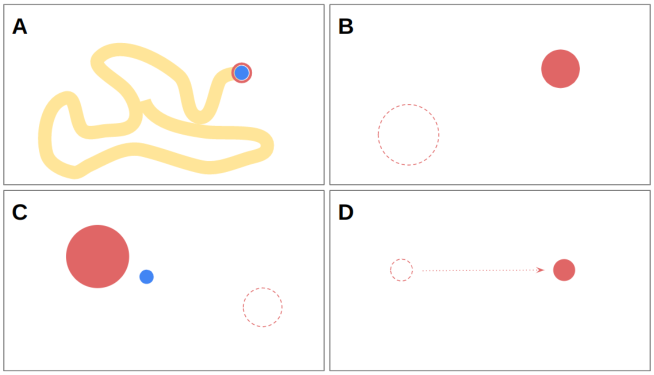

Fig. 1: The screen-based tasks implemented in Unity2D. Red dashed circles represent where the target dot was previously, and filled circles represent where the current target dot is. **A:** The Painting Task. Participants controlled a blue cursor while a red circle indicated a 5-minute countdown. **B:** The Focus Task *without* visual feedback for gaze position. Targets fade away as participants fixate on the target continuously for 2 seconds, and the next target appears. **C:** The Focus Task *with* visual feedback for gaze position. Participants have a blue dot representing the gaze position measured by the eye gaze tracking system for visual feedback. **D:** The Tracking Task. Participants track moving targets on the screen.

Fig. 3: A participant doing the Painting Task. ArUco markers on the corners of the screen are used for transforming the gaze position in the scene to gaze position on the screen.

C. Tracking Task

Features of interest: Smooth pursuit.

Task description: Twenty-four moving targets (circles) appear on the screen one after another. Participants are tasked with following the targets with their gaze. Targets move at a speed of approximately 15 degrees of visual angle per second.

Visual feedback: Participants are given feedback on the proximity of their measured gaze to the moving target by a variation in the opacity of the target. (More opacity indicating being closer to the target.)

An illustration of each of the screen-based tasks described is shown in Fig. 1. The code for the data collection pipeline is made publicly available on github.

IV. METHODOLOGY

A characterization study was carried out to collect eye gaze user data from individuals with motor impairments.

A. Participants

In total, 11 participants were recruited: 1 individual with ALS (62 y.o., female) and 10 individuals with spinal cord injury (SCI) (41.7 ± 11.3 y.o., (9 male, 1 female)). All participants were screened for ability to use an eye gaze tracker to interact with the screen via a simple test described in the experiment procedure. All participants gave their informed, signed consent to participate in the experiment, which was approved by Northwestern University's Institutional Review Board (STU00217297).

B. Hardware and Materials

Fig. 2: Tobii Pro Glasses 3 (TPG3) head-mounted eye tracker [23].

The study used a head-mounted eye gaze tracker (Tobii Pro Glasses 3, 100 Hz gaze data rate, 1920p by 1080p @ 25 fps scene camera) as the eye gaze interface (Fig. 2). The Tobii Pro Glasses 3 (TPG3) uses corneal reflection dark pupil eye tracking, and provides scene camera video, gaze position in the scene camera frame, as well as 3D gaze position. Using the scene camera video, gaze position was transformed in real-time to position on the screen and the transformed signal was used as control input for study tasks. While not analyzed in this work, the tracker also provides pupil diameter and IMU sensor information, which could be used as additional sources of implicit input to a control system. The TPG3 has been shown to yield more accurate eye tracking results than the earlier model (Tobii Pro Glasses 2) model, which has been used extensively in previous eye tracking research [24].

In this work, the streamer node (Section III) of the data collection pipeline provided a Python websocket client to communicate with the TPG3 in order to read signal streams which were then published to ROS2 topics. Since study tasks were controlled using the position of gaze on the screen, other screen-based and head-mounted eye trackers could be interfaced with the data collection pipeline as future work.

C. Experiment Procedure

Participants were seated at a 60 cm viewing distance from a 55.9 cm (22-inch) 1600p by 900p screen and fitted with the head-mounted TPG3 eye gaze tracker (setup shown in Fig. 3). For those who wear glasses, corrective lenses matching their glasses power were added to the TPG3. The eye gaze tracker then was calibrated using the TPG3's standard built-in calibration procedure. After calibration, a simple sanity check test was conducted to verify that participants were able to use the eye gaze tracker to issue control inputs—participants were asked to look at the four corners of the screen and the recorded signal was verified to match the expected values. After this, participants proceeded with the study tasks.

Each task consisted of a training phase and a testing phase. In the training phase, participants were introduced to and familiarized with the task via a shortened version of the full-length task. The testing phase then consisted of executing the given task for the prescribed duration or number of repetitions (as described in Section III). The presentation of tasks was fixed across participants: (1) Painting Task, (2) Focus Task *without* gaze position feedback, (3) Focus Task *with* gaze position feedback, and (4) Tracking Task. After each task, participants were asked to fill out a NASA Task Load Index (TLX) [25] survey.

After completing all 4 tasks, participants were asked to fill out a questionnaire including Likert-scale questions about their experience using the eye gaze tracker, as well as open-ended questions about eye gaze trackers and assistive devices in their daily lives.

D. Analysis Methods

Eye movement event detection is an established field of research with various algorithmic approaches proposed in the literature [26]. The process of eye movement event detection often involves a pre-processing step, that de-noises the gaze signal by way of filtering, followed by the classification of gaze signal segments as different types of eye movements such as *saccades*, *fixations*, and *(smooth) pursuits*. This classification, also known as event detection, can be done by parametric or non-parametric methods, wherein parametric methods make use of velocity, acceleration, and temporal thresholds to identify types of eye movements, and non-parametric methods often make use of machine learning techniques to extract inherent patterns in the gaze signal. For this work, we used the open-source REMoDNaV eye movement classifier implemented in Python [27] to do offline classification of gaze movements.

V. Results and Discussion

This section presents the eye movement characteristics and metrics gathered while operating our eye-gaze characterization tasks. We direct our discussion to the context of possible use cases for powered wheelchair driving control. A pain point in current eye-based input methods for driving powered wheelchairs is the lack of commercial availability of continuous control [15], and so we focus in particular on continuous inputs. Reported statistical significance between groups is determined by a non-parametric ANOVA (Kruskal-Wallis H-test). Where specified, eye movement distance is reported in degrees of visual angle (DVA), computed using screen size, viewing distance, and screen resolution (with our experimental setup, 1 pixel $\equiv 0.0269$ DVA).

A. Spatial Layout of User Interface

During the Painting Task, participants were encouraged to explore the space of the screen as much as possible. Fig. 4 shows the spatial distribution of recorded eye gaze signals and fixations. We observe a large inter-participant variability for both the spatial distributions as well as fixations.

To ensure users are able to issue intended commands, the user interface should be designed in a way that allows users to operate in the area of the screen that they are most comfortable and skilled in. For instance, S002 would likely benefit from a user interface that expects eye gaze input in the upper-left region of the screen, while S010 would likely benefit from an interface that expects eye gaze input in the lower region of the screen.

B. Virtual Button Sizes

In Fig. 5 the number of successful targets (out of 10 targets total) is shown for the Focus Task, grouped by target size and both *without* visual feedback and *with* visual feedback. From the results, we can see that most participants were able to successfully fixate for 2 seconds consecutively on targets that had a diameter larger than 5 degrees of visual angle. This would suggest that, in general, virtual on-screen icons that require dwell to activate should not be smaller than 5 degrees of visual angle in size. Customization of icon size can also be done according to users' success at such a task.

When visual feedback was provided, participants were able to successfully achieve targets with an even smaller diameter (4.0 degrees of visual angle), suggesting that on-screen icons may be even smaller if users have visual feedback.

C. Continuous Pursuit Motion

Continuous inputs can likely be achieved by pursuit motions, which are slower and more controlled than saccadic motions. Figs. 6 and 7 show the distributions of the time duration and magnitude (in degrees of visual angle) of smooth pursuits. In general, smooth pursuit segments during the Tracking Task tend (statistically significant in 9 of 11 participants) to be longer in mean duration and of a greater magnitude than those during the Painting Task. This is likely due to the Tracking Task providing a moving target for the eyes to focus on, while the Painting Task requires intentional motion of the eye to move the cursor on the screen.

One interpretation of this result for interface design could be to use the median of an individual's magnitude of smooth pursuit to inform the control resolution of a virtual joystick on a screen—individuals able to more frequently provide larger distances of smooth pursuit when controlling a cursor with their eyes might be able to control a virtual joystick over a larger visual area, offering higher control resolution.

Another interpretation of this result is that for some individuals, it may be possible to differentiate whether the eyes are being used for control input versus being used to scan the environment, based on the duration of smooth pursuits over a short time history. This may be used to more intelligently overcome the 'Midas touch' problem.

D. Visual Feedback for Gaze Location

Fig. 8 shows the overall target completion success rate and mean target completion time for the Focus Task, under both visual feedback conditions. In terms of task success, we see that participants performed better overall with visual feedback than without visual feedback. However, participants tended to take longer to successfully complete targets with visual feedback. Due to the limited sample size, statistical significance was not observed in these results; however, they suggest the presence of a potential difference.

Interestingly, for participants that performed better without feedback, anecdotal evidence suggests that visual feedback (in this case the blue dot displaying measured gaze position on the screen) was distracting and resulted in 'cursor chasing' due to perceived inconsistencies between the perceived gaze location and the displayed measured gaze location [28].

E. Task Workload

Fig. 9 shows the NASA TLX scores for all participants after completing each of the tasks. A high TLX score insinuates more cognitive workload. From the results, there is a lot of variability in the perceived mental workload across participants, with scores ranging from zero workload

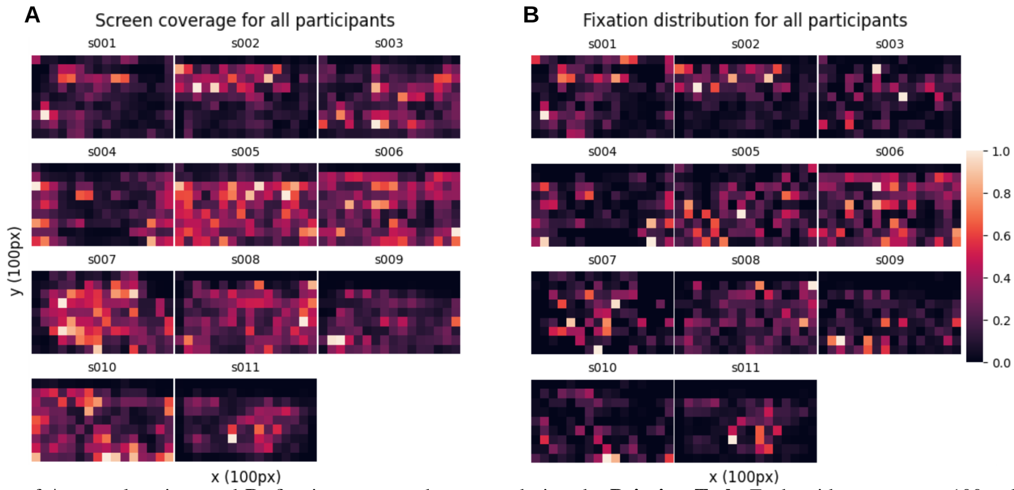

Fig. 4: Distirbutions of **A:** gaze locations and **B:** fixations, across the screen during the **Painting Task**. Each grid represents a 100px by 100px area. The intensities of the heatmap are normalized on a scale from 0 to 1 for each participant, and normalization is done relative to the counts in the most frequently occurring 100px by 100px area in the heatmap.

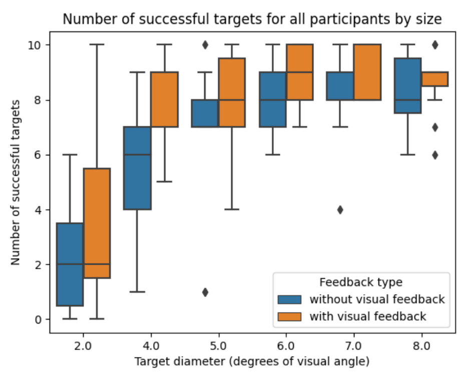

Fig. 5: Number of successful targets, grouped by target size, for all participants during the **Focus Task** under both feedback conditions.

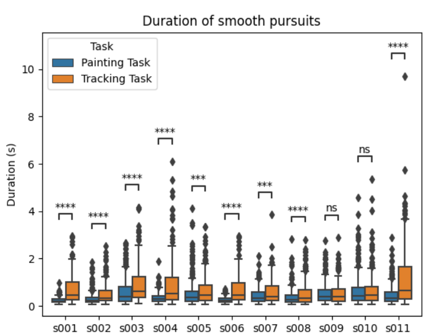

Fig. 6: Duration of smooth pursuit segments for all participants during the **Painting Task** and **Tracking Task**.

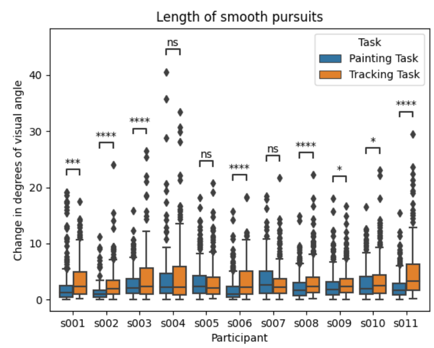

Fig. 7: Length (magnitude) of smooth pursuit segments in degrees of visual angle for all participants during the **Painting Task** and **Tracking Task**.

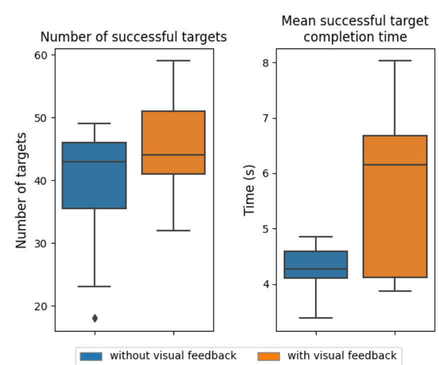

Fig. 8: Number of successful targets (*left*) and mean time taken to complete successful targets (*right*) for all participants during the **Focus Task**, comparing *without* and *with* visual feedback for measured gaze position.

to near maximal. For a given participant, we further observe instances of consistency across tasks (e.g., consistently high cognitive load for all tasks) as well as marked variability (e.g., some tasks rated low and others high).

Most participants reported lower TLX scores for the Tracking Task (4th task). Since the Tracking Task requires participants to follow a moving target (reactionary) as opposed to controlling a cursor (as in the Painting Task), it makes sense that participants would find it less mentally taxing.

Anecdotally, participants reported that one of the largest difficulties they faced with using the eye gaze tracker was a perceived steep learning curve. The TLX scores across the tasks may have captured learning effects in some participants. In particular, S002 and S007 generally reported decreased levels of workload from the first to the last task while S006 self-reported zero workload for the 3rd and 4th tasks, citing that they had now learned and understood the system.

F. Limitations

This study is subject to several limitations. A small sample size of participants makes it challenging to establish statistical significance, and the fixed task order for all participants may have influenced the results through learning or fatigue effects, which may bias TLX scores. Additionally, the gaze

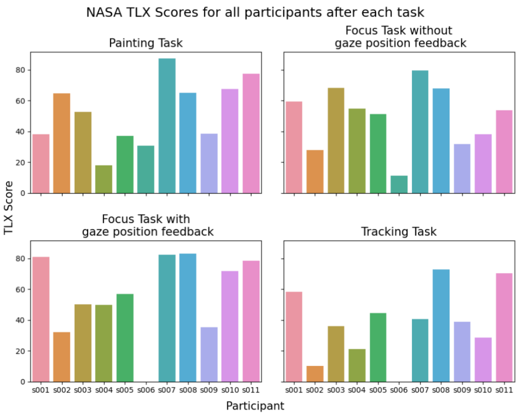

Fig. 9: NASA TLX Scores for all participants across the different tasks

characterization tasks developed for this study lack validation in accurately measuring the intended gaze characteristics, and the study was conducted under specific experimental conditions and may not be representative of eye tracking in the real world applications. Future work could address these limitations by expanding the participant pool, conducting validation studies to demonstrate reliability and accuracy of these tasks, and varying the study environment to better mimic real world scenarios.

VI. CONCLUSIONS

In this paper, we present an open-source data collection pipeline to characterize eye gaze for device control. This pipeline consists of a suite of screen-based assessment and data gathering tasks to characterize eye gaze movements during eye-based control, and a system to use an eye gaze tracker for real-time control input with ROS. We probed individuals' abilities to interact with a screen using eye gaze and a variety of eye movements: namely saccadic motions, fixations, and smooth pursuits. Probing these characteristics is the first step towards allowing us to identify parameters towards designing a customizable and adaptive control interface design. Towards this end, we conducted an end-user study with our eye-gaze characterization system and present insights from the data with regards to the design of an explicit continuous eye gaze tracking user interface. Our future work will present users with different eye input user interface designs based on these insights and assess usability.

ACKNOWLEDGMENT

The authors thank Dr. Colin K. Franz, Assistant Professor of Physical Medicine & Rehabilitation and Neurology at Northwestern University, and Ed Hitchcock, Occupational Therapist at the Shirley Ryan AbilityLab, for their advice on study design and eye gaze interfaces, and assisting with participant recruitment.

REFERENCES

[1] G. Kelliher, J. Farrelly, Q. Afridi, F. Arubi, and B. Kane, "Designing control interfaces for powered wheelchair users," *Preface 4 Acknowledgements 6 Keynote Address 11 IxDA industry Session 15 Full Papers*, p. 78.

[2] K. C. Kanning, A. Kaplan, and C. E. Henderson, "Motor neuron diversity in development and disease," *Annual review of neuroscience*, vol. 33, pp. 409–440, 2010.

[3] A. Calvo, A. Chiò, E. Castellina, F. Corno, L. Farinetti, P. Ghiglione, V. Pasian, and A. Vignola, "Eye tracking impact on quality-of-life of als patients," in *International Conference on Computers for Handicapped Persons*. Springer, 2008, pp. 70–77.

[4] "Gazedriver – mobilitet," https://gazedriver.com/en/, (Accessed on 06/28/2022).

[5] T. Technologies, "Ability drive adt," https://tolttech-media.s3.us-west-2.amazonaws.com/product-catalog/ADT-Product-Card-v2201.pdf, (Accessed on 06/28/2022).

[6] R. Parasuraman and V. Riley, "Humans and automation: Use, misuse, disuse, abuse," *Human Factors*, vol. 39, no. 2, pp. 230–253, 1997.

[7] P. Majaranta and A. Bulling, "Eye tracking and eye-based human–computer interaction," *Advances in physiological computing*, pp. 39–65, 2014.

[8] R. J. Jacob, "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*, 1990, pp. 11–18.

[9] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[10] P. E. Cairns and A. L. Cox, *Research methods for human-computer interaction*. Cambridge University Press, 2008.

[11] R. M. Aronson and H. Admoni, "Eye gaze for assistive manipulation," in *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, 2020, pp. 552–554.

[12] B. D. Argall, "Autonomy in rehabilitation robotics: An intersection," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, p. 441, 2018.

[13] S. Li, X. Zhang, and J. D. Webb, "3-d-gaze-based robotic grasping through mimicking human visuomotor function for people with motion impairments," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 12, pp. 2824–2835, 2017.

[14] C. Carreto, D. Gêgo, and L. Figueiredo, "An eye-gaze tracking system for teleoperation of a mobile robot," *Journal of Information Systems Engineering & Management*, vol. 3, no. 2, p. 16, 2018.

[15] J. M. Araujo, G. Zhang, J. P. P. Hansen, and S. Puthusserypady, "Exploring eye-gaze wheelchair control," in *ACM Symposium on Eye Tracking Research and Applications*, 2020, pp. 1–8.

[16] J. Stolzenwald and W. W. Mayol-Cuevas, "Rebellion and obedience: The effects of intention prediction in cooperative handheld robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 3012–3019.

[17] C.-S. Lin, C.-W. Ho, W.-C. Chen, C.-C. Chiu, and M.-S. Yeh, "Powered wheelchair controlled by eye-tracking system." *Optica Applicata*, vol. 36, 2006.

[18] M. Gillham, M. Pepper, S. Kelly, and G. Howells, "Feature determination from powered wheelchair user joystick input characteristics for adapting driving assistance," *Wellcome Open Research*, vol. 2, 2017.

[19] K. Pfeuffer, M. Vidal, J. Turner, A. Bulling, and H. Gellersen, "Pursuit calibration: Making gaze calibration less tedious and more flexible," in *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 2013, pp. 261–270.

[20] L. E. Sibert and R. J. Jacob, "Evaluation of eye gaze interaction," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 2000, pp. 281–288.

[21] A. T. Duchowski, *Eye tracking methodology: Theory and practice*. Springer, 2017.

[22] Unity Technologies, "Unity," Game engine, 2021, version: 2021.3.10f1. [Online]. Available: https://unity.com/

[23] (2021) Tobii pro glasses 3—latest in wearable eye tracking. [Online]. Available: https://www.tobii.com/products/eye-trackers/wearables/tobii-pro-glasses-3

[24] V. Onkhar, D. Dodou, and J. de Winter, "Evaluating the tobii pro glasses 2 and 3 in static and dynamic conditions," 05 2023.

[25] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 2006.

[26] M. Startsev and R. Zemblys, "Evaluating eye movement event detection: A review of the state of the art," *Behavior Research Methods*, pp. 1–62, 2022.

[27] A. H. Dar, A. S. Wagner, and M. Hanke, "Remodnav: robust eye-movement classification for dynamic stimulation," *Behavior research methods*, vol. 53, no. 1, pp. 399–414, 2021.

[28] S. Vickers, H. Istance, A. Hyrskykari, N. Ali, and R. Bates, "Keeping an eye on the game: Eye gaze interaction with massively multiplayer online games and virtual communities for motor impaired users." 2008.