Eye Tracking for Interactive Accessibility

A Usability Analysis of Alternative Communication Interfaces

Kaan Sergen Arslan

Cognitive Science Department, Middle East Technical University, Turkey, kaan.s.arslan@gmail.com

Cengiz Acartürk

Cognitive Science Department, Middle East Technical University, Turkey, acarturk@metu.edu.tr

ABSTRACT

At the late stages of ALS, (Amyotrophic Lateral Sclerosis) individuals may lose their ability to speak, eat, move, and even breathe without ventilator support. Alternative modalities of communication, such as gaze, may have a potential to provide a communication interface that may increase the quality of life in cases as such. This study aims at developing interactive interfaces for gaze-based communication, and it reports a small-scale usability analysis for a comparison of the interfaces. In particular, we focus on two dimensions in the usability investigation: The interface layout design and methods of eye tracking. The interface layouts include a virtual QWERTY keyboard, an alphabetical keyboard, and a cascades menu. The eye tracking methods include a wearable eye tracker and a webcam-based eye tracker.

CCS CONCEPTS

• Human-centered computing~Accessibility~Empirical studies in accessibility

• Human-centered computing~Accessibility~Accessibility design and evaluation methods

KEYWORDS

Eye-tracking, Gaze communication, Usability analysis

1 Introduction:

The number of ALS diagnosed patients was 222,800 in 2015, which is expected to increase by 69% to 376,674 in 2040 due to the aging of the population (Arthur et al., 2016). According to the ALS Association, someone is diagnosed with the ALS disease in every 90 minutes (ALS Association, 2021) and 5,000 new cases diagnosed each year in the US (Hopkins Medicine, 2021). The statistics indicate an urgent need for solutions that may contribute to remedy the situation in multiple fronts. Recent developments in eye-tracking technologies bring an opportunity to improve the quality of life of ALS diagnosed citizens by providing an alternative communication modality as such. The potential of computer-based speech synthesizers using eye-tracking technology has already been recognized as a means to increase the quality of life of citizens with motor disabilities (National Institute of Neurological Disorders & Stroke, 2019). In this study, a comparative analysis of alternative interface designs is reported. In particular, we compare the use of an on-screen keyboard vs. a hierarchical window in a simulated setting, where participants with no motor disabilities tested the systems for functionality and usability. We employed the Haar-like approach, which is a feature-based approach used in hierarchical applications. The approach is advantageous in designing complexity, also in terms of relatively low hardware requirements. The drawback is the lower resolution when compared to infrared eye tracker technologies. Therefore we used the infrared eye tracker both in the on screen keyboard application and the hierarchical application. We compared the two approaches in terms of their functionality and usability. In particular, we aimed at contributing to available research by studying the functionality and usability of alternative interfaces. We report four applications, which are software programs developed for a desktop or a laptop computer. The applications use either a webcam or an eye tracker hardware (in this case, an open-source eye tracker, viz. Pupil Core headset). The applications aim to detect and analyze eye movements and create a communication environment.

2 Background and Relevant Work

In Human Computer Interaction (HCI), multiple methodologies have been proposed as methods of interaction and communication for users with motor disabilities. The computer control methods use various characteristics of communication channels, such as the ones that employ the functionalities of the users' tongue, eye movements, and brain signals (Salem & Zhai, 1997; Jacob, 1990; Barreto et al., 1999; Barreto et al., 2000). These methods have been used for running multiple system commands for controlling the interface, in particular cursor movement and/or icon selection mechanisms. Besides the methods that employ tongue and biosignals, eye movements have been used for cursor and selection operations in computer interfaces. Gaze tracking applications have been developed either with computer webcams or IR cameras (Wanluk et al., 2016; Liu et al., 2010). In certain cases, where the level of motor functionality is very limited, for example when the users have only the ability to move their eyes, eye tracking applications have the potential to provide an alternative communication channel. Both on-screen keyboards and mouse cursor applications have been used for this purpose.The eye tracking methods also have the potential to be complementary methodologies to other BCI interfaces.

In terms of the design and development of interfaces for this purpose, a major technical requirement is the detection of the face and the gaze within the face image. There are two different approaches for face detection techniques: The feature-based approach and image-based approach (Ranjan et al., 2019 for a review). In this study, the feature-based approach was selected due to implementation advantages and the requirement of high CPU power consumption. The feature-based approach has advantages in terms of implementation due to the availability of libraries for development, especially when using the image-based approach. For example, the feature-based approach requires feature extraction for object detection, face detection in our case. For this purpose, Haar-like features are used. The Haar-like features (Viola & Jones, 2001) have been implemented in numerous studies. We employed the Haar algorithm and OpenCV library to detect face and eye. OpenCV (Open Source Computer Vision Library) is an open-source software library with the purpose of computer vision and machine learning (OpenCV Team, 2019). OpenCV has been employed for various eye tracking applications such as detecting the driver's drowsiness and fatigue (Nguyen & Demidenko, 2015; Li et al., 2011), real-time attitude recognition (Lv et al., 2011), smart wheelchair design (Wanluk et al., 2016), and low-cost communication system (Lupu et al., 2013). More generally, eye-tracking is a method whereby the position of the eye is used to know where the individual is looking at any given time and sequence in which the eyes of the individual are moved (Poole & Ball, 2006). Since eye tracking is important for researchers trying to understand the movements of the eye while individuals express indifferent actions, different techniques are developed over the years (Lupu & Ungureanu, 2013). Various eye-tracking applications have been developed for people with severe motor impairments and Neuro-motor disabilities. For example, the *EyeAssist* is a low cost, user-friendly communication aid developed for citizens with neuro-motor disabilities (Khasnobish et al., 2017). The program uses an eye-tracker hardware of the *EyeTribe* company (which is obsolete since 2017). The next assistive application to communicate with gaze movements is developed by using *Tobii EyeX* eye tracker hardware (Pal et al., 2017). The target of the application is to increase the quality of life of citizens with motor disabilities.

In summary, various eye-tracking applications have been developed using different eye trackers. The applications have different user interfaces for users to state their needs. In the present study, we present two different interfaces developed whose are images of daily needs and on-screen keyboard to type needs.

3 Methodology

3.1 Web-cam Application: Cascade Window Application

The Cascade Window Application, which was developed by the purpose of the present study, tracks the eye movements of the user by processing the video frames recorded by the computer webcam. In the application, a window containing an eye is obtained by the pupil detection algorithm (Lupu et al., 2013; Nguyen et al., 2015; Ciesla & Koziol, 2012). By analyzing the eye movements, the program interprets the user's choice on the screen.

The program starts with pupil detection and then continues with the analysis of the location of the pupil. The pupil detection starts with webcam initialization. After an image is acquired from the webcam, it is converted to the grayscale image. Then Haar Cascade Face filter and Haar Cascade Eye Filter algorithms are applied to the the grayscale image. If an eye image is detected after the algorithms, the blob detector algorithm is applied to the eye image, which result with pupil data. The pupil data and eye image are analyzed together in order to detect the location of the pupil. The location of the pupil on the horizontal axes is used to decide where the user is gazing at.

The working principle of the application is that the application window shows the user sixteen images and detects which picture the user is looking at. The application consists of two sets of roles: An administrator (selecting images on GUI, starting calibration steps) and a user (selecting a picture with eyes, approving the decision).

There are two calibration steps through the program's operation. In the first calibration step, the threshold value for the pupil detection is set. The next step is to calibrate the computer screen for the user. When the calibration process is completed, the program is ready to operate for the user. The user is expected to look left or right side of the screen according to the need.

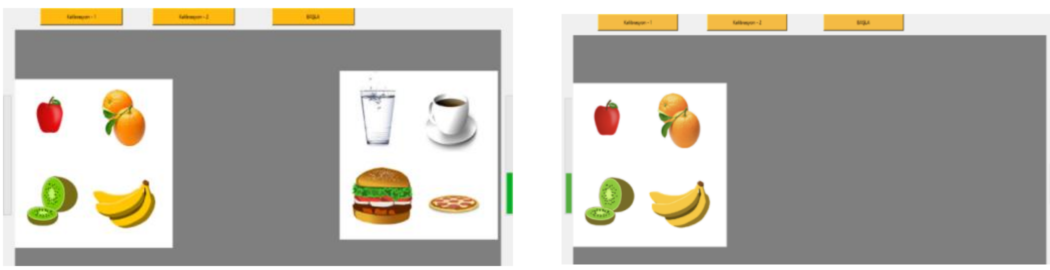

Figure 1: Screenshots from the Cascade Window application. The user selects the left image (left) and the selected image is highlighted on the display (right).

After displaying the picture, the program waits for user response. On the user side, the approval of the selected image was indicated by closing the eyelids. When the user closes the eyes for two seconds program makes the interpretation that the selection is approved by the user. Otherwise, the program waits for user selection.

According to the gaze data, the program decides which side of the screen the user is looking at and it operates according to the result. However, the major challenge with the Cascade Window application was that the user selection pool was limited to 16 images. Although the program could be modified for a larger set of images, a keyboard layout provides better flexibility for the user. Therefore, applications with different keyboards were developed in this study. Due to the requirements for higher spatial resolution for detecting eye gaze, a feasible equipment that would provide accessibility to the user is an eye tracker. The next section describes those keyboard applications that were used with an eye tracker.

3.2 Eye Tracker Applications

In particular, three applications were developed using open-source eye tracker hardware, viz. Pupil Core Headset. The Pupil Core is an open platform used by a global community of researchers (Pupil Labs, 2019). The Pupil Core headset operates by defining an area of interest containing the AprilTag (Sagitov et al., 2017).) marker. The area of interest contains either a keyboard or images in the applications. After the spot is detected, where the eye is gazing at, the applications show where the user is gazing at. One of the applications has a QWERTY keyboard interface for the user. The other application has a Alphabetical-keyboard interface. The last application's interface is the same as Cascade Window Application. The Cascade Window was modified such that the application works with the eye-tracker hardware.

The keyboard applications start with clicking the starting button. When the user triggers the program, the program starts capturing gaze data. The program collects 400 gaze data satisfying that the data confidence value must be greater than 0.80, and in the region of AOI. Then, the application detects the letter which the user is looking. This process last three seconds satisfying that the calibration step completed successfully. The pink dots on the interface show gaze data.

Figure 2: Virtual QWERTY keyboard (left) and virtual alphabetical keyboard (right).

Besides the QWERTY keyboard, an application with an alphabetical keyboard has been developed. The working principle of the application is the same as the QWERTY Keyboard Application. The only difference is the ordering of the letters on the interface. The letters are in alphabetical order. In addition to the keyboard applications, a cascade window application has been adopted to the eye tracking device, too. The application is the same interface with the Cascade Window Application, but the Pupil Core eye-tracker (www.pupil-labs.com) was used to track eye movements. The next step is to evaluate the usability and performance of these applications. For this purpose, the applications were tested by four participants.

4 Results

In this study, a comparative usability test has been performed for four applications in order to evaluate and compare the usability of the applications. The present study reports a usability investigation of a small sample of four participants. The participants were expected to use all four applications twice in random order. The tasks were the same for all the participants: *asking a glass of water* and *calling the nurse*. Accordingly, the user was expected to choose the water image in the Cascade Window applications, whereas being expected to write rereferring words or phrases related to the task in the keyboard applications. Four participants (two females, mean age 25.8, SD =3.5) with no motor disabilities tested the applications. The applications were used by the participants in random order. In addition to the usability analysis, a performance analysis is evaluated by a set of dependent variables: The *success rate*, the *operation time*, the *number of errors*, and the number of letters typed in in the keyboard applications. Each participant answered the System Usability Scale, i.e., SUS (Brooke, 1996) questionnaire for each application.

The results showed that participants accomplished all the tasks in all the applications. The SUS scores showed that were similar across the Cascade Window application with the webcam eye tracker (M=79.3, SD=7.4), the QWERTY keyboard application (M=80.0, SD=16.2), and the Cascade Windows application the wearable eye tracker (M=84.4, SD=21.6), which were all higher than the SUS score of the alphabetical keyboard application (M=66.2, SD=16.0). The results for functionality are shown in Table 1.

Table 1: Task completion times in seconds (The parentheses show standard deviations. EyeTr.= Eye Tracker, QWER=QWERTY Keyboard, ABC=Alphabetical Keyboard).

	Cascade Window Applications				Keyboard Applications			
	Webcam 1st task	Webcam 2nd task	EyeTr. 1st task	EyeTr. 2nd task	QWER 1st task	QWER 2nd task	ABC 1st task	ABC 2nd task
Task Completion Time	163.8 (70.1)	109.8 (10.9)	81.8 (6.72)	66.5 (15.1)	82.8 (17.5)	146.8 (30.0)	151.8 (32.4)	219.3 (18.6)

According to the results, the Cascade Window Eye Tracker Application is better than Cascade Window Webcam Application in terms of functionality. Moreover, QWERTY Keyboard is better than Alphabetical Keyboard in terms of functionality.

After tests were completed, every participant answered the standard usability questionnaires, System Usability Scale, i.e. SUS (Brooke, 1996) for each application. The results for usability are shown in Table 2.

Table 2: SUS results (The parentheses show standard deviations. EyeTr.= Eye Tracker, QWER=QWERTY Keyboard, ABC=Alphabetical Keyboard).

	Cascade Window Applications		Keyboard Applications	
	Webcam	EyeTr.	QWER	ABC
SUS Scores	79.3 (7.40)	84.4 (21.6)	80 (16.2)	66.2 (16)

According to the results, the usability score of the Alphabetical Keyboard Application is relatively low, though with high variations among the individual scores. Moreover, the Cascade Window and Eye Tracker QWERTY Keyboard Applications' SUS scores are around 80, meaning that the applications are usable.

In addition to the performance analysis, we observed the expertise of the participants during the tasks. The participants feel comfortable when using Cascade Window Application, since they have familiar with computer webcam and setup process is simple. However, on the eye tracker applications, some of the participants faced with difficulties. One of the participant's calibration is lost while typing on QWERTY Keyboard interface. Even though they face with difficulties, the participants stated that communication with eye movements was a good experience.

5 Conclusion and Future Work

Our findings reveal that both webcam eye trackers and wearable eye trackers may be appropriate candidates for gaze-based communication. It indicates that low-cost, webcam-based solutions have potential to be employed as methods of gaze-based interaction, as long as enough space is allocated for correct discrimination of gaze on specific Areas of Interest (AOIs) on the screen. The findings also show the advantage of using habitual keyboards instead of alphabetical ones in gaze-based interfaces. Nevertheless, the results of the study have limited generalizability since the tests were conducted with non-disabled participants at the prototype development stage. The future research should expand the sample size to allow statistical comparisons, also to fine tune the BCI approach prior to examining the interface on motor-impaired individuals, which is eventually the target population for application. In particular, Future research should address disabled citizens as participants, after the development of the prototype into an end-user system. We plan to provide the end-user system as an open source tool.

6 References

[1] ALS Association (2021, July). Retrieved August 08, 2021, from https://www.als.org/understanding-als.

[2] ALS, Amyotrophic Lateral Sclerosis, Lou Gehrig's disease (2019, June), Retrieved August 08, 2021, from https://www.hopkinsmedicine.org/neurology_neurosurgery/centers_clinics/als/conditions/als_amyotrophic_lateral_sclerosis.html

[3] Arthur, K. C., Calvo, A., Price, T. R., Geiger, J. T., Chio, A., & Traynor, B. J. (2016). Projected increase in amyotrophic lateral sclerosis from 2015 to 2040. *Nature communications*, *7*(1), 1-6.

[4] Amyotrophic Lateral Sclerosis (ALS) Fact Sheet. (2019, August 13). Retrieved October 17, 2019, from https://www.ninds.nih.gov/Disorders/Patient-Caregiver-Education/Fact-Sheets/Amyotrophic-Lateral-Sclerosis-ALS-Fact-Sheet.

[5] Salem, C., & Zhai, S. (1997, March). An isometric tongue pointing device. In *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems* (pp. 538-539).

[6] Jacob, R. J. (1990, March). 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* (pp. 11-18).

[7] Barreto, A. B., Scargle, S. D., & Adjouadi, M. (1999). A real-time assistive computer interface for users with motor disabilities. *ACM SIGCAPH Computers and the Physically Handicapped*, (64), 6-16.

[8] Barreto, A., Scargle, S., & Adjouadi, M. (2000). A practical EMG-based human-computer interface for users with motor disabilities.

[9] Brooke, J. (1996). SUS-A quick and dirty usability scale. Usability evaluation in industry, 189(194), 4-7.

[10] Wanluk, N., Visitsattapongse, S., Juhong, A., & Pintavirooj, C. (2016, December). Smart wheelchair based on eye tracking. In *2016 9th Biomedical Engineering International Conference (BMEiCON)* (pp. 1-4). IEEE.

[11] Liu, S. S., Rawicz, A., Ma, T., Zhang, C., Lin, K., Rezaei, S., & Wu, E. (2010). An eye-gaze tracking and human computer interface system for people with ALS and other locked-in diseases. *CMBES Proceedings*, *33*.

[12] Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., ... & Chellappa, R. (2019). A fast and accurate system for face detection, identification, and verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, *1*(2), 82-96.

[13] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. *International journal of computer vision*, *57*(2), 137-154.

[14] OpenCV. (2019, December 23). Retrieved from https://opencv.org/.

[15] Nguyen, T. P., Chew, M. T., & Demidenko, S. (2015, February). Eye tracking system to detect driver drowsiness. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)* (pp. 472-477). IEEE.

[16] Li, X., Han, G., Ma, G., & Chen, Y. (2011, November). A new method for detecting fatigue driving with camera based on OpenCV. In *2011 International Conference on Wireless Communications and Signal Processing (WCSP)* (pp. 1-5). IEEE.

[17] Lv, Y., Wang, S., & Shen, P. (2011, August). A real-time attitude recognition by eye-tracking. In *Proceedings of the Third International Conference on Internet Multimedia Computing and Service* (pp. 170-173).

[18] Lupu, R. G., Ungureanu, F., & Siriteanu, V. (2013, November). Eye tracking mouse for human computer interaction. In *2013 E-Health and Bioengineering Conference (EHB)* (pp. 1-4). IEEE.

[19] Poole, A., & Ball, L. J. (2006). Eye tracking in HCI and usability research. In Encyclopedia of human computer interaction (pp. 211-219). IGI Global.Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, *189*(194), 4-7.

[20] Pupil Labs. (2019). Retrieved November 20, 2019, from https://pupil-labs.com/.

[21] Khasnobish, A., Gavas, R., Chatterjee, D., Raj, V., & Naitam, S. (2017, March). EyeAssist: A communication aid through gaze tracking for patients with neuro-motor disabilities. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)* (pp. 382-387). IEEE.

[22] Pal, S., Mangal, N. K., & Khosla, A. (2017, March). Development of assistive application for patients with communication disability. In *2017 international conference on innovations in green energy and healthcare technologies (IGEHT)* (pp. 1-4). IEEE.
[23] Ciesla, M., & Koziol, P. (2012). Eye pupil location using web-cam. *arXiv preprint arXiv:1202.6517*.
[24] Sagitov, A., Shabalina, K., Sabirova, L., Li, H., & Magid, E. (2017, July). ARTag, AprilTag and CALTag Fiducial Marker Systems: Comparison in a Presence of Partial Marker Occlusion and Rotation. In ICINCO (2) (pp. 182-191).