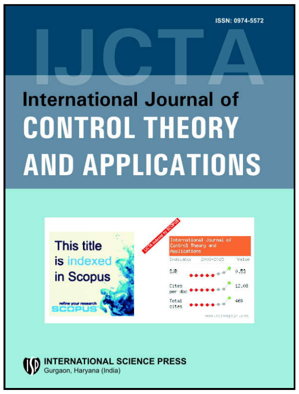

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 15 • 2017

Helping Hands–An Eye Tracking enabled User Interface Framework for Amyotrophic Lateral Sclerosis Patients

Tushar Sawhney[1], Sereddy Pravin Reddy[1], Amudha J.[1] and C. Jyotsna[1]

1 Dept. of Computer Science &Engineering, Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, Amrita University, India, Emails: tushars94@yahoo.com,s.pravinreddy30@gmail.com,j_amudha@blr.amrita.edu, c_jyotsna@blr. amrita.edu

Abstract: Amyotrophic Lateral Sclerosis weakens the nervous system of a person due to death of the neurons that are responsible for the muscular activities of our body, allowing him to move and communicate with the help of his eye gaze and eye movements. This inactivity leads to a person being in a completely paralyzed state slowly making him succumb to his condition. In a vast majority of cases of ALS, the cause is still unknown. However, some studies claim that the involvement of multiple genes and environmental factors contribute to ALS in various cases. Indeed, ALS is primarily a polygenic (multiple genes involved) disease (70%–90) is inherited in 30% of total ALS cases. Some other causes include cigarette smoking, viral infections and ingestion of non-protein amino acids that may engender to this disease. The initial symptoms of ALS involves muscle weakness and/or muscle atrophy. Other presenting symptoms include trouble swallowing or breathing, cramping, or stiffness of affected muscles. However these symptoms prove to be too subtle to be detected and are often overlooked. This state of being makes it impossible for the patient to communicate his basic needs to others. Eye tracking is the advanced techniques which enable the people with Amyotrophic lateral Sclerosis (ALS) and other locked-in diseases to communicate with normal people and make their life easier. The framework developed proves to be user-friendly and customizable according to the needs of the patient.

Keywords: Eye-gaze, Eye-tracking, Human Computer Interface, Amyotrophic-Lateral Sclerosis

1. INTRODUCTION

Eyes are the best source for communication for people who are unable to communicate using the conventional means. Gaze is the direction where the person looks. Tracking the gaze of a person's eye using several eye-gazing techniques have gained popularity among several renowned researchers around the globe for their usual application in human machine interfaces, monitoring driver vigilance, and identifying website key objects. This technique of eye gaze tracking and capturing can be incorporated in several commercial purposes too.

This paper describes the assistive application developed for ALS patients which helps them to convey their basic needs through HCI (Human Computer Interaction)[1][6].The patient suffering from Amyotrophic Lateral Sclerosis does not allow him to communicate with his peers using any kind of physical means. This state of being makes it impossible for the patient to communicate his basic needs to others.

Unfortunately, existing systems prove to be too expensive. Cost is not the only factor behind developing this proposed system, but to make it a technology that can be used for niche applications. Hence, the proposed system helps him, which consists of a basic GUI can be used as a framework, where the patient can convey his needs by gazing at different images, which prove to be a less stressful task for a patient, rather than typing the message again by his eye movements .

The advantage of the proposed system is that the patient can use it as a framework, and can modify the GUI with the images of his own needs. It is a framework programmed in python using methods from the existing libraries, which can be customized by including more libraries and their methods, and also incorporating multiple layouts.

2. BACKGROUND AND RELATED WORK

Various augmented and alternative methods have been proposed and tested on the ALS patients to track their needs and behavior. These methods may include gestures, eye blinks or communication boards with letters or symbols.

In its more advanced state, electronic communication devices or computers allow the user to have voice output, send email, and surf the web.

An inexpensive, dependable and mobile communication support system for the disabled is developed using Electro- Oculography. Four directional eye movements and blink are tracked using 5 electrode EOG configuration for communication using a visual board for reference. The board was specifically designed based on modeling of English language to increase the speed by reducing the number of eye movements required. Using these system users has achieved an average speed of 16.2 letters/min [4][6].Chart based encoding is another technique which have and index of codes and their corresponding phrase or request. Peers and family create a chart of the most often patient requests and create an established system for the patient to indicate yes—whether an eye blink, grunt, small thumb movement, etc. The chart is placed on the wall of the room and if the patient has something to say, caregiver asks "is it a number? ". If yes, caregiver recites number until the patient gives the signal and now caregiver knows what patient wants[4].

Another Hi-Tech solution includes an Electronic Eye Gaze which has an attachment attached to them that tracks the patient's eye movements. The patient needs to have functional eye movement in order to operate electronic eye gaze systems. Some of these systems are, however, very expensive and needs an assistance of an expert to operate it.

3. SYSTEM ARCHITECTURE

The following are the details of the application developed.

Language: Python

Operating system: Linux

Primary Libraries Used: Tkinter, OS, Sys

Software Development Kit: Eye-Tribe SDK

Eye tracking device: Eye-Tribe Eye Tracker.

3.1. Approach

The approach involves developing a Python programmed Graphic User Interface framework which can be controlled by an eye-tracking device. The eye-tracking device used in the proposed system is Eye-Tribe eye-tracker. The GUI consists of images in the form of grids, which are included according to the need of the patient[2]. For each grid on the layout, the coordinates of the grid have been recorded in a separate file.

The calibration of the patient's eyes are done using Eye-Tribe SDK, which allows the eye tracker to have a knowledge about various parameters of the patient's eyes such as the speed of the movement, the distance your eyes can move over the screen and the accuracy of the focus[3].

Once the calibration is completed, the patient is allowed to control the cursor movements with the help of his eyes. The movement of the cursor over the grids on the interface helps him to describe his needs to his peers, as the interface has been designed in a way which dynamically changes itself, once the user hovers over the grid whose coordinates have been initially recorded in the separate file.

To run the eye-tracker smoothly on the machine, it should be connected to a USB 2.0 port. This eye-tracker has a database use to various parameters which can be parsed and used to derive various observations. It also provides a unique feature of controlling the cursor movements with the help of eye-movements.

The system constitutes of a client server model. In the proposed system have 2 clients, where one acts as the display system and other as the monitor system.

The server sends request to client system to establish a connection. In order to perform a successful connection, the display system has to perform a calibration. If the calibration is performed successfully, the application is started. If the calibration result is negative, the server asks the display system to calibrate again.

3.2. Calibration

Calibration is performed on the client side on the request of server for the system to function in a smooth fashion. In this step, the patient's gaze is captured on the client side of the application and the results are displayed to the monitor system. The calibration can be a 1-point-9point calibration depending on the accuracy required by the system.

After the calibration is performed successful, the server is connected to both the clients.

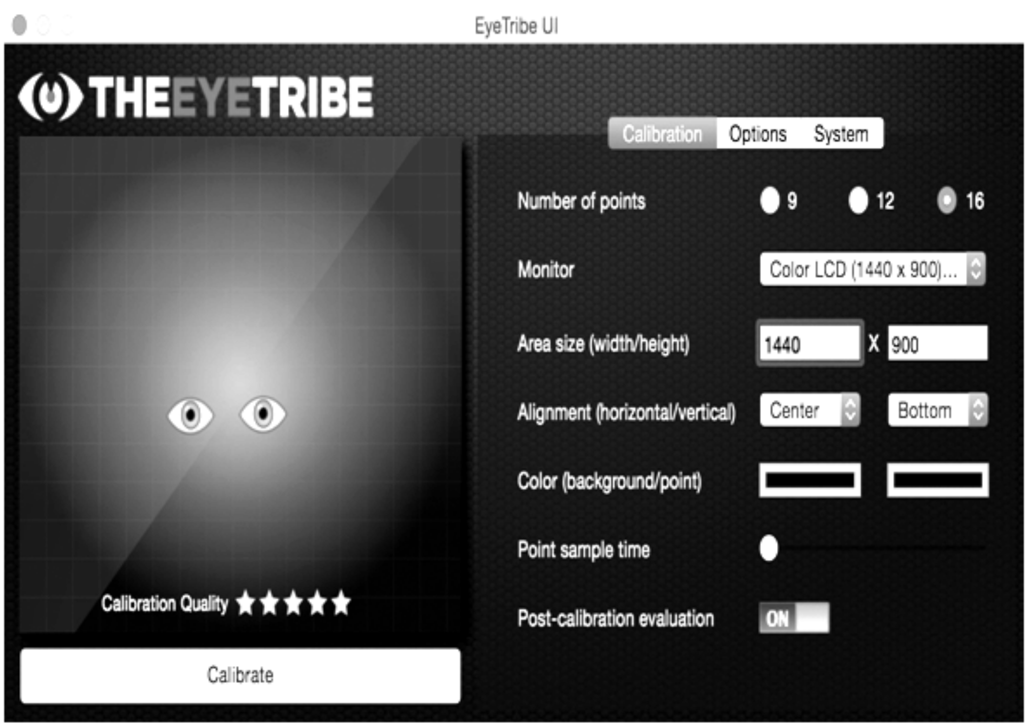

Figure.1. Calibration

4. SYSTEM IMPLEMENTATION

The application is divided into a 6-9 grid system where each grid constitutes a request which the display system has to send to the server. The request of the system is recorded according to the eye-gaze of the patient which is focused on a particular grid. If the request is captured successfully by the server, the response is sent to both to display system and the monitor system. Multiple requests can accepted from the display system and sent to the monitor system according to the number of grids each layout of the application contains.

4.1. Application Code

To capture the gaze

def motion(event):

x, y = event.x, event.y

The above code captures the co-ordinates of the points of the cursor when it hovers over the interface. Once captured, it's stored in the variables as the x and y coordinates. The event is a predefined function defined in the Tkinter library.

4.1.1. To display the layout of the application

root = Tk()

photo = PhotoImage(file="C:/Users/Desktop//image1.gif") label = Label(root,image=photo)

label.pack() root.bind('<Motion>', motion) root.mainloop()

Figure 2: Main Interface

In the above code, the layout in figure 2 is created initially using the tk() function and it's reference is stored in the root variable. The layout is initially a blank layout. In order to bind the custom interface over the blank interface the image is browsed from the user's machine and is bound over the blank layout.

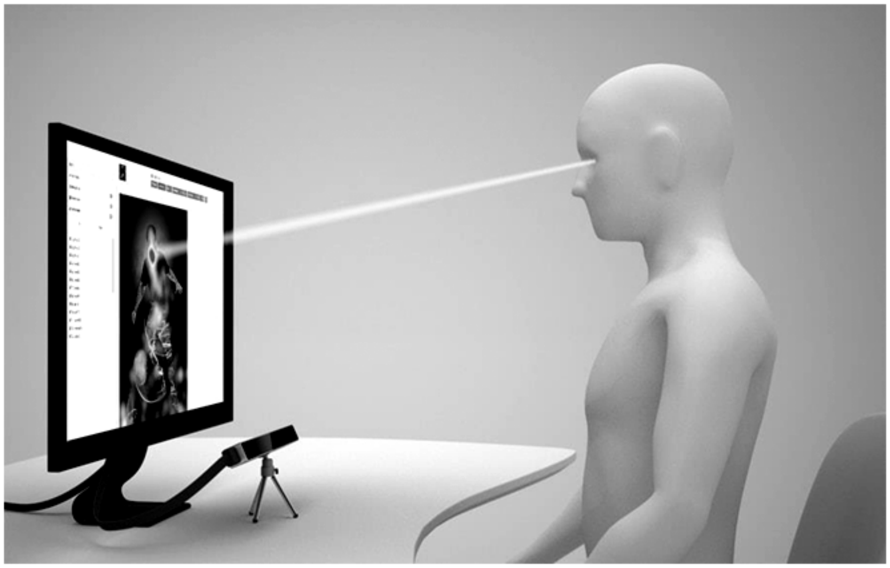

Figure 3: Eye gaze Captured. [7]

4.1.2. To dynamically change the layout

```
if((x1<x<x2 &&
 y1<y<y2))
change_layout();

def change_interface();
root.destroy();
root1=tk()

label =Label(root,
 image=PhotoImage(file="/Desktop/img1.gif))
root1.bind('<Motion>',motion)
```

Every image on the layout has a set of pre-defined co-ordinates. In the above code, if the coordinates of the cursor lies in the range of any of the image present in the layout, the 'if' condition is executed. According to the pre-defined co-ordinates of the images on the layout, the layout changes dynamically to the required layout, showing the result both on the monitor system and the display system.

The above procedure can be included for multiple layouts

Figure 4: Changed Layout

Figure 4 is derived dynamically from figure 1 after the patient focuses or gazes on the co-ordinates that specified water as one of his need. Figure 5 shows the procedure followed by the patient from the beginning till the end in the form of an activity diagram.

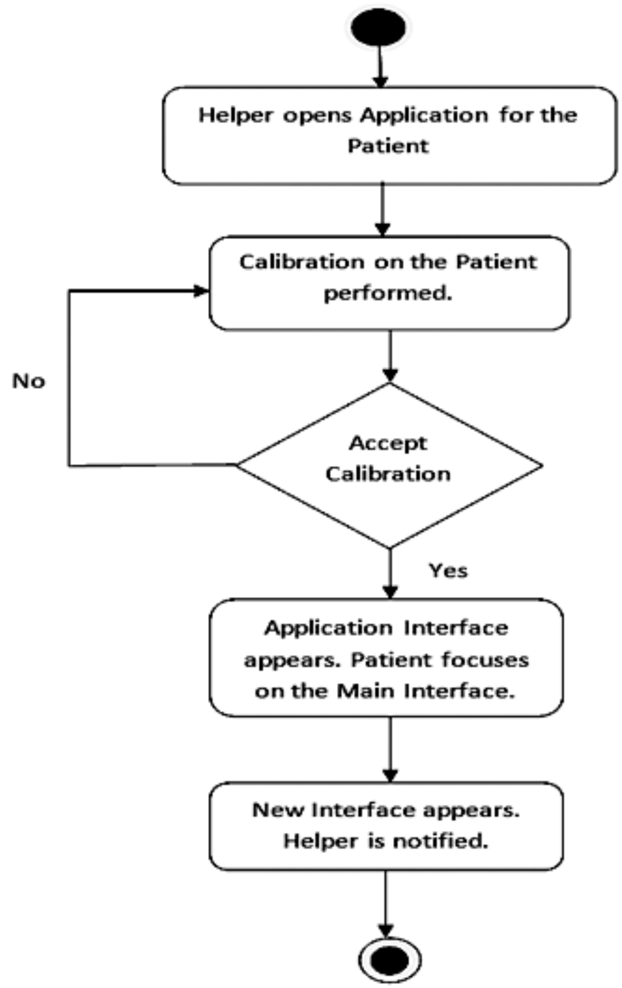

Figure 5: Activity Diagram

5. RESULT ANALYSIS

In the proposed system, the patient is made to select a particular need from the grid displayed in the Figure 2.The eye gaze of the patient is captured by the eye-tracker kept in front of the patient as shown in Figure 3.After the

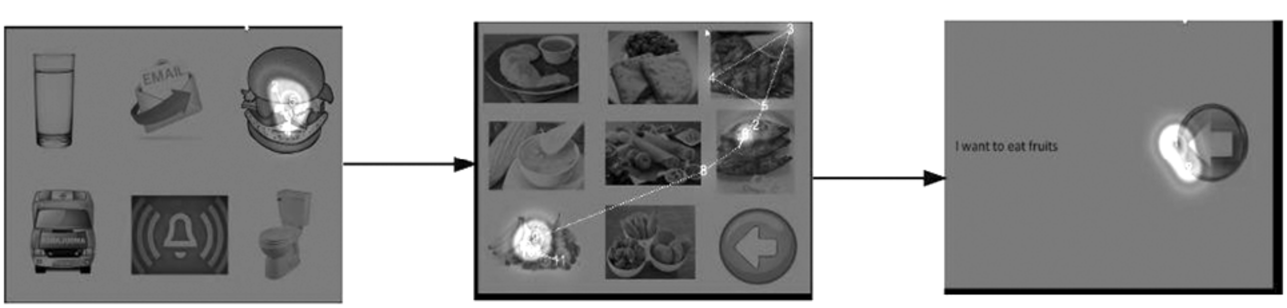

Figure 6: Visualizations

coordinates of the gaze are captured, are passed as the arguments in the python code. When the python code runs successfully, corresponding to the coordinates of the gaze, a new interface is created dynamically, alerting his peers about his needs

The patients who were tested upon did not have any trouble while focusing their eyes on the interface. Their eye movements were easily calibrated on the device. Once the patient was made to focus on the interface and asked to pick a suitable option for himself, there was a smooth transition noticed between different options that were provided to the patient, as can be seen by the visualization patterns in Figure 6. The visualization patterns used in Fig 6 are Heat Maps and Scan paths. In the first image, the patient focuses on one of the six options, which can be shown by the heat map. The higher the frequency of the heat map color, the specific becomes the area of his interest. After the interface changed, the patient was made to select a specific product from the list of given products under the chosen demand. The eye gaze patterns of the patient are depicted in the form of scan paths in image 2.Finally in the image 3, the heat map depicted again shows the area of interest of the patient on the interface. This system proves to be an effective and a healthy alternative for the existing systems. The programmer was able to modify the interface according to the needs of the patient. Fig.6 shows how the proposed framework can be extended to multiple levels to make the request by the patient more specific to his peer.

6. CONCLUSION AND FUTURE SCOPE

A person suffering from ALS leaves him no option of communicating with his peers for his basic needs other than with the help of eye movements and eye gaze. The existing methods for communication prove to be expensive as well as complex for some patients. They may assist patients with specific needs, but a multi-purpose framework has not been programmed which can cater to the general needs of the ALS patients. The proposed system proves to be a healthy alternative for the existing technologies and is provides accurate results to considerate amount. The patient can use the application with ease, as the application is user-friendly and does not strain his eyes either, when he stares at the interface. The ease of transition between various co-coordinates on the interface can be seen in Fig. 6. Advancements in the system can be made by incorporating a text to speech feature in the framework, in case the patients wants to contact with the monitor system if the vigilance towards the patient is not present[5].Another way of making the system more flexible is to connect it to a cloud. In the cloud, more requirements can be stored by the application developer, which can be downloaded by the patient and included in the list of his needs dynamically, in case he does not find a suitable option in case of emergency. Movies, songs, or books can also be pre-loaded in the patient's system, which can start executing when the respective options included in the interface are gazed upon by the patient. If the patient is sitting at a distance more than 100cm, the system fails to track the eyes. The system might also fail if the patient is wearing his spectacles as it causes an obstruction between the patient's eyes and the eye-tracker. Sometimes, if the calibration is not performed accurately, the person may feel the strain of calibrating several time until a standard degree of accuracy is obtained.

REFERENCES

[1] Jacob, Robert JK. "Eye movement-based human-computer interaction techniques: Toward non-command interfaces" Advances in human-computer interaction 4 (1993): 151-190.

[2] Amudha J, R. Ramya Sai, M. Vani Shree, T. Shravani, "Assistive Communication Application for ALS Patients", Proceedings of 2nd IEEE International Conference on Human Computer Interactions (ICHCI'16), March 2016.

[3] DivyaVenugopal, Amudha J, C. Jyotsna, "Developing An Application Using Eye Tracker ", IEEE International Conference on Recent Trends in Electronics Information, May 2016

[4] ALS Association Website, www.alsa.org

[5] Arai, Kohei, and Ronny Mardiyanto. "Eye-based human-computer interaction allowing phoning, reading e-book/e-comic/e-learning, Internet browsing and TV information extraction." International Journal of Advanced Computer Science and Applications 2.12 (2011).

[6] Pradeep, S. G., Akesh Govada, and Kendaganna Swamy. "Eye Controlled Human Machine Interface (e-VISION)." Eye 2.5 (2013).

[7] AbilityNet Website www.abilitynet.org.uk