App to Help Children with Special Needs to Improve Their Eye Movements and Focus

Christeena Varghese*
christeena.varghese@study.thws.de
Technical University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany

Vincent Wahyudi*
vincent.wahyudi@study.thws.de
Technical University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany

Viony Tengguna*
viony.tengguna@study.thws.de
Technical University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany

ABSTRACT

This paper presents a low-cost eye-tracking system and training game application aimed to improve eye gaze control of children aged three to six, especially for those with conditions such as Dyspraxia and Autism Spectrum Disorders. The traditional method of manually evaluating eye movement can often result in inaccurate results. Our application combines eye-tracking technology with a game-like approach designed to encourage children to focus their gaze in different directions. The application tracks their gaze using a laptop's built-in camera and provides comprehensive analytics. This work has the potential for further development and application in the field of eye gaze treatment.

KEYWORDS

Eye-tracking application, Autism Spectrum Disorder, children, Computer Vision, OpenCV

1 INTRODUCTION

Eye contact plays a crucial role in human social interaction. Individuals with conditions such as Dyspraxia and Autism Spectrum Disorders face difficulties in shifting their gaze, which leads to difficulty in synchronizing their central and peripheral vision [9]. Impaired eye movement can increase the risk of accidents, restlessness in movement, and increase stress. If detected in its early stages, the dexterity of the eye gaze can be improved through the implementation of therapeutic exercises, commonly referred to as "eye gaze treatments".

Traditionally, the therapist uses an item (a stick or a pen) for the eye gaze treatment sessions. The therapist moves the stick in various directions, encouraging the child to focus on the object. The therapist visually evaluates the movement of the child's eye and manually tracks the improvements. However, this method is extremely imprecise and subjective.

The inaccuracy of the results can be reduced by the use of eye-tracking systems, which are readily available for purchase and can accurately determine the position and movement of the gaze [10]. However, these systems can be costly and there are a limited number of affordable eye-tracking systems on the market.

Our eye-tracking software aims to improve traditional methods in a cost-effective way by providing more accurate results in eye gaze treatment. The source code of this project is available on GitHub at the following link: https://github.com/vincentw1997/Eye_Tracking_Project_Module.git. The eye-tracking application is covered up as an interactive game that can be appealing to children of ages three to six [11]. The game that is connected to the eye-tracking system mimics the traditional treatment in an attractive way that helps the children to improve their eye gaze in a joyful way. The therapist can evaluate each child's progress accurately from the results generated by the application.

The software tracks the child's gaze through the built-in laptop's camera and provides analytics about the estimated gaze for the therapist. The system records the coordinate data of the child's eye movement and compares it to the recorded coordinates of the moving characters in the game. These data are useful for therapists to assess the speed and accuracy of the eye gaze movement during the therapy session.

This paper is structured into four sections. The first section provides a concise overview of various eye-tracking methods and offers explanations of similar works related to cost-effective eye-tracking pipelines. Next, this paper describes the method used to develop the eye-tracking application, including elements such as head-tracking, game components, and data analytics. The third section presents the experimental setup, the results of the controlled experiments, and a discussion of the obtained results. Lastly, the fourth section provides a short conclusion and potential improvements for future works.

2 RELATED WORK

Generally, eye-tracking methods can be categorized into two main groups [4]: appearance-based and model-based approaches. The appearance-based approach tackles the problem by learning a mapping function from eye images to gaze estimation. This approach usually requires large training data and a properly trained model [13]. On the other hand, the model-based approach aims to calculate a 3D gaze direction vector, based on the known anatomy of the human eye, such as the iris, sclera, and pupil. This method usually relies on precise metric information such as camera calibration, positioning, monitor positioning, and orientation. In this paper, we implemented the model-based approach by simplifying assumptions and skipping some traditional steps in order to fit into our specific use case. Cazzato et al. [2] approaches eye-tracking problems for soft biometrics using the model-based approach that uses cheaper hardware such as Microsoft Kinect and ASUS Xtion Pro Live. These are commercial depth sensors, which are considerably more accessible to the general public compared to Tobii infrared eye trackers or similar laboratory hardware that can cost multiple times more. Our paper aims to leverage the concept of making eye-tracking technology more accessible to the general public, without

*Equal contribution

DOI: https://doi.org/10.26493/scores23.05

the need for additional hardware such as commercial depth sensors. However, this approach comes at the cost of lowered tracking accuracy.

3 METHOD

In this section, we divide the description of our application into four parts. The first description begins with an overview of the eye-tracking architecture. Secondly, we elaborate on the head-tracking architecture and its integration with the eye-tracking architecture. Thirdly, we explain the game component designed to captivate the child's attention. Lastly, we provide data analytics based on the data collected from the first three architectures. We visualize the data for performance analysis evaluated by the therapist. An overview of how the application functions can be seen in Figure 1. A flowchart of all the development processes can be seen in the Github repository.

3.1 Eye-Tracking

The eye-tracking architecture's main objective is to isolate the eye with respect to the whole screen captured by the webcam of the laptop. It starts with grayscaling [14] the frames captured to reduce color complexity. Facial landmarks are further detected from the grayscaled frames using dlib library [5] that assigns specific coordinates to the facial landmarks. Masking is executed to isolate the precise locations of the right and left eye coordinates derived from dlib landmarks [12], thereby producing frames that exclusively isolate the eye shape, which we call eye-only frames. The estimation of the pupils' position is achieved by detecting circular shapes in the eye-only frames.

General pipelines for shape detection are implemented to estimate the pupil's position in eye-only frames. These include grayscaling, filtering of the frames to emphasize contours of the iris and the sclera (white part of the eye), and thresholding of the frames where pixels are replaced into either black or white depending on the pixel value being higher or lower than the predetermined threshold value. The contours of the eye are clearly shown after the thresholding step and the circular shape of the iris will be clearly outlined.

The determined iris shape can be used to detect the region of interest (ROI). In the ROI frame, pupils are estimated by detecting the circular shape using Circular Hough Transform [8] inside the iris. The coordinates of the estimated iris and pupil are stored. Plotting the stored coordinates with respect to time will provide comparable data with the character movement in the game part. The ROI is implemented by magnifying the area on only one of the eyes as the movement of the left and right eye are assumed to be similar. The ROI frame can be seen in Figure 1. I. Eye Tracking (magnified version of the frame) with a green circle highlighting the iris of the child's eye.

3.2 Head-Tracking

The head-tracking architecture detects and follows the movement of a person's head, which includes the position and orientation of the head. The integration of head-tracking with eye-tracking ensures a controlled alignment between the camera and the child. Good alignment provides accurate readings of the estimated eye movements. Face detection, facial landmarks, and pose estimation are the main components of the head-tracking part. In this model, we use OpenCV [1] for camera live input, Dlib for the facial detector, and solvePnP [7] for pose estimation. A warning will show in the application if the patient is not aligned perfectly with the camera as seen in Figure 1. II. Head Tracking.

3.3 Game

The game architecture is created to serve as a facade for eye-tracking therapy. The game offers several settings such as cartoon character selections, the speed of the character's movement, and the direction. This promotes different movement variations for eye gaze training. The available direction options are horizontal, vertical, and free movement. The movement speed of the character can be set to slow, medium, or high.

The game was built using Python and libraries such as Pygame and Tkinter. SQLite was used as the backend database to store information such as game progress, patient details, and login credentials. Therapists can register patients, view patient records, start therapy sessions, and create an account using the "Create Account" option before logging in. These features provide secured information storage that the therapy can use.

3.4 Data Analytics

Data analytics can be done once eye-tracking, head-tracking and the game architecture are combined together into one integrated application. Data analytics analyzes the eye gaze performance of the child. Eye gaze data are only collected as long as the child maintains good head orientation during the session. Alerts in the application will remind the user to fix the user's head position and orientation as explained in the previous section.

The gathered data are represented in the form of two graphs [15] for each therapy session. The first graph contains coordinates that represent the pupil gaze positions in the ROI frame. It is plotted with respect to time and shows all the positions of the pupil from the start of the session until the session ends. This graph can be seen in Figure 2. (Pupil position in the ROI). The second graph contains the comparison of the cartoon character game coordinates and the pupil gaze coordinates in the ROI frame. Matching these coordinate systems with time will show whether the child is following the movement of the cartoon character or not. If the child is looking at something else besides the cartoon character, there will be a big difference between the two coordinates. An example of the graph can be seen in Figure 2. (Comparison between Game and Eye Coordinates).

4 EXPERIMENTAL SETUP, RESULTS AND DISCUSSION

In this section, we will explain the experimental setup, results, and discussion. The experimental setup describes the implemented settings. The result shows the data collected when implementing the settings in the experimental setup and discussion on the results obtained.

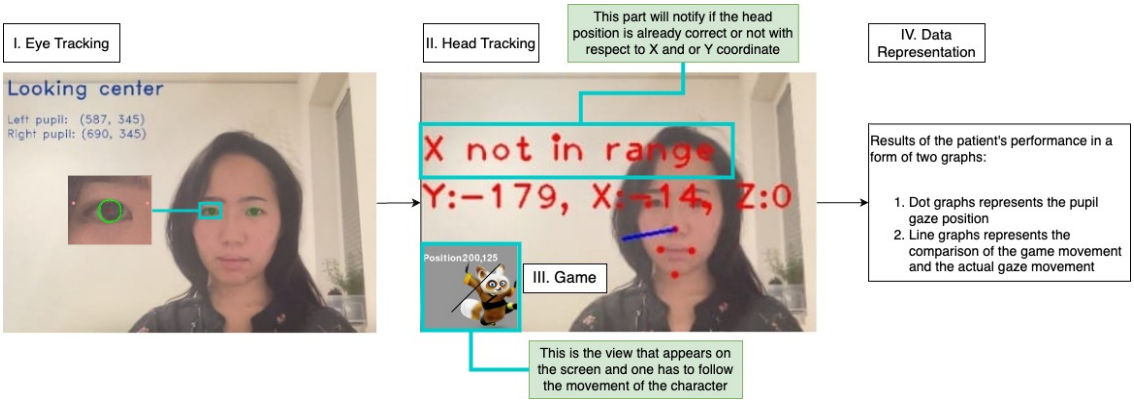

Figure 1: An Overview of the eye-tracking application. I. Eye Tracking shows the Region of Interest (ROI) frame that isolates the eye and highlights the iris as a green circle. II. Head Tracking shows the head position and orientation of the user in red. "Good" means the user is in the perfect position. "X/Y not in range means that the user needs to adjust his/her positioning. The small rectangle on the bottom left (III. Game) is the view that the user will see when the application runs. The user must follow the cartoon character's movement while remaining still. IV. Data representation represents the recorded data into graphs for analytics purposes."

4.1 Experimental Setup

The application is ideally designed for home usage and we simulate similar conditions during our experiment. The application is running on a regular laptop with a built-in webcam with a resolution of 720p. The room was well-lit with one warm white light in the ceiling and a window overlooking the side of the laptop. Experiments were done with no direct sunlight shining into the room. The tester is required to remove glasses during the experiment as it adds noise to the data.

The application was run on 5 different settings. The first and second setups run at medium speeds with different movement, which is horizontal and vertical respectively. The third and fourth setups have the same horizontal movement with different speed settings, low speed, and high speed respectively. The fifth setup will have the cartoon character moving at medium speed with a free movement function. These setups aim to show the effect of different parameters on a common use case.

The tester must be perpendicular to the camera, remain still, and maintain his left eye in between the pink diamonds in the ROI frame during the testing session. If the tester is maintaining an ideal position, the application display will show "Good!" in red color. Otherwise, the application will show warnings, which the tester must adjust their position similar to Figure 2. II. Head Tracking. X not in range means the tester is not in the right orientation horizontally.

4.2 Result

Figure 2. shows the recorded result for the first experimental setup (Horizontal movement and medium speed). The first graph in Figure 2. represents the estimated pupil position in the ROI frame. Both axes are in pixel units. The time recorded for the estimated pupil position is determined by the color. Darker colors represent earlier time stamps in the session and color lightens as time increases. The second graph in Figure 2. shows the movement data of the cartoon character (Game X and Y coordinate) and the eye movement in the ROI frame (Eye X and Y coordinate).

Figure 3. shows the recorded result for the second experimental setup (Vertical movement and medium speed). The two graphs in Figure 3. follows the same format as explained in the previous paragraph. The second setup maintains the same speed parameter while changing the cartoon character's movement from horizontal to vertical to test the tracking capabilities of the application. Results from the third to the fifth experimental setup are not shown here and can be seen in the GitHub repository.

4.3 Discussion

Figure 2. and Figure 3. demonstrates the effect of different movement directions. The pupil position in the ROI graph in Figure 2. shows the initial dots of the estimated pupil positions start moving from the bottom left corner of the screen which suggests the tester is trying to find the cartoon character on the screen. In the following time stamps, the positions moved toward the center of the ROI frames and oscillated horizontally between 250 to 300 pixels X position. These movements indicate that the tester is trying to follow the movement of the cartoon character that is moving horizontally. The horizontal movement of the pupil is further supported by the result of the Eye X coordinate in the second graph of Figure 2. (green line). The Eye X coordinate mimics the oscillating movement of the Game X coordinate (blue line) and has similar peaks. As this setup only evaluates the horizontal movement, the Game Y coordinate (orange line) remains in place for the whole session. The Eye Y coordinate (red line) will never be as constant as the Game Y coordinate (orange line) as humans have micro-movement in their gaze when tracking any moving object. The first graph in Figure 3. shows similar behavior in the first few dots. It started to move from the bottom left towards the center of the ROI frame which suggests the tester trying to track the cartoon character on the screen. Subsequent movements suggest that the estimated pupil

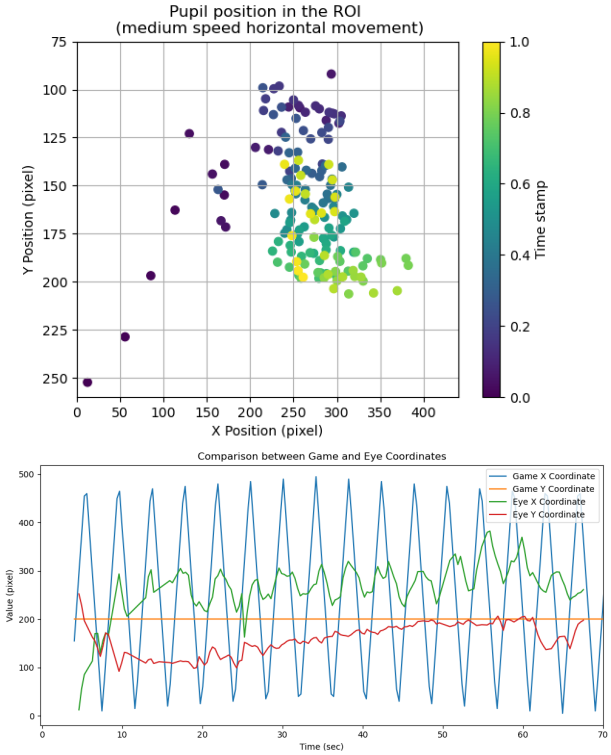

Figure 2: Results for the first experimental setup. Parameter settings for this setup are horizontal movement and medium speed. The first graph represents the tracked estimated pupil position in the ROI frame. The axes are in pixels. Darker points mean an earlier time stamp. The tracked pupil positions are more horizontally spread, and oscillations from left to right can be observed between 250 to 300 pixels X Position. The second graph illustrates the movement of the cartoon characters in the game as Game X/Y coordinates and the tracked eye movement as Eye X/Y coordinates. The Eye X coordinate (green line) mimics the pattern of the Game X coordinate (blue line) when the user tracks the horizontally moving character.

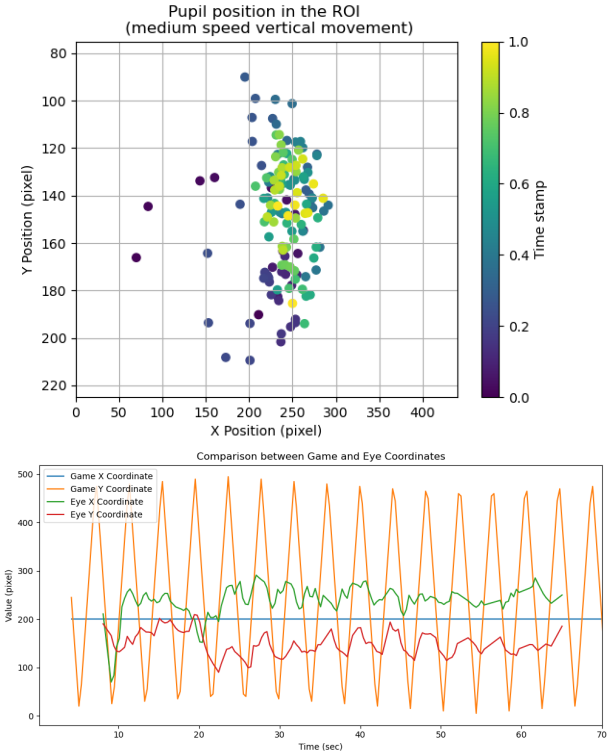

Figure 3: Results of the second experimental setup. Parameter settings for this setup are vertical movement and medium speed. The representation is the same as the previous Figure 2. The estimated pupil positions in the ROI are more vertically spread between 100 to 200 Y pixels Position. The estimated pupil positions generally remain centered on the X-axis. The second graph shows the Eye Y coordinate (red line) mimics the Game Y coordinate (orange line). The movements on the vertical plane are not as defined as the movements on the horizontal plane as in Figure 2.

position also centers around 200 to 250 pixels X position. Due to the vertical movement of the cartoon character, the estimated pupil positions oscillate vertically between 100 and 200 Y pixels position. This is further reinforced by the result in the second graph in Figure 3. Eye Y coordinate (red line). It mimics the movement of the Game Y coordinates (orange line) and has similar peaks. However, these peaks and nearly constant values of the Eye coordinates are not as obvious as in the previous horizontal setup. This shows the weakness in the application in capturing the estimated vertical movement of the eye. Another point to address is the difference between the Eye X coordinate and the Game X coordinate very far apart. This is due to the difference in the coordinate values during the development and the limitation of the application in capturing the extreme values on the edge of the ROI frames. This means that the Eye coordinates will have less range compared to the Game coordinates.

5 CONCLUSION AND FUTURE WORKS

This paper develops an affordable game-like eye-tracking application that does not need additional hardware. The application provides analytics for the therapist to track the performance of the child. Based on the experimental results, the application performs better in tracking horizontal movement compared to vertical movement. Other results testing the effect of speed suggest that the application performs best in a medium-speed environment. The free movement setup shows the limitation of the application in capturing extreme coordinates in the ROI frames. Based on the discussion section, the application still has a lot of sectors to improve. Children with glasses cannot use this application due to the effect of glass on the recorded video. Gwon et al. [3] use additional hardware to solve this limitation. Differences in the Game and Eye coordinate values can be tackled by building both coordinates of

the same magnitude making it easier to compare both values. The application has limited controls (no pause or run time settings) during the data recording session which made it hard to operate the application. The implemented approach in this paper is a naive approach that has its limitations in the accuracy and usability sector. Rebuilding the whole architecture using the appearance-based approach [6] requires a larger pre-trained model with a larger dataset might be a better solution that can improve the tracking accuracy of the application.

6 ACKNOWLEDGEMENTS

The authors express great appreciation to Prof. Dr. Magda Gregorová for her constructive suggestions during the experiments and writing of this paper.

REFERENCES

[1] G. Bradski. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[2] Dario Cazzato, Andrea Evangelista, Marco Leo, Pierluigi Carcagnì, and Cosimo Distante. 2015. A low-cost and calibration-free gaze estimator for soft biometrics: An explorative study. *Pattern Recognition Letters* 82 (11 2015). https://doi.org/10.1016/j.patrec.2015.10.015

[3] Su Gwon, Chul Cho, Eui Chul Lee, Won Lee, and Kang Park. 2014. Gaze Tracking System for User Wearing Glasses. *Sensors (Basel, Switzerland)* 14 (02 2014), 2110–34. https://doi.org/10.3390/s140202110

[4] Dan Witzner Hansen and Qiang Ji. 2010. In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 3 (2010), 478–500. https://doi.org/10.1109/TPAMI.2009.30

[5] Davis E. King. 2009. Dlib-ml: A Machine Learning Toolkit. *Journal of Machine Learning Research* 10 (2009), 1755–1758.

[6] Kyle Krafka, Aditya Khosla, Petr Kellnhofer, Harini Kannan, Suchendra Bhandarkar, Wojciech Matusik, and Antonio Torralba. 2016. Eye Tracking for Everyone. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] Eric Marchand, Hideaki Uchiyama, and Fabien Spindler. 2016. Pose Estimation for Augmented Reality: A Hands-On Survey. *IEEE Transactions on Visualization and Computer Graphics* 22, 12 (Dec. 2016), 2633 – 2651. https://doi.org/10.1109/TVCG.2015.2513408

[8] Jiri Matas, C. Galambos, and J. Kittler. 2000. Robust Detection of Lines Using the Progressive Probabilistic Hough Transform. *Computer Vision and Image Understanding* 78 (04 2000), 119–137. https://doi.org/10.1006/cviu.1999.0831

[9] Michael Miller, Leanne Chukoskie, Marla Zinni, Jeanne Townsend, and Doris Trauner. 2014. Dyspraxia, motor function and visual–motor integration in autism. *Behavioural brain research* 269 (2014), 95–102.

[10] Pramodini A. Punde, Mukti E. Jadhav, and Ramesh R. Manza. 2017. A study of eye tracking technology and its applications. In *2017 1st International Conference on Intelligent Systems and Information Management (ICISIM)*. 86–90. https://doi.org/10.1109/ICISIM.2017.8122153

[11] Tony Renshaw, Richard Stevens, and Paul Denton. 2009. Towards understanding engagement in games: An eye-tracking study. *On the Horizon* 17 (09 2009), 408–420. https://doi.org/10.1108/10748120910998425

[12] Christos Sagonas, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 2013. 300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge. 397–403. https://doi.org/10.1109/ICCVW.2013.59

[13] Yusuke Sugano, Yasuyuki Matsushita, and Yoichi Sato. 2014. Learning-by-Synthesis for Appearance-Based 3D Gaze Estimation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 1821–1828. https://doi.org/10.1109/CVPR.2014.235

[14] C. Tomasi and R. Manduchi. 1998. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. 839–846. https://doi.org/10.1109/ICCV.1998.710815

[15] Antony Unwin. 2020. Why Is Data Visualization Important? What Is Important in Data Visualization? *Harvard Data Science Review* 2, 1 (jan 31 2020). https://hdsr.mitpress.mit.edu/pub/zok97i7p.