Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 201 (2022) 166–173

Procedia Computer Science

www.elsevier.com/locate/procedia

The 13th International Conference on Ambient Systems, Networks and Technologies (ANT)
March 22 - 25, 2022, Porto, Portugal

EyeBox: A Toolbox based on Python3 for Eye Movement Analysis

Ling Zhang[a], Xiao Liu[a], Qian Chen[a], Yun Zhou[a,*], Tao Xu[b]

[a] Faculty of Education, Shaanxi Normal University, 199 South Chang'an Road, Xi'an, 710062, P.R.China
[b] School of Software, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an, 710072, P.R.China

Abstract

Eye tracking technology can reflect human attention and cognition, widely used as a research tool. To analyze eye movement data, users need to determine a specific area known as areas of interests (AOIs). Although existing tools offer dynamic AOIs functions to process visual behavior on moving stimuli, they may ask users to use markers to specify contours of moving stimuli in the physical environment or define AOIs manually on screen. This paper proposes a toolbox named Eyebox to 1) recognize dynamic AOIs automatically based on SIFT and extract eye movement indicators, as well as 2) draw fixations. We also design a user-friendly interface for this toolbox. Eyebox currently supports processing data recorded from the Pupil Core device. We compared results processed by manual with by Eyebox in a custom eye-tracking dataset to evaluate this toolbox. The accuracy of 3/4 data for AOI1 is above 90%, and the accuracy of 4/5 data for AOI2 is higher than 90%. Finally, we conducted a user study to test the usability and user experience of EyeBox.

Keywords: Dynamic Area-of-Interest; Eye Tracking; Pupil Core

1. Introduction

In recent decades, eye tracking has been widely used as a research tool [1, 11] in many areas, such as business [9], education [21], medicine [3], and human-computer interaction [18]. By tracking the user's gaze and analyzing eye movement and pupil size, users can study cognitive states, like attention patterns and learning preferences [3, 4]. The eye trackers currently in use can be categorized into two types: remote eye trackers and head-mounted eye trackers [7]. Remote eye trackers, such as EyeLink 1000 and Tobii Pro Fusion, require users to sit in front of the monitor and interact with screen-based content. The advantage is that the temporal and spatial resolutions are both high, but the disadvantage is that the subject's movement is limited. It is generally used for researches based on a laboratory environment. Head-mounted eye trackers are also called glasses-type eye trackers, such as DG3, Pupil Core, and

* Corresponding author. Tel.: +86 29 85308047 ; fax: +86 29 85308047.

E-mail address: zhouyun@snnu.edu.cn

10.1016/j.procs.2022.03.024

Tobii Pro Glasses. These eye trackers have good flexibility and no physical restrictions. It is usually equipped with a pupil camera to photograph one or two eyes. There is also a scene camera used to photograph the world in front of the user and record the environment [12]. Glasses-type eye trackers are designed for real-life eye movement research.

To analyze eye movement data, users need to determine a specific area known as areas of interests (AOIs) in eye tracking experiments [13]. Although existing tools offer dynamic AOIs functions to process visual behavior on moving stimuli, they may ask users to use markers to specify contours of moving stimuli in the physical environment or define AOIs manually on screen. For eye tracking provided by glasses devices, AOIs usually change position or even disappear entirely due to the wearer's movement, which brings difficulties for defining AOIs. In many previous studies, users have drawn boundaries of AOIs frame by frame manually [10], bringing a considerable workload. Some software help users draw AOIs, but most are not open-sourced. Current open-source tools employ markers to track dynamic AOIs in the environment [14]. This method may cause one issue: the markers split the user's attention.

In this work, we propose EyeBox, an open-source toolbox used to process eye gaze data collected by Pupil Core headset. Pupil Core is an eye tracking platform with an open-source software suite and a wearable headset. It has been involved in many research work [18] [17] [14]. Without using markers, it can recognize dynamic AOIs automatically based on SIFT and extract eye movement indicators, including total fixations duration, mean fixations duration, first fixations duration, max fixations duration, min fixations duration, total fixations count, leap count between AOIs, and fixations duration before leap. Besides, it can visualize the original eye tracking data on videos. We also design a user-friendly interface to make this toolbox easy to use. Current Eyebox supports processing data recorded from the Pupil Core device. The generated video is stored in .mp4 format, and the extracted eye movement indicators are stored in .csv format. We compared results processed by manual with by Eyebox in a custom eye-tracking dataset to evaluate this toolbox. The accuracy of 3/4 data for AOI1 is above 90%, and the accuracy of 4/5 data for AOI2 is higher than 90%. Finally, we conducted a user study to test the usability and user experience of EyeBox.

2. Related Work

For processing AOIs, the suppliers provide software and libraries. For example, Tobbi provides Tobbi Pro Lab™ to analyze eye gaze data [8], used for experimental design and analysis. It supports visualization, AOIs selection and indicators extraction, including fixations, saccades, and blinks. Sensomotoric Instruments provides SMI BeGaze™ [20], supporting semantic gaze mapping, heat map generation, fixation mapping, AOIs analysis, and extraction of key performance indicators such as number of revisits to AOI. Additionally, it can extract and export several parameters related to saccades, fixations, blinks, and pupil size. SR Research provides the Data Viewer™ [22] for EyeLink devices. It allows the displaying, filtering, and reporting output of EyeLink I, EyeLink II, EyeLink 1000, and EyeLink 1000 Plus EDF data files. The EyeLink Data Viewer supports three trial-based viewing modes: the Spatial Overlay View, the Temporal Graph View, and the Animation View. These tools can detect multiple AOIs and dynamic AOIs, but they are not automatic, and not free to use. They overlay videos with polygons at various time points. Between any two time points, dynamic AOIs are calculated by interpolating the size and positions of the polygons. However,Pupil [14], the data processing tool of the eye tracker Pupil Core, which is open-source and free to use. It consists of two parts: Pupil Capture and Pupil Player. It also can extract the fixations, saccades, and blinks. A series of markers are displayed around the screen at a certain distance and used for the detection of the AOIs. So the shape of the AOI is single.

Among the open-sourced eye-tracking tools, PyTrack [8] can be used to extract parameters of interest, generate and visualize a variety of gaze plots from raw eye-tracking data. But it cannot deal with multiple AOIs and dynamic AOIs, and lack of analysis GUI. Iris [6] can display a variety of gaze behaviors such as the current gaze point, duration, and saccades. It can also allow adjustment of color, opacity, and other style elements to allow users to visualize their gaze. The design carries out creative and detailed control. None of the above tools can realize the identification of multiple AOIs and dynamic AOIs.

Some software can handle dynamic interest areas. ISeeCube [15], combines the Space-Time Cube (STC) with a linked timeline visualization to support analysis of dynamic AOIs. It draws bounding boxes at keyframes in the video to create new AOIs. AMAT [19], is used to process Tobii Pro Glasses 2 eye tracker data. It uses the AKAZE point feature detection to detect viewpoint features in the video and match the predefined AOI. Dynamic Regions Tracker [2], provides seven tracking algorithms for users to choose from. Researchers find that the MOTLD(Multi

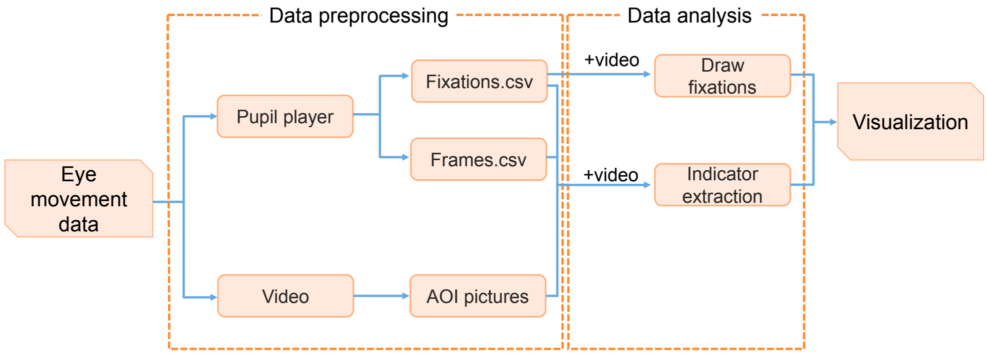


Fig. 1: The processing pipeline of EyeBox.

Object Tracking-Learning-Detection) algorithm works best. The disadvantage of these three tools is that they cannot detect fixations, saccades, and blinks. We compare these eye tracking analysis software as shown in Table.1.

Table 1: The comparison of eye gaze analysis software and tools.

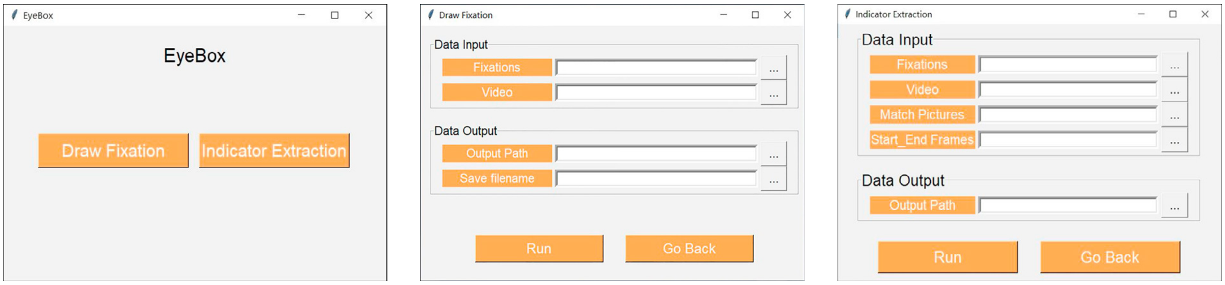
	Tobbi Pro Lab	SMI BeGaze	Data Viewer	Pupil	PyTrack	Iris	ISeeCube	AMAT	Dynamic Region Tracker
Free to use	×	×	×	✓	✓	✓	✓	✓	✓
Blink parameters	✓	✓	✓	✓	✓	✓	×	×	×
Fixation parameters	✓	✓	✓	✓	✓	✓	×	×	×
Saccade parameters	✓	✓	✓	✓	✓	✓	×	×	×
Different AOI shapes	✓	✓	✓	×	✓	×	×	✓	×
Multiple AOIs	✓	✓	✓	×	×	×	✓	✓	✓
Dynamic AOIs	✓	✓	✓	✓	×	×	✓	✓	✓
Fixation plot	✓	✓	✓	✓	✓	✓	✓	×	×
Analysis GUI	✓	✓	✓	✓	×	✓	✓	×	✓

3. Design and Implementation

EyeBox[1] is developed under a 64-bit Windows 10 environment (Microsoft Corp., Redmond, WA, USA) using Python3. As shown in Fig.1, the pipeline consists of four steps for eye movement data analysis, including data preparation, preprocessing, data analysis, and visualization.

Eyebox is open and accessible. Considering the different levels of users' knowledge of Python programming, we also design a friendly interface for Eyebox (see Fig.2). Users with limited programming experience can use this end-to-end tool. Users with rich programming experience can modify open code to meet their requirements. This section first illustrates the main functions and then offers instructions. Eyebox has two modules: fixation drawing module and indicators extraction module (see Fig.2b and Fig.2c). The fixation drawing module uses a fixation data file exported in Pupil Player and draws fixations on the originally recorded videos. The indicator extraction module uses the image matching algorithm to recognize the target image in the video, find and draw AOIs, and compare the fixation data file with AOIs to analyze whether each fixation falls into AOIs. This module extracts the eye movement indicators, including total fixations duration, mean fixations duration, first fixations duration, max fixations duration, min fixations duration, total fixations count, leap count between AOIs, and fixations duration before leap [5].

[1] https://github.com/zling00/EyeBox/releases



(a) (b) (c)

Fig. 2: The interface of Eyebox: (a) entrance interface; (b) visualization module; and (c) indicators extraction module.

3.1. Data Preprocessing

3.1.1. Generate Fixation File

First, download and install the Pupil Core software[2], open Pupil Player, and import the previously recorded data file of a single subject and drag it in. Next, configure the parameters. Set the Minimum data confidence under the General Settings toolbar to 0.6, select Blink Detector, fixations Detector, Raw Data Exporter, World Video Exporter under the Plugin Manager toolbar, and set Maximum Dispersion under the fixations Detector toolbar to 1.5. The Minimum Duration is set to 50, and the Maximum Duration is set to 600. This is because a person's gaze is usually between 50-600 milliseconds. Export the data after completing the fixations detection.

3.1.2. Record Frames of AOIs

The AOIs that users need to identify may not always appear in the video. Sometimes, users need to identify different AOIs at different periods. Thus it is required to record the start and end timestamps of AOIs. The operations are as follows. Create a new .csv (comma-separated) file, and record the number of the recognized interest areas in the first column of the table (such as 1, 2, 3). The second column records the number of frames that AOIs starts. The third column records the end frame number of AOIs.

3.1.3. Pictures of AOIs

EyeBox uses SIFT (Scale Invariant Feature Transform) feature detection algorithm and FLANN matching algorithm. The SIFT algorithm is a scale-invariant feature conversion with image scale invariance and rotation invariance [16]. The FLANN matching algorithm is a fast nearest neighbor search algorithm. This matching algorithm can select the most suitable algorithm to process the data according to the data itself, which is ten times faster than other search algorithms.

The operations are as follows. Open world.mp4 in the original recorded file, import pictures of AOIs, and save as in .jpg format. For example, as shown in Fig.3, the left is one cropped screenshot of the video. The middle is the picture of AOI1, and the right is the picture of AOI2.

3.2. Navigation Instruction

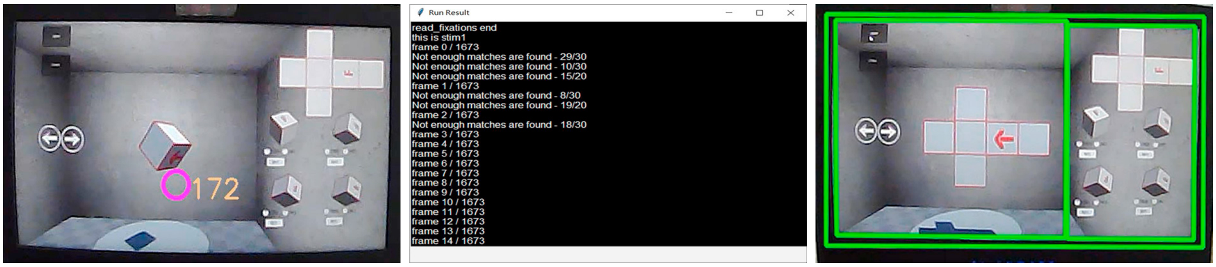
There are three interfaces in the program. The first interface (see Fig.2a) is the primary interface with two buttons. Click the left button to go to the sub-interface for drawing fixations points (see Fig.2b). Click the right button to go to the eye tracking indicator extraction sub-interface (see Fig.2c).

For the fixations drawing operation interface, click the "..." button on the right of Fixations and Video to upload the fixations file exported in Pupil Player and the video to be processed. The result is automatically saved in .mp4 format.

[2] https://docs.pupil-labs.com/core/

(a) (b) (c)

Fig. 3: An example: (a) cropped screenshot of the video; (b) AOI1; and (c) AOI2.



(a) (b) (c)

Fig. 4: (a) Visualization of fixations;(b) The process of the indicator extraction;(c) The result of drawing AOIs.

For the indicator extraction interface, two controls are added to the data input part. Click the ”...” button on the right of Match Pictures and Start_End Frames to upload the captured interest area picture folder and the file recording the number of frames in the area that needs to be recognized.

3.3. Visualization

After uploading the file and clicking Run, the drawing process will be automatically presented on the screen (see Fig.4a). The visualization result will be saved as a file in .mp4 format. For the visualization of the results of the indicator extraction, a new interface (see Fig.4b) will pop up after the program runs, which will output the running status of the program and the image matching results. In this interface, image matching results are output frame by frame. When AOIs match well, the interface will display Current frame/All frames. When the image matching effect is not good, the interface will display not enough matches are found. At the same time, a rectangular box will be drawn on the AOI in the video(see Fig.4c), and the result will be saved as a .mp4 file. The extraction of indicators will be saved as a .csv file.

4. Experiment and User Study

In this experiment, the target is to find two AOIs, AOI1 and AOI2. First, we use the EyeBox to process the data to obtain the number of fixations in the two AOIs. Then we process this data manually. Two people process the same data. If the difference is less than or equal to 0.2, the average value processed by the two people is accepted. If it is greater than 0.2, two other people will process it. A total of 20 pieces of data were processed in this study (see table.2). The accuracy of 3/4 data in AOI1 is higher than 90%. The accuracy of 4/5 data in AOI2 is higher than 90%. But the accuracy of the AOI2 for the 17th data is 0.46. The accuracy of the AOI1 and AOI2 for the 18th data are 0.64 and

0.18. Further analysis reveals that due to environmental light problems during the experiment, the recorded video is too dark or exposed, which affects the image matching, resulting in low accuracy of some data.

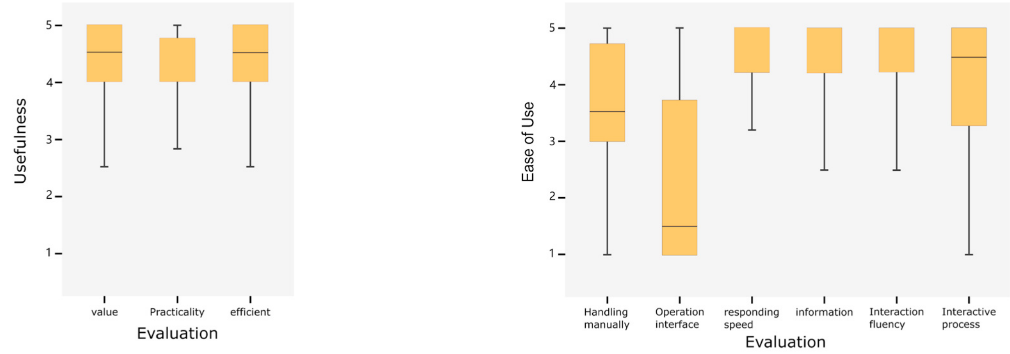
Table 2: The comparison of processed results of eye tracking data by manual and by EyeBox.

Sample	AOI	EyeBox	Manual	Accuracy	Sample	AOI	EyeBox	Manual	Accuracy
1	AOI1	933	910	0.97	**11**	AOI1	1923	1721	0.88
	AOI2	3100	3213	0.96		AOI2	1545	1699	0.91
2	AOI1	1067	1060	0.99	**12**	AOI1	1596	1426	0.88
	AOI2	2340	2587	0.90		AOI2	1967	2256	0.87
3	AOI1	1113	1095	0.98	**13**	AOI1	1903	2011	0.95
	AOI2	2685	2761	0.97		AOI2	2253	2303	0.98
4	AOI1	1280	1150	0.89	**14**	AOI1	1889	2059	0.92
	AOI2	1883	2173	0.87		AOI2	4601	4865	0.95
5	AOI1	1360	1340	0.99	**15**	AOI1	4352	4600	0.95
	AOI2	2078	2133	0.97		AOI2	2604	2861	0.91
6	AOI1	1506	1525	0.98	**16**	AOI1	2340	2183	0.93
	AOI2	1583	1618	0.98		AOI2	2818	3098	0.91
7	AOI1	891	910	0.98	**17**	AOI1	1794	1932	0.93
	AOI2	2086	2105	0.99		AOI2	27	58	0.46
8	AOI1	1087	1132	0.96	**18**	AOI1	583	428	0.64
	AOI2	1294	1323	0.98		AOI2	171	946	0.18
9	AOI1	852	898	0.95	**19**	AOI1	2264	2171	0.96
	AOI2	1906	1917	0.99		AOI2	2697	2870	0.94
10	AOI1	852	896	0.95	**20**	AOI1	2674	2369	0.87
	AOI2	838	852	0.98		AOI2	1804	2058	0.88

Accuracy: EyeBox processing result/Manual processing result.

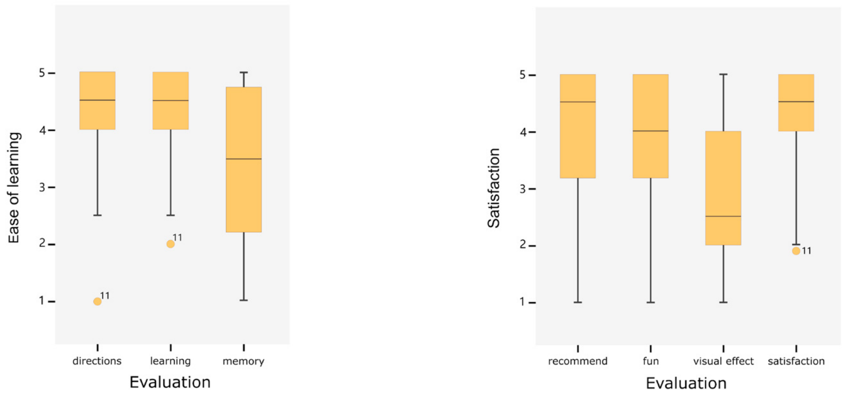
Additionally, we conducted a user study to evaluate the current interface of EyeBox. A total of 12 participants (10 females and 2 males) were recruited, all of whom were students from University. Each participant was asked to perform the task of processing the data, and finally filled out a questionnaire. The questionnaire contained four dimensions: usefulness, ease of use, ease of learning, and satisfaction. A total of 23 questions were designed. Among these, sixteen questions were used to obtain participants' user experience on EyeBox. The rest were about participants' basic information, understanding of eye tracking data processing, and an open question on participants' suggestions. Each question was set with five scoring levels (1-negative, 5-positive), so the maximum score was 80. The results showed that the average score of the questionnaire was 63.83 (SD = 10.50). A small number of participants (30.8%, n = 12) had experience in eye movement data processing, and they all processed eye movement data manually without using any tools.

As shown in Fig.5. For the tool's usefulness, we evaluated the value, practicality, and efficiency. The median score of the three questions was above 4, indicating that EyeBox was effective. Regarding ease of use, results showed that participants thought manually preparing the data (median = 3.5) was slightly complicated, and the interface (median =



(a) (b)

Fig. 5: Evaluation results. (a) usefulness; (b) ease of use.



(a) (b)

Fig. 6: Evaluation results. (a) ease of learning; (b) satisfaction.

1.5) was not attractive enough. But EyeBox response speed (median = 5) and interaction fluency (median = 5) scores were higher. As shown in Fig.6, for the results of the ease of learning, the memory of various operations and functions (median = 3.5) scores were slightly lower. Finally, the visual effect of the interface (median = 2.5) scores were low. Some participants also made suggestions, hoping that the interface could be carefully designed to reduce the manual processing part, consistent with the questionnaire results.

5. Conclusion

In this work, we introduce EyeBox for analyzing and visualizing the data of Pupil Core head-mounted eye trackers. EyeBox could significantly reduce the time for data analysis. By comparing with the manual processing results, we found EyeBox achieved high accuracy. Finally, we conducted a user experience test to evaluate the interface of EyeBox. Results showed that EyeBox has good response speed and interaction fluency. However, some participants hoped to simplify and optimize the interface. We will realize the automatic recognition of a certain area, reduce the manual processing part, and redesign the interface in future work. The current version only processes the data from Pupil Core. We plan to upgrade EyeBox to process the data obtained by different eye trackers.

Acknowledgements

This work was supported by projects of the National Natural Science Foundation of China (62077036), and the national key research and development program of China under grant No.: 2018AAA0100501.

References

[1] B. Adhanom, I., Lee, S.C., Folmer, E., MacNeilage, P., 2020. GazeMetrics: An Open-Source Tool for Measuring the Data Quality of HMD-based Eye Trackers, in: ACM Symposium on Eye Tracking Research and Applications, Association for Computing Machinery, New York, NY, USA. pp. 1–5. doi:10.1145/3379156.3391374.

[2] Bonikowski, L., Gruszczyński, D., Matulewski, J., 2021. Open-source Software for Determining the Dynamic Areas of Interest for Eye Tracking Data Analysis. Procedia Computer Science 192, 2568–2575. doi:10.1016/j.procs.2021.09.026.

[3] Boraston, Z., Blakemore, S.J., 2007. The application of eye-tracking technology in the study of autism. The Journal of Physiology 581, 893–898. doi:10.1113/jphysiol.2007.133587.

[4] Carter, B.T., Luke, S.G., 2020. Best practices in eye tracking research. International Journal of Psychophysiology 155, 49–62. doi:10.1016/j.ijpsycho.2020.05.010.

[5] Chamberlain, L., 2007. Eye Tracking Methodology; Theory and Practice. Qualitative Market Research: An International Journal 10, 217–220. doi:10.1108/13522750710740862.

[6] D'Angelo, S., Brewer, J., Gergle, D., 2019. Iris: A tool for designing contextually relevant gaze visualizations, in: Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications, Association for Computing Machinery, New York, NY, USA. pp. 1–5. doi:10.1145/3317958.3318228.

[7] Geisler, D., Fox, D., Kasneci, E., 2018. Real-time 3D Glint Detection in Remote Eye Tracking Based on Bayesian Inference, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 7119–7126. doi:10.1109/ICRA.2018.8460800.

[8] Ghose, U., Srinivasan, A.A., Boyce, W.P., Xu, H., Chng, E.S., 2020. PyTrack: An end-to-end analysis toolkit for eye tracking. Behavior Research Methods 52, 2588–2603. doi:10.3758/s13428-020-01392-6.

[9] Guo, F., Ding, Y., Liu, W., Liu, C., Zhang, X., 2016. Can eye-tracking data be measured to assess product design?: Visual attention mechanism should be considered. International Journal of Industrial Ergonomics 53, 229–235. doi:10.1016/j.ergon.2015.12.001.

[10] Hessels, R.S., Benjamins, J.S., Cornelissen, T.H.W., Hooge, I.T.C., 2018. A Validation of Automatically-Generated Areas-of-Interest in Videos of a Face for Eye-Tracking Research. Frontiers in Psychology 9, 1367. doi:10.3389/fpsyg.2018.01367.

[11] Hosp, B., Eivazi, S., Maurer, M., Fuhl, W., Geisler, D., Kasneci, E., 2020. RemoteEye: An open-source high-speed remote eye tracker: Implementation insights of a pupil- and glint-detection algorithm for high-speed remote eye tracking. Behavior Research Methods 52, 1387–1401. doi:10.3758/s13428-019-01305-2.

[12] Huang, M.X., Kwok, T.C., Ngai, G., Chan, S.C., Leong, H.V., 2016. Building a Personalized, Auto-Calibrating Eye Tracker from User Interactions, in: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. pp. 5169–5179. doi:10.1145/2858036.2858404.

[13] Jayawardena, G., Jayarathna, S., 2020. Automated Filtering of Eye Gaze Metrics from Dynamic Areas of Interest, in: 2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI), pp. 67–74. doi:10.1109/IRI49571.2020.00018.

[14] Kassner, M., Patera, W., Bulling, A., 2014. Pupil: An open source platform for pervasive eye tracking and mobile gaze-based interaction, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Association for Computing Machinery, New York, NY, USA. pp. 1151–1160. doi:10.1145/2638728.2641695.

[15] Kurzhals, K., Heimerl, F., Weiskopf, D., 2014. ISeeCube: Visual analysis of gaze data for video, in: Proceedings of the Symposium on Eye Tracking Research and Applications, Association for Computing Machinery, New York, NY, USA. pp. 43–50. doi:10.1145/2578153.2578158.

[16] Lowe, D., 1999. Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, pp. 1150–1157 vol.2. doi:10.1109/ICCV.1999.790410.

[17] Lu, S., Sanchez Perdomo, Y.P., Jiang, X., Zheng, B., 2020. Integrating Eye-Tracking to Augmented Reality System for Surgical Training. Journal of Medical Systems 44, 192. doi:10.1007/s10916-020-01656-w.

[18] Majaranta, P., Bulling, A., 2014. Eye Tracking and Eye-Based Human–Computer Interaction, in: Fairclough, S.H., Gilleade, K. (Eds.), Advances in Physiological Computing. Springer, London. Human–Computer Interaction Series, pp. 39–65. doi:10.1007/978-1-4471-6392-3_3.

[19] Meyer, L., Josefsson, B., Vrotsou, K., Westin, C., Lundberg, J., 2021. Evaluation of an AoI Mapping and Analysis Tool for the Identification of Visual Scan Pattern, in: 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), pp. 1–8. doi:10.1109/DASC52595.2021.9594500.

[20] Papenmeier, F., Huff, M., 2010. DynAOI: A tool for matching eye-movement data with dynamic areas of interest in animations and movies. Behavior Research Methods 42, 179–187. doi:10.3758/BRM.42.1.179.

[21] Que, Y., D'Andrea-Penna, G., Hu, X., Dong, Y., Chiba, A., Iversen, J., 2021. Reading with and without Background Music: An Exploration with EEG, Eye Movement and Heart Rate.

[22] Wang, Z., 2021. Eye Movement Data Analysis and Visualization, in: Wang, Z. (Ed.), Eye-Tracking with Python and Pylink. Springer International Publishing, Cham, pp. 197–224. doi:10.1007/978-3-030-82635-2_8.