# Fixation Detection with Ray-casting in Immersive Virtual Reality

Najood Alghamdi[1]
Computer Science Department
King Abdulaziz University, Jeddah, Saudi Arabia

Wadee Alhalabi[2]
Computer Science Department
King Abdulaziz University, Jeddah, Saudi Arabia

***Abstract*—This paper demonstrates the application of a proposed eye fixation detection algorithm to eye movement recorded during eye gaze input within immersive Virtual Reality and compares it with the standard frame-by-frame analysis for validation. Pearson correlations and a sample paired t-test indicated strong correlations between the two analysis methods in terms of fixation duration. The results showed that the principle of eye movement event detection in 2D can be applied successfully in a 3D environment and ensures efficient detection when combined with ray-casting and event time.**

***Keywords—Eye Movement; Eye Tracking; Virtual Reality; Fixation Detection; HMD***

## I. Introduction

Analysis of eye movement data recorded by eye trackers has been shown to be a valuable tool for psychological research and diagnosis. During the eye-tracking experiment, the data sample represented as a stream of data exhibits specific behaviors which can be used to detect oculomotor events. It is important to determine specific limits or thresholds before the detection of eye movement events. These thresholds of eye movement event detection have no standards or accurately defined values, but can be determined by trial and error, or by experience, or taken from literature in the same paradigm, which may lead to subjective results [1-3] whether the analysis is performed manually or algorithmically.

However, the algorithms used in eye movement research are based on general principles for detecting oculomotor events. The gaze coordinates (x, y) for a given stimulus display are calculated and represented as a data sample, and, based on the spatial and temporal characteristics of these gazes, events can be detected. Frequent events include fixations and saccades, along with their associated characteristics such as fixation duration and saccadic amplitudes. A fixation is represented as a group of consecutive gaze points resulting from the eye stopping to look at a target, where the fixation duration is the time window between onset and offset of a fixation, which in most studies, is bounded between 200 ms and 400 ms and is rarely less than 100 ms. A saccade is represented as spaced-out gaze points where no visual processing can occur because of the rapid jumps in the eye movement towards a target.

The spatial characteristics of eye movements are the velocity and dispersion of eye gazes, and the duration is a temporal characteristic. To measure the velocity, it is necessary to sample gaze data at a high rate. This allows the velocity between consecutive gaze points to be calculated. Thresholds can then be applied to this velocity to detect and classify events. Fixations are indicated by low velocities between consecutive gaze points (<100 °/s), while saccades are indicated by high velocities (>300 °/s) [3]. Dispersion is also used to identify fixations and saccades, because tightly clustered data points tend to indicate a fixation, whereas data points which are more widely spread spatially tend to indicate a saccade.

However, in eye movement research, detection algorithms are often not described clearly, and their measurements rely mainly on the hardware specification and the research objectives. This makes it challenging to perform a meaningful comparison of algorithms for movement detection. To address this problem and create a road map of algorithmic validation, in [3], Salvucci and Goldberg proposed a novel classification of fixation identification algorithms based on the principal techniques used in implementations of these algorithms.

On the other hand, the majority of eye movement research utilizes commercial analysis tools provided by the eye tracker manufacturers. In commercial software, the detected oculomotor events, mainly fixations, represented using different gaze visualization techniques include scan paths, areas of interest (AOI), and attentional or visual maps.

### A. Eye Movements in Virtual Reality

Virtual reality (VR) technology provides the opportunity to conduct, in realistic environments, experiments which would be very expensive or unsafe to conduct in real environments. Thus, the ability to use eye tracking and detect oculomotor events within these environments will open a new door for researchers to dig deep inside human behavior and user experience (UX), to support training and education, and many other applications, in safe and convenient environments.

On the other hand, despite all the advantages of VR for scientific research, creating these environments is still a significant challenge that consumes a lot of time and effort, as a given research goal typically requires specific visual objects and characters. Also, studying eye movements within a specific experiment necessitates defining the visual behavior to be measured and its significance to the research question: primarily, the eye metrics, such as fixation or pupil size, and their characteristics that are associated with the visual behavior. Consequently, combining VR environments with eye-tracking technology will increase complexity and will be time-consuming.

One of the possible solutions to reduce the time and cost in such a complex environment is to use off-the-shelf components and suitable VR environments from previous related studies. However, eye-tracking studies have mainly represented the gaze coordinates as (x, y) on a 2D display representing the contents of the environment. This means that the tools available for visualization and analysis in this context are not adequate for analysis of eye movements in a 3D environment [4], as they can only be applied to scenes which are composed of frames. However, measuring and analyzing eye movements in immersive VR is a relatively new sub-field in eye-tracking research. In this area of research, the stimulus within the VR environment contains visual objects or scenes for specific purposes, and the experiment aims to test a hypothesis through objects which are known as objects of interest (OOI) or through AOI in the scene, similar to the principles of experiments with head-mounted eye-tracking systems. Eye-tracking studies in real environments are still time-consuming, as the display will be video recorded, and must then be examined manually, such as frame-by-frame. Frame-by-frame analysis is an appropriate tool, acceptable in the analysis of eye movements on video records. However, it has a significant drawback for researchers conducting an eye-tracking experiment involving video recording, where the gaze coordinate must be checked in each frame to see whether it is located on a specific predefined AOI or not. This is very time-consuming, so only small samples can be taken [5][6]. The same method of video recording and manual analysis [3][7] is applied to eye movement within the VR environment.

Currently, there is an ongoing effort to develop eye-tracking systems that are adequate to emerge with immersive VR. In 2018, embedded 60-120 Hz eye trackers in high-performance head-mounted displays (HMDs) appeared on the ground: these are FOVE [8], SMI and Tobii [9]. These HMDs with embedded binocular eye trackers (HMD-ET) include multiple infra-red sources and use the pupil-corneal reflection technique to measure the distance from the corneal reflected light "glint" to the extracted pupil center, then calculate the gaze direction by measuring the changing distance between the glint and the moving pupil center to allow free tracked head movements in the helmet. The accuracy and reliability of these new devices depend on all the hardware and software technologies utilized to build these systems. Until now, this emerging technology has been still in development, and there are many challenges for researchers in designing adequate environments and software tools for analysis and visualization of eye movements that utilize and take advantage of such technology. Unfortunately, this technology also has no technical limitation on producing massive quantities of hardware and tools with a variety of techniques like previous eye-tracking systems [4], and most manufacturers do not disclose the algorithmic solutions used in their products. Therefore, the difficulty of comparing studies of eye tracking in 3D will continue, just as in 2D classical studies.

Unlike eye tracking in a classical 2D environment, several streamed data can be generated using HMD-ET that can be used for more in-depth analysis of eye movements and additional understanding of human behavior compared to classical 2D. Point of regard (POR), for example, is the gaze point mapped onto the projected image on an HMD screen, identical to 2D gaze, but with a 3D vector ray-casting representing the direction from a virtual camera origin for both eyes to the POR. Another example is 3D vectors representing the actual direction of both eyes when looking at the physical world and their origins at the center of the eyeballs. Currently, reading these streamed data is only possible by creating a client-side file dump [10].

This paper proposes a simple methodology to detect eye fixations and OOI using HMD-ET in current VR environments. To validate the proposed algorithm's analysis of eye movements within VR, its outcome was compared with the outcome of a frame-by-frame gaze location analysis.

## II. Previous Works

Duchowski et al. [11] introduced novel techniques for binocular eye tracking within VR. These are 3D calibration techniques and 3D eye movements analysis techniques. The study was conducted on an aircraft inspection simulation training application. Stellmach et al. [7] proposed a set of gaze visualization techniques with a prototype toolkit for supporting eye movement analysis in static 3D environments similar to heat maps and scan paths that are used in gaze visualizations for 2D contents, and investigated the usefulness of their techniques. In another similar recent study in [12], the gaze fixations were mapped onto a geometrical model to generate 3D attention volumes similar to heatmaps. Boukhalfi et al.'s [13] work in progress presented the development of a multimodal brain–computer interface (BCI) at the Montreal Philippe-Pinel Institute for different studies related to forensic psychiatry (including the integration of eye tracking glasses within a 4-wall CAVE-like VR environment). The majority of the previous studies including VR and eye tracking used a semi-immersive VR approach, as in [14], which showed that average 3D gaze errors between 3D object and the eye gaze increased linearly with the distance of the virtual planes. In contrast to this, HMD-ET captures the eye from a close distance that leads to a more accurate gaze estimation.

## III. Methodology

### A. Hardware and Software Requirements Specification

Despite the progress in GPUs, it is still challenging to render 3D graphics in a resolution adequate for the resolution of the eye. Currently, minimum requirements should be considered for the machine that will be used to run the VR environment and the eye tracker plugin. Also, it is important to provide a high-frame-rate VR [12] to avoid VR sickness. The hardware requirements are: an msi laptop with GeForce GTX 1080 8 GB, Intel Core i7 $7^{th}$ Gen (2.80 GHz), 16 GB RAM, connected to an HMD from VIVE HTC [15] with a field of view (FOV) of 110 degrees and a resolution of 1080 × 1200 pixels per eye and a refresh rate of 90 Hz. The HMD is integrated with a pair of eye trackers from SensoMotoric Instruments (SMI) which track the subjects' gaze in the 3D environment with a typical accuracy of 0.5° [10]. Each eyepiece of the goggles is trimmed with a small ring of a miniature infrared camera and six infrared lights positioned in a circle around each eye. This tracker works with the C++/C# SDK for various VR engines. The setting platform is the

Windows 10 operating system, and the 3D application development tool used is Unity. Eye-tracing data were recorded in the Unity VR engine relative to an origin coordinate at the upper left corner of the HMD screens, as specified by the SMI plugin documentation [10]. The recorded eye-tracking data were sent to a dump file which was processed by the proposed algorithm implemented in Matlab. The tools used for video recording and manual analysis were ApowerREC, Kinovea, and MS Excel. Finally, SPSS was employed for statistical tests.

### B. Participants

Five participants volunteered in the experiment, three males, and two females, age 26 ± 8.2 years (mean ± SD). All participants are healthy with no motor or neural abnormalities.

### C. Stimuli

The task scene is a VR environment used to demonstrate gaze interaction with Unity provided by SMI including an eye-tracking plugin. The scene is presented as a room with cube-shaped objects on the walls. Although in VR all visual objects are named, some of the cubes were defined ahead as OOI (i.e., MiddleObj, CornerObj, RightMidObj, LeftObj, LeftMidObj, and BackObj) for the purpose of comparison between the eye movement analysis methods. The eye-tracking plugin software represented the participant's gaze point on the display as a small pointer-like blue circle (gaze cursor).

### D. Experiment Procedure

Each participant was given a brief explanation of the simple task that they would be required to complete, in which they wear the helmet, complete the calibration procedure and look around for one minute, fixing their gazes on the objects on the walls freely and unaware of the defined OOI. After each participant understood the instructions for the task, the HMD-ET was adjusted to the participant and then calibrated. A standard calibration procedure was used in which each participant was instructed to stare at a red point which appears at the beginning and keep tracking its movement. Once the calibration procedure has succeeded, the virtual room was presented to the participant's view.

In each participant session, in the same order, the eye-tracking data streams were recorded in a dump file during the experiment. A complete video recording of each participant's view contents was also saved. When a task was completed, the dump file was analyzed offline using the proposed algorithm, and the video record was analyzed using manual gaze location detection in frame-by-frame analysis.

### E. Eye Movement Data Analysis Methods

*1) Proposed dispersion-based algorithm:* The proposed methodology to detect fixation and OOI in VR environments using HMD-ET is based on several of the previously known principles in the literature. The main steps of the proposed algorithm are reading streamed data, denoising, fixation detection, and OOI hit detection. The eye-tracking data stream was collected into the dump file and the algorithm was implemented in Matlab according to the previously determined measures.

*a) Measurements used in the implementation:* In streamed eye movement data, there is always some noise or undesirable data depending on factors such as the equipment used and the environment. Therefore, denoising is essential to remove such data. Here, denoising was achieved by eliminating data points corresponding to blinking. When a participant's eye is closed, the pupil diameter is zero, so the tracker returns the coordinates (0, 0). Also, any data points outside the boundaries of the environment were eliminated.

To detect gaze fixation, the concept of dispersion was used, based on its usage in 2D fixation detection. Choosing a suitable threshold depends on the hardware and the experiment specifications. Here, an HTC HMD was used with a FOV of 110 degrees and a resolution of 1080 × 1200 pixels per eye. In most human computer interaction (HCI) eye-tracking studies, the typical fixation size is one degree, so to convert the threshold of one degree into pixels using Pythagoras’ theorem, one degree is approximately equivalent to 15 pixels in this display. The precision of the fixation size, which can be determined by the dispersion between the coordinates of successive concussive gazes, can be varied depending on the required task and the virtual objects in the VR.

The distance between two concussive gazes is calculated using the Euclidean distance of the gazes' coordinates (x, y) as follows:

$$\text{Distance} = \text{sqrt}((x_2 - x_1)^2 + (y_2 - y_1)^2) \quad (1)$$

According to the literature, the minimum duration of a fixation is 100–250 ms. Again, the duration can be varied depending on many factors such as stimulus complexity and the required task to be accomplished. The HMD used has a 90 Hz refresh rate, which means 90 gazes recorded per second, so nine gazes can be recorded in 100 ms. Consequently, when more than nine gazes are found within the determined threshold (15 pixels), it forms a cluster, which represents a fixation within the corresponding timestamp.

The 3D vector of ray-casting originating from the participant’s eye is traced to check if an OOI is hit when the event occurs. When fixation forming coincides with the ray-casting hitting the OOI, the fixation is assigned to the OOI.

*b) The output of the algorithm:* In addition to detection of the start and end time, and the number of gazes in each fixation, the proposed algorithm was able to detect small fixations that are difficult to detect manually. It also shows the corresponding OOI that was fixated on. Graphical representation of the timeline shows the gazes in the XY plane during a time slot and the corresponding 3D vector of the ray-casting direction in the 3D environment (sample shown in Fig. 1).

*2) Frame-by-frame analysis:* The video software Kinovea used for the frame-by-frame analysis shows 1800 frames per minute for each participant's view content. Each frame was checked for the gaze cursor location and was coded into a spreadsheet by assigning the gaze to one of the predefined visual objects that overlay the gaze cursor. When the gaze cursor was located outside those OOI or when there was no signal for the gaze cursor, it was assigned to No OOI (sample shown in Fig. 2).

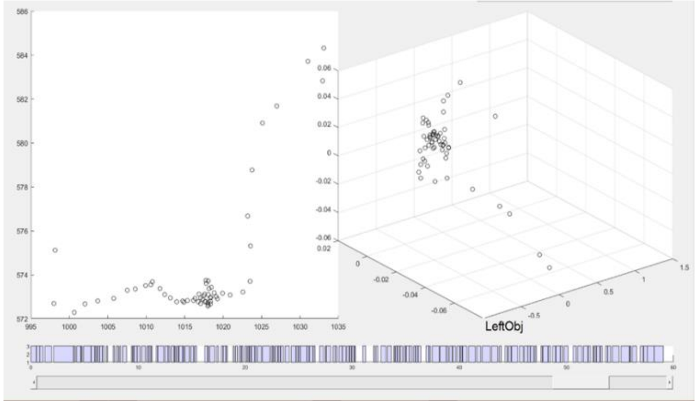

Fig. 1. Gaze Direction and POR Gazes in the Selected Timeslot.

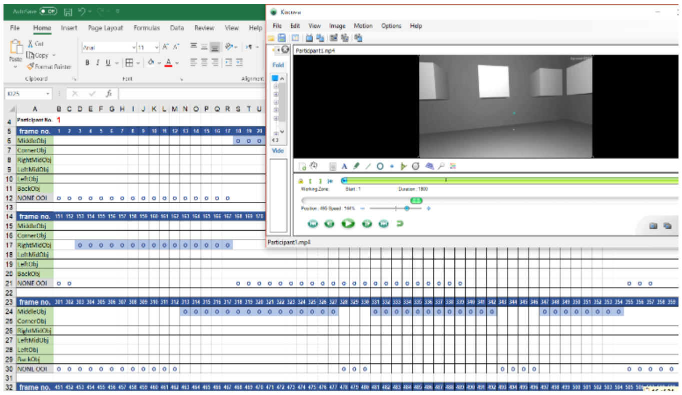

Fig. 2. Frame-by-Frame Analysis Spreadsheet.

The minimum fixation threshold implemented in the first method was 100 ms, which is equivalent to 3 frames of 1800 frames per minute. Therefore, at least three consecutive frames should be encoded for the same OOI to form a fixation, and the number of frames indicates the length of the fixation duration.

### F. Comparison Method

The number of fixations in the frame-by-frame analysis is much less than in the algorithm, as long fixations were recorded in the frame-by-frame analysis in cases where there were several consecutive small fixations on the object, which can only be detected by the algorithm. Also, sometimes consecutive fixations were found in the frame-by-frame analysis due to the absence of the gaze cursor signal. To compare between the two methods, the total duration of fixations on each corresponding OOI for each participant was calculated in MS Excel, using the number of frames in frame-by-frame analysis and using the number of gazes in the proposed algorithm. The fixation durations were compared in SPSS to validate the proposed algorithm with frame-by-frame analysis using a Pearson correlation and a paired sample t-test.

## IV. Results

Pearson correlations were carried out for each pair of OOI (e.g., MiddleObj in algorithm output versus MiddleObj in the frame-by-frame output). The results of the correlation analysis are shown in Table I.

Very significant and strong correlations are exhibited between the two methods. This indicates that both methods are very similar in terms of fixation duration. The exception to this is LeftObj, which exhibits a non-significant correlation (p = 0.098). However, the correlation coefficient is nevertheless strong (r = 0.808).

A t-test was used for evaluating differences between the two methods. This procedure is summarized in Table II. No significant differences were found between the two methods for any of the paired OOIs. As such, it can be said that the output of the proposed fixation detection algorithm and the frame-by-frame analysis are statistically identical in terms of fixation duration.

The fixation duration percentage results from the fixation detection algorithm and the frame-by-frame method, for the six OOIs in the analysis, can be found in Fig. 3, confirming that the proposed analysis algorithm produces similar results to the frame-by-frame analysis.

TABLE. I. Pearson Correlations

| **Pair (OOI in the algorithm – OOI in frame-by-frame)** | **Correlation** | **Sig.** |
|---|---|---|
| MiddleObj - MiddleObj | 0.984 | 0.003 |
| CornerObj - CornerObj | 0.929 | 0.022 |
| RightMidObj - RightMidObj | 0.980 | 0.004 |
| LeftObj - LeftObj | 0.808 | 0.098 |
| LeftMidObj - LeftMidObj | 0.940 | 0.018 |
| BackObj - BackObj | 1.000 | 0.000 |

TABLE. II. Paired Samples T-Test for the Proposed Algorithm and Frame-by-Frame Analysis

| **Pair (OOI in the algorithm – OOI frame-by-frame)** | **Mean** | **Std. Dev.** | **Std. Error mean** | **t** | **Sig. 2-tailed** |
|---|---|---|---|---|---|
| MiddleObj-MiddleObj | 171.1 | 931.3 | 416.5 | 0.411 | 0.702 |
| CornerObj-CornerObj | 1337.8 | 1379.8 | 617.1 | 2.168 | 0.096 |
| RightMidObj-RightMidObj | 424.4 | 973.5 | 435.3 | 0.975 | 0.385 |
| LeftObj- LeftObj | 395.6 | 1698.7 | 759.7 | 0.521 | 0.630 |
| LeftMidObj-LeftMidObj | 735.6 | 1094.3 | 489.4 | 1.503 | 0.207 |
| BackObj- BackObj | -86.7 | 193.8 | 86.7 | 1.000 | 0.374 |
| df = 4 | | | | | |

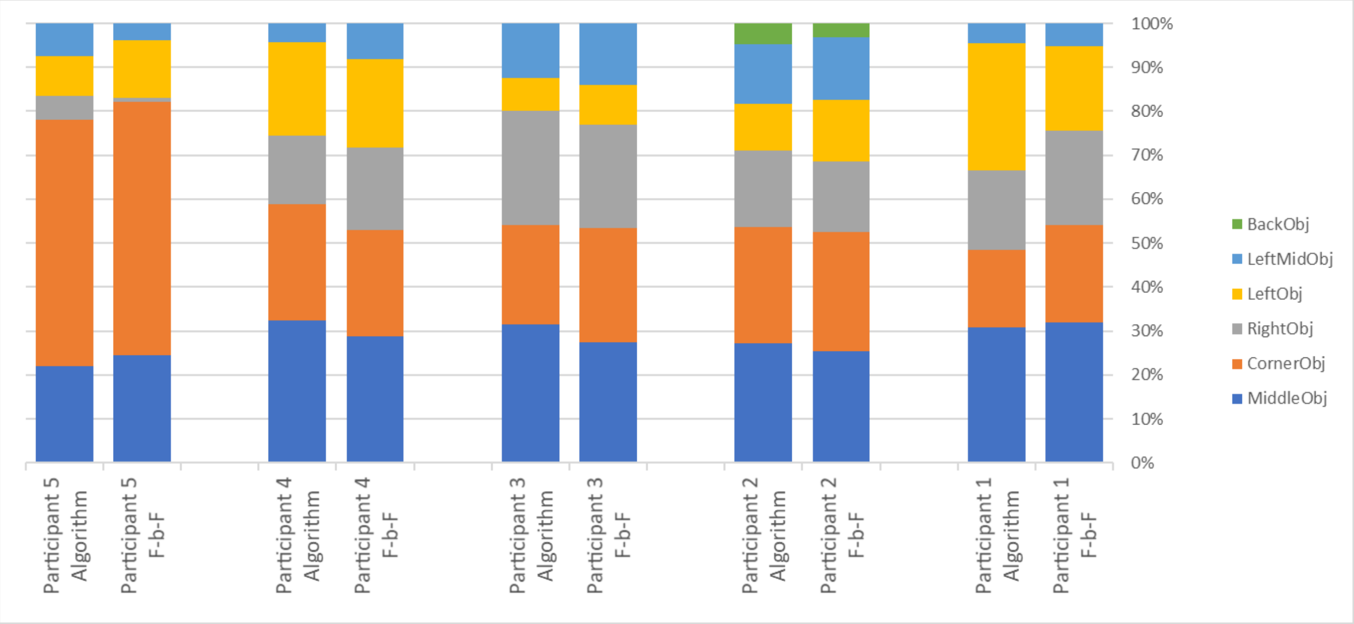

Fig. 3. Duration Percentages for OOIs in the Fixation Detection Algorithm and the Frame-by-Frame Analysis.

## V. Conclusion and Future Work

Eye movement analysis in VR is still a new research area among eye-tracking studies. This paper has presented an initial investigation of eye gaze within the VR environment and a demonstration of a proposed fixation detection algorithm within current VR environments using SMI HMD-ET. Continued development of the proposed algorithm will be considered for future eye movement analysis studies.

In VR, the position of a pixel within a projected scene will reflect different objects each time rendering occurs. One of the solutions to this issue lies in the ray-casting originating from the participant's eye; it is a convenient technique for finding where the user is looking, thus providing the opportunity to study the user's gaze behavior in VR environments. This ray-casting can be utilized to get the hit information for an OOI, when it exists, and determine the gaze direction inside the VR world.

The results of the proposed algorithm indicate the possibility of utilizing the same principles of eye movement analysis algorithms that are applied to the spatial or temporal characteristics of gazes in 2D to study eye gaze behavior in a 3D environment, in conjunction with the time of events and ray-casting, in many applications. Nevertheless, further investigation into the data stream recorded by eye-tracking devices in immersive VR needs to be considered in future work to utilize these data streams efficiently.

To validate the proposed algorithm, the result of its analysis was compared with frame-by-frame manual analysis, which is employed and recognized as a tool to analyze eye movement data with the similar technology of eye-tracking devices mounted on the head like eyeglasses. The results of both methods correlated very highly, using Pearson and paired t-tests. Nevertheless, there are many differences between the two methods that cannot be ignored. Despite the effort and time spent on the frame-by-frame analysis, it is not useful in determining an actual number of fixations and their spatial and temporal characteristics. With such drawbacks, algorithmic solutions for eye movement analysis are required. While this is difficult in the natural environment using head-mounted eye tracking, the integration of eye tracking with immersive VR represented in HMD-ET technology provides an alternative solution for many cases. With the tremendous advance of graphics, VR offers a safe and controlled realistic environment, which has allowed many costly or dangerous research studies to be transferred from nature to the laboratory and eye-tracking studies are not an exception with the emergence of this HMD-ET. Moreover, it opens the door for discoveries and research in different multidisciplinary fields.

Lastly, it is worth mentioning that many applications or studies require a precise selection of visual objects from within a dense visual field or an estimation of fixation depth, such as in joint attention and visual communication studies. These requirements require the search for a new paradigm and innovative approaches to analyzing eye movements within the depth of the virtual environment.

## Acknowledgment

This research was supported by King Abdulaziz City for Science and Technology (KACST) grant no. 1425-37- أط

## References

[1] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, H. Jarodzka, and J. Van De Weijer, Eye Tracking: A comprehensive guide to methods and measures, First pub. OxfordUniversity Press, 2011.

[2] R. Andersson, L. Larsson, K. Holmqvist, M. Stridh, and M. Nystrom, "One algorithm to rule them all? An evaluation and discussion of ten eye movement event-detection algorithms.," Behav. Res. Methods, vol. 49, no. 2, pp. 616–637, Apr. 2017.

[3] D. D. Salvucci and J. H. Goldberg, "Identifying Fixations and Saccades in Eye-Tracking Protocols," ACM Press, no. Proceedings of the symposium on Eye tracking research & applications-ETRA '00, pp. 71–78, 2000.

[4] T. Blascheck, K. Kurzhals, M. Raschke, M. Burch, D. Weiskopf, and T. Ertl, "State-of-the-Art of Visualization for Eye Tracking Data," Euographics Conf. Vis. EuroVis, 2014.

[5] A. Klostermann, R. Kredel, and E.-J. Hossner, "The quiet eye without a target: The primacy of visual information processing.," Journal of Experimental Psychology: Human Perception and Performance, vol. 40, no. 6. Bern, Switzerland, pp. 2167–2178, 2014.

[6] P. Vansteenkiste, G. Cardon, R. Philippaerts, and M. Lenoir, "Measuring dwell time percentage from head-mounted eye-tracking data – comparison of a frame-by-frame and a fixation-by-fixation analysis," Ergonomics, vol. 58, no. 5, pp. 712–721, 2015.

[7] S. Stellmach, L. Nacke, and R. Dachselt, "Advanced gaze visualizations for three-dimensional virtual environments," Proc. 2010 Symp. Eye-Tracking Res. Appl. - ETRA '10, p. 109, 2010.

[8] "Home - FOVE Eye Tracking Virtual Reality Headset." [Online]. Available: https://www.getfove.com/. [Accessed: 04-May-2019].

[9] "Tobii.com," 2019. [Online]. Available: https://www.tobii.com/. [Accessed: 10-May-2019].

[10] SMI, "Eye Tracking Solutions," 2019. [Online]. Available: https://www.smivision.com/. [Accessed: 19-Mar-2018].

[11] A. T. Duchowski, E. Medlin, A. Gramopadhye, B. Melloy, and S. Nair, "Binocular eye tracking in VR for visual inspection training," Proc. ACM Symp. Virtual Real. Softw. Technol. - VRST '01, p. 1, 2001.

[12] C. Hanel, B. Weyers, B. Hentschel, and T. W. Kuhlen, "Visual Quality Adjustment for Volume Rendering in a Head-Tracked Virtual Environment," IEEE Trans. Vis. Comput. Graph., vol. 22, no. 4, pp. 1472–1481, 2016.

[13] T. Boukhalfi, C. Joyal, S. Bouchard, S. M. Neveu, and P. Renaud, "Tools and techniques for real-time data acquisition and analysis in brain computer interface studies using qEEG and eye tracking in virtual reality environment," IFAC-PapersOnLine, vol. 28, no. 3, pp. 46–51, 2015.

[14] S. Wibirama, H. A. Nugroho, and K. Hamamoto, "Evaluating 3D gaze tracking in virtual space : A computer graphics approach," Entertain. Comput., vol. 21, pp. 11–17, 2017.

[15] Bethesda Game Studio, "VIVETM | VIVE Virtual Reality System," Bethesda Softworks , 2018. [Online]. Available: https://www.vive.com/us/product/vive-virtual-reality-system/. [Accessed: 23-Mar-2018].