VEDBViz: The Visual Experience Database Visualization and Interaction Tool

Sanjana Ramanujam*
Christian B. Sinnott*
sramanujam@nevada.unr.edu
csinnott@nevada.unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Bharath Shankar
bshankar@nevada.unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Savannah Halow
savvyhalow@gmail.com
University of Nevada, Reno
Reno, Nevada, USA

Brian Szekely
bszekely@nevada.unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Kamran Binaee
kbinaee@unr.edu
University of Nevada, Reno
Reno, Nevada, USA

Paul R. MacNeilage
pmacneilage@unr.edu
University of Nevada, Reno
Reno, Nevada, USA

ABSTRACT

Mobile, simultaneous tracking of both the head and eyes is typically achieved through integration of separate head and eye tracking systems because off-the-shelf solutions do not yet exist. Similarly, joint visualization and analysis of head and eye movement data is not possible with standard software packages because these were designed to support either head or eye tracking in isolation. Thus, there is a need for software that supports joint analysis of head and eye data to characterize and investigate topics including head-eye coordination and reconstruction of how the eye is moving in space. To address this need, we have begun developing VEDBViz which supports simultaneous graphing and animation of head and eye movement data recorded with the Intel RealSense T265 and Pupil Core, respectively. We describe current functionality as well as features and applications that are still in development.

CCS CONCEPTS

• **Human-centered computing → Visualization toolkits**; *Visualization design and evaluation methods*; Ubiquitous and mobile computing systems and tools.

KEYWORDS

eye-tracking, head-tracking, data visualization, eye-movements, head-movements, gaze

ACM Reference Format:
Sanjana Ramanujam, Christian B. Sinnott, Bharath Shankar, Savannah Halow, Brian Szekely, Kamran Binaee, and Paul R. MacNeilage. 2021. VEDBViz: The Visual Experience Database Visualization and Interaction Tool. In *2021 Symposium on Eye Tracking Research and Applications (ETRA '21 Adjunct), May 25–27, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3450341.3458486

*Both authors contributed equally to this research.

ETRA '21 Adjunct, May 25–27, 2021, Virtual Event, Germany

ACM ISBN 978-1-4503-8357-8/21/05…$15.00
https://doi.org/10.1145/3450341.3458486

1 INTRODUCTION

Joint tracking of both head and eye position is necessary to address a range of applied and basic scientific research questions [Hausamann et al. 2020; Kothari et al. 2020]. For example, it allows investigating coordinated movements of both head and eyes and reconstructing movements of the eye in space, which ultimately determines motion sensed at the retina [MacNeilage et al. 2019; Matthis et al. 2018]. Analysis methods for eye movement data are highly developed, but methods for joint analysis of head and eye movement are less mature.

To facilitate joint analysis of head-eye tracking data, we have developed the Visual Experience Database Visualizer (VEDBViz). While the visualizer is built initially for use with Visual Experience Database (VEDB) data, we plan to accommodate loading of joint head-eye data from any system, as long as the data is formatted appropriately. The VEDB is a collaborative effort across three universities (University of Nevada, Reno; North Dakota State University; Bates College) to collect more than 200 hours of data, including joint head-eye movements, across a diverse range of individuals during natural behaviors. Data are collected using the VEDB headset which combines a Pupil Core eye tracker with the Intel RealSense T265 (T265) for tracking head motion.

We first focus on the functionality of the visualizer, including real-time head-eye animation, time-series visualization, and video playback/scrubbing. We also describe current and future applications of the visualization software including data inspection, event annotation, figure generation, and other applications.

1.1 Previous Work

Most commercially available eye tracking and motion tracking systems come bundled with manufacturer-specific software for data processing, including rudimentary data visualization. Typically this software can only visualize eye data collected with that tracker. Visualizations are typically limited to a few options that the manufacturer deems appropriate, but these options may not be suitable for all purposes. With the exception of the Pupil software suite [Kassner et al. 2014], this commercial software is not open source, meaning that the researcher has limited ability to modify the visualization or customize the analysis.

Furthermore, no commercial solutions exist for joint analysis of head and eye movements. To our knowledge, two commercial mobile eye trackers come equipped with an inertial measurement unit (IMU) that could be used to estimate head movement at the same time as eye movements: Pupil Invisible [Tonsen et al. 2020] and the Tobii Pro Glasses 3 [Tobii 2020]. Like other mobile eye-trackers without IMUs, these come bundled with purpose-built data processing and visualization software (Pupil Player and Tobii Pro Lab, respectively), but they do not include options for visualizing head movements.

Similarly, there are a variety of commercial motion tracking solutions that may be used to track head motion including IMUs (xSens), outside-in optical tracking (Vicon, OptiTrack, OptoTrak), and inside-out tracking (T265), and these all provide some sort of visualization capability. However, they are typically closed-source and therefore do not allow for incorporation of eye movement data.

Within the research community, recent reports have provided open-source software solutions and open-access data for image segmentation [Chaudhary et al. 2019; Yiu et al. 2019], gaze mapping [Kassner et al. 2014], gaze event classification [Kothari et al. 2020], mobile gaze tracking [Hausamann et al. 2020; Kinsman et al. 2012; Tomasi et al. 2016], evaluation of gaze data quality [Adhanom et al. 2020], and gaze-contingent experimental design [Dalmaijer et al. 2014]. But in their current form, none of these allow for joint analysis and visualization of head and eye movement data.

Thus, there is a clear gap, and subsequently, a need for joint head-eye visualization software. While the ultimate needs for analysis and visualization are likely to differ significantly across labs, the general ability to quickly visualize data is highly valuable. Even if it does not meet every users' exact needs, effective visualization can serve researchers as a first-line tool for rapid and intuitive pre-processing, inspection, diagnostics, and event annotation. To begin addressing this gap, we have developed a functioning prototype of VEDBViz software that allows for real-time playback and visualization of joint head-eye data.

2 CURRENT WORK

2.1 Scope

We set out to create a tool for visualization of data collected for the VEDB, specifically head and eye movement data. The VEDB acquisition device records data from several sensors including a FLIR Chameleon 3 forward-facing world camera (3.2 MP, 2048 × 1536 pixel resolution, 55 Hz frame rate), an Intel RealSense T265 forward-facing stereoscopic tracking camera (two global shutter fish-eye world cameras: 173° diagonal FOV, 30 Hz frame rate, 848x800 pixel resolution; 3 DOF accelerometer: ±4g range, 62.5 Hz sampling rate; 3 DOF gyroscope: ±2000° range; 200 Hz sampling rate), as well as two eye-facing cameras from the Pupil Core (200 Hz, 192x192 pixel resolution). The LG G6 Android Smartphone is worn separately from the headset on the body and provides additional IMU and global positioning system (GPS) data.

Currently, the software loads in only the head and eye movement data: 6 degrees of freedom (DOF) position of the head, and 2 DOF position of each eye. The visualizer generates an animation of head and eye movement along with time-series graphing of head movement data. Controls allow for playback and scrolling through the data. The visualizer is lightweight and functions both on a computer with dedicated graphics processing unit (Nvidia GeForce GTX 1650) and without (Intel Iris Plus Graphics).

2.2 Visualization

In order to inspect simultaneous head-eye tracking data, we constructed two data visualization modules. These were implemented in Unity 2020.2.7f1 as this platform is well-suited for generating high-quality graphics. Users first interact with both visualization modules by indicating the data they wish to visualize. Users enter the file path to the appropriate data file via keyboard input. Once complete, users click a "Load" button which imports appropriate data from the file, and starts both visualization modules.

The first module allows graphing of 6 DOF head data and 3 DOF gaze data. In the software's default configuration, a panel displays 3 DOF linear head velocity as a time-series graph (1A). The parameter currently plotted can be changed via drop-down menu, allowing users to plot their choice of linear or angular head velocity as well as gaze norm position. Time-series data are graphed using the BitSplash IO Chart and Graph package, available in the Unity Asset Store [IOSplash 2021].

In addition to the graphing module, we created a head-eye kinematic animation module (1B). This module displays a real-time animation consisting of a human skull with eyes such that the movement recreates the recorded head and eye movement of the participant. The head moves based on the 6 DOF head data recorded by the T265 and movement of each eye is governed the by 2 DOF eye data recorded by Pupil Core. The animation is implemented in Unity as a single parent object (the skull) and two child objects (each eye) in a 3D space. Assets used for the skull and eye were found online and used under Creative Commons license [Creativo 2021; martinjario 2016].

The user's view of the animation begins in a default position and orientation, with the viewport pointed directly at the front of the skull and eyes. Because the animation includes changes in the linear position of the head, it is possible that the skull and eyes move out of the field of view (FOV) of the camera. In this case, users are able to adjust the camera view by first clicking a button to toggle camera controls, then using the mouse to radially move the camera about the head and eyes.

2.3 Playback

A key component of VEDBViz is the ability for the user to quickly play through and navigate data recorded by the VEDB headset. Playback ability allows a user to access any moment of time in the recording session. It is essential for efficient visual inspection, since playback affords the user the ability to check for data tracking errors or examine someone's head and eye movements.

The playback features in this application consist of several buttons and a frame indicator (Fig 1C). The buttons allow the user to play and pause the animation via keyboard. The frame indicator shows the current sample in the animation relative to total time. Assets and scripts were used or modified from [Weimann 2018].

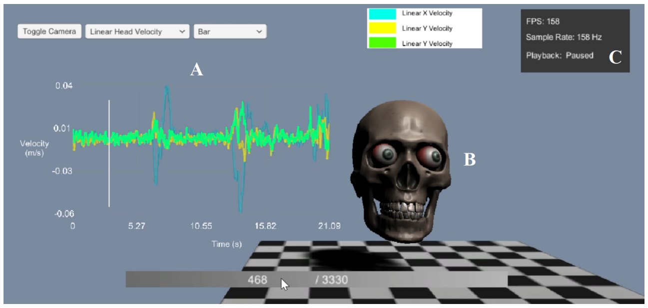

Figure 1: Screenshot of VEDBViz. Time-series visualization with legend is shown in (A). Head-eye kinematic module is shown in (B). Playback status is shown in (C). Still image is taken when participant was fixated on a target.

3 DISCUSSION

3.1 Summary

Here we present ongoing work to create VEDBViz; a gaze visualization and interaction tool for use with the VEDB. This first iteration of VEDBViz allows users to load and visualize joint head-eye movements via both animation and graphing. The user is also able to playback through data using a set of standard media playback tools. These represent first steps towards a joint head-eye movement visualization tool. To our knowledge there are currently no dedicated visualizers for joint head-eye movement available for purchase in the commercial sector or open-source in the research community.

3.2 Future Directions

Future work on VEDBViz will focus on adding functionality to the current iteration of the software. Currently, our software accepts a single *.csv file with both head and eye data, and this must be generated ahead of time with the proper frame rate and format. In the future, we aim to implement more flexibility such that head and eye data with different frame rates can be loaded as long as the time stamps are overlapping. We also aim to accommodate the loading of head and eye data from other devices so that the visualizer may be used with other datasets.

In addition to the Pupil Core and the T265, the VEDB headset features multiple other sensors not supported in this iteration of VEDBViz. These include a high-resolution world camera, the FLIR Chameleon 3, as well as an LG smartphone tracking positional data via GPS. Future iterations of VEDBViz will seek to expand sensor integration. These include playback for video files generated by the FLIR world camera, the T265, and Pupil Core's two eye cameras. Future sensor integration will also include visualization of GPS and IMU data recorded on the smartphone worn on the participant's body.

Next, multiple future features focus on increasing functionality of VEDBViz for data pre-processing. The ability to implement various filters on data in the visualization environment allow researchers an easy, intuitive way of seeing how these filters may affect their data. Data annotation and labeling is also a key future development; as these processes are made more expedient through data visualization.

Finally, the current iteration of VEDBViz relies upon a licensed package (BitSplash IO Chart and Graph) for the time-series visualization module. Future iterations will move towards a scratch-built time-series visualization module. In addition to providing us with greater transparency on how the visualization itself is rendered, moving towards our own implementation will allow us to move the entire toolset towards an open-source solution. This fits with the stated purpose of the VEDB: to be a large, open-access dataset with associated open-source software tools.

ACKNOWLEDGMENTS

All authors conceived the visualization software. S.R. and S.H. coded the main interface and visualization modules. B.S., S.H., and B.S. coded the playback module. C.S. wrote the main manuscript text. All authors reviewed the manuscript.

This research was supported by NSF-EPSCOR under grant number OIA-1920896.

REFERENCES

Isayas B. Adhanom, Samantha C. Lee, Eelke Folmer, and Paul MacNeilage. 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* (Stuttgart, Germany) *(ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA, Article 19, 5 pages. https://doi.org/10.1145/3379156.3391374

Aayush K. Chaudhary, Rakshit Kothari, Manoj Acharya, Shusil Dangi, Nitinraj Nair, Reynold Bailey, Christopher Kanan, Gabriel Diaz, and Jeff B. Pelz. 2019. RITnet: Real-time Semantic Segmentation of the Eye for Gaze Tracking. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (Oct 2019). https://doi.org/10.1109/iccvw.2019.00568

Oscar Creativo. 2021. *Eye Free Model 3D.* https://sketchfab.com/3d-models/eye-free-model-3d-by-oscar-creativo-5d466ea41c874fc5b376c92a313a9bb3 Accessed: 2021-3-12.

Edwin S Dalmaijer, Sebastiaan Mathôt, and Stefan Van der Stigchel. 2014. PyGaze: An open-source, cross-platform toolbox for minimal-effort programming of eyetracking experiments. *Behavior Research Methods* 46, 4 (2014), 913–921.

Peter Hausamann, Christian Sinnott, and Paul R. MacNeilage. 2020. Positional head-eye tracking outside the lab: An open-source solution. In *Eye Tracking Research and Applications Symposium (ETRA) (ETRA '20 Short Papers)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3379156.3391365

IOSplash. 2021. *Graph and Chart.* https://assetstore.unity.com/packages/tools/gui/graph-and-chart-78488 Accessed: 2021-3-12.

Moritz Kassner, William Patera, and Andreas Bulling. 2014. Pupil: An Open Source Platform for Pervasive Eye Tracking and Mobile Gaze-based Interaction. In *Adjunct Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 1151–1160. https://doi.org/10.1145/2638728.2641695

Thomas Kinsman, Karen Evans, Glenn Sweeney, Tommy Keane, and Jeff Pelz. 2012. Ego-Motion Compensation Improves Fixation Detection in Wearable Eye Tracking. In *Proceedings of the Symposium on Eye Tracking Research and Applications* (Santa Barbara, California) *(ETRA '12)*. Association for Computing Machinery, New York, NY, USA, 221–224. https://doi.org/10.1145/2168556.2168599

Rakshit Kothari, Zhizhuo Yang, Christopher Kanan, Reynold Bailey, Jeff B Pelz, and Gabriel J Diaz. 2020. Gaze-in-wild: A dataset for studying eye and head coordination in everyday activities. *Scientific Reports* 10, 1 (2020), 1–18.

Paul R MacNeilage, Luan Nguyen, and Christian Sinnott. 2019. Characterization of natural head and eye movements driving retinal flow. *Journal of Vision* 19, 10 (2019), 147d–147d.

martinjario. 2016. *Skull downloadable.* https://sketchfab.com/3d-models/skull-downloadable-1a9db900738d44298b0bc59f68123393 Accessed: 2021-3-12.

Jonathan Samir Matthis, Jacob L Yates, and Mary M Hayhoe. 2018. Gaze and the control of foot placement when walking in natural terrain. *Current Biology* 28, 8 (2018), 1224–1233.

Tobii. 2020. Tobii Pro Glasses 3. https://www.tobiipro.com/product-listing/tobii-pro-glasses-3/. Accessed: 2021-3-12.

Matteo Tomasi, Shrinivas Pundlik, Alex R. Bowers, Eli Peli, and Gang Luo. 2016. Mobile gaze tracking system for outdoor walking behavioral studies. *Journal of Vision* 16, 3 (2016).

Marc Tonsen, Chris Kay Baumann, and Kai Dierkes. 2020. A High-Level Description and Performance Evaluation of Pupil Invisible. (Sep 2020). arXiv:2009.00508 [cs.CV]

Jason Weimann. 2018. Unity Video Player with Controls & Time Scrubber. https://unity3d.college/2018/04/25/unity-video-player-controls-time-scrubber/. Accessed: 2021-3-10.

Yuk-Hoi Yiu, Moustafa Aboulatta, Theresa Raiser, Leoni Ophey, Virginia L. Flanagin, Peter zu Eulenburg, and Seyed-Ahmad Ahmadi. 2019. DeepVOG: Open-source pupil segmentation and gaze estimation in neuroscience using deep learning. *Journal of Neuroscience Methods* 324 (2019), 108307. https://doi.org/10.1016/j.jneumeth.2019.05.016