Published in final edited form as:

Proc Eye Track Res Appl Symp. 2021 May ; 2021: . doi:10.1145/3450341.3458881.

Post-processing integration and semi-automated analysis of eye-tracking and motion-capture data obtained in immersive virtual reality environments to measure visuomotor integration

Haylie L. Miller[*],
University of Michigan

Ian R. Zurutuza[*],
University of North Texas

Nicholas E. Fears,
University of Michigan

Suleyman O. Polat,
University of North Texas

Rodney D. Nielsen
University of North Texas

Abstract

Mobile eye-tracking and motion-capture techniques yield rich, precisely quantifiable data that can inform our understanding of the relationship between visual and motor processes during task performance. However, these systems are rarely used in combination, in part because of the significant time and human resources required for post-processing and analysis. Recent advances in computer vision have opened the door for more efficient processing and analysis solutions. We developed a post-processing pipeline to integrate mobile eye-tracking and full-body motion-capture data. These systems were used simultaneously to measure visuomotor integration in an immersive virtual environment. Our approach enables calculation of a 3D gaze vector that can be mapped to the participant's body position and objects in the virtual environment using a uniform coordinate system. This approach is generalizable to other configurations, and enables more efficient analysis of eye, head, and body movements together during visuomotor tasks administered in controlled, repeatable environments.

Keywords

mobile eye tracking; visuomotor integration; motion capture; machine learning; computer vision; object detection; kinematic; oculomotor

millerhl@umich.edu.

[*] Co-first authors.

1 INTRODUCTION

Eye-tracking technology has advanced considerably in the past decade, in part due to increased affordability and decreased size of high-speed cameras and mobile recording devices. Early studies required uncomfortable instrumentation and fixed-head tasks, neither of which lend themselves to naturalistic use of gaze. Mobile eye-tracking has enabled more naturalistic studies with free head and body movement, for example, during walking [Tomasi et al., 2016]. This approach has great potential to inform our understanding of how humans acquire and use visual information in the wild, but data processing and analysis techniques have failed to keep pace.

Eye-tracking research is limited by the time and human resources required to analyze video-based data, particularly as tasks increase in complexity and duration. Many studies still use frame-by-frame coding of areas of interest (AOIs) or events (hit, miss) [Geruschat and Hassan, 2005] rather than capitalizing on the increased temporal and spatial precision now available [Diaz et al., 2013]. Software-based solutions often focus on increasing the speed or efficiency of manual coding [Benjamins et al., 2018] rather than on automated approaches. While a few researchers have attempted to leverage advances in computer vision algorithms for automated object detection with some success [Brone et al., 2011; De Beugher et al., 2014], these solutions historically have not performed well under suboptimal task conditions or are not sufficiently flexible.

Data acquisition challenges still exist, particularly under circumstances involving (1) unconstrained head or body movement, (2) special populations (e.g., children; those with behavioral or physical differences), or (3) recording of eye-tracking data in conjunction with other measures (e.g., motion capture, neuroimaging) or in complex environments (e.g., virtual reality, video games). Combined, these challenges limit the efficacy and generalizability of eye-tracking, and must be addressed in order to broaden and refine the use of this technology.

1.1 Measuring Eye Movement in Challenging Contexts

Solutions for analysis of gaze during unconstrained head and body movement typically fall into two categories: (1) coordinate-based, requiring information about the location of the participant in the world relative to target objects, and (2) target-based, which can be trained to detect and label objects of interest. Coordinate-based solutions require information about the participant's location relative to the visual world, such as a local coordinate system that plots the participant's gaze point on a scene camera video, or a global coordinate system like a motion-capture volume. Target-based solutions require that objects of interest be visible in a point-of-view video of the visual environment. Both have inherent challenges; an optimal solution would enable analysis of unconstrained head and/or body movement data relative to both eye movements and the environment.

Eye-tracking can be challenging in special populations with behavioral, cognitive, physical, or developmental differences. Known issues exist with regard to whether calibration of mobile eye-tracking systems are robust to perturbation or slipping, and how often they must be re-calibrated [Niehorster et al., 2020]. These concerns are particularly relevant to populations whose inhibitory control and/or sensory needs may result in frequent adjustment of wearables (e.g., children with Autism Spectrum Disorder, ASD). Other concerns include measurement of pathological eye movement in clinical populations, for example schizophrenia [Morita et al., 2019] or ASD [Brandes-Aitken et al., 2018], where there are known oculomotor differences that are difficult to accommodate using standard algorithms for detection and annotation of eye movement events. As such, there is a need for data processing pipelines that are robust to these issues.

1.2 Integrating Mobile Eye-Tracking with Motion-Capture & Virtual Reality

The data output from mobile eye-tracking systems is often recorded in a local coordinate system (with the exception of systems that have proprietary add-on packages for head tracking). As a result, there is a need for efficient, flexible solutions to transform data from multiple local coordinate systems to one global coordinate system, so that these data can be analyzed in combination rather than isolation. This will facilitate more precise use of eye-movement data for both measurement of gaze characteristics in research and gaze-controlled commercial applications (e.g., gaming, adaptive technology). Particularly in the case of virtual reality technology, this is critical given the need for high fidelity to reduce cybersickness and increase user tolerance.

2 METHOD

2.1 Participants

We compared eye-movement and body-movement data from two 7-year-old male children, one with ASD and one TD. These participants were selected from a sample obtained as part of a larger study (see Appendix B), for the purpose of demonstrating outcome measures obtained from our post-processing and analysis approach.

2.2 Apparatus

Testing was performed in an immersive virtual environment utilizing integrated mobile eye-tracking (SMI ETG 2w) and motion-capture systems (Motion Analysis Corp.). Main specifications of the data collection systems, virtual environment, and tasks are described below; additional detail can be found elsewhere [Miller et al., 2017].

Mobile Eye-Tracking System: The SMI ETG 2.5w binocular eye-tracking glasses (SensoMotoric Instruments) have 3 digital cameras that record at a sampling rate of 60 Hz. The scene camera on the nose bridge of the glasses has a 60°horizontal × 46°vertical FOV and a 960 × 720 p resolution at 30 FPS. Two eye-tracking cameras on the bottom inner frame use pupil tracking with parallax compensation in a proprietary algorithm to yield a 3D gaze vector in a coordinate space relative to the scene camera. Data are recorded on a Samsung Galaxy S4 equipped with iView ETG data acquisition software. The glasses yielded real-time eye movement data for each eye independently and for a combined “gaze vector” with a spatial accuracy of 0.5°of visual angle. The glasses were calibrated using a 3-point method for each participant, and calibration was verified using 13 targets in the virtual environment.

The motion-capture system consisted of 12 Eagle-4 infrared cameras (Motion Analysis Corp.) that recorded the 3D coordinates (spatial accuracy = 0.5 mm) of 28 reflective markers placed on the participant's body and the eye-tracking glasses at a sampling rate of 120 Hz. Data were streamed in real-time to a control computer equipped with the Cortex software suite (Motion Analysis Corp.), where they were visualized to ensure quality.

The Computer-Assisted Rehabilitation Environment Network (CAREN; Motek ForceLink) is an immersive virtual rehabilitation system used to generate virtual environments with static and dynamic targets and user-controlled objects. A battery of visuomotor integration tasks, described elsewhere [Miller et al., 2017], required participants to employ varying degrees of visuomotor integration depending on task complexity. The height of the virtual environment and objects in it was adjusted to the Y coordinate of a reflective marker located on the right top of the glasses frame to ensure that the visual experience was relatively uniform regardless of participant height.

2.3 Data Analysis

Eye-tracking Analysis: Automatic event-detection for most low-speed (< 240 Hz) eye-tracking systems is based on a dispersion model with fixations, saccades and blinks as the primary events. In this model, saccades could include rapid pursuit eye movements. More sophisticated approaches are needed to analyze pursuit eye movement and subsequent gaze point in relation to the precise location and trajectory of objects (or their projected images) in the real world. For this reason, we are developing software to temporally and spatially align data captured by mobile eye-tracking with data from motion capture and visual projection systems.[1]

Temporal Alignment of Data: Alignment of data from motion-capture and the virtual environment with eye-tracking data *could* be achieved via manual annotation of the start of tasks in the eye-tracking scene camera videos. This approach is common, but inefficient, and introduces variable amounts of human error. Instead, we trained an algorithm to automatically identify objects of interest in videos, which we align with their presentation timing. We performed transfer learning on Inception V2 [Huang et al., 2017]. This neural network, originally trained on the COCO [Lin et al., 2015] image dataset, was fine-tuned to identify and generate bounding boxes for 24 objects, 12 of which are usable for temporal alignment with the VR presentation. Initially, we labeled approximately 2000 frames from the scene video using labelImg software [darrenl, 2021]. This included more than 150 frames for each object used to automate temporal alignment. Images were split into 70% train and 30% test sets before transfer learning.

Spatial Alignment of Data: Objects in the virtual world are projected onto the inside of a half-cylinder screen, radius = 2.49m, with the participant at its axis, standing at global (motion-capture) coordinates (x,y,z)=(0,0,0). The virtual coordinate system is the same as the global system. We can easily compute the location of object projections (incidence on screen) given a point of view and the screen's location, using just Pythagorean's theorem.

[1] We will make reusable components available to other researchers in a later stable version.

Determining where the participant's gaze is on the screen is more challenging. First, we have to learn an eye-tracking scene-camera model relative to three markers on the glasses. We define a glasses coordinate system, which includes those three markers, the camera's scene viewpoint, and its orientation. The camera's viewpoint and orientation are unknown and must be *fit* to the data collected.

We use a least squares method with the error (residual) between the virtual objects' actual projection locations and their perceived locations computed based on their location in the scene-camera's captured image and the (in-progress) learned camera model. We can compute the perceived projection locations by applying rigid-body transformation [Arun et al., 1987] and coordinate translation to the camera model based on the three glasses markers' locations. This determines the point on the screen corresponding to the center pixel in our image and the image's orientation. The screen location corresponding to other pixels is determined by the camera's image resolution (960 × 720 pixels) and field of view (60 × 46 degrees). Once the camera model is learned, we can easily determine where a participant's gaze strikes the screen based on transformation of the camera model, as before, followed by computing the relative gaze angle given the recorded gaze pixel. This allows us to compute metrics to compare a participant's gaze and a target location. (See Appendix A for an approach to derive relevant equations.)

3 RESULTS

3.1 Object Detection, Temporal Alignment, & Spatial Alignment Performance

The object detector's recall on the test set was 97.4% for the fixation cross which appears between trials. (True positives are defined as correctly labeled bounding boxes where the intersection with the gold-standard bounds over their union (IoU) is 0.75+.) In each task, there are at least 14 cross presentations; so the probability of missing at least 50% of crosses on their first frame is $0.026^7 < 10^{-12}$. Thus, the probability that we fail to detect a sufficient number of object appearances for temporal alignment is essentially zero. To perform effective spatial alignment, we only need a fraction of the thousands of image frames with target balls. Therefore, we filter out the detections that violate the symmetry of balls – the smaller of width or height must be at least 95% of the larger. The root mean squared error on the detected objects is 3.7 pixels, but the error is gaussian, so the simple mean error was 0.2 pixels in our sample. The latter error is more relevant, since the alignment seeks a least squared error fit, allowing for accurate spatial alignment from the detected object locations. Using a Nvidia K80 GPU, it takes around 13.5 minutes to label five minutes of video (30 FPS; 9000 frames). While time consuming, this requires no manual annotation and runs unattended. By contrast, manual labeling would take several hours.

3.2 Visuomotor Performance Metrics

Our approach yields a wide array of metrics designed to analyze and mine spatial and temporal patterns of gaze behavior during naturalistic tasks requiring visuomotor integration. Figure 1 presents a visualization of one such metric, distance-to-target, across a single trial for an interception task, which required the participant to move their body to position a user-controlled object in the path of an oncoming target object. In this task, although average reaction time over trials was similar for participants (TD = 257 ms, ASD = 234 ms), the average time-to-target was notably lower for the TD participant (483 ms) than for the participant with ASD (770 ms).

4 DISCUSSION

Our novel post-processing data integration pipeline is generalizable to other configurations using a head-mounted scene camera and 3D coordinate data for objects of interest in real or virtual environments. We were able to successfully demonstrate its efficacy as an approach for generating precise visuomotor performance metrics with a lower cost of time and human resources typically required for manual coding of eye-tracking data. We also demonstrated the value of this approach for use with challenging populations (e.g., children, ASD), where unconstrained head and body movement may make manual temporal alignment and coding of areas of interest difficult or impossible. We presented participant data here for the purposes of illustrating the rich temporal and spatial performance metrics that can be automatically generated, rather than for the purpose of evaluating clinically-significant differences. However, it is clear that our approach is useful for examining differences between typical and pathological visuomotor task performance.

This approach has the potential to serve as a tool to improve efficiency of data analysis for other researchers interested in maximizing the temporal and spatial resolution of mobile eye-tracking solutions in combination with other technologies such as motion-capture and virtual reality. Future directions include refinement of our object detection approach to include edge detection, enabling more precise analysis of spatial accuracy, and use of this approach to automatically detect visuomotor biomarkers in clinical populations.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

ACKNOWLEDGMENTS

We thank the families and community partners who supported this work, as well as the National Science Foundation (SMA-1514495) and the National Institutes of Health (KL2-TR001103; K01-MH107774).

REFERENCES

Tomasi Matteo, Pundlik Shrinivas, Bowers Alex, Peli Eli, and Luo Gang. 2016. Mobile gaze tracking system for outdoor walking behavioral studies. Journal of Vision, 16(27). DOI: 10.1167/16.3.27

Geruschat Duane and Hassan Shirin. 2005. Driver behavior in yielding to sighted and blind pedestrians at roundabouts. Journal of Visual Impairment and Blindness, 995, 286–302. DOI: 10.1177/0145482X0509900504

Diaz Gabriel, Cooper Joseph, Kit Dmitry, and Hayhoe Mary. 2013. Real-time recording and classification of eye movements in an immersive virtual environment. Journal of Vision, 13(5). DOI: 10.1167/13.12.5

Benjamins Jeroen S., Hessels Roy S., and Hooge Ignace T. C.. 2018. Gazecode: open-source software for manual mapping of mobile eye-tracking data. ETRA ‘18: Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications, June 2018, article no. 54, 1–4. DOI: 10.1145/3204493.3204568

Brone Geert, Oben Bert, and Goedeme Toon. 2011. Towards a more effective method for analyzing mobile eye-tracking data: integrating gaze data with object recognition algorithms. PETMEI ‘11: Proceedings of the 1st international workshop on pervasive eye tracking & mobile eye-based interaction, September 2011, 53–56. DOI: 10.1145/2029956.2029971

De Beugher Stijn, Brone Geert, and Goedeme Toon. 2014. Automatic analysis of in-the-wild mobile eye-tracking experiments using object, face, and person detection. In VISAPP 2014: International Conference on Computer Vision Therapy and Applications, Vol. 9.

Niehorster Diederick C., Santini Thiago, Hessels Roy S., Hooge Ignace T. C., Kasneci Enkelejda, & Nystrom Marcus. 2020. The impact of slippage on the data quality of head-worn eye trackers. Behav Res, 52, 1140–1160. DOI: 10.3758/s13428-019-01307-0

Morita Kentaro, Miura Kenichiro, Kasai Kiyoto, and Hashimoto Ryota. 2019. Eye movement characteristics in schizophrenia: A recent update with clinical implications. Neuropsychopharmacology Reports, 40, 2–9. DOI: 10.1002/npr2.12087 [PubMed: 31774633]

Brandes-Aitken Anne, Anguera Joaquin A., Rolle Camarin E., Desai Shivani S., Demopoulos Carly, Skinner Sasha N., Gazzaley Adam, and Marco Elysa J.. 2018. Characterizing cognitive and visuomotor control in children with sensory processing dysfunction and autism spectrum disorders. Neuropsychology 32, 2 (2 2018), 148–160. DOI: 10.1037/neu0000404 [PubMed: 29376661]

Miller Haylie L., Bugnariu Nicoleta L., Patterson Rita M., Wijayasinghe Indika, and Popa Dan O.. 2017. Development of a novel visuomotor integration paradigm by integrating a virtual environment with mobile eye-tracking and motion-capture systems. In 2017 International Conference on Virtual Rehabilitation (ICVR), 1–6. DOI: 10.1109/ICVR.2017.8007481

Huang Jonathan, Rathod Vivek, Sun Chen, Zhu Menglong, Korattikara Anoop, Fathi Alireza, Fischer Ian, Wojna Zbigniew, Song Yang, Guadarrama Sergio, and Murphy Kevin. 2017. Speed/accuracy trade-offs for modern convolutional object detectors. arXiv:1611.10012 [cs] (4 2017). (Specifically, we used the model named “Faster R-CNN” from the TensorFlow object detection API.)

Lin Tsung-Yi, Maire Michael, Belongie Serge, Bourdev Lubomir, Girshick Ross, Hays James, Perona Pietro, Ramanan Deva, Zitnick C. Lawrence, and Dollár Piotr. 2015. Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs] (2 2015).

darrenl. 2021. tzutalin/labelImg. Retrieved March 14, 2021 from https://github.com/tzutalin/labelImg“ https://github.com/tzutalin/labelImg

Arun KS, Huang TS, and Blostein Steven. 1987. Least-squares fitting of two 3-D point sets. IEEE T Pattern Anal. Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-9, (10 1987), 698–700. DOI: 10.1109/TPAMI.1987.4767965

CCS CONCEPTS

• Human-centered computing; • General and reference; • Computing methodologies;

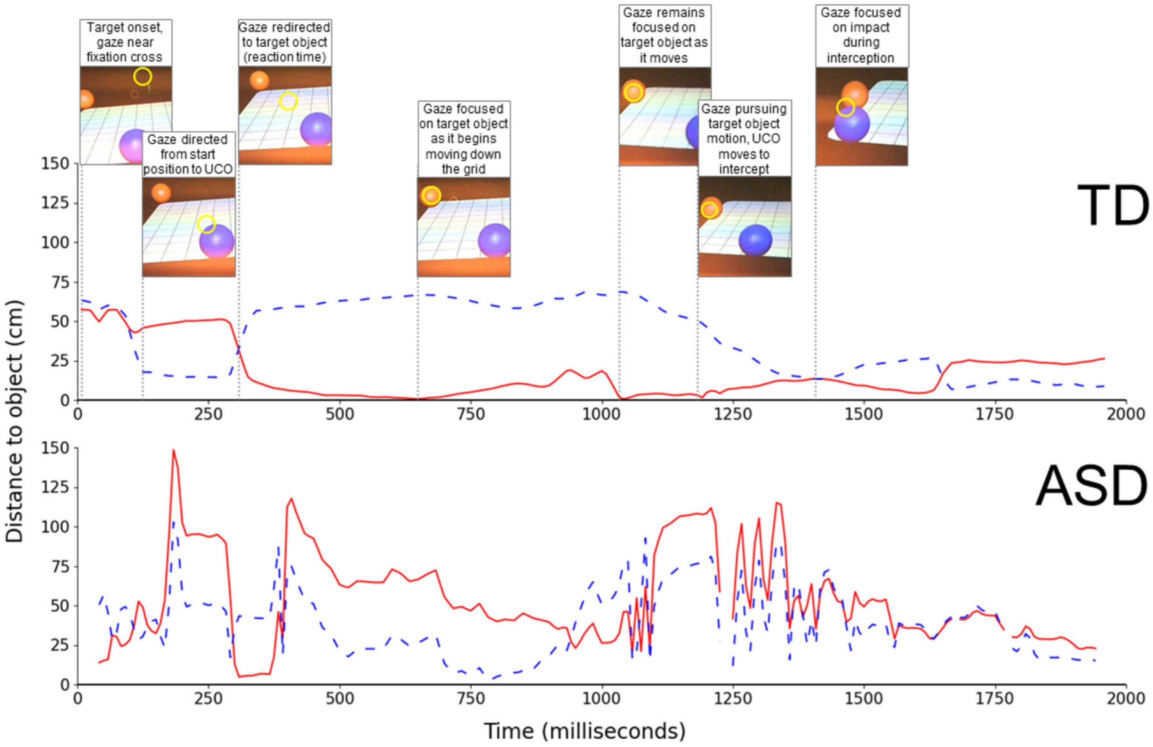

Figure 1:

Distance from gaze point to the target object (solid line) and the user-controlled object (UCO, dashed line) are shown during a single trial of the intercept task for a TD child (top panel) and a child with ASD (bottom panel). Vertical lines in the top panel correspond to images showing the target object (red ball), the UCO (blue ball), and gaze (yellow circle).