EPiC Series in Computing
Volume 82, 2022, Pages 141–149
Proceedings of 37th International Conference on Computers and Their Applications

An Application for Simulating Patient Handoff Using 360°Video and Eye Tracking in Virtual Reality

Christopher Lewis[1], Sven Diaz-Juarez[1], Steven J. Anbro[2], Alison J. Szarko[2], Ramona A. Houmanfar[2], Laura H. Crosswell[3], Michelle J Rebaleati[4], Luka A Starmer[4], and Frederick C Harris, Jr.[1]

[1] Dept. of Computer Science and Engineering
[2] Behaviour Analysis Program
[3] Reynolds School of Journalism
[4] @Reality

University of Nevada, Reno
Reno, NV, USA, 89557
christopher_le1@nevada.unr.edu

Abstract

Virtual reality (VR) is a relatively new and rapidly growing field which is becoming accessible by the larger research community as well as being commercially available for entertainment. Relatively cheap and commercially available head mounted displays (HMDs) are the largest reason for this increase in availability. This work uses Unity and an HMD to create a VR environment to display a 360°video of a pre-recorded patient handoff between a nurse and doctor. The VR environment went through different designs while in development. This works discusses each stage of it's design and the unique challenges we encountered during development. This work also discusses the implementation of the user study and the visualization of collected eye tracking data.
Keywords: Medical Simulation, VR, Eye Tracking, 360 Video, HMD

1 Introduction

This work describes an application used for displaying a 360°video in VR to students practicing to become doctors and nurses. The 360°video started with a doctor coming into a patient room while a nurse is taking care of a patient, in this case a training mannequin. The doctor and nurse initiate a patient handoff while the user watches. A patient handoff is the very common action of transferring patient care from one medical professional to another. The doctor and nurse get cut short as the patient has a crisis that requires both of their attention. During this crisis, they quiz the user on what they discussed during the patient handoff, where the user has to respond verbally. This scenario was given to the user two times. Once at the beginning of their training for patient handoff, and once at the end of their training. During this scenario, this application records eye tracking data, which is collected from the HTC Vive [7] HMD with

TOBII [12] beta eye tracking integration. This hardware is no longer in beta, and is an early implementation of the eye tracking that is a part of the HTC Vive Pro Eye HMD. The virtual space itself was created using the Unity [14] game engine and then integrating it with SteamVR and the TOBII beta eye tracking SDK, which is the beta version of the SRanipal Unity SDK[5].

The rest of this paper is structured as follows: In Section 2 we discuss a few related works to this project focused on 360°video, eye tracking, and VR medical simulators. In Section 3 we cover the implementation of the user study and eye tracking data visualization. In Section 4 we cover different versions of this project's virtual space with specific notes about corresponding eye tracking with objects/people in 360°video. In Section 5 we discuss the overall conclusions found from this application. Finally, in Section 6 we introduce future work that would make this application better and allow further research in this area.

2 Background and Related Work

There are quite a few umbrella terms that might classify this work, from software engineering to human-computer interaction. The unique aspects of this work, however, come from an intersection of three main topics: 360°video displayed in VR, eye tracking in VR, and VR medical simulators.

2.1 360°Video in VR

360°video displaying in VR is a somewhat new use case for VR. Due to the lightweight portability that HMDs provide, VR is seeing much more prevalent use in research, industry, and in general entertainment. This increase of use means that streaming 360°video, rather than having the user store video on their machine, is a much needed addition to make VR more dynamic. However, streaming 360°video in high resolution is not an easy task due to the fact that these videos contain much more data than a standard video stream. Much research has gone into fixing this problem, but more recent research uses neural networks, and other machine learning techniques, to predict where a user might fixate next inside of a 360°video[3]. This technique is very useful in limiting the total amount of information that needs to be rendered for any given 360°video, but it requires a significant amount of user data on a wide diversity of videos to train. This problem is currently being investigated, but a useful step in the right direction is the creation of datasets containing user eye tracking data on specific 360°videos[8]. This data set captures both sensor data and content data. Sensor data are pieces of data collected from the HMD sensors, like head position and orientation. Content data are pieces of data collected from the video, like saliency maps and motion maps.

Perhaps one of the largest innovations for 360°video streaming in VR is within the advent of tiling. Tiling is the process of breaking up specific sections, or tiles, of a 360°video and streaming only tiles that are relevant, or close to relevant, to the user. This means only the tiles of the video that are being currently viewed by the user, or about to be viewed by the user, are streamed to the user. Two problems exist within this approach, one lies within the ability to know what the user might want to view next, and the other problem is in how small the tiles are in each approach. An approach that aims to fix both of these problems is within an optimization framework for cellular networks[10].

2.2 Eye Tracking in VR

Eye tracking in VR is yet another new use case for VR. With the integration of eye tracking built directly into commercially available, though expensive, HMDs[6], researchers have more access to eye tracking technologies for VR than ever before. With more researchers getting their hands on this new technology, an introduction to this field is very prudent. The authors of [2] have done just that. Their work shows valuable insights into the creation of eye tracking applications for VR and how to deal with certain inherent aspects of VR, like motion sickness and fatigue. They also ran a pilot study for an example application. All of this culminates in a very helpful introduction into research for this field.

Another application to integrate eye tracking into VR is in facial expression classification[4]. This work uses a convolutional neural network (CNN) to tie a user's expression to a perceived emotion, through the use of images collected from the HMD's eye tracking cameras. Interestingly, this work shows that using an emotionless image from each user serves as a good baseline for the CNN. Overall, they found their approach to be about 70% accurate, a great step towards realizing facial expressions into VR for greater immersion in multi-user applications.

2.3 VR Medical Simulators

VR medical simulators are vast in terms of their diversity. Some simulators focus on specific surgeries or type of surgery, like ArthroSim[13], a VR knee and shoulder arthroscopy simulator. Some simulators are general purpose, like ORamaVR[9], a general purpose VR surgical training simulator. These simulators are vast and often times very specific, but researchers have found that the visual realism, interaction realism, and tactile realism are lacking in most trainers [11]. The interaction and tactile realism aspects of this problem can be clearly shown with the technology used for each simulator, more specifically, if the simulator uses a commercial HMD or not.

HMDs are general purpose and very good at specific types of interaction; however, they do not excel at certain aspects of what surgical simulators require, which are dexterity and feedback. Most HMDs use rather bulky controllers, due to the need for greater accuracy inside of VR. This is an issue for surgical training due to the decrease in dexterity the controllers provide. These controllers are also very different in shape, size, feel, and weight than any tool a surgeon might need to use, causing a disconnect from what they are trying to train for. The controllers themselves also only have haptic feedback, which can't give meaningful feedback to the user. Haptic feedback does not provide the proper resistance or feel that surgeons can expect to work with, making a further disconnect from interaction realism and tactile realism. Lastly, visual realism often suffers due to a lack of diversity in the development team for these applications. Most researchers are not proficient in 3D modeling, so visual realism suffers in a large amount of these simulators, something 360°video and techniques like photogrammetry aim to improve.

3 Implementation

3.1 User Study

The user study for this application consisted of male and female students practicing to become nurses and doctors. Each user used the application twice. The first time was before they received training on patient handoffs and the second time was after they received training.

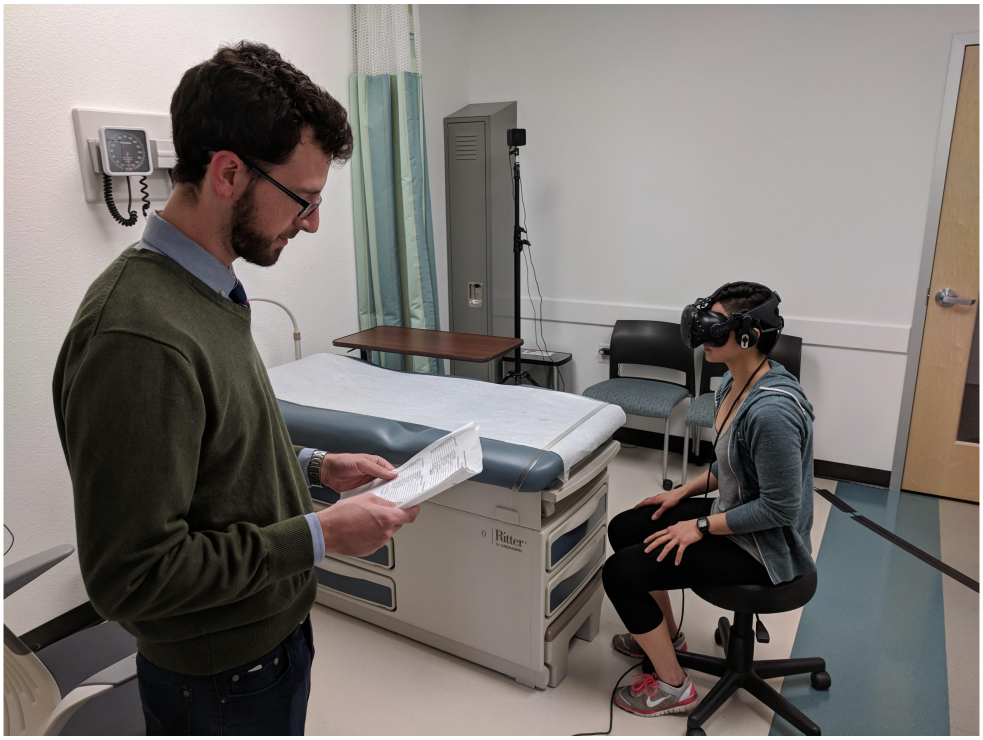

Figure 1: A picture taken of a participant while being given pre-written instructions.

During both rounds, each user was ushered into a standard patient room where they would sit down in a chair and be given pre-written instructions, as seen in Figure 1.

Once the instructions were read, the user put on the HMD with the help of the authors and followed the instructions. The instructions described the process of calibrating eye tracking and how the user should interact with the application. The user was required to watch the 360°video and to respond to what was said in the video verbally. During this time, the person giving the instructions stayed in the room to address any concerns the user might have and to assist the user in getting out of the HMD. The user was also recorded and observed during their entire participation inside of the standardized patient rooms, as seen in Figure 2. For more information on this user study, and the findings from it, please see Anbro *et al.*[1].

3.2 Eye Tracking Data Visualization

Eye tracking data was gathered using a beta TOBII SRanipal SDK for Unity. The data collected was the participant ID (the ID is just the order in which the users participated), a boolean stating if the user is a nursing student or a medical student, and a list of timestamps indicating when the user looked at a given area of interest (AOI). This data was then evaluated for the time of first fixation for each AOI and how long their attention stayed on the AOI. This occurred throughout the scenario - before, during, and after there was a “crisis” with the patient/mannequin. The data was also evaluated for total number of fixations and total time fixated throughout the video on each AOI.

All of the data evaluated was then combined into a timeline for each AOI fixation, as seen

Figure 2: A picture of the standardized patient room control station.

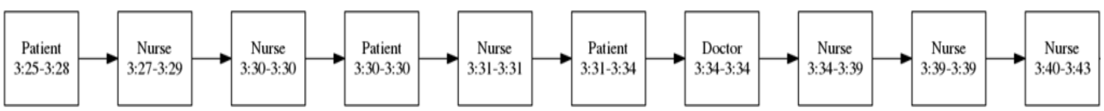

Figure 3: A small section of visualized eye tracking data notating when a given user looked at specific AOIs and for how long.

in Figure 3. This figure shows each time a user fixated on a different AOI. It is important to note that each element of this figure is of a new fixation for each AOI and when in the video it happened. When an element is followed by another element of the same name with a gap in time between them, it notates either a drop in fixation, or that the user was fixating on something that was not labeled as an AOI. When an element shares it's fixation time with another element, it indicates that both AOIs were being fixated on, meaning they shared space with one another during the video. The reason we allowed AOIs to conflict in timestamp is due to the fact that it is not clear which AOI the user is trying to fixate on at a time. For example, a user might be focusing on the patient, but during the "crisis" event the nurse and doctor share the same space as where the patient's AOI would be, so the user would end up focusing on their AOIs instead. The user in this situation is still fixating on the patient, or at least that was their goal, but they are now also fixating on the doctor or nurse. Including all timestamps for all AOIs solves this issue of conflicting AOIs and it gives researchers using this data for their evaluation a better grasp on what happened at any given data point.

4 Experimental Setup

The virtual room needed to accomplish two main goals. The first was to play 360°video without any defects or distortions. The second was to allow eye tracking measurements to be taken at any point in the video feed. Different iterations of the virtual room were then developed to accomplish these goals, and correct weaknesses in previous iterations.

4.1 Skybox

The skybox was the first iteration of the virtual room. A skybox is a default screen for most game development platforms, and since this project used Unity[14] for it's development platform a skybox was easy to create. A skybox is a rendering of a 3D pseudo-infinite space. This space is similar in look to a sky that goes on forever with no ground under it. This skybox surrounds the user constantly, even after moving. At this point in the development process, the authors didn't have the beta eye tracker from TOBII[12] or the SRanipal Unity SDK[5] (which was still in development throughout this study), but did have 360°video to render, so we used the skybox to render the video instead of the infinite space. This displayed the video accurately and with no noticeable defects or distortions. This was done by continuously updating the skybox with the next frame in the video. However, once the authors received the eye tracker and integrated the eye tracking API into the application, we realised the skybox wouldn't work as the display medium due to how the eye tracking API works.

4.2 Eye Tracking

The TOBII eye tracking API took measurements from an updating eye tracker on the inside of the Vive HMD [7]. These measurements would be either the angle towards the object the user is focusing on from the user's virtual HMD or a null measurement, meaning either nobody is in the HMD to look or the user is not fixating on anything. This angle measurement would then need to be raycast from the middle of the virtual headset in the direction of the angle returned by the HMD. This raycast would then need to collide with an object that is looking for raycasts to hit it, which then calls a customizable script. This script was used to write the time the object was hit into a file for each participant, and was later used for data analysis. Due to the fact that skyboxes do not interact with collisions from raycasts, the skybox would not work with eye tracking.

4.3 Box

The second iteration of the virtual room was a hollow box. This box is an object in the unity space that can interact with the raycasts at each of it's sides. The user's avatar was put in the center of the box and a video renderer was added to it's inner sides. The same script to render the 360°video onto the skybox was used on the regular box, as a skybox is similar to a box with a video renderer on the inside. This made the 360°video render smoothly, with no defects or distortions, and allowed the eye tracker to work on the 360°video. But, the box itself is a single collider, this means that the raycasts sent from the eye tracker will always result in the box being hit. A method was needed to correlate what the user is looking at in the video. This is where AOIs come in.

Areas of interest (AOI) are locations inside of a video that are relevant to the audience of a video. AOIs are generally objects of some relevance or individual people. For this application the relevant AOIs were the doctor, nurse, patient, monitor, and table as seen in Figure 4. These

Figure 4: A screenshot of the 360°video playing for each participant

AOIs were identified by the authors, and then we created invisible boxes that would move if the correlating AOI in the video moved. This means that when the nurse moved, so did her corresponding AOI.

The way we made the AOIs move was by manually shifting the coordinates of the AOI in the Unity world space everytime the AOI shifted in the video. We recorded the coordinates, and the correlating time in the video, to then have the shifts be repeated automatically during the simulation. These manual shifts mean that the AOI measurements are very close approximations to the ones in the video, but not perfect.

The authors also noticed an error in the design of the box as a display medium for eye tracking. Any eye tracking coordinates obtained from the raycasting on the background, the box itself, would be incorrect. This is due to the recording format of the 360°video being a sphere.

4.4 Sphere

The final iteration of the virtual space was the sphere. The sphere was very similar to the box iteration, but it solved the issue of displaying a sphere inside of a box. This issue is created in the corners of the box, where the coordinates when raycast to from the center of the box would be deeper than they would be in a sphere, making them inaccurate.

5 Conclusions

This work created an application to be used in patient handoff training that integrated 360°video within VR. This application also mapped 360°video with a user's eye tracking data in real time to be used in evaluating training methods for the patient handoff. The application is novel due to the intersection of three techniques; VR, 360°video rendering, and eye tracking. The application itself was built using a beta version of the TOBII SRanipal SDK for eye tracking. This SDK, and the TOBII eye tracking hardware, are now integrated natively with the HTC

VIVE Pro Eye HMD. This work also provides details as to what worked, and did not work, within the VR space to allow for eye tracking to be used effectively.

6 Future Work

There are several things that can be looked at to improve work in this area. This section presents several of the areas we think would be beneficial for future researchers to consider.

AOI Tracking: AOIs were created and moved manually. The AOIs were paired with a list of locations and timestamps for this specific video. Using computer vision that paired AOIs to their associated person or object for any video would limit the manual labor needed to recreate this application with a different video.

Heat Map: A heat map for where a user was looking would be a great addition to this project. The heat map would allow another visual metric to be displayed and evaluated by researchers. This heat map could be static and evaluate overall fixation by the user, but it could also show the fixation over time for any given video during a playback. Using these heat maps and comparing them to other user heat maps can show a large amount of metrics and show aggregate data useful in research and marketing.

Virtual Environment: A virtual environment for patient handoffs would be a good change for this project. Allowing the user to move around and interact with things inside of this virtual environment would allow for deeper immersion and greater use of VR technologies. This change could allow many different scenarios to play out, not just patient handoff. It also opens up the possibility of having users interact in a multiplayer and collaborative environment which might further their knowledge and experience with skills like the patient handoff.

Streaming Video: To take full advantage of the VR format for medical training, it would prove useful to stream video rather than host it on a user's local machine. With AOI tracking established, it is feasible to create many 360°videos that allow eye tracking data to be collected. To allow this broader goal to be reached, a new UI for selecting videos/trainings would need to be established and the list for selecting these videos/trainings would need to be updated from a server with new videos/trainings when they get released. Much research has been done on improving the large resource load that streaming 360°video requires [3, 10], some of this research must be implemented to allow for streaming video for medical training to be feasible in a broader scope.

7 Acknowledgements

We would like to acknowledge the UNR School of Medicine and the Orvis School of Nursing for their collaboration and for having their students participate.

This material is based in part upon work supported by the National Science Foundation under grant numbers 2019609 and IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Steven J. Anbro, Alison J. Szarko, Ramona A. Houmanfar, Amber M. Maraccini, Laura H. Crosswell, Frederick C. Harris, Michelle Rebaleati, and Luka Starmer. Using virtual simulations to assess situational awareness and communication in medical and nursing education: A technical feasibility study. *Journal of Organizational Behavior Management*, 40(1-2):129–139, April 2020.

[2] Viviane Clay, Peter König, and Sabine U. König. Eye tracking in virtual reality. *Journal of Eye Movement Research*, 12(1), April 2019. https://doi.org/10.16910/jemr.12.1.3.

[3] Ching-Ling Fan, Jean Lee, Wen-Chih Lo, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. Fixation prediction for 360° video streaming in head-mounted virtual reality. In *Proceedings of the 27th Workshop on Network and Operating Systems Support for Digital Audio and Video*. ACM, June 2017.

[4] Steven Hickson, Nick Dufour, Avneesh Sud, Vivek Kwatra, and Irfan Essa. Eyemotion: Classifying facial expressions in VR using eye-tracking cameras. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, January 2019.

[5] HTC Corporation. Documentation - Developer Resources: SRanipal Unity SDK for the HTC Vive Pro Eye. [online]. https://developer-express.vive.com/resources/vive-sense/eye-and-facial-tracking-sdk/documentation/ (Last accessed 28 November 2021).

[6] HTC Corporation. VIVE Pro Eye Overview: VIVE United States. [online]. https://www.vive.com/us/product/vive-pro-eye/overview/ (Last accessed 30 November 2021).

[7] HTC Corporation. VIVE United States: Next-level VR Headsets and Apps. [online]. https://www.vive.com/us/ (Last accessed 27 November 2021).

[8] Wen-Chih Lo, Ching-Ling Fan, Jean Lee, Chun-Ying Huang, Kuan-Ta Chen, and Cheng-Hsin Hsu. 360° video viewing dataset in head-mounted virtual reality. In *Proceedings of the 8th ACM on Multimedia Systems Conference*. ACM, June 2017.

[9] ORamaVR. ORamaVR - the world's most intelligent VR medical training simulations. [online]. https://oramavr.com/ (Last accessed 15 October 2021).

[10] Feng Qian, Lusheng Ji, Bo Han, and Vijay Gopalakrishnan. Optimizing 360 video delivery over cellular networks. In *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*. ACM, October 2016.

[11] G S Ruthenbeck and K J Reynolds. Virtual reality for medical training: the state-of-the-art. *Journal of Simulation*, 9(1):16–26, February 2015.

[12] Tobii AB. Providing technologies & solutions for a better world - Tobii. [online]. https://tobii.com/ (Last accessed 27 November 2021).

[13] Touch of Life Technologies Inc. TolTech - ArthroSim Arthroscopy Simulator. [online]. https://www.toltech.net/medical-simulators/products/arthrosim-arthroscopy-simulator (Last accessed 15 October 2021.

[14] Unity Technologies. Unity real-time development platform: 3D, 2D VR & AR engine. [online]. https://unity.com/ (Last accessed 27 November 2021).