

CLOVR: Collecting and Logging OpenVR Data from SteamVR Applications

Esteban Segarra Martinez*
University of Central Florida

Ayesha A. Malik†
University of Central Florida

Ryan P. McMahan‡
University of Central Florida

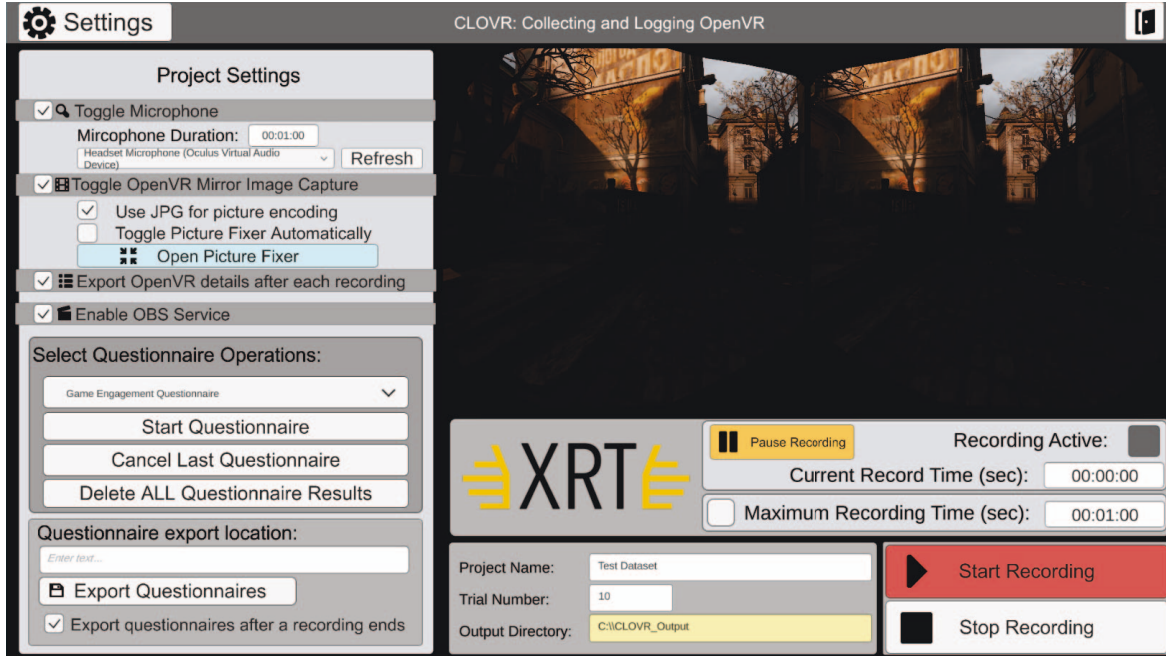


Figure 1: CLOVR is an easy-to-use tool with an interface that can be used to collect VR interaction data from any OpenVR based application combined with an easy to use GUI for experienced or novice users.

ABSTRACT

Due to the growing popularity of consumer virtual reality (VR) systems and applications, researchers have been investigating how tracking and interaction data from VR applications can be used for a wide variety of purposes, including user authentication, predicting cybersickness, and estimating cognitive processing capabilities. In many cases, researchers have to develop their own VR applications to collect such data. In some cases, prior researchers have provided open datasets from their own custom VR applications. In this paper, we present CLOVR, a tool for Capturing and Logging OpenVR data from any VR application built with the OpenVR API, including closed-source consumer VR games and experiences. CLOVR provides an easy-to-use interface for collecting interaction data from OpenVR-based applications. It supports capturing and logging VR device poses, VR actions, microphone audio, VR views, VR videos, and in-VR questionnaires. To demonstrate CLOVR’s capabilities, we also present six datasets of a single user experiencing six different closed-source SteamVR applications.

Index Terms: Human-centered computing—Systems and tools for interaction design; Human-centered computing—Interactive systems and tools Computing methodologies—Virtual reality

*e-mail: esteban.segarra@ucf.edu

†e-mail: ay070076@ucf.edu

‡e-mail: rpm@ucf.edu

1 INTRODUCTION

Since the introduction of consumer virtual reality (VR) systems, researchers have investigated how tracking and input data from VR applications can be leveraged for a wide variety of purposes. Numerous researchers have investigated using VR interaction data for authenticating and identifying VR users [17, 26, 30, 32, 40]. Several researchers have also investigated using VR interaction data to predict cybersickness [13–15]. Finally, some researchers have investigated VR interaction data for predicting cognitive processes, such as mental workload [41], knowledge acquisition [31], and skills retention [33].

In most cases, researchers have had to develop their own VR applications to collect the tracking and input data necessary for their respective investigations. This approach is often time consuming, due to developing the required underlying VR application and data collection process [16]. A number of open-source tools have been developed to help expedite the development process, particularly the collection of VR interaction data [46, 49]. However, these tools do not expedite the actual data collection process of conducting participants through studies to collect the data.

In recent years, researchers have begun openly sharing datasets of VR interactions to alleviate the time required to collect one’s own dataset and to facilitate the generalization of knowledge across multiple VR applications. Most of these datasets are collected from custom VR applications and scenarios, such as viewing moving targets [23], shooting an arrow [21], throwing a ball [28], and assembling structures [30]. Some datasets have been collected from viewing 360° images or videos [4, 8, 42]. Most recently, researchers have collected and shared data from closed-source, consumer VR

applications, such as *Beat Saber* [35] and *Half-Life: Alyx* [39], by leveraging open application programming interfaces (APIs), such as Valve’s *OpenVR* API.

In this paper, we present *CLOVR*¹—a tool for *Capturing and Logging OpenVR* data from VR applications that utilize the *SteamVR* plugin, including closed-source experiences like *VRChat* and *Rec Room*. By facilitating the capture of VR interaction data from pre-existing, closed-source applications, CLOVR helps researchers avoid the time required to implement their own custom VR applications while also providing critical data on consumer VR experiences. CLOVR automatically records the positions and rotations of every tracked VR device (even additional HTC Vive trackers), every input action like button presses or headset doffing events, the binocular images rendered to the VR headset, and even the VR headset’s microphone. CLOVR also supports capturing high frame-rate, non-distorted videos of the VR perspective by leveraging *OBS Studio* to directly record the SteamVR window.

In addition to capturing VR interaction data from any OpenVR-based application, CLOVR also supports the administration of questionnaires directly within the VR experience, even closed-source experiences like *Beat Saber* and *Half-Life: Alyx*. In-VR questionnaires have been demonstrated to collect comparable results to their out-of-VR counterparts and to be more enjoyable [1]. With CLOVR, researchers can now administer questionnaires like the Simulator Sickness Questionnaire (SSQ) [19], the Game Engagement Questionnaire (GEQ) [3], the Spatial Presence Experience Scale (SPES) [11], or the Preliminary Embodiment Short Questionnaire (pESQ) [7] directly in VR for any OpenVR-based application. In addition to administering standardized questionnaires like the SSQ and GEQ, researchers can also administer their own questionnaires by using CLOVR’s XML questionnaire format.

After discussing the design and capabilities of CLOVR, we also present six datasets of a single user experiencing six different closed-source, OpenVR-based applications², including *The Lab*, *VRChat*, *Rec Room*, *Beat Saber*, *Half-Life: Alyx*, and *Boneworks*. These datasets demonstrate the broad range of consumer VR applications that interaction data can be captured from with CLOVR. Furthermore, these single-user datasets are expected to help advance the state of the art with regard to identifying individual VR users across applications, which has been underexplored.

2 RELATED WORK

In this section, we discuss existing tools for capturing VR interaction data, currently available VR datasets, and their prior applications.

2.1 Existing VR Recording Tools

OpenVR Recorder [2] is a proprietary and close-source tool used to record the VR equipment poses from any connected device. It’s only use is to collect motion data and does not record button presses or interactions with the controller. *Virtual Reality Scientific Toolkit (VRSTK)* [12] presents questionnaires to the user, however it has to be incorporated directly into the source code of a Unity application, and does not incorporate video or pictorial data capture. *VRHook* [48] is another more direct approach to capturing VR motion data, which offers the ability to hook into a VR application using dynamic hooking to interface with a runtime application and capture poses, images from the HMD, and interaction actions of controllers. However, it has to directly inject its code into runtime memory to acquire the information it has to record. This presents an issue with VR applications that use anti-cheating software that prevent memory access.

RealityCheck is another project that utilized the OpenVR API to inject data into the HMD in order to pass-through and overlay picture

¹<https://github.com/xrtlab/clovr>

²<https://doi.org/10.17605/OSF.IO/GUF7X>

Dataset Name	Year Published	Public?
Behavioral Biometrics [37]	2019	No
TTI [27]	2020	Upon request
Body Normalization [22]	2021	Yes
Obfuscation [32]	2021	No
Body Sway [5]	2021	Yes
You Can’t Hide [45]	2022	No
Motion Matching [38]	2022	Yes
Personal Identifiability [26]	2023	Upon request
Who is Alyx? [39]	2023	Yes
BOXRR-23 [35]	2023	Yes
VR.Net [47]	2023	Yes

Table 1: Table of current open-source datasets.

data from real world cameras [10]. Finally, *DreamStream* [44] was an evolution of this interface and improved the scene injection by being able to collect the graphical qualities of an active VR scenario such as RGB, depth, and also injecting Kinect depth/camera data. Neither *RealityCheck* or *DreamStream* have publicly available releases for use.

2.2 Current VR Datasets

Several VR datasets exist as public-domain and machine learning ready to be used for training, ground truth, or identification purposes. Table 1 shows a selection of VR-related datasets.

As seen in table 1, there is a diverse application to the use of the collected datasets and also how they are being applied. These range from subject identification [26, 27, 32, 37, 38, 45], general datasets [35, 39], and environmental effect studies in VR [5]. Subject identification is the process of precisely identifying a subject based on multiple characteristics of the individual, which can include motion patterns [38], behavioural biometrics [37], skeletal poses, and/or the approaches of individuals to tasks [45]. Individual characteristics that can be identified from this data include sex, age, and personally identifiable behaviors.

General motion capture datasets include *Who is Alyx?* [39], which observes users interacting in two 50 minute scenarios of the popular video game *Half-Life Alyx*. Another dataset [35] *BOXRR-23*, was collected from the popular video games *Beat Saber* and *Tilt Brush*, and includes 100000 participants with a diverse span of personal characteristics. *VR.net* [47] is a dataset collected using *VRHook* [48] compromising data from 10 video games, which includes RGB data, motion flow images, depth output, pose, interaction, scene, camera, lights information, and self-reported user surveys.

2.3 Applications of VR Datasets

Researchers have multiple applications for VR datasets, particularly for applications such as gesture recognition [28], virtual experience exposure [17], tracking-based ML learning [31], or motion-based identification [40]. More data-heavy applications may be utilized for applications such as artificially generating animations from generative adversarial networks [20] or GAN-enhanced autoencoders [29]. As presented in one paper, video game avatars can be produced with generative AI models [50], showing one potential avenue for demand of VR motion capture data. The impact of latency and cybersickness is also of concern, as shown in Stauffert et al. [43], where latency can lead to individuals to suffer cybersickness.

VR pose data can be used to identify users as detailed in one paper [32], where users can be identified and authenticated using velocity pose records. A survey paper by Lin & Latoschik [24] describes how VR data could be a vector to expose user’s identities and privacy and demonstrates how this is being addressed in the current literature. Another paper detailed the use of pose data in order to predict learning gains from a learning session [33]. Another

paper [9] highlighted the use of VR hand motion to predict hand trajectories to predict ballistic hand motions in VR. Other notable machine learning techniques include DeepGRU [25], embedded GRU models [40], or convolutional neural networks [18].

3 CLOVR: AN OPEN-SOURCE DATA COLLECTION TOOL

CLOVR is an easy-to-use tool and interface (see Figure 1) for collecting VR interaction data from any OpenVR-based VR application, including close-source games like *Beat Saber* and *Half-Life: Alyx*. OpenVR is Valve’s API for developing VR applications for *SteamVR*. We developed CLOVR using the Unity game engine (2021.3.30f) and open sourced the tool to enable other researchers to easily capture and record data directly from OpenVR-based consumer games and experiences. Below, we discuss the system architecture of CLOVR and its key features.

3.1 CLOVR System Architecture

Figure 2 depicts the interaction of the different components that comprise CLOVR, including the interaction of external applications such as OBS and OpenVR. It also describes how data is collected, where source information is being drawn from in OpenVR, and how is CLOVR interfaces with them. The details of each key feature are described in the following sections.

3.2 Capturing and Logging Pose Data of VR Devices

As discussed, the poses (i.e., positional and rotational data) of tracked VR devices can be leveraged for a wide variety of purposes, including the authentication and identification of users [17, 30, 40], the prediction of cybersickness [13, 14], and the prediction of cognitive processes [31, 33, 41].

CLOVR uses the `GetDeviceToAbsoluteTrackingPose` function of OpenVR to obtain positional and rotational data for each connected VR device, such as headsets, controllers, base stations, and additional devices like Vive trackers. For each device, CLOVR captures the device type (e.g., “HMD 0”, “Controller LeftHand”, “TrackingReference 1”), its absolute position as a 3D vector, its absolute rotation as a 4D quaternion, its linear velocity as a 3D vector, and its angular velocity as a 3D vector. This pose data is collected at the refresh rate of the headset or at a default, user adjustable, rate of 72 frames per second (fps) if the headset does not specify a refresh rate. Furthermore, CLOVR currently assumes the captured VR application uses a standing tracking space, but are planning to add support for seated tracking in the near future and support whichever environment the *SteamVR* runtime is set to.

Temporal synchronisation between CLOVR’s own pose and interaction data is guaranteed since the OpenVR data is originating from the *SteamVR* runtime. However, we cannot guarantee temporal synchronization between CLOVR and a target application since CLOVR cannot directly obtain poses and interactions from the target application.

CLOVR logs the captured pose data of the VR devices to a file named *Poses_and_Interactions_YYYY-MM-DD-HH-MM-SS-MS.csv*, where *YYYY-MM-DD-HH-MM-SS-MS* represents the exact time that the recording began. This naming convention avoids the potential overwriting of data and facilitates researchers in determining when a VR experience was captured. We chose to use a comma-separated value (CSV) file format, as opposed to recent alternatives (e.g., the XR Open Recording format [35]), because many statistical and machine learning packages support importing CSV data. Additionally, the CSV file format is human-readable, which makes it easy to work with. For each tracked VR device, CLOVR appends a new set of columns to the CSV file corresponding to the captured properties of that device, including “Device”, “Position X”, “Position Y”, “Position Z”, “Rotation X”, “Rotation Y”, “Rotation Z”, “Rotation W”, “Velocity X”, “Velocity Y”, “Velocity Z”, “Ang. Velocity X”, “Ang. Velocity Y”, and “Ang. Velocity Z”.

3.3 Capturing and Logging VR Actions

In general, most researchers have focused on leveraging VR poses for their purposes. Hence, VR actions, such as controller button presses, are an underutilized set of features. However, research by Nair et al. [34] demonstrated that contextual features, such as the slashing of specific *Beat Saber* blocks, can be used as input features to improve machine learning models.

Given an OpenVR JSON action manifest file (e.g., `default.bindings`), CLOVR binds the provided list of actions to actual VR input components and uses OpenVR’s `UpdateActionState` function to capture the current state of all the actions. CLOVR then uses OpenVR’s `GetDigitalActionData` function to read the states of boolean actions (e.g., a button) and the `GetAnalogActionData` function to read the states of vector-based actions (e.g., a trigger, a joystick).

CLOVR logs the captured VR actions to the same file as the VR pose data (i.e., *Poses_and_Interactions_YYYY-MM-DD-HH-MM-SS-MS.csv*). For each VR action, CLOVR appends a new set of columns to the CSV file corresponding to the captured state of the action. For boolean actions (e.g., a button), these columns include the name of the action, the input source (e.g., the left or right controller), what its current state is, and whether its state has changed. For vector-based actions (e.g., a trigger, a joystick), these columns also include the name of the action and the input source, but also include “X”, “Y”, “Z” values and their deltas (i.e., “dX”, “dY”, “dZ”).

3.4 Capturing and Recording Microphone Audio

VR users may talk about or verbally respond to their experiences when immersed, particularly during moments of extreme emotions like excitement or fear. In some cases, researchers may explicitly ask VR participants to think aloud [36] or emotive aloud [6] to reveal insights about their cognitive and affective states, respectively. In short, recording the speech and utterances of VR users may be extremely useful to researchers.

CLOVR uses Unity’s `Microphone` class to capture audio clips from an available device. Captured audio clips are then exported as Waveform Audio File (WAV) files using the naming convention *Audio_Recording_YYYY-MM-DD-HH-MM-SS-MS.wav*. By default, the system’s default microphone will be recorded, but the CLOVR interface allows the researcher to specify which microphone to record from to ensure the VR headset’s microphone is selected. Temporal synchronization is achieved as this system is integrated directly into CLOVR and records alongside pose and interactions.

3.5 Capturing and Recording VR Views

The user’s view of the VR experience is another underutilized feature in prior work. However, what the user is seeing obviously can have significant effects of the user’s VR experience. Some researchers have considered the user’s VR view [13], but this rich perceptual data should be investigated more in the near future.

CLOVR utilizes OpenVR’s `GetMirrorTextureD3D11` function to obtain a native pointer to the texture rendered to each eye and then uses the `AsyncGPUReadback` function to read in the current texture (i.e., the current VR view), display it within the interface (see Figure 1), and save the texture data for both the left and right eyes. By default, CLOVR saves the 2D texture data in JPG files by leveraging Unity’s `ImageConversion.EncodeToJPG` function. However, researchers that desire lossless images can choose to save the texture data in PNG files, which leverages Unity’s `ImageConversion.EncodeToPNG` function. By default, CLOVR records these VR views for both the left and right eyes once per second (i.e., 1 fps) due to the substantial computations required to encode and write the images to disk.

Images obtained from OpenVR are HMD-corrected, which means they are vertically mirrored (i.e., the top left of the view appears at the bottom right) and the picture brightness is lowered. To correct

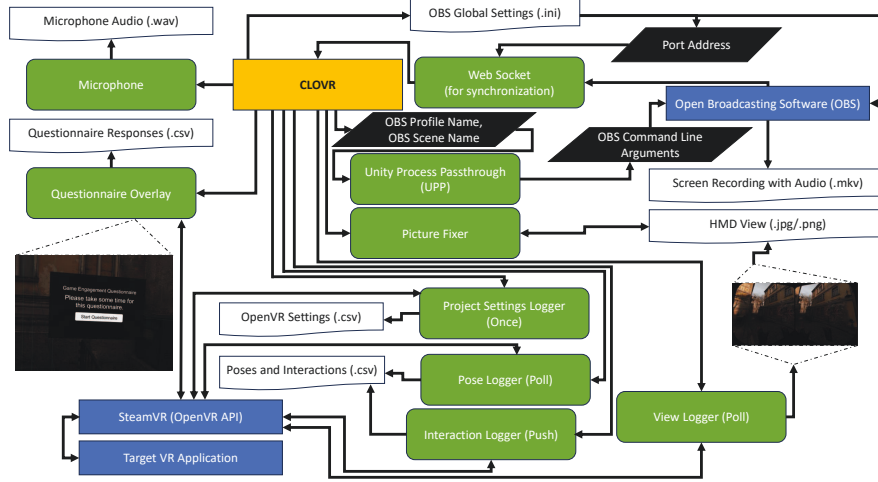


Figure 2: System architecture of CLOVR.

this inversion problem, we provide a “Picture Fixer” utility built into CLOVR. This utility takes in all of the pictures that were generated for a prior recording session or in a location provided by the researcher and inverts them by first linearly inverting the picture vertically and then linearly inverting the picture horizontally. This process is slow and recommended to be applied after a recording session has been completed. Researchers have the option to “Open Picture Fixer” via the CLOVR interface (see Figure 1). We expect to fix the HMD-corrected image problem in a future update.

3.6 Capturing and Recording VR Videos

While CLOVR can directly capture and record VR views from the OpenVR API, we found this process too computationally expensive to record videos of VR interactions. In particular, converting the raw textures from OpenVR to JPG or PNG file format introduces a substantial amount of computation. As a result, CLOVR is not capable of capturing VR views at the same frame rate as the headset’s refresh rate. Furthermore, we were unable to find an open-source solution for capturing in-game audio, particularly from closed-source VR applications and games.

In order to seamlessly record videos with in-game audio from closed-source VR applications, we incorporated the ability to call the *Open Broadcast Software (OBS) Studio* from CLOVR. OBS Studio is a free, open-source application for video recording and live streaming screens and other applications³. In addition to optimized screen recordings, OBS Studio supports several custom settings, such as selecting which application or scene to capture, what input sources to record, how to mix the system and microphone audio channels, and other settings like the video codec, audio bitrate, and multipass mode to use.

In order to call OBS Studio from CLOVR, we developed a custom dynamic-link library (DLL) called *Unity Process Passthrough (UPP)*, which can launch external applications and remove their processes when requested by the Unity application. This library is used to start and destroy the OBS Studio executable when CLOVR respectively starts and stops recording. This ensures that OBS Studio can cleanly start and stop when requested by CLOVR, avoiding the potential of a rogue process indefinitely recording in the background, which we have observed when attempting to use the OBS API instead of calling OBS Studio. Note, UPP’s source code is available along with CLOVR.

When OBS is enabled within CLOVR and an experimenter starts recording, UPP is used to call OBS Studio with a desired application and recording profile. By default, CLOVR will attempt to change the recording profile of OBS to record the SteamVR application. The name of the recorded video will be `yyyy-mm-dd-hh-mm-ss-ms.mkv`. Videos will only be as long as the start and stop of the recording session. Experimenters can adjust the recorded view through OBS if they are choosing to record through SteamVR’s *VR View*, Oculus’s *Oculus Mirror* VR viewer, or a custom application that they desire to record. This change may be necessary in certain situations where a Meta headset is used and SteamVR’s *VR View* does not display a mirrored view. Temporal synchronization is approximately guaranteed between CLOVR and OBS by connecting a websocket connection between the two applications and starting the OBS recording only after the connection is successful and closed. A delay between the two applications in the magnitude of microseconds can be expected.

3.7 Administering and Logging In-VR Questionnaires

Questionnaires have long been used to gather information about VR experiences and compare them [16]. Recent research indicates that questionnaires administered within VR (i.e., in-VR questionnaires) are comparable to traditional questionnaires administered outside of VR but are more enjoyable [1]. Hence, the capability of administering in-VR questionnaires, especially validated and standard ones, is valuable to the VR research community.

CLOVR provides the capability to dynamically start a selected questionnaire from its interface (see Figure 1) to display within the VR experience as an OpenVR overlay (see Figure 3). For a selected questionnaire, each question and its anchors are displayed one at a time. As an OpenVR overlay, the VR user is able to use a virtual pointer to select choices for each question. Temporal synchronization is not guaranteed in our current questionnaire system at this time, however all questionnaires will be stored to their respective recording session.

CLOVR automatically captures and logs selected questionnaire responses to a CSV file named `<questionnaire name>-yyyy-mm-dd-hh-mm-ss-ms.csv`. Questionnaires are exported using the list of responses as the header and selected responses are appended below the header. Questionnaires can be exported at the end of each recording session or manually to a desired location.

Questionnaires to select from within the CLOVR interface are loaded through XML formatted files. By default, CLOVR comes with four preloaded questionnaires. These are standard, validated

³<https://obsproject.com>

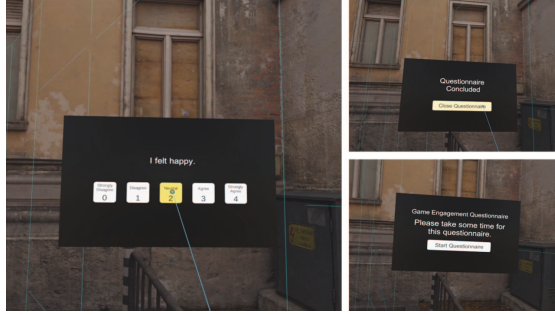


Figure 3: **Left image:** A typical question panel. **Right top image:** A panel to demonstrate end-of-questionnaire. **Right bottom image:** A panel to show the start of the questionnaire.

questionnaires and include the Simulator Sickness Questionnaire (SSQ) [19], the Game Engagement Questionnaire (GEQ) [3], the Spatial Presence Experience Scale (SPES) [11], and the Preliminary Embodiment Short Questionnaire (pESQ) [7]. Additional questionnaires can be added by adding XML files for each questionnaire.

3.8 Project Properties File

This file documents the runtime configuration of OpenVR when the researcher performed a recording. The data is sourced from here includes OpenVR details found from calling the following functions: `OpenVR.Chaperone.GetPlayAreaRect`, `OpenVR.Chaperone.GetPlayAreaSize`, `OpenVR.Applications.GetApplicationCount`, and `CVRSystem.GetRuntimeVersion`.

3.9 Extensions to CLOVR

CLOVR is designed to be easily extended by a third party developer or researcher. Extensions can be implemented to the open-source Unity project, such as the Tobii Eyetracking API. Other extensions can be incorporated for future headsets support different sensors such as electrocardiogram (ECG) meters, different eye-tracking software, brain-computer interfaces (BCI), or custom-made sensors. These extensions will benefit of the temporal synchronization between OpenVR and CLOVR.

4 PRELIMINARY RESULTS

To validate the functionality of CLOVR, we collected six datasets from six distinct close source applications from a single individual. The targeted applications were commercially available VR applications that are popular at the time of this paper, which include the following: *The Lab*, *VRChat*, *Rec Room*, *Boneworks*, *Half-Life Alyx*, and *Beat Saber*. These games were chosen in part because they were popular applications, are close source, and provide a diverse set of environments to record and interact with. To demonstrate our recording capability without direct application injections, *VRChat* uses *Easy Anti-Cheat* to prevent memory hacks.

4.1 Single-Shot Dataset

We collected a dataset with per-application datasets to demonstrate the capabilities of CLOVR in collecting OpenVR data as well as OBS video and audio. All of the targeted applications were collected on a computer with i9-10850k CPU with 128GB of DDR4 RAM and a RTX 2080 using the Oculus Quest 3 and the applications were installed locally on a 1TB Samsung 860 EVO SSD. Figure 4 shows preview captures of in-game footage from the collected applications. Figure 5 shows previews of the traced positions from each game and their respective patterns collected from the user’s HMD and controllers.

4.1.1 The Lab

This free application involves the use of interacting with different environments in a demo-like experience of VR to a novice user. The application involves the use of seven experiences in VR with an additional one being the hub area where the user can access the different locations. This hub area contains items that are placed around a central table that are unlocked as the user explores the different locations. The collected dataset contains a detailed recording of the table, one demo scene, an exploration of the hub area, and using action items on top of the table.

4.1.2 VRChat

VRChat is a social interaction application where users are allowed to use unique avatars and personas to interact with other users. This game by default logs you into a hub area where the user can select a unique avatar and walk through portals to different virtual environments. During data collection, we avoided exposing private information of users by restricting our records to the hub area of VRChat.

4.1.3 Rec Room

Rec Room is similar to VRChat in the aspect of being a social interaction VR game. Users explore different virtual environments by navigating a user menu and teleporting there. The game begins with a tutorial environment that demonstrates what users can do in Rec Room. We recorded this tutorial environment and recorded a few demo locations including a paintball environment and a dune buggy virtual environment.

4.1.4 Boneworks

Boneworks is a roleplaying game and shooter game where the user explores a virtual environment filled with different entities and interactable elements. The game starts at an introductory location where the user is taught the basics of interaction with the different aspects of the game. These include a parkour location, a climbing demo, and a demo location to interact with different objects in the environment such as picking up objects and interacting with machines. The recording focused primarily on the introductory tutorial environment.

4.1.5 Half-Life Alyx

Half-Life Alyx is a VR RPG game where users are put into a first-person experience in a story. We primarily focused to cover two chapters of the game, Chapters 1 and 3. This was done to provide a comparison point between the dataset generated using CLOVR and the *Who is Alyx?* [39] paper dataset.

4.1.6 Beat Saber

Beat Saber is a VR rhythm game where players swing light sabers at directional boxes in VR space to destroy them at the rhythm of a beat. Players play maps, or rhythm-adjusted patterns of these boxes and attempt to complete the map. The dataset generated from this game was of seven maps played with varying levels of success.

5 DISCUSSION

We provide six individual datasets from different applications and record them utilizing the all of CLOVR’s recording features. These datasets can serve different purposes, from user identification, motion studies, and pose and interaction capture. From the trials we ran, CLOVR did not have a direct impact on the runtime render of the applications, avoiding cybersickness on the trialed applications. We believe that CLOVR can serve as a free and complete alternative to current implementations as a tool for collecting pose, interaction, and questionnaires. We are also releasing the project open source and documented in order to foster interest in the research community for a tool that can be utilized for rapid data collection.

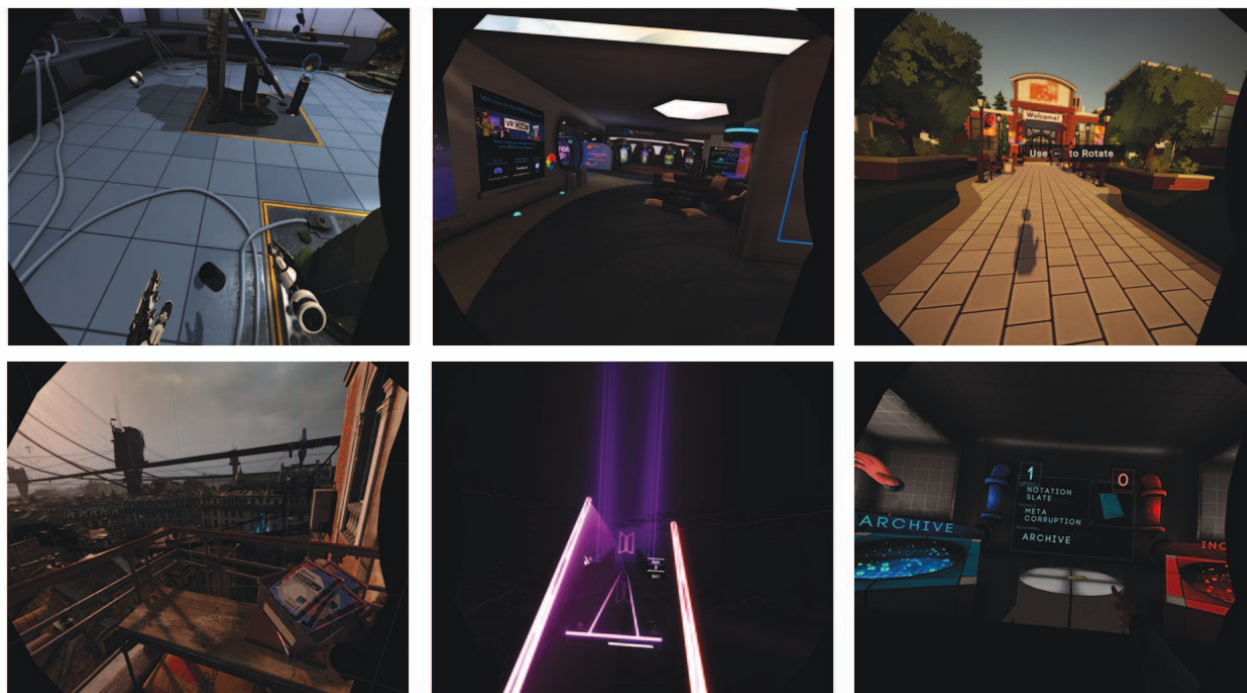


Figure 4: OpenVR image data from applications displayed from top left corner to bottom right corner: *The Lab*, *VR Chat*, *Rec Room*, *Half-Life Alyx*, *Beat Saber*, and *Boneworks* (All images as seen from the perspective of the left eye)

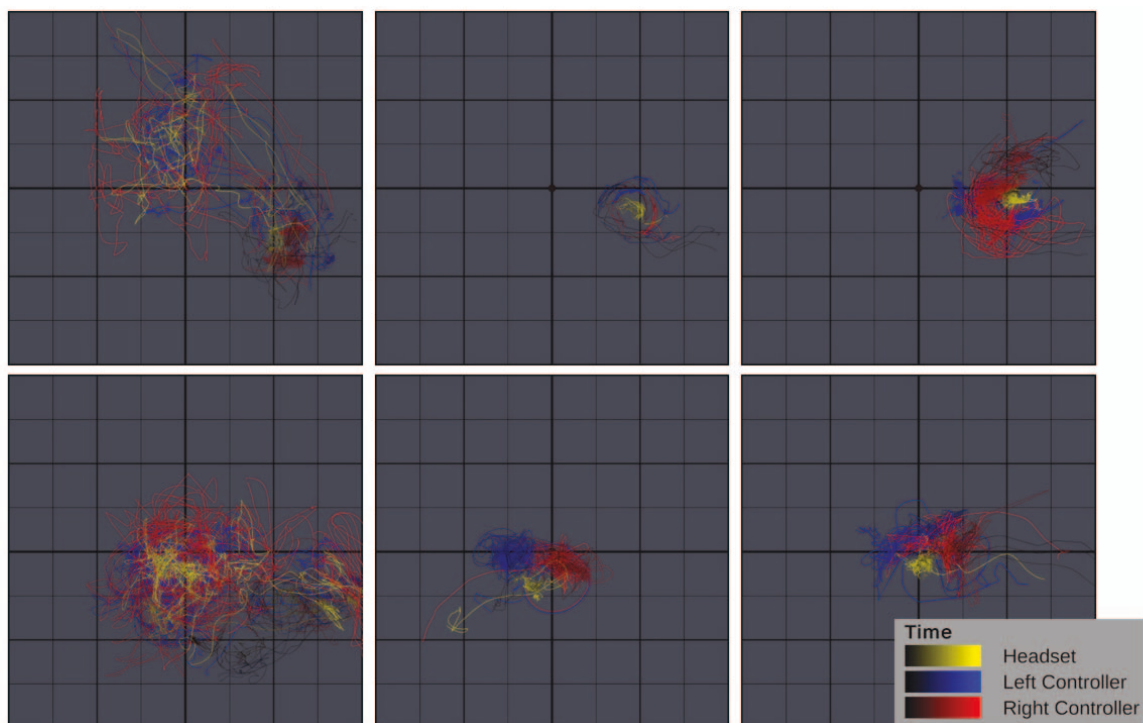


Figure 5: Motion traces from recorded applications displayed from top left corner to bottom right corner: *The Lab*, *VR Chat*, *Rec Room*, *Half-Life Alyx*, *Beat Saber*, and *Boneworks* (All images as seen from the perspective of the left eye)

5.1 Current Limitations

The provided dataset is limited to the results of one participant, limiting its applicability to general user cases. Additional limitations include the lack of guaranteed temporal synchronization between different CLOVR recording components, OpenVR as a platform becoming outdated, and the need of installing OBS onto a platform for recording video and audio. An important note is that despite the lack of temporal synchronization guarantees, CLOVR is still a useful tool for collecting OpenVR related data. CLOVR data has use cases, such as simultaneous localization and mapping (SLAM), pose estimations, gesture recognition, and general survey capturing.

5.2 Benefits of CLOVR

CLOVR offers simultaneous recording of different information sources and removes the requirement of the researcher having to create specialised tools for collecting and organizing the information. By having a graphical interface, researchers can also configure and observe the current status of the recording, present questionnaires, and configure settings related to the recording. Since CLOVR is tapping into the pose stream presented by the SteamVR and OpenVR API, data will be the same as presented to the application.

5.3 Replicability and Reproducibility

A common issue in the realm of VR datasets is that not all are public due to privacy statements or clauses that prevent a public release. Additionally, the data may not be easy to use for outside the intended researcher's applications. CLOVR addresses this by creating a platform from which VR researchers can record datasets consistently for general VR data. Researchers can also share the standardized XML files for the questionnaires that they use.

6 FUTURE WORKS

We plan to extend the capabilities of CLOVR by including additional data available such as the predictive or realtime skeleton hand information provided by OpenVR. Additionally, we also are extending the recording capabilities through more recording options such as refresh rate, reference world frame (standing, sitting, or raw and uncalibrated), a websocket service to stream OpenVR data, and the use of default SteamVR bindings. We are also working on improving the recording capability of HMD referenced images so as to mitigate the bottlenecks currently in the frame capture system.

7 CONCLUSION

We present CLOVR, an open-source tool for directly collecting pose and image data from OpenVR. The tool can additionally collect video and audio data from OBS through a built-in DLL that can enable and disable OBS recording. Because of OBS, users can select the stream of information that is being collected, whether that be directly from SteamVR VR View or Oculus Mirror VR view. Data is automatically sorted into different folders depending on the type of collected information, compartmentalizing the data to be machine learning ready. Finally, we present six different datasets collected from different applications using CLOVR.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation award #2232448 – “CCRI: Planning-C: Capturing and Logging Ecological Virtual Experiences and Reality (CLEVER)”.

REFERENCES

- [1] D. Alexandrovsky, S. Putze, M. Bonfert, S. Höffner, P. Michelmann, D. Wenig, R. Malaka, and J. D. Smiedinck. Examining design choices of questionnaires in vr user studies. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, p. 1–21. Association for Computing Machinery, 2020. doi: 10.1145/3313831.3376260
- [2] J. Brekelmans. Motion capture using vive/oculus/meta & other vr devices, 2024. Accessed on 2024-1-5.
- [3] J. H. Brockmyer, C. M. Fox, K. A. Curtiss, E. McBroom, K. M. Burkhart, and J. N. Pidruzny. The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*, 45(4):624–634, 2009. doi: 10.1016/j.jesp.2009.02.016
- [4] B. David-John, D. Hosfelt, K. Butler, and E. Jain. A privacy-preserving approach to streaming eye-tracking data. *IEEE Transactions on Visualization and Computer Graphics*, 27(5):2555–2565, 2021. doi: 10.1109/TVCG.2021.3067787
- [5] S. Dent, K. Burger, S. Stevens, B. D. Smith, and J. W. Streepey. The effect of music on body sway when standing in a moving virtual environment. *PLOS ONE*, 16(9):e0258000, Sept. 2021. doi: 10.1371/journal.pone.0258000
- [6] S. D'Mello and R. A. Calvo. Beyond the basic emotions: What should affective computing compute? In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, p. 2287–2294. Association for Computing Machinery, 2013. doi: 10.1145/2468356.2468751
- [7] J. C. Eubanks, A. G. Moore, P. A. Fishwick, and R. P. McMahan. A preliminary embodiment short questionnaire. *Frontiers in Virtual Reality*, 2, 2021.
- [8] S. Fremerey, A. Singla, K. Meseberg, and A. Raake. Avtrack360: An open dataset and software recording people's head rotations watching 360° videos on an hmd. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, p. 403–408. Association for Computing Machinery, 2018. doi: 10.1145/3204949.3208134
- [9] N. M. Gamage, D. Ishtaweera, M. Weigel, and A. Withana. So predictable! continuous 3d hand trajectory prediction in virtual reality. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, p. 332–343. ACM, Oct. 2021. doi: 10.1145/3472749.3474753
- [10] J. Hartmann, C. Holz, E. Ofek, and A. D. Wilson. Realitycheck: Blending virtual environments with situated physical reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, p. 1–12. ACM, May 2019. doi: 10.1145/3290605.3300577
- [11] T. Hartmann, W. Wirth, H. Schramm, C. Klimmt, P. Vorderer, A. Gysbers, S. Böcking, N. Ravaja, J. Laarni, T. Saari, F. Gouveia, and A. Maria Sacau. The spatial presence experience scale (spes). *Journal of Media Psychology*, 28(1):1–15, 2016. doi: 10.1027/1864-1105/a000137
- [12] D. Hepperle, T. Dienlin, and M. Wölfel. Reducing the human factor in virtual reality research to increase reproducibility and replicability. (arXiv:2110.15687), Oct. 2021. arXiv:2110.15687 [cs, stat].
- [13] R. Islam, K. Desai, and J. Quarles. Cybersickness prediction from integrated hmd's sensors: A multimodal deep fusion approach using eye-tracking and head-tracking data. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 31–40, 2021. doi: 10.1109/ISMAR52148.2021.00017
- [14] R. Islam, Y. Lee, M. Jaloli, I. Muhammad, D. Zhu, P. Rad, Y. Huang, and J. Quarles. Automatic detection and prediction of cybersickness severity using deep neural networks from user's physiological signals. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 400–411, 2020. doi: 10.1109/ISMAR50242.2020.00066
- [15] W. Jin, J. Fan, D. Gromala, and P. Pasquier. Automatic prediction of cybersickness for virtual reality games. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pp. 1–9, 2018. doi: 10.1109/GEM.2018.8516469
- [16] J. J. L. Jr, E. Kruijff, R. P. McMahan, D. Bowman, and I. P. Poupyrev. *3D User Interfaces: Theory and Practice*. Addison-Wesley Professional, Apr. 2017. Google-Books-ID: fxWSDgAAQBAJ.
- [17] N. Kang, D. Ding, M. B. Van Riemsdijk, N. Morina, M. A. Neerincx, and W.-P. Brinkman. Self-identification with a virtual experience and its moderating effect on self-efficacy and presence. *International Journal of Human-Computer Interaction*, 37(2):181–196, Jan. 2021. doi: 10.1080/10447318.2020.1812909
- [18] T. Kang, M. Chae, E. Seo, M. Kim, and J. Kim. Deepphndsvr: Hand interface using deep learning in immersive virtual reality. *Electronics*,

- 9(1111):1863, Nov. 2020. doi: 10.3390/electronics9111863
- [19] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.
- [20] Y. Li, M. Min, D. Shen, D. Carlson, and L. Carin. Video generation from text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(11), Apr. 2018. doi: 10.1609/aaai.v32i11.12233
- [21] J. Liebers, M. Abdelaziz, L. Mecke, A. Saad, J. Auda, U. Gruenefeld, F. Alt, and S. Schneegass. Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, p. 1–11. Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445528
- [22] J. Liebers, M. Abdelaziz, L. Mecke, A. Saad, J. Auda, U. Gruenefeld, F. Alt, and S. Schneegass. Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, p. 1–11. Association for Computing Machinery, May 2021. doi: 10.1145/3411764.3445528
- [23] J. Liebers, P. Horn, C. Burschik, U. Gruenefeld, and S. Schneegass. Using gaze behavior and head orientation for implicit identification in virtual reality. In *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, VRST '21. Association for Computing Machinery, 2021. doi: 10.1145/3489849.3489880
- [24] J. Lin and M. E. Latoschik. Digital body, identity and privacy in social virtual reality: A systematic review. *Frontiers Virtual Reality*, 3, 2022. doi: 10.3389/frvir.2022.974652
- [25] M. Maghoumi and J. J. LaViola Jr. Deepgru: Deep gesture recognition utility. (arXiv:1810.12514), Oct. 2019. arXiv:1810.12514 [cs]. doi: 10.48550/arXiv.1810.12514
- [26] M. R. Miller, E. Han, C. DeVeaux, E. Jones, R. Chen, and J. N. Bailenson. A large-scale study of personal identifiability of virtual reality motion over time. (arXiv:2303.01430), Mar. 2023. arXiv:2303.01430 [cs]. doi: 10.48550/arXiv.2303.01430
- [27] M. R. Miller, F. Herrera, H. Jun, J. A. Landay, and J. N. Bailenson. Personal identifiability of user tracking data during observation of 360-degree vr video. *Scientific Reports*, 10(11):17404, Oct. 2020. doi: 10.1038/s41598-020-74486-y
- [28] R. Miller, N. K. Banerjee, and S. Banerjee. Combining real-world constraints on user behavior with deep neural networks for virtual reality (vr) biometrics. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, p. 409–418, Mar. 2022. doi: 10.1109/VR51125.2022.00060
- [29] M. S. Mirzaei, K. Meshgi, E. Frigo, and T. Nishida. Animgan: A spatiotemporally-conditioned generative adversarial network for character animation. May 2020. arXiv:2005.11489.
- [30] A. G. Moore, T. D. Do, N. Ruozzi, and R. P. McMahan. Identifying virtual reality users across domain-specific tasks: A systematic investigation of tracked features for assembly. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 396–404, 2023. doi: 10.1109/ISMAR59233.2023.00054
- [31] A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Extracting velocity-based user-tracking features to predict learning gains in a virtual reality training application. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, p. 694–703. IEEE, Nov. 2020. doi: 10.1109/ISMAR50242.2020.00099
- [32] A. G. Moore, R. P. McMahan, H. Dong, and N. Ruozzi. Personal identifiability and obfuscation of user tracking data from vr training sessions. In *2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, p. 221–228, Oct. 2021. doi: 10.1109/ISMAR52148.2021.00037
- [33] A. G. Moore, R. P. McMahan, and N. Ruozzi. Exploration of feature representations for predicting learning and retention outcomes in a vr training scenario. *Big Data and Cognitive Computing*, 5(33):29, Sept. 2021. doi: 10.3390/bdcc5030029
- [34] V. Nair, W. Guo, J. Mattern, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song. Unique identification of 50,000+ virtual reality users from head & hand motion data, 2023.
- [35] V. Nair, W. Guo, R. Wang, J. F. O'Brien, L. Rosenberg, and D. Song. Berkeley open extended reality recordings 2023 (boxrr-23): 4.7 million motion capture recordings from 105,852 extended reality device users. (arXiv:2310.00430), Sept. 2023. arXiv:2310.00430 [cs].
- [36] M. Nørgaard and K. Hornbæk. What do usability evaluators do in practice? an explorative study of think-aloud testing. In *Proceedings of the 6th Conference on Designing Interactive Systems*, DIS '06, p. 209–218. Association for Computing Machinery, 2006. doi: 10.1145/1142405.1142439
- [37] K. Pfeuffer, M. J. Geiger, S. Prange, L. Mecke, D. Buschek, and F. Alt. Behavioural biometrics in vr: Identifying people from body motion and relations in virtual reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, p. 1–12. Association for Computing Machinery, May 2019. doi: 10.1145/3290605.3300340
- [38] J. L. Ponton, H. Yun, C. Andujar, and N. Pelechano. Combining motion matching and orientation prediction to animate avatars for consumer-grade vr devices. *Computer Graphics Forum*, 41(8):107–118, 2022. doi: 10.1111/cgf.14628
- [39] C. Rack, T. Fernando, M. Yalcin, A. Hotho, and M. E. Latoschik. Who is alyx? a new behavioral biometric dataset for user identification in xr. Aug. 2023. arXiv:2308.03788 [cs]. doi: 10.3389/frvir.2023.1272234
- [40] C. Rack, K. Kobs, T. Fernando, A. Hotho, and M. E. Latoschik. Extensible motion-based identification of xr users using non-specific motion data. (arXiv:2302.07517), July 2023. arXiv:2302.07517 [cs]. doi: 10.48550/arXiv.2302.07517
- [41] D. Reinhardt, S. Haesler, J. Hurtienne, and C. Wienrich. Entropy of controller movements reflects mental workload in virtual reality. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*, pp. 802–808, 2019. doi: 10.1109/VR.2019.8797977
- [42] A. Singla, S. Fremerey, W. Robitz, and A. Raake. Measuring and comparing goe and simulator sickness of omnidirectional videos in different head mounted displays. In *2017 Ninth International Conference on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, 2017. doi: 10.1109/QoMEX.2017.7965658
- [43] J.-P. Stauffert, F. Niebling, and M. E. Latoschik. Latency and cyber-sickness: Impact, causes, and measures. a review. *Frontiers in Virtual Reality*, 1, 2020. doi: 10.3389/frvir.2020.582204
- [44] B. Thoravi Kumaravel and A. D. Wilson. Dreamstream: Immersive and interactive spectating in vr. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22, p. 1–17. Association for Computing Machinery, Apr. 2022. doi: 10.1145/3491102.3517508
- [45] P. P. Tricomi, F. Nenna, L. Pajola, M. Conti, and L. Gamberini. You can't hide behind your headset: User profiling in augmented and virtual reality. *arXiv*, (arXiv:2209.10849), Sept. 2022. arXiv:2209.10849 [cs]. doi: 10.48550/arXiv.2209.10849
- [46] S. Villenave, J. Cabezas, P. Baert, F. Dupont, and G. Lavoué. Xrecho: A unity plug-in to record and visualize user behavior during xr sessions. In *Proceedings of the 13th ACM Multimedia Systems Conference*, MM-Sys '22, p. 341–346. Association for Computing Machinery, 2022. doi: 10.1145/3524273.3532909
- [47] E. Wen, C. Gupta, P. Sasikumar, M. Billingham, J. Wilmott, E. Skow, A. Dey, and S. Nanayakkara. Vr. net: A real-world dataset for virtual reality motion sickness research. *arXiv:2306.03381*, 2023.
- [48] E. Wen, T. I. Kaluarachchi, S. Siriwardhana, V. Tang, M. Billingham, R. W. Lindeman, R. Yao, J. Lin, and S. Nanayakkara. Vrhook: A data collection tool for vr motion sickness research. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–9, 2022.
- [49] M. Wölfel, D. Hepperle, C. F. Purps, J. Deuchler, and W. Hettmann. Entering a new dimension in virtual reality research: An overview of existing toolkits, their features and challenges. In *2021 International Conference on Cyberworlds (CW)*, pp. 180–187, 2021. doi: 10.1109/CW52790.2021.00038
- [50] J. Zhang, Z. Jiang, D. Yang, H. Xu, Y. Shi, G. Song, Z. Xu, X. Wang, and J. Feng. Avatargen: A 3d generative model for animatable human avatars. In L. Karlinsky, T. Michaeli, and K. Nishino, eds., *Computer Vision – ECCV 2022 Workshops*, Lecture Notes in Computer Science, p. 668–685. Springer Nature Switzerland, Cham, 2023. doi: 10.1007/978-3-031-25066-8_39