Enhanced Videogame Livestreaming by Reconstructing an Interactive 3D Game View for Spectators

Jeremy Hartmann
University of Waterloo
Waterloo, Ontario, Canada
jeremy@mtion.tv

Daniel Vogel
University of Waterloo
Waterloo, Ontario, Canada
dvogel@uwaterloo.ca

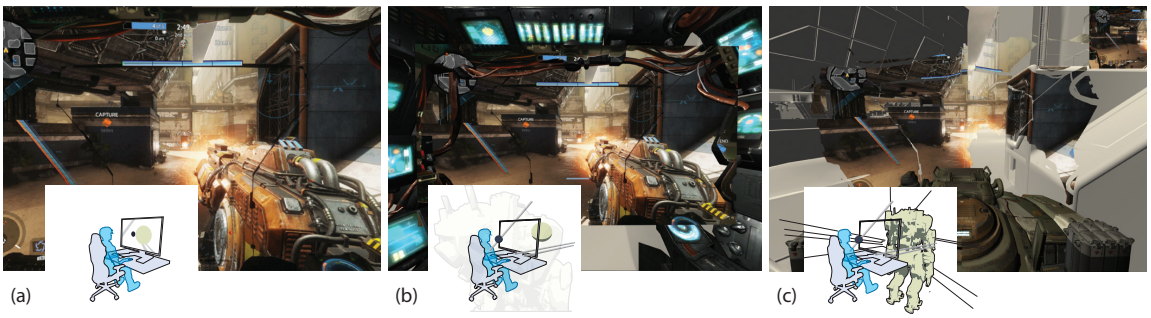

Figure 1: Enhanced videogame livestreaming examples for the game Titanfall 2 on a desktop environment: (a) a single RGB frame captured from the game depicts the default screen space view; (b) the 3D projected geometry creates a volumetric space, setting the spectator inside the titan with enhanced camera interactions; (c) the 3D projected geometry of the game is composited with a low-fidelity environment, creating a world space view that is decoupled from the streamer.

ABSTRACT

Many videogame players livestream their gameplay so remote spectators can watch for enjoyment, fandom, and to learn strategies and techniques. Current approaches capture the player's rendered RGB view of the game, and then encode and stream it as a 2D live video feed. We extend this basic concept by also capturing the depth buffer, camera pose, and projection matrix from the rendering pipeline of the videogame and package them all within a MPEG-4 media container. Combining these additional data streams with the RGB view, our system builds a real-time, cumulative 3D representation of the live game environment for spectators. This enables each spectator to individually control a personal game view in 3D. This means they can watch the game from multiple perspectives, enabling a new kind of videogame spectatorship experience.

CCS CONCEPTS

• **Human-centered computing** → **Virtual reality**; **Interactive systems and tools**; • **Information systems** → **Multimedia streaming**; • **Computing methodologies** → *Computer graphics*; Graphics file formats; • **Software and its engineering** → *Interactive games.*

CHI '22, April 29-May 5, 2022, New Orleans, LA, USA

ACM ISBN 978-1-4503-9157-3/22/04…$15.00
https://doi.org/10.1145/3491102.3517521

KEYWORDS

videogame streaming, virtual reality, graphics hacking, 3D video

ACM Reference Format:
Jeremy Hartmann and Daniel Vogel. 2022. Enhanced Videogame Livestreaming by Reconstructing an Interactive 3D Game View for Spectators. In *CHI Conference on Human Factors in Computing Systems (CHI '22), April 29-May 5, 2022, New Orleans, LA, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3491102.3517521

1 INTRODUCTION

Videogame live-streaming has become a popular pastime for both the streamers producing content and for the spectators consuming it [24, 49]. Web-streaming services, like Twitch [32] and YouTube Gaming [31], provide a platform for not only distribution of this video content but also a way for audiences to engage with the streamers and each other.

The typical stream consists of a primary game view containing the actual gameplay footage, and a composited picture-in-picture feed of the streamer captured through an external front-facing camera. All this footage is acquired through an external application, like Open Broadcast Software (OBS) [5], that duplicates the rendered videogame frame, encodes it, and then transports it to a streaming media server for distribution. The final content can then be viewed on various devices such as a desktop computer, mobile phone, or television screen. In this current structure, the role of the spectator is asymmetric to that of the streamer: the spectator's primary role is to passively watch the streamer with an optional and minimal chat interface for shared discussion. However, there is a growing trend of adding interactive elements into the stream for spectators. These are typically composited animations

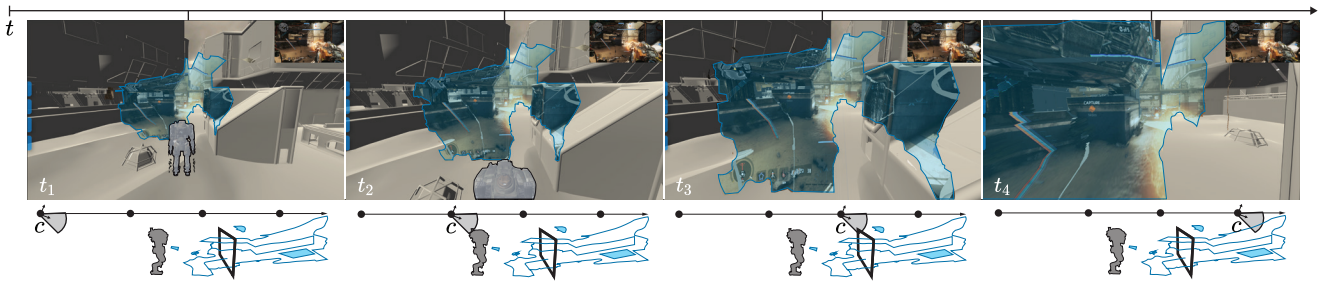

Figure 2: A timeline representing a sequence of frames captured as the spectator translates their camera c from behind the titan (t_1) into the 3D projected geometry (t_4). The illustrations depict where the spectator's camera c is relative to the titan (gray) and the projected 3D geometry (blue).

and graphics that react to specific keywords in the chat, but these can also consist of more complex arrangements where the spectator is given the ability to invoke an action directly within a predefined virtual environment [50]. Our work explores how to increase spectator interaction by generating different levels of an immersive 3D videogame streaming experience.

Despite the maturity of the streaming, the rise of virtual reality (VR) headsets has remained a challenge for both streamers and their spectators in subtly different ways. For streamers playing VR games, they have the challenging task of communicating what they are doing. One popular solution for non-VR spectators is to swap the streamer's first-person headset view for a third-person perspective using software like *LIV* [30]. However, the effectiveness and benefit of this simple approach remain unclear [16]. For spectators watching a videogame stream in VR, they are relegated to using a virtual theatre-like environment with a *big* screen [7]. These environments are effective at creating social spaces [11, 40] but they are incapable of taking advantage of the 3D environment of the videogame in any meaningful way. Our work is focused on the latter, we show how our 3D videogame streaming methods can enable immersive and interactive experiences for spectators in VR.

We introduce a method to dynamically generate a live 3D reconstruction of a 3D videogame environment at run time, and use it in a system to generate different levels of immersive and interactive experiences for spectators using desktop or VR. The method intercepts depth and virtual camera data exposed by low-level graphics rendering pipelines in the streamer's computer, then analyzes it for efficient transport and 3D reconstruction. The reconstructed 3D environment is used to create new visual and interactive capabilities for the spectator. For example, the spectator can view the streamer's actions from inside the game environment with full control over their position and vantage point. We demonstrate the flexibility of our approach through a design space spanning three levels of immersion, "screen space," "volumetric space," and "world space", for conventional 2D displays and 3D VR. Figure 1 illustrates these levels for a desktop environment and Figure 2 provides an example of world space locomotion. Figure 3 illustrates these levels for a VR environment. A key research question is whether spectators value these new levels of immersion and interaction. We conducted a study where participants experienced all three levels of immersion with three different videogames using a desktop interface or in VR. Our results suggest more immersive experiences are preferred, especially in VR.

In summary, we make the following contributions:

- A new streaming paradigm that leverages available 3D data from realtime gameplay to enable new ways for people to experience videogame streams;
- An end-to-end live-streaming system that demonstrates the approach is technically feasible, scalable, and generalizable;
- A study showing the effectiveness of our approach for 2D displays and VR.

2 RELATED WORK

The role of the spectator is asymmetrical to that of the performer, where the primary means of participation is accomplished through the simple act of *looking* [48]. There are intrinsic asymmetric qualities to the roles the streamer and their spectators have within the medium of videogame live-streaming. In this section, we elaborate on works in the areas of videogame live-streaming with emphasis on the spectator and how VR and other techniques can be used to enhance the spectator experience when watching videogames.

2.1 Videogame Spectatorship

There is a significant amount of research around the motivations, preferences, and reasons why people watch others play videogames [16, 24, 49]. For the most part, these investigations fall under two contexts: when the spectator is collocated with the player and when they are remote.

Collocated gaming and spectatorship has been studied in the context of audiences [36] to smaller intimate at-home play with only a few people [51]. To describe the relationships between the spectators and players, Downs et al. [14] proposed that the spectator can take on the role of a bystander, audience member, or player where participation can range from passive observance to active engagement [42]. Recently, it is becoming more common for games to blur what the type of role a spectator can have within a game, where they can take on a more direct role or even be a critical part of the game's design [19, 54]. One of our goals for this paper is to enable ways for the spectator to transition from passive to active engagement and become an active "audience member" in a purely remote setting.

In contrast to collocated spectatorship, watching others play games remotely is becoming an increasingly popular passtime, one that is comparable to traditional sports [10, 24, 38, 47]. To better understand the motivations behind why people engage in spectating activity, Sjöblom and Hamari looked at intrinsic and extrinsic factors that motivate users to watch others play videogames online [49]. They found that the total number of hours watched is positively associated with information seeking, tension release, and affective motivations. Expanding this to VR, Emmerich et al. investigated the live-streaming of VR games and found first- or third-person perspectives of the VR streamer can affect the spectators overall experience [16]. Their findings suggest that a third-person perspective of the VR player is not as effective as the view taken directly from the HMD, and can sometimes be detrimental to the viewing experience. However, this was limited to a fixed perspective with no spectator agency over the view. In this paper, we build off these insights to explore the inverse problem, VR users spectating non-VR videogame livestreams.

Though there has been plenty of research surrounding the user's affective experience and motivation for watching others play videogames, there has been little investigation into the specific ways systems can be enhanced to create new interactive capabilities for the spectator when watching videogames in VR or on desktop.

2.2 Systems that Enhance Spectatorship

Research has investigated ways in which an external non-VR user can view what another VR user is doing while in an immersive virtual environment. Silhouette Games [39] explores this through a mirror metaphor by compositing the mirror reflection of the VR user inside the videogame world for external viewers. ShareVR [22] uses Spatial Augmented Reality (SAR) to communicate what players in VR are doing to other collocated players external to them in a room-scale experience. TransceiVR [53] explored communication between a VR and external user in the context of productivity applications. RealityCheck [25] used a reconstruction of the VR player's physical environment for communication with external users. Though our work builds off of the insights explored in these works, we specifically look at the inverse problem: spectating non-VR games remotely in VR and on desktop.

Directly augmenting a head-mounted display (HMD) has been used for external communication across AR and VR. This has been explored through the direct placement of touchscreen displays onto the HMD [23] and through the attachment of small actuated pico projectors [26, 34, 56]. All of these systems specifically focus on how to bring context outside the virtual environment so external users can observe and interact without needing to be inside the same virtual space.

In contrast to exploring external non-VR spectatorship of VR users, is to spectate them while *in* VR. This has largely focused on live music concerts [35, 37] and live theatre [27]. Yakura and Goto looked at the individual audience member and their affective experience while inside a virtual concert event with others [60]. They proposed a machine-learning approach to synthesize audience movement when virtual concert attendance is minimal. Investigating multi-user collocated VR, Herscher et al. proposed a system and design hypotheses for enabling collective VR experiences for large theatre productions [28].

While there has been significant exploration of viewing VR users and for evaluating VR spectatorship experiences, little work has explored ways in which we can enhance current non-VR videogames for spectators in VR. The existing approaches are relegated to applications like BigScreen [7] and AltSpaceVR [3] that give the user a virtual place in which to watch different kinds of media on a 2D screen. In contrast, we explore a system and its uses for enhancing spectatorship for existing non-VR videogames.

3 ENHANCED VIDEOGAME SPECTATORSHIP

There are inherent differences between spectators and streamers in a livestreaming system, as the primary role of the streamer is to entertain their spectating audience and for the spectator to watch. What they watch is typically a 2D live video feed, where a front facing camera view of the streamer is overlaid on top of the main videogame content in a picture-in-picture arrangement. Additional graphical information is commonly composited into this arrangement to provide the spectators with information about the stream and to notify them about events. If we were to imagine an optimal form of videogame specatorship, the spectator would be immersed right into the videogame environment side-by-side with the streamer, where they could choose a vantage point, interact with the game world and the streamer, and be able to share their experience with other spectators in the real game space. This would require spectators to have access to a perfect realtime 3D reconstruction of the entire videogame environment. An example of this is present in the game Fortnite [18], where upon a player's defeat, they are able to watch their defeater from a third-person view. A more complete version of this exists in some games like League of Legends [20] where game owners can watch esports matches as spectators.

Though these examples provide the best possible experience for the spectator, real-world implementation issues make this impractical at scale. For example, current solutions require that spectators have access to all game assets, which can result in substantial download times and storage costs on per-game basis. For a massive online battle arena (MOBA) game like League of Legends [20], assets could be on the order of 10 GB, and for games like Call of Duty [1], well over 100 GB. Another consideration is the monetary cost associated with purchasing each videogame for purposes of spectatorship. For esports games, many of these games are free-to-play, but they also only make up a subset of games watched by spectators. Finally, there is additional development effort on the game creator to add game-specific spectatorship modes. Considering this, it is important to identify trade-offs between immersion, agency, fidelity, and interaction to make real-world applications for enhanced videogame spectatorship scalable, cost effective, and usable by a wide audience.

3.1 Design Space

We consider the trade-offs associated with possible enhancements across two technical dimensions: (1) the *medium* used by the spectator, and (2) the amount of videogame data needed to enable an experience. We explore these dimensions in three discrete *immersion levels*: screen, volumetric, and world. These represent increasing

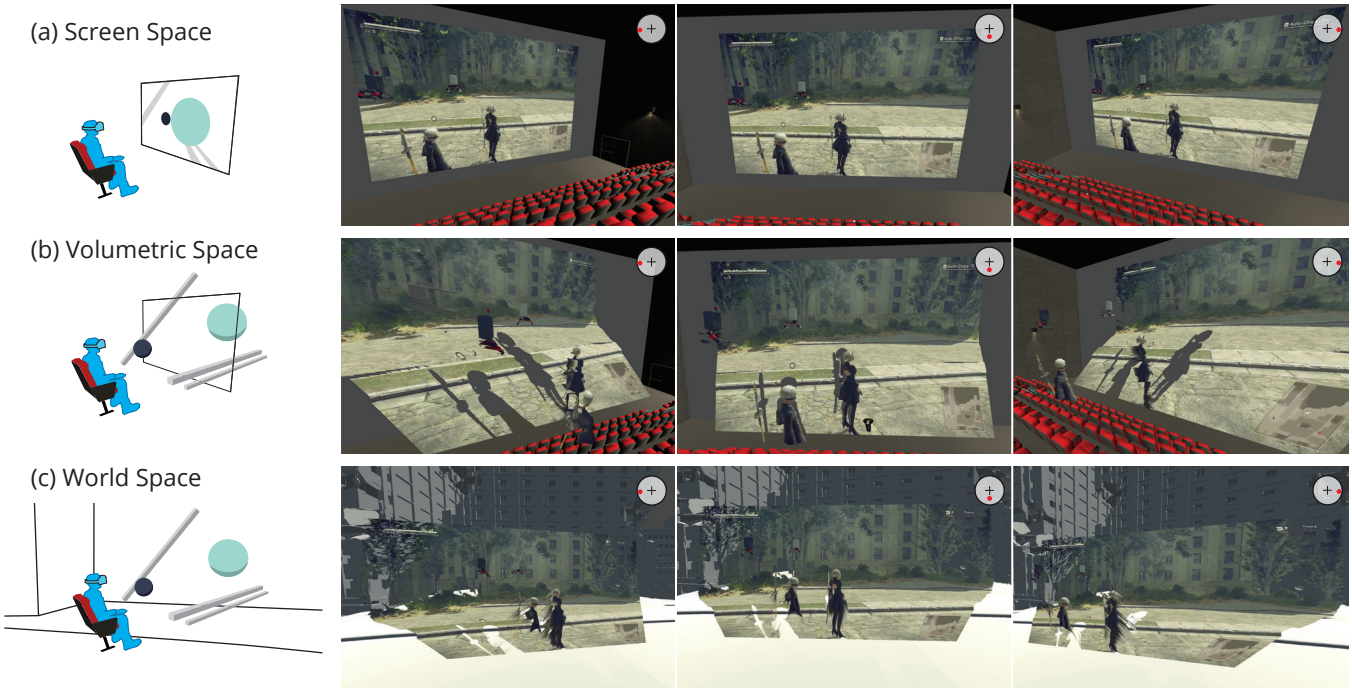

Figure 3: Spectator viewing levels when in VR: (a) screen space view uses the 2D video frames from the stream to recreate a cinema experience; (b) volumetric view uses the depth data to provide a 3D effect with limited locomotion and interaction with the projected geometry; (c) the world space view uses both the depth data and low-fidelity models from the game to create an environment that maximizes the spectator's locomotion and interaction capabilities letting them move around the space uncoupled from the streamer.

amounts of videogame data to produce spectator experiences that vary in the amount of agency and control they have within the spectating system. Each of these levels can be generalized to two broad categories of mediums used by the spectator: 2D display (i.e. a desktop computer) and 3D immersion (i.e. VR). Figure 3 illustrates each level conceptually and with screen captures from our system. The accompanying video figure also demonstrates several examples of spectating experiences across these levels and mediums.

Screen. The screen space level can be considered the canonical 2D live streaming experience. On desktop, the output of the game is displayed on a flat 2D display,where the spectator can either passively watch or engage with others through a real-time chat system. This is similar to how websites like Twitch [32] and YouTube [31] work. Alternatively, the spectator could watch in VR on a large virtual cinema style screen. This is equivalent to existing experiences provided through applications like BigScreen [7]. An advantage of this level of representation is that it allows the spectator to passively watch a videogame stream with minimal requirements around how they interact with other spectators or the streamer.

Volumetric. The volumetric level projects the incoming game data into a 3D environment to reconstruct parts of the game world for the spectator, which could be thought of as a kind of "3D movie." This act of projection transforms the stream from the space of the screen into a separate virtual world that encapsulates it. Now, both the spectator and the stream occupy the same virtual space, where the spectator can act on the stream independently of the streamer producing it. This arrangement opens up new opportunities for spectating with additional interactive elements designed to take advantage of the virtual space containing the spectators and stream data. For example, setting the user inside a diegetic room where the projected videogame data is composited within it, or by allowing them to shoot orbs at the reconstructed geometry of the stream and have it react to the spectator's actions.

Conceptually, we can think of this shared environment as a liminal space that sits in between both the physical environment of the spectator and the virtual environment of the videogame. This gives a designer the freedom to think of this space as being separate from the videogame environment, where there is no narrative connection. Alternatively, it might be desirable to create deliberate connections between the space the spectator is in and the videogame environment. These diegetic spaces could be used to advance the story in interesting and novel ways outside the primary narrative. And similar to screen space, this viewing mode also allows passive spectating with the added enhancements of viewing the content in a more immersive setting.

World. The world level combines the 3D volumetric projection with the positional and rotational information from the streaming game viewport. This provides not only the geometry from the game, but also where in the game this geometry is located. When combined together, new experiences can be created that place the

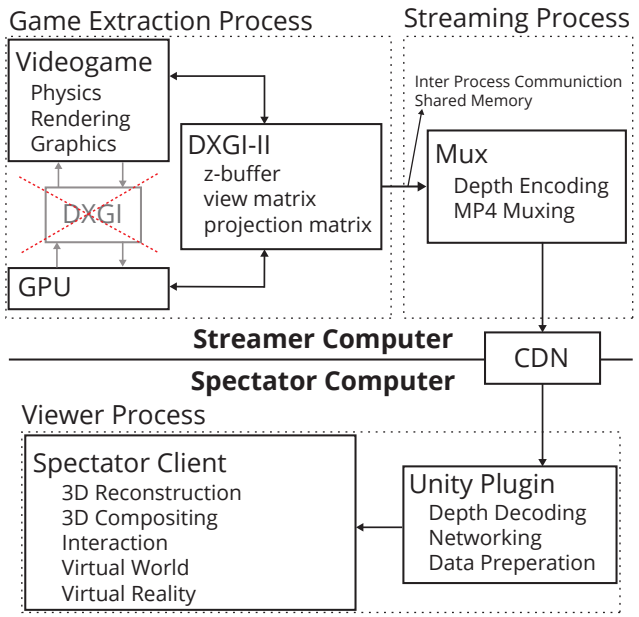

Figure 4: System diagram

spectator inside an approximation of the game being streamed. This gives the spectator the most agency over what they can do in the context of the videogame stream. For example, they now have the choice to follow along with the streamer as they play, or detach from the streamer to explore the areas around them (Figure 2).

An alternative to reconstructing the videogame environment at runtime is to utilize a low-fidelity 3D model of the videogame environment and composite the runtime 3D view on top of it. This type of configuration requires extra environmental information that is outside the current stream, but will also give the spectator extra context as to where they are in the videogame world and will effectively fill in information that could be lost when relying only on runtime reconstruction. One advantage of this is when multiple streamers are playing on the same map in a competitive battle royal or e-sports setting. A designer could tag specific spectator vantage points into the 3D environment to enable curated view such as a top-down view of all the players within the environment. However, this also requires more active participation from the spectator as they are directly in control over where they are and what they look at during the live stream. This contrasts to the more passive screen and volumetric spectating levels.

Considering all three spectating modes together, it is clear that each offer there own unique spectating experience along with a set of advantages and disadvantages for the spectator. Later, we will evaluate each of these levels in a remote study to examine how they affect the viewer experience, however first we discuss the system infrastructure and technologies that enable these experiences.

4 SYSTEM ARCHITECTURE

Current live streaming pipelines can be broken into three broad phases: data acquisition, content distribution, and client-side playback. We make modifications to each of these in order to create a streaming architecture that is capable of extracting and transporting the additional data we use for our reconstruction and visualizations of 3D videogame streams. This allows us to extract and transport the RGB and depth frame as well as the view and projection matrix of a game running on a Windows PC using DirectX 11. An overview of our architecture is in Figure 4. We describe each phase in detail next.

4.1 Data Acquisition

We take as a starting point the problem of extracting data from the rendering pipeline of a videogame. Previous work showed how the OpenVR DLL (Dynamic Link Library) can be exploited to extract the z-buffer from a VR game [25]. Hartmann et al. used a method that "hijacks" specific API calls, which is a general approach used in software analysis and reverse engineering [29]. However, this is limited to only OpenVR and fails to generalize to other games that do not bind to this specific protocol. Further, it is not clear how other videogame data, like the view or projection matrices, can be extracted through this higher-level technique.

We utilize the general idea of hijacking a DLL and extend it to work directly with the graphics API layer, bypassing higher level APIs like OpenVR. This allows us to directly intercept graphics data passed from the game to the GPU. We accomplish this be wrapping all DirectX 11 Graphics Interface (DXGI) definitions for all of `IDXGISwapChain`, `ID3D11Device`, and `ID3D11DeviceContext` interface classes, forcing the videogame process to link to our implementation of these APIs. This is visualized in Figure 4 as `DXGI-II`, and is an approach used within the game modding community through tools like Special K and Reshade that adjust stylistic aspects of the rendering pipeline [12, 43].

With a backdoor into the videogame rendering pipeline, we are able to build methods that extract the data we need to enable our novel spectator experiences. The data consists of RGB textures, depth textures, as well as view and projection matrices. Together, a single frame is composed of all four data types. We explain how each of these are extracted and bundled into a single frame next.

4.1.1 Texture Extraction. Extracting the RGB texture can be obtained by copying the backbuffer associated with the graphic device swap chain before `IDXGISwapChain::Present` is called. However, extracting the z-buffer texture is more involved.

A fully featured videogame uses many different z-buffers during a render pass. Typically, a z-buffer is used to ensure sufficient object culling, but it is also used for other post-processing passes, like screen space ambient occlusion (SSAO) [6]. In a videogame scene, every virtual camera will produce a z-buffer. This includes not only the players' view, but any other view into the scene. For example, it is common for an in-game world map to be rendered using actual environment geometry from a separate camera pass.

Since we are interested in the player's view of the game environment, we search for the z-buffer that corresponds to the primary RGB texture of the main game view. We accomplish this by directly utilizing and expanding on the injection system presented within the Reshade post-processing framework [43] by analyzing incoming data during calls to `D3D11DeviceContext::Draw`. During each draw call, pointers to associated depth textures are cached along

with simple statistics, like the number of draw calls, vertices rendered, and the texture dimensions. This is then used to choose a z-buffer that best corresponds to our target RGB texture. Alternatively, since we retain all the pointers to the z-buffers rendered during a frame, they can be displayed to the streamer through a simple interface overlaid on top of the game view. If the selection heuristics are wrong, the desired z-buffer can be manually selected. Note that this is typically a one-time task at the start of a gameplay session or level.

4.1.2 Matrix Data Extraction. Getting both the view and projection matrices is a far more challenging task as there is no direct way to extract this data without source code access. To overcome this, we employ two approaches: shader reflections and constant buffer (`cbuffer`) analysis.

The shader reflection approach is composed of three steps: (1) parse the shader byte code; (2) look for shader variables that contain typical view and projection matrix naming conventions like `view` or `proj`; and (3) store an index and offset into the constant buffer containing the matrix data for fast recall later. During runtime, the index and offset for the constant buffer is used to copy the matrix data associated with that particular frame.

The second approach requires analysis of the constant buffer during runtime. For each videogame frame, the system analyses the incoming constant buffer data passing through the DXGI calls to `D3D11DeviceContext::VSSetConstantBuffers`. For each of these constant buffers, the underlying raw information is extracted and specific signatures associated with a view or projection matrix are searched for. For the view matrix, we use a set of heuristics to ensure the transformation is well formed. This includes checking for a valid determinate, well-formed rotation matrix, and a translation vector within reasonable bounds. For the projection matrix, a similar approach is used that looks for common signatures found within the data. This includes the proper placement of coefficients, well formed focal length values, and reasonable near and far planes. If the data passes these checks, the index and offsets corresponding to each matrix are stored.

The collection of candidates is then visually presented to the streamer during initial setup. In cases where an incorrect projection or view matrix is selected, the streamer can manually override the default selection to ensure the correct matrices are used.

4.2 Data Preparation and Distribution

Once the data is extracted from the game, each frame will contain an RGB and z-buffer texture along with the view and projection matrices. Before the data can be distributed to remote spectators, it first needs to be transformed into a format with a sufficiently small memory profile suitable for streaming over the internet. We do this in a four step process: (1) get the raw uncompressed frame; (2) encode the RGB texture data using a H.264 video codec; (3) encode the depth data using a special codec; and (4) package all the data inside a customized MPEG-4 container.

4.2.1 Transferring data between processes. The streamer pipeline is composed of two processes. The first co-opts the videogame process and is responsible for the data extraction discussed above. The second process runs separately on the desktop and is tasked with preparing data for streaming. This is visualized by the line leading out of the *Game Extraction Process* into the *Streaming Process* in Figure 4. Keeping these separate has advantages. First it ensures that any issues in the data preparation process does not affect the streamer's gameplay. Second, it reduces computational overhead which can impact game performance. To transfer the data from one process to the other, we utilize an inter-process communication (IPC) bridge and shared virtual memory. This provides an efficient means to transfer data between the videogame process and the process used for stream preparation.

4.2.2 Encoding RGB Textures. Methods to encode RGB textures are well understood as specific standards have been developed [44]. We use a lossy H.264 [57] codec for all encoding and decoding of colour texture data.

4.2.3 Encoding Depth Textures. The extracted depth texture (z-buffer) is typically composed of 32-bit pixels, where 24 bits represent the distances of objects in camera space and the remaining 8 bits are used as a stencil. Unlike RGB textures, there is no established method to efficiently encode depth data for streaming. Previous work has proposed methods that repurpose existing encoding technology to transform depth data into a suitable format for compression. However, these approaches are expensive to run [45] or have explicit assumptions on pixel bitness [41]. To overcome these limitations, we use a "double-helix" encoding technique to transform depth data into a colour space The method is inspired by the cube-helix transform [21], and it ensures that the mapping between a 1D depth space to a 3D RGB colour is error-tolerant when compressed using the standard H.264 codec and multiplexed through a media server. Closely related depth transformation methods, like the approach proposed by Pece's et al. [45], would in principle be compatible with our system.

4.2.4 Data Multiplexing. Multiplexing video data typically consists of first encoding the images, audio, subtitles, and other data into an appropriate representation, and then placing the encoded data into a media container with metadata to describe the content. This step is visualized as the `Mux` container in Figure 4. For on-demand video and livestreaming, there are a number of media container formats that are typically used, such as Webm [46], HLS [4], and MPEG-4 [33]. We use the MPEG-4 (.mp4) family of formats due to its flexibility and extensibility.

Unlike video files that contain only a video and audio track, our stream contains five data types: RGB, depth, view and projection matrices, and audio. This requires multiplexing more data than what a media container typically handles.

Encapsulating Videogame Data. An MPEG-4 container file is composed of boxes called Atoms [52]. These define what type of data is contained within the the MPEG-4 container and how a media server should prepare that data for transport when playing files remotely. Each type of data is contained within an atom called a `trak`. This could be video, audio, subtitles, or something else. Associated with the `trak` atom are handlers (`hdlr`) that describe how the data within a `trak` atom is structured. This can include the type of encoding method used, framerate, and other metadata. This is then used by a media player to properly decode and transform the data for playback.

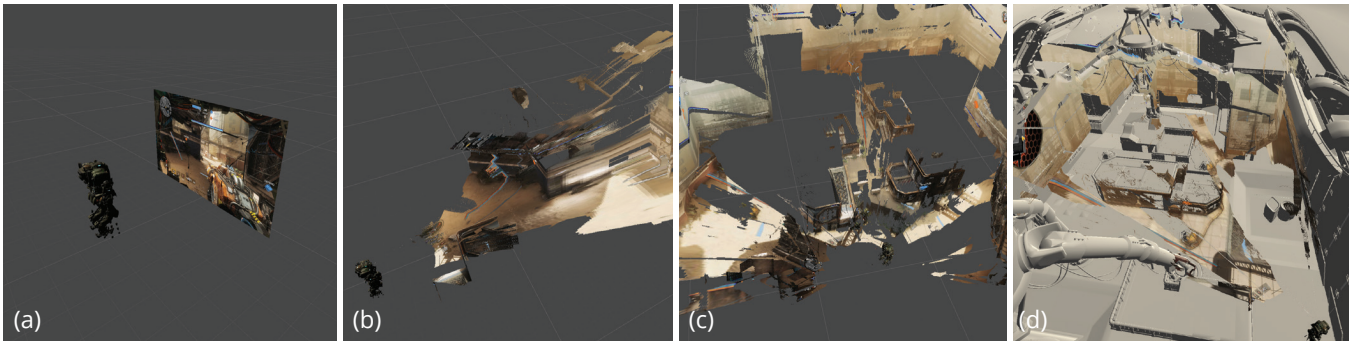

Figure 5: Visual representations: (a) screen, (b) volumetric; (c) reconstruction; and (d) world composite with low fidelity environment model. All frames captured from Titanfall 2.

We define four MPEG-4 `traks` that contain unique specifiers based on their data type. Two of these are dedicated to video content. The `trak` containing RGB video data data uses default MPEG-4 atoms. However, the `trak` containing the depth data needs to be identified during decoding in order to convert it from double-helix colour space back into depth space using the specialized depth encoding method described earlier.

The last two `traks` contain the view and projection matrix data. These each consist of a compressed array of 64 bytes, representing the 4×4 matrix. We define two additional handler types, one for the view matrix (`vmtn`) and one of the projection matrix (`pmtn`). During the parsing and decoding process, we intercept these data packets in order to process the view and projection matrix separately from the video data. An advantage of encapsulating the view and projection matrices inside the MPEG-4 is that it guarantees synchronization between all data with little extra overhead.

5 SPECTATOR VIEWER

To view and interact with the live streaming content, we built a prototype spectator player that is capable of playing our modified MPEG-4 formatted stream from a remote media server. This is visualized as the `Spectator Computer` in Figure 4. The streaming data can be rendered as either a 2D video, a 3D projection, or a 3D reconstruction of the environment being streamed. The rendering can additionally be targeted for a desktop or VR experience.

We use Unity 2019.4 LTS to implement the viewer application. This allows us to compose 3D objects and build out a user experience inside a game engine-like environment. However, all streaming and reconstruction functionality is contained in separate C++ libraries integrated into Unity through a plugin. This loose coupling means other editors or game engines could be used in the future.

5.1 Playback Engine

We connect to a remote media source through a custom media player with an API for media control and to access the raw decoded frames. This reads our enhanced MPEG-4 file either locally or from a uniform resource locator (URL). We use FFMpeg [17] for reading packets with a custom extension to delegate incoming `AVFrames` to specific routines for processing based on their underlying data and `hdlr` types embedded in metadata.

The RGB and depth video data types are decoded using the H.264 codec. For depth, additional decoding using the depth colour transformation method recovers the high-quality z-buffer texture. The two matrix data types are decoded using the LZ4 [13] compression algorithm. Together, all the decompressed data is composed into a single `DataFrame` in our library, then accessed on demand by any calling application.

Within the playback engine, the `DataFrame` is processed on a separate thread at the frame rate encoded by the enhanced video stream, which is typically 30 FPS. This is in contrast to the actual application which runs at a consistent 60 Hz when in the desktop mode and 90 Hz for VR. Since the application runtime is decoupled from `DataFrame` processing, frame processing time, connection issues, or dropped frames will not break the viewer experience or noticeably affect interaction with local game elements.

5.2 Environment and Reconstruction

The data packaged by the playback engine allows us to create different visual representations of the video stream for the spectator. An overview of these can be seen in Figure 5.

A 2D representation of the stream is comparable to typical video streaming experiences seen on websites such as Twitch or Youtube. This type of video can be viewed on a desktop computer or can be viewed within a VR theatre-like environment. This kind of experience directly relates to the *screen space* immersion level discussed in our Design Space (Figure 5a).

By utilizing the depth data associated with the frame, a 3D projection of the current view can be generated (Figure 5b). The 3D projection is created using a single perspective into the video game environment based on the projection matrix extracted from the game. The reconstructed geometry of the view has a one-to-one correspondence with the geometry in the videogame. This corresponds to the *volumetric space* immersion level in our Design Space.

Utilizing all the data contained in the 3D video frame, a reconstruction of the videogame environment is possible (Figure 5c). This corresponds to the *world space* immersion level in our Design Space. We accomplish this by utilizing the view matrix data, which perfectly represents the 3D pose of the camera at the time of capture. When combined with the depth data, we can then assign a specific position and rotation to the projected mesh geometry. Each frame

is then added to the previous, building up a static rendering of the environment as viewed from the virtual camera in the videogame. In our implementation, we overwrite any existing geometry with the new geometry produced by that video frame in order to keep all changes in the mesh current with what the streamer is viewing. This is similar to what simultaneous location and mapping (SLAM) algorithms do to build up 3D representations of a physical space, where the camera can only see what is inside its frustum and builds up its iterative reconstruction over time [9].

Further visualizations are possible by combining both the 3D video data with a low-fidelity model of the environment taken directly from the videogame through offline methods like videogame data mining (Figure 5d). The low-fidelity model is used as a backdrop from which the 3D projected mesh is composited directly on top of. This gives the spectator further context as to how the videogame environment is structured during a livestream. The model could be directly extracted from the game as an array of vertex buffers or extracted offline through data mining techniques. We used a videogame data mining approach for the study experiences described in Section 6.

Figure 6: Spectator interaction with streaming geometry: (a) light orbs shot into scene interact with the geometry, making it glow bright orange; (b) waypoint markers are placed to mark points of interest and notify others.

5.3 Interaction and Control

The spectator can interact and move around the reconstructed environments with varying levels of agency. This can range from no control for the simple 2D video case to complete 6DoF control over their viewport in the reconstructed case. For example, in the *screen space* immersion level, no interaction is possible with the stream itself. This is equivalent to existing streaming media websites.

In the *volumetric space* immersion level, the user is placed in a virtual space with the 3D projected geometry, allowing for a number of interactive enhancements. The first of these is locomotion. Since the volumetric space sits outside the videogame, locomotion is limited to a predefined area, like a social theatre environment. The spectator can move in this space using a keyboard and mouse on a desktop and controllers in VR. The VR controls consist of two handheld controllers that allow the spectator to manipulate objects through direct interactions and teleport to specific locations in the scene through raycasting. The second is interaction with the 3D projected geometry from the videogame. Since the 3D point of each pixel is known, salient objects from the videogame frame can be segmented using depth discontinuities and spatial locality of grouped pixels. Even though the semantic information around these objects are not known, experiences can be created that allow the spectator to play along with the streamer. For example, the spectator could aim and shoot light orbs at an on-screen enemy. At the point of intersection, the enemy mesh changes colour to indicate a hit (Figure 6a). Input uses mouse and keyboard on desktop and controllers in VR.

Finally the *world space* immersion level expands capabilities to bring more agency and freedom of movement to the spectator. The spectator can control where they want to go with respect to the streamer, as if they were playing their own first-person game. This includes following the streamer as they move through their environment or detaching from the streamer to explore other environment areas. Alternatively, the spectator could fly above the videogame map to get an aerial perspective of the action taking place in a table-top style environment.

5.4 Extensions and Enhancements

Additional experiences are possible by providing ways for the spectator to interact directly with the 3D projected videogame geometry. The viewer application allows the spectator to play along with the streamer by allowing the spectator to shoot orbs into the scene (Figure 6a). The orbs interact with the reconstructed videogame frame by causing an area of effect at the point of intersection, making the mesh glow brightly. Additionally, the spectator can add waypoints to the videogame environment that are decoupled from the streamers current view (Figure 6b). The waypoints can be composted directly within the current view from the streamer or used to indicate to other spectators where points of interest are located in either the current frame or past ones. These extensions and enhancements are implemented in our viewer, but not tested in our user study.

6 USER STUDY

The goal of this study is to evaluate how differing levels of immersion of a videogame livestream can affect the experience of the viewer who watches it. We explore these effects across across two mediums: *desktop* and *VR*.

Levels of immersion differ in both the agency the user has while spectating the videogame stream and the amount of 3D data used in the experience. We evaluate the three levels of our Design Space: *screen*, *volumetric*, and *world*. At the lowest level is *screen* which consists of only a 2D RGB video feed of the videogame stream. The next level is *volumetric* which projects the videogame view into 3D space. At the highest level is *world* which utilizes the 3D videogame projection with with a low-fidelity environment to geometrically composite them into a unified experience.

Figure 7: WORLD immersion for VR across each VIDEOGAME type: (a) Titanfall 2; (b) NieR:Automata; and (c) Homeworld: Desert of Kharak. The spectator can watch from above or teleport into the scene below, demonstrated by the picture-in-picture view.

Videogames were chosen to be representative of common gameplay genres and camera perspectives. These consist of: *Titanfall 2*, a first-person shooter (FPS); *NieR Automata*, a third-person action role-playing game (RPG); and *Homeworld: Desert of Kharak*, a top-down real-time strategy game (RTS). An overview of the *world* immersion level in *VR* for each *videogame* type can be viewed in Figure 7, which demonstrates an aerial perspective of the videogame stream with the ability to teleport down into the map using handheld controllers.

6.1 Participants

We initially gathered participant interest for our remote study by posting a general call to popular social media outlets like Reddit, Facebook, and Twitter. A total of 126 users (118 male, 4 female, 1 nonbinary, and 3 who preferred not to disclose their gender) responded by filling out a form consisting of general demographic information and questions to confirm they had the necessary computing power or VR equipment. From initial respondents, 30 where removed because they did not meet technical requirements.

The remaining 96 were contacted in an email outlining the details of the study, its requirements, and what participant responsibilities would be. From this, 30 people confirmed their interest in participating. In this group 2 encountered issues with their VR equipment and 10 became unresponsive to further emails from the researchers.

We ran our study with the remaining 18 participants, ages 16 to 36, of which 2 were female and 16 male. Each participant used their own VR headset tethered to a desktop gaming PC. This included: 14 Oculus Quest 2 and 4 Oculus Rift. All reported familiarity with using a VR headset: 17 reported they use VR at least once a week and 1 reported at least once a month. All but one participant reported they watch videogame livestreaming at least once a week on services like Twitch or YouTube. Internet speed across all participants averaged 133.9 Mbps (σ = 206.6). Participants were distributed across 2 continents: Europe and North America. Each participant received $15 USD for successful completion of the study.

6.2 Apparatus

A modified version of our spectator viewer (Section 5) is used with the participant's own gaming desktop computer and VR headset. Their computer needed to have at least an Intel i7 or AMD Ryzen 9 CPU, and at least an Nvidia GTX 1070 or AMD Radeon RX 580 GPU. We required a "tethered" VR headset to ensure consistent graphic fidelity across all participants. No headset was used in standalone mode.

The spectatorship software accessed each 3D stream through a global content distribution network (CDN) provided through Amazon Web Services (AWS). Endpoints were distributed across all major continents, ensuring low latency and high bandwidth access to each of the 3D video files for the entire participant pool.

6.3 Procedure

For each participant, the study started with a 15 minute onboarding session to outline the experiment procedure and the participant's responsibilities. Then the participant used our spectator viewer to view a series of 45 to 60 second 3D streams of the game in different immersion and medium conditions, ensuring that direct comparisons can be made effectively. They watched 18 streams in total: 3 different immersion levels in 2 different mediums, each with 3 videogames. Each stream encoded the movement data directly from the streamer, which was then cached on a content-distribution network for access later by the participant. The levels of immersion and interaction available to the participant are described in Section 5. For desktop, the participant viewed the streams on their computer monitor and interacted using mouse and keyboard input, similar to a first-person shooter game. For VR, they watched the streams using a VR HMD with all interaction using the standard handheld controllers, where teleportation is used as the primary mode of locomotion. The pacing and completion of each viewing was self-directed by the participant without any direct supervision by the researchers. Breaks in between were encouraged.

Upon completion of each individual stream, participants filled out a survey consisting of 6 preference questions. Upon completion of all 18 streams, a closing questionnaire captured final thoughts on their experiences across the entire experiment.

Overall, the study lasted approximately 90 minutes: 15 minute onboarding, 60 minutes for stream evaluations, and 15 minutes for the closing questionnaire. The study had to be completed within 3 days from the onboarding interview.

6.4 Design

This is a within subjects design with two primary independent variables: MEDIUM with 2 levels (VR, DESKTOP); and IMMERSION with

3 levels (SCREEN, VOLUMETRIC, WORLD). VIDEOGAME, which consists of three levels (TITANFALL, NEIR, HOMEWORLD), form secondary independent variables. Each combination of MEDIUM and IMMERSION were repeated 3 times, one for each of the VIDEOGAME types. The combination of MEDIUM and IMMERSION were counterbalanced using a Latin square to mitigate ordering bias. A random task order was used for VIDEOGAME.

The primary measures consisted of two subjective ratings asking if participants felt like they were immersed inside the videogame, and about their overall preference. Another, composite metric introduced by Venkatesh [55] was used to evaluate *perceived enjoyment*. This uses 4 separate questions to measure how much enjoyment the participant felt while watching the stream [49]. The composite metric was verified through factor analysis, verifying that each question contributed to the same measure ($\lambda = [3.47, 0.23, 0.16, 0.13]$). All measures are on a 5-point interval scale, where a "1" represents the most negative sentiment and a "5" the most positive.

In summary: 2 MEDIUM × 3 IMMERSION × 3 VIDEOGAME = 18 data points per question per participant.

6.5 Results

Aligned Rank Transform (ART) [58] and post hoc pairwise ART-C [15] tests with Holm correction were used for all non-parametric preference measures. Figure 8 provides an overview of the results.

6.5.1 Overall Preference. Across both mediums, participants preferred volumetric and world experiences over the baseline screen experience. There is a main effect of IMMERSION on overall user preference ($F_{2,304} = 15.3, p < 0.001$). Post hoc tests show that VOLUMETRIC ($\mu = 3.3$, $\sigma = 1.3$) and WORLD ($\mu = 3.5$, $\sigma = 1.1$) are both preferred over SCREEN ($\mu = 2.8$, $\sigma = 0.8$) irrespective of MEDIUM (all $p < 0.001$).

For medium type, participants preferred VR over desktop. There is a main effect of MEDIUM on overall user preference ($F_{1,305} = 11.2, p < 0.001$). A post hoc test shows VR ($\mu = 3.4$, $\sigma = 1.1$) is preferred over DESKTOP ($\mu = 3$, $\sigma = 1.2$) ($p < 0.001$).

For videogame type, participants preferred both third person NieR and first person Titanfall over Homeworld, the top down strategy game. There is a main effect of VIDEOGAME on overall user preference ($F_{2,304} = 12.7, p < 0.001$) Post hoc tests show that NIER ($\mu = 3.3$, $\sigma = 1.1$) and TITANFALL ($\mu = 3.5$, $\sigma = 1.2$) are preferred to HOMEWORLD ($\mu = 2.9$, $\sigma = 1.1$) (all $p < 0.005$). There is no significant difference between NIER and TITANFALL ($p = 0.09$).

Overall, participants preferred the world immersion level across both desktop and VR. There is an interaction between IMMERSION and MEDIUM on overall user preference ($F_{2,289} = 6.4, p < 0.002$). For VR, post hoc tests found that VOLUMETRIC ($\mu = 3.7$, $\sigma = 1.0$) and WORLD ($\mu = 3.7$, $\sigma = 1.1$) are preferred over SCREEN ($\mu = 2.8$, $\sigma = 0.9$) (all $p < 0.001$). No effect is reported between VOLUMETRIC and WORLD ($p = 1$). For DESKTOP, post hoc tests found WORLD ($\mu = 3.3$, $\sigma = 1.2$) to be preferred over SCREEN ($\mu = 2.9$, $\sigma = 0.9$) ($p < 0.045$). No other differences were found between any of the other IMMERSION types for DESKTOP (all $p > 0.4$).

6.5.2 Feeling immersed inside the videogame. Participants felt more inside the videogame for both the volumetric and world immersion levels when compared with the baseline screen experience. There is a main effect of IMMERSION on the participant's affective experience of of being present within the videogame with the streamer ($F_{2,304} = 18.7, p < 0.001$). Post hoc tests show that both VOLUMETRIC ($\mu = 3.1$, $\sigma = 1.4$) and WORLD ($\mu = 3.2$, $\sigma = 1.3$) are more aligned with feeling inside the videogame then the baseline SCREEN ($\mu = 2.3$, $\sigma = 1.1$) (all $p < 0.001$). There is no significant difference between VOLUMETRIC and WORLD ($p = 0.61$).

For medium, participants felt more inside the videogame for VR when compared with desktop. There is a main effect of MEDIUM on the user's affectual experience of being inside the videogame ($F_{1,305} = 4.8, p < 0.03$). A post hoc test shows that participants felt more inside the videogame for VR ($\mu = 3$, $\sigma = 1.3$) when compared with DESKTOP ($\mu = 2.7$, $\sigma = 1.3$) ($p < 0.03$).

Overall, we found the world immersion level to be the most effective at evoking feelings of being in the game regardless of medium type. There is an interaction between IMMERSION and MEDIUM on overall feelings of being inside the game with the streamer ($F_{2,289} = 7.6, p < 0.001$). For VR, post hoc tests show that participants felt more inside the game for VOLUMETRIC ($\mu = 3.3$, $\sigma = 1.2$) and WORLD ($\mu = 3.6$, $\sigma = 1.2$) when compared with SCREEN ($\mu = 2.2$, $\sigma = 1.1$) (all $p < 0.001$). No other differences are observed between VOLUMETRIC and WORLD ($p = 0.46$). For DESKTOP, post hoc tests show that WORLD ($\mu = 3.1$, $\sigma = 1.4$) felt more inside the game then SCREEN ($\mu = 2.4$, $\sigma = 1.1$) ($p < 0.003$). No other differences are observed for DESKTOP (all $p > 0.15$).

6.5.3 Perceived Enjoyment. Participants reported the most enjoyment from both the world and volumetric immersion levels over the baseline screen experience. There is a main effect of IMMERSION on perceived enjoyment ($F_{2,304} = 11.67, p < 0.001$). Post hoc test show that WORLD ($\mu = 3.4$, $\sigma = 1.1$) and VOLUMETRIC ($\mu = 3.2$, $\sigma = 1.3$) are preceived as being more enjoyable when compared with SCREEN ($\mu = 2.8$, $\sigma = 0.9$) (all $p < 0.001$). There is no significant difference between WORLD and VOLUMETRIC ($p = 0.19$).

Participants enjoyed the videogame experiences more in VR than they did on desktop. There is a significant effect of MEDIUM on perceived enjoyment ($F_{1,305} = 14.24, p < 0.001$). A post hoc test shows that VR ($\mu = 3.3$, $\sigma = 1.1$) is perceived more enjoyable when compared with DESKTOP ($\mu = 2.9$, $\sigma = 1.15$) ($p < 0.001$).

Overall, participants perceived the world immersion level as being the most enjoyable regardless of medium type. There is an interaction between IMMERSION and MEDIUM for perceived enjoyment ($F_{2,301} = 3.7, p < 0.03$). For VR, post hoc tests show that WORLD ($\mu = 3.6$, $\sigma = 1.0$) and VOLUMETRIC ($\mu = 3.6$, $\sigma = 1.1$) are perceived more enjoyable then SCREEN ($\mu = 2.8$, $\sigma = 1.0$) (all $p < 0.001$). There is no significant difference between WORLD and VOLUMETRIC ($p = 0.98$). For DESKTOP, post hoc tests show that WORLD ($\mu = 3.6$, $\sigma = 1.0$) is perceived as more enjoyable then SCREEN ($\mu = 2.8$, $\sigma = 1.0$) ($p < 0.045$). No other differences are observed (all $p > 0.44$).

6.5.4 Ranked Preferences. Participants ranked their top three experiences grouped by IMMERSION. The immersion level of VOLUMETRIC is ranked as the most preferred across DESKTOP and VR ($N = 8$), followed by WORLD ($N = 5$), and finally STREAM as the least preferred ($N = 1$). A non-preference is reported by 4 participants.

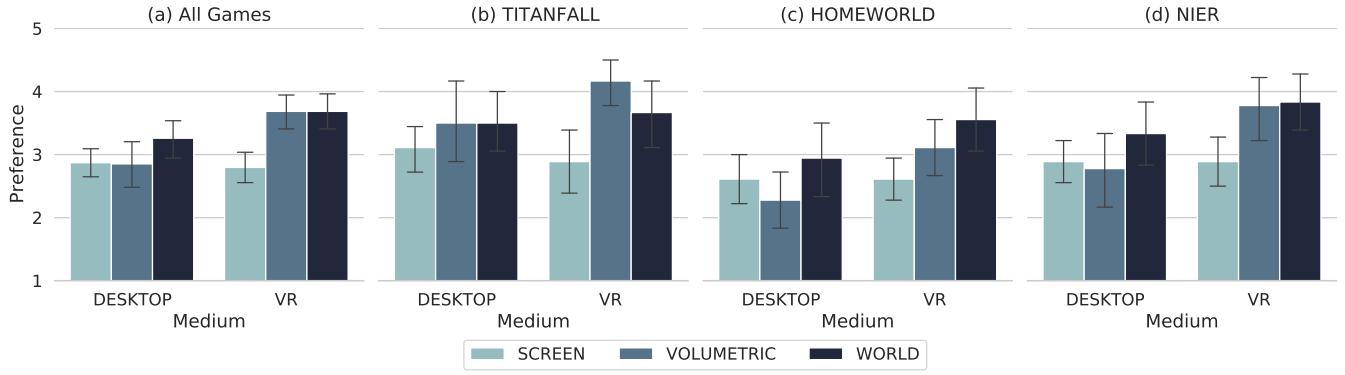

Figure 8: Overall preference ratings by (a) IMMERSION and MEDIUM and (b, c, d) VIDEOGAME type (error bars 95% CI).

7 DISCUSSION

We found compelling differences between the medium and immersion types in how they affected participant sentiments towards specific visualizations of the videogame streams.

3D Streams Enhance Enjoyment and Immersion. Overall, participants found that watching a 3D videogame stream enhances their overall enjoyment and immersion when compared directly to a 2D stream of the same content. This is true regardless of the videogame type or the medium they watched it in. This can be seen in our results which report both the world and volumetric levels as being the most preferred (Section 6.5.1 and Section 6.5.4). However, when participants were asked to explicitly rank their preferences, the majority preferred the volumetric experience slightly more than the world space experience. This may be due to how the volumetric experience gives the user a 3D experience with interactive elements in a more passive manner, which contrasts the world space experiences that require direct engagement with the stream. This observation is reinforced by a participant who stated they were "*blown away by the 3d theatre experience*"[P6] and found it "*relaxing to watch compared to the interactive versions*"[P6] in world space.

Preference for 3D Streams in VR. We reported an overall preference for VR over desktop, including a greater sense of perceived immersion and enjoyment. This may be partially due to how the stereoscopic displays in a VR headset render a scene, letting the participant experience the three dimensional aspects of the projected videogame frame in a more pronounced way. This can be seen in the participants remarks, stating that the "*watching experience [was] greatly improved by the 3D reconstruction*"[P4], that they "*liked the fact that the screen was 3d like in a movie theatre*"[P6], and that they "*loved being able to switch perspectives from looking at the map to actually being inside it*"[P12] when viewing a stream in the world space immersion level.

The difference between a 2D and 3D stream was more apparent in VR then on desktop. Participants stated that they "*felt like [they were] in a 3D cinema*"[P2], and that it made them "*feel like [they were] playing along*"[P15] when spectating in VR. In contrast, viewing the 3D reconstruction on desktop was mixed. This is reflected in the lack of differences between immersion levels for desktop and in the participant's individual comments. Some did not see "*any value in adding false depth*"[P3] or thought that it did not provide "*any benefit on a monitor*"[P1] screen. However, some other participants felt it continued to make them feel "*like [they were] there with the streamer*"[P8] and that it was able to provide "*additional context to the game being played*"[P15], even when viewing on a desktop screen.

A number of participants stated that they felt 'in' the game with the streamer when watching in 3D (8 participants). Commenting how it "*felt like I was in the game right behind the player*"[P12], "*felt like I was part of the battle*"[P10], and how the characters seemed "*larger than life [where] the action seemed to be particularly clearer and real as a result of the level of depth*"[P17]. This sentiment is also reflected in our reported results, where the overall effect of the 3D reconstruction had an impact on participant's feelings of being there with the streamer.

Suitability of 3D Streams for Game Types. We reported an overall preference for Titanfall 2 and NieR:Automata over the videogame Homeworld: Deserts of Kharak. This may be due to two intrinsic qualities that Homeworld has that the other two videogames do not. The first of these being that it is a real-time strategy game (RTS) and the other is in how the virtual environment is rendered through the game camera. Some participants stated their general dislike for RTS games in general, where they felt bored as they did not "*care about the subject*"[P8] matter presented to them. Other participants commented on the general 'flatness' of the scene due to the camera vantage point, stating that "*everything looks flat*"[P5] when viewing the 3D reconstruction and that the "*perspective and distance [made it] too hard to tell what the player's doing*"[P1]. The 'flatness' some observed is the result of positioning the camera very far from the game geometry, making the depth effect less pronounced. However, in contrast to this sentiment, some participants explicitly stated that Homeworld was their "*favorite way to view a stream*" and felt that it created a type of "*2.5D game*"[P17] experience, which emphasizes the two-dimensional aspects of the experience with added 3D effects.

Across all videogame types, the volumetric 3D rendering experience had the most pronounced effect inside of VR with the exception of Titanfall 2, which saw a moderately positive increase on desktop as well. This could be due to the diegetic environment of the titan when rendering the 3D volumetric stream. In this scenario,

the spectator view was actually inside the titan which may have contributed to the feelings of being more immersed in the experience. However, there was no interaction effect between immersion, medium, and videogame ($p = 0.89$) so no definite conclusions can be made.

Technical Limitations of Reconstructed 3D Streams. Some participants commented on the inherent limitations of the live-streaming system. Due to how we capture depth data from the videogame, sections of the scene will be occluded by objects directly in front of the camera. These are known as depth shadows. In total, 5 participants directly or indirectly made comments about these depth reconstruction artifacts. For example, they noted that the cutout from the "*gun*"[P14] in Titanfall 2 or the "*shadow*"[P3] created by 2B in NieR:Automata could sometimes be distracting. Other participants commented directly on the quality of 3D reconstruction, stating that the geometry could be "*spiky*"[P4] and that the image would become more distorted around complex geometry like trees [P2]. This can occur at times when the graphics shaders do not detect depth discontinuities properly in the depth buffer, which can result in geometry being generated where it should not be and may cause slight motion sickness (noted by [P16]). Another possible explanation could be due to how the depth codec reconstructed the scene. At lower bandwidths, it would have to reconstruct more lost depth data which can affect visual fidelity.

System Robustness. Our study was conducted entirely remotely and took place across two continents. This gave us the opportunity to test our infrastructure and system at scale. There are trade offs to this, one being that we did not have precise control over what equipment the participants use or the network bandwidth and latency. However, we gained valuable insight as to extent and feasibility of deploying such a system across a wide geographic region. For the most part, participants did not report many issues related to network connectivity or reconstruction. Participants that did report issues found they were typically resolved once the CDN network cached packets closer to their physical location.

7.1 Limitations and Future Work

While our system and infrastructure is adequate for the study we conducted, there are areas that could be refined and opportunities for future work.

Depth Shadows. We capture the depth buffer directly from the videogame we stream. The advantage of this is that it gives us an exact replica of the geometry as it was rendered. However, the data from anything occluded during rendering will be lost causing a "depth shadow." As mentioned in our discussion, some participants commented on this. One possible solution is to use an array of virtual cameras in the game view to generate light field video [8]. However, this would require extra rendering passes per virtual camera in the array, which could affect the frame rate of the videogame. Another approach to consider is inpainting via neural irradiance fields to fill in missing geometry and pixels [59]. Both of these are interesting directions for future work.

Remote Study. We conducted a distributed study across two continents with 18 participants. Though a remote study has its advantages like sampling from a wider participant pool with different setups and configurations, there are disadvantages in level of control over variables such as bandwidth and equipment, and the amount of supervision that can be reasonably given. Though we did attempt to normalize these variables across participants, they are harder to control when compared with an in-lab study.

Videogame Vignettes. We sampled a set of videogames that were representative of three prominent game genres. The vignettes were pre-recorded and streamed on-demand to participants from a content-distribution network. Our goal was to simulate a livestream on a technical level in a condensed form suitable for a within-subjects study to enable direct comparisons. However, this does not capture how an immersive 3D streaming experience might affect the bidirectional relationship and social dynamics between streamers and spectators. Conducting a more narrow study using the world immersion level with one game and one medium would be an interesting direction for future work.

Scene Changes. During environment reconstruction for the world space immersion level, we iteratively build up a 3D scene over a sequence of frames. There are instances were this simple approach could break when the continuity of the scene abruptly changes or when large objects move in front of the camera. For example, when the streamer changes levels or views a menu screen that is detached from the game world. In many cases, simple heuristics like detecting abrupt changes in the view matrix or discontinuity in the video frame, could identify these moments and render a suitable scene change for the spectator. A more robust approach using computational geometry techniques that take into account the actual geometric shapes and textures in each frame would be an interesting direction for future work [2].

Extensions. Many participants suggested use cases and extensions. For example, suggesting it could work in an e-sports setting (4 participants) or having the ability to dynamically 'switch' between views would be beneficial (3 participants). In particular, [P3] suggested using the 3D reconstruction of the videogame "*as a replay environment [where you] could pause and rewind, and move the camera to check out details*"[P3].

8 CONCLUSION

We presented a system and study that demonstrated the feasibility of capturing, encoding, transporting, and rendering immersive 3D streams for spectators to view on desktop or VR. A distributed study demonstrated our approach at scale, and the results show that immersive 3D streams enhance the overall spectator experience. In the future we plan to explore how our system can enhance the streamer to spectator relationship and how our system can be adapted to virtual tubing (VTubing) to leverage depth data and immersion for communication and entertainment.

ACKNOWLEDGMENTS

Special thanks to Remy Siu[1] for providing initial feedback on early prototypes and for their in-depth discussions around the philosophy of virtual spaces and our relationship with them. This work made

[1] https://remysiu.com/

possible by NSERC Discovery Grant 2018-05187, Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces," and the Ontario Early Researcher Award ER16-12-184.

REFERENCES

[1] Activision. 2022. Call of Duty®: Black Ops - Cold War | Popular FPS Game. https://www.callofduty.com/ca/en/blackopscoldwar. (Accessed on 01/06/2022).

[2] Helmut Alt. 2009. *The Computational Geometry of Comparing Shapes.* Springer Berlin Heidelberg, Berlin, Heidelberg, 235–248. https://doi.org/10.1007/978-3-642-03456-5_16

[3] AltspaceVR. 2021. AltspaceVR | Be there, together. https://altvr.com/. (Accessed on 05/06/2021).

[4] Apple. 2021. HTTP Live Streaming (HLS) - Apple Developer. https://developer.apple.com/streaming/. (Accessed on 09/01/2021).

[5] Hugh Bailey. 2021. Open Broadcaster Software | OBS. https://obsproject.com/. (Accessed on 09/03/2021).

[6] Louis Bavoil and Miguel Sainz. 2008. Screen space ambient occlusion. *NVIDIA developer information: http://developers. nvidia. com* 6 (2008).

[7] Bigscreen. 2021. Bigscreen. https://www.bigscreenvr.com/. (Accessed on 05/02/2021).

[8] Michael Broxton, John Flynn, Ryan Overbeck, Daniel Erickson, Peter Hedman, Matthew Duvall, Jason Dourgarian, Jay Busch, Matt Whalen, and Paul Debevec. 2020. Immersive light field video with a layered mesh representation. *ACM Transactions on Graphics* 39, 4 (jul 2020), 15. https://doi.org/10.1145/3386569.3392485

[9] Carlos Campos, Richard Elvira, Juan J.Gomez Rodriguez, Jose M.M. Montiel, and Juan D. Tardos. 2021. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Transactions on Robotics* 37, 6 (2021), 1874–1890. https://doi.org/10.1109/TRO.2021.3075644 arXiv:2007.11898

[10] Gifford Cheung and Jeff Huang. 2011. Starcraft from the Stands: Understanding the Game Spectator. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11.* ACM Press, New York, New York, USA, 763. https://doi.org/10.1145/1978942.1979053

[11] Elizabeth F Churchill, David N Snowdon, and Alan J Munro. 2012. *Collaborative virtual environments: digital places and spaces for interaction.* Springer Science & Business Media.

[12] Andon M. Coleman. 2021. The Complete Guide to SK | Special K - The Official Wiki. https://wiki.special-k.info/. (Accessed on 05/11/2021).

[13] Yann Collet. 2021. LZ4 - Extremely fast compression. https://lz4.github.io/lz4/. (Accessed on 09/04/2021).

[14] John Downs, Frank Vetere, Steve Howard, Steve Loughnan, and Wally Smith. 2014. Audience Experience in Social Videogaming: Effects of Turn Expectation and Game Physicality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* ACM, New York, NY, USA, 3473–3482. https://doi.org/10.1145/2556288.2556965

[15] Lisa A Elkin, Matthew Kay, James J Higgins, and Jacob O Wobbrock. 2021. An aligned rank transform procedure for multifactor contrast tests. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST '21).* https://doi.org/10.5281/zenodo.594511

[16] Katharina Emmerich, Andrey Krekhov, Sebastian Cmentowski, and Jens Krueger. 2021. Streaming VR Games to the Broad Audience: A Comparison of the First-Person and Third-Person Perspectives. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems.* 1–14. https://doi.org/10.1145/3411764.3445515 arXiv:2101.04449

[17] FFmpeg. 2021. FFmpeg. https://www.ffmpeg.org/. (Accessed on 08/06/2021).

[18] Epic Games. 2021. Fortnite | Free-to-Play Cross-Platform Game - Fortnite. https://www.epicgames.com/fortnite/en-US/home. (Accessed on 08/04/2021).

[19] Resolution Games. 2019. Acron: Attack of the Squirrels! https://www.resolutiongames.com/acron. (Accessed on 09/06/2021).

[20] Riot Games. 2022. League of Legends. https://www.leagueoflegends.com/en-us/. (Accessed on 01/06/2022).

[21] D. A. Green. 2011. A colour scheme for the display of astronomical intensity images. *Bulletin of the Astronomical Society of India* 39, 2 (aug 2011), 289–295. arXiv:1108.5083 http://arxiv.org/abs/1108.5083

[22] Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2017. ShareVR: Enabling Co-Located Experiences for Virtual Reality between HMD and Non-HMD Users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* ACM, New York, NY, USA, 4021–4033. https://doi.org/10.1145/3025453.3025683

[23] Jan Gugenheimer, Evgeny Stemasov, Harpreet Sareen, and Enrico Rukzio. 2018. FaceDisplay: Towards Asymmetric Multi-User Interaction for Nomadic Virtual Reality. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems.* ACM, New York, NY, USA, 1–13. https://doi.org/10.1145/3173574.3173628

[24] William A. Hamilton, Oliver Garretson, and Andruid Kerne. 2014. Streaming on twitch: Fostering participatory communities of play within live mixed media. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14.* ACM Press, New York, New York, USA, 1315–1324. https://doi.org/10.1145/2556288.2557048

[25] Jeremy Hartmann, Christian Holz, Eyal Ofek, and Andrew D. Wilson. 2019. RealityCheck: Blending Virtual Environments with Situated Physical Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19.* ACM Press, New York, New York, USA, 1–12. https://doi.org/10.1145/3290605.3300577

[26] Jeremy Hartmann, Yen-ting Yeh, and Daniel Vogel. 2020. AAR: Augmenting a Wearable Augmented Reality Display with an Actuated Head-Mounted Projector. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* ACM, New York, NY, USA, 445–458. https://doi.org/10.1145/3379337.3415849

[27] Linjia He, Hongsong Li, Tong Xue, Deyuan Sun, Shoulun Zhu, and Gangyi Ding. 2018. Am I in the theater? Usability Study of Live Performance Based Virtual Reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology.* ACM, New York, NY, USA, 1–11. https://doi.org/10.1145/3281505.3281508

[28] Sebastian Herscher, Connor DeFanti, Nicholas Gregory Vitovitch, Corinne Brenner, Haijun Xia, Kris Layng, and Ken Perlin. 2019. CAVRN: An exploration and evaluation of a collective audience virtual reality nexus experience. In *UIST 2019 - Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology.* ACM, New York, NY, USA, 1137–1150. https://doi.org/10.1145/3332165.3347929

[29] Galen Hunt and Doug Brubacher. 1999. Detours: Binary interception of Win32 functions. *3rd USENIX Windows NT Symposium* (1999). http://research.microsoft.com/sn/detours

[30] LIV Inc. 2021. LIV | Your VR capture toolbox. https://www.liv.tv/. (Accessed on 05/02/2021).

[31] Youtube Inc. 2021. YouTube Gaming. https://www.youtube.com/gaming. (Accessed on 05/04/2021).

[32] Twitch Interactive. 2021. Twitch. https://www.twitch.tv/. (Accessed on 05/04/2021).

[33] ISO/IEC 14496-11 2015. *Information technology — Coding of audio-visual objects — Part 11: Scene description and application engine.* Standard. International Organization for Standardization, Geneva, CH.

[34] Pascal Jansen, Fabian Fischbach, Jan Gugenheimer, Evgeny Stemasov, Julian Frommel, and Enrico Rukzio. 2020. ShARe: Enabling Co-Located Asymmetric Multi-User Interaction for Augmented Reality Head-Mounted Displays. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* ACM, New York, NY, USA, 459–471. https://doi.org/10.1145/3379337.3415843

[35] Tatsuyoshi Kaneko, Hiroyuki Tarumi, Keiya Kataoka, Yuki Kubochi, Daiki Yamashita, Tomoki Nakai, and Ryota Yamaguchi. 2019. Supporting the sense of unity between remote audiences in VR-based remote live music support system KSA2. In *Proceedings - 2018 IEEE International Conference on Artificial Intelligence and Virtual Reality, AIVR 2018.* 124–127. https://doi.org/10.1109/AIVR.2018.00025

[36] Dennis L. Kappen, Pejman Mirza-Babaei, Jens Johannsmeier, Daniel Buckstein, James Robb, and Lennart E. Nacke. 2014. Engaged By Boos and Cheers: The Effect of Co-Located Game Audiences on Social Player Experience. In *Proceedings of the first ACM SIGCHI annual symposium on Computer-human interaction in play.* ACM, New York, NY, USA, 151–160. https://doi.org/10.1145/2658537.2658687

[37] Shunichi Kasahara and Jun Rekimoto. 2014. JackIn: Integrating first-person view with out-of-body vision generation for human-human augmentation. In *ACM International Conference Proceeding Series.* Association for Computing Machinery. https://doi.org/10.1145/2582051.2582097

[38] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira, and Chedy Raïssi. 2012. Watch me Playing, I am a Professional: a First Study on Video Game Live Streaming. In *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion.* ACM Press, New York, New York, USA, 1181. https://doi.org/10.1145/2187980.2188259

[39] Andrey Krekhov, Daniel Preuß, Sebastian Cmentowski, and Jens Krüger. 2020. Silhouette Games: An Interactive One-Way Mirror Approach to Watching Players in VR. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play.* New York, NY, USA. https://doi.org/10.1145/3410404.3414247

[40] Jie Li, Yiping Kong, Thomas Röggla, Francesca De Simone, Swamy Ananthanarayan, Huib de Ridder, Abdallah El Ali, and Pablo Cesar. 2019. Measuring and Understanding Photo Sharing Experiences in Social Virtual Reality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems.* ACM, New York, NY, USA, 1–14. https://doi.org/10.1145/3290605.3300897

[41] Yunpeng Liu, Stephan Beck, Renfang Wang, Jin Li, Huixia Xu, Shijie Yao, Xiaopeng Tong, and Bernd Froehlich. 2015. Hybrid Lossless-Lossy Compression for Real-Time Depth-Sensor Streams in 3D Telepresence Applications. In *Advances in Multimedia Information Processing – PCM 2015,* Vol. 9314. Springer International Publishing, Cham, 442–452. https://doi.org/10.1007/978-3-319-24075-6_43

[42] Bernhard Maurer, Ilhan Aslan, Martin Wuchse, Katja Neureiter, and Manfred Tscheligi. 2015. Gaze-Based Onlooker Integration: Exploring the In-Between of

Active Player and Passive Spectator in Co-Located Gaming. In *Proceedings of the 2015 Annual Symposium on Computer-Human Interaction in Play*. ACM, New York, NY, USA, 163–173. https://doi.org/10.1145/2793107.2793126

[43] Patrick Mours. 2021. Reshade. https://reshade.me/. (Accessed on 09/01/2021).

[44] Moving Picture Experts Group (MPEG). 2021. MPEG – The Moving Picture Experts Group. https://www.mpegstandards.org/. (Accessed on 05/13/2021).

[45] Fabrizio Pece, Jan Kautz, and Tim Weyrich. 2011. Adapting standard video codecs for depth streaming. In *Joint Virtual Reality Conference of EGVE 2011 - The 17th Eurographics Symposium on Virtual Environments, EuroVR 2011 - The 8th EuroVR (INTUITION) Conference*. 59–66. https://doi.org/10.2312/EGVE/JVRC11/059-066

[46] WebM Project. 2021. The WebM Project | Welcome to the WebM Project. https://www.webmproject.org/. (Accessed on 09/01/2021).

[47] Jana Rambusch, Anna - Sofia Alklind Taylor, and Tarja Susi. 2017. A pre-study on spectatorship in eSports. In *Spectating Play. 13TH ANNUAL GAME RESEARCH LAB SPRING SEMINAR*. 24–25.

[48] Jean-Paul Sartre. 1943. *Being and Nothingness*. Éditions Gallimard. 628 pages.

[49] Max Sjöblom and Juho Hamari. 2017. Why do people watch others play video games? An empirical study on the motivations of Twitch users. *Computers in Human Behavior* 75 (oct 2017), 985–996. https://doi.org/10.1016/j.chb.2016.10.019

[50] Bijan Stephen. 2021. CodeMiko will see you now - The Verge. https://www.theverge.com/22370260/codemiko-twitch-interview-stream-technician. (Accessed on 05/02/2021).

[51] Burak S. Tekin and Stuart Reeves. 2017. Ways of Spectating: Unravelling Spectator Participation in Kinect Play. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, Vol. 2017-May. ACM, New York, NY, USA, 1558–1570. https://doi.org/10.1145/3025453.3025813

[52] Emmanuel Thomas. [n.d.]. The 'MP4' Registration Authority. https://mp4ra.org/. (Accessed on 05/18/2021).

[53] Balasaravanan Thoravi Kumaravel, Cuong Nguyen, Stephen DiVerdi, and Bjoern Hartmann. 2020. TransceiVR: Bridging Asymmetrical Communication Between VR Users and External Collaborators. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 182–195. https://doi.org/10.1145/3379337.3415827

[54] Wolfgang Tschauko. 2021. VR Giants on Steam. https://store.steampowered.com/app/1124160/VR_Giants/. (Accessed on 09/06/2021).

[55] Viswanath Venkatesh. 2000. Determinants of perceived ease of use: Integrating control, intrinsic motivation, and emotion into the technology acceptance model. *Information systems research* 11, 4 (2000), 342–365.

[56] Chiu-Hsuan Wang, Seraphina Yong, Hsin-Yu Chen, Yuan-Syun Ye, and Liwei Chan. 2020. HMD Light: Sharing In-VR Experience via Head-Mounted Projector for Asymmetric Interaction. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*. ACM, New York, NY, USA, 472–486. https://doi.org/10.1145/3379337.3415847

[57] Thomas Wiegand, G.J. Sullivan, G. Bjontegaard, and Ajay Luthra. 2003. Overview of the H.264/AVC Video Coding Standard. *IEEE Transactions on Circuits and Systems for Video Technology* 13, 7 (jul 2003), 560–576. https://doi.org/10.1109/TCSVT.2003.815165

[58] Jacob O Wobbrock, Leah Findlater, Darren Gergle, and James J Higgins. 2011. The aligned rank transform for nonparametric factorial analyses using only anova procedures. In *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. ACM Press, New York, New York, USA, 143. https://doi.org/10.1145/1978942.1978963

[59] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. 2021. Space-time Neural Irradiance Fields for Free-Viewpoint Video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9421–9431.

[60] Hiromu Yakura and Masataka Goto. 2020. Enhancing Participation Experience in VR Live Concerts by Improving Motions of Virtual Audience Avatars. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 555–565. https://doi.org/10.1109/ISMAR50242.2020.00083