



## **TOGETHERVR: A FRAMEWORK FOR PHOTO-REALISTIC SHARED MEDIA EXPERIENCES IN 360-DEGREE VR**

M. J. Prins, S. Gunkel, O.A. Niamut

TNO, The Netherlands

### **ABSTRACT**

Virtual Reality (VR) and 360-degree video are reshaping the media landscape, creating a fertile business environment. In 2016 many new 360-degree cameras and VR headsets entered the consumer market. Distribution platforms are being established and new production studios are emerging. VR is a hot topic in research and industry, and many new and exciting interactive VR content and experiences are emerging. The biggest gap we see in these experiences is the lack of social and shared aspects of VR usage, as today's VR applications tend to be an isolated endeavour. In this paper, we present TogetherVR, a web-based framework for the creation and evaluation of social and shared VR experiences in which users can communicate with a high degree of presence and in photo-realistic video quality. We further elaborate on three multi-user VR cases: watching TV together in VR, social collaboration in VR, and social VR conferencing in a mixed reality setting.

### **INTRODUCTION**

The last few years have seen a major uptake of virtual reality technology, enabling the creation of immersive videogames and training applications, but also paving the way for new forms of video entertainment. Major sports events are being broadcast in 360-degree video<sup>1</sup>, offering consumers new levels of immersion and to experience an event like never before. Also, Hollywood is experimenting with VR as a promotional tool with recent releases of VR movie tie-ins such as Ghost in the Shell VR<sup>2</sup>.

Unfortunately, many VR experiences are still an isolated experience. People wearing VR Head Mounted Displays (HMD) can sometimes feel like being in a different place and do not see or hear their physical surroundings. However, isolation is not a necessary consequence of new media formats: people can and do feel like being in another place with others 'Slater et al. (1)'. Multi-user VR experiences do exist, but these tend to focus on creating artificial experiences, in which users meet in a rendered VR environment, and the users are portrayed by avatars 'Thomas et al. (2)'. Avatar-based approaches however

---

<sup>1</sup> <http://digitalsport.co/how-sports-teams-and-brands-using-360-degree-video>

<sup>2</sup> <http://dragons.org/creators/recent-work/work/ghost-in-the-shell/>



may be too restrictive for interactions in many use cases where non-verbal communication is important, such as video conferencing, presentations, watching 360-degree videos together, and many more.

In this paper, we present our ongoing efforts towards the creation of social and shared VR experiences in which users can communicate with a high degree of presence and in photo-realistic video quality. For this purpose, we leverage the web browser as a VR-enabled application platform by utilizing emerging web technologies for audio-visual communications and currently available off-the-shelf hardware. The goal is an easy deployment as well as easy access in the home for interactive shared VR applications.

## RELATED WORK

Many 360-degree video delivery solutions and trials for live VR have been launched in 2016. Most major sports are offering some sort of VR-experience, enabled by tools from e.g. NextVR<sup>3</sup> and VideoStitch<sup>4</sup>, which offer virtual experiences for individual use. Social VR experiences have been enabled by AltspaceVR<sup>5</sup> and vTime<sup>6</sup>, who offer virtual spaces where users, represented as avatars, can come together and interact and possibly play games.

Further, companies like Facebook and LivelikeVR<sup>7</sup> are enabling shared experiences between VR users at remote locations, in which they can interact via avatars while watching omnidirectional video, whilst video service Hulu allows Gear VR users to watch regular video with friends in VR. The avatars typically reflect user movements, as captured by the user's HMD, external sensors or controller input. While this is a step in the right direction, we believe that offering a photo-realistic view of other people in a shared VR space will further increase social experiences.

One enhancement for real-time avatars is the use of highly detailed photorealistic point clouds. However, we currently lack efficient 3D point cloud video compression and transmission methods as presented by 'Doumanoglou et al. (3)'. A demonstration of networked streaming of humans as captured by a depth-camera was shown by 'Woodworth et al. (4)'. While the technology around 3D point clouds are constantly improving, with our work we aim to close the gap of utilizing current technology for social VR.

## ENABLING SOCIAL VR WITH WEB TECHNOLOGY

To be able to rapidly prototype and evaluate social VR experiences we focus on the web browser as a target application platform. Developing applications on top of web technology brings many advantages: a modern web-browser comes equipped with many multimedia

---

<sup>3</sup> <http://www.nextvr.com/>

<sup>4</sup> <http://www.video-stitch.com/>

<sup>5</sup> <https://altvr.com/>

<sup>6</sup> <https://vtime.net/>

<sup>7</sup> <http://www.livelikevr.com/>

features: secure and adaptive video streaming and playback (HTML5 video, Media Source Extensions and Encrypted Media Extensions), real-time audio- and video communications (WebRTC), positional audio (Web Audio) and support for VR (WebVR and WebGL). Additionally, these multimedia capabilities are supported by a broad range of web browsers on many devices and not tied to specific HMD vendors or operating systems. Currently however the web-based approach to VR has some clear drawbacks in terms of performance, compared to native solutions (for instance created in Unity or Unreal Engine). Still we see in WebVR the potential as an enabler for wide and easy deployment of VR applications and to make them massively available. Thus, with our work we also like to identify further bottlenecks and help to improve and stretch how current web technologies can be supported in or combined with WebVR.

## FRAMEWORK AND ARCHITECTURE

We developed a web-framework called TogetherVR for enabling, testing and evaluating social VR experiences enabled by the above stated web browser features. We earlier reported on this framework in 'Gunkel et al. (5)'. The framework is based on Node.js/Express, Angular.js and A-Frame<sup>8</sup>. A-Frame is a popular and easy to use application framework for creating WebVR applications on top of HTML. It enables web developers to create VR applications without diving into the technical aspects of WebGL and WebVR. But more importantly, A-Frame supports embedding HTML elements like videos and images inside a VR environment, allowing us to stream video or show the webcam-feed of another users. This allows us to create virtual environments in which we can place participants in a photo-realistic 360-degree panoramic room, overlaid with media-elements such as videos, screen shares or captures of other participants.

An overview of our web framework is given in Figure 1. TogetherVR consists of a frontend component, which completely runs in the web browser, and a backend component, which hosts the VR applications, and facilitates inter-user communication and media playback control. Users connect to TogetherVR via a web browser on a laptop, with several peripheral devices, such as an HMD, a camera and gamepad.

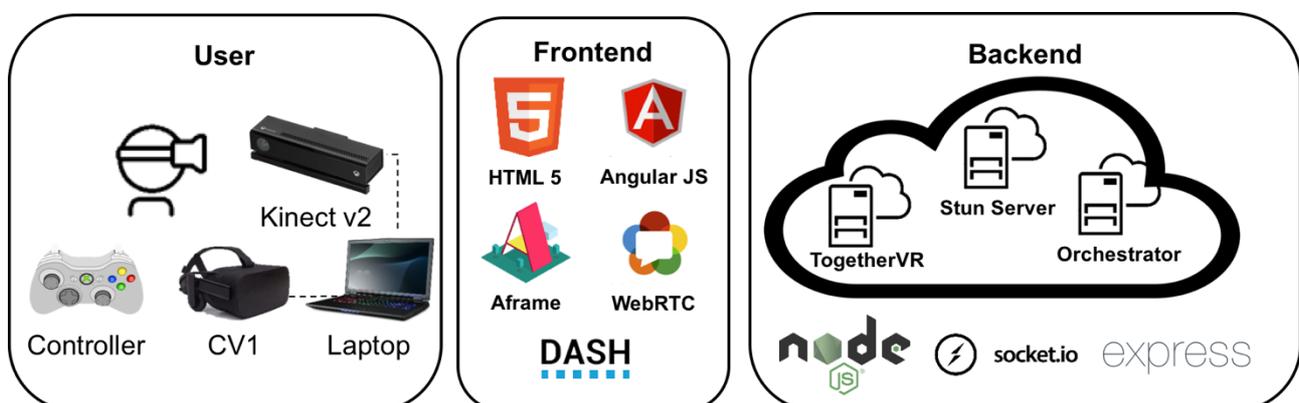


Figure 1 – Overview of the TogetherVR application framework.

<sup>8</sup> <https://aframe.io/>



The entry point to our application is provided by the TogetherVR server. This server offers a modular web application based on AngularJS and Node/Express. The application is based on five modules, (i) to display content in VR, (ii) to setup WebRTC connections, (iii) to watch video delivered via MPEG-DASH, (iv) to share interactive content (like a video game), and (v) an administration console to dynamically manipulate properties of individual clients on the fly:

(i) The VR room utilizes the A-Frame framework, which allows to view VR content with a VR HMD, such as a Oculus Rift, in a web browser. Furthermore, we developed a shader to alpha-blend people into the environment. We record users either with a normal camera against a green-screen background or use a depth-sensing camera with a Kinect to remove the background and convert it into an alpha channel.

(ii) The SimpleWebRTC<sup>9</sup> library is used to support direct communications between users, via audio and video (currently we use mono audio and a resolution of 960x540).

(iii) The dash.js<sup>10</sup> enabled video client can play any type of MPEG-DASH video. We synchronize the playout of video between clients via the Orchestrator.

(iv) We use the browser's screen share functionality to create a video stream of any kind of application window. This stream can be shared remotely or simply shown locally. For instance, we can start a game on each of the two computers and display it in our interactive VR living room. Or share a web page, presentation, etc.

(v) All client properties can be modified in real-time. Particularly, this means the content being displayed (e.g. a game or movie), but also the correct placement of objects.

## EXAMPLE SCENARIOS

The framework allows us to rapidly create compelling VR experiences in which VR participants can jointly experience watching video, playing a game or collaborate in a 360-degree VR environment while seeing each other in VR. We have currently created three interactive proof-of-concept applications: a social VR TV experience, an interactive social VR experience and social VR conferencing in a mixed reality setting. The former two experiences have already been concluded with a small user experiment. We will report on these experiences in the following.

### Social VR TV experience

The first application we created allows two VR participants to remotely watch TV together, as if the users are sitting next to each other on a couch in a living room. Participants are recorded with a webcam from the side against a green screen background on a normal chair (Figure 2). The resulting video stream is transmitted to the other participant via WebRTC and rendered in A-Frame. The video is aligned in the VR environment, such that it appears that the other participant is sitting next to you on a couch (Figure 3), with the

---

<sup>9</sup> <https://simplewebrtc.com/>

<sup>10</sup> <https://github.com/Dash-Industry-Forum/dash.js/>

audio spatially aligned such that his/her voice comes from that direction as well. The room itself consists of a 360-degree panoramic photo of our medialab.

Recording users against a green screen allows for easy blending of the webcam feed in the virtual space of the other user via chroma-keying. In the VR space, the users can watch a video projected on a large video screen (

Figure 4), the playback being synchronized, such that users can experience the content at the same time.



Figure 2 - Capturing a VR user against a green screen background.



Figure 3 - A user rendered in the VR space.



Figure 4 - A video projected in the photorealistic VR space.

We made several important observations from our experiment with this setup. The first was the effectiveness of communication: people appreciated seeing each other, even though part of their face was occluded by the HMD. But they could see the person laugh, talk and they could see some non-verbal expressions. Also, the positional audio effectively worked as cues for the participants: when one user started talking, the other would hear this sound from the left (or right) and this user would automatically look in this direction (and then see the other participant). The participants however were distracted by the lack of a self-view, i.e. people could not see their own body and mentioned the lack of interaction possibilities with the VR space itself.

### Interactive Social VR experience

For our second experiment, we extended our framework to support shared interaction with objects in the VR space next to conversing with each other. We integrated a pong-like game<sup>11</sup> to be shown and played instead of watching TV, with the game view being synchronized between participants (Figure 5). The users control the paddles via gamepads and try to defeat the opponent by getting the ball behinds



Figure 5 – VR view of another user who is participating in a game of Pong.

<sup>11</sup> <http://blog.nikc.org/2012/02/11/how-i-created-a-pong-clone-using-node-js-canvas-websocket-and-deviceorientation/>

the opponents paddle. The game itself is rendered on a HTML5 canvas which we then project in the VR space.

One of the things we noticed when we ran the experiment was that people did not comment about the lack of the self-view in the interactive setting. We believe this was caused by the fact that people needed to focus on what was happening on the screen and act accordingly: they had to control the paddle with a gamepad to prevent the opponent from scoring. So, the addition of a shared interactive activity in which actions of the user directly impact the other user helped increase the sense of immersion. Or stated differently: it withdrew the attention from the users to limitations of the VR environment, for instance the fact that they could not see their hands.

### **Social VR conferencing in a mixed reality setting**

For our third experiment, we explored to what extent VR can be used to bring a remote user into a meeting room, the meeting room containing several people. In other words, a VR alternative to dialling into a teleconference. The main incentive for this experience is to allow a person in VR to interact and communicate with people outside of VR. For this purpose, we combined a static 360-degree background image of a room, which is overlaid with a live video-feed of that same room that is being recorded to capture the participants currently present in that room, as can be seen in Figure 6.

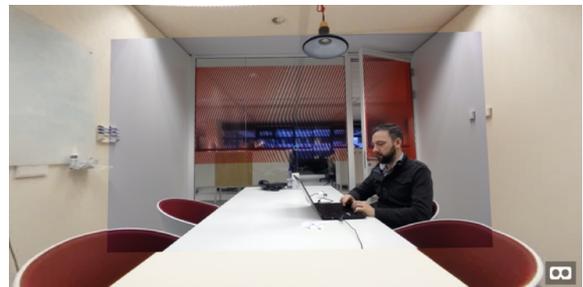


Figure 6 - Shared collaboration in a mixed reality setting.

We also explore how we can facilitate screen sharing, such that the VR participant for instance can have a view of an application being shared by one of the other participants. Screen sharing is still a novel and experimental feature of WebRTC, but our early tests show that we can share any type of content and show this in the VR space. For instance, a standalone application, such as a game can be rendered in the VR space (see Figure 7). We are yet to evaluate this feature and see to what extent we can use this for interactivity purposes as well.



Figure 7 - Embedding non-browser applications via WebRTC screen share functionality.

## EVALUATION

We recently had the chance to evaluate our interactive social VR application with a larger audience. We held a 3-day demo session in which 75 participants played a two-player game (Pong) in an informal and uncontrolled setting at a conference space (Figure 8). This version used an updated version of our framework, in which the users were captured using a depth-camera (a Microsoft Kinect v2), such that we could perform background removal without having access to a flat background with a green screen. Due to venue limitations, we unfortunately could not physically separate the participants of the experiments, limiting a proper evaluation of audio-components. We collected feedback from the participants through a short questionnaire. We evaluated the quality of the (visual) experience and the sense of (co-)presence, asking participants to rank their experience on a 5-point Likert scale. People appreciated the overall quality (77% of the participants scored 4 or higher) of the experience. They also felt involved in the virtual environment experience (72% of the participants scored 4 or higher). Overall, we saw a high degree of activity and interaction between users, mainly because the video quality and delay between people was perceived as “good”. In informal conversations after the demo, people did not report on any experience of motion sickness. The latency of the game itself was mentioned often as a distracting factor. Further, not seeing yourself (lack of self-view) as well as standing up or getting too close to the camera were mentioned as problems by the users.



Figure 8 - Evaluation setup.

## CONCLUSIONS

With our web-based approach to social VR we have developed a framework in which we can rapidly test and evaluate VR experiences. From our experiments and proof-of-concept applications, we see that VR can be made more social by leveraging real-time communication in VR settings using the multimedia functionalities already available in modern web browsers. Our approach of capturing users with a camera and blending them in the VR space proved promising. Users were not really concerned with seeing the other participants wearing a HMD, as people could still see non-verbal communications which was not possible before. Also, users had a high degree of immersion, high degree of presence and overall pleasant communication interactions. Overall with this web framework we have a general testbed, that allows us to quickly execute more studies, to evaluate different technology and its impact to the user perception (i.e. immersion, interaction and quality).



## FUTURE WORK

As the biggest challenge for social VR applications we currently see the following problems, that we also intend to address in the future:

- Being able to see yourself. Currently participants cannot see their own body. A common approach taken in VR is rendering virtual arms that follow the movement of real arm movements. We are investigating to what extent we can create a self-view which allow you to see your own body.
- High immersion and presence for communication. We made a start with enabling photorealistic representations of other people, but in our current approach the HMD is still visible, so we intend to apply HMD removal techniques. Also, we aim to look at 3D and point-cloud based video rendering to improve the VR experience.
- Being able to interact and engage in VR with more users simultaneously. Our current examples focus on social VR experiences for two users. In the future, we will look at accommodating more users at the same time, with all the associated communication challenges, both in terms of network and user interactions.

Additionally, we like to explore other use cases to test and evaluate social VR. We are currently aiming at creating a social virtual skybox experience, in which you can watch a sports game or a concert with your friends. We also plan to investigate to what extent social VR can be utilized for educational services, for instance remote participation in a class room for sick or immobile people.

## REFERENCES

1. Slater M., Steed A., Usoh M., 2013. Being There Together: Experiments on Presence in Virtual Environments (1990s). Technical Report, Dep. of Computer Science, University College London, UK
2. Thomas J., Bashyal R., Goldstein, S., Suma E., 2014. MuVR: A Multi-user Virtual Reality Platform. Proc. IEEE VR 2014. pp. 115-116.
3. Doumanoglou, A. Alexiadis D. S. Zarpalas D. and Daras, P. (2014). Toward real-time and efficient compression of human time-varying meshes, IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 12, pp. 2099–2116
4. Woodworth, J.W. Ekong, S. and Borst, C.W., 2017. Virtual Field Trips with Networked Depth-Camera-Based Teacher, Heterogeneous Displays, and Example Energy Center Application, Proc. IEEE VR 2017. pp. 471-472
5. Gunkel S., Prins M., Stokking H., and Niamut O., 2017. WebVR Meets WebRTC: Towards 360-Degree Social VR Experiences, Proc. IEEE VR 2017. pp. 457-458