Mixed Reality Experience
How to Use a Virtual (TV) Studio for Demonstration of Virtual Reality Applications

Jens Herder[1], Philipp Ladwig[1], Kai Vermeegen[1], Dennis Hergert[1], Florian Busch[1], Kevin Klever[1],
Sebastian Holthausen[1] and Bektur Ryskeldiev[2]

[1] *Faculty of Media, Hochschule Düsseldorf, University of Applied Sciences, Münsterstr. 156, 40476 Düsseldorf, Germany*

[2] *Spatial Media Group, University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan*

Keywords: Virtual Reality, Mixed Reality, Augmented Virtuality, Virtual (TV) Studio, Camera Tracking.

Abstract: The article discusses the question of "How to convey the experience in a virtual environment to third parties?" and explains the different technical implementations which can be used for live streaming and recording of a mixed reality experience. The real-world applications of our approach include education, entertainment, e-sports, tutorials, and cinematic trailers, which can benefit from our research by finding a suitable solution for their needs. We explain and outline our Mixed Reality systems as well as discuss the experience of recorded demonstrations of different VR applications, including the need for calibrated camera lens parameters based on realtime encoder values.

1 INTRODUCTION

Virtual reality (VR) applications for head mounted displays (HMD) have recently become popular because of advances in HMD technology, including tracking and computer graphics, and decrease in costs of VR-related hardware. The VR applications are often promoted or explained through videos, which become more attractive and understandable when users are shown as part of the virtual environment. The process of creation of such mixed reality videos is explained by (Gartner, 2016).

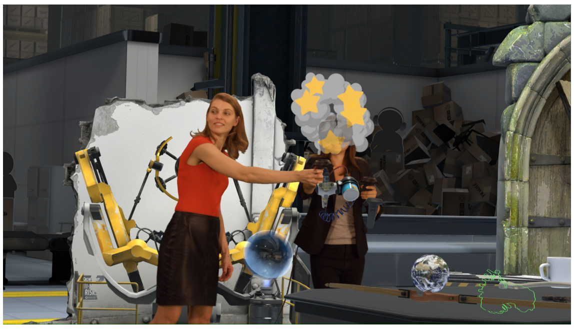

Figure 1: External camera view with actors in the VR application "The Lab," interacting with balloons.

The Unity game engine supports mixed reality environments via (Valve Corporation, 2017b) plug-in. The plug-in offers a special rendering mode for an external camera which can be tracked. We discuss such a simple setup and bring it to a more sophisticated level in a virtual studio with professional studio equipment including camera, camera tracking, and hardware keyer. We integrated a professional camera tracker using a developed OpenVR driver and used the system for a demonstration video for evaluation[1]. An introduction to virtual studios can be found in (Gibbs et al., 1998).

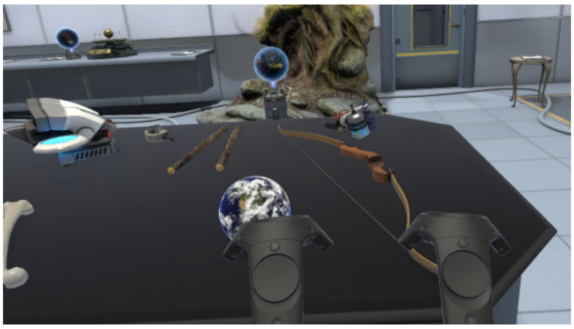

Figure 2: Egocentric view with the HMD within the VR application "The Lab".

[1] Demonstration video is available at http://vsvr.medien.hs-duesseldorf.de/productions/openvr/.

Herder, J., Ladwig, P., Vermeegen, K., Hergert, D., Busch, F., Klever, K., Holthausen, S. and Ryskeldiev, B.
Mixed Reality Experience - How to Use a Virtual (TV) Studio for Demonstration of Virtual Reality Applications.
DOI: 10.5220/0006637502810287
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 1: GRAPP*, pages 281-287
ISBN: 978-989-758-287-5

2 PRODUCTION IMPLEMENTATION

2.1 Views and Cuts: Creative Process

For a virtual environment broadcast we use several different views, to create a compelling story. We distinguish between exocentric (see Figure 1) and egocentric (see Figure 2) views. Those can be mixed by using windows, split screens, and cuts. The egocentric view (also known as subjective shot (Mercado, 2013)) from inside of HMD is easy to record and represents the user experience inside the virtual environment, allowing viewers to see the scene from an actor's viewpoint. However, in such case showing actors' interaction can be challenging, since their hands or body might not be visible. Furthermore, it is also hard to visually convey emotions and facial expressions, such as smiling, to spectators. One of the workarounds in such cases is recording audio in order to capture laughing or other emotional expressions to alleviate the lack of visual display.

The exocentric view ("bird's eye view" or third person perspective) provides a more complete overview of a scene. In such case the full body of an actor is visible, showing motions and emotions. However, most 3D user interfaces, such as a menu floating in space or attached to a user's perspective, are designed for egocentric view, and do not work well with exocentric perspectives. Placing an external camera behind a user or instructing a user to hold controllers in such a way that those are oriented towards the camera helps but limits the live experience.

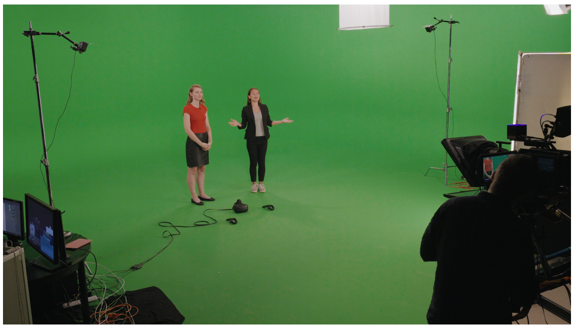

Figure 3: Studio stage with lighthouse and Vive.

2.2 Studio Camera Tracking

Our first approach utilized a DSLR camera and a controller with a 3D-printed adapter (seen in Figure 4[2]). The DSLR was connected to a computer and mixing and keying were done in accompanying software. In standard virtual studio setups, the camera signal is delayed before mixing to compensate for time used by tracking and rendering. With the DSLR, the video signal over USB had too much delay and synchronized mixing was not possible.

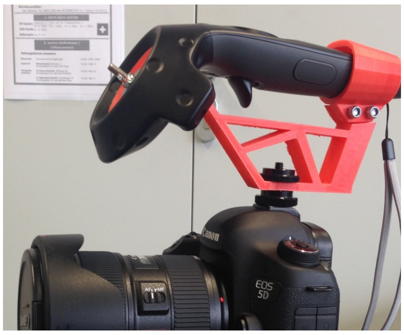

Figure 4: DSLR camera with Vive controller and custom-printed adapter.

The generic HTC Vive tracker in Figure 5 simplifies the setup. Furthermore, a webcam does not have such a high delay while connected over USB. Both approaches require a time-consuming calibration process and lens parameters cannot be changed during broadcast. Even an autofocus has to be switched off because focus also changes the field of view (as seen in Figure 7).

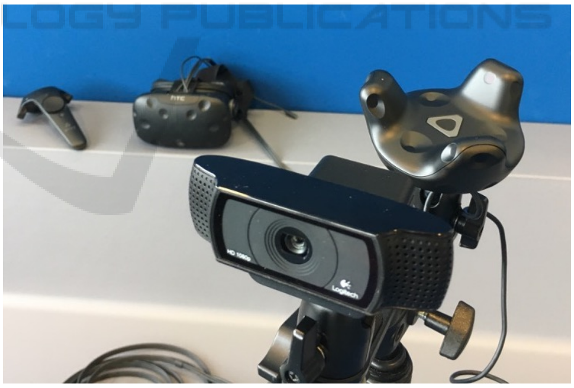

Figure 5: Webcam with lower delay with Vive tracker.

Professional virtual studio cameras have either lens encoders (seen in Figure 6) or a lens which has a serial output for zoom and focus data.

In a onetime process, a professional studio camera and its lens get calibrated for providing field of view, nodal point shift, center shift, lens distortion, and focal distance. Figure 7 shows that field of view depends on zoom and focus in a non linear way. Depending on the lens, the focus might change the field of view dramatically (in this example up to 8 degrees).

[2] Downloadable freely from the accompanying website.

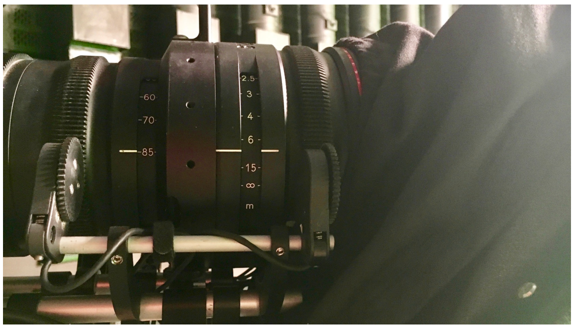

Figure 6: Lens encoder for zoom and focus.

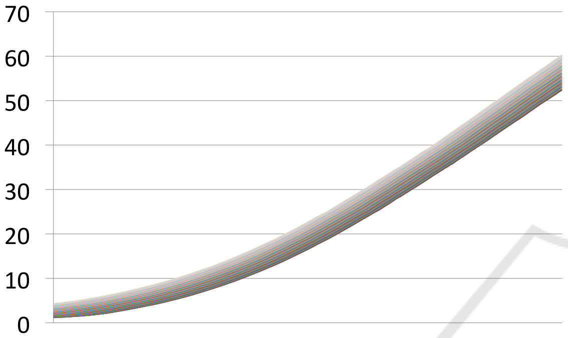

Figure 7: Calibrated field of view with the parameters zoom (horizontal) and focus (color) for the Canon KJ17ex7.7B-IASE, the maximal influence of focus is 8°.

Lens distortion must also be applied to the graphics so that virtual and real objects stay in place for all camera changes. Usually lens distortion is rendered using a shader. Figure 8 shows distortion with the k_1 parameter with a negative value, generating a barrel distortion. With the k_2 parameter, the distortion can be better modeled. The lens calibration is especially important for AR applications because real and virtual objects are in contact all over the screen.

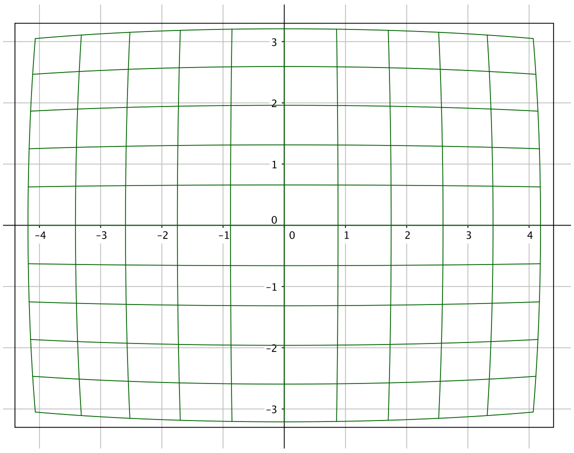

Figure 8: Distortion with the parameter $k_1 = -0.0025$.

Using a controller or tracker of a HMD system is a very cost-effective solution. However, this comes with a cost: the camera motions are limited to the relatively small tracking area of the HMD's tracking system. Especially if a production in a virtual studio is desired, the limited tracking space has to be considered because the studio camera is most of the time outside of the tracking area of the HMDs tracking systems. That problem gets even worse, if recordings with a high focal length are desired and a long distance between camera and actors is required. For covering a large tracking area external tracking systems can be used. Figure 9 shows the StarTracker tracking system, based on an auxiliary camera with LED ring on the studio camera facing to the ceiling with retro-reflective marker. It is an inside-out tracking which allows covering a much larger area compared to a common HMD tracking system.

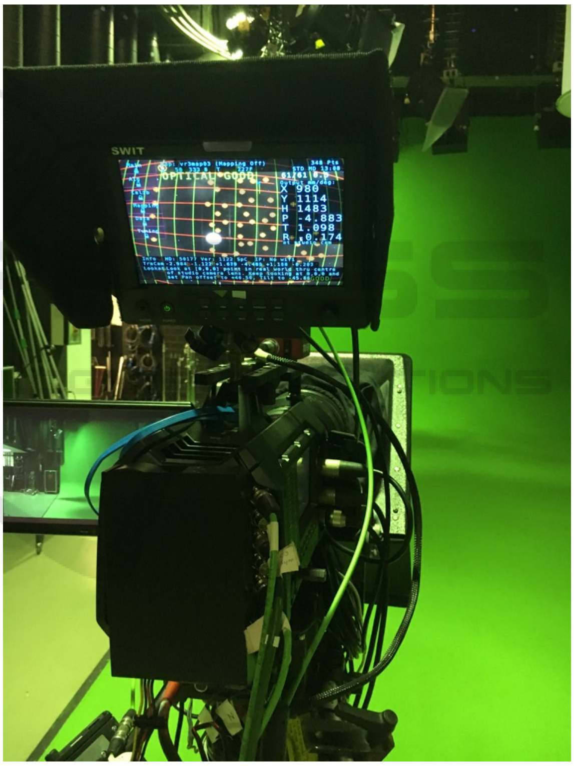

Figure 9: Monitoring the StarTracker in the viewfinder.

In our studio production, we deployed the Vizrt Tracking Hub[3] (see Figure 10) for interfacing camera tracking data and game engine and created an OpenVR driver (Valve Corporation, 2017a). This approach enables us to use many different studio camera tracking systems and also ease the configuration setup and matching of the different coordinate systems (i.e., HMD and studio camera).

[3] www.vizrt.com

2.3 Mixing and Keying

Figure 10 shows the signal flow of video and tracking data. The keying and mixing takes place in the external hardware chroma keyer from Ultimatte (Smith and Blinn, 1996). In this case the rendered image includes graphics which are before and behind the actors (standard implementation). The mask determines which part is transparent or is in front. This does not work well for all situations. If the foreground is translucent and the background contains a pattern then this will be added to the video image so that the background shines through the actor. Table 1 classifies different implementation possibilities. An advanced approach renders foreground and background graphics including a mask for the foreground separately requiring at least three video outputs. The most sophisticated method, “internal,” does not take a layer approach and requires that the engine handles video input and keying, which goes beyond standard implementations.

2.4 Timing, Delays, and Rendering Processes

Tracking systems for HMDs and controllers is optimized for low latency and is often based on more than one sensor type, such as optical and inertial sensors. Different sensors types have different update frequencies and delays. Adequate sensor fusion algorithms merge the data and take into account correct timings between different sensor types. In a Mixed Reality setup, such as that shown in Figure 10, an external studio camera tracking systems is added which is not part of the tracking system of the HMD and hence is not processed by the sensor fusion algorithms of the HMD’s tracking system. Additionally, the real-world image of the studio camera takes time for transmitting and have to be synchronized as well. The synchronization of these three systems (HMD tracking, external camera tracking, and transmission of the camera image) is compelling for good visual results. Usually the tracking of the external camera is the slowest subsystem and the other subsystems have to be delayed until all systems are synchronized. This implies adding a delay to the HMD’s controllers for the exocentric view. It has to be considered that only the controllers of the exocentric view must be delayed since a delay of the controllers for the egocentric view would adversely affect the user’s interaction with virtual objects.

Furthermore, other systems must be taken into account for synchronization. A camera tracking system also uses several sensors, which have again different delays. Mostly the delay of lens encoders (see Figure 6) differs from the camera position tracking. Often those delays are already compensated and configured within the studio camera tracking system. Moreover, the image sources have different signal flows and require different delays as well. When switching or blending between the different views (external camera vs. HMD or exocentric vs. egocentric) also the timing has to be aligned.

Utilizing two different tracking systems (tracking of the HMD and tracking of an external camera such as StarTracker shown in Figure 9) requires an exact match of both coordinate systems. Not only the orientation and origin of both systems must match but also the format of data. As the case may be Euler degrees and quaternions have to be translated as well as units or different decimal powers must be recalculated.

Realistic rendering and interaction with virtual objects can be achieved by occlusion of actors which can be seen in Figure 1. The actor with the HMD stands behind the table. This is possible since the render engine knows the position of the HMD (which is the position of the actor) and renders the foreground layer and the alpha mask accordingly. The rendering of foreground and mask is achieved by locating the far clipping plane at the position of the HMD which avoids the rendering of virtual object behind the actor. The foreground layer and alpha mask are mixed and composed in a final step for achieving the broadcast stream with occlusion. But incorporating occlusion works only for the actor who wears the HMD. An optional marker-less actor tracking systems (Daemen et al., 2013) can translate the physical location of every actor into the respective virtual location in the virtual scene which allows incorporating occlusion for every actor.

The combination of a game engine with the hardware of a virtual studio requires special software, since SDI is the standard for the transmission of video signals. We developed a software which records the render window of the game engine and transmits the captured window via SDI to the chroma keyer. We utilize a Rohde & Schwarz DVS Atomix LT video board for streaming which supports two SDI output channels.

2.5 Actor Augmentation

For enhancing storytelling, actors can be augmented partly with computer graphics. The pose of head mounted display is available within the system, so attaching a helmet would not be hard to implement, but the original goal of conveying emotions would be less achievable because a face would be completely cov-

Table 1: Various mixing implementations.

Implementations	Layers[a]	Restrictions	Benefits
simple	BG, V, VM	no graphics in front of actors	low cost, simple
standard	(BG+FG), M, V, VM	backgrounds shines through video in case of translucent foreground	used in most virtual studio setups
advanced	BG, FG, M, GM, V, VM	more than two rendering outputs, special hardware keyer	correct mixing with translucent objects
internal	V, VM, Scene	resampling of the video can lower image quality	special effects like distortion of the video or reflections of the video onto the scene

[a] BG = background graphics; V = Video from the studio camera; VM = Mask from the chroma keyer
FG = Foreground graphics; M = Mask for the foreground graphics, controlling transparency
Scene = Three-dimensional scene not layer, video input is used as video texture

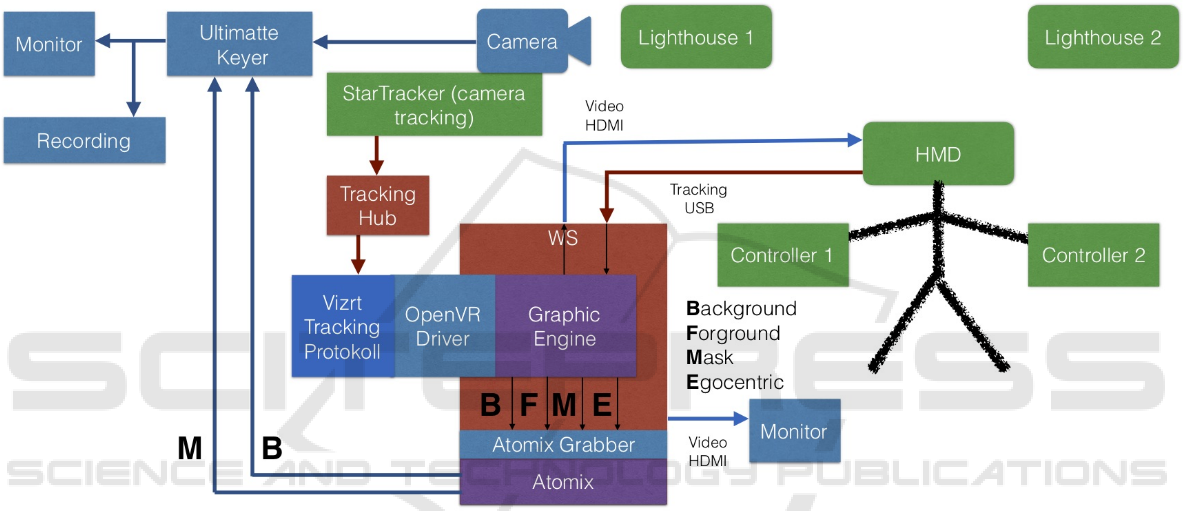

Figure 10: Signal flow for our mixed reality production with HTC Vive.

ered. Depending on a story it could be entertaining. If you have more motion data available like using a markerless actor tracking system, then other parts of a body can be augmented for creating a cyborg (Herder et al., 2015). For recreating the emotions based on eye sight and eye contact, a head mounted display can be augmented by an animated actors face either using eye tracking (Frueh et al., 2017) or just relying on the image of the rest of the face (Burgos-Artizzu et al., 2015). In a standard setting, the controllers get rendered also in the external camera view, thus occluding the real hand and real controllers. Depending on the application, it would be better to switch off that rendering and just showing the real controllers. When actors come to close to the graphics, the occlusion cannot be handled properly because rendering uses only one distance from camera to actor. If mixing would use a depth image of the actor, occlusion could be handled for each pixel correctly.

2.6 Feedback to Presenters and Acting Hints

Since the presenters in Figure 1 without a HMD cannot see the virtual environment while talking to a user with a HMD, it is important to provide some other kind of feedback to them (Thomas, 2006). The presenter looks too obviously at monitor to get informed what happens in the virtual environment. Besides common use of monitors outside the view of the external camera, other approaches project in keying color onto the background, place monitors in keying color, use proxy objects (Simsch and Herder, 2014), or explore spatial audio (Herder et al., 2015) for feedback.

Table 2: Various productions setups for mixed reality recordings.

	Post production	Simple	Virtual studio with fixed lens	Virtual studio with variable lens
description	rendering output and video feeds will be recorded and mixed in a post production	live mixing in the workstation	external chroma keyer and video delay	calibrated lens
camera lens	fixed	fixed	fixed	variable zoom and focus
application area	trailer	trailer, low budget streaming	broadcast	broadcast
live streaming	no	yes	yes	yes
problems	long workflow	wrong delays for graphics and video	no zoom and focus	precise lens calibration necessary
advantages	mistakes can be corrected	easy setup		camera can be used freely
costs	high	low	middle	high
production quality	high	low	middle	high

2.7 Production Classification and Comparison

Table 2 discusses the various productions setups for different applications, considering costs and implementation details. Post production allows corrections of the recording and can be used to adjust for different delays. While this approach might lead to high quality, inherently no live broadcasting is possible. The simple approach with using a tracker or controller from the HMD system as a camera tracker is cost effective but limits the camera deployment area and does not allow changes in zoom and focus, which is a strong handicap for a good camera work. We experienced tracking problems with the HMD tracker because of the strong light in the studio. We reduced the light, which led to lower quality in the key and noise close to the floor. Using a professional studio for production generates high costs but provides best quality. For reaching broadcast quality, lens parameters need to be calibrated and used for the rendering.

3 OPENXR PROPOSAL

We propose to standardize a mode for mixed reality broadcasting of a scene with an external camera. This could take place within the OpenXR[4] working group. An interface driver should support all parameters of a studio camera tracking system as outlined in Table 3. The driver itself would be implemented by different vendors using their specific tracking protocols. How distortion is rendered needs to be specified or distortion rendering has to be implemented by a driver similar to the distortion for the lenses of a HMD. Providing a calibrated focus distance could be also important for next generations of HMDs with eye tracking while the focus point has not necessarily located on the line orthogonal to the sensor. Delays for all devices, especially controllers, which contribute to mixed reality rendering needs to be configurable. In general, when virtual and real objects are close to each other, the mixing does not work well. Therefore, the graphics for interaction devices should be configurable for rendering of external view, as graphics usually do not overlay properly with user's hands.

Table 3: Additional parameters for camera tracking.

parameters	unit	additional information
horizontal and vertical field of view	degrees	
distortion k_1 and k_2	co-efficients	in lens coordinates, distorted pixel $\vec{p}'$[a]
nodal point shift	m, same as pose	
focus	m, same as pose	calibrated distance

[a] $\vec{p}' = u(\vec{p} + \vec{c}) - \vec{c}$ with $u = 1 + Rk_1 + R^2k_2$; R is the distance from image center; c is the center shift; see also Figure 8

[4] https://www.khronos.org/openxr

4 EVALUATION

We interviewed four experts from the broadcasting industry regarding the different ways to present a virtual reality application to a broader audience. All rank the different view modalities nearly equally important: Egocentric (first person) view, exocentric view of how a person is using the HMD, exocentric view within a virtual scene. All agree that it is necessary to show egocentric and exocentric perspectives together. One expert commented that it is important to be able to observe the interaction not only from the actor's point of view but also from a distance. A sense of space in time is crucial for film making. Another expert stated that the personal experience of the actor is an important emotional feedback. If we can notice how and why the protagonist is acting in a certain way, we have a chance to enact actions like in a film-plot. All experts agree that is important to use a camera with zoom and focus during production. Half of the experts think that the produced video is better than other videos that do not show the users within the virtual environment. All experts felt immersed while watching the produced video.

5 CONCLUSIONS

We introduced and discussed different implementations of virtual studio productions for virtual reality applications. For achieving best quality and free studio camera use, the standard interface as well as the rendering process must be extended. Most important missing parameter for the rendering process is a variable field of view, provided by a camera tracking system, but others cannot be neglected either. If this becomes standard, then it will be easy to bring any virtual reality application into a virtual studio without additional modifications. Showing persons' interactions using VR devices is not only instructive, but also empathic (i.e. conveying user emotions).

ACKNOWLEDGEMENTS

Bianca Herder and Ida Kiefer contributed as presenters. Vizrt Austria GmbH supported this project by providing the viz tracking hub software. The production took place at vr3 / Voss GmbH TV Ateliers, where Dirk Konopatzki and Jochen Schreiber were very helpful. Thanks belong to Christopher Freytag and Antje Müller for getting the studio equipment running. First experiments with the custom camera adapter were done with Thomas Stütz and his class at the FH Salzburg.

REFERENCES

Burgos-Artizzu, X. P., Fleureau, J., Dumas, O., Tapie, T., LeClerc, F., and Mollet, N. (2015). Real-time expression-sensitive hmd face reconstruction. In *SIGGRAPH Asia 2015 Technical Briefs*, pages 9:1–9:4, New York. ACM.

Daemen, J., Haufs-Brusberg, P., and Herder, J. (2013). Markerless actor tracking for virtual (tv) studio applications. In *Int. Joint Conf. on Awareness Science and Technology & Ubi-Media Computing*, iCAST 2013 & UMEDIA 2013. IEEE.

Frueh, C., Sud, A., and Kwatra, V. (2017). Headset removal for virtual and mixed reality. In *ACM SIGGRAPH 2017 Talks*, SIGGRAPH '17, pages 80:1–80:2, New York. ACM.

Gartner, K. (2016). Making high quality mixed reality vr trailers and videos. http://www.kertgartner.com/making-mixed-reality-vr-trailers-and-videos.

Gibbs, S., Arapis, C., Breiteneder, C., Lalioti, V., Mostafawy, S., and Speier, J. (1998). Virtual studios: An overview. *IEEE Multimedia*, 5(1):18–35.

Herder, J., Daemen, J., Haufs-Brusberg, P., and Abdel Aziz, I. (2015). Four metamorphosis states in a distributed virtual (tv) studio: Human, cyborg, avatar, and bot - markerless tracking and feedback for realtime animation control. In Brunnett, G., Coquillart, S., van Liere, R., Welch, G., and Váša, L., editors, *Virtual Realities*, volume 8844 of *Lecture Notes in Computer Science*, pages 16–32. Springer International Publishing.

Mercado, G. (2013). *The Filmmaker's Eye: Learning (and Breaking) the Rules of Cinematic Composition*. Taylor & Francis.

Simsch, J. and Herder, J. (2014). Spiderfeedback - visual feedback for orientation in virtual TV studios. In ACE'14*, 11th Advances in Computer Entertainment Technology Conf.*, Funchal, Portugal. ACM.

Smith, A. R. and Blinn, J. F. (1996). Blue Screen Matting. In *Computer Graphics Proceedings, Annual Conferences,* SIGGRAPH, SIGGRAPH '96, pages 259–268, New York, NY, USA. ACM.

Thomas, G. A. (2006). Mixed reality techniques for tv and their application for on-set and pre-visualization in film production. In *International Workshop on Mixed Reality Technology for Filmmaking*. BBC Research.

Valve Corporation (2017a). *OpenVR API Documentation*. https://github.com/ValveSoftware/openvr/wiki/API-Documentation.

Valve Corporation (2017b). SteamVR Unity Plugin. https://github.com/ValveSoftware/steamvr_unity_plugin.