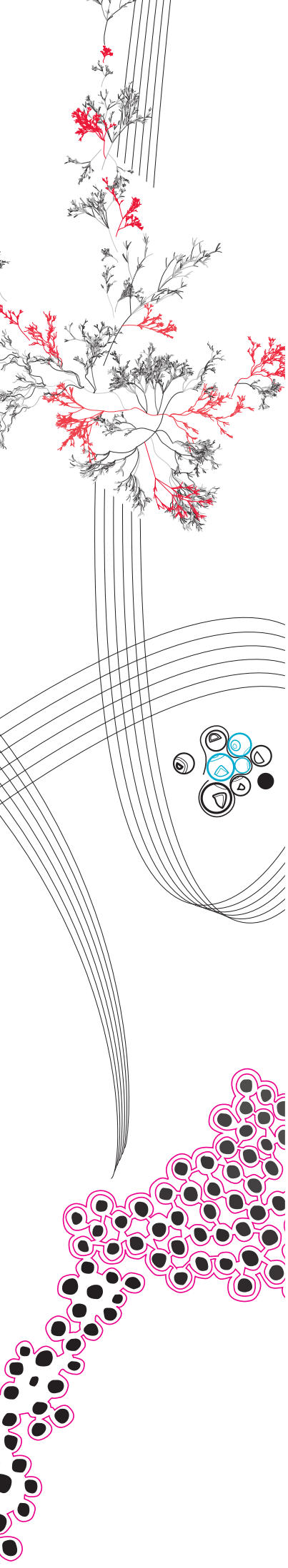

Video Quality in VR

Measuring Impact of Bandwidth-Saving Approaches on Quality of Experience and Social Interaction in a HMD-mediated Video Application.

Master thesis at TNO &
University of Twente

Student:	**Jonathan Simsch**
Track:	**M.Sc. HCID**
1^{st} Supervisor UT:	**Jan van Erp**
1^{st} Supervisor TNO:	**Hans Stokking**
2^{nd} Supervisor UT:	**Dirk Heylen**
2^{nd} Supervisor TNO:	**Omar Niamut**
Handed in:	**August 14, 2016**

eit Digital

TNO

UNIVERSITY OF TWENTE.

Acknowledgements

I would like to express my great appreciation to Hans Stokking for his enthusiasm and guidance during the process of my thesis at TNO and to Jan van Erp for his supervision and valuable feedback that helped me getting the most out of my thesis. I would also like to thank Omar Niamut for his very constructive feedback throughout the thesis, and Dirk Heylen for his support not only regarding the thesis but also regarding the coordination of university processes. Furthermore, I would like to thank the TNO Media-Lab personnel and especially Arjen Veenhuizen, Simon Gunkel and Peter Hoefsloot for their invaluable assistance. Additionally, I would like to thank the UTwente Designlab-Team and the helpful and open employees at TNO – Anna van Buerenplein, who made such an extensive user-test participation possible. Finally I would like to thank my friends and family who supported me during my entire master studies.

"So the key issue for developing satisfying virtual environments is measuring the disappearance of mediation, a level of experience where the VR system and the physical environment disappear from the user's phenomenal awareness."

(Riva et al. [Riv99])

Abstract

Video delivery over an IT-network is often challenging because bandwidth limitations are bound to affect the quality of experience. In order to reduce bandwidth demand of a video while affecting the QoE as little as possible, parameters like delay, resolution and frame rate need to be optimised with care. While these bandwidth-negotiations have been researched extensively for standalone displays, virtual reality headsets with their special optical properties have barely been assessed in this regard. In this thesis, a video pipeline was created in hardware and software to assess the impact of bandwidth-negotiations on the quality of experience for video viewed in a virtual reality headset. I describe a series of user studies that assess the impact of different video resolutions and video delay on quality of experience and social interaction conducted through virtual reality headsets. The findings described herein are expected to support video distribution of mixed- and virtual reality applications over IT-networks.

Keywords virtual reality, mixed reality, 360°-video, video quality, video resolution, video delay, bandwidth, user testing, presence, transportation, immersion, collaboration, tele-communication, social interaction, network

Contents

Glossary

API – Application Programming Interface
A set of tools and aids for the development of software applications.

AR – Augmented Reality
Technology that "augments" (live-)video by adding visual content. The added content often appears in form of 3D-models and seems to be integrated into the presented image.

CRT – Cathode Ray Tube
Common monitor design for older generations of displays like (e.g. TVs or computer monitors) based on electric illumination of phosphor molecules.

DPI – Dots per Inch
Pixel count of a digital image or a display, based on the amount of pixels per inch and focussed on pixel-density. This is in contrast to "resolution", where only the total pixel count is measured.

FoV – Field of View
The size of a visual field as perceived by a viewer or presented by a medium, measured in angle. The FoV for visual perception is dependent on the optical system (e.g. human eyes). The FoV of display technology is dependent on the viewing distance and physical size of the presented image.

fps – Frames per Second
Frequency of presented images (frames) in a video stream. Measured in images presented per second.

HMD – Head Mounted Display
A device that provides visual information via a display that is strapped to the user's head.

IT – Information Technology
The utilisation of computer-systems to process data. This includes, data-storage, -analysis, -generation and -distribution.

MR – Mixed Reality
The combination of real and computer generated content. Ranges from the real world without additions over "Augmented Reality" to "Virtual Reality", consisting exclusively on computer generated content.

MUD – Multi User Dungeon
Virtual world (often computer game like) that enables users to meet and interact remotely through computer technology.

QoE – Quality of Experience
"[T]he degree of delight or annoyance of the user of an application or service. It results from the fulfillment of his or her expectations with respect to the utility and / or enjoyment of the application or service in the light of the user's personality and current state." [BBDM+13]

QoS – Quality of Service
Quality of technically measurable factors of a communication service from the viewpoint of a user. Amongst other things, often related to measurements of delay, packet-loss, jitter and bandwidth.

RoI – Region of Interest
A region that is of higher priority. In relation to visual media (e.g. video), the RoI describes a section of an image or video that is of higher interest or importance to the viewer or a technological application.

VE – Virtual Environment
An environment that is generated with computer technology and can be explored by a user with virtual reality-technology such as head mounted displays.

VoD – Video on Demand
Services that deliver video over a network on demand of a user. Popular examples are YouTube or Netflix.

VR – Virtual Reality
Technology and concepts, that present virtual environments and other computer generated content in an immersive manner.

1 Introduction

1.1 Motivation

Even before computer technology was developed far enough to facilitate Virtual Reality (VR) applications, philosophers pondered over the "ultimate medium", able to transport the mind of people to other spheres far away from their physical bodies [BL13]. In order to accomplish this in VR, the technology needs to generate a feeling of "transportation" and "non-mediation" [Riv99, LD97] which results in a high degree of "presence" [SVS05]. This could ultimately allow remote social interaction on a level that would render airlines obsolete. But the full coverage of sensory input and output that is needed for such an application still bares numerous problems to be solved. In pursuance of supporting the development of VR in this direction, the here described research is addressing the visual aspects of VR.

Today, computer technology has evolved to a point where Virtual Reality (VR) can be made available for a broad audience. Already acknowledged by the military [RPL+11, BM07] and now fostered by the gaming- and entertainment industry, a great deal of related hardware and software appeared on the market in the last years. This trend is being reflected by a vivid community and might indicate the beginning of a far reaching focus-shift from 2-dimensional displays to 3-dimensional/stereoscopic visualisations. This shift is supported by big market players such as HTC, Facebook and Steam and their interest in this technology is not surprising, considering the numerous application scenarios VR potentially allows.

On this background, VR appears to be the next technology platform for media consumption and remote (social-) interaction. Game-like Multi User Dungeons (MUDs) such as AltspaceVR[1] show great potential for social interaction already and the appearance of video based VR-services like NextVR[2] suggest that video-content will still play an

[1] `http://altvr.com/`, last accessed: August 3, 2016

[2] `http://www.nextvr.com/`, last accessed: August 3, 2016

important role in the future. This expectation is shared with researchers like van der Vorst et al. [vdVBvKB14] who are forecasting that down-streaming of video will have a great share of future bandwidth use. This share is also expected to increase during upcoming years.
This in turn means that bandwidth-limitations of IT-infrastructures will remain a challenge in the future. This is especially true for video-streaming in VR, since immersive 360°-videos are rich on data and therefore require a significant amount of bandwidth and a stable network. The research in this thesis aims at supporting the developments of video based tele-communication in VR. It therefore focusses on the visual aspects of spherical 360°-video and the effects of bandwidth-negotiation concepts on its Quality of Experience.

1.2 Research Goal

One of the main problems of video delivery over an IT-network is bandwidth. It limits the amount of data that can be sent – or "streamed" – over a certain period of time and can affect the Quality of Service (QoS) and Quality of Experience (QoE). Therefore, it has been the focus of extensive research [DM98, AAS+14, OSN+14] to find relations between QoE and bandwidth negotiations for providing better experiences with less data. But data-reduction processes that are created for 2-dimensional displays like TVs or PC-monitors are not necessarily applicable to novel VR-HMDs since they have different optical viewing conditions. On the same page, video applications for VR – like e.g. immersive 360°-video – have often higher bandwidth requirements than video for 2D-displays. Because of this, further research needs to be conducted to find novel negotiation tactics that ensure a good QoE for video in HMDs.

One solution for dealing with bandwidth limitations of video-streams is a switch in video-resolution. Matching the video-resolution to the available bandwidth helps avoiding stutter and delay. The process is called "Adaptive Streaming" and is standardised under the name "MPEG DASH"[3]. It is supported by numerous market players[4] and has furthermore been implemented by leading Video on Demand-services like Netflix or YouTube. This success generates the question, whether altering the video resolution is a

[3] **D**ynamic **A**daptive **S**treaming over **H**TTP
[4] I.e. Microsoft, Netflix, TNO, Samsung and other. `http://dashif.org/members/`, last accessed: August 3, 2016

feasible approach for VR-video as well. Therefore the overarching research question this study aims to answer is:

Which effects does a resolution-based bandwidth-saving approach have on a HMD-mediated video-application?

This generates a number of sub-questions:

1. *What are the effects of a decrease in video resolution on the QoE in a HMD?*
2. *Which display resolution and video resolution is sufficient for facilitating a face-to-face-communication and a collaboration task with a HMD?*
3. *What are the effects of a decrease in video resolution on social interaction that is mediated by a HMD?*
4. *How much video-delay is acceptable with a HMD for tele-communication?*

The core assessment of this study is therefore a QoE measurement for different video qualities in a HMD that was gained through user tests. For this purpose, the VR-video-pipeline as shown in Figure 1.1 was used as the baseline to create an experiment that allows to conduct these measurements. The pipeline was condensed into a controllable emulation, while preserving aspects that were identified as crucial for a pleasant VR-experience. Finally, a collaboration-based user study was conducted with the modified system to assess if a change in video-resolution has an effect on the QoE and social interaction.

1.3 Thesis Structure

The document consists of four main parts. Chapter 2 as the first part provides a broad background that illustrates recent challenges of VR. This includes challenges of creating presence through VR, challenges of HMD-development and challenges of creating and delivering immersive video for HMD-applications.
Based on this, Chapter 3 deals with the setup of necessary hardware and the execution of a user test study that aims at providing evidence for the assessment of the sub-questions. The prototypes are designed to emulate the pipeline of Figure 1.1 by breaking it down to a simpler, more controllable setup.
Thirdly, the experiments will be concluded by an analysis of the results in chapter 4.

Finally, chapter 5 accumulates the findings and states concrete recommendations to cope with (some of) the bandwidth problem of video for VR-purposes. Additionally, an outlook on possible future research is being presented that pursues the issues, discovered by this study.

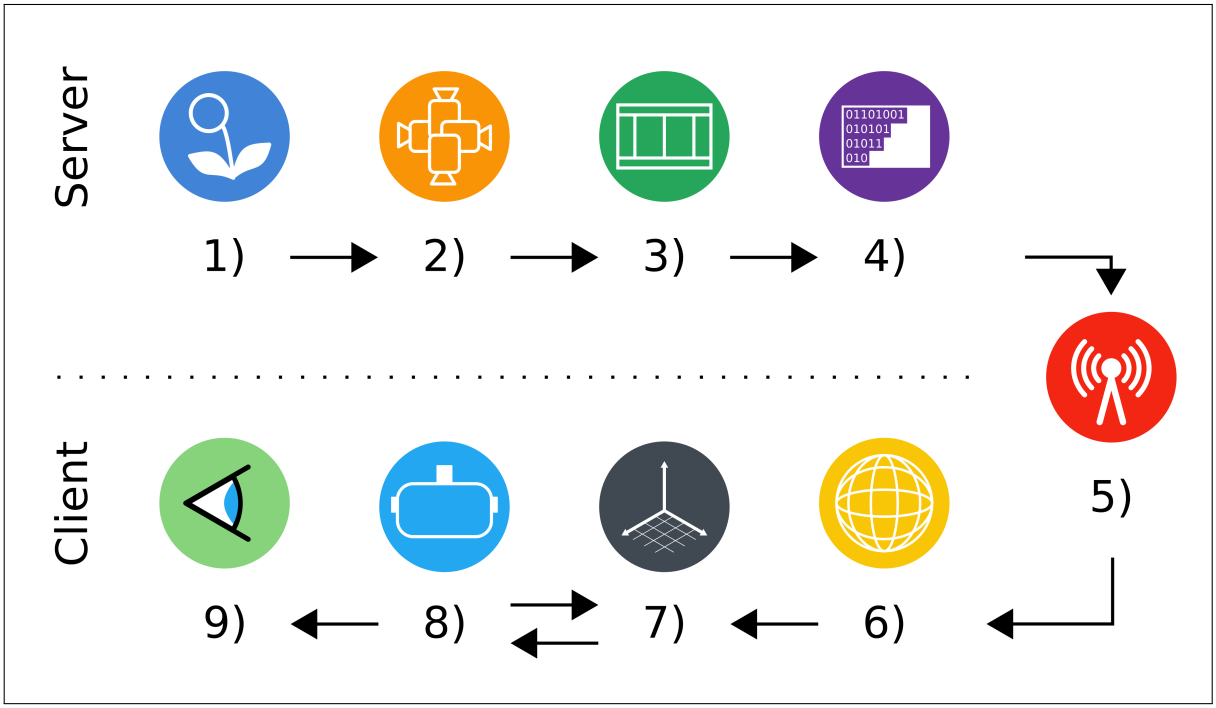

Figure 1.1: The here shown VR-videopipeline records a scene (1) with a multi-camera-system (2). The footage of all cameras is being stitched into one rectangular image (3). The image is then encoded (4) for delivery over a network (5). The image is received on the client side and warped around a geometry to generate a 360°-video (6) that can be viewed from the inside. Processing of the video (7) is done to allow for image distortion and to incorporate orientation-input from the VR-HMD. Ultimately the HMD displays the image (8), thus making it visible for the human visual system (9).

2 Background

Before the challenges of creating good VR-experiences are examined, it is needed to clarify the relation between Virtual Reality (VR) and Mixed Reality (MR). As illustrated in Figure 2.1, VR can be seen as the extreme end of the MR-spectrum. But the technology that is needed for VR can be used to augment or replace a real environment in varying degrees. Therefore, if this document is referring to VR or the related technology, these degrees are incorporated in its meaning. Especially since this thesis focusses on content that is not entirely generated by a computer (i.e. a video in contrast to a video-game) but is displayed with VR-technology, it would not be practical to make this distinction from this point on. Tackling the here described challenges however might affect technology and applications across the entire MR-spectrum.

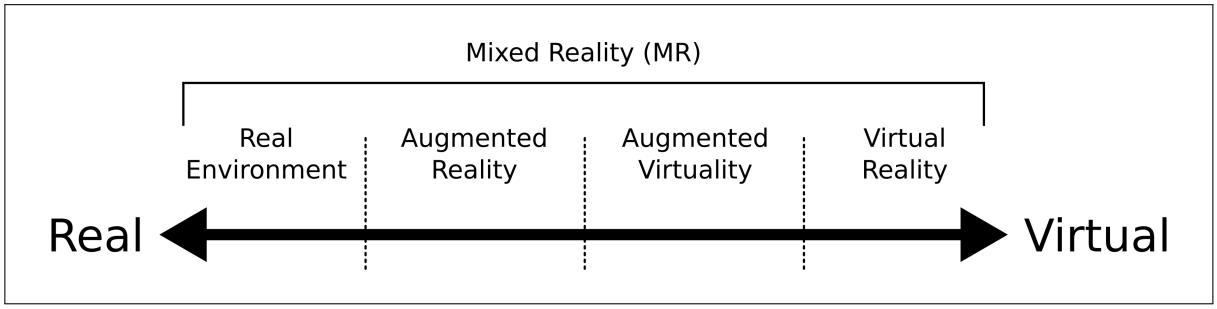

Figure 2.1: "Virtuality Continuum" as illustrated by Milgram et al. [MK94]

2.1 General Technology Benchmarks

Developers of software and hardware alike examined the human perception apparatus. Thus, several benchmarks were formulated that are to be reached to bring VR closer to the "ultimate display"-experience, Biocca et al. [BL13] are articulating.
Based on the human visual perception, a positional and oriental change of VR-content according to the user's (head-) movement should happen with the lowest possible latency.

According to Oculus' developers[5], this "motion-to-photon-latency" should range below 15 ms if not below 7 ms. Head movements can gain up to 60 deg/sec in velocity and 500 deg/sec^2 in peak acceleration. Since purely measurement-based changes are not fast enough to compensate for such values, prediction models and sensor fusion come into play.

For display technology, the pronounced targets are a 4Kx4K resolution (better: 8Kx8K) at a Field of View (FoV) of 90 deg vertical and horizontal.[6] But since a human's total FoV is rather around 190 deg horizontally [HR95, p. 32], HMDs with increased FoV that match human vision are likely to appear on the market at some point.In order to fill this higher FoV with visual content, more pixels are required if the same visual quality is desired to be achieved. This applies to the hardware as well as to the software/content.

For emulating a "real"-feel, display technology is facing another challenge called "judder"[7]. Judder is caused by two main components, namely "strobing" and "smearing" of the displayed image. Display-related strobing stands for "blinking" of an object that is moving over the screen. Since displays do not display content continuously, a moving object appears to be jumping step-wise in a certain direction instead of moving smoothly. Smearing appears when a display is not capable of switching its pixels fast enough. The result is a "ghost"-image of a moving object at its old position. A popular example is a moving football displayed on an LCD-screen, turning into a comet with a tail rather than remaining a round object. An "advantage" of smearing is its potential to reduce strobing. Both issues could be tackled by raising the refresh rate to 1000 Hz.[8] However, recent technology with refresh rates below 300 Hz is fairly distant from this goal.

[5] `https://developer.oculus.com/blog/the-latent-power-of-prediction/`, last accessed: June 9, 2016

[6] `http://blogs.valvesoftware.com/abrash/when-it-comes-to-resolution-its-all-relative/`, last accessed: June 9, 2016

[7] `http://blogs.valvesoftware.com/abrash/down-the-vr-rabbit-hole-fixing-judder/`, last accessed: July 20, 2016

[8] `http://blogs.valvesoftware.com/author/mabrash/`, last accessed: May 12, 2016

2.2 Hardware

2.2.1 VR-HMDs

The benchmarks described in the introduction of this chapter are far away from today's technology standards. Therefore, VR-HMDs will remain a great challenge for developers since integrating the necessary technology while keeping the weight down and comfort high is contradictory at this level.
The HMDs on the market are based on various technology setups but can mainly be separated into two sections, the all-in-one systems (usually powered by a mobile phone) and the computer-powered systems (as extensions of high powered, mostly immobile processing units). Regardless of which technology is used, the challenges mostly remain the same. However it is more likely to reach technology-benchmarks with a computer-powered HMD first due to the higher processing capability and lower mobility requirements. Integrating the same functionality in an all-in-one system with the same quality will then probably be more a matter of time than a matter of feasibility.
Momentarily, a significant difference of these two groups however is the bandwidth-question. PCs with a wired internet connection are more likely to have access to a stable internet connection and a high bandwidth rate than mobile phones in a mobile network. Delivering content to an all-in-one HMD will therefore be more challenging.

2.2.2 Displays

Display technology in general will add to the fall or success of VR-technology. Due to their beginnings in form of Cathode-Ray-Tubes (CRTs), novel display technologies still inherit certain behaviours when it comes to image generation. Scan lines and scan-out order might prevent a pixel from being displayed immediately after rendering, thus creating motion-to-photon-latency. One possible exploit called "racing-the-beam" might help but sparks several drawbacks as described by Michael Abrash[9]. A further, more general problem is that the high requirements for display hardware are at the moment only justified through the perspective of VR. Other applications will hardly benefit from exponential raise of refresh-rate and pixel density along other technology leaps. It thus hardly motivates

[9] `http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/`, last accessed: June 9, 2016

the industry to bridge these gaps any time soon if commercial interests are not aligned. Not complying to the stated criteria for HMDs might hinder the development towards a high-quality remote interaction in conclusion.

2.2.3 Tracking and Sensors

VR is heavily dependent on motion-tracking to present visuals that behave in relation to the user's body movements. With regards to HMDs, head tracking is applied to show the right camera angle and perspective. It is usually realised with a combination of "inside-out"-, "outside-in"- and inertia tracking. Here, sensors for magnetic field tracking and inertia tracking, embedded in the HMDs (inside-out) are complemented by optical tracking through cameras and other (outside-in).
Aside from trying to create natural viewing experiences, tracking can be used to reduce bandwidth of streamed 360°-videos. By analysing the user's head-orientation, the viewing angle can be determined and video that is "behind" the user can be neglected or played with lower resolution.
Similar potentially applies for eye-tracking which has potential to support bandwidth savings by facilitating e.g. fovea-based rendering or image blurring, based on Region of Interest (RoI). This way, content that falls in the user's peripheral view can be rendered with less quality than the visual centre, thus reducing bandwidth with only a low impact on QoE [CCL02].

2.3 Content Generation and Processing

Creating immersive material can be approached from two sides. The first is to model a 3D-world, similar to a computer game. The virtual view within the Virtual Environment (VE) can then be controlled by moving the tracked HMD. Creating VR-content that is based on video however is an entirely different challenge and requires special camera systems and content processing. The desired product of such a video-capture will at this point be called "spherical 360°-video" and is illustrated in Figure 2.2. The following subsections illustrate the capture and processing of such video-footage and points out deriving challenges.

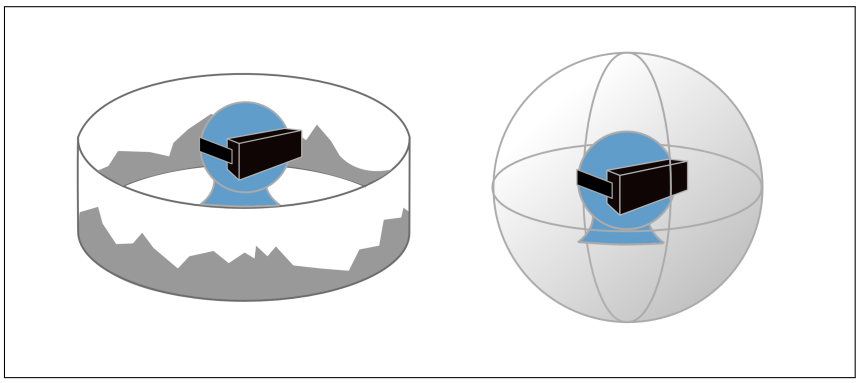

Figure 2.2: The term "360°-video" can be confusing. The left side illustrates a "panoramic" video that shows a 360°-view on the horizontal level. The right side illustrates a "spherical" 360°-video that encloses the viewer completely, thus showing 360° of video content in every plane.

2.3.1 Video Capture for VR

Image capture for VR-purposes requires novel camera-systems that can capture spherical 360°-content in high resolution, high framerate and as binocular footage (see section 2.4). Data compression and content delivery as well as storage devices and cameras struggle with the amount of data, such a video might accumulate. Additionally, challenges that derive from this kind of video are video-stitching (combining several camera images for a 360° view around the viewer) and consequently bandwidth management.
What is more, the whole way films are being told and produced needs to be reinvented. How is the image being divided? How is the video being shot without having a camera crew in the video or letting the viewer lose the focus of the story? This in combination with how the footage is being edited and post-produced asks for novel and comprehensive solutions and addressing them would exceed the focus of this thesis.

One of the main reasons why a spherical 360°-video adds up significantly more data than a video for rectangular screens is that the amount of pixels needed to enclose the viewer is higher than the amount needed for filling a regular display. Although current HMDs also only work with a planar rectangular screen, the additional video-data is needed to provide the option to look around in real-time and to compensate for random head-movements.
Figure 2.3 shows the schematic of a 6-sided 360°-camera. Here, 6 cameras capture images that are adjacent to each other. This way, a 360°-video-file would have approximately 6 times the size of a rectangular video file from one of the cameras. This is a very general concept and applies mostly to multi-camera-rigs and less to e.g. wide-angle-cameras that

capture a wider FoV than regular cameras. However, content processing workflows have evolved that try to decrease the file size for this kind of video. Some of these processes are explained in section 2.3.2.

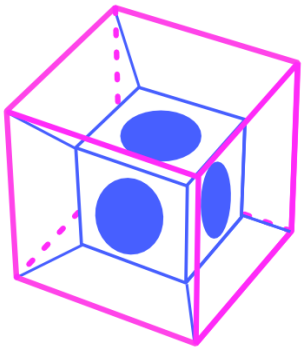

Figure 2.3: Schematic of a 360°-camera. The 6 cameras (blue circles) on each side of the cube capture a 2-dimensional and rectangular image (sides of the pink cube). These images can be "stitched" together to form a video that surrounds the viewer.

2.3.2 Content Processing

In general, video consists of flat and mostly rectangular images, that are shown in rapid succession. Small differences in a row of pictures let's the viewer perceive a motion or continuous change. This kind of data has been subject to decades of data-reduction and video-encoding efforts and is well researched. Standardisation committees like the "Moving Picture Experts Group" (MPEG)[10] dedicate their work to forming standards for handling video-data in a way that covers the most common video-applications. These standards however are also not entirely suitable for handling spherical 360°-videos. But with the gaining popularity of spherical 360°-videos, correlating standards can be expected in the future.

Video-footage for spherical 360°-videos is stored the same way as planar videos in form of a succession of rectangular images. 360°-video however needs to be wrapped around the viewer (Figure 2.2) before displaying. Creating a panoramic image does not generate that many problems since they can be created by connecting two ends of an image to form a cylinder. Enclosing the viewer in a full sphere of video however needs more steps

[10] `http://mpeg.chiariglione.org/`, last accessed July 19, 2016

to achieve and generates additional problems.
The wrapping creates overlap and geometric distortions in the video. This is similar to wrapping a rectangular piece of paper around a ball as illustrated in Figure 2.4. The equator of the ball might be wrapped smoothly but the poles have excess-material that needs to be wrinkled or cut in order to fit the surface of the ball. This excess needs to be taken care of to avoid visual artefacts (e.g. visible colour-errors at the poles).

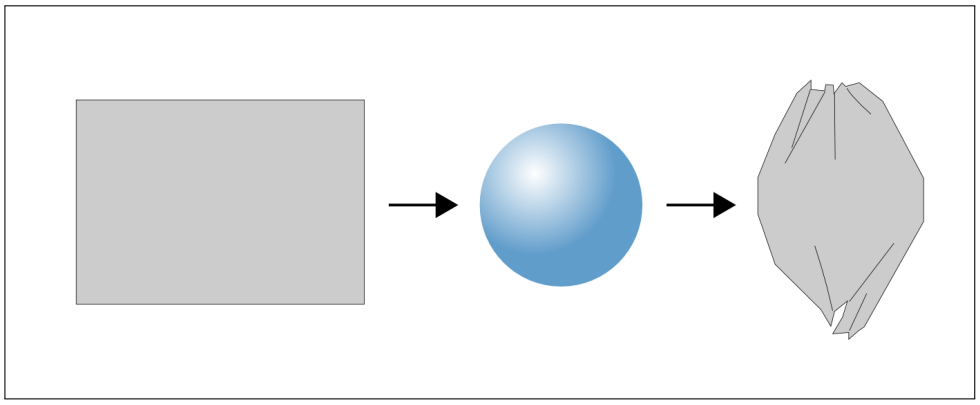

Figure 2.4: Wrapping a rectangular plane around a ball generates excess material near the poles.

In order to deal with the wrapping issues, several processes have been created. Facebook for instance promotes the "cube-mapping"[11] which takes the video footage and maps it on the inside of a 6-sided cube. This approach is beneficial because the rectangular video-footage can be cut in a way that pixel-loss is minimal, as it is easier to form a cube out of a rectangular plane than forming a sphere. The viewer can then watch the video from inside the cube and sees footage in every direction much like the sphere in Figure 2.2.
This approach has been further improved in terms of data-reduction by mapping the video on a four-sided pyramid. Further explanations regarding this would exceed the limits of this thesis and can be viewed on Facebook's official development-homepage[12].

[11] `https://code.facebook.com/posts/1638767863078802/under-the-hood-building-360-video/`, last accessed: July 19, 2016
[12] `https://code.facebook.com/posts/1126354007399553/next-generation-video-encoding-techniques-for-360-video-and-vr/`, last accessed: July 19, 2016

2.3.3 Content Delivery

Subsection 2.3.2 addressed the way, spherical 360°-videos can be processed to be viewed as such. Additional processes along these lines are concerned with providing video as fast as possible over an internet connection. Especially when viewed over the mobile network (e.g. with a smart-phone or a tablet), video delivery needs to be highly optimised to ensure a certain quality. What has been challenging with "normal" rectangular video already becomes even harder when dealing with spherical 360°-videos due to their higher data volume.
One process that could potentially handle this problem was already mentioned before: "adaptive streaming" [Sto11, ABD11, LMT12]. It facilitates a real-time switching of video-resolutions, depending on the quality of the internet connection. This way, the stream with the appropriate amount of data can be chosen to match the available bandwidth to foster a stutter-free video experience.
A further approach to flexible content delivery is called "tiled streaming" [VBNPS11, AMV11]. This concept acknowledges changing bandwidth requirements and the need for delivering ultra high resolution video to end devices over wired and mobile internet connection. This is done by dividing a video-stream into several tiles for further processing. Therefore, customised and scalable content can be provided to clients, based on the type of used replay-device and the requested RoI of a video. That way video-streams can be altered to suit a device's resolution and internet bandwidth or a user's preferred content within a video-stream (e.g. camera angle or head-orientation in a HMD). This avoids the transmission of unused data and frees bandwidth for the actually requested data.

The above described processes serve several different use cases but have a certain feature in common: Switching the amount of pixels being sent over a network which alters the data volume of the streamed video. In some cases these switches might take place unnoticed, if the receiving device has for instance a lower display-pixel count than the requested footage. Switching to a lower resolution which actually fits the pixel count of the device might have little to no effect on the visual QoE and can improve the viewing experience by lowering or avoiding stutter or long loading times.
The expansive research efforts to keep the bandwidth to a minimum by lowering the resolution of a (streamed) video [DM98, CCL02, AAS+14, OSN+14] suggests high potential for eventual VR-applications. A question which derives from this is, in which range such a degradation can take place before a viewer would notice a decrease in

QoE.

2.4 The Human Visual System and VR-HMDs

The human visual system is highly complex and relies on numerous factors. Each of these factors has to be taken into account, when designing hardware and software for VR applications. Neglecting these can lead to distress if not calibrated correctly [MWWR93]. The main factors are being described in the following.

2.4.1 3D-Vision / Spatial Perception

The spatial perception is based on the slightly different perspectives the two eye-balls are providing. When looking at an object, its size as well as its position can be determined by the small differences in perspective and their relation to the environment. This requires, that the object can be focussed on with both eyes at the same time, resulting in binocular fusion [BB85]. With the foreground-background relation, positional cues can also be obtained with monocular vision. The visual shift of an object against a background is known as "parallax" (Figure 2.5). In VR-applications, the difference between monocular and binocular vision can occur if content is only rendered for one eye or a camera image is duplicated and presented to both eyes simultaneously in the HMD. Increasing the distance between two camera-sensors that shoot binocular footage can lead to hyper-stereo vision, if the distance is bigger than the user's interocular distance. This increases the depth perception in greater distances.
In case a streamed video is being viewed, either two camera streams or a slightly altered version of one camera feed needs to be transmitted to provide binocular vision. Along with the bandwidth issues shown in section 2.3, this is yet another origin of eventually higher bandwidth requirements for VR-applications in contrast to regular video-transmission.

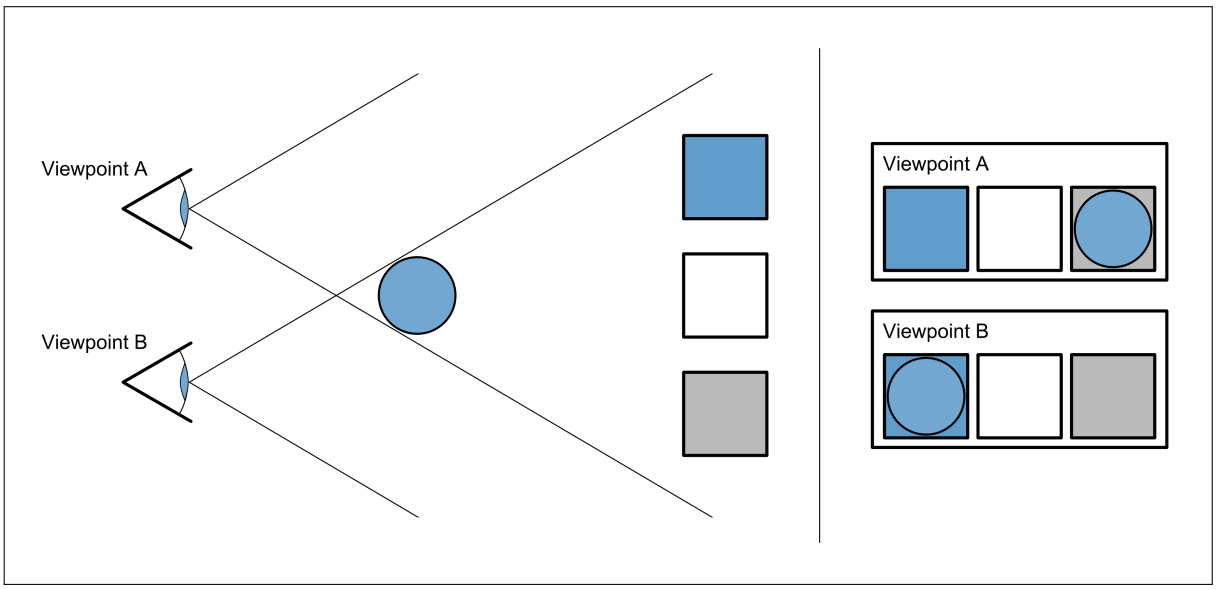

Figure 2.5: Visual shift of an object's (ball) position against the background (square tiles) after change of viewpoint (A→B). This shift is called parallax.

2.4.2 Orientation

Vision is a big part of the human orientation. In order to walk or keep balance, humans take visual cues from the environment amongst other sensorial input as reference [Lee78]. If prominent visual marks such as the horizon are not static or start to "swim" against the user's head movement, discomfort and nausea can develop [LSP83]. This can be the case for VR-applications if the motion-to-photon-latency is too high.

2.4.3 (Con-)Vergence and Accommodation

Vergence describes the movement of eyeballs' against each other, convergence describes the opposite. In order to look at a distinct object, the human eyes cross their lines of sight in the position of the object to achieve binocular fusion (Figure 2.6).
In case of most VR-HMDs, focussing eye-movement can only be emulated by providing a wider scene to look at. Thus, the eyes can wander over the display. However, the virtual cameras that are rendered by a game engine for instance do not orientate accordingly so that the actual perspective does not change when changing the focus without altering position and orientation. Same applies to a fixed binocular camera setup as it is often used for Augmented Reality (AR)-applications or robotic vision. Advanced HMD-concepts like the "Fove" implement eye-tracking to alter the displayed information

according to the user's focus point. This could become standard if the application scenarios turn out to increase the QoE and (eye-)tracking technology becomes cheaper and smaller.

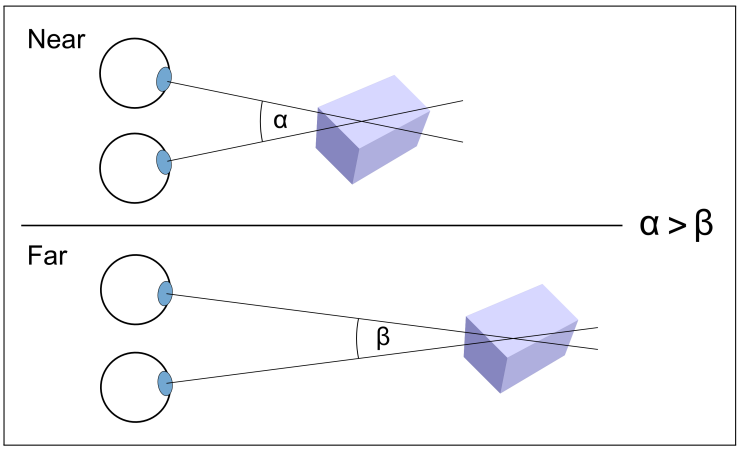

Figure 2.6: The angle between the two eyes' lines of sight differ when focussing on objects in different distances. The line of sight angle α for close distances is greater than angle β for distant objects.

The accommodation of the human eye describes the changes in lens form to focus on an object in a specific distance. The lens is being contracted or stretched by the ciliary muscle in order to change the focal distance to form a sharp image on the retina.
Since electronic displays are usually a two dimensional medium, real depth in an optical sense can hardly be visualised. The human accommodation reflex is therefore working against the architecture of modern HMDs since in this case, depth can only be illustrated through perspective and not through natural visual blur or focus that is caused by the viewer's eye. Also in this regard, the Fove HMD could be a step in the right direction since it allows artificial image blur, based on the eye tracking information. However, a HMD-display that is able to show real depth is yet to be released.

The fixed display-eye-distance in HMDs generally requires the wearer to use her eyes in a way that is not in accordance with human visual reflexes. These reflexes and relations as described by Finchman [Fin51] generate a strong correlation of vergence and accommodation. A constant violation of familiar viewing habits requires conscious efforts to work against these reflexes. This in turn has a high potential of causing eye-strain. In fact, manufacturers like Oculus actually recommend[13] a certain virtual distance between the user's eyes and objects in a VE to ease up these effects.

[13] `https://developer.oculus.com/documentation/intro-vr/latest/concepts/bp_intro/`, last accessed: July 20, 2016

2.4.4 Screen Door Effect

The "Screen Door Effect" describes the visibility of a display's pixel grid. If the spacing between the physical pixels is visible, the display can be perceived as a barrier between the viewer and the displayed content. This is a crucial aspect for VR-applications with regards to immersion and presence. What is more, this effect could cause a decrease in perceived content quality since the visual quality is limited by the pixel-count of the viewed display. What has not been a problem for television screens viewed from a distance can very well become a distracting factor in VR-hardware. To tackle this problem, displays with a high Dots per Inch (DPI)-count are required.

2.4.5 Lenses and Distortion

The compact design of most VR-HMDs provides only little distance between the user's eyes and the HMD's display(s). This requires an optical system to make the content comfortably visible. For this purpose, a variety of lenses is being used. Early developments in the developer versions from Oculus and other manufacturers used one bi-convex lens for each eye. The easy to produce lenses however cause chromatic aberrations – colour shifts – at the edge of the FoV. The lenses are also accountable for a significant "pincushion" distortion (see Figure 2.7). Apart from easing the visual focus for the beholder, the lenses serve another purpose: The FoV in the virtual world needs to approximately fit the human FoV for a high degree of immersion. Since displays in most goggles are too small to cover the entire FoV, magnification through optics is used. This step also increases the pincushion distortion. In order to rectify the content's perspective distortion, it needs to be distorted in the other direction. This is usually done by a barrel-distortion shader towards the end of the rendering process (Figure 2.7).

With the release of consumer HMDs from Oculus and HTC the lens design has changed from bi-convex to a hybrid, incorporating fresnel lens attributes. These lenses are similar to the ones used in lighthouses and have little concentric steps that focusses the light towards the centre. This rectifies the chromatic aberrations more on a physical level. They however spark another problem often referred to as "god-rays" or "flare", describing the appearance of a halo at the FoV's edges by light that is falsely redirected through the fresnel steps. The main gain by the new design is that content needs less colour-correction by shaders. The perspective issues however still remain.
Apart from improvements in lens design, chromatic aberrations are also tackled by a

colour correction in the distortion shader. The way the content is being warped for the lenses can be seen in Figure 2.8.

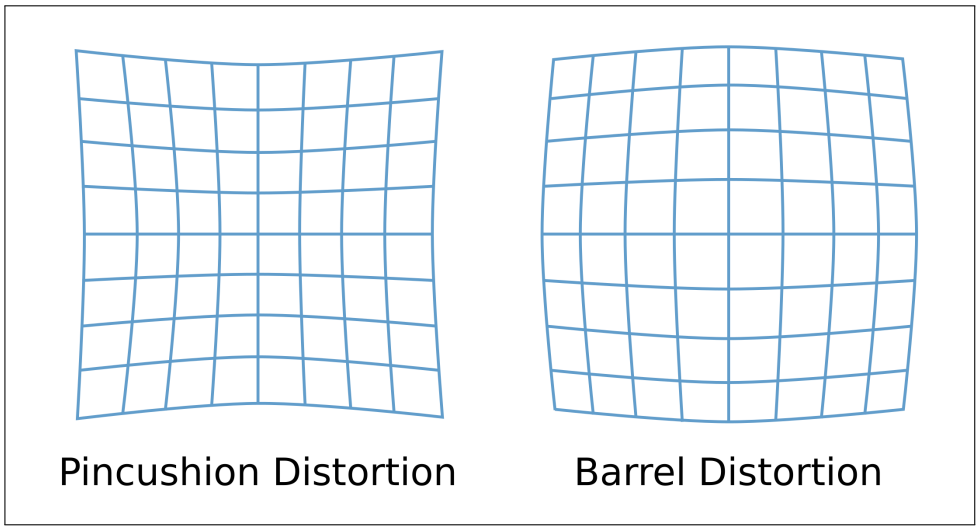

Figure 2.7: Types of distortion, appearing in VR hardware. The pincushion distortion (left) that is caused by the lenses and wide FoV-rendering is rectified by the barrel distortion (right).

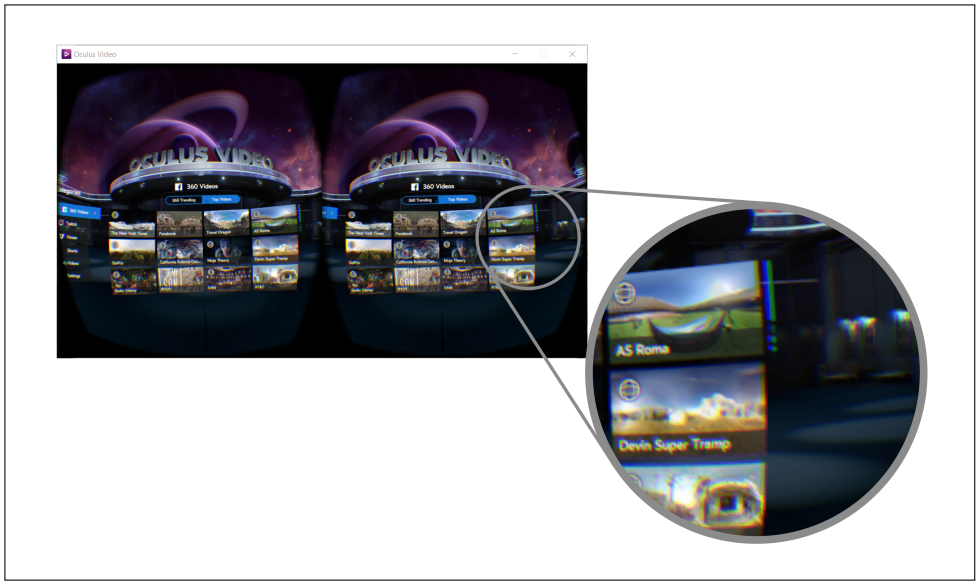

Figure 2.8: Screenshot of spectator display for the Oculus Video App. The black borders originate from warping the image corners towards the middle. Colour splits can be seen at light edges towards the borders of the image (see magnification). These are computer-generated to counter the lenses' chromatic aberration.

2.5 Design Challenges

The last section shed a light on recent challenges in VR from a technology perspective. These are mainly aligned with the needs of the human body to make synthetic content to be accepted as plausible reality. This view is generating measurable benchmarks for technology but highly disregarding the psychological and social aspects of human beings. To cover these areas, designers are required to develop content that is as acceptable as the technology it is displayed with. In the following, benchmarks for design, interaction and usability are explored.

2.5.1 Comfort

In order to be able to use the system for longer, the comfort is a crucial factor. Heavy gear, constricting wires or low robustness of the system can make it unusable. Parallel to the immediate level of comfort, longer sessions should also be possible without experiencing a decrease in comfort. Restriction of movement or natural behaviour is therefore to be avoided. The challenges that derive from the need for comfort are referring back to the technology. Making ever so smaller sensors, lighter materials and conveniently wearable hardware/head-gear is therefore crucial.

2.5.2 Safety

Safety is often opposing usability. Since VR-technology fully occupies the human perception, it is crucial to cover for risks that are originating from the user's physical environment. What is more, content should not trigger behaviour that puts the user in danger such as sudden and wide movements or extreme exercise.
Apart from the physical threat, psychological safety should be considered. How VR is affecting our psych is not very well researched. Nevertheless, the existence of such effects are already acknowledged in the established fields, i.e. psychology and therapy [SVS05, Hof04, LG13]. In this regard the challenge is less to develop *for*, but to investigate more *about* VR and its effect on humans.

2.5.3 Presence

In the fields of Human-Computer-Interaction researchers and psychology-/behavioural scientists alike, presence describes the feeling of "being there" where "there" is not equivalent to the position of one's own body but the place the VR-content suggests. This dislocation of mind in regards to the body might even be the "holy grail" of VR. Trying to convince the user that what he or she is being presented with is as real as what is perceived in their everyday life is maybe the hardest challenge of the field.

Sanchez-Vives et al. [SVS05] state that "Of particular importance is the degree to which simulated sensory data matches proprioception — for example, as the participant's head turns, how fast and how accurately does the system portray the relevant visual and auditory effects." Furthermore, they found that seeing and perceiving one's own body in VR is of equal importance. The perception needs to match the experience in real life, meaning that the body is viewed from a familiar perspective and also behaves like it. This factor is resonated in the research community and communicated by several researchers such as Riva et al. [RM00]: "[...] a VE, particularly when it is used for real world applications, is effective when 'the user is able to navigate, select, pick, move and manipulate an object much more naturally'[...]." Another described factor, closely connected to the bodily experience is the range and degree of body mobilisation within the VE. Being able to walk around and explore the VE should be an integral part of VR content since it appears to foster the feeling of presence [SVS05]. Riva et al. [Riv99] provide further distinctions of presence, that might serve as potential checkpoints for VE designers:

- social richness
- realism
- immersion
- transportation
- social actor within medium
- medium as social actor

It is safe to say that a complete VR-experience closely correlates with a high degree of presence. This is especially challenging, since reaching a certain degree of presence includes the use of numerous technologies that have to correlate as precise as a clockwork.

2.5.4 Degree of Realism

Since the first graphical interface for computers was released, developers strive for ever more realistic renderings. In times of today's potent graphic processors, content designers also explore more artistic approaches. These however are no longer inspired by the lack, but the existence of rendering power. The question that derives from these technological gains is which degree of realism is needed to provide an immersive VR-experience?
Sanchez et al. [SVS05] argue, that a high degree of realism is not equivalent to a high degree of presence: "Surprisingly, the evidence so far does not support the contention that visual realism is an important contributory factor to presence, and only one study, which used a driving simulator, has shown this to be the case[...]."
Designers are limiting the vast potential of VR-technology if the overarching goal is a sophisticated copy of our everyday life. Reality defying environments can still be highly immersive, if the user is well integrated. Thus, the important question is how VR can be exploited in a way that newly gained possibilities are incorporated without deleting the human factor out of the equation. In what way this applies to video-based VR however is yet to be found out.

2.5.5 From Space to Place

Harrison et al. [HD96] make a clear distinction between places and spaces: "Space is the opportunity; place is the understood reality." This concise definition constitutes a major philosophy VR-designers are invited to incorporate in their work. It is crucial, that "Virtual Reality" is not thought of as a space where people can go, enabled through sophisticated technology. It is rather important, that VR is understood as a medium to create places with a meaning. The moment VR is developed based on that goal, immersion will be enhanced significantly. Putting on some VR-goggles should not be a part of the experience. Diving in and visiting a place with a meaning is the experience sought after. HMDs and supporting technologies are merely a tool to facilitate the visit. Spherical 360°-video has the potential to generate these places. A 360°-camera in the ranks of a football stadium for instance is such case. By putting the viewer in a certain place and surrounding him with meaningful video, VR can become more than an entertainment device. This concept is also imaginable for meetings or family gatherings that connects people around the globe.

2.5.6 Communication and Social Paradigms

Communication and social behaviour are bound to affect the quality of a VR-experience. Humans are social beings and follow a vast set of rules and incorporated behaviours that only become conscious if violated. Therefore, VR-content needs to acknowledge these rules. Social behaviour is an extensive field on its own and can come in manifold shapes such as concrete measurements like direction and duration of gaze, reaction time and many other. Social avatars or even other human beings need to (re-)act "naturally" to let the user experience immersion. If communication and/or collaboration is limited or fails, the VR-experience will be less immersive.
How fine these aspects can be is demonstratively shown by Vinayagamoorthy et al. [VGSS04] and van Eijk et al. [vEKDI10]. These fine lines, marking the difference between a deep dive into the "uncanny valley" [MMK12] and a successful VE still provide room for improvement. We are therefore heavily dependent on further knowledge gain in order to get these things right. Especially video based tele-communication via VR-hardware bears significant challenges in this regard. One example of this is the display of facial expressions and eye-gaze despite wearing HMDs.

2.5.7 Collaboration

Technology developments are often inspired by the desire to reach other people and to facilitate communication. Different technologies provide different levels of communication. A telephone transports voice and other sounds but lacks display of gestures and facial expressions. 2D video-conferencing adds an image to the auditory information but still presents a clear border between the communicating parties. Kuster et al. [KRZ+12] summarize the shortcomings of 2D-teleconferencing as follows:

- Restrained to sitting behind a desk
- No gesture support
- No eye-contact
- Only upper body capture
- No depth information

The wish to "...develop a communication system that seamlessly integrates the remote person in the environment of the other participants..." in order to create "...a fully immersive 3D telepresence experience..." [KRZ+12] creates another great potential and

challenge for VR. How can we facilitate meaningful interaction and collaboration over great distances through VR? And can spherical 360°-video offer a solution?
If broken down to a basic understanding, collaboration can be seen as an exchange of information and actions. The struggles to facilitate this in a holistic way through technology are not novel and certainly not overcome yet. Why is it so difficult and what has to be done to come closer to a tele-collaboration that feels natural and eliminates the distance between participants?

In order to approach this problem, it is crucial to understand which information is needed and which information can be reduced or even disregarded. As stated in the definition of usability, this is correlating with the context of use. A short phone call might clarify an issue. An email provides sufficient information to keep on working. But it is not fully explored yet, what possibilities and boundaries VR inherits. We are yet to find out, if VR is a base for a whole new way of communication or just the next step to a more holistic communication pipeline.
In any case, more research is needed to shed light on ways to utilise VR for human collaboration. Novel concepts are required, that enable users to create a collaboration with outcomes, comparable to a face-to-face meeting. Probably the most challenging part is to create VR-tools, that cover a variety of (collaboration-)tasks to provide manifold collaboration scenarios through a single system.

2.6 Summary

This chapter illustrated different factors that have to be taken into account when creating VR-hardware and -software. They show that both need to be calibrated very carefully and are affected by each other in numerous ways. The following aspects have been recognised as the most important and had great influence on the experiment design and the related hardware development which are described in the next chapter:

- **Wired HMD** – A wired HMD is likely a more potent platform that allows easier prototyping than a mobile-system.
- **Tracking** – The tracking needs to be very precise and fast, to keep the motion-to-photon latency low and viewing comfort high.
- **FoV** – Humans have a wide horizontal FoV. It is therefore crucial to provide content that makes best possible use of the HMD's FoV.

- **Displays** – High display resolution, low persistence and a high DPI-count are desirable to counter the screen-door-effect, judder and to provide a natural "viewing" experience.
- **Content generation** – In order to avoid wrapping-issues and other side-effects of spherical 360°-video, the content for the experiment should be created in another way if possible. Furthermore, used video needs to be prepared for being viewed in a HMD. This is especially important with regards to distortion.
- **Content** – The viewed content should match human viewing habits and take the impairment of HMDs into account. Therefore an experiment should be designed in a way that keeps eye-strain minimal.
- **Comfort** – the used hardware needs to be comfortable in terms of weight as well as restriction through wires and other.
- **Safety** – The experiment needs to keep the participants safe. Therefore, the content as well as the user-task must not foster dangerous elements or behaviour.

3 Testing the Impact of Video Parameters on QoE in VR

3.1 Hypotheses

Before the experiment-hardware was developed, assumptions about acceptable ranges were made regarding video-resolution and video-delay in relation to QoE. These were based on literature and recommendations from VR-developers. Figure 3.1 illustrates these assumptions.

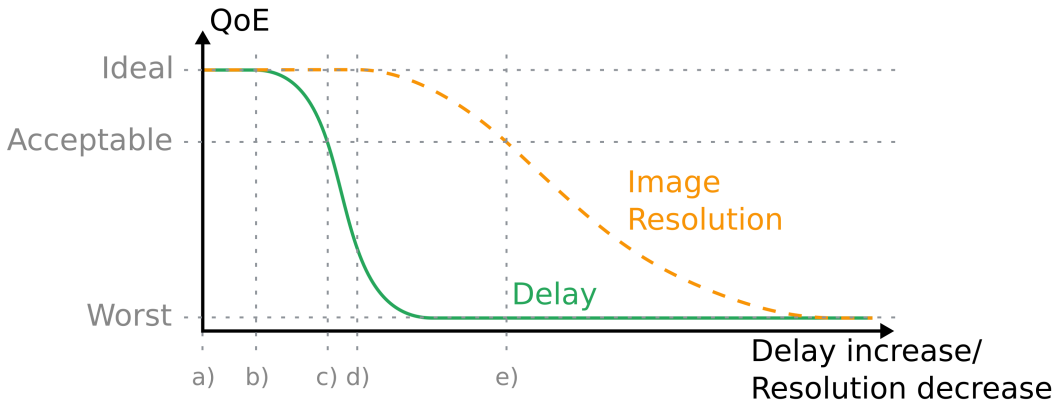

Figure 3.1: Relations of QoE related to image-resolution and video-delay: a) value for ultimate QoE, b) threshold for ideal video-latency, c) threshold for latency-acceptability, d) threshold for ideal video-resolution, e) threshold for resolution-acceptability

Figure 3.1 shows that a decrease in image-resolution was expected to have a wider acceptability-interval (a-e) than an increase in delay (a-c). It is worth mentioning in this regard that the acceptability threshold might alter, based on the application scenario. Since video-resolution could theoretically increase infinitely to improve whereas latency can only improve by shrinking towards zero, it can be argued that video quality

also has a bigger ideal-interval (a-d) than latency (a-b). Therefore, d) would mark a threshold above which an increase in video-resolution would bear no further advantage for the QoE. Based on the findings shown in Figure 3.1, Hypothesis 1 and 2 were formed:

> Hypothesis 1: *Altering the video-resolution in favour of bandwidth savings will lower the QoE after a certain threshold.*

> Hypothesis 2: *Altering the video-resolution in favour of bandwidth savings will worsen social interaction after a certain threshold.*

A suspected threshold where this would apply could however not be stated before development and was therefore not included in the hypothesis. It was furthermore assumed, that video-delay would affect the QoE and social interaction. Based on recommendations of leading HMD-manufacturers regarding motion-to-photon latency, it was suspected that video delay would have a greater effect on a HMD-mediated application than on video-tele-communication with a regular monitor or TV.

> Hypothesis 3: *Video-delay in ranges that are acceptable for video-based tele-communication (*$100 - 600ms$*)*[14] *will already affect the QoE and social interaction negatively in a HMD-application.*

The last hypothesis however was made on the base line, that the motion-to-photon-latency is closely linked to video-delay since it was planned to strap the cameras to the HMD for the experiment. The tolerance for video delay might not necessarily be that small if the footage was captured with a stationary 360°-camera, thus allowing direct orientation change of the HMD in a spherical 360°-video, independent of the video-delay itself. In that case it might even be possible, that delays of $100 - 600ms$ become applicable again.

3.2 Design Goal

In order to design experiments where the effects of alterations in video-parameters on QoE and social interaction can be examined, a system was designed where a camera sends live-video to a HMD. By this, the video pipeline as described in Figure 3.2 was condensed

[14] This range was stated by [GJK+14] as acceptable end-to-end delay for monitor-based video-tele-communication.

to a local, controllable setup that allowed for user experiments. The simplified schematics are shown in Figure 3.3. The following sections will illustrate the development of the video-pipeline in hardware and software. Afterwards, experiments will be described that were designed to examine the parameters' impact with participants.

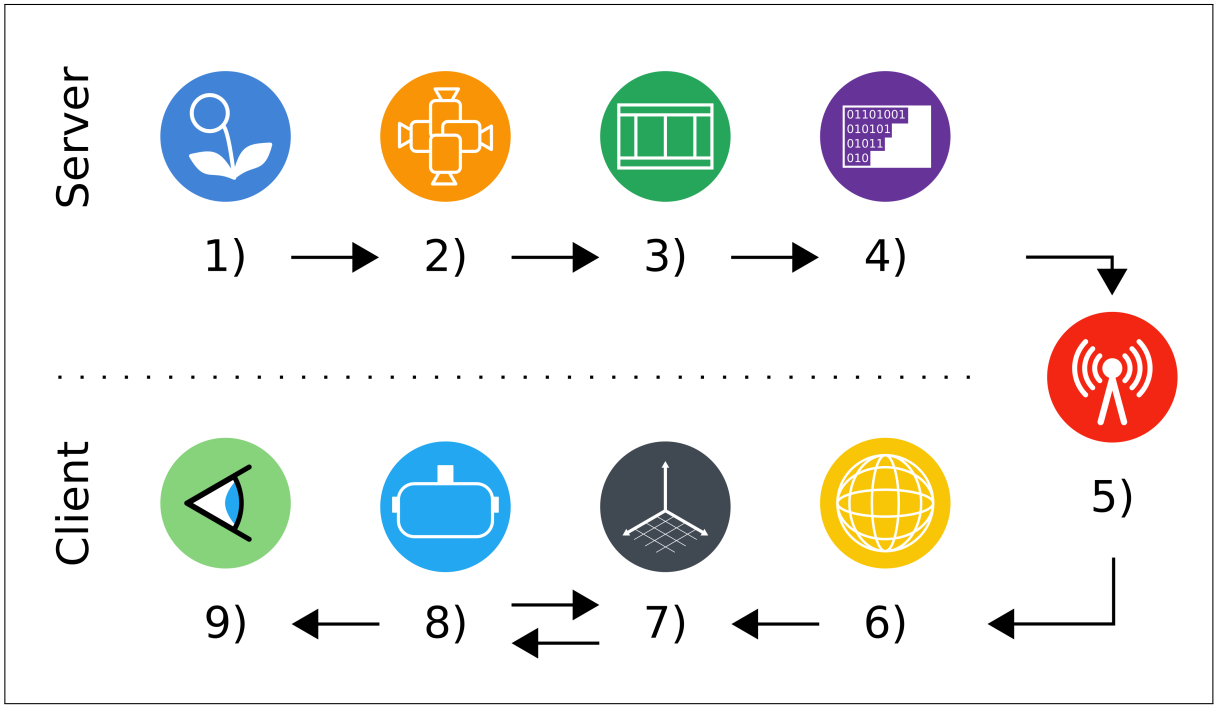

Figure 3.2: Video pipeline for spherical 360°-video sent over a data-connection as shown in Figure 1.1, Chapter 1.

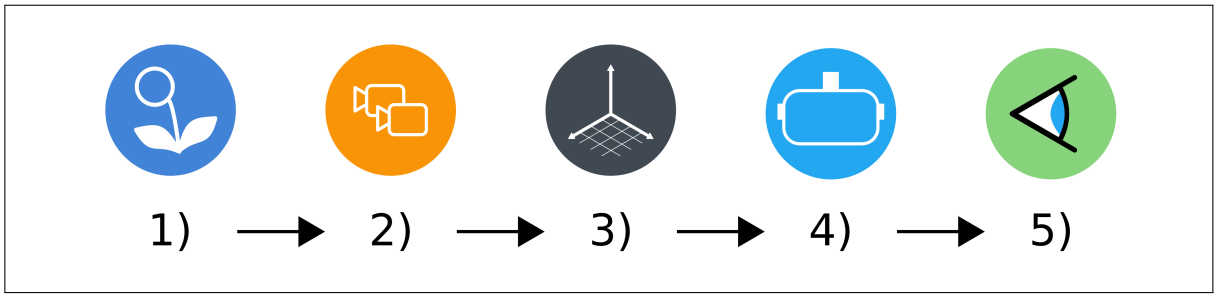

Figure 3.3: The condensed experiment-video-pipeline: A scene (1) is captured by Cameras (2). The video is being altered and processed for the HMD (3). (This time, the orientation-feedback from the HMD to the processing unit is not given.) Phase (3) is also the point where the video is being altered manually in terms of resolution and delay for the experiments. It is then displayed in a HMD (4) and presented to the human visual system (5).

3.3 Technology Setup

Three prototypes have been developed to find a suitable camera-HMD setup and each prototype was created with the factors described in Chapter 2 in mind. For simpler prototyping, wired PC-based systems were chosen as HMDs. The different hardware setups are illustrated in Figure 3.4 and their specifications are listed in Table 3.1. The first prototype was described extensively by Steptoe[15] and Pankratz et al. [PK15] and provided as a base for version two and three. The main focus regarding the software development was to provide controlled alteration of video quality and latency in the video pipeline. The latency-feature however was disregarded in later stages, based on findings elaborated in subsection 3.4.1
In each setup, cameras were strapped to HMDs in order to align the video-feed with the user's head-movement. This would make the processes needed for creating spherical 360°-video unnecessary. Additionally, since the cameras were directly attached, delay between head movement and physical camera movement was considered non-existent. By doing this, the tracking issue could be disregarded and latency-issues were only dependent on the delay between image-capture and image-display.

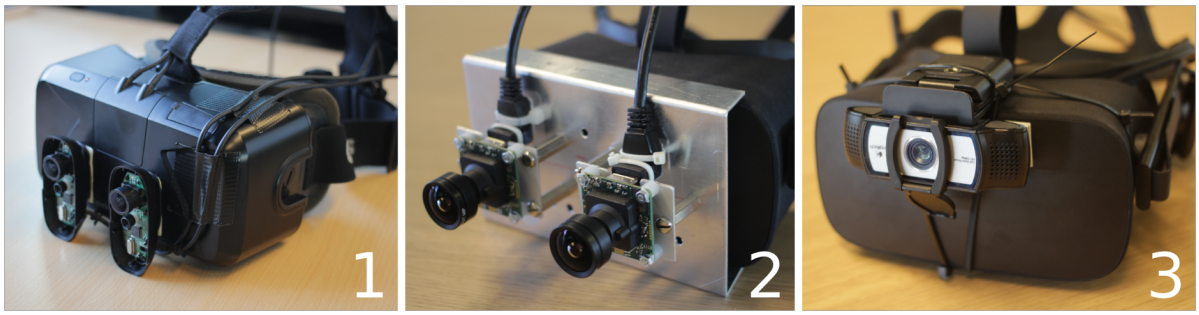

Figure 3.4: Prototype 1-3 (left to right). Setup 1 was used for the preliminary tests in subsection 3.4.1 and 3.4.2. Setup 3 was used for the main experiment, described in subsection 3.5.

The cameras were chosen based on their resolution. Ideally, the capture devices would have the exact resolution of the HMD's display resolution for one eye. This would help avoid sending video-data that eventually cannot be displayed. Furthermore, this potentially helps keeping the latency down and the frame rate up by avoiding unnecessary bandwidth-use. A comparison of the different resolutions is illustrated in Figure 3.5. Additionally, the cameras' FoVs were compared to the HMDs' FoVs. A close match of both would reduce perspective mismatches as illustrated in Figure 3.6.

[15] `http://willsteptoe.com/post/66968953089/ar-rift-part-1`, last accessed: July 7, 2016

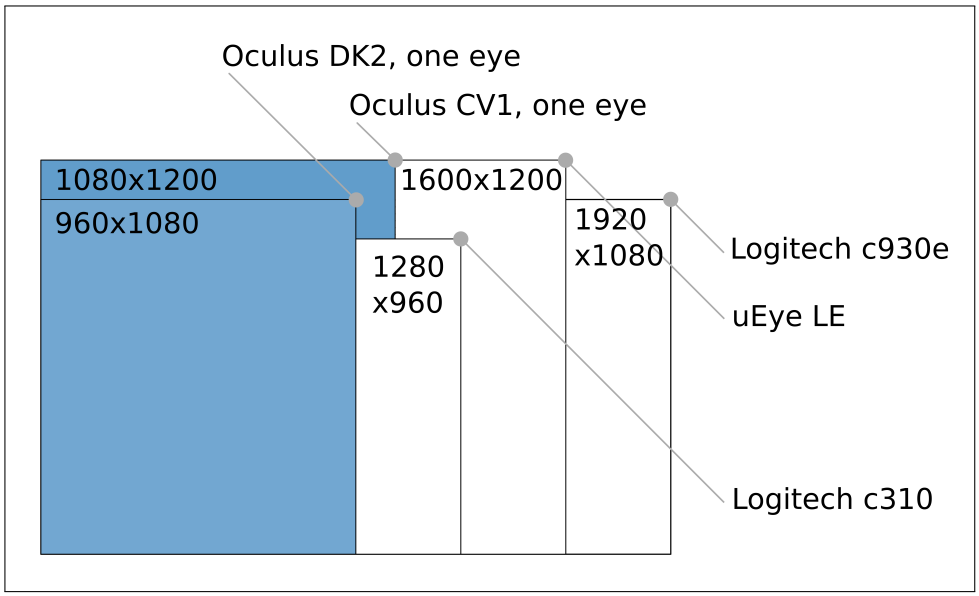

Figure 3.5: Camera resolution (white boxes) compared to HMD display resolutions (blue).

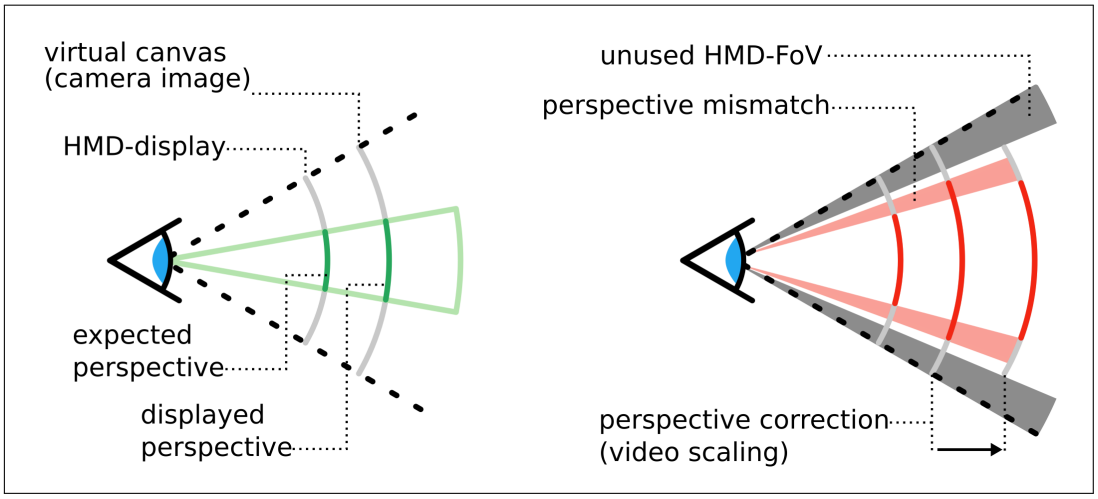

Figure 3.6: Perfect match of camera-FoV and HMD-FoV (left) against mismatch with a too small camera-FoV (right). The mismatch results in a mismatch of expected and displayed perspective. The case illustrated here causes objects to appear too big and requires a down-scale of video size which results in visible black borders. In contrast, a loss of video-pixels would occur if the camera-FoV was too big and objects would appear too small. The required up-scaling would make the video bigger than the display-area.

The lenses of the Prototype 1-cameras were equipped with wide-angle lenses to improve the FoV-match. After the HMD was switched in favour of a better display and optical properties, different cameras with higher resolution were chosen to adapt to the change. The setup is described as Prototype 2. Its cameras fit most of the above mentioned requirements and were highly customisable. Frame rates of 60 Frames per Second (fps) or higher as stated by the manufacturer however could not be achieved despite matching hardware configurations. A maximum frame rate of about 15 fps motivated the choice to switch to a consumer webcam, capable of 30 fps for the main experiment. This combination was labelled as Prototype 3.
The major drawbacks of switching the cameras were loss of binocular vision and a smaller FoV, resulting in unused display-pixels since the video had to be down-scaled for perspective correction (see Figure 3.6). A quick comparison however showed that the higher frame rate was needed in order to facilitate a certain experiment duration without generating significant eye-strain or worse repercussions. Since only one camera was used, the video image had to be copied for the second eye. Therefore, the total FoV was not only limited by the camera's optical parameters but also by the fact that both eyes would share the exact same FoV. While the monocular vision limited the capabilities for hand-eye-coordination and spatial navigation, it appeared to be still enough to provide general orientation.
An additional benefit besides higher frame rates however was the weight-loss. The aluminium-plate and the the longer eye-camera-distance of Prototype 2 made the setup appear quite heavy and therefore uncomfortable to wear.

3.3.1 Video Pipelines

Creating a video-pipeline that would connect the camera(s) with the HMDs sparked several challenges. Firstly, a way of sending an image to the HMD had to be created. Secondly, the video had to be altered in real-time after capture so the test-parameters could be adjusted. Lastly, displaying the video correctly for the optical conditions in the HMD required further image-alteration (i.e. distortion and positioning).

A first approach with the game engine "Unity" provided for quick results for the DK2. Here, two canvas-elements were rendered with a "webcam-texture" to show the cameras' video feed. The two canvases were positioned in front of the virtual camera, representing the HMD's view in the VE. Unfortunately the HMD-tracking could not be deactivated

which led to an inconsistent positioning of the canvas-elements in the virtual space. The result was a jumping image whenever the user would move his head. Furthermore, altering video parameters was not a default feature in Unity.
Since the DK2-HMD could be handled as a second display in a Linux distribution (Xubuntu), the GStreamer libraries could be used. These libraries offer numerous functionalities to play, record, send and alter video in a precise manner. The GStreamer-command-lines shown in appendix C generate windows that feature side-by-side-videos of the two webcams. These windows could be dragged into the DK2 with the mouse and switched to fullscreen. The combination of DK2, Linux and GStreamer allowed for very fast alteration and optimisation.

Switching to the CV1-HMD brought a better display but also a more enclosed software interface. Therefore, the prior developed video-pipelines were not applicable anymore and the "A-Frame"-framework[16] was used instead. A-Frame enables web-developers to create 3D-content in a browser with HTML-code in the same way a 2D-webpage would be created. It also provides a simple way to send content to the HMD and to deactivate the motion tracking. In combination with the "GetUserMedia"-Application Programming Interface (API)[17] video could be presented in the CV1 again. The API also allowed switching of video quality by requesting different resolutions from the camera. Apart from the different interface, the approach is similar to the Unity-pipeline with its virtual canvas-elements. Since Prototype 2 and 3 were both based on the same HMD, the video pipeline could remain the same after changing the cameras.

In order to measure the video-latency from camera to HMD, a visual approach was used which could be used with every device. Firstly a camera was placed behind the HMD so that the optics could be recorded. The camera was able to record 240 fps and had a wide FoV so that it could record the area around the HMD as well. The setup was switched on to show the video-feed in the HMD. When recording with the high-fps camera, a light-source was switched on and off in the view of all cameras. The 240-fps recording was then used to count the frames needed from switching on the light-source until appearing on the HMD-display. With 240 fps, each frame accounts for 1/240 or $\approx 4.2ms$. The maximum base-latency for each setup can be viewed with the other prototype-features in Table 3.1.

[16] `https://aframe.io/`, last accessed: July 16, 2016
[17] `https://developer.mozilla.org/en-US/docs/Web/API/MediaDevices/getUserMedia`, last accessed: July 16, 2016

Table 3.1 shows that Prototype 3 has the best suitable combination of settings of the three. It has the lowest weight and is able to record and display ≤ 30 fps. The resolution of the display is a significant improvement against the DK2-display and visual assessments have shown that the content resolution would be sufficient for the planned experiment, thus providing enough headroom for degradation to facilitate measurements. Additionally, the measured delay is the lowest of all the prototypes, even with the high resolution setting.

Table 3.1: Prototype specifications.

	Prototype 1	**Prototype 2**	**Prototype 3**
HMD	Oculus DK2	Oculus CV1	Oculus CV1
Camera(s)	2x Logitech c310	2x IDS uEye LE	1x Logitech c930e
Camera Resolution	1280x960	1600x1200	1920x1080
HMD Resolution (Display, 1 eye)	960x1080	1080x1200	1080x1200
Camera FoV (Horizontal after HMD-mounted, 1 cam)	$\approx 90°$	$\approx 100°$	$\approx 90°$
Horizontal HMD FoV (1 eye)	$\approx 94°$	$\approx 94°$	$\approx 94°$
Camera fps	$\leq$ 30fps	$\leq$ 110fps	$\leq$ 30fps
Video Pipeline fps	$\leq$ 30fps	$\leq$ 15fps	$\leq$ 30fps
Software Interface (Video Pipeline)	GStreamer	A-Frame	A-Frame
Video Latency (in milliseconds)	230	167	160
Video Pipeline	GStreamer	A-Frame	A-Frame
Weight	Middle	Highest	Lowest

3.4 Preliminary User Tests

In order to assess what can be considered an extreme and/or bearable decrease in quality of video parameters with regards to VR-video, preliminary tests were conducted. Two of these tests were especially insightful with regards to visual quality and latency. Each test was conducted with Prototype 1 without correcting barrel distortion and only for the purpose of general assessment.

3.4.1 Preliminary Test 1: Maze Drawing with Latency

The maze drawing test was a seated scenario where participants were asked to solve a very simple maze with a pen. The red marker had to be placed at the entrance and the maze was to be "walked through" with the marker in one stroke. Time was being measured from beginning to end of the drawing and the participant was asked to solve the maze as fast as possible while staying in the boundaries. This process was repeated four times. The first try was done without any impairment (no HMD) and used as a benchmark for the following three attempts. The second attempt was done while wearing the HMD but without additional delay, thus accounting for about $115ms$ of total video-latency from camera to display. The third and fourth attempt had an additional artificial delay of $300ms$ and $500ms$ respectively.
The measured impact of different degrees of video-delay on hand-eye-coordination can be viewed in Table 3.2, representing a total of seven participants. Figure 3.7 shows the drawing-attempts of one of the participants.

Table 3.2: Maze test measurements with a total of seven participants.

Condition	Avrg. Time Needed	Avrg. Oversteps
a) No HMD	7s	0
b) HMD, +0ms	9.5s	0.6
c) HMD, +300ms	15.2s	3.7
d) HMD, +500ms	22.5s	4.6

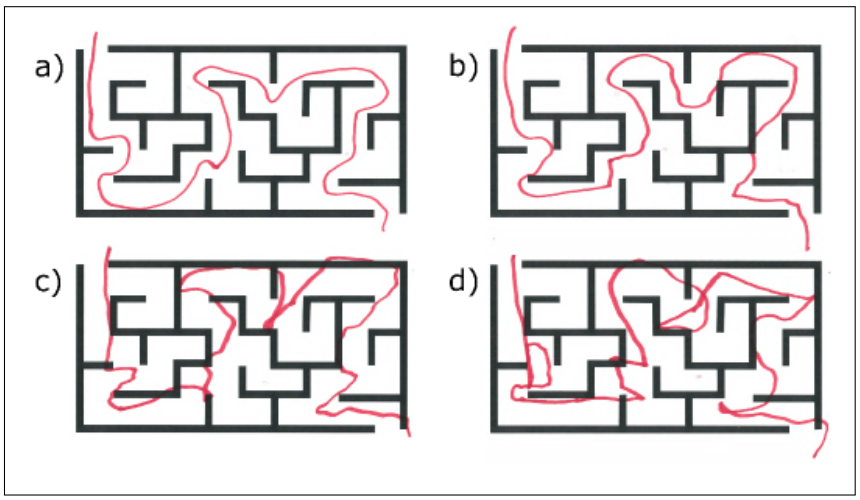

Figure 3.7: Drawing test with different degrees of latency: a) No HMD b) HMD without additional delay ($\approx 115ms$ delay) c) $+300ms$ ($\approx 415ms$ delay) d) $+500ms$ ($\approx 615ms$ delay). The time needed for each attempt by this participant was ca. $8s$, $17s$, $19s$ and $25s$ respectively.

A conclusion from this test was that an experiment-setup should not come close to the latency used in attempt 3 ($\approx 415ms$ delay) to minimize the impact on hand-eye-coordination. What is more, users indicated that they experienced slight visually-induced discomfort and nausea during the experiment. Especially on the background that a seated position and a task that required little to no head movement was the baseline of the test, such a strong impact suggests extra caution when it comes to introducing additional delay. This is along the lines of a study conducted by TNO and the U.S.-military [EJRP12] suggesting that a video-delay below $150ms$ in a comparable setup (without hand-eye-coordination tasks) is generally bearable.

Finally, the outcomes led to the decision to neglect experiments about the impact of delay on social communication and to focus solely on the impact of video quality. This decision was supported by prior studies stating that social interaction through tele-communication would still accept delays between $100 - 600ms$ [GJK+14]. Thus, testing for it would exceed the delay limits for a comfortable HMD-use before impairing social interaction. What is more, this finding supports Hypothesis 3. When inducing nausea, the QoE is bound to decrease.

3.4.2 Preliminary Test 2: Visual Acuity with Playing Cards

An initially desired test scenario was a poker game with one player being visually impaired by the HMD. With proficient players, this would have potentially led to an elaborate assessment of how much their general Poker skills and social interaction during the game would be decreased by the HMD-setup in relation to video delay and -quality. Additionally, Poker requires a certain degree of visual acuity to "read" the other players and to see the actions on the table. An eventual experiment would have generated insights in how much video-quality is needed to play the game in an acceptable manner.

In order to evaluate the possible range of artificial visual impairment for the HMD, a vision test with the best possible video delivery was created. The quality was based on the encoding quality of the motion-jpeg-encoder in GStreamer and set to 85% since higher values would cause significant video-delay.
The test featured a table with three covered rows of poker-sized playing cards (88.9 x 63.5mm). Each participant was asked to sit down in front of the table in an upright position while wearing the headset (Figure 3.8). At first, the participants were given five playing cards to hold in their hands and asked to read the value and suit out loud. Subsequently, each row on the table was uncovered from closest to furthest which the participants were asked to read out loud as well. The reading distances were $50 - 70cm$ for the hand cards and ca. $1m$, $1.5m$ and $2m$ for the table rows from close to far. The test was conducted with a total of seven participants and the success-rates can be viewed in Table 3.3.

Table 3.3: Visual acuity measurements with a total of seven participants.

Row	**Distance**	**Avrg. Score**
Hand	0.5-0.7m	5/5
Close	1m	1.14/5
Middle	1.5m	0.14/5
Far	2m	0/5

The results show that it was easily possible to read the cards on the hand with the highest video quality in Prototype 1. The rows on the table were less successfully distinguished and the low averages for the middle and far row showed that the system decreases in visual acuity in distances greater than 0.7 meters. It would therefore not be sufficient for a Poker game scenario since the game requires the players to read the cards on their

own. Thus, further artificial impairment would not take place within an acceptable range. Furthermore, providing aids to help the player identify the cards would alter the game mechanics and the way proficient players play too much. In consequence, the test scenario would become far less representative. Therefore, the experiment as described in section 3.5 was created.

Figure 3.8: Test setup for assessing visibility of gaming cards in poker-typical distances.

3.5 Final Experiment Setup

This experiment formed the main experiment of this study. It aimed at assessing the impact of degradation in video resolution on QoE and (social-) interaction. The test was conducted with two participants at a time to facilitate social interaction. One was participating without any alteration whereas the second person was wearing Prototype 3. Both persons were being seated at a table and not allowed to touch each other (Figure 3.9). In order to spark a conversation between the participants, a riddle was being presented. The riddle took shape in the puzzle game "Tangram" (Figure 3.10). These puzzle tiles assured a certain degree of visibility despite the visual impairment. Furthermore, the game is explained easily and provides the needed collaboration.
The video feed of the camera was displayed within the HMD and altered in resolution. The two depicted resolutions were 1920x1080 as the high resolution and 1024x576 as lower

resolution. The high value was picked since it was the maximum resolution of the camera. The low value was picked as a negotiation between a significant bandwidth decrease and a resolution that appeared to still allow for a) recognition of faces in conversational distances (≈ 2m) and b) visibility of the puzzle tiles.

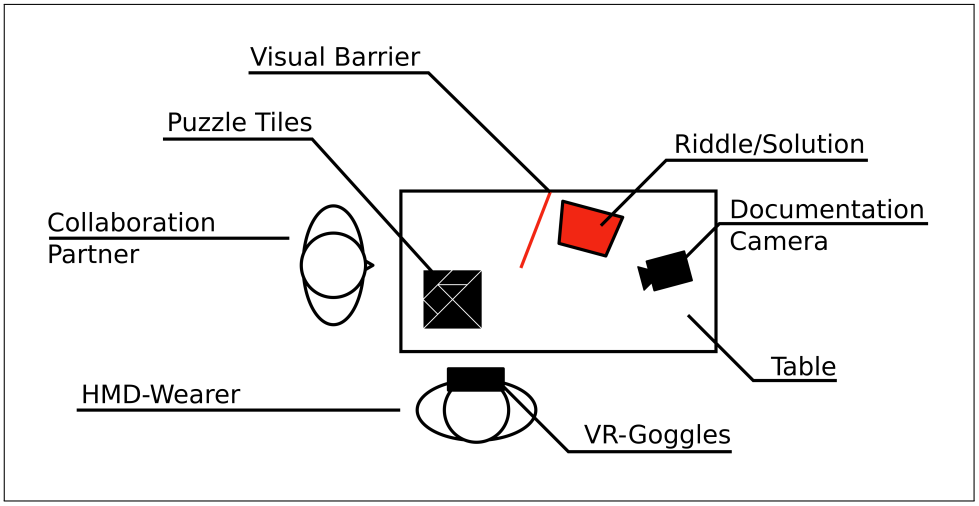

Figure 3.9: The participants are sitting at a table with a vision-barrier to hide the solution from the executing participant. The barrier was removed for the first phase of the test.

The HMD-wearer was not allowed to use his hands to touch or point at anything or manipulating the puzzle tiles during the whole experiment. This limitation was set in order to emulate his physical absence/remote presence. Furthermore, that person was instructed to only move the head and not alter the seating position.
The test was conducted in two stages. The first stage (five minutes) was spent trying to figure out puzzles in collaboration and without a presented solution. In case a puzzle was solved before the end of that phase, another riddle was presented.
After the collaboration time, a solution for another riddle was presented to the HMD-participant. This was done to trigger instructional communication. During the instructional phase, the HMD-wearer had to explain to the other person how the puzzle is to be solved (only through voice). This phase also lasted five minutes. The test ended after ten minutes.

3.5.1 Variables and Measurement

The independent variable is the video quality of the video feed, directed to the VR-HMD. Based on the capabilities of the camera, the high and low resolutions were set to 1920x1080

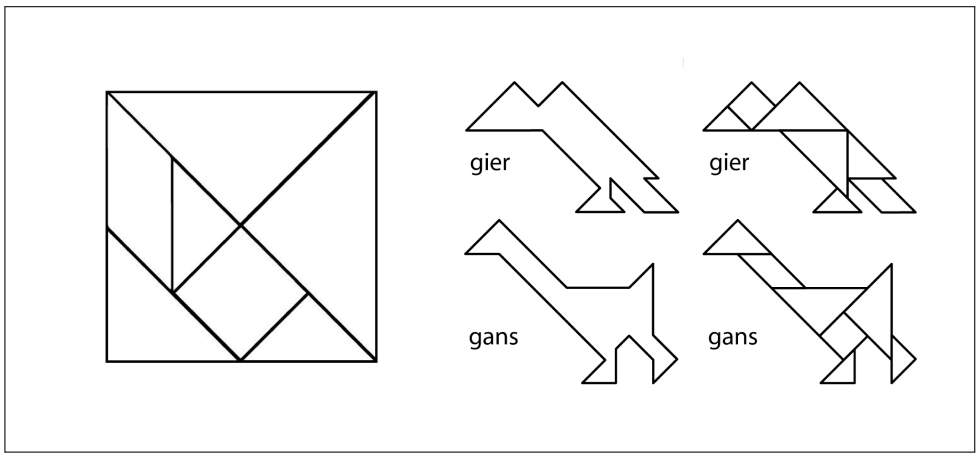

Figure 3.10: Chinese puzzle game Tangram. The tiles (left) are used to lay predefined silhouettes (right).

and 1024x576 respectively which accounts for a resolution decrease of approximately 70% . This in turn means a potentially significant decrease in required bandwidth[18]. Dependent variables are the subjective quality of communication between the two participants and the subjective level of comfort for the HMD-wearer in relation to the presented video quality. Only one quality level is presented during each experiment round since the used setup did not allow for swift switching. Since it was assumed that a learning effect could occur for the non-wearer during the initial test phase, the roles were not switched and each participant only took part in the experiment once.

The participants were recorded in video and audio for reference material. After the experiment, the participants were asked to fill out a questionnaire. The questionnaire was using the Likert scale and category ratings. In order to register eventual further remarks and insights, a comment section was also included. The questionnaire can be viewed in appendix B. Each question is designed to feed a dimension for the assessment of the system. The dimensions are

- Visual Acuity
- (Social-) Awareness
- Video Parameters
- General Assessment

The first dimension aims at getting an understanding on how well the hardware displays a remote physical environment. Similar to this, the (social-) awareness category features

[18]The nominal decrease is still dependent on degree and way of encoding and not directly proportional to the amount of pixels!

questions that assesses how well entities like collaboration- or communication partners are perceived and how well the system supports interaction (in form of visual feedback and other). The video parameters are based on subjective QoE-measurements and aim at giving insights in the needed quality for such a system. The general assessment category covers parameters like overall quality and possible endurance of potential users.

3.5.2 Test Group

The experiment was conducted with students at the University of Twente and TNO-employees at the TNO office, located in the centre of The Hague, Netherlands. This split guaranteed a certain number of expected rounds and a higher variety in age distribution. In total, 60 people participated in 30 rounds, accounting for 30 HMD-wearers and 30 collaboration partners. Since the resolution was only altered between rounds, these 30 rounds are split in 15 high resolution sessions and 15 low resolution sessions. The total participants' age ranged between 19 and 62 with an average age of 29 (see Figure 3.11).

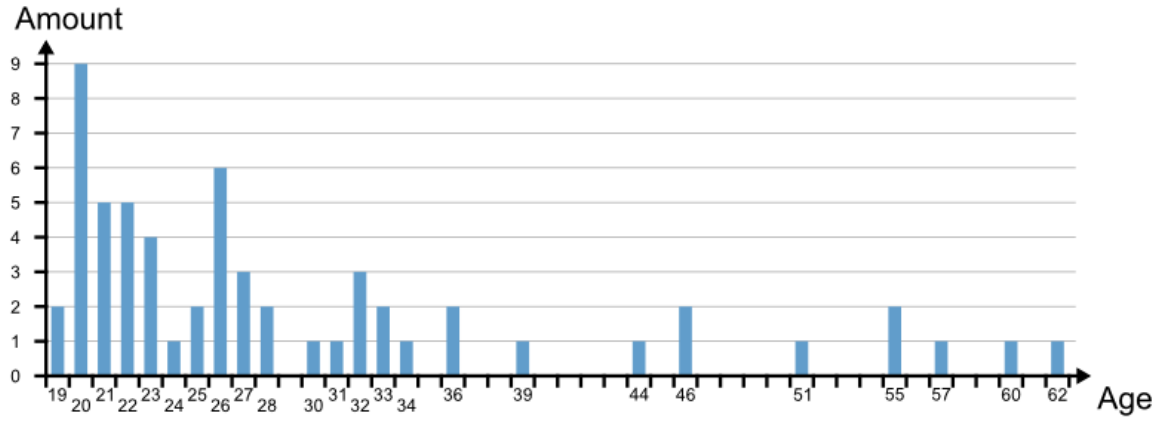

Figure 3.11: Age distribution of the test participants.

If persons with vision-correcting glasses participated, they were asked to play the role of the collaboration partner in order to keep the visual assessment of the video footage as unbiased as possible. Therefore, HMD-priority was given to the persons without the need for optical aids or contact lens wearers. The resolutions were altered after every experiment and regardless of the participant for a random distribution. This also ensured

that the number of experiments for both groups stayed equal (± 1) in case the experiment had to be stopped.

3.5.3 Ethical Considerations

In case user experiments are conducted by TNO, an Institutional Review Board is required to approve the test setup. The following considerations are being implemented to keep the test participants safe and to meet the board's criteria.
Since the test is altering visual perception, it has a slight potential to induce nausea or a general decrease in comfort. A study that featured a remote controlled robot, equipped with a camera-gimbal that was controlled by the pilot's head-movement shows that a similar system with ca. $150ms$ delay is generally bearable [EJRP12]. Only two out of 18 participants had to stop early because of nausea or other discomfort. Responses regarding nausea, dizziness or discomfort accounted to 5 in total. The described feelings of discomfort were assumed to be related to several possible factors:

- Delay between user's head movement and the robot's camera movement
- Delay in video
- Weight/design of gear
- Stereo vision setup
- Claustrophobia/lack of space

The most critical factor in the proposed test setup is the video delay between the head-mounted cameras and the video display in the HMD. The maximum delay measured[19] in the current setup is ca. $160ms$ for the high resolution and $140ms$ for the low resolution. It is therefore close to the situation, described in the study with the remote-controlled robot [EJRP12]. The main difference however is the connection between the participant's head movement and the camera movement. Since the cameras are directly attached to the HMD, a direct orientation-manipulation is assured. Furthermore, positional camera-changes on a large scale as undertaken with the robot are not part of the proposed test. Thus, only intended movements from the participants will have an effect on the video feed (contrary to e.g. movement through the robot's driving motions.)

[19] See subsection 3.3.1 for the measurement method.

3.5.4 Limitations and Expectations

The used setup did not allow for real-time monitoring of frame rate and other parameters since it would risk a stable performance. In fact, during frame rate measurements the perceived quality actually worsened due to additional jitter while A-Frame's VR-mode was enabled. Although there is little evidence that the values fluctuated significantly during the experiment, measuring them would have solidified the experiment outcomes.

Probably the greatest limitation is the alteration of the video feed through the HMD on several levels. Firstly, the actual video-resolution undergoes a change while being sent to the HMD through video-scaling for perspective correction. This could change the QoE despite a stable pixel-count of the video itself. Secondly, the visual input is altered through distortion as described in section 2.4.5. Combined with the influence of the optical setup of HMDs, it might alter the perceived video quality as well, depending on the used model. Therefore, measurements can only give insight about similar HMDs. What is more, the requested resolution was almost never displayed by A-Frame during preparation. Therefore, higher resolutions were requested to actually reach the desired resolutions. It turned out only after the experiment that the code presented in appendix D had the values for width and height switched around. A comparing test with the resolution measurement tool however revealed that the presented resolution was the same after the code was corrected and the actual desired resolution was requested. Therefore it can be assumed that the bug had no impact on the results.
An additional limitation was generated by the need to scale the video-feed for perspective correction as described in Figure 3.6. Possible effects in a HMD-mediated scenario can only be assumed at this point and are recommended as a future research subject. Since the scaling was not altered during the switch in resolution, a direct comparison of both is still valid for this setup.
Lastly, the frame rate for the high resolution setup (25 fps) could not be matched to the frame rate of the low resolution (30 fps). Since measurements during the tests were not possible, presented frame rates can only be stated with limited confidence. It is however likely that an average 5-fps-difference remained during the user tests. The reason for the occurrence of this difference is suspected to be related to bandwidth issues. The higher resolution is likely to decrease the frame rate if requesting too much bandwidth. A potential effect of this difference is a lower rating for the video quality if users do not differentiate between resolution and frame rate. Additionally, there might be a higher

chance to experience nausea with the high resolution setting than with the low resolution setting.

4 Test Results & Discussion

4.1 Ratings

Figure 4.1 and Figure 4.2 illustrate the degree of agreement for questions 1-10, given by the HMD-wearers of the high resolution group (H) and low resolution group (L) respectively. The box-plots indicate a mean-rating with the line in the middle of the box. The box itself spreads from quartile 1 to quartile 3 and includes 50% of the ratings. The "whiskers" indicate occurrences outside the two quartiles within a range of 1.5 times the interquartile range. Each dot is indicating a rating outside the whiskers range.[20]
Ratings of 1.0 would indicate strong disagreement. 4.0 would indicate neutrality whereas 7.0 would show a high degree of agreement. In the following, the mean ratings are being presented. Question 11-14 were rated on a 1-5 scale, asking either for a quality estimation (questions 11&12) or a noticeability rating (questions 13&14). All ratings have been accumulated with 15 votes each, if not further specified. The related questions are listed in Table 4.1 and the original questionnaire can be viewed in appendix B.

Question 1 with average ratings of 5.9(H) and 6.2(L) shows that the setup was generally sufficient for distinguishing high contrast forms in size of $10 - 20cm$ over a distance of up to $2m$. The statements for question 2 with a rating of 6.3(H) and 6.5(L) indicate a high degree of presence for the collaboration partner as perceived by the HMD-wearer in both groups. This is supported by a 6.1(H) and 6.2(L) for question 3, asking for readability of the partner's gestures.
Questions 4 only registered a (H)-count of 13 and an (L)-count of 14. Question 5 is represented similarly with 7 (H)-votes and 7 (L)-votes. The low counts for both questions were based on participants stating that they were not looking at their partners' faces

[20] For more information see [Acz96, p.32-35] or `http://mathworld.wolfram.com/Box-and-WhiskerPlot.html`, last accessed: July 28, 2016

Table 4.1: Average high (H) and low (L) ratings per questionnaire question.

#	Question	Ratings (H)	Ratings (L)
1	I was able to identify the riddles and solutions (table and billboard).	5.9	6.2
2	I was aware of my partner's actions.	6.3	6.5
3	I was able to read my partner's gestures (e.g. pointing at something, etc.).	6.1	6.2
4	I was able to read my partner's facial expressions.	3.0	3.4
5	I could tell if my partner was looking at me.	3.0	2.9
6	I felt distracted by the quality of the video.	3.9	4.7
7	During the instruction task: I could follow my partner's actions and was able to intervene if necessary.	5.8	6.0
8	I can imagine using the setup for telecommunication.	5.4	4.6
9	I could have used this system for longer than the duration of the test.	4.5	4.5
10	I felt distracted by the delay of the video.	4.4	5.3
11	The overall video quality was...	3.3	3.1
12	Compared to a non-mediated face-to-face communication, the communication with my partner was...	3.0	2.9
13	Delay in our communication was...	3.9	4.2
14	The overall video delay was...	3.5	3.3

and could therefore not assess the question. Possible reasons for that will be assessed in the discussion (Section 4.5). Question 4 was rated with an average of 3.0(H) and 3.4(L). Question 5 gained another 3.0(H) and a 2.9(L).
Question 6 about the degree of distraction caused by the video quality was rated with an average 3.9(H) and 4.7(L). In this case a low grade is desired. The (H)-grading close to a 4.0 indicates moderate concern about video quality whereas the (L)-grading shows a worse grade. The wide distribution of answers in both groups however suggest that this measurement is highly individual in terms of perception. The degree of attention towards the partner addressed in question 7 was rated 5.8(H) and 6.0(L) in average, suggesting a high degree of communication potential in both cases. This rating stands in contrast to the recorded statements about questions 4 and 5. Possible reasons for that are stated in the discussion section (Section 4.5).

When asked to imagine the potential for this setup as a tele-communication system

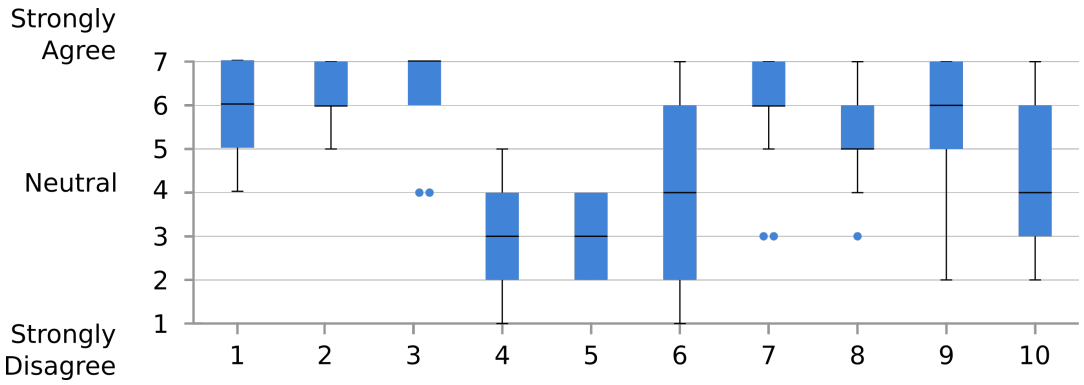

Figure 4.1: Degree of agreement for questions 1-10 of the questionnaire for the high resolution group. For question 6 and 10 a low score is desired.

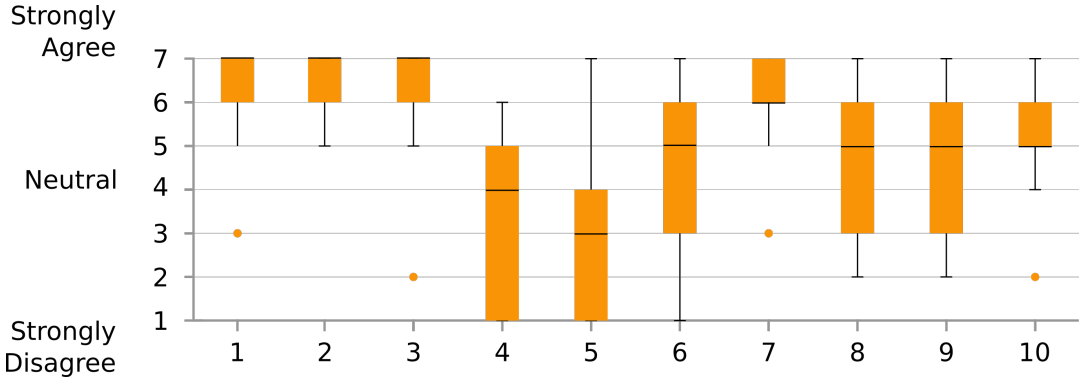

Figure 4.2: Degree of agreement for questions 1-10 of the questionnaire for the low resolution group. For question 6 and 10 a low score is desired.

(question 8), the average rating reached a 5.4(H) and a 4.6(L). The high distribution in the (L)-group signifies that Prototype 3 would not make up for a satisfying system to all users in the group, yet. The slightly more focussed (H)-group is also still not entirely convinced about the current setup. Also the willingness to use the system for longer than the 10 test minutes (question 9) was only moderately existent with an average rating of 4.5 for both (H) and (L). For question 10 asking about how distracted the HMD-wearers were by the video delay, the average response was a 4.4(H) and a 5.3(L). This also indicates need for improvement.

The ratings for question 11 and 12 for the high resolution group are shown in Figure 4.3. The low resolution group is represented in Figure 4.4 . The groups rated the overall video quality as “fair” with an average of 3.3(H, 14 ratings) and 3.1(L). Compared to a non-mediated face-to-face communication (question 12), the communication during the test was rated with “fair” as well by an average grade of 3.0(H) and 2.9(L, 14 ratings).

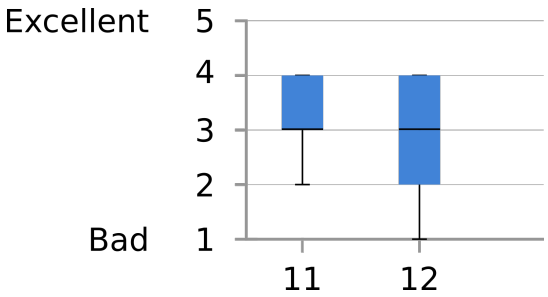

Figure 4.3: Rating for questions 11 and 12 of the questionnaire for the high resolution group. High grades are desired.

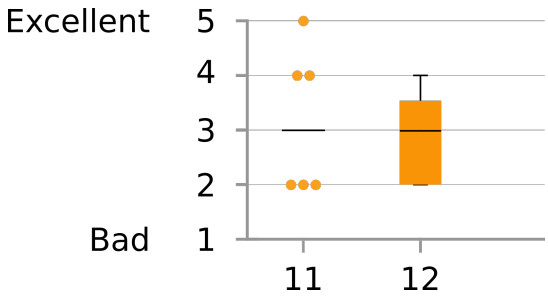

Figure 4.4: Rating for questions 11 and 12 of the questionnaire for the low resolution group. High grades are desired.

The answers for questions 13 and 14 are illustrated in Figure 4.5(H) and Figure 4.6(L). When asked if a delay in communication was perceived (question 13), the groups confirmed with a 3.9(H, 14 ratings) and a 4.2(L, 13 ratings) on the noticeability-scale. This is almost aligned with the noticeability of the overall video delay, regarded with a 3.5(H) and a 3.3(L).

The collaboration partners (no HMD) answered question 12 with positive averages of 4.1(H) and 3.8(L) indicating that the communication was good when compared to a non-mediated situation. A communication delay (question 13) was rated with a confident average of 4.0(H) and 3.7(L) as slightly noticeable.

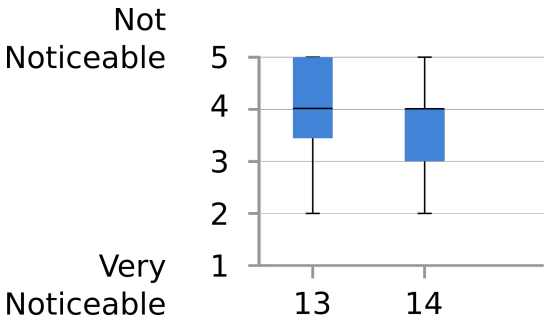

Figure 4.5: Noticeability rating for questions 13 and 14 of the questionnaire for the high resolution group. A low grade is desired.

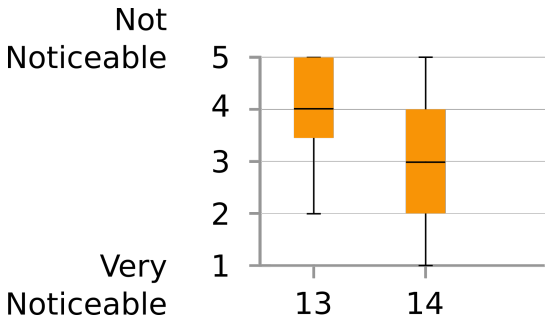

Figure 4.6: Noticeability rating for questions 13 and 14 of the questionnaire for the low resolution group. A low grade is desired.

4.2 Statistical Analysis

Calculating the statistical difference of the two group's means as a p-value with a t-test for each question revealed that none of the values are below the used alpha-value of 0.05. It can therefore be concluded that the two groups have no statistically significant difference in any of the questions. This in turn suggests, that the assessed parameters show little to no correlation with the alteration of resolution in the tested range. Hypothesis 1 and 2 can therefore be rejected for values within this range.

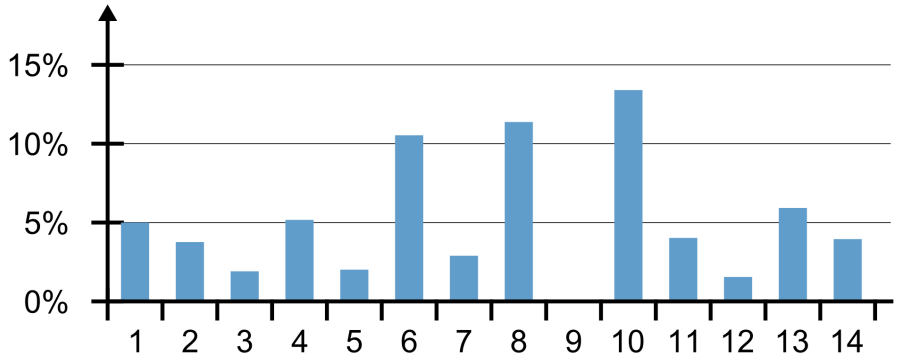

Figure 4.7: Percentaged difference of average rating between the two groups per question.

Figure 4.7 shows the difference of average rating between the two groups for each question in percentage. Question 6, 8 and 10 show a strong deviation from the rest. This in combination with the high distribution of ratings indicates that the outcome might change if a bigger population is being tested.
The distribution in question 6 (degree of distraction by video quality) could have been caused by the fact that participants have not been trained to distinguish between different aspects of video quality and therefore intended to rate different things.
The difference of question 8 (willingness for prolonged use) can have a variety of different and very individual causes such as deviating tolerances for motion sickness, the gear's weight and other. That this is solely based on the video quality is therefore unlikely.
Question 10 about the degree of distraction due to delay has an unexpected distribution within the two groups. The measurements show that the high quality setting has $20ms$ more delay than the low quality setting and still the low resolution group shows a more concise rating towards higher distraction than the high-resolution group. This spawns the question whether the combination of video parameters has a higher

impact on the QoE than each parameter on its own as e.g. illustrated by Zinner et al. [ZHAH10].

4.3 Participant Comments

The participants had the opportunity to leave comments at the end of the questionnaire. This section was used especially by the HMD-wearers. The different comments can be summarised in the following categories:

1. Impairment of communication by HMD
2. Discomfort
3. Video parameters
4. Difference in communication compared to non-mediated scenario

Category number one includes comments about about the limited FoV (5 comments), not looking at the partner (4) and general impairment (2). It was generally stated that the limited FoV pushed the collaboration partner out of focus. Therefore, the HMD-wearers were mainly focussing on the task at hand and disregarded (social) communication cues, they might have picked up otherwise. Here, nodding and eye-contact were mentioned as such cues. Analysing the videos showed that both partners generally focussed on the puzzle and less on each other. This was even the case during the instructional phase where a more dependent communication was expected. The HMD-wearers however focussed more on the solution and on how the tiles were handled and the collaboration partners relied more on their partner's voice than on visual cues. Since a HMD covers most of the face, communicating through facial expressions was very limited.
The discomfort category summarises comments about nausea/visually induced discomfort (8 comments), the weight of the HMD (1) and eye strain (1). These comments indicate, that the video-delay and frame rate were not ideal. Based on comparison of the setup with recommendations from developers and manufacturers, this was an expected outcome. Despite several statements about slight discomfort only one participant felt the need to take a short break between the two experiment phases to recover.
Category number three revealed opinions about the low video quality (4 comments), the low frame rate (2) and a lack in depth perception (1). Comments in these dimensions were also expected, based on the experience gained from the prototype development. A lack in depth perception however is an interesting insight since spatial orientation or

hand-eye-coordination was not required from the participant. But a count of only one comment regarding this suggests that missing depth perception did not seem to be a general problem. A further interesting remark, stated by several participants was that in order to reduce the risk of getting motion-sick, they would restrict their head movement. This might add to the FoV-problem that was described earlier.
Interestingly enough, 4 collaboration partners (non-HMD) stated that the communication was not significantly different, compared to a communication scenario without the partner wearing a HMD. This could be based on the fact that the HMD-wearer was still present in person and could be seen at the desk. Also, the audio-channel was not mediated through technology so that spoken communication was not particularly skewed. Additionally, although movements were restricted for the HMD-wearer, body language was still used on a basic level which might have improved the communication.

4.4 Video analysis

The videos of the experiment-sessions were taken to capture minor events that might have not been reflected in the questionnaires. A re-occurring theme that could be observed with many of the HMD-participants was a – sometimes seemingly unconscious – tendency to use small hand-gestures whenever vocal communication would not be sufficient. What is more, the videos confirm that the HMD-participants barely looked at the collaboration partner and focussed more on the puzzles. This is however also true for the collaboration partners, and it was often stated afterwards that they did not know what the HMD-wearer would see. Similar statements indicated that the collaboration-partner's visual focus was on the tiles and they would mainly listen to the HMD-wearers voice. This can be seen in the videos as well.

4.5 Discussion

Prior to the video-pipeline development, several hypotheses have been generated (Section 3.1). Hypothesis 1 stated that a decrease in video resolution would affect the QoE:

> Hypothesis 1: *Altering the video-resolution in favour of bandwidth savings will lower the QoE after a certain threshold.*

The results of the questionnaire indicate that the quality of the video led to distraction. It can however not be said clearly, if this is solely based on the resolution or other video-parameters as well. It is highly possible, that a combination of video-parameters rather than a sole decrease in resolution led to the occasionally very low ratings. The overall quality of the video was rated as "fair" in both groups. This suggests, that the change in video resolution within the tested range only has a marginal effect on the QoE, if any.
Hypothesis 2 was connecting the change in resolution to the assessment of quality in social interaction:

> Hypothesis 2: *Altering the video-resolution in favour of bandwidth savings will worsen social interaction after a certain threshold.*

The results of the main experiment indicate that altering the resolution within the tested range affects the quality of social interaction only marginal, if any. This in turn suggests, that the decrease in video-resolution in a HMD has not fallen below the acceptability threshold for the described scenario.

Hypothesis 3 connected the video-delay to the quality of social interaction:

> Hypothesis 3: *Video-delay in ranges that are acceptable for video-based tele-communication (*$100-600ms$*) will already affect the QoE and social interaction negatively in a HMD-application.*

The maze-test already showed that video delays between $300 - 600ms$ were affecting the hand-eye-coordination. Furthermore, it was stated that delay above $150ms$ would have a high chance of visually inducing nausea and motion-sickness. Therefore it can be concluded that Hypothesis 3 is generally true. It is however crucial to distinguish between different kinds of delay as already illustrated in Section 3.1.

Based on the findings, the specifications of Prototype 3 can be located on the QoE-curves as illustrated in Figure 4.8. The delay in the setup falls below the acceptability range. This assessment is based on the fact that participants had to restrict their head-movements in order to avoid discomfort. The mainly positive rating about the delay in communication however suggest that the video delay itself was not the problem but its connection to the motion-to-photon-latency. this supports the assumption, that splitting up this relation might allow for a higher video-delay as it is the case for video-based tele-communication, presented on a standalone monitor.

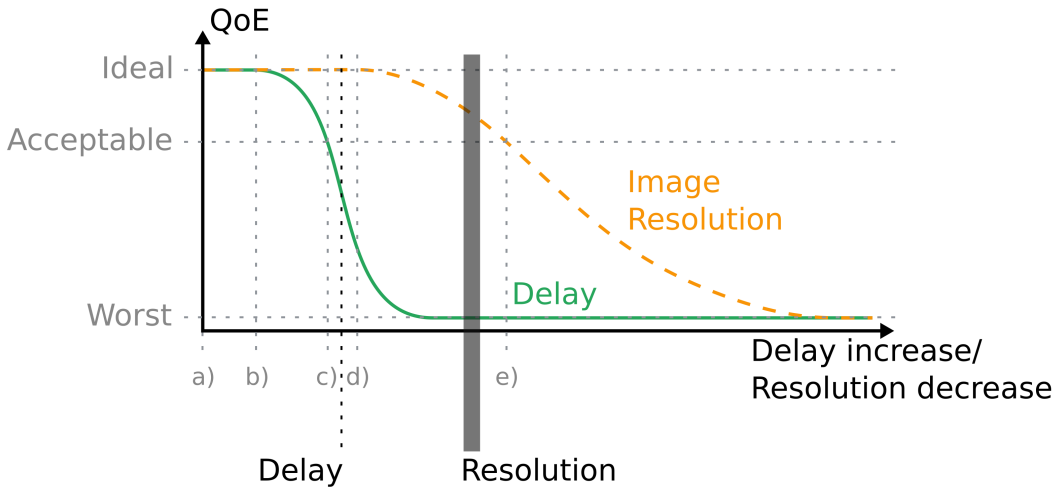

Figure 4.8: Positioning of the specifications from Prototype 3 on the QoE-curves for delay and resolution.

The tested resolution range is marked within the acceptability interval for resolution. It is however located in the lower end since statements about low video quality were made. The visual acuity appeared to be generally sufficient and social interaction was fairly possible for both test-groups. A more pressing problem seemed to be the FoV and the low frame rate in combination with the delay which restricted the HMD-wearer. This trend can also be seen in the comments. It appears therefore that this combination of video-parameters ultimately led them to keep their head rested and focussed on the task at hand rather than switching between the task and their partner.

It was generally expected, that a change in resolution would make a difference, especially for the close eye-display-distances that are common in HMDs. Why the big decrease in resolution did not appear to have a measurable effect can only be assumed at this point. Possible explanations might be found in the fact, that the video was scaled for perspective correction so that the impact of resolution-alteration was derogated. Further explanations might be found in the unique viewing conditions, the HMD provides. Image distortion and other factors might alter the image in a way, that a change in resolution has a lesser visual effect than under regular viewing conditions with a standalone display.

5 Conclusion

This study was conducted in order to answer a set of research questions that aimed at assessing whether resolution-based bandwidth negotiations for video could be applied to spherical 360°-video that is presented in a HMD. Here, the main focus was on its impact on QoE and social interaction. The main research question was:

Which effects does a resolution-based bandwidth-saving approach have on a HMD-mediated video-application?

In order to focus the research, sub-questions were formulated:

1. *What are the effects of a decrease in video resolution on the QoE in a HMD?*
2. *Which display resolution and video resolution is sufficient for facilitating a face-to-face-communication and a collaboration task with a HMD?*
3. *What are the effects of a decrease in video resolution on social interaction that is mediated by a HMD?*
4. *How much video-delay is acceptable with a HMD for tele-communication?*

Sub-question 1 can only be answered for the resolution-range between 1920x1080 and 1024x576. Significant effects in this range were not registered for the QoE in the measured dimensions.

The answers for sub-question 2 similarly clear. The experiment featured an Oculus "CV1" as HMD, which has a resolution of 1080x1200 per eye. Social interaction and face-to-face-communication with this display-resolution was possible. It can therefore be regarded as sufficient for remote face-to-face communication and remote collaboration. Since both groups showed no statistically significant difference in their assessment for any of the questions, it can be concluded that even the low video resolution of 1024x576 was generally sufficient for this purpose as well.

Sub-question 3 can be answered very similar to sub-question 1. The measurements did not yield major effects on social interaction when changing the resolution in the tested range. Here, a too small FoV, low frame rate and a high delay appear to be more influential than resolution as standalone video-parameters.

Sub-question 4 is very much dependent on the way, video-delay affects the viewing conditions. Since motion-to-photon-latency was closely linked to the video-delay in the setup, acceptable margins range below $150ms$, if not in the range of only $15ms$ or even lower. If video delay and motion-to-photon-delay would be separated as it might be in an actual spherical 360°-video the case, the acceptable margins might be higher. Further testing is required to assess this.

Based on the assessment of the sub-questions, the main research question can be answered: It can be concluded that an alteration of video resolution in favour of bandwidth savings – within the tested range of 1920x1080 and 1024x576 – does not have any significant effect on the QoE and on social interaction when mediated with a VR-{hmd.

5.1 Future Work

It became apparent throughout the course of this thesis that certain parameters seem to have greater effects on social interaction than other when mediated through the described hardware. These dominant parameters appear to be frame rate, motion-to-photon-latency and FoV. It is therefore desirable to test each of these parameters independently with expanded ranges to find clearer acceptability thresholds. The tests could at the same time benefit from trained users, to really distinguish between these factors. Furthermore, testing combinations of the given parameters might generate further insights in feasible bandwidth negotiation-tactics.
In order to assess acceptable margins for video delay in more detail, it would be helpful to repeat the tests in an actual spherical 360°-video application to separate video-delay from motion-to-photon-latency.
Another interesting question would be whether a group conference with one person attending via tele-communication is more successful if he/she uses a traditional video conferencing tool (e.g. Skype, Google Hangouts) or a HMD-based setup with the possibility to look around in the remote location.

Bibliography

[AAS+14] Sebastian Arndt, Jan-Niklas Antons, Robert Schleicher, Sebastian Möller, and Gabriel Curio. Using electroencephalography to measure perceived video quality. *IEEE Journal of Selected Topics in Signal Processing*, 8(3):366–376, 2014.

[ABD11] Saamer Akhshabi, Ali C Begen, and Constantine Dovrolis. An experimental evaluation of rate-adaptation algorithms in adaptive streaming over http. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 157–168. ACM, 2011.

[Acz96] Amir D Aczel. *Complete business statistics*. Richard d Irwin, 1996.

[AMV11] Patrice Rondao Alface, Jean-François Macq, and Nico Verzijp. Evaluation of bandwidth performance for interactive spherical video. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2011.

[BB85] Randolph Blake and Karin Boothroyd. The precedence of binocular fusion over binocular rivalry. *Perception & Psychophysics*, 37(2):114–124, 1985.

[BBDM+13] Kjell Brunnström, Sergio Ariel Beker, Katrien De Moor, Ann Dooms, Sebastien Egger, Marie-Neige Garcia, Tobias Hossfeld, Satu Jumisko-Pyykkö, Christian Keimel, Mohamed-Chaker Larabi, et al. Qualinet white paper on definitions of quality of experience. 2013.

[BL13] Frank Biocca and Mark R Levy. *Communication in the age of virtual reality*. Routledge, 2013.

[BM07] Doug A Bowman and Ryan P McMahan. Virtual reality: how much immersion is enough? *Computer*, 40(7):36–43, 2007.

[CCL02] Kirsten Cater, Alan Chalmers, and Patrick Ledda. Selective quality rendering by exploiting human inattentional blindness: looking but not seeing. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 17–24. ACM, 2002.

[DM98] Andrew T Duchowski and Bruce H McCormick. Gaze-contingent video resolution degradation. In *Photonics West'98 Electronic Imaging*, pages 318–329. International Society for Optics and Photonics, 1998.

[EJRP12] Linda R Elliott, Chris Jansen, Elizabeth S Redden, and Rodger A Pettitt. Robotic telepresence: Perception, performance, and user experience. Technical report, DTIC Document, 2012.

[Fin51] Edgar F Fincham. The accommodation reflex and its stimulus. *The British journal of ophthalmology*, 35(7):381, 1951.

[GJK+14] Simon NB Gunkel, Jack Jansen, Ian Kegel, Dick CA Bulterman, and Pablo Cesar. The optimiser: monitoring and improving switching delays in video conferencing. In *Proceedings of Workshop on Mobile Video Delivery*, page 1. ACM, 2014.

[HD96] Steve Harrison and Paul Dourish. Re-place-ing space: the roles of place and space in collaborative systems. In *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, pages 67–76. ACM, 1996.

[Hof04] Hunter G Hoffman. Virtual-reality therapy. *SCIENTIFIC AMERICAN-AMERICAN EDITION*, 291:58–65, 2004.

[HR95] Ian P Howard and Brian J Rogers. *Binocular vision and stereopsis*. Oxford University Press, USA, 1995.

[KRZ+12] Claudia Kuster, Nicola Ranieri, H Zimmer, JC Bazin, C Sun, T Popa, M Gross, et al. Towards next generation 3d teleconferencing systems. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, pages 1–4. IEEE, 2012.

[LD97] Matthew Lombard and Theresa Ditton. At the heart of it all: The concept of presence. *Journal of Computer-Mediated Communication*, 3(2):0–0, 1997.

[Lee78] David N Lee. The functions of vision. *Modes of perceiving and processing information*, pages 159–170, 1978.

[LG13] Danielle E Levac and Jane Galvin. When is virtual reality "therapy"? *Archives of physical medicine and rehabilitation*, 94(4):795–798, 2013.

[LMT12] Stefan Lederer, Christopher Müller, and Christian Timmerer. Dynamic adaptive streaming over http dataset. In *Proceedings of the 3rd Multimedia Systems Conference*, pages 89–94. ACM, 2012.

[LSP83] Herschel W Leibowitz, CL Shupert, and Robert B Post. The two modes of visual processing: Implications for spatial orientation. In *Peripheral vision horizon display (PVHD), NASA conference publication*, volume 2306, pages 41–44. Citeseer, 1983.

[MK94] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[MMK12] Masahiro Mori, Karl F MacDorman, and Norri Kageki. The uncanny valley [from the field]. *Robotics & Automation Magazine, IEEE*, 19(2):98–100, 2012.

[MWWR93] Mark Mon-Williams, John P Warm, and Simon Rushton. Binocular vision in a virtual world: visual deficits following the wearing of a head-mounted display. *Ophthalmic and Physiological Optics*, 13(4):387–391, 1993.

[OSN+14] Peter Orosz, Tamás Skopkó, Zoltán Nagy, Pál Varga, and László Gyimóthi. A case study on correlating video qos and qoe. In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pages 1–5. IEEE, 2014.

[PK15] Frieder Pankratz and Gudrun Klinker. [poster] ar4ar: Using augmented reality for guidance in augmented reality systems setup. In *Mixed and Augmented Reality (ISMAR), 2015 IEEE International Symposium on*, pages 140–143. IEEE, 2015.

[Riv99] Giuseppe Riva. From technology to communication: Psycho-social issues in developing virtual environments. *Journal of Visual Languages & Computing*, 10(1):87–97, 1999.

[RM00] Giuseppe Riva and Giuseppe Mantovani. The need for a socio-cultural perspective in the implementation of virtual environments. *Virtual Reality*, 5(1):32–38, 2000.

[RPL+11] Albert Rizzo, Thomas D Parsons, Belinda Lange, Patrick Kenny, John G Buckwalter, Barbara Rothbaum, JoAnn Difede, John Frazier, Brad Newman, Josh Williams, et al. Virtual reality goes to war: A brief review of the future of military behavioral healthcare. *Journal of clinical psychology in medical settings*, 18(2):176–187, 2011.

[Sto11] Thomas Stockhammer. Dynamic adaptive streaming over http–: standards and design principles. In *Proceedings of the second annual ACM conference on Multimedia systems*, pages 133–144. ACM, 2011.

[SVS05] Maria V Sanchez-Vives and Mel Slater. From presence to consciousness through virtual reality. *Nature Reviews Neuroscience*, 6(4):332–339, 2005.

[VBNPS11] Ray Van Brandenburg, Omar Niamut, Martin Prins, and Hans Stokking. Spatial segmentation for immersive media delivery. In *Intelligence in Next Generation Networks (ICIN), 2011 15th International Conference on*, pages 151–156. IEEE, 2011.

[vdVBvKB14] T van der Vorst, R Brennenraedts, D van Kerkhof, and RNA Bekkers. Fast forward: How the speed of the internet will develop between now and 2020. 2014.

[vEKDI10] Rob van Eijk, André Kuijsters, Klaske Dijkstra, and Wijnand A IJsselsteijn. Human sensitivity to eye contact in 2d and 3d videoconferencing. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pages 76–81. IEEE, 2010.

[VGSS04] Vinoba Vinayagamoorthy, Maia Garau, Anthony Steed, and Mel Slater. An eye gaze model for dyadic interaction in an immersive virtual environment: Practice and experience. In *Computer Graphics Forum*, volume 23, pages 1–11. Wiley Online Library, 2004.

[ZHAH10] Thomas Zinner, Oliver Hohlfeld, Osama Abboud, and Tobias Hoßfeld. Impact of frame rate and resolution on objective qoe metrics. In *Quality of Multimedia Experience (QoMEX), 2010 Second International Workshop on*, pages 29–34. IEEE, 2010.

List of Figures

A Comments

Table A.1: Participant comments.

Category	**HMD**	**Res.**	**Comment**
Impairment of communication by HMD	Yes	High	"Do to the field of view and trying to not move my head (also to cope with delay) I did not see the face of my partner.[...] Definitely restricting the communication."
	Yes	High	"I could not see if he was turning his head towards me, as this was not in my peripheral vision"
	Yes	Low	"[...] the viewport of the headset is far less wide than what I see, which was less than ideal."
	Yes	low	"It was somewhat sweaty and limited my field of vision."
	Yes	Low	"[...] but the field of view was quite narrow[...]"
Discomfort	Yes	Low	"I got dizzy and felt stomach discomfort.[...] The biggest problem is the discomfort and dizziness it causes."
	Yes	High	"it caused slight naussea with rapid movements"
	Yes	Low	"Yes, I felt[...]dizzy. I tried to keep my head as still as possible to reduce this effect."
	Yes	High	"Turning my head made things a bit fuzzy."

Table A.2: Participant comments (continuation of table A.1).

Category	HMD	Res.	Comment
	Yes	High	"Slight nausea"
	Yes	High	"Yes, slightly nausiating. I didn't try to check his facial expressions because I didn't want to move my head too much (nausiating)"
	Yes	Low	"Yes, minor eye-strain."
	Yes	High	"Yes, nausea"
	Yes	High	"Firstly, it was quite heavy and warm."
	Yes	High	"I didn't try to check his facial expression because I didn't want to move my head too much (nausiating)"
Video parameters	Yes	Low	"[...]resolution was noticeably low and the frame rate was disturbingly low"
	Yes	Low	"The delay when moving the head is very uncomfortable"
	Yes	High	"Due to the discomfort of the refresh rate when moving, I kept focussed on the solution and puzzle[...]."
	Yes	High	"Feeling dizzy because of the video quality and delay."
	Yes		"[...] Also because looking at the instruction and switching towards the place of the puzzle peaces on the table is quite uncomfortable if done quickly."
	Yes	Low	"The visual quality was not very bad. I could identify the shapes and work with my partner."
	Yes	High	"video quality & delay can be annoying for longer use, also the small fov makes the task more difficult."
	Yes	High	"My eyes felt betrayed."
	Yes	High	"Biggest issue was lack of depth perception."
	Yes	High	"The quality of the video was annoying, it's like you're not wearing glasses when you should."

Table A.3: Participant comments (continuation of table A.2).

Category	**HMD**	**Res.**	**Comment**
Difference in communication compared to non-mediated scenario	No		"Yes communication was difficult, but I'm not sure if that was due to the headset. It could also be the fact that I've never talked to this person before. [...]"
	No		"Because you don't see the face of the other it is more like an one way communication. Your not looking at the one who is leading you, only at the task because his face is covered by a mask. normally a nod is enough to confirm something now he had to say it."
	No		"Did not feel any difference with [compared to] non-headset colleague."
	No		"I noticed I didn't look at the other user, something I would normally do."
	No		"Very little difference from normal face to face communication."
	No		"I have no clue what was shown in the glasses, but communicating was quite natural."
	No		"Yes, Communication seemed less direct. I tend not to look at the other participant as I assume he could not see me. Maybe due to the presence of the goggles."

B Questionnaire

Experiment title: “Impact of video quality on a social interaction-/coordination task, mediated by a virtual reality head mounted display”

Experiment nr. & category (to be filled out by test supervisor): _____ — _____

I was wearing the headset during the test: Yes ◯ No ◯
(Questions with a * should also be answered by the **participant without HMD**.)

Please rate the following statements according to the indicated categories. Check **only one** circle per question/statement.

	Strongly Disagree		Somewhat Disagree		Somewhat Agree		Strongly Agree
1) I was able to identify the riddles and solutions (table and billboard).	○	○	○	○	○	○	○
2) I was aware of my partner's actions.	○	○	○	○	○	○	○
3) I was able to read my partner's gestures (e.g. pointing at something, etc.).	○	○	○	○	○	○	○
4) I was able to read my partner's facial expressions.	○	○	○	○	○	○	○
5) I could tell if my partner was looking at me.	○	○	○	○	○	○	○
6) I felt distracted by the quality of the video.	○	○	○	○	○	○	○

	Strongly Disagree		Somewhat Disagree		Somewhat Agree		Strongly Agree
7) During the instruction task: I could follow my partner's actions and was able to intervene if necessary.	○	○	○	○	○	○	○
8) I can imagine using the setup for telecommunication.	○	○	○	○	○	○	○
9) I could have used this system for longer than the duration of the test.	○	○	○	○	○	○	○
10) I felt distracted by the delay of the video.	○	○	○	○	○	○	○

	Bad	Poor	Fair	Good	Excellent
11) The overall video quality was...	○	○	○	○	○
12) *Compared to a non-mediated face-to-face communication, the communication with my partner was...	○	○	○	○	○

	Very Noticeable				Not Noticeable
13) *Delay in our communication was...	○	○	○	○	○
14) The overall video delay was...	○	○	○	○	○

Please answer the questions below in written form. Use the space in the boxes. In the last box, you are given space for comments and additional input.

*Did you experience any problems in completing the task?

Did you experience any discomfort with the headset? In what way?

*Space for comments

C GStreamer Code

C.1 GStreamer Command Line Code

Double vision with black bar in the middle:

```
gst-launch-1.0 -ev compositor name=c sink_0::xpos=1500 background=black !
    queue ! videocrop top=100 bottom=100 ! capsfilter caps="video/x-raw,width
    =2460,height=1080,framerate=(fraction)30/1,format=I420" ! tee name=t !
    queue ! textoverlay text="1920x1080x30 (85)" halignment=left valignment=
    top ! jpegenc quality=85 ! decodebin ! xvimagesink sync=false v4l2src
    device=/dev/video1 ! capsfilter caps="video/x-raw,width=1280,height=960,
    framerate=(fraction)30/1" ! videoflip method=counterclockwise ! queue ! c.
     v4l2src device=/dev/video0 ! capsfilter caps="video/x-raw,width=1280,
    height=960,framerate=(fraction)30/1" ! videoflip method=counterclockwise !
     queue ! c.
```

Start several Windows with different delay:

```
gst-launch-1.0 -ev compositor name=c sink_0::xpos=960 ! queue ! videocrop top
    =100 bottom=100 ! capsfilter caps="video/x-raw,width=1920,height=1080,
    framerate=(fraction)30/1,format=I420" ! tee name=t ! queue ! textoverlay
    text="1920x1080x30 (85, 0s)" halignment=left valignment=top ! jpegenc !
    queue ! decodebin ! queue max-size-buffers=0 max-size-time=0 max-size-
    bytes=0 min-threshold-time=10 ! xvimagesink sync=false t. ! queue !
    textoverlay text="1920x1080x30 (85, 0.5s)" halignment=left valignment=top
    ! jpegenc ! queue ! decodebin ! queue max-size-buffers=0 max-size-time=0
    max-size-bytes=0 min-threshold-time=500000000 ! xvimagesink t. ! queue !
    textoverlay text="1920x1080x30 (85, 0.3s)" halignment=left valignment=top
    ! jpegenc ! queue ! decodebin ! queue max-size-buffers=0 max-size-time=0
    max-size-bytes=0 min-threshold-time=300000000 ! xvimagesink sync=false
```

```
v4l2src device=/dev/video1 ! capsfilter caps="video/x-raw,width=1280,
height=960,framerate=(fraction)30/1" ! videoflip method=counterclockwise !
 queue ! c. v4l2src device=/dev/video0 ! capsfilter caps="video/x-raw,
width=1280,height=960,framerate=(fraction)30/1" ! videoflip method=
counterclockwise ! queue ! c.
```

C.2 Python Script for Shader Inclusion

The code below is an altered version of Floren Thiery's "gst-oculus-fpv" python script[21]. The script integrates a shader into a GStreamer pipeline. The here used shader is an early development version of the original Oculus barrel distortion shader. Alterations were made to add functionalities like delay request, resolution request and optical parameters such as interpupillary distance and degree of distortion.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# Copyright 2015, Florent Thiery

import sys
import time
import json
import logging
logger = logging.getLogger('FpvPipeline')

import gi
gi.require_version('Gst', '1.0')
from gi.repository import GObject, Gst
GObject.threads_init()
Gst.init(None)
Gst.debug_set_active(True)
Gst.debug_set_colored(True)
Gst.debug_set_default_threshold(Gst.DebugLevel.WARNING)
#Gst.debug_set_threshold_for_name("glimage*", 5)
```

[21] https://github.com/fthiery/gst-oculus-fpv, last accessed: July 7, 2016

```
config_default = {
    #'headtracker_enable': True,
    'headtracker_enable': False,
    'headtracker_fov': 70,
    #'render_fps': 60,
    'render_fps': 60,
    'font_size': 30,
    'bitrate_video': 4000,
    'display_width': 1920,
    'display_height': 1080,
    #'display_width': 1280,
    #'display_height': 800,
    #'display_width': 2160,
    #'display_height': 1200,
    #'display_width': 3840,
    #'display_height': 2160,
    'benchmark_mode': False,
    #'benchmark_mode': True,
}

#source = "v4l2src ! video/x-raw, format=(string)YUY2, width=(int)640, height
    =(int)360, pixel-aspect-ratio=(fraction)1/1, interlace-mode=(string)
    progressive, colorimetry=(string)1:4:7:1, framerate=(fraction)30/1"
#pipeline_source = 'videotestsrc is-live=true ! video/x-raw, format=(string)
    YUY2, width=(int)720, height=(int)480'
#pipeline_source = 'filesrc location=../sim.mp4 ! qtdemux ! h264parse !
    avdec_h264 ! queue'

#the following line is working code (1cam not 90 degree turn) and taps
    directly into the webcam feed
#pipeline_source = 'v4l2src device=/dev/video0 ! video/x-raw,framerate=30/1,
    width=1920,height=1080 ! queue'

#the following pipeline is for double vision
pipeline_source= ' v4l2src device=/dev/video1 ! capsfilter caps="video/x-raw,
    width=1280,height=960,framerate=(fraction)30/1" ! videoflip method=
```

```
    counterclockwise ! queue ! c. v4l2src device=/dev/video0 ! capsfilter caps
    ="video/x-raw,width=1280,height=960,framerate=(fraction)30/1" ! videoflip
    method=counterclockwise ! queue ! compositor name=c sink_0::xpos=1500
    background=black ! queue ! videocrop top=100 bottom=100 ! capsfilter caps
    ="video/x-raw,width=2460,height=1080,framerate=(fraction)30/1,format=I420"
     ! tee name=t ! queue ! jpegenc quality=85 ! decodebin'

#the following line is working code (1cam not 90 degree turn) and taps
    directly into the webcam feed
#pipeline_source = 'v4l2src device=/dev/video0 ! video/x-raw,framerate=30/1,
    width=1280,height=960 ! videoflip method=counterclockwise ! queue'

#the following line is working code (1cam not 90 degree turn) and taps
    directly into the webcam feed
#pipeline_source = 'v4l2src device=/dev/video0 ! video/x-raw,framerate=30/1,
    width=1280,height=960 ! videoflip method=counterclockwise ! queue'

#test pipeline to access fifo "/tmp/pipe" (not working yet)
#pipeline_source = 'filesrc location=/tmp/pipe blocksize=1843200 ! video/x-raw
    ,framerate=30/1,width=1280,height=960,format=I420 ! queue'

# as of gstreamer 1.6.2 glimagesink currently does not yet post key presses
    on the bus, so lets use xvimagesink to toggle recording using the "r" key
    for testing and "q" for quitting (ctrl+c also works)
#pipeline_preprocess_pattern = 'tee name=src ! queue name=qtimeoverlay !
    timeoverlay name=timeoverlay font-desc="Arial {font_size}" silent=true !
    glupload ! glcolorconvert ! glcolorscale ! videorate ! video/x-raw(memory:
    GLMemory), width=(int){display_width}, height=(int){display_height}, pixel
    -aspect-ratio=(fraction)1/1, interlace-mode=(string)progressive, framerate
    =(fraction){render_fps}/1, format=(string)RGBA ! gltransformation name=
    gltransformation ! glshader location=oculus.frag ! gldownload ! queue !
    videoconvert ! xvimagesink name=glimagesink'
pipeline_preprocess_pattern = 'tee name=src ! queue name=qtimeoverlay !
    glupload ! glcolorconvert ! glcolorscale ! videorate ! video/x-raw(memory:
    GLMemory), width=(int){display_width}, height=(int){display_height}, pixel
    -aspect-ratio=(fraction)1/1, interlace-mode=(string)progressive, framerate
    =(fraction){render_fps}/1, format=(string)RGBA'
```

```
pipeline_headtracker = 'gltransformation name=gltransformation'
pipeline_sink = 'glshader name=glshader ! glimagesink name=glimagesink'

pipeline_encoder_pattern = 'src. ! queue ! videoconvert ! x264enc tune=
    zerolatency speed-preset=1 bitrate={bitrate_video} ! mp4mux ! filesink
    location=test.mp4'
#pipeline_encoder_pattern = 'src. ! queue ! vaapipostproc ! vaapiencode_h264
    rate-control=2 bitrate={bitrate_video} ! h264parse ! mp4mux ! filesink
    location=test_vaapi.mp4'

shader_pattern = '''
#version 100
#ifdef GL_ES
precision mediump float;
#endif
varying vec2 v_texcoord;
uniform sampler2D tex;

const vec4 kappa = vec4(1.0,1.7,0.7,15.0);

const float screen_width = {display_width}.0;
const float screen_height = {display_height}.0;

//default = 0.9
const float scaleFactor = 0.8;

const vec2 leftCenter = vec2(0.2, 0.5); //x, y
const vec2 rightCenter = vec2(0.8, 0.5); //x, y

//default: -0.5 but that seems too less, so: 0.005
const float separation = -0.05;

const bool stereo_input = false; //"true" changes the image in an interesting
     way

// Scales input texture coordinates for distortion.
```

```
vec2 hmdWarp(vec2 LensCenter, vec2 texCoord, vec2 Scale, vec2 ScaleIn) {{
    vec2 theta = (texCoord - LensCenter) * ScaleIn;
    float rSq = theta.x * theta.x + theta.y * theta.y;
    vec2 rvector = theta * (kappa.x + kappa.y * rSq + kappa.z * rSq * rSq +
        kappa.w * rSq * rSq * rSq);
    vec2 tc = LensCenter + Scale * rvector;
    return tc;
}}

bool validate(vec2 tc, int eye) {{
    if ( stereo_input ) {{
        //keep within bounds of texture
        if ((eye == 1 && (tc.x < 0.0 || tc.x > 0.5)) ||
            (eye == 0 && (tc.x < 0.5 || tc.x > 1.0)) ||
            tc.y < 0.0 || tc.y > 1.0) {{
            return false;
        }}
    }} else {{
        if ( tc.x < 0.0 || tc.x > 1.0 ||
             tc.y < 0.0 || tc.y > 1.0 ) {{
             return false;
        }}
    }}
    return true;
}}

void main() {{
    float as = float(screen_width / 2.0) / float(screen_height);
    vec2 Scale = vec2(0.4, as); //vec2(0.5, as);
    vec2 ScaleIn = vec2(2.0 * scaleFactor, 1.0 / as * scaleFactor);
    vec2 texCoord = v_texcoord;
    vec2 tc = vec2(0);
    vec4 color = vec4(0);

    if ( texCoord.x < 0.5 ) {{
        texCoord.x += separation;
        texCoord = hmdWarp(leftCenter, texCoord, Scale, ScaleIn );
```

```
        if ( stereo_input ) {{ //default: !stereo_input
            texCoord.x *= 2.0;
        }}

        color = texture2D(tex, texCoord);

        if ( !validate(texCoord, 0) ) {{
            color = vec4(0);
        }}
    }} else {{
        texCoord.x -= separation;
        texCoord = hmdWarp(rightCenter, texCoord, Scale, ScaleIn);

        if ( stereo_input ) {{ //default: !stereo_input
            texCoord.x = (texCoord.x - 0.5) ;//* 2.0;
        }}

        color = texture2D(tex, texCoord);

        if ( !validate(texCoord, 1) ) {{
            color = vec4(0);
        }}
    }}

    gl_FragColor = color;
}}
'''

def save_config(config_dict, config_fpath):
    with open(config_fpath, 'w') as config_file:
        json.dump(config_dict, config_file, sort_keys=True, indent=4,
            separators=(',', ': '))

def read_config(config_fpath):
    with open(config_fpath, 'r') as config_file:
        return json.load(config_file)
```

```
try:
    config_fpath = 'config.json'
    config = read_config(config_fpath)
    for k in config_default.keys():
        changed = False
        if not config.get(k):
            config[k] = config_default[k]
            changed = True
        if changed:
            save_config(config, config_fpath)
            print('Config updated')
except Exception as e:
    print('Error while parsing configuration file, using defaults (%s)' %e)
    config = config_default

if config['headtracker_enable']:
    from rift import PyRift
    import math

if config['benchmark_mode']:
    import os
    os.environ['vblank_mode']="0"
    NUM_BUFFERS = 1000
    #pipeline_source = 'filesrc location=/dev/zero num-buffers=%s blocksize
        =1382400 ! videoparse format=rgba width=720 height=480' %NUM_BUFFERS
    #pipeline_source = 'filesrc location=/dev/zero num-buffers=%s blocksize
        =518400 ! videoparse format=i420 width=720 height=480' %NUM_BUFFERS
    #pipeline_source = 'filesrc location=/dev/zero num-buffers=%s blocksize
        =691200 ! videoparse format=yuy2 width=1920 height=1080' %NUM_BUFFERS
    pipeline_source = 'filesrc location=/tmp/pipe num-buffers=%s blocksize
        =1843200 ! videoparse format=I420 width=1280 height=960' %NUM_BUFFERS
    #pipeline_source = 'filesrc location=/temp/pipe num-buffers=%s blocksize
        =691200 ! capsfilter caps="video/x-raw,width=1920,height=1080,
        framerate=(fraction)30/1,format=I420 ! queue !" ' %NUM_BUFFERS
    pipeline_sink = 'glshader name=glshader ! glimagesink name=glimagesink
        sync=false'
```

```
class FpvPipeline:
    def __init__(self, mainloop=None):
        self.post_eos_actions = list()
        self.record = False
        self.mainloop = mainloop
        if config['headtracker_enable']:
            self.rift = PyRift()
            logger.debug('OpenHMD detection result:')
            logger.debug(self.rift.printDeviceInfo())
    def start(self):
        if self.is_running():
            self.disable_headtracker_fov()
            self.add_post_eos_action(self.start)
            self.stop()
            return
        logger.info("Record: %s" %self.record)
        pipeline_desc = self.get_pipeline_description(self.record)
        logger.debug("Running %s" %pipeline_desc)
        self.pipeline = self.parse_pipeline(pipeline_desc)
        if self.record:
            self.set_record_overlay()
        self.activate_bus()
        self.update_shader()
        if config['headtracker_enable']:
            self.enable_headtracker_fov()
        self.pipeline.set_state(Gst.State.PLAYING)
        self.start_time = time.time()
    def toggle_record(self):
        self.record = not self.record
        logger.info('Toggling record to state %s' %self.record)
        self.start()
    def stop(self):
        GObject.idle_add(self.send_eos)
    def exit(self):
        self.schedule_exit()
        self.stop()
    def schedule_exit(self):
```

```
        logger.info('Exiting cleanly')
        if self.mainloop:
            logger.debug('Stopping mainloop')
            self.add_post_eos_action(self.mainloop.quit)
        else:
            logger.warning('Exiting (sys.exit)')
            self.add_post_eos_action(sys.exit)

    # Initializations

    def get_pipeline_description(self, record=False):
        pipeline_preprocess = pipeline_preprocess_pattern.format(**config)
        pipeline_encoder = pipeline_encoder_pattern.format(**config)

        if config['headtracker_enable']:
            elts = [pipeline_source, pipeline_preprocess, pipeline_headtracker,
                pipeline_sink]
        else:
            elts = [pipeline_source, pipeline_preprocess, pipeline_sink]

        p = ' ! '.join(elts)
        if record:
            p = "%s %s" %(p, pipeline_encoder)
        return p

    def activate_bus(self):
        self.bus = self.pipeline.get_bus()
        self.bus.add_signal_watch()
        self.bus.connect('message', self._on_message)
        self.bus.connect('message::eos', self._on_eos)
        self.bus.connect('message::error', self._on_error)
    def enable_headtracker_fov(self):
        gltransformation = self.pipeline.get_by_name('gltransformation')
        gltransformation.set_property('fov', config['headtracker_fov'])
        gltransformation.set_property('pivot-z', 30)
        self.headtracker_tid = GObject.timeout_add(int(1000/config['render_fps
            ']-2),self.poll_oculus, priority=GObject.PRIORITY_HIGH)
```

```
def disable_headtracker_fov(self):
    GObject.source_remove(self.headtracker_tid)
def poll_oculus(self):
    self.rift.poll()
    x, y, z, w = self.rift.rotation
    yaw = math.asin(2*x*y + 2*z*w)
    pitch = math.atan2(2*x*w - 2*y*z, 1 - 2*x*x - 2*z*z)
    roll = math.atan2(2*y*w - 2*x*z, 1 - 2*y*y - 2*z*z)
    #logger.debug("rotation quat: %f %f %f %f, yaw: %s pitch: %s roll: %s"
        % (x, y, z, w, yaw, pitch, roll))
    self.update_headtracker_fov(pitch, -roll, yaw)
    return True

def update_headtracker_fov(self, rot_x, rot_y, rot_z):
    gltransformation = self.pipeline.get_by_name('gltransformation')
    gltransformation.set_property('rotation-x', rot_x)
    gltransformation.set_property('rotation-y', rot_y)
    gltransformation.set_property('rotation-z', rot_z)
def update_shader(self):
    shader = shader_pattern.format(**config)
    glshader = self.pipeline.get_by_name('glshader')
    try:
        glshader.set_property("fragment", shader)
    except TypeError:
        with open('/tmp/shader.frag', 'w') as f:
            f.write(shader)
        glshader.set_property("location", "/tmp/shader.frag")
def set_record_overlay(self):
    o = self.pipeline.get_by_name('timeoverlay')
    o.set_property('text', 'Rec')
    o.set_property('silent', False)

# Event callbacks
def _on_eos(self, bus, message):
    logger.info("Got EOS")
    self.pipeline.set_state(Gst.State.NULL)
```

```
        if config['benchmark_mode']:
            took = time.time() - self.start_time
            fps = NUM_BUFFERS/took
            logger.info('Benchmark: %.1f fps' %fps)
            self.schedule_exit()
        self.run_post_eos_actions()
    def _on_error(self, bus, message):
        err, debug = message.parse_error()
        error_string = "{0} {1}".format(err, debug)
        logger.error("Error: {0}".format(error_string))
    def _on_message(self, bus, message):
        t = message.type
        if t == Gst.MessageType.ELEMENT:
            struct = message.get_structure()
            sname = struct.get_name()
            if sname == 'GstNavigationMessage':
                event = struct.get_value('event')
                estruct = event.get_structure()
                if estruct.get_value('event') == "key-release":
                    key = estruct.get_value('key')
                    self._on_key_release(key)
                #self.print_struct_content(estruct)
        else:
            #logger.debug("got unhandled message type {0}, structure {1}".
                format(t, message))
            pass
    def _on_key_release(self, key):
        logger.info('Key %s released' %key)
        if key == "r":
            self.toggle_record()
        elif key == "q":
            self.exit()

    # GStreamer helpers
    def parse_pipeline(self, pipeline):
        return Gst.parse_launch(pipeline)
    def is_running(self):
```

```
        if not hasattr(self, 'pipeline'):
            return False
        return self.pipeline.get_state(Gst.CLOCK_TIME_NONE)[1] == Gst.State.
            PLAYING
    def print_struct_content(self, struct):
        for i in range(struct.n_fields()):
            field_name = struct.nth_field_name(i)
            field_value = struct.get_value(field_name)
            print('%s = %s' %(field_name, field_value))
    def send_eos(self, *args):
        if hasattr(self, "pipeline") and self.is_running():
            logger.info("Sending EOS")
            event = Gst.Event.new_eos()
            Gst.Element.send_event(self.pipeline, event)
        else:
            logger.info("No pipeline or pipeline not running, skipping EOS
                emission")
            self.run_post_eos_actions()
    def run_post_eos_actions(self):
        for action in self.post_eos_actions:
            logger.debug('Calling %s' %action)
            action()
        self.post_eos_actions = list()
    def add_post_eos_action(self, action):
        if callable(action):
            self.post_eos_actions.append(action)
        else:
            logger.error('Action %s not callable' %action)

if __name__ == '__main__':
    logging.basicConfig(
        level=getattr(logging, "DEBUG"),
        format='%(asctime)s %(name)-12s %(levelname)-8s %(message)s',
        stream=sys.stderr
    )

    ml = GObject.MainLoop()
```

```
f = FpvPipeline(ml)
GObject.idle_add(f.start)
#GObject.timeout_add_seconds(3, f.toggle_record)
#GObject.timeout_add_seconds(13, f.exit)

try:
    ml.run()
except KeyboardInterrupt:
    logger.info('Ctrl+C hit, stopping')
    f.exit()
```

D A-Frame Code

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>webcam2</title>
    <meta name="description" content="Webcam ? A-Frame">
    <script src="aframe.min.js"></script>
    <!-- <script src="https://aframe.io/releases/0.2.0/aframe.min.js"></script-->
  </head>
  <body>
    <a-scene>
      <a-assets>
     <video id="webcamvideo" autoplay loop="true" src="..">
  </a-assets>

  <!--a-entity id="KeyControl" keyboard-controls> </a-entity-->
      <a-camera id="player" position="0 0 0" fov="100" look-controls="enabled: false">
<!-- set the size [video is 16/9 ] -->
                <a-video id="webvideo" src="#webcamvideo" width="16" height="9"
 position="0 0 -10" scale="-1 -1" rotation=" 0 0 0"></a-video>
      </a-camera>
  </body>

  <script>

var video = document.querySelector('#webcamvideo');
////Logitech c390////
```

```
var wIDTH = 3840; //=1920x1080
var hEIGHT = 2160;

//var wIDTH = 1920; //= 1024x576
//var hEIGHT = 1080;

navigator.getUserMedia = navigator.getUserMedia || navigator.webkitGetUserMedia
|| navigator.mozGetUserMedia || navigator.msGetUserMedia ||
navigator.oGetUserMedia;

if (navigator.getUserMedia) {
    // getting the webcam and setting input resolution
    navigator.getUserMedia({video: { width: hEIGHT, height: wIDTH} },
    handleVideo, videoError);
    //window.setTimeout(check_fps_image,5000);
}

function handleVideo(stream) {
    console.log("now play video" + window.URL.createObjectURL(stream));
    console.log("video playin: width = " + video.videoWidth + ";
    height = " + video.videoHeight );
    video.src = window.URL.createObjectURL(stream);
    video.play();
    //window.setTimeout(printResolution,1000);
}

function printResolution(){
  var is_playing = !(video.paused || video.ended || video.seeking ||
  video.currentTime <= 0 || video.readyState < video.HAVE_FUTURE_DATA);
  if (!is_playing){
    window.setTimeout(printResolution,2000);
    return;
  }
  console.log("video playin: width = " + video.videoWidth + ";
  height = " + video.videoHeight );
```

```
  window.setTimeout(printResolution,2000);
}

function videoError(e) {
    // do something
}

var lastImg = "";
var frameCounter = 0;
var refresh = Date.now();

// calculate the frame rate of a video element
function check_fps_image(){

//check whether the video is playing
  var is_playing = !(video.paused || video.ended || video.seeking ||
  video.currentTime <= 0);

  if (!is_playing){
   window.setTimeout(check_fps_image,100);
    return;
  }

// the scale factor helps to get a good performance
scaleFactor = 0.05;

  var w = video.videoWidth * scaleFactor;
var h = video.videoHeight * scaleFactor;

var canvas = document.createElement('canvas');
canvas.width  = w;
canvas.height = h;
var ctx = canvas.getContext('2d');
ctx.drawImage(video, 0, 0, w, h);
```

```
var pngImg = canvas.toDataURL();

if (lastImg != pngImg){
frameCounter++;
lastImg = pngImg;
}

if (Date.now() - refresh > 1000){
    console.log("time interval: " + (Date.now() - refresh));
  console.log("videoleft fps - Image: " + frameCounter);
      console.log("Resolution " + video.videoWidth + "x" + video.videoHeight);
    frameCounter = 0;
refresh = Date.now();
}
window.setTimeout(check_fps_image,5);
}

</script>
</html>
```