Mixed Voxel Reality: Presence and Embodiment in Low Fidelity, Visually Coherent, Mixed Reality Environments

Holger Regenbrecht, *Member, IEEE*, Katrin Meng, Arne Reepen, Stephan Beck, and Tobias Langlotz

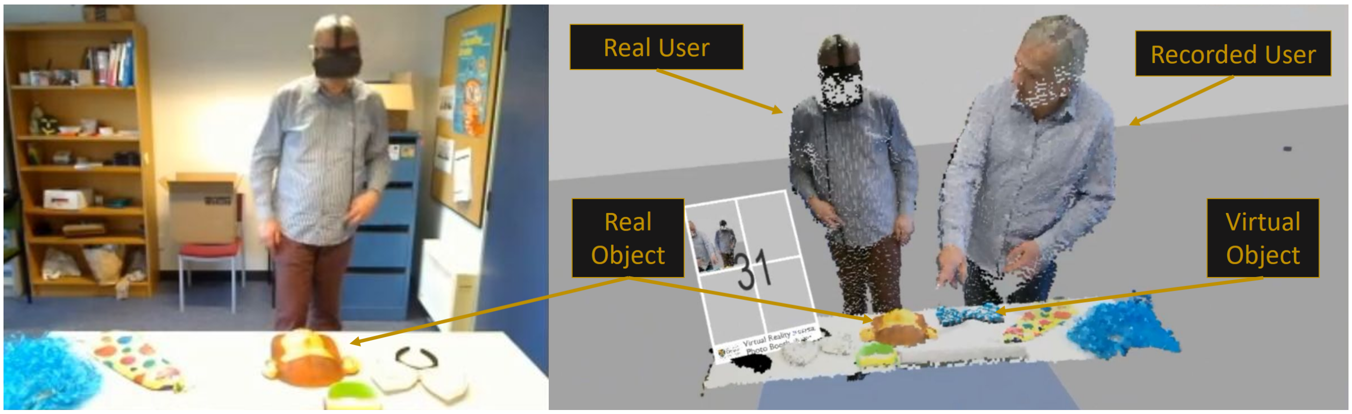

Figure 1: Left: real-world view as captured by a web cam Right: User in the mixed voxel reality seeing himself in a virtual mirror with a mix of real and virtual objects and another person

ABSTRACT

Mixed Reality aims at combining virtual reality with the user's surrounding real environment in a way that they form one, coherent reality. A coherent visual quality is of utmost importance, expressed in measures of e.g. resolution, framerate, and latency for both the real and the virtual domains. For years, researchers have focused on maximizing the quality of the virtual visualization mimicking the real world to get closer to visual coherence. This however, makes Mixed Reality systems overly complex and requires high computational power. In this paper, we propose a different approach by decreasing the realism of one or both visual realms, real and virtual, to achieve visual coherence. Our system coarsely voxelizes the real and virtual environments, objects, and people to provide a believable, coherent mixed voxel reality. In this paper we present the general idea, the current implementation and demonstrate the effectiveness of our approach by technical and empirical evaluations. Our mixed voxel reality system serves as a platform for low-cost presence research and studies on human perception and cognition, a host of diagnostic and therapeutic applications, and for a variety of Mixed Reality applications where users' embodiment is important. Our findings challenge some commonplace assumptions on "more is better" approaches in mixed reality research and practice—sometimes less can be more.

Keywords: mixed reality, augmented reality, believability, presence, voxel grid

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 INTRODUCTION

Computer-mediated realities (CMR) include all techniques on the continuum between Virtual Reality (VR) and Mixed Reality (MR). They are defined around the concepts of being computer-generated, interactive, three-dimensional, rendered in real-time, and to allow for the development of a sense of presence. While VR worlds are fully computer-generated, MR combines virtual computer-generated reality with the physical reality in a way that both worlds spatially align, i.e. virtual and real objects or subjects (persons) are perceived as being in the same space. However, already Milgram et al. separated visual fidelity of a MR environment ("Reproduction Fidelity") from the feeling of presence the MR environment creates ("Extent of Presence Metaphor") [24]. This raises the question of how much visual fidelity is needed and sufficient for (a) the acceptance of visually reconstructed real objects and subjects, (b) the acceptance of artificial objects, and (c) achieving a sense of presence in such an environment.

For VR, some researchers would argue that an increased visual realism leads to higher presence in such an environment [35], which would include acceptance and usability. Others would argue that realism does only marginally contribute to presence, much stronger factors being the spatial self-location and involvement aspects of presence [32]. Those aspects are mainly influenced by the user's possibilities to interact with the environment, actual or imagined [31]. For embodiment, and in particular for ownership in e.g. therapeutic applications, a certain degree of realism is required and is mediated by the interaction modalities [36, 29]. However, it remains unclear, which degree of visual realism is sufficient to be believable, usable, acceptable, and leading to a sense of presence in such a CMR, in particular MR environments.

In this paper we propose Mixed Voxel Reality—a CMR embodiment system which allows us to study the influence of real and virtual environments (and objects and persons) on presence and embodiment. The presented Mixed Voxel Reality system does not aim to generate photo-realistic realities but focuses on a different approach by decreasing the realism of one or both visual realms, real and virtual, to achieve visual coherence. If we are able to show that low-fidelity, non-photo-realistic MR environments can create a feeling of presence we are not only able to contribute to the presence literature but also would decrease the hardware requirements for presence research in general and would open it to other low-cost platforms.

To achieve our goal, our Mixed Voxel Reality systems utilises a voxel-based rendering technique. A voxel here represents a colour point on a regular grid in three-dimensional space leading to a much simplified processing and rendering technique. While voxel techniques traditionally require more computational power and memory than polygonal techniques, today's GPU-based, and memory-rich computers offset for this and allow us to exploit the advantages of voxels. In particular, a unified model handling (everything is represented as voxels in a fixed grid), an inherently built-in occlusion handling, much easier collision detection, the ease of providing procedural models, and the modifiability and destructibility of objects are of advantage.

For our scenario, a high voxel resolution is of less interest. The voxel size, and therefore voxel resolution, is determined by taking into account the spatial resolution of the off-the-shelf depth camera (MS Kinect2) which is used to capture the real environment. The key idea is here that we sample voxels of digital objects into the same voxel space as physical objects scanned in real-time. While this allows for coherent spatial resolution between real and virtual we also need to take into account other artifacts that are a result of the used camera hardware (e.g. noise in color and depth data). Inspired by previous work for simulating the camera imperfections for 2D cameras [19] we are approximating the most relevant characteristics of the used depth camera and apply them similarly to the virtual objects. The entire system runs interactively using off-the-shelf hardware and supports a coherent visualization of real-time captured objects and persons, static objects (scanned or CAD data), and pre-recorded dynamic objects or persons (avatars).

While our targeted applications are primarily situated in the realm of therapy and rehabilitation, the system presented here (Mixed Voxel Reality) and our findings can be used for a host of other applications in e.g. human behaviour simulation, telepresence, entertainment and gaming. Wherever embodiment, i.e. agency, body ownership, and self-location/presence, is key, our approach can inspire or be used to improve user experience and to study human behaviour. With our work described in this paper, we

1. Show that even a low resolution voxel system can achieve the experience of presence in MR

2. Present a system platform which allows for the study of MR perception as well as for further MR applications

3. Present an effective technique for coherent real/virtual voxelized object rendering

4. Present an effective technique for the integration of voxel avatars in a MR scene

5. Contribute a prototypical software platform to the community which can be used for further presence studies with low to moderate hardware requirements.

In the remainder we discuss related work in presence and visually coherent rendering techniques, voxel-based visualization, and other MR and VR systems aiming for coherent, believable experiences. We describe in detail our Mixed Voxel Reality system followed by a technical evaluation of that system. We present a user study with 22 participants using our system to show levels of presence and embodiment achieved. Finally we discuss future opportunities and application areas of our system and approach.

2 Related Work

The sense of presence, or short presence, and the experience of embodiment are inter-related concepts. Presence can be decomposed into three factors [32]: spatial presence, involvement, and realism. Embodiment is believed to comprise [18]: agency, body ownership, and self-location. The factors spatial presence and self-location both refer to the same perception of spatially being part of an environment.

Mixed reality inherently combines virtual reality and reality in a way that both blend seemingly together [24]. In many or even most cases it is desirable to provide a user experience where both domains form one reality, i.e. the user is unable to distinguish between real and virtual. For instance in BurnAR, where flames are presented at a user's hand, it is the goal to give the user the impression that his or her hand is set on fire—real hand and virtual flames should form one reality [42]. Similarly, in MR telepresence applications the participating (reconstructed) 3D avatars should seamlessly blend with each other and with the surrounding virtual environment. This is usually achieved by trying to achieve the highest possible photorealism for for rendered and reconstructed content (environment, objects, people) [10, 39, 2].

A different approach is to alter the visual fidelity or visualization style for the real or virtual domain. For instance, Fischer et al. [7] stylize both the video stream and the virtual content in a video see-through AR system in a way that both form one visually coherent experience. Another approach is to stay within the realm of VR and render all three domains (environment, objects, people) with the same fidelity within one virtual environment [37]. While both of those approaches work well in their respective domains (video see-through (VST) AR and VR), applying the same techniques to three-dimensional MR is complex and also computationally expensive.

Today's computational power and advancements in algorithmic solutions allow for an affordable implementation of voxel-based approaches in VR and MR. While mixed reality volumetric rendering in the past (e.g. [30]) required specialised and expensive hardware, modern computers equipped with GPU capabilities and intelligent data handling techniques (e.g. [25]) lend themselves to a voxel-based approach for visual coherence. Real objects and people can be voxelized (e.g. [15]) and rendered in the mixed reality environment (e.g. [34]). Usually, those voxel representations are then turned into triangulated virtual objects to achieve higher photorealism [42, 2, 9, 6]. In particular for the rendering of human characters (persons) with a desire to render fine detail this triangulation effort is the most computational expensive and also requires domain-specific algorithms. In addition, even if a very high fidelity is achieved, there is a danger that the so called *uncanny valley* effect [11] occurs—first investigated with anthropomorphic robots: that any type of human-like object that has an almost, but not perfectly, realistic human appearance leads to the illusion to break; and with this leads to a break in presence.

When targeting presence and embodiment, is there really a need to improve photorealism? Couldn't a rather coarse resolution, voxel-based mixed reality system deliver presence and embodiment too? We are closing this gap in research by providing a system platform and evaluation which emphasizes visual coherence instead of visual fidelity, which is computational inexpensive, and could potentially applied in the realms of applications targeted by the other techniques described above, like virtual rehabilitation or telepresence.

3 Mixed Voxel Reality System

Our Mixed Voxel Reality system targets presence and embodiment in mixed-reality environments. Our main focus is to investigate presence and embodiment while having a coherent but low fidelity representation of the physical and virtual environment which includes objects and people (including the interacting user). Its tech-

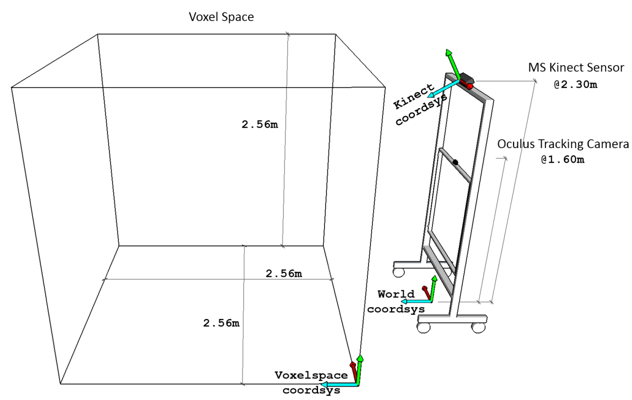

Figure 2: General setup of Mixed Voxel Reality system: The user interacts within a 2.56^3 m^3 space in front of a Kinect camera while wearing a head-mounted display tracked by a camera in front of him/her.

nical and functional requirements stem from therapeutic and diagnostic applications, but the presented basic architecture and implementation is not domain specific. The main requirements include affordability, the provision of an interactive experience (minimum guaranteed framerate of 60 fps, latency below 100 ms), perceived embodiment by the users (agency, ownership, self-location), and the experience of presence and believability of the environment, including its objects and subjects (people). Similarly to other mixed-reality systems our Mixed Voxel Reality system is built using different components responsible for capturing the environment, tracking the user in a defined space, and a rendering and display component that visually mixes a representation of the physical and virtual environment. We start with presenting the overall system before providing more details on how to achieve a coherent visualization within our Mixed Voxel Reality system.

System Overview Our Mixed Voxel Reality system is build around an interaction space of 2.56^3 m^3 in which the users of the system can later freely move. The dimensions were empirically defined by the area that can be well monitored by a depth camera (in our case a Microsoft Kinect v2 RGBD camera), allows accurate tracking of a head-mounted display (here an Oculus Rift CV1 head-mounted display) and allows us to easily map it into a voxel space with edges of equal length. The interaction space is located in front of a metal frame (Fig. 2). This metal frame serves as a mount for the Kinect sensor later used for capturing the environment, and the tracking camera for tracking the head-mounted display.

The user is wearing an immersive head-mounted display within the defined voxelspace and is able to freely look around and to interact with objects displayed as voxels (see Fig. 3). Within the mixed reality space experienced by the user not only the first person perspective elements are visible as voxel (objects, own hands and legs, etc.,) but also a virtual mirror as an effective form to develop the illusion of virtual body ownership [23]. This virtual mirror exactly reflects what is seen in the mixed reality scene (even if future versions might use the mirror otherwise or don't use it at all).

Hardware The Mixed Voxel Reality system in its current implementation is fully functional, stable, and has been tested for feasibility at different locations. Apart from the previously mentioned Oculus Rift CV1 head-mounted display, and a Microsoft Kinect v2 RGBD camera it comprises a standard personal computer (currently an Intel i7-6700 Quad-Core Processor, 4.0 GHz, 16GB of DDR4 Memory with an NVIDIA GTX 970 graphics card with 4GB of VRAM; operating system is Windows 10 64Bit).

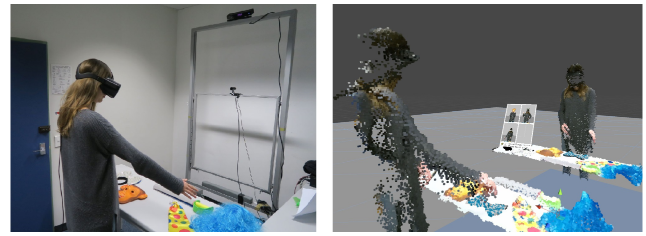

Figure 3: Left: User interacting with real objects in voxelspace; Right: Virtual third person view of the mixed scene.

Implementation The Kinect sensor's raw depth and color data are mapped into the 2.56^3 m^3 voxelspace. An exemplary voxel size of 1cm would result in a 256^3 voxel grid with 16.7 million potential voxels filling the space. While the Kinect sensor data would fill an angled frustum of rays with depth values, our voxel space is equally spaced and perpendicular to the interaction space. In its current implementation, the capture program utilizes the Kinect for Windows SDK 2.0. RGB color, depth, and skeleton data are retrieved and pre-processed before they are visualised. To accommodate for future needs of the system and to realize a more modular structure, capturing and rendering processes are separated and linked with a network component using a UDP socket connection. The capturing component sends UDP packets with the voxel information according to their position in the voxel space grid, indexed with a 16-bit unsigned integer for each of the three axes. The contained voxel is assigned an RGB color value with 8 bits per channel. Additionally, we encode a body index value (8 bit) derived from the Kinect SDK. Therefore each packet is a 10 byte/voxel structure. The networking is implemented as a local loop adaptor (one computer) or as a 1 Gb/s network connection (two computers). The rendering component uses the Unity Game Engine scenegraph and amends it in a way to achieve an effective networking and voxel rendering. We implement voxel objects inside Unity for fast and reliable handling and rendering of all voxels. The system defines different kinds of voxel objects to handle the different classes of elements present in the scene. These classes include body voxels that contain all voxel which have been identified as part of the user, environment voxels which are all voxels that do not belong to a user but are present in the real environment and captured by the depth sensor, and finally, virtual voxel objects. Virtual voxel objects are different from the other two classes as they do not represent any real world elements.

Unity is not a voxel-based game engine, instead it works with vertices which form triangles to make up a mesh. In our case, each voxel is defined by a position and a color which shall be rendered as a uniformly colored cube. There is no need to work with all four vertices and 36 triangle indices, instead we keep simple lists of position and color for as long as possible. When all modifications are done, we translate our data to a Unity mesh. This is more efficient, as Unity does not expect meshes to be changed frequently. Unity meshes are addressed using 16-bit unsigned integers and thus are limited to 65,536 vertices per mesh. If a voxel object contains more positions, it needs to be split into multiple mesh objects. Ultimately, to create the representation of a voxel, a geometry shader is used to calculate the necessary vertices that make up a cube and assign the corresponding color value. The visual output is rendered using the Oculus Utilities for Unity 5 and displayed on the HMD.

Voxel Representation To a great extent, the visual quality of the voxel space is determined by the voxel resolution (size, number) and by the way the voxel scene is lit and shadowed. The size of each voxel and therefore the resolution of our voxelspace is limited by the resolution of the Kinect data. Figure 4 illustrates the changing area each pixel covers at different distances to the sensor.

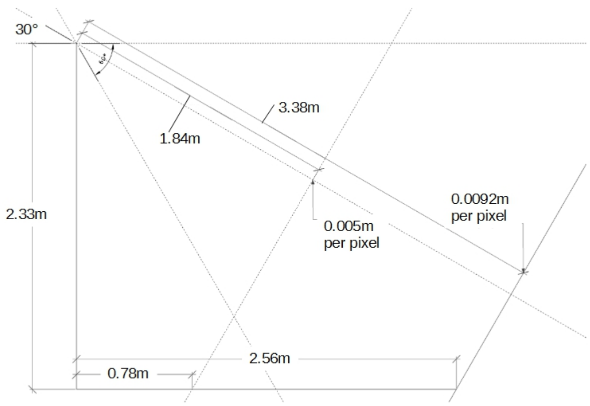

Figure 4: Resolution of Kinect depth data in relation to distance from sensor.

As we need a depth and a color value for each voxel the limiting factor is the resolution of the depth sensor, which is 512 x 424 pixels with a field of view (FoV) of 70 x 60 degrees. This means that one degree in the FoV is covered by a 7x7 pixel area. As the area covered by one degree gets bigger with increasing distance, increasingly more space of the real world needs to be mapped onto those 7x7 pixels. We can calculate the space that is mapped to one pixel at a certain distance. So for a distance greater than 1.84m from the Kinect sensor, each pixel of the depth image covers more than 0.005m. This distance is measured along the view direction of the sensor and equates to a distance of 0.78m from the frame. Details smaller than that cannot be captured by the sensor. In contrast, the space covered by one voxel is consistent over distance.

Voxel Size On one hand, if we choose a voxel size bigger than the precision we loose detail possibly provided by the sensor. On the other hand, by choosing a smaller size there will be gaps between the voxels as there are not enough valid depth values provided at further distances Fig. 5 on the left shows a voxel size of 0.5cm resulting in a very sparse voxel grid, because the sensor cannot provide valid values for each voxel at this distance. But it preserves a more detailed visualization, especially noticeable at the hand and face areas of the user. Considering the given interaction space the user would usually be at this position or even farther away. Consequently, the effect and the gaps get worse with increasing distance. Increasing the voxel size to 0.8cm gives a much better representation at this distance. Thus, we settled on a voxel size of 0.8cm, as this offers a good trade-off between a detailed image and usable interaction space.

Lighting A good approximate to coherent lighting is achieved by estimating the light situation in the real environment and by implementing similar lighting situations within the virtual environment. The most common methods are summarized in the work by Jacobs and Loscos [12]. As we have a relative coarse voxel structure, deviations from ideal coherent lighting are less noticeable than in photorealistic scenes. We use the ambient light from Unity and one directional light source to achieve a lighting situation similar to our real world. As the main light in the real environment is at the ceiling we use one directional light facing down to match that. The position of directional lights in Unity is not relevant. The light sources have an intensity property that needs to be adjusted for the right brightness. Due to other light sources in the real environment like daylight we manipulate the intensity of the ambient light in Unity as well to match the situation as good as possible.

Figure 5: Visual effect of different voxel sizes at interaction distance: Left: voxel size = 0.5cm, Right: voxel size = 0.8cm. Increasing voxel resolution does not necessarily leads to better visual quality.

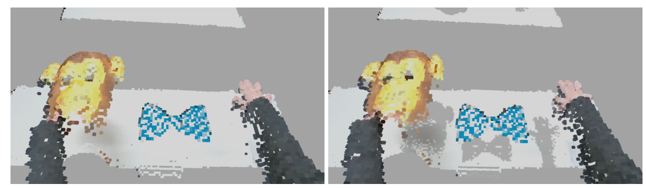

Figure 6: Influence of shadowing between real objects (mask), virtual objects (bowtie), user and environment: Left: shadowing off, Right: shadowing on. Coherent shadows cast by each voxel for all object types help to visually integrate them.

In addition to coherent lighting, a coherent shadow visualization similar to the real world should be provided. Because we have the information of the geometry from both worlds, they can influence each other equally. By using the shadows Unity provides for directional lights in the mixed scene we achieve the same shadows for real and virtual objects as seen in Figure 6. The left images show the scene without virtual shadows. In the upper left picture a soft shadow from the real mask is seen on the real table caused by the light in the real environment. The images on the right show the scene with virtual shadows. As they are casted by each individual voxel of the voxelized objects, the shadows show holes and noise. However, it is more important that the objects do influence each other rather than that the behavior is physically correct. Sugano et al. showed that even if the shadows of virtual objects are behaving contradictory to the real shadows of the scene, the virtual objects felt more present than without any shadow [40]. Shadows can help to visually integrate real and virtual objects in the shared environment.

Virtual Subjects Our current Mixed Voxel Reality system also allows for the recording and replay of voxelized characters. We are able to capture a user inside the interaction space, save the voxels that represent the body as a clip and later replay it inside the voxelspace. This gives us the option to present a user with a recording of him-/herself or of a different person inside a coherent virtual environment. The user can freely move around and watch the volumetric clip in 3D independently of the original recording. Furthermore, apart from just storing position and color data we also record skeleton tracking data provided by the Kinect camera. This information can be used in a variety of ways, some of which have been already implemented in our Mixed Voxel Reality system, while others will be the focus of future work.

4 Coherent Rendering for Voxel Realities

The Mixed Voxel Reality framework's basic functionality described in the preceding section allows for the visualization of anything *real*

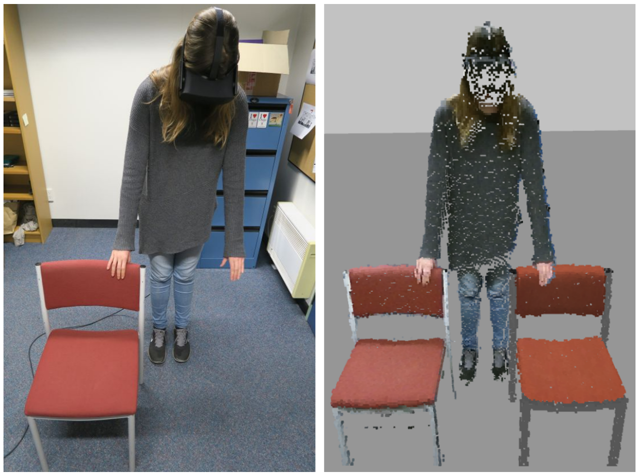

Figure 7: User with two chairs: one virtual and one real

in the environment, including mirroring, scaling, moving, and animated playback of subjects. Additionally, our Mixed Voxel Reality environment should be able to accommodate real and *virtual* elements at the same time and in a visually coherent way. We therefore have to alter the appearance of any virtual elements brought into the mixed environment so that they are indistinguishable from any real elements present. At least, they shouldn't produce a break in coherent experience for the user. To implement a coherent visualization we took inspiration from earlier work on simulating camera imperfections for 2D color cameras [19]. In our work we are approximating the most relevant properties of the RGBD sensor (here a Kinect v2) capturing process in our Mixed Voxel Reality space and apply them to the imported virtual elements (3D objects):

1. Characteristics of the RGBD sensors; resolution, transformation and FoV
2. Voxel representation at a given resolution and size
3. Noise and holes depending on the view of sensor and user
4. colors, affected by the illumination of the real world

To get a matching representation all of these factors have to be reproduced for the virtual objects. We use the following steps to achieve this. First of all we need a 3D Model to start with. This model needs to be converted from a triangulated mesh to a voxel representation. Then the view restrictions of the sensor are applied by culling. If we want to consider the resolution of the sensor and be closer to the way of capturing of the sensor we have to use a raycasting technique. With this we can do voxelization and culling in one step. To achieve a convincing visual coherence, we have to apply temporal noise to the final static voxel model.

4.1 3D modelling

In principle, any virtual object modelled with a 3D modelling program, scanned and reconstructed, or produced otherwise could be imported into the Mixed Voxel Reality scene and appropriately voxelized. However, it helps to understand some basic modelling aspects for the achievement of best results within the system. We therefore describe our way of modelling 3D objects—similar principles apply to other techniques for modelling.

Our models were made with the 3D creation software Blender[1]. A real chair in our office was used as a ground truth for comparing

[1] www.blender.org

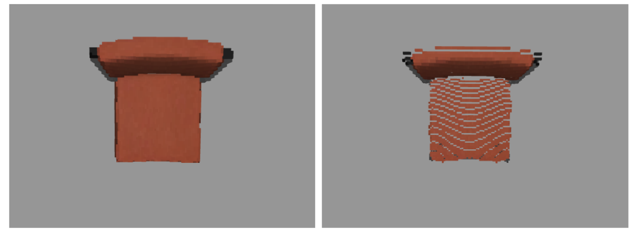

Figure 8: Left: Rendering by voxelization only; Right: Rendering after Raycasting

virtual and real models in terms of rendering quality, scale, etc. (see Fig 7). A most crucial part is to texturize the model in a way so that the colors are recreated in a quality manner. colors are influenced by many factors:

1. Material of the object
2. Lighting situation in the surrounding environment
3. Capturing sensor
4. Way of storing the color values
5. Displaying medium
6. Eye of the observer

To get a similar color impression between the virtual and real models we texturized the model with a texture captured by the Kinect color sensor in the usual lighting situation. For our system it is not needed to achieve exactly the same color impression—two objects don't have to look the same but they should look as being in the same environment.

4.2 Voxelization

Raycasting is used for the simulation of the capturing by a virtual camera. This method is the closest to the way the Kinect sensor works and the real data are generated. For each pixel of the sensor a ray is cast into the scene and the position and color of the first intersection of the ray with the mesh is stored. This takes all of the camera characteristics into account. The transformation of the sensor and the FoV determine the rays that are cast. In that way, no geometry outside of the view frustum is checked. Additionally, as only the first position and color seen by the sensor is stored we do not have to handle occlusion separately.

In Fig.8 the result of the raycasting is compared to the result of a simple voxelization step. In particular the stripes of missing voxels in the ray casting result are worth noting. When capturing real objects those stripes are present more or less predominantly depending on the distance, orientation, and resolution of the sensor. With our raycasting approach we are performing voxelization, culling, and occlusion handling in one effective step.

4.3 Noise

Up until here, the rendering of the virtual objects take into account the voxel representation in the given resolution, the FoV, the resolution and position of the Kinect sensor and the lighting of the real scene. However, this rendering remains temporally static—there is a very noticeable difference between the virtual and real objects as the real objects change their appearance over time due to the presence of noise of the depth sensor. The raw data is used without any processing. Most applications using depth data use smoothing filters, for example an edge preserving bilateral filter. As we do not want to improve the data to achieve a higher fidelity, we do not use any filter to keep the processing of the real data as realistic as possible. Consequently we have to consider the characteristics of time-of-flight (TOF) sensors in general, and of the Kinect v2 specifically. What factors are influencing the noise and how can this be modeled and transferred to the virtual objects?

A so called metrological characterization for the TOF sensor of the Kinect v2 is provided by Corti et al. [5] and Gonzales-Jorge et al. [8] who compared the v1 and the v2 sensors. They analyzed the random and systematic components of uncertainty in the measurements of the depth sensor and identified the following parameters influencing the noise present in the depth image of the Kinect v2 depth sensor:

1. Distance of the captured object to the sensor
2. Angle between the camera and the captured surface
3. Distance of a pixel in the depth image to the central pixel
4. Reflectivity of the captured material
5. Additive noise present in all TOF sensors but not specified yet
6. Wiggling error, also a specific TOF characteristic
7. Mixing pixels effect shown between objects

These factors were found by testing the behaviour of the Kinect depth sensor in different situations. The resulting depth values follow a Gaussian distribution with different standard deviations, depending on the mentioned parameters. Belhedi et al. [3] introduced a model for the noise of TOF sensors. They used a 3D thin-plate-spline function to get the standard deviation for each distance and pixel position. They also state that the noise for each pixel follows a Gaussian distribution. We used a simpler function to generate noise, as most of the effects just cause errors of a few millimeters. This would not make any difference for our system, because in the voxelization step they are mapped to the same values nonetheless. More noticeable is the mixing pixel effect, which is introduced by interpolating between depth values in a preprocessing step included in the Kinect sensor.

4.4 Render results

Due to the fact, that the capturing and the rendering camera for the real object differ there is the need for rendering the virtual objects in a way that adapts the static real world camera. The best way to implement the effects of the sensor on the geometry is a raycasting technique. Fig.9 shows the result of a straight forward surface voxelization combined with culling methods and the result of the raycasting each next to the real chair. The virtual objects in our current system are either positioned statically or not moving a lot, so the computational costs of raycasting are minimal. Our technique is close to the function of the Kinect sensor and reproduces appealing artifacts moderated by the resolution and the view of the camera.

5 Evaluation

Technically, our Mixed Voxel Reality system can serve as a platform for further developments and to study presence and embodiment. The developers and other fellow researchers and students in the lab are able use and control the system. But, can we show that the system achieves the desired performance and will it support the development of our targeted dimensions of interest presence and embodiment? Our Mixed Voxel Reality system was evaluated technically and empirically. Our evaluations prove the effectiveness of our system to provide an interactive, believable, embodied mixed reality experience.

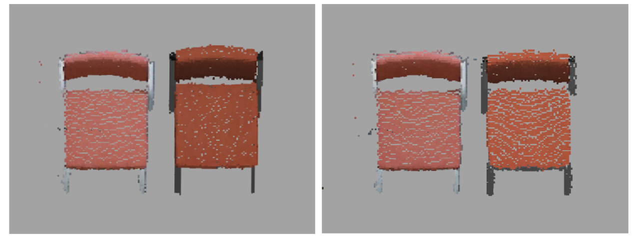

Figure 9: Comparison of a real voxelized (left) and raycasted chair (right), side-by-side real and virtual (the real chair has lighter colored legs than the virtual one for illustration purposes)

5.1 Technical Evaluation

We tested our Mixed Voxel Reality system on a standard MS Windows PC (quadcore i7 6700, 3.4 GHz, with 16GB of RAM) equipped with a mid-of-the-range graphics board (nVidia GTX 970 with 4GB of RAM). We are interested in how the system performs with an increasing number of voxels simultaneously processed and shown in the environment. The test scene we worked with during development consisted of one person, a table, and some objects on the table—everything also shown in a virtual mirror placed within the environment. Such a scene is composed of less than 100,000 voxels (2x50,000). During development and demonstrations within our lab we perceived the performance as fast and almost latency free. But, how scalable is our system? When do we reach limitations of the computing system (memory, CPU, GPU) and when do we reach the limits of interactivity. We would consider the system as real-time interactive if a frame update rate of 60 Hz or more is achieved at an end-to-end latency of less than 100ms.

To measure the update rate of the system we implemented a callback function with a timer and displayed the time between update cycles and the respective update rate on screen. Memory and GPU usage have been measured with HWMonitor (www.cpuid.com/softwares/hwmonitor.html) and the CPU usage with the Windows-built-in task manager.

The most difficult part was to measure the latency of the system. Because we are using not only visual information (RGB sensor), but also depth information and turn both combined into our dynamic voxel representations we tried to find a solution in the related work of VR and AR systems. One of the earliest work in measuring latency in VR uses a pendulum to which a tracking sensor (in that case Polhemus Isotrak) is attached. The resulting (virtual) movement of the tracker is visualized on a screen. A camera captures both the real tracking sensor at the pendulum as well as the virtual representation in the same field of view. The angular difference between the real and virtual sensor is used to calculate the resulting latency [20]. Steed used a similar setup, but measured frequencies / phase differences instead of angles and with this could even improve on the quality of the measures [38]. Because of the absence of actual tracking sensors in our system this configuration is hard to mimic. Olano et al. and Jacobs et al. used a pulsed light source, captured it with a camera and measured its representation on the monitor screen with a photo sensor [26, 13]. Using an oscilloscope they could determine the time difference representing the delay. Similarily Sielhorst et al. [33] made technical modifications to a VST AR system to measure latency. They encode the time in the image and decode the time after camera feedback. Histogram visualization is used to determine latency, amongst other factors. Difficulties for the Mixed Voxel Reality system arise from the requirement of having a light source and a photo sensor within the setup. While the light time difference measurement would allow us to determine one part of the overall latency, it would not allow for including the depth component of the sensor and with this

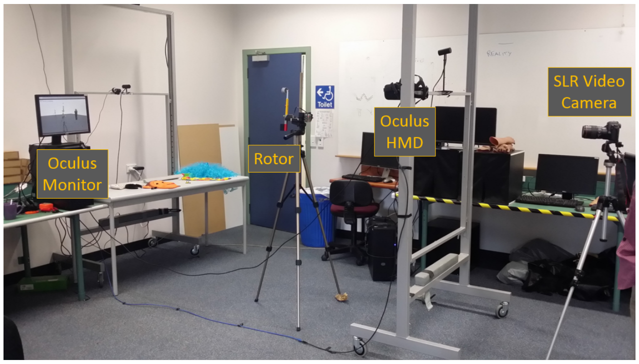

Figure 10: Setup of latency measurement: SLR Video Camera is capturing the Oculus Monitor *and* the Rotor; the Oculus Monitor presents what is seen by the Oculus HMD looking at the Rotor.

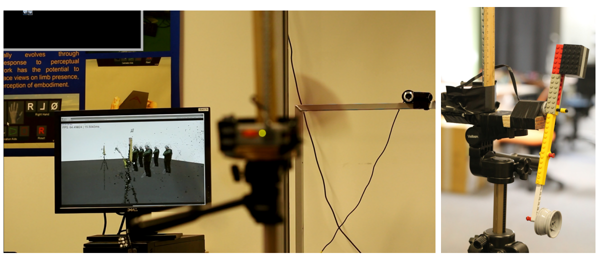

Figure 11: Left: Example frame taken from video stream recorded by SLR Video Camera showing both the Oculus Monitor and the Rotor in action; Right: Photo of the Rotor build for the latency measurement rotating with about 360 revs/min.

the actual movements of objects or body parts in the scene. This makes also approaches like IR-LED arrays or checkerboards unsuitable [4, 21]. Besides other measures within the overall system lag flow, Swindells et al. used a modified phonograph turntable in combination with a half-silvered mirror in a desktop monitor setup to measure the angular difference between a real and a virtual disc (recorded by a camera) [41]. Apart from the very specialised type of movement pattern, the RGBD sensor would hardly pick-up the depth component (but certainly the visual component).

What we need is a visually and spatially distinctive element in the real scene which is captured with the RGB *and* the D components of the Kinect sensor and which is big enough to end up processed and rendered as one or more voxel within the Mixed Voxel Reality environment. We opted for a configuration with a slow (enough) moving rotor with an end-object attached to it which can be clearly tracked by the sensor and system. The rotor is facing the Kinect sensor and is rotating with about 360 revs/minute. An SLR video camera on a tripod is capturing the rotor and the resulting visualization on screen at the same time from behind. Fig 10 illustrates the latency estimation setup.

From the video stream individual images can be extracted which show the physical rotor and its voxel representation at the same time. Those images give two estimates: (1) the blurred image part from the beginning to the end of the angular movement gives the actual revs/min speed (with the known shutter speed) and (2) the difference between last angle of this segment and the angle of the completely voxelized virtual rotor arm gives the latency (considering the revs/min speed). Fig 11 shows the rotor used (right) and an example frame taken from the video.

To incrementally increase the number of voxels in the scene in a realistic way we introduced pre-recorded, animated voxel avatars to the scene one-by-one. Each of those voxel avatars including mir-

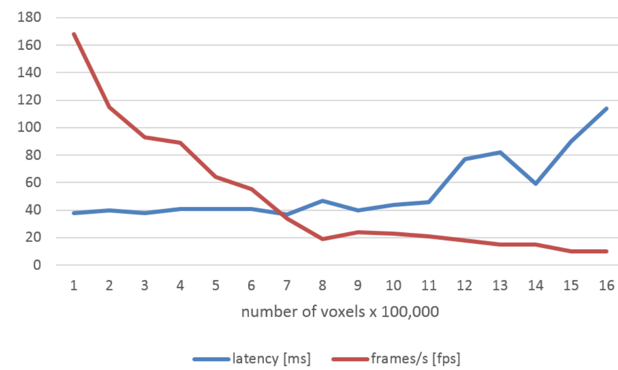

Figure 12: Framerate and latency developing with increasing numbers of voxels present in scene; measured at 100,000 voxels intervals.

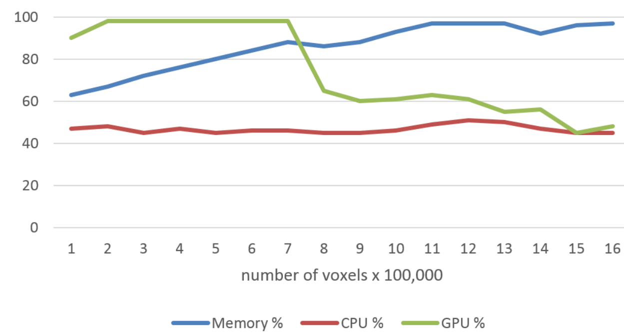

Figure 13: System behaviour under increasing voxel number load; i7 6700, 3.4 GHz, 16GB RAM, GTX970, 4GB

roring consists of about 100,000 voxels. We added voxel avatars in those 100,000 voxel increments. After each increment we measured the frame update rate, the latency, memory usage (in per cent of maximum), CPU and GPU usage. Fig 12 combines in one illustration the increasing latency and the falling frame rate as the number of voxels in the scene increases. While there is some variability in the precision of our measurement it can be seen that about 500,000 voxels can be rendered in interactive real-time (60Hz) with a latency of about 40ms, which is well within our given limits. What also can be seen is that at about 800,000 voxels the frame rate drops below 20 Hz, which makes interaction more difficult, but still possible given the corresponding latency of about 50 ms.

While the computer system resources have been used at almsot maximum capacity they still stayed just within boundaries and allowed for rendering of up to 1.6 million voxels. See Fig 13 for an illustration of this effect.

Overall, we have been very satisfied with the results of the technical evaluation. The number of voxels we can render simultaneously in interactive real-time on a standard PC is certainly high enough for our envisaged application scenarios. Those would require less than 0.5 million voxels and therefore can be implemented using our current system.

5.2 User Study

Technically, our Mixed Voxel Reality system is able to deliver a real-time interactive performance. The scenario used for the technical test now serves as a basis for evaluating the actual user experience. We developed a demonstration and test application which uses the same elements, e.g. a real table, real and virtual elements

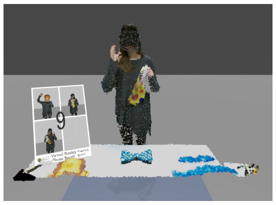

Figure 14: User's view of the Mixed Reality Photo Booth demonstration application, including a mirror image of the user, table, real and virtual props, and a photo preview with counter.

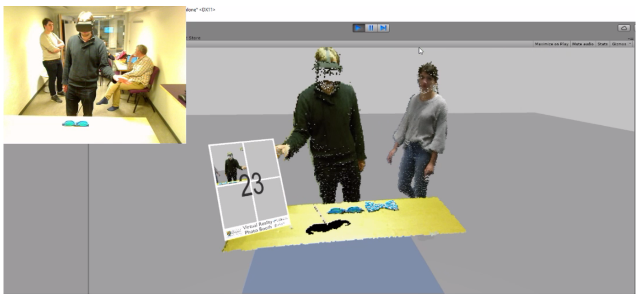

Figure 15: Still image of recording of a session including real world camera view (inset top left) and virtual character in scene.

on the table, and a virtual mirror, but embeds and extends this environment in a way that it can (a) be used for demonstration purposes and (b) forms the basis for an empirical laboratory study. We first describe the demonstration application and then our lab study with 22 participants.

5.2.1 Demonstration and Test Application

To test our system's practical feasibility we developed a demonstration application called the *Mixed Reality Photo Booth*. Much like a photo booth normally found at railway stations, shopping malls, etc. users enter a closed box environment where four consecutive pictures are taken—usually just for the sake of fun. Instead of taking photos of the person in front of a camera our application takes photos of the users seeing themselves in the virtual environment (Fig 14).

The users are wearing head-mounted displays (i.e. the face isn't visible) and can use props provided on a table in front of them. Inherently, the Mixed Voxel Reality system captures all real objects as well as the user within the defined voxelspace in front of the Kinect sensor. In addition to the actual objects present we added virtual props and occasionally appearing recorded virtual characters to the scene. We prepared ten predefined scenes with different characters (authors and colleagues, also sometimes wearing a Gorilla costume). Added to the scene is a preview of the to be shot photos of the scene including a countdown timer until the next photo will be taken. A real, external web camera was added to capture the real environment for post-hoc evaluation purposes (Fig 15).

Before starting the scene the users agreed or declined to have real camera and virtual camera recordings stored for research purposes. At the end of the sessions, which lasted for usually five minutes, the four photos have been printed on a 4x6 color printer for the users to take home (Fig 16).

The Mixed Reality Photo Booth was demonstrated to about fifty people in Dunedin, New Zealand as part of a public two-day science festival and to about three more dozens of people at locations in Koblenz, Germany and Oslo, Norway. Because the term Mixed Reality is not commonly used by the general public we called the application *Virtual Reality Photo Booth*.

Figure 16: Example of photo a user would take home after experiencing the MR Photo Booth demo application.

The photo booth demonstration application was very well received. Apparently, the implementation quality is of a high enough standard to be usable and acceptable. We did not perform any formative evaluations of the system during our demonstrations. Where consent was given we recorded the users in the scene (real and simultaneously virtual) including audio tracks. Those recordings informed design decisions on modifications of the Mixed Voxel Reality system and we used them to design our laboratory study.

5.2.2 Laboratory Study

While our Photo Booth application demonstrated the general usability and acceptance of the Mixed Voxel Reality system it does not serve as a proof for our targeted dimensions presence and embodiment. We therefore designed a laboratory experiment which is based on the photo booth application and measures the users' sense of presence and feeling of embodiment subjectively. The photo booth application was simplified to one scene only (instead of ten) where four real objects and two virtual objects lay on the real table in front of the participants. The participants saw themselves in the virtual mirror placed about two metres away from them. For each of the four photos taken a virtual character was entering the scene next to the position of the participants. Those virtual characters have been recorded at Bauhaus University's Virtual Reality Systems lab using a four Kinect recording setup [2]. The Mixed Voxel Reality system was set up in a separate room in our lab environment. Twenty-two participants (15 male, 7 female, avg age=25.91, S.D.=10.23) have been recruited from university's staff and students. They have been greeted outside the Mixed Voxel Reality room, read an information sheet and signed a consent form, according to the university's ethics guidelines. They put on the head-mounted display outside of the Mixed Voxel Reality room, i.e. they did not see the real environment before entering the photo booth scene. With the HMD put on the participants were guided into the room to stand in front of the table and were asked to simply do the VR photo booth task, like in the demonstration application. After completing the task they filled in a combined questionnaire containing: items from the igroup presence questionnaire [32], items on body ownership from [1], items on body ownership from [22], items from the Mixed Reality Experience Questionnaire [28], and the simulator sickness questionnaire [17]. After completion the participants were given grocery vouchers as a token of appreciation for their time.

The igroup presence questionnaire (IPQ) [32] is an instrument designed to measure a person's sense of presence in a virtual environment. It consists of the subscales spatial presence, involvement, and realism—all of which are also relevant for mixed reality environments. The IPQ is a standard questionnaire to measure presence and was successfully applied in hundreds of studies. We left out the items which are only applicable to purely virtual environments and applied eight (of 14) items of the IPQ.

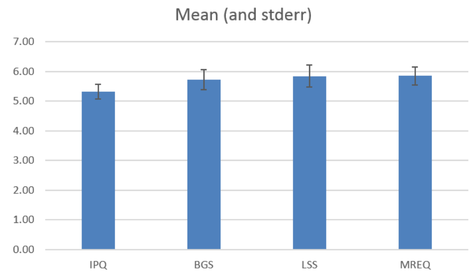

Figure 17: Means and standard errors for four presence and embodiment questionnaires (from left to right): IPQ - igroup presence questionnaire [32], BGS - body ownership [1], LSS - body ownership [22], MREQ - Mixed Reality Experience Questionnaire [28]

.

Research on measuring presence in augmented or mixed reality environments is still in its infancy. In addition to the application of the IPQ we administered a sub-set of items from the Mixed Reality Experience Questionnaire (MREQ) [28]. Two items target the relationship between the user and the virtual objects, one the relationship between the virtual and the real objects and one the relationship between the virtual objects.

Our second dimension of interest, the feeling of embodiment, was targeted with items from two different works: First, we took four items from [1] targeting questions about the body, mirroring, visual body features, and the feeling of having two bodies. Second, we took two items from [22], one targeting ownership and the other agency.

All questions used the originally proposed Likert-like scales (7-point and 10-point, respectively), including their original labeling and anchors. For analysis purposes the 10-point scales have been converted later to 7-point scales.

All 22 participants completed the task and nobody withdrew from the experiment. Some of the participants reported simulator sickness symptoms as measured by the scoring system described in [16]: one participant with "symptoms are a concern" and one with "significant symptoms". All other twenty participants reported minimal, negligible, or no symptoms at all. On one hand, viewing this in a positive light, it means that only 10 per cent of the participants had considerable simulator sickness symptoms and even the two who had, continued with the experiment voluntarily. On the other hand, two participants with symptoms is two too many—we should strive for zero simulator sickness with MR. Possible sources for sickness symptoms could be: (1) participants did not spend enough time to adjust the HMD properly (and should have been encouraged more to do so), (2) an end-to-end latency of around 40ms might be still too high, (3) the overall tracking robustness is not high enough (even if we already used two Oculus tracking cameras during the study), or (4) individual differences amongst participants lead to symptoms, regardless of the immersion quality.

Figure 17 shows the averages scored on each of the four other questionnaires. Presence and embodiment are reported to be very high. All four questionnaires' means are significantly above mid-point as tested with a one-tailed t-Test assuming unequal variances (df=21). With a $t_{critical}$ of 1.721 all t_{stat} are higher ($p<0.05$) than $t_{critical}$ (IPQ: 10.3, BGS: 4.33, LSS: 7.77, MREQ: 9.25) and therefore the means are significantly higher than mid-point (4.0). With those results we can show that our Mixed Voxel Reality system is able to achieve a sense of presence and embodiment. The users developed a spatial sense of the mixed environment, developed agency with their virtual body counterparts, and perceived ownership of those voxel bodies. The high ratings for the scale items suggest that presence and embodiment can be robustly achieved with even this kind of low fidelity system as long as visual coherence is maintained. While this finding does not necessarily mean that there is no need for higher fidelity MR systems, it shows, that low (computational) cost mixed reality is possible.

There were no indicators during the study which would suggest that the system experience is not believable or is unacceptable. On the contrary, when asked about their experience, participants reported about their positive and enjoyable (short) time in the environment.

6 Discussion and Future Work

We conceptually developed, implemented, and tested our Mixed Voxel Reality system as an enabling, low-cost, low-fidelity, coherent Mixed Reality platform to study (and design for) presence and embodiment. We could demonstrate the effectiveness of our Mixed Voxel Reality approach with technical and empirical evaluations. While we intentionally did not compare them, we would anticipate that other RGBD sensors and other methods for voxel generation, processing, and rendering could be used leading to similar results, perhaps even point-cloud based methods without an explicit spatial data structure (cf [14]). Also, we did not compare the fidelity of our results against other approaches for the same reason. We want to initiate a discussion on whether higher fidelity is always needed or, as in our presented case, that less can be more. However, this does not mean that we simply stop here and do not try to increase fidelity aspects of such a system. For instance, the use of multiple RGBD sensors to acquire a more comprehensive model of 3D objects, the environment, and people is desirable and subject to our future work. Akin to other 3D telepresence systems using depth sensors, e.g. [27] and [2], we are going to include a number of Kinect sensors not only for capturing and later replay but also for real-time communication and collaboration. On the basis of our current development we are going to implement a range of prototypical applications targeting mainly three different scenarios: (1) body perception studies, (2) telepresence, and (3) human behaviour studies in manufacturing processes.

We can't foresee all possible application scenarios for Mixed Voxel Reality and therefore invite other researchers and practitioners to build their own applications. Our Mixed Voxel Reality development environment is provided as open-source and can be accessed through www.hci.otago.ac.nz/research.html. It allows for the realisation of affordable mixed voxel realities, essentially requiring a mid-range computing system and an MS Kinect sensor. In particular for applications where presence and embodiment are important, but budgets are limited, Mixed Voxel Reality provides a suitable platform.

The authors wish to thank Noel Park, Jacob Young, Thomas Schubert, Stefan Mueller, Chris Edwards, Michael Wagner and the Otago HCI group. Parts of this research have been financially supported by a University of Otago Research Grant. The study was approved by the university's Ethics Committee (D17/047).

References

[1] D. Banakou, R. Groten, and M. Slater. Illusory ownership of a virtual child body causes overestimation of object sizes and implicit attitude changes. *Proceedings of the National Academy of Sciences*, 110(31):12846–12851, 2013.

[2] S. Beck, A. Kunert, A. Kulik, and B. Froehlich. Immersive group-to-group telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 19(4):616–625, 2013.

[3] A. Belhedi, A. Bartoli, S. Bourgeois, K. Hamrouni, P. Sayd, and V. Gay-Bellile. Noise modelling and uncertainty propagation for tof sensors. In *European Conference on Computer Vision*, pages 476–485. Springer, 2012.

[4] M. Billeter, G. Röthlin, J. Wezel, D. Iwai, and A. Grundhöfer. A led-based ir/rgb end-to-end latency measurement device. In *Mixed*

and Augmented Reality (ISMAR-Adjunct), 2016 IEEE International Symposium on, pages 184–188. IEEE, 2016.

[5] A. Corti, S. Giancola, G. Mainetti, and R. Sala. A metrological characterization of the Kinect V2 time-of-flight camera. *Robotics and Autonomous Systems*, 75:584–594, jan 2016.

[6] M. Dou, H. Fuchs, and J. Frahm. Scanning and tracking dynamic objects with commodity depth cameras. In *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013, Adelaide, Australia, October 1-4, 2013*, pages 99–106. IEEE Computer Society, 2013.

[7] J. Fischer, D. Bartz, and W. Straber. Stylized augmented reality for improved immersion. In *Virtual Reality, 2005. Proceedings. VR 2005. IEEE*, pages 195–202. IEEE, 2005.

[8] H. Gonzalez-Jorge, P. Rodríguez-Gonzálvez, J. Martínez-Sánchez, D. González-Aguilera, P. Arias, M. Gesto, and L. Díaz-Vilariño. Metrological comparison between Kinect I and Kinect II sensors. *Measurement*, 70:21–26, jun 2015.

[9] T. Ha, S. Feiner, and W. Woo. Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 219–228, Sept 2014.

[10] S. Hauswiesner, M. Straka, and G. Reitmayr. Image-based clothes transfer. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 169–172, Oct 2011.

[11] C.-C. Ho and K. F. MacDorman. Measuring the uncanny valley effect. *International Journal of Social Robotics*, pages 1–11, 2016.

[12] K. Jacobs and C. Loscos. Classification of illumination methods for mixed reality. *Computer Graphics Forum*, 25(1):29–51, 2006.

[13] M. C. Jacobs, M. A. Livingston, et al. Managing latency in complex augmented reality systems. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 49–ff. ACM, 1997.

[14] S. Kahn. Reducing the gap between augmented reality and 3d modeling with real-time depth imaging. *Virtual Reality*, 17(2):111–123, 2013.

[15] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg. Omnikinect: Real-time dense volumetric data acquisition and applications. In *Proceedings of the 18th ACM Symposium on Virtual Reality Software and Technology*, VRST '12, pages 25–32, New York, NY, USA, 2012. ACM.

[16] R. S. Kennedy, J. M. Drexler, D. E. Compton, K. M. Stanney, D. S. Lanham, and D. L. Harm. Configural scoring of simulator sickness, cybersickness and space adaptation syndrome: Similarities and differences. *Virtual and adaptive environments: Applications, implications, and human performance issues*, page 247, 2003.

[17] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The international journal of aviation psychology*, 3(3):203–220, 1993.

[18] K. Kilteni, R. Groten, and M. Slater. The sense of embodiment in virtual reality. *Presence: Teleoperators and Virtual Environments*, 21(4):373–387, 2012.

[19] G. Klein and D. Murray. Compositing for small cameras. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 57–60. IEEE Computer Society, 2008.

[20] J. Liang, C. Shaw, and M. Green. On temporal-spatial realism in the virtual reality environment. In *Proceedings of the 4th annual ACM symposium on User interface software and technology*, pages 19–25. ACM, 1991.

[21] P. Lincoln, A. Blate, M. Singh, T. Whitted, A. State, A. Lastra, and H. Fuchs. From motion to photons in 80 microseconds: Towards minimal latency for virtual and augmented reality. *IEEE transactions on visualization and computer graphics*, 22(4):1367–1376, 2016.

[22] J. Llobera, M. V. Sanchez-Vives, and M. Slater. The relationship between virtual body ownership and temperature sensitivity. *Journal of the Royal Society Interface*, 10(85):20130300, 2013.

[23] J.-L. Lugrin, M. Landeck, and M. E. Latoschik. Avatar embodiment realism and virtual fitness training. In *Virtual Reality (VR), 2015 IEEE*, pages 225–226. IEEE, 2015.

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

[25] S. Mock, P. Lensing, and W. Broll. Achieving flexible 3d reconstruction volumes for rgb-d and rgb camera based approaches. In *International Conference on Computer Vision and Graphics*, pages 221–232. Springer, 2016.

[26] M. Olano, J. Cohen, M. Mine, and G. Bishop. Combatting rendering latency. In *Proceedings of the 1995 symposium on Interactive 3D graphics*, pages 19–ff. ACM, 1995.

[27] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A. Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, pages 741–754. ACM, 2016.

[28] H. Regenbrecht, C. Botella, R. Baños, and T. Schubert. Mixed reality experience questionnaire (MREQ)—reference. Discussion paper 2017/01, Department of Information Science, University of Otago, Dunedin, New Zealand, Feb. 2017.

[29] H. Regenbrecht, S. Hoermann, C. Ott, L. Muller, and E. Franz. Manipulating the experience of reality for rehabilitation applications. *Proceedings of the IEEE*, 102(2):170–184, 2014.

[30] H. Regenbrecht and S. Jacobsen. Augmentation of volumetric data in an airplane cabin. In *Demonstration at IEEE and ACM International Symposium on Mixed and Augmented Reality, Darmstadt*, 2002.

[31] H. Regenbrecht and T. Schubert. Real and illusory interactions enhance presence in virtual environments. *Presence: Teleoperators and virtual environments*, 11(4):425–434, 2002.

[32] T. Schubert, F. Friedmann, and H. Regenbrecht. The experience of presence: Factor analytic insights. *Presence: Teleoperators and virtual environments*, 10(3):266–281, 2001.

[33] T. Sielhorst, W. Sa, A. Khamene, F. Sauer, and N. Navab. Measurement of absolute latency for video see through augmented reality. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 1–4. IEEE Computer Society, 2007.

[34] K. H. Sing and W. Xie. Garden: A mixed reality experience combining virtual reality and 3d reconstruction. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '16, pages 180–183, New York, NY, USA, 2016. ACM.

[35] M. Slater, P. Khanna, J. Mortensen, and I. Yu. Visual realism enhances realistic response in an immersive virtual environment. *IEEE computer graphics and applications*, 29(3), 2009.

[36] M. Slater, D. Pérez Marcos, H. Ehrsson, and M. V. Sanchez-Vives. Inducing illusory ownership of a virtual body. *Frontiers in neuroscience*, 3:29, 2009.

[37] B. Spanlang, J.-M. Normand, D. Borland, K. Kilteni, E. Giannopoulos, A. Pomés, M. González-Franco, D. Perez-Marcos, J. Arroyo-Palacios, X. N. Muncunill, et al. How to build an embodiment lab: achieving body representation illusions in virtual reality. *Frontiers in Robotics and AI*, 1:9, 2014.

[38] A. Steed. A simple method for estimating the latency of interactive, real-time graphics simulations. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 123–129. ACM, 2008.

[39] M. Straka, S. Hauswiesner, M. Rüther, and H. Bischof. *A Free-Viewpoint Virtual Mirror with Marker-Less User Interaction*, pages 635–645. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[40] N. Sugano, H. Kato, and K. Tachibana. The effects of shadow representation of virtual objects in augmented reality. *Proceedings - 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2003*, pages 76–83, 2003.

[41] C. Swindells, J. C. Dill, and K. S. Booth. System lag tests for augmented and virtual environments. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 161–170. ACM, 2000.

[42] P. Weir, C. Sandor, M. Swoboda, T. Nguyen, U. Eck, G. Reitmayr, and A. Dey. Burnar: Feel the heat. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 331–332, Nov 2012.