frontiers in ICT

ORIGINAL RESEARCH
published: 24 April 2019
doi: 10.3389/fict.2019.00007

Preaching Voxels: An Alternative Approach to Mixed Reality

Holger Regenbrecht[1]*, *Jung-Woo (Noel) Park*[1], *Claudia Ott*[1,2], *Steven Mills*[1,2], *Matthew Cook*[1] and *Tobias Langlotz*[1]

[1] Human-Computer Interaction Lab, Information Science, University of Otago, Dunedin, New Zealand, [2] Visual Computing Lab, Computer Science, University of Otago, Dunedin, New Zealand

For mixed reality applications, where reality and virtual reality are spatially merged and aligned in interactive real-time, we propose a pure voxel representation as a rendering and interaction method of choice. We show that voxels—gap-less volumetric pixels in a regular grid in space—allow for an actual user experience of a mixed reality environment, for a seamless blending of virtual and real as well as for a sense of presence and co-presence in such an environment. If everything is based on voxels, even if coarse, visual coherence is achieved inherently. We argue the case for voxels by (1) conceptually defining and illustrating voxel-based mixed reality, (2) describing the computational feasibility, (3) presenting a fully functioning, low resolution prototype, (4) empirically exploring the user experience, and finally (5) discussing current work and future directions for voxel-based mixed reality. This work is not the first that utilizes voxels for mixed reality, but is the first that uses voxels for all internal, external, and user interface representations as an effective way of experiencing and interacting with mixed reality environments.

Keywords: virtual reality, augmented reality, presence, visual coherence, interaction, voxels, mixed reality, user study

OPEN ACCESS

Edited by:
Daniel Thalmann,
École Polytechnique Fédérale de Lausanne, Switzerland

Reviewed by:
Selim Balcisoy,
Sabanci University, Turkey
Soraia Raupp Musse,
Pontifícia Universidade Católica do Rio Grande do Sul, Brazil

***Correspondence:**
Holger Regenbrecht
holger.regenbrecht@otago.ac.nz

Specialty section:
This article was submitted to Virtual Environments, a section of the journal Frontiers in ICT

Received: *07 November 2018*
Accepted: *03 April 2019*
Published: *24 April 2019*

Citation:
Regenbrecht H, Park J-W, Ott C, Mills S, Cook M and Langlotz T (2019) Preaching Voxels: An Alternative Approach to Mixed Reality. Front. ICT 6:7. doi: 10.3389/fict.2019.00007

1. INTRODUCTION

Imagine a mixed reality experience composed of voxels, and voxels only. The surrounding environment, the objects, and the people in such an environment are composed of voxels, regardless of whether they are real or virtual. Such a mixed reality environment would be inherently coherent in its visual appearance, would benefit from naturally occurring occlusions, collisions and interactions, and would provide a system of computational and engineering simplicity and elegance. A pure voxel-based approach would benefit from decades of research and development in 2D techniques, like image storage and compression, and transfer this into the 3D world of voxels. Also, it would benefit from existing voxel research. A voxel-based *Mixed Reality* (MR) experience would be like walking through a volumetric pixel world.

While the concept of voxels is not new and has been widely used in simulations and for other computational purposes, the actual real-time experience of being immersed in an interactive voxel space is still in its infancy. In this paper, as the title suggests, we want to make a case for using voxels in MR systems and will demonstrate that, even with today's achievable low-fidelity representations, voxels can be an effective and efficient way to deliver MR experiences. We highlight the current possibilities and the future potential of voxels for mixed reality even if today's visual realism is inferior to other approaches like triangulated rendering. Furthermore, we argue that the future of MR environments will be shaped by voxels.

Our research and development is targeting application scenarios in the areas of telepresence and collaboration, entertainment, education and training, and evaluating users' behavior in Mixed Reality environments. To serve those and other scenarios we have to be able to capture, voxelize, store, process, transmit, and display users and objects from multiple viewpoints and in interactive real-time.

Voxel-based mixed reality uses voxels, and only voxels, to visually represent spatially merged and aligned real and virtual objects, subjects, and the environment in interactive real-time. Normally, an exocentric voxelization technique will be used to represent the visible shapes of reality in a voxel-based MR system, e.g., by using external RGB-D cameras to capture a scene.

Voxels are volumetric pixels, which are spaced in a regular grid in three-dimensional space and are perceived without gaps between them. In contrast to the similar concept of point clouds, voxels can only inhabit discrete positions in space dictated by the grid. The grid is regular, i.e., throughout the entire MR volume, and all possible voxel positions are equally spaced. Voxels can be of any shape, but have to be all of the same shape and, if placed next to each other in space, should give the viewer a gap-less impression. Cube shapes lend themselves to be used, but other shapes can be used as well as long as the gap-less perception is maintained.

Our voxel-based approach can be seen as the next logical step after the highly developed, two-dimensional pixel rendering techniques that are omnipresent today. We are interested in how a 2D (interactive) video experience can be turned into a three-dimensional, interactive, mixed reality experience, even with today's technical possibilities. We focus on the real-time aspect of delivering such a volumetric pixel experience where we trade 3D experience for resolution.

Everything in a voxel-based MR environment is represented by voxels to achieve visual coherence. Real objects, real people, and the surrounding environment are voxelized before being presented in real-time or recorded in the mixed environment. Even if voxels allow for actual volumetric data representation, i.e., solids and voids, normally only the visual hull of reality will be captured and then turned into voxels. One or more capturing devices (e.g., depth cameras) are used to reconstruct the real environment or individual objects, and then captured surface elements (points, meshes) are converted into voxels while maintaining the captured colors at the respective positions. Usually this process is performed from exocentric viewpoints, i.e., independent of the current viewing position and orientation of the viewer. Users can then virtually walk through the space of voxels constructed from exocentric views, as for instance recently demonstrated with a meshed environment by Lindlbauer and Wilson (2018).

In addition, all purely virtual elements are voxelized in a similar way. Normally this means that the visual resolution of the object will be intentionally degraded to achieve visual coherence. 3D models, like *Computer Aided Design* (CAD) objects, are brought to real scale and are then voxelized for later integration into the mixed environment, as shown in **Figures 1**, **2**. All mixed reality real-time, recorded, and virtual elements are merged and spatially aligned into one space with the aim of being experienced as one coherent MR space.

We will look into the related work around voxel representations; show that voxels are mathematically and computationally efficient also in the context of MR applications; explain why voxels can be very effective for voxel-based MR; present our working prototype solution for a voxel-based MR system called *Mixed Voxel Reality* (MVR); use the MVR system to explore presence and co-presence of low resolution mixed reality scenes (**Figure 1**); and finally discuss ramifications, application scenarios, and areas of current and future work.

2. RELATED WORK

Our assertion is that voxels are suitable for a wide range of MR applications. The literature supporting this falls into three main categories. Firstly, there is the support for voxels both as an efficient representation for spatial reasoning and as a basis for solutions to a number of tasks related to MR applications. This supports our assertion that voxels are ready for widespread use as the basis for MR applications. Secondly, there is a range of MR research that uses non-photorealistic modeling techniques. This literature shows that photorealism is not essential to MR experiences, and so the fact that we cannot (yet) deploy voxel models with fidelity approaching reality is not a fundamental impediment to their immediate use. Finally, we review existing applications of voxels in MR systems, illustrating the range of techniques that are already supported by this representation.

2.1. Voxel-Based Representations

Voxels have a long history of use in visual computing, with both regular voxel grids (Cleary and Wyvill, 1988) and octree structures (Meagher, 1982) having been used to accelerate graphics computations for several decades. Other tree-based subdivision methods such as *k*-D trees (Bentley, 1975) and BSP trees (Fuchs et al., 1980) have a similarly long history. While point- and mesh-based graphics have become dominant, supported by hardware acceleration on GPUs, voxels, and other rectangular box structures are still used to accelerate tasks such as raytracing (Cohen and Sheffer, 1994; Sramek and Kaufman, 2000; Mahovsky and Wyvill, 2004), topological analysis (Cohen-Or and Kaufman, 1995), volume estimation (Reitinger et al., 2003), collision detection (Nießner et al., 2013a), shadow rendering (Kämpe et al., 2016), and other complex lighting effects (Crassin et al., 2011). Many of these techniques use voxels to approximate more complex geometry, but in a purely voxel-based system such approaches are no longer approximations.

One area of particular interest for voxel-based MR is the ability to efficiently construct voxel-based models of the world in real time. Depth cameras have become an invaluable source of such reconstructions, and while the raw data from a RGB-D camera is typically a point cloud, voxels are used as intermediate representations in systems such as KinectFusion (Izadi et al., 2011; Newcombe et al., 2011). Voxels are used here to collate information from multiple views by constructing a fixed occupancy grid structure. The fixed grid size limits the reconstruction volume, although this can be mitigated by

FIGURE 1 | The principle of voxel-based Mixed Reality illustrated with our implementation of a prototype system with a voxel grid resolution of 8 mm (5/16 inches). Real, recorded, and virtual objects and people are coherently experienced. (Consent was obtained from individuals depicted).

moving the working volume over time (Roth and Vona, 2012) using octrees (Zeng et al., 2012) or voxel hashing (Nießner et al., 2013b). Such methods have been extended to large-volume mapping (Dai et al., 2017) and real-time reconstruction of dynamic scenes (Dou et al., 2013, 2016; Newcombe et al., 2015; Innmann et al., 2016). Voxels also provide a convenient representation for silhouette-based reconstruction (Slembrouck et al., 2015) as they can explicitly represent free-space. Again, these approaches have been successfully extended to real-time reconstruction of dynamic scenes (Cheung et al., 2000; Sridhar and Sowmya, 2009). GPU acceleration and the use of sparse voxel grids are often key components in these reconstructions (Loop et al., 2013), and this is likely to become more widespread with the release of nVidia's GVDB library to support such computations (Hoetzlein, 2016) This work shows that it is practical to convert even complex scenes into voxel-based representations in real-time, which is vital for voxel-based MR.

Voxels are also fundamental in several recent approaches to semantic labeling. While such labels are not required in all MR applications, they are very useful in assistive technologies and some aspects of collaborative virtual workspaces. Häne et al. (2013, 2017) use voxels to represent spatial constraints on scene labels. They are also able to infer the labels of unobserved voxels, allowing for volume estimation and volumetric segmentation. While their method is limited in the number of labels it can assign, this can be extended somewhat trough a block-based subdivision of the voxel space (Cherabier et al., 2016). The regular voxel grid also provides an ideal basis for deep *convolutional neural networks* (CNNs), as has been recently demonstrated by Zhou and Tuzel (2018), who converted 3D point clouds to a voxel-based representation before applying a CNN to detect pedestrians, cars, and cyclists in data captured from a moving vehicle. Since the convolution layers work by applying weighted connections over a grid, sub-volumes of voxel space provide a direct analog to image patches. Interactive labeling is also supported via voxels, as in Semantic Paint (Valentin et al., 2015) where users can provide labels through gestures and voice. These labels are used to train classifiers to support semi-automated labeling as more information is provided.

Photorealistic Mixed Reality research and development tries to seamlessly merge virtual objects into real scenes, paying most of its attention to consistent illumination (Kronander et al., 2015). If this cannot be done flawlessly, especially when human characters are to be integrated into the mixed environment, then the MR illusion can break quite rapidly. This so-called uncanny valley effect (Ho and MacDorman, 2017) can be avoided by, e.g., applying non-photorealistic rendering techniques (see below) or by using real objects or people as the source for mixing (Beck and Froehlich, 2017), which is also demonstrated in this paper.

2.2. Non-photorealistic Mixed Reality

Purely voxel-based rendering, at the current level of fidelity, does not provide a photorealistic view of the world. For voxel-based MR we advocate voxelization of the real world, so that the real and virtual elements become indistinguishable (Regenbrecht et al., 2017a). Non-photorealistic rendering has been shown to reduce the appearance differences between real and virtual objects (Fischer et al., 2005) when both are rendered in a stylized fashion, or to draw attention to stylized objects in an otherwise unchanged scene (Haller, 2004).

Chen et al. (2008) use a watercolor-inspired rendering technique to blur the distinction between real and virtual objects, Steptoe et al. (2014) compare realistic, stylized (edge-enhanced), and "virtualized" (edge detection and desaturated color), finding that the stylized environment gave the optimal blending of real and virtual elements and that all three treatments gave a high degree of presence and embodiment.

For video see-through AR it is often desirable to model the noise characteristics of the physical camera when rendering objects (Klein and Murray, 2010). Again, this aims to blur the distinctions between real and virtual by making the virtual objects have the same noisy appearance as low-cost cameras.

The common theme through this research is that realism is not essential to establishing a sense of presence or a seamless experience between real and virtual elements in an environment. Rather a key factor seems to be for the real and virtual elements to have similar visual character, allowing the user to accept the virtual as real.

2.3. Voxel-Based Mixed Reality

Voxels have been used in mixed reality applications for some time. As outlined earlier, voxels are often used in modeling and rendering virtual elements. A true voxel-based mixed reality system, however, uses voxels as the explicit and visible representation of the world. Many rendering systems that exploit voxels, such as KinectFusion (Izadi et al., 2011; Newcombe et al., 2011) and its derivatives, convert these back to textured triangle meshes for rendering. Furthermore, voxels are an underlying data

structure for several of the methods discussed earlier, particularly those that deal with semantic labels (Häne et al., 2013, 2017; Valentin et al., 2015; Zhou and Tuzel, 2018), and provide a direct way to represent free space, allowing for natural modeling of occlusions.

A recent example exploiting these benefits in mixed reality is "Remixed Reality" (Lindlbauer and Wilson, 2018). Users see a live reconstruction of the space that they are in, and virtual, remote, or pre-recorded elements can be inserted into this view. The underlying data structure that supports this is a regular voxel grid, and interactions, modifications, and composition of models all happen at the voxel-scale. The user, however, is presented with a more traditional rendering, where the voxels are converted to a mesh model. Such a conversion is not, however, necessary to present the user with a compelling experience. Regenbrecht et al. (2017a) show that a direct voxel rendering, like the other non-photorealistic rendering methods discussed above, give a strong sense of presence and embodiment. As with previous work on merging real and virtual in video see-through displays (Klein and Murray, 2010), a key step in this process is to render virtual elements with similar noise characteristics to the physical sensors (in this case Kinect depth sensors) so that real and virtual become indistinguishable.

3. CHARACTERISTICS OF A VOXEL-BASED MR SYSTEM

Voxel-based MR can provide simplicity and elegance in computation and software technology. This new simplicity applies to the way the virtual worlds are stored, computed, and presented and to the way we interact with them and hence develop an understanding of the underlying concept. Voxels, representing this simplicity, are easy to understand and easy to handle. If we reduce every aspect of our system to voxels in the purest possible fashion, we gain a number of positive effects in computing and engineering as well as in user experience, which are hard to achieve with other approaches. However, we do not want to benchmark voxel-based techniques against other techniques here, technically or empirically; we rather want to show that voxels are effective for our targeted mixed reality experience.

In the following, let us look at some more advantages voxels can have in delivering a real-time MR experience: visual coherence, unified model handling (everything is represented as voxels in a fixed grid), an inherently built-in occlusion handling, and much easier collision detection.

3.1. Visual Coherence

Visual coherence can be inherently achieved because everything is represented in the form of voxels. Today's state-of-the-art mixed reality systems suffer from a fidelity difference between real and virtual elements. Either reality is high resolution and virtual reality is not (e.g., optical see-through augmented reality) or virtual reality is high resolution and reality is not (e.g., video see-through augmented reality). Even projected augmented reality (commonly referred to as spatial AR) cannot normally deliver the necessary quality for a seamless blending. While there are approaches to minimize these differences, voxel-based MR might be the obvious method of choice to scale with demand. In addition to resolution matching, all spatial, temporal, and appearance flaws of the real world capturing process have to be simulated, too, to achieve coherence. For example, flickering and noise effects of depth-sensing cameras are applied to the virtual objects to blend them in.

3.2. Scalability of Resolution

The overall desired resolution is scalable. **Figure 2** shows example resolutions for a voxelized 3D model as rendered with our Mixed Voxel Reality system described below. The rendered voxel resolution is able to match the effective voxel resolution of the real-world capturing devices. For instance, with our $(2.5\ \mathrm{m})^3$ interaction volume system incorporating Microsoft Kinect cameras we can achieve a voxel resolution of 8mm (voxel edge length). If the resolution of those devices increases, the virtual resolution can match this increase, and **Figure 2** shows an example with a voxel size down to 1/8th of a millimeter. However, current off-the-shelf computing hardware can practically handle a 1–10 million-voxel scene with a resolution of 4–8 mm (Regenbrecht et al., 2017a) while voxel resolutions are yet not able to satisfy the interactivity requirements for MR applications.

3.3. Visual Occlusion, Interaction, and Collisions

While traditionally occlusion handling in virtual reality, and particularly mixed reality (Collins et al., 2017), is a challenging research topic, a voxel-based mixed reality approach solves this inherently. There is a "natural" depth position for each voxel and therefore all scene occlusions are handled correctly (within the limitations of the voxel resolution). Because real and virtual objects are treated identically, there is not even a real-virtual occlusion conflict, which for instance normally occurs with manual interactions. **Figure 3** illustrates how mutual occlusions between a virtual object and interacting hands are solved.

Similarly, collisions, especially for manual interactions between all objects in the scene, can be handled in a computing-efficient and more predictable way, e.g., the collision of a user's hand with a virtual object in a voxel environment (see **Figure 3**). Computationally, it is very inexpensive to calculate the collisions between the voxels in question. Also, the user's interaction with the object is based on the actually interacting points (voxels) and not on any invisible proxy geometries (like colliders) which are potentially confusing. Also, because of the underlying voxel grid, computationally inexpensive colliders, like spheres, can be used.

4. THE MIXED VOXEL REALITY SYSTEM

In our laboratory environment we implemented a prototypical system which allows for the actual experience of voxel-based mixed reality, i.e., to the users there are no other elements presented than voxels in the environment. This requires (a) the capturing and voxelization of the real environment, (b) the recording and playback of the real environment (elements) as voxels, (c) the voxelization of 3D models (CAD models), and,

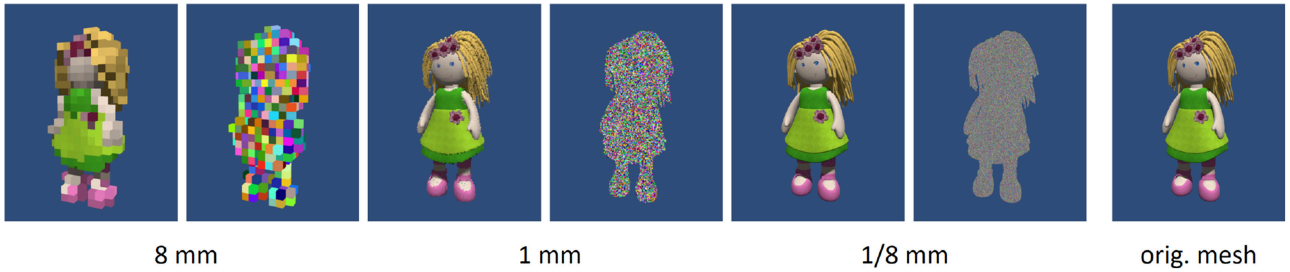

FIGURE 2 | Example voxel resolutions as rendered with our system accompanied by random color images to illustrate voxel densities. The achievable update rates for a $(2.5\ m)^3$ interaction space with current hardware and without optimizations are above 60 fps @ 8 mm, 8 fps @ 1 mm, and 0.07 fps at 1/8 mm.

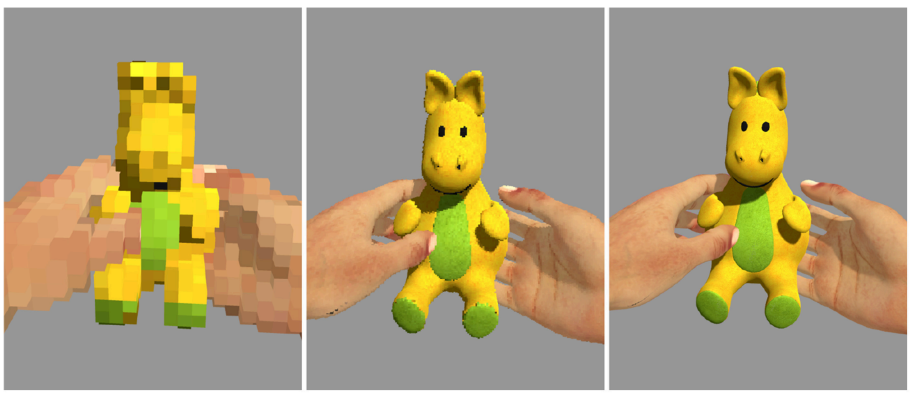

FIGURE 3 | Occlusions and interactions in voxel-based MR are effectively handled by the underlying voxel grid—in this example one (real) hand is occluding the (virtual) toy while the toy is occluding the other hand in the back (from left to right voxel resolutions: 8, 1, 1/8 mm).

for non-co-located environments, the transmission of voxel data over networks. Our system is integrated into a Unity scenegraph and our source code will be made publicly available.

4.1. Capture

An earlier version of the system only supported co-located (not networked) users and used only one Kinect camera (Regenbrecht et al., 2017a); here the real environment is captured with three Kinect cameras (**Figure 5**). Kinect camera #1 is our primary camera and is connected to a (Windows) computer, which serves also as our main computer to provide the visual voxel experience. The head-mounted display and its tracking system are also connected to this main computer.

In addition, two more Kinect cameras (#2 and #3) are connected to a second (Linux) computer, which pre-processes the two RGB-D data streams and sends the data via dedicated network ports to the main computer (**Figure 4**). Our *multi-Kinect server* (MKS) is based on Beck et al.'s implementation for calibrating multiple RGB-D sensors (Beck and Froehlich, 2017). We used the same Libfreenect2 library they used to implement a multi-Kinect server, and the C++ Boost library for concurrent RGB-D data acquisition on multiple threads. We also used the ZeroMQ middleware framework for sending RGB-D data locally because their system provides methods for receiving RGB-D data (using ZeroMQ), voxelizing the RGB-D data, and sending voxel data via UDP. When the MKS acquires new RGB-D image frames, the color image is adjusted to fit the field of view of the depth sensor. Because we use concurrent Boost threads for each Kinect device, the Boost barrier object is used to synchronize all Kinect threads to ensure each thread processes RGB-D data at the same time. Otherwise, with no synchronization, some threads might process RGB-D image frames faster than other threads, leading to different images at a point in time. The processed RGB-D frame from each Kinect is then written to a ZMQ message and then sent to the voxelization server. The voxelizer application receives these ZMQ messages, reads out the RGB-D frames, obtains colored point cloud data from the RGB-D images, and then voxelizes the point cloud based on a user defined bounding box (or voxel space). The voxel position is then truncated into an unsigned short integer and sent to our main MVR system via UDP. We also send all RGB-D data as separate instances using different ports so we can manually calibrate each Kinect (using the Unity scenegraph) at the cost of unsynchronized frames and calibration simplicity. While we could also have implemented Beck et al.'s multi Kinect calibration method, we opted for a manual alignment procedure because we do not require sub-millimeter accuracy as our voxel size is restricted by the resolution of the Kinect (resulting in 8mm voxel resolution). In addition, unsynchronized frames are sufficient here, because we either capture a static environment/objects only

and the dynamic objects (recorded people) are not moving fast enough to lead to issues with voxel alignment. Future higher voxel resolutions and possible recordings of faster moving objects will require synchronization, though.

The skeleton data we receive from the main Kinect is used to assign voxels to body parts for future use and is not relevant for the study and application scenarios reported here. Also, while the infrastructure would allow for more than three Kinect cameras, we limited the number to three. This also mitigates interference problems between multiple Kinect cameras resulting in temporal noise (flickering of voxels in mid-air and around objects).

4.2. Recording and Playback

The same system setup is used to record objects or people in the environment as for later playback in the voxel-based MR scene. The recording is done on the main computer and the recorded scenes are stored frame-by-frame.

In our system, a voxel is represented as a 3D position and a color. The voxel position in our system represents one corner of a cube. The voxel color is represented using the RGBA color space. In total one voxel can be represented in 16 bytes: three floating point data types for the x, y, z positions (12 bytes), and four byte data types for each red, green, blue colors and the alpha channel (four bytes). One voxel frame is stored as a list of voxels, and a voxel (recording) video is stored as a list of voxel frames.

The size of the recorded voxel file depends on the current voxel resolution—higher resolution increases the overall voxel count of the scene. As a guideline, 8 mm voxels generate about 200–250 kB per voxel frame. One second of recording takes up about 7 MB and a 30-s recording adds up to about 200 MB. When the recording is finished, the captured voxel frames are serialized using the C# standard library and stored as a .binary file.

Recorded voxel data is stored relative to the user-defined voxelspace and size and therefore can be directly deserialized and played back into the Unity scene. Furthermore, assuming the defined voxelspace and size is the same as the recording, it will be played back at its original location. Thirty frames per second are recorded based on the frame rate of the Kinect.

We have not invested in compression technique for this yet, our recorded clips are stored and loaded without any significant delay. We do not use any optimization or compression techniques yet as for our purposes the recorded files are still small enough to be handled efficiently.

4.3. Networking

To support our application scenarios for telepresence and collaboration, we implemented a simple but effective networking protocol to transmit and receive streams of voxel data in local area and wide area networks. Especially for wide area networks, a balance has to be found between the number of packets to be sent and the size of each of the packets. We determined the effective packet size by testing different configurations in the field between two very distant locations using fast, standard university internet connections.

We perform lossless compression, which reduces the voxel position data from 12 to 6 bytes. In our Unity system, one unit equals 1 m—that means a voxel size of 8 mm is represented by a floating point value of 0.008. Because the maximum precision in floating point values stores up to the 1/1,000th decimal place, we could multiply by 1,000 to obtain a whole number. Then we store the whole number as a 16-bit signed integer—essentially we are converting the units from meters to millimeters. Of course this compression method only works as long as the voxel position value as a C# short data type is within the range of −32,768 and 32,767. However, based on our uniform voxel resolution ($320 \times 320 \times 320$) and 8 mm voxels, each voxel position axis can be represented between −2,560 and 2,560—well within the short data type range limits. For the voxel color, we can reduce it from 4 to 3 bytes if the alpha channel is kept constant (255). To decompress the voxel data, we divide the compressed position values by 1,000 (convert unit from millimeter to meter) and add the constant alpha value of 255 (restore back to 32-bit RGBA). Therefore, we send [int16 x, int16 y, int16 z, byte red, byte green, byte blue] per voxel with packet sizes of up to 166 voxels, i.e., 9×166 bytes per packet.

We choose UDP because it is designed for fast data transmission (appropriate for real-time performance) at the cost of reliability. For our application, we can tolerate some packet loss assuming it does not significantly affect the visual appearance of the voxelized scene. Based on the maximum UDP packet size (65,535 bytes), our initial telepresence network protocol transmitted $9 \times 7{,}000$ bytes per UDP packet (7,000 voxels per packet) which works perfectly well in a lab environment. But in order to reduce the overhead related with packet fragmentation/reassembly at the IP layer for real WAN internet connections, we reduced the voxel packet size down to 9×166 bytes per UDP packet (166 voxels per packet)—which just fits the ethernet MTU (1,500 bytes).

We implemented the sending and receiving of voxels on concurrent threads. When the sender thread receives a new frame we segment the frame into smaller segments (166 voxels per packet). For each voxel, we reduce the voxel data in the aforementioned method, write them into the packet buffer, and send them to the receiver thread once the packet buffer is filled. We continue until all remaining voxels are processed. When all voxels in a frame are processed, a final 1 byte packet is sent to indicate the end of a frame. On the receiver thread, we implemented a dual buffer for concurrent reading and writing (render/receiver thread). We first check if the received packet is a 1 byte packet. If so we allocate the latest frame buffer to the unused buffer (front/back). We then check if both the dual buffers are unused by the render and receiver thread. If so, we swap the buffers and raise a new frame flag to the render thread. When an end of frame packet is not received, we extract all bytes from the packet, decompress the voxels, and then write them to a local buffer (latest frame buffer).

4.4. CAD Voxelization

To provide a coherent mixed reality environment, CAD models are voxelized and modified to match the appearance of captured objects. We closely match the Kinect capturing and voxelization result by modeling a virtual Kinect that casts rays into the scene based on the depth resolution (512×424) and horizontal/vertical field of view (70/60°). The Kinect device is positioned 2.3 m

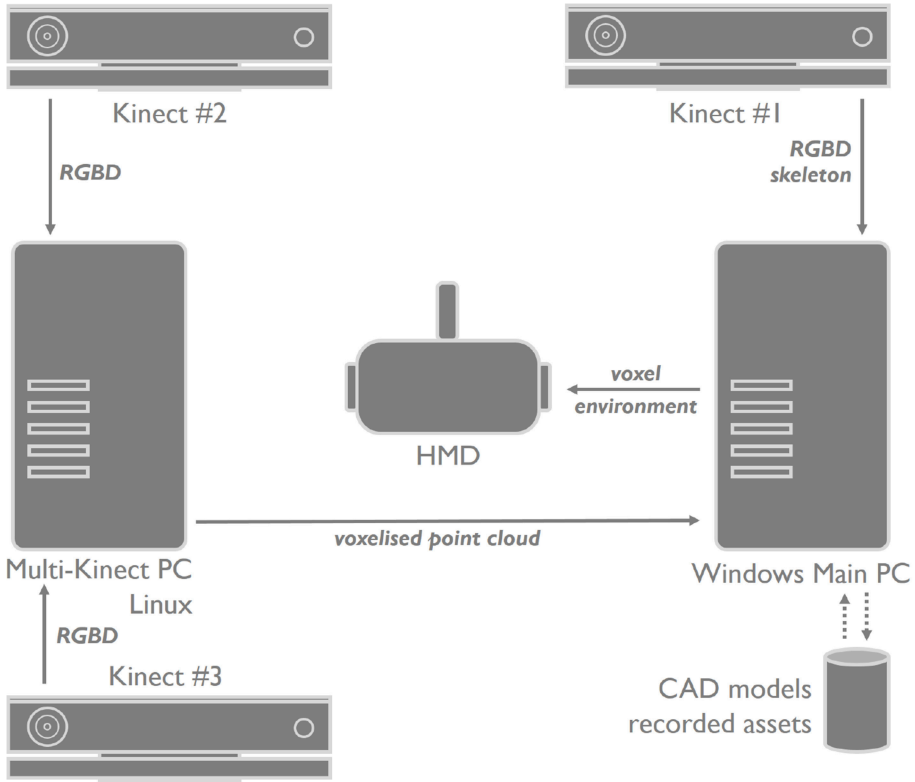

FIGURE 4 | Overview of our MVR system configuration.

above the world origin facing downwards at about 30° (**Figure 5**); therefore, rays are cast from the same Kinect position at the same angle in the Unity scene. For each depth image pixel, a ray is cast through the near clipping plane (0.5 m) into the scene. If a ray intersects with a mesh, the intersection point and color are obtained for all 512 × 424 rays (essentially a uniformly spaced point cloud). This is then mapped into our voxelspace producing voxelized CAD models. We also benefit from raycasted voxelization because by using these Kinect device properties, we define a view frustum which automatically provides frustum culling.

This technique alone only produces a static voxel model; however, we have to consider temporal noise similar to Kinect voxelization. Kinect noise is influenced by (1) the distance of the captured object to the sensor; (2) the angle between the camera and the captured surface; (3) the distance of a pixel in the depth image to the central pixel; and (4) the reflectivity of the captured material. We reproduced this noise for the CAD voxelization using a simple Gaussian function where the standard deviation is computed based on the distance between voxel and virtual Kinect position.

4.5. Integration and Rendering

All objects are voxelized into a regular grid space of 320^3 voxels ($2.56^3 m^3$ at 8 mm grid resolution) and rendered within a Unity3D system. We achieve this by using a geometry shader with a Point List input type and TriangleStream List output type. We send a single location for each generated voxel to the shader, where the GPU calculates the vertices and faces that make up the respective cube for each voxel every frame. To work within the limits of Unity we submit a maximum of 65,536 points per shader, spawning additional GameObjects based on the number of voxels in the scene. This approach allows for orders of magnitude more voxels to be updated and rendered with interactive frame rates compared with a CPU- and GameObject-based solution.

Individual voxels are unlit by the renderer—all the generated vertices for a single voxel have the same color, as determined during reconstruction, and the cube faces are not affected by ambient or directional lighting. Despite this, whole objects appear to be lit because the reconstruction of physical objects includes the real world lighting, and voxel generation for virtual objects includes the virtual scene lighting.

Where multiple voxels would be generated at the same point from overlapping objects (e.g., physical and virtual) we only render a single voxel without color averaging. This eliminates z-fighting and ensures that separate objects remain distinct rather than blending into one another. The final visual output is then rendered using the Oculus Utilities for Unity 5 and displayed on the HMD.

The entire system is operated in an office room (with a physical table in the middle). Three Kinect cameras are placed at roughly ceiling height, forming a triangle. For consistent

lighting, three additional lamps are placed next to the Kinect sensors. We use three Oculus tracking cameras. This setup gives us enough freedom and stability to track our interaction space (**Figure 5**). With our MVR system we are targeting our focus Mixed Reality application areas, namely telepresence and telecollaboration, the study of human behavior in different settings, entertainment, and training and education. Therefore, we provide functionalities within the MVR system specifically in support of those applications. In the next section, we are exploring the empirical feasibility of our concrete prototype implementation, the Mixed Voxel Reality system. We test a number of assumptions about the perception of a voxel-based MR environment with respect to our application targets with a laboratory study, in particular the sense of presence and co-presence as defining elements for MR.

5. USER STUDY

We designed and executed a user study using the Mixed Voxel Reality system we have implemented. Of particular interest have been the questions around whether people would perceive our voxel-based MR experience as convincing, whether people would be able to distinguish between real and recorded and purely virtual objects and people, and whether a sense of presence and co-presence could be developed. While answers to those questions might look trivial, for our low-resolution experience they needed to be addressed and confirmed.

As a summary, we found that

- our system is able to deliver a sense of presence and co-presence,
- recorded characters are convincing and trigger responses equivalent to real-world situations,
- people cannot distinguish real from virtual or recorded objects.

Hence, in principle, we are able to support our target application scenarios: Communication and collaboration can be enabled, virtual and real object and environment interaction can be supported, and educational training can be provided with pre-recorded, real or virtual content. In addition, all aspects are integrated into one coherent mixed reality experience. In the following we describe the experiment with our MVR system in detail.

5.1. Participants and Procedure

Twenty participants (15 male, 5 female) between 18 and 59 years (average 32 years) of different ethnicities (11 Caucasian amongst them) took part in the study. Nine of the participants had previous experiences with VR and HMDs.

Each participant was individually exposed to the voxel-based MR system wearing a HMD. The HMD was fitted outside the experimental office room. This way, participants did not see the real environment or the person operating the system. The investigator led the participant into the room, explained the tasks, and collected answers and took notes. The operator switched between different scenes and acted as a real character in one of the scenes. A questionnaire was administered after the study to measure the participants' sense of presence in the environment, their spatial perception, and signs of experienced simulator sickness.

5.2. Conditions and Scenes

We were interested in five questions addressed in the following subsections.

5.2.1. Are People Able to Distinguish Between Virtual (Recorded or Voxelized) and Real (Physically Present) Objects?

Each participant was shown a table with objects (toys) of mixed origin. An object could be (a) real and physically present, (b) recorded by the three Kinect setup and replayed, or (c) voxelized from a CAD object. Two different scenes with six objects each were prepared and participants were asked to name the objects and then specify the real objects in each scene (**Figure 6**). In each of the two scenes, the participant was asked to rate their confidence after identifying the real objects on a scale of low, medium, or high. The participant was allowed to approach the table and probe their assumptions by touching after completing the tasks. People could not confidently and correctly distinguish real from virtual objects in our voxel-based environment as overall confidence levels were low to medium and the ratio of correct vs. incorrect identification was not higher than by chance. More research is needed to explain what factors influence the perception of objects perceived as real or virtual.

5.2.2. Do People React With an Immediate Response to a Virtual (Recorded) Character?

To investigate if recorded characters are convincing enough to trigger an immediate response, a recording was played showing the instructor reading a piece of paper for about three seconds and then handing over the piece of paper to the participant across the table to explore if participants would act immediately by reaching out for the document (**Figure 7**, left). Data were collected as of immediate, hesitant, or no response at all. We found that recorded virtual characters are convincing and trigger an immediate response in most cases (18 participants), but incoherent elements in the environments (in our case voice not aligned with position) may lead to some hesitation (2 participants).

5.2.3. Are People Able to Distinguish Between a Virtual (Recorded) and Real (Physically Present) Character?

We wanted to find out if recorded characters are in the same way convincing as real characters. A scene was loaded showing two recorded and one real character, this way the participant was presented with a scene where three people standing behind a table and were asked to point to the real character within 10–20 s (**Figure 7**, right). We found that people could not confidently and correctly distinguish real and virtual characters as our results show that over half of the participants identified the real character incorrectly or were unable to distinguish at all.

FIGURE 5 | The interaction space of our voxel-based MR prototype system: three Kinect cameras observe a volume of $(2.56\ m)^3$, which is filled with real and virtual objects for the user. (Consent was obtained from individual depicted).

FIGURE 6 | Real setups **(Top)** and what the participants saw **(Bottom)**. There was temporal voxel noise present stemming from interferences of the three Kinect cameras—none of the participants reported negatively about this effect.

FIGURE 7 | (Left) Screenshot from user's perspective of recorded person (instructor) handing over a document. The two images on the right depict a scene setup in reality **(Left)** and as seen by the participants **(Right)**. (Consent was obtained from individuals depicted).

5.2.4. Are People Able to Identify the Gender of Virtual (Recorded) Characters?

To explore the level of recognition regarding recorded characters, we prepared two scenes with recordings of four people for each scene. We were mindful to record people providing not too much of a suggestion of the gender (e.g., no facial hair) and none of the female actors were wearing a skirt or a dress on the day of the recording. Participants were asked to specify the gender of the characters and tell us when they recognized a character they knew in person (**Figure 8**, left). People were able to specify the gender of a recorded character with very high accuracy but that with increasing distance (in our case over 2.3 m) the recognition was less spontaneous, but still correct.

5.2.5. Did People Develop a Spatial Awareness of Themselves and Others Within the Environment?

To explore this question, we prepared a scene with a virtual mirror. We placed two real scarfs on the table and participants were asked to try on a scarf of their choice and to observe their mirror reflection. After a couple of seconds a recorded character entered the scene from the side (**Figure 8**, right). The instructor took a note if participants (a) turned their head, or (b) realized and commented, or (c) did nothing at all. Our expectation that most participants would turn their head to check out the character beside them was not met. Only one participant turned the head immediately. Sixteen participants commented that somebody was in the room and would either reach out to the correct side to feel if somebody was there or would point to the correct position when asked where the character was in relation to them. Three participants did not react at all but reported, when asked, that they were aware that somebody was beside them. Even though we did not get the response we expected we are confident that all participants were spatially aware of the recorded character's position.

5.2.6. Self-Reported Presence and Believability

After completing the tasks, participants filled in a combined questionnaire with a total of 21 items. The first eight items were chosen from the igroup presence questionnaire IPQ (Schubert et al., 2001). The IPQ is an instrument to measure a person's sense of presence in a virtual environment assessing spatial presence, involvement, and realism, which are also relevant for mixed reality environments. We left out six items which are only applicable to pure virtual environments. In addition to the application of the IPQ, we administered an eight item subset of the Mixed Reality Experience Questionnaire (MREQ) (Regenbrecht et al., 2017b). Co-presence was measured by choosing the three co-presence items of BAIL (Bailenson et al., 2005), and we added two items asking about the believability of objects and characters in the voxel-based MR system (BoOC). All questions used Likert-like scales (7-point).

All four questionnaires' means (IPQ, MREQ, BAIL, BoOC) are significantly above midpoint as tested with a one-sample *t*-Test assuming unequal variances (df $= 19$). With a *t*-critical of 1.73 all *t*-stats are higher ($p < 0.05$) than *t*-critical (IPQ: 7.80, MREQ: 14.99, BAIL: 4.43, BoOC: 6.01) and therefore the means are significantly higher than mid-point (4.0). Given these results, we conclude that our voxel-based MR system is able to achieve a sense of presence, co-presence and that objects and characters were reported to be convincing. After reviewing the three co-presence questionnaire items we realized that they were a bit misleading because of the actual co-presence of the instructor in the room during the experiment. These items should have been rephrased to emphasize the co-presence of the characters in the mixed reality scene only.

5.2.7. Summary

The feasibility of the MVR system was explored by testing people's ability to distinguish real (physically presents) objects from virtual objects (recorded or voxelized from CAD models) as well as real characters from recorded characters in the MR environment. In line with our expectations, we found that in both scenarios, participants were not able to identify real objects or characters with high accuracy, and that participants reported low to medium confidence in their choices. Using a recorded character to trigger an immediate response was successful in the majority of cases, and we found that all participants correctly identified the gender of recorded characters with high confidence. In a simple setup using a virtual mirror, we explored if people developed spatial awareness within the MVR system. Although we did not observe the reaction we anticipated, we found that people were fully aware of the spatial arrangement.

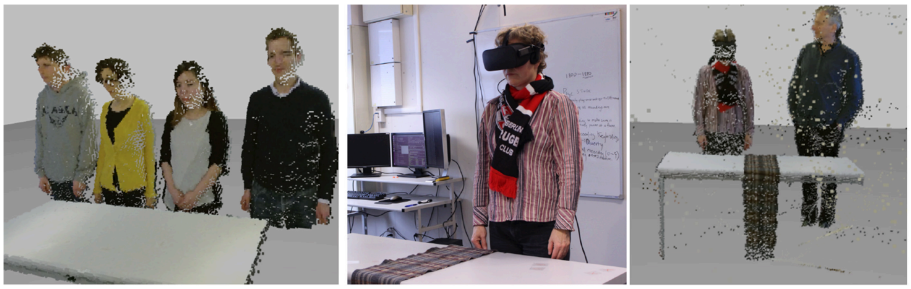

FIGURE 8 | (Left) Screenshot of user's view when asked to identify the gender of the recorded characters. The two images on the **(Right)** depict the real setup and the user's (mirror) view when a recorded character entered the scene. (Consent was obtained from individuals depicted).

These observations have been reinforced by the collected self-report measures.

6. CURRENT APPLICATIONS

In the following, we want to show how our VMR system was and is already used in different application scenarios. The first scenario makes use of the effect whereby people respond to recorded characters and the fidelity of the voxel resolution can be controlled. The second and third application scenarios build on the system's ability to deliver a perceived sense of co-presence in an entertainment and telepresence prototype scenario, respectively.

6.1. Human Behavior Study

The system's ability to finely tune levels of abstraction can be used for studies on human behavior in certain repeatable, controlled, and coherent environments. As demonstrated in one of our scenes above, voxel-based MR can elicit user responses like reaching for an offered document. The advantage of the coarse voxel character lies in the controllable balance between realism and abstraction: a person can be identifiable on a spectrum between just recognizable as a person to "that woman who just brought me into this room." This real-abstract balance allows for the study of human behavior under laboratory conditions where certain generic characteristics of a human (actor) are to be displayed (like gender), but controlling other confounding features (like empathy). Recently, our MVR system was used by behavioral psychologists for the study of prejudice in children where different ethnic characters are presented in our MVR environment and childrens' responses are measured. One hundred children with one of their parents present took part in a laboratory study to investigate parental impact on child prejudice to find out whether they tend to help the own-race individual first and whether they tend to choose the own-race individual to play with. The children wore a head-mounted display and saw different scenes with two recorded characters: one Asian and the other Caucasian sitting next to each other and pretend-interacting with the child (see **Figure 9**). The fidelity of the voxels for the characters was determined in a pilot study to find the right balance between realism and abstraction. The children's behavior was observed and brought into relation with parents' attitudes. The findings of this study are beyond the scope of this paper and will be published separately by our colleagues.

6.2. Telepresence

Three-dimensional telepresence systems would allow for more natural communication and cooperation over distance. Instead of showing head-and-shoulder videoconferencing views, meeting participants can take part as 3D bodily representations acting in 3D conferencing space. With coarse voxel resolutions we lose the ability to sense finer facial expressions, but we gain non-verbal communication cues like one's posture and gesturing. Voxels can be transmitted efficiently over networks and could be an enabler for scalable resolution 3D telepresence. Our MVR system allows two (for now) parties to meet in mutual voxel space. An internet-based connection is established between two voxel-based MR computers, and voxel data of the bodies of the participating people are transmitted via a proprietary voxel format. The transmission of audio data is left to a different (existing) channel.

Both parties can talk to each other, see each others voxel representations and therefore can interact with each other. **Figure 10** is showing a virtual handshake between two people in different rooms in our lab environment. Even if there is no haptic sensation, we observed a strange feeling of a real handshake. In addition, we successfully trialed this system over a real distance of many thousand kilometers and are going to study this application in more detail in the future.

6.3. Voxelvideos

MVR allows for the production of an alternative, next generation of video clips—truly three-dimensional videos which we call voxelvideos. In our example presented here we individually recorded audio streams and voxel representations of two folk musicians playing an Irish folk song (**Figure 11**). The voxel recordings are then brought into the virtual environment, the recorded audio tracks are positioned spatially where the virtual instruments are (here a fiddle and a guitar), and the audio is finally synchronized with the voxel recordings. During playback, users wearing a head-mounted display with stereo headphones can navigate throughout the scene by simply walking and/or teleporting and with this are experiencing a 3D visual and audio scene interactively.

FIGURE 9 | Screenshot **(Left)** and photograph of a real scene **(Right)** of an experimental study using our system involving 100 children determining prejudice bias. (Consent was obtained from individuals depicted).

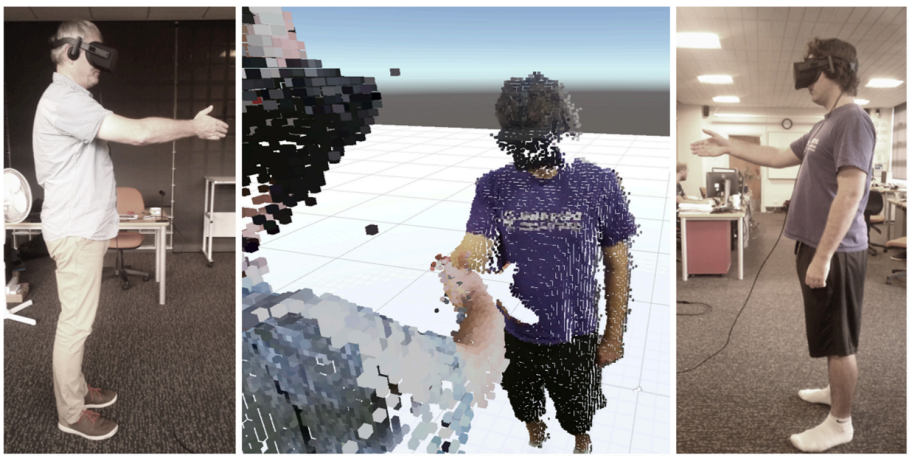

FIGURE 10 | Two users (**Left** and **Right**) at different physical locations virtually shake hands: screenshot taken of view over a user's shoulder **(Center)**. (Consent was obtained from individuals depicted).

7. FUTURE WORK AND CONCLUSION

Our application examples illustrate the *current* potential of voxel-based MR, i.e., what can be achieved with low-fidelity, early-stage voxel-based technology. Because of the state of infancy for voxel-based MR systems, we did not even make an attempt to show any superiority or inferiority in comparison to other techniques—we simply show feasibility and future prospects. Technology at large will scale, strictly following Moore's law or not, but also, if our prediction is right, specialized voxel solutions will evolve. For instance, instead of mesh-optimized graphics systems, manufacturers might offer voxel-optimized systems (as they did with volume rendering systems in the past) or more specialized algorithms will be developed not to optimize vertex geometries but voxel interactions. Such systems need not even be voxel-specific—voxel grids can be viewed as (3D) tensors, much as images are viewed as matrices, so modern tensor-based systems can be exploited. nVidia's voxel library GVDB (Hoetzlein, 2017) or Intel's True View technique for the NFL (View, 2017) show that there is interest in industry in voxel-based techniques.

Future commodity hard- and software for camera-based depth sensing will allow for the application of systems that are less spatially constrained, more portable and mobile, and perhaps even ubiquitous. Extending the idea of the Office of the Future (Raskar et al., 1998), where for instance all ceiling lights in an office are replaced by computer controlled cameras and projectors, we envisage that those cameras will be used for a fine-grained voxelized reconstruction of everything within that office. For instance, a new version of the seminal AR experience *Three Angry Man* (MacIntyre et al., 2002) could be delivered. We are working on combining many RGB-D and other sensors to (a) increase voxel fidelity, (b) extend the capture range, and (c) to achieve more complete voxelizations of real objects, people, and the environment.

If such an ubiquitous sensing space also incorporates techniques that allow capturing the *inside* of objects then a truly solid voxel experience can be delivered, i.e., one is able to really look into objects and to experience a world of solids and voids. Currently, this would be achievable for

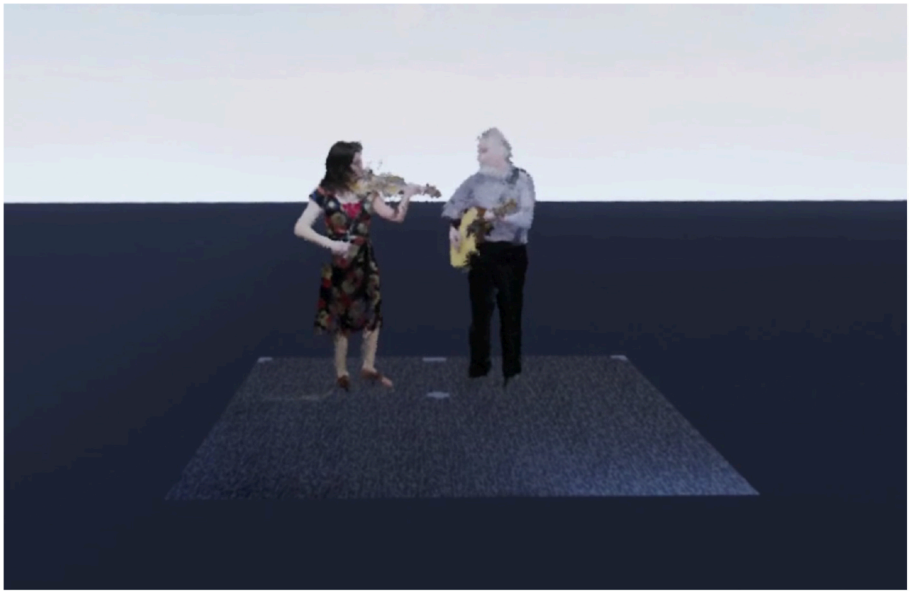

FIGURE 11 | Two musicians performing in a voxelvideo, which can be explored interactively (incl. spatial sound). (Consent was obtained from individuals depicted).

virtual voxel models stemming from solid CAD models, like those produced with constructive solid geometry modelers, or deriving from volumetric data like 3D CT scans. Further away future sensing systems might go beyond light (visual, nearly visible) toward techniques like pervasive MRI. For now, we will concentrate on turning CAD models into solid voxel representations, including ways to "intelligently" fill geometries which are given by their hulls only.

While building those hull-based or solid voxel objects and environments, we can assign meaning to each and every one of the voxels. This can and should include each voxel's origin (belongs to CAD object X), its physical properties, its relationship to other voxels, its label, etc. This can be done automatically, e.g., while converting CAD models, interactively, e.g., by "semantically painting" voxels (c.f. Valentin et al., 2015), by way of machine learning (recognizing objects), or by any other or a combination of those techniques. All three ways are subject to our current research activities.

Beyond visual, this voxel-based MR approach can be extended to other sensory modalities, e.g., each voxel, and group of voxels, can be assigned acoustic or tactile properties, making it a very elegant and simple system to deliver a multi-sensory experience. Currently, we are working on early research projects which demonstrate how even today voxel-based techniques can provide alternatives to other techniques making voxels more suitable because of their unique, integrated nature—everything is voxels. Voxels are not only used for internal (e.g., fast 3D array) representations but also for external (e.g., octree file format), visualization (massive parallel, GPU-powered rendering), and interaction (e.g., occlusion handling) purposes.

Today's mixed reality voxel worlds are coarse—tomorrow's voxel worlds will be fine (ambiguity intended). In this paper we are making a case for using voxels in MR by way of technical and practical argument, illustrating examples, and exploratory user study. We think that the potential of voxel-based MR is not unleashed yet. However, we could show that voxels are effective in terms of users' interactions and technical implementation, that voxels can lead to a sense of presence, and that voxels are computationally efficient and scalable.

We hope that we have made a convincing enough argument for researchers and practitioners to consider voxel-based mixed reality as an option for new user experiences to be designed and developed.

ETHICS STATEMENT

The study was approved by the Ethics Committee of the University of Otago (D17/279). The participants provided written informed consent to take part in this study.

AUTHOR CONTRIBUTIONS

HR conceived the idea, managed the project, oversaw the development of the system, study, and paper and wrote major parts of sections. J-WP developed the major parts of the underlying system (based on earlier work) and co-executed the study. CO was responsible for the user study and wrote the major part of that section. MC contributed to critical parts of the system implementation. SM was the main author of the related work section. TL co-led the management of the project and made major revisions to the paper. All authors

contributed to manuscript revision, read, and approved the submitted version.

FUNDING

Parts of this project have been funded by University of Otago Research Grants 2016 and 2018 and by NZ's National Science Challenge (SfTI) funding.

ACKNOWLEDGMENTS

We would like to thank Katrin Meng and Arne Reepen for their contributions to earlier versions of the system, our participants and the HCI Lab's people time and effort, the makers of the CAD models used in our studies, and Kevin from Weta for encouraging discussions about voxels.

REFERENCES

Bailenson, J. N., Swinth, K., Hoyt, C., Persky, S., Dimov, A., and Blascovich, J. (2005). The independent and interactive effects of embodied-agent appearance and behavior on self-report, cognitive, and behavioral markers of copresence in immersive virtual environments. *Presence Teleoper. Virtual Environ.* 14, 379–393.

Beck, S., and Froehlich, B. (2017). "Sweeping-based volumetric calibration and registration of multiple RGBD-sensors for 3D capturing systems," in *Proceedings of IEEE Virtual Reality 2017* (Los Angeles, CA), 167–176.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 509–517. doi: 10.1145/361002.361007

Chen, J., Turk, G., and MacIntyre, B. (2008). "Watercolor inspired non-photorealistic rendering for augmented reality," in *ACM Symposium on Virtual Reality Software and Technology* (Bordeaux), 231–234.

Cherabier, I., Häne, C., Oswald, M. R., and Pollefeys, M. (2016). "Multi-label semantic 3D reconstruction using voxel blocks," in *Proceedings of the Fourth International Conference on 3D Vision (3DV 2016)* (Stanford, CA).

Cheung, G. K., Kanade, T., Bouguet, J.-Y., and Holler, M. (2000). "A real time system for robust 3D voxel reconstruction of human motions," in *IEEE Conference on Computer Vision and Pattern Recognition, 2000, Vol. 2* (Hilton Head, SC), 714–720.

Cleary, J. G., and Wyvill, G. (1988). Analysis of an algorithm for fast ray tracing using uniform space subdivision. *Visual Comput.* 4, 65–83. doi: 10.1007/BF01905559

Cohen, D., and Sheffer, Z. (1994). Proximity clouds–an acceleration technique for 3D grid traversal. *Visual Comput.* 11, 27–38. doi: 10.1007/BF01900697

Cohen-Or, D., and Kaufman, A. (1995). Fundamentals of surface voxelization. *Graph. Models Image Process.* 57, 453–461. doi: 10.1006/gmip.1995.1039

Collins, J., Regenbrecht, H., and Langlotz, T. (2017). Visual coherence in mixed reality: a systematic enquiry. *Presence Teleop. Virt. Environ.* 26, 16–41. doi: 10.1162/PRES_a_00284

Crassin, C., Neyret, F., Sainz, M., Green, S., and Eisemann, E. (2011). Interactive indirect illumination using voxel cone tracing. *Comput. Graph. Forum* 30, 1921–1930. doi: 10.1111/j.1467-8659.2011.02063.x

Dai, A., Nießner, M., Zollhöfer, M., Izadi, S., and Theobalt, C. (2017). BundleFusion: real-time globally consistent 3D reconstruction using on-the-fly surface reintegration. *ACM Trans. Graph. (TOG)* 36:24. doi: 10.1145/3072959.3054739

Dou, M., Fuchs, H., and Frahm, J. (2013). "Scanning and tracking dynamic objects with commodity depth cameras," in *IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2013* (Adelaide, SA), 99–106.

Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., et al. (2016). Fusion4D: real-time performance capture of challenging scenes. *ACM Trans. Graph.* 35:114. doi: 10.1145/2897824.2925969

Fischer, J., Bartz, D., and Straßer, W. (2005). "Stylized augmented reality for improved immersion," in *IEEE Virtual Reality* (Washington, DC).

Fuchs, H., Kedem, Z. M., and Naylor, B. F. (1980). "On visible surface generation by a priori tree structures," in *ACM Siggraph Computer Graphics, Vol. 14* (Seattle, WA), 124–133.

Haller, M. (2004). "Photorealism or/and non-photorealism in augmented reality," in *ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry* (Singapore), 189–196.

Häne, C., Zach, C., Cohen, A., Angst, R., and Pollefeys, M. (2013). "Joint 3D scene reconstruction and class segmentation," in *IEEE Conf. Computer Vision and Pattern Recognition*, 97—104.

Häne, C., Zach, C., Cohen, A., and Pollefeys, M. (2017). Dense semantic 3D reconstruction. *IEEE Trans Pattern Anal. Mach. Intell.* 39, 1730–1743. doi: 10.1109/TPAMI.2016.2613051

Ho, C.-C., and MacDorman, K. F. (2017). Measuring the uncanny valley effect. *Int. J. Soc. Robot.* 9, 129–139. doi: 10.1007/s12369-016-0380-9

Hoetzlein, R. (2017). Raytracing sparse volumes with nvidia gvdb in designworks. Available online at: https://developer.nvidia.com/sites/default/files/akamai/designworks/docs/GVDB_TechnicalTalk_Siggraph2016.pdf

Hoetzlein, R. K. (2016). "GVDB: Raytracing sparse voxel database structures on the GPU," in *High-Performance Graphics Conference* (Dublin), 109–117.

Innmann, M., Zollhöfer, M., Nießner, M., Theobalt, C., and Stamminger, M. (2016). "VolumeDeform: real-time volumetric non-rigid reconstruction," in *European Conference on Computer Vision* (Amsterdam: Springer), 362–379.

Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., et al. (2011). "KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (Santa Barbara, CA), 559–568.

Kämpe, V., Sintorn, E., Dolonius, D., and Assarsson, U. (2016). Fast, memory-efficient construction of voxelized shadows. *IEEE Trans. Visual. Comput. Graph.* 22, 2239–2248. doi: 10.1145/2699276.2699284

Klein, G., and Murray, D. W. (2010). Simulating low-cost cameras for augmented reality compositing sign in or purchase. *IEEE Trans. Visual. Comput. Graph.* 16, 369–380. doi: 10.1109/TVCG.2009.210

Kronander, J., Banterle, F., Gardner, A., Miandji, E., and Unger, J. (2015). Photorealistic rendering of mixed reality scenes. *Comp. Graph. Forum* 34, 643–665. doi: 10.1111/cgf.12591

Lindlbauer, D., and Wilson, A. D. (2018). "Remixed reality: manipulating space and time in augmented reality," in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal, QC), 129.

Loop, C., Zhang, C., and Zhang, Z. (2013). "Real-time high-resolution sparse voxelization with applications to image-based modeling," in *High-Performance Graphics Conference* (Anaheim, CA), 73–79.

MacIntyre, B., Bolter, J. D., Vaughan, J., Hannigan, B., Moreno, E., Haas, M., et al. (2002). "Three angry men: dramatizing point-of-view using augmented reality," in *ACM SIGGRAPH 2002 Conference Abstracts and Applications* (San Antonio, TX), 268–268.

Mahovsky, J., and Wyvill, B. (2004). Fast ray-axis aligned bounding box overlap test with Plüker coordinates. *J. Graph. Tools* 9, 35–46. doi: 10.1080/10867651.2004.10487597

Meagher, D. (1982). Geometric modeling using octree encoding. *Comput. Graph. Image Process.* 19, 129–147. doi: 10.1016/0146-664X(82)90104-6

Newcombe, R. A., Fox, D., and Seitz, S. M. (2015). "Dynamicfusion: reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Boston, MA), 343–352.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., et al. (2011). "KinectFusion: real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Boston, MA), 127–136.

Nießner, M., Siegl, C., Schäfer, H., and Loop, C. T. (2013a). "Real-time collision detection for dynamic hardware tessellated objects," in *Eurographics (Short Papers)* (Girona), 33–36.

Nießner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. (2013b). Real-time 3D reconstruction at scale using voxel hashing. *ACM Trans. Graph.* 32, 169:1–169:11. doi: 10.1145/2508363.2508374

Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., and Fuchs, H. (1998). "The office of the future: a unified approach to image-based modeling and spatially immersive displays," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques* (Orlando, FL), 179–188.

Regenbrecht, H., Meng, K., Reepen, A., Beck, S., and Langlotz, T. (2017a). "Mixed voxel reality: presence and embodiment in low fidelity, visually coherent, mixed reality environments," in *2017 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* (Nantes), 90–99.

Regenbrecht, H., Schubert, T., Botella, C., and Baños, R. (2017b). *Mixed Reality Experience Questionnaire (mreq)-Reference.* University of Otago.

Reitinger, B., Bornik, A., and Beichel, R. (2003). "Efficient volume measurement using voxelization," in *Proceedings of the 19th Spring Conference on Computer Graphics, SCCG '03* (New York, NY:ACM), 47–54.

Roth, H., and Vona, M. (2012). "Moving volume KinectFusion," in *BMVC, Vol. 20* (Surrey), 1–11.

Schubert, T., Friedmann, F., and Regenbrecht, H. (2001). The experience of presence: factor analytic insights. *Presence Teleoper. Virtual Environ.* 10, 266–281. doi: 10.1162/105474601300343603

Slembrouck, M., Van Cauwelaert, D., Veelaert, P., and Philips, W. (2015). "Shape-from-silhouettes algorithm with built-in occlusion detection and removal," in *International Conference on Computer Vision Theory and Applications (VISAPP 2015)* (Berlin: SCITEPRESS).

Sramek, M., and Kaufman, A. (2000). Fast ray-tracing of rectilinear volume data using distance transforms. *IEEE Trans. Visual. Comput. Graph.* 6, 236–252. doi: 10.1109/2945.879785

Sridhar, A., and Sowmya, A. (2009). "Sparsespot: using a priori 3-d tracking for real-time multi-person voxel reconstruction," in *Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology, VRST '09*, (New York, NY: ACM), 135–138.

Steptoe, W., Julier, S., and Steed, A. (2014). "Presence and discernability in conventional and non-photorealistic immersive augmented reality," in *IEEE International Symposium on Mixed and Augmented Reality* (Munich), 213–218.

Valentin, J., Vineet, V., Cheng, M.-M., Kim, D., Shotton, J., Kohli, P., et al. (2015). Semanticpaint: interactive 3d labeling and learning at your fingertips. *ACM Trans. Graph.* 34, 154:1–154:17. doi: 10.1145/2751556

View, I. T. (2017). Get closer to the game with intel true view. Available online at: https://www.intel.com/content/www/us/en/sports/nfl/overview.html

Zeng, M., Zhao, F., Zheng, J., and Liu, X. (2012). "A memory-efficient KinectFusion using octree," in *Computational Visual Media* (Beijing: Springer), 234–241.

Zhou, Y., and Tuzel, O. (2018). "VoxelNet: end-to-end learning for point cloud based 3d object detection," in *International Conference on Computer Vision and Pattern Recognition* (Salt Lake City, UT).

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.