Holoportation: Virtual 3D Teleportation in Real-time

S. Orts-Escolano **C. Rhemann** **S.R. Fanello** **D. Kim** **A. Kowdle** **W. Chang**
Y. Degtyarev **P. Davidson** **S. Khamis** **M. Dou** **V. Tankovich** **C. Loop**
Q. Cai **P. Chou** **S. Mennicken** **J. Valentin** **P. Kohli** **V. Pradeep**
S. Wang **Y. Lutchyn** **C. Keskin** **S. Izadi**

Microsoft Research

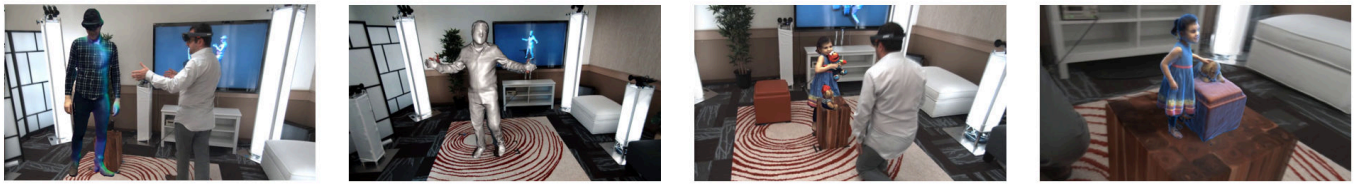

Figure 1. Holoportation is a new immersive telepresence system that combines the ability to capture high quality 3D models of people, objects and environments in real-time, with the ability to transmit these and allow remote participants wearing virtual or augmented reality displays to see, hear and interact almost as if they were co-present.

ABSTRACT

We present an end-to-end system for augmented and virtual reality telepresence, called Holoportation. Our system demonstrates high-quality, real-time 3D reconstructions of an entire space, including people, furniture and objects, using a set of new depth cameras. These 3D models can also be transmitted in real-time to remote users. This allows users wearing virtual or augmented reality displays to see, hear and interact with remote participants in 3D, almost as if they were present in the same physical space. From an audio-visual perspective, communicating and interacting with remote users edges closer to face-to-face communication. This paper describes the Holoportation technical system in full, its key interactive capabilities, the application scenarios it enables, and an initial qualitative study of using this new communication medium.

Author Keywords

Depth Cameras; 3D capture; Telepresence; Non-rigid reconstruction; Real-time; Mixed Reality; GPU

ACM Classification Keywords

I.4.5 Image Processing and Computer Vision: Reconstruction; I.3.7 Computer Graphics: Three-Dimensional Graphics and Realism,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
UIST 2016, October 16-19, 2016, Tokyo, Japan.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-4189-9/16/10..$15.00.
DOI: http://dx.doi.org/10.1145/2984511.2984517

INTRODUCTION

Despite huge innovations in the telecommunication industry over the past decades, from the rise of mobile phones to the emergence of video conferencing, these technologies are far from delivering an experience close to *physical co-presence*. For example, despite the myriad of telecommunication technologies, we still spend over $1 trillion/year globally on business travel, with the US making up $300 billion/year and over 482 million flights/year[1]. And this is without counting the cost on the environment. Indeed telepresence has been cited as key in battling carbon emissions in the future[2].

However, despite the promise of telepresence, clearly we are still spending a great deal of time, money, and CO_2 getting on planes to meet face-to-face. Somehow much of the subtleties of face-to-face co-located communication — eye contact, body language, physical presence — are still lost in even high-end audio and video conferencing. There is still a clear gap between even the highest fidelity telecommunication tools and physically being there.

In this paper, we describe Holoportation, a system that attempts to close this gap. Holoportation is a new communication tool that leverages consumer augmented reality (AR) and virtual reality (VR) display technologies, and combines these with a state-of-the-art, real-time 3D capture system. This system is capable of capturing in full 360^o the people, objects and motions within a room, using a set of custom depth cameras. This 3D content is captured and transmitted to remote participants, in real-time.

Any person or object entering the instrumented room will be captured in full 3D, and *virtually teleported* into the remote participants space. Each participant can now see and hear

[1] `http://www.forbes.com/sites/kenrapoza/2013/08/06/business-travel-market-to-surpass-1-trillion-this-year/`
[2] `http://www.scientificamerican.com/article/can-videoconferencing-replace-travel/`

these remote users within their physical space when they wear their AR or VR displays. From an audio-visual perspective, this gives users an impression that they are co-present in the same physical space as the remote participants.

Our main contribution is a new end-to-end system for high-quality, real-time capture, transmission and rendering of people, spaces, and objects in full 3D. Apart from the system as a whole, another set of technical contributions are:

- A new active stereo depth camera technology for real-time high-quality capture.
- A novel two GPU pipeline for higher speed temporally consistent reconstruction.
- A real-time texturing pipeline that creates consistent colored reconstructions.
- Low latency pipeline for high quality remote rendering.
- Spatial audio that captures user position and orientation.
- A prototype for headset removal using wearable cameras.

Furthermore, this paper also contributes key interactive capabilities, new application scenarios, and an initial qualitative study of using this new communication medium.

RELATED WORK

There has been a huge amount of work on immersive 3D telepresence systems (see literature reviews in [16, 4, 32]). Given the many challenges around real-time capture, transmission and display, many systems have focused on one specific aspect. Others have constrained the scenarios, for example limiting user motion, focusing on upper body reconstructions, or trading quality for real-time performance.

Early seminal work on telepresence focused on capturing dynamic scenes using an array of cameras [17, 27]. Given both the hardware and computational limitations of these early systems, only low resolution 3D models could be captured and transmitted to remote viewers. Since this early work, we have seen improvements in the real-time capture of 3D models using multiple cameras [57, 55, 31]. Whilst results are impressive, given the real-time constraint and hardware limits of the time, the 3D models are still far from high quality. [48, 36] demonstrate compelling real-time multi-view capture, but focus on visual hull-based reconstruction —only modeling silhouette boundaries. With the advent of Kinect and other consumer depth cameras, other real-time room-scale capture systems emerged [39, 43, 25, 47]. However, these systems again lacked visual quality due to lack of temporal consistency of captured 3D models, sensor noise, interference, and lack of camera synchronization. Conversely, high quality offline reconstruction of human models has been demonstrated using both consumer, e.g. [13], and custom depth cameras, e.g. [11]. Our aim with Holoportation is to achieve a level of visual 3D quality that steps closer to these offline systems. By doing so, we aim to achieve a level of presence that has yet to be achieved in previous real-time systems.

Beyond high quality real-time capture, another key differentiator of our work is our ability to support true motion within the scene. This is broken down into two parts – remote and local motion. Many telepresence systems limit the local user to a constrained seating/viewing position, including the seminal TELEPORT system [19], and more recent systems [47, 10, 38, 5, 62], as part of the constraint of the capture setup. Others constrain the motion of the remote participant, typically as a by-product of the display technology used whether it is a situated stereo display [10] or a more exotic display technology. Despite this, very compelling telepresence scenarios have emerged using situated autostereo [42, 45], cylindrical [28], volumetric [24], lightfield [1] or even true holographic [7] displays. However, we feel that free-form motion within the scene is an important factor in creating a more faithful reconstruction of actual co-located presence. This importance of mobility has been studied greatly in the context of co-located collaboration [37], and has motivated certain tele-robotics systems such as [26, 33], and products such as the Beam-PRO. However, these lack human embodiment, are costly and only support a limited range of motion.

There have been more recent examples of telepresence systems that allow fluid user motion within the space, such as blue-c [21] or [4]. However, these systems use stereoscopic projection which ultimately limits the ability for remote and local users to actually inhabit the exact same space. Instead interaction occurs through a 'window' from one space into the other. To resolve this issue, we utilize full volumetric 3D capture of a space, which is overlaid with the remote environment, and leverage off-the-shelf augmented and virtual reality displays for rendering, which allow spaces to be shared and co-habited. [40] demonstrate the closest system to ours from the perspective of utilizing multiple depth cameras and optical see-through displays. Whilst extremely compelling, particularly given the lack of high quality, off-the-shelf, optical see-through displays at the time, the reconstructions do lack visual quality due to the use of off-the-shelf depth cameras. End-to-end rendering latency is also high, and the remote user is limited to remaining seated during collaboration.

Another important factor in co-located collaboration is the use of physical props [37]. However, many capture systems are limited to or have a strong prior on human bodies [9], and cannot be extended easily to reconstruct other objects. Further, even with extremely rich offline shape and pose models [9], reconstructions can suffer from the effect of "uncanny valley" [44]; and clothing or hair can prove problematic [9]. Therefore another requirement of our work is to support arbitrary objects including people, furniture, props, animals, and to create faithful reconstructions that attempt to avoid the uncanny valley.

SYSTEM OVERVIEW

In our work, we demonstrate the first end-to-end, real-time, immersive 3D teleconferencing system that allows both local and remote users to move freely within an entire space, and interact with each other and with objects. Our system shows unprecedented quality for real-time capture, allows for low end-to-end communication latency, and further mitigates remote rendering latency to avoid discomfort.

To build Holoportation we designed a new pipeline as shown

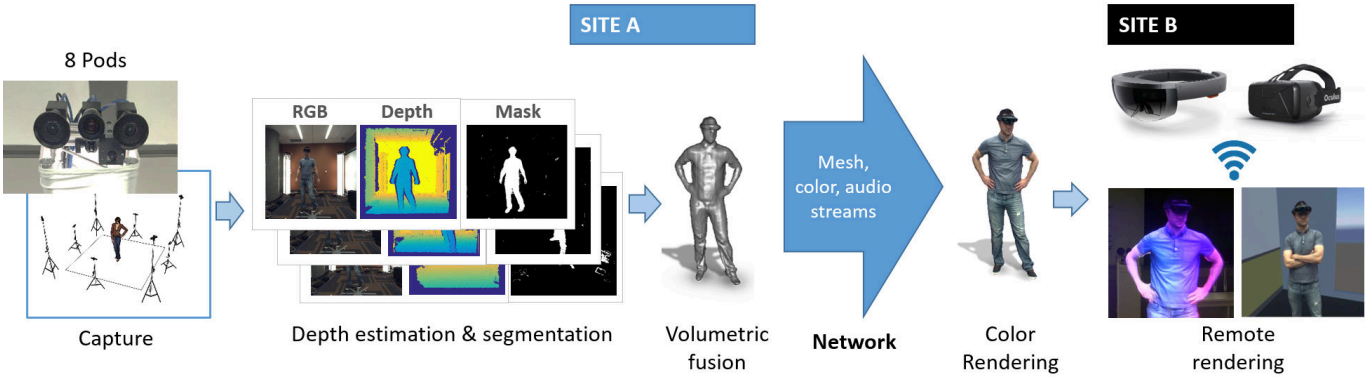

Figure 2. Overview of the Holoportation pipeline. Our capture units compute RGBD streams plus a segmentation mask. The data is then fused into a volume and transmitted to the other site. Dedicated rendering PCs perform the projective texturing and stream the live data to AR and VR displays.

in Fig. 2. We describe this pipeline in full in the next sections, beginning with the physical setup, our algorithm for high quality depth computation and segmentation, realistic temporal geometry reconstruction, color texturing, image compression and network, and remote rendering.

Physical Setup and Capture Pods

For full 360° capture of the scene, we employ $N = 8$ camera *pods* placed on the periphery of the room, pointing inwards to capture a unique viewpoint of the subject/scene. Each pod (Fig. 3, middle) consists of 2 Near Infra-Red cameras (NIR) and a color camera mounted on top of an optical bench to ensure its rigidity. Additionally a diffractive optical element (DOE) and laser is used to produce a pseudo-random pattern (for our system we use the same design as the Kinect V1). We also mount NIR filters on top of each NIR camera to filter out the visible light spectrum. Each trinocular pod generates a color-aligned RGB and depth stream using a state-of-the-art stereo matching technique described below. Current baseline used in the active stereo cameras is 15 centimeters, giving an average error of 3 millimeters at 1 meter distance and 6 millimeters at 1.5 meter distance.

In total, our camera rig uses 24 4MP resolution Grasshopper PointGrey cameras. All the pods are synchronized using an external trigger running at 30fps. Fig. 3, right, shows an example of images acquired using one of the camera pods.

The first step of this module is to generate depth streams, which require full intrinsics and extrinsics calibration. In this work we use [63] for computing the camera parameters. Another standard calibration step ensures homogeneous and consistent color information among the RGB cameras. We perform individual white balancing using a color calibration chart. Once colors have been individually tuned, we use one RGB camera as a reference and warp the other cameras to this reference using a linear mapping. This makes the signal consistent across all the RGB cameras. This process is performed offline and then linear mapping across cameras is applied at runtime.

Depth Estimation

Computing 3D geometry information of the scene from multiple view points is the key building block of our system. In our case, the two key constraints are estimating consistent depth information from multiple viewpoints, and doing so in *real-time*.

Depth estimation is a well studied problem where a number of diverse and effective solutions have been proposed. We considered popular depth estimation techniques such as structured light, time-of-flight and depth from stereo for this task.

Structured light approaches allow for very fast and precise depth estimation [59, 49, 6, 3, 15]. However, in our setup of multi-view capture structured light-based depth estimation suffers from interference across devices. Another alternative strategy is time-of-flight based depth estimation [22], which has grown to be very popular, however, multi-path issues make this technology unusable in our application. Depth from RGB stereo is another possible solution that has seen several advances in real-time depth estimation [51, 61, 56, 52]. Passive stereo uses a pair of rectified images and estimates depth by matching every patch of pixels in one image with the other image. Common matching functions include sum of absolute differences (SAD), sum of squared differences (SSD), and normalized cross-correlation (NCC). This is a well understood approach to estimating depth and has been demonstrated to provide very accurate depth estimates [8]. However, the main issue with this approach is the inability to estimate depth in case of texture-less surfaces.

Motivated by these observations, we circumvent all these problems by using active stereo for depth estimation. In an active stereo setup, each camera rig is composed of two NIR cameras plus one or more random IR dot pattern projector (composed of laser plus DOE). Each serves as a texture in the scene to help estimate depth even in case of texture-less surfaces. Since we have multiple IR projectors illuminating the scene we are guaranteed to have textured patches to match and estimate depth.

Additionally, and as importantly, we do not need to know the pattern that is being projected (unlike standard structured light systems such as Kinect V1). Instead, each projector simply adds more texture to the scene to aid depth estimation. This circumvents the issues of interference which commonly occur when two structured light cameras overlap. Here, overlapping projectors actually result in more texture for our matching algorithm to disambiguate patches across cameras,

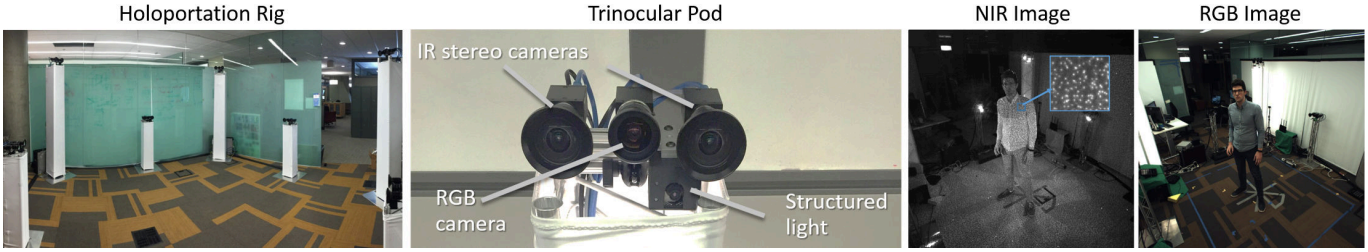

Figure 3. Left: Holoportation rig. Middle: Trinocular Pod: Right: NIR and RGB images acquired from a pod.

so the net result is an improvement rather than degradation in depth quality.

With the type of depth estimation technique narrowed down to active stereo, our next constraint is real-time depth estimation. There are several approaches that have been proposed for depth estimation [53]. We base our work on PatchMatch stereo [8], which has been shown to achieve high quality dense depth maps with a runtime independent of the number of depth values under consideration. PatchMatch stereo alternates between random depth generation and propagation of depth based on the algorithm by [2], and has recently been extended to real-time performance by assuming fronto parallel windows and reducing the number of iterations of depth propagation [65]. In this work, we develop a real-time CUDA implementation of PatchMatch stereo that performs depth estimation at as high as 50fps on a high-end GPU such as an NVIDIA Titan X in our case. Details of the GPU implementation are provided later in the paper. Examples of depthmaps are shown in Fig. 4.

Foreground Segmentation

The depth estimation algorithm is followed by a segmentation step, which provides 2D silhouettes of the regions of interest. These silhouettes play a crucial role – not only do they help in achieving temporally consistent 3D reconstructions [14] but they also allow for compressing the data sent over the network. We do not use green screen setups as in [11], to ensure that the system can be used in natural environments that appear in realistic scenarios.

The input of the segmentation module consists of the current RGB image I_t, and depth image D_t. The systems also maintains background appearance models for the scene from the perspective of each camera pod in a manner similar to [29]. These background models are based on depth and RGB appearance of the empty scene (where the objects of interest are absent). They are represented by an RGB and depth image pair for each camera pod $\{I_{bg}, D_{bg}\}$, which are estimated by averaging over multiple frames to make the system robust against noise (e.g. depth holes, light conditions etc.).

We model the foreground/background segmentation labeling problem using a fully-connected Conditional Random Field (CRF) [30] whose energy function is defined as:

$$E(\mathbf{p}) = \sum_i \psi_u(p_i) + \sum_i \sum_{j \neq i} \psi_p(p_i, p_j) \qquad (1)$$

where $\mathbf{p}$ are all the pixels in the image. The pairwise terms $\psi_p(p_i, p_j)$ are Gaussian edge potentials defined on image gradients in the RGB space, as used in [30]. The unary potential is defined as the sum of RGB $\psi_{rgb}(p_i)$ and depth $\psi_{depth}(p_i)$ based terms: $\psi_u(p_i) = \psi_{rgb}(p_i) + \psi_{depth}(p_i)$. The RGB term is defined as

$$\psi_{rgb}(p_i) = |H_{bg}(p_i) - H_t(p_i)| + |S_{bg}(p_i) - S_t(p_i)|, \quad (2)$$

where H and S are the HSV components of the color images. Empirically we found that the HSV color space leads to better results which are robust to illumination changes. The depth based unary term is a logistic function of the form of $\psi_{depth}(p_i) = 1 - 1/(1 + \exp(-(D_t(p_i) - D_{bg}(p_i))/\sigma))$, where σ controls the scale of the resolution and is fixed to $\sigma = 5$cm.

Obtaining the Maximum a Posterior (MAP) solution under the model defined in Eq. 1 is computationally expensive. To achieve real-time performance, we use the efficient algorithm for performing mean field inference that was proposed in [58], which we implemented efficiently on the GPU. In Fig. 4 we depict some results obtained using the proposed segmentation algorithm.

Temporally Consistent 3D Reconstruction

Given the $N = 8$ depth maps generated as described previously, we want to generate a 3D model of the region of interest. There are mainly 3 different strategies to achieve this. The simplest approach consists of visualizing the 3D model in the form of point clouds, however this will lead to multiple problems such as temporal flickering, holes and flying pixels (see Fig. 5).

A better strategy consists of fusing the data from all the cameras [23]. The fused depth maps generate a mesh per frame, with a reduced noise level and flying pixels compared to a simple point cloud visualization. However the main drawback of this solution is the absence of any temporal consistency: due to noise and holes in the depth maps, meshes generated at each frame could suffer from flickering effects, especially in difficult regions such as hair. This would lead to 3D models with high variation over time, making the overall experience less pleasant.

Motivated by these findings, we employ a state of the art method for generating temporally consistent 3D models in real-time [14]. This method tracks the mesh and fuses the data across cameras and frames. We summarize here the key steps, for additional details see [14].

In order to fuse the data temporally, we have to estimate the nonrigid motion field between frames. Following [14], we parameterize the nonrigid deformation using the embedded deformation (ED) model of [54]. We sample a set of K "ED

Figure 4. Top: RGB stream. Bottom: depth stream. Images are masked with the segmentation output.

Figure 5. Temporal consistency pipeline. Top: Point cloud visualization, notice flying pixels and holes. Bottom: temporally reconstructed meshes.

nodes" uniformly, at locations $\{\mathbf{g}_k\}_{k=1}^{K} \subseteq \mathbb{R}^3$ throughout the mesh $\mathcal{V}$. The local deformation around an ED node $\mathbf{g}_k$ is defined via an affine transformation $A_k \in \mathbb{R}^{3\times3}$ and a translation $\mathbf{t}_k \in \mathbb{R}^3$. In addition, a global rotation $R \in SO(3)$ and translation $T \in \mathbb{R}^3$ are added. The full parameters to estimate the nonrigid motion field are $G = \{R, T\} \cup \{A_k, \mathbf{t}_k\}_{k=1}^{K}$. The energy function we minimize is:

$$E(G) = \lambda_{data} E_{data}(G) + \lambda_{hull} E_{hull}(G) + \lambda_{corr} E_{corr}(G) + \lambda_{rot} E_{rot}(G) + \lambda_{reg} E_{reg}(G), \quad (3)$$

which is the weighted sum of multiple terms that take into account the data fidelity, visual hull constraints, sparse correspondences, a global rotation and a smoothness regularizer. All the details regarding these terms, including the implementation for solving this nonlinear problem, are in [14].

Once we retrieve the deformation model between two frames, we fuse the data of the tracked nonrigid parts and we reset the volume to the current observed 3D data for those regions where the tracking fails.

Color Texturing

After fusing depth data, we extract a polygonal 3D model from its implicit volumetric representation (TSDF) with marching cubes, and then texture this model using the 8 input RGB images (Fig. 6). A naive texturing approach would compute the color of each pixel by blending the RGB images

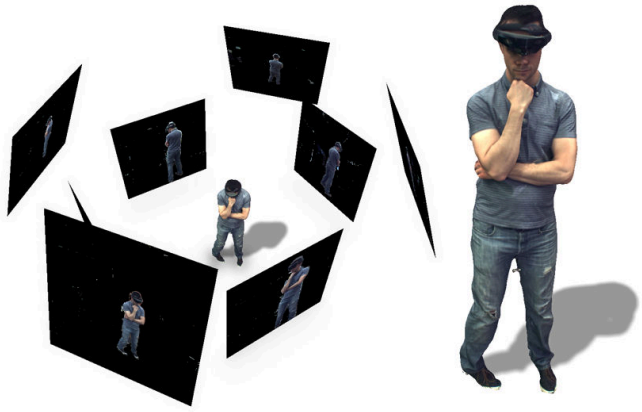

Figure 6. Projective Texturing: segmented-out RGB images and reconstructed color.

according to its surface normal $\hat{\mathbf{n}}$ and the camera viewpoint direction $\hat{\mathbf{v}}_i$. In this approach, the color weights are computed as $w_i = vis \cdot max(0, \hat{\mathbf{n}} \cdot \hat{\mathbf{v}}_i)^{\alpha}$, favoring frontal views with factor α and vis is a visibility test. These weights are non-zero if the textured point is visible in a particular view (i.e., not occluded by another portion of the model). The visibility test is needed to avoid back-projection and is done by rasterizing the model from each view, producing *rasterized depth maps* (to distinguish from *input depth maps*), and using those for depth comparison. Given imperfect geometry, this may result in so-called "ghosting" artifacts (Fig. 7a), when missing parts of geometry might cause wrongly projected color to the surfaces behind it. Many existing approaches use global optimization to stitch [18] or blend [64, 46] the incoming color maps to tackle this problem. However, these implementations are not real-time and in some cases use more RGB images.

Instead we assume that the surface reconstruction error is bounded by some value e. First, we extend the visibility test with an additional depth discontinuity test. At each textured point on a surface, for each input view we search for depth discontinuity in its projected 2D neighborhood in a rasterized depth map with radius determined by e. If such a discontinuity is found, we avoid using this view in a normal-weighted blending (the associated weight is 0). This causes dilation of edges in the rasterized depth map for each view (Fig. 7bc). This can eliminate most ghosting artifacts, however it can classify more points as unobserved. In this case, we use regular visibility tests and a majority voting scheme for colors:

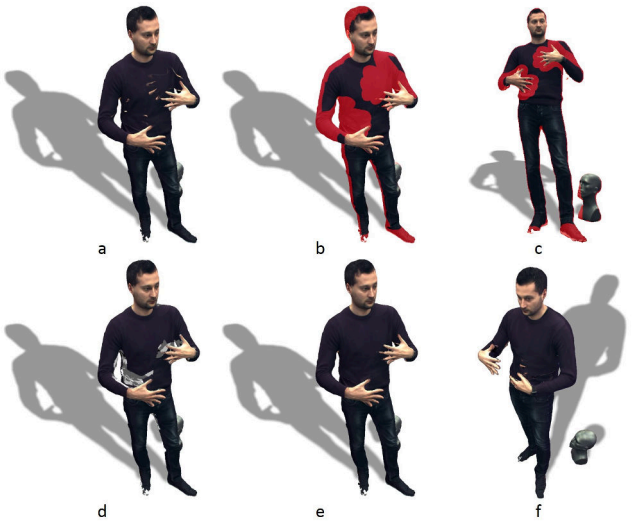

Figure 7. (a) Ghosting artifacts, (b) Dilated depth discontinuities, (c) Dilated depth discontinuities from pod camera view, (d) Number of agreeing color (white = 4), (e) Proposed algorithm, (f) Voting failure case.

for a given point, to classify each view as trusted, the color candidates for this view must agree with a number of colors (Fig. 7d) from other views that can see this point, and this number of agreeing views should be maximum. The agreement threshold is multi-level: we start from as small value and increase the threshold if no previous views agreed. This leads to a more accurate classification of trusted views as it helps to avoid false-positives near color discontinuities. Finally, if none of the views agree, we pick the view with the minimal RGB variance for a projected 2D patch in the input RGB image, since larger color discontinuity for a patch is more likely to correspond to a depth discontinuity of a perfectly reconstructed surface, and thus that patch should not be used.

While this approach works in real time and eliminates most of the artifacts (Fig. 7e), there are some failure cases when two ghosting color candidates can outvote one true color (Fig. 7f). This is a limitation of the algorithm, and a trade-off between quality and performance, but we empirically derived that this occurs only in highly occluded regions and does not reduce the fidelity of the color reconstruction in the region of interest, like faces. We also demonstrate that the algorithm can handle complex, multi-user scenarios with many objects (Fig. 8) using only 8 RGB cameras.

Spatial Audio

Audio is the foundational medium of human communication. Without proper audio, visual communication, however immersive, is usually ineffective. To enhance the sense of immersion, auditory and visual cues must match. If a user sees a person to the left, the user should also hear that person to the left. Audio emanating from a spatial source outside the user's field of view also helps to establish a sense of immersion, and can help to compensate for the limited field of view in some HMDs.

In Holoportation we synthesize each remote audio source, namely the audio captured from a remote user, as coming from the position and orientation of the remote user in the local user's space. This ensures that the audio and visual cues are matched. Even without visual cues, users can use the spatial audio cues to locate each other. For example, in Holoportation, it is possible for the first time for users to play the game "Marco Polo" while all players are geographically distributed. To the best of our knowledge, Holoportation is the first example of auditory augmented or virtual reality to enable communication with such freedom of movement.

In our system, each user is captured with a monaural microphone. Audio samples are chunked into 20ms frames, and audio frames are interleaved with the user's current head pose information in the user's local room coordinate system (specifically, x, y, z, yaw, pitch, and roll). The user's audio frame and the head pose information associated with that frame are transmitted to all remote users through multiple unicast connections, although multicast connections would also make sense where available.

Upon receiving an audio frame and its associated head pose information from a remote user, the local user's system transforms the head pose information from the remote user's room coordinate system to the local user's room coordinate system, and then spatializes the audio source at the proper location and orientation. Spatialization is accomplished by filtering the audio signal using a head related transfer function (HRTF).

HRTF is a mapping from each ear, each audio frequency, and each spatial location of the source relative to the ear, into a complex number representing the magnitude and phase of the attenuation of the audio frequency as the signal travels from the source location to the ear. Thus, sources far away will be attenuated more than sources that are close. Sources towards the left will be attenuated less (and delayed less) to the left ear than to the right ear, reproducing a sense of directionality in azimuth as well as distance. Sources above the horizontal head plane will attenuate frequencies differently, giving a sense of elevation. Details may be found in [20].

We also modify the amplitude of a source by its orientation relative to the listener, assuming a cardioid radiation pattern. Thus a remote user facing away from the local user will sound relatively muffled compared to a remote user facing toward the local user. To implement audio spatialization, we rely on the HRTF audio processing object in the XAUDIO2 framework in Windows 10.

Compression and Transmission

The previous steps generate an enormous amount of data per frame. Compression is therefore critical for transmitting that data over the wide area network to a remote location for rendering. Recent work in mesh compression [11] as well as point cloud compression [12] suggest that bit rates on the order of low tens of megabits per second per transmitted person, depending on resolution, are possible, although *real-time* compression at the lowest bit rates is still challenging.

For our current work, we wanted the rendering to be real time, and also of the highest quality. Hence we perform only a very lightweight real time compression and straightforward wire

Figure 8. Examples of texturing multiple users and objects.

format over TCP, which is enough to support 5-6 viewing clients between capture locations over a single 10 Gbps link in our point to point teleconferencing scenarios. For this purpose it suffices to transmit a standard triangle mesh with additional color information from the various camera viewpoints. To reduce the raw size of our data for real-time transmission, we perform several transformations on per-frame results from the capture and fusion stages, as we now describe.

Mesh Geometry

As described previously, we use a marching cubes polygonalization of the volumetric data generated in the fusion stage as our rendered mesh. The GPU implementation of marching cubes uses 5 millimeters voxels. We perform vertex deduplication and reduce position and normal data to 16 bit floats on the GPU before transfer to host memory for serialization. During conversion to our wire format, we use LZ4 compression on index data to further reduce frame size. For our current capture resolution this results in a per-frame transmission requirement of approximately 2MB per frame for mesh geometry (60K vertices, 40K triangles). Note that for proper compositing of remote content in AR scenarios, the reconstruction of local geometry must still be available to support occlusion testing.

Color Imagery

For color rendering, the projective texturing described above is used for both local and remote rendering, which requires that we transfer relevant color imagery from all eight color cameras. As we only need to transmit those sections of the image that contribute to the reconstructed foreground objects, we leverage the foreground segmentation calculated in earlier stages to set non-foreground regions to a constant background color value prior to transmission. With this in place, the LZ4 compression of the color imagery (8 4MP Bayer images) reduces the average per-frame storage requirements from 32MB to 3MB per frame. Calibration parameters for projective mapping from camera to model space may be transferred per frame, or only as needed.

Audio Transmission

For users wearing an AR or VR headset, the associated microphone audio and user pose streams are captured and transmitted to all remote participants to generate spatial audio sources

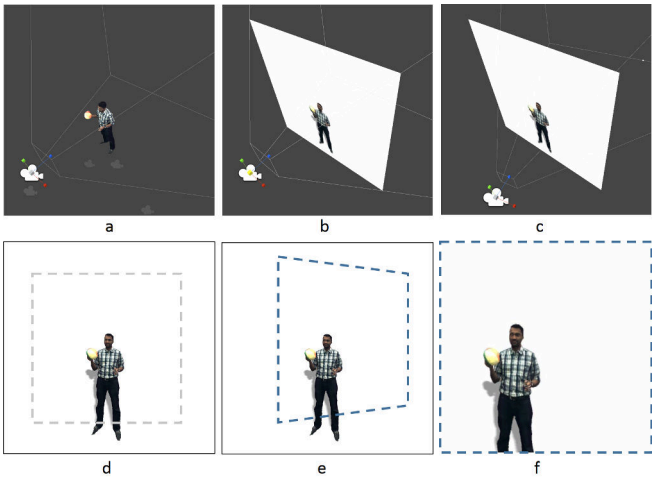

Figure 9. Render Offloading: (a) Rendering with predicted pose on PC, (b) Rendered image encoding, (c) Rendering with actual pose on HMD, (d) Predicted view on PC, and its over-rendering with enlarged FOV, (e) Pixels in decoded video stream, used during reprojection, (f) Reprojected actual view.

correlated to their rendered location, as described in the Audio subsection above. Each captured audio stream is monaural, sampled at 11 KHz, and represented as 16-bit PCM, resulting in a transmitted stream of 11000*16 = 176 Kbps, plus 9.6 Kbps for the pose information. At the receiving side, the audio is buffered for playback. Network jitter can cause the buffer to shrink when packets are not being delivered on time, and then to grow again when they are delivered in a burst. If the buffer underflows (becomes empty), zero samples are inserted. There is no buffer overflow, but the audio playback rate is set to 11025 samples per second, to provide a slight downward force on the playback buffer size. This keeps the playback buffer in check even if the receiving clock is somewhat slower than the sending clock. The audio+pose data is transmitted independently and bi-directionally between all pairs of remote participants. The audio communication system is peer-to-peer, directly between headsets, and is completely independent of the visual communication system. We do not provide AV sync, and find that the audio and visual reproductions are sufficiently synchronized; any difference in delay between audio and visual systems is not noticeable.

Bandwidth and Network Topology

For low-latency scenarios, these approaches reduce the average per-frame transmission size to a 1-2 Gbps transfer rate for an average capture stream at 30fps, while adding only a small overhead (<10ms) for compression. Compressed frames of mesh and color data are transmitted between capture stations via TCP to each active rendering client. For 'Living Memories' scenarios, described later in the applications section, these packets may be intercepted, stored, and replayed by an intermediary recording server. Large one-to-many broadcast scenarios, such as music or sports performance, requires further compression and multicast infrastructures.

Render Offloading and Latency Compensation

For untethered VR or AR HMDs, like HoloLens, the cost of rendering a detailed, high quality 3D model from the user's

perspective natively on the device can be considerable. It also increases motion-to-photon latency, which worsens the user experience. We mitigate this by offloading the rendering costs to a dedicated desktop PC on the receiving side, i.e., perform remote rendering, similar to [41, 34]. This allows us to maintain consistent framerate, to reduce perceived latency, and to conserve battery life on the device, while also enabling high-end rendering capabilities, which are not always available on mobile GPUs.

Offloading is done as follows: the rendering PC is connected to an untethered HMD via WiFi, and constantly receives user's 6DoF (six degree of freedom) *pose* from the HMD. It predicts a headset pose at render time and performs scene rendering with that pose for each eye (Fig. 9a), encodes them to a video stream (Fig. 9b), and transmits the stream and poses to the HMD. There, the video stream is decoded and displayed for each eye as a textured quad, positioned based on predicted rendered pose (Fig. 9c), and then reprojected to the latest user pose (Fig. 9f) [60].

To compensate for pose mispredictions and PC-to-HMD latency, we perform speculative rendering on the desktop side, based on the likelihood of the user pose. The orientation misprediction can be compensated by rendering into a larger FoV (field of view) (Fig. 9d), centered around the predicted user direction. The HMD renders the textured quad with actual display FoV, thus allowing some small misprediction in rotation (Fig. 9e). To handle positional misprediction, we could perform view interpolation techniques as in [34]. However, it would require streaming the scene depth buffer and its reprojection, which would increase the bandwidth and HMD-side rendering costs. Since the number of objects are small and they are localized in the capture cube, we approximate the scene depth complexity with geometry of the same (textured) quad and dynamically adjust its distance to the user, based on the point of interest.

IMPLEMENTATION

We use multiple GPUs across multiple machines to achieve real-time performance. At each capture site, 4 PCs compute depth and foreground segmentation (each PC handles two capture pods). Each PC is an Intel Core i7 3.4 Ghz CPU, 16 GB of RAM and it uses two NVIDIA Titan X GPUs. The resulting depth maps and segmented color images are then transmitted to a dedicated dual-GPU fusion machine over point-to-point 10Gbps connections. The dual-GPU unit is an Intel Core i7 3.4GHz CPU, 16GB of RAM, with two NVIDIA Titan X GPUs.

Depth Estimation: GPU Implementation

The depth estimation algorithm we use in this work comprises of three main stages: initialization, propagation, and filtering. Most of these stages are highly parallelizable and its computation is pixel independent (initialization). Therefore, a huge benefit in terms of compute can be obtained by implementing this directly on the GPU. Porting PatchMatch stereo [8] to the GPU required a modification of the propagation stage in comparison with the original CPU implementation. The original propagation stage is inherently iterative and is performed in row order starting at the top-left pixel to the end of the image and in the reverse order, iterating twice over the image. Our GPU implementation modifies this step in order to take advantage of the massive parallelism of current GPUs. The image is subdivided into neighborhoods of regular size and the propagation happens locally on each of these neighborhoods. With this optimization, whole rows and columns can be processed independently on separate threads in the GPU. In this case, the algorithm iterates through four propagation directions in the local neighborhood: from left to right, top to bottom, right to left, and bottom to top.

Finally, the filtering stage handles the removal of isolated regions that do not represent correct disparity values. For this stage, we implemented a Connected Components labeling algorithm on the GPU, so regions below a certain size are removed. This step is followed by an efficient GPU-based median algorithm to remove noisy disparity values while preserving edges.

Temporal Reconstruction: Dual-GPU Implementation

The implementation of [14] is the most computationally expensive part of the system. Although with [14] we can reconstruct multiple people in real time, room-sized dynamic scene reconstructions require additional compute power. Therefore, we had to revisit the algorithm proposed in [14] and implement a novel dual-GPU scheme. We pipeline [14] into two parts, where each part lives on a dedicated GPU. The first GPU estimates a low resolution frame-to-frame motion field; the second GPU refines the motion field and performs the volumetric data fusion. The coarse frame-to-frame motion field is computed from the raw data (depth and RGB). The second GPU uses the frame-to-frame motion to initialize the model-to-frame motion estimation, which is then used to fuse the data volumetrically. Frame-to-frame motion estimation does not require the feedback from the second GPU, so two GPUs can run in parallel. For this part we use a coarser deformation graph (i.e. we sample an ED node every 8 cm) and run 8 Levenberg-Marquardt (LM) solver iterations, with coarser voxel resolution and only 2 iterations are used for model-to-frame motion estimation.

Throughput-oriented Architecture

We avoid stalls between CPU and GPU in each part of our pipeline by using pinned (non-pageable) memory for CPU-GPU DMA data transfers, organized as ring buffers; overlapping data upload, download, and compute; and using sync-free kernel launches, maintaining relevant subproblem sizes on the GPU, having only its max-bounds estimations on the CPU. We found that overhead of using max-bounds is lower than with CUDA Dynamic Parallelism for nested kernel launches. Ring buffers introduce only processing latency into the system and maintain the throughput we want to preserve.

Computational Time

Image acquisition happens in a separate thread overlapping the computing of the previous frame with the acquisition of

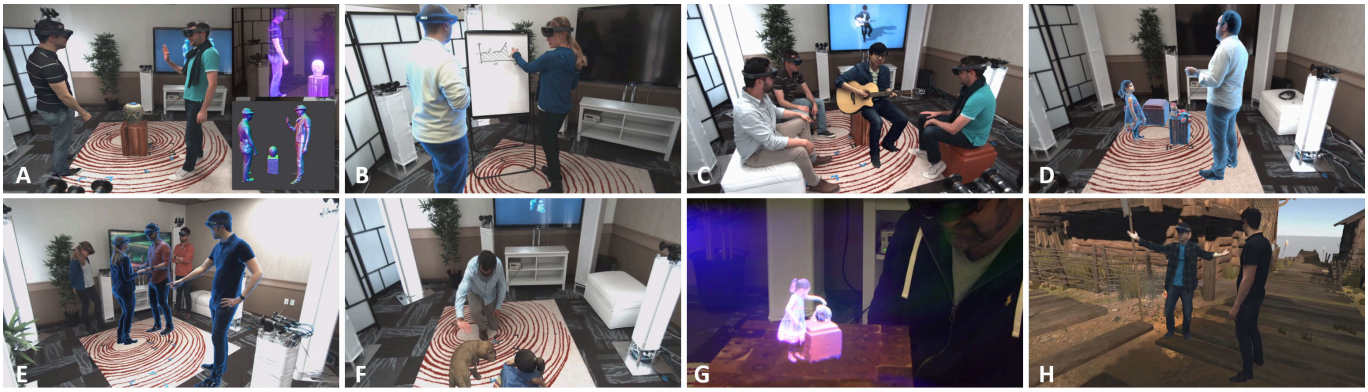

Figure 10. Applications. A) One-to-one communication. B) Business meeting. C) Live concert broadcasting. D) Living memory. E) Out-of-body dancing instructions. F) Family gathering. G) Miniature view. H) Social VR/AR.

the next one and introducing 1 frame latency. The average time for image acquisition is 4ms.

Each machine on the capture side generates two depth maps and two segmentation masks in parallel. The total time is 21ms and 4ms for the stereo matching and segmentation, respectively. In total each machine uses no more than 25ms to generate the input for the multiview non-rigid 3D reconstruction pipeline. Segmented RGBD frames are generated in parallel to the nonrigid pipeline, but do also introduce 1 frame of latency.

A master PC, aggregates and synchronizes all the depth maps and segmentation masks. Once the RGBD inputs are available, the average processing time (second GPU) to fuse all the depth maps is 29ms (i.e., 34fps) with 5ms for preprocessing (18% of the whole fusion pipeline), 14ms (47%) for the nonrigid registration (2 LM iterations, with 10 PCG iterations), and 10ms (35%) for the fusion stage. The visualization for a single eye takes 6ms on desktop GPU, and thus enables to display graphics at native refresh rate of 60Hz for HoloLens and 75Hz for Oculus Rift DK2.

APPLICATIONS

As shown in the supplementary video and Fig. 10, we envision many applications for Holoportation. These fall into two broad categories. First, are one-to-one applications: these are scenarios where two remote capture rigs establish a direct connection so that the virtual and physical spaces of each rig are in one-to-one correspondence. Once segmentation is performed any new object in one space will appear in the other and vice versa. As shown in the video, this allows remote participants to be correctly occluded by the objects in the local space. Because our system is agnostic to *what* is actually captured, objects, props, and even furniture can be captured.

One to one calls are analogous to a telephone or video chat between two parties. However, with the ability to move around the space, and benefit from many physical cues. The second category consists of one-to-many applications: this is where a single capture rig is broadcasting a live stream of data to many receivers. In this case the physical space within the capture rig corresponds to the many virtual spaces of the remote viewers. This is analogous to the broadcast television model, where a single image is displayed on countless screens.

One-to-one

One-to-one applications are communication and collaboration scenarios. This could be as simple as a remote conversation. The distinguishing characteristic that Holoportation brings is a sense of physical presence through cues such as movement in space, gestures, and body language.

Specific one-to-one applications we foresee include:

- Family gatherings, where a remote participant could visit with loved ones or join a birthday celebration.
- Personal instruction, where a remote teacher could provide immediate feedback on dance moves, or yoga poses.
- Doctor patient, where a remote doctor could examine and interact with a patient to provide a diagnosis.
- Design meetings, where remote parties could share and interact with physical or virtual objects.

One-to-many

A unique characteristic of Holoportation for one-to-many scenarios is *live* streaming. Other technologies can broadcast pre-processed content, as in Collet et al [11]. We believe that live streaming provides a much more powerful sense of engagement and shared experience. Imagine a concert viewable from any seat in the house, or even from on stage. Similarly, live sporting events could be watched, or re-watched, from any point. When humans land on Mars, imagine waiting on the surface to greet the astronaut as she steps foot on the planet.

In both one-to-one or one-to many scenarios, Holoportation content can be recorded for replay from any viewpoint, and any scale. Room-sized events can be scaled to fit on a coffee table for comfortable viewing. Content can be paused, rewound, fast forwarded, to view any event of interest, from any point of view.

Beyond Augmented Reality: A Body in VR

Finally, we point out that Holoportation can also be used in a single or multi player immersive VR experiences, where one's own body is captured by the Holoportation system and inserted into the VR world in real-time. This provides a virtual body that moves and appears like ones own physical body, providing a sense of presence.

As shown in the supplementary video, these experiences capture many new possibilities for 'social' AR and VR applications in the future.

USER STUDY

Having solved many of the technical challenges for real-time high quality immersive telepresence experiences, we wanted to test our prototype in a preliminary user study. We aimed to unveil opportunities and challenges for interaction design, explore technological requirements, and better understand the tradeoffs of AR and VR for this type of communication.

Procedure and Tasks

We recruited a total of 10 participants (3 females; age 22 to 56) from a pool of partners and research lab members not involved with this project. In each study session, two participants were placed in different rooms and asked to perform two tasks (social interaction and physical object manipulation) using each of the target technologies, AR and VR. We counterbalanced the order of technology conditions as well as the order of the tasks using the Latin square design. The whole procedure took approximately 30 minutes. For the duration of the study, two researchers observed and recorded participants' behavior, including their interaction patterns, body language, and strategies for collaboration in the shared space. After the study, researchers also conducted semi-structured interviews to obtain additional insight about the most salient elements of the users' experience, challenges, and potential usage scenarios.

Social Interaction Task: Tell-a-lie

To observe how people use Holoportation for verbal communication, we used a tell-a-lie task [66]. All participants were asked to state three pieces of information about themselves, with one of the statements being false. The goal of the partner was to identify the false fact by asking any five questions. Both deception and deception detection are social behaviors that encourage people to pay attention to and accurately interpret verbal and non-verbal communication. Thus, it presents a very conservative testing scenario for our technology.

Physical Interaction Task: Building Blocks

To explore the use of technology for physical interaction in the shared workspace, we also designed an object manipulation task. Participants were asked to collaborate in AR and VR to arrange six 3D objects (blocks, cylinders, etc.) in a given configuration (Fig. 11). Each participant had only three physical objects in front of him on a stool, and could see the blocks of the other person virtually. During each task, only one of the participants had a picture of the target layout, and had to instruct the partner.

Study Results and Discussion

Prior to analysis, researchers compared observation notes and partial transcripts of interviews for congruency. Further qualitative analysis revealed insights that fall into 5 categories.

Adapting to Mixed-reality Setups

Participants experienced feelings of spatial and social co-presence with their remote partners, which made their interaction more seamless. P4: *"It's way better than phone calls.*

Figure 11. User study setting: Two participants performing the building blocks task in an AR condition (left) and a VR condition (right).

[...] Because you feel like you're really interacting with the person. It's also better than a video chat, because I feel like we can interact in the same physical space and that we are modifying the same reality."

The spatial and auditory cues gave an undeniable sense of co-location; so much so that many users even reported a strong sense of *interpersonal space* awareness. For example, most participants showed non-verbal indicators/body language typical to face-to-face conversations in real life (e.g. adopting the "closed" posture when lying in the social task; automatically using gestures to point or leaning towards each other in the building task). P6: *"It made me conscious of [my own image]. It was kind of realistic in that sense, because whenever we are talking to someone around the table we are conscious of not wanting to slouch in front of the other person [and] about how we look to the other person."*

Participants also quickly adapted and developed strategies for collaborating in the mixed-reality setup. For example, several participants independently came up with the idea to remove their own blocks to let their partner arrange their shapes first, to avoid confusing the real and virtual objects. While participants often started by verbally instructing and simple pointing, they quickly figured out that it is easier and more natural for them to use gestures or even their own objects or hands to show the intended position and orientation. P2: *"When I started I was kind of pointing at the shape, then I was just doing the same with my shape and then just say 'Ok, do that. Because then she could visually see, I could almost leave my shapes and wait until she put her shapes exactly in the same location and then remove mine."*

Natural Interaction for Physical and Collaborative tasks

The shared spatial frame of reference, being more natural than spatial references in a video conversation, was mentioned as a main advantage of Holoportation. For example, P9 found that *"[the best thing was] being able to interact remotely and work together in a shared space. In video, in Skype, even just showing something is still a problem: you need to align [the object] with the camera, then 'Can you see it?', then 'Can you turn it?' Here it's easy."*

The perception of being in the same physical space also allows people to interact simply more naturally, even for full body interaction, as nicely described by P4: *"I'm a dancer [...] and we had times when we tried to have Skype rehearsals. It's really hard, because you're in separate rooms completely. There's no interaction, they might be flipped, they might not. Instead with this, it's like I could look and say 'ok, this is his right' so I'm going to tell him move it to the right or towards me."*

Influence of Viewer Technology

To probe for the potential uses of Holoportation, we used two viewer technologies with different qualities and drawbacks. Characteristics of the technology (e.g. FoV, latency, image transparency, and others) highly influenced participants' experiences, and, specifically, their feelings of social and spatial co-presence. To some participants, AR felt more realistic/natural because it integrated a new image with their own physical space. In the AR condition, they felt that their partner was "here", coming to their space as described by P7 *"In the AR it felt like somebody was transported to the room where I was, there were times I was forgetting they were in a different room."* The VR condition gave participants the impression of being in the partner's space, or another space altogether. P6: *"[With VR] I felt like I was in a totally different place and it wasn't all connected to my actual surroundings."*

Another interesting finding in the VR condition was the confusion caused by seeing a replica of both local and remote participants and objects. There were many instances where users failed to determine if the block they were about to touch was real or not, passing their hands directly through the virtual object. Obviously, rendering effects could help in this disambiguation, although if these are subtle then they may still cause confusions.

Related to AR and VR conditions was also the effects of latency. In VR the user's body suffered from latency due to our pipeline. Participants could tolerate the latency in terms of interacting and picking up objects, but a few users suffered from discomfort during the VR condition, which was not perceived in the AR condition. Despite this we were surprised how well participants performed in VR and how they commented on enjoying the greater sense of presence due to seeing their own bodies.

Freedom to Choose a Point of View

One of the big benefits mentioned, was that the technology allows the viewer to assume the view they wanted, and move freely without constraints. This makes it very suitable for exploring objects or environments, or to explain complicated tasks that benefit from spatial instructions. P10 commented: *"I do a lot of design reviews or things like that where you actually want to show somebody something and then explain in detail and it's a lot easier if you could both see the same things. So if you have a physical object and you want to show somebody the physical object and point to it, it just makes it a lot easier."* However, this freedom of choice might be less suitable in case of a "directors narrative" in which the presenter wants to control the content and the view on it.

Requirements for Visual Quality

While a quantitative analysis of small sample sizes needs careful interpretation, we found that most participants (70%) tended to agree or strongly agree that they perceived the conversation partner to look like a real person. However, some feedback about visual quality was less positive; e.g., occasional flicker that occurred when people were at the edge of the reconstruction volume was perceived to be off-putting.

Eye Contact through the Visor

One key factor that was raised during the study was the presence of the headset in both AR and VR conditions, which clearly impacted the ability to make direct eye contact. Whilst we perceived many aspects of physical co-located interaction, users did mention this as a clear limitation.

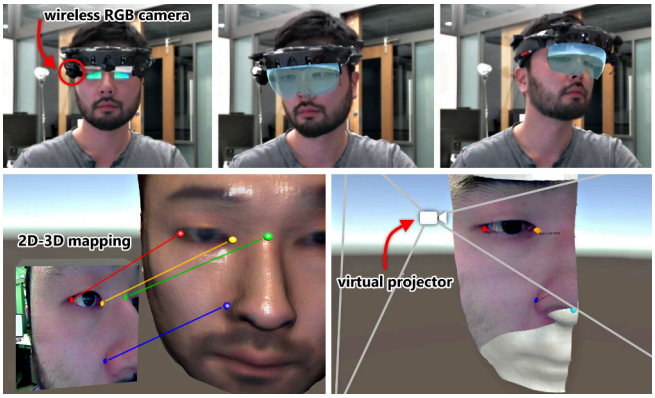

Figure 12. Visor Removal. The left image in the top row shows the placement of the inward looking cameras on the HMD while those on the center and right show the projections. The bottom row shows the facial landmarks localized by our pipeline to enable the projective texturing.

To overcome this problem, we have begun to investigate a proof-of-concept prototype for augmenting existing see-through displays with headset removal. We utilize tiny wireless video cameras, one for each eye region that is mounted on the outer rim of the display (see Fig. 12). We then project the texture associated with the occluded eye-region to a 3D reconstruction of the user's face.

To perform the projective texturing, we need to not only estimate the intrinsics calibration of the camera, but also the extrinsic parameters, which in this instance corresponds to the 3D position of the camera with respect to the face of the user. Given the 3D position of facial keypoints (e.g. eye corners, wings of the nose) in an image, it is possible to estimate the extrinsics of the camera by leveraging the 2D-3D position of these landmarks using optimization methods like [35]. We use a discriminately trained cascaded Random Forest to generate series of predictions of the most likely position of facial keypoints in a manner similar to [50]. To provide predictions that are spatially smooth over time, we perform a mean-shift filtering of the predictions made by Random Forest.

The method we just described requires the 3D location of keypoints to estimate the camera extrinsics. We extract this information from a high fidelity reconstruction of the user's face obtained by aggregating frames using KinectFusion [23]. This process requires human intervention, but only takes a few seconds to complete. Note that once we have the 3D model, we render views (from known camera poses) and pass these to our Random Forest to estimate the 2D location of the facial keypoints. Knowing the camera poses used to render, we can ray-cast and estimate where the keypoints lie on the user's 3D face model. We also perform color correction to make sure that the texture coming from both cameras look

compatible. At this stage, we have all the components to perform real-time projective texture mapping of eye camera images onto 3D face geometry and to perform geometric mesh blending of the eye region and the live mesh. Our preliminary results are shown in Fig. 12 and more results are presented in the attached video.

LIMITATIONS

While we present the first high quality 360^o immersive 3D telepresence system, our work does have limitations. The amount of high-end hardware required to run the system is high. Currently, a 10 Gigabit Ethernet connection is used to communicate between rooms, allowing low latency communication between the users. More efficient compressions schemes need to be developed to address this requirement and to be able to compress mesh and color information into 1 Gigabit connections. Moreover, during the novel viewpoint color rendering process we observed some color artifacts caused by extreme occlusions, multiple people and furniture in the scene. More complex approaches that take advantage of the non-rigid tracking of the mesh could be developed to further reduce the artifacts and the number of color cameras that are required. Regarding the 3D non-rigid reconstruction, we note that during some of the social interaction tasks, the 3D reconstruction of the hands produced some artifacts, like missing fingers caused by occlusions during the manipulation of the objects. This is an artifact of the limited number of viewpoints of the depth cameras. Finally, it is worth noting the necessity of developing algorithms for headset removal and realistic eye in-painting since wearing VR and AR headsets clearly impacted the ability to make direct eye contact.

CONCLUSION

We have presented Holoportation: a new end-to-end system for high-quality real-time capture, transmission and rendering of people, spaces, and objects in full 3D. We combine a new capture technology with mixed reality displays to allow users to see, hear and interact in 3D with remote colleagues. This creates an experience akin to physical presence. We have demonstrated many different interactive scenarios, from one-to-one communication and one-to-many broadcast scenarios, which capture live/real-time interaction, to record and playback, which creates 'living memories'. We hope that practitioners and researchers will begin to expand the application space further, leveraging new possibilities enabled by this type of live 3D capture.

REFERENCES

1. Balogh, T., and Kovács, P. T. Real-time 3d light field transmission. In *SPIE Photonics Europe*, International Society for Optics and Photonics (2010), 772406–772406.
2. Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM SIGGRAPH and Transaction On Graphics* (2009).
3. Batlle, J., Mouaddib, E., and Salvi, J. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition 31*, 7 (1998), 963–982.
4. Beck, S., Kunert, A., Kulik, A., and Froehlich, B. Immersive group-to-group telepresence. *Visualization and Computer Graphics, IEEE Transactions on 19*, 4 (2013), 616–625.
5. Benko, H., Jota, R., and Wilson, A. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proceedings of the SIGCHI conference on human factors in computing systems*, ACM (2012), 199–208.
6. Besl, P. J. Active, optical range imaging sensors. *Machine vision and applications 1*, 2 (1988), 127–152.
7. Blanche, P.-A., Bablumian, A., Voorakaranam, R., Christenson, C., Lin, W., Gu, T., Flores, D., Wang, P., Hsieh, W.-Y., Kathaperumal, M., et al. Holographic three-dimensional telepresence using large-area photorefractive polymer. *Nature 468*, 7320 (2010), 80–83.
8. Bleyer, M., Rhemann, C., and Rother, C. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *BMVC* (2011).
9. Bogo, F., Black, M. J., Loper, M., and Romero, J. Detailed full-body reconstructions of moving people from monocular rgb-d sequences. In *Proceedings of the IEEE International Conference on Computer Vision* (2015), 2300–2308.
10. Chen, W.-C., Towles, H., Nyland, L., Welch, G., and Fuchs, H. Toward a compelling sensation of telepresence: Demonstrating a portal to a distant (static) office. In *Proceedings of the conference on Visualization'00*, IEEE Computer Society Press (2000), 327–333.
11. Collet, A., Chuang, M., Sweeney, P., Gillett, D., Evseev, D., Calabrese, D., Hoppe, H., Kirk, A., and Sullivan, S. High-quality streamable free-viewpoint video. *ACM TOG 34*, 4 (2015), 69.
12. de Queiroz, R., and Chou, P. A. Compression of 3d point clouds using a region-adaptive hierarchical transform. *Transactions on Image Processing* (2016). To appear.
13. Dou, M., and Fuchs, H. Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In *Virtual Reality (VR), 2014 iEEE*, IEEE (2014), 39–44.
14. Dou, M., Khamis, S., Degtyarev, Y., Davidson, P., Fanello, S. R., Kowdle, A., Escolano, S. O., Rhemann, C., Kim, D., Taylor, J., Kohli, P., Tankovich, V., and Izadi, S. Fusion4d: Real-time performance capture of challenging scenes. *ACM Trans. Graph. 35*, 4 (July 2016), 114:1–114:13.
15. Fanello, S., Rhemann, C., Tankovich, V., Kowdle, A., Orts Escolano, S., Kim, D., and Izadi, S. Hyperdepth: Learning depth from structured light without matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2016).

16. Fuchs, H., Bazin, J.-C., et al. Immersive 3d telepresence. *Computer*, 7 (2014), 46–52.

17. Fuchs, H., Bishop, G., Arthur, K., McMillan, L., Bajcsy, R., Lee, S., Farid, H., and Kanade, T. Virtual space teleconferencing using a sea of cameras. In *Proc. First International Conference on Medical Robotics and Computer Assisted Surgery*, vol. 26 (1994).

18. Gal, R., Wexler, Y., Ofek, E., Hoppe, H., and Cohen-Or, D. Seamless montage for texturing models. In *Computer Graphics Forum*, vol. 29, Wiley Online Library (2010), 479–486.

19. Gibbs, S. J., Arapis, C., and Breiteneder, C. J. Teleport–towards immersive copresence. *Multimedia Systems 7*, 3 (1999), 214–221.

20. Gilkey, R. H., and Anderson, T. R., Eds. *Binaural and Spatial Hearing in Real and Virtual Environments*. Psychology Press, 2009.

21. Gross, M., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-Meier, E., Svoboda, T., Van Gool, L., Lang, S., et al. blue-c: a spatially immersive display and 3d video portal for telepresence. In *ACM Transactions on Graphics (TOG)*, vol. 22, ACM (2003), 819–827.

22. Hansard, M., Lee, S., Choi, O., and Horaud, R. P. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012.

23. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R. A., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A. J., and Fitzgibbon, A. W. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *UIST 2011, Santa Barbara, CA, USA, October 16-19, 2011* (2011), 559–568.

24. Jones, A., Lang, M., Fyffe, G., Yu, X., Busch, J., McDowall, I., Bolas, M., and Debevec, P. Achieving eye contact in a one-to-many 3d video teleconferencing system. *ACM Transactions on Graphics (TOG) 28*, 3 (2009), 64.

25. Jones, B., Sodhi, R., Murdock, M., Mehra, R., Benko, H., Wilson, A., Ofek, E., MacIntyre, B., Raghuvanshi, N., and Shapira, L. Roomalive: Magical experiences enabled by scalable, adaptive projector-camera units. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, ACM (2014), 637–644.

26. Jouppi, N. P. First steps towards mutually-immersive mobile telepresence. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, ACM (2002), 354–363.

27. Kanade, T., Rander, P., and Narayanan, P. Virtualized reality: Constructing virtual worlds from real scenes. *IEEE multimedia*, 1 (1997), 34–47.

28. Kim, K., Bolton, J., Girouard, A., Cooperstock, J., and Vertegaal, R. Telehuman: effects of 3d perspective on gaze and pose estimation with a life-size cylindrical telepresence pod. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ACM (2012), 2531–2540.

29. Kohli, P., Rihan, J., Bray, M., and Torr, P. H. S. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *IJCV 79*, 3 (2008), 285–298.

30. Krähenbühl, P., and Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in Neural Information Processing Systems: 25th Annual Conference on Neural Information Processing Systems 2011. Granada, Spain.* (2011), 109–117.

31. Kurillo, G., Bajcsy, R., Nahrsted, K., and Kreylos, O. Immersive 3d environment for remote collaboration and training of physical activities. In *Virtual Reality Conference, 2008. VR'08. IEEE*, IEEE (2008), 269–270.

32. Kuster, C., Ranieri, N., Agustina, Zimmer, H., Bazin, J. C., Sun, C., Popa, T., and Gross, M. Towards next generation 3d teleconferencing systems. 1–4.

33. Kuster, C., Ranieri, N., Zimmer, H., Bazin, J., Sun, C., Popa, T., Gross, M., et al. Towards next generation 3d teleconferencing systems. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, IEEE (2012), 1–4.

34. Lee, K., Chu, D., Cuervo, E., Kopf, J., Degtyarev, Y., Grizan, S., Wolman, A., and Flinn, J. Outatime: Using speculation to enable low-latency continuous interaction for mobile cloud gaming. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, ACM (2015), 151–165.

35. Lepetit, V., Moreno-Noguer, F., and Fua, P. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision 81*, 2 (Feb. 2009).

36. Loop, C., Zhang, C., and Zhang, Z. Real-time high-resolution sparse voxelization with application to image-based modeling. In *Proceedings of the 5th High-Performance Graphics Conference*, ACM (2013), 73–79.

37. Luff, P., and Heath, C. Mobility in collaboration. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, ACM (1998), 305–314.

38. Maimone, A., and Fuchs, H. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symposium on*, IEEE (2011), 137–146.

39. Maimone, A., and Fuchs, H. Real-time volumetric 3d capture of room-sized scenes for telepresence. In *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), 2012*, IEEE (2012), 1–4.

40. Maimone, A., Yang, X., Dierk, N., State, A., Dou, M., and Fuchs, H. General-purpose telepresence with head-worn optical see-through displays and projector-based lighting. In *Virtual Reality (VR), 2013 IEEE*, IEEE (2013), 23–26.

41. Mark, W. R., McMillan, L., and Bishop, G. Post-rendering 3d warping. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, ACM (1997), 7–ff.

42. Matusik, W., and Pfister, H. 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. In *ACM Transactions on Graphics (TOG)*, vol. 23, ACM (2004), 814–824.

43. Molyneaux, D., Izadi, S., Kim, D., Hilliges, O., Hodges, S., Cao, X., Butler, A., and Gellersen, H. Interactive environment-aware handheld projectors for pervasive computing spaces. In *Pervasive Computing*. Springer, 2012, 197–215.

44. Mori, M., MacDorman, K. F., and Kageki, N. The uncanny valley [from the field]. *Robotics & Automation Magazine, IEEE 19*, 2 (2012), 98–100.

45. Nagano, K., Jones, A., Liu, J., Busch, J., Yu, X., Bolas, M., and Debevec, P. An autostereoscopic projector array optimized for 3d facial display. In *ACM SIGGRAPH 2013 Emerging Technologies*, ACM (2013), 3.

46. Narayan, K. S., and Abbeel, P. Optimized color models for high-quality 3d scanning. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, IEEE (2015), 2503–2510.

47. Pejsa, T., Kantor, J., Benko, H., Ofek, E., and Wilson, A. D. Room2room: Enabling life-size telepresence in a projected augmented reality environment. In *CSCW 2016, San Francisco, CA, USA, February 27 - March 2, 2016*, D. Gergle, M. R. Morris, P. Bjrn, and J. A. Konstan, Eds., ACM (2016), 1714–1723.

48. Petit, B., Lesage, J.-D., Menier, C., Allard, J., Franco, J.-S., Raffin, B., Boyer, E., and Faure, F. Multicamera real-time 3d modeling for telepresence and remote collaboration. *International Journal of Digital Multimedia Broadcasting 2010* (2009).

49. Posdamer, J., and Altschuler, M. Surface measurement by space-encoded projected beam systems. *Computer graphics and image processing 18*, 1 (1982), 1–17.

50. Ren, S., Cao, X., Wei, Y., and Sun, J. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), 1685–1692.

51. Rhemann, C., Hosni, A., Bleyer, M., Rother, C., and Gelautz, M. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR* (2011).

52. S. Kosov, T. T., and Seidel, H.-P. Accurate real-time disparity estimation with variational methods. In *ISVC* (2009).

53. Scharstein, D., and Szeliski, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision 47*, 1-3 (Apr. 2002), 7–42.

54. Sumner, R. W., Schmid, J., and Pauly, M. Embedded deformation for shape manipulation. *ACM TOG 26*, 3 (2007), 80.

55. Tanikawa, T., Suzuki, Y., Hirota, K., and Hirose, M. Real world video avatar: real-time and real-size transmission and presentation of human figure. In *Proceedings of the 2005 international conference on Augmented tele-existence*, ACM (2005), 112–118.

56. Tombari, F., Mattoccia, S., Stefano, L. D., and Addimanda, E. Near real-time stereo based on effective cost aggregation. In *ICPR* (2008).

57. Towles, H., Chen, W.-C., Yang, R., Kum, S.-U., Kelshikar, H. F. N., Mulligan, J., Daniilidis, K., Fuchs, H., Hill, C. C., Mulligan, N. K. J., et al. 3d tele-collaboration over internet2. In *In: International Workshop on Immersive Telepresence, Juan Les Pins*, Citeseer (2002).

58. Vineet, V., Warrell, J., and Torr, P. H. S. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. In *ECCV 2012 - 12th European Conference on Computer Vision*, vol. 7576, Springer (2012), 31–44.

59. Will, P. M., and Pennington, K. S. Grid coding: A preprocessing technique for robot and machine vision. *Artificial Intelligence 2*, 3 (1972), 319–329.

60. Williams, O., Barham, P., Isard, M., Wong, T., Woo, K., Klein, G., Service, D., Michail, A., Pearson, A., Shetter, M., et al. Late stage reprojection, Jan. 29 2015. US Patent App. 13/951,351.

61. Yang, Q., Wang, L., Yang, R., Wang, S., Liao, M., and Nistr, D. Real-time global stereo matching using hierarchical belief propagation. In *BMVC* (2006).

62. Zhang, C., Cai, Q., Chou, P. A., Zhang, Z., and Martin-Brualla, R. Viewport: A distributed, immersive teleconferencing system with infrared dot pattern. *IEEE Multimedia 20*, 1 (2013), 17–27.

63. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell. 22*, 11 (Nov. 2000), 1330–1334.

64. Zhou, Q.-Y., and Koltun, V. Color map optimization for 3d reconstruction with consumer depth cameras. *ACM Transactions on Graphics (TOG) 33*, 4 (2014), 155.

65. Zollhöfer, M., Nießner, M., Izadi, S., Rhemann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., and Stamminger, M. Real-time non-rigid reconstruction using an rgb-d camera. *ACM Transactions on Graphics (TOG) 33*, 4 (2014).

66. Zuckerman, M., DePaulo, B. M., and Rosenthal, R. Verbal and nonverbal communication of deception. *Advances in experimental social psychology 14*, 1 (1981), 59.