Immersive Group-to-Group Telepresence

Stephan Beck, André Kunert, Alexander Kulik, and Bernd Froehlich, *Member, IEEE*

Fig. 1. Two groups of users meet virtually while being surrounded by a virtual city. On the left the local users greet the life-size 3D representations of the remote users. On the right the two groups discuss the tower of a church in a WIM by pointing and gesturing.

Abstract—We present a novel immersive telepresence system that allows distributed groups of users to meet in a shared virtual 3D world. Our approach is based on two coupled projection-based multi-user setups, each providing multiple users with perspectively correct stereoscopic images. At each site the users and their local interaction space are continuously captured using a cluster of registered depth and color cameras. The captured 3D information is transferred to the respective other location, where the remote participants are virtually reconstructed. We explore the use of these virtual user representations in various interaction scenarios in which local and remote users are face-to-face, side-by-side or decoupled. Initial experiments with distributed user groups indicate the mutual understanding of pointing and tracing gestures independent of whether they were performed by local or remote participants. Our users were excited about the new possibilities of jointly exploring a virtual city, where they relied on a world-in-miniature metaphor for mutual awareness of their respective locations.

Index Terms—Multi-user virtual reality, telepresence, 3D capture

1 Introduction

Marvin Minsky originally coined the term "telepresence" to describe the ability of controlling the instruments of a remote robot as if operating directly with one's own hands [29]. In this sense, the term refers to remote manipulation paired with high-quality sensory feedback. Bill Buxton later transferred the concept of telepresence to the domain of telecommunication [8]. He distinguished between the task space and the person space in collaborative work and argued that "effective telepresence depends on quality sharing of both". Considering a shared person space, Buxton et al. suggested representing each participant of a teleconference by an individual terminal equipped with audio and video facilities [7]. A shared task space was provided with additional interconnected electronic whiteboards. Ishii and Kobayashi's Clearboard [15] expanded on the metaphor of a transparent whiteboard between two distant users, which merges the shared person and task space for one-to-one telecommunication. Since then, the advances toward a shared person space have been impressive (e.g. [3]), while the idea of an integrated shared space for groups of people and tasks has received much less attention.

We created an immersive telepresence system that allows distant groups of users to collaborate in a shared task space. We used two projection-based multi-user 3D displays [22, 10] to provide the means for local collaboration. These two systems were driven and coupled using the distributed virtual reality framework AVANGO [21]. A cluster of depth cameras continuously captured participants and physical objects at each site. The captured 3D data was then transferred to the remote location in real time and displayed within the shared virtual environment. This setup allowed us to realize direct face-to-face group meetings as if occurring locally. Furthermore, we explored other configurations where the groups were placed next to each other or navigating completely independently in the shared virtual world. Both groups of users were informed of their respective locations through a world-in-miniature (WIM) that was attached to a stationary navigation device in front of the display (Figure 1).

Our work was inspired by many other immersive telepresence projects, including the early TELEPORT system [12], the National Tele-Immersion-Initiative (NTII) and the blue-c project [13]. In these systems the capturing technology remained a challenging problem, which is now simplified by the availability of commodity depth cameras [47]. Recent work based on depth cameras produced promising results for one-to-one telepresence [24, 4] using 3D video avatars. Sev-

- *Stephan Beck is with the Virtual Reality Systems Group at Bauhaus-Universität Weimar. E-mail: stephan.beck@uni-weimar.de.*
- *André Kunert is with the Virtual Reality Systems Group at Bauhaus-Universität Weimar. E-mail: andre.kunert@uni-weimar.de.*
- *Alexander Kulik is with the Virtual Reality Systems Group at Bauhaus-Universität Weimar. E-mail: kulik@uni-weimar.de.*
- *Bernd Froehlich is with the Virtual Reality Systems Group at Bauhaus-Universität Weimar. E-mail: bernd.froehlich@uni-weimar.de.*

Manuscript received 13 September 2012; accepted 10 January 2013; posted online 16 March 2013; mailed on 16 May 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

eral projects tried to reconstruct the surrounding local space of the participants, as seen with [24]. Others focused on capturing the involved persons, their postures, gestures and actions to embed them into a virtual environment [13, 23, 34]. We followed this last approach with an additional interest in capturing handheld objects and the interactions with a navigation device. However, none of the existing systems provided multi-user 3D display capabilities and in most cases they were too small for colocated group interaction in a shared task space or they did not provide life-size 3D representations of the users.

We built the first telepresence system that provides an integrated shared space in immersive virtual reality for groups of people and their tasks. Our system displays virtual objects in the shared space between the two groups of users as well as surrounds them with a consistent virtual environment. The main contributions of our work fall into three areas:

- 3D capturing and reconstruction: We introduce the use of a depth correction volume for precise calibration of individual depth cameras, which is the key to registering multiple depth cameras over a larger space.
- Interaction: Our interface allows users to couple both groups in registered face-to-face or side-by-side situations. Alternatively, both groups can move independently through a virtual world. We suggest the use of a WIM to provide awareness of the locations of the respective other group in the environment.
- User study: Our study confirms that local pointing can be exploited as a means for direct communication between local and remote participants. In both situations, we also observe similar limitations in accuracy, albeit for different reasons. Local pointing is affected by the accommodation-convergence mismatch while remote pointing suffers from the precision of the 3D reconstruction of a user's finger or hand.

In our work we provided basic audio communication through a single microphone and speaker at each site. While our application ran in a distributed configuration for driving two different projection setups, we did not focus on the data distribution aspect. All our machines were connected to our local 10 GbE network. Nevertheless, significant amounts of engineering are necessary to build and run such a complex distributed system, including the setup of and the communication with multiple depth camera servers, the calibration and registration of the different camera coordinate systems and the stereoscopic real-time rendering of up to 10 image streams consisting of color and depth information.

The 3D reconstructions from a set of depth images still contain many visual artifacts despite our significantly improved registration of multiple depth cameras and their depth accuracy. In particular, the reconstruction of the shutter glasses leaves much to be desired. However, our users ignored these artifacts for the most part and focused on the tasks. They confirmed that taking turns and collaboration as well as communication through gestures worked well.

2 Related Work

Numerous systems have been proposed that aimed for a shared person space in teleconferencing. Four complementary approaches can be distinguished:

1. 2D videoconferencing systems that enable eye contact through optimized camera placement [15, 9, 46, 31],
2. the embodiment of remote conference participants in a local audio and video terminal, also called situated avatar [7, 18, 43, 17, 20, 32, 1],
3. virtual avatars that are controlled by the users' body movements [40], sometimes including eye movements [41], and
4. 3D capturing of remote participants to enable multiple perspectives including face-to-face with direct eye contact [45, 14, 27, 38, 17, 3, 24, 4].

The latter approach had been envisioned by Fuchs and Neuman as early as 1993 [11] and first encouraging steps were presented five years later [35]. The recent availability of commodity depth cameras facilitates 3D capturing and consequently boosted developments in all directions. On the basis of 3D video data the angular disparity of cameras in 2D videoconferencing can be removed by aligning the perspective artificially [38], situated avatars can provide a live size 3D appearance of remote participants by displaying 3D video on the surface of a physical terminal [1] and realistic virtual avatars can be generated by scanning the users' physical appearances [44].

In the domain of real-time 3D capturing and reconstruction the system presented by Maimone and Fuchs [24] was particularly impressive. They achieved a high quality surface reconstruction by fusing overlapping depth contributions based on a per pixel quality analysis. Later, the same authors proposed an improved 3D calibration method for the depth cameras [25] in order to capture a larger space. They also suggested physical vibration of the depth cameras to reduce interferences of the superimposed infrared patterns in a setup with multiple Kinects [26]. The resulting motion blur causes the projected patterns from other devices to vanish in the individual camera images, while the respective own patterns remain stable. At the same time Butler et al. investigated this method and provided a detailed analysis of the optimal vibration parameters [6]. We followed this idea and attached a vibrating motor to the body of the Kinect devices. More recently, Kainz et al. [19] presented a system that used up to 10 depth cameras to cover an area of about two by two meters for omnidirectional 3D capturing of a person in real time. Similar to ours their system also involved a mobile device besides the stationary installed depth cameras. However, it cannot be applied for group-to-group telepresence. The available capturing volume is too small for a group of users and no means are provided for displaying remote participants. Newcombe, Izadi and colleagues [30, 16] presented a 3D scanning system that uses a single Kinect. Starting from a coarse reconstruction, their method iteratively refines the surface of static objects by a volumetric integration of successive frames over time. Although their method can capture objects in real time it is limited in range and it is not designed to handle dynamic objects.

Most early 3D teleconferencing systems had focused on symmetric one-to-one or asymmetric one-to-many setups. Symmetric group-to-group interaction was first realized in a 2D videoconferencing system by Nguyen et al. [31]. They provided individual capturing and display for each participant to achieve a consistent shared space among them, but the system could not provide relevant depth cues like stereo vision and motion parallax. Each participant could perceive the shared space only correctly from a dedicated sweet spot. In contrast we created an immersive telepresence system allowing distant groups of users to meet face-to-face in a shared task space. Our system seamlessly combines local and remote collaboration spaces in a 3D space that spans over the whole range of mixed reality [28] from purely virtual content over remotely captured 3D video to local physical reality.

3 3D Capturing and Reconstruction

Our technical setup is based on two projection-based multi-user display systems, two clusters of Microsoft Kinects and the distributed virtual reality framework AVANGO [21], which provides a shared virtual world for the spatially separated displays. Each Kinect-cluster captures a group of users in front of their display (Figure 2). The Kinect data streams are processed by multiple server processes and are then rendered as 3D reconstructions inside the virtual world.

Our technical setup requires an accurate calibration of individual Kinects as well as a precise registration of multiple Kinects to a joint world coordinate system. Each Kinect comes with slightly different optical characteristics due to manufacturing tolerances, which requires a good calibration of the cameras' intrinsics as well as an individual correction of each camera's distance measurements. The registration procedure must allow for camera configurations in which the cameras might not simultaneously see a single calibration target at a fixed position. During runtime, the Kinect data streams are acquired and processed by dedicated Kinect servers, which includes real-time rec-

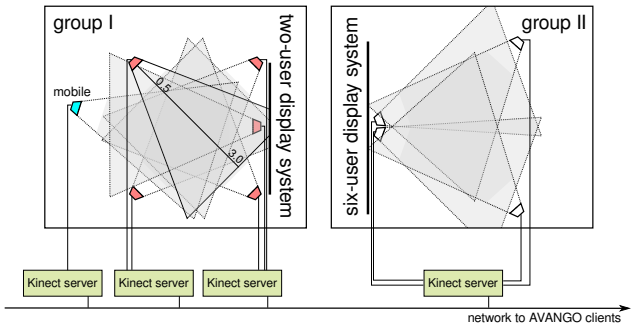

Fig. 2. Schematic diagram of our physical setup (overhead view). Two Kinect clusters in front of our two projection-based multi-user display systems and the Kinect servers that process the data streams. The blue Kinect is not attached to a fixed mount and can be interactively placed. The five red Kinects capture the users of the two-user system and the other four white Kinects capture the users of the six-user system. Accurate depth data can be provided in a range of 0.5 to 3.0 meters.

tification, filtering, and transference of the depth and color images to the rendering nodes of our distributed VR-application. The rendering nodes further process the incoming data streams and finally fuse them into 3D reconstructions.

In the next subsections we describe the most important steps that are involved to realize the described setup. Firstly, our dedicated calibration procedures are explained, followed by the software pipeline that processes and renders the image streams during runtime.

3.1 Calibration

Intrinsic Calibration Further processing of color and depth images requires rectification. For this task we use a tool provided by Nicolas Burrus [5] that computes the intrinsic parameters for the color- and depth-camera optics of each Kinect from a set of images of a checkerboard. The resulting parameters are stored in a file along with an affine transformation matrix that maps depth pixels to rgb pixels for each Kinect. Based on this data, standard OpenCV functionality can be applied for frame-wise image rectification during runtime.

Depth Calibration A Kinect's depth sensor reports a single raw depth value per pixel, encoded as an 11bit parallax value $v(x,y)$. This value can be converted to a metric distance $d(v(x,y))$ using the *raw-to-metric* mapping of Burrus [5].

While this mapping is sufficient for small scale environments we observed that it is not accurate enough for our requirements. Our system requires particularly accurate depth measurements because our Kinect clusters observe a relatively large volume of approximately 4 by 4 by 3 meters. Accurate depth data is a key requirement for all further processing and in particular the registration of multiple Kinects.

For an analysis of the depth accuracy, we placed several Kinects in front of a planar wall and compared the measured distances to the real wall-to-lens distance. With Burrus' mapping we observed errors of up to 15 cm between the real and the measured distance. These errors were different for individual devices and they also varied with distance and in different areas of the depth image. However, they were very well reproducible.

We parametrized Burrus' mapping so that it matches two reference measurements at different depths. This already reduced the error by a certain degree, but as we started to use multiple (mutually registered) Kinects, we observed that their measurements only matched at a few sweet spots. Apparently, such a simple linear correction does not reduce the variance in the measurements and does not even keep the mean error close to zero (Figure 4 left). Maimone and Fuchs [25] as well as Kainz et al. [19] observed similar issues. Both groups proposed an integrated solution for the calibration and registration of multiple Kinects. They simultaneously measure a calibration sphere at several positions with multiple Kinects. Maimone and Fuchs use an affine transformation of the data of each device to match the measured positions in 3D space. A limitation of this approach is that it may only reduce a linear bias. Kainz et al. fit a three-dimensional polynomial function to the acquired set of 3D depth errors which can be used at runtime to estimate and correct measurement errors.

We developed a more precise approach which produces a 3D-lookup table for per-pixel per-depth mapping of a Kinect's measurements. Our depth calibration method uses the almost perfectly even floor of our laboratory as the reference plane. A Kinect is mounted such that it orthogonally faces the floor (Figure 3). Counterweights facilitate leveling the camera which allows us to ensure an angular precision of less than one degree. The actual distance of the depth camera to the floor is measured with a well calibrated optical tracking system [2] serving as our ground truth. A motor winch moves the depth camera automatically from the ceiling of our laboratory towards the floor for generating the 3D-lookup table. This procedure for a depth range of $d_t = 0.5$ to 3.1 meter takes about 5 minutes.

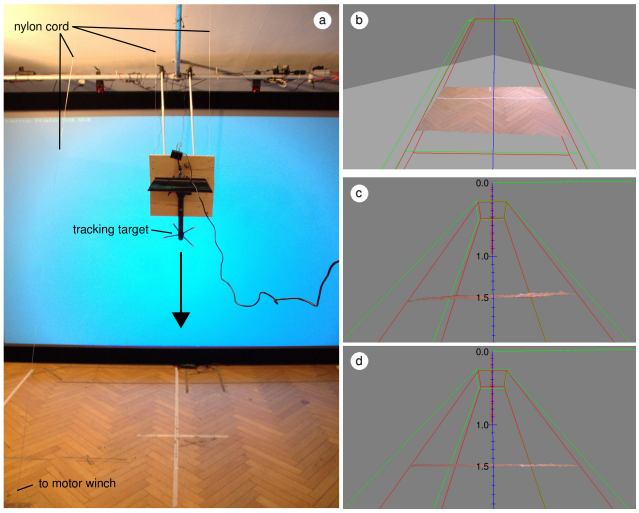

Fig. 3. Overview of the depth calibration process: a) A Kinect is attached to our custom mount. During calibration it is lowered by a motor winch from 3.1 to 0.5 meter above the floor that serves as the reference plane. b) The Kinect's line of sight (blue axis) is orthogonal to the ground during calibration. Here the normal of the gray semi-transparent plane coincides with the Kinect's line of sight. c) The inaccurate reconstruction of the ground floor without our depth calibration. d) With our depth calibration the reconstructed ground floor is almost perfectly flat and at the correct distance of 1.5 meter.

Our 3D-lookup table $D[640,480,2048]$ is sized corresponding to the resolution of the Kinect's depth image and 11 bit depth values. While the Kinect is slowly moving toward the floor we simultaneously acquire the distance values d_t from the tracking system and the depth images V from the Kinect. For each acquisition step, we add the correct distance value d_t to the 3D-lookup table by performing $D[x_i,y_j,V[x_i,y_j]] += d_t$ for each pixel (x_i,y_i) of the depth image V. Note that the Kinect may report the same depth value for different real distances d_t.

Once this process is complete, the resulting value in each cell of D is divided by the respective number of added distance values. This normalization results in an averaging of the real-distance values that corresponds to a reported raw Kinect value. The few cells of D that remain empty are filled with values interpolated from adjacent cells. Values outside the calibrated range are tagged to be invalid. Hence our valid depth-range is limited to 3.1 meter for each Kinect. For accurate 3D capturing at larger distances we would require a higher depth resolution than provided by current hardware.

The 3D-lookup table D maps every possible raw value at each pixel in a Kinect's depth image to a highly accurate metric distance (Figure 4). Absolute accuracy depends on various factors, including the

accuracy of the tracking system and the noise of the Kinect sensor. Thus, the standard error for individual measurements at each pixel position mostly remains in the range of 1 cm while the mean error is very close to zero for all distances. Such an individual 3D-lookup table needs to be generated for each Kinect

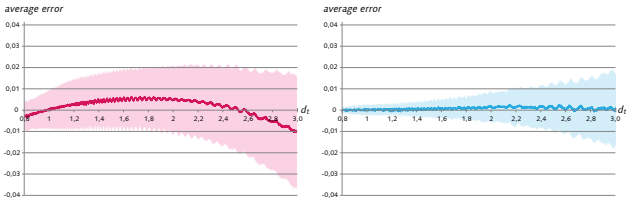

Fig. 4. Comparison of the depth measurement accuracy of a single Kinect with the parametrized version of Burrus' mapping (left) and with our mapping based on the 3D-lookup table (right). The y-axis shows the average error with respect to the real distance d_t across all pixels of a depth image with the standard deviation as an envelope. The x-axis shows the distance in a range from 0.8 - 3.0 meter. With Burrus' mapping, the mean of the error varies with distance and the standard error goes up to 3 cm (depending on the device). Our calibration method improves on both issues.

External Registration The calibrated data from all involved Kinect depth cameras must be registered to a joint coordinate system. Our approach uses a large tracked box as a physical reference. The Kinects derive their transformation relative to the calibration box whose position and orientation is constantly monitored by our optical tracking system [2].

Before starting the registration process for each Kinect, we take a depth shot of a static scene without the calibration box. Then the box is placed in front of the device so that three sides can be fully seen in the acquired image (Figure 5 left). By comparing every depth sample of a depth frame to the depth shot a set of 3D positions that belong to one of the three sides is selected as candidate points for further processing. Then local normals for all candidate points are computed by nearest neighbor search and a plane fitting step. Based on the orientation of their respective normals the algorithm associates the candidate points to one of the three sides of the box. The point sets define the corresponding surface only with a certain accuracy, therefore, within each set outliers are removed whose normal deviates more than a specified tolerance angle a_{min} from the average normal. The algorithm then searches for the best fitting planes through the three filtered point sets using the least squares method. A coordinate system is constructed by intersecting the three detected planes. In contrast to the corresponding corner of the reference cube, the derived coordinate system is not necessarily orthogonal (due to noise in the depth image). An optimization process thus adapts the tolerance angle a_{min} in a range from 0 to 20 degrees and outputs the coordinate system with the minimum overall deviation error from orthogonality over all axes. Depending on the orientation of a Kinect and the calibration box this error is mostly below 0.3 degrees. This coordinate system is then orthogonalized (by fixing the axis corresponding to the most stable plane) and then defines the relative transformation between the Kinect and the calibration box (Figure 5 right).

Mobile Kinect Our system also supports a mobile depth camera that can be positioned dynamically. In order to enable tracking of its pose during runtime, it is equipped with a tracking target (Figure 5 left). Initially, we register the mobile Kinect with the method described above. As a result of the registration step we obtain the pose of the calibration box, the pose of the attached target and the relative transformation between the depth camera and the calibration box. Given these transformations the relative transformation between the Kinect's sensor and its attached target can be derived.

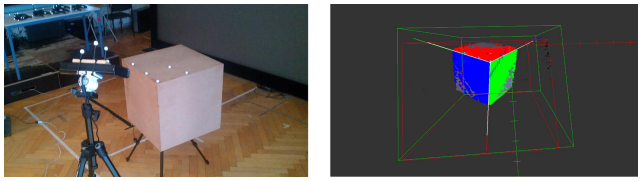

Fig. 5. Left: The calibration box with markers attached and a Kinect that has to be registered. The three sides of the box are exactly orthogonal. Right: The three sides of the box are detected in the depth image. The deviation between the computed coordinate system (colored axes) and its orthonormal form (white axes) is below 0.3 degree per axis.

3.2 Real-Time Processing Pipeline

The two calibrated and registered clusters of depth camera observe the space in front of each display and capture the participating users. The challenge is to provide individual 3D reconstructions for each user's views (up to eight users) from the image streams of up to ten Kinects - five capturing the space in front of the display supporting up to six tracked users, four in front of the other display supporting up to two users plus one mobile Kinect. Therefore all image streams must be processed simultaneously and in real time. For a single Kinect the processing steps from capturing to the final 3D-view are outlined in Figure 6. Note that these steps are not performed on a single workstation in our system but the figure rather shows the processing pipeline along a single Kinect image stream.

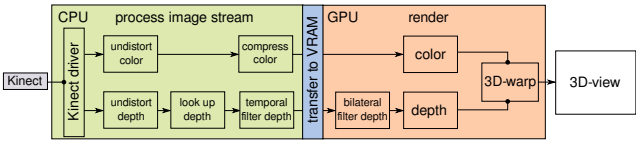

Fig. 6. Overview of the processing pipeline for the depth and color image streams of a single Kinect: From left to right: The Kinect generates new data at 30 Hz. The color images are rectified and compressed to DXT1 texture format. The corresponding depth frames are rectified, converted to metric distances and temporally filtered in parallel to the color streams. Both streams are then transferred to texture memory. On the GPU the bilateral filter is applied on the depth images and finally a 3D view is reconstructed using a 3D warping step.

Up to four Kinects can be connected to one multicore workstation that processes the depth and color data in up to eight parallel threads. Each Kinect provides raw color and depth images at 30 Hz. Both data streams are processed in parallel on the CPU starting with a distortion correction (using OpenCV) based on the cameras' intrinsic parameters. Using fastdxt [36], the color images are then compressed to the DXT1 texture format, which can directly be handled by modern graphics APIs such as OpenGL. After compression, each color image occupies 150 kB instead of 900 kB for the raw data. This saves network and also GPU-transfer bandwidth. We found that the quality after compression is almost indistinguishable from the original images. The raw values in the depth images are simultaneously converted to metric distances based on our 3D-lookup table described above. We further apply a temporal filter to the depth images. If the current depth value deviates more than 3 cm from the filtered value, we ignore the filtered value and use the actual depth value. In our experience this improves the capturing quality of static parts of the scene while dynamic content remains unaffected.

After the initial steps on the CPU, the captured data is further processed on the GPU. For a more stable reconstruction we apply a hardware-accelerated bilateral filter of size 5 by 5 pixels to the depth image. The results of bilateral filtering compared to unfiltered depth images are shown in Figure 7. Eventually, the filtered depth image is reprojected into 3D and rendered with a proxy triangle-mesh according to the perspective of the user. Depending on the number of Kinect

image streams and the available processing power we apply two different reconstruction methods. The faster but also less accurate approach only uses the geometry stage of the OpenGL pipeline to analyze and improve the quality of the reconstructed proxy mesh. Triangles that are too large in relation to the capturing distance are eliminated as well as those that are almost facing away from the viewing direction of the capturing device. In the fragment stage the compressed color image is accessed and applied to the warped triangle mesh through a texture look-up. Alternatively we apply a 3D-reconstruction method similar to the one introduced by Maimone et al. [24]. This approach yields better quality for surfaces that are seen by more than one Kinect but is computationally more expensive.

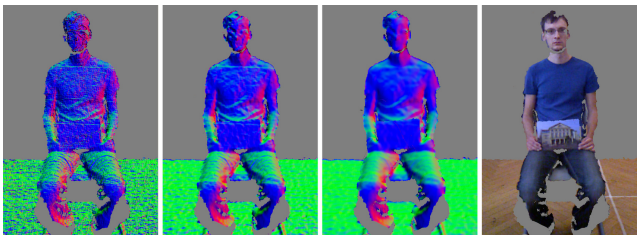

Fig. 7. Comparison of unfiltered depth images and the results achieved with the bilateral filter. The surface normal is computed from the depth gradient and mapped to rgb to visualize the smoothing effect. From left to right: no bilateral filter, 5 by 5 pixel filter kernel, 10 by 10 pixel filter kernel, reconstruction with photo texture applied for 5 by 5 pixel filter kernel.

The whole capturing-and-reconstruction pipeline of our system is distributed over several workstations. Figure 8 shows an overview. The raw data from the Kinect devices is acquired and processed by several Kinect servers that provide their results to client applications on other workstations via a multi-cast network connection. We currently do not focus on compression techniques that are specialized for medium bandwidth networks such as Pece et al. [33]. However, network traffic is slightly reduced through compression of the color images.

For each user, an individual client application receives the processed capturing data from the network and performs the 3D reconstruction and the stereoscopic rendering on a separate graphics card. The data transfer to the GPU and the rendering on the GPU are performed sequentially. On the workstation that drives the two-user display, two clients are running, while a further six clients are running on another workstation that drives the six-user display. Each client reconstructs and renders the captured data of all involved Kinects' depth and color image streams on a dedicated graphics card for the left and the right eye of a user. If a WIM is used, then additionally all Kinect streams are rendered to show the local and remote users in the WIM.

4 Group-to-Group Interaction

Many applications can benefit from an immersive telepresence system that allows two or more groups of users at different locations to meet in a shared virtual environment. Architectural applications are a prime example since they traditionally perform face-to-face discussions around a small scale model and appreciate the possibility for a joint exploration of a life-size 3D model. For enabling a local group to work together in a colocated setup, we use a projection-based multi-user system, which provides each tracked user with a perspectively correct visualization of the 3D scene. In such a system colocated users can show details of the model to each other by simply pointing with their bare hand. Salzmann et al. [37] showed that an accuracy of two to three centimeters can be achieved. They also reported that tracing along the boundaries of small objects allows other users to identify the object more securely than by simply pointing at it. Remote participants, represented by life-size video avatars in our telepresence system, can join this gestural communication just as naturally.

Besides bare handed pointing, navigation is an important part of the interface. In our system each group is equipped with a stationary

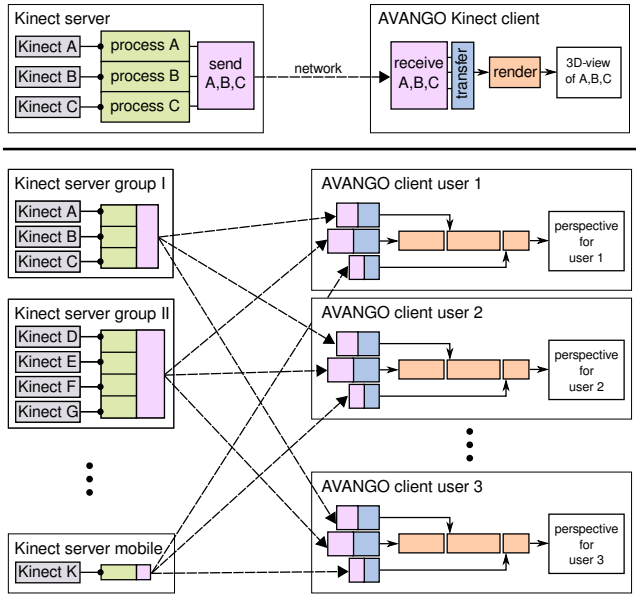

Fig. 8. Overview of the data flow within our distributed VR-application: Top: A Kinect server processes the data of several attached Kinects in parallel and then sends the processed frames via a multi-cast connection to a client application. The client receives the frames in an asynchronous manner and transfers incoming frames to the GPU for rendering. Bottom: In our distributed application setup several Kinect servers process and send the streams of multiple Kinects to each participating AVANGO client which renders the stereoscopic view for a particular user. The lengths of the individual processing blocks indicate the relative processing times.

input device, the Spheron, which is placed approximately 1.5 m in front of the screen. The Spheron is equipped with a 3D joystick on the side for moving through the scene and a large 3D trackball on the top for changing the viewing direction [22] in navigation mode. In manipulation mode the input is applied to a selected object.

Our interface also provides a world-in-miniature (WIM [42]) to facilitate wayfinding and the coordination between groups. The WIM can be manipulated with a handheld prop, but if this reference object is placed on the Spheron, the WIM becomes attached to the stationary device. It is then visualized directly above the trackball and can be further manipulated with input from the Spheron. We observed that users on both sides prefered this alternative for collaborative way planning and navigation.

We differentiate among three different configurations of the two user groups that support different requirements: face-to-face, side-by-side or independent. In the following sections, we explain these configurations. Figures 9-11 illustrate the resulting situations in the exemplary scenario of a joint tour through a 3D city model.

4.1 Face-to-Face Meeting

A face-to-face configuration of the two groups is the most natural starting point for a meeting. The Spheron serves as a centrally shared object and both groups are placed on opposite sides of the device. As a result, the positions of the two Spheron devices match in the virtual environment. Figure 9 shows this configuration in the WIM. In this situation the participants can directly see each other. The remote participants' hands and partially their bodies appear in front of the screen. If a remote and a local user are touching the same point on a virtual model, their fingers will also meet in space. In this configuration remote and local users can show each other details in the WIM. Furthermore, if they zoom into the WIM - to their actual location in the virtual world - they can see their own 3D representations acting in real-time in the virtual world. In this configuration as well as in the side-by-side configuration, there is only a single WIM model. Both

Fig. 9. Two groups of users are standing across from each other and around the overlapping Spheron input devices. They zoomed into the WIM to the location where they are currently located in the city model. Here they can see video-avatar representations of themselves.

Fig. 10. The two groups stand next to each other. The user on the right side is standing behind the remote users and thus he can see their virtual representations from behind.

Spheron devices affect the same model. We simply add the motion input from both parties to a combined input. This generally does not result in interferences between input from both groups, since users see each other accessing the control of their local device and directly resolve disagreements by talking and gesturing.

4.2 Side-by-Side Coupling for Joint Navigation

Face-to-face meetings are most suitable for exploring small-scale virtual content displayed between the two groups. For jointly exploring a large scale virtual model, the two user groups can place themselves so that they are both facing in the same direction. In this case their Spheron devices as well as their screens align. Both groups can take over the navigation through the virtual scene by operating the Spheron. Typically, they coordinate in taking turns by talking to each other. If one group stands slightly left of their Spheron and the other group slightly to the right, they are truly in a side-by-side configuration. In this configuration none of the virtual user representations can be seen. Only if remote users move forward will they partially appear to the local users as they enter the local users' viewing frustum. At this point it is possible that the remote user points to a virtual object and the local user is able to correctly see this gesture if there is no occlusion. However, such a true side-by-side placement of the users is not required and local and remote users can stand in arbitrary locations in front of the screens as in Figure 10. In the side-by-side configuration, surprising things can happen such as a virtual arm of a remote user extending from the body of a local user.

4.3 Independent Navigation

At times both user groups might want to be more independent of each other. Thus they can detach from the coupled face-to-face or side-by-side configuration and switch to an independent navigation mode. Both groups are equipped with a local Spheron input device and all other interface elements, including the WIM, are provided to each group individually. However, the groups can still see each other in the shared virtual world by looking at the WIM or in configurations where one group is standing or moving in front of the other. For example, one group may follow the other (Figure 12, left) or, they can meet each other during individual tours (Figure 12, right). Both situations mimic real-world situations with the limitation that each display system provides only a window into the virtual world through which the others can be seen. If one group cannot directly see the other, the WIM comes in handy. A handle sticking out of the WIM shows where the other group is (Figure 11). They can then zoom into the WIM to see what the others are doing. Using the handle they can move themselves closer to the others. Alternatively, the handle also allows one group to pick up the other group and move them to a different location, e.g. place them in front of a particular landmark or close to their own location. Such an act may cause confusion - if not discomfort. Performing actions that strongly affect the other users, should at least be announced. If the other group is not in viewing distance, the only possible communication method is via audio, i.e. voice.

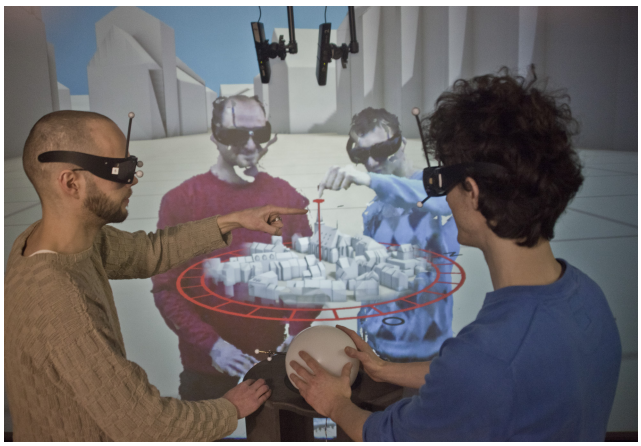

Fig. 11. A handle sticking out of the WIM shows the position of the local and remote group. In this case both handles overlap because the users are coupled. Each user could pick up the remote group and move it to a different location.

5 Results, Evaluation and Discussion

We set up two multi-user telepresence platforms, both building on multi-user stereo projection displays that accommodate two or up to six users respectively. Our setup incorporates 10 Microsoft Kinects, each equipped with a vibration motor. The PC workstations for Kinect servers are equipped with two Intel Xeon X5680 six-core processors running at 3.33 GHz and 48 GiB of main memory. On the six-user projection system the AVANGO client applications run on a single workstation equipped with two Intel Xeon X5680 six-core processors running at 3.33 GHz, 96 GiB of main memory and three NVIDIA Quadro Plex 7000 graphics subsystems. On the 2-user projection system the AVANGO client applications run on a single workstation equipped with two Intel Xeon X5680 six-core processors running at 3.33 GHz, 48 GiB of main memory and one NVIDIA Quadro Plex 2200 D2 graphics subsystem. The screen size is 4.3 by 2.7 meters for the six-user and 3 by 2 meters for the two-user projection system. Both projection systems have a resolution of 1920 by 1200 pixels. All workstations are running Ubuntu Linux 11.04, 64-bit, kernel version

Fig. 12. Left: One group trails another group on a city tour. Right: The two groups meet inside a building.

Table 1. Timings in milliseconds per frame for the different processing steps. The server time includes the pull time from the Kinect driver, rectification of color and depth images, applying our depth calibration, compression of the color image and the temporal smoothing of the depth image. Transfer time is over our 10 GbE network. Bilateral filtering uses a 5x5 filter kernel.

# Kinects	server	transfer	bil. filter	3D-warp	**sum**
1	50	2.7	4	3.1	59.8
2	50	5	7	6.5	68.5
3	50	6.7	10	10.2	76.9
4	50	8.3	15	14.2	87.5

Fig. 13. Two groups of users meet in a virtual city model. The image to the left shows the situation at the six-user display. The same situation on the other side with the two-user display can be seen on the right.

is 2.6.38-13 and NVIDIA driver version 304.43. They are connected via a 10 GbE fibre network connection.

Using this setup we typically achieved application frame rates of around 30 Hz for scenarios that do not use the WIM. With the WIM enabled, 3D video avatars of all participants must be rendered for each of these additional scene representations. Correspondingly, the frame rate dropped to 20 Hz with one WIM or 12 Hz with two WIMs. The system's latency is different for purely virtual content as compared to the 3D reconstructions of remote users. At 30 Hz the local rendering takes about 100 ms from tracking the user's motion input to the display update. The 3D capturing and reconstruction of remote users, however, takes significantly longer. Between 300 to 500 ms elapse before an action of a remote user can be perceived locally. Thus, independent of the users' individual capacity of reaction, we have to consider the round trip time of about one second before we could see the reaction of a remote participant to a gesture of a local participant. These estimates of the end-to-end latency are based on measurements of individual steps in our processing pipeline (Table 5). They also include the basic latency of the Kinect as well as the rendering time.

The network transfer is certainly another factor in our system, but not the dominating one. For example, the data of compressed color images and uncompressed depth images from four Kinects sums up to approximately 5.3 MB per frame. In our 10 GbE network it takes 8.3 ms on average to transfer this amount of data. In a truly distributed setting the latency of the remote 3D video avatars would be somewhat higher. The network bandwidth would become the main issue. Therefore, one would have to use a higher compression for the color images and also compress the depth values as suggested by Pece et al. [33].

For the best possible coverage of a groups's interaction space, a large number of depth cameras would be beneficial. However, the interactivity would suffer from the increased latency and reduced frame rate. Thus, we decided to optimize the capturing quality of the two-user display using five Kinects plus one mobile Kinect, while ensuring sufficient coverage of the larger platform to enable bidirectional gestural communication by using four Kinects. The setups can be seen in Figure 2. In both setups we perform an outside-in capturing to cover the users from all sides. The smaller setup provides a decent reconstruction of two persons interacting in a 1 m radius around a sweet spot 1.5 m in front of the screen center (Figure 13, right). One of these cameras is the mobile Kinect and can be repositioned during runtime for close-up shots or for the compensation of missing viewing angles. For the six-user display four Kinects provide basic acquisition capabilities (Figure 13, left). Here a surround reconstruction of people is only available close to the sweet spot 1.5 m in front of the screen. Furthermore occlusion of individuals through other people can rarely be compensated by other viewing angles. For the exchange of audio information we set up a basic speaker and microphone at each site.

5.1 Usability Experiments

During regular visits from guests to our laboratory, we received very positive feedback about the system and observed that it allows for rich gestural communication between local and remote participants. It facilitates the examination of virtual objects shown between the two groups as well as the exploration of virtual environments surrounding the users. To verify our impressions we invited four groups of three users for formal evaluation sessions each lasting one hour, with about 45 minutes in our systems. The four groups were subsequently scheduled and all went through the same protocol, collaborating remotely with two experimenters on the planning of sightseeing tours through a virtual model of our local city. The participants always used the larger telepresence system (Figure 13 left), while both experimenters coordinated the group and remotely performed tests using the smaller telepresence system (Figure 13 right).

5.1.1 Participants

We recruited 12 students of various disciplines from our university. They were aged between 20 to 31. All but one participant were male.

Fig. 14. A remote user pointing at a building and a local user identifying it during the pointing study.

We divided them into four groups of three users that were invited for independent test sessions. The stereo parallax was adjusted for each user according to their respective eye distance to ensure accurate spatial perception.

5.1.2 Procedure

After a short introduction all sessions proceeded through three main phases:

Welcome and Introduction: The guests were welcomed by the 3D video avatars of the experimenters, who asked everybody to shake hands, making sure that the visual perception allows for direct interpersonal interaction. Thereafter we enabled the miniature visualization of our city model. We mentioned that the model showed our actual geographic environment and asked the users to identify known buildings by pointing and speaking.

Understanding Pointing Gestures of Remote Participants: In the second phase we tested the comprehensibility of pointing gestures of remote participants. The experimenters subsequently pointed at 10 individual building models. We asked the participants to identify them using a virtual ray to eliminate any possible confusion in their selection. To involve all participants, the pointing device was handed over to another participant after each of the 10 pointing tasks and the other two were always asked to verify the other user's selection. The virtual building models used for this study were all of similar size of around 2 x 2 x 2 cm and directly connected to other surrounding buildings (Figure 14). We knew that misunderstandings can be rapidly clarified in the local group as also with the remote communication partners. To study the immediate communication capabilities of our system, we asked the other participants to hold back with their support until the active user had made a selection.

After the remote pointing tests, the experimenter who performed the pointing joined the group of participants to repeat the same test again with a different set of buildings and local pointing. This allowed us to compare the local and remote pointing capabilities of our system. After this test the experimenter moved back to the smaller telepresence platform. To balance the order of the pointing tasks, two groups performed the sequence in reverse order.

Virtual City Tour: For a joint tour through the life-size city model, we decoupled the two telepresence platforms. Each group had to individually control their traveling using the Spheron device situated in front of the screen. Instructions on how to use it were given by the remotely present experimenters. We also showed them their miniature avatars in the WIM and asked everybody to identify themselves in the WIM. We started with a tour of sites that had been selected by the experimenters, the group of participants followed. We instructed them to use the WIM in case of losing eye contact with the experimenters' group ahead. Eventually, we asked the participants to show the experimenters their home or other places of interest by guiding both groups there in the virtual model.

Thereafter we switched off the systems and asked the participants to rate several characteristics of the system in a questionnaire. The questionnaire consisted of 10 topics that were covered by groups of two to four separate questions that had to be answered using Likert scales with varying orientation. For the evaluation we aligned the orientation of the scales and averaged the responses to these groups of questions (Figure 15).

5.1.3 Results

During the introductory phase users immediately understood the system and engaged in playful greeting gestures with the remote group of experimenters. They also recognized the city model and started discussing the relevance of various places in the city for themselves. Pointing and tracing gestures were used to communicate naturally with each other. Although we observed no communication issues in this informal setting, we found in pointing tests that pointing gestures could be misunderstood, both in colocated as well as in remote communication. While two groups identified all buildings correctly, each of the two other groups made one identification error in the remote setting. The subjective feedback of the participants indicated that the captured finger was not reconstructed well enough and appeared to be blurry on these two occasions. One group also had two identification errors in the colocated pointing condition. We attribute this to the perceptual conflict of vergence and accommodation that may hamper depth perception [37].

The subjective user ratings were very positive (Figure 15). In particular, all participants were enthusiastic about the overall system and everybody agreed that this type of system would be a useful extension to telecommunications and that they would like to use it if it comes to the collaborative analysis of virtual 3D models. All of them found the virtual content as well as the 3D video avatars to convey a strong sense of spatiality. The WIM representation including the miniature video avatars were rated to be very helpful for orientation and group-to-group coordination during decoupled travel.

The users found group communication and coordination using a combination of gestures and spoken language to be very effective, although they clearly stated further potential in gestural communication and even more so in gaze communication as only head orientation could be estimated. We assume that this also affected the illusion of physical copresence of remote participants which was rated as only slightly positive. Clearly, our 3D video avatars are of limited quality as can be seen from most figures in this paper and thus cannot compete with the actual physical presence of the colocated participants.

The feedback on the shutter glasses was very inconsistent and spread across the entire scale with an almost neutral average. Various issues might have affected this rating: The shutter glasses are custom-build and are not as comfortable and light weight as commercial shutter glasses. They cannot be captured by the Kinect sensor due to the diffuse black material of the housing and the high reflectivity of the shutters made of glass. As a result there are disturbing holes in head reconstructions of the 3D video avatars (Figure 14). As a workaround we now render virtual 3D models of the shutter glasses at the respective positions (Figure 1) and observed that this was already a huge improvement. Furthermore, this adaptation improves the perception of head orientation.

The overall enthusiastic subjective ratings confirm the usability of the system. Although we believe that part of its impressiveness is also due to its novelty. The lower agreement on the system's support for gaze communication, physical copresence and the acceptance of the shutter glasses point to the limitations of our system.

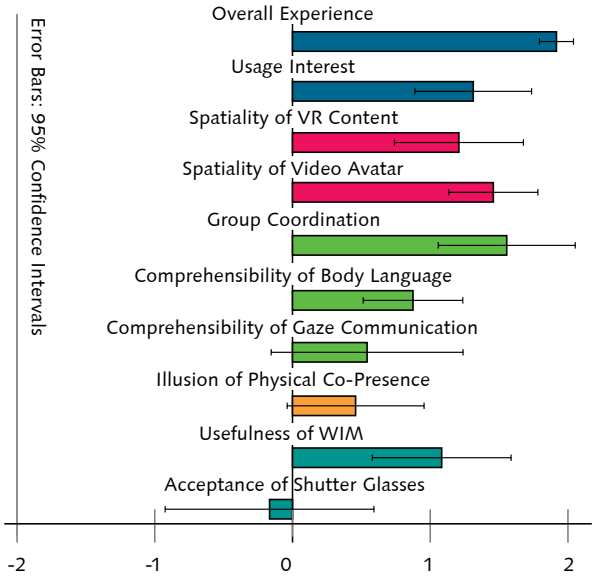

Fig. 15. Average user ratings of system characteristics, clustered by similarity of topics.

5.2 Discussion

The general aim of our work was to explore different situations and support for interactions of distributed groups of people that meet in a shared virtual environment. We chose to represent the remote users as life-size 3D video avatars as we found this to be the most realistic representation. After first seeing a single remote video avatar we were amazed how natural the movements of the remote person were perceived even though the reconstruction was incomplete in places, this is much in contrast to interactions with regular avatar models, which are complete 3D models but their motions are typically much more artificial. We did not focus on the quality of the reconstruction as in other work, but wanted to cover a larger space and involve a small group of people instead of a single person. However, the positional accuracy of the reconstructions became quickly important as soon as the remote users wanted to use bare handed pointing to show something to the local users. The accuracy of depth measurements of the Kinects was identified as the main issue. Once our depth calibration procedure was in place, we appreciated the ability to observe others during pointing at small objects, which also benefited from an increased stability of the merging of contributions from different Kinects.

We achieved significant improvements in the quality of both capturing and reconstruction, but the visual representation of remote users remains noisy as well as perforated due to occlusions. The noisiness is also partly due to our basic reconstruction algorithm that was used during the study. A higher visual quality of the 3D video avatars would most certainly improve the illusion of copresence of remote participants. Note that the presence of actually colocated participants in our multi-user systems defines the baseline for the perception of copresence, which will be challenging to compete with even if perfect capturing and reconstruction can be achieved in the future. From our experiences with using the system, we are convinced that for many applications, the support for direct communication between team members, including subtle body language, is more important than the complete and accurate reconstruction of the remote user representation – even though it is highly desirable.

Despite these current limitations, we observe that even naive users can immediately start to interact with our telepresence setup for direct gestural communication. Our pilot study demonstrated the capability of the system to support remotely connected groups of colocated users in realistic tasks involving object inspection and the joint exploration of virtual environments.

6 Conclusions and Future Work

We realized the first symmetric 3D telepresence system that provides a shared space for two groups of users based on two coupled stereoscopic multi-viewer systems. Our system captures the users in front of each display using a set of color-and-depth cameras and reconstructs life-size 3D avatars at the respective remote site. Navigation techniques support the exploration of a shared virtual world in different configurations, which are particularly suitable for architectural applications. We demonstrated advances in registering multiple depth cameras based on the volumetric calibration of each involved Kinect sensor. Experiments with the system show that we achieved an average accuracy of 1-2 cm. A user study confirmed that this is precise enough to understand deictic gestures. The participants expressed that turn taking and collaboration as well as communication through gestures worked well while they were fascinated by the directness and naturalness of the interaction with the remote collaborators.

We have only begun to explore the area of 3D user interfaces for these holodeck-like encounters. In our city tour prototype we displayed the city and the participants life-sized or appropriately scaled in the WIM. However, it is certainly interesting to explore the city as a giant or dwarf if sufficient detail in the city model is available. In collaborative applications, such scaling of user representations could disturb the perception of others in the shared 3D environment and affect the social behavior of the participants.

Despite the quality we have achieved, we are not yet satisfied with the visual appearance of the captured 3D video avatars. Higher resolution depth cameras and more intelligent reconstruction algorithms for dealing with the occlusions will significantly improve the experience in our immersive 3D telepresence system. However, higher network bandwidth and faster graphics processing would be also required. A further enhancement to the system is spatial audio. In particular the listener-position-independent 3D audio reconstruction of a wavefield synthesis system would be ideally suited for our configuration as already shown in [39].

The excitement of our users toward our system – despite all its limitations – has convinced us that this is a first glimpse into the future of online 3D meetings and also 3D network gaming. The walk-up capability of the system, the active involvement of multiple users and the real-time reconstruction of life-size 3D representations of one or more remote groups of people are the main factors that contribute to the positive immersive telepresence experience.

Acknowledgments

This work was supported in part by the European Union under grant AAT-285681 (project VR-HYPERSPACE), the German Federal Ministry of Education and Research (BMBF) under grant 03IP704 (project Intelligentes Lernen), and the Thuringian Ministry of Education and Cultural Affairs (TKM) under grant B514-08028 (project Visual Analytics in Engineering). We thank the participants of our studies and the members and students of the Virtual Reality Systems group at Bauhaus-Universität Weimar (`http://www.uni-weimar.de/medien/vr`) for their support.

References

[1] S. Alers, D. Bloembergen, M. Bügler, D. Hennes, and K. Tuyls. Mitro: an augmented mobile telepresence robot with assisted control (demonstration). In *Proc. of AAMAS 2012*, pages 1475–1476, Richland, SC, 2012. International Foundation for Autonomous Agents and Multiagent Systems.

[2] ART. Advanced realtime tracking gmbh, 2012. "`http://www.ar-tracking.com/home/`".

[3] T. Balogh and P. T. Kovács. Real-time 3d light field transmission. In *Proc. of SPIE 2010*, volume 7724, pages 772406–772406–7, 2010.

[4] H. Benko, R. Jota, and A. Wilson. Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proc. of CHI 2012*, pages 199–208, New York, NY, USA, 2012. ACM Press.

[5] N. Burrus. Rgbdemo: Demo software to visualize, calibrate and process kinect cameras output, 2012. Available at "`http://labs.manctl.com/rgbdemo/`".

[6] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim. Shake'n'sense: reducing interference for overlapping structured light depth cameras. In *Proc. of CHI 2012*, pages 1933–1936, New York, NY, USA, 2012. ACM Press.

[7] W. Buxton, A. Sellen, and M. Sheasby. Interfaces for multiparty videoconferences. *Video-Mediated Communication*, pages 385–400, 1997.

[8] W. A. S. Buxton. Telepresence: integrating shared task and person spaces. In *Proc. of Graphics Interface 1992*, pages 123–129, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[9] M. Chen. Leveraging the asymmetric sensitivity of eye contact for videoconference. In *Proc. of CHI 2002*, pages 49–56, New York, NY, USA, 2002. ACM Press.

[10] B. Fröhlich, R. Blach, O. Stefani, J. Hochstrate, J. Hoffmann, K. Klüger, and M. Bues. Implementing multi-viewer stereo displays. In *Proc. of WSCG 2005*, pages 139–146, Plzen, Czech Republic, 2005.

[11] H. Fuchs and U. Neumann. A vision of telepresence for medical consultation and other applications. In *Proc. of the 6th Symposium on Robotics Research*, pages 565–571, Hidden Valley, PA, USA, 1993. Intl. Foundation for Robotics Research.

[12] S. J. Gibbs, C. Arapis, and C. J. Breiteneder. Teleport: Towards immersive copresence. *Multimedia Systems*, 7(3):214–221, 1999.

[13] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. In *Proc. of SIGGRAPH 2003*, pages 819–827, New York, NY, USA, 2003. ACM Press.

[14] M. Gross, S. Würmlin, M. Naef, E. Lamboray, C. Spagno, A. Kunz, E. Koller-Meier, T. Svoboda, L. Van Gool, S. Lang, K. Strehlke, A. V. Moere, and O. Staadt. blue-c: a spatially immersive display and 3d video portal for telepresence. *ACM Trans. Graph.*, 22(3):819–827, July 2003.

[15] H. Ishii and M. Kobayashi. Clearboard: a seamless medium for shared drawing and conversation with eye contact. In *Proc. of CHI 1992*, pages 525–532, New York, NY, USA, 1992. ACM Press.

[16] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proc. of UIST 2011*, pages 559–568, New York, NY, USA, 2011. ACM Press.

[17] A. Jones, M. Lang, G. Fyffe, X. Yu, J. Busch, I. McDowall, M. Bolas, and P. Debevec. Achieving eye contact in a one-to-many 3d video teleconferencing system. In *Proc. of SIGGRAPH 2009*, pages 64:1–64:8, New York, NY, USA, 2009. ACM Press.

[18] N. P. Jouppi. First steps towards mutually-immersive mobile telepresence. In *Proc of CSCW 2002*, pages 354–363, New York, NY, USA, 2002. ACM Press.

[19] B. Kainz, S. Hauswiesner, G. Reitmayr, M. Steinberger, R. Grasset, L. Gruber, E. Veas, D. Kalkofen, H. Seichter, and D. Schmalstieg. Omnikinect: real-time dense volumetric data acquisition and applications. In *Proc. of VRST 2012*, VRST '12, pages 25–32, New York, NY, USA, 2012. ACM Press.

[20] K. Kim, J. Bolton, A. Girouard, J. Cooperstock, and R. Vertegaal. Telehuman: effects of 3d perspective on gaze and pose estimation with a life-size cylindrical telepresence pod. In *Proc. of CHI 2012*, pages 2531–2540, New York, NY, USA, 2012. ACM Press.

[21] R. Kuck, J. Wind, K. Riege, and M. Bogen. Improving the avango vr/ar framework: Lessons learned. In *5th Workshop of the GI-VR/AR Group*, pages 209–220. Shaker, 2008.

[22] A. Kulik, A. Kunert, S. Beck, R. Reichel, R. Blach, A. Zink, and B. Froehlich. C1x6: a stereoscopic six-user display for co-located collaboration in shared virtual environments. *ACM Trans. Graph.*, 30(6):188:1–188:12, Dec. 2011.

[23] G. Kurillo, R. Bajcsy, K. Nahrsted, and O. Kreylos. Immersive 3d environment for remote collaboration and training of physical activities. In *Proc. of VR 2008*, pages 269–270, Washington, DC, USA, 2008. IEEE Computer Society.

[24] A. Maimone and H. Fuchs. Encumbrance-free telepresence system with real-time 3d capture and display using commodity depth cameras. In *Proc. of ISMAR 2011*, pages 137–146, Washington, DC, USA, 2011. IEEE Computer Society.

[25] A. Maimone and H. Fuchs. A first look at a telepresence system with room-sized real-time 3d capture and large tracked display. In *Proc. of ICAT 2011*, New York, NY, USA, 2011. ACM Press.

[26] A. Maimone and H. Fuchs. Reducing interference between multiple structured light depth sensors using motion. In S. Coquillart, S. Feiner, and K. Kiyokawa, editors, *Proc. of VR 2012*, pages 51–54, Washington, DC, USA, 2012. IEEE Computer Society.

[27] W. Matusik and H. Pfister. 3d tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Trans. Graph.*, 23(3):814–824, Aug. 2004.

[28] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, E77-D(12):1321–1329, 1994. Available at "`http://vered.rose.utoronto.ca/people/paul_dir/IEICE94/ieice.html`".

[29] M. Minsky. Telepresence. *Omni*, 2(6):45–52, 1980.

[30] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Proc. of ISMAR 2011*, pages 127–136, Washington, DC, USA, 2011. IEEE Computer Society.

[31] D. Nguyen and J. Canny. Multiview: spatially faithful group video conferencing. In *Proc. of CHI 2005*, pages 799–808, New York, NY, USA, 2005. ACM Press.

[32] O. Oyekoya, W. Steptoe, and A. Steed. Sphereavatar: a situated display to represent a remote collaborator. In *Proc. of CHI 2012*, pages 2551–2560, New York, NY, USA, 2012. ACM Press.

[33] F. Pece, J. Kautz, and T. Weyrich. Adapting standard video codecs for depth streaming. In *Proc. of EGVE 2011*, pages 59–66, 2011.

[34] B. Petit, J.-D. Lesage, C. Menier, J. Allard, J.-S. Franco, B. Raffin, E. Boyer, and F. Faure. Multicamera real-time 3d modeling for telepresence and remote collaboration. *International Journal of Digital Multimedia Broadcasting*, 2010:247108–12, 2009.

[35] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In *Proc. of SIGGRAPH 1998*, pages 179–188, New York, NY, USA, 1998. ACM Press.

[36] L. Renambot. Fastdxt: A fast implementation of a dxt compressor.

[37] H. Salzmann, M. Moehring, and B. Froehlich. Virtual vs. real-world pointing in two-user scenarios. In *Proc. of IEEE VR 2009*, pages 127–130, Washington, DC, USA, 2009. IEEE Computer Society.

[38] O. Schreer, I. Feldmann, N. Atzpadin, P. Eisert, P. Kauff, and H. Belt. 3dpresence -a system concept for multi-user and multi-party immersive 3d videoconferencing. In *Proc. of CVMP 2008*, pages 1–8, nov. 2008.

[39] J. P. Springer, C. Sladeczek, M. Scheffler, J. Hochstrate, F. Melchior, and B. Froehlich. Combining wave field synthesis and multi-viewer stereo displays. In *Proc. of VR 2006*, pages 237–240, Washington, DC, USA, 2006. IEEE Computer Society.

[40] W. Steptoe, J.-M. Normand, W. Oyekoya, F. Pece, G. Giannopoulos, F. Tecchia, A. Steed, T. Weyrich, and J. Kautz. Acting rehearsal in collaborative multimodal mixed reality environments. *Presence - Teleoperators and Virtual Environments*, 21(1), 2012.

[41] W. Steptoe, R. Wolff, A. Murgia, E. Guimaraes, J. Rae, P. Sharkey, D. Roberts, and A. Steed. Eye-tracking for avatar eye-gaze and interactional analysis in immersive collaborative virtual environments. In *Proc. of CSCW 2008*, pages 197–200, New York, NY, USA, 2008. ACM Press.

[42] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proc. of CHI 1995*, pages 265–272, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[43] T. Tanikawa, Y. Suzuki, K. Hirota, and M. Hirose. Real world video avatar: real-time and real-size transmission and presentation of human figure. In *Proc. of ICAT 2005*, pages 112–118, New York, NY, USA, 2005. ACM Press.

[44] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan. Scanning 3d full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):643–650, 2012.

[45] H. Towles, W.-C. Chen, R. Yang, S.-U. Kum, H. Fuchs, J. Mulligan, K. Daniilidis, L. Holden, B. Zeleznik, and A. Sadagic. 3d tele-collaboration over internet2. In *International Workshop on Immersive Telepresence, Juan Les Pins*, 2002.

[46] R. Vertegaal, I. Weevers, C. Sohn, and C. Cheung. Gaze-2: conveying eye contact in group video conferencing using eye-controlled camera direction. In *Proc. of CHI 2003*, pages 521–528, New York, NY, USA, 2003. ACM Press.

[47] Z. Zhang. Microsoft kinect sensor and its effect. *IEEE MultiMedia*, 19(2):4–10, Apr. 2012.