International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2015)
M. Imura, P. Figueroa, and B. Mohler (Editors)

An HMD-based Mixed Reality System for Avatar-Mediated Remote Collaboration with Bare-hand Interaction

Seung-Tak Noh[1], Hui-Shyong Yeo[1,2] and Woontack Woo[1]

[1]UVR Lab, KAIST, South Korea
[2]SACHI, University of St Andrews, UK

Abstract

We present a novel system for mixed reality based remote collaboration system, which enables a local user to interact and collaborate with another user from remote space using natural hand motion. Unlike conventional system where the remote user appears only inside the screen, our system is able to summon the remote user into the local space, which appears as a virtual avatar in the real world view seen by the local user. To support our avatar-mediated remote collaboration concept, we derive a systematic framework design that consists of the hardware and software configuration with various devices. We explore novel techniques for calibrating and managing the coordinate system in asymmetric setup, sensor fusion between devices and generating human-like motion for the avatar. For validating our proposal, we implemented a proof-of-concept prototype using off-the-shelf hardware and report the experimental results. We believe that our system overcomes not only several limitations of previous systems but also creates new possibilities in remote collaboration domain.

Categories and Subject Descriptors (according to ACM CCS): H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented, and virtual realities.

1. Introduction

The common idea of Mixed Reality (MR) is combining the real and virtual imagery [BGR*98, MTUK95]. One main issue on these platforms is how to give a seamless experience to the users by blurring the line between virtual and reality. A head-mounted display (HMD) is the main component that makes MR experience possible, but due to its low accessibility, it has inevitably been used only in highly controlled environment such as research laboratory.

Recently, consumer level HMD has become more common, as popular devices such as Oculus Rift and Google Cardboard were released to the public at affordable price. Although these consumer HMDs still suffer from problems such as heavy weight and encumberment, they posses a great potential to be an entry point for many average users to the mixed reality environment, similar to how handheld device popularized Augmented Reality (AR) in the past decade.

Now that HMD has become a commodity device, we turn our attention to other related yet challenging research issues when using HMD, such as providing natural input technique with hand tracking [HFW14, PAK*14, JNC*15] and supporting bidirectional telepresence [MYD*13]. With these technology, immersive remote collaboration can be achieved. There are some academic works on this field [STB*12, TAH12, VKL*11], but they ultimately are limited in the lab environment.

Remote collaboration is very useful in many contexts such as education, business and health care. For example, imagine a remote surgery scenario where surgeons from different locations are operating on the same patient (Figure 1). The patient only physically resides on the local side of the main surgeon but on the remote side, co-surgeon can also operate on the patient through telepresence technology. Each surgeon on his own side can see other co-surgeons from remote locations as if they are teleported to him. The actions executed by the remote surgeons are tracked in real time and replicated at the main surgeon location by means of robot or surrogate. Perhaps if not an actual surgery, a simulation surgery on a virtual patient would not require any physical action to be replicated on the other side, but can be very beneficial for training and learning purposes.

To realize the concept, we present a novel HMD-based MR remote collaboration system which enables a local user to collaborate with another user remotely. Each user remains in his local space and wears a see-through HMD. The remote collaborator is summoned into local space as a virtual avatar. With this avatar, users can collaborate on shared virtual objects in a collaborative space. In addition, vision-based hand tracking allows users to interact directly with these shared

DOI: 10.2312/egve.20151311

objects using their bare hand without holding any additional device or controller.

One of our primary goals is to achieve inexpensive and portable setup that can be readily deployed by average users. Hence, we also implemented a proof-of-concept prototype using only off-the-shelf hardware. We report on the performance and limitations of our system and discuss the issues we found. Our contributions are followed:

- Our system is novel in terms of the overall integration, as it generates new possibility in remote collaboration. To the best of our knowledge, there is no previous system that allows summoning remote user into local space and supporting direct interaction using only bare-hand.
- We describe the detail on avatar summoning for remote collaboration. It includes how to utilize global and local pose tracker together, fuse the body and hand tracking information from different sensors, and generate pseudo body motion using only limited body information with lightweight networking requirement.
- We implemented and evaluated the system using off-the-shelf hardware. We validate our concept by showing several potential application scenarios. The results and discussion of the findings provide insightful guidelines and implications for further improvements.

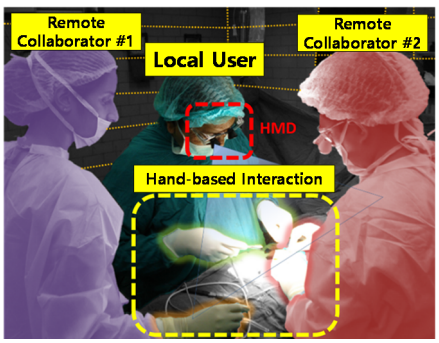

Figure 1: *Remote collaborators are summoned into local space as avatar, and they exist in the same co-space.*

2. Background

2.1 Remote User Positioning in Collaborative Space

For the past few decades, the major part of teleconference system remain the same, as the communication channels are limited to voice and video only, i.e., using camera system to capture user in front of the screen [IK92, KPB*12]. For enhancing the presence, the system usually also captures the remote environment. In this approach, the screen is an analog of window which connects the different spaces [MF11] (Figure 2a). The main drawback is that user cannot exceed one's own territory and enter the opponent's space. Hence, these systems often emphasize on verbal communication and eye contact (e.g. [KPB*12]) rather than supporting actual collaborative work between the users.

This limited territory problem can be partially solved (Figure 2b) through immersive display technology such as a large three-dimensional display [BFFK13, YKN*14] which can provide depth cue of the remote user's appearance. TELEPORT system [GAB99] is a pioneer realization of this concept that utilizes the wall-sized screen for merging the distant space as a connected room. However, a single display has limited viewing angle where user always need to look at the screen, even if the head position is tracked and correct perspective is rendered. We call this as the "2.5D" problem.

Meanwhile, situated avatar is another representative approach for displaying remote person (Figure 2c) [TDY*11, OSS*13]. It uses a physical surrogate such as SphereAvatar [STB*12] or robotic hardware to for representing the remote person, often supporting physical movement [TDY*11] or mirroring user's body motion [OSS*13]. The merit of this approach is that remote user is also "situated" in the local user's space, albeit looks different from the actual person.

One way to avoid the aforementioned problems is by using CAVE-like omnidirectional display system with image-based reconstruction by multiple cameras [GWN*03, VKL*11] to support various viewpoints and perspective. However, a major problem of this approach is the complex and expensive setup of the hardware and environment. It lacks scalability and portability, as the interaction space is limited to the pre-configured environment only.

Yet another major problem is occlusion handling. In the shared territory, the scene is cluttered with a mixture of real and virtual obstacles. In light of this, keeping the perception of these objects is important to provide natural collaboration experience to the user. In case of 3D display, it has no problem as long as the hand does not exceed the virtual object or avatar area. However, if the virtual object is located between hand and user's eye, the system cannot show the virtual object because user's hand physically blocks the screen.

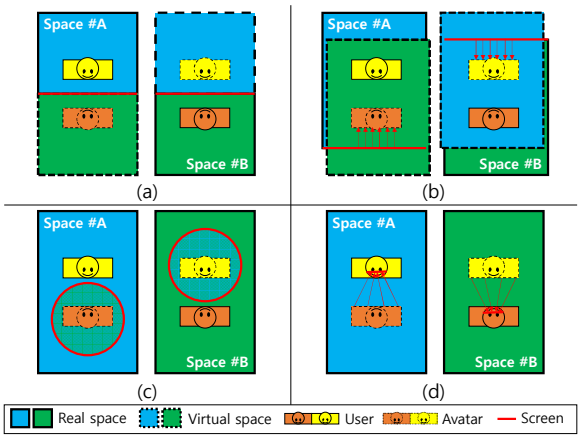

Figure 2: *Comparison among conventional remote collaboration system (a-c) and our approach (d). We note the 2.5D problem usually happens in other system.*

A silver bullet to this problem is See-Through (ST) HMD. It can overcome 2.5D and occlusion problems since the screen is closely located in front of the user's eye. See-through HMD itself can be further differentiated by optical see-through and video see-through. Optical ST HMD offers an unhindered real-world view and the virtual object is overlaid on the real-world view, but the virtual object appears

like floating on the air. While Maimone et al. [MYD*13] used projector novelly to make a dark area for the virtual object so that it appears opaque, it cannot be easily adapted to an uncontrolled situation that we aim. Therefore, we chose Video ST HMD, and adopted depth mask generation using short range depth camera [PAK*14, HFW14]. With our approach (Figure 2d), our avatar representation is completely virtual imagery augmented on user's HMD without requiring physical surrogate.

In summoning remote avatar in arbitrary local space, it becomes important on how to register virtual objects in MR space with HMD localization. Unfortunately, many existing systems use the fixed coordinate system predefined by authors such as HMD-centered mid-air [JNC*15], table-top [PAK*14], and small indoor space [HFW*14]. To overcome this problem, we combine local tracker based on simple marker tracking with global HMD tracker [BW10], hence our system enables novice user to adjust coordinate easily.

2.2 Natural Interaction for HMD-based System

The most common and ideal way for 3D user interface is direct interaction using bare-hand and finger. Humans are familiar with using their hand for daily tasks. Besides, human's fingers are very dexterous and have a high degree of freedom (DoF). However, providing hand interaction in MR is very challenging due to the difficulties of tracking hand and fingers in real time with enough robustness. Conventional devices for tracking hand are data glove and mocap system with IR markers (e.g., Vicon). These devices are very expensive and hinder the naturalness of user experience. A common alternative is by using low-cost camera combined with a computer vision technique to track the hand and fingers [EBN*07] in real time.

Recently, several state-of-the-art methods that enable the tracking of user's hand in high DoF are presented by using a single commercial RGB-D camera only. Despite many promising results [SKR*15] are presented, typical vision-based hand tracking algorithm assumed exocentric camera placement, which is not suitable for hand interaction when wearing a HMD. G-SIAR [PAK*14] is an AR based object manipulation system that uses similar hardware to our system, but they rely on an external top view depth camera to capture and track the hand (based on 3Gear system) instead of utilizing the HMD-attached camera. In other words, their approach has a spatial limitation because it is not possible to interact with objects outside the fixed region.

Egocentric based hand tracking is perhaps more suitable for HMD-based hand interaction because user's hand often follow where the head direction is facing. However, egocentric hand tracking is more challenging because of self-occlusion problem [JNC*15]. Recently, Ha et al. [HFW14] presented a system that allows the user to manipulate virtual objects with bare-hand through proxy virtual hand in wearable AR environment. Introducing proxy hand in AR environment permits the user to manipulate distant virtual objects, but it may be confusing for the user. Meanwhile, those representations is not be useful in the arm-reachable area.

3. System Design

3.1 Conceptual Design and Considerations

The main goal of our system is to support immersive and intuitive remote collaboration using direct hand-based interaction. Using summoned avatar as a representation of the remote collaborator, it is able to mirror the motion of its owner. In addition, the user's local space becomes a coexistence-space [YKN*14] where local user and remote users share the virtual objects and manipulate them together.

Here we describe the main considerations of our framework design. First, we aim to overcome the 2.5D problem that usually occurs in previous systems. Even the state-of-the-art systems [BKKF13, YKN*14] still set to a single display in front of the user, where the view direction and working space is severely limited. In contrast, we employ an HMD as our primary display. With this choice, we can avoid the 2.5D problem and summon remote user as avatar into the local space. Hence, the collaborative space is not limited in front of screen but is expanded into the user's local space.

Next, we try to design the framework using only commodity hardware for utilizing the system in not-in-a-lab context. Existing systems usually employ environmentally tethered sensors and displays [BKKF13]. Although those approaches achieve highly accurate capturing and 3D reconstruction, they have a serious disadvantage on flexibility and cost.

Lastly, we focus on the actual collaboration tasks with remote user in the co-space, not merely enhancing the presence as explored in previous works. Although some systems allowed collaboration tasks, but the remote users are nonetheless limited inside the screen [BJW12] or only the hand part is shown [TAH12], thus limiting the immersion.

3.2 Hardware Configuration

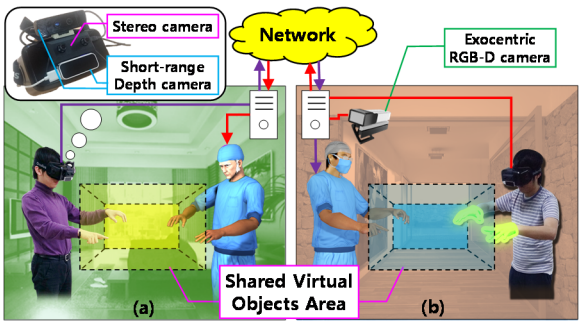

Figure 3: *Possible hardware configuration. (a) Portable setup which does not use exocentric camera and (b) residential setup which adopts exocentric camera.*

Our system requires minimal hardware (Figure 3): A computing unit, a see-through HMD and a short-range depth sensor. An optional exocentric RGB-D camera can be included. The HMD provides an immersive view of virtual objects and avatar while the stereo camera supports the see-through view of real-world to the user. The short-range depth sensor supports hand tracking and permits the user to interact with virtual objects. It can be also used for generating an occlusion

mask [PAK*14]. An optional exocentric RGB-D camera tracks and maps the full body motion of the local user to an avatar which then appears at the remote user's space. Finally, a computing unit responsible for the rendering, managing the coordinate system in co-space, and communicating with the collaborator's side through network.

3.3 Management of Shared Object Area in Co-Space

Without relying on the environmentally tethered sensors and display, it becomes important how to track and register the coordinate system for maintaining co-space. Conventional object-based localization method commonly used in AR system is not suitable for our system because the marker needs to remain inside user's viewpoint, and it causes the aforementioned 2.5D problem. Thus, we take a hybrid approach inspired by Baek and Woo [BW10] to localize user's HMD pose and register co-space locally.

In our hybrid approach, we adopted two types of tracker: an outside-in global tracker and an inside-out local tracker. The global tracker has more flexibility as the marker can be tracked as long as it is inside the defined space. In local tracker, however, the marker must remain in sight at all time, thus limiting the camera view direction. Although global tracker lifts the restriction on the user's viewpoint problem, it does not allow to register co-space in user's world coordinate. For this purpose, we utilize local tracker for registering the local marker as the basis of co-space (Figure 4). For example, a user only needs to watch a local object during the beginning of remote collaboration session. For this reason, we generally utilize global tracker for our system, and only utilize the local tracker once during the initialization stage, thus getting the best of both worlds.

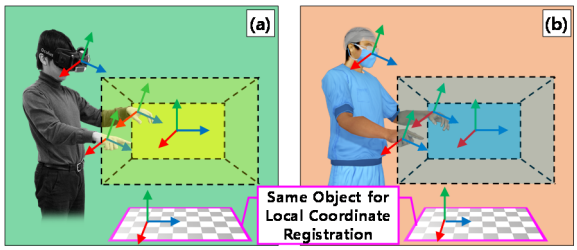

Figure 4: *Co-space in local (a) and remote (b) space. Remote user's avatar is augmented by relative pose derived from the relationship between HMD and registered object.*

Once the pose of the local object is registered in the global tracker, co-space coordinate information can also be calculated in a straightforward way (Figure 4). Registered object pose becomes the basis of shared virtual objects in user's space. User's joint data is also converted to local coordinate based on basis object pose, and send to remote user's space.

4. Implementation and Results

We set up our prototype environment using commodity devices and sensors. To construct a low-cost see-through stereoscopic HMD, we employed the Oculus Rift DK2 and attached a stereo camera module provided by OvrVision. The Oculus DK2 supports position and rotation tracking with an external IR camera. For near-range depth sensor, we experimentally adopted two cameras: Creative Senz3D and LeapMotion. Finally, a Kinect v2 sensor is used as the exocentric RGB-D camera. All hardware are available on both sides except for the Kinect, which is installed on the residential side only, as shown in Figure 3b. The reason is for supporting asymmetric setup and comparing avatar representation with and without full body tracking, which will be explored in section 4.2. In short, the user only needs to wear a single HMD without holding any additional devices.

4.1 Software Modules

We implemented our initial prototype in Unity Engine. Figure 5 illustrates a conceptual relationship among our software modules. Most image processing modules such as chessboard tracker and mesh generator are implemented in C++ with multi-threading enabled for better performance. Then, it is imported as binary plugin for Unity. In addition, we imported Intel hand skeletal tracking library (HSKL) [MKO13] and LeapMotion V2 skeletal tracking API for SenZ3D and LeapMotion sensor, respectively. Lastly, we use official Kinect for Windows SDK and RUIS toolkit [Tak14] for avatar related body tracking.

Through the HMD, the user can see virtual stereoscopic images overlaid on the real-world background images which are captured by the stereo camera. These real-world images are originally acquired by fisheye lens, so undistortion and rectification are needed. One of the undistorted images is used as an input image of the local tracker. In our initial implementation, we used the left camera image. Hence basis of the transformation in calibration stage is $\mathbf{T}_{rgb\_L}$.

Virtual stereoscopic images are rendered by Unity Engine based on virtual left ($\mathbf{T}_{rgb\_L}$) and right ($\mathbf{T}_{rgb\_R}$) camera pose which are sub-transformation of HMD pose ($\mathbf{T}_{hmd}$). We also set the root of the depth camera as $\mathbf{T}_{depth}$ to manage the information comes from depth camera such as articulated hand tracking and real-world depth. For tracking $\mathbf{T}_{hmd}$, we utilized the built-in tracker of Oculus DK2.

Our system utilizes the hand tracking result of local user for manipulating and interacting with virtual objects. The hand tracking information and head pose are sent to the remote space through network in real-time. In remote space, this information are replicated by avatar motion, allowing users to collaborate with a high level of presence.

Masking mesh. Occlusion handling is a non-trivial problem in see-through HMD. We generate a masking mesh for handling the occlusion of the hand between user's eye and virtual objects. First, we convert depth image into 3D point cloud and then generate a mesh by simple triangulation on this point cloud. By changing the shader, the generated masking mesh can be half or full transparent. If the shader is half transparent [PAK*14], the user can see through the hand. Else, if the shader is full transparent [HFW14], it generates a void area in the virtual image, so that the user's real hand is fully visible in the HMD view, and it blocks anything behind the hand. In our pilot test, full transparent shader is more preferred, as it present an experience more closely related to how a human perceives the real world.

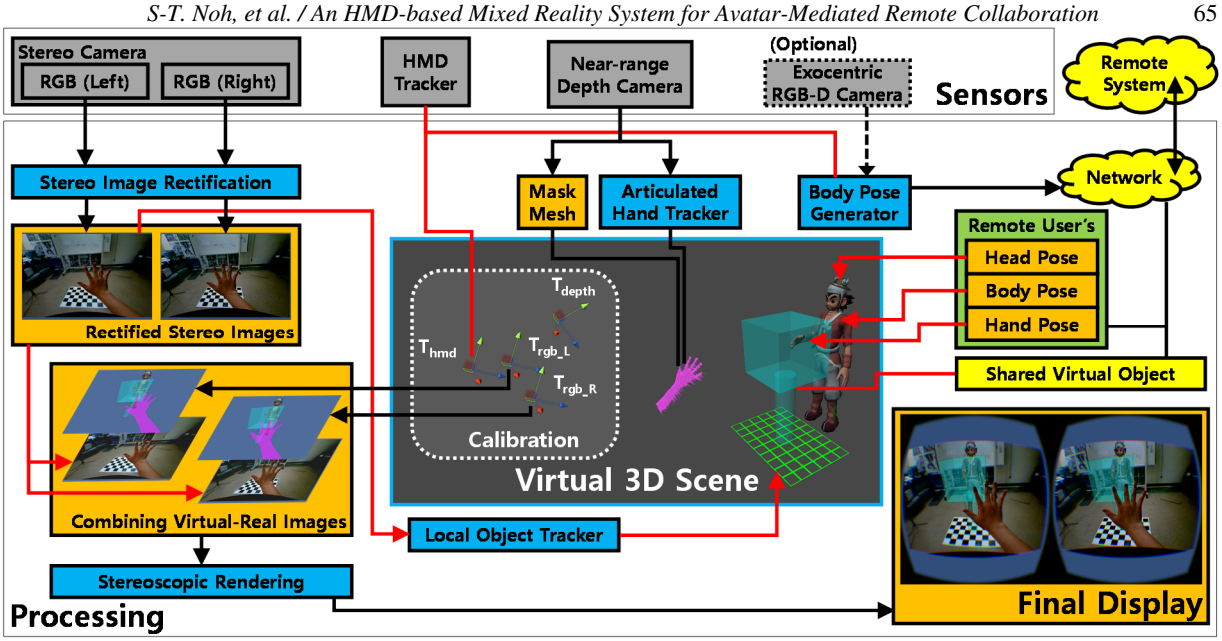

Figure 5: *Software diagram of our proposed concept and our initial implementation.*

Articulated hand tracking. Despite vision based hand tracking has been a popular research topic recently [EBN*07, SKR*15], the number of available library is still amazingly low. Therefore, we use Intel HSKL and LeapMotion API, which are widely available. We found that LeapMotion tracking to be superior in many aspects, especially in HMD use-cases with both hands visible (egocentric view). Thus, for the rest of the paper, we use the LeapMotion for tracking the hand, while we keep the Intel Senz3D sensor for generating the masking mesh that support occlusion handling.

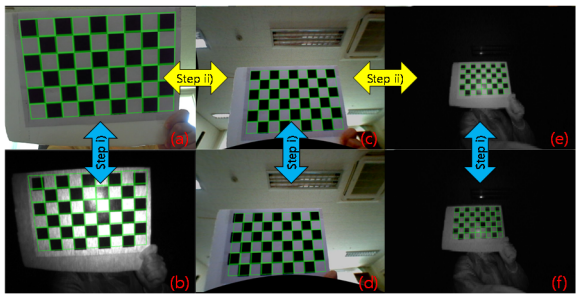

Figure 6: *Calibration result. These images are captured by each camera on the HMD, (a,b) Creative Senz3D, (c,d) OvrVision and (e,f) LeapMotion. Tracking is only done by stereo left camera, and the other overlay images are calculated by calibration result.*

Calibration. Before the platform is ready for use, a 2-steps calibration process to acquire the intrinsic and extrinsic parameters of the cameras is needed: i) calibration inside the same module ii) calibration between modules.

In the first step, we calibrate the two cameras included in each module. In case of OvrVision and LeapMotion, provided tools and predefined values are accurate enough for our purpose, so we utilize it as it is. In case of Senz3D, the intrinsic parameters were acceptable but extrinsic parameters are not provided explicitly. In addition, the mapping function provided in SDK is not accurate enough. Therefore, we calibrate this module by ourselves using chessboard.

In the second step, we calibrate cameras in different camera modules. After the first step, it is certain that two cameras in the same module are well calibrated. Therefore, we only need to focus on the calibration among representative cameras in each module (Figure 6a, 6c and 6e), by calculating relative pose between cameras. Finally, we verify and show our calibration result in Figure 6.

4.2 Fusing the Hand and Body Tracking Data

An external body tracking sensor such as Kinect can track the full body skeleton with 25 joints. With these data, we can control and scale a virtual avatar according to the real user (Figure 7a). Thus, the avatar mirrors the body motion and real-world size of the tracked user. Due to limited hand tracking supported by Kinect, we rely on LeapMotion for full articulated hand tracking. Since the hand and body tracking data come from different sources, we need to fuse them together so that the final result looks natural when viewed from local or remote side (Figure 8a, 8b).

A straightforward approach is to attach the hand information acquired from LeapMotion to the Kinect wrist joint position. This approach works well in most cases, except when the hand is pointing towards the Kinect camera. In this occurrence, the occlusion problem causes the wrist and elbow joints tracked by Kinect to be rather unstable. In result, the hands of avatar appear trembling when viewed by the remote user. To avoid this, we retarget the forearm tracked by Kinect to the palm tracked by LeapMotion, as the latter is more stable. We fallback to the Kinect hand position when the hand is out of view of the LeapMotion.

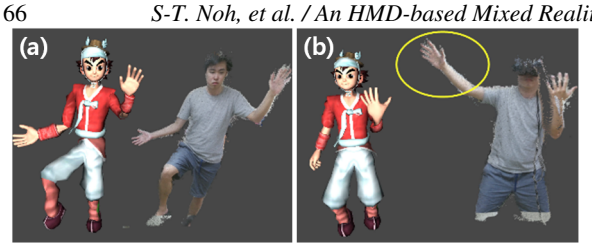

Figure 7: *a) Avatar control with full body tracking using Kinect. b) Limited avatar control with only hand and head pose information using IK. Note that the right hand is not tracked because it is outside LeapMotion tracking area.*

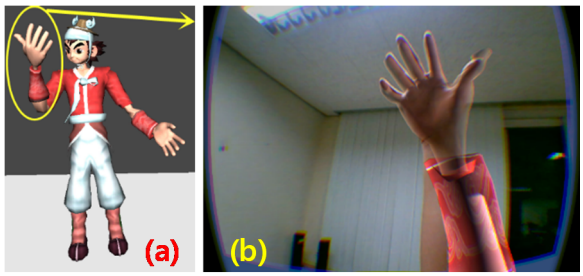

Figure 8: *a) Sensor fusion between Kinect and LeapMotion b) First person view from the yellow circled area, the virtual hand is overlaid and aligned with real hand view.*

Since Kinect is optional; in portable setup that does not include Kinect (Figure 3a), we cannot provide a full-body controlled avatar because only the head and hand information are known. Nonetheless, even with this limited information, it is possible to utilize inverse kinematics (IK) provided in Unity engine to generate human-like motion for the upper-body especially on the arm and elbow joint (Figure 7b). By using the head position and the floor plane, it is also possible to generate lower-body motion such as walking or crouching with predefined animation. Although this is not a perfect representation of the remote user, the result appears to be quite acceptable because lower body motion is often not needed in simple collaboration tasks, but mainly for visualization and immersion purpose.

4.3 Remote Avatar in Local Space

As both the local and remote users share the co-space, it is straightforward to just summon the remote user as virtual avatar into local space. Initialization of the avatar in the real world is done by placing a chessboard marker on the floor plane. For simplicity, we re-purpose the chessboard tracker from Section 4.3, although other image-based tracker is possible. This marker acts as a virtual anchor for the summoned remote space, and can be physically repositioned by the local user as desired. It also generates a virtual floor plane that align with the real world floor plane. Hence the summoned avatar can be placed nicely on top of this plane.

In terms of networking, we only send the HMD pose and skeleton joint data from the body and articulated hand to the other side. Thus, the bandwidth requirement is relatively light compared to other telepresence system. While it is possible to capture, send and visualize the live view of the user by means of colored point cloud or mesh, the obvious drawbacks are heavy processing and bandwidth usage. Moreover, the user's face is still blocked by the HMD.

To make matters worse, only 2.5D point cloud (Figure 9) is visualized due to our inexpensive environment setting that uses only single exocentric RGB-D camera. Using a multiple camera setup may overcome this limitation, but it usually requires complex setup and calibration process, let alone the increased cost. Even in a state-of-the-art setup with multiple cameras [BKKF13], the quality of point cloud is still not perfect due to interference and problem in stitching.

Figure 9: *Our experimental visualization of the remote user using colored point cloud. Although frontal view (a) has acceptable quality, it suffers from 2.5D problem in a different viewpoint, especially from the side (b).*

4.4 Overall System Performance

We measured the system performance on our prototype computer with an i7-5820k CPU, 16 GB RAM, and GTX-970 GPU. On average, each frame takes 11 ms to process, which effectively results in 90 FPS, as shown in Figure 10. Acquiring and rectification of the stereo images from OvrVision takes about 9ms whereas the other processes has negligible processing time (<1ms each). The body tracking is capped at 30fps due to Kinect's limitation. On the network bandwidth, it takes about 4.8Mbits per second as we are sending all joints data in real-time, uncompressed (17 finger joints and the head pose at 60 fps whereas 25 body joints at 30fps).

As a comparison, we also tested sending the colored body point cloud data (subtracted from background scene). Each frame is about 4.5Mbits depending on the body size and distance towards the sensor (as body nearer to the sensor is larger). Due to heavy processing, it is only able to process about 15 frames per second which results in 67.5 Mbits per second, not including routing overhead. We tested in our local area network (LAN) with 100Mbps Ethernet, and it almost saturates the network bandwidth. To achieve 30fps, it will require a faster computer and Gigabit internet connectivity. Even at 15fps only, we felt that it is too demanding for real world usage, not taking into account the latency and packet loss when transmitting across the Internet. Thus, we argue that our avatar approach is better in many aspects.

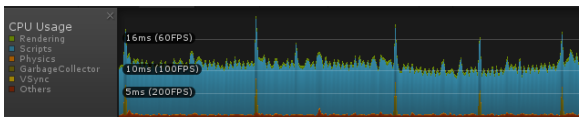

Figure 10: *Overall performance profiled by Unity profiler.*

5. Discussions

5.1 Applications

We implemented a few sample applications to showcase potential scenarios for remote collaboration (Figure 11). In the first scenario, local and remote users collaborate on a puzzle made from a board of tiles. The user can use fingers to touch and flip the tiles or pinch to move it in 3D space.

In the second scenario, we reproduce the remote surgery simulation we envisioned in the introduction section. Remote users diagnose a Human body with internal organs and perform surgery operation. The more expert user might guide another novice user by using a mixture of deictic and metaphorical gestures such as pointing, rapid hand movement, and physical shaping action. The novice user is able to see the action clearly and mimic it accordingly.

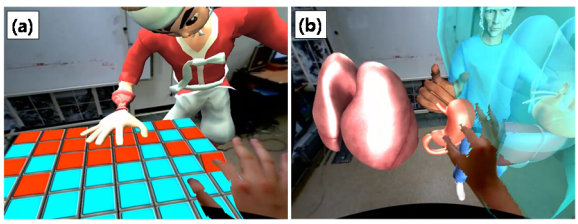

Figure 11: *a) puzzle solving b) surgery simulation.*

5.2 Observations

From our preliminary test with students who have worked related to AR and VR, we have gathered several characteristics of our system. We also found several open problems especially in the component modules, which we categorize and explain the details below.

Face-to-face to side-by-side. Most remote collaboration systems assumed face-to-face situation only, and set up the equipment based on this assumption. As a consequence, the system usually suffers from 2.5D problem. In our portable setting, however, we can support a smooth transition from face-to-face situation to side-by-side which tightly follows the real user movement in the physical world (refer supplementary video). Other systems support similar functionality but require rotating the anchor manually using a controller.

Live capture vs. virtual character. There is some controversy about the appearance of the avatar. Expert users understand that for the convenience of portable setup, there is no camera for capturing a live view of the user. They also agreed that there is no absolute necessity to visualize live capture data for scenarios that focus more on solving the collaborative tasks instead of emphasizing face and eye contact. However, they also concern in terms of technology acceptance to novice users as they noted that the characters used in our implementation are too cartoonish and far from representing the real person's appearance. Thus, it might give a false impression to novice users that they are collaborating with an AI instead of a real person on the remote side. Finally, due to the tracking instability of Kinect and LeapMotion, the avatar's body and hand tremble slightly.

Shared virtual area and height of the user. We found that the difference of user's height is a considerable factor in general collaboration context. For instance, if two users have extremely different heights, a simple arrangement of the shared virtual object area makes the either user uncomfortable to reach those objects by hand. While it is also possible to reposition the avatar to either sink into the floor or to float on the air to accommodate another user, it might appear unnatural and unrealistic, thus breaking the level of presence.

This phenomenon not only happen in standing situation only but also in a various situation such as table-top. It may be more complex because we need to consider not only user's sitting height but also the height of table and chairs.

5.3 Limitations of Initial Implementation

In our initial implementation, we utilized Oculus DK2 tracker as the global tracker. While it is possible to change viewpoint freely, it is still limited inside the FOV of external IR camera. One potential solution would be using SLAM-like inside-out environmental tracking approach, but it requires heavy computation and suffers from drift.

In the portable setup without external Kinect and using IK, it is hard to differentiate body actions such as body leaning forward or walking forward, given that only the head and hands information are known. Although it is possible to use the head orientation to predict the intended action, we found that it is still rather unreliable.

Although we have demonstrated our prototype, we have ultimately tested in LAN environment only. We are uncertain on how the system handles a long distance network with high latency and packet loss. Nonetheless, given that the bandwidth requirement of our system is relatively lightweight, it should perform reasonably fine in the real world.

6. Conclusion and Future Work

In this paper, we proposed a novel framework for immersive remote collaboration system that supports direct hand interaction. We implemented a prototype that explored the possibilities and potential of our system. The results and discussions gave insights into a practical remote collaboration system and serve as guidelines for further improvement.

It is likely that our current prototype is sufficient for many applicable scenarios, yet there are many possible directions to be explored. One promising direction is on extending the number of collaborators. In this paper, we only focused on the one-to-one scenario, but our system has scalability and can be easily extended. We can explore the group-to-group scenario [BKKF13] and collaboration among three or more different physical spaces in the future.

Further exploration on the utilization of full articulated hand tracking result is needed. The human hand has high DoF, which can be useful in many scenarios in MR. Yet, most of research including our work only shows simple usage which cannot highlight the merit of articulated hand.

We have used the virtual character in our implementation and argue that it is better than live capture in 2.5D. However,

it may give a false impression of interacting with AI instead of a real human. One alternative approach is to generate a realistic avatar based on the actual appearance of people captured by exocentric camera. Generating and transferring this realistic avatar should be done before or at the initialization step of remote collaboration to save bandwidth.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP, MEST) (2010-0029751, NRF-2014R1A2A2A01003005)

References

[BGR*98] BENFORD S., GREENHALGH C., REYNARD G., BROWN C., KOLEVA B.: Understanding and constructing shared spaces with mixed-reality boundaries. *ACM Trans. Comput.Hum. Interact. 5*, 3 (Sept. 1998), 185–223.

[BJW12] BENKO H., JOTA R., WILSON A.: Miragetable: freehand interaction on a projected augmented reality tabletop. In *Proc.* CHI (2012), ACM, pp. 199–208.

[BKKF13] BECK S., KUNERT A., KULIK A., FROEHLICH B.: Immersive group-to-group telepresence. *IEEE TVCG on 19*, 4 (2013), 616–625.

[BW10] BAEK W., WOO W.: Efficient inter-camera management for multiple objects tracking in mobile ar environments. *In Proc. of ICAT 2010*, 89–95.

[EBN*07] EROL A., BEBIS G., NICOLESCU M., BOYLE R. D., TWOMBLY X.: Vision-based hand pose estimation: A review. *Computer Vision and Image Understanding 108*, 1 (2007), 52– 73.

[GAB99] GIBBS S. J., ARAPIS C., BREITENEDER C. J.: Teleport–towards immersive copresence. *Multimedia Systems 7*, 3 (1999), 214–221.

[GWN*03] 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 TOG* (2003), vol. 22, ACM, pp. 819–827.

[HFW14] HA T., FEINER S., WOO W.: Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *Proc. ISMAR,* (2014), IEEE, pp. 219–228.

[IK92] ISHII H., KOBAYASHI M.: Clearboard: a seamless medium for shared drawing and conversation with eye contact. In *Proc. CHI* (1992), ACM, pp. 525–532.

[JNC*15] JANG Y., NOH S.-T., CHANG H. J., KIM T.-K., WOO W.: 3d finger cape: Clicking action and position estimation under self-occlusions in egocentric viewpoint, *IEEE TVCG on 21*, 4 (2015), 501–510.

[KPB*12] KUSTER C., POPA T., BAZIN J.-C., GOTSMAN C., GROSS M.: Gaze correction for home video conferencing. *ACM Transactions on Graphics (TOG) 31*, 6 (2012), 174.

[MF11] MAIMONE A., FUCHS H.: A first look at a telepresence system with room-sized real-time 3d capture and life-sized tracked display wall. In *Proc. of ICAT 2011*, 4–9.

[MKO13] MELAX S., KESELMAN L., ORSTEN S.: Dynamics based 3d skeletal hand tracking. In *Proc. GI* (2013), Canadian Information Processing Society, pp. 63–70.

[MTUK95] MILGRAM P., TAKEMURA H., UTSUMI A., KISHINO F.: Augmented reality: a class of displays on the reality-virtuality continuum, 1995.

[MYD*13] MAIMONE A., YANG X., DIERK N., STATE A., DOU M., FUCHS H.: General-purpose telepresence with headworn optical see-through displays and projector-based lighting. In *Proc. VR, IEEE* (2013), , pp. 23–26.

[OSS*13] OYEKOYA O., STONE R., STEPTOE W., ALKURDI L., KLARE S., PEER A., WEYRICH T., COHEN B., TECCHIA F., STEED A.: Supporting interoperability and presence awareness in collaborative mixed reality environments. In *Proc. VRST* (2013), ACM, pp. 165–174.

[PAK*14] PIUMSOMBOON T., ALTIMIRA D., KIM H., CLARK A., LEE G., BILLINGHURST M.: Grasp-shell vs gesture-speech: A comparison of direct and indirect natural interaction techniques in augmented reality. In *Proc. ISMAR,* (2014), IEEE, pp. 73–82.

[SKR*15] SHARP T., KESKIN C., ROBERTSON D., TAYLOR J., SHOTTON J., LEICHTER D. K. C. R. I., WEI A. V. Y., KRUPKA D. F. P. K. E., FITZGIBBON A., IZADI S.: Accurate, robust, and flexible real-time hand tracking. In *Proc. CHI* (2015), ACM.

[STB*12] STEED A., TECCHIA F., BERGAMASCO M., SLATER M., STEPTOE W., OYEKOYA W., PECE F., WEYRICH T., KAUTZ J., FRIEDMAN D., ET AL.: Beaming: an asymmetric telepresence system. *IEEE computer graphics and applications*, 6 (2012), 10–17.

[TAH12] TECCHIA F., ALEM L., HUANG W.: 3d helping hands: A gesture based mr system for remote collaboration. In *Proc. VRCAI* (2012), ACM, pp. 323–328.

[Tak14] TAKALA T. M.: Ruis: a toolkit for developing virtual reality applications with spatial interaction. In *Proc. Spatial User Interaction* (2014), ACM, pp. 94–103.

[TDY*11] TSUI K. M., DESAI M., YANCO H., UHLIK C., ET AL.: Exploring use cases for telepresence robots. In *HumanRobot Interaction (HRI), 2011 6th ACM/IEEE International Conference on* (2011), IEEE, pp. 11–18.

[VKL*11] VASUDEVAN R., KURILLO G., LOBATON E., BERNARDIN T., KREYLOS O., BAJCSY R., NAHRSTEDT K.: High-quality visualization for geographically distributed 3-d teleimmersive applications. *Multimedia, IEEE Transactions on 13*, 3 (2011), 573–584.

[YKN*14] YOU B.-J., KWON J. R., NAM S.-H., LEE J.-J., LEE K.-K., YEOM K.: Coexistent space: toward seamless integration of real, virtual, and remote worlds for 4d+ interpersonal interaction and collaboration. In *SIGGRAPH Asia 2014 Autonomous Virtual Humans and Social Robot for Telepresence* (2014), ACM, p. 1.