Integrating AR and VR for Mobile Remote Collaboration

Jeremy Venerella[1], Lakpa Sherpa[1], Tyler Franklin[1], Hao Tang[1,2], Zhigang Zhu[2,3]

[1]Dept.of CIS, Borough of Manhattan Community College - CUNY

[2]Visual Computing Lab, The City College of New York – CUNY

[3]Dept. of Computer Science, The CUNY Graduate Center

ABSTRACT

In many complex tasks, a remote expert may need to assist a local user or to guide his or her actions in the local user's environment. Existing solutions also allow multiple users to collaborate remotely using high-end Augmented Reality (AR) and Virtual Reality (VR) head-mounted displays (HMD). In this paper, we propose a portable remote collaboration approach, with the integration of AR and VR devices, both running on mobile platforms, to tackle the challenges of existing approaches. The AR mobile platform processes the live video and measures the 3D geometry of the local environment of a local user. The 3D scene is then transited and rendered in the remote side on a mobile VR device, along with a simple and effective user interface, which allows a remote expert to easily manipulate the 3D scene on the VR platform and to guide the local user to complete tasks in the local environment.

Keywords: augmented reality, virtual reality, remote collaboration, 3D mesh.

Index Terms: Human-centered computing—Human computer interaction—Interaction paradigms—Mixed/augmented reality

1 INTRODUCTION

Although communication and collaboration is the most effective for people in close proximity, we have to spend a large amount of our time apart from each other to avoid long-distance commuting and traveling. Nowadays, ubiquitous smartphones play an important role in tele-communications, yet the interactions are limited to written, verbal and video content, which are sometimes insufficient for workplace tasks. Examples of such tasks may include a customer calling a call center to help troubleshoot a PC, a novice technician calling an expert for guidance in repairing a complex device (e.g., a car engine), or a medical expert guiding a local doctor through a complex procedure.

Fussell et al. [21] proposed a shared visual representation of a local user's space, including the task object, tools used, and a view of the local user's actions to simplify the communication. The collaboration with video-mediated dialogs, however, was less efficient since the remote helper could not refer efficiently to task objects in the local environment. Therefore, Kirk et al. [22, 23] and Gergle et al. [24] have tried to address this issue by using either monitors or projectors to overlay remote helpers' gestures on top of the local video stream to be sent back and shown to the local users on a display. This may, however, cause a fractured ecology in which the worker needs to split attention between the task and the external display [25].

email: jeremy.venerella@stu.bmcc.cuny.edu
email: lakpa.sherpa5382@gmail.com
email: tylerjasonfranklin@gmail.com
email: htang@bmcc.cuny.edu
email: zzhu@ccny.cuny.edu

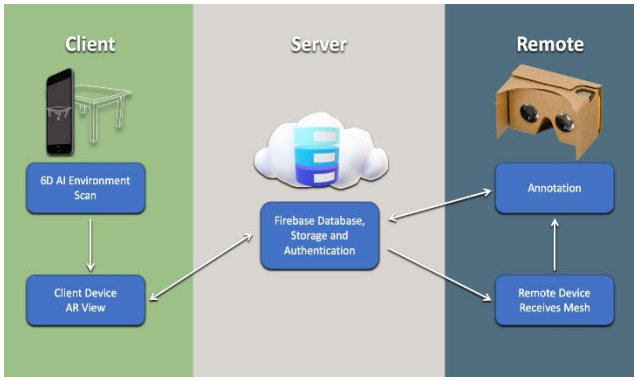


Figure 1. The overview of the proposed MCoAVR system.

Augmented Reality (AR) allows users to see virtual objects seamlessly superimposed over the real world, which allows for co-located users to view shared 3D virtual objects that they interact with, or a remote helper to annotate the live video feed of a local worker, enabling them to collaborate from a distance [4]. Similar AR-based collaboration systems are proposed in [7, 10, 17]; in each of these systems, a remote helper is able to guide the local user in complex physical tasks. Such an AR system can stick the annotations needed to the original positions in the 3D world.

Recently, head-mounted displays (HMDs) have become increasingly popular as immersive display devices for remote collaboration. AR-based HMDs, including HoloLens and other smart glasses, offer users see-through experience, whereas VR-based HMDs, such as Oculus Rift and HTC Vive, offer fully immersive and large field-of-view VR experience. In [8], video of local workspace is captured and sent to a VR-based HMD worn by the remote user. AR-based smart glasses are used in [9] for both local and remote users in collaboration.

In remote guidance scenarios, a remote user often has to view the local scene from the point of view of the local user. A few studies [5, 13] present collaborative AR systems that allow the remote user to have an independent view into the shared task space. A 3D model is reconstructed by a depth sensor and rendered on a VR HMD worn by the remote user. This allows the remote users to view and annotate the 3D scene from a different perspective, leading to faster task completion. More studies follow the idea in [5, 13] to provide 3D models of local scenes [6, 11, 15, 20], and use either independent depth sensors or HMDs with embedded depth sensor (e.g., Microsoft HoloLens) for the local user.

To render smooth and high-quality 3D models on the remote side, high-end VR HMDs are widely used. Although the high-end VR HMDs provide high frame rate, low latency, large field of view (FOV) and high visual quality, they often require wired connections to a high-end workstation, which may not be feasible in many real-world applications. In [14], a mobile remote assistance solution is offered as a commercial product, which streams video while local user stays focused on one area but the remote user's annotation is limited on 2D, making it ineffective in many complex 3D

environments. While high-end AR HMDs provide users with decent AR experience, but they are costly.

For the reason above, it would be beneficial to develop a portable and affordable remote collaboration solution with effective communication and interaction. This presents some challenges**:**

Lack of portable depth sensors. Only a few high-end portable devices have depth sensors and offer real-time 3D modeling features, thus allowing remote users with independent perspectives and effective referencing object.

Lack of 3D texture modeling. Even existing AR SDKs [1,2,3] don't provide 3D texture modeling out of box. Without 3D textured modeling, annotations and animations are only limited to 2D image/video and may not be effective for remote assistance in many complex 3D environments.

Lack of effective interfaces. Mobile VR solutions have many limitations in their interfaces and functions, which makes it difficult for remote users to effectively annotate and interact.

In this paper, we propose a completely portable solution, a *Mobile Collaborative Augmented and Virtual Reality* (MCoAVR) to tackle the above challenges and provide simple and effective remote collaboration experiences. The collaboration system offers a more flexible yet affordable solution than the high-end A/VR system. The mobile A/VR solution offers a more self-contained experience, that is easier to use and significantly less expensive and therefore, may provide a more convenient educational/training solution. For example, one local user can easily work with multiple remote users simultaneously. A 3D mesh model reconstructed from the client side using a normal smartphone can be rendered at the remote side via VR. The 3D mesh allows the remote users to learn the position and orientation of objects in the real-world scene and improves the users' spatial perception.

In the following, we first describe our proposed system design and implementation in Section 2 and then we present a preliminary experiment and discussion in Section 3, before we conclude the paper in Section 4.

2 System Design and Implementation

MCoAVR is a multi-user system with cloud storage (Figure 1). A user on the client side uses a mobile AR system (e.g. a smartphone) to scan the local environment and build a 3D model (mesh) of the client's scene. It then uploads the textured 3D mesh to a cloud storage (e.g. Google Firebase platform). Users on the remote side use a mobile VR system (e.g. a smartphone with Google Cardboard) to query the client's 3D scene via the textured mesh available on the cloud, allowing remote users to view client's environment from an arbitrary perspective. The interface on the remote side also allows users to annotate virtual objects and add virtual landmarks in the client's environment, either to guide the client user or to collaborate with client user to complete a complex task. In this paper, both client-side and local users refer to same users who receive assistance from remote users.

2.1 AR: Client Side and 3D Mesh Generation

The mesh generation is provided by 6D.AI [1]. This platform uses AR point cloud data captured during a user's scan to generate a 3D mesh of the surrounding environment from a mobile device. A scan is done by deliberately moving the mobile devices camera to allow for full coverage of the environment with feedback given visually of what is unscanned. The networking (i.e. the cloud storage) is built with Google's Firebase platform. The networked mesh allows a remote user to interact with a client's 3D environment. When a client user designates the desired workspace, detected changes in the mesh results (within a user defined region of interest) in a serialization of the mesh chunks within the bounds of the desired workspace. This serialized data is then uploaded asynchronously into the Firebase storage system. This upload produces a callback triggering a download for the remote user. The downloaded serialized data is then converted back into the mesh format. Figure 2 shows the 3D meshes; the calculated 3D geometry is roughly correct (the house siding is perpendicular to the ground) hence allows the remote user to navigate the local scene in a VR environment by simply moving in the real world (without using remote controls).

The client-side projections are generated during client-side mesh generation. This is done by capturing workspace point of view (POV). Each captured view is automatically sent along with the client device coordinates to the remote user over the cloud server.

2.2 VR: Remote Side and Its Navigation

Once the remote user receives the captured image and coordinates via a callback, a Unity projector [26] is generated at the specified coordinates, projecting the image onto the 3D mesh so remote users can see 3D client environment from any POV.

The remote side runs on a Google cardboard device which comes with very simple interface (one button), hence it's not easy to navigate in the VR world, compared to high-end VR devices with remote controls. This would typically require additional virtual buttons to perform movements in the VR scene and obtain a better POV for more accurate and clearer 3D reference and annotation [16]. Modern desktop VR systems use inside-out tracking for tracking a user's movement without the need over external sensors. This technique can be replicated with Google cardboard and the addition of the ARKit SDK [3].

We apply the ARKit SDK on the rear camera to obtain the user's 6-DOF motion. This allows the remote users to easily navigate the client 3D scene in the VR view by simply moving in the real world using detected feature points across video frames and applying this model to the device.

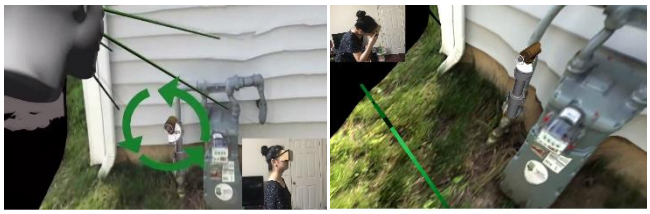


Figure 2. A remote user navigates in a 3D VR environment by simply moving in her office, with different perspectives. The remote user first starts with a normal perspective (left image) and then she moves a few steps, looks down and turns her head to have a better view of the 3D object – the valve (right image).

Using multiple fingers on the screen of a device only allows for 2D manipulation of an annotation, with a loss in depth. With the addition of AR, depth is added in the form of a user's physical movements. These replicate natural movements such as walking over and placing a pencil on a table, but in this case, it is an annotation.

A limitation of using such a solution is the user's physical environment. Any obstructions such as wall or furniture can limits a user's range of motion when placing an annotation in the correct location. Our solution to this is the ***interactive mini-map***. When a user confronts a physical obstruction, the user can move to an obstruction free area and then use the mini-map to move the user's AR transform closer to the desired position. This allows the user an unobstructed space for continued annotation.

2.3 Integration: Interaction Between Client and Remote Sides

Simple and effective interaction and communication is the key to a successful remote collaboration system. The proposed system

includes different types of interaction features: prebuilt/draw annotations; virtual landmarks with 3D surface alignment; and synchronized or independent POVs with the user's head gaze. We will briefly discuss each of them below.

Prebuilt and/or drawn annotations (circular arrows and red flags in Figure 3b) placed or moved in the remote scene automatically send their coordinates, rotation, annotation type, Globally Unique Identifier (GUID), and color to the client side via database callback. Upon receiving the data, the client application generates a new annotation into the client scene via AR (rendered at the same location/rotation as it's annotated in the remote VR view) or updates an existing based on the GUID.

A virtual laser allows the remote user to instruct the client using a beam that is ray casted from the remote user. This allows for quick pinpointing of an object in the scene or to enhance understanding of accompanying annotations.

(a)

(b)

Figure 3. (a) the remote VR view is superimposed to the top left corner of the client AR view, and the head gaze with POV of the client user is rendered in the VR view so the remote users know what their collaborators see; (b) includes a pre-built annotation (green circular arrows) and a virtual landmark with text annotation (red flag with "move pipe" annotation) in the remote VR view. Note the virtual landmark with annotation is aligned with the 3D wall surface for easy visualization.

Virtual landmark (Figure 3b) has proved effective to reference 3D object in the client's scene. Hence, we implement this feature, allowing users on both sides to easily refer to 3D objects. The orientation of virtual landmark is aligned with 3D surface of objects in the local environment (Figure 3b).

Remote users can either choose an *independent POV* by default for better perspective of reference 3D objects, or choose a *synchronized POV* with the client user so users on both sides will have the exact same POV if it's more convenient for them to collaborate. Additionally, a user on one side can see another user's POV and head gaze (Figure 3a), with boundary of what the users can see through their display (Figure 3a). It can inform users of what their collaborators can see.

In addition, multi-user networked annotations allow for the ability of multiple remote users to produce annotations for the client-side user. These annotations are used to help guide the client to complete a real-world task.

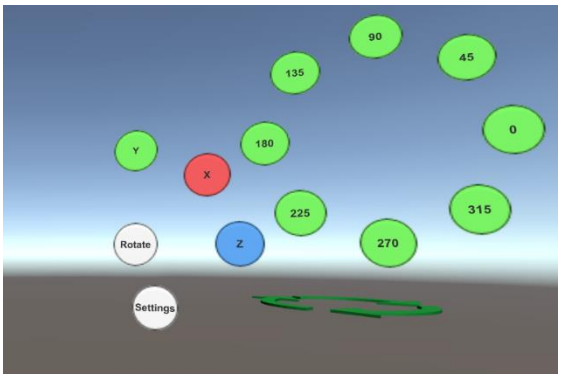


Figure 4. A floating menu to rotate a 3D object in the VR environment at the remote side. The feature allows user to easy interaction with only button on a Google Cardboard.

2.4 Interface: Floating Menus on Google Cardboard

In order to allow the remote user to easily perform annotation on a Google cardboard, we have designed one-click interactivity that the google cardboard VR requires. A *floating menu system* is built upon the unity floating canvas system. The floating canvas system allows for world space UI elements that include images, buttons and text. An initial canvas is created that holds only the initial settings button and sub buttons. Each annotation settings button holds an array of customizable sub buttons such as rotate and instantiate. A sub button is created using a button that performs the desired action on click. These sub buttons also can acts as setting buttons, which allow for multiple nested sub menus. An example of this is the rotation button that has a submenu of an axis of rotation, which in turn has a sub buttons for degrees to rotate (Figure 4).

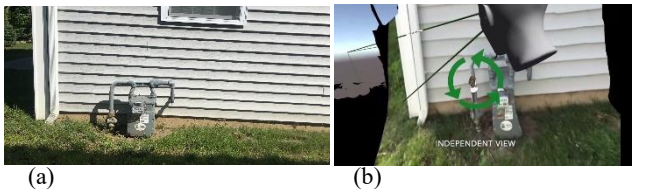
3 Experiment and Discussions

3.1 Preliminary Experiments

We conducted some preliminary experiments using the proposed system. Figure 5 shows an experiment to replace a pipe and valve of a water meter in a residential house. The local platform is running on an iPad. The user walks around the meter and the local system reconstructs 3D meshes and textures of the meter and surrounding scenes. These meshes are sent to the remote user via the cloud server and then the remote user can view the 3D model. The remote user can navigate the 3D model in the VR environment by simply moving in the real world (a room in Figure 2) using Google cardboard and ARKit (Sec 2.2).

In the experiment, the remote user needs to walk around to see the details of the meters from different perspectives and then she uses the floating menu to create a virtual pipe and valve in the VR environment. She also moves and turns her cardboard in her real-world environment until placing the 3D pipe and valve at the correct 3D location, then she uses the floating menu to create 3D annotations. The entire animation (the motion trajectory of the virtual 3D models) has been sent back to the local side at the correct location in local user's real-world environment so the local user can easily follow the animation and annotation to complete the task.

We also performed the same experiment using Skype video chat [19] for collaboration, and a commercial software Vuforia chalk (free evaluation) [14]. The feedback from the participants show the proposed system is easier to understand and follow.



(a) (b)

(c) (d)

Figure 5. (a) an image captured at the local side, (b) a textured 3D model was recovered at the local view and rendered in a VR environment at the remote side; (c) and (d) are the animation to demonstrate the pipe and valve replacement in the local and remote view, respectively. Note, both views have independent perspectives.

3.2 Discussions

There are several issues revealed through-out preliminary experiments: quality of the rendering, latency, and user experience. We discuss each of the issues and propose possible solutions as ongoing and future work.

Quality. One concern is that the quality of 3D mesh (mobile 3D reconstruction using RGB images only) is not high. Is the proposed remote collaboration method still effective with the low-quality 3D model? The experiments reveal the 3D model improves the users' spatial perception in the following aspects.

a) The 3D mesh helps remote users to easily and accurately position a 3D annotation and superimpose a virtual 3D model at a specific 3D location.
b) Because of the available 3D mesh, the proposed system offers users independent views which allows remote users to easily explore the 3D environment from different perspectives.

Latency. Currently the system is configured with a few seconds of latency between the local and remote sides because the local side always needs to update the 3D scan and sends it to the remote user. Now we are working on reducing the latency. Once the first big mesh of the 3D scene has been sent to the cloud, we aim at only send the incremental updates of the local scan so the latency can be reduced and the collaboration process can achieve real-time performance.

User Experience. We plan to conduct a comprehensive user study to test the proposed system applied to complete various remote tasks. We will evaluate the system with a baseline which only uses Skype video chat or Vuforia chalk for a remote collaboration task. The completion time will be recorded for all trails for performance analysis. Additionally, each user will complete a survey, including various questions such as ease of use, fatigue, awareness of remote actions, communication efficacy, ease of annotation and overall satisfaction rate.

4 Conclusion

In this paper, we propose MCoAVR, a lightweight remote collaboration system, with the integration of mobile AR and VR devices, which are more affordable and accessible. A local user uses an AR device to reconstruct simple 3D mesh models of the local environment. The 3D model is transmitted and rendered at the remote side on a mobile VR device, along with a simple and effective user interface. The proposed mobile A/VR solution allows a remote expert to easily manipulate the 3D scene on a mobile VR platform to guide the local user to complete a complex task. Furthermore, the solution offers an easy to use and significantly less expensive option, and therefore may be more practical training tool.

Acknowledgments

This research is supported by the US National Science Foundation via a Smart and Connected Community (S&CC) planning grant (Award #CNS-1737533) and a Partnerships for Innovation (PFI) grant (Award #IIP-1827505), the Intelligence Community Center for Academic Excellence (IC-CAE) Critical Technology Studies Program at Rutgers University, a U.S. Department of Education: Minority Science Engineering Improvement (MSEIP) Grant, and a PSC-CUNY Research Grant.

References

[1] 6D.AI http://6D.AI, last access 6/25/2019
[2] ARCore https://developers.google.com/ar/ last access 6/25/2019
[3] ARKit https://developer.apple.com/arkit/ last access 6/25/2019
[4] Ronald Azuma, (1997). A Survey of Augmented Reality. In Presence: Teleoperators and Virtual Environments, vol. 6, no. 4, pp. 355-385.
[5] Matthew Tait and Mark Billinghurs: The Effect of View Independence in a Collaborative AR System Computer Supported Cooperative Work 24(6) · August 2015
[6] F. Tecchia, L. Alem and W. Huang. 2012. 3D Helping Hands: A Gesture Based MR System for Remote Collaboration. In Proc. ACM VRCAI. 323-328.
[7] Matt Adcock and Chris Gunn. 2015. Using Projected Light for Mobile Remote Guidance. Computer Supported Coop. Work 24, 6 (December 2015), 591-611.
[8] Judith Amores et al. ShowMe: A Remote Collaboration System that Supports Immersive Gestural Communication, CHI 2015
[9] Kevin Wong, HandsOn: A Portable System for Collaboration on Virtual 3D Objects Using Binocular Optical Head-Mounted Display, Master thesis, MIT 2015
[10] Pavel Gurevich, Joel Lanir, Benjamin Cohen, and Ran Stone. (2012). TeleAdvisor: a versatileaugmented reality tool for remote assistance. In Proceedings of the SIGCHI Conference on HumanFactors in Computing Systems. New-York: ACM Press. pp. 619–622.
[11] Thammathip Piumsomboon, Lee, Y., Lee, G. and Billinghurst, M, 2017. CoVAR: a collaborative virtual and augmented reality system for remote collaboration SIGGRAPH Asia 2017 Emerging Technologies, ACM, Bangkok, Thailand, 1-2.
[12] Ranjan, A., Birnholtz, J. P., and Balakrishnan, R. Dynamic Shared Visual Spaces: Experimenting with Automatic Camera Control in a Remote Repair Task. In Proc. of CHI (2007), 1177-1186.
[13] R.S. Sodhi, Jones, B. R., Forsyth, D., Bailey, B.P. and Maciocci, G. 2013. BeThere: 3D mobile collaboration with spatial input. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM, Paris, France, 179-188.
[14] Vuforia chalk, https://chalk.vuforia.com/ last access 6/25/2019
[15] Wang et al. Augmented Reality as a Telemedicine Platform for Remote Procedural Training, Sensors 2017, 17, 2294
[16] Benjamin Nuernberger et al., Multi-view gesture annotations in image-based 3D reconstructed scenes, 22nd ACM Conference on Virtual Reality Software and Technology, 129-138.

[17] Steffen Gauglitz, Benjamin Nuernberger, Matthew Turk, Tobias Höllerer, World-Stabilized Annotations and Virtual Scene Navigation for Remote Collaboration, the 27th annual ACM symposium on User interface software and technology, Honolulu, Hawaii, October, 2014, 449-459.

[18] ARKit World Tracking https://developer.apple.com/documentation/arkit/understanding_world_tracking

[19] Skype, https://www.skype.com/en/ last access 6/25/2019

[20] Piumsomboon, T., Day, A., Ens, B., Lee, Y., Lee, G. and Billinghurst, M. 2017. Exploring enhancements for remote mixed reality collaboration SIGGRAPH Asia 2017 Mobile Graphics & Interactive Applications, ACM, Bangkok, Thailand, 1-5.

[21] Fussell, A. R., Setlock, L. D., Yang J., Ou J., Mauer E., and Kramer, A. D. I. Gestures Over Video Streams to Support Remote Collaboration on Physical Tasks, Human-Computer Interaction, 19: 3, 273 - 309 (2004)

[22] Kirk, D., Fraser, D. S., Comparing remote gesture technologies for supporting collaborative physical tasks. In Proc. CHI 06, ACM Press (2006), 1191-1200.

[23] Kirk, D., Rodden, T., Fraser, D. S. Turn it this way: grounding collaborative action with remote gesture. In Proc. CHI 07, ACM Press (2007), 1039-1048.

[24] Gergle, D., Kraut, R.E., and Fussell, S.R. Action as language in a shared visual space. In Proceedings of. CSCW04, ACM Press (2004), 487-496

[25] O'Neill J., Castellani S., Roulland F., Hairon N., Juliano C, Dai L. From ethnographic study to mixed reality: a remote collaborative troubleshooting system. Proceedings of CSCW11, ACM Press. 225-234. (2011)

[26] Unity Projector, https://docs.unity3d.com/Manual/class-Projector.html