Demo: A Low-Cost Wireless System Implementation for Interactive and Immersive Teaching

Minghui Weng[†], Xiangqi Kong[*], Lianfen Huang[†], Bin Li[*]

[†]Department of Communication Engineering, Xiamen University, China
[*]Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, USA
Email: 23320151154078@stu.xmu.edu.cn, lfhuang@xmu.edu.cn, xqkong@uri.edu, binli@uri.edu

ABSTRACT

In recent years, virtual/augmented reality (VR/AR) technology has received great attention due to its capability of creating various levels of immersive experiences. However, current wireless VR/AR devices are quite expensive, which hinders its large-scale deployment in practice. In this demo, we present a wireless interactive VR/AR teaching system based on popular Android phones. In such a demo, when a teacher explains a 3D heart model, multiple students can see it from exactly the same perspective as the teacher does through VR/AR glasses. When one student has a concern or question regarding a particular part of the heart model, he/she can point it out, and a corresponding blue cursor will appear on screens of all users. Moreover, in the absence of 3D models in Android phones, we broadcast 3D models based on their visual priorities.

CCS CONCEPTS

• **Networks → Wireless local area networks**; • **Computer systems organization → Embedded systems**;

KEYWORDS

Wireless Networks, Virtual Reality, Augmented Reality, Low-cost Implementation, Quality of User Experience

ACM Reference Format:
Minghui Weng[†], Xiangqi Kong[*], Lianfen Huang[†], Bin Li[*]. 2018. Demo: A Low-Cost Wireless System Implementation for Interactive and Immersive Teaching. In *Mobihoc '18: The Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 26–29, 2018, Los Angeles, CA, USA.* Los Angeles, CA, USA, 2 pages. https://doi.org/10.1145/3209582.3225208

1 INTRODUCTION

VR/AR technology creates various levels of immersive experiences and has a lot of promising and exciting applications in entertainment (e.g., interactive gaming and 360-degree spherical videos), education (e.g., immersive education and training), 3D design and art, and social interactions. However, existing commercial VR/AR products, especially wireless VR/AR devices, are quite expensive (e.g., $5,000 for one commercial HoloLens). This greatly hinders its large-scale deployment in practical systems, such as interactive and immersive classroom teaching.

To this end, in this demo, we develop an interactive VR/AR teaching system based on popular Android phones. In particular, in our developed wireless interactive VR/AR teaching system as shown in Fig. 1, when a teacher explains a 3D heart model, multiple students can see it from exactly the same perspective as the teacher does through VR/AR glasses. When one student has a concern or question regarding a particular part of the heart model, he/she can point it out, and a corresponding blue cursor will appear on screens of all wireless devices. The developed system only contains Android phones, self-designed remote controllers, wireless routers, and servers, which are much more affordable compared to the existing commercial wireless VR/AR products.

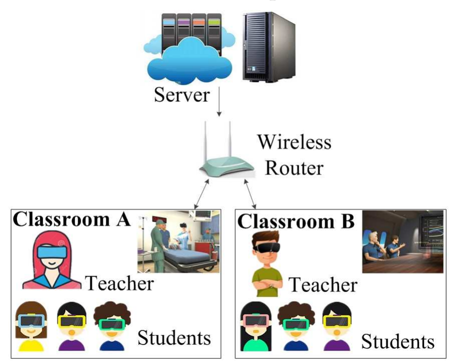


Figure 1: Interactive VR/AR teaching system.

Moreover, in the absence of 3D models in Android phones, we broadcast 3D models based on their visual priorities. Indeed, different from conventional video streaming that delivers contents in consecutive frames, data transmission in VR/AR applications highly depends on user's input movement. In particular, whenever there is an input movement, we expect the corresponding images to be displayed on the screen in time. In the case of 3D building design using VR/AR technology, when users open a door of one conference room, they expect to see inside the conference room immediately. Since the data volume of a whole 3D design typically is large, it causes unacceptable high latency if the data of a whole design is delivered through wireless networks. Instead, it is sufficient to deliver the images around the door in time. More generally,

L. Huang is the corresponding author.

Mobihoc '18, June 26–29, 2018, Los Angeles, CA, USA

ACM ISBN 978-1-4503-5770-8/18/06…$15.00
https://doi.org/10.1145/3209582.3225208

users' immediate and surrounding models should be loaded first and then the rest of models.

2 SYSTEM IMPLEMENTATION

In this section, we discuss the implementation details of our interactive VR/AR teaching system, which consists of remote controllers, Android phones, and cloud/edge servers. In our demo, each remote controller is associated with a unique Android phone and communicates with it through Bluetooth Low Energy (BLE), while each Android phone communicates with an edge server through WiFi, as shown in Fig. 2.

Figure 2: Communication paradigms

2.1 Remote Controller Design

Fig. 3 shows our designed remote controller that contains a STM32F103C8T6 micro-control unit (MCU), a battery, four switches, four buttons, two rockers, fifteen LEDs, and a CC2541 BLE module. The MCU can immediately respond to the actions of switches, buttons or rockers, and then send a control signal to an Android phone through BLE every 20ms for satisfactory user experience (see [1]).

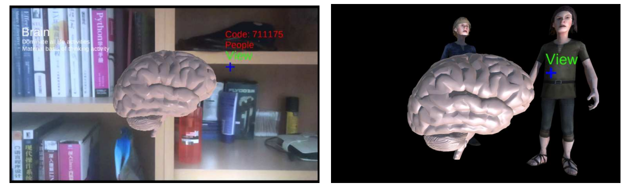
Figure 3: Remote controller

2.2 Android Phone Design Platform

The software design on Android phone basically includes VR displaying or AR rendering, BLE and WiFi communications. In particular, we use the Unity3D to develop virtual 3D models for VR scenario and then further combine with real-time camera background for AR application through rendering frame by OpenGL. Here, the operation instruction is highly time-sensitive and should be sent to the server in time. In addition, some 3D models cannot be fully stored in an Android phone due to its huge volume of data and thus it should be dynamically downloaded from edge/cloud servers. Moreover, whenever there is an action, in order to provide better user experience, the user's immediate and surrounding models should be loaded first and then the rest of models. Fig. 5 shows a snapshot of our VR/AR application.

2.3 Server Platform Design

Our server platform includes the cloud and edge servers, which are configured by Express framework of Node.js and combined with MySQL database to facilitate the data management. In particular, the cloud server provides the storage of all user data and 3D models, while the edge server extracts the required data from the cloud server, so that users can respond quickly to some actions and display corresponding models in time through TCP/UDP connections via WiFi.



(a) AR application (b) VR application

Figure 4: Snapshots of our VR/AR application.

3 DEMONSTRATION

In this section, we show how to operate our interactive VR/AR system in our demo. Our video demo is available at www.ele.uri.edu/faculty/binli/demos/ARdemo.mp4

In our demo, users can watch VR/AR scenes and their real-time operations at the same group. Each user can use its remote controller to select controlling, watching or freedom mode. In the controlling mode, the operation will be shared among users within each group while users can only view the operation in the watching mode. In the free mode, users can operate VR/AR models freely, but other users within the same group will not see the operation.

In addition, users can conduct a series of operations on the VR/AR scenes by rockers and buttons of their remote controllers. In particular, VR/AR operations include the rotation and zooming of 3D models, splitting and combination of 3D models, displaying and hiding of texts or the AR background as well as the resetting operation. Fig. 5 shows some operations of our VR/AR model.

(a) Model rotation (b) Model splitting and combination

Figure 5: Model operations

Moreover, in the absence of 3D models, traditional wireless designs aiming for optimizing the user experience for video streaming need to load and render all large volume of 3D models, and thus users need to wait for a long time to observe an action, which is unacceptable for wireless VR/AR systems. Intuitively, whenever there is an action, users' immediate and surrounding models should be loaded first and then the rest of models so that users just need to wait for much shorter time to see the operations.

ACKNOWLEDGEMENTS

This work is supported in part by Key Laboratory of Digital Fujian on IoT Communication, Architecture and Security Technology (Grant number 2010499), and US NSF grant CNS-1717108.

REFERENCES

[1] 2016. Making Immersive Virtual Reality Possible in Mobile. Qualcomm White Paper (2016).