An Interactive and Immersive Remote Education Platform based on Commodity Devices

Jiangong Chen*  Feng Qian†  Bin Li*

*Department of ECBE, University of Rhode Island, Kingston, Rhode Island, USA
†Department of CSE, University of Minnesota - Twin Cities, Minneapolis, Minnesota, USA
Motivation

• High demand for remote education, especially under the context of the pandemic

• Virtual Reality (VR) – more interactive and immersive experiences

• Main challenges:
  • High-quality panoramic image rendering: typically, 2560x1440 pixels and 60 FPS
  • Synchronization among users: smooth interactions
Wired V.S. Wireless

Wired VR Headset

- High computation capability
- Multi-Gbps link transmission rate
- Low flexibility

Wireless VR Headset

- Low computation capability
- Limited wireless bandwidth
- High flexibility
Synchronization

- **The same view** among teacher and students
- The teacher controls the movement
- The students can interact with the teacher

Frame in 0ms

3ms

Teacher

3ms

RTT=3ms

Edge Server

RTT=30ms

Student 1

RTT=100ms

Student 2

30/2+3/2=16.5ms

100/2+3/2=51.5ms

(Ignore the rendering time)
Synchronization (Cont’)

• If the server knows the trace in the future, it can send diverse frames based on the RTT to different clients instead of the frame corresponding to the current time slot.

The same frame will be displayed at the same time. At each time slot, they can share the same view.
Offline Rendering Engine

• Split the VR world into grids
• Offline generate all panoramic frames in each grid
• Use equirectangular projection to process the panoramic frame
• Compress the frames using the H.264 codec

Grid world

Earth in equirectangular projection

Teacher’s position
Server and Clients

• **Server:**
  • Deliver both the **pose** (6DoF, from the teacher) and the **rendered frame** to both the teacher and students
  • Utilize **motion prediction** to alleviate the asynchronization problem
    • predict the future positions as the same as the **last position**
    • use the **Autoregressive Process (AR) model** to predict the future orientation
  • Collect the **display time** of each frame for each client

• **Clients:**
  • Always wait for frames from the server
  • Send **ACK** to the server when a frame is displayed
  • Periodically send **sampled pose** to the server on the teacher client
Prototype

Teacher

Internet

Edge Server

Ethernet

More students...

Student 1

Student 2

Router

Bluetooth

WiFi

Router

Router

Router
Evaluation

• 5 users
• Add extra RTTs by Linux TC: 0,10,20,30,40ms
• 3 methods: Local Rendering, Server Rendering, Our Design
• Frame Rate: displayed frames per second
• Frame Latency Difference: display time difference of similar frames between the teacher and the students

![Frame Rate](image1.png)

![Frame Latency Difference](image2.png)
Conclusion

• Develop a VR-based remote education platform using commercial off-the-shelf mobile devices
• Provide high-resolution (2560×1440 pixels) panoramic content to the clients with a high frame rate (60 FPS)
• Utilize motion prediction to mitigate the impact of heterogeneous RTTs between clients and the server
• Evaluation results show at least 60% and 20% synchronization improvement compared to existing methods, respectively
• A demo video can be found at https://www.ele.uri.edu/~jiangongchen/demo/VRDemo.mp4
Thank you!