InstantXR: Instant XR Environment on the Web Using Hybrid Rendering of Cloud-based NeRF with 3D Assets

Moonsik Park¹, Byonghyun Yoo²³, Jee Young Moon², Ji Hyun Seo²⁴

¹ESTsoft Corporation, ²Korea Institute of Science and Technology, ³University of Science and Technology, ⁴Korea University
Introduction

Moonsik Park

- Software Engineer, ESTsoft Corporation (2022. 07. - Current)
- Research Intern, Korea Institute of Science and Technology (2022. 03. - 2022. 06.)
- Undergraduate Student, The Catholic University of Korea (2020. 03. - Current)

Research Interest

- Neural Rendering technologies and their applications
Neural Radiance Fields - Idea

- Synthesizing novel views of complex scenes by optimizing an underlying continuous volumetric scene function using a sparse set of input views
- Surprisingly photorealistic output
- Very slow and compute-intensive
  - Several days to optimize NeRF, several hours to render a video with novel views
Neural Radiance Fields - Advances

- NVLabs “Instant Neural Graphics Primitives”
- Sped up NeRF significantly
- Optimizing
  - Simple scene under 5 seconds
  - Complicated scene under 5 minutes
- Rendering (depending on the resolution)
  - 50 ~ 100 ms
Neural Radiance Fields - Applications

Using NeRF to map large environments

Using NeRF depth estimation to navigate robots


Sharing Environments in Extended Reality (XR)


3D Reconstructed Model
Point Cloud Streaming
3D Camera Streaming

Using NeRF to Share Environments

NeRF’s exceptional ability

- Excellent novel view synthesis of complex scenes with photorealistic quality
- No preprocessing necessary other than taking a video of the environment
- Very small output (under 50 megabytes)

Roadblocks exist

- Rendering the novel view is compute-intensive
- Modifying the already optimized scene is hard
- Can’t make modifications to the “rendering pipeline”
InstantXR

Cloud Rendering

- Scalable architecture with a render server and multiple cloud renderers
- Adding objects to NeRF scenes using depth harmonization

Transport

- Novel low latency streaming method
- Low latency head position sharing

Client

- No requirements to the client other than a web standards compliant browser
InstantXR Architecture
Real-time Cloud-based NeRF with InstantXR

Designing a “distributed remote rendering system”

- Lowering Motion-to-Photon latency using a cloud render farm
- ‘Cloud Encoding Server’ paired with multiple ‘Cloud Renderer’s’
- Scheduling required
  - Dropping frames that take too long to render
  - Distributing render requests based on rendering speed, latency, and difficulty to render
  - Order of render request and artifacts should be honored
- XR devices have stereoscopic displays: two video streams!
- Low level operations and optimizations required
  - Renderer and server written in C++
Streaming InstantXR

User’s head position

- Periodically sends the position every 20 ms
- Using web standards WebXR and WebSockets API

Novel streaming strategy to reduce latency

- Sending the user raw compressed packets
- Does not have the notion of “timestamps” (show frames as soon as they arrive)
- No audio/video synchronization
- Using web standards WebCodecs and WebSockets API
Harmonizing 3D assets with InstantXR scenes

Scene and estimated depth from NeRF

Torus rendered using OpenGL

Depth Testing

Harmonized Result
InstantXR Demo
Conclusion

Contribution

● Our method provides a mirror world of an existing space as an immersive XR environment without the cumbersome modeling process
● One of the first attempts to apply NeRF to real world use cases

Opinion

● NeRF has a potential to be a new data storage format for scenes
● Research in handling NeRF scenes should be done
● Research on delivery of volume rendered artifacts should be done
Future Work

- User experience comparison with other methods with:
  - 3D reconstruction model transfer
  - point cloud streaming
  - 3D camera streaming
- Lowering latency even more
- Improving user experience
  - Judder Control
  - Noise reduction of head position
- Method like asynchronous time warp, for volume rendering

3D Rendering pipeline with asynchronous time warp

---

Thank you!

Visit https://moonsikpark.github.io/instantxr/ for demo video and code release