Edge-based Provisioning of Holographic Content for Contextual and Personalized Augmented Reality

Michael Glushakov, Yunfan Zhang, Yuqi Han, Timothy James Scargill, Guohao Lan, Maria Gorlatova

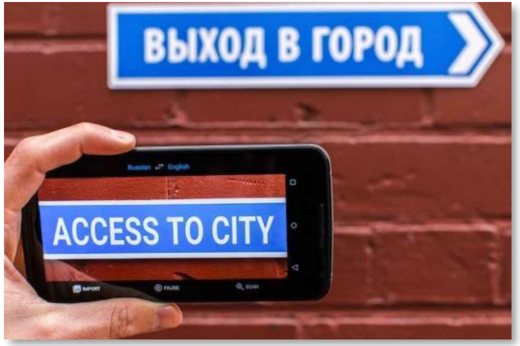
March 23rd, 2020

Talk Outline

- **Edge-supported augmented reality**
- Related work
- Google ARCore hologram provisioning case study
- Magic Leap One case study
- Challenges and directions

Augmented Reality (AR): A Definition

- The [virtual] content is laid out around a user **in the same spatial coordinates as the physical objects surrounding her/him***



*From: Baldassi et al, Challenges and New Directions in Augmented Reality, Computer Security, and Neuroscience, June 2018.

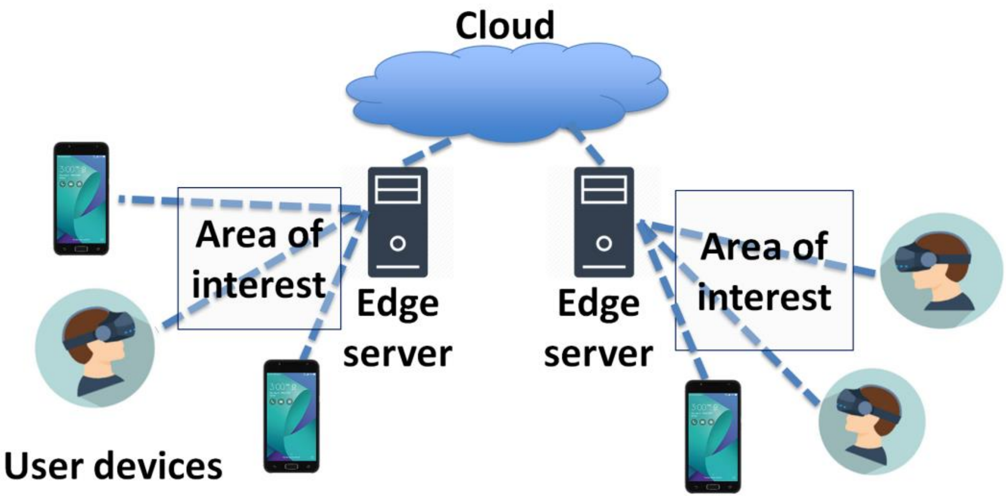
Core Mobile Technology of the Future

"AR will redefine our relationship with technology"

"It is the next big thing, and will pervade our entire lives"

Edge Computing as an Enabler of Next-Generation AR

- Low-latency connections to computationally capable nodes
- Access to stationary persistent local sensors



Example Application

- Ongoing project at Duke Lemur Center to enhance visitor experiences using mobile AR
- Personalized, contextual holograms cached on and delivered by intelligent edge nodes

Personalization

- Edge servers process data on environmental conditions and user profile to provide most suitable content
- Examples include an animated virtual guide for children, or in-depth scientific information for researchers

Talk Outline

- Edge-supported augmented reality
- **Related work**
- Google ARCore hologram provisioning case study
- Magic Leap One case study
- Challenges and directions

Background and Related Work

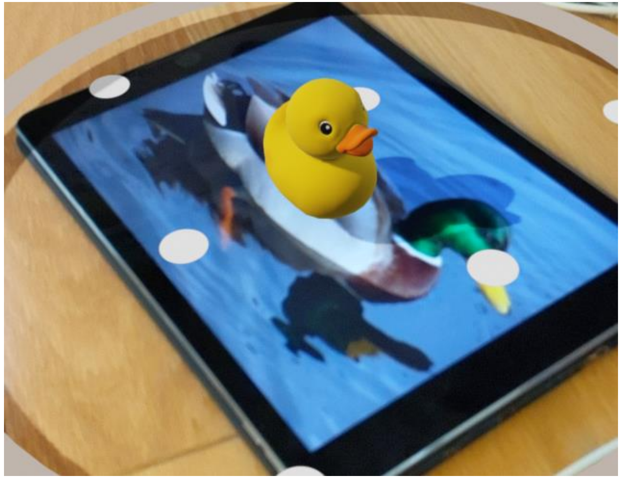
- Context-aware AR: multiple applications
  - Do not consider how to obtain holographic content
- On-demand loading of 3D content: supported by many gaming platforms
  - Not optimized, not examined in edge conditions
- Edge computing and edge caching
  - Edge for AR: object recognition, video rendering, SLAM
  - Edge for generic content caching
  - This work is *first to develop edge-based holographic loading frameworks for common AR platforms*

Talk Outline

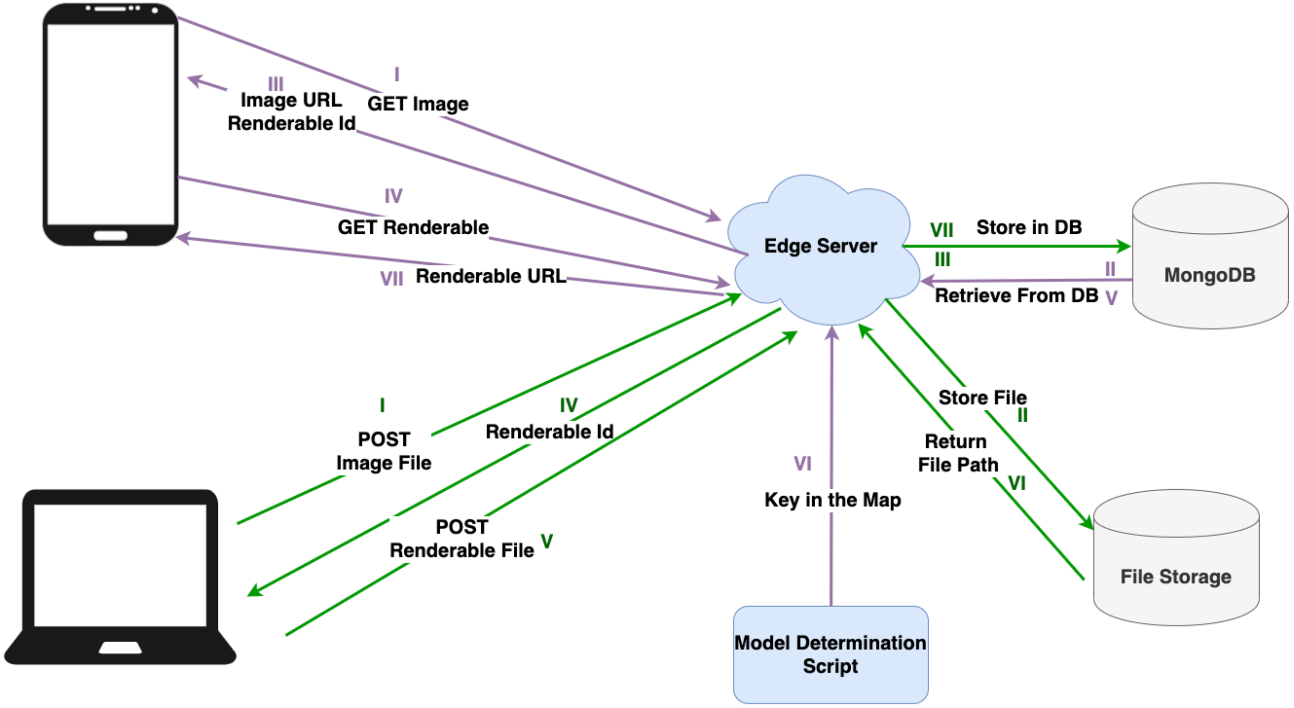
- Edge-supported augmented reality
- Related work
- **Google ARCore hologram provisioning case study**
- Magic Leap One case study
- Challenges and directions

Google ARCore Case Study: Motivations

- Target users: museums, zoos etc.
- Problem: high development cost and low budget
- Solution: code-free AR experiences
- Server: NodeJS with MongoDB
- Client: ARCore enabled Android app
- Can be extended to ARKit

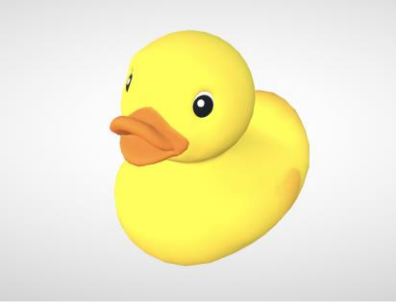


API Overview

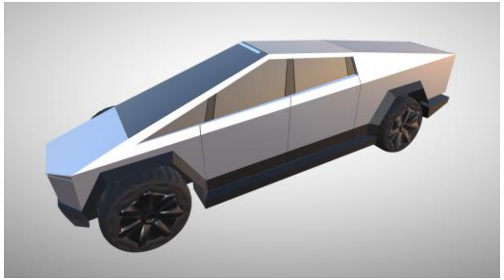


Performance Metrics

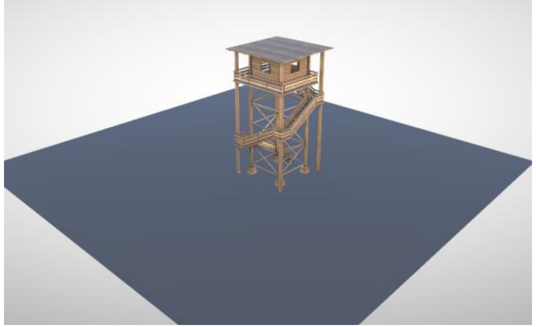
- 3 Models
- Edge Server vs. Cloud Server
- SFB vs GLB/glTF



Duck
GLB: 120 KB GLB: 238KB



Cybertruck
GLB: 321KB SFB:646 KB



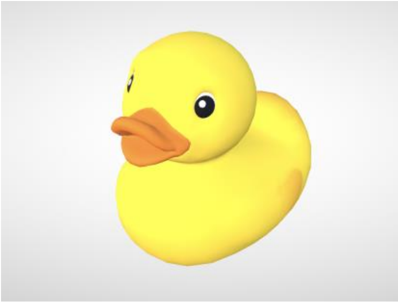
Tower
GLB: 19.1 MB SFB: 19.4 MB

File Transfer Latency

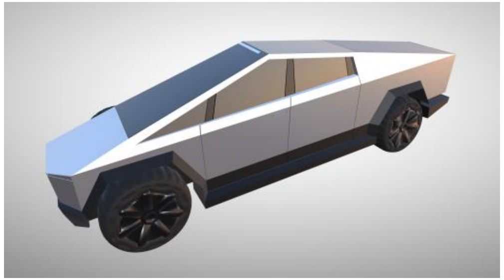
3D Model	Phone	Time Saved (GLB vs. SFB) Edge (ms)	Time Saved (GLB vs. SFB) Cloud (ms)	Time Saved (Edge vs. Cloud) GLB (ms)	Time Saved (Edge vs. Cloud) SFB (ms)
Duck	Nokia 7.1	37.9	38.6	42.6	43.3
	Pixel 3	14.6	42.8	87.4	115.6
Cybertruck	Nokia 7.1	37.2	106.1	34.9	103.8
	Pixel 3	73.9	21.7	80.4	28.2
Tower	Nokia 7.1	-151.5	-402	556.4	305.9
	Pixel 3	326.1	267	-270.4	-329.5

File Transfer Latency Takeaways

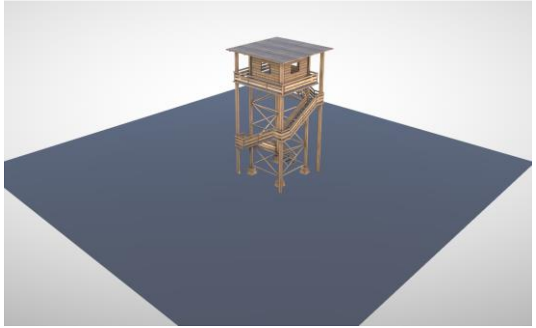
- Generally faster load times from Edge Server
- Generally faster for GLB than SFB
- Doesn’t hold for large files



Duck
GLB: 120 KB GLB: 238KB



Cybertruck
GLB: 321KB SFB:646 KB



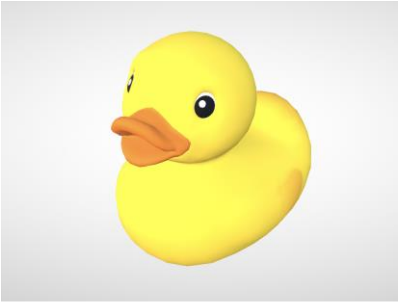
Tower
GLB: 19.1 MB SFB: 19.4 MB

File Storage Load & Processing Times

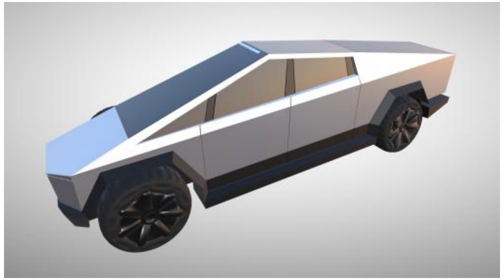
3D Model	Phone	Load Time GLB (ms)	Load Time SFB (ms)	Time Lost on Processing (ms)
Duck	Nokia 7.1	171.0	40.8	130.2
	Pixel 3	94.2	39.1	55.1
Cybertruck	Nokia 7.1	370.1	66.0	304.1
	Pixel 3	203.7	55.6	148.1
Tower	Nokia 7.1	2425.3	605.6	1819.7
	Pixel 3	1079.9	344.4	735.5

Performance Metrics Takeaways

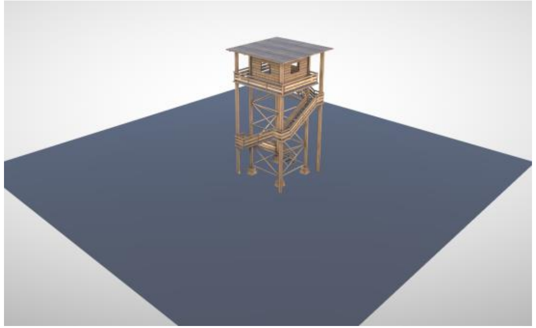
- Much longer processing times for GLB files
- Total Load + Processing times is shorter for SFB that GLB



Duck
GLB: 120 KB GLB: 238KB



Cybertruck
GLB: 321KB SFB:646 KB



Tower
GLB: 19.1 MB SFB: 19.4 MB

Talk Outline

- Edge-supported augmented reality
- Related work
- Google ARCore hologram provisioning case study
- **Magic Leap One case study**
- Challenges and directions

Magic Leap One Case Study: Introduction

- AR Headsets: new but quickly immerging AR consumer device
- Limitations: battery life, computational power, memory, size constraints
- Magic Leap 1: one of the most adopted AR headsets in the market today

Magic Leap One Case Study: Motivation

- 3D Models on modern interactive apps usually ranges from a few to a few hundred MBs in size
- Challenges with AR application:
  - Model usage dependent on user’s environment
  - Large number of models needed
  - Difficult to download models in advance
- Solution: runtime model offloading

Magic Leap One Case Study: Setup

- The models are loaded onto ML1 at runtime
- We used 3 models to emulate low, medium, and high model quality
- The models have 5K, 33K, and 87K triangles. Texture size is 2K by 2K pixels. Compressed with LZMA before transmission (more on next slide)
- Edge server is 10ms RTT away with 50Mbps bandwidth. ML1 is connected to internet with 5Ghz Wi-Fi

File Transfer Latency

3D Model	Triangle Count	Texture Size (In Pixels)	File Size Before Compression (in MB)	File Size After Compression (in MB)	Transfer Time (in Milliseconds)	Decompression Time (in Milliseconds)
Bunny	5K	2K by 2K	7.4	1.4	1,702	267
Lucy	33K	2K by 2K	10.6	1.7	1,715	372
Dragon	87K	2K by 2K	79	7.7	3,325	175

User Experience on ML1

- The hologram runtime loading on ML1 introduced a certain amount of delay
- The delay is acceptable as users would naturally try to familiarize themselves with the environment first before start to look for holograms, so the delayed appearance of holograms are not particularly noticeable

Future Research Directions

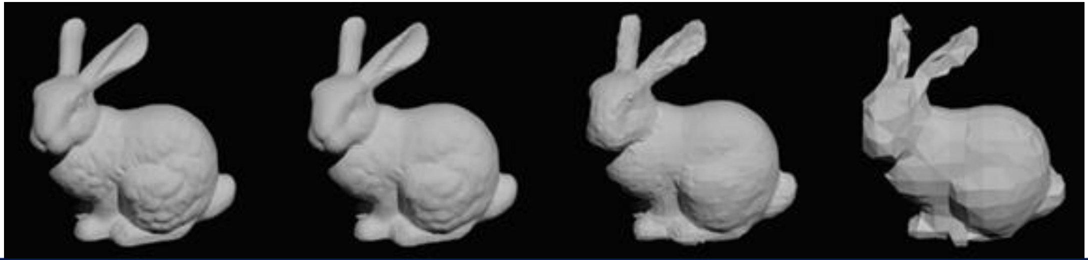
- Preload holograms to further conceal delays
- Can be implemented by predicting user interaction or recognize the location that the user is in
- Lack of dataset on user behaviors in AR applications
- Current commercial AR devices do not have sufficient APIs to implement such features
- We did a brief experiment by sniffing the ML1's mac address and preload holograms once user enters a room. Indeed improved perceived user experience.

Talk Outline

- Edge-supported Augmented Reality: an introduction
- Background
- Google ARCore hologram provisioning case study
- Magic Leap One case study
- **Challenges and directions**

Challenges and Directions: On-Demand Hologram Transmissions

- Challenge: user-facing latency
  - ➢ Many users
  - ➢ Poor wireless channel conditions
  - ➢ High edge server loads
- Solution: edge server supplies holograms of different quality levels depending on conditions



Challenges and Directions: Proactive Hologram Transmissions

- Advantages:
  - No user-facing latency
  - Can optimize bandwidth utilization over time
- Research directions: identifying the right holograms to transmit and cache
  - Hologram caching and eviction policy design
  - Leveraging user locations in hologram transmissions

Summary

- Propose to personalize AR experiences by using edge servers to transmit appropriate holograms to the users
- Conduct case studies with two popular AR platforms

ARCore

magic leap

- Identify challenges and research directions
- Make all code available on GitHub

Acknowledgements

- Lord Foundation of North Carolina
- NSF awards CSR-1903136, CNS-1908051

- 3D model repositories

The Stanford 3D Scanning Repository

cgtrader

KHRONOS GROUP

Questions?

- mykhaylo.glushakov@duke.edu