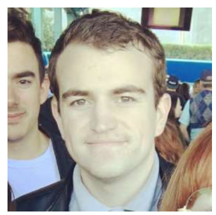
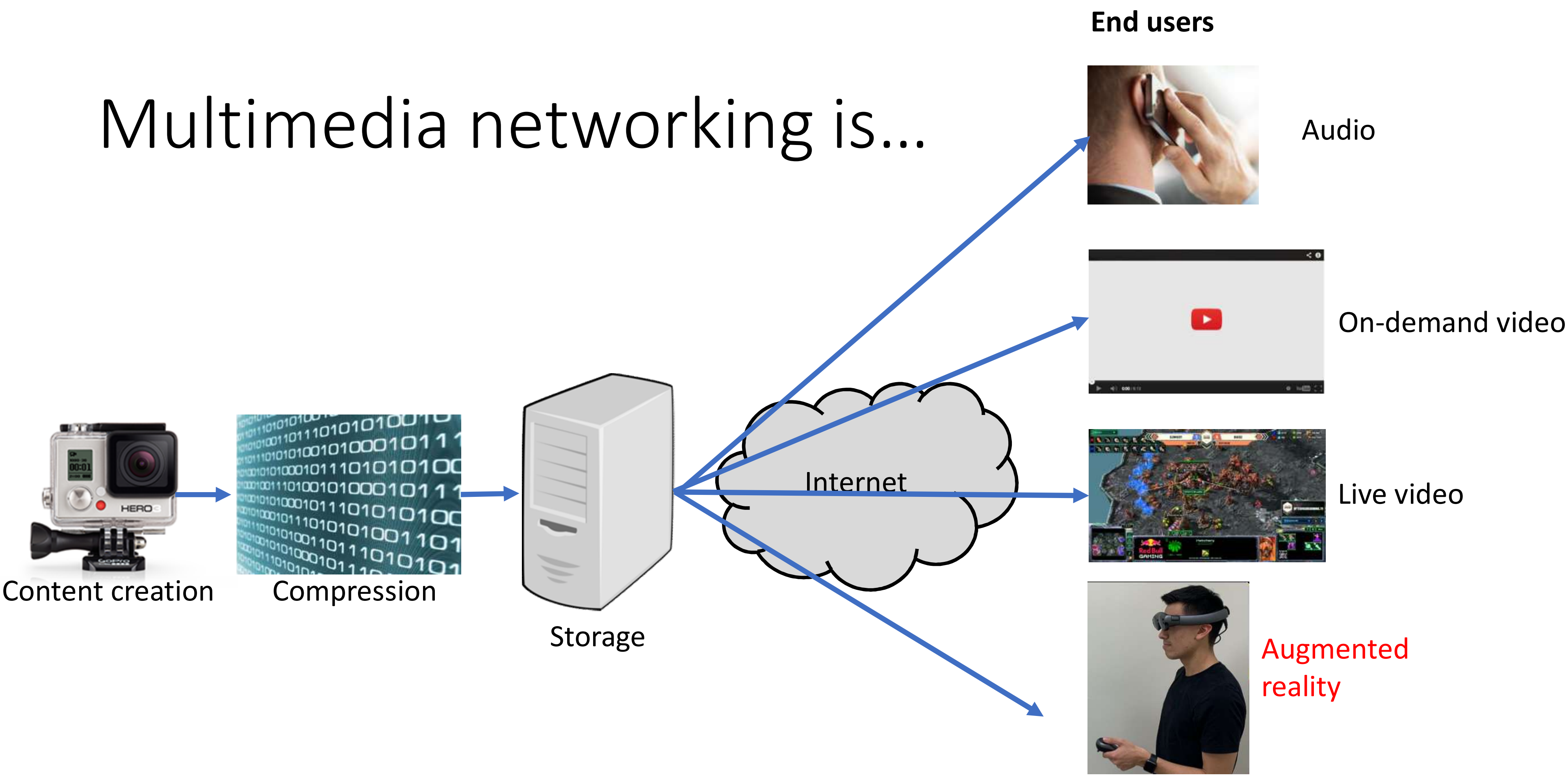
ShareAR: Communication-Efficient Multi-User Mobile AR


Xukan Ran*, Carter Slocum*, Maria Gorlatova[1], **Jiasi Chen***

*University of California, Riverside [1]Duke University

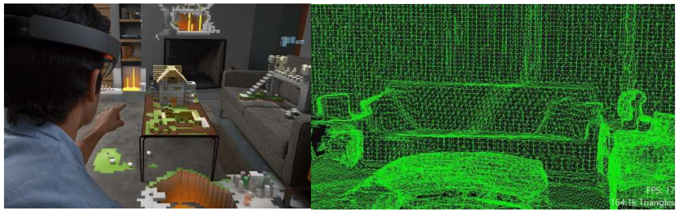
Multimedia networking is…



Who is Using Augmented Reality?

Head-mounted hardware:

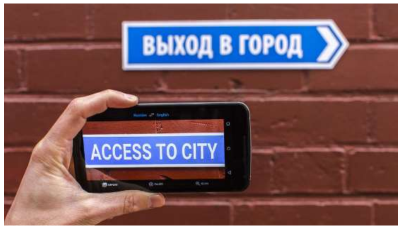
Magic Leap



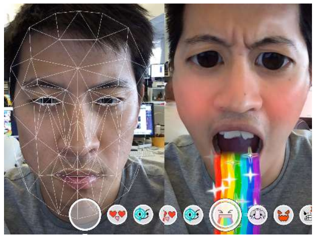
Microsoft Hololens

Smartphone hardware:

Pokemon Go



Google Translate (text processing)



Snapchat filters (face detection)

Mobile AR software platforms:

ARCore

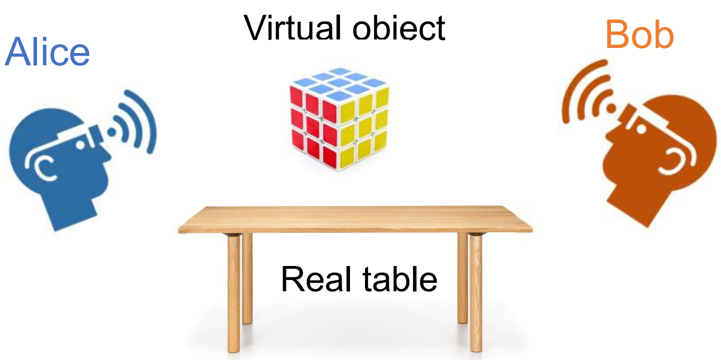
Google Play Services for AR

Apple ARKit

Apple ARKit

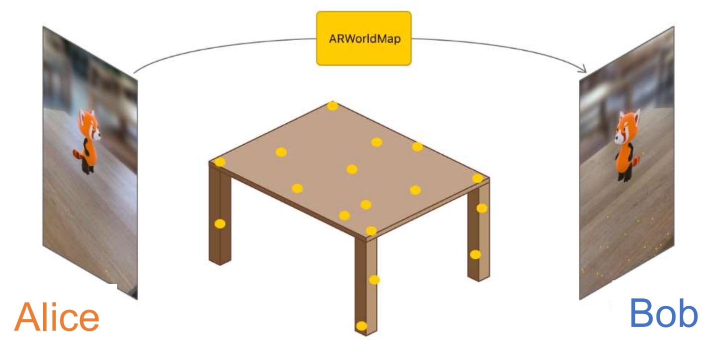
Our scenario: Multi-user mobile AR

- Multiple users in the same physical area
- Users view the same virtual content
    1. Alice places the virtual object
    2. Bob views the virtual object

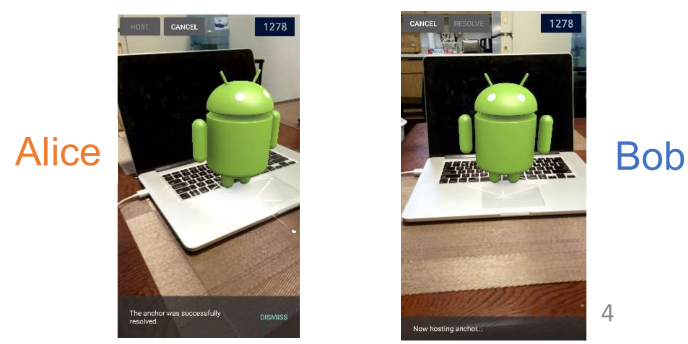


Examples from industry:

Apple's AR Multiuser



Google's CloudAnchor

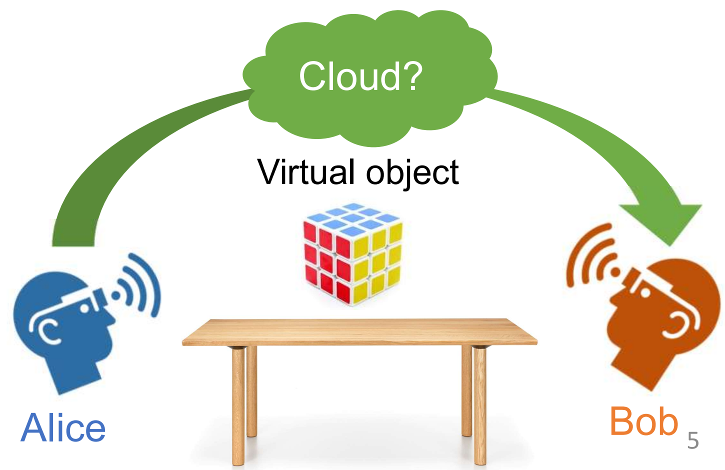


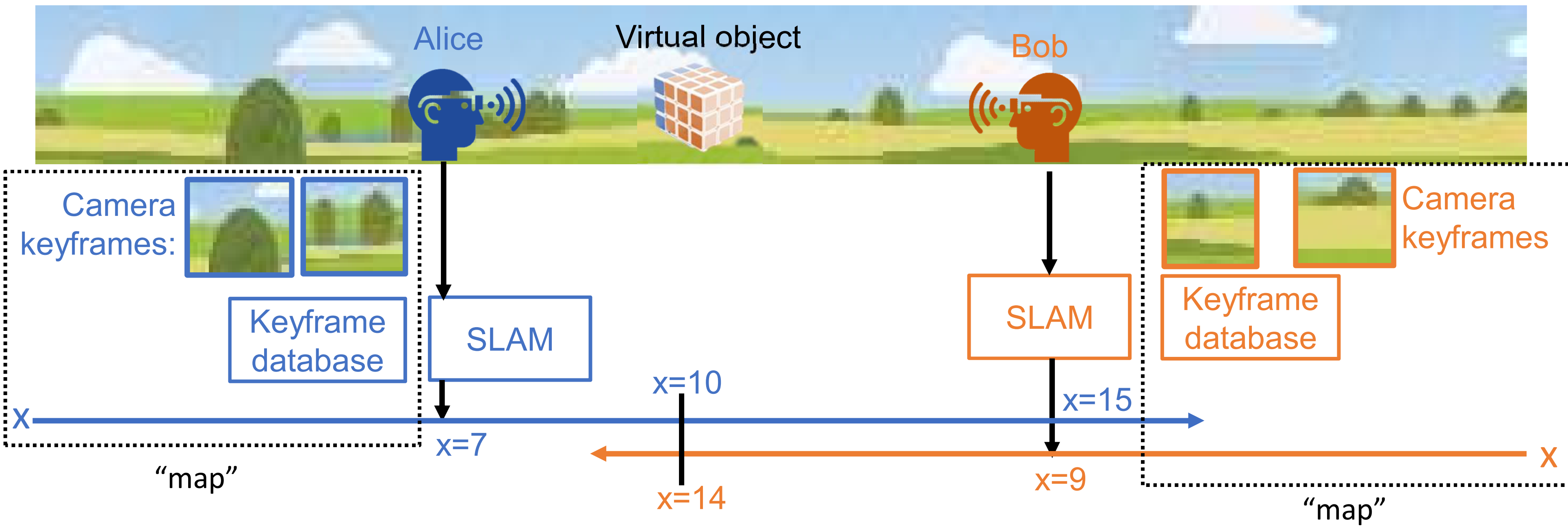
Motivation: Lack of understanding of multi-user AR

- Researchers have focused on object detection for single-user AR
- In multi-user AR, information is exchanged over the network

What kind of network traffic does AR generate?
What kind of network support does AR need for good user experience?

- Research agenda
    1. **Who** to send to?
    2. **What** information to send?
    3. How to **evaluate** multi-user AR quality?
    4. How to **test** any proposed solutions?





- Devices have different coordinate systems (computed by SLAM)
- Camera keyframes record history of where the user has been
- Keyframe database enables **map alignment** (coordinate system matching)

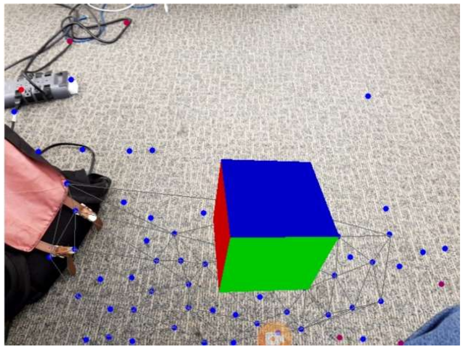
e.g., Keyframe database +  = "user B is at x=15" → aligned x = 15 with x = 9

- **Map alignment** is needed for both users to render the same virtual object

  → Bob knows x = 10 means x = 14, renders his virtual cube

4. How to test any proposed solutions?

- Challenge: current multi-user AR platforms are closed-source
  - Google ARCore's CloudAnchor
  - Apple ARKit's ARWorldMap
  - Microsoft Hololens' Spatial Anchor



- Proposed platform: ShareAR research prototype
  - Extended existing open-source Android AR platform [1] with multi-user abilities
  - Enables full control over networking and computer vision processing

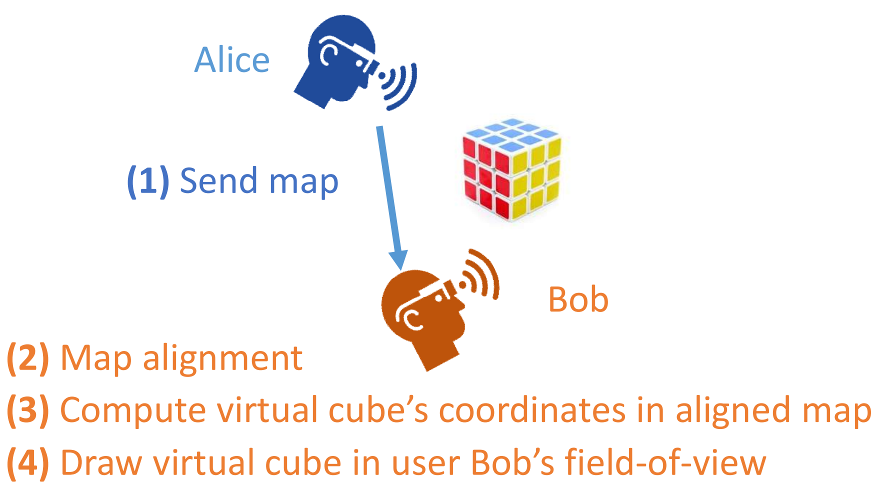
[1] Qin, T., Li, P., and Shen, S. Vins-mono: A robust and versatile monocular visual-inertial state estimator. IEEE Transactions on Robotics 34, 4 (Aug 2018).

1. Who to send to?
   (compute latency)

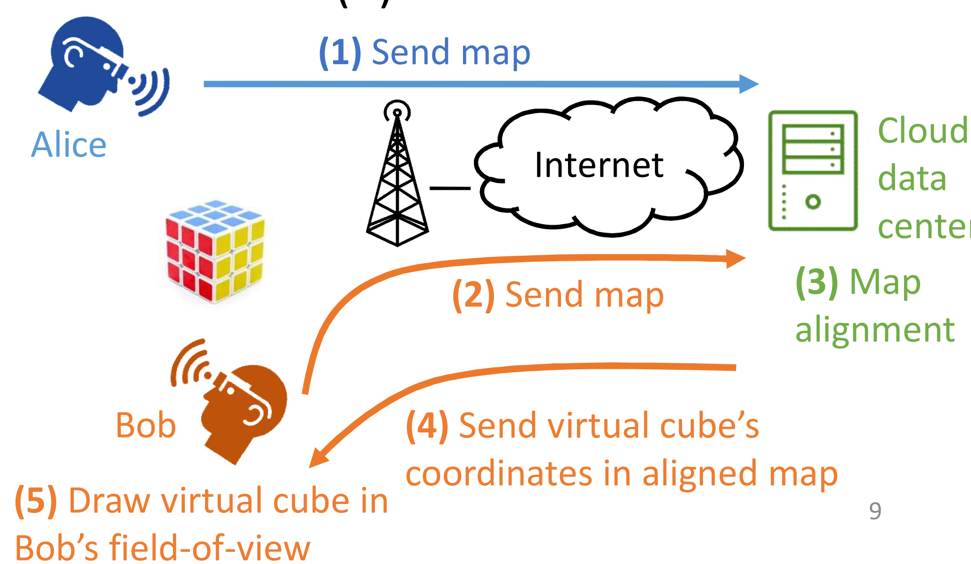
P2P vs. cloud architectures

- Examined communication patterns of existing platforms
  - Google ARCore, Apple ARKit, Microsoft Hololens
- Inferred two main architectures (details are closed-source)

(a) Peer-to-peer



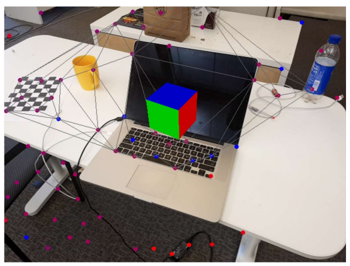
(b) Cloud-based



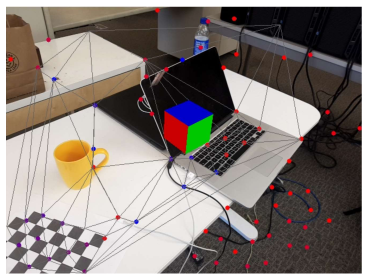
How much latency do users perceive?

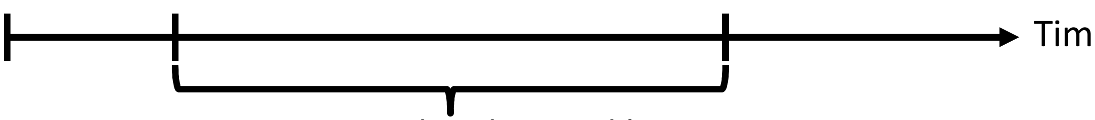
- Measured latency breakdown of existing Android and iOS AR apps
  - Two Samsung Galaxy S7 devices with 50 Mbps up/down WiFi

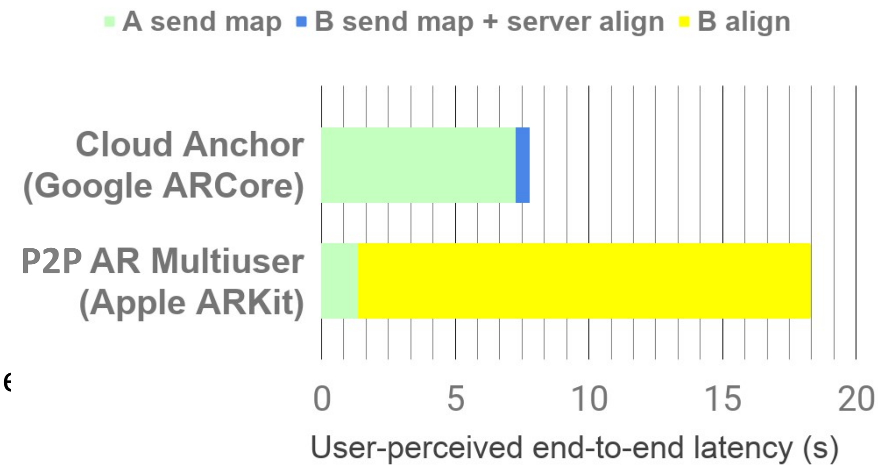
1. Alice places virtual object



2. Bob sees virtual object



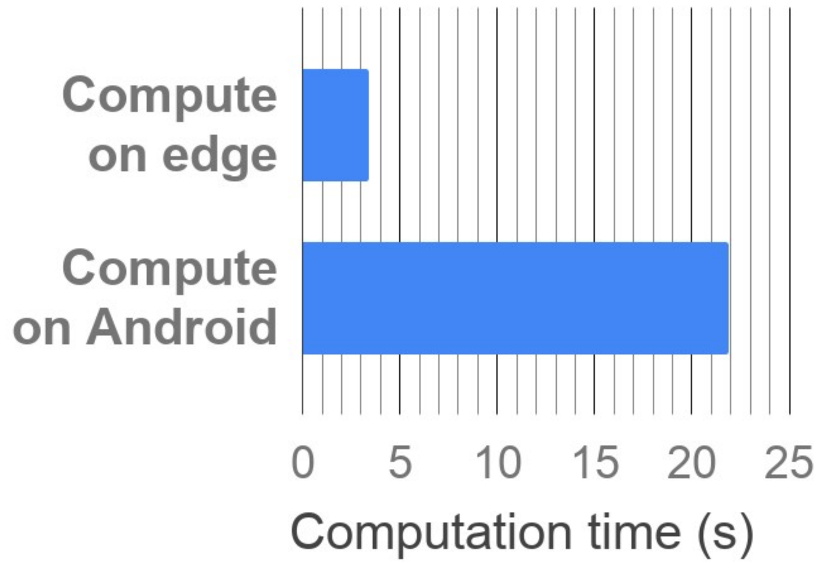




User-perceived end-to-end latency is 7-17 seconds! Would like 100s of ms
P2P app experiences longer latency

How long does AR computation take?

- Measured map alignment computation latency on ShareAR

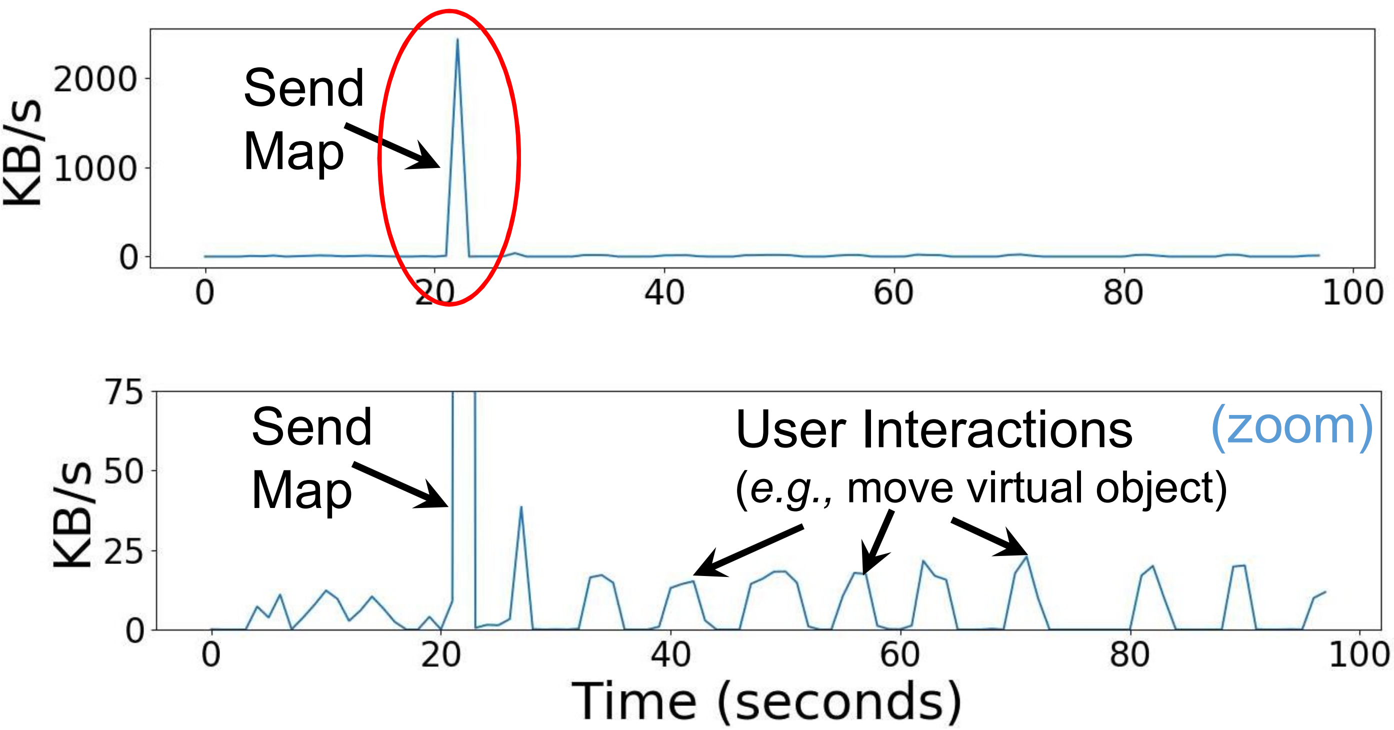


Edge-based computation can reduce computation latency
(at the expense of communication latency?)

2. What to send?
(communication latency)

What does AR traffic look like?

- Recorded bandwidth traces from the previous experiment



→Large initial bursts (>20Mb) corresponding to map data

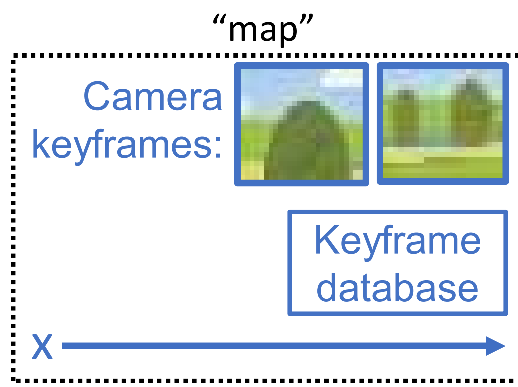
→Unpredictable smaller user interactions

How to design a QoS class for AR traffic?

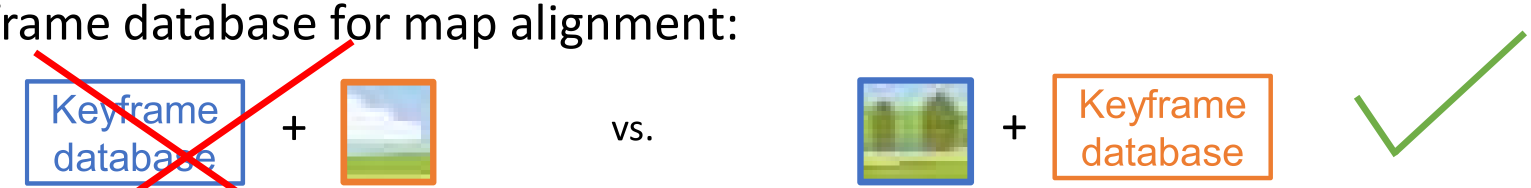
- **High bandwidth for initial burst**
- **Low latency for later bursts**

Can we reduce the communication latency?

- What data should we transmit in the map?
  - Camera keyframes → no, only need features (~150 KB/frame)
  - Camera features → yes, for map alignment (20-50 KB/frame)
  - Keyframe database → yes, for map alignment (50 MB+)



- How to adapt the map data to the network bandwidth? (like MPEG-DASH)
  - Camera features: Cull down instances in **time**? **Space**?
  - Keyframe database for map alignment:



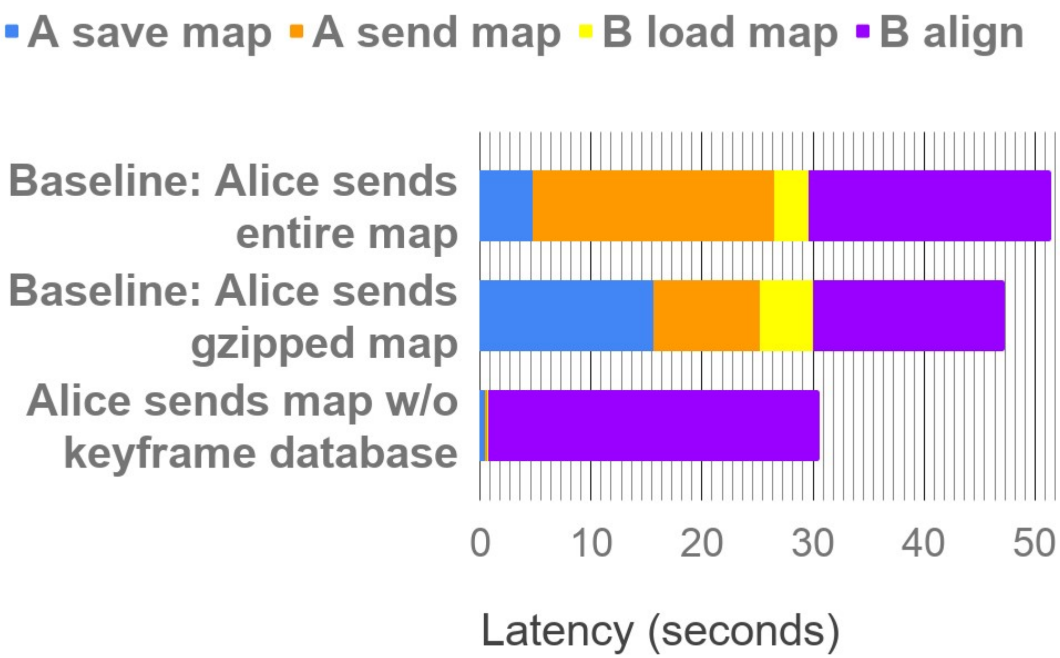
→ Transmit Alice's keyframe database to Bob
→ 50 MB data transmission

→ Transmit Alice's camera features to Bob
→ 50 KB data transmission

How much does suppressing the keyframe database help?

+ Keyframe database

→ Transmit Alice's camera features to Bob

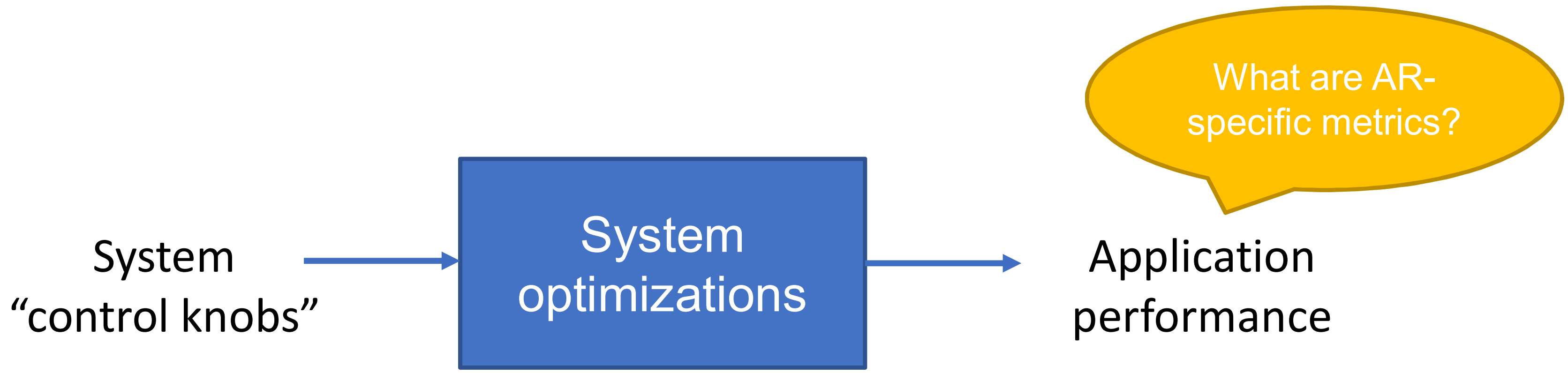


Not sending the keyframe database drastically reduces latency (map alignment still successful)

*Bandwidth simulated at 5 Mbps for these experiments

3. How to evaluate multi-user AR?

Typical systems optimization approach



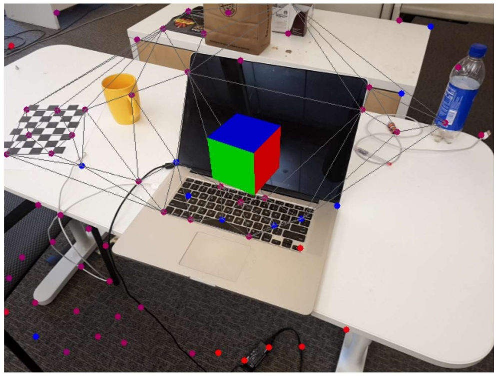
- *e.g.*, video
  - Stalls
  - Bit rate
  - Bit rate switches
- AR
  - Latency
  - ???

Beyond latency: Spatial consistency

- We often observed devices with different views of the same virtual object

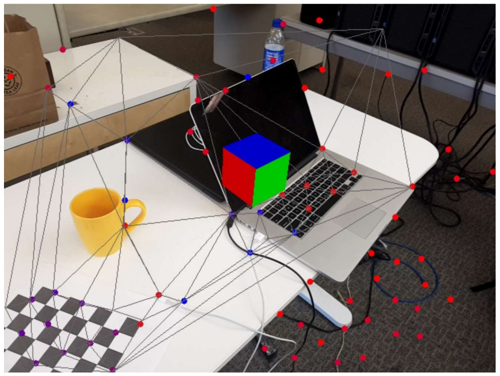
Alice's view

Cube is centered above laptop



Bob's view

Cube is above left of laptop

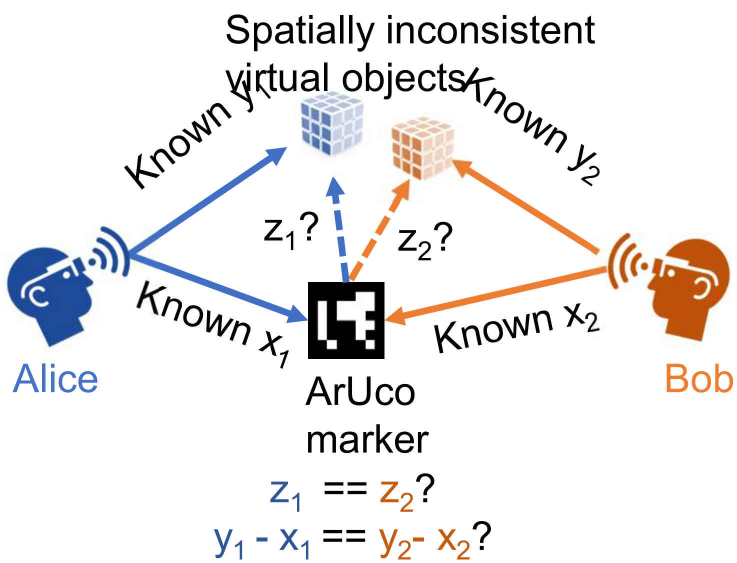


- How to quantify AR quality?
  - Human-computer interactions community: conduct user surveys → too slow
  - Multimedia community: manually label ground truth virtual object → too slow

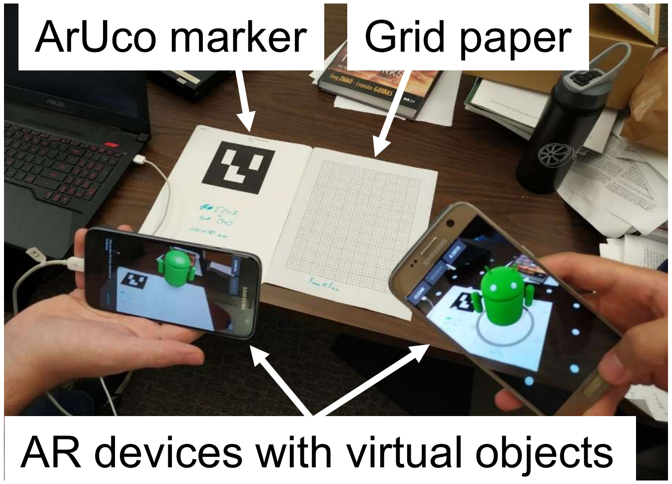
→ We need a real-time, quantifiable measurement tool

AR quality tool to measure spatial consistency

- Main idea: place ArUco markers to act as reference points
- Evaluation: Use grid paper to measure "ground truth" virtual object position

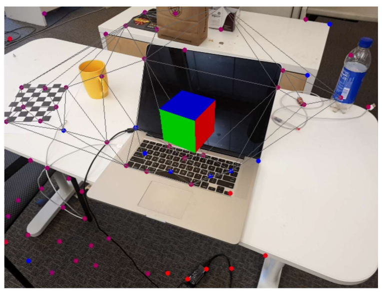


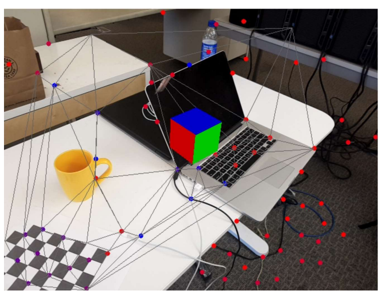
Our setup in the lab:



- Initial results shows accuracy on the order of centimeters

Key Take-Aways





Users suffer from **10s of seconds of end-to-end latency** in multi-user AR

Edge computing and **adaptive data transmissions** can reduce end-to-end latency

Beyond latency, we need a way to measure AR virtual object **spatial consistency**

Thank you!
Questions?