EECE5698
Networked XR Systems

Lecture Outline for Today

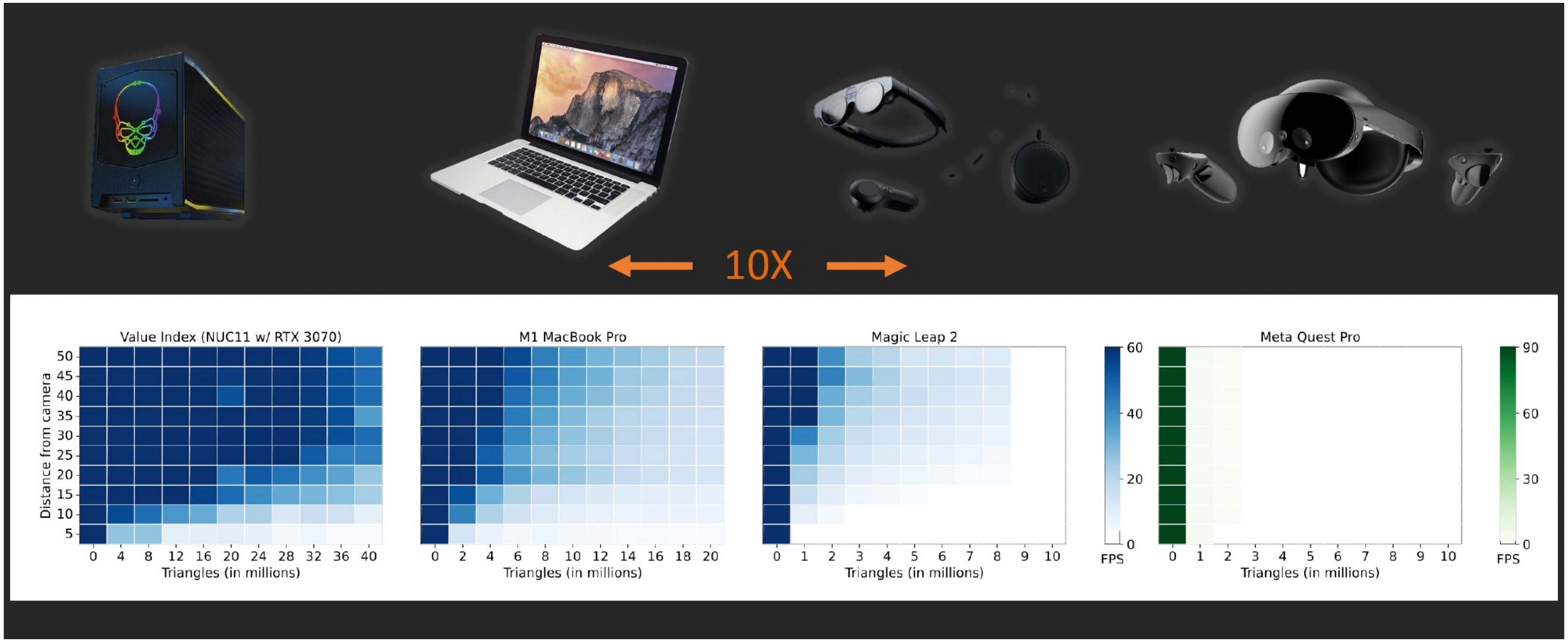
- Offloading Rendering Computation
- Remote, Cloud, Distributed, Edge, Hybrid Rendering
- Streaming Rendered Video
  - WiFi
  - mmWaves, THz, and Optical links

Rendering Performance

- *Frames per second*
  - *Speed*
- *Polygons per frame*
  - *Related to detail*
- *Latency*
  - *How long before system input to updated frame*
- *Power*
  - *Computation and data transfer*

Rendering Performance

Across different XR devices



Rendering Performance

- How about real-time rendering on ultra-thin wearable XR devices like glasses?

Rendering Performance

- Rendering computation is expensive
  - Offload rendering computation elsewhere for high-quality

- Remote rendering
- Cloud rendering
- Edge rendering
- Distributed rendering

Local vs. Remote Rendering

- **Local Rendering**: The traditional approach where rendering is done on the same device that is being used for display and interaction.
- **Remote Rendering**: Offloading the rendering process to a remote server or dedicated hardware and streaming the output back to the local device.
- Advantages and Disadvantages:
  - Local rendering leverages direct access to the GPU, minimizing latency but can be limited by the device's hardware capabilities.
  - Remote rendering allows for more powerful processing and potentially better graphics quality but can introduce network latency and require stable connectivity.

Cloud Rendering

- Using cloud computing resources to perform rendering tasks, with the rendered content streamed back to the user's device.
  - Scalability, access to high-performance hardware, and the ability to offload intensive computational tasks from local devices.
- Considerations: Requires reliable and fast internet connection, and there can be concerns about data security and latency.

Cloud Rendering

- Two-way latency
  - Need to wait until the user’s pose is sent to the Cloud, render the content, and receive the rendered video

Edge Server Rendering

- Edge rendering is done at the edge of the network, near the user, rather than on centralized data centers or the user's device.
  - The purpose is to reduce latency, decrease the bandwidth needed for high-quality graphics, and alleviate the computational load on user devices.
- Key Benefits:
  - Faster content delivery due to proximity to the user.
  - Improved performance for real-time applications.

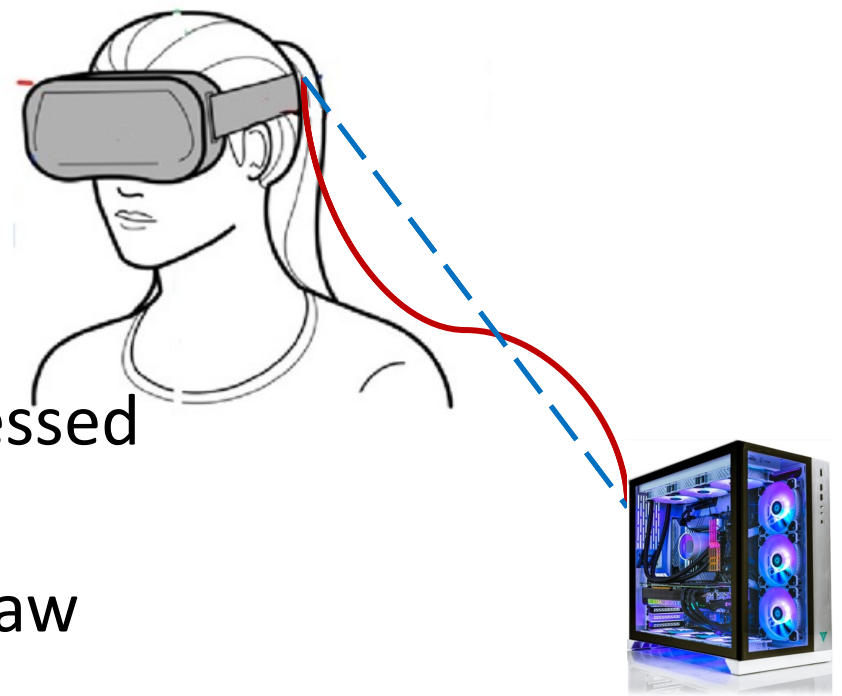
Edge Server Rendering

- Cellular Networks
  - Rendering is placed at the Base station

- Need to stream rendered video from Base station
  - Base stations are placed at a few miles away
  - High frequencies provide high bandwidth but LOS problem
  - Lower frequencies are okay but low bandwidth
  - Latency is also a problem

Edge Server Rendering

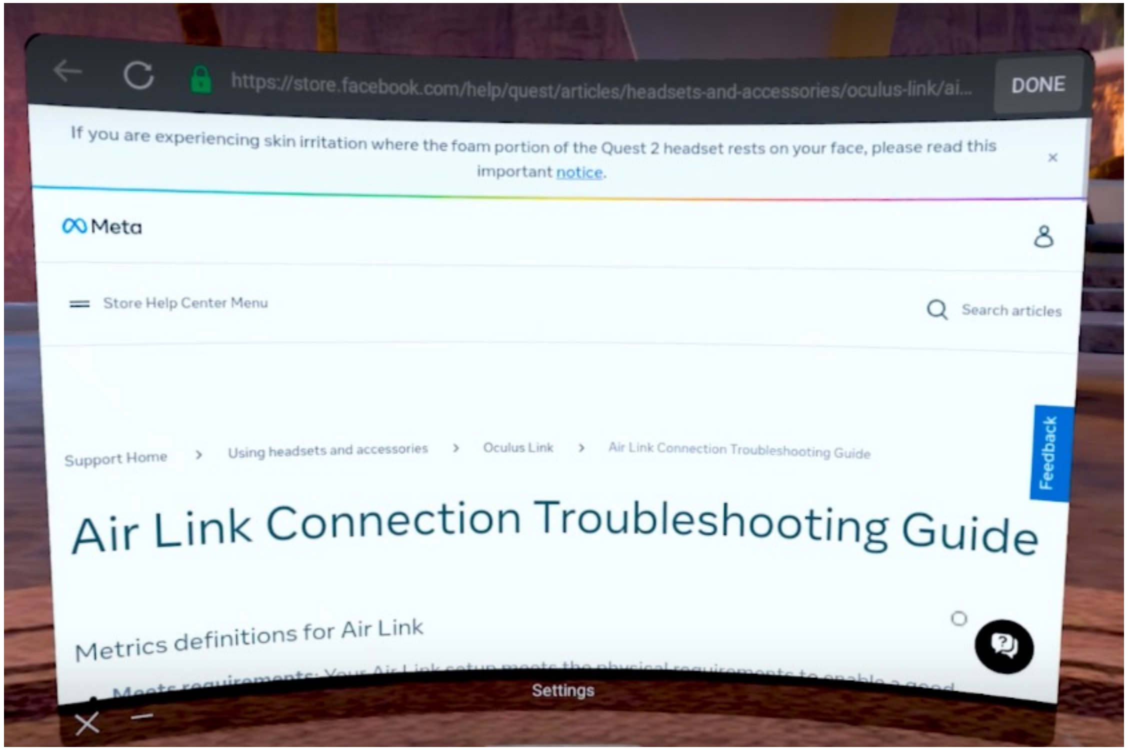
- WiFi
  - Rendering is placed a computer within the same WIFi LAN
  - Closer to users
  - Low latency



- Works for streaming compressed rendered content
- What if we want to stream raw video?

Edge Server Rendering

- WiFi
  - Connect Meta Quest to your PC over Wi-Fi with Air Link



Edge Server Rendering

- Why do we want to stream raw video to XR devices?
  - Eliminate the computation demands of compression and decompression
  - Also saves latency

- mmWaves, THz or Optical links for higher bandwidths

Edge Server Rendering

- Problem with higher frequency wireless links
  - Links are not reliable – narrow wavelength
  - Environmental impact
  - Line of sight

- Problem with XR devices
  - Users move around
  - Mobility impact

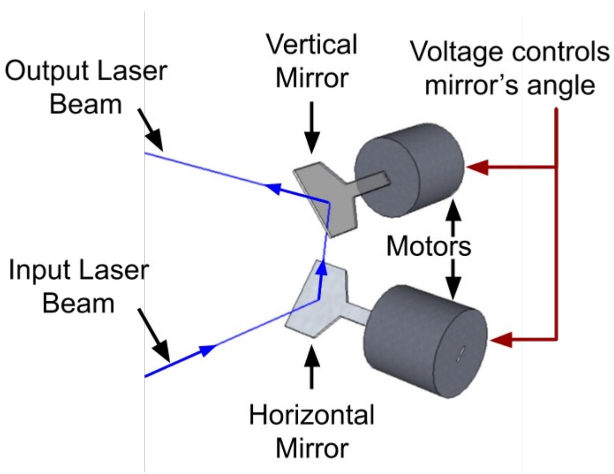
Edge Server Rendering

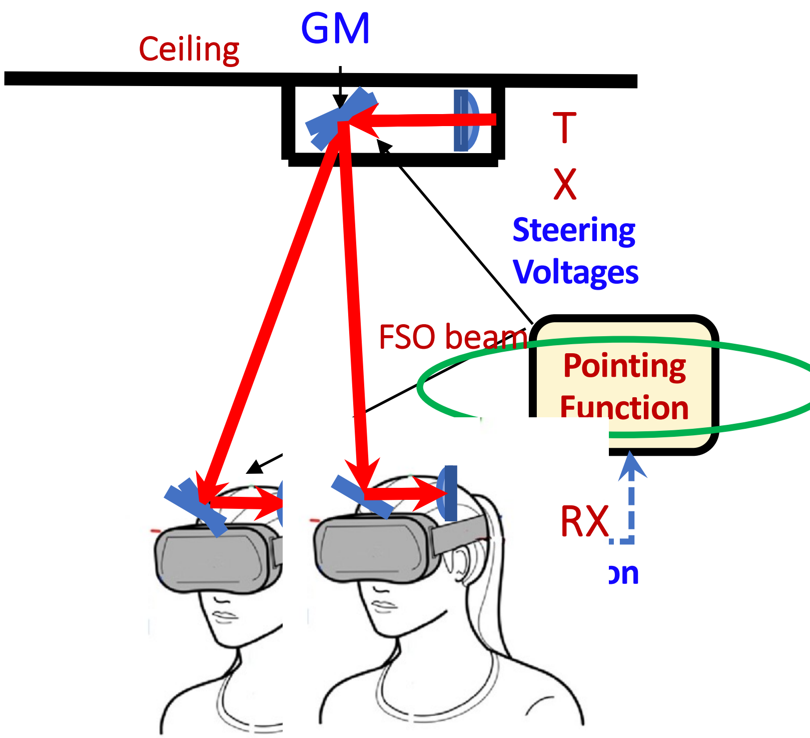
- Let’s take an example scenario with Free space optics (FSO)
  - Narrow laser links, collimated beams

FSO-based VR Wireless Link

- TX (renderer) fixed on ceiling.
- RX (VRH) moves
- To realign the beam:
  a. Localize RX [mm accuracy; via VRH's in-built localization]
  b. Steer TX and RX [using Galvo Mirrors (GMs)]

(2-axis) Galvo Mirror





Pointing Function:

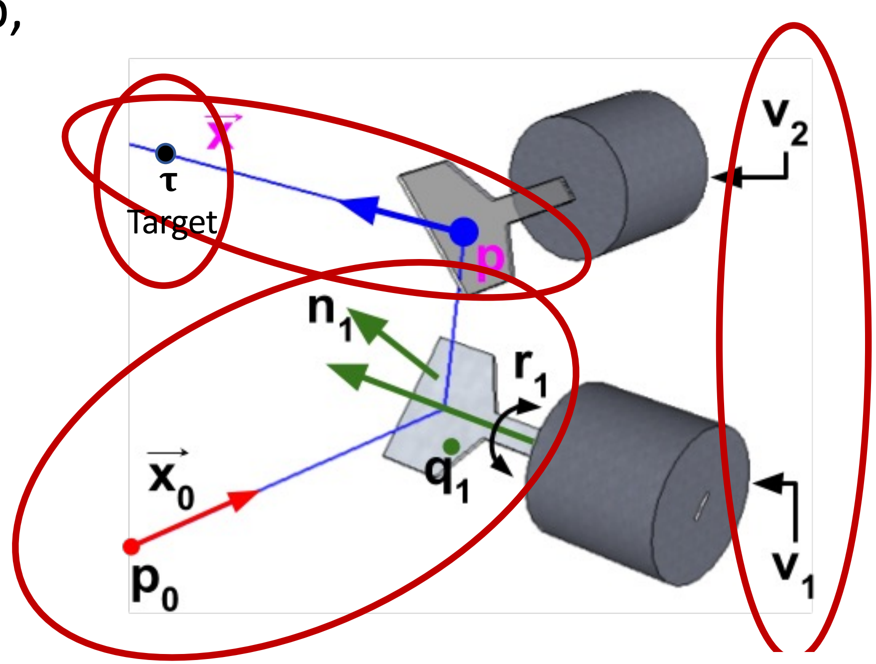
- Pointing function P:
  - Input: VRH/RX location [In the unknown VRH coordinate system]
  - Output: 4 GM Voltages [To steer TX and RX to realign beam]
- Learning P directly from (input, output) samples is infeasible
- Our approach:
  1. Learn GM models (two functions G and G') [Offline]
     a) In the GM's coordinate system (a known space).
     b) Map to the VRH coordinate system.
  2. Use GM functions to compute P. [Real-time]

1a. Learn GM Model (in GM space)

Function G: $(v_1, v_2) \longrightarrow$ Output beam $(p, \vec{x})$.

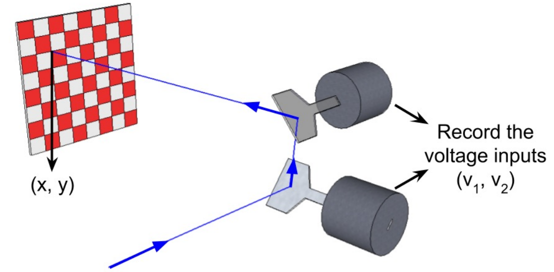
i. Derive an expression for G from its physical configuration.

ii. Learn the parameter values, using training data.



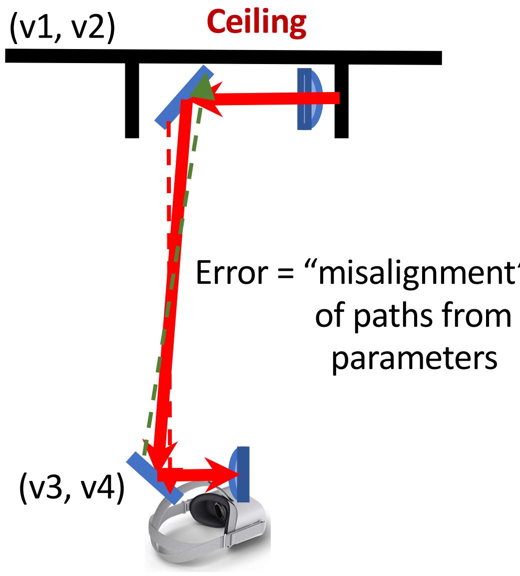
Function G': (target point τ) $\longrightarrow$ (v_1, v_2)

- Use G iteratively to estimate G'.



1b. Map GM Functions to VRH Space

- Tantamount to estimating GMs' positions in VRH space.
- Need to estimate 12 parameters (6 for each GM).

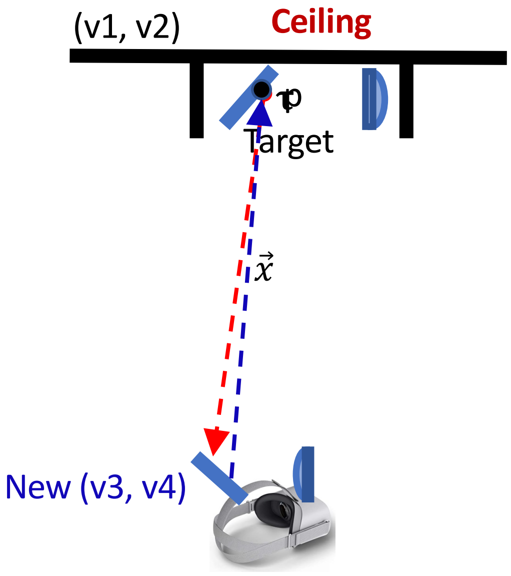


1. Gather training samples (aligned beam state).
   - (VRH Position, 4 voltages) for each sample.
2. Define an error function for given parameter values.
3. Determine parameter values that minimize the total loss over samples.

2. Pointing Function P from GM Functions

Pointing Function P:

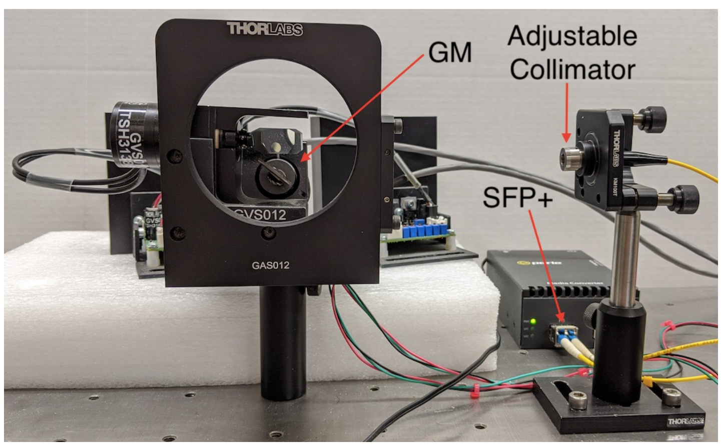
- Input: VRH position.
- Output: 4 Voltages.
- Approach (Real-Time):
  - Initialize voltages v_1, v_2, v_3, v_4
  - (p, $\vec{x}$) = G (v_1, v_2) TX-beam output specs
  - New (v_3, v_4) = G'($\boldsymbol{\tau}$ = p) RX-beam should hit p.
  - Similarly, compute new (v_1, v_2).
  - Iterate.



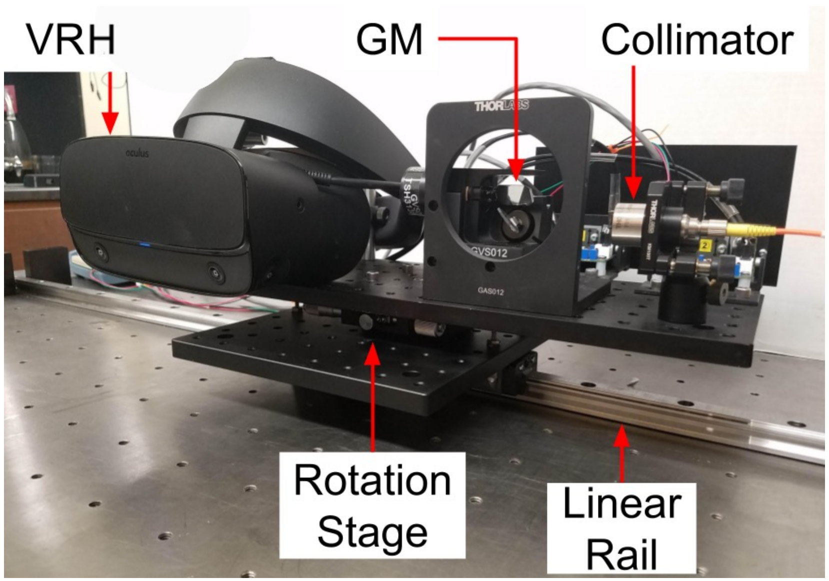
FSO-VR Prototype Design

- Link Design
  - Divergent beam offered higher movement tolerance.
  - 10 and 25 Gbps links.
- Prototype:

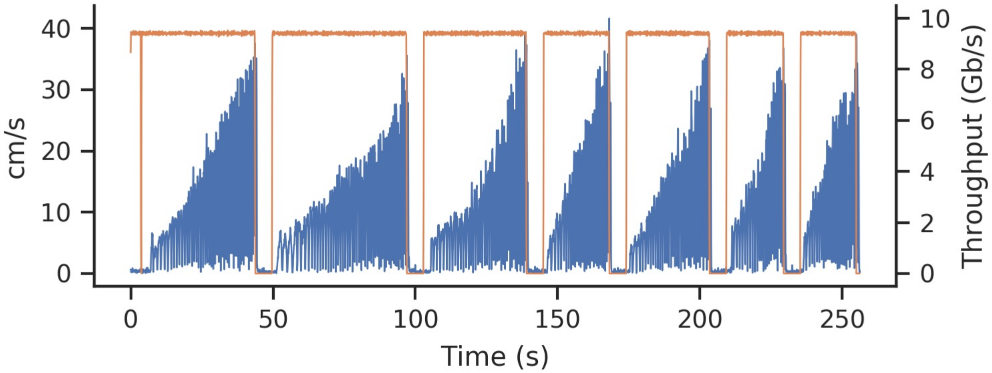
TX

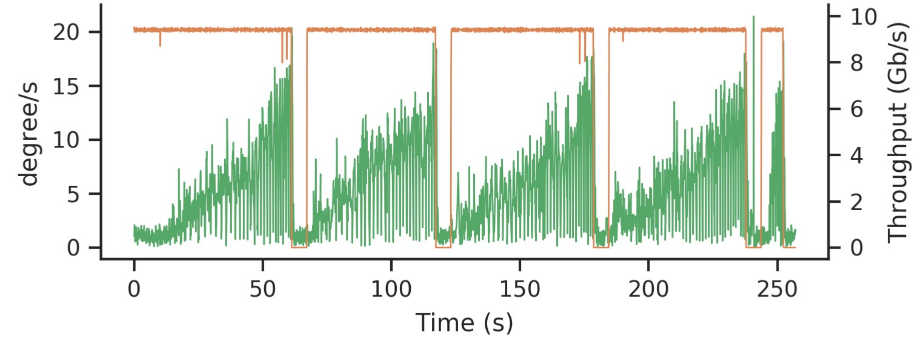


RX



FSO-based VR Link Performance

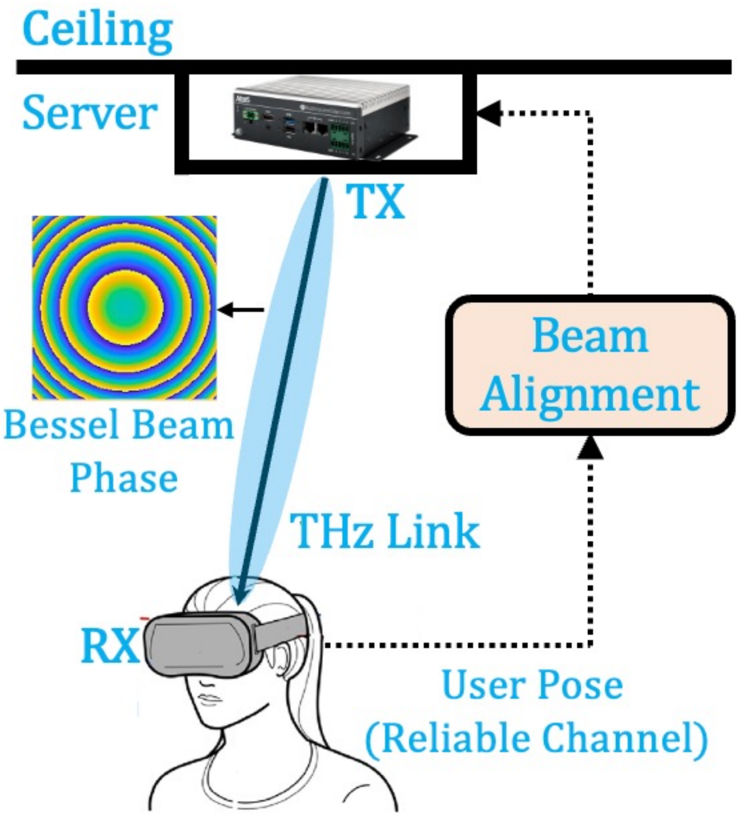




- Performance could be much improved, with customized components.
  - E.g., higher tracking frequency, customized optical components.

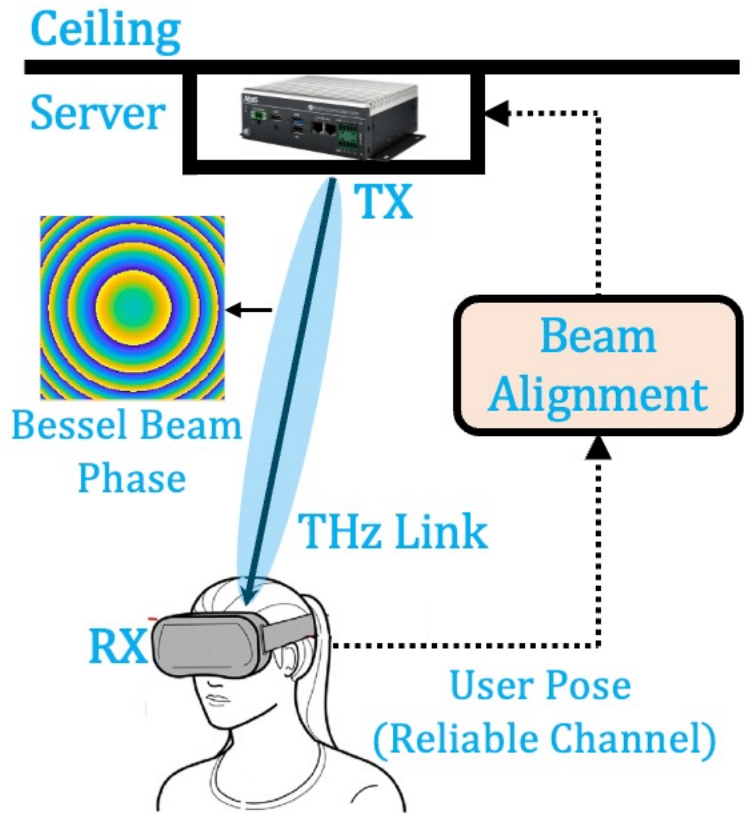
THz Band based VR Link

- Above 100GHz Radio frequencies
  - Affected smaller obstacles e.g., raindrops or atmospheric effects, in additional regular blockage issues

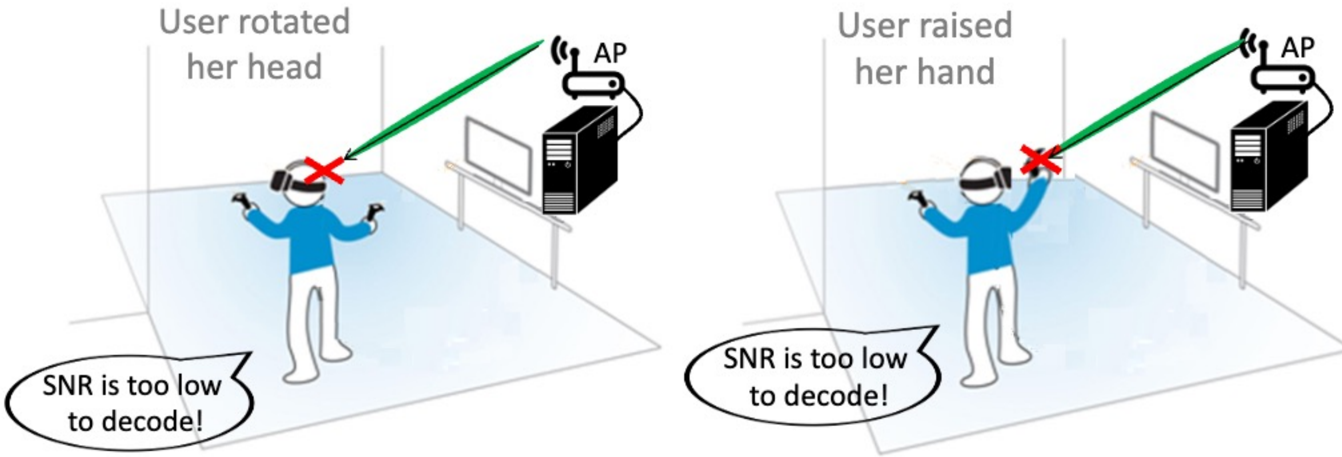


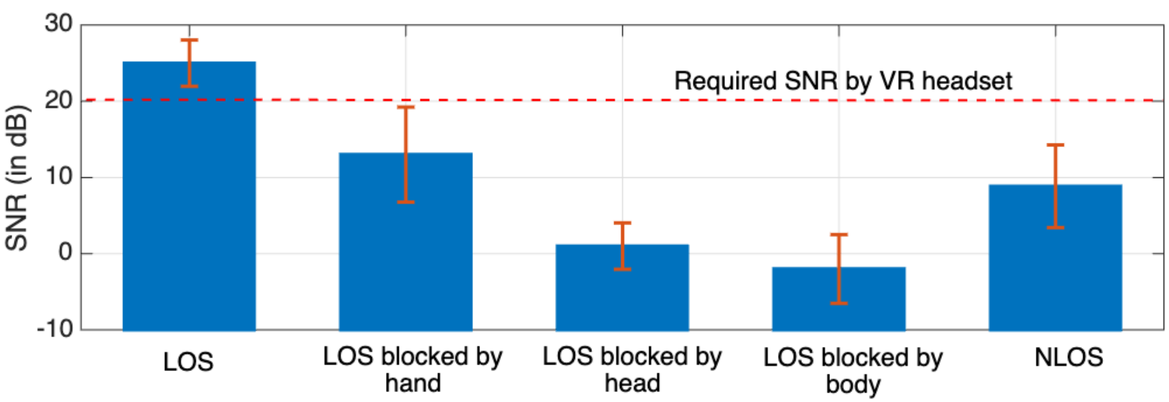
THz Band based VR Link

- Need Beam alignment algorithms
  - RF anchors can be placed in the environment for absolute location estimate
- Predict, track and point beams based on mobility models



mmWave based VR Links





Abari et.al

mmWave based VR Links

- Build a highly directional antenna by packing multiple antenna elements into an array, and controlling the phase of each element.

mmWaves based VR Links

- HTC Vive

Distributed or Parallel Rendering

- Splitting rendering tasks across multiple machines or nodes, often used in high-end graphics production and complex simulations.
  - Each node processes a portion of the rendering task, and the results are combined to produce the final image or animation.

Distributed or Parallel Rendering

- Pixar’s RenderFarm

Render their
big-screen 3d
animated
films

Summary of the Lecture

- Different types of rendering
- Rendered video streaming over wireless

Next up: Hybrid rendering