EEC5512
Networked XR Systems

Last Class - Recap

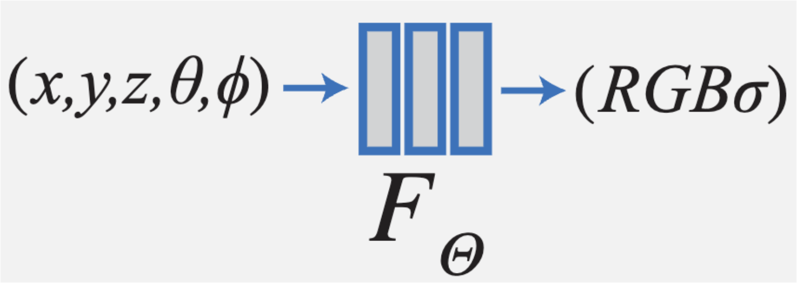
- Quiz
- XR Data Structures, 3D Representations, formats
  - 2D Videos
  - Stereo/3D Videos
  - Multi-view 2D Videos
  - 2D/Flat 360 Degree Videos
  - Stereo/3D 360 Degree Videos
  - 3D/6-DoF Videos (point clouds, meshes, depth maps)
  - Implicit Neural Representations
  - Gaussian splats

Lecture Outline for Today

- Remaining XR/3D Data Representations
  - Implicit Neural Representations
  - Gaussian splats
- View Immersion
- Capturing 3D Videos for Network Transmission
  - Scene Capture
  - Network & Application Interplay
  - Capture Scenarios: Outside-in vs. Inside-out Capture
  - Offline vs. Live Capture
  - Depth Maps, Point Cloud, and Mesh Capture
  - Compute, Bandwidth vs. Latency Trade-offs

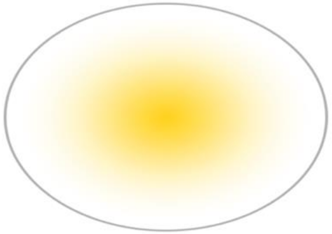
Implicit Neural Representation

- A fully-connected neural network that can generate novel views of complex 3D scenes, based on a partial set of 2D images.
- Set of weights
- To render a view, need to query the neural network by inputting the pose info

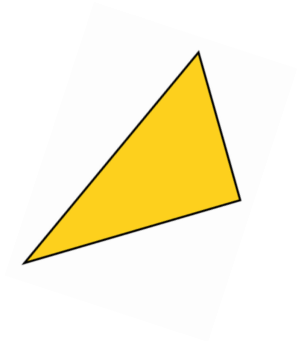


https://www.matthewtancik.com/nerf

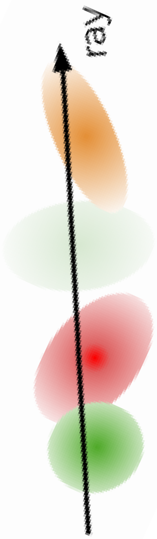
Gaussian Splats



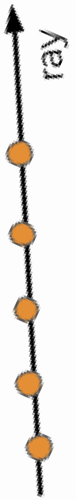
Scene is represented with a number of Gaussian distributions



Mesh is made up of triangles



Gaussian Splat



NeRF

- **Position**: where it's located (XYZ)
- **Covariance**: how it's stretched/scaled (3x3 matrix)
- **Color**: what color it is (RGB)
- **Alpha**: how transparent it is (α)

View Immersion

- Monocular
- Stereoscopic
- Multi-view

View Immersion

- Mono or monocular
  - Single camera
  - Simple, low cost
- Limitations
  - No depth perception
  - No interaction
  - No motion parallax
  - Limited FoV

View Immersion

- Stereo or Stereoscopic
  - 2 cameras
- Depth perception depends on the baseline
- Limited by small field of view

Apple spatial videos

View Immersion

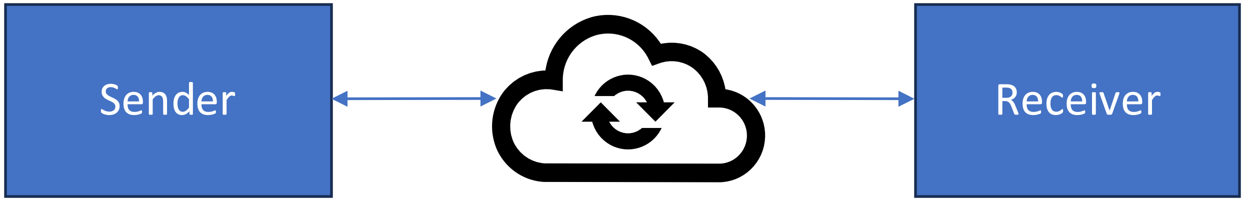
- Multi-view videos
  - Typically, tens to hundreds of cameras are deployed to get full 3D 360^{o} view of the scene of interest
  - Highest level of immersion
  - Costly
  - Very infra heavy
  - Bandwidth heavy
  - Compute heavy
  - Hard (almost impossible) to get in real-time/live

Lecture Outline for Today

- Remaining XR/3D Data Representations
  - Implicit Neural Representations
  - Gaussian splats
- View Immersion
- Capturing 3D Videos for Network Transmission
  - Scene Capture
  - Network & Application Interplay
  - Capture Scenarios: Outside-in vs. Inside-out Capture
  - Offline vs. Live Capture
  - Depth Maps, Point Cloud, and Mesh Capture
  - Compute, Bandwidth vs. Latency Trade-offs

Networked XR System

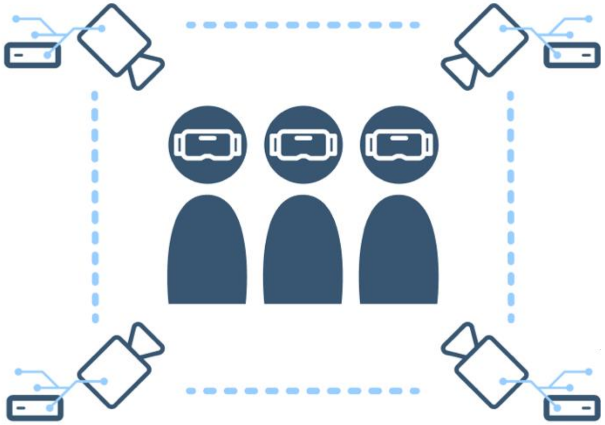
Classical networked application pipeline



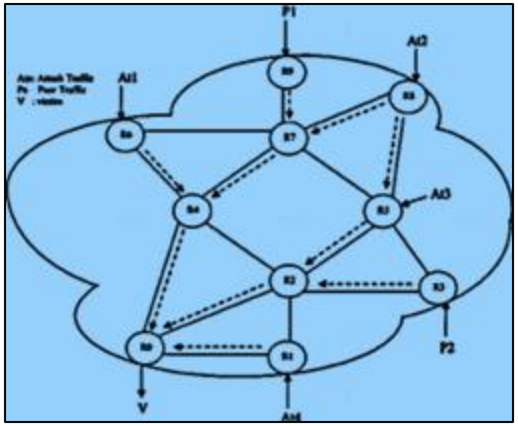
Sender

Receiver

XR networked application pipeline



Digitize 3D spaces



Network Transport

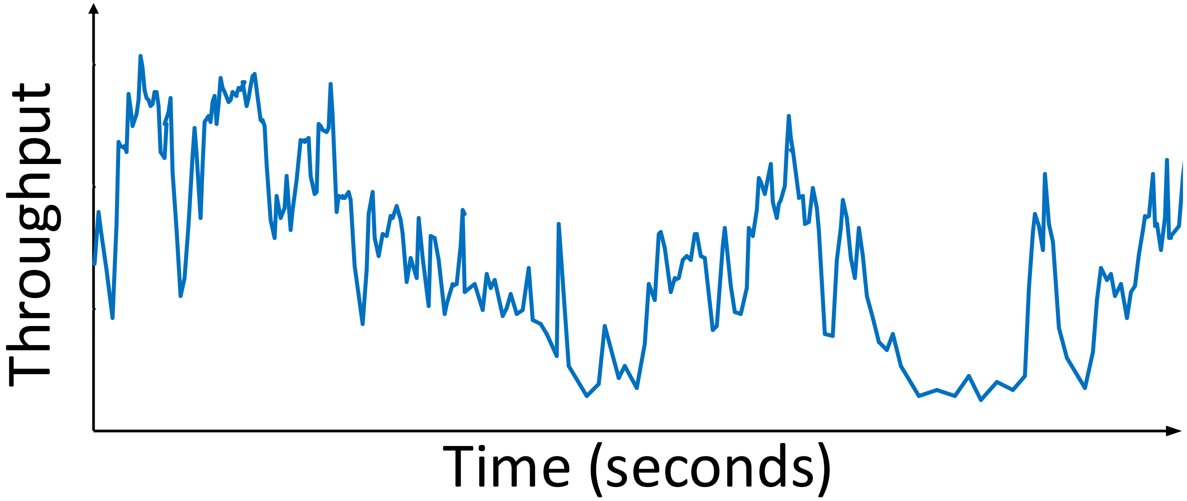
Display Interfaces

Scene Capture

- Storage vs. Network Transmission
- What are the requirements?
  - Storage: Less data is better
  - Network: Low data rate is better

Scene Capture

- Data rates should be flexible to change as the network conditions changes – introduces some overhead



Capturing 2D Scenes or Videos

- Mostly mature – work done for nearly 3 decades
- Plenty of hardware support to process 2D video streams
- Still a lot of research happening to reduce power consumption or improving the quality of experience under poor network conditions
    - Advances in low power image sensors

Scene Capture for Network Transmission

- Why transmit over network
  - Share 3D content with others
  - Machine to machine 3D analytics
  - Access 3D movies

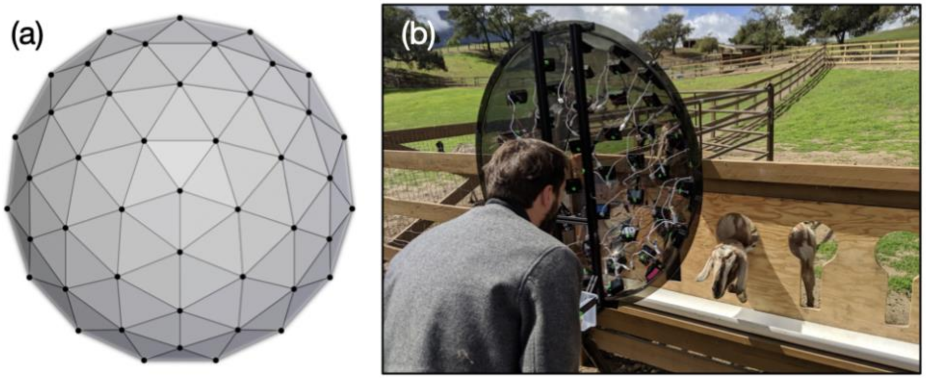
  - Many use cases that we saw in the previous lectures

Capture Scenarios

- Inside-out: Mobile Devices or Headsets
  - iPhone Lidar capture or stereo/spatial videos
  - 2 color cameras and a depth camera
  - Or Vision Pro or Quest3 captures

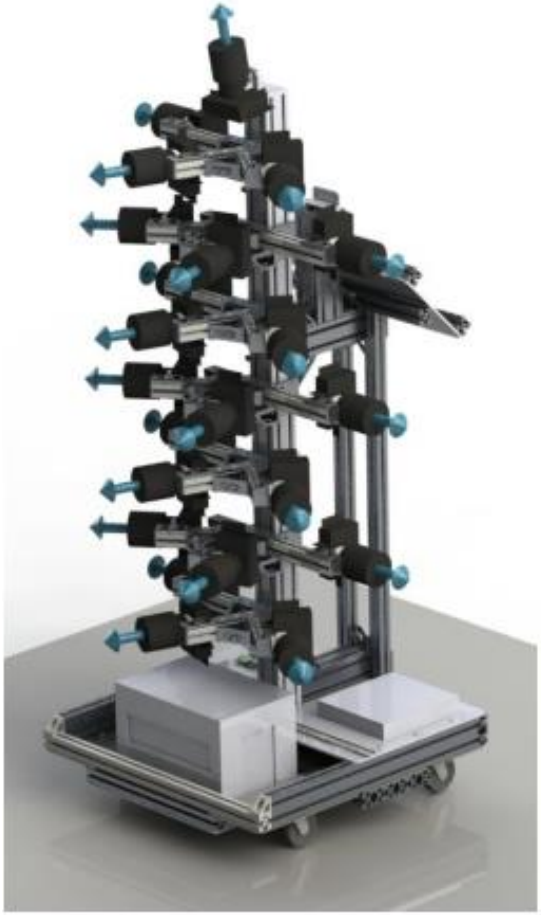
Capture Scenarios

- Inside-out: Multi-camera infrastructure
  - Cameras are placed at vertices of an icosahedral tiling of a 0.92 m diameter hemisphere. This yields an average inter-camera spacing of 18 cm.



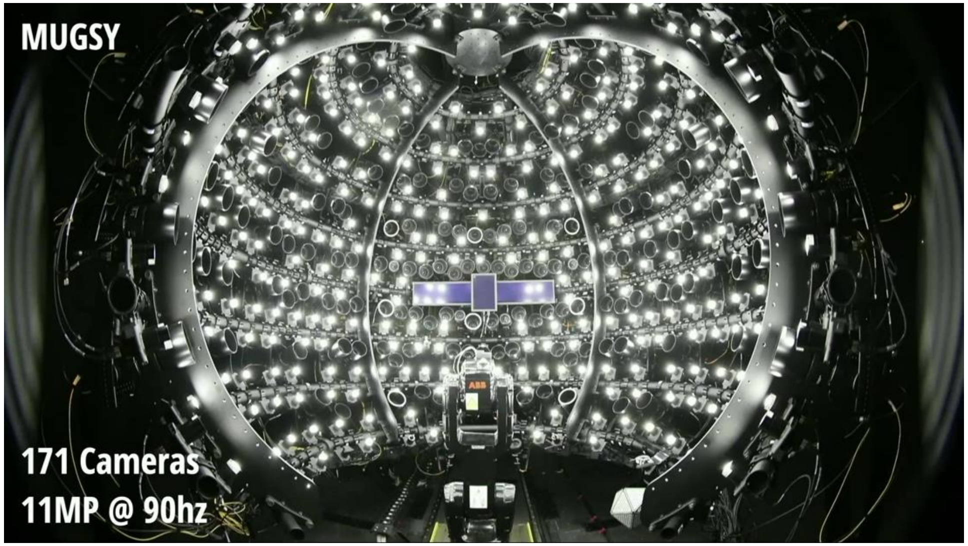
Capture Scenarios

- Inside-out : Multi-camera infrastructure
  - 80×80 cm base with a 1.8 m vertical pole for 22 cameras that are distributed on 7 levels with 3 cameras each, plus one upward-facing camera at the top



Capture Scenarios

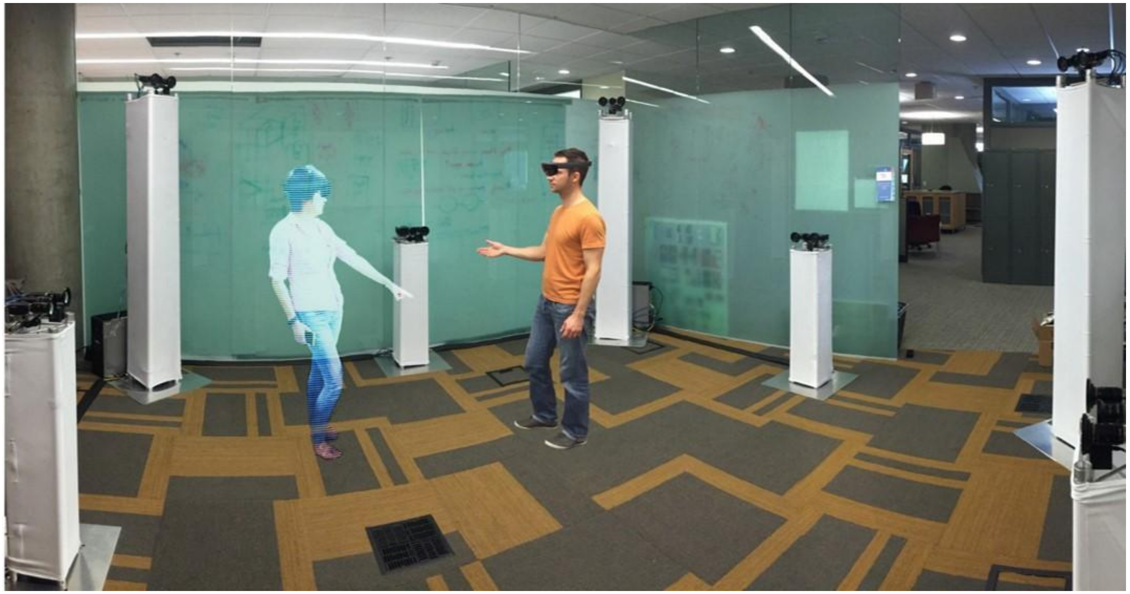
- Outside-in: Multi-camera infrastructure



Meta’s Mugsy

Capture Scenarios

- Outside-in: Multi-camera infrastructure



RGB & Depth cameras

Live Capture vs. Offline

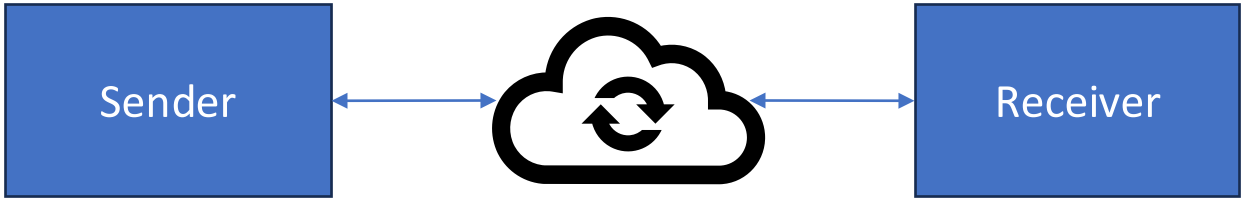
- Offline capture does not pose problems
    - Enough time and resources to process the content
- Live capture has stringent requirements
    - Low latency (<100ms)
    - Trade quality with latency and bandwidth

Live 3D Capture

- Many options
  - Our favorite data structures:
    - Depth Maps
    - Point Clouds
    - Triangle Meshes

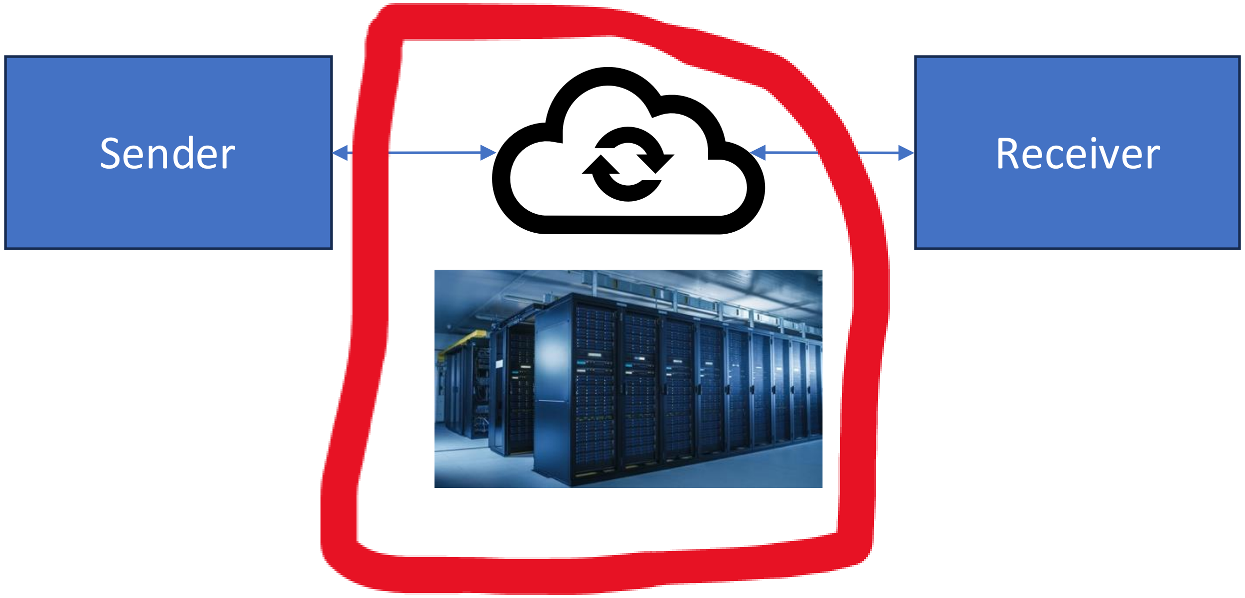
Live 3D Capture

- Different data structures captured at the sender have different implications on the network and receiver device
  - Rendering input: Triangles
  - Where you place the triangle extraction i.e., 3D mesh reconstruction computation matters (particularly for devices like headsets or phones).



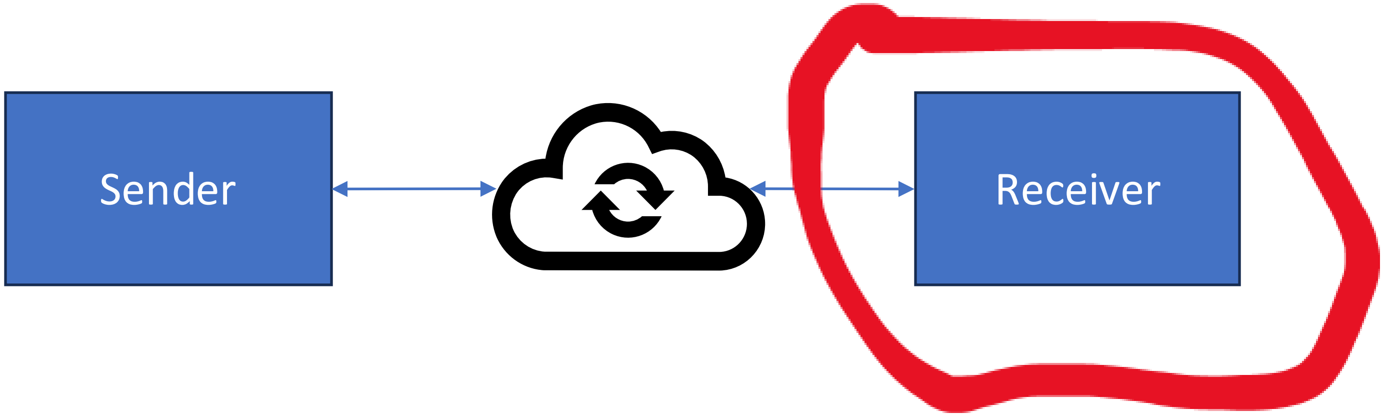
Capturing Depth Maps

- Possible end-to-end streaming pipelines
  - Cloud based mesh reconstruction
    - In general, many resources – Fast, High Quality
    - Caution on bandwidth requirement



Capturing Depth Maps

- Possible end-to-end streaming pipelines
  - Receiver-side mesh reconstruction
    - Fewer resources – Slow, Low Quality
    - Additional power consumption due to reconstruction computation – bad for XR devices

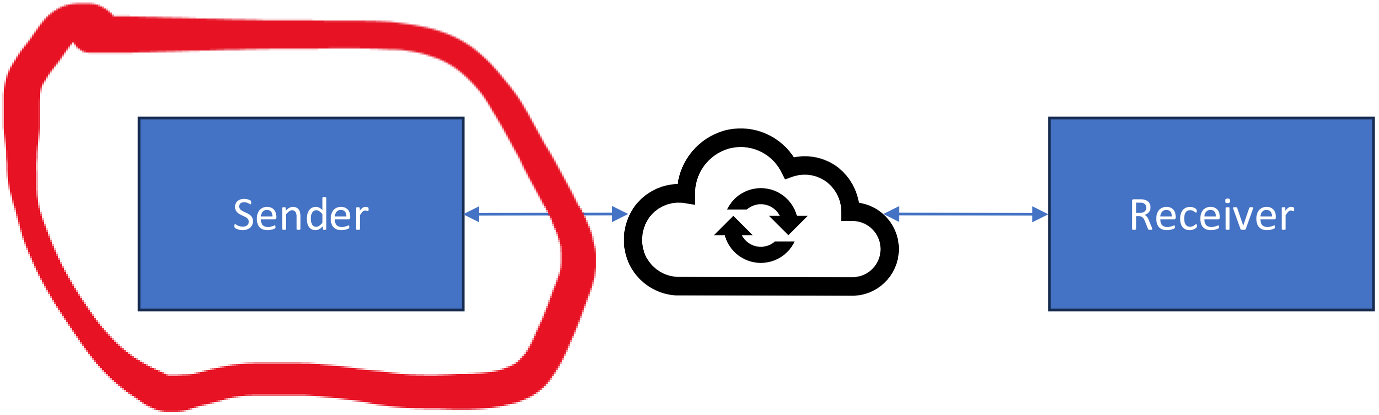


Capturing Point Clouds

- Natively available on the sensor like Depth maps (e.g., Lidar)
- Or a depth map can be converted to a point cloud with a simple transformation
  - Very little computation for transformation
  - i.e., sender-side pipeline is not affected as much
- Possible end-to-end streaming pipelines?
  - Similar to Depth maps, including the implications

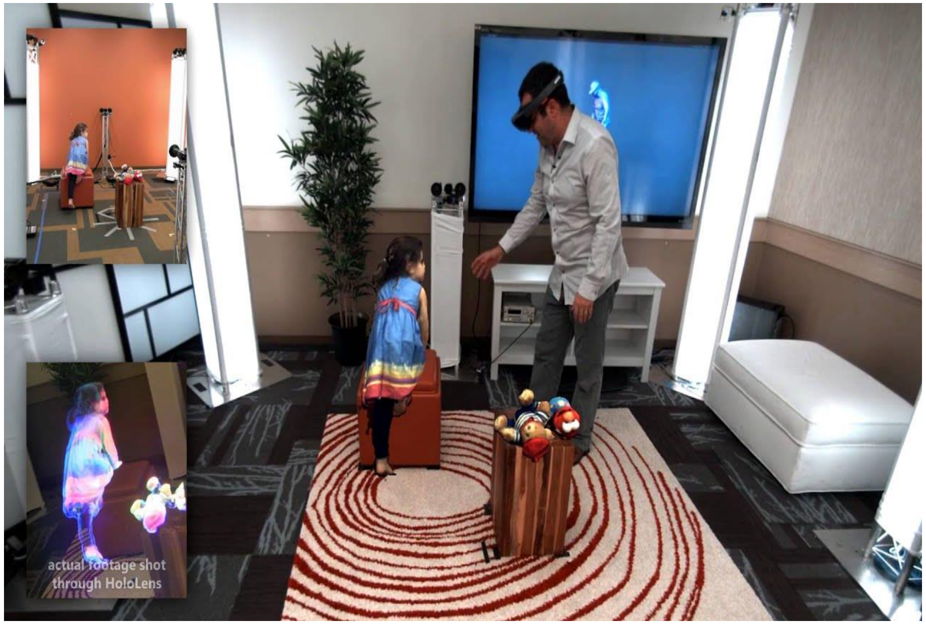
Capturing Meshes

- Meshes are not available natively on the sensor
  - Computation burden on the sender
  - No need for cloud (at least not for reconstruction; for rendering maybe – we'll talk about that later)
  - Triangle mesh is readily available for receivers – no overhead of reconstruction, less power consumption
  - Sender overhead depending on outside-in or inside-out or the number of cameras



Real-world Examples

- Microsoft Holoportation
    - Extracts mesh on the sender-side
    - Outside-in capture
    - Infra heavy
    - Sufficient resources for 3D reconstruction



Real-world Examples

- Google Project Starline
  - 8 Depth videos are streamed
  - Reconstruction computation is placed on the receiver
  - Both sender and receiver have similar computation resources

Real-world Examples

- Apple Vision Pro
  - Sender-side reconstruction
  - 3D reconstruction maybe fast but still consumes power
  - Receivers could be other XR headsets

Live 3D Capture

- Depth Map vs. Point Cloud vs. Mesh

- Outside-in
    - Most scenarios sender has more resources
    - Sender-side reconstruction strikes a good balance

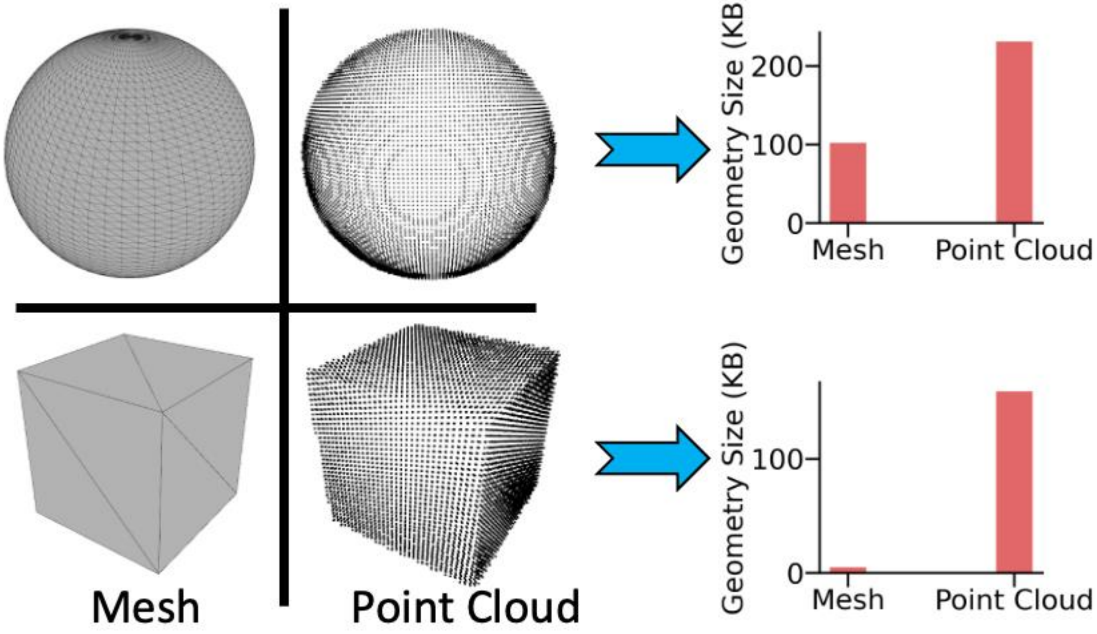
- Inside-out
    - Most scenarios senders do not have enough resources (e.g., phones or headsets)
    - Cloud is a good option

Live 3D Capture

- Depth Map vs. Point Cloud vs. Mesh
- Implications on the network?
  - Each data structure has significantly different bandwidth requirement
  - It is unclear which is better – still in experimental research phase, no consensus yet; need to study diverse scenarios.

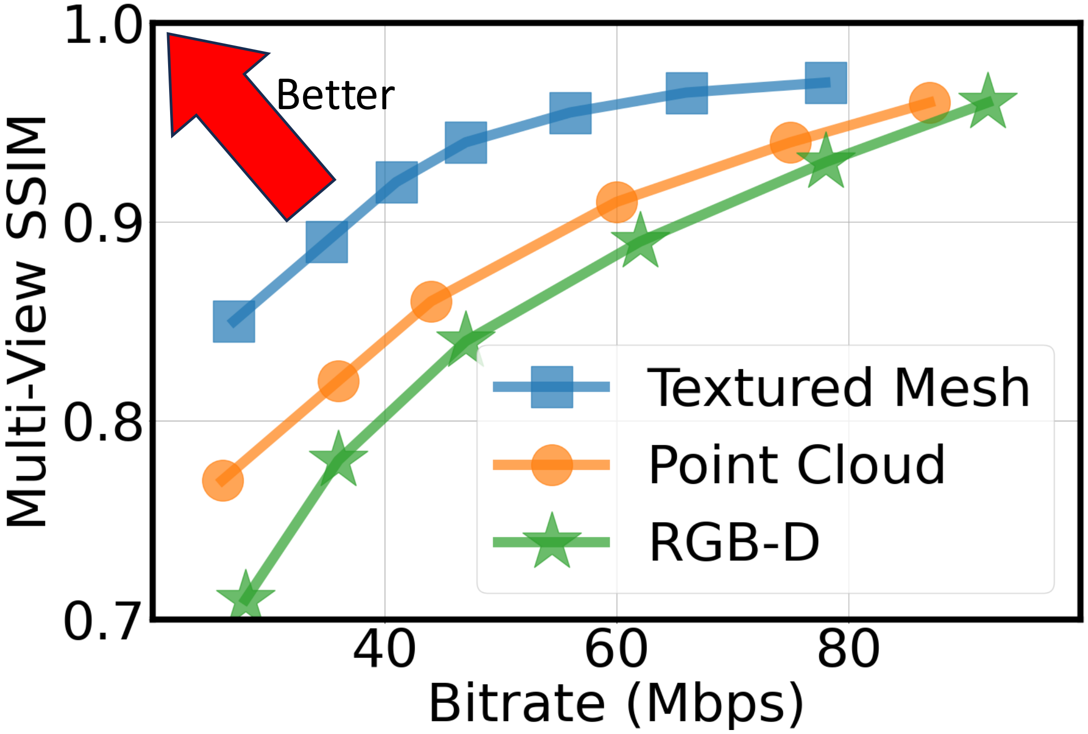
Early Findings

- Mesh is compact



Early Findings

- Mesh requires relatively lower bandwidth for a given final rendering visual quality



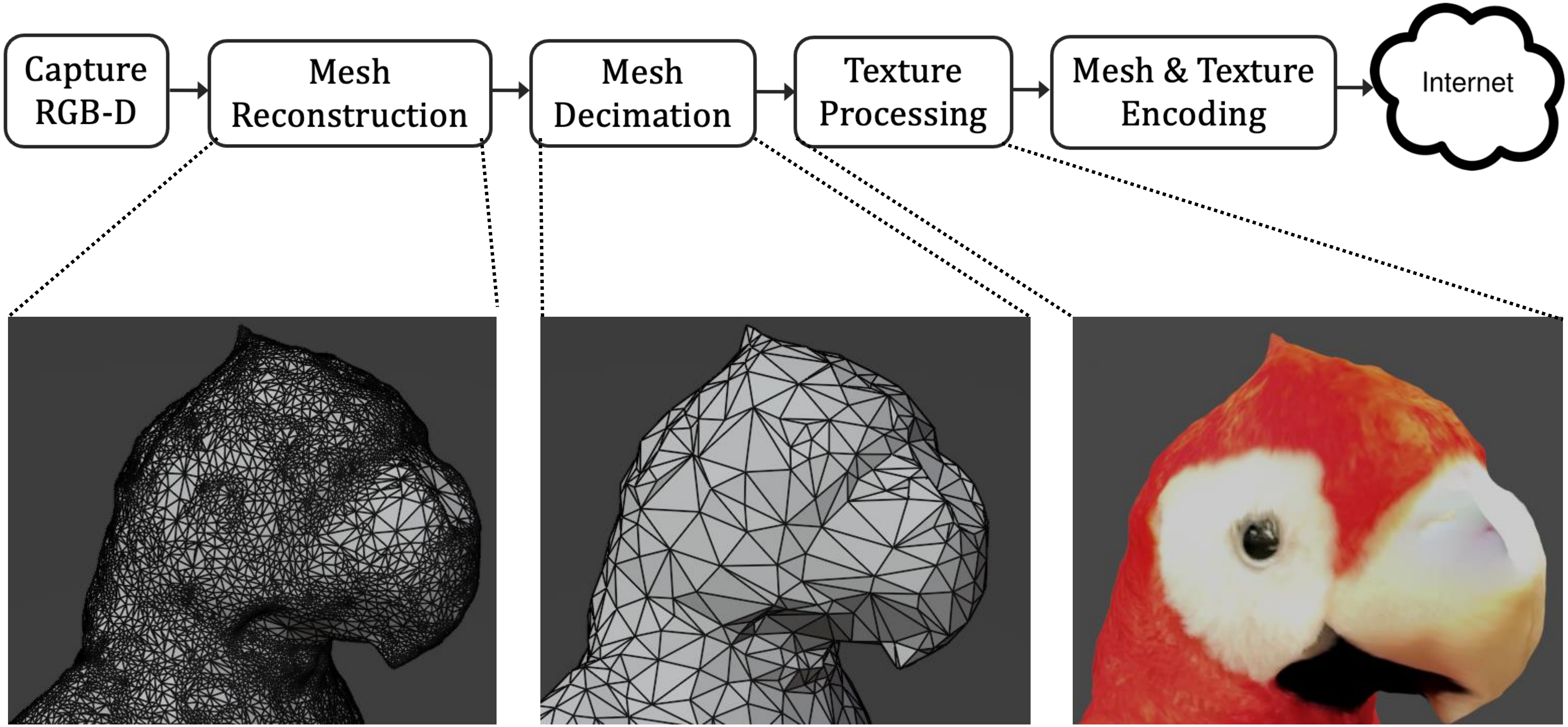
Live 3D Capture

- Depth Map vs. Point Cloud vs. Mesh
- Meshes are generally superior – assuming we can tackle the computation challenge on the sender side
- Several reasons
    - Compact
    - High resolution texture
    - Compatible for rendering hardware - triangles

Live Capture of Meshes

- Texture is given – we can use existing hardware pipelines for 2D videos to capture and stream textures
- Extracting meshes is a complex process
  - Involves a series of computationally expensive reconstruction steps
  - Outside-in scenario: fusing multiple scenes together; adds additional computation

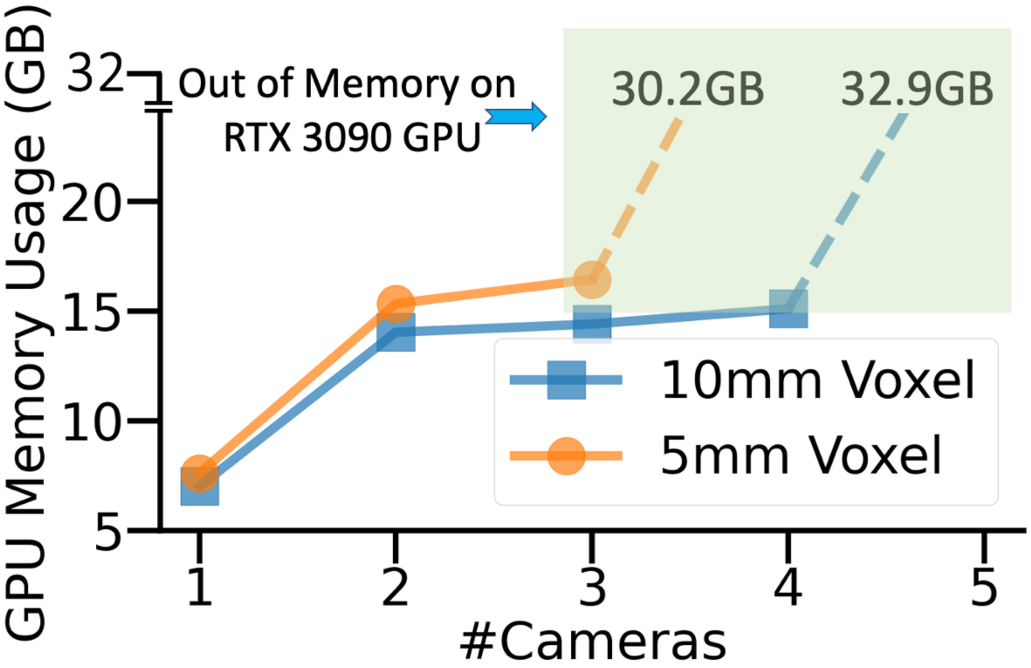
Live Capture of Meshes



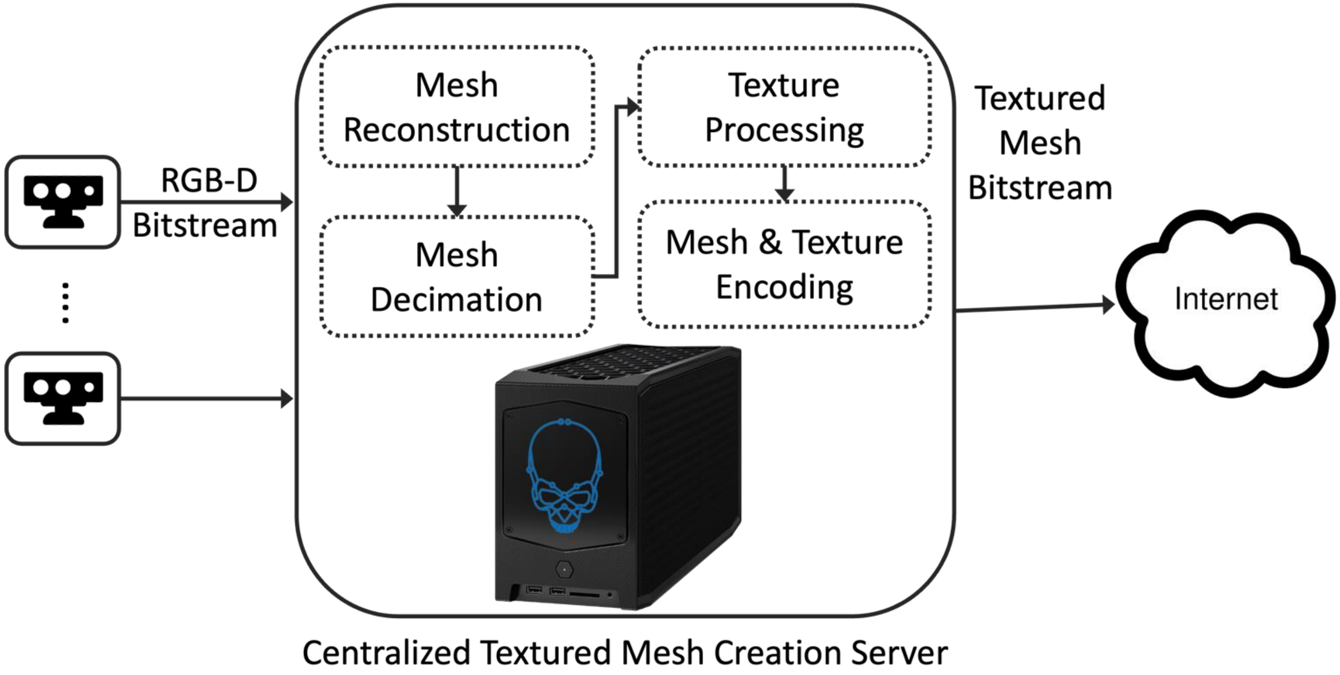
Capture RGB-D → Mesh Reconstruction → Mesh Decimation → Texture Processing → Mesh & Texture Encoding → Internet

Live Capture of Meshes

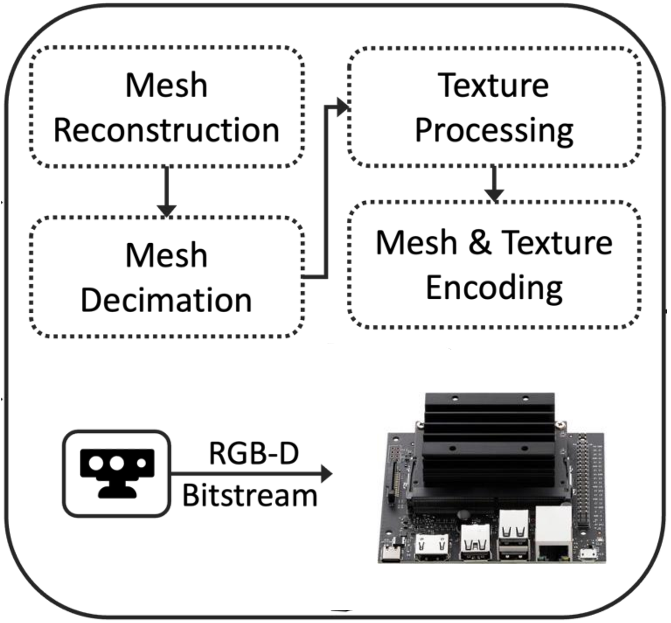
- Single camera vs. multi camera reconstruction
  - GPU memory runs out of memory quickly
  - Depends on the voxel resolution
  - What is voxel?



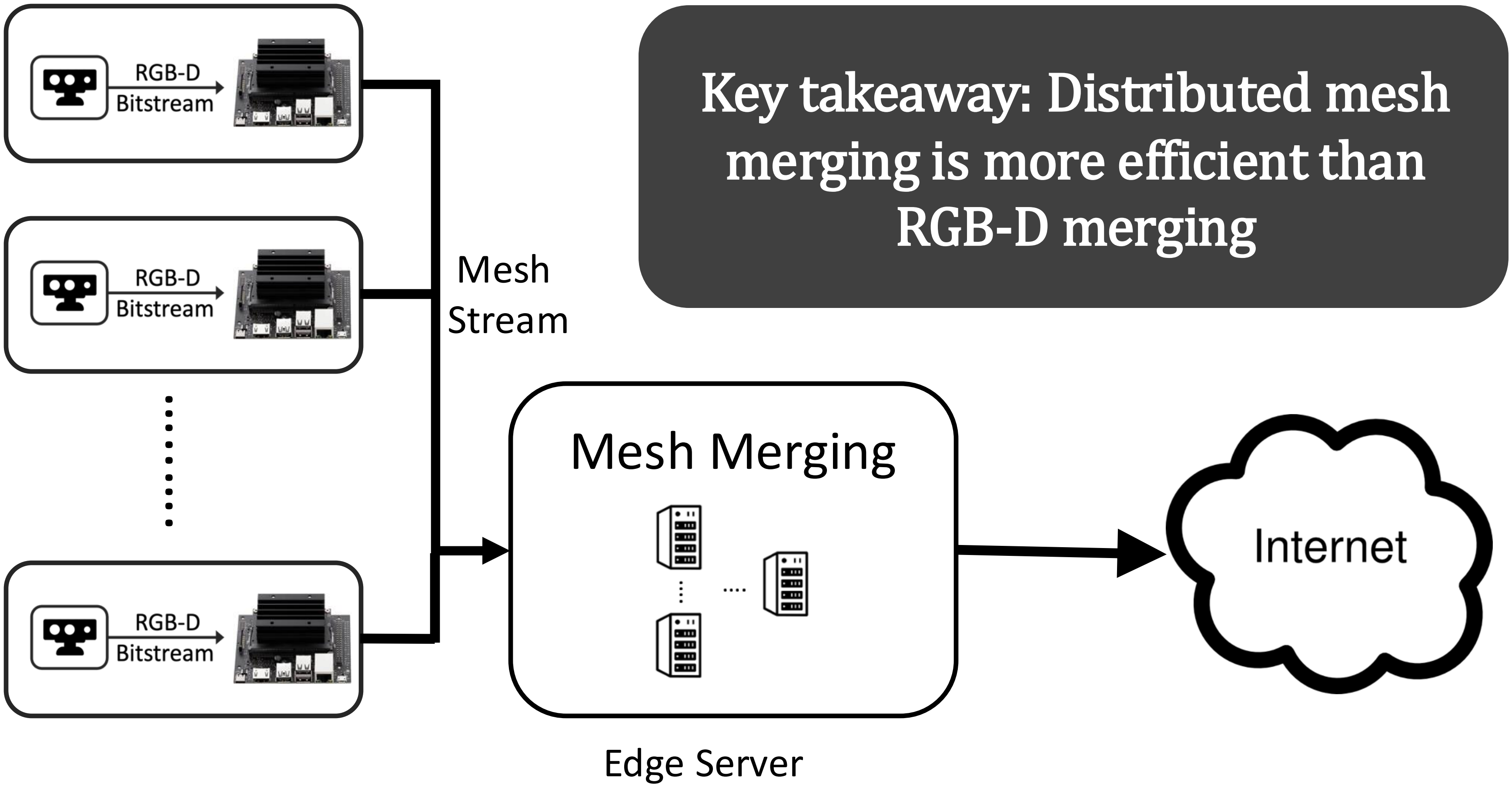
Live Capture of Meshes



Live Capture of Meshes



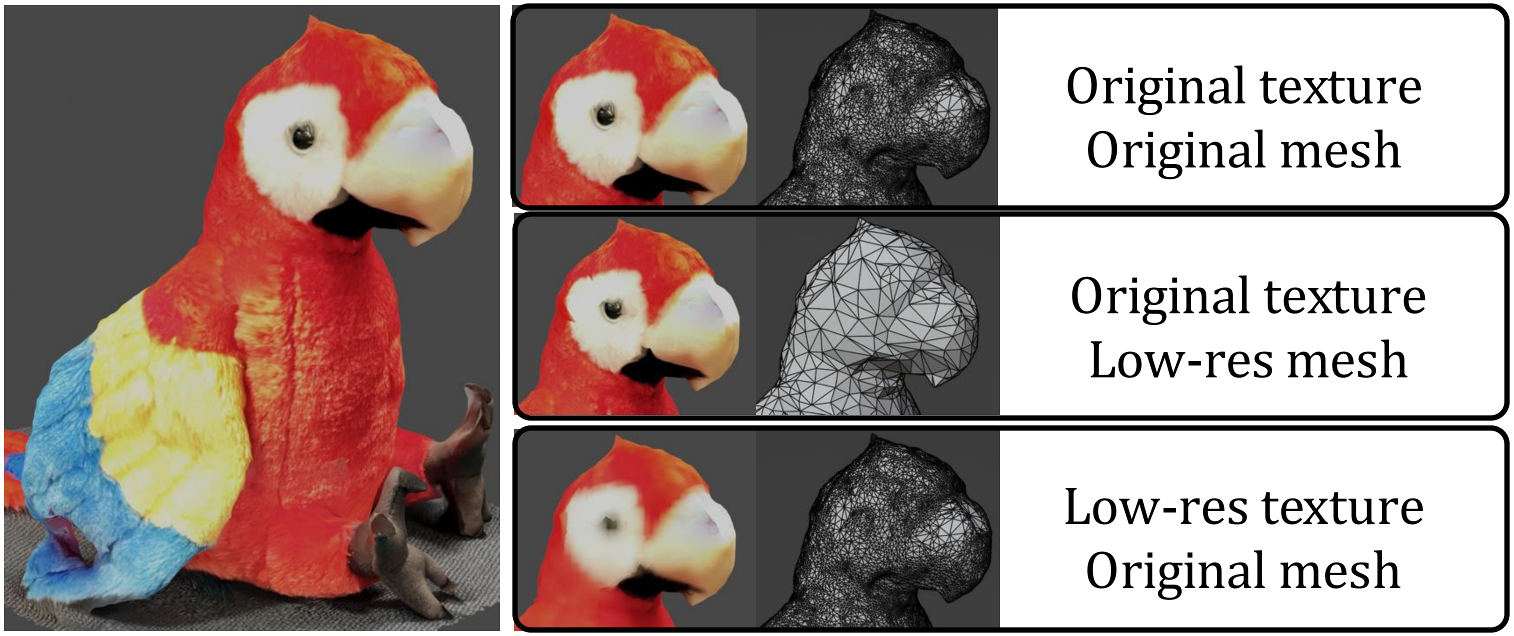
Live Capture of Meshes



Key takeaway: Distributed mesh merging is more efficient than RGB-D merging

Live Capture of Meshes

- Texture vs. Mesh bandwidth



Original texture Original mesh	164MB
Original texture Low-res mesh	8.2MB
Low-res texture Original mesh	8.2MB

Summary of the Lecture

- Scene Capture
    - Computation, bandwidth, latency implications
- Capturing different 3D Data Structures
- Sender, Cloud and Receiver-driven Pipelines
- Distributed Mesh Reconstruction

Next Up: Compression Fundamentals