MuV2: Scaling up Multi-user Mobile Volumetric Video Streaming via Content Hybridization and Sharing

Yu Liu1, Puqi Zhou2, Zejun Zhang1, Anlan Zhang1, Bo Han2, Zhenhua Li3, Feng Qian1

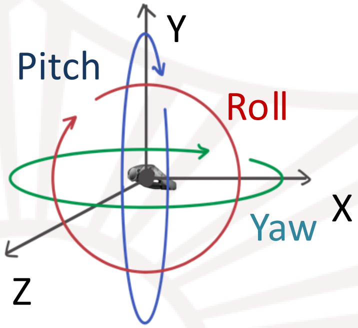
1 University of Southern California (USC)

2 George Mason University (GMU)

3 Tsinghua University

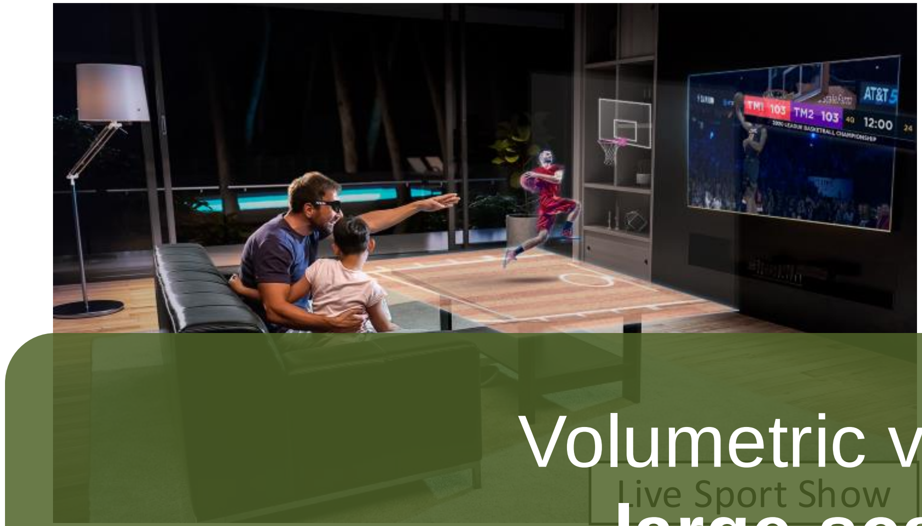
Extended Reality and Metaverse is growing

Volumetric Videos: Introduction

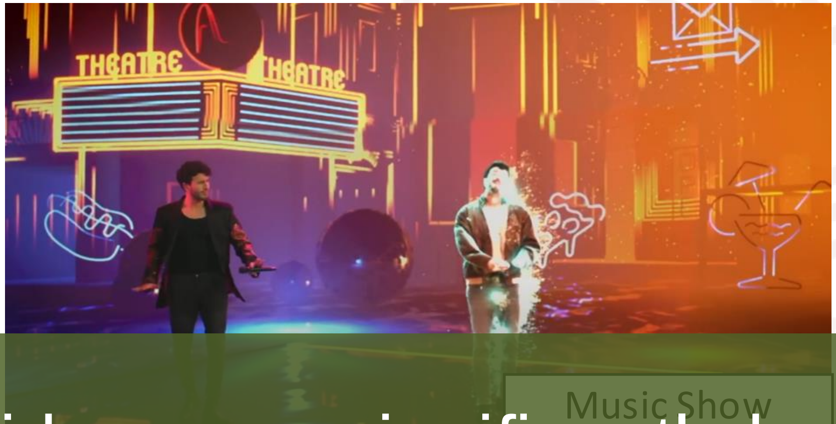


- A time series of **fully 3D** representation captured with multiple RGB(-D) cameras
- Support **6 Degree-of-Freedom (DoF)** movement
- Multiple representations:
  - **Point Cloud**: a group of unsorted points
  - **3D Mesh**: a collection of vertices, edges, and faces
  - **Neural Models:** A trained neural model representing the 3D scene(NeRF, Gaussian Splatting)

Volumetric Video Streaming: Application



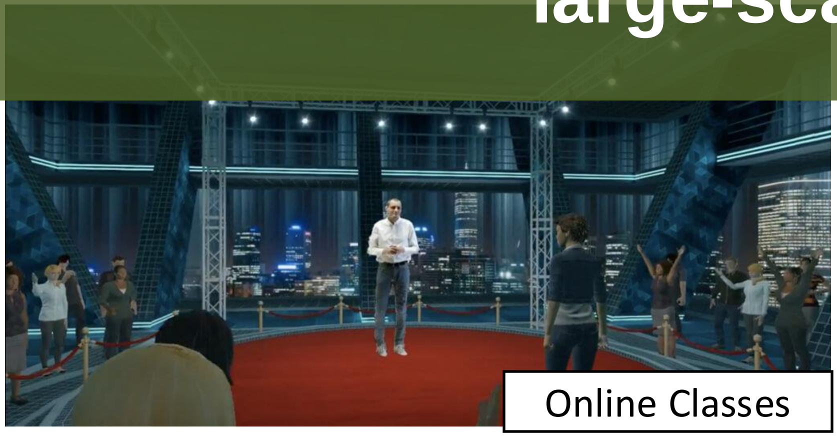
Live Sport Show



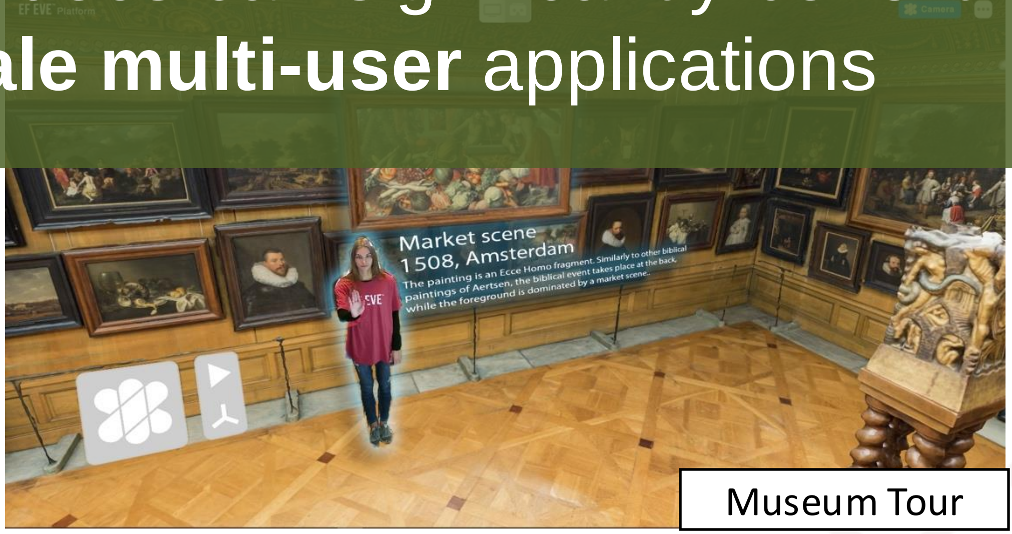
Music Show

Online Meeting

Volumetric videos can significantly benefit **large-scale multi-user** applications



Online Classes



Museum Tour

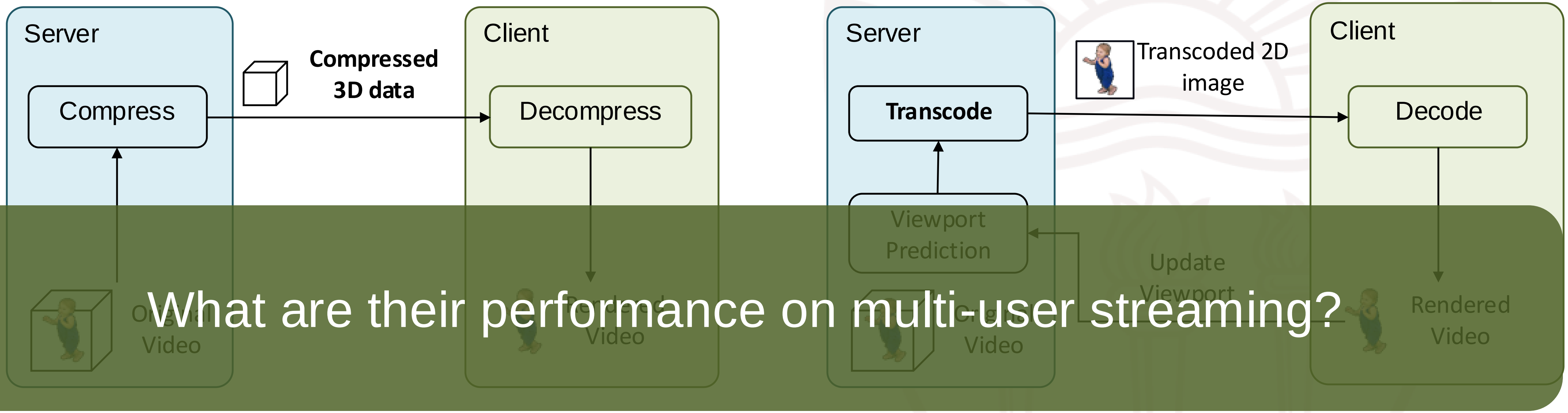
Online Lecture

Volumetric Video Streaming: Challenges

- Data volume and bandwidth consumption
  - A medium-quality point cloud (PtCl) volumetric video featuring a single person (~160K points/frame) requires more than 500 Mbps (raw) or 100 Mbps (compressed) to stream at **30 FPS**
- Processing and compression overhead
  - Most **CPU-accelerated** compression
  - State-of-the-art compression method* can only achieve **6:1** compression ratio
    - H264 can achieve about 2000:1**
- Multi-dimension user movement
  - 6-DoF movement leads to **complex** user movement pattern
  - More challenging for **predicting** users' viewport

- * Draco: https://github.com/google/draco
- ** https://www.rgb.com/h264-profiles#:~:text=High%20Profile-,H.,ratio%20of%20about%202000%3A1.

Volumetric Video Streaming: State-of-the-Art



What are their performance on multi-user streaming?

Directly stream compressed volumetric video to clients

> Reduce the size and proportion of the original video to be streamed

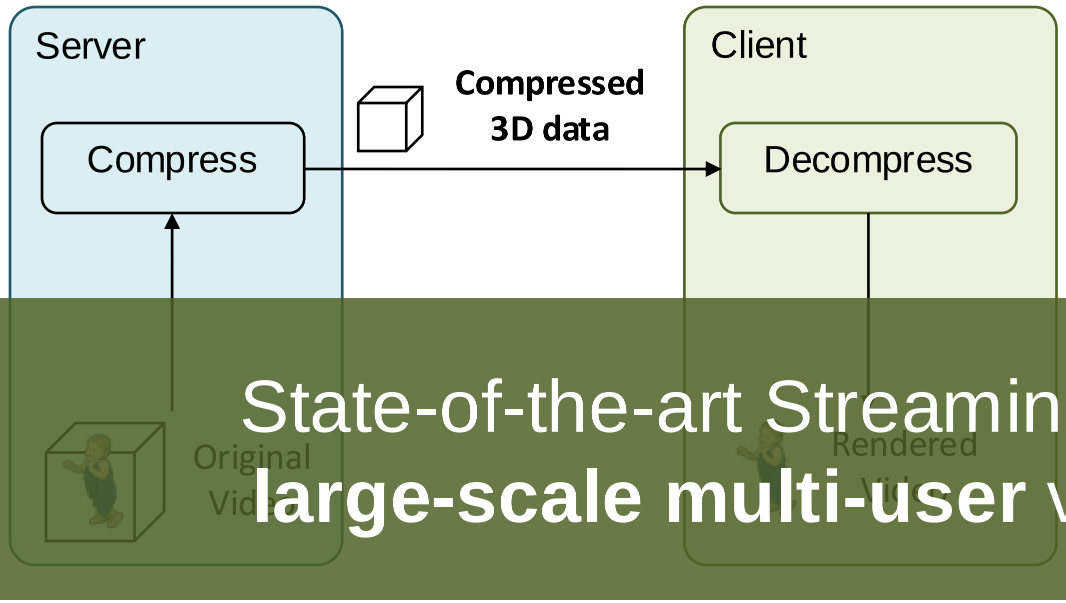
> Improve encoding efficiency

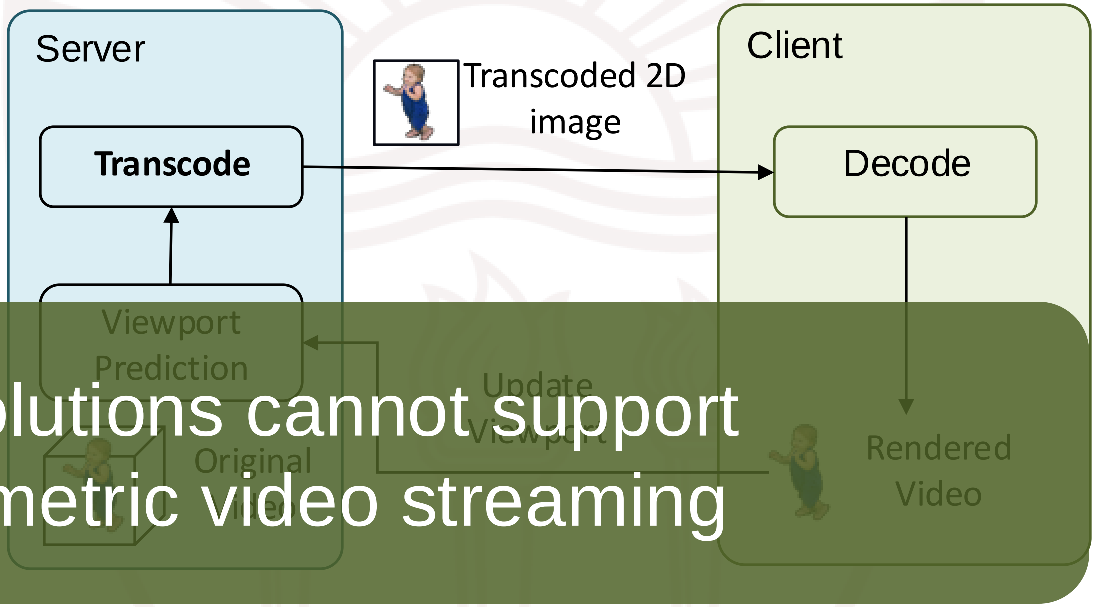
Transcode volumetric videos into 2D video stream

> Image-based rendering or multi-view to reduce distortion

> Improve viewport prediction accuracy

Volumetric Video Streaming: Single-user to Multi-user





State-of-the-art Streaming Solutions cannot support **large-scale multi-user** volumetric video streaming

Directly stream

> Sending multiple copies of compressed volumetric video incurs **higher bandwidth requirement**

> Can support at most **5~6** users losslessly at **30 FPS**

Transcode stream

> Performing remote rendering and encoding for multiple users incurs **higher computational overhead**

> Can support at most **8** users at **30 FPS** (with Nvidia 2080Ti)

Observations and System Design Considerations

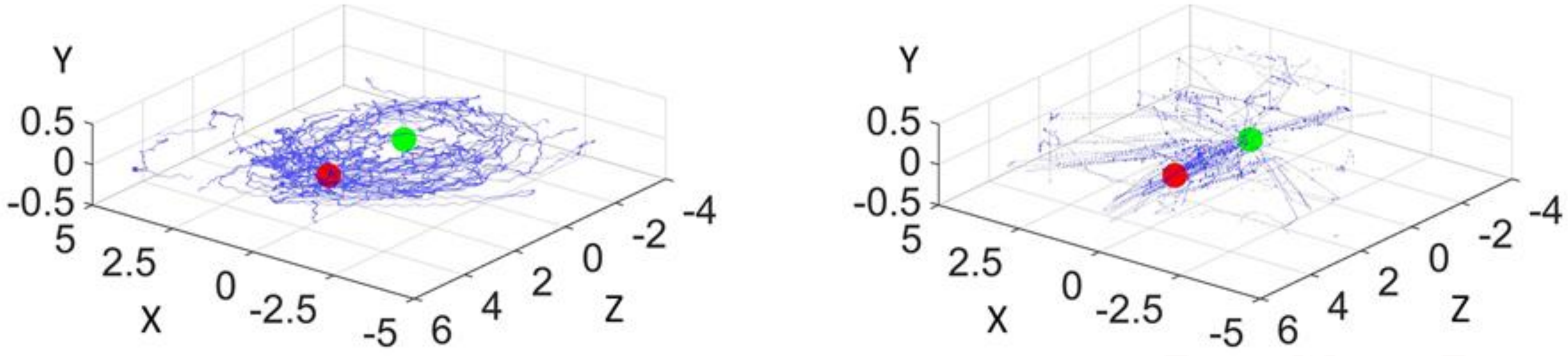
- Limited resource becomes the bottleneck for scaling up the system's capacity
    - **Bandwidth** and **network** resources for direct streaming
    - **Computational** resource for transcode streaming
- Resource requirements scale linearly with the increase in the number of users
- System design principles:
    - Maximize resource utilization
    - Decouple resource requirements from number of users
    - Ensure high and fair quality across users

Our Solution: Content Hybridization

- **Maximize** resource utilization
  - Streaming volumetric content does **not** require computational resources on the edge server
  - Streaming transcoded views requires much **lower** bandwidth resources
- Hybrid streaming approach:
  - Stream **transcoded views** to users by default
  - Stream compressed **volumetric content** to some users under bandwidth limit
  - Compensate **visual quality** drop caused by transcoded views
- Hybrid streaming decision:
  - Stream volumetric content to users that are more likely to have lower visual quality if streamed transcoded views

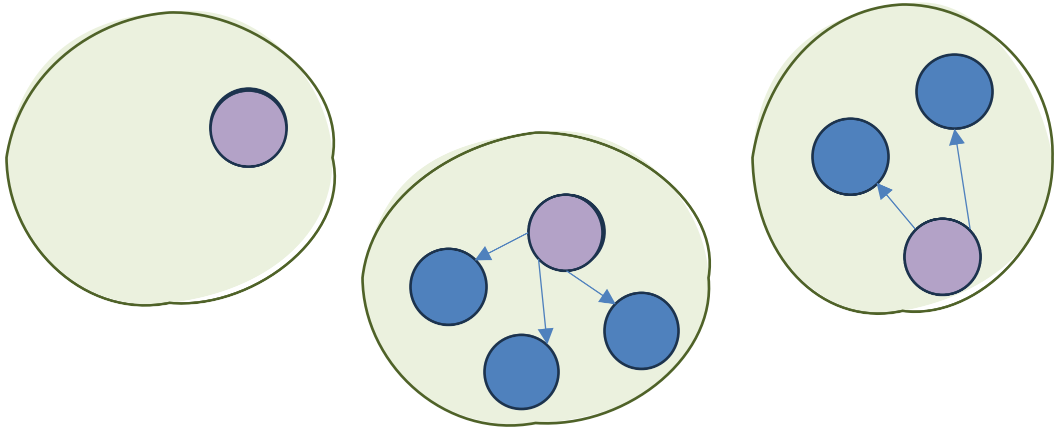
Our Solution: View-Sharing

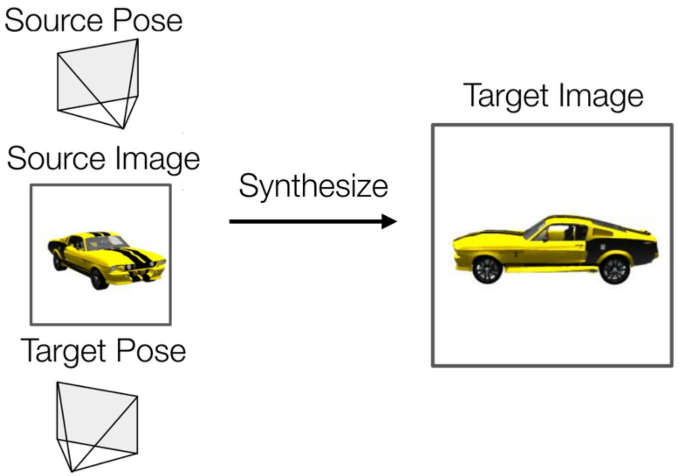
- **Decouple resources requirement** from the number of users
- Observe that users have similar movement patterns while watching
- **Share** the same transcoded view across multiple users to avoid extensive rendering and encoding



Our Solution: View-Sharing

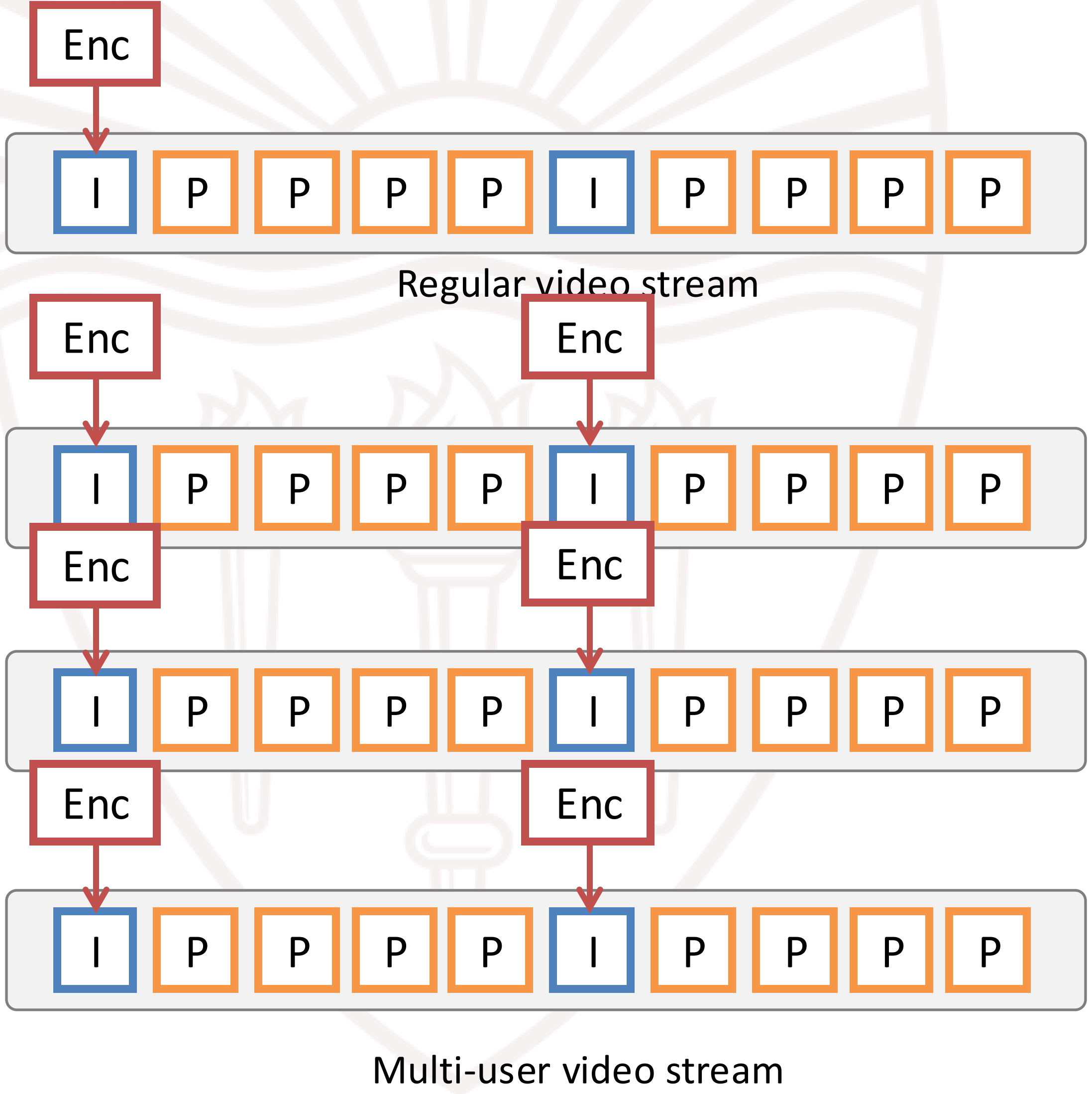
- **Group users** and select **one view** to **share** inside each group
  - Achieve minimum visual distortion across all users
  - Modified greedy algorithm for K-Median Problem
    - Use visual distortion and fairness as "distance" between users
- Use **image warping** to generate novel views for each user





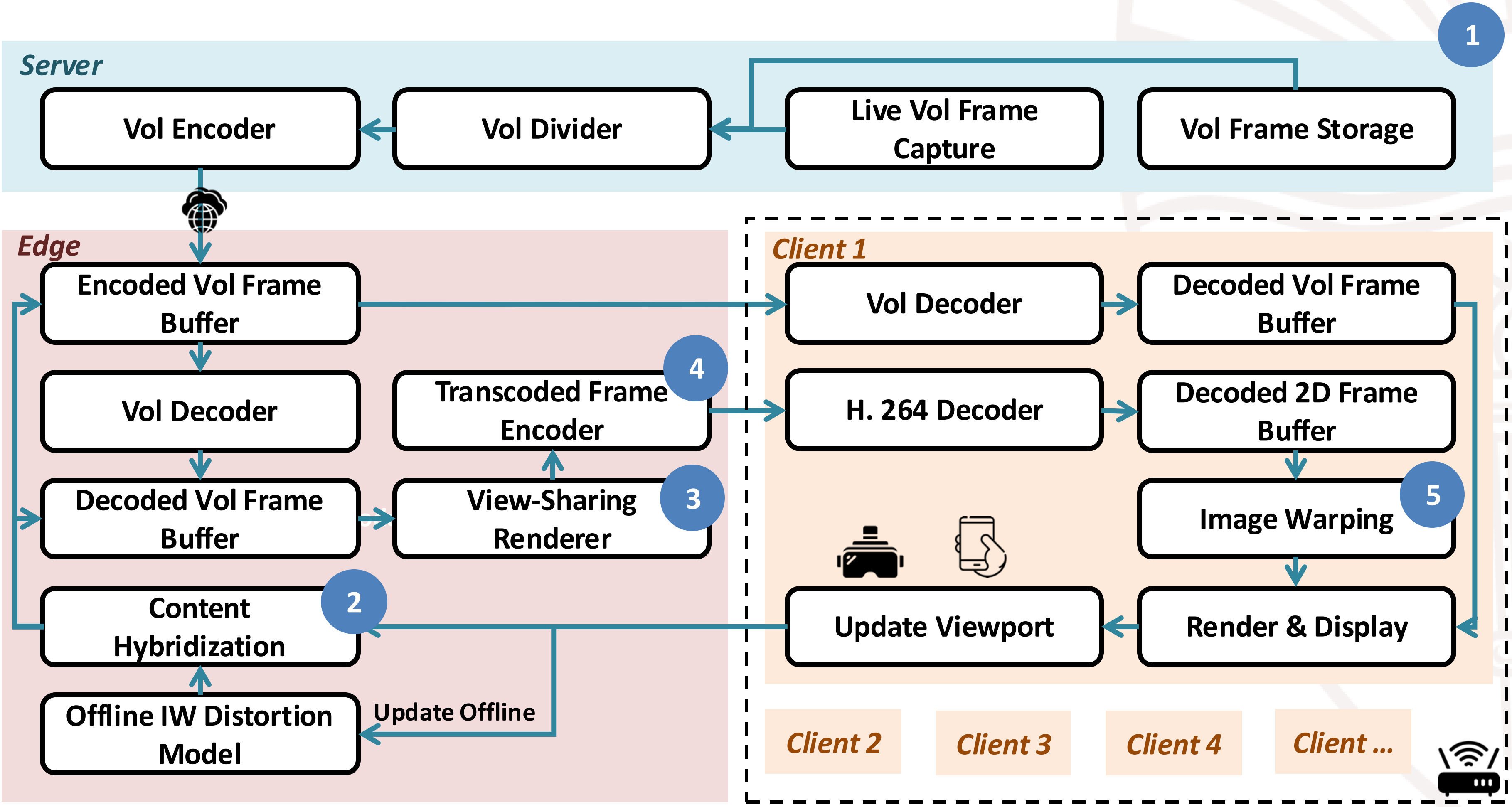
Our Solution: Encoder Multiplexing

- Multiplex **multiple encoding tasks** into **limited hardware resources**
- Infra-frame compression algorithm:
  - Divide video frames into group-of-frames (GoP)
  - I-frame: independently decodable
  - P-frame: require the previous I-frame
  - P-frames have a higher compression ratio
- Round-robin encoding scheme for multi-user with single encoder instance



USC

System Design for MuV2



1 Live or on-demand volumetric content server

2 Content Hybridization decision

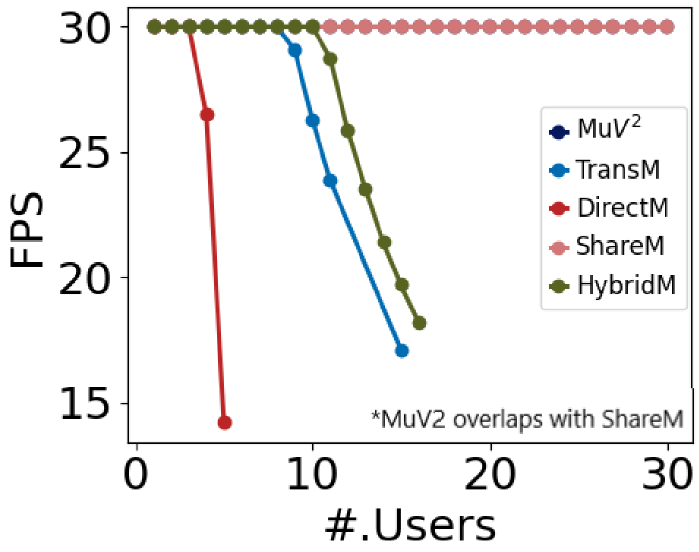
3 User grouping based on real-time viewports

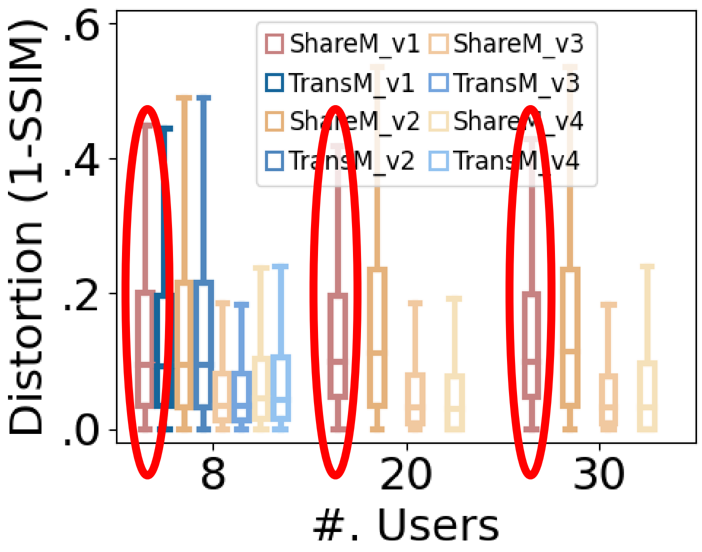
4 Rendering and encoding shared views

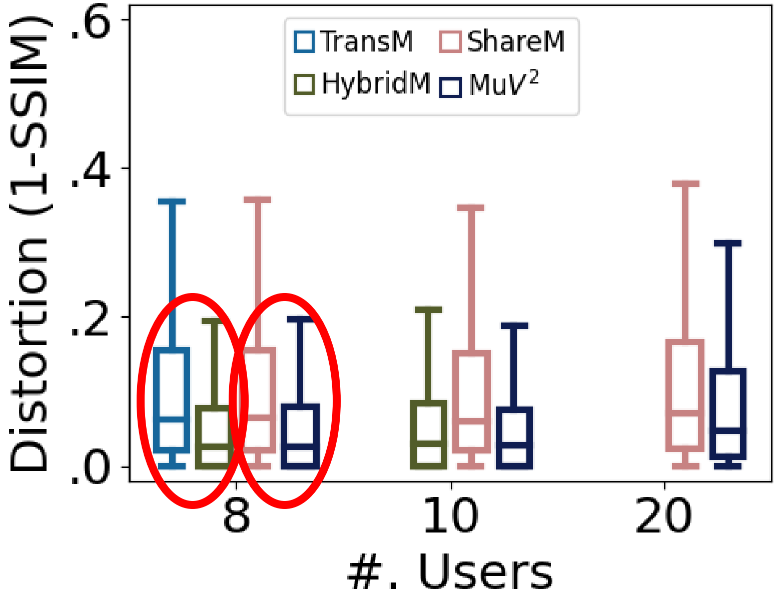
5 Client-side image warping and rendering

USC

Evaluation:



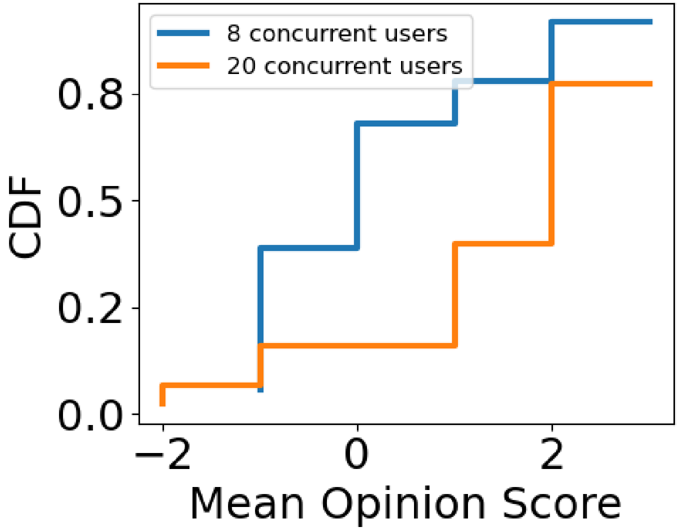


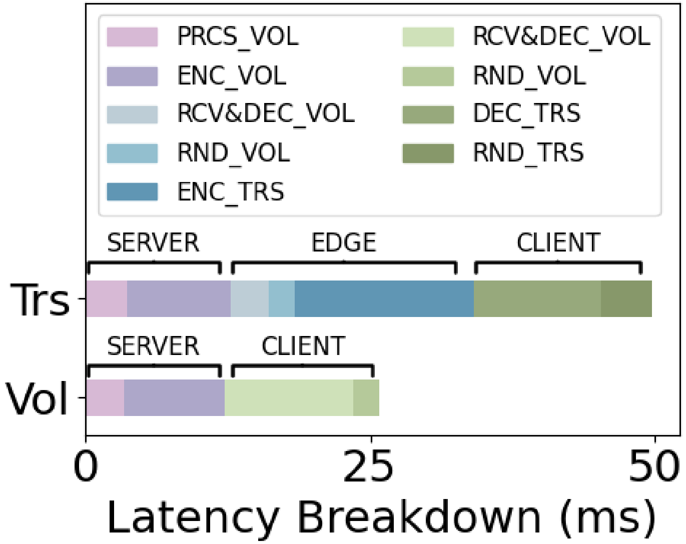


- Supports more than **30** users without frame rate drop
- View-sharing increases visual distortion by only **6%** when increasing from 8 to 20 users
- Hybrid streaming volumetric video frames reduce visual distortion by **48%**

USC

Evaluation:





- User study shows that MuV2 achieves a **90% positive** score compared to using transcode streaming alone with **20 users**
- MuV2 achieves **50ms** processing latency for transcoded views, and **26ms** for volumetric frames
- MuV2 only increases the end-to-end latency by **66ms** for transcoded views and **36ms** for volumetric frames in a real-world live-streaming test scenario

Thank you