# VR/AR Video Streaming Systems

Jounsup Park (jounsup@illinois.edu)
Postdoctoral Research Associate
Coordinated Science Lab
University of Illinois at Urbana-Champaign

04/06/2020

# Outline


Video Streaming Systems

VR Video Streaming Systems

AR Video Streaming Systems

# Outline

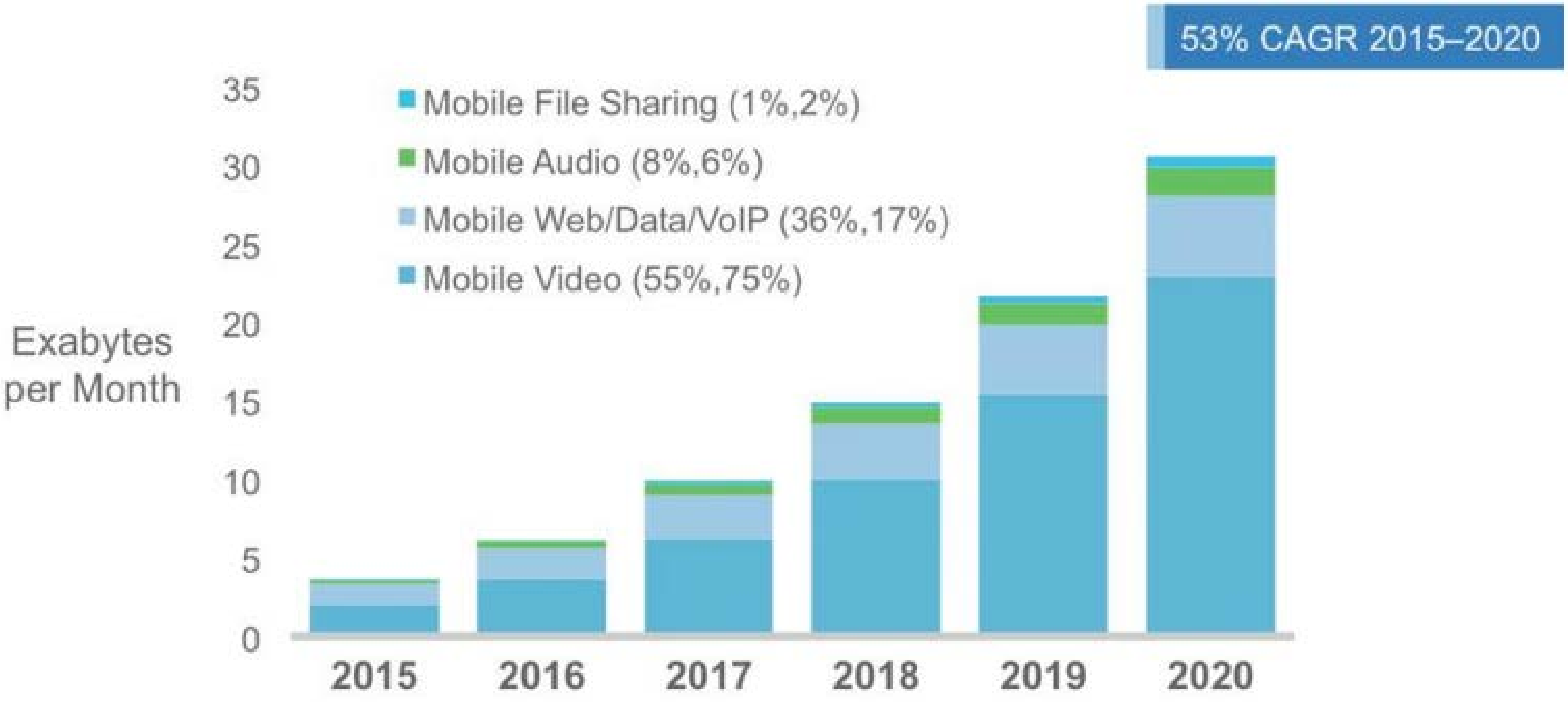

Video Streaming Systems

VR Video Streaming Systems

AR Video Streaming Systems

# Introduction

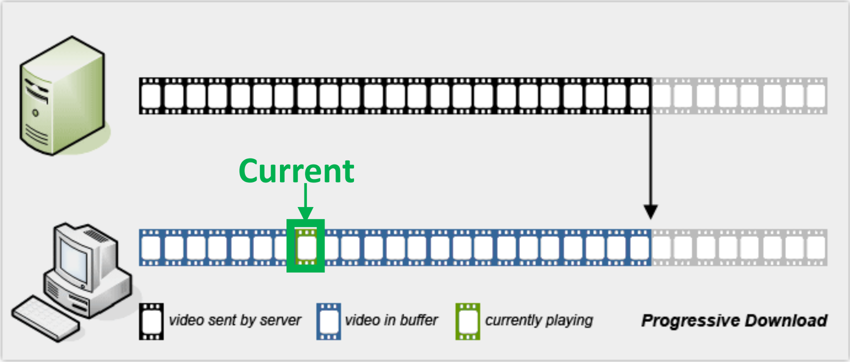
- Video traffic occupies more than 70% of mobile internet traffic



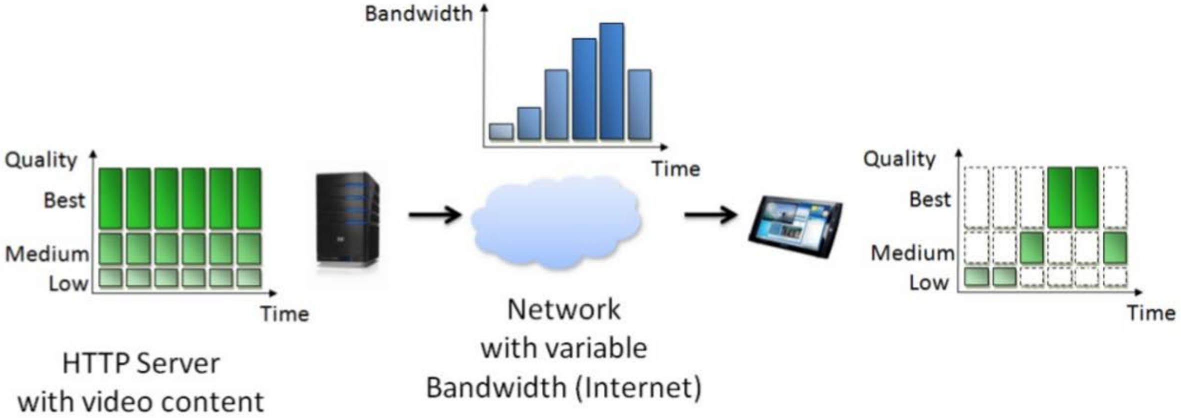

Figures in parentheses refer to 2015 and 2020 traffic share.
Source: Cisco VNI Mobile, 2016

# Video over the Internet

- Video Download
  - Playback starts after download completes
- Progressive Download
  - Playback starts before download completes

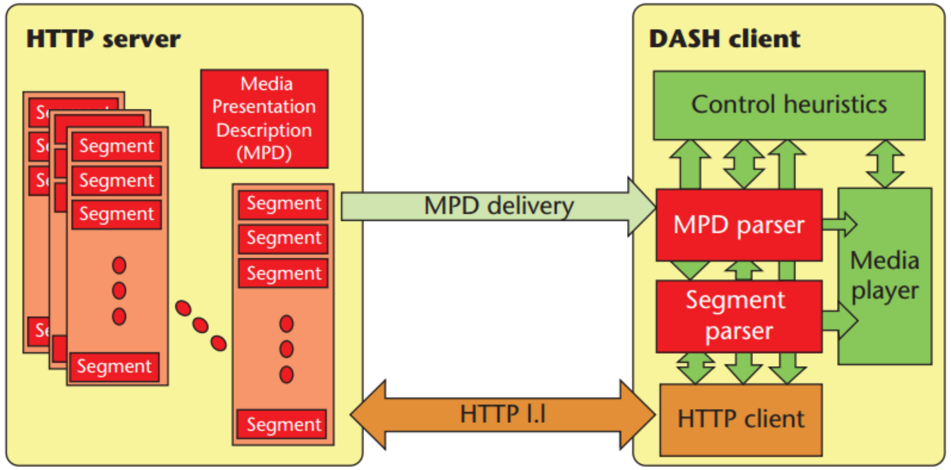



- Adaptive Bitrate Streaming (ABS)
  - Video data stored as chunks with different bitrate
  - Client request video rate

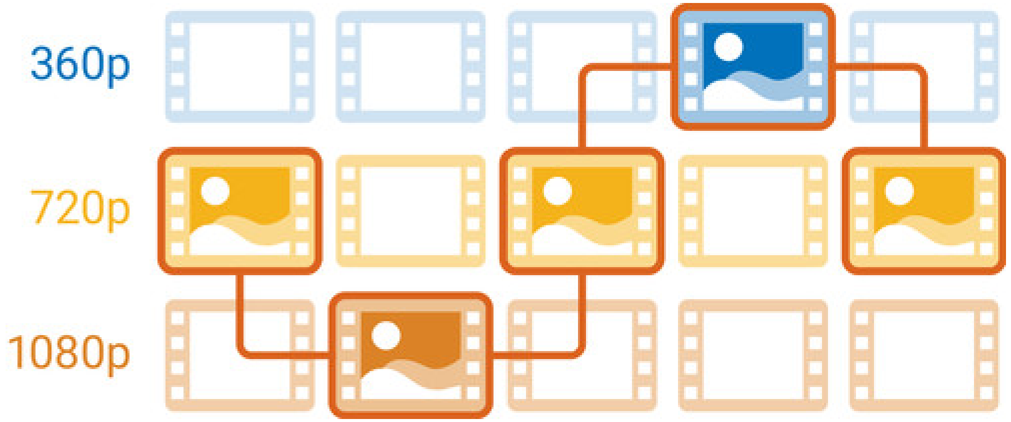


# Dynamic Adaptive Streaming on HTTP (DASH)

- Adaptive Bitrate Streaming platform
  - Small video chunks (Segment) are stored in the server
  - Server provides Media Presentation Description (MPD)
  - Client-centric rate adaptation




- Rate Adaptation
  - Network condition changes
  - Based on network condition and buffer status
  - Choose segment quality to request



# Outline


Video Streaming Systems

VR Video Streaming Systems

AR Video Streaming Systems

# VR broadcasting

cnet REVIEWS NEWS VIDEO HOW TO SMART HOME CARS DEALS DOWNLOAD

## Head trip: The experience of broadcasting an NBA game in VR

NextVR made sports history by broadcasting weekly NBA games in virtual reality. The next step is getting people to watch.

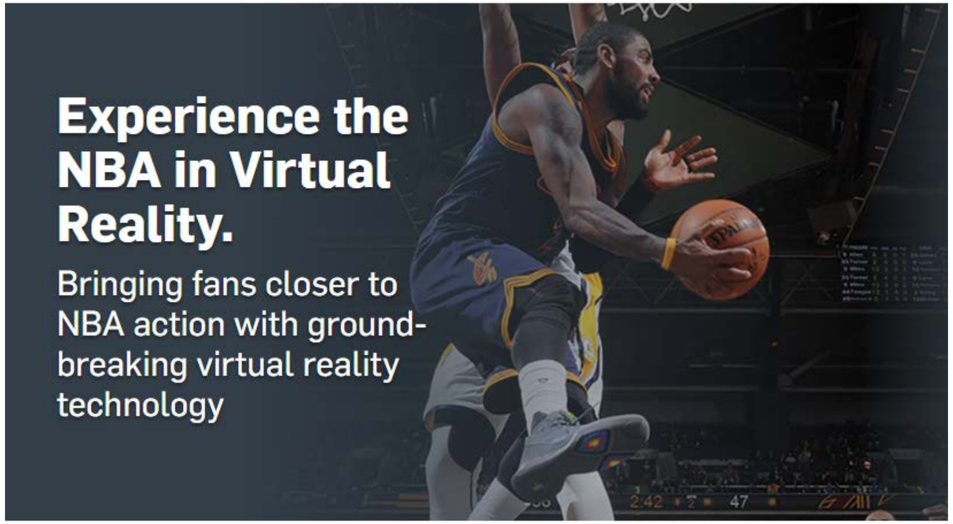


Software

by **Terry Collins**
April 27, 2017 5:00 AM PDT
@terryscollins

A technician makes last-minute preparations inside NextVR's production truck prior to the company's broadcast of a Warriors-Timberwolves game.

NBA Scores Schedule Playoffs News Video Standings Stats NBA LEAGUE PASS Store Tickets




NBA LEAGUE PASS NEXTVR

**NBA LEAGUE PASS VR Game of the Week**

Courtside seats. Wherever you are.

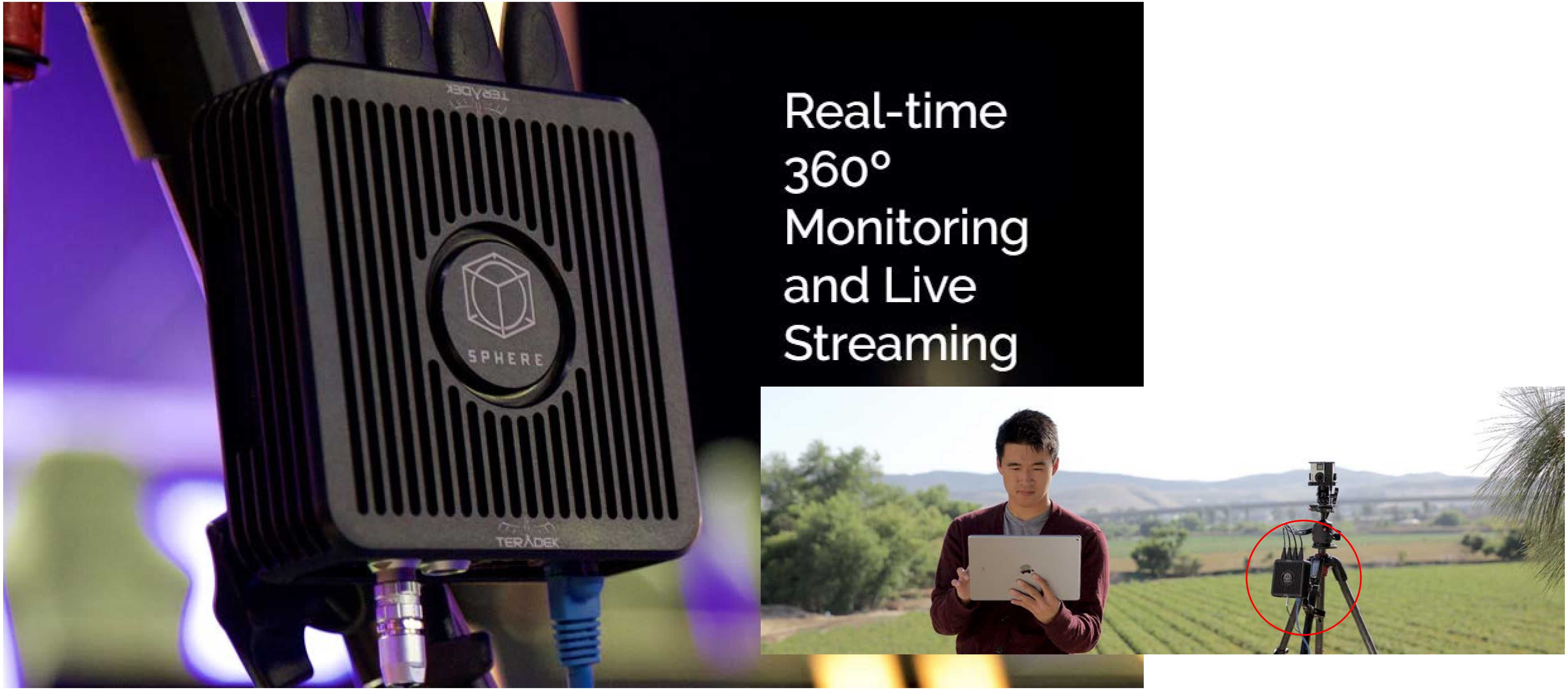
# 360-degree Cameras

Nokia OZO

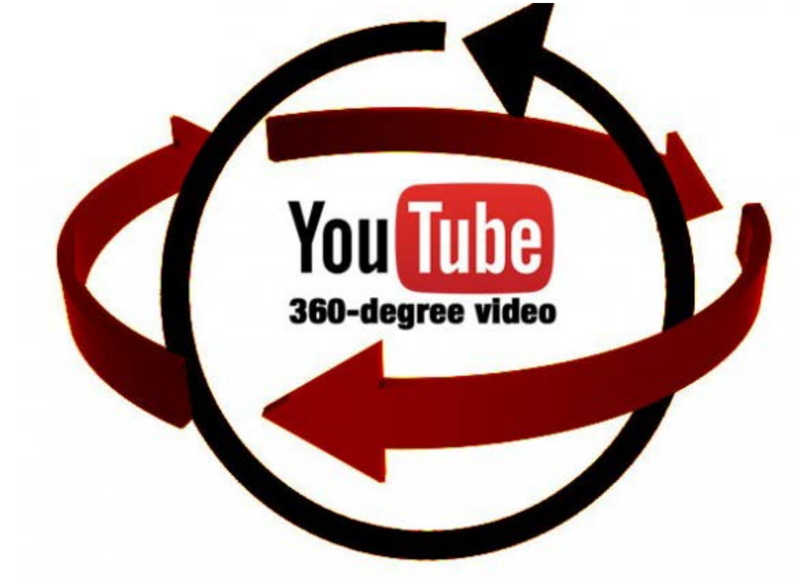
GoPro Omni


Giroptic 360cam

# Streaming platform is available



# YouTube 360-degree video





# Challenges in Networking

- Virtual Reality (VR) and Augmented Reality (AR) traffic will increase **12-fold** between 2017 and 2022 globally

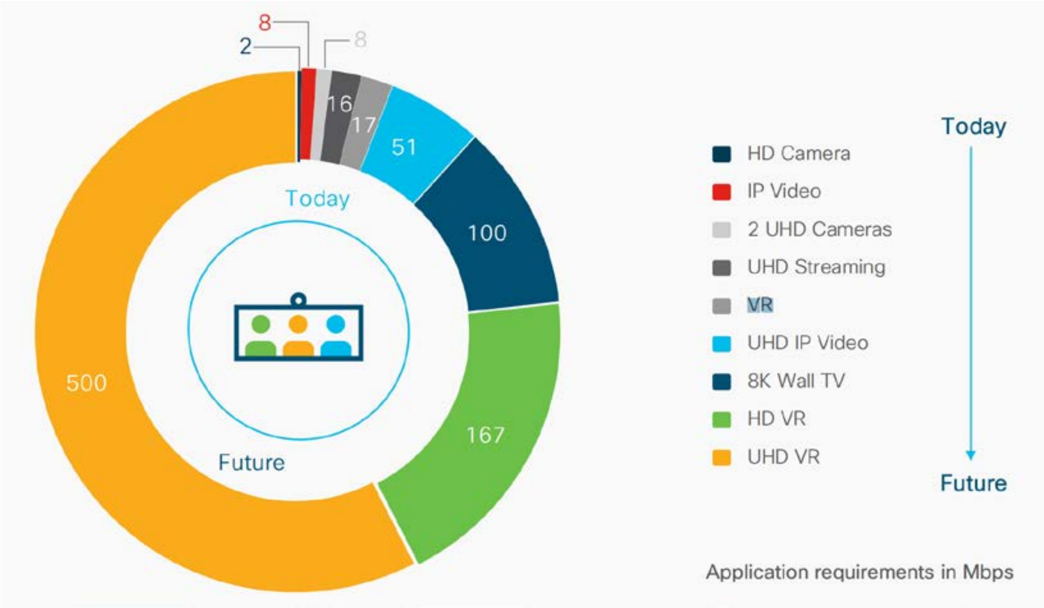



Table 6. Projected average mobile network connection speeds (in Mbps) by region and country

| | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|
| **Global** | | | | | | |
| Global speed: All handsets | 8.7 | 13.2 | 17.7 | 21.0 | 24.8 | 28.5 |
| Western Europe | 16.0 | 23.6 | 31.2 | 37.2 | 43.8 | 50.5 |
| Central and Eastern Europe | 10.1 | 12.9 | 15.7 | 19.5 | 22.8 | 26.2 |
| Middle East and Africa | 4.4 | 6.9 | 9.4 | 11.2 | 13.2 | 15.3 |
| North America | 16.3 | 21.6 | 27.0 | 31.9 | 36.9 | 42.0 |
| Asia Pacific | 10.6 | 14.3 | 18.0 | 21.7 | 25.3 | 28.8 |
| Latin America | 4.9 | 8.0 | 11.2 | 13.0 | 15.3 | 17.7 |

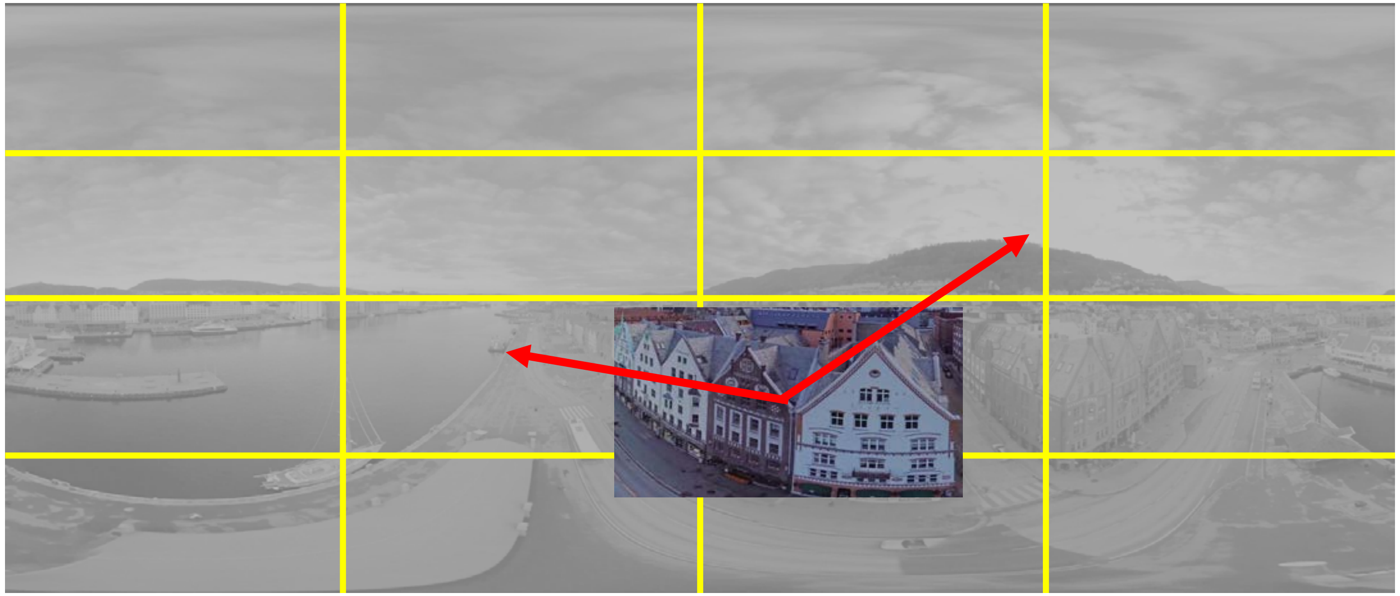
Table 7. Projected average Wi-Fi network connection speeds (in Mbps) by region and country

| Region | 2017 | 2018 | 2019 | 2020 | 2021 | 2022 |
|---|---|---|---|---|---|---|
| Global | 24.4 | 30.3 | 36.3 | 42.2 | 48.2 | 54.2 |
| Asia Pacific | 26.7 | 34.5 | 42.2 | 47.6 | 56.0 | 63.3 |
| Latin America | 9.0 | 10.6 | 12.1 | 13.8 | 15.2 | 16.8 |
| North America | 37.1 | 46.9 | 56.8 | 63.6 | 74.4 | 83.8 |
| Western Europe | 25.0 | 30.8 | 36.3 | 37.7 | 44.6 | 49.5 |
| Central and Eastern Europe | 19.5 | 22.6 | 24.1 | 27.4 | 30.1 | 32.8 |
| Middle East and Africa | 6.2 | 7.0 | 7.9 | 9.6 | 10.2 | 11.2 |

Source: Cisco VNI, 2018.

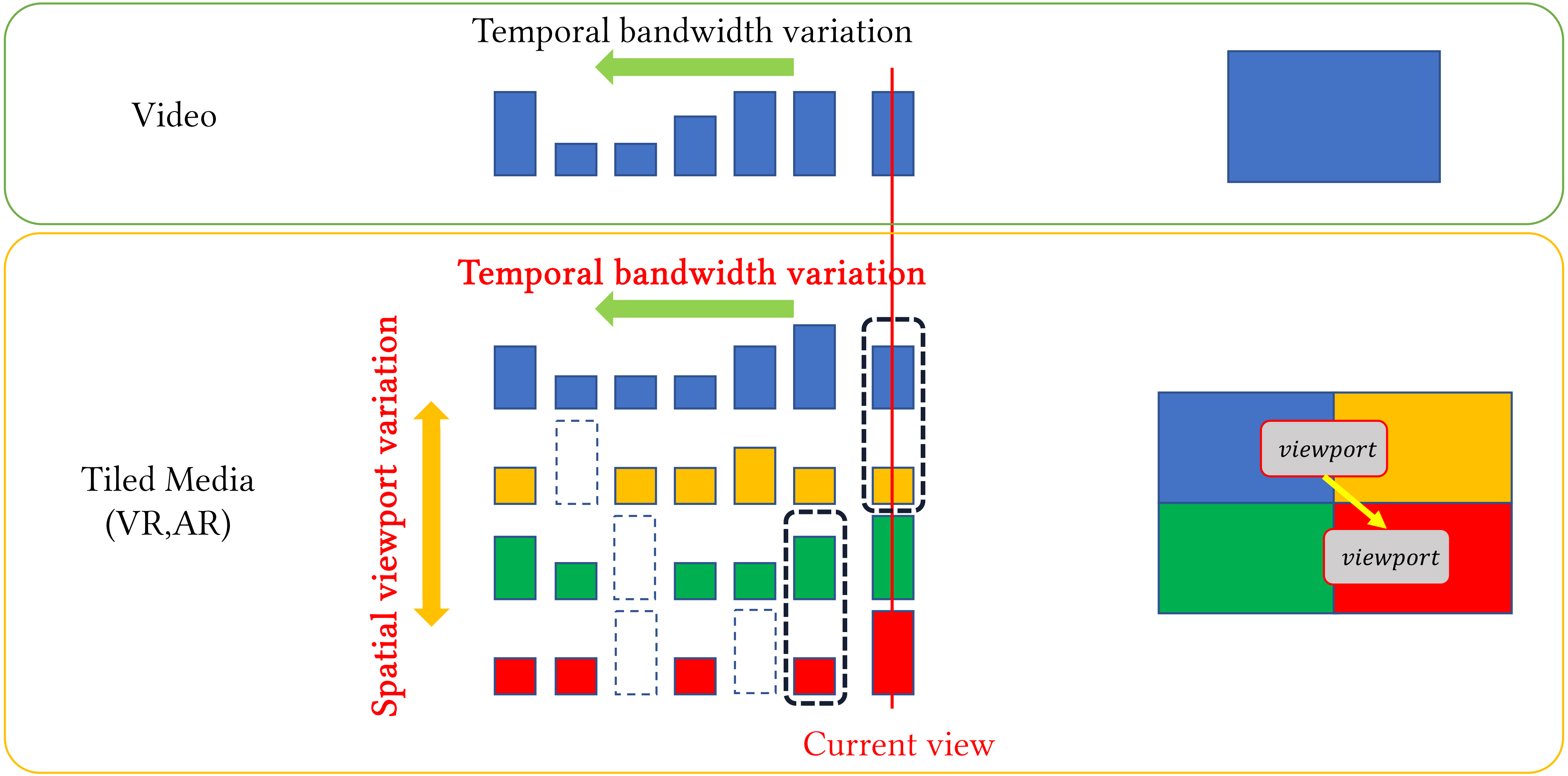
# Challenges in 360-degree video streaming

- Much higher video rate (4~6 times more than conventional videos): at least 4K
- Can save up to 80% of bandwidth by removing non-visible part of the video (typically viewport is only 20% of whole 360-degree video)
- 360-degree Video is divided into tiles representing tiled media
  - Spatial Relationship Descriptor (SRD) describes tile configuration



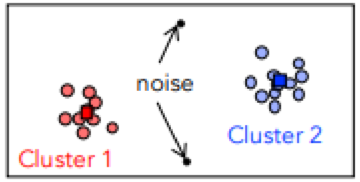
# Challenges in 360-degree video streaming

- View prediction is important to request future segments
  - Information of **Spatial and Temporal variation of view**



# Cross-user Learning based System (CLS)

- User Fixation Clustering
  - Density Based Clustering Algorithm (DBSCAN)




Fixations on one segment

| | | | |
|---|---|---|---|
| 0.0 | 0.0 | 0.0 | 0.0 |
| 0.1 | 0.1 | 0.1 | 0.0 |
| 0.1 | 0.2 | 0.1 | 0.0 |
| 0.1 | 0.1 | 0.1 | 0.0 |

Calculated from users in class 1

| | | | |
|---|---|---|---|
| 0.0 | 0.1 | 0.1 | 0.1 |
| 0.0 | 0.1 | 0.2 | 0.1 |
| 0.0 | 0.1 | 0.1 | 0.1 |
| 0.0 | 0.0 | 0.0 | 0.0 |

Calculated from users in class 2

- User Classification for Prediction
  - SVM for classification
  - Data in the window $\Delta t = [t - B_{max} - \delta, t - B_{max}]$ to train the SVM model

- **Server performs the view prediction**
- **Errors in clustering affect prediction performance**

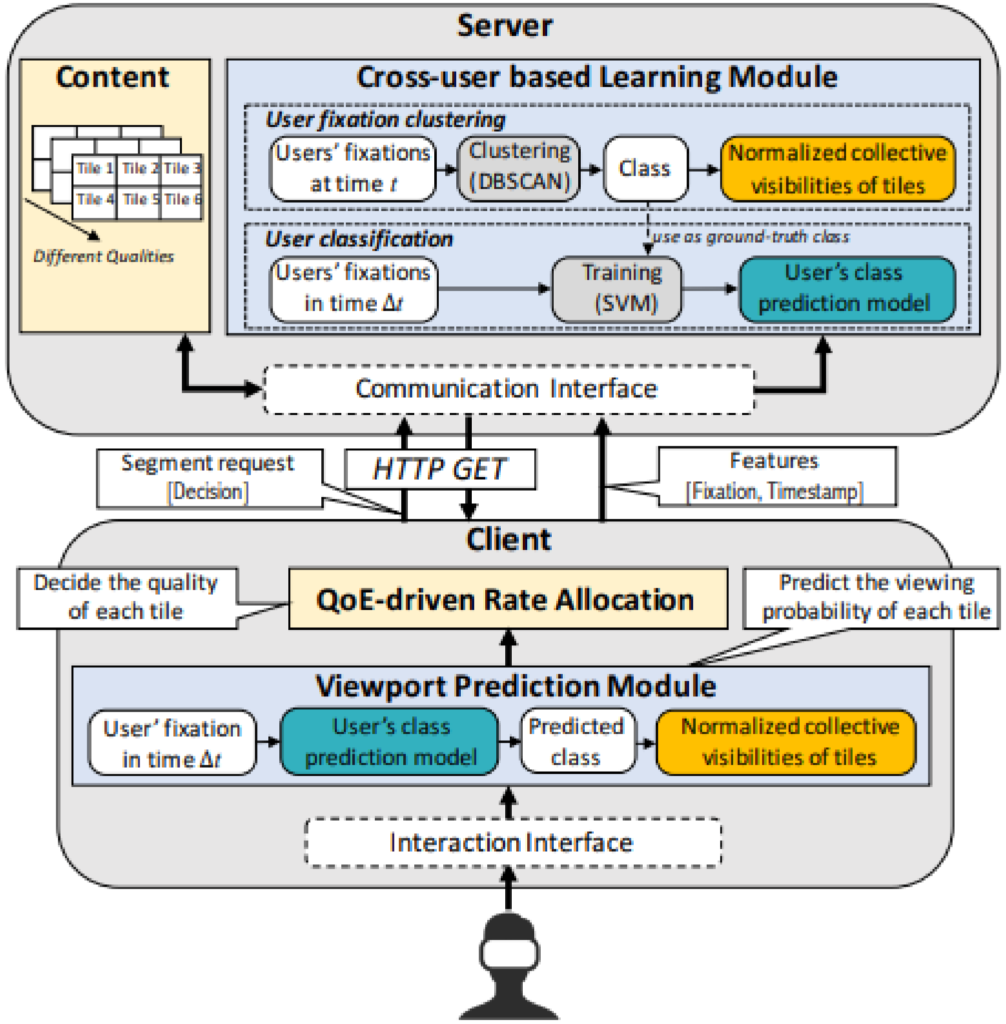



**Figure 6: Diagram of CLS architecture.**

Ref: Lan Xie, Xinggong Zhang, and Zongming Guo. 2018. CLS: A Cross-user Learning based System for Improving QoE in 360-degree Video Adaptive Streaming. In Proceedings of the 26th ACM international conference on Multimedia (MM '18). ACM, New York, NY, USA, 564-572.

# Tiled-video Rate Selection Algorithm

- Utility maximization problem

$$U = \max_{x} \sum_{t=1}^{T} \sum_{q=1}^{M} x_{t,q} u_{t,q} P_t$$

$$\text{subject to} \sum_{t=1}^{T} \sum_{q=1}^{M} x_{t,q} R_{t,q} \leq BW$$

$$x_{t,q} \in \{0,1\}$$

$x_{t,q}$ = selection parameter
$u_{t,q}$ = utility
$P_{t,q}$ = View probability of a tile
$R_{t,q}$ = bitrate
$BW$ = available throughput
$t$ = tile index
$q$ = quality index
$V_t$ = quality index selected for a tile $t$

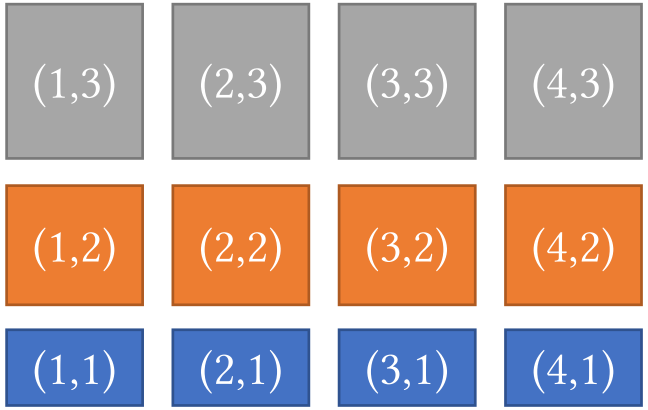
Patrice R. Alface, et al, "Interactive Omnidirectional Video Delivery:A Bandwidth-Effective Approach", Bell Labs Technical Journal (2012)

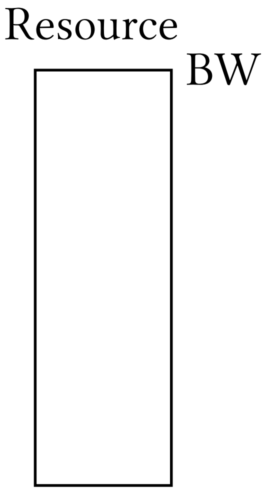
# Tiled-video Rate Selection Algorithm

1. Sort all ratio $(t, q)$ pairs in increasing order of their expected **utility-over-cost ratio** $\frac{\tilde{u}_{t,q} P_t}{C_{t,q}}$
2. Set $current\_BW = 0$
3. For all $t$, mark $Q_t$ as “not selected”
4. Do while $current\_BW <= BW$
   a. Mark $Q_t$ as “selected at quality level $q$”
   b. Update $current\_BW = current\_BW + C_{t,q}$

$$C_{t,q} = \begin{cases} R_{t,q} - R_{t,q-1}, & q \geq 2 \\ R_{t,q}, & q = 1 \end{cases}$$

$$\tilde{u}_{t,q} = \begin{cases} u_{t,q} - u_{t,q-1}, & q \geq 2 \\ u_{t,q}, & q = 1 \end{cases}$$




Patrice R. Alface, et al, “Interactive Omnidirectional Video Delivery:A Bandwidth-Effective Approach”, Bell Labs Technical Journal (2012)

# Outline


Video Streaming Systems

VR Video Streaming Systems

AR Video Streaming Systems

# Hologram Streaming Systems

- Goal
  - Streaming the volumetric media over the internet
  - Augmented Reality Applications: Holo-presence
- Challenges
  - Network variation
  - Large data/small pipe
  - User interaction

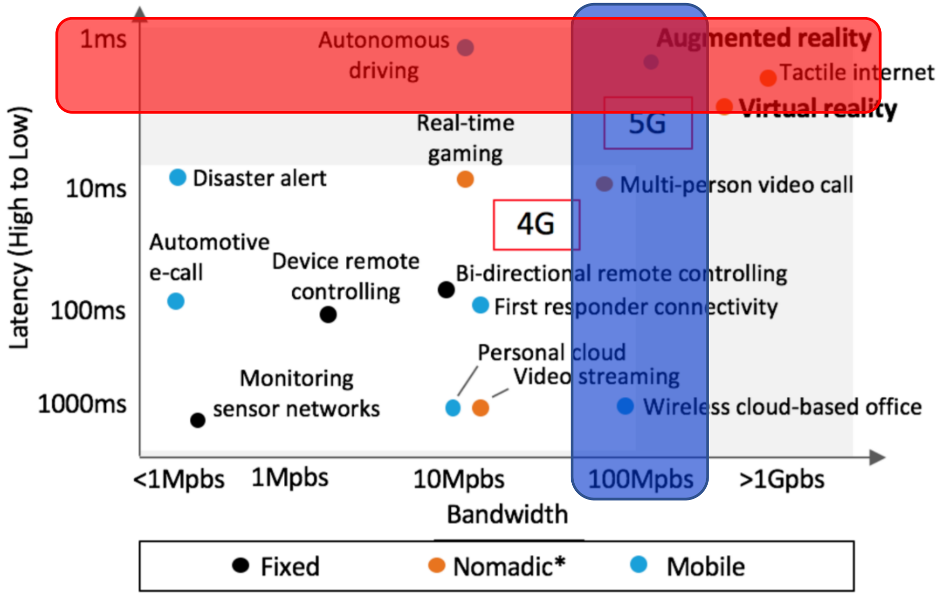
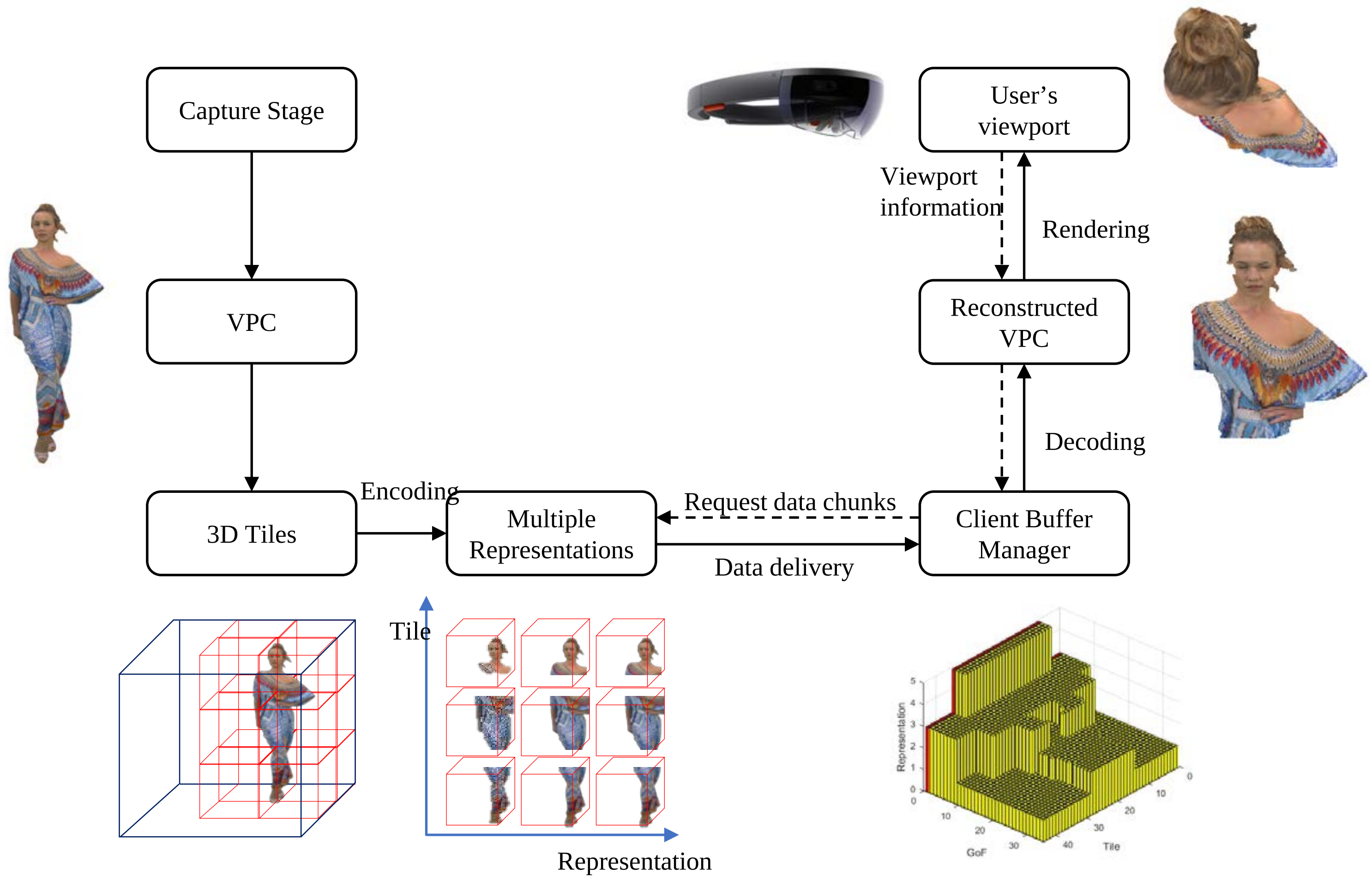



Figure 1: Projected network demands of emerging technologies[35, 43]
*Nomadic: Connectivity is wireless but not ubiquitous - limited to disjoined hotspots


# Volumetric Media Streaming

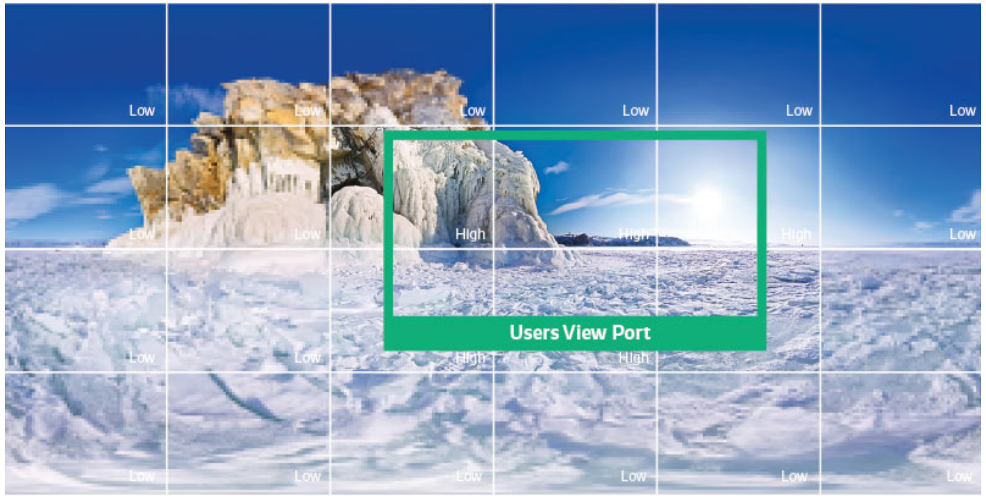
- Voxelized Point Cloud (VPC) created in Capturing Stage
- 3D tiles encoded with multiple representations
- Client Buffer Manager requests data chunks based on
  - User's Viewport
  - Network condition
  - Buffer status



# User Adaptation for VR/AR

- Users only see part of video
  - Only selected parts (tiles) are transmitted to save bitrate
  - Tiles can have different representations (quality)

Tiled 360-degree video

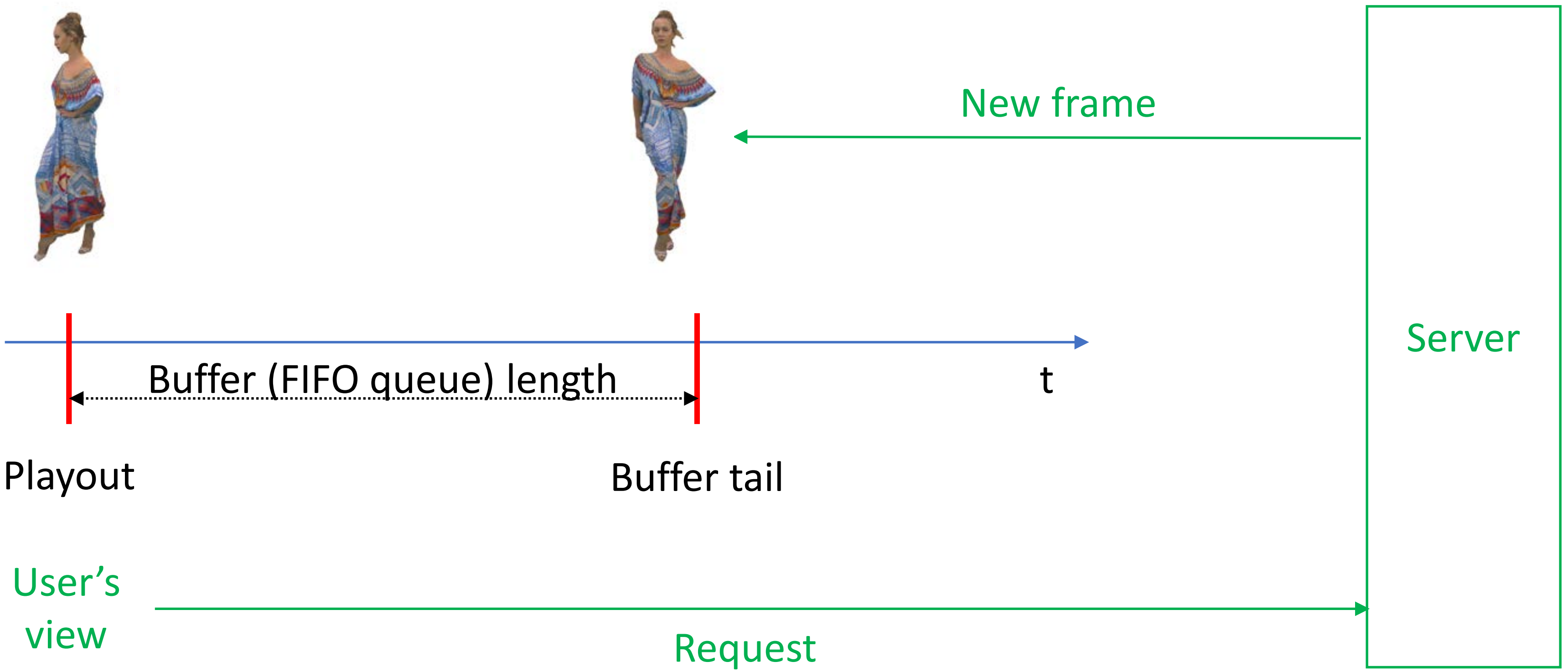



User's view

Transmitted tiles

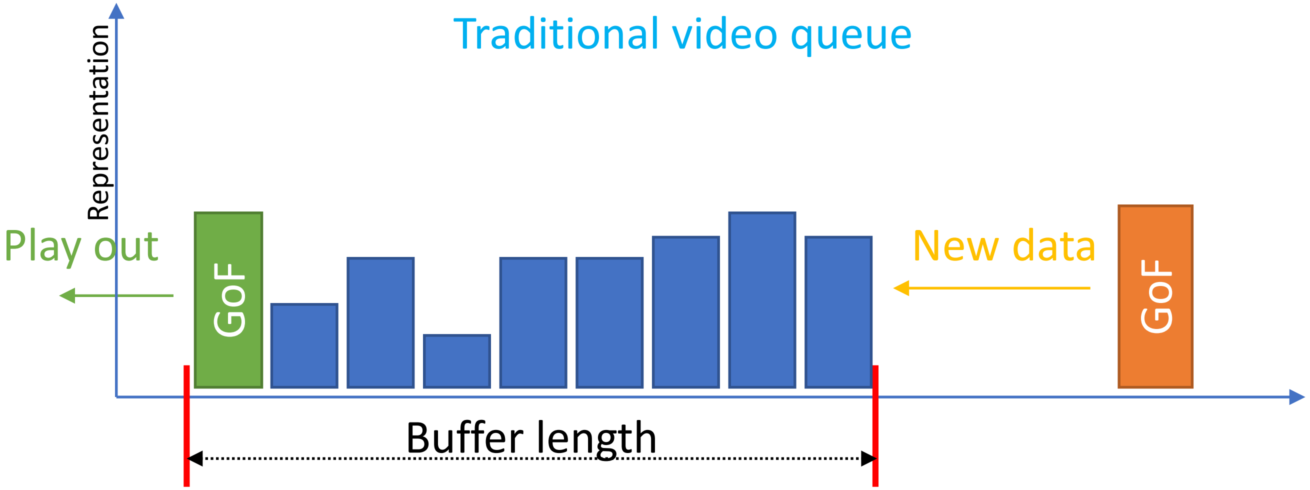
# View Prediction

- Delay is gap between playout frame and requesting frame

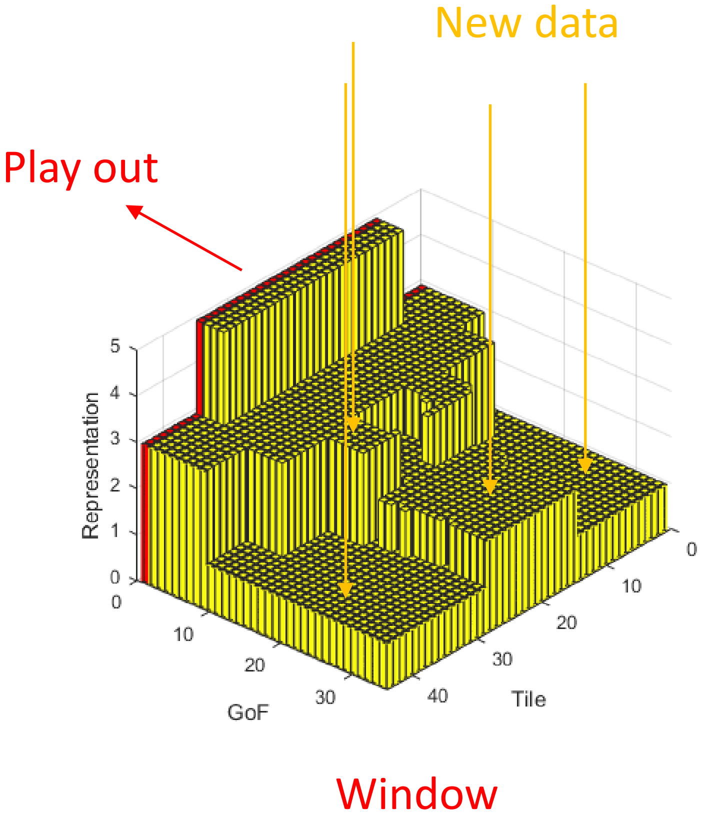


# Window based algorithm

- Our proposed algorithm for volumetric media streaming
  - Uses Window (not a queue)
  - Lower delay with robust buffer size
  - Supports multiple assets and multiple tiles




GoF : Group of Frames



# Utility

- Mathematical expression of expected visual quality
  - Quality : determined by bitrates ($b_i$)
  - Level of Detail (LoD)
  - Visibility ($P_i$) : user's view

$$U_k(n_k) = u(B_{n_k}) \times \max_{v \in V}\{LOD(n_k, v) \times P_k(v)\},$$

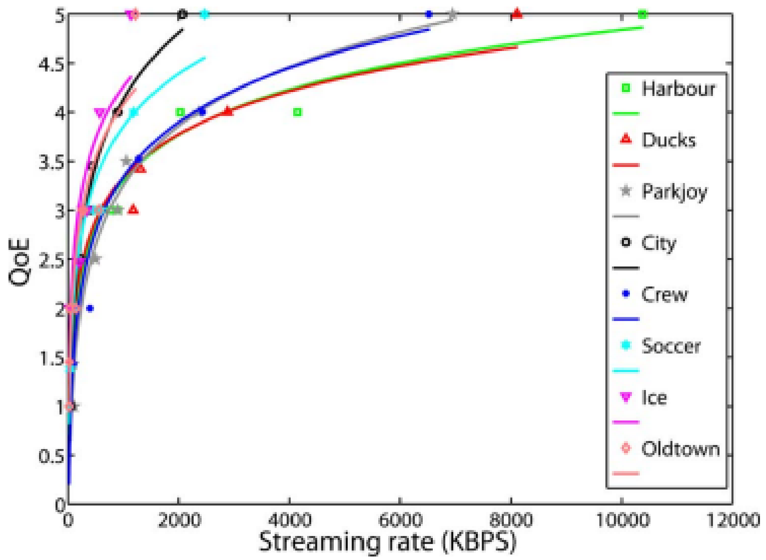
- Utility maximization algorithm
  - Select the best bitrates for tiles of GoFs and Assets to maximize utility within given bandwidth

# Quality

- Logarithmic law
  - Better quality with higher bitrate
  - Quality saturates with higher bitrate
  - Mean Opinion Score (MOS) usually follow this model

$$u(B) = \begin{cases} \alpha \log \beta B, & B > 0 \\ 0, & B = 0 \end{cases},$$

$\alpha, \beta$: constants
$b_i$:bitrates allocated for $i$



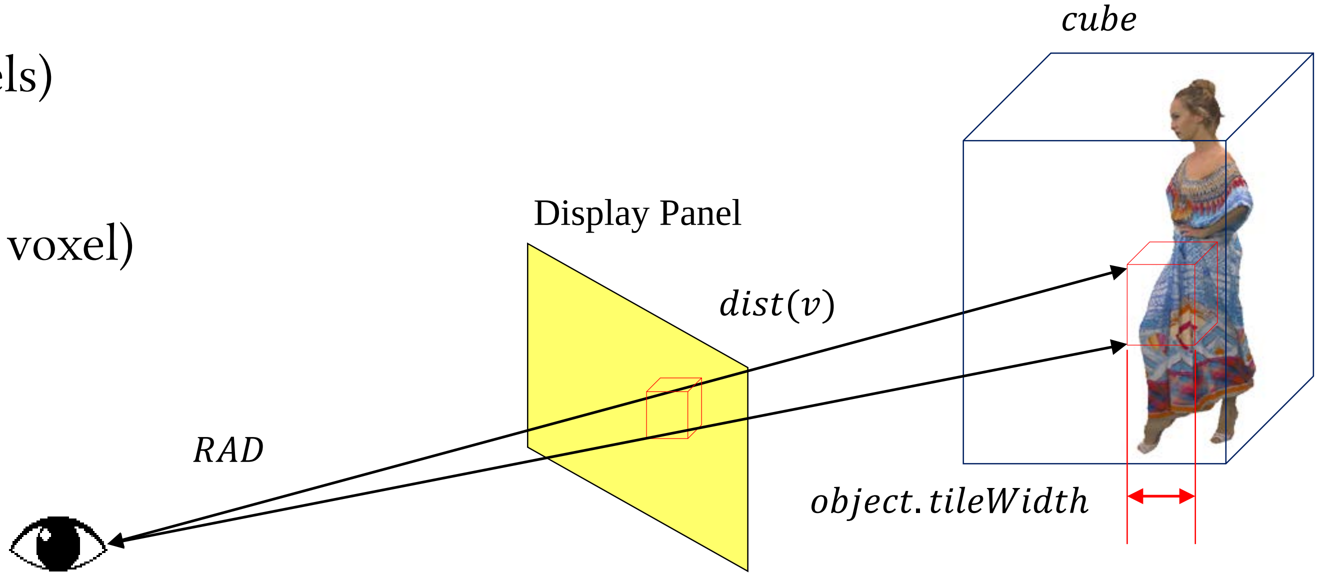
# Level of Details (LoD)

- Far from camera (single pixel contains many voxels)

→ utility is limited by resolution of display

- Close to camera (multiple pixels represents single voxel)

→ utility is limited by asset LoD (size of voxel)

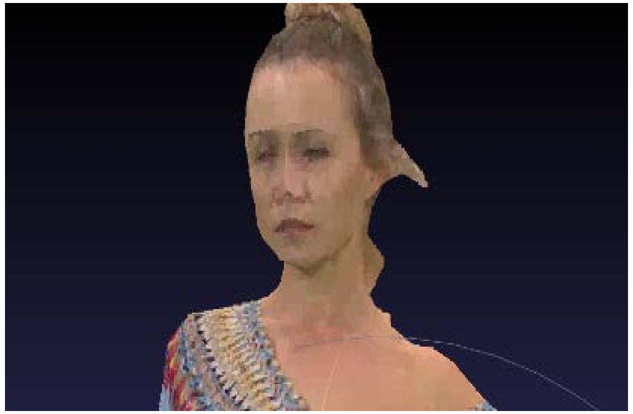



$$LOD_k(n,v) = [RAD_k(v) \times \min\{VPR_k(n,v), PPR_k(v)\}]^2$$

$$RAD_k(v) = \frac{objec.tileWidth \times object.cubeToObjectScale}{dist(v)}.$$

$$VPR_k(n,v)$$
$$= \frac{object.representation[n].width \times dist(v)}{object.\max Width \times object.cubeToObjectScale}.$$

$$PPR_k(v) = \frac{display.horzPixels}{veiw(v).\,frustum.horzFOV}$$

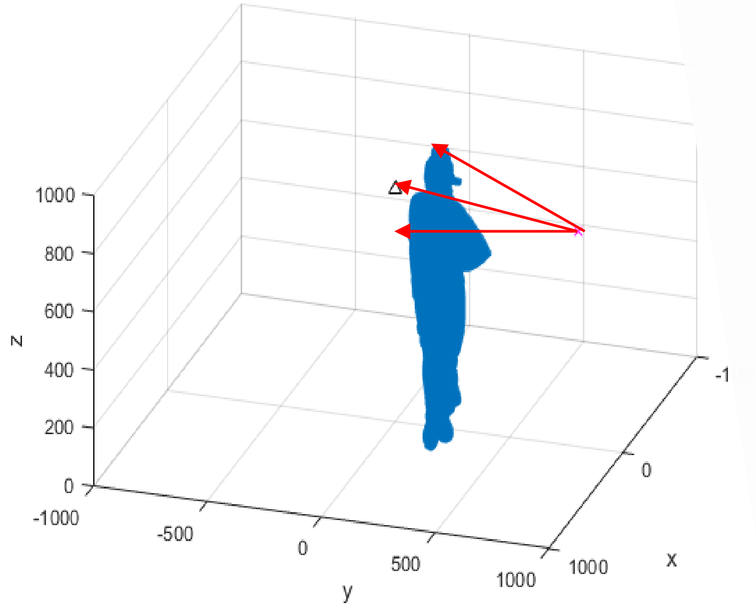


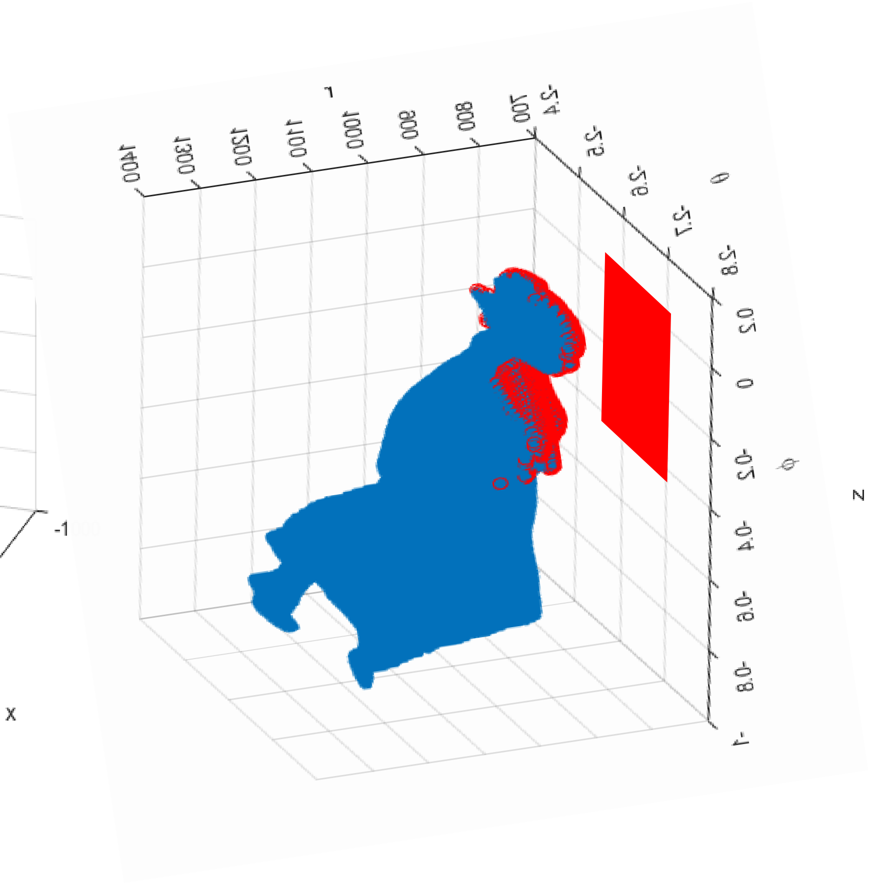
VPR<PPR

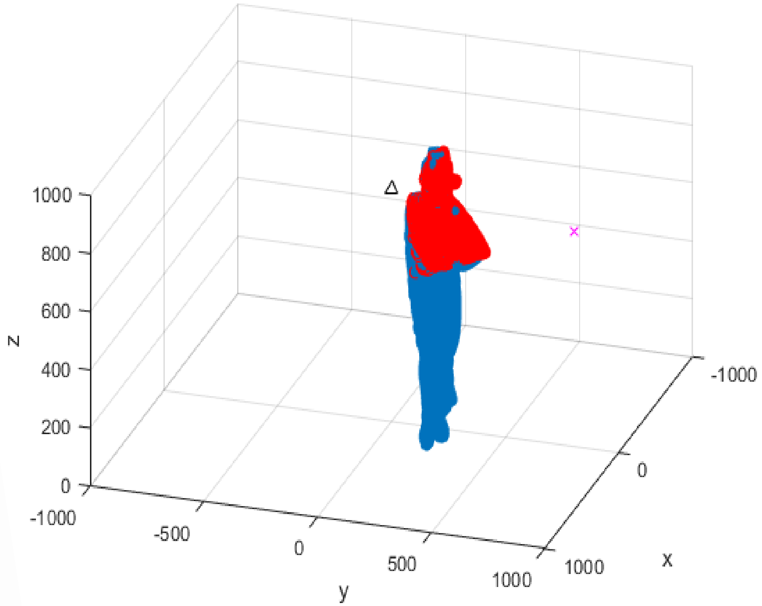
VPR>PPR

# Visibility

- Cartesian coordinate → Spherical coordinate
- Find the closest voxel in a FoV, for hidden tile removal



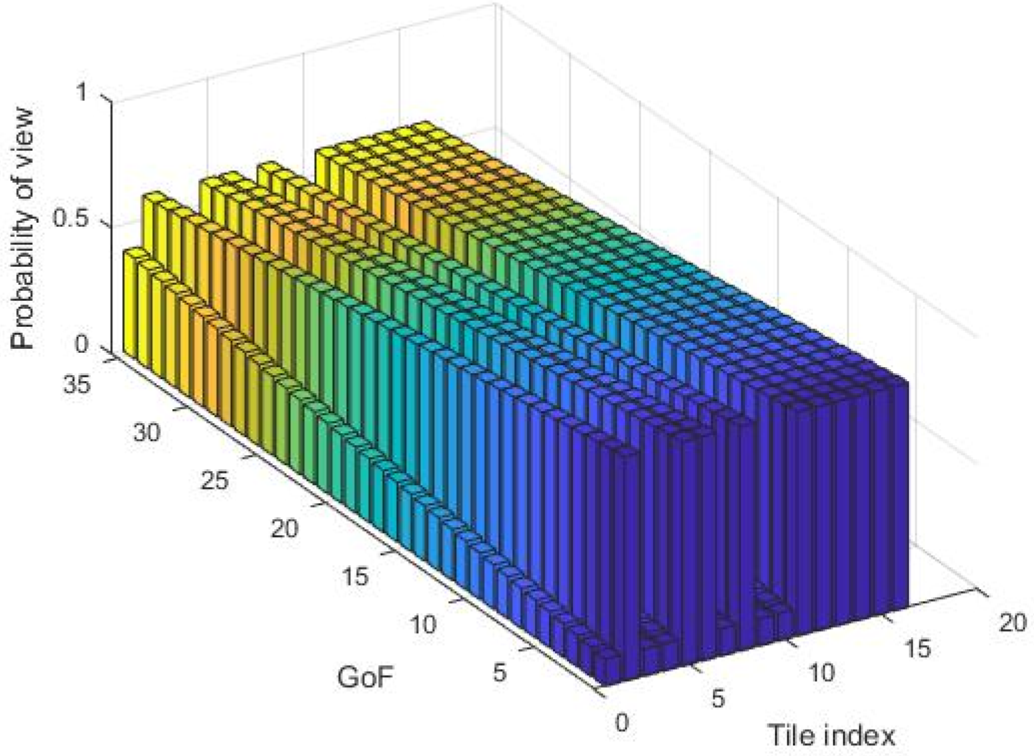


# Visibility and prediction

- Prediction model
  - Current view at GoF = 0 in a window
  - Prediction is not accurate as it is far from GoF = 0

$$P_{err} = 0.1 + 0.3 * \min\left\{1, \frac{GoF}{35}\right\}$$

$$P_i \ = \ \text{in the view?} \ (1 - P_{err}) : \ P_{err}$$

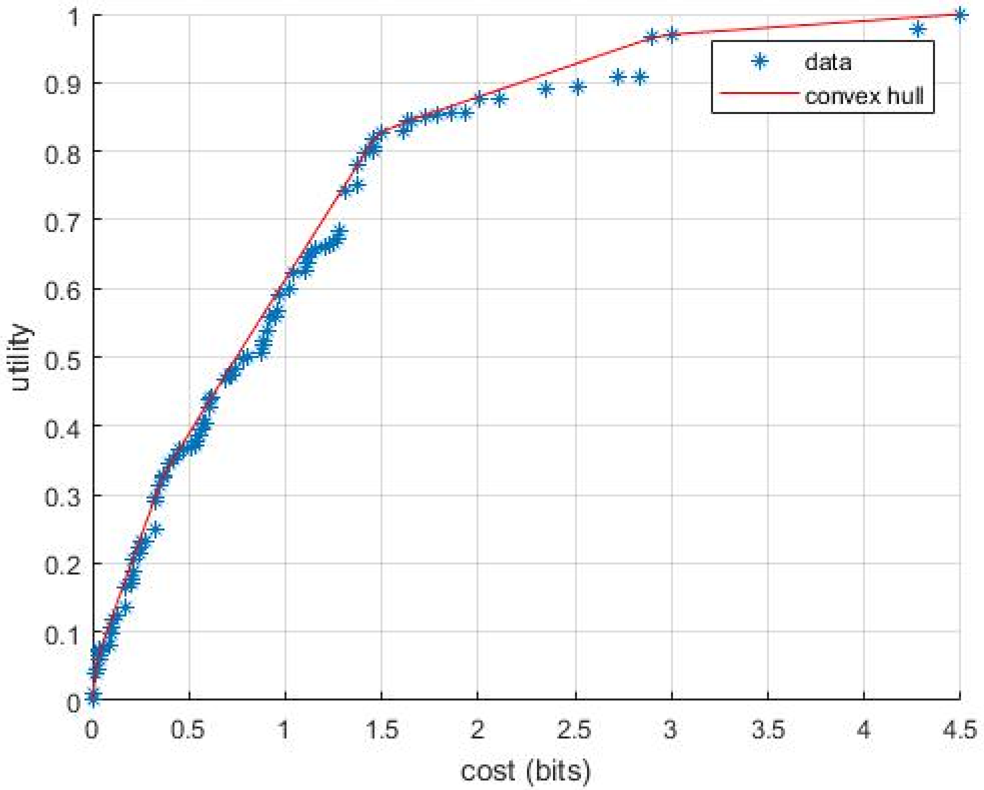


# Utility Maximization Problem

- Utility maximization with bandwidth constraint

$$\max_{b_1,\ldots,b_M} \sum_{i}^{M} u(b_i) \qquad \text{subject to} \sum_{i}^{M} b_i \leq R$$

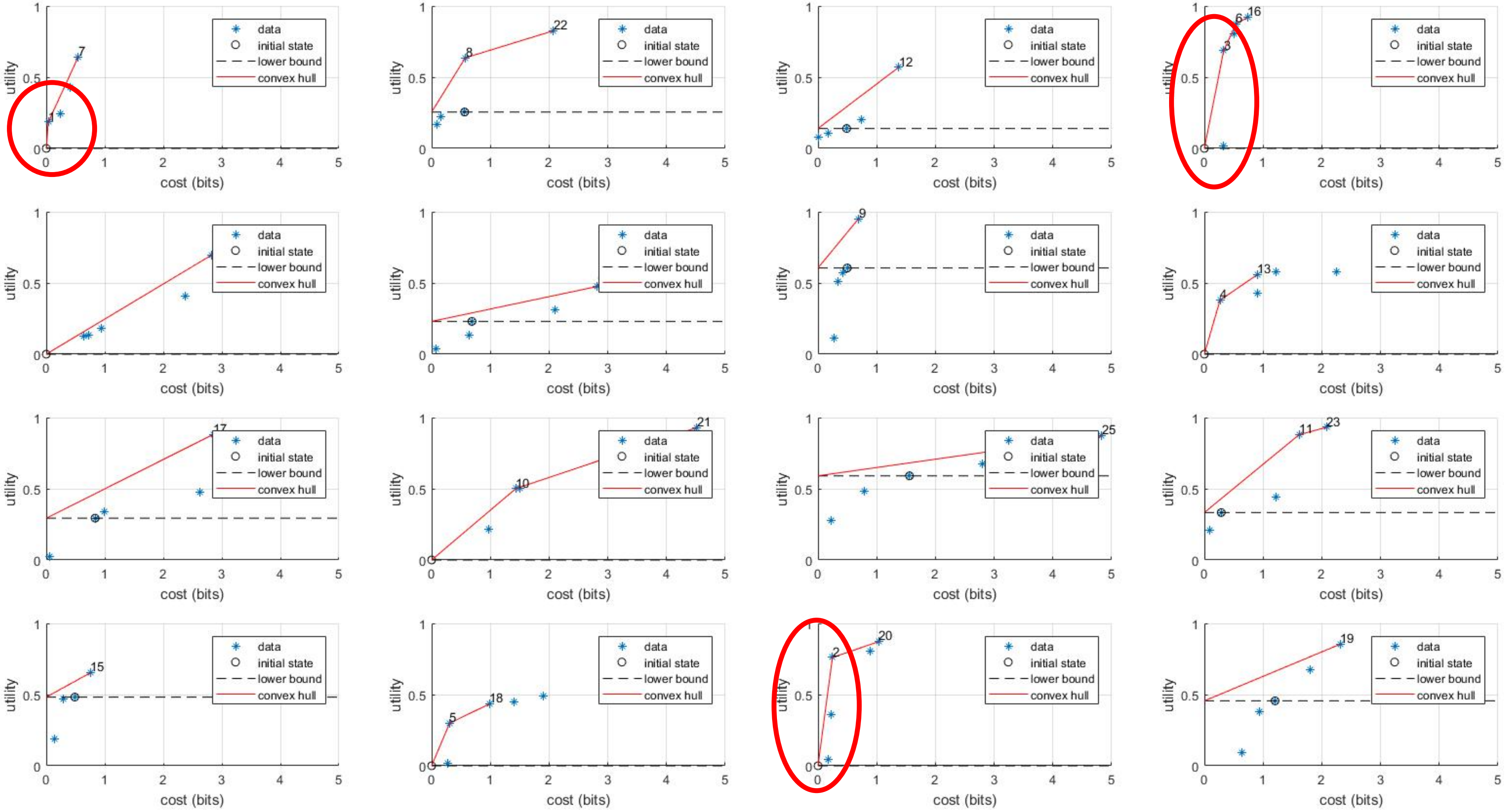
$$\text{where } u(b_i) = \alpha \log(\beta b_i) \cdot LoD \ \cdot P_i$$
$$M = \text{numAssets} * \text{numTiles} * \text{numGoF}$$




- Greedy Algorithm
  - Sort with efficiency
  - Allocate bitrates until whole resource used

# Greedy rate selection algorithm

$M = 16$



# Representations

- MPEG : 5 representations

| Test Dataset | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Queen | 3 | 5* | 15* | 30 | 55* |
| Loot | 3.5* | 5 | 8* | 16 | 27* |
| Redandblack | 3.5* | 6 | 9* | 18 | 30* |
| Soldier | 3.5* | 6 | 11* | 20 | 37.1* |
| Longdress | 3.9* | 6 | 13* | 27 | 42.7* |

# Representations

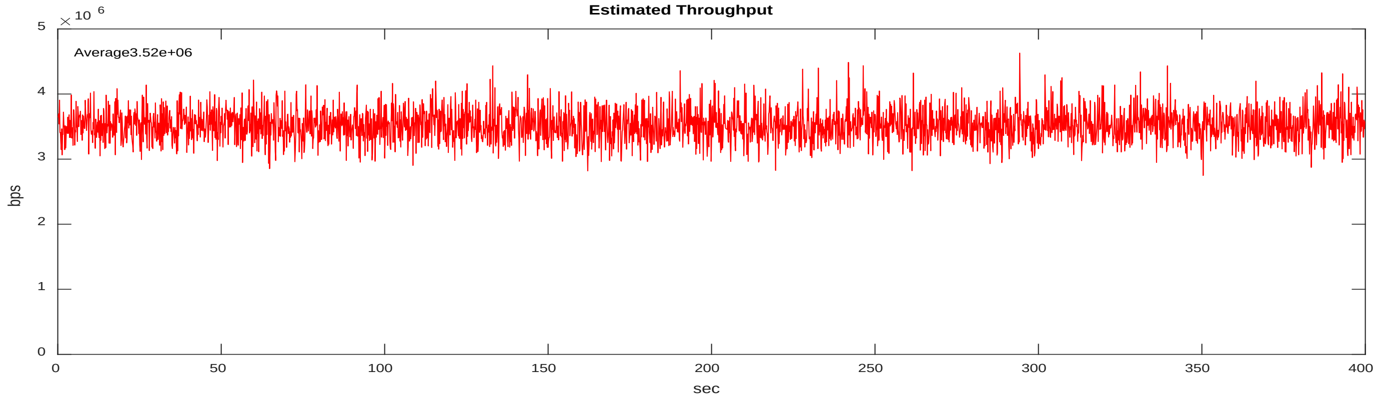
- MPEG : 5 representations

| Test Dataset | 1 | 2 | 3 | 4 | 5 |
| --- | --- | --- | --- | --- | --- |
| Queen | 3 | 5* | 15* | 30 | 55* |
| Loot | 3.5* | 5 | 8* | 16 | 27* |
| Redandblack | 3.5* | 6 | 9* | 18 | 30* |
| Soldier | 3.5* | 6 | 11* | 20 | 37.1* |
| Longdress | 3.9* | 6 | 13* | 27 | 42.7* |

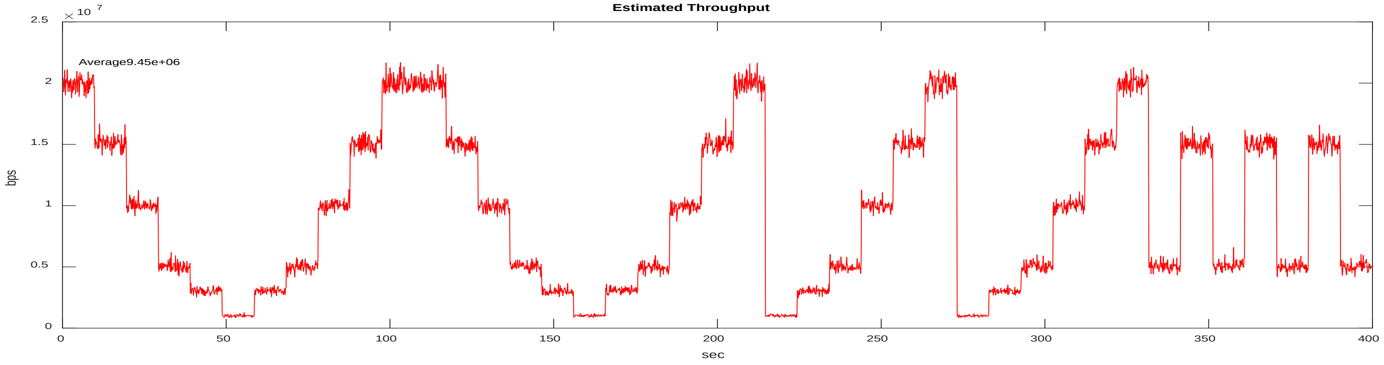
# Network model used in simulations

- Packet arrivals can be modeled as random Poisson process in a stable network condition
- Change network throughput variation manually

Scenario 1 : Stable network



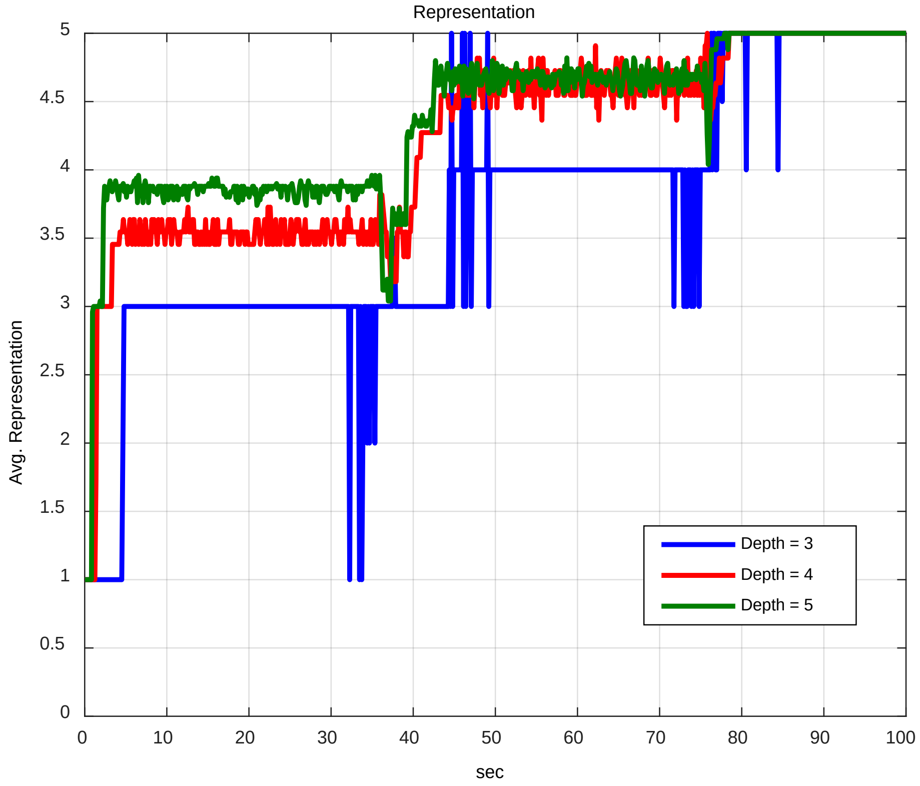

Scenario 2: Varying network



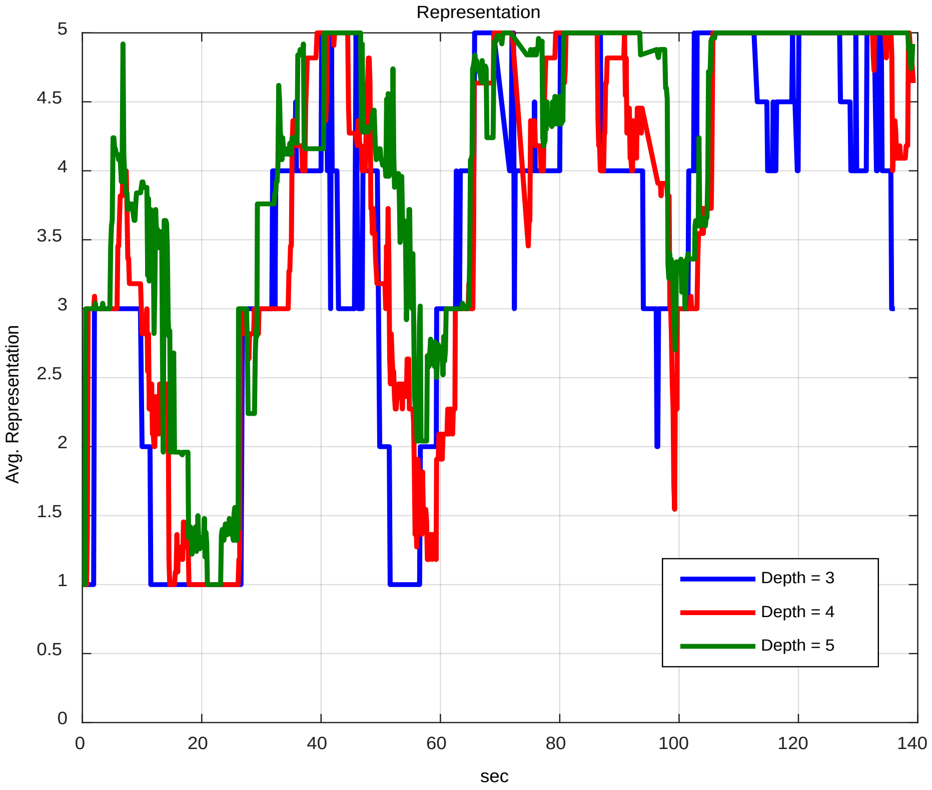
# Simulation Results (1/2)

- Average Quality of user's view with different sizes of tiles

Stable network




Varying network



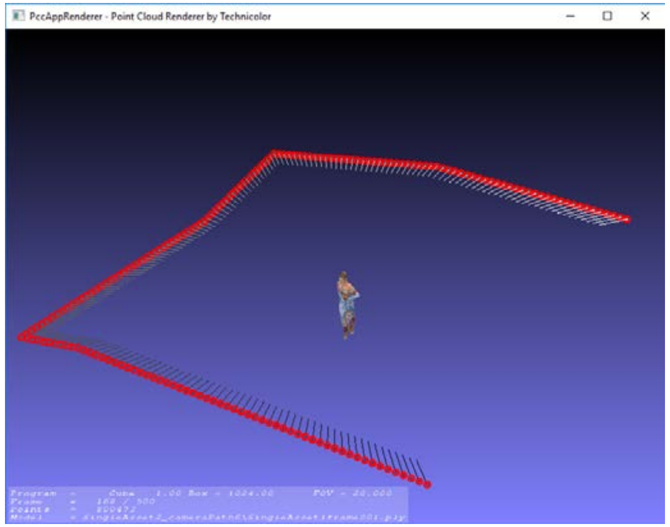
# Simulation Results (2/2)

- Videos
  - CameraPath1
    - Buffer status, utility, representations in a user's view
  - CameraPath2
    - Buffer status, utility, representations in a user's view

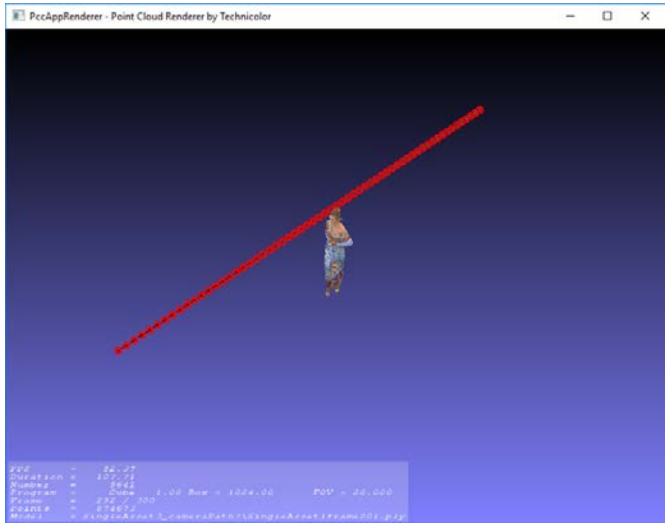
- User's view
  - CameraPath1
  - CameraPath2

- Multiple Assets
  - CameraPath1

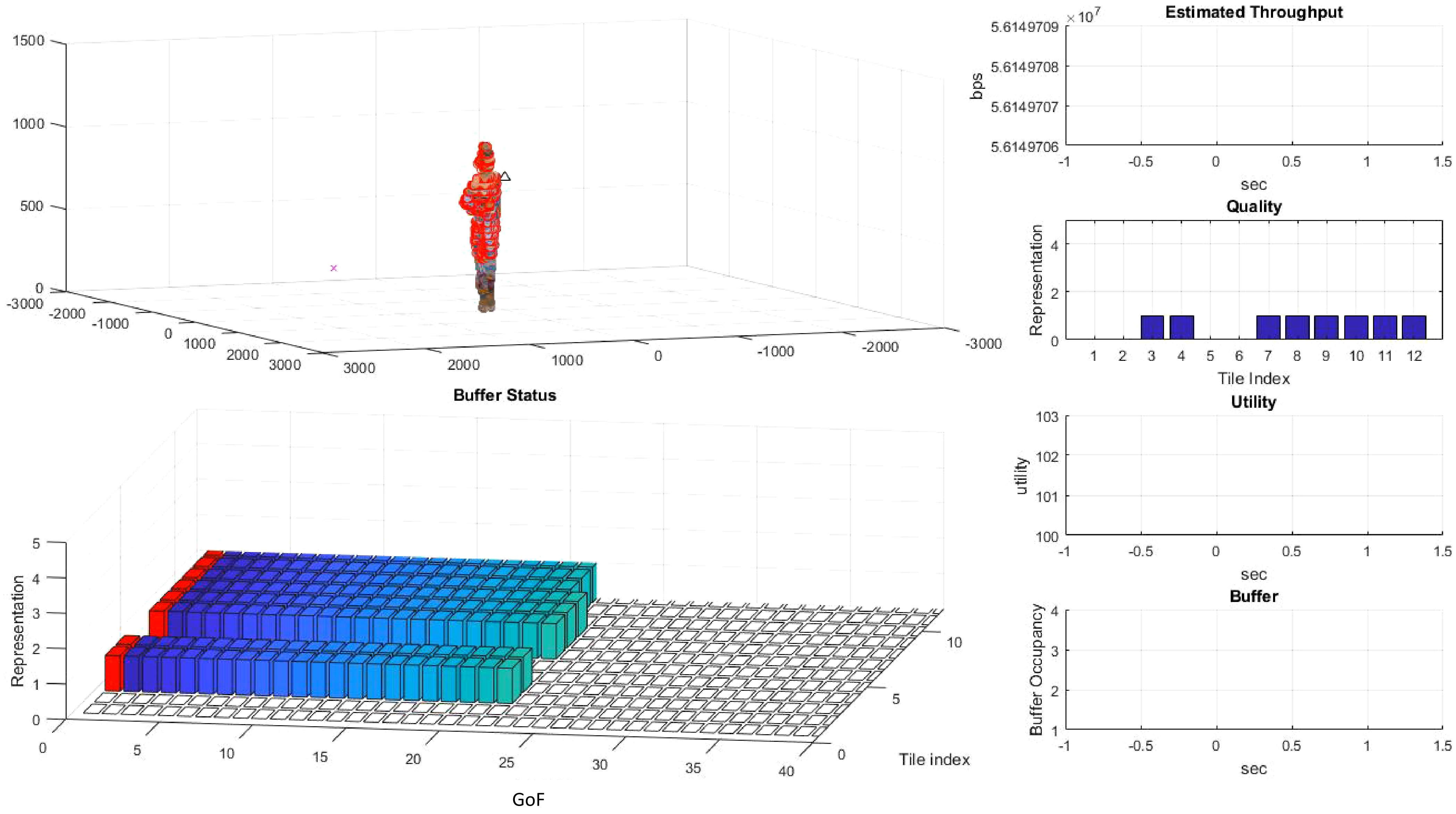
CameraPath1



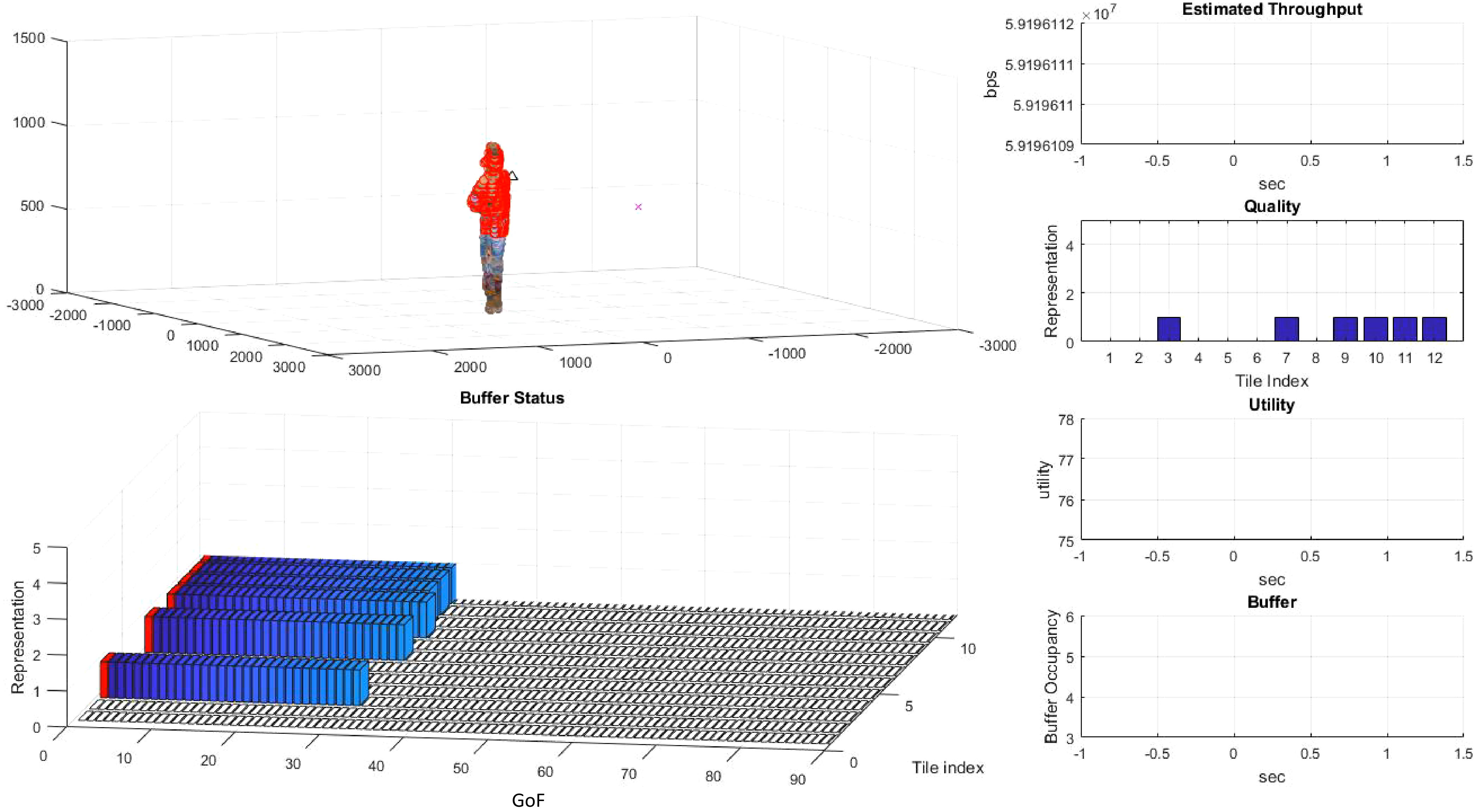

CameraPath2



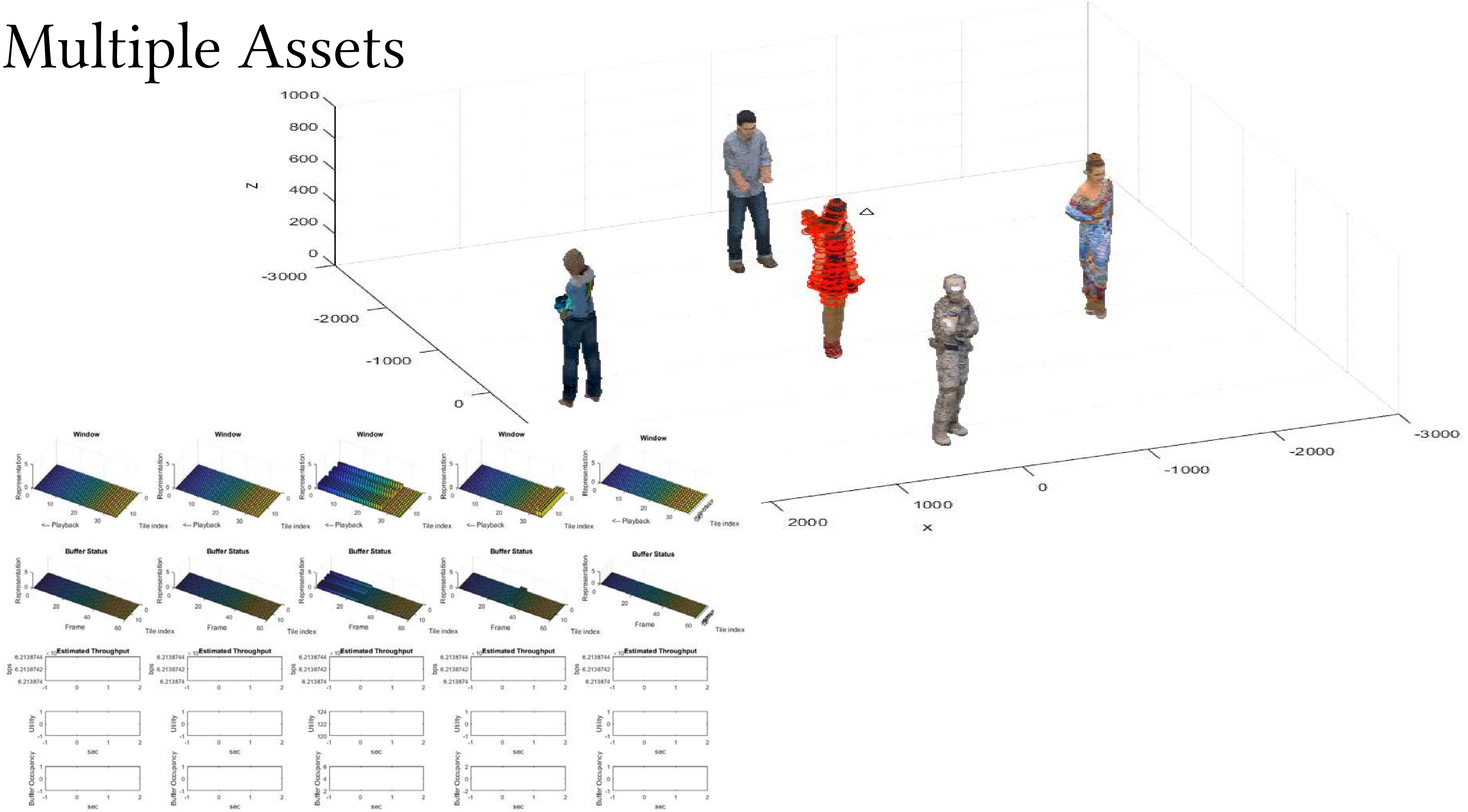
# Camera Path 1



# Camera Path 2



# Multiple Assets



# Q&A

# Thanks