ĐẠI HỌC
BÁCH KHOA HÀ NỘI
HANOI UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Optimizing Immersive Media Streaming: From 360° Video To Volumetric Experiences

Presenter: **Assoc.Prof.Truong Thu Huong**

Affiliation: School of Electrical and Electronic Engineering
Hanoi University of Science and Technology

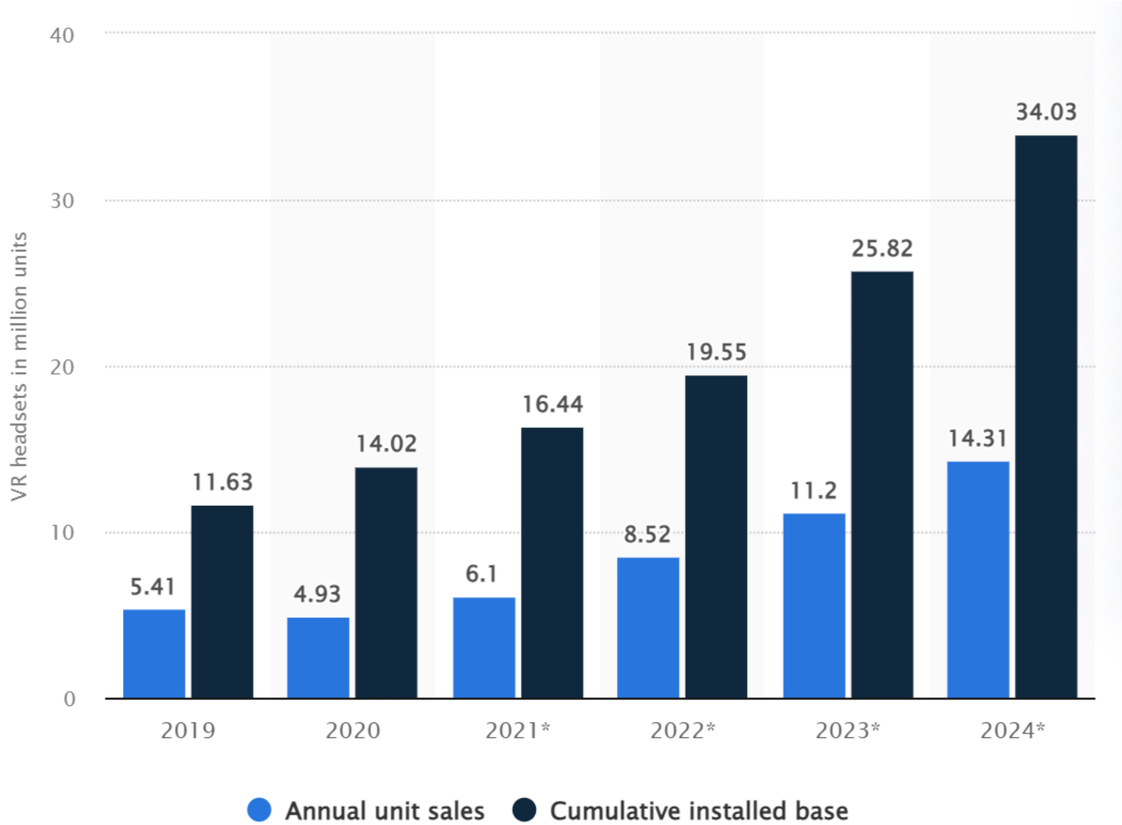
ONE LOVE. ONE FUTURE.

FROM NEEDS AND PROBLEMS TO MOTIVATIONS

ONE LOVE. ONE FUTURE.

Extended Reality Technologies

XR includes VR, AR, and MR, providing immersive experiences for natural interaction.

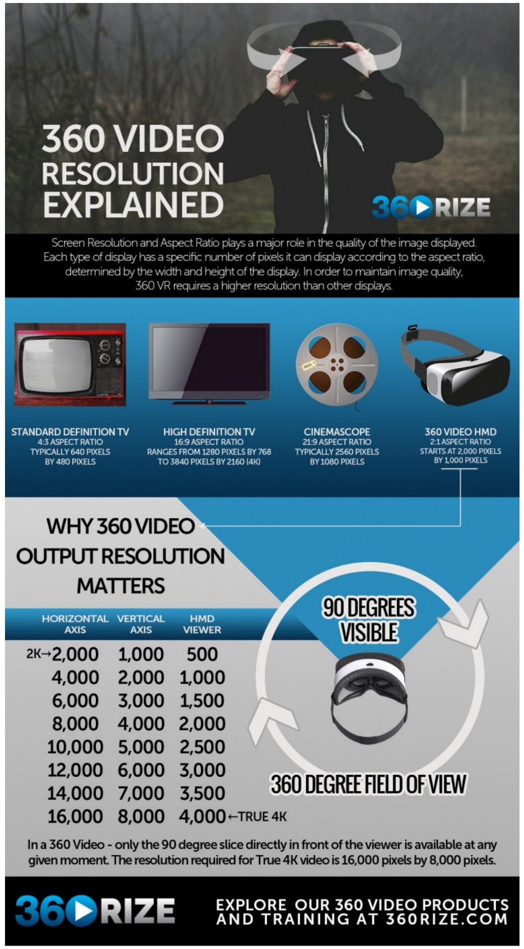


Motivation for XR Media

Challenges of immersive service Delivery

Massive Bandwidth Requirements

- 360° video captures **the entire environment**, even though users view only part of it at any moment.
- To avoid visible pixelation in head-mounted displays, **4K–8K+ resolution** is often required.
- This leads to **extremely high bitrates**, stressing networks and increasing delivery costs.

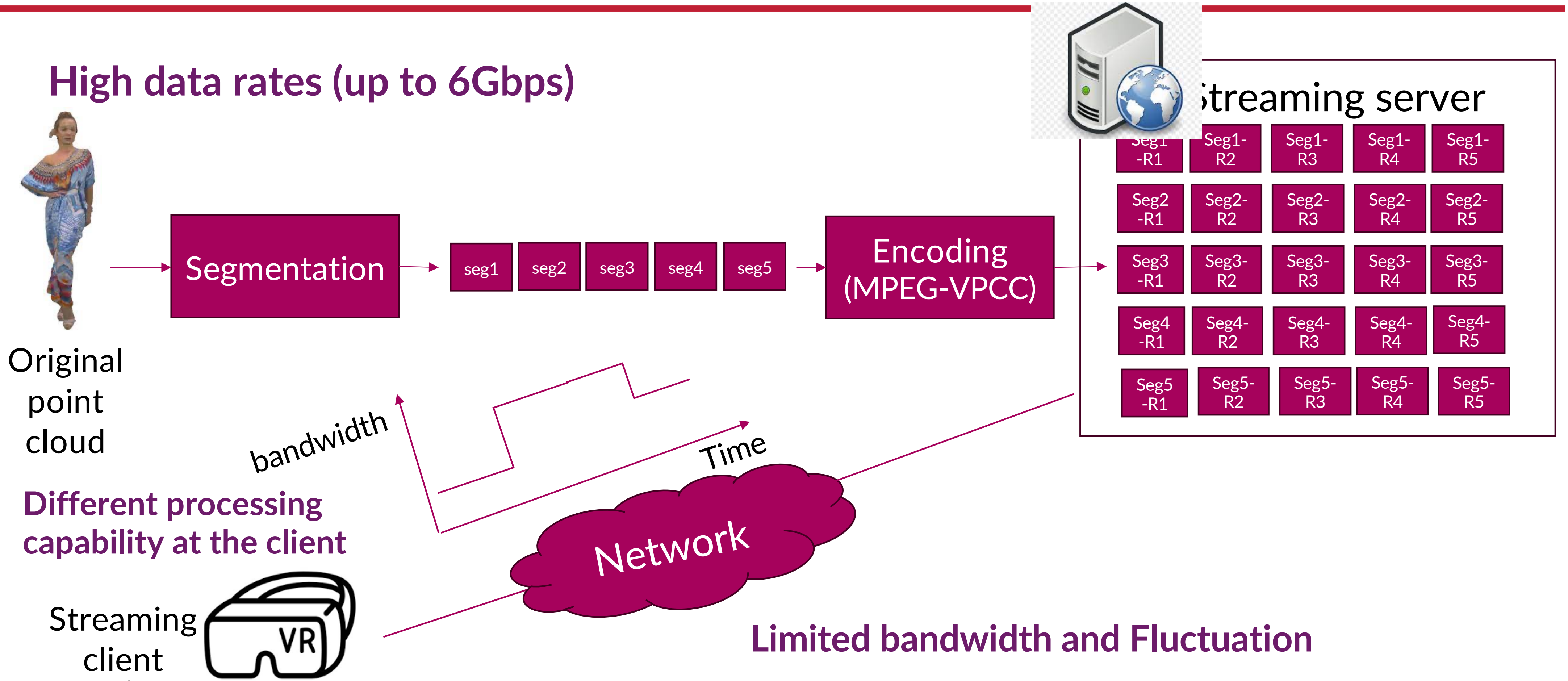


	HORIZONTAL AXIS	VERTICAL AXIS	HMD VIEWER	
2K→	2,000	1,000	500	
	4,000	2,000	1,000	
	6,000	3,000	1,500	
	8,000	4,000	2,000	
	10,000	5,000	2,500	
	12,000	6,000	3,000	
	14,000	7,000	3,500	
	16,000	8,000	4,000	←TRUE 4K

Challenges of 360-degree-video Delivery

- **Inefficient Data Usage (Viewport Problem)**
  - Only **10–20% of the video sphere** is visible at once.
  - Yet, traditional streaming delivers the **entire frame**, wasting bandwidth.
- **Limited bandwidth**
- **Need for real-time rendering on resource-constrained devices.**

Challenges of Point Cloud Video Delivery



Market Leadership

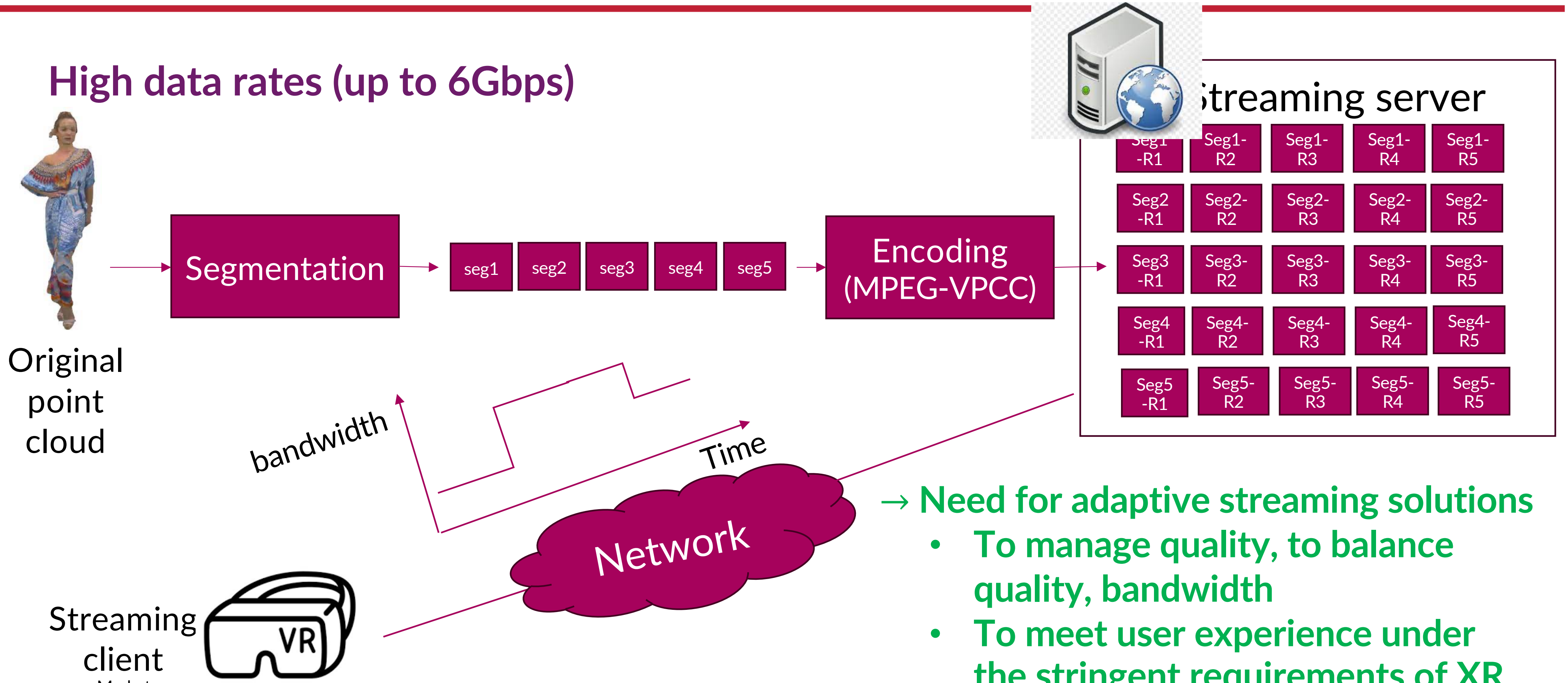
FROM MOTIVATIONS TO ACTIONS:

ADAPTIVE STREAMING TECHNIQUES TO SOLVE THE XR ISSUES

ONE LOVE. ONE FUTURE.

A need of Adaptive Streaming Solutions

High data rates (up to 6Gbps)



→ Need for adaptive streaming solutions

- To manage quality, to balance quality, bandwidth
- To meet user experience under the stringent requirements of XR application

Challenges of Adaptive Streaming Solutions

Viewport-adaptive streaming exists but is difficult to implement accurately and at scale.

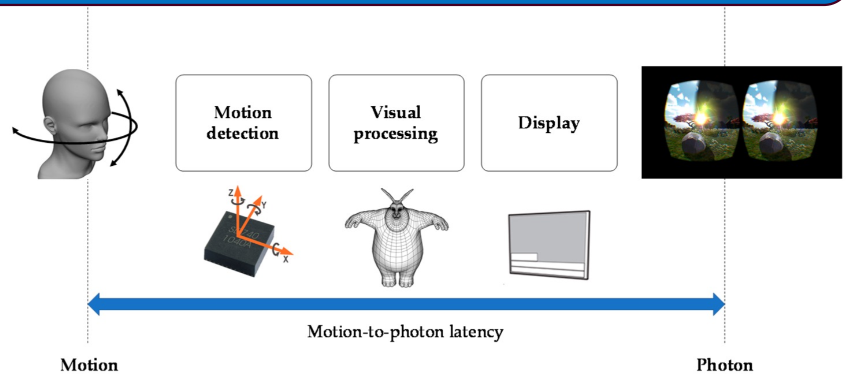
- Additional computational overhead
- Head/Eye movement prediction

Maintaining latency below 20 milliseconds is critical to ensure motion-to-photon responsiveness and user comfort in XR experiences.

Point cloud-adaptive streaming

- Limited processing power, to handle tile-based or multi-object volumetric streams efficiently
- Fast rate of 6Gbps at 30 frames per second over bandwidth-constrained networks.

Low-latency delivery is especially hard on **mobile networks** or congested Wi-Fi



Challenges of Adaptive Streaming Solutions

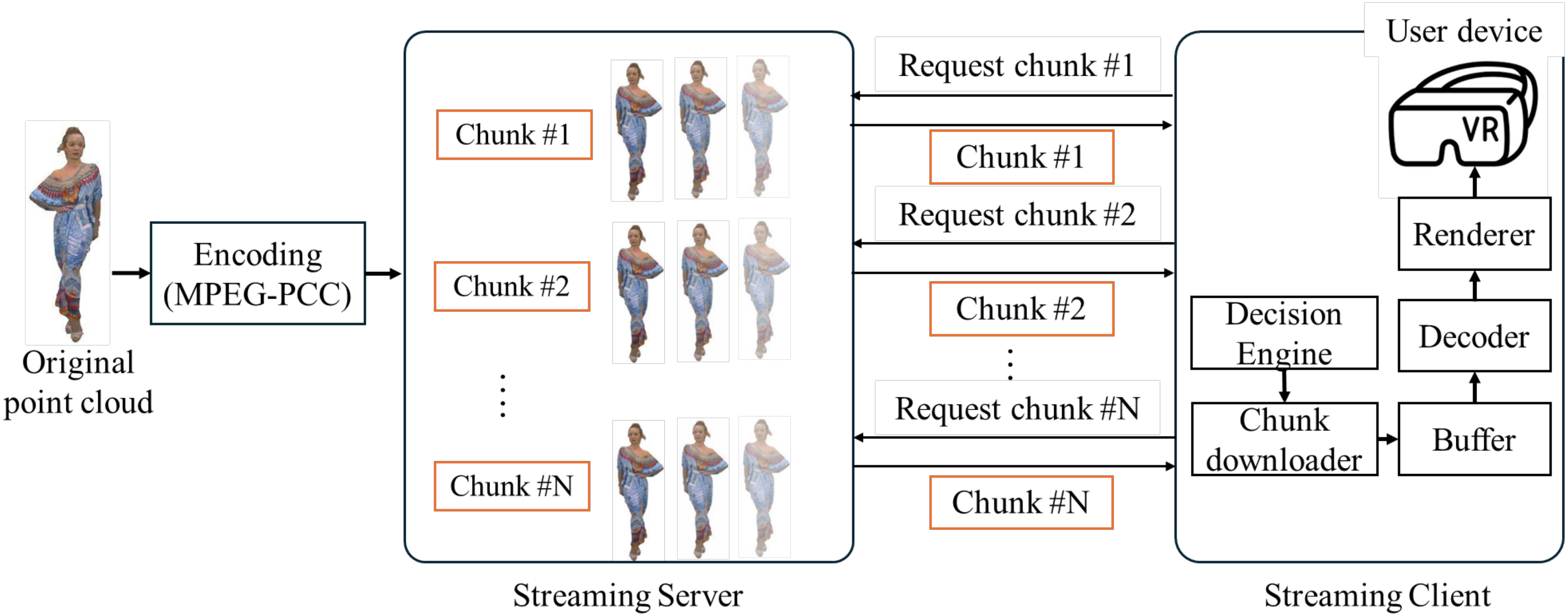
Viewport-adaptive streaming exists but is difficult to implement accurately and at scale.

- Additional computational overhead
- Head/Eye movement prediction
- **QoE degradation** with stalling and temporal quality variation

Point cloud-adaptive streaming

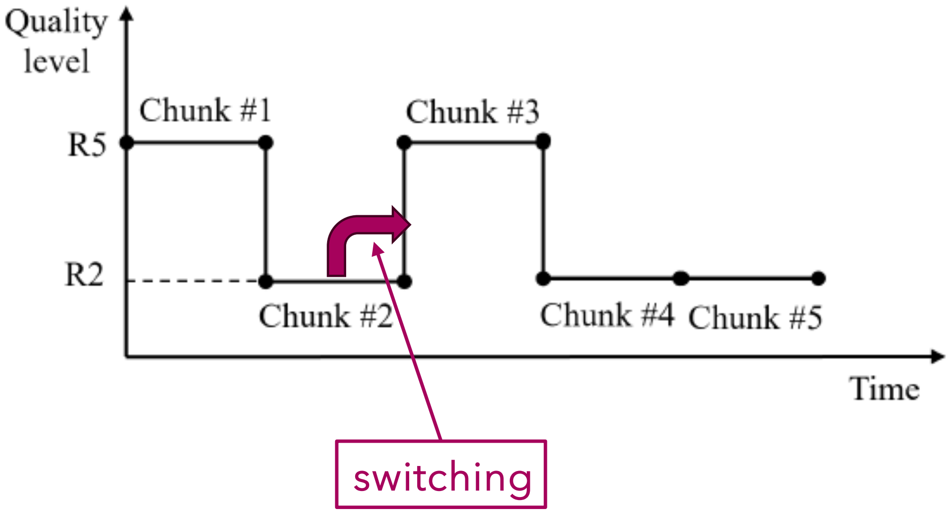
- Limited processing power, to handle tile-based or multi-object volumetric streams efficiently
- Fast rate of 6Gbps at 30 frames per second over bandwidth-constrained networks.

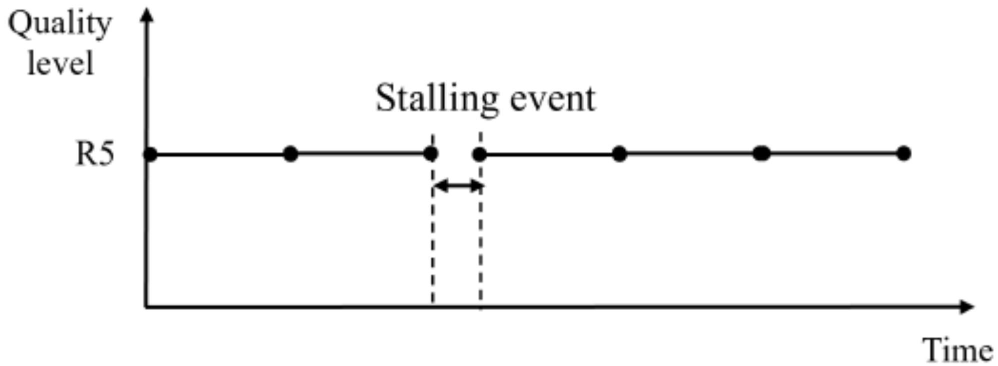
How Adaptive Point Cloud Video Streaming cause QoE degradation



How Adaptive Point Cloud Video Streaming cause QoE degradation

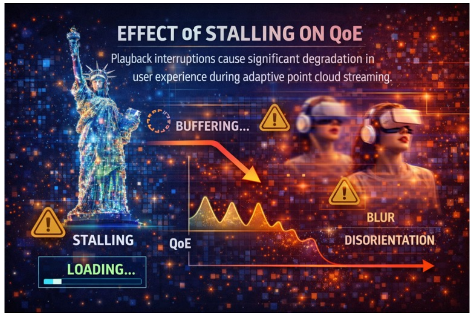
Key factors influencing QoE: temporal quality variation and stalling



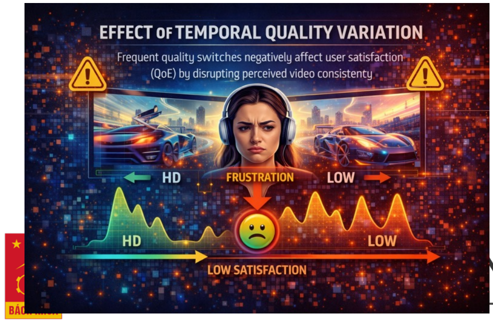


Ow Users' Quality of Experience is affected

Stalling and temporal quality variation complicate adaptive streaming for immersive media users.



Playback interruptions cause significant degradation in user experience during adaptive point cloud streaming.



Frequent quality switches negatively affect user satisfaction by disrupting perceived video consistency.

Therefore

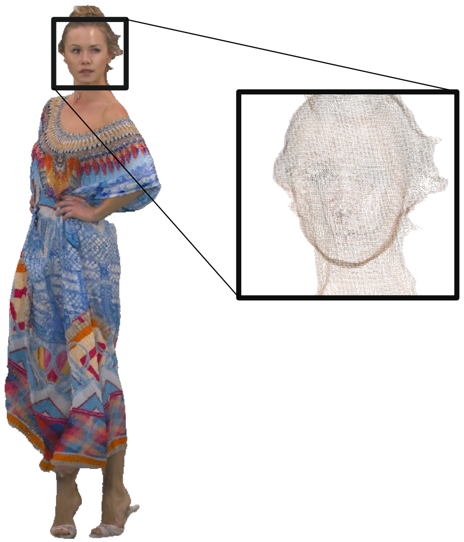
In adaptive streaming, we must trade off between the balance between computational complexity and visual quality in adaptive volumetric streaming.

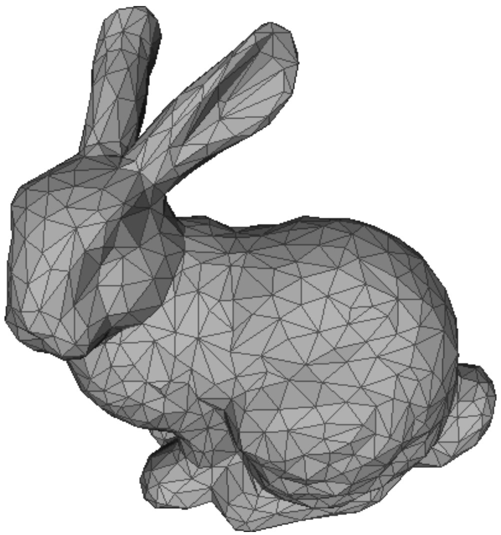
Background on Volumetric content

ONE LOVE. ONE FUTURE.

Background: What is volumetric video?

Background: Type of Volumetric video?





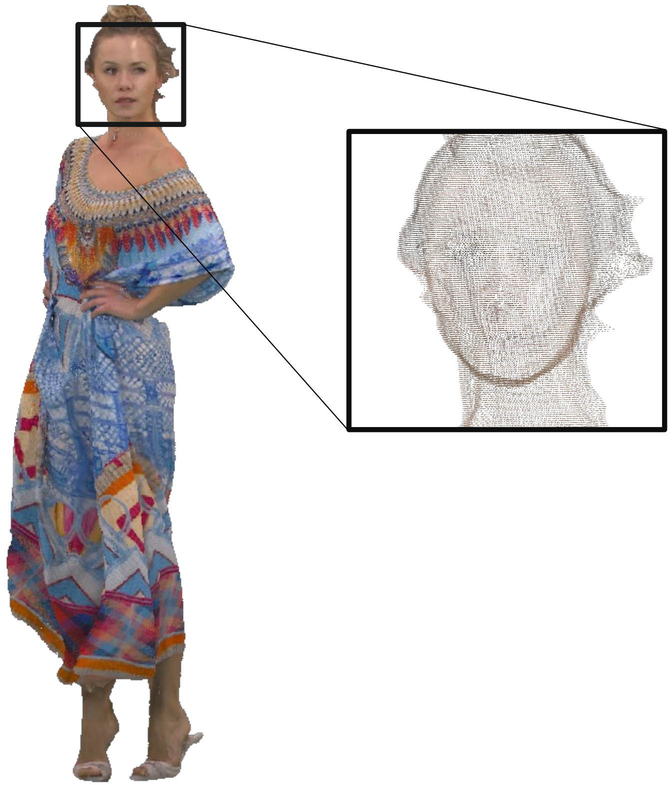
- **Large data capacity**

A shot of a single person requires at least 3.5 Gbps to stream without freezing[1] → four people require 19.2 Gbps [1].

- **High computational load**

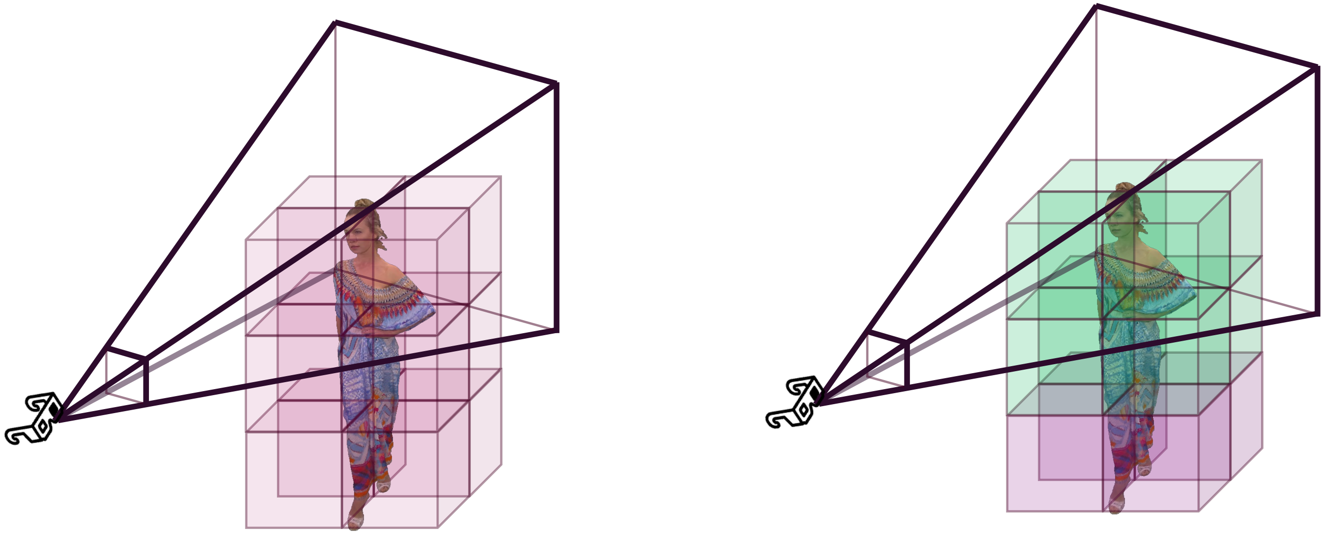
The decompression time for one frame must, on average, be shorter than the display time of that frame[1,2].

Reference:
[1]. 8i VFBv2
[2]. MPEG-DASH

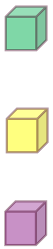


Video "longdress" (8iVFB v2 [1])

Background: State of the Art



- 3D tiling[3, 4, 5, 6, 7, 8, 9].
- Viewport Adaptive Streaming.

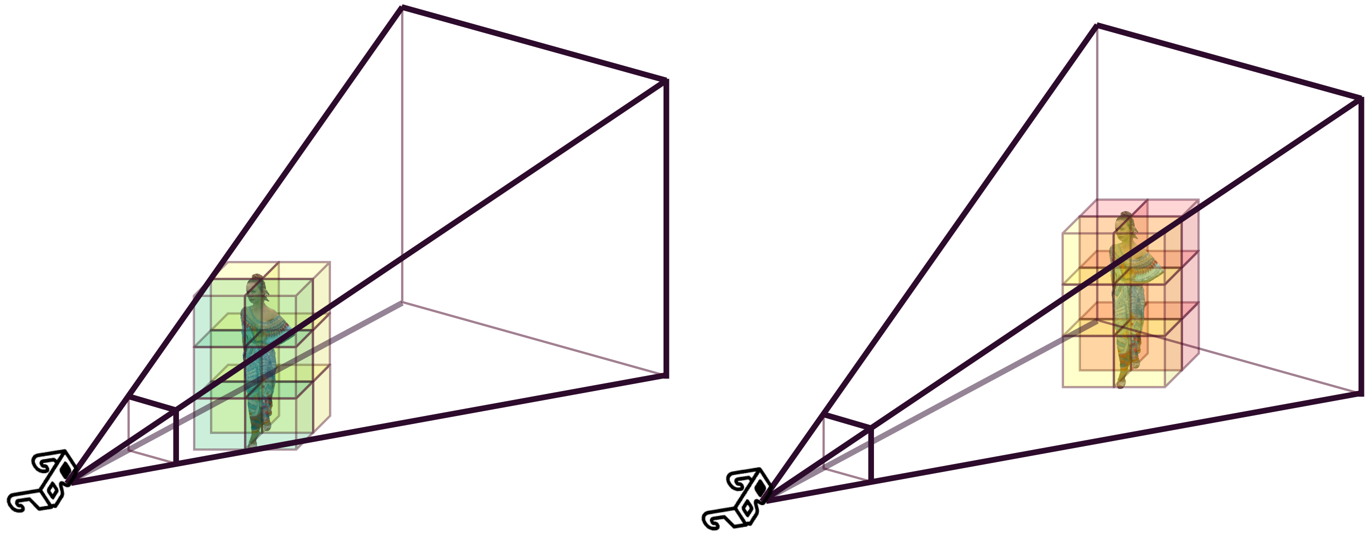


: High Quality

: Medium Quality

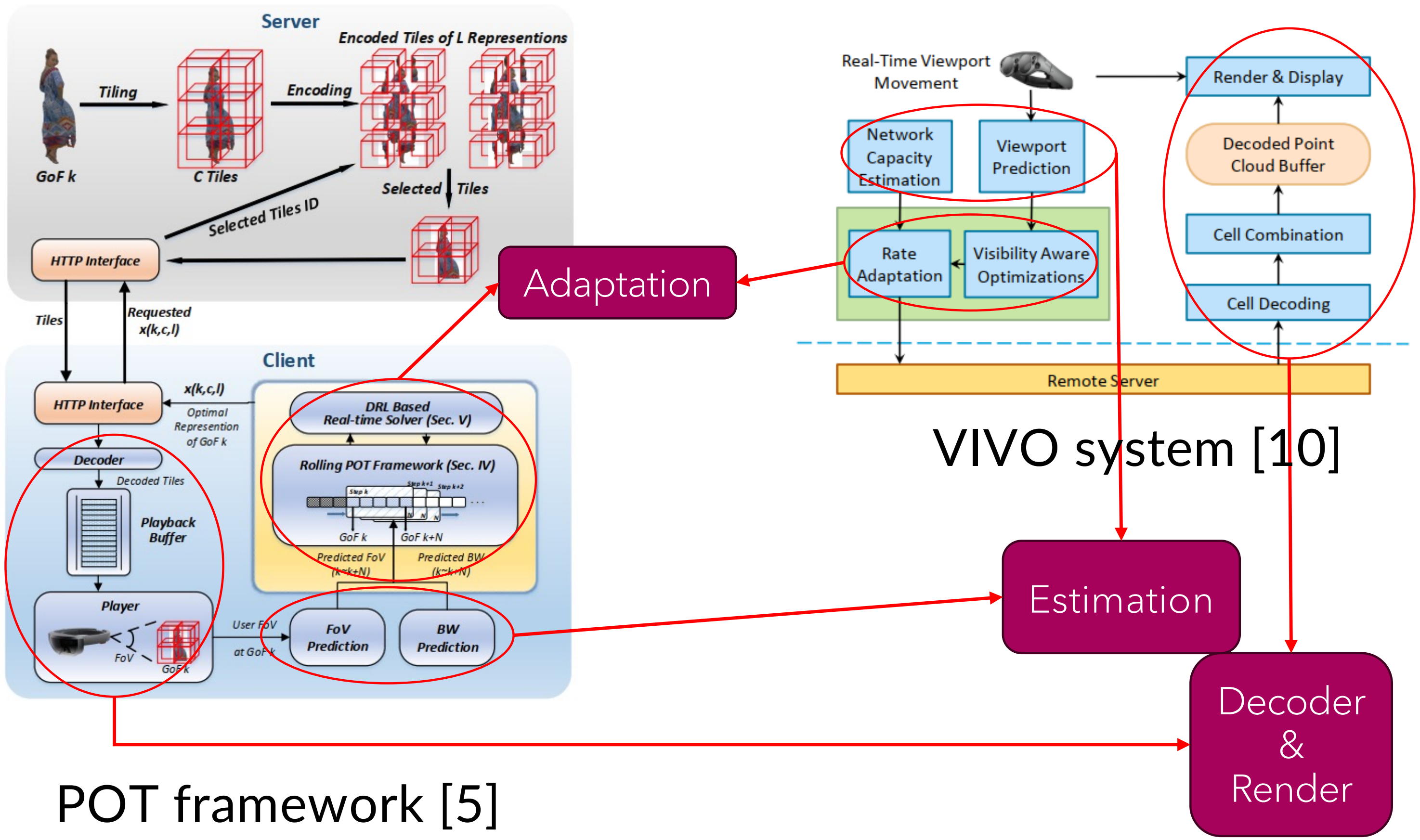
: Low Quality

Background: State of the Art



- The quality of the 3D tiles will be adapted based on the viewing distance and the visible area. [10, 11]

Background: State of the Art

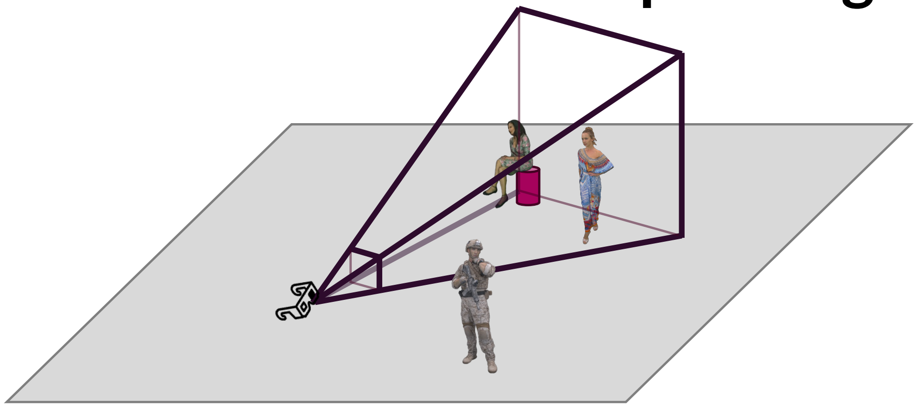


Background: State of the Art

- Current systems support streaming single-object scenes
- [3, 4, 5, 6, 7, 8, 9].
- With the 3D tiling technique, a larger number of objects must be decompressed independently, creating a burden on the viewer's device.

→ How to stream multi-object scenes?

→ How to optimize resources for decompressing multi-object scenes?

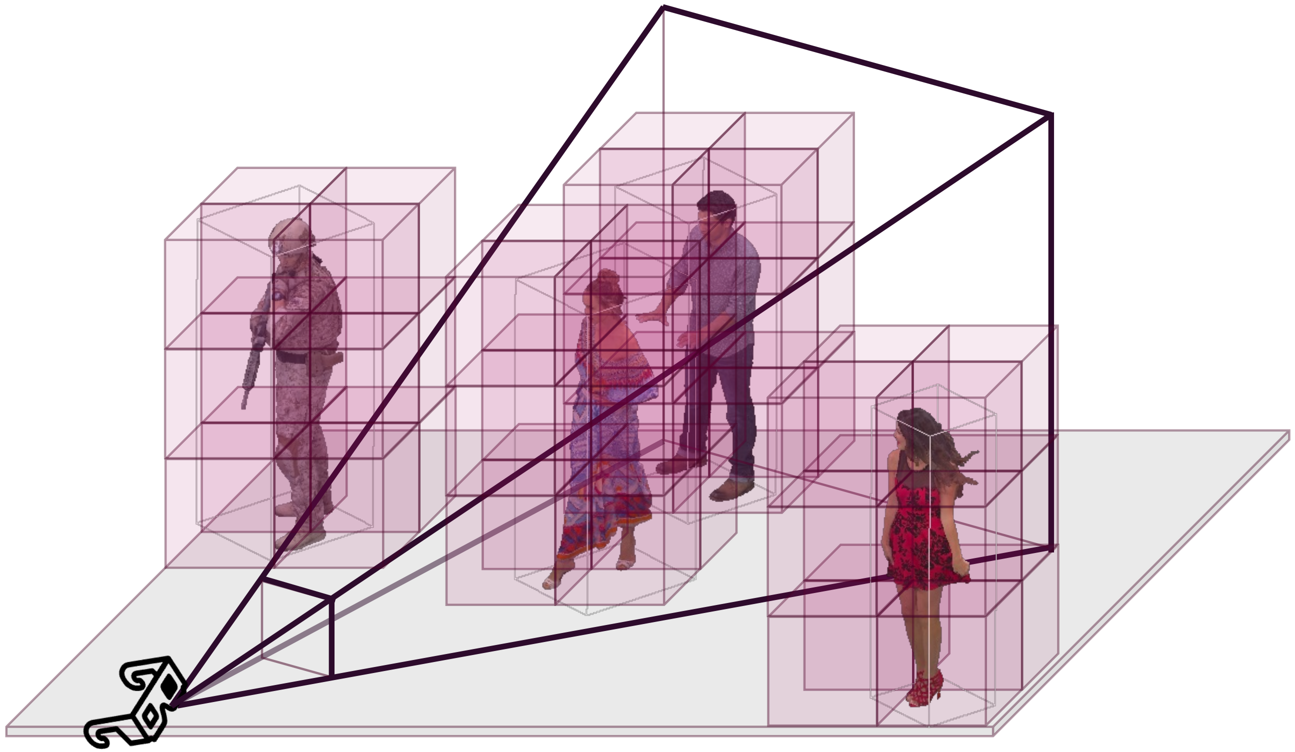


Objective: Streaming multi-object volumetric video

#1: Splitting into many tiles

- Requires more decompression, making it easier to freeze

- Adapts well and makes resource optimization easier.

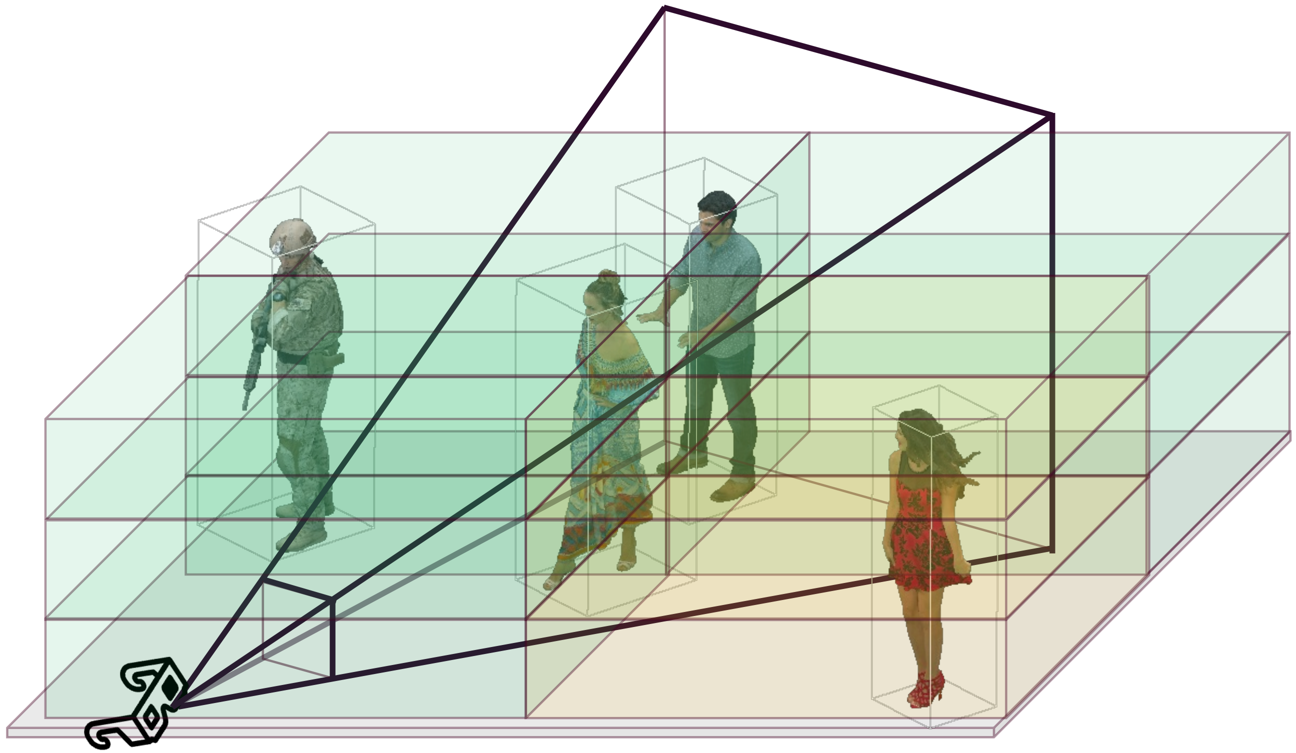


Objective: Streaming multi-object volumetric video

#2: Splitting into fewer tiles

- Requires less decompression, making freezing less likely

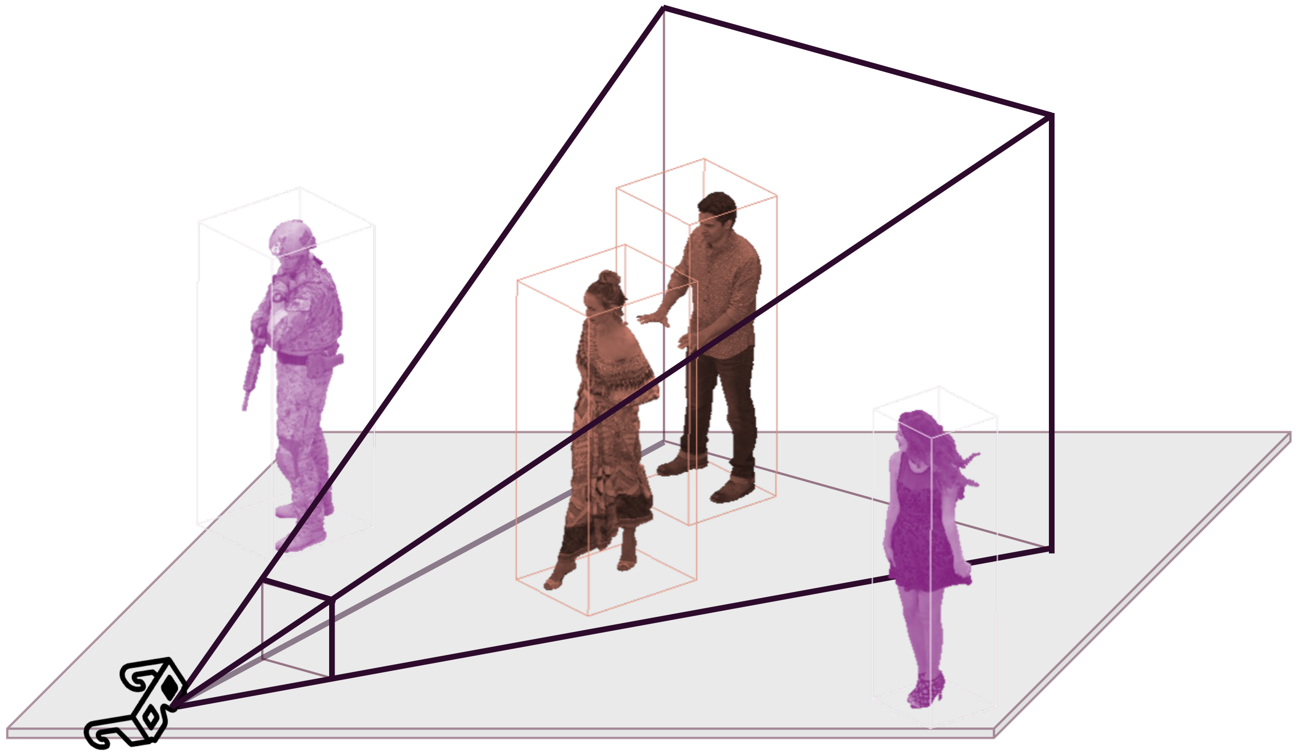
- Adapts poorly and makes resource optimization difficult



#3: No tiling

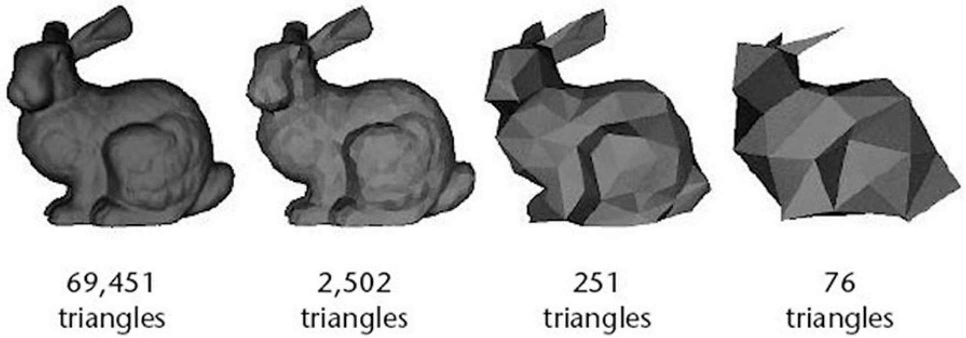
- Requires little decompression, making freezing unlikely

- Adapts well and is easy to optimize for resource usage



Simple compression techniques

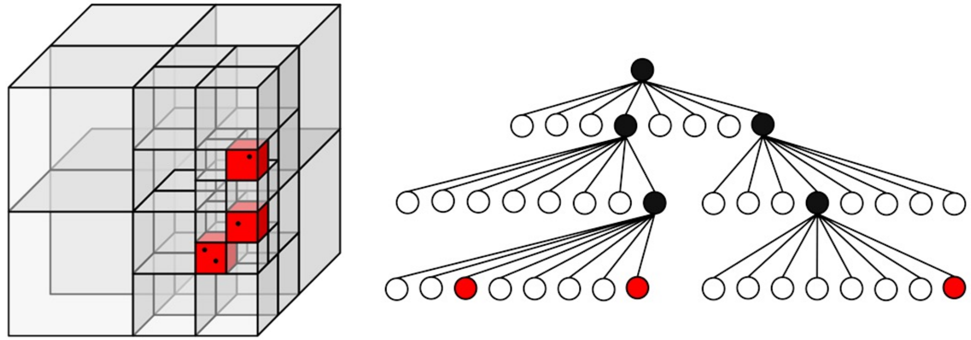
LoD (Mesh): reducing the edges and vertices while keeping the visual quality at a certain level.

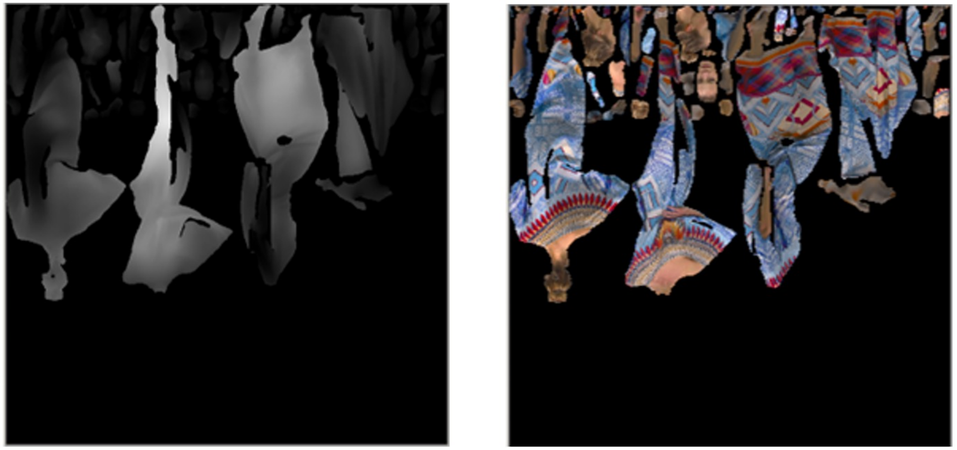


Subsampling: keeping only a subset of the point cloud based on some rules.

Advanced compression techniques

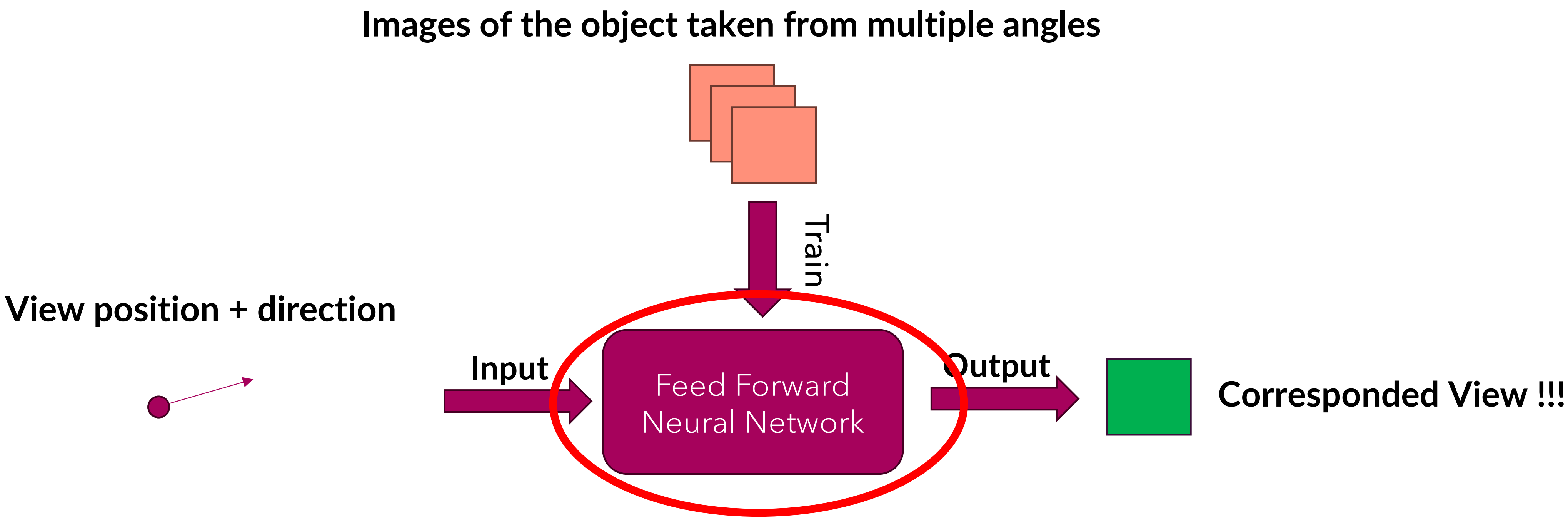
Octree Coding (point cloud): insert each 3D points into a Octree and output the serialized octree in a bitstream.





Projection-based coding (point cloud): project 3D surfaces of a point cloud into 2D plane, and use 2D video compression to compress.

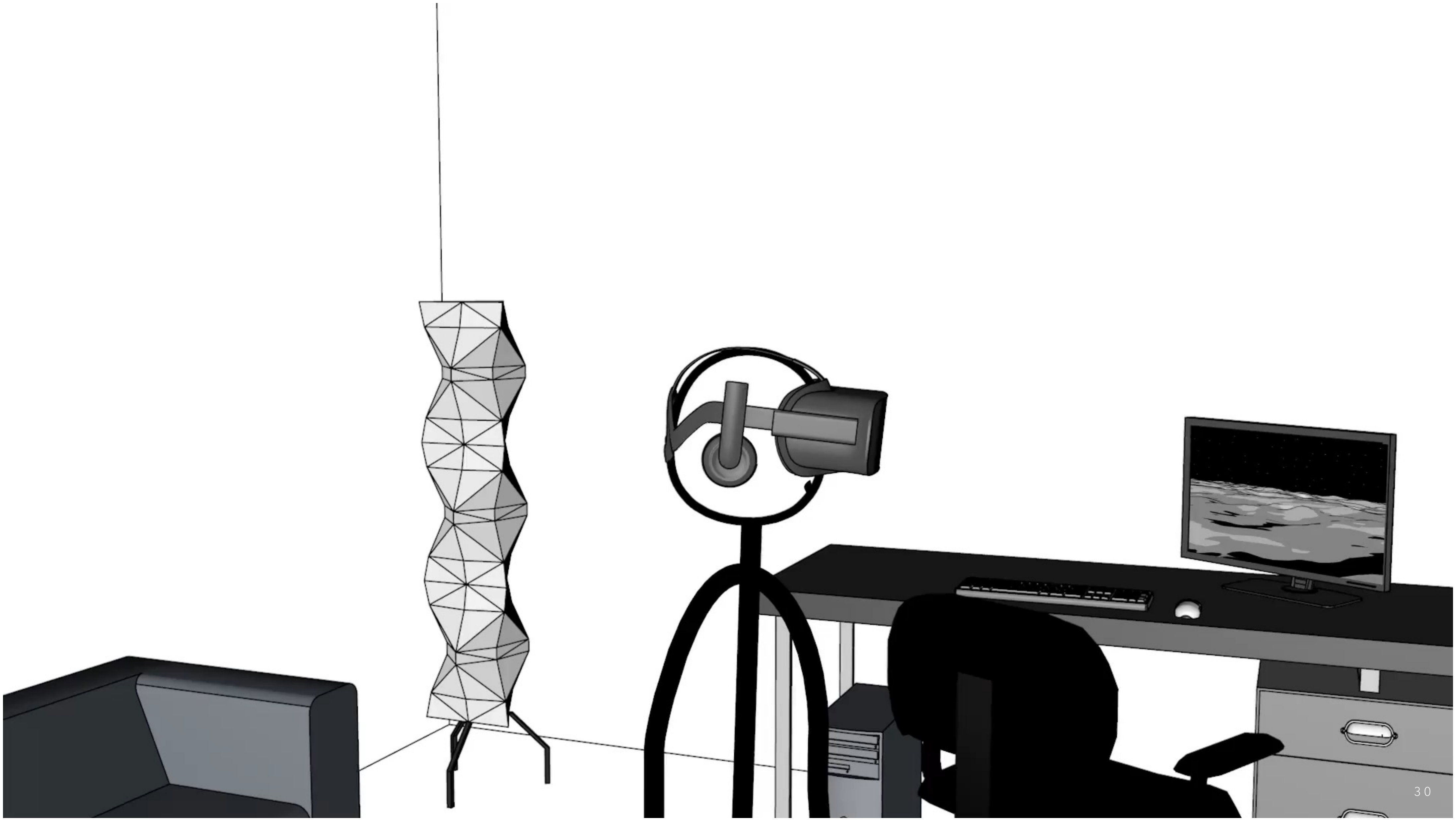
NeRF: a 3D representation and also A COMPRESSION TECHNIQUE !



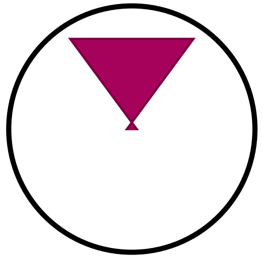
STORE and SHARE the Neural Network !!!

Background on 360-degree videos

ONE LOVE. ONE FUTURE.



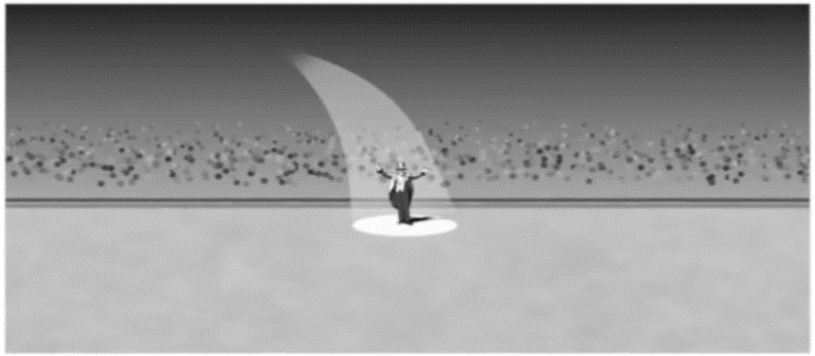
What is 360-degree video?

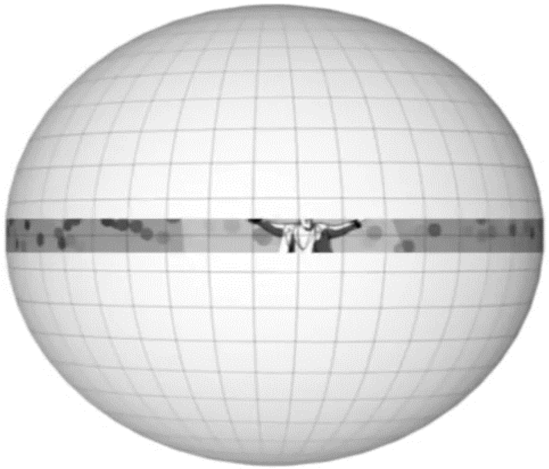


Viewport

Viewing direction

What is 360-degree video?

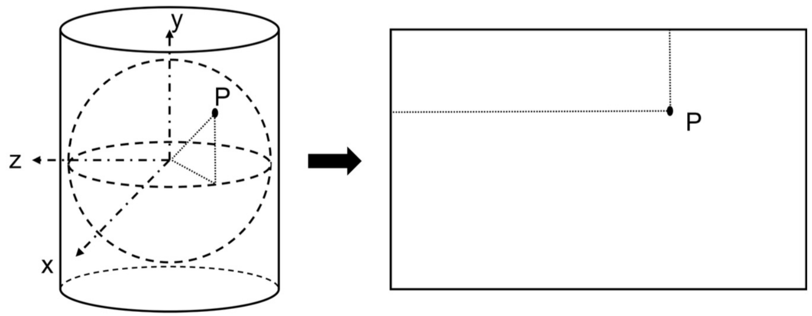




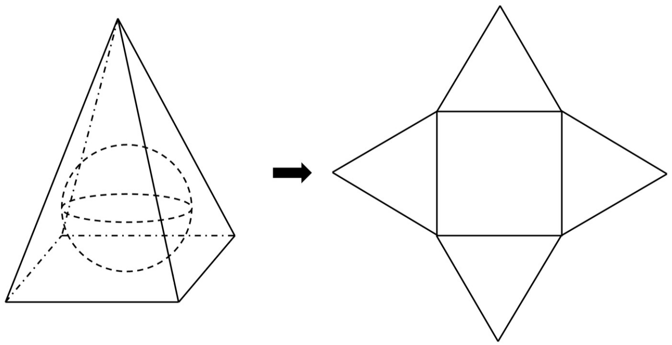
360-degree video background

360° Video Mapping Techniques to convert spherical video into rectangular video before encoding:

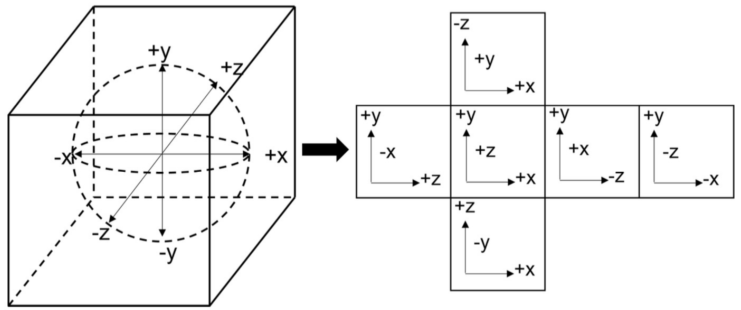
- cylindrical mapping
- cubic mapping
- pyramid mapping



Equirectangular Projection - ERP



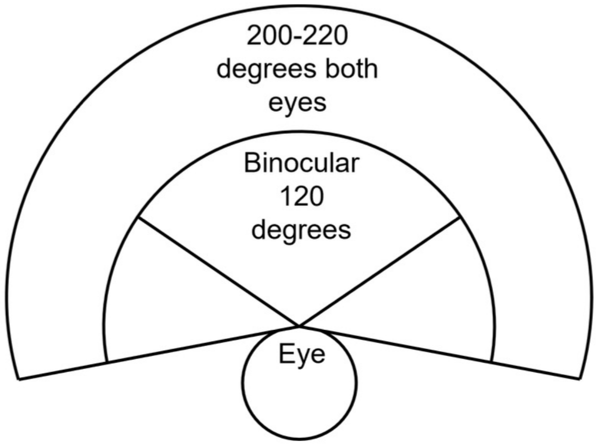
Pyramid Projection



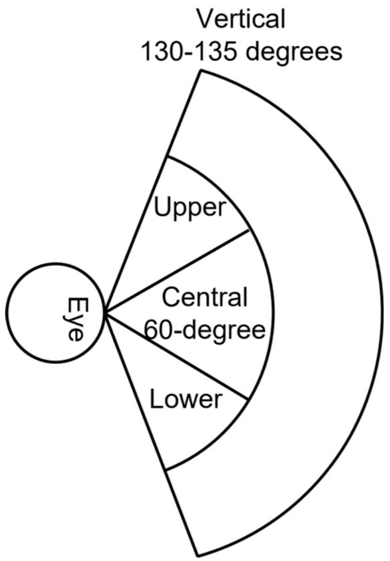
Cubemap Projection - CMP

360-degree video background

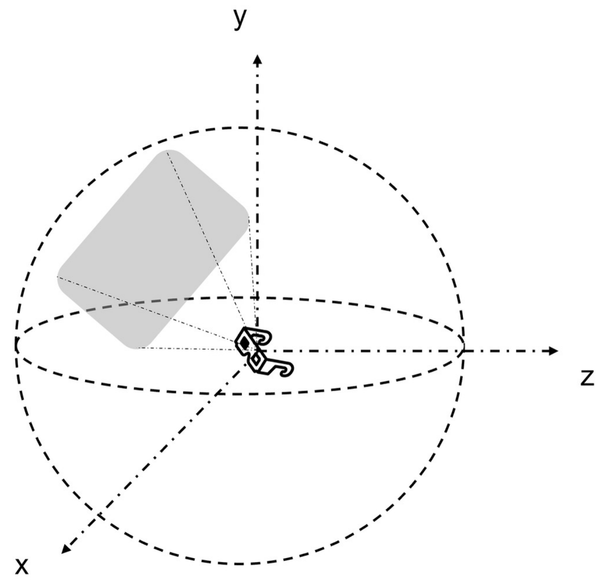
Viewport/Field of View concept



Horizontal Field of View



Vertical Field of View



Viewport

Problem statement

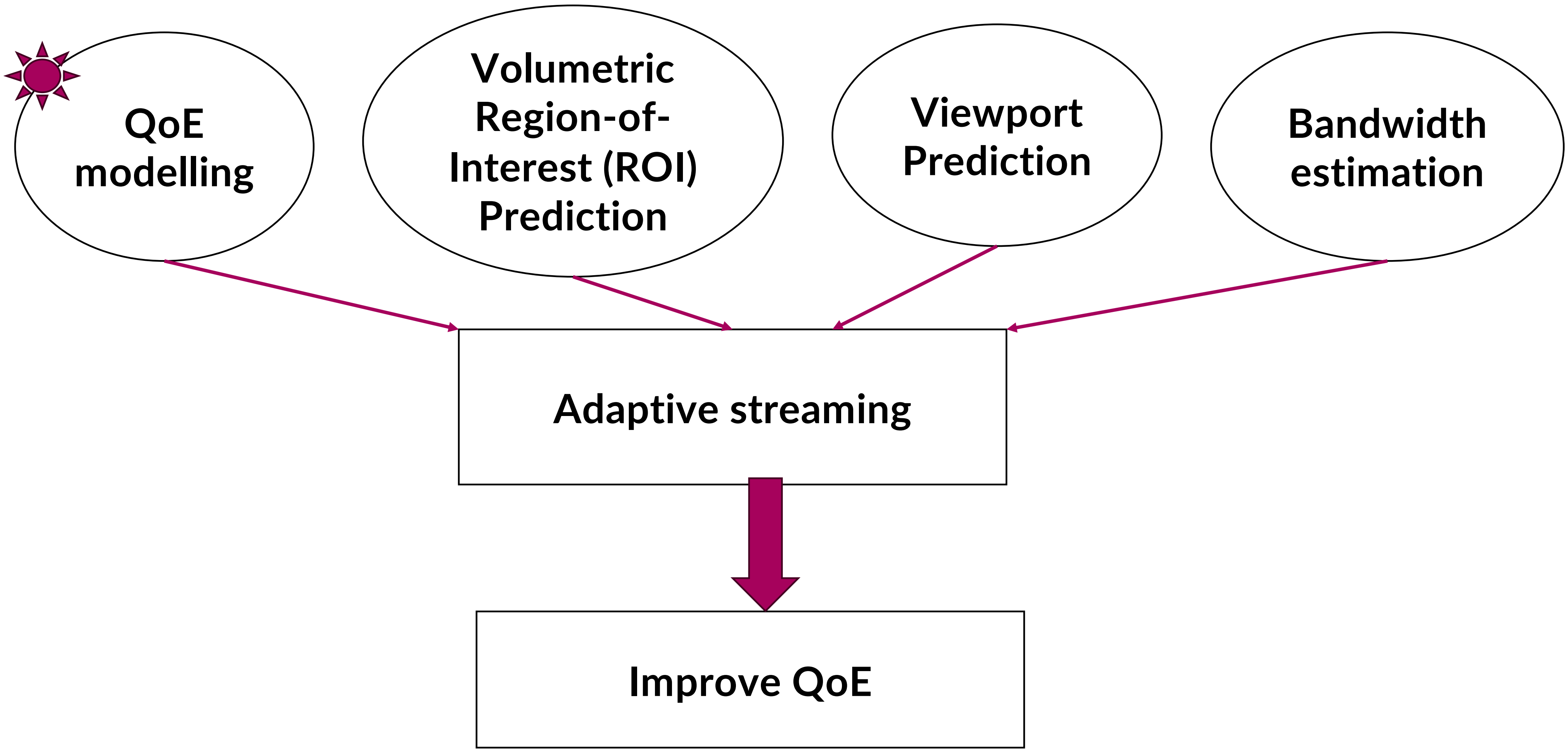
360-degree key features

- Users can freely rotate their heads to explore the video in all directions.
- 360° video is usually captured and delivered at ultra-high resolution (>4K).
- Live 360° video streaming demands high bandwidth

How to live stream 360-degree video over mobile networks?

- with good QoE
- Low-latency playback without buffering or stalls
- Efficient resource usage

Remember this diagram before we start



Background on QoE and QoE modelling

ONE LOVE. ONE FUTURE.

Needs for QoE modelling

VR content has large file sizes

Applications that require real-time processing

Adaptive methods for video/image adjustment (or tuning)

Reducing the size of VR content

Ensuring user experience

Method evaluation

Subjective

- *With a large number of participants*
- *Direct observation and evaluation*

Time-consuming and costly

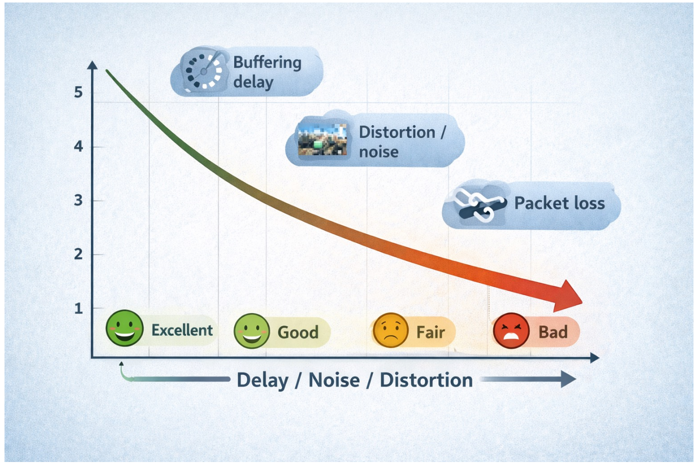
Objective

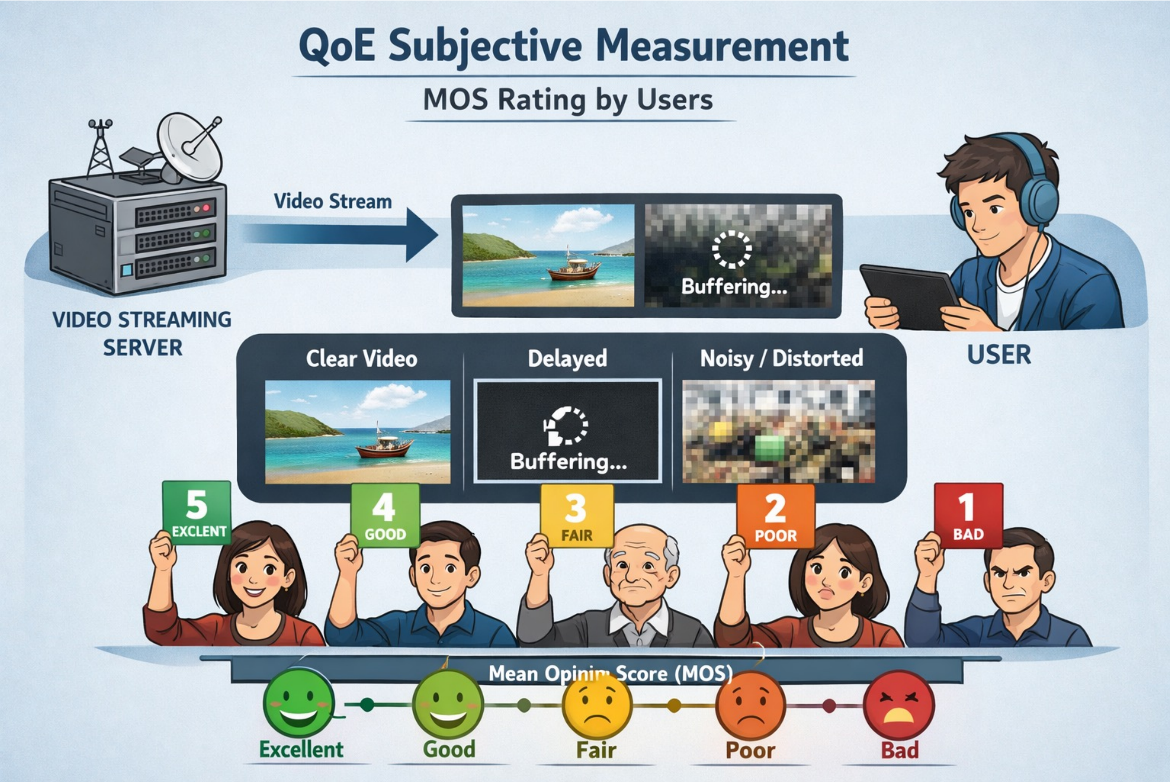
- *Does not require user involvement*
- *Conducted on a computer*

Requires initial research but helps reduce costs in the long term

Subjective QoE assessment

MOS	Quality
1	Excellent
2	Good
3	Fair
4	Poor
5	Bad





Objective QoE assessment is a cure

Subjective

- *With a large number of participants*
- *Direct observation and evaluation*

Mapping

Objective

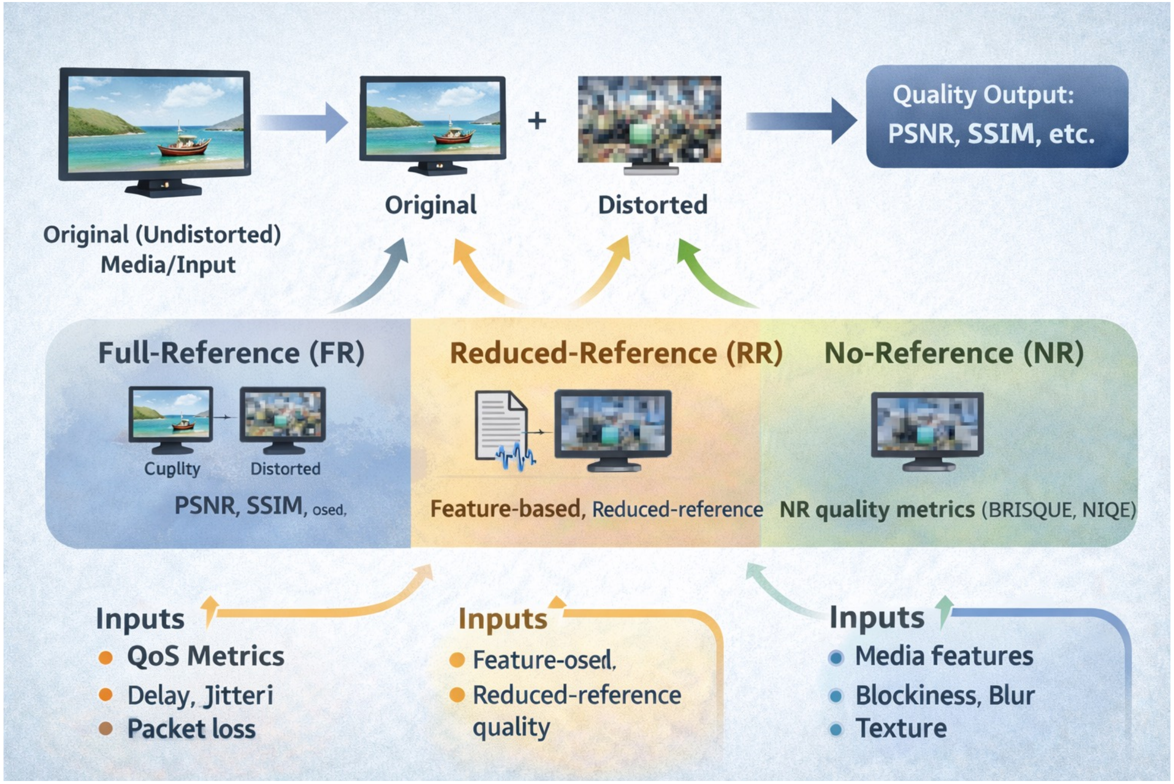
- *Does not require user involvement*
- *Conducted on a computer*

Time-consuming and costly

Requires initial research but helps reduce costs in the long term

MOS = f(Objective parameters)

Objective QoE assessment

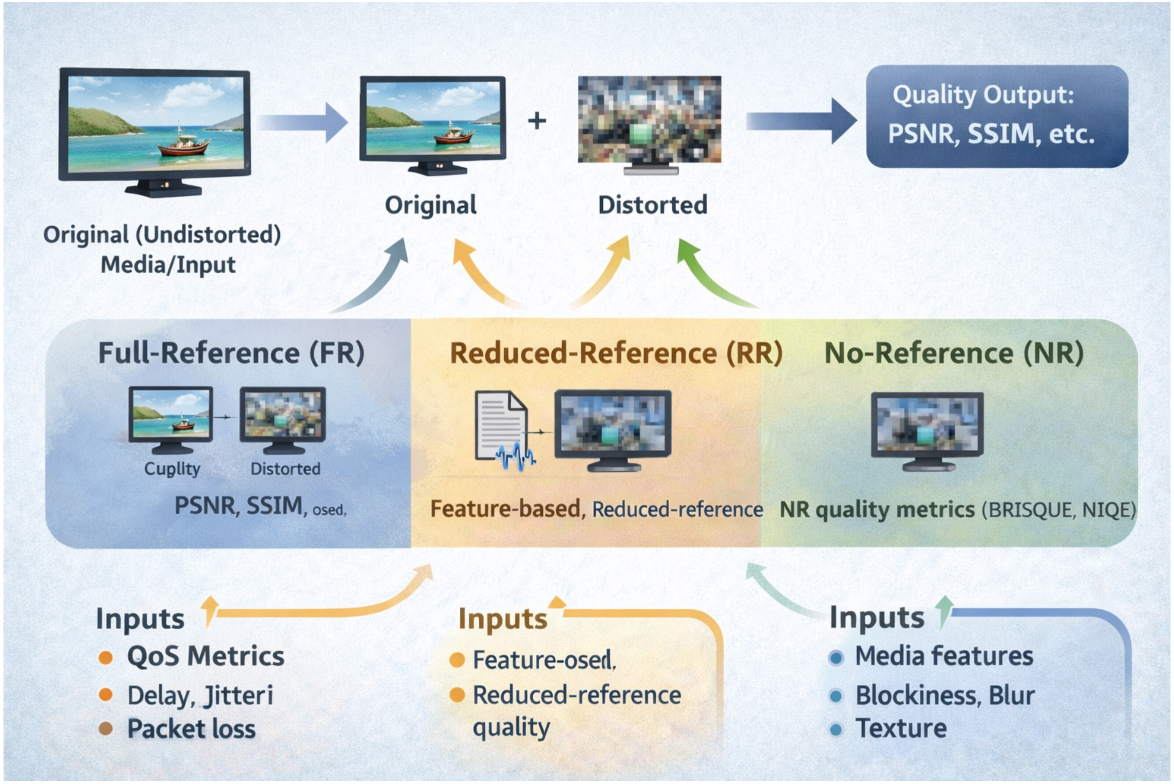


Full-Reference (FR) Methods

Require access to the original (undistorted) signal.

- **PSNR** (Peak Signal-to-Noise Ratio)
- **SSIM** (Structural Similarity Index)
- **MS-SSIM** (Multi-Scale SSIM)
- **VQM** (Video Quality Metric)
- **VMAF** (Video Multi-Method Assessment Fusion)

Objective QoE assessment

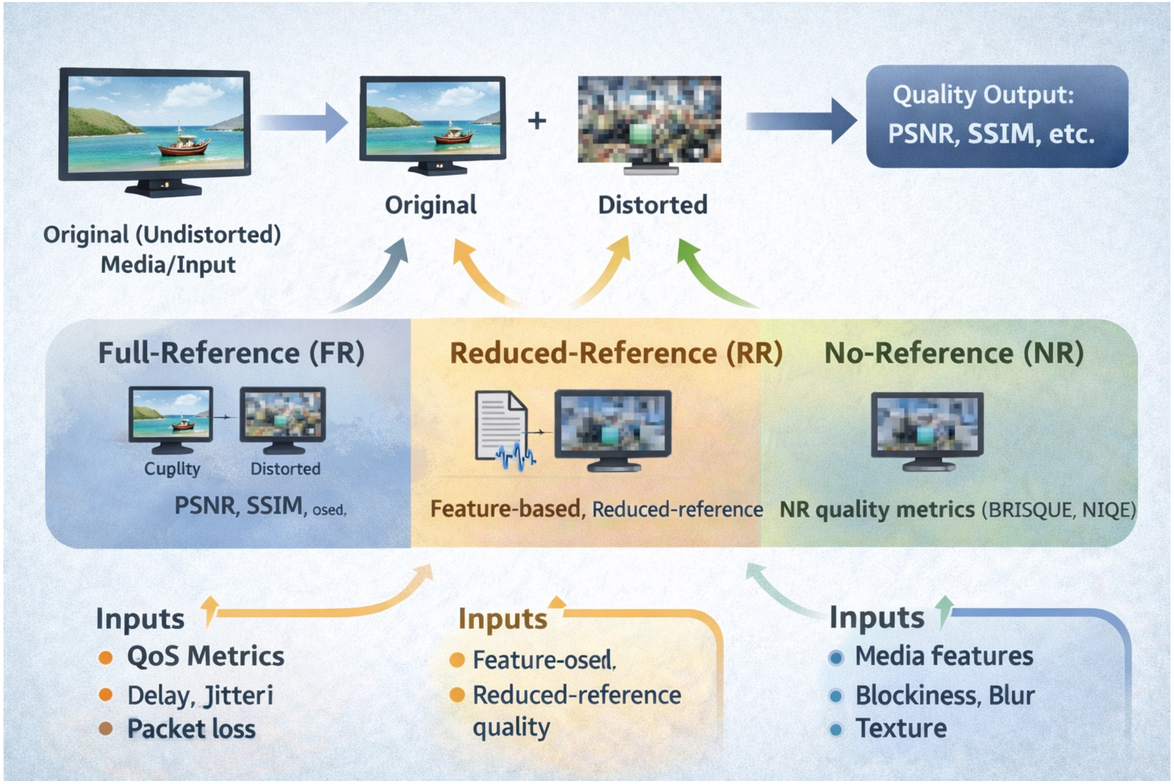


Reduced-Reference (RR) Methods

Use partial information about the original signal.

- Feature-based quality metrics
- Reduced-reference video quality models

Objective QoE assessment



No-Reference (NR) / Blind Methods

Use only the received signal.

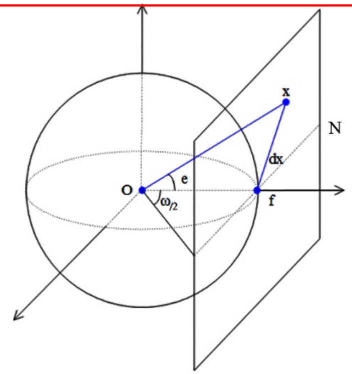
- Blockiness, blur, noise estimators
- NR video quality metrics (e.g., BRISQUE, NIQE)
- Deep-learning–based quality predictors

Example 1: objective assement for 360-degree image

$$\text{WVPSNR} = 10\,\log_{10}\frac{(\sum_{n=1}^{N} w^2)\text{MAX}^2}{\sum_{n=1}^{N}[v(x_n)-g(x_n)]^2\, w^2}\ (\text{dB})[1]$$

W = f(fc)

Spatial frequency(fc)



$$f_{cx}{}' = \frac{f_{cx}}{(\cos e_x)^2}$$

Constrast Sencitivity Function

$$\text{fc} = \frac{e2\,Ln(\frac{1}{CT_0})}{\alpha(e+e2)}\left(\frac{cycles}{degree}\right)$$

e2 = ?

[1] Sanghoon Lee, M. S. Pattichis, and A. C. Bovik, "*Foveated video compression with optimal rate control*", IEEE Transactions on Image Processing, Volume 10 Issue 7, July 2001, Pages 977-992

Example 1: objective assement for 360-degree image

Coefficient Determination Process

[3] Joint Video Exploration Team, "360Lib." [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_360Lib/tags/360Lib-2.0.1/

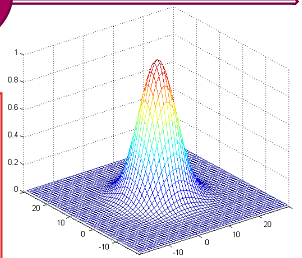
[2] P.913, Recommendation ITU-T, "Methods for the subjective assessment of video quality, audio quality and audiovisual quality of Internet video and distribution quality television in any environment," 2014

Gaussian Filtering for Image Blurring

360lib Software [3]

360^o original image

360-degree images differ in complexity, time, and objects



Blurred 360^oimage

viewport extraction

Viewport

MOS

Curve fitting phép đo với điểm MOS tương ứng

Coefficient $for\ 360\ image^o$

Absolute Category Rating method[2] với 5 mức điểm 1-5.

Tối ưu PCC

Example 2: QoE modelling for point cloud

Step 1

Construct a QoE database for Point cloud video

Construct a large QoE database

Evaluate impacts of temporal quality variation and stalling on QoE in adaptive point cloud video streaming in a VR setting.

Step 2

Model QoE

Using machine learning to develop prediction models

Develop models for predicting users' Quality of Experience given the impacts of temporal quality variation and stalling.

Example 2: QoE modelling for point cloud

Step 1

Construct a QoE database for Point cloud video

Construct a large QoE database

Evaluate impacts of temporal quality variation and stalling on QoE in adaptive point cloud video streaming in a VR setting.

Example 2: QoE modelling for point cloud:Construction of QoE database

(a) Longdress (b) Loot (c) RedandBlack (d) Soldier

Quality	GQP	TQP	Loot	RedandBlack	Soldier	Longdress
R1	32	42	2.27	3.38	4.37	4.64
R2	28	37	3.48	4.88	6.96	7.97
R3	24	32	5.62	7.55	11.58	14.05
R4	20	27	9.41	12.76	19.95	25.97
R5	16	22	16.67	22.91	35.29	46.77

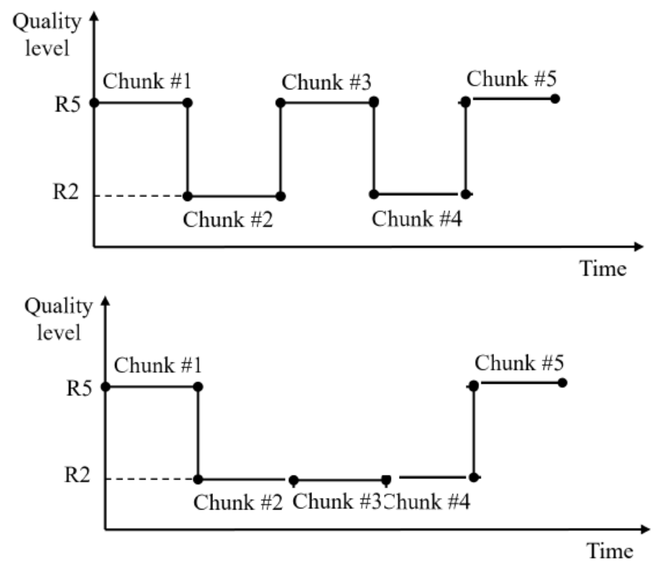
4 original point cloud videos from 8i Voxelized Full Body Dataset:

- Each video is 10 second long at 30 frames per second
- Each video is divided into five 2-second chunks, and each chunk is encoded into five quality levels (versions) using MPEG V-PCC compression standard

Test Stimuli Patterns for Temporal Quality Variation

29 stimuli with various temporal quality variation patterns are generated for each point cloud video by concatenating chunk versions based on pre-defined patterns:

- **Constant** (5 patterns): All chunks have the same quality level of either R1, R2, R3, R4, or R5.
- **Spike** (4 patterns): R5-Rx-R5-Rx-R5, where Rx is either R4, R3, R2, or R1.
- **InverseSpike** (4 patterns): Rx-R5-Rx-R5-Rx, where Rx is either R4, R3, R2, or R1.
- **SingleDrop** (12 patterns): R5-Rx-R5-R5-R5 or R5-Rx-Rx-R5-R5 or R5-Rx-Rx-Rx-R5
- **SingleIncrease** (4 patterns): Rx-R5-Rx-Rx-Rx with Rx is either R4, R3, R2, or R1.



Stalling Patterns in Test Stimuli

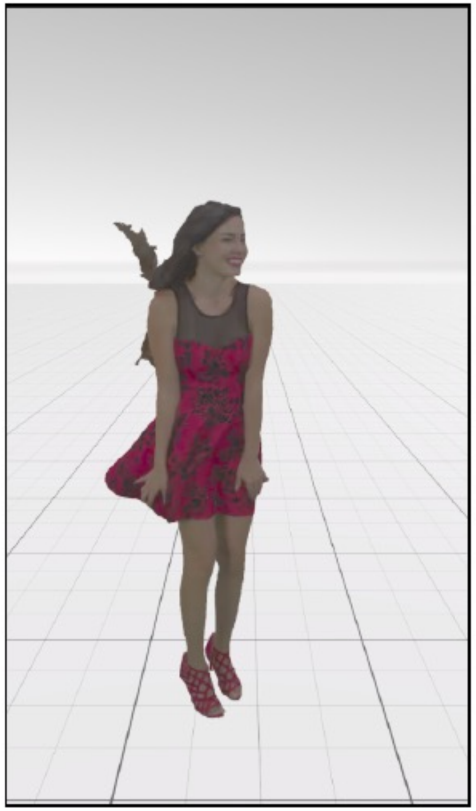
33 stalling patterns are generated for each point cloud video at R5 with 8 stalling durations of 0.25s, 0.5s, 0.75s, 1s, 1.5s, 2s, 3s, and 4s:

- **Single-Stall** (16 patterns)**:** either at the end of the first chunk or the end of the fourth chunk
- **Double-Stall** (8 patterns)**:** 2 stalling occur either at 1) the end of the first and third chunks or 2) the end of the second and third chunks. Stalling in a stimulus has the same duration of either 0.25s, 0.5s, 1s, or 2s.
- **Triple-Stall** (6 patterns): 3 stalling occur either 1) the end of the first, third, and fourth chunks or 2) the end of the first, second, and third chunks. Stalling in a stimulus has the same duration of either 0.25s, 0.5s, or 1s.
- **Quadruple-Stall** (3 patterns): A stalling occurs at the end of all chunks except the last one. Stalling in a stimulus has the same duration of either 0.25s, 0.5s, or 1s.

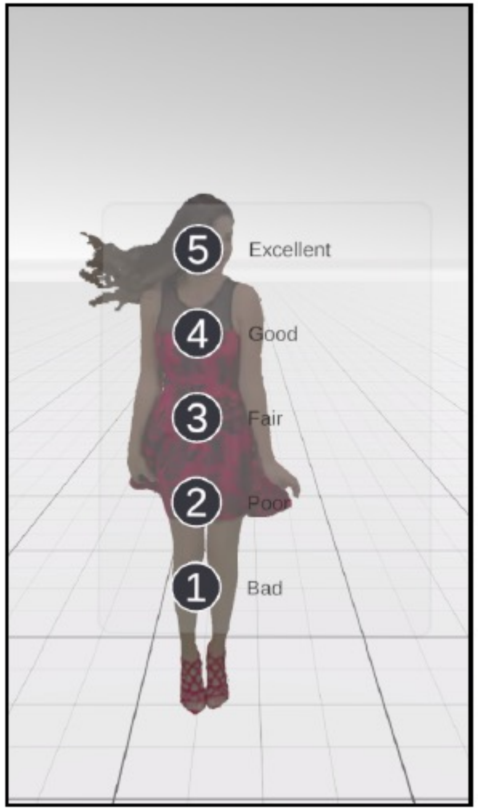
→ Total test stimuli: (29+33) *4 = 248

Test Environment and Test Procedure

- Unity and HTC Vive Pro headset
- 43 participants between 19 and 45, all with normal or corrected-to-normal vision.
- At least 17 participants rate each stimulus.
- Each stimulus's mean opinion score (MOS) is calculated as the average score given by all valid participants.

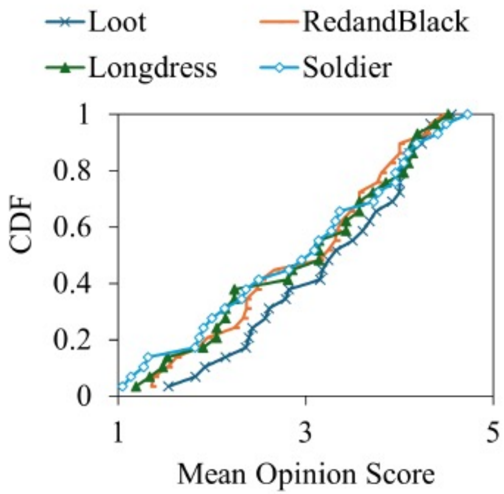


(a) A test stimulus from the participant's viewpoint.

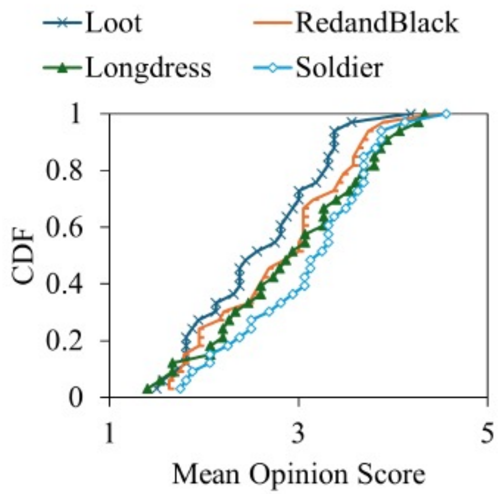


(b) The rating window.

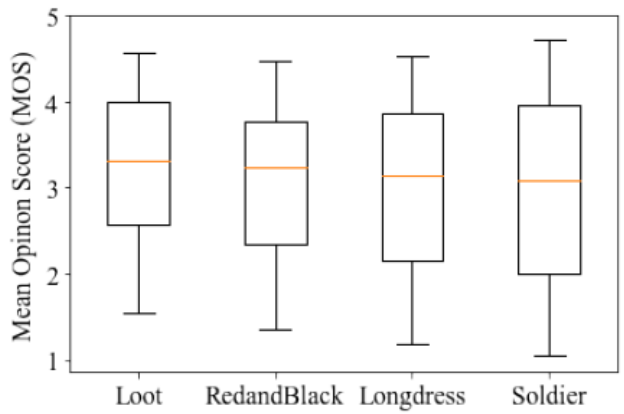
Test Results



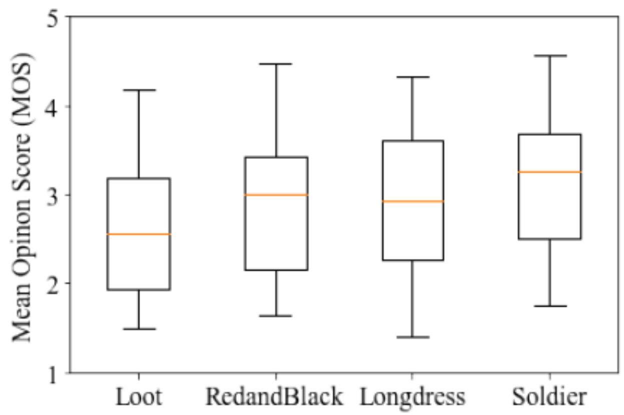
(a) Test stimuli with temporal quality variations



(b) Test stimuli with stalling



(a) Test stimuli with temporal quality variations



(b) Test stimuli with stalling

Example 2: QoE modelling for point cloud

Step 2

Model QoE

Using machine learning to develop prediction models

Develop models for predicting users' Quality of Experience given the impacts of temporal quality variation and stalling.

QoE Modeling for Temporal Quality Variation

6 features for GQP and TQP:

$$x_1^{qp} = \frac{1}{N}\sum_{i=1}^{N}(GQP_i + TQP_i) \tag{1a}$$

$$x_2^{qp} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GQP_i + TQP_i - x_1^{qp})^2} \tag{1b}$$

$$x_3^{qp} = \min(GQP_1 + TQP_1, \ldots, GQP_N + TQP_N) \tag{1c}$$

$$x_4^{qp} = \max(GQP_1 + TQP_1, \ldots, GQP_N + TQP_N) \tag{1d}$$

$$x_5^{qp} = \sum_{i=1}^{N-1} \mathbf{1}(GQP_{i+1} + TQP_{i+1} - GQP_i - TQP_i)) \tag{1e}$$

$$x_6^{qp} = \sum_{i=1}^{N-1} \mathbf{1}(GQP_i + TQP_i - GQP_{i+1} - TQP_{i+1})) \tag{1f}$$

4 features for bitrate:

$$x_1^{br} = \frac{1}{N}\sum_{i=1}^{N} r_i \tag{2a}$$

$$x_2^{br} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(r_i - x_1^{br})^2} \tag{2b}$$

$$x_3^{br} = \min(r_1, r_2, \ldots, r_N) \tag{2c}$$

$$x_4^{br} = \max(r_1, r_2, \ldots, r_N) \tag{2d}$$

QoE Modeling for Temporal Quality Variation

The user's QoE is predicted as a weighted sum of the extracted features :

$$QoE^{Pred} = \sum_{j=1}^{6} w_j^{qp} \times x_j^{qp} + \sum_{j=1}^{4} w_j^{br} \times x_j^{br}$$

To learn the appropriate values of the model parameters, the least square method is utilized and the mean square error with L2-regularization is used as the loss function to avoid over-fitting:

$$L = \frac{1}{N_s} \sum_{i=1}^{N_s} (QoE_i^{Pred} - QoE_i)^2 + \alpha\left(\sum_{j=1}^{6} (w_j^{qp})^2 + \sum_{j=1}^{4} (w_j^{br})^2\right)$$

QoE Modeling for Stalling

5 features for stalling:

$$x_1^s = \sum_{i=1}^{N} s_i \tag{5a}$$

$$x_2^s = \sum_{i=1}^{N} \mathbf{1}(s_i) \tag{5b}$$

$$x_3^s = \min(s_1, s_2, \ldots, s_N) \tag{5c}$$

$$x_4^s = \max(s_1, s_2, \ldots, s_N) \tag{5d}$$

$$x_5^s = \sum_{i=1}^{N} \mathbf{1}(s_i) \times 2^{i-1} \tag{5e}$$

QoE Modeling for Stalling

Let x denote the input feature vector, the proposed QoE model F(x) is a weighted sum of M base learners (i.e., decision trees) hm(x):

$$F(x) = \sum_{i=1}^{M} \gamma_m h_m(x)$$

The multiplier γ_m and the base learner $h_m(x)$ are the model's parameters and are learned iteratively using gradient tree boosting learning method [18].

$$QoE^{Pred} = F_M(x)$$

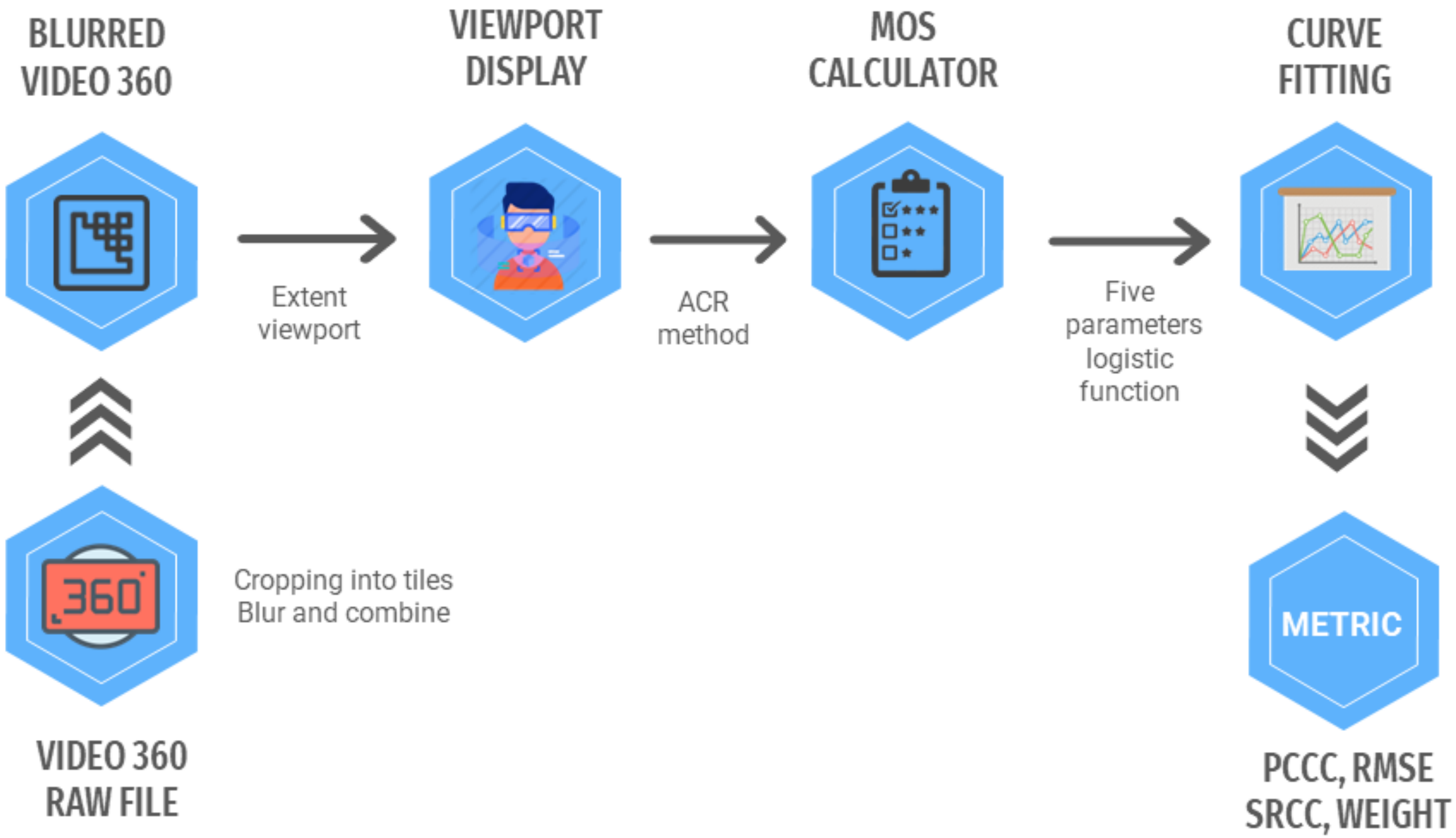
Performance Evaluation of the Proposed QoE Models

- The constructed QoE database is randomly split into a training set containing 80% of the samples and a test set containing the remaining 20% of the samples.

- The performance of the QoE prediction models is measured in Pearson Linear Correlation Coefficient (PLCC), Spearman's Rank Order Correlation Coefficient (SROCC), and Root Mean Squared Error (RMSE).

Point Cloud Video	QoE Model #1 (Temporal Quality Variation)						QoE Model #2 (Stalling)					
	Training Set			Test Set			Training Set			Test Set		
	PLCC	SROCC	RMSE	PLCC	SROCC	RMSE	PLCC	SROCC	RMSE	PLCC	SROCC	RMSE
Longdress	0.98	0.98	0.24	0.97	0.97	0.26	0.99	0.99	0.09	0.95	0.94	0.24
Loot	0.97	0.94	0.25	0.97	0.92	0.25	0.99	0.99	0.08	0.95	0.93	0.22
RedandBlack	0.98	0.98	0.20	0.97	0.97	0.25	0.99	0.98	0.11	0.95	0.95	0.26
Solider	0.98	0.97	0.30	0.97	0.96	0.32	0.97	0.96	0.16	0.93	0.88	0.31
All	0.97	0.96	0.25	0.96	0.94	0.27	0.99	0.99	0.11	0.94	0.94	0.26

Example 3: Retina-Based QoE Modeling



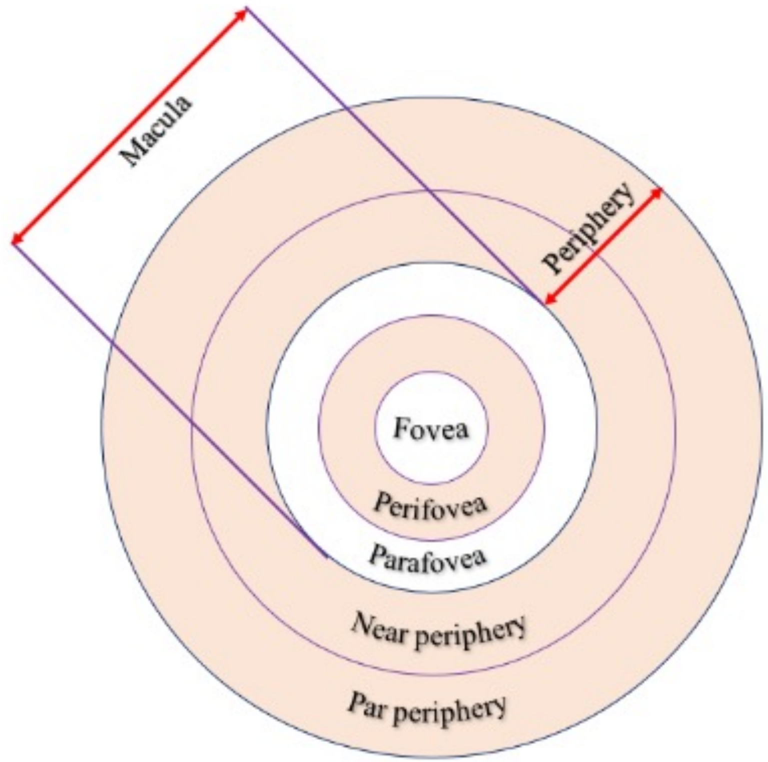
Example 3: Retina-Based QoE Modeling

Angular deviation from the region center

Region	Z_1	Z_2	Z_3	Z_4	Z_5
Deviation	0, 2.5	2.5, 4	4, 9	9, 30	30, ∞

✓ A new QoE metrics for 360-degree video

✓ To find a new mapping function to predict QoE score based on QoE metrics.



The retina is divided into five regions

Example 3: Retina-Based QoE Modeling

$$WZUQI = \sum_{k=1}^{K} w_k UQI_k$$

Chỉ số (UQI) được định nghĩa như sau:

$$UQI = \frac{1}{M}\left[\frac{4\sigma_{xy}\overline{x}\overline{y}}{(\sigma_x^2+\sigma_y^2)\times[(\overline{x})^2+(\overline{y})^2]}\right]$$

M: The number of pixels in each image.

σ_x: Correlation loss value..

σ_y: Luminance distortion..

σ_{xy}: Correlation distortion..

x: is computed as the average of$\{x_i \mid i = 1, 2, 3, \ldots, N\}$

y: is computed as the average of$\{y_i \mid i = 1, 2, 3, \ldots, N\}$

Example 3: Retina-Based QoE Modeling

MOS Assessment Criteria Table

MOS	Quality score
1	Very blurry / very uncomfortable
2	Blurry and uncomfortable
3	Slightly blurry and slightly uncomfortable
4	Slightly blurry but not uncomfortable
5	Very good

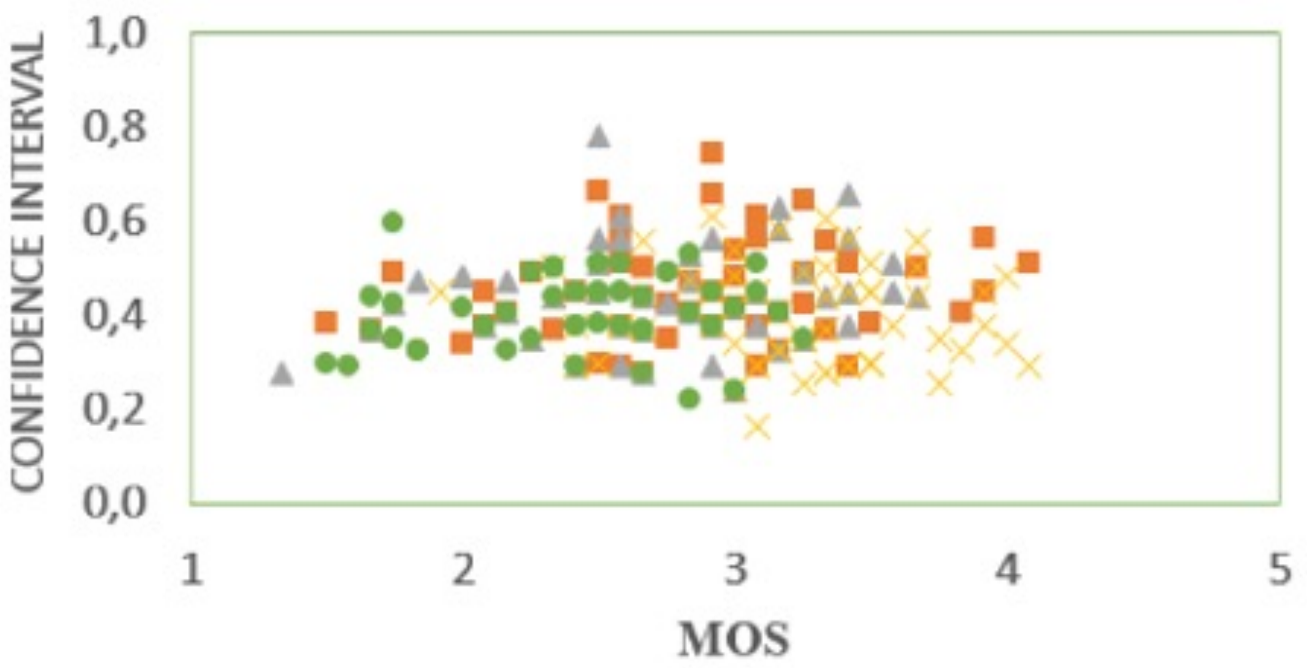
$$\widehat{MOS} = \alpha_1\left(\frac{1}{2} + \frac{1}{1+e^{\alpha_2(WZUQI-\alpha_3)}}\right) + \alpha_4 WZUQI + \alpha_5$$

Where

α_i=1,2,3,4,5 are parameters that are precomputed in advance.

- ✓ A five-parameter logistic function is used to predict MOS (Mean Opinion Score) values from the WZUQI value.

Evaluation of a New QoE Modeling Approach



95% confidence interval of 240 MOS value

Evaluation of a New QoE Modeling Approach

Experimental setting: 360-degree videos used in the experiment

(a) Diving_1 (b) Diving_2 (c) Paris_1 (d) Paris_2

(e) Rollercoaster_1 (f) Rollercoaster_2 (g) Venice_1 (h) Venice_2

Evaluation of a New QoE Modeling Approach

Characteristics of the four 360-degree videos used for the experiments

VIDEOS	YOUTUBE ID	MÔ TẢ NỘI DUNG VIDEO	CHUYỂN ĐỘNG HOẠT ĐỘNG
Diving	2OzlksZBTiA	Ban ngày, cảnh biển	Thấp
Paris	EkshFcLESPU	Các điểm tham quan ở Paris, ban ngày, du khách tản bộ	Thấp
RollerCoaster	8lsB-P8nGSM	Tàu lượn siêu tốc, ngoài trời, ban ngày	Cao
Venice	s-AJRFQuAtE	Tòa nhà ở Venice, ngoài trời, đèn ngủ	Thấp

Evaluation of a New QoE Modeling Approach

Two performance metrics are considered:
Pearson Correlation Coefficient (PCC), Root Mean Square Error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(\widehat{MOS}_i - MOS_i)^2}{N}}$$

Trong đó:

- N: là số lượng bức ảnh;
- $\widehat{MOS}_i$: là giá trị dự đoán MOS của bức ảnh i;
- MOS_i: là giá trị MOS thực tế của ảnh i;
- Cuối cùng, sau khi fit với MOS, ta thu được kết quả các giá trị trọng số của từng vùng trong ảnh và có được phép đo $ZWUQI$.

$$PCC = \frac{\sum_{i=1}^{N}(M_i - \overline{M})(MOS_i - \overline{MOS}_i)}{\sqrt{\sum_{i=1}^{N}(M_i - \overline{M})^2}\sqrt{\sum_{i=1}^{N}(MOS_i - \overline{MOS}_i)^2}},$$

trong đó $\overline{M}$, và $\overline{MOS}$ được tính như sau:

$$\overline{M} = \frac{1}{N}\sum_{i=1}^{N} M_i; \qquad \overline{MOS} = \frac{1}{N}\sum_{i=1}^{N} MOS_i,$$

với:

- N: là số lượng bức ảnh;
- M_i: là giá trị tỷ lệ tín hiệu-tỷ lệ nhiễu của video theo trọng số của ảnh thứ i.

Evaluation of a New QoE Modeling Approach

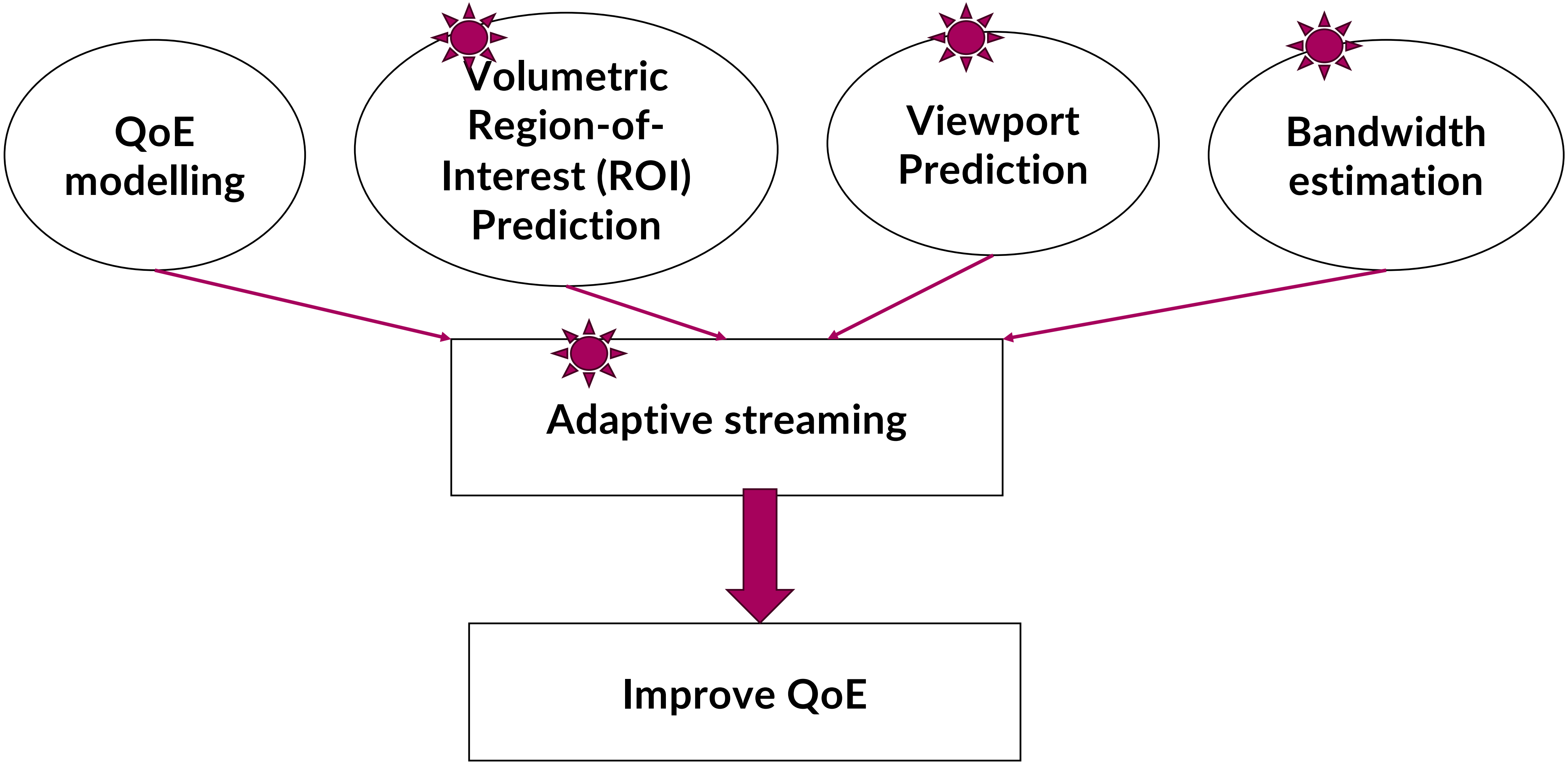
Bảng 2.4 Giá trị PCC của các chỉ số đánh giá chất lượng khách quan với từng video.

	Videos			
Metric	**Video #1**	**Video #2**	**Video #3**	**Video #4**
MSE [40]	0.620	0.270	0.135	0.793
SSIM [26]	0.019	0.096	0.132	0.449
MS-SSIM [27]	0.002	0.098	0.125	0.419
UQI [28]	0.212	0.409	0.410	0.790
VIFp [30]	0.000	0.615	0.436	0.743
VIF [30]	0.012	0.210	0.282	0.101
NQM [31]	0.214	0.093	0.174	0.316
IW-PSNR [32]	0.096	0.121	0.340	0.066
IW-SSIM [32]	0.021	0.142	0.260	0.087
FSIM [33]	0.295	0.114	0.077	0.439
FSIMc [33]	0.262	0.154	0.069	0.500
SR-SIM [35]	0.239	0.140	0.253	0.370
RFSIM [34]	0.089	0.007	0.201	0.325
ADD-SSIM [36]	0.212	0.158	0.080	0.391
PSIM [37]	0.319	0.203	0.400	0.786
WSNR [31]	0.236	0.170	0.229	0.337
FMSE [39]	0.246	0.103	0.126	0.463
FPSNR [38]	0.245	0.095	0.092	0.340
F-SSIM [29]	0.232	0.177	0.087	0.212
GSIM [41]	0.221	0.166	0.083	0.199
PSNR [12]	0.318	0.251	0.063	0.450
ZWF [13]	0.244	0.217	0.000	0.791
WZUQI	**0.888**	**0.808**	**0.844**	**0.885**

Bảng 2.5 Giá trị RMSE của các chỉ số đánh giá chất lượng khách quan với từng video.

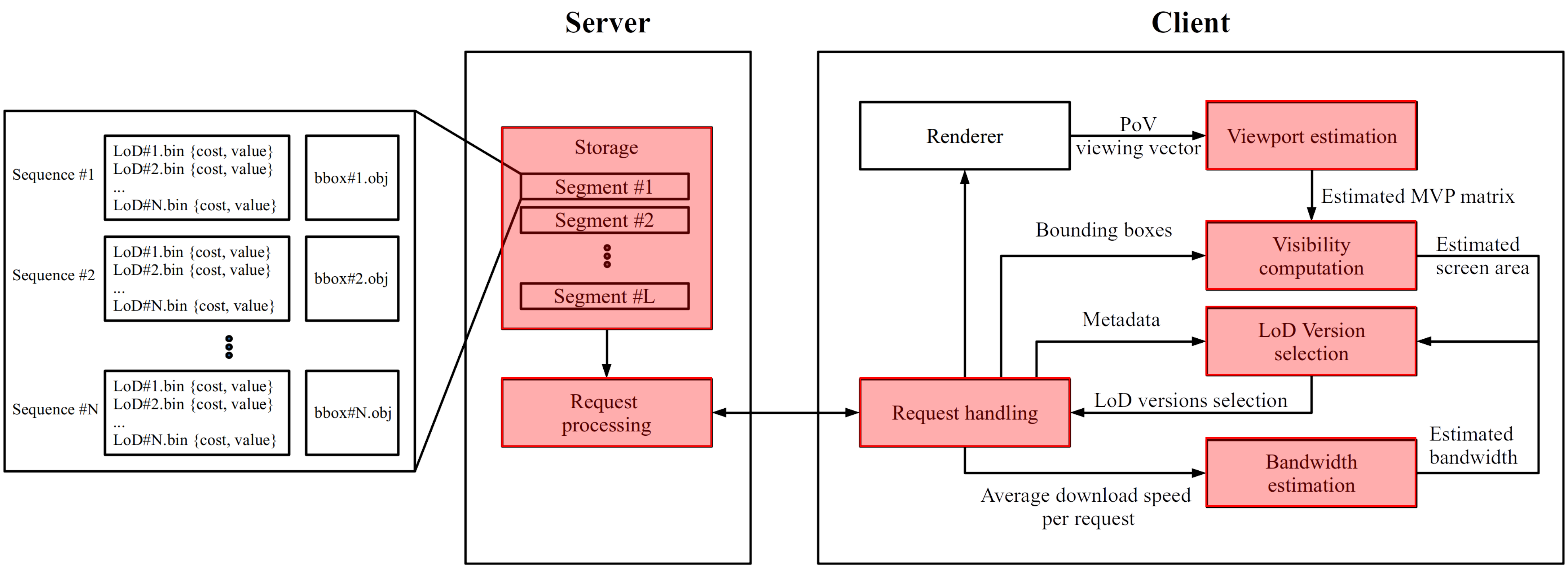
	Videos			
Metric	**Video #1**	**Video #2**	**Video #3**	**Video #4**
MSE [40]	0.375	0.395	0.505	0.295
SSIM [26]	0.478	0.408	0.505	0.433
MS-SSIM [27]	0.478	0.408	0.505	0.440
UQI [28]	0.467	0.374	0.464	0.297
VIFp [30]	0.478	0.323	0.458	0.325
VIF [30]	0.478	0.401	0.489	0.482
NQM [31]	0.467	0.408	0.502	0.460
IW-PSNR [32]	0.476	0.407	0.479	0.484
IW-SSIM [32]	0.478	0.406	0.492	0.483
FSIM [33]	0.457	0.407	0.508	0.436
FSIMc [33]	0.461	0.405	0.508	0.420
SR-SIM [35]	0.464	0.406	0.493	0.451
RFSIM [34]	0.476	0.410	0.499	0.459
ADD-SSIM [36]	0.467	0.405	0.508	0.446
PSIM [37]	0.453	0.401	0.467	0.299
WSNR [31]	0.465	0.404	0.496	0.457
FMSE [39]	0.463	0.408	0.505	0.430
FPSNR [38]	0.463	0.408	0.507	0.456
F-SSIM [29]	0.465	0.404	0.507	0.475
GSIM [41]	0.466	0.404	0.508	0.475
PSNR [12]	0.453	0.397	0.508	0.433
ZWF [13]	0.469	0.401	0.716	0.384
WZUQI	**0.348**	**0.301**	**0.362**	**0.344**

Remember this diagram before we start

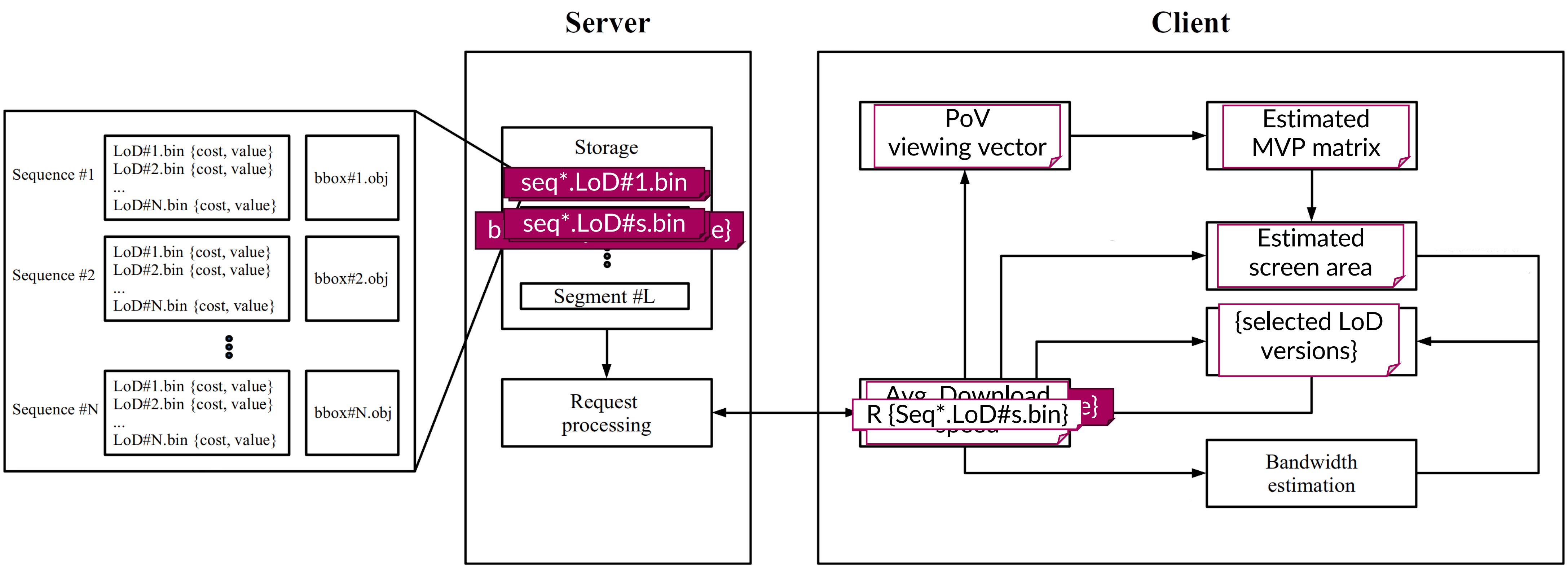


VOLUMETRIC ADAPTIVE STREAMING

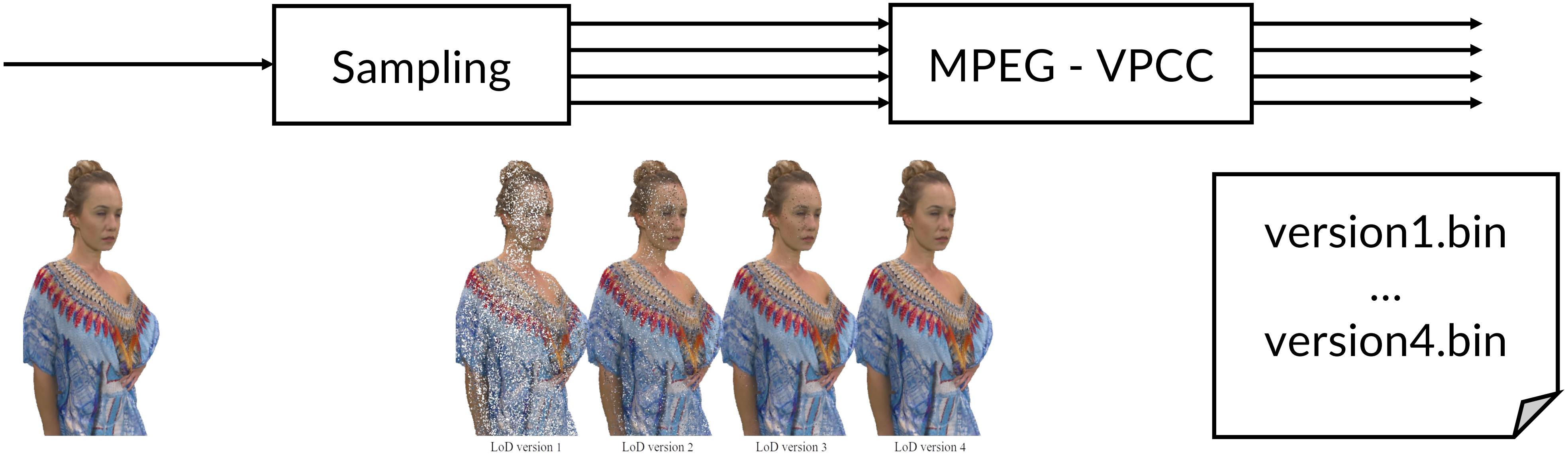
SYSTEM ARCHITECTURE FOR VOLUMETRIC STREAMING



Proposal: System Architecture



Proposal: Storage

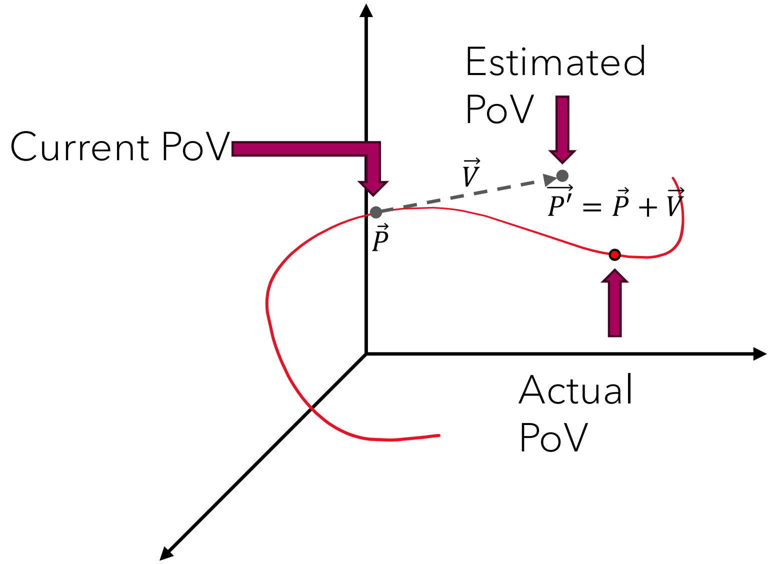


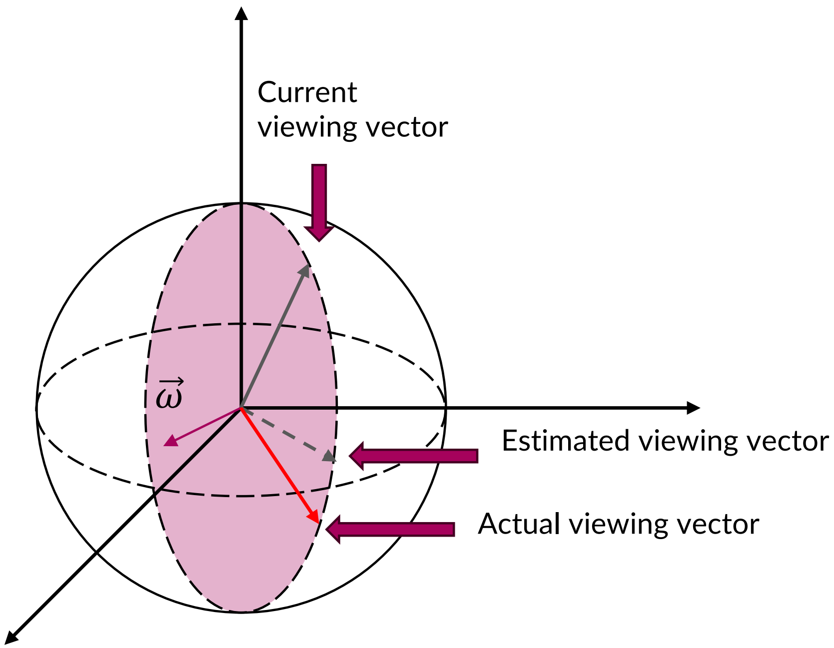
Original version

Compressed data

Viewport estimation

$$\hat{P}(t_e) = P(t_c) + \frac{P(t_c) - P(t_c - \Delta t)}{\Delta t} \cdot (t_e - t_c)$$

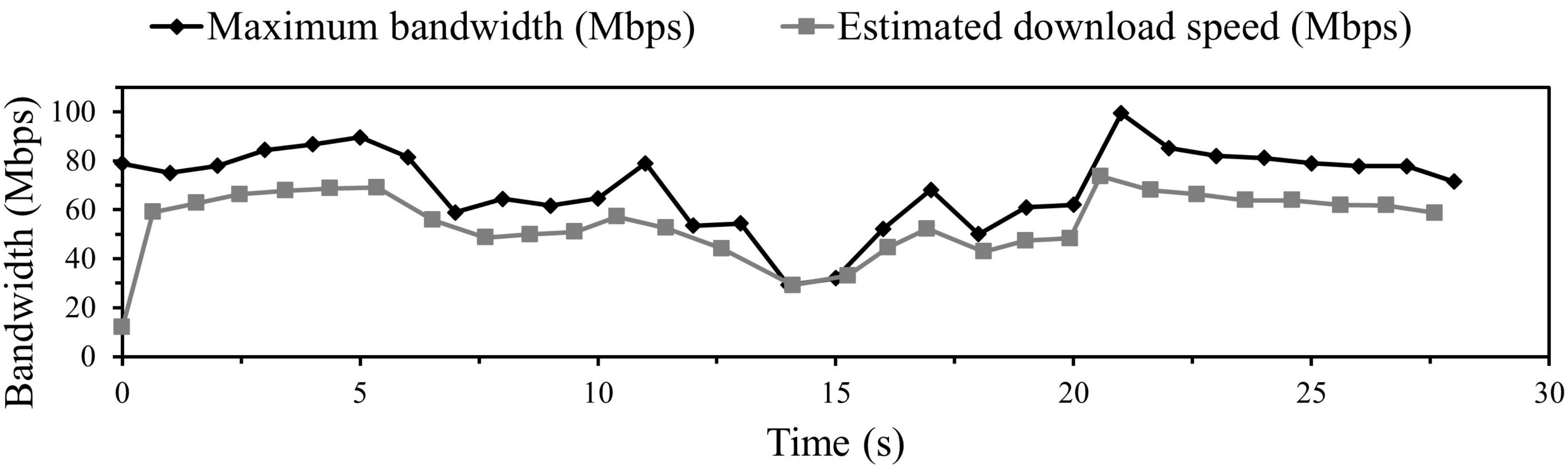




$$\hat{V}(t_e) = R_{\hat{\omega}}\left(\frac{\Delta\theta}{\Delta t} \cdot (t_e - t_c)\right) * V(t_c)$$

Bandwidth estimation

$$R^a = \frac{M}{\sum_{i=1}^{M} \frac{1}{R_i}}$$



Version selection

Optimize

$$OV = \sum_{m=1}^{M} w_m \cdot V(m, n_m)$$

with:

$$\sum_{m=1}^{M} C(m, n_m) \leq R^a$$

$$w_m = \frac{a_m}{\sum_{i=1}^{M} a_i}$$

Where:

M : Number of point clouds

N : Number of versions per point cloud

n_m: Selected LoD version for the point cloud m

V(m, n) : “value” of version *n* of Point cloud *m*.

C(m, n) : “cost” of version *n* of Point cloud *m*. (bitrate sau mã hóa nguồn).

R^a : Available bandwidth of the client.

a_m : Estimated screen area of Point cloud *m*.

Version adaptation

Dynamic Programming based solution

Maximize

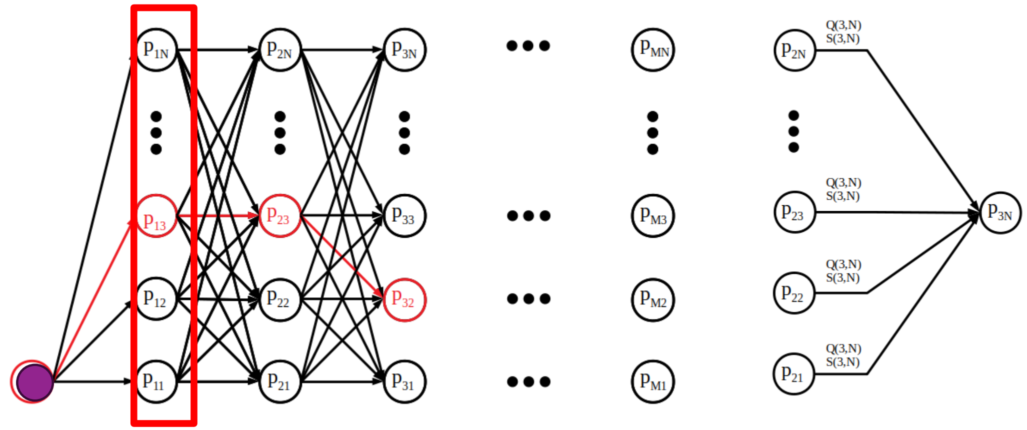
$$OV = \sum_{m=1}^{M} w_m \cdot V(m, n_m)$$

Với:

$$\sum_{m=1}^{M} C(m, n_m) \leq R^a$$

$$V(m,n) = PSNR(m,n)$$

$$= 20 \cdot log \frac{{d_m}^{bb}}{RMS(m,n)}$$



N versions for point cloud ***m***

|V| = N * M
|E| = N ^ M

Visibility area Computation

Version adaptation:

Dynamic Programming based solution

Maximize

$$OV = \sum_{m=1}^{M} w_m \cdot V(m, nm)$$

with:

$$\sum_{m=1}^{M} C(m, nm) \le R^a$$

$$V(m,n) = PSNR(m,n) = 20 \cdot log \frac{{d_m}^{bb}}{RMS(m,n)}$$

Algorithm 1: *Dynamic Programming-based Solution*

```
Input  : {w_m}, {C(m,n)}, {V(m,n)}, R^a
Output : Optimal LoD versions selection χ_s
χ_s ← {}, χ ← {}, V̄ ← 0;
initialization(G, R^a);
pulse(0, 1, χ, R^a, V̄, χ_s);
return χ_s;
Function pulse(m, n, χ, R^a, V̄, χ_s):
    if checkDominance(p_mn, χ) == true OR
      checkFeasibility(p_mn, χ, R^a) == false OR
      checkBounds(p_mn, χ, V̄) == false then
        return;
    end
    χ' ← χ ∪ n;
    if m == M then
        χ_s ← χ';
        V̄ ← OV(χ_s);
        return;
    end
    For k ← 1 to N do
        pulse(m + 1, k, χ', R^a, V̄, χ_s);
    end
return
```

Recursive BFS

→ O(|V|+|E|) = O(N^M)

Version selection

Version adaptation:

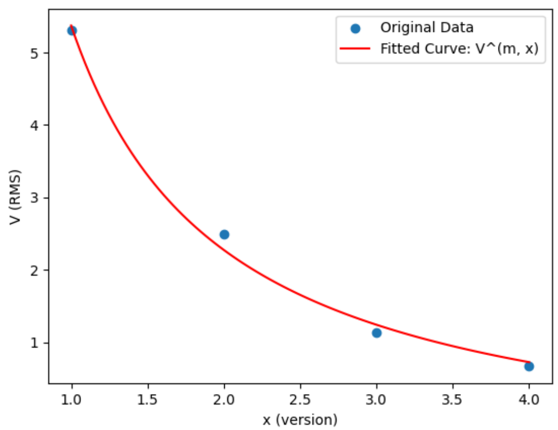
Lagrange Multiplier - based solution

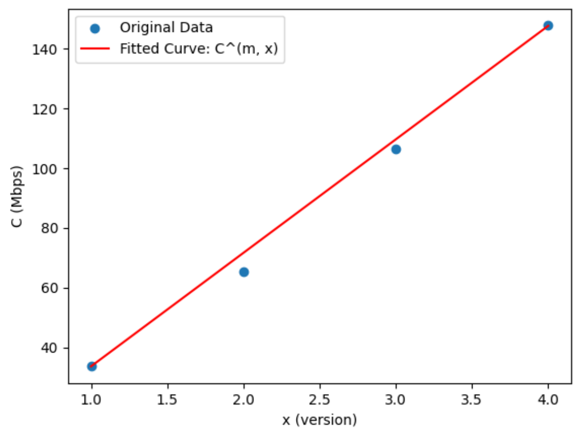
Minimize:

$$OV = \sum_{m=1}^{M} w_m \cdot V(m, n_m)$$

Với:

$$\sum_{m=1}^{M} C(m, n_m) \leq R^a$$





$$V(m, n) = RMS(m, n)$$

$$\hat{V}(m, x) = \frac{A_m}{x} + Bm$$

$$\hat{C}(m, x) = Cm \cdot x + Dm$$

Version selection

Version adaptation

Lagrange Multiplier - based solution

Minimize

$$OV = \sum_{m=1}^{M} w_m \cdot V(m, n_m)$$

Với:

$$\sum_{m=1}^{M} C(m, n_m) \leq R^a$$

$$\hat{V}(m,x) = \frac{A_m}{x} + Bm \ \ and \ \ \hat{C}(m,x) = Cm \cdot x + Dm$$

$$1 \leq \ xm \ \leq N$$

Algorithm 2: *Lagrange Multiplier-based Solution*

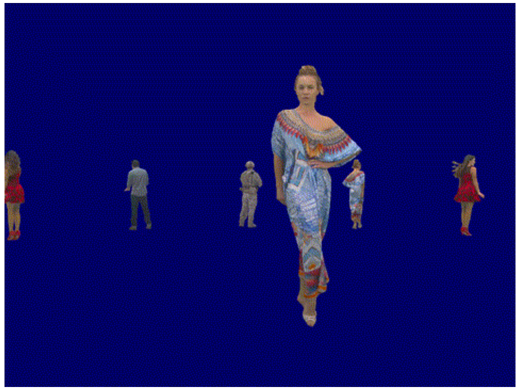
```
Input  : {C(m,n)}, {w_m}, {A_m}, {B_m}, {C_m},
         {D_m}, R^a
Output: LoD versions selection χ_s
χ_s ← {}, Ω ← {m, m ← 1 to M};
LagrangeSelect(Ω, χ_s, R^a);
return χ_s;
Function LagrangeSelect(Ω, χ_s, R^a):
    do
        TouchBound ← false;
        For m ∈ Ω do
            update(χ_s[m]);
            if χ_s[m] < 1 OR χ_s[m] > N then
                χ_s[m] ← (χ_s[m] < 1) ? 1 : N;
                R^a ← R^a − Ĉ(m, χ_s[m]);
                Ω ← Ω \ m;
                TouchBound ← true;
            end
        end
    while TouchBound == true;
    χ_s ← RoundHalfUp(χ_s), R^u ← 0;
    For m ← 1 to M do
        R^u ← R^u + C(m, χ_s[m]);
    end
    if R^u > R^a then
        χ_s ← int(χ_s);
    end
return
```

O(M^2)

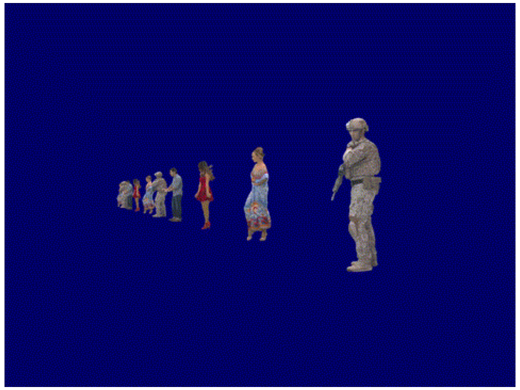
Performance Evaluation: Experimental Environment

Reference Methods:

- **Equal:** Evenly distributes the available network resources among the Point Clouds within the viewport.
- **Hybrid [11]:** Determines the quality of each Point Cloud heuristically based on its position in the ranking list of projected screen area.

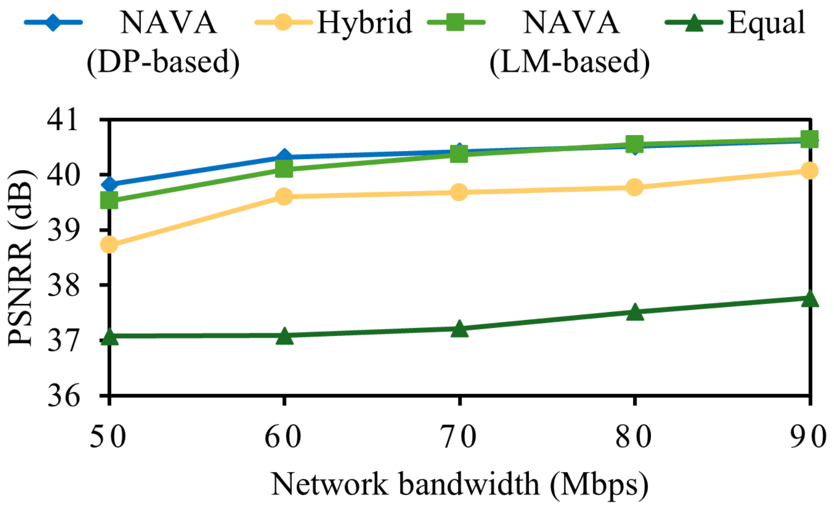


Scene 1

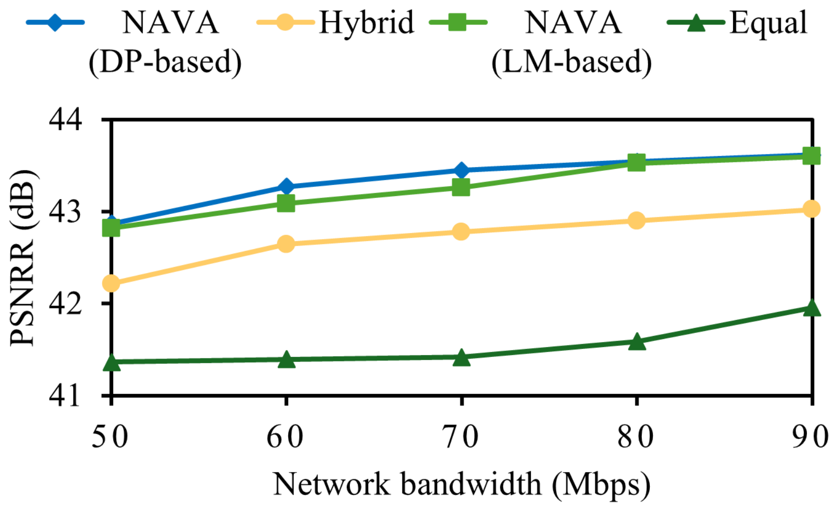


Scene 2

Performance Evaluation: Fixed network (Constant bandwidth)

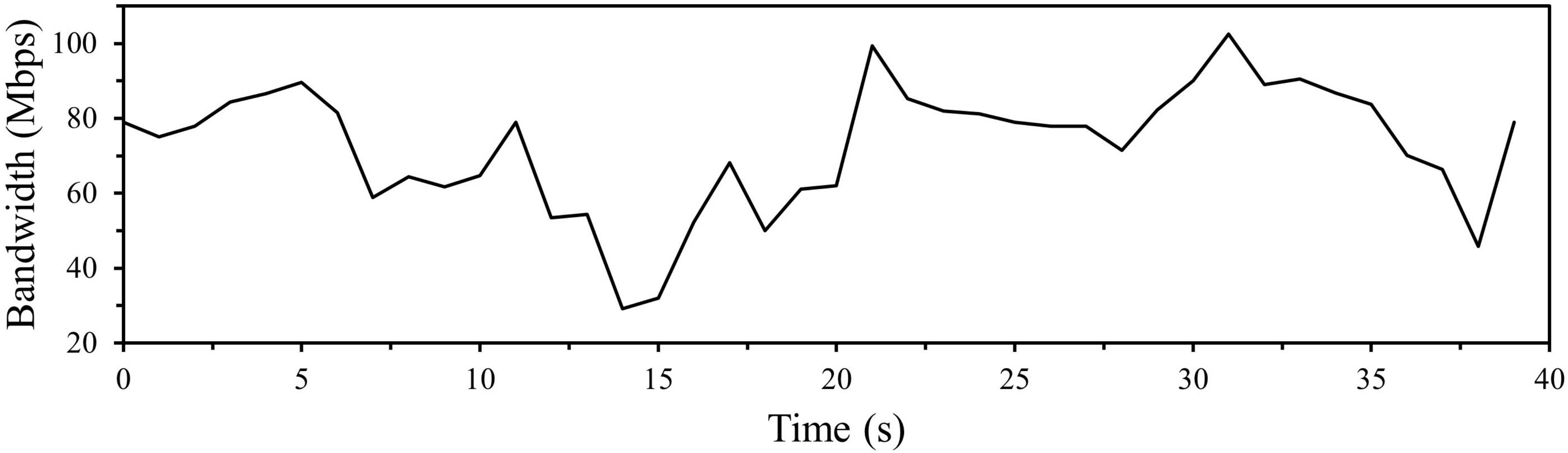


Scene 1



Scene 2

Performance Evaluation: Mobile Network



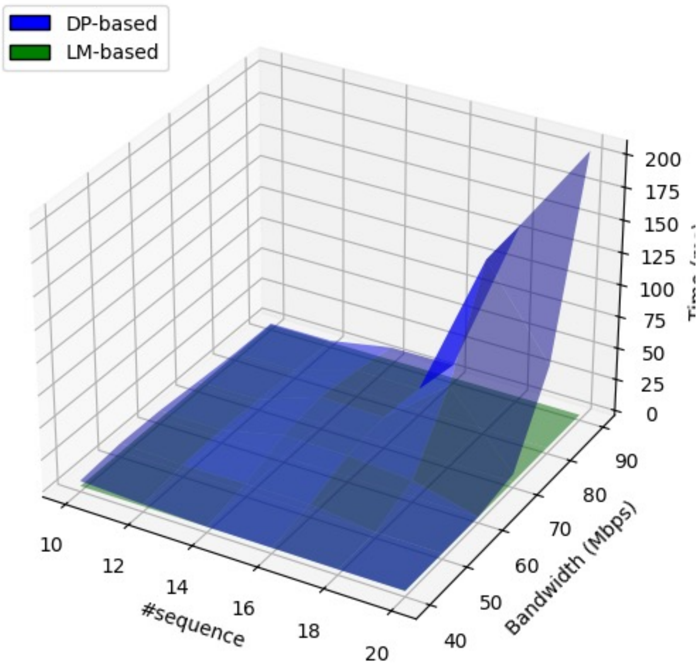
Performance Evaluation: Mobile Network (Fluctuating bandwidth)

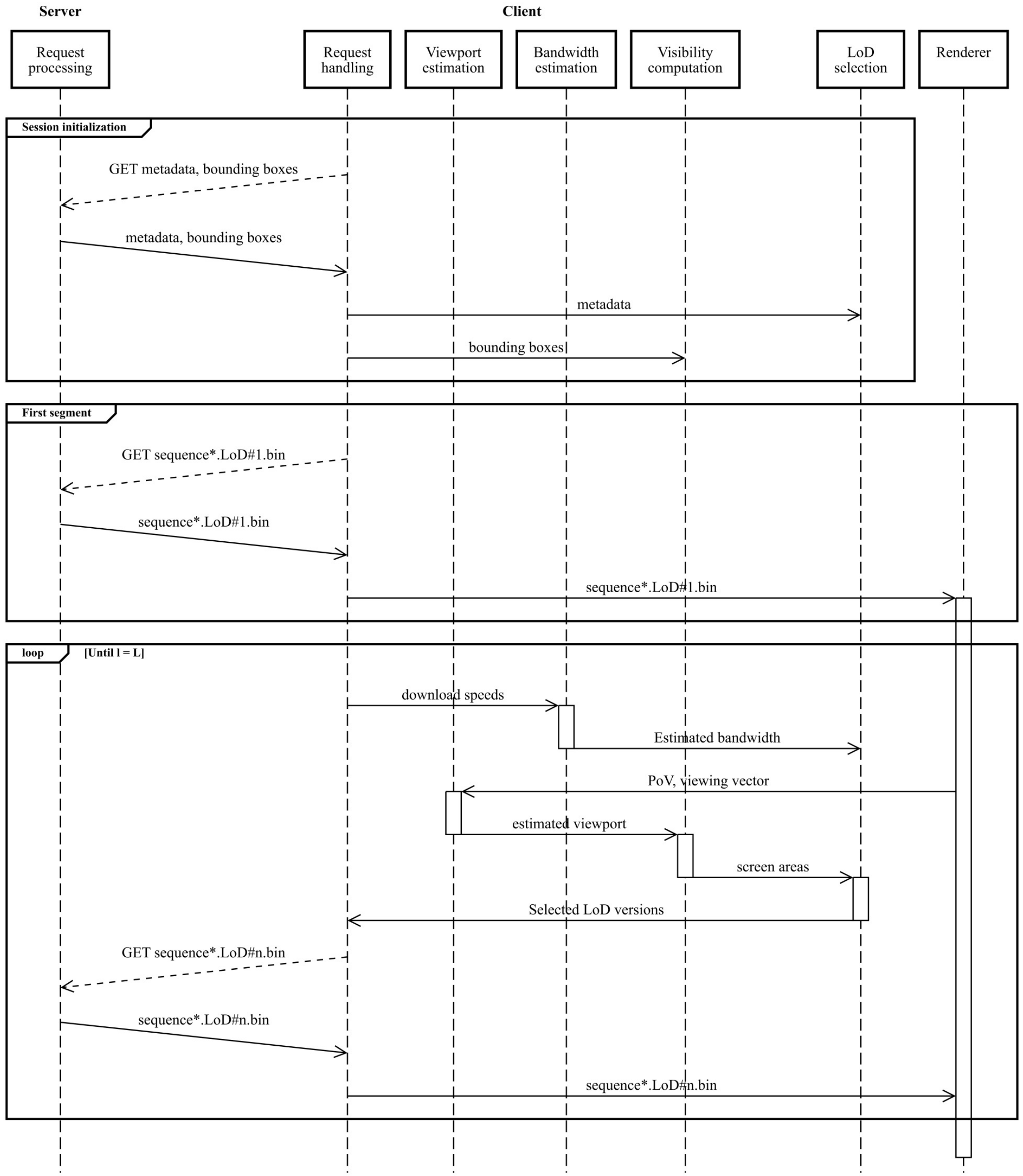
Method	Avg PSNR (dB)	Avg #Stall	Avg Stall Duration (s)
NAVA (DP-based)	44.22	8.5	1.1675
NAVA (LM-based)	44.17	7.25	1.0875
Hybrid	43.72	13	1.5475
Equal	42.18	0	0

Performance Evaluation: Processing time

Average processing time (milliseconds) of the DP-based / LM-based solution:

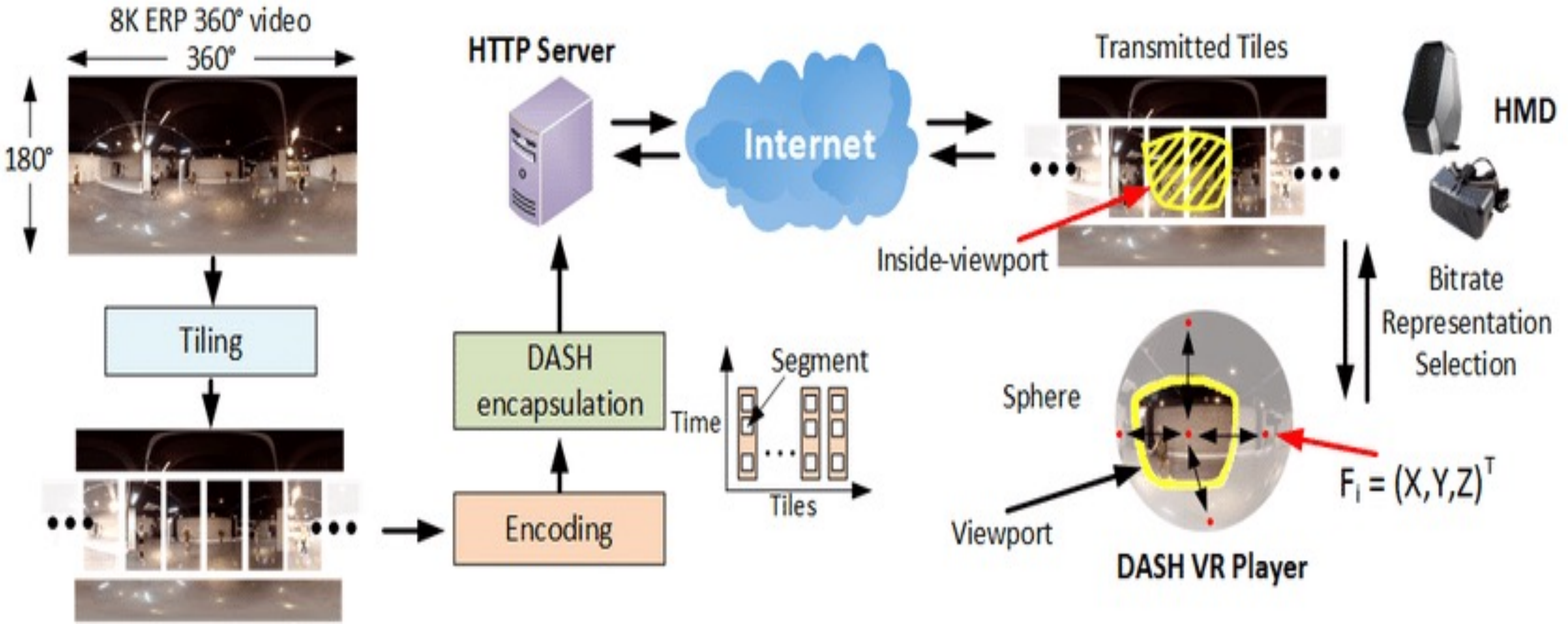
BW \ #	10 sequence	12 sequence	14 sequence	16 sequence	18 sequence	20 sequence
40Mbps	4.5 / 0.40	2.08 / 0.47	0.59 / 0.54	0.59 / 0.58	0.37 / 0.63	0.69 / 0.52
50Mbps	7.31 / 0.39	5.72 / 0.46	6.52 / 0.52	6.15 / 0.55	0.71 / 0.57	0.65 / 0.51
60Mbps	6.85 / 0.39	6.49 / 0.46	17.86 / 0.52	17.30 / 0.55	13.39 / 0.57	0.68 / 0.51
70Mbps	4.32 / 0.37	4.70 / 0.41	21.94 / 0.47	20.92 / 0.49	60.93 / 0.54	6.75 / 0.46
80Mbps	2.07 / 0.33	2.72 / 0.38	17.60 / 0.43	17.27 / 0.46	133.75 / 0.48	66.13 / 0.45
90Mbps	1.31 / 0.28	1.56 / 0.35	10.52 / 0.39	10.96 / 0.43	136.93 / 0.46	205.94 / 0.41




Server
Client
Request processing
Request handling
Viewport estimation
Bandwidth estimation
Visibility computation
LoD selection
Renderer
Session initialization
GET metadata, bounding boxes
metadata, bounding boxes
metadata
bounding boxes
First segment
GET sequence*.LoD#1.bin
sequence*.LoD#1.bin
sequence*.LoD#1.bin
loop
[Until l = L]
download speeds
Estimated bandwidth
PoV, viewing vector
estimated viewport
screen areas
Selected LoD versions
GET sequence*.LoD#n.bin
sequence*.LoD#n.bin
sequence*.LoD#n.bin

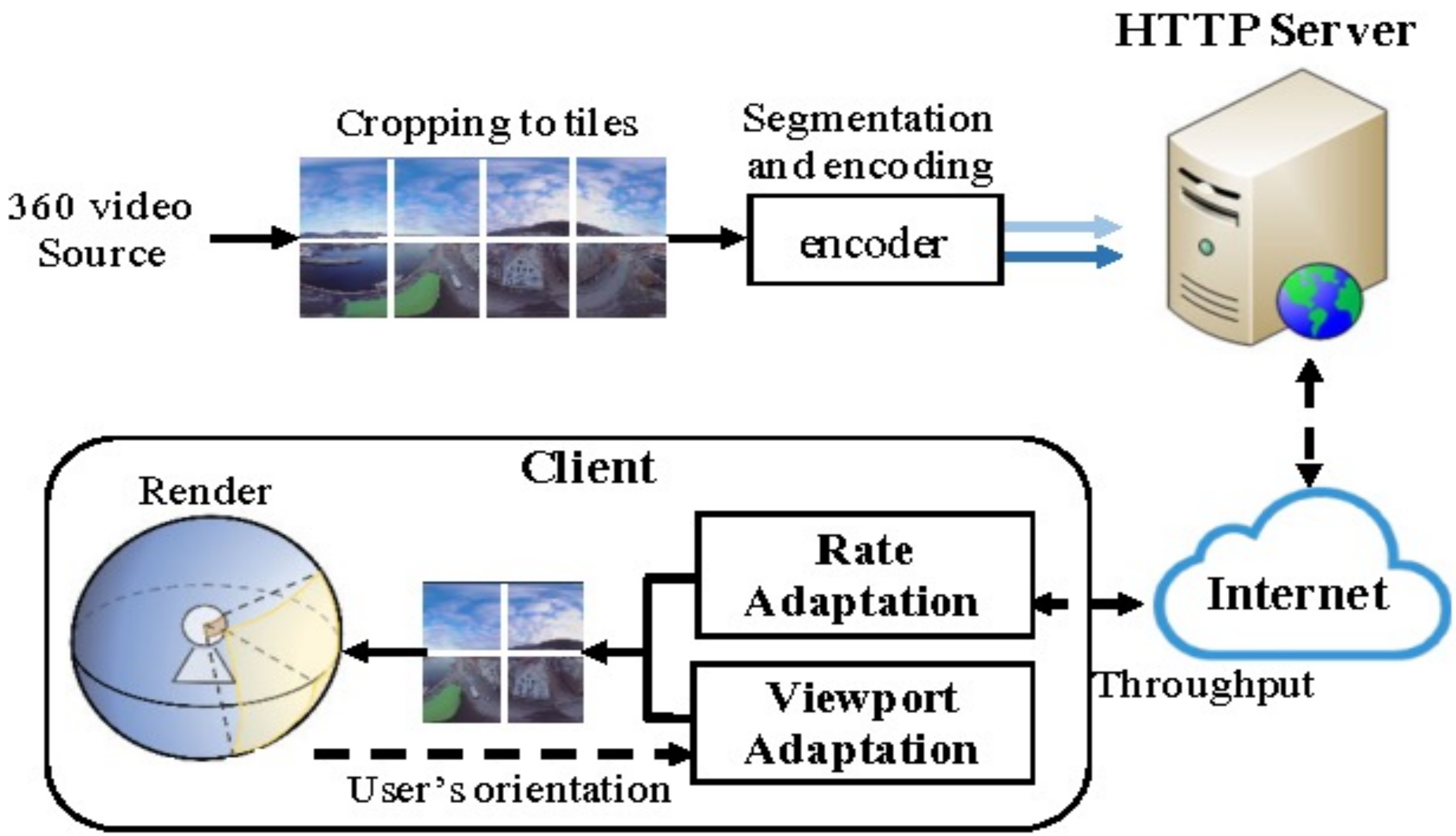
ADAPTIVE 360° VIDEO STREAMING

CURENT STREAMING METHOD



Source: https://www.semanticscholar.org/

VIEWPORT ADAPTIVE STREAMING



Source: https://slideplayer.com/slide/13895227/

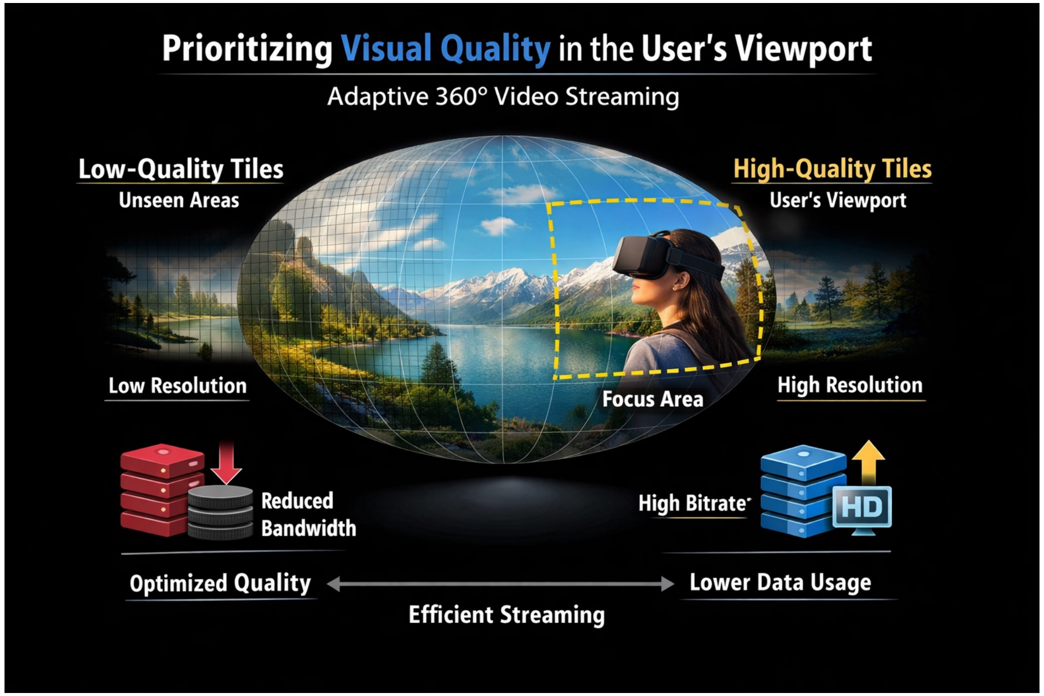
What to adapt while streaming

Prioritizes visual quality in the user’s viewport

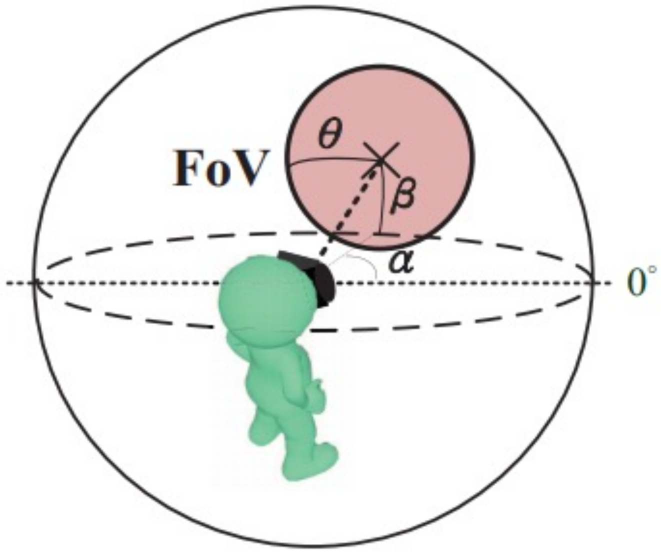
Adaptive systems dynamically:

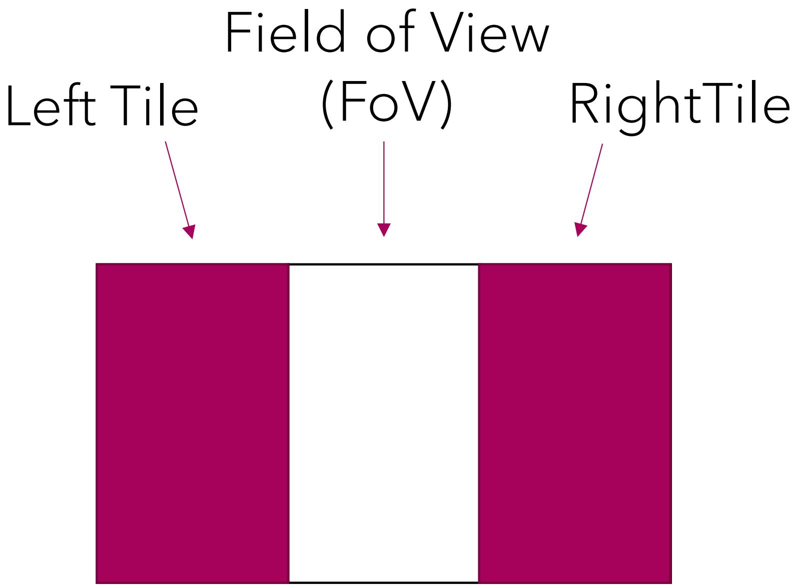
- Stream **high-quality video tiles** for the current (or predicted) viewport
- Stream **lower-quality tiles** for non-viewed regions

This preserves **perceived quality** while reducing bandwidth usage.

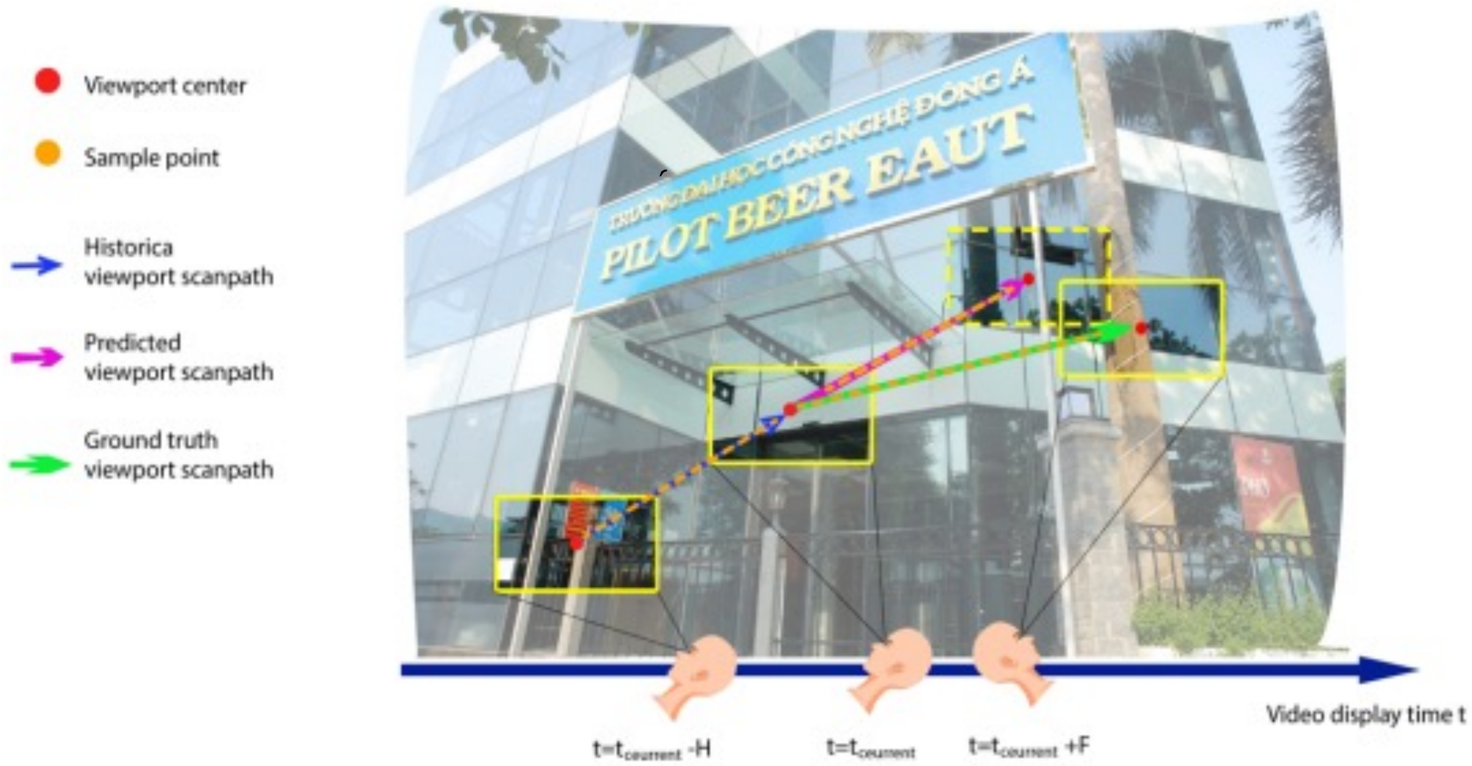


VIEWPORT ADAPTIVE STREAMING



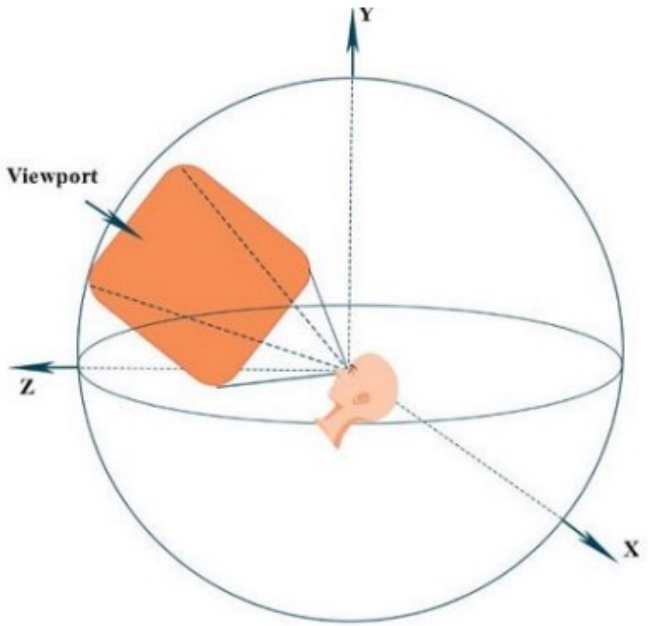


Viewport Prediction

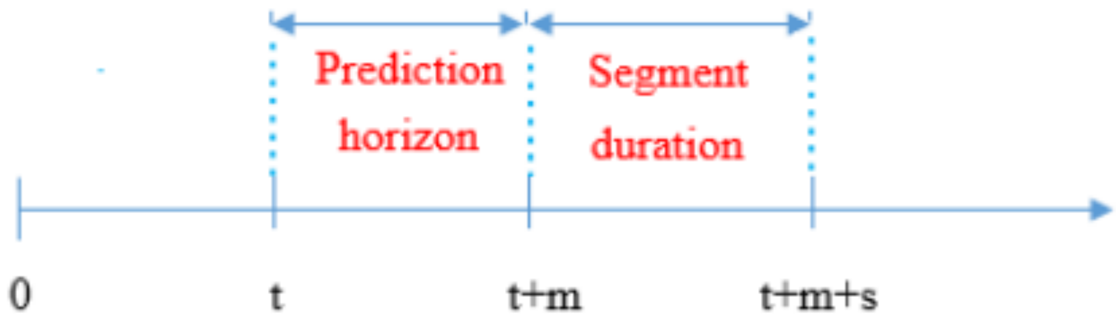


Comparing the viewport scanning speed over the past H seconds to predict the viewport scanning speed in the next F seconds.

Temporal Motion Behavior

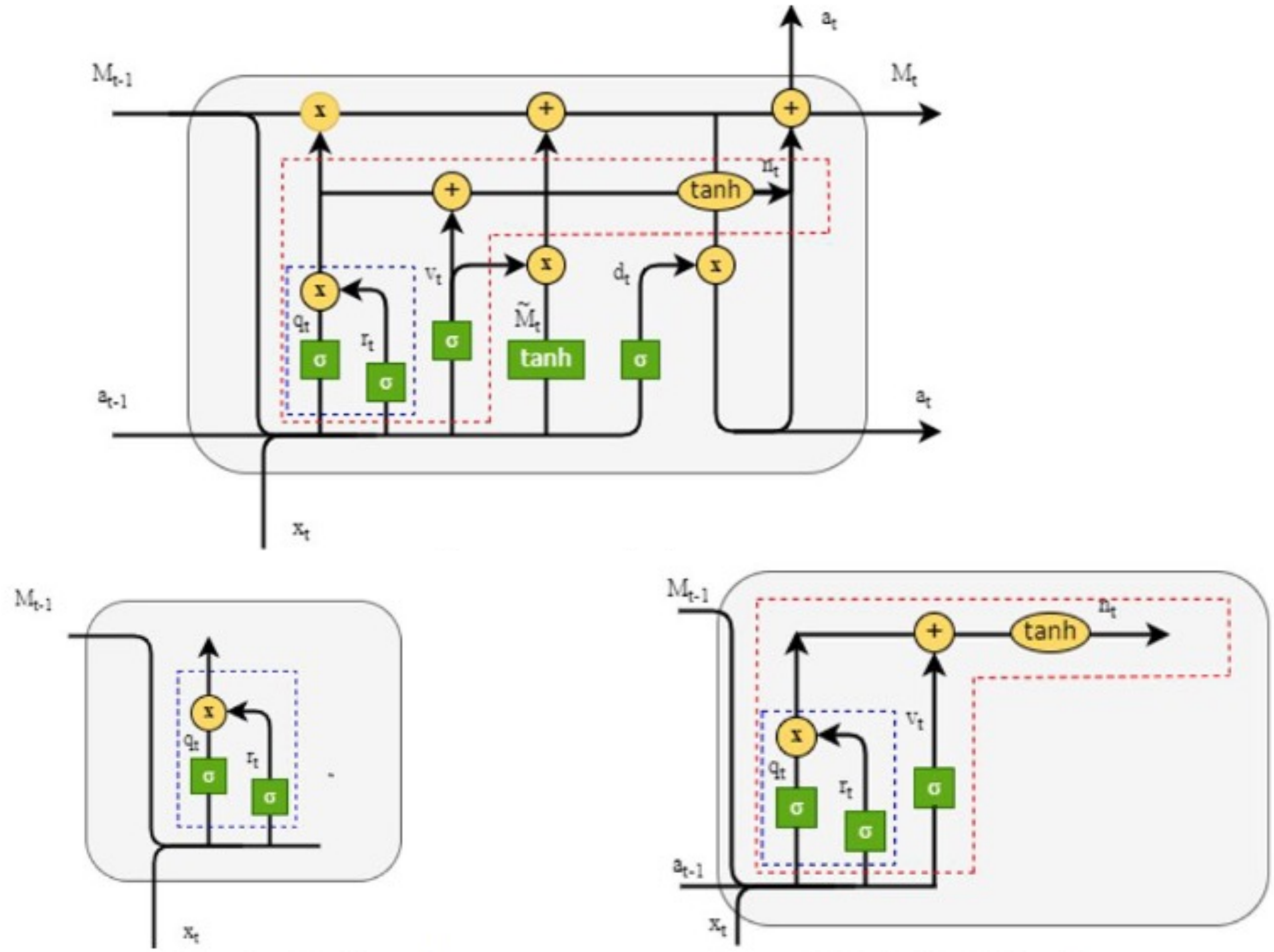


The user's viewport at time *t*



Problem Formulation of the Viewport Prediction

Viewport position Prediction: Approach based on head movement



(a) Bước 01: Thêm cổng r_t

(b) Bước 02: Thêm cổng n_t

Algorithm 1: *Ước tính viewport*

Input: $q_t, v_t, r_t, d_t, \tilde{M}_t$

Output: $\{M_t, a_t\}$

for $t = 1$ ***to*** N **do**

- Calculate $v_t = \sigma(v_t)$
- Calculate $q_t = \sigma(q_t)$
- Calculate $\tilde{M}_t = \tanh(\tilde{M}_t)$
- Calculate $d_t = \sigma(d_t)$
- Calculate $r_t = \sigma(v_t + M_{t-1})$
- Calculate $n_t = \tanh(v_t + (r_t \otimes q_t))$
- $M_t = ((r_t * q_t) \otimes M_{t-1}) + (v_t \otimes M_t)$
- $a_t = n_t + d_t * \tanh(M_t)$

end

return M_t, a_t

Supplemented with reset gates- r_t, n_t to control how many previous states are retained.

Evaluation of the head-movement based Viewport position Prediction

Values		Proposed	LAST	LINEAR	GRU	LSTM
Position of viewport 1	Accuracy	94.28	84.64	76.58	85.33	75.76
Position of viewport 2	Accuracy	94.04	84.38	80.93	84.38	75.51

Achieving an improvement of 10% to 19.70% compared to existing methods.

Bảng 3.3 Độ chính xác (%) của HEVEL với các phương pháp tham chiếu

Videos	GRU	RNN	GLVP	HEVEL
BAR	61.65	62.12	74.21	**84.54**
Ocean	60.52	71.30	73.21	**85.86**
Po. Riverside	88.65	87.23	90.11	**91.62**
Sofa	74.21	58.16	74.21	**85.86**
Turtle	72.57	71.62	74.21	**75.58**
Average	**70.09**	**68.62**	**76.64**	**84.54**

Bảng 3.4 RMSE của HEVEL với các phương pháp tham chiếu

Videos	GRU	RNN	GLVP	HEVEL
BAR	0.266	0.264	0.221	**0.194**
Ocean	0.271	0.230	0.224	**0.191**
Po. Riverside	0.185	0.188	0.182	**0.179**
Sofa	0.221	0.282	0.221	**0.191**
Turtle	0.226	0.229	0.221	**0.217**
Average	**0.234**	**0.239**	**0.214**	**0.194**

Algorithm 2: *Viewport Estimation*

Input: c_{t-1}, h_{t-1}, x_t

Output: c_t, h_t, y_t

for $t = 1$ ***to*** N **do**

Calculate $i_{\alpha t} = \sigma(W_{i\alpha} \otimes (h_{t-1}, x_t) + b_{i\alpha}$

Calculate $i_{\beta t} = \tanh(W_{i\beta} \otimes (h_{t-1}, x_t) + b_{i\beta}$

Calculate $i_t = i_{\alpha t} * i_{\beta t}$

Calculate $f_t = \sigma(W_f \otimes (h_{t-1}, x_t) + b_f$

Calculate $c_t = c_{t-1} * f_t + i_t$

Calculate $o_{\alpha t} = \sigma(W_{o\alpha} \otimes (h_{t-1}, x_t) + b_{o\alpha}$

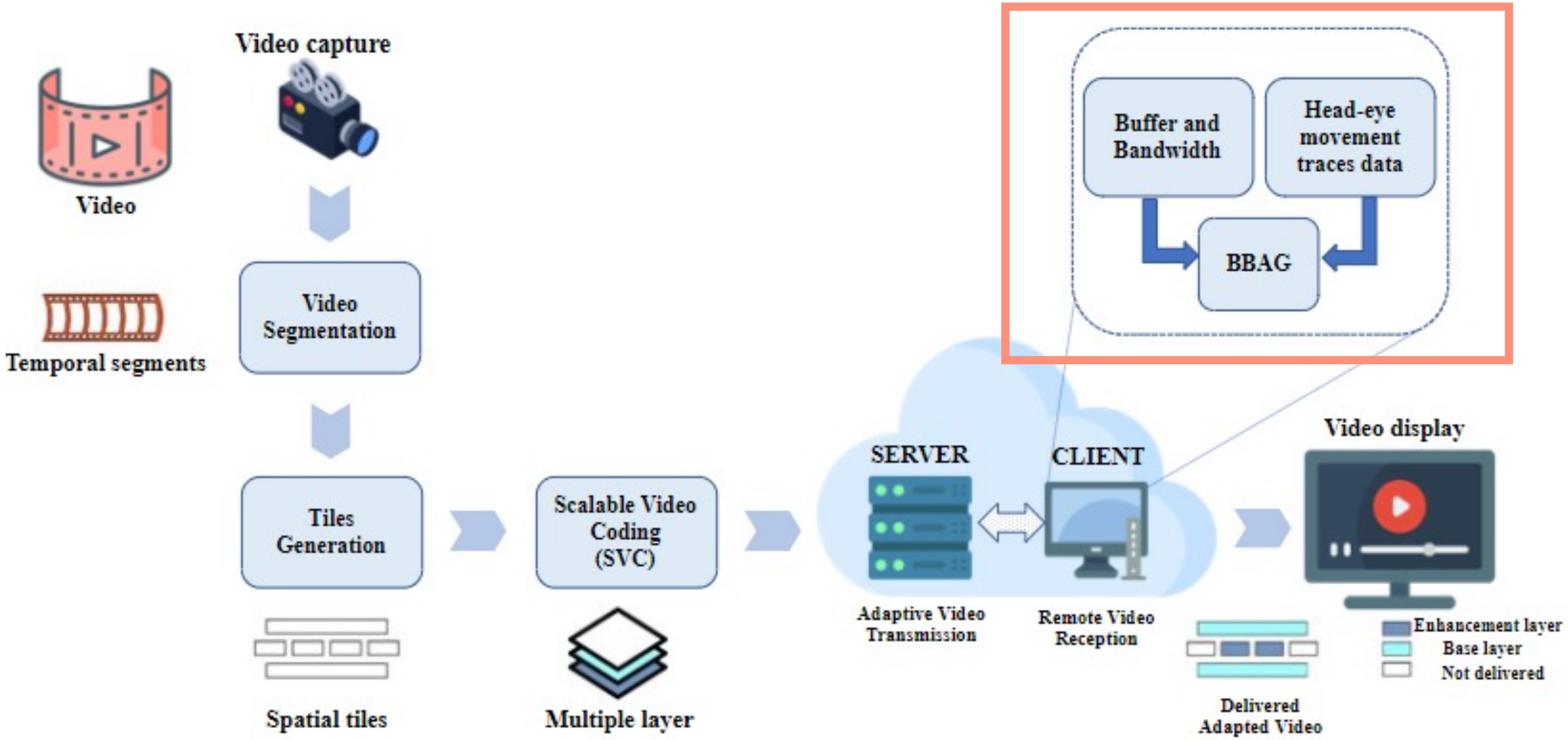
Calculate $o_{\beta t} = \tanh(W_{o\beta} \otimes c_t + b_{o\beta}$

Calculate $h_t, y_t = o_{\alpha t} * o_{\beta t}$

end

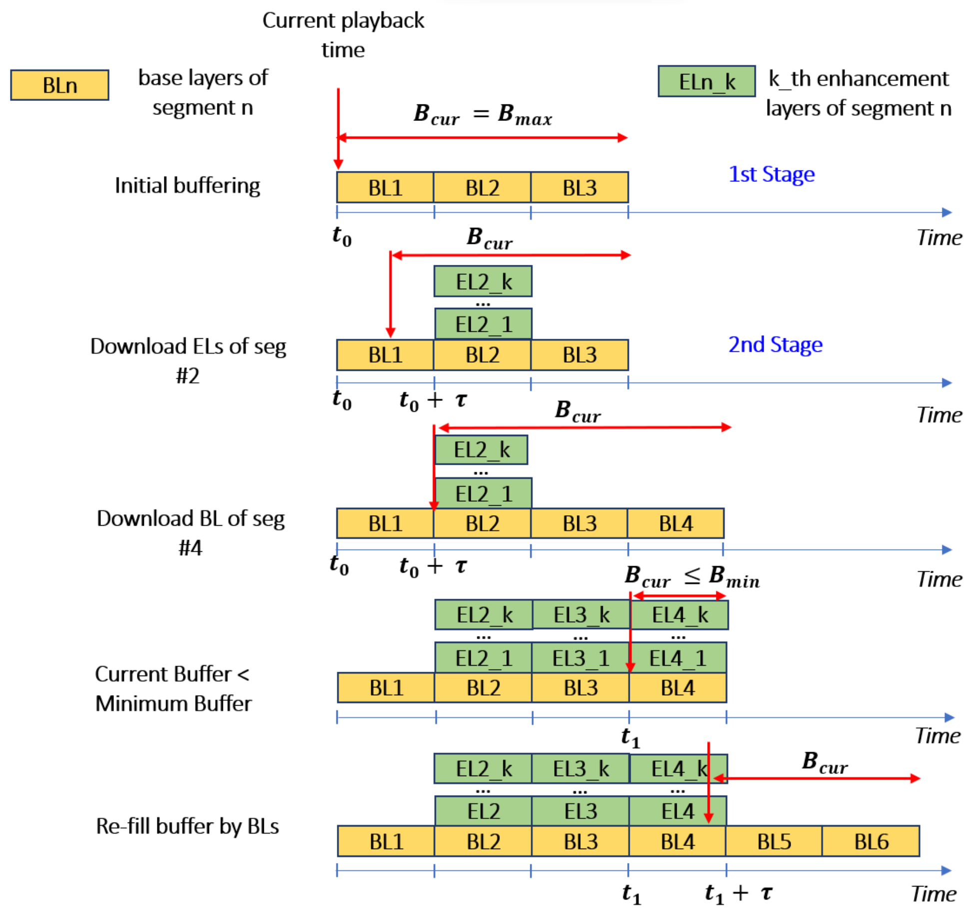
c_t, h_t, y_t

Adaptive streaming system



Version adaptation based on buffer level

- ✓ The objective is to minimize the occurrence of **re-buffering** events.
- ✓ The buffer is divided into four ranges—critical, low, medium, and high—based on predefined thresholds denoted as B_{min}, B_{low}, B_{high}, và B_{max}



QoE modelling

$$QoE = \sum_k (\alpha \times bitrate_k - \beta \times rebuffer_k - \gamma \times smooth_k)$$

✓Rebuffer$_k$ denotes the re-buffering duration at segment k

$$rebuffer_k = \begin{cases} |B_k - t_{segment}|, & (B_k > t_{segment}) \\ 0 & \end{cases}$$

✓Smooth *k* denotes the bitrate difference between two consecutive segments R_k và R_{k+1}

$$smooth_k = | R_{k+1} - R_k |$$

✓Given the network conditions and the user's viewport, determine the optimal set of layer values $\{l_1, l_2, \ldots, l_N\}$ to maximize the user's Quality of Experience (QoE).

Select layer for tiles

Algorithm 3: *Lựa chọn layer cho các tile*

Input: $N, R_k^{thresh}, R_{l,n,k}, V_k, w_n(V_k), B_{low}, B_{high}, B_{cur}$

Output: $\{l_n\}_{1 \le n \le N}$

$l_n \leftarrow 0$ **for** $1 \le n \le N$;

$\Delta R \leftarrow R_k^{thresh} - \sum_{n=1}^{N} \sum_{l=0}^{l_n - 1} R_n^{l_n}$;

Sắp xếp tile theo w: $sortedTile \leftarrow sort(w_n(V_m))$;

$B_{cur} \leftarrow$ Mức bộ đệm hiện tại;

for $l = 1$ *to* $L - 1$ **do**

    foreach $n \in sortedTile$ **do**

        if $B_{cur} \ge B_{high}$ **then**

            if $l_n < L - 1$ **and** $R_{l_n+1,n,k} < \Delta R$ **then**

                $\Delta R \leftarrow \Delta R - R_{l_n+1,n,k}$

                $l_n \leftarrow l_n + 1$

            else

                $\Delta R \leftarrow \Delta R - R_{l_n,n,k}$

                $l_n \leftarrow l_n$

            end

        else if $B_{low} \le B_{cur} < B_{high}$ **then**

            if $l_n < L - 1$ **and** $R_{l_n+1,n,k} < \Delta R$ **then**

                $\Delta R \leftarrow \Delta R - R_{l_n,n,k}$

                $l_n \leftarrow l_n$

            else

                $\Delta R \leftarrow \Delta R - R_{l_n-1,n,k}$

                $l_n \leftarrow l_n - 1$

            end

        else

            for $j = l_n; j \ge 0; j--$ **do**

                if $l_n < L - 1$ **and** $R_{l_n+1,n,k} < \Delta R$ **then**

                    $\Delta R \leftarrow \Delta R - R_{j,n,k}$

                    $l_n \leftarrow j$

                end

            end

        end

    end

end

return $\{l_n\}_{1 \le n \le N}$;

Experimental settings

(a) Bar

(b) Porto Riverside

(c) Sofa

(d) Turtle

- The end-to-end latency is set to 10ms
- The number of throughput samples S is set to 3
- The coefficient values α, β, and γ are set to 1; 1.85; and 1 respectively as in [110]).

360-degree videos used

[110] C. Zhou, Y. Ban, Y. Zhao, L. Guo, and B. Yu, "Pdas: Probability-driven adaptive streaming for short video," in Proceedings of the 30th ACM International Conference on Multimedia, ser. MM '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 7021–7025. [Online]. Available: https://doi.org/10.1145/3503161.3551571

Experimental settings

Average bitrate of a tile corresponding to the five test videos (kbps)

Scalable Layer	Average tile bitrate (kbps)				
	Bar	Turtle	Porto Rive-rside	Sofa	Ocean
Enhancement Layer 4	824	322	224	131	263
Enhancement Layer 3	644	202	124	67	207
Enhancement Layer 2	323	98	62	33	98
Enhancement Layer 1	194	63	41	24	80
Base Layer	101	30	27	19	46

Statistical table of the bandwidth trace dataset (Mbps)

	7Train1	7Train2	Bus57	Bus62	Long Island Rail Road	QTrain
Average	6.81	9.07	2.47	0.09	4.29	8.49
Median	5.28	8.42	0.008	0.003	3.08	7.75
Max	31.40	25.40	23.20	8.26	16.50	28.40
Min	0.02	0.02	0	0	0	0
STD	5.51	6.26	4.96	0.49	3.90	6.39

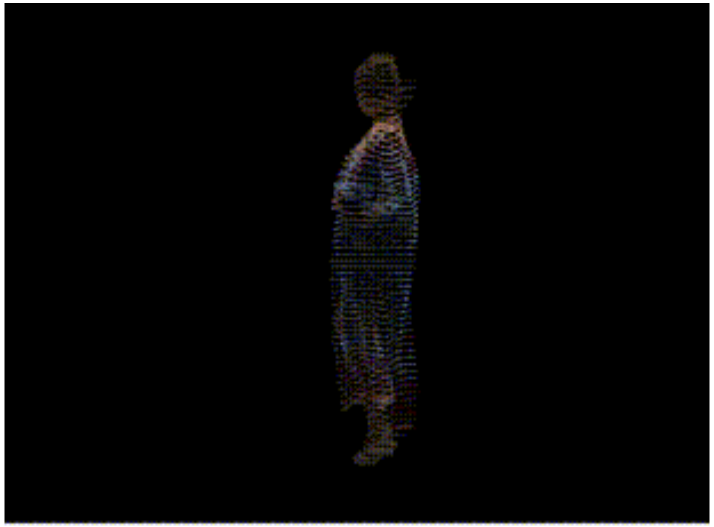
Performance evaluation

Average bitrate (BR) and average buffer level (BL) (s) over time of the proposed and reference methods under a simple bandwidth trace.

Video đơn giản		Turtle	Sofa	Bar	Porto Riverside
TLGA	Avg. viewport BR	50.48	19.38	61.81	32.43
	Avg. BL	2.71	2.80	3.43	2.83
	Min BL	0	1	0	2
SHVH	Avg. viewport BR	159.25	85.30	151.01	133.91
	Avg. BL	2.90	2.32	2.20	2.24
	Min BL	2	1	1	1
S-VAS	Avg. viewport BR	154.90	88.93	170.85	148.07
	Avg. BL	2.33	3.00	2.11	2.33
	Min BL	1	2	1	1
BBAG	Avg. viewport BR	183.78	97.31	214.09	155.65
	Ave. BL	3.06	3.14	3.74	3.13
	Min BL	2	2	3	2

Average QoE under a simple bandwidth scenario

Video	Avg. QoE			
	TLGA	SVSH	S-VAS	BBAG
Turtle	12.85	43.82	43.05	55.87
Sofa	9.77	38.80	40.66	43.90
Bar	8.28	65.20	44.05	85.69
Porto Riverside	13.26	63.44	63.83	86.59



THANK YOU FOR YOUR ATTENTION!