Building a Next-Generation Cloud Gaming Platform with Planar Map Streaming and Distributed Rendering

Pin Chun Wang

May, 2017

Outline

Why Cloud Gaming

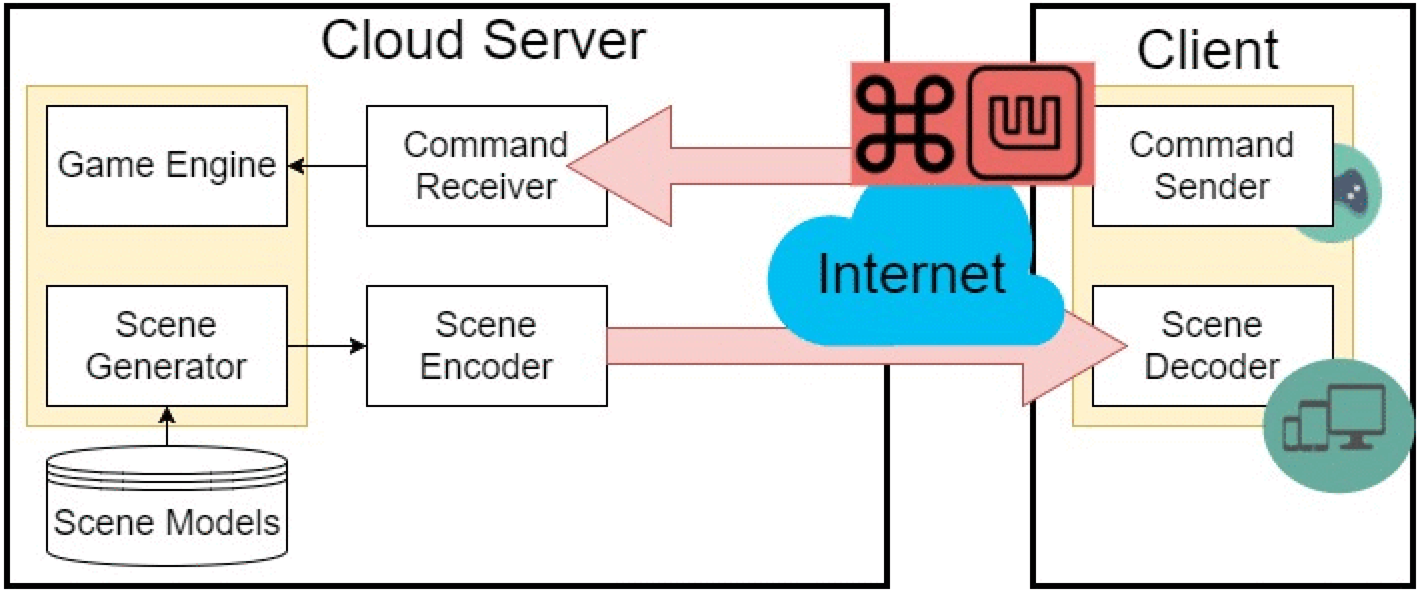
For gamers,

- require constantly updating hardware
- are locked to a specific computer

For developers,

- need to solve hardware compatible issues
- need to support multiple platforms

Cloud Gaming Platforms



- Perform game logic execuation on server
  ➔ thin client is enough
- Gaming result is streamed to client
  ➔ supports clients on multiple platforms

Limitaitons of Existing Solutions

Existing Solutions

-
-

SONY PS NOW

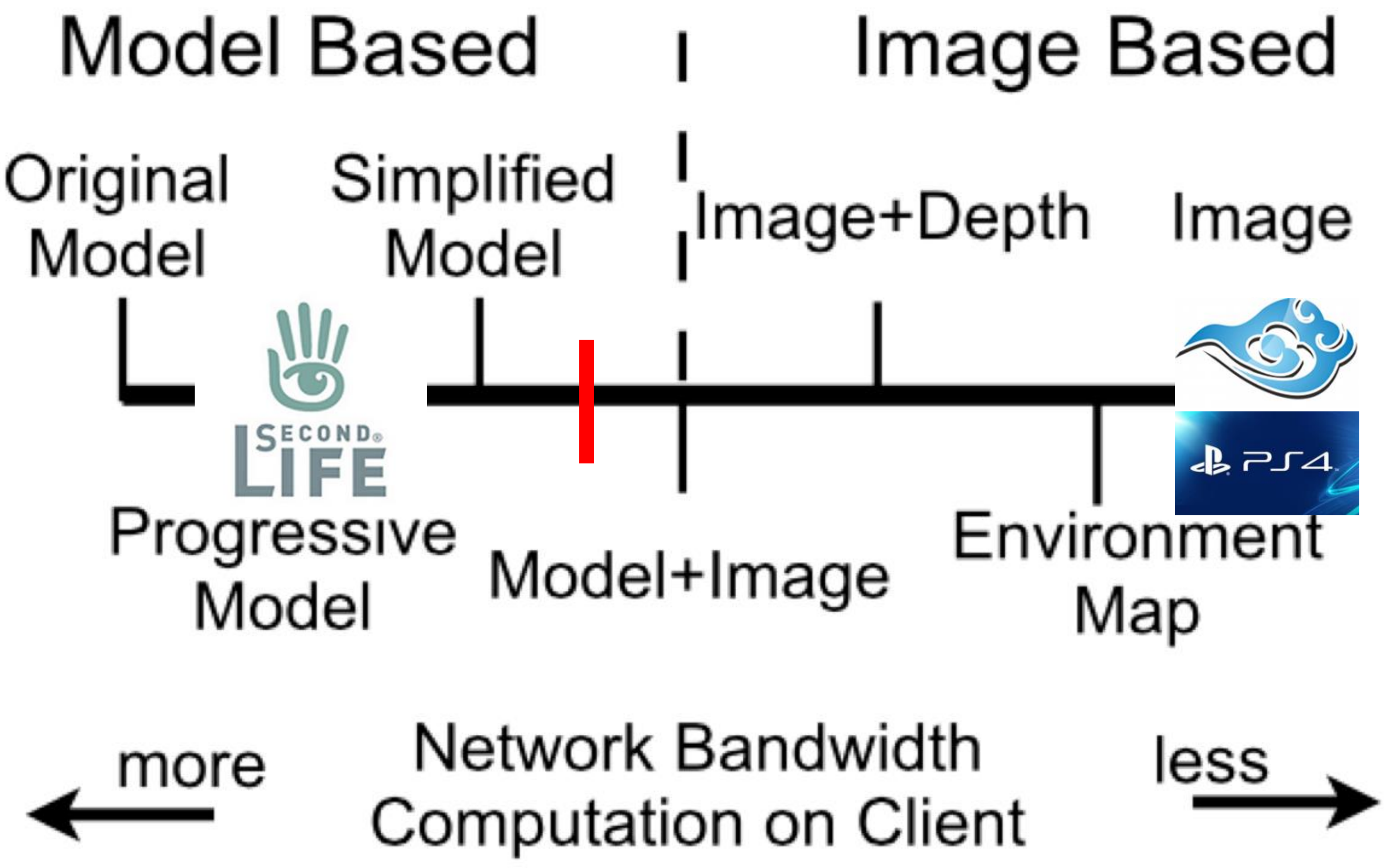
Gaming Anywhere

Limitations

- High bandwidth consumption
- Limited scalability
- Little room for optimizations

PS Now: https://www.playstation.com/en-us/explore/playstationnow
Gaming Anywhere: http://gaminganywhere.org/doc.html

Classifications of Cloud Gaming Platforms



[1] S. Shu, and C. Hsu. "A survey of interactive remote rendering systems." *ACM Computing Surveys (CSUR)*, 2015

Contributions

- We propose a **distributed rendering server-client framework** for cloud gaming platforms.
- We optimize the proposed system by exploring and compressing a new datatype, named planar map [1].

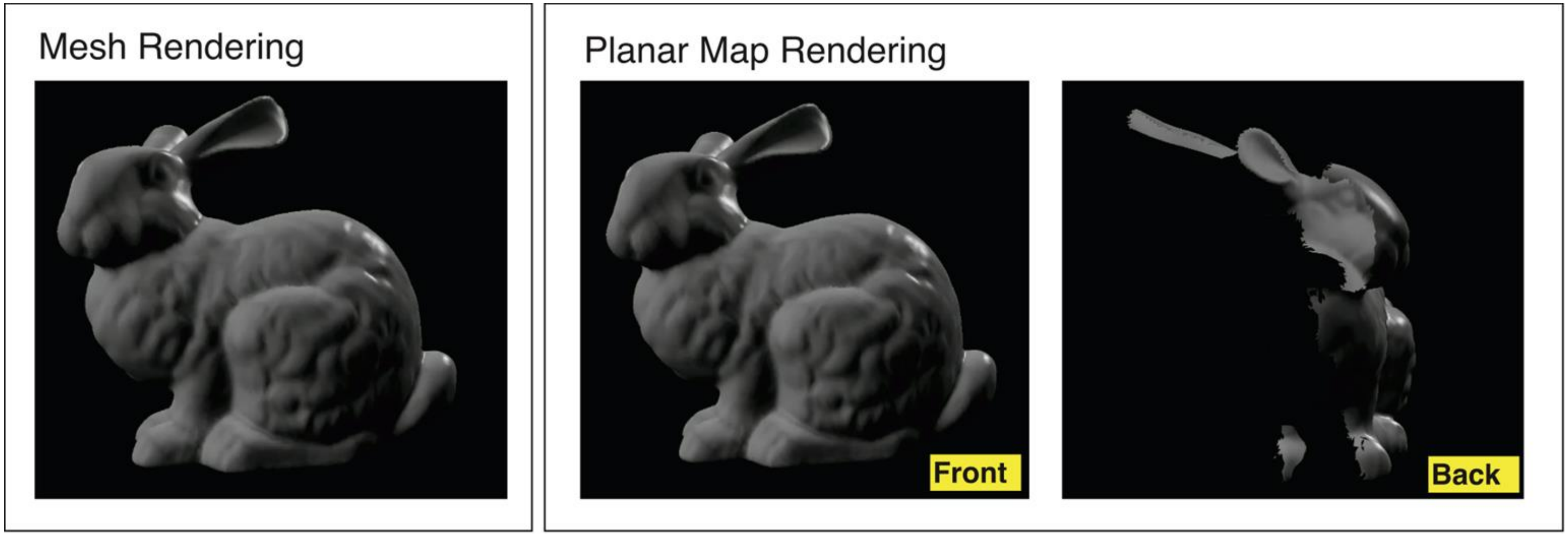
[1] **P. Wang**, A. Ellis, J. Hart, C. Hsu. "Optimizing Next-Generation Cloud Gaming Platforms with Planar Map Streaming and Distributed Rendering. In Proc. of IEEE Workshop on Network and Systems Support for Games (NetGames '17)

Outline

1. Introduction
2. **Planar Map**
3. System Architecture
4. Planar Map Compression
5. Evaluations
6. Future Work and Conclusion

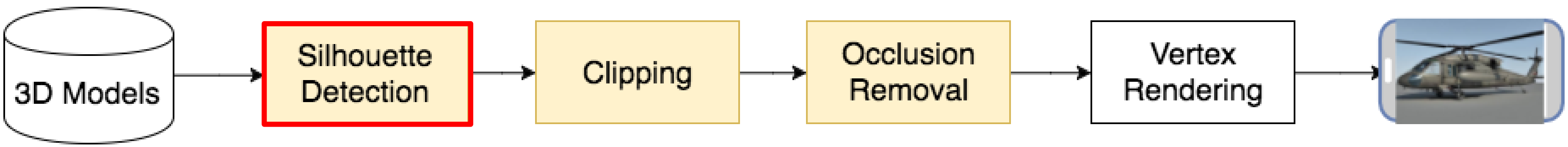
Planar Map Rendering

- Planar map is a sequence of visible triangles
- 3D mesh rendering pipeline draws everything in 3D world while 2D planar map rendering pipeline **only renders visible triangles**
- Ellis et al. [1] proposed a real-time planar map pipeline

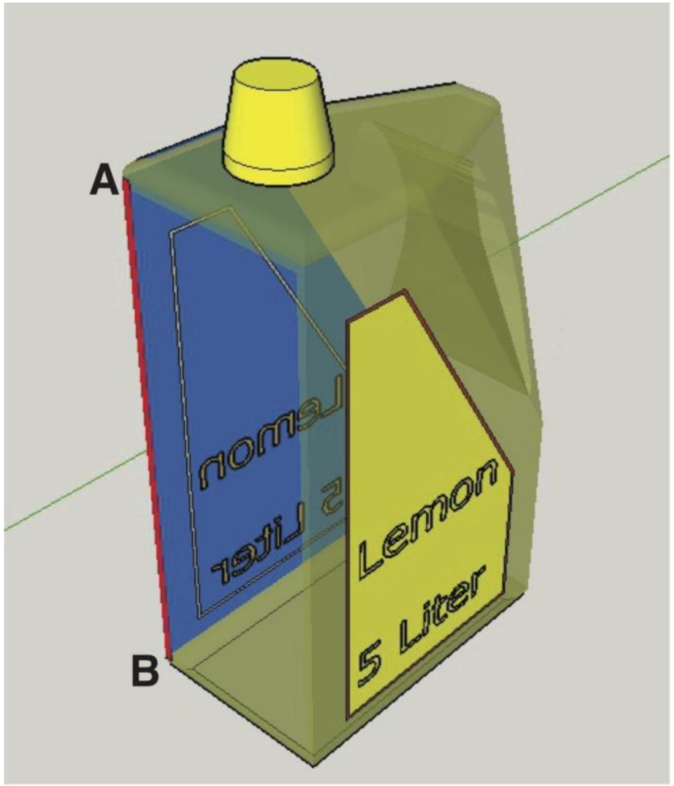


[1] A. Ellis, W. Hunt, and J. Hart. Svgpu: real time 3d rendering to vector graphics formats. In *Proc. of High Performance Graphics (HPG'16)*, pages 13–21, 2016.

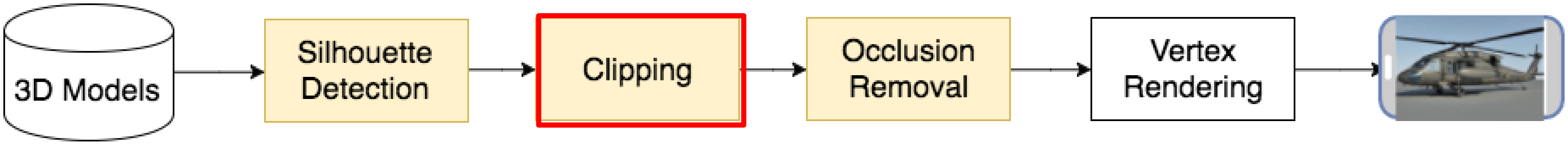
Silhouette Detection



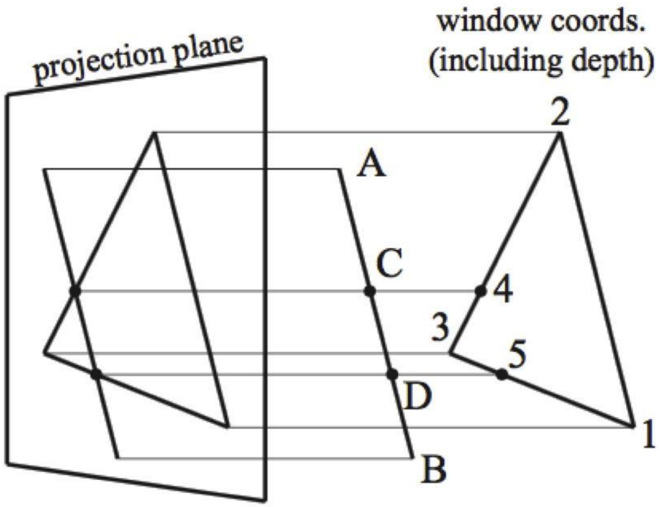
- Silhouette is an edge shared by a frontfacing and a backfacing triangle
- We use a hash table to record all the edges in 3D scene as entry
- We check whether the corresponding faces' normal are opposite signs in the *z*-axis



Silhouette Clipping

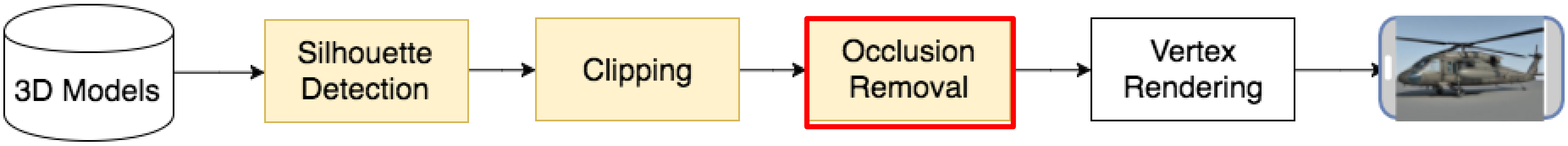


1. Project the triangle and silhouette onto the view plane
2. Clip each triangle against its list of overlapping silhouette
3. Walks through the polygon's vertices determine whether to: (i) output a polygon edge, (ii) generate a new vertex, (ii) output the clipped edge



[1] A. Ellis, W. Hunt, and J. Hart. Svgpu: real time 3d rendering to vector graphics formats. In *Proc. of High Performance Graphics (HPG'16)*, pages 13–21, 2016.

Triangle Occlusion



- Leverage the property that no triangle is partially occluded to remove triangles
- Discard the triangles whose centroid is overlapped by any other triangle
- Then, we obtain a set of triangles with the depth complexity of one

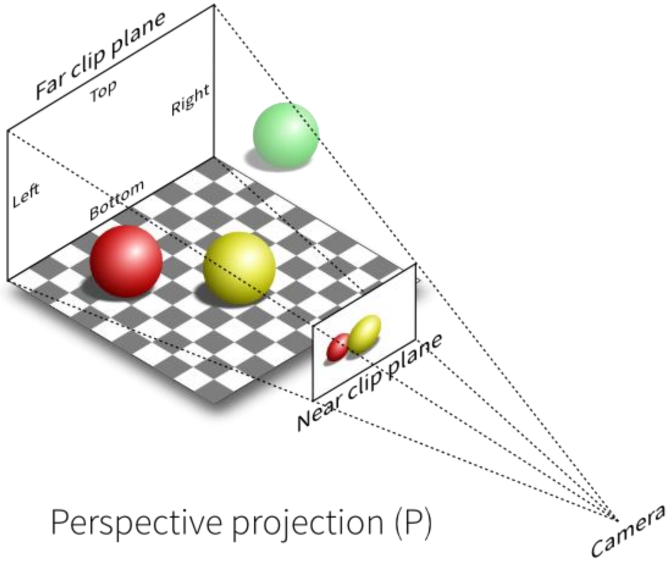
Outline

1. Introduction
2. Planar Map
3. **System Architecture**
4. Planar Map Compression
5. Evaluations
6. Future Work and Conclusion

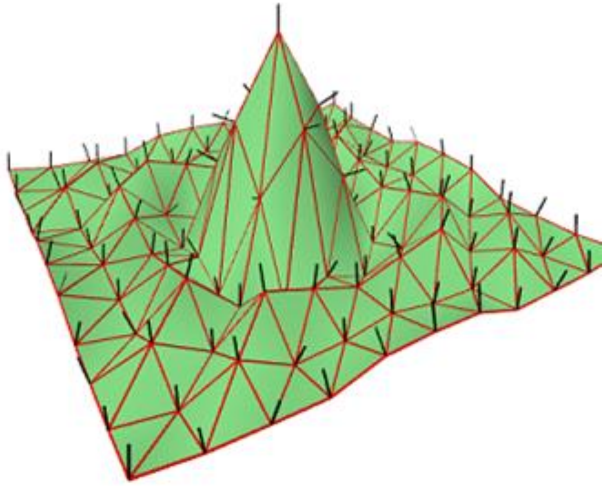
Planar Map Format

In every planar map, we need to describe:

- Viewing matrix
- Draw call number and size
- 2D triangle decription, including geometric information, texture coordinates, and normals



Perspective projection (P)



Coordinate Systems

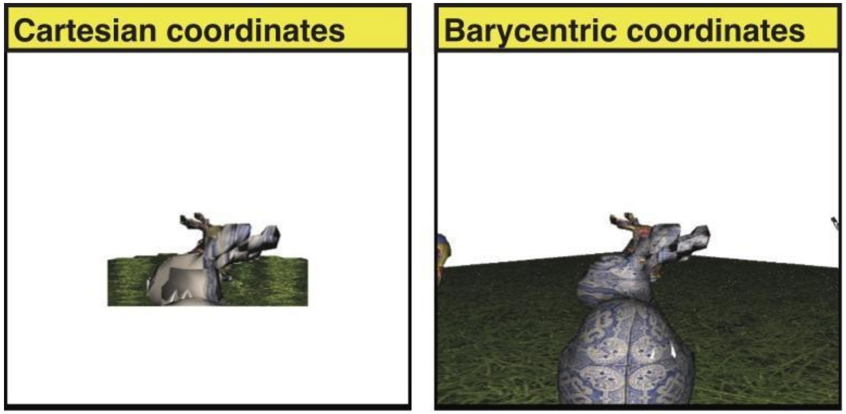
Cartesian Coordinates,

- Pros: preserve spatial property within triangles
- Cons: no range limitation, require 30 entries for a triangle

Barycentric Coordinates, describe vertex information, within a triangle

- Pros: shorter indexes for triangles, common triangle patterns on unclipped triangles

- Cons: correlation among vertices harder to be leveraged



Coordinate Systems

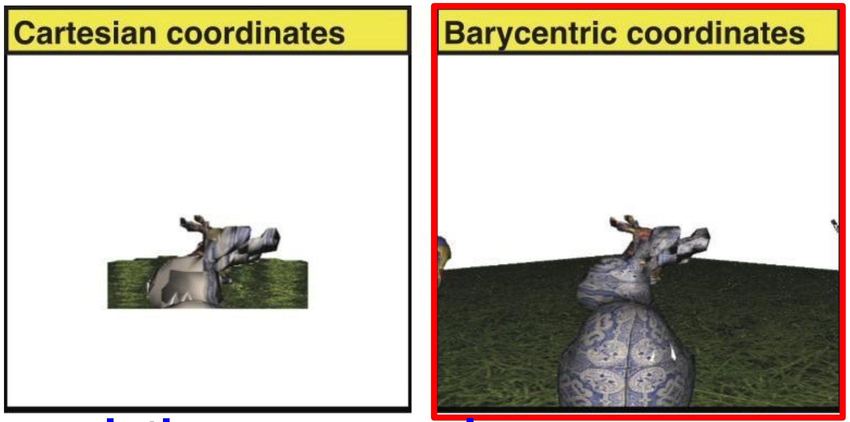
Cartesian Coordinates,

- Pros: preserve the spatial property within triangles
- Cons: no range limitation, require 30 entry for a triangle

Barycentric Coordinates, describe vertex information, within a triangle

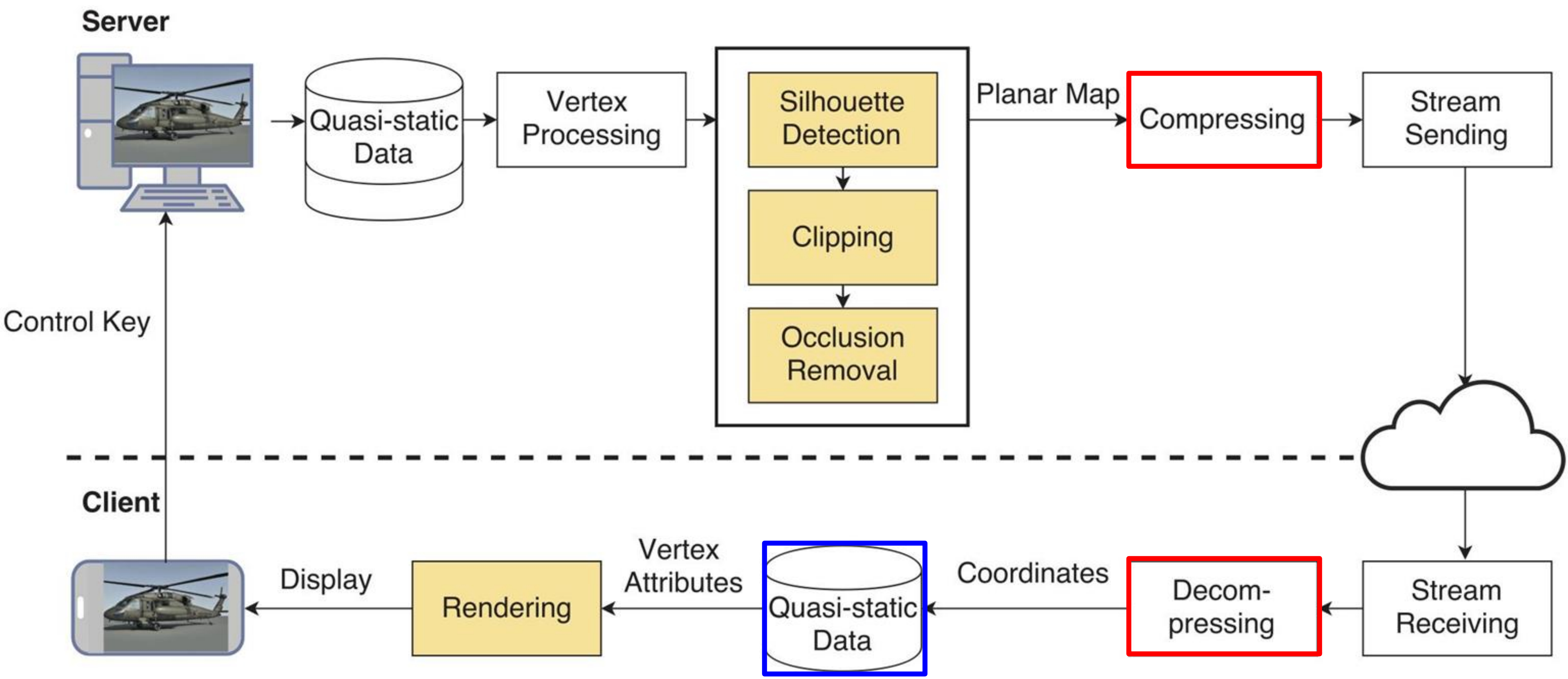
- Pros: shorter indexes for triangles, common triangle patterns on unclipped triangles

- Cons: correlation among vertices is harder to be leveraged



degradation on normal
-> missing triangle

System Overview



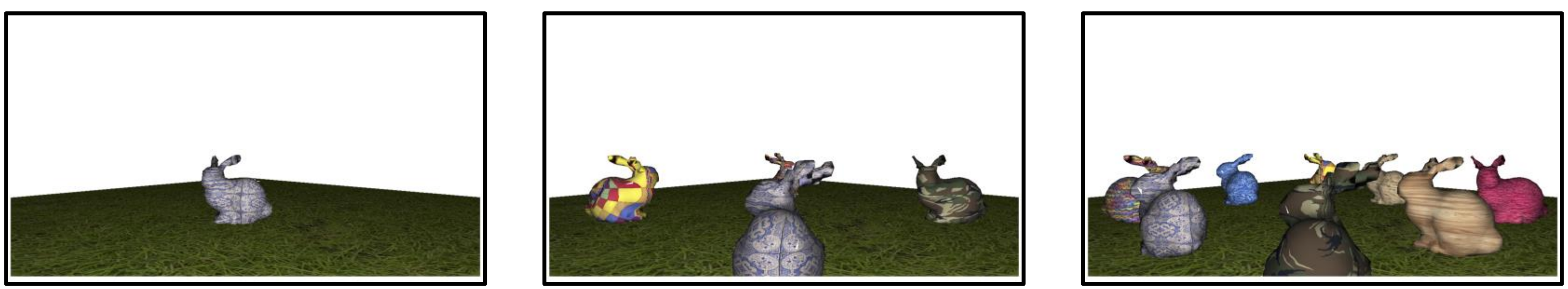
Outline

1. Introduction
2. Planar Map
3. System Architecture
4. **Planar Map Compression**
5. Evaluation
6. Future Work and Conclusion

Mesh Compression Pipeline

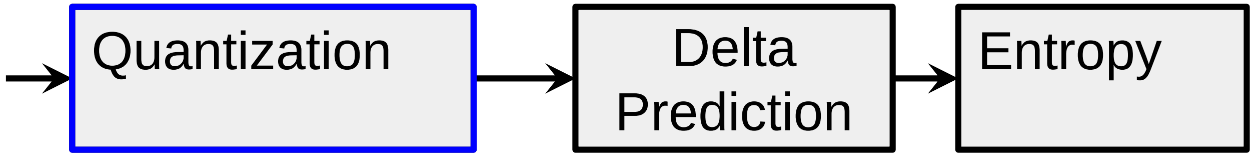
- We record three scenes in 720p resolution, in which we vary the number of the popular Bunny model, among 1, 2, and 8

- We evaluate the performance by video quality metrics and compression ratio

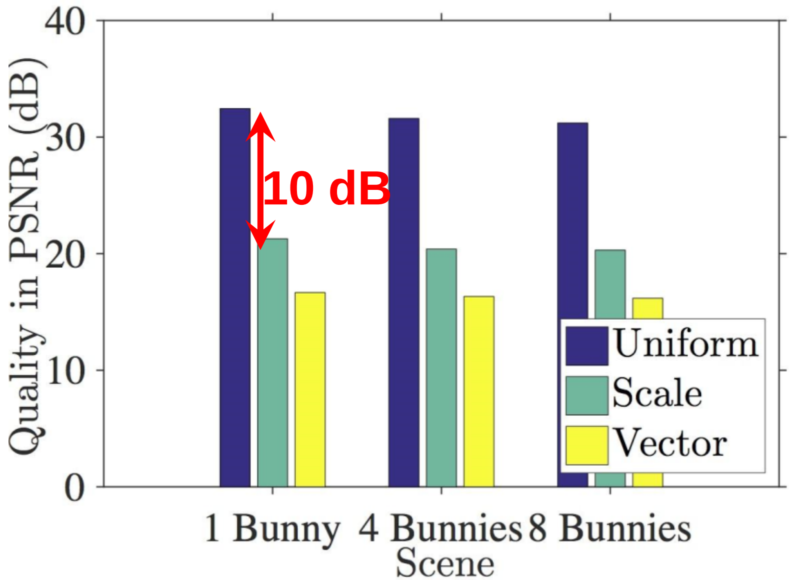


[1] J. Peng, C. Kim, and C. Kuo, "Technologies for 3D mesh compression: a survey," *Journal of Visual Communication and Image Representation*, vol. 16, no. 6, pp. 688–733, 2005.

Quantization



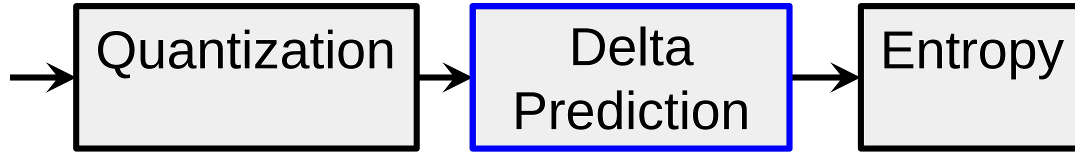
- Uniform: divide the representative range **in equal size**
- Scale: apply Lloyd's algorithm on **individual dimensions sequentially**
- Vector: **all dimensions** are jointly quantized using K-means



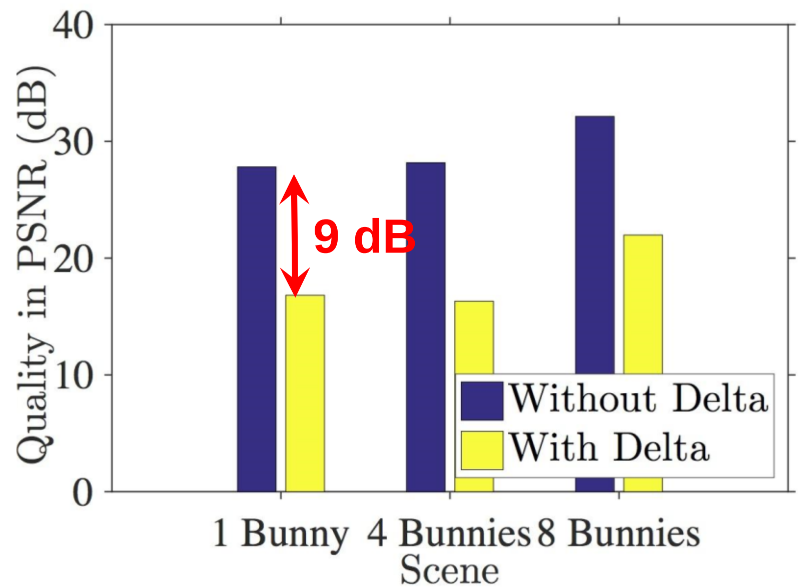
Decision: Uniform Quantization

Uniform quantization outperforms other modules since barycentric coordinates range between 0~1.

Delta Prediction



To leverage the property that close-by vertices share similar information, a prediction algorithm may use previous coordinates to predict current coordinates



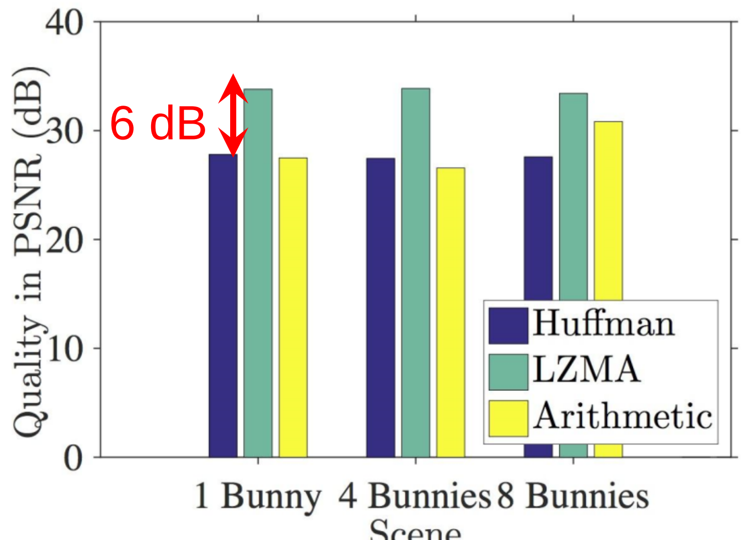
Decision: no delta prediction

Barycentric coordinates do not share spacial property
Therefore, delta prediction harms its compression ratio

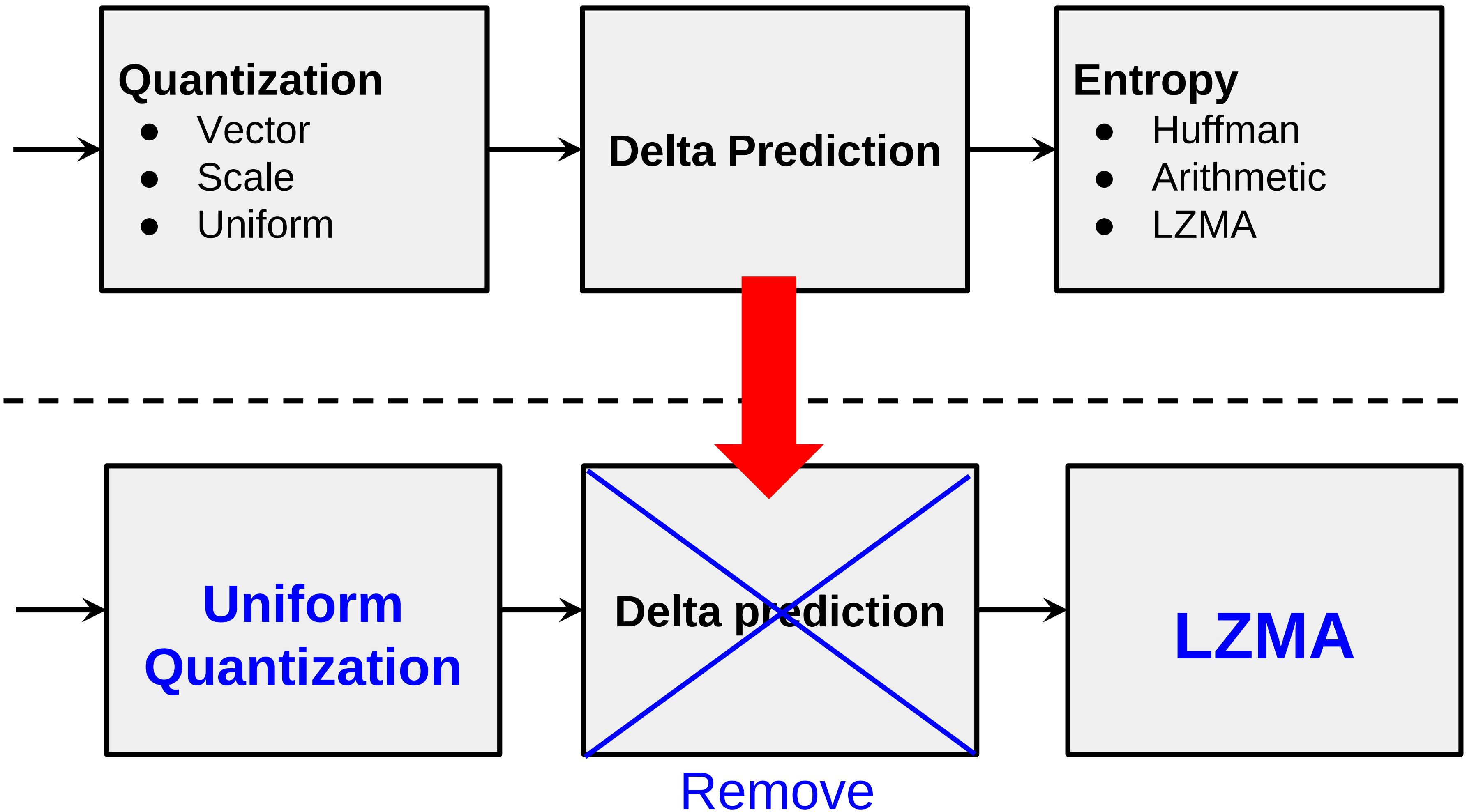
Entropy Encoding

- Huffman: builds a tree with the more frequent elements at lower levels of the tree
- Arithmetic: converts symbol sequence into a floating point between 1 and 0, and shrinks the interval based on the symbol probability
- LZMA: a dictionary compressor, which encodes a stream with an adaptive binary range coder

Decision: LZMA encoding



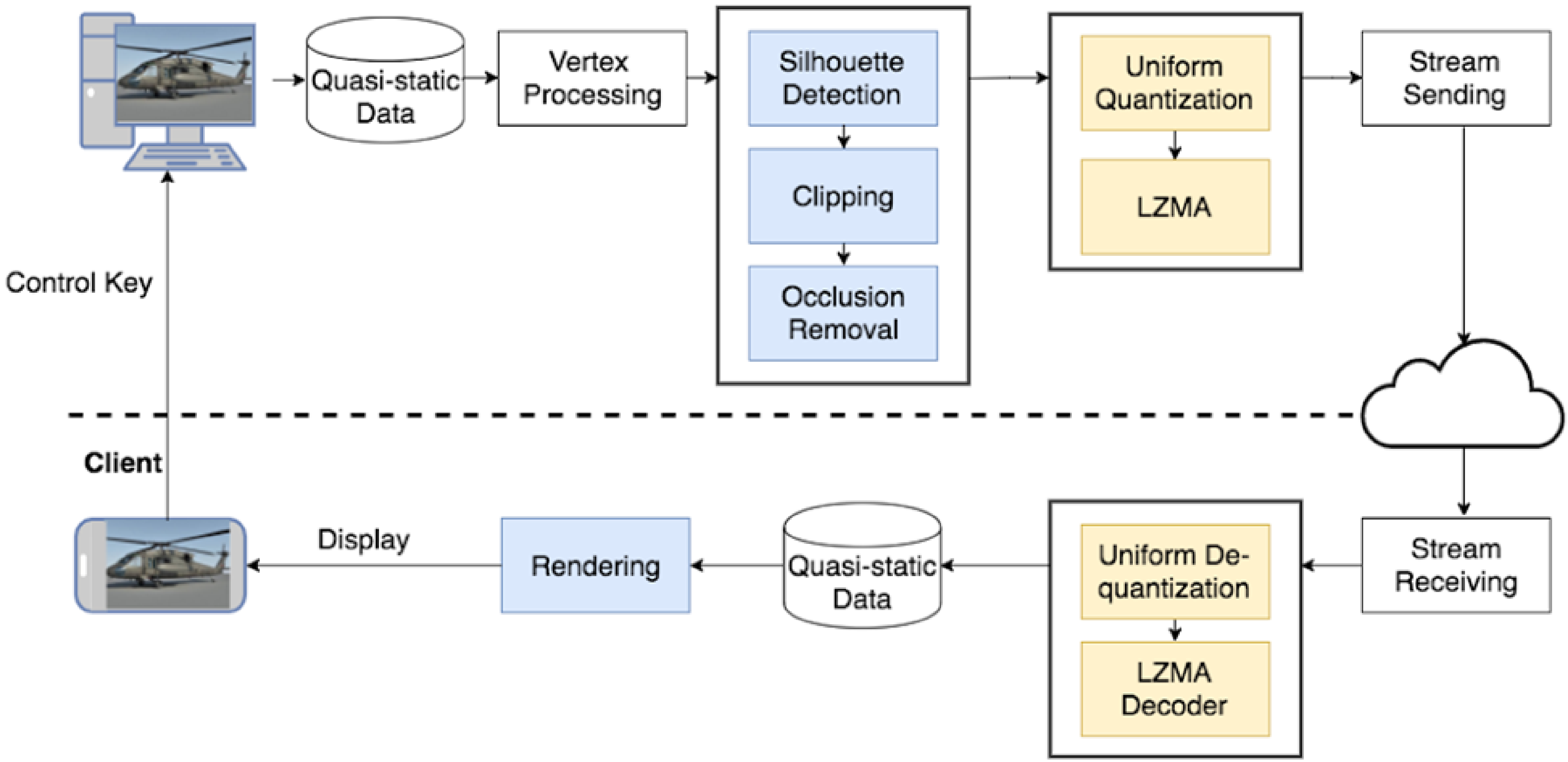
Compression Pipeline (Summary)



Outline

1. Introduction
2. Planar Map
3. System Architecture
4. Planar Map Compression
5. **Evaluations**
6. Future Work and Conclusion

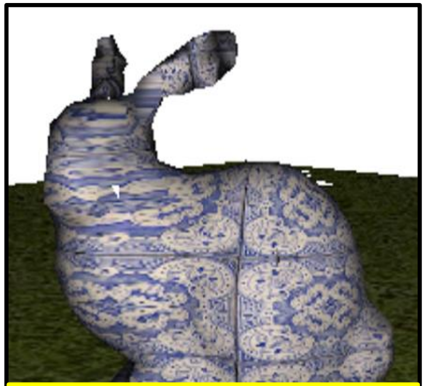
System Architecture (Revised)

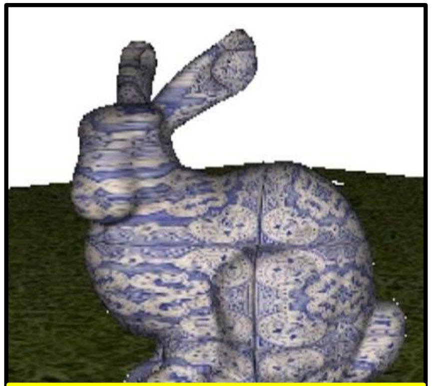


Setup

Environment:

- i7 3.4 GHz
- GPU: NVidia Quadro M4000


37,67 vertices


200,700 vertices

Baseline:

- image-based solution
- state-of-art x265 video codec

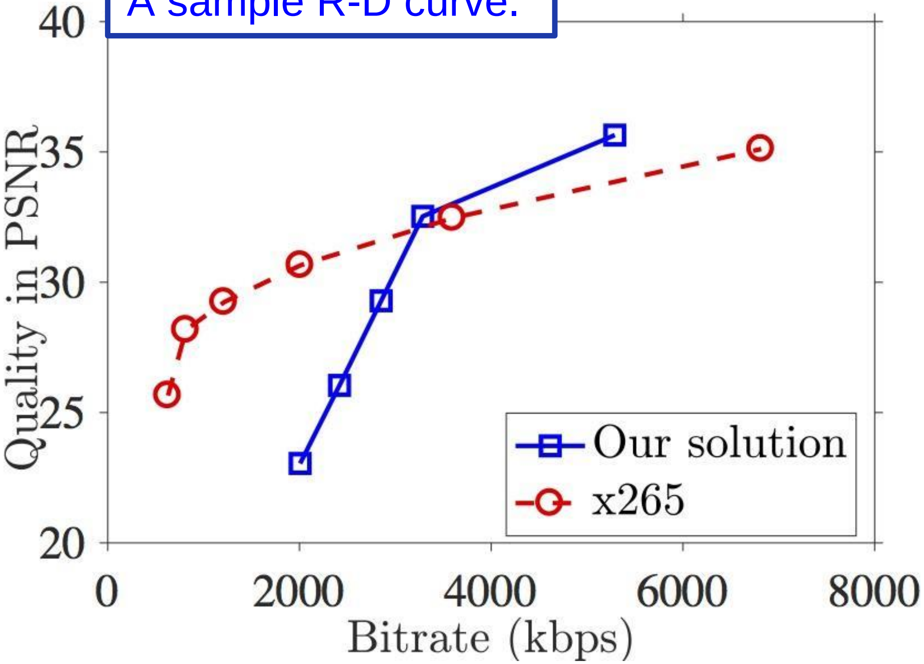
Scenes:

- diverse model complexity : { Basic, Fine Grained }
- numbers of bunny : { 1, 4, 8 }
- camera speed : { slow, fast }

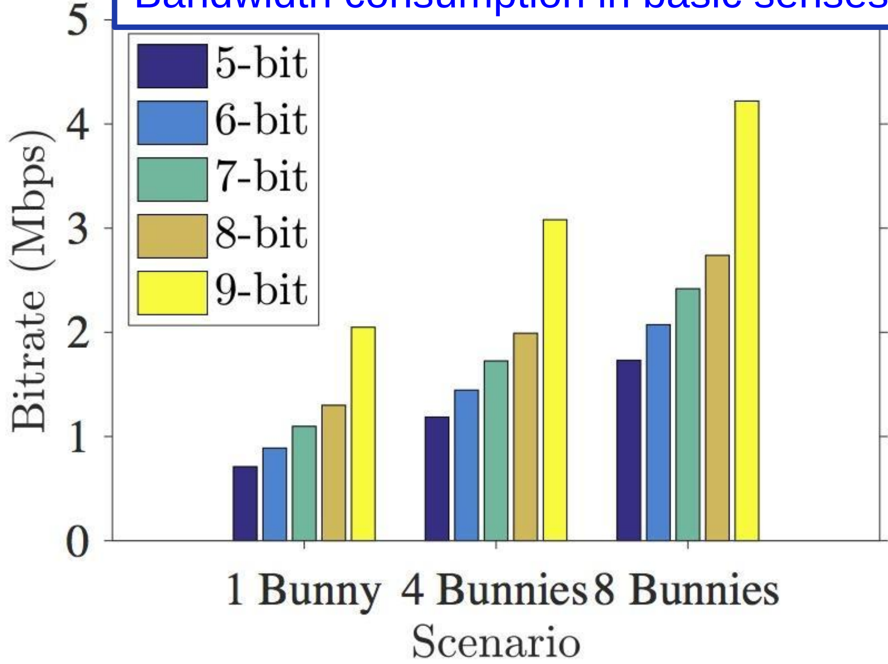
Bandwidth Consumption

- We vary compression depths in compression pipeline
- Bandwidth consumption ranges from 1.5 to 4 Mbps for basic bunnies, and 2.5 to 17 for fine-grained bunnies

A sample R-D curve.



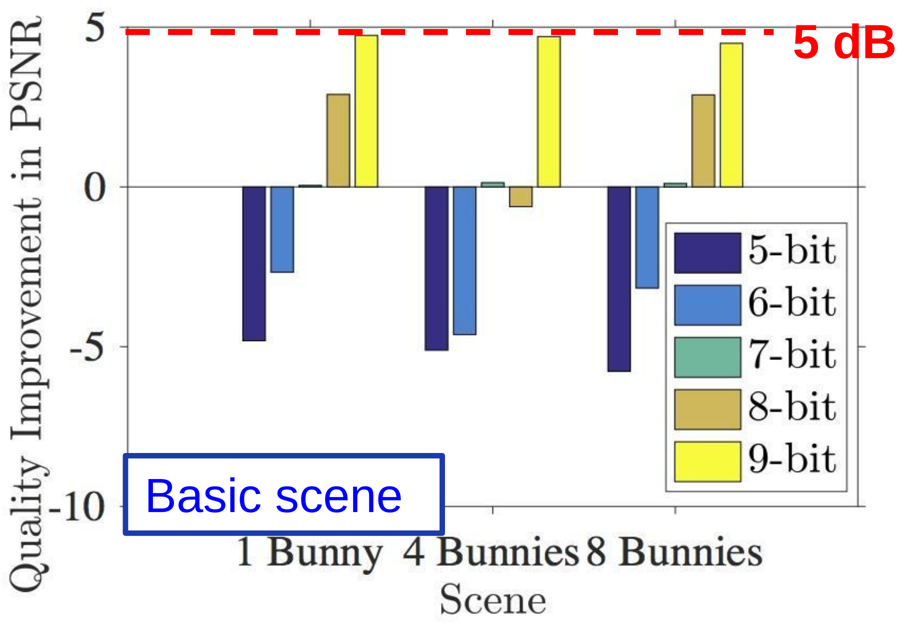
Bandwidth consumption in basic senses.

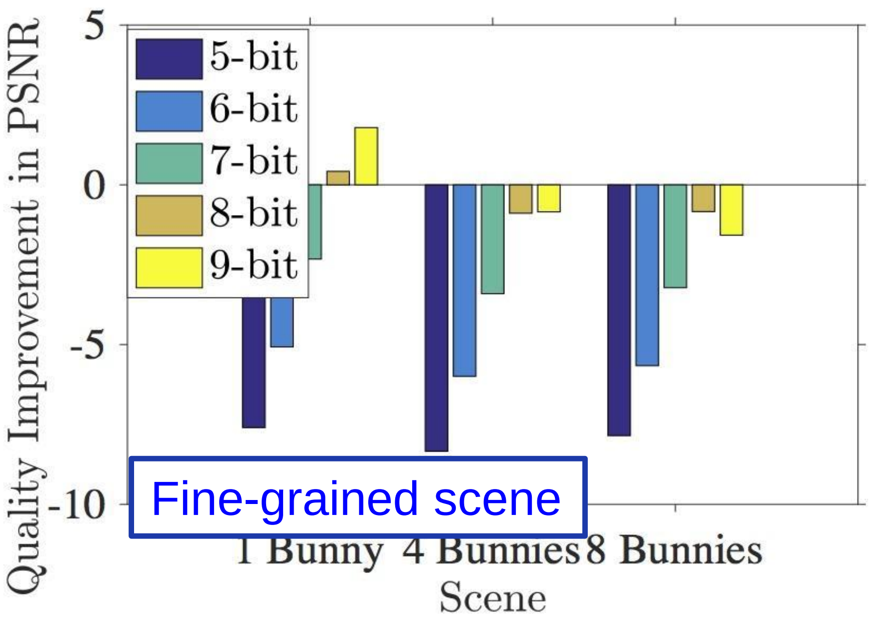


Outperforms x265 in PSNR under 3 Mbps

Video Quality (PSNR)

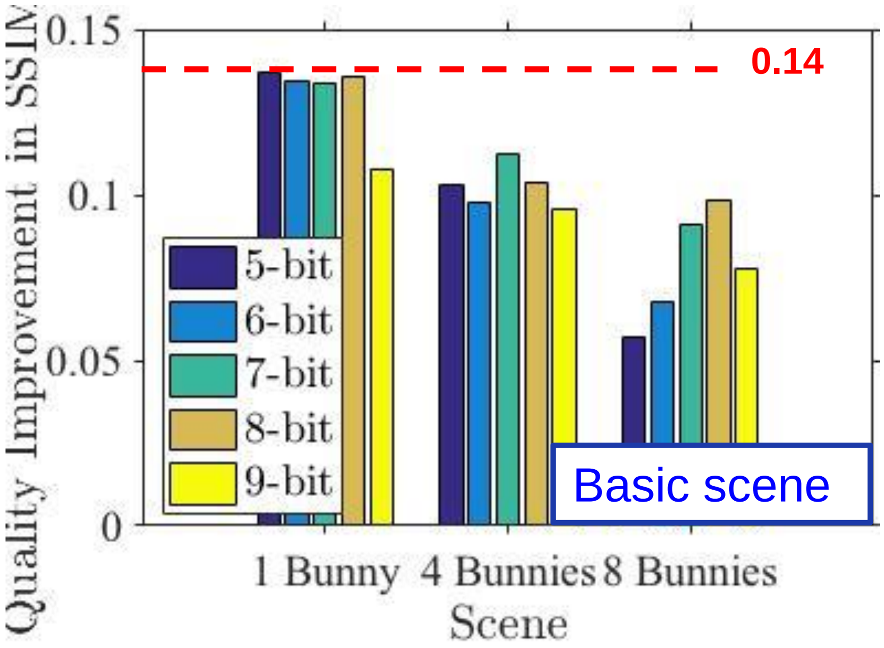
- Up to 5 dB improvement is possible, and as long as the bit-depth is ≥ 7 bits, our proposed solution results in higher PSNR in basic bunnies scenes.

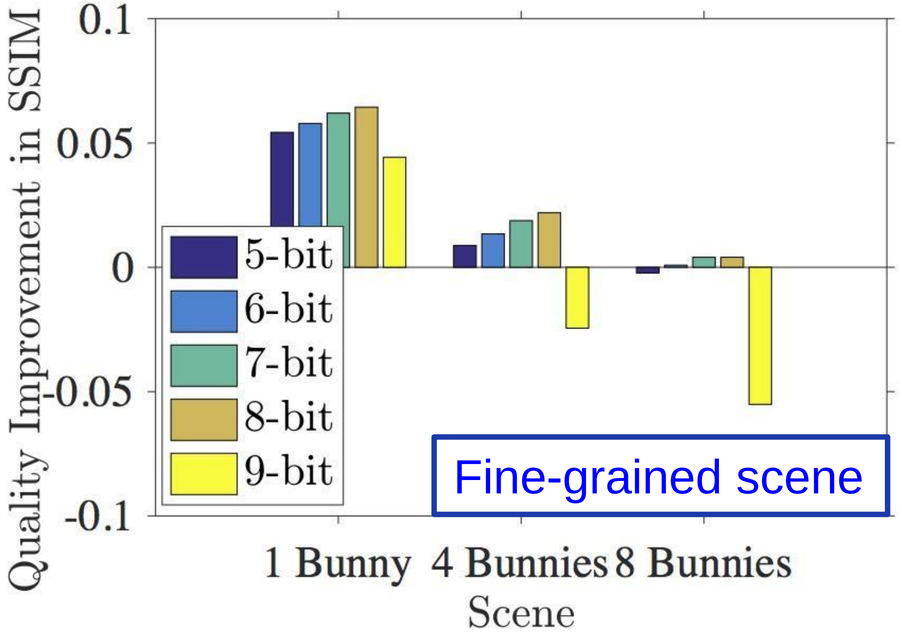




Video Quality (SSIM)

- Distributed rendering preserves image structures in low bitrate situation
- The gap is as high as 0.14 in SSIM

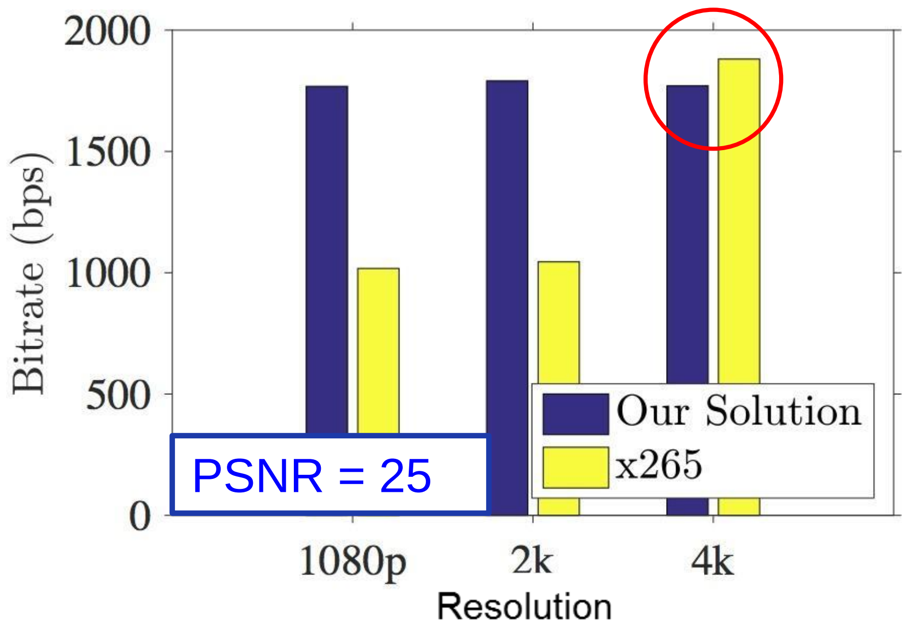


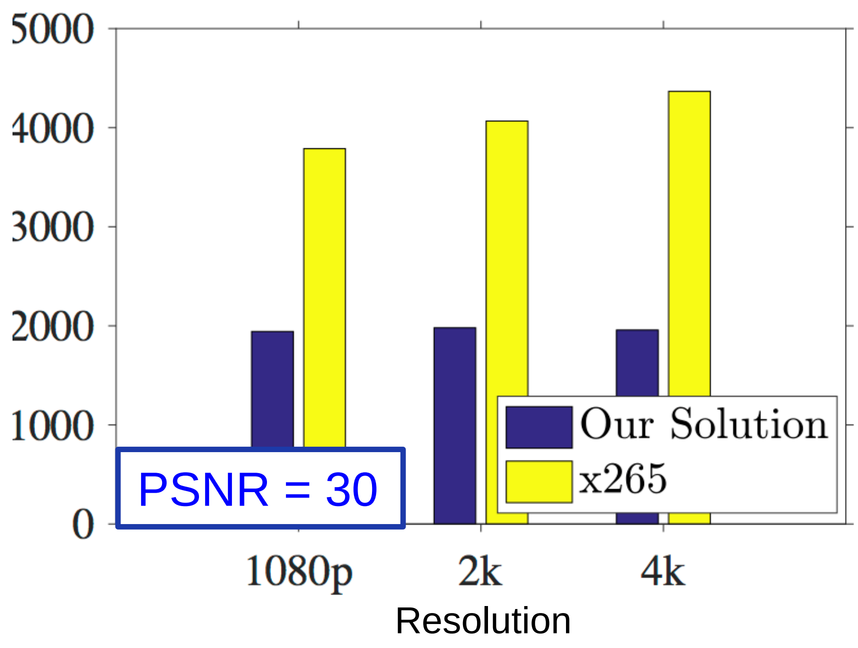


Outperforms x265 in SSIM under low bitrate scanario

Ultra-High Resolution Display

- Our proposed solution scale well to high resolution applications, such as 360 videos and VR





Our solution scales well to high resolution applications

Running Time

- Run computation demanding process on server side (< 1%)
- Achieve real-timeness (basic scene)

	Server side (mean / max) in ms.			**Client**
	Detection	Clipping	Occlusion	Rendering
Basic	0.27 / 0.28	25.56 / 33.41	1.94 / 2.68	0.22 / 0.48
F.G.	3.66 / 4.73	60.55 / 88.14	17.43 / 24.02	0.83 / 3.13

= 28 ms

Basic scene

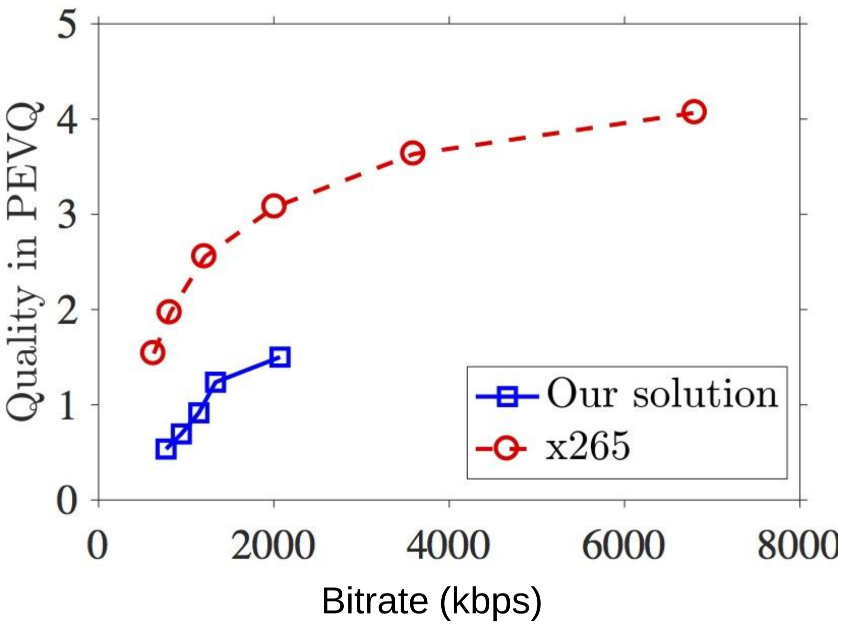
Fine-grained scene

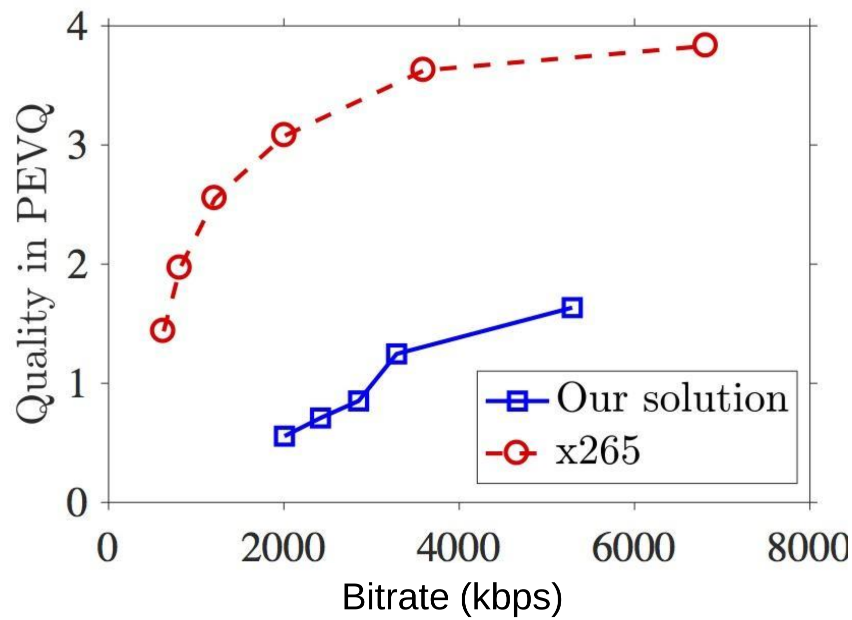
Outline

1. Introduction
2. Planar Map
3. System Architecture
4. Planar Map Compression
5. Evaluations
6. **Future Work and Conclusion**

Discussion

- PEVQ is an metrics of Perceptual Evaluation of Video Quality (PEVQ) [1] described in ITU-T J.247 Annex B

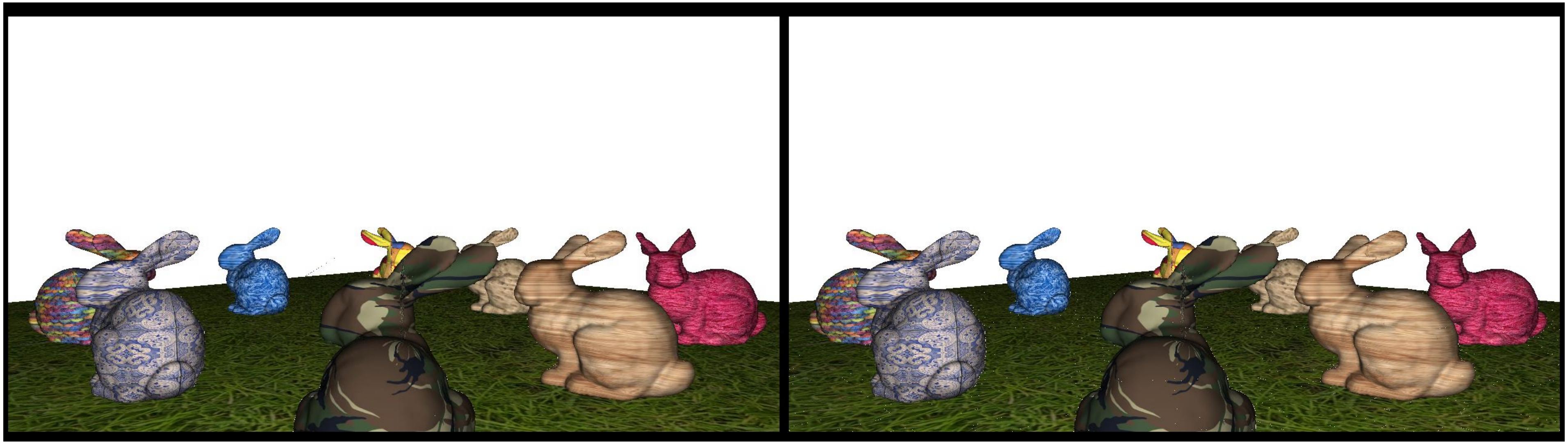




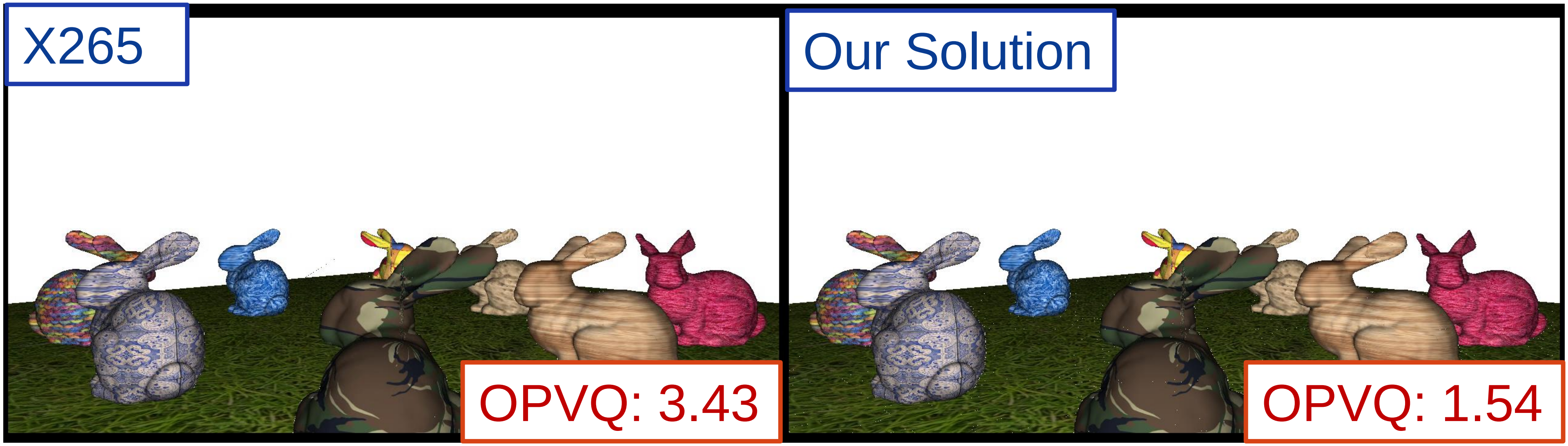
The curve of our solution is fairly flat compared to x265

[1] K. Skarseth, H. Bjørlo, P. Halvorsen, M. Riegler, and C.Griwodz. OpenVQ: a video quality assessment toolkit. In *Proc. of ACM International Conference on Multimedia (MM'16), OSSC paper*, pages 1197–1200, 2016.

Sample Video

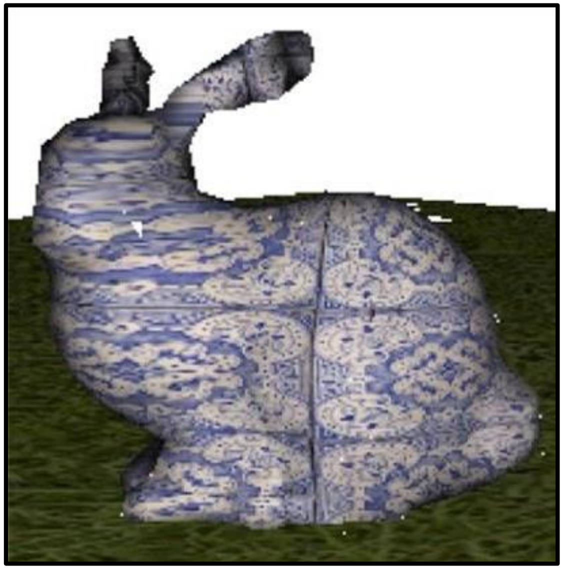


Sample Video



Hole Filling

- When zooming in the rendered scene, we find there are small holes within vertices, which may leads to lower PEVQ scores



Future Work

- Filling holes by morphological antialiasing
- Integrate with game engine
- Evaluate system performance with user study

OGRE

Mini3D

a minimalistic, cross platform, open source 3d Game Engine Framework

GODOT Game engine

[1] Reshetov, Alexander. "Morphological antialiasing." Proceedings of the Conference on High Performance Graphics 2009. ACM, 2009.

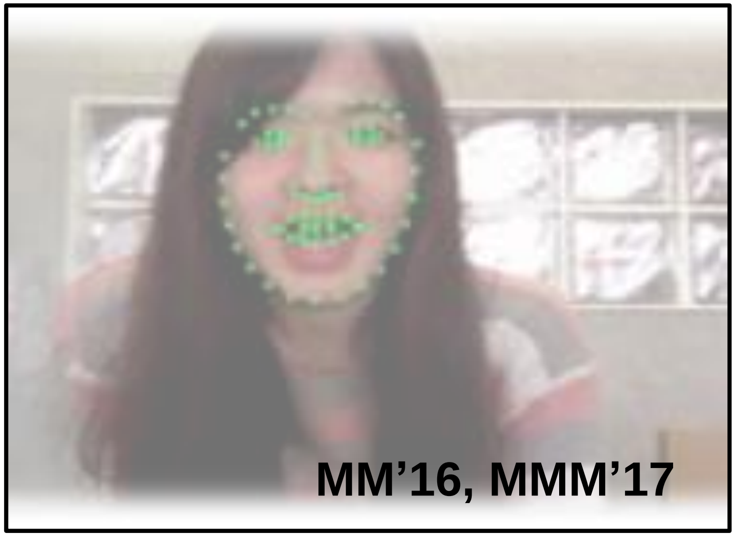
[2] Open source game engine: https://godotengine.org/; http://www.ogre3d.org/; http://www.mini3d.org/

Conclusion

- We proposed a cloud gaming platforms with planar map
- Compared to video streaming based platforms, our planar map based platform:
  - achieves higher video quality at the same bitrate
  - runs fast, especially at the client side
  - scales well to ultra-high-resolution displays

Q & A


NetGames'17


MM'16, MMM'17


IP Camera
Access Point
Internet
Other Sites
Projector & Display
Local Site

APNOMS'16