Capitalizing Light-Field Technology
in Head-Mounted Virtual Reality
頭戴式虛擬實境中之光場技術應用

Yu-Ming Lai
Advisor: Cheng-Hsin Hsu

Networking Multimedia Systems Lab
CS Dept. National Tsing-Hua University

Outline

01

Introduction

Explore the World on your Couch

- Augmented and Virtual Reality (AR/VR) is thriving in various fields
- 360 video (aka spherical video, omnidirectional video) provides visual experience from all direction
- People experience VR with head-mounted displays (HMDs) for more immersive viewing experience

What’s limiting us?

360 Video Limitations [1]

1 Fixed focal length

- Traditional images/videos have their default focal length that *can't* be adjusted
- Objects' proper distances from the eye gaze cannot be recognized when wearing HMDs

Discomfort!

- Require **focal length adaptation** to solve the issue

2 Fixed viewpoint

- Images/videos are always taken from single viewpoint
- Scenes in HMDs won't change when users move their head/body
- Conflict between what the body feels and what the eyes see

Discomfort!

- Use **multi-viewpoint video** to solve the issue

J. Moss, J. Scisco, and E. Muth. Simulator sickness during head mounted display (HMD) of real world video captured scenes. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 52(19):1631–1634, September 2008.

Focal Length Adaptation

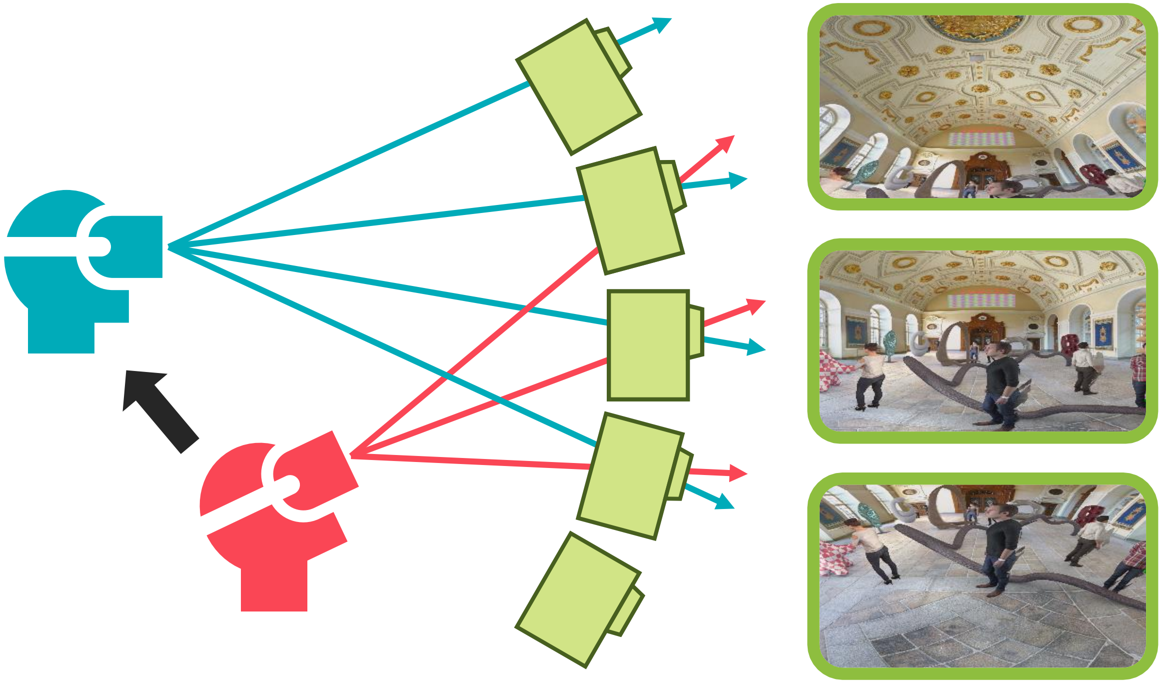
"Adapt the focal plane based on where user is gazing"

Short focal length

Long focal length

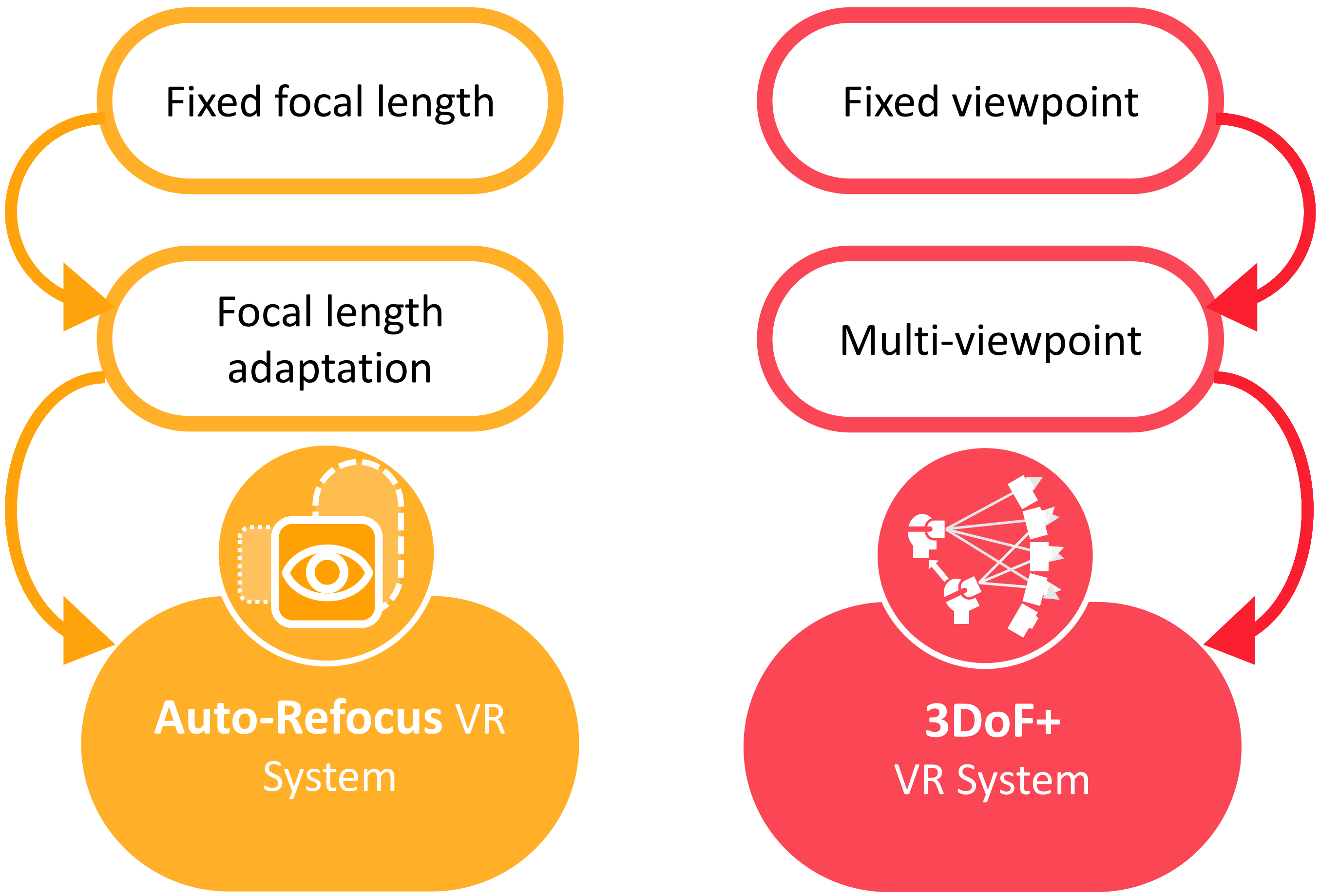
Multi-Viewpoint Video

" Take images from multiple different viewpoints "



Establish 2 Systems

For handling the limitations



Research Problems

Auto-Refocus VR System

A

Refocusing images in real-time

- Refocus image with gazing coordinate
- Optimize the refocus processing for smooth video playout

3DoF+
VR System

B

Selecting the proper views for view synthesis

- Reduce the reference view number
- Consider not only geometry but also 3D space coverage

Contributions

1. Propose an auto-refocus panorama system based on user eye gaze
2. Propose a novel view selection algorithm for 3DoF+ systems
3. Establish real systems and evaluate their performance

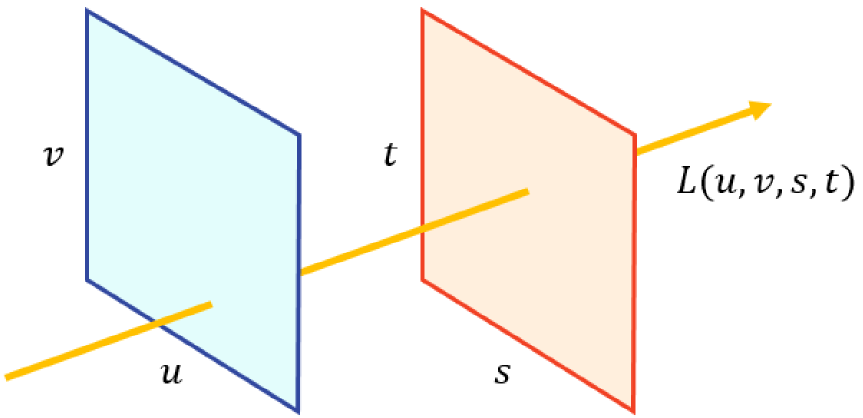
02

Background

Light Field [2]

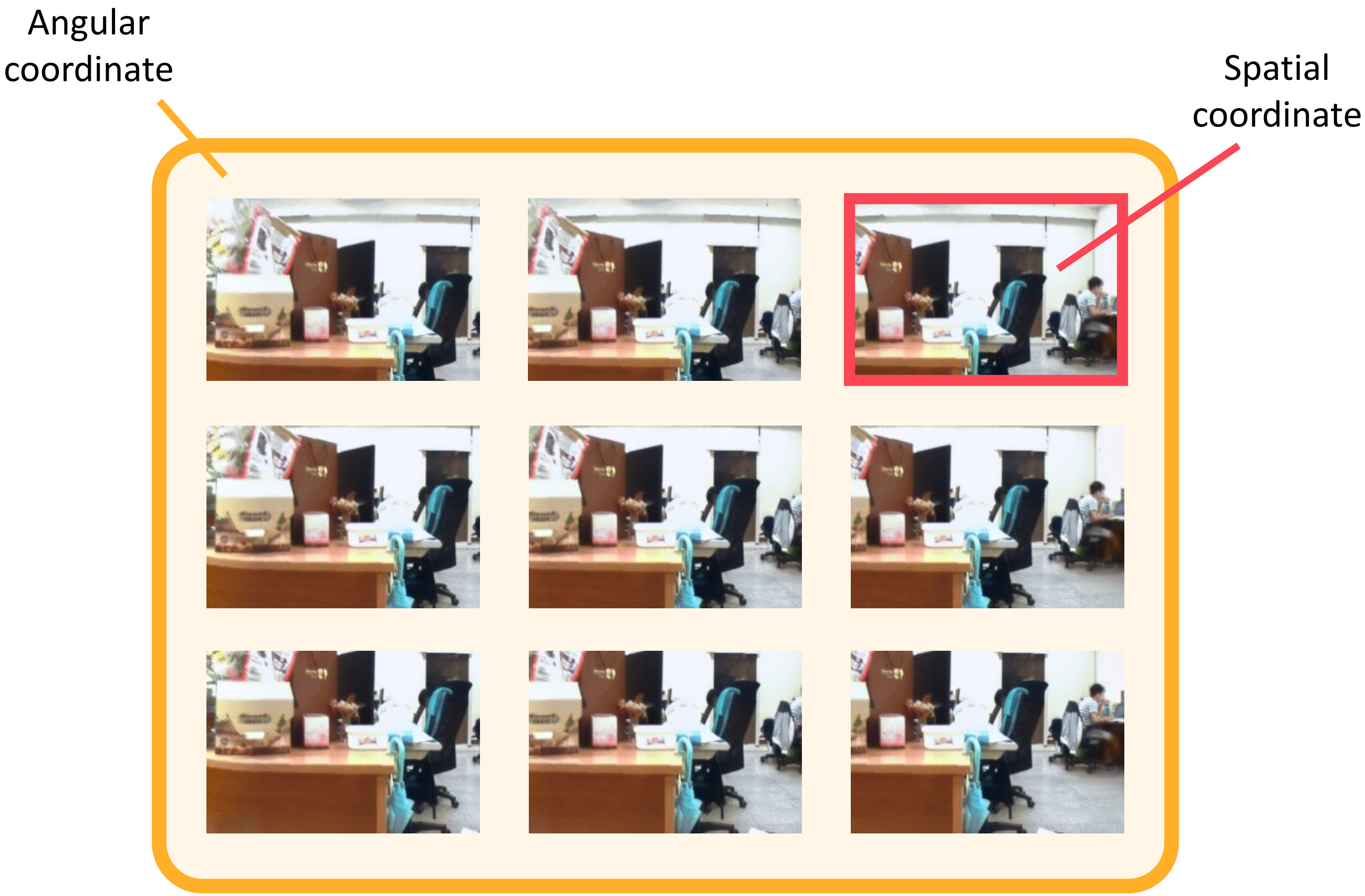
- 4D data format, including angular and spatial information
- UV: angular coordinates; ST: spatial coordinates
- **Scene synthesis**, **image refocusing**, depth estimation…

> "Capture all lights from all directions"



[2] M. Levoy and P. Hanrahan. Light field rendering. In Proc. of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96), pages 31–42, New Orleans, USA, August 1996.

Sub-Aperture (View) Image

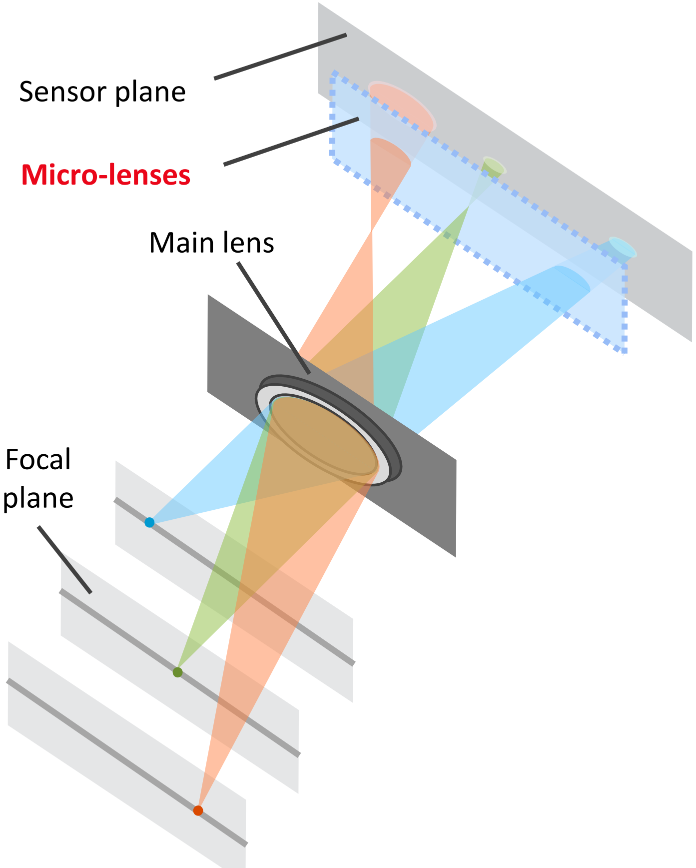


Micro-lenses Camera System

- A **micro-lens array** is placed in front of the image sensor
- Disperse the light information into **different views**
- A rather *small* scale of light field
- Small sub-aperture image disparity

Plenoptic camera

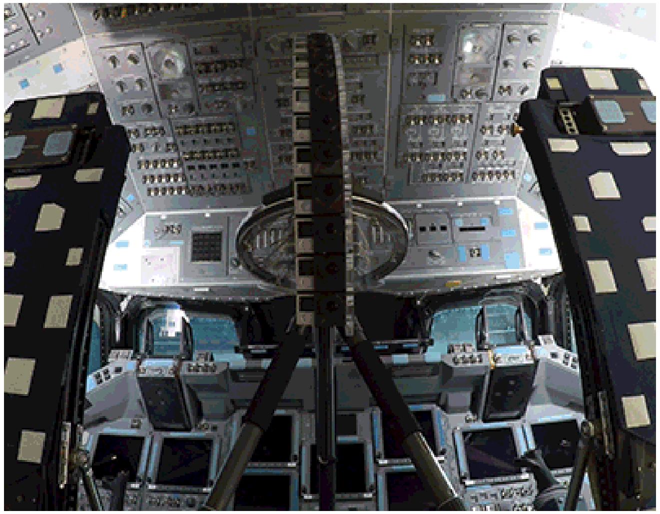
Lytro Illum

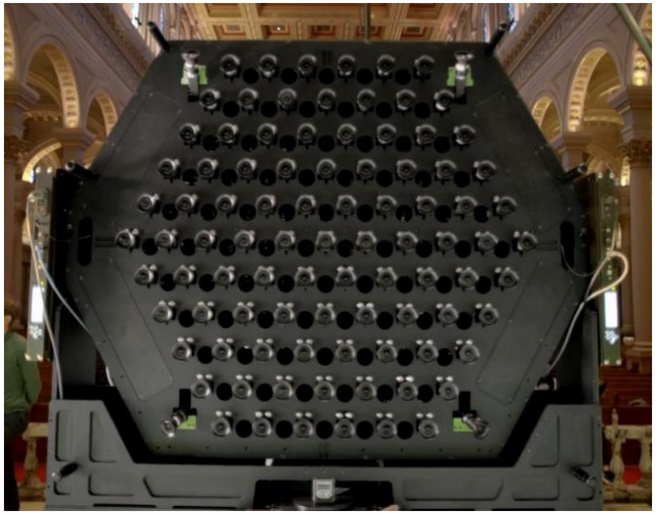


Light-field camera

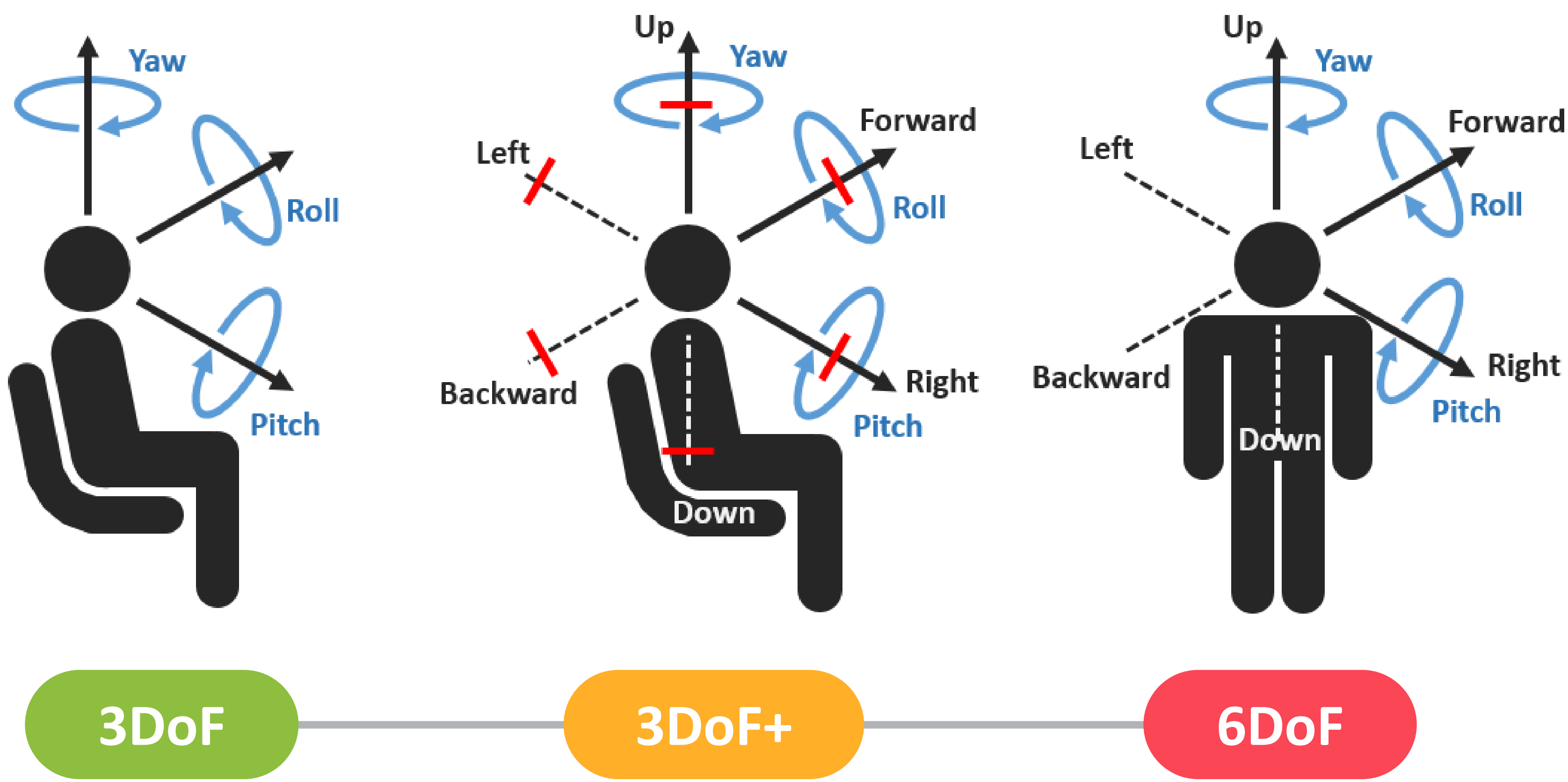
Camera Array System

- Align multiple cameras (straight line, spherical, random) in space to capture light information from **different perspectives**
- Resemble to the **multi-view system**
- A much *larger* scale of light field





3DoF to 6DoF [4]



X. Wang, L. Chen, S. Zhao, and S. Lei. From OMAF for 3DoF VR to MPEG-I Media Format for 3DoF+, Windowed 6DoF and 6DoF VR. ISO/IEC JTC1/SC29/WG11 MPEG2017/M41197, 2017. Meeting held at Torino, Italy.

03

Auto-Refocus VR System

Overview [5]

1. Use a **eye movement detection-support HMD** for focal length adaptation

2. Get **LF panoramas** by stitching LF images

3. Utilize LF properties for **image refocusing**

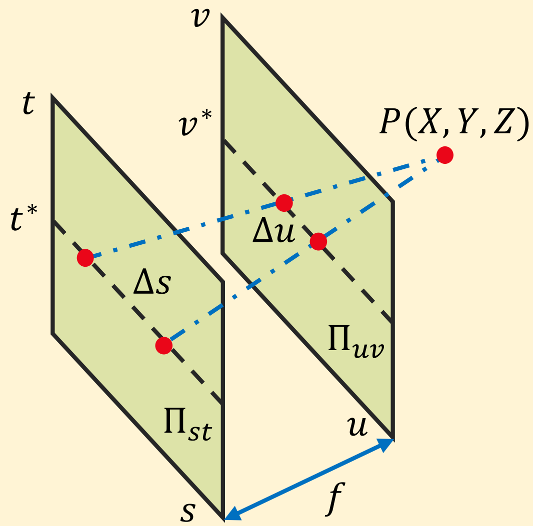
4. Require scene **depth map** for accurate focal length extraction

Y. Lai and C. Hsu. Refocusing Supports of Panorama Light-Field Images in Head-Mounted Virtual Reality. In Proceedings of the 3rd International Workshop on Multimedia Alternate Realities, AltMM'18, pages 15–20. ACM, 2018.

LF Image Refocusing

Shift-Sum Algorithm [6]

1. Calculate the pixel-shift amount of a sub-aperture image based on its angular coordinate (u, v) and target depth Z



$$slope = \frac{Z}{Z - f}$$

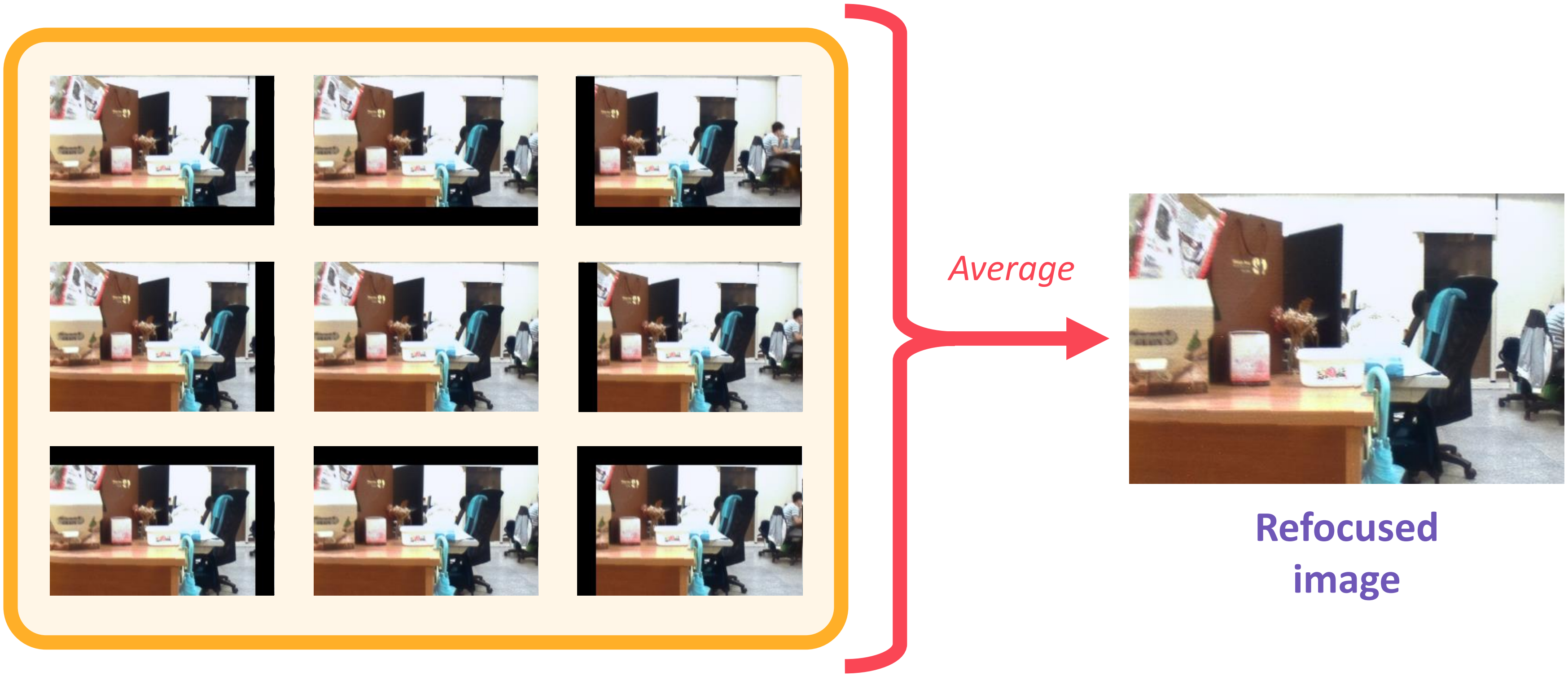
$$u_{\text{shift}} = U \times slope \times \left(\frac{u}{U-1} - \frac{1}{2}\right)$$

$$v_{\text{shift}} = V \times slope \times \left(\frac{v}{V-1} - \frac{1}{2}\right)$$

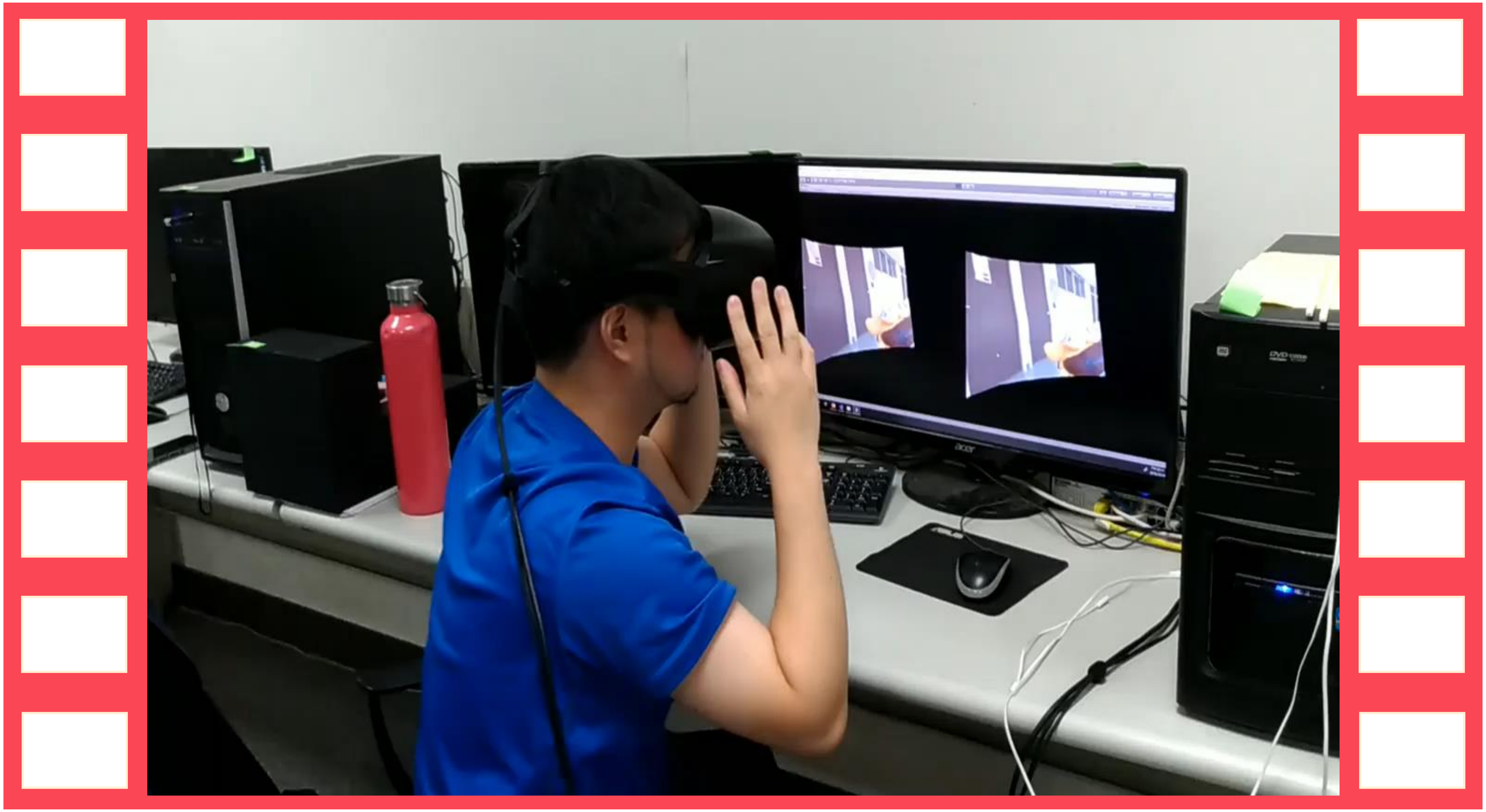
2. Shift each sub-aperture image by the corresponding pixel-shift amounts
3. Calculate the average value of the images pixel-wisely and get the result

R. Ng, M. Levoy, M. Br´edif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Stanford Tech Report CTSR 2005-02, 2005.

Shift-Sum Algorithm

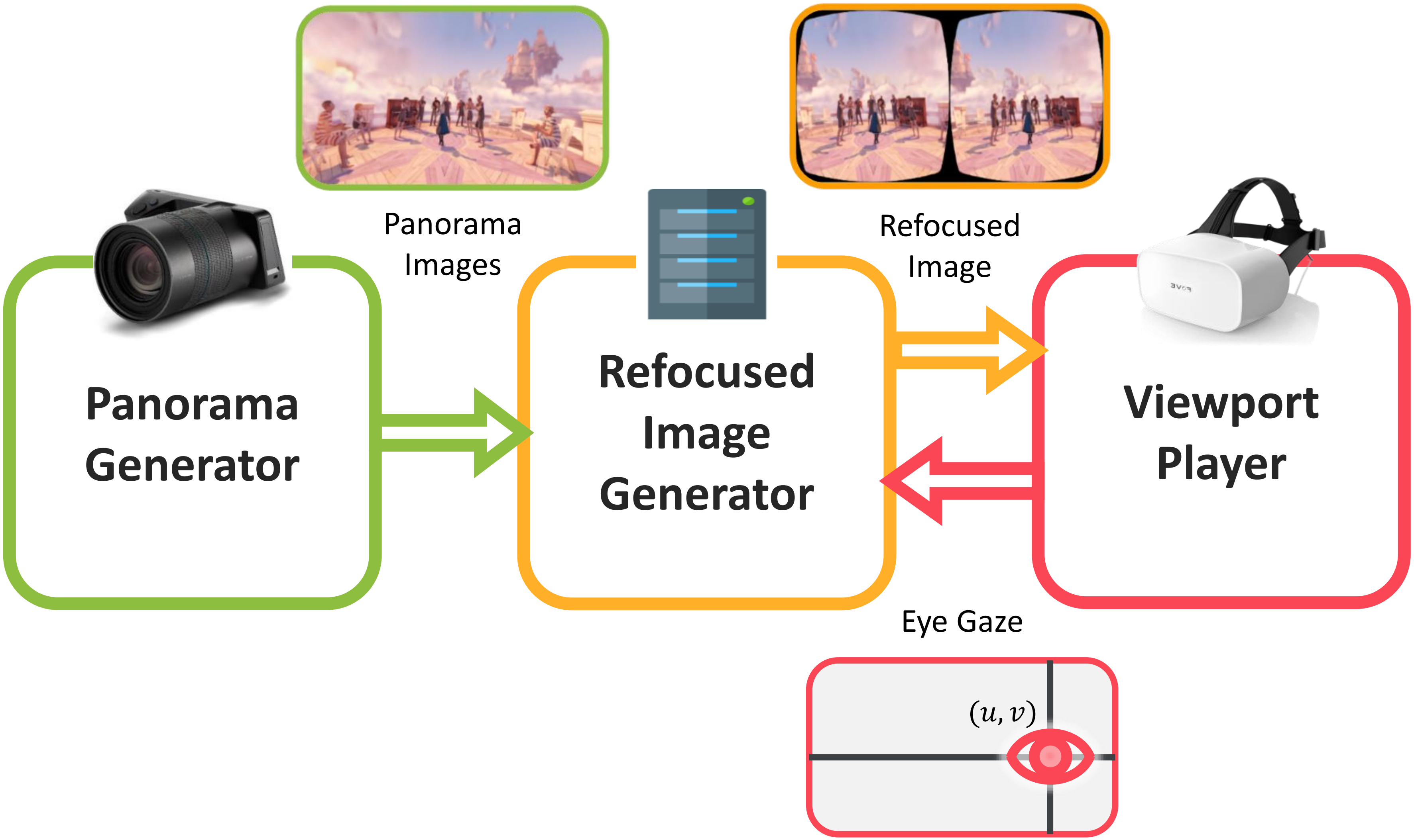


Demonstration



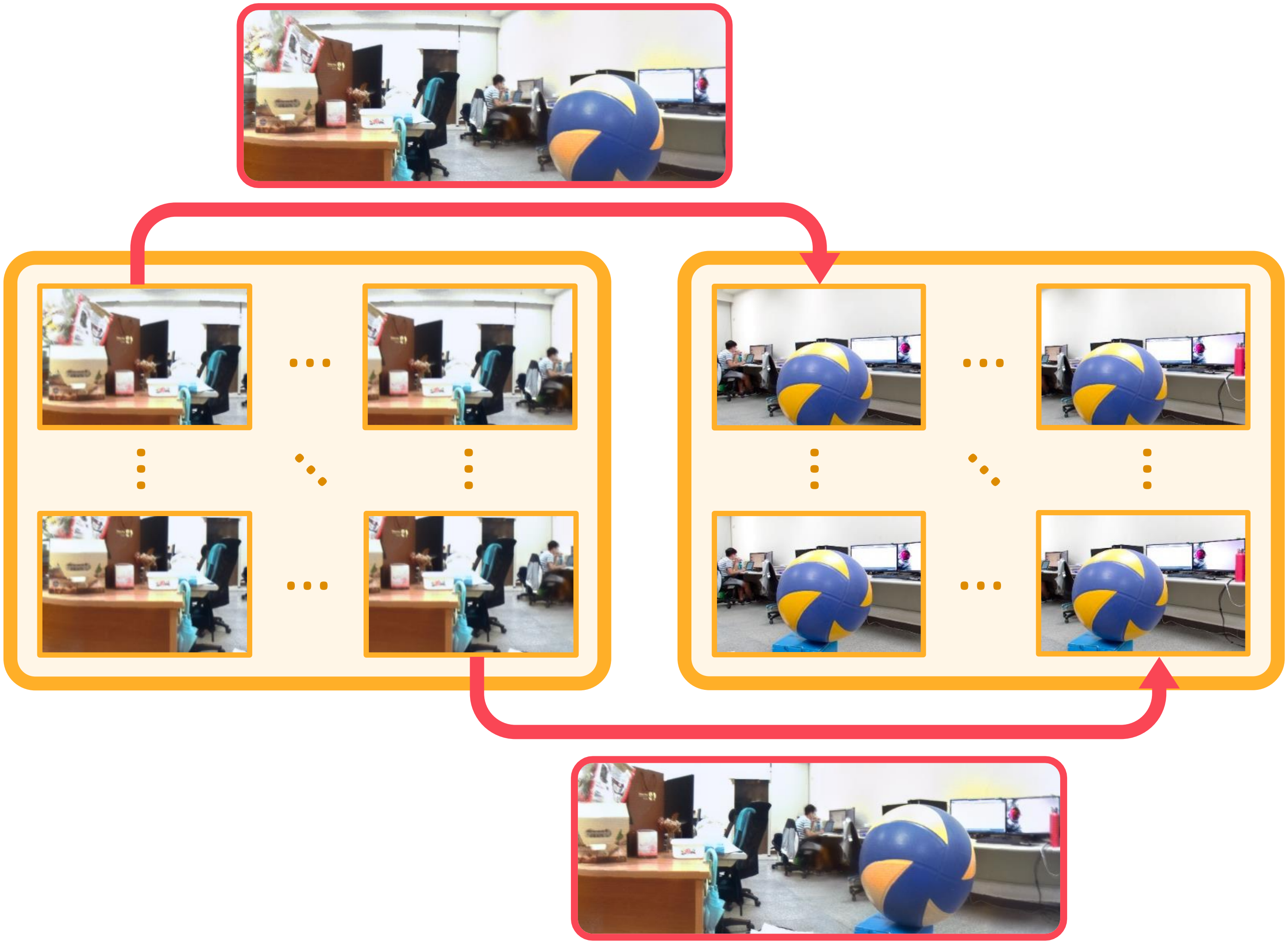
YouTube Link

System Architecture



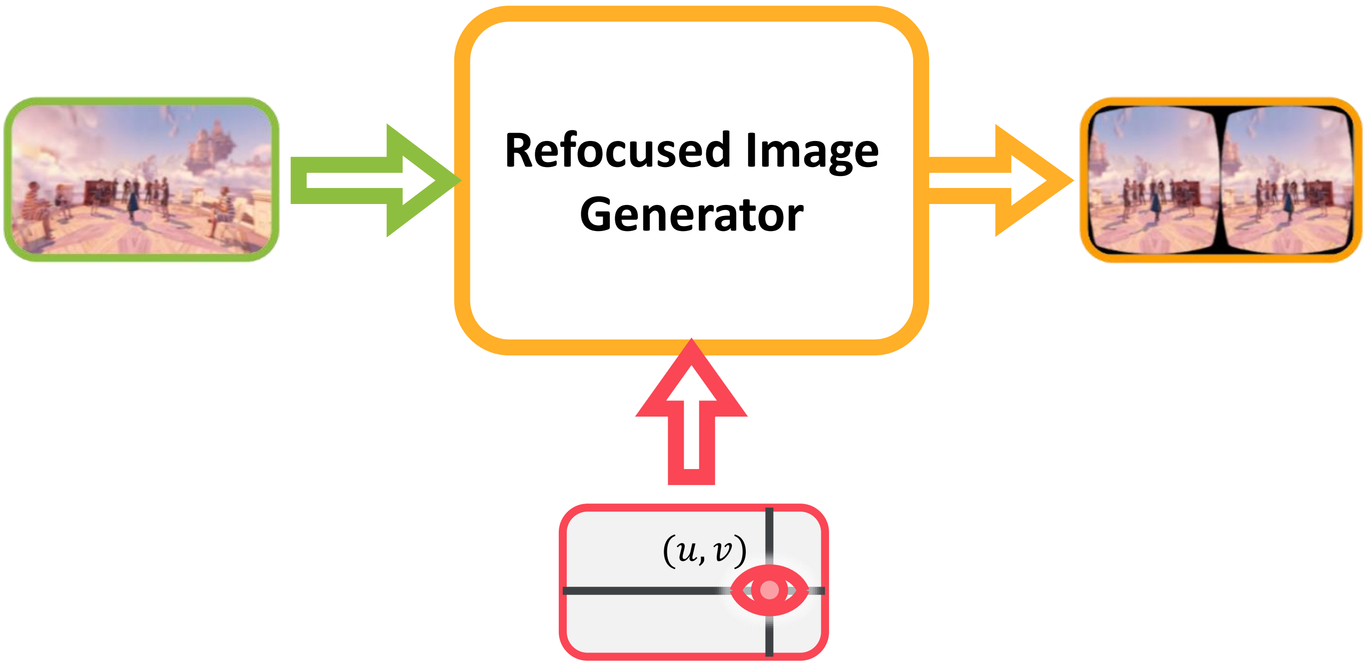
Panorama Generator

LF Panorama Stitching

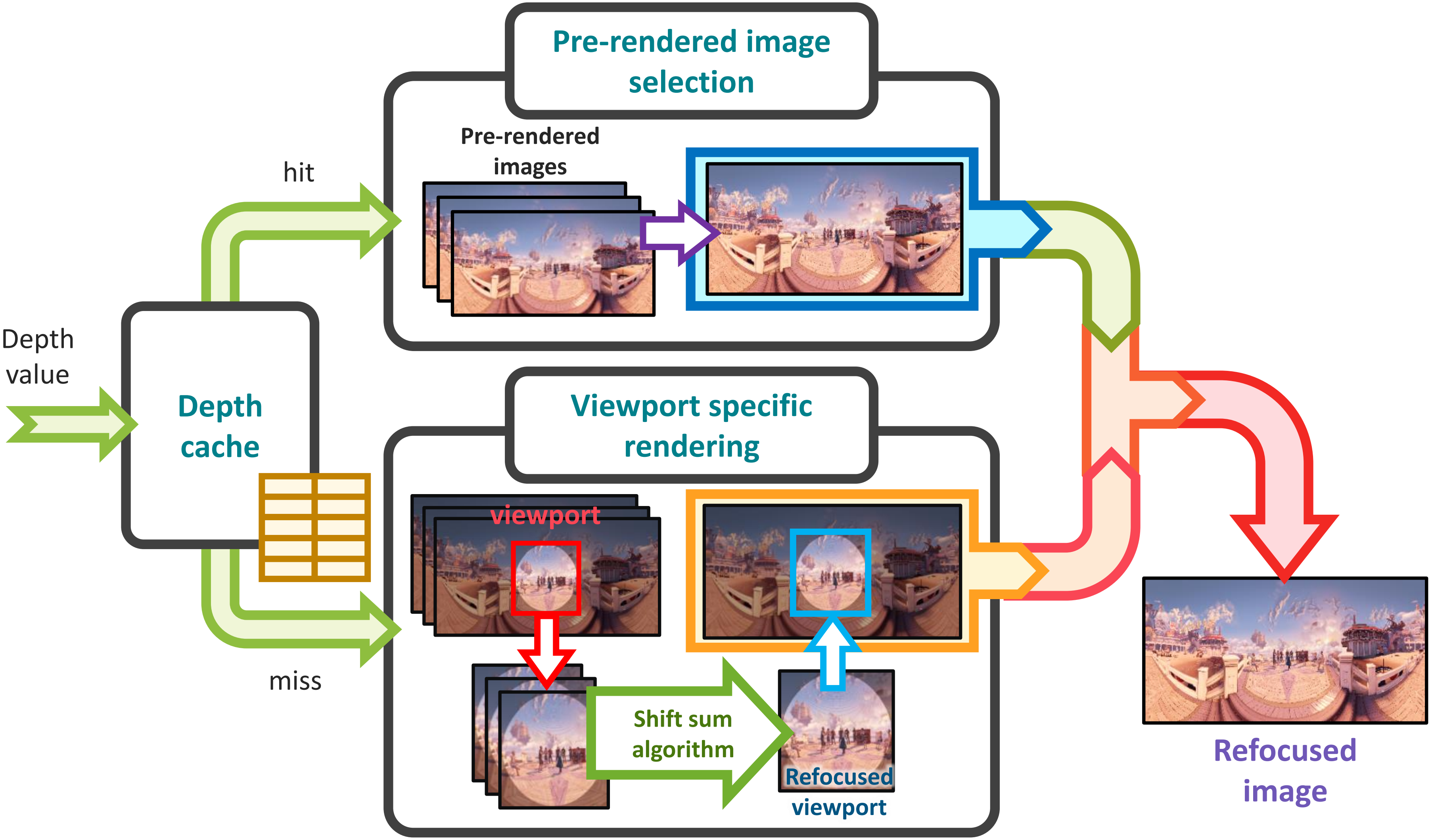


Refocused Image Generator

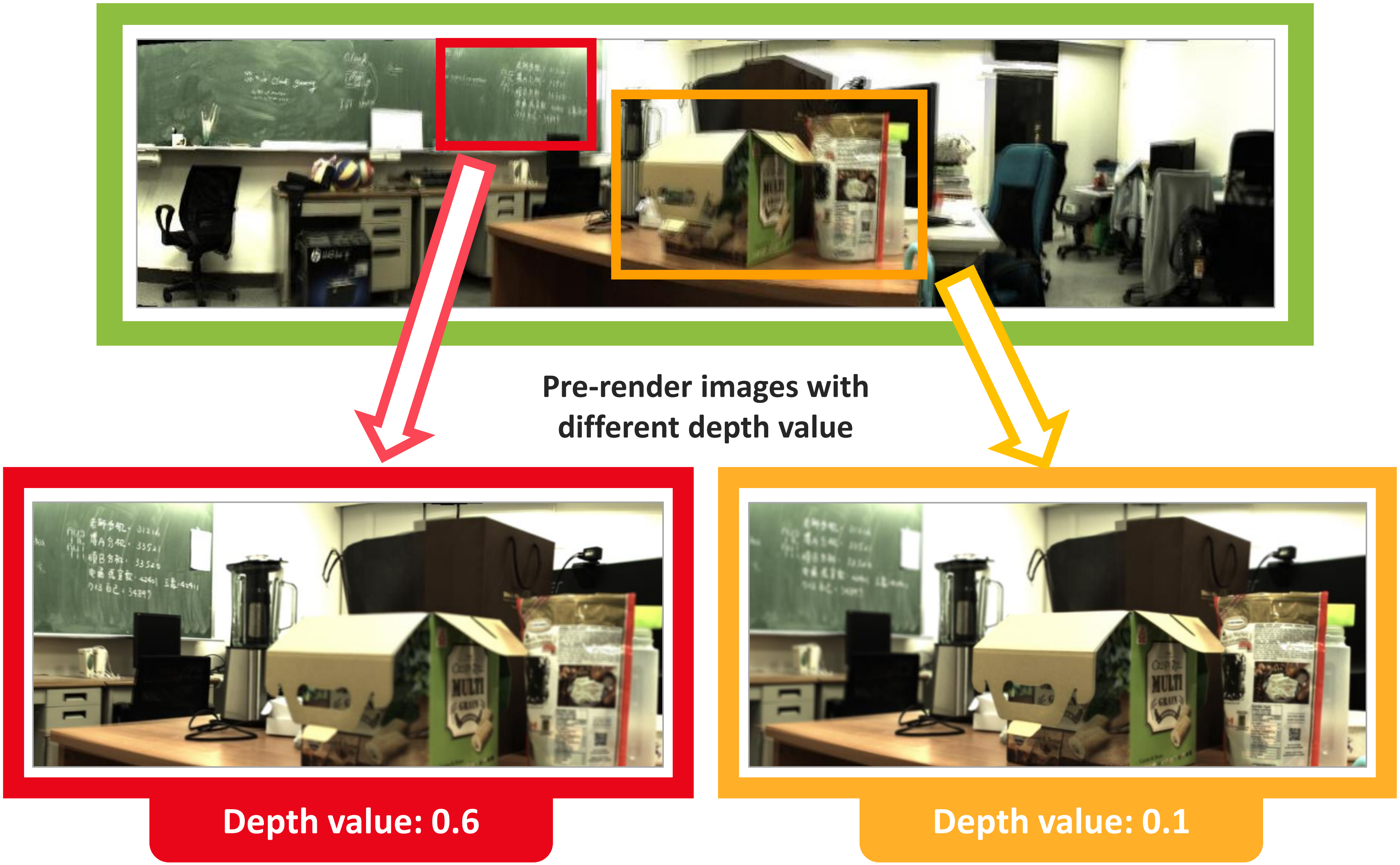
- Generate the refocused image based on the gazing coordinate from viewport player
- Two optimization schemes are proposed
  - **Pre-rendered image caching**
  - **Viewport specific rendering**



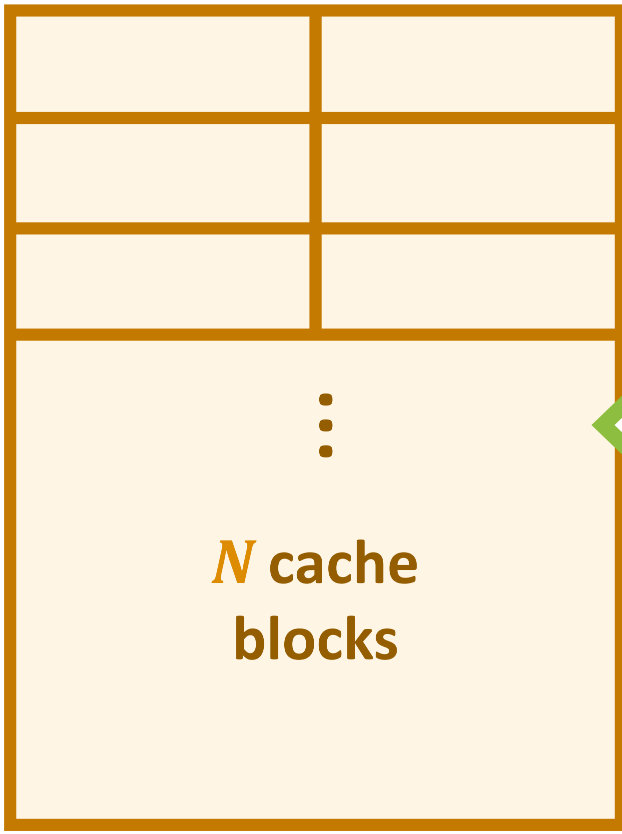
Light Field Processing



Pre-rendered Image Caching



Depth Cache

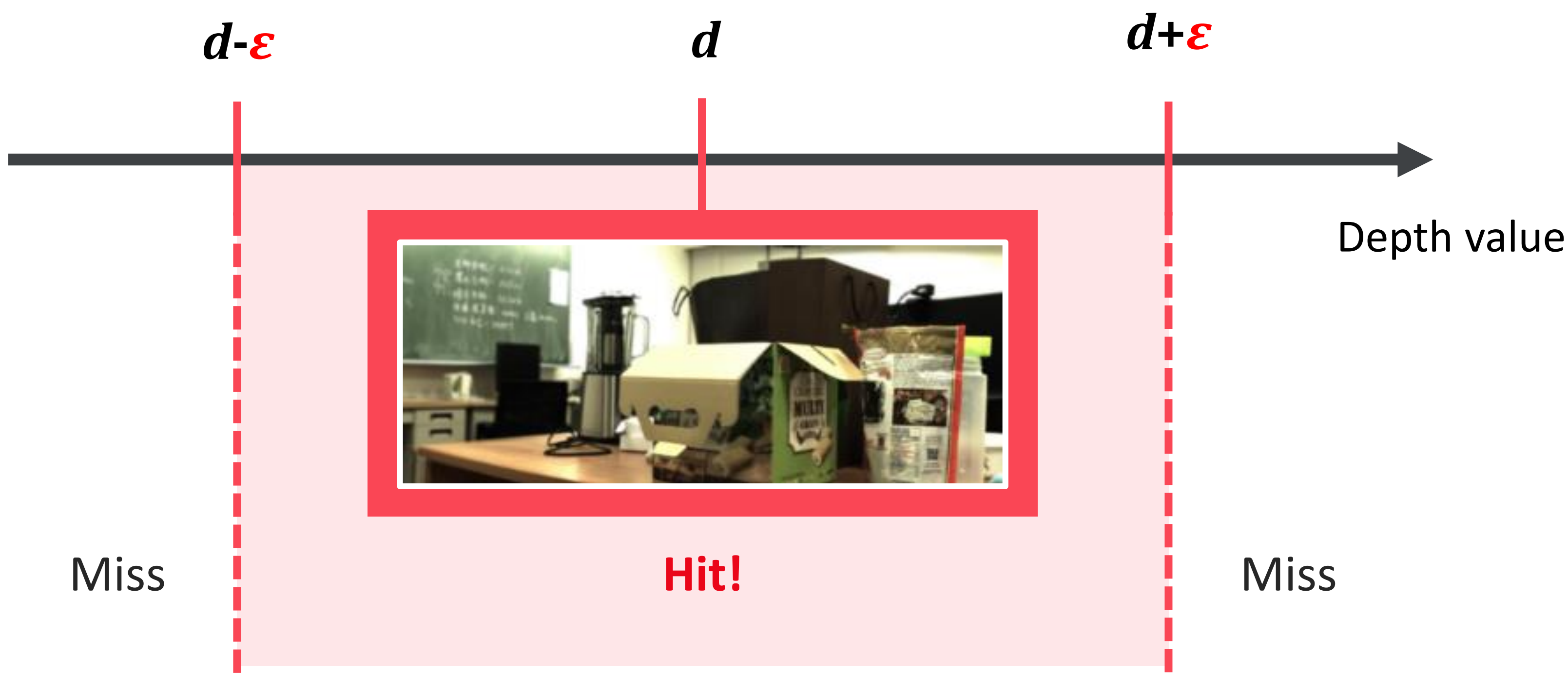


Caching information
(in each block)

- Selected depth value
- Pre-rendered image corresponding to the value

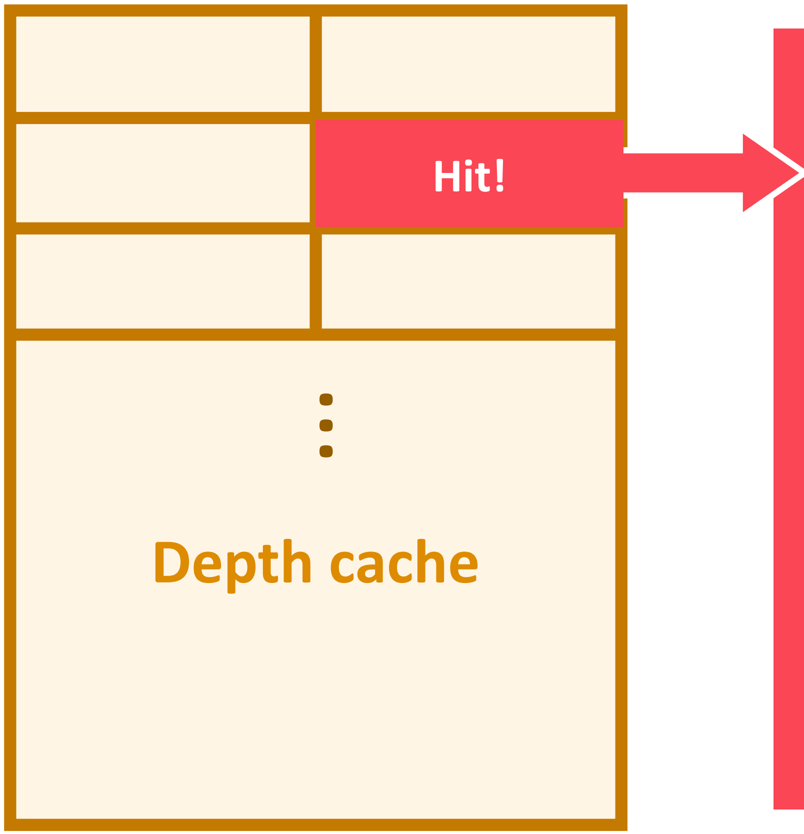
Depth Tolerance ε

Determine whether a target depth value
hit the cache



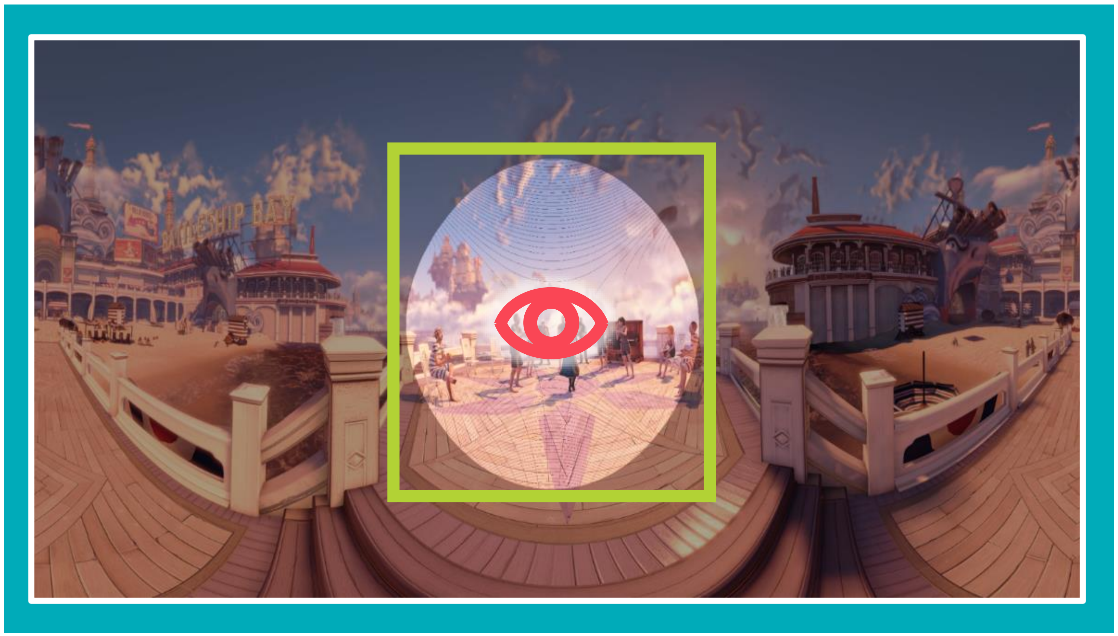
d: depth value in cache

Pre-rendered Image Selection

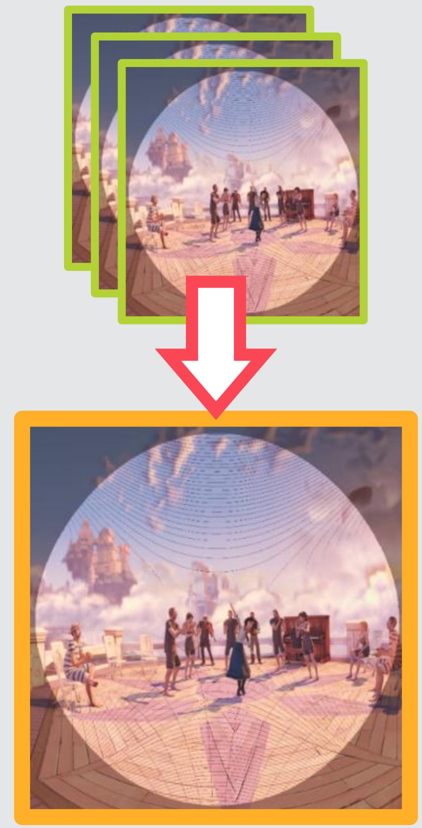


- Extract the **pre-rendered image** from cache
- Directly use the image as the result of the target depth

Viewport Specific Rendering



Sub-aperture images of viewport



- The viewport takes only about 15% of the whole panorama image (FoV = 100˚)
- Only apply refocusing process to the viewport region

Implementation

Panorama Generator	LYTRO
Refocused Image Generator	unity OpenCV# for Unity
Viewport Player	FOVE

Experiment Design

Objective

Metrics

Processing latency

Baseline

Refocusing process w/o optimization

Dataset

Light field panorama scene of our lab

Subjective

Metrics

Mean opinion score (MOS)

Baseline

Central sub-aperture image

Dataset

Light field panorama scene of our lab

Performance

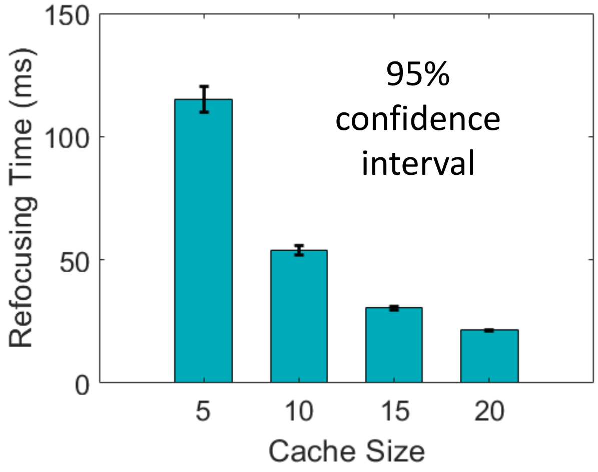
Light field size: **5x5x1920x3840x3**

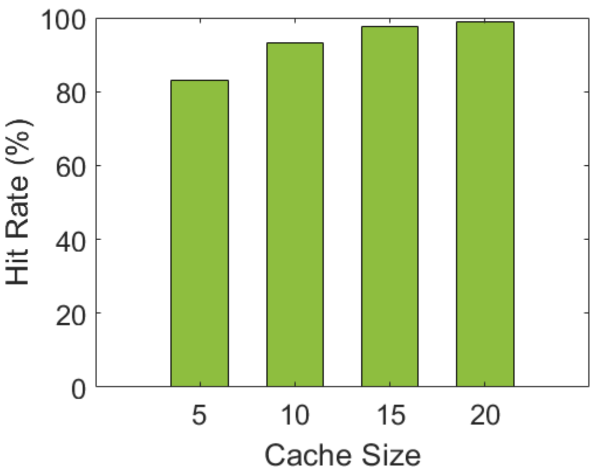
Average value of **300 refocused images**

W/o Optimization	Pre-rendered Image Selection	Viewport Rendering
Per Process	Cache hit	Cache miss Process 1056x1034 pixels
5.56s	355x ⬇ 16ms	9.4x ⬇ 594ms
stdev: **484.2ms**	stdev: **0.82ms**	stdev: **102.7ms**

Different Cache Size (N)

- Average refocusing time drops as N grows
- More candidate images for selection
- Average results from 3 runs

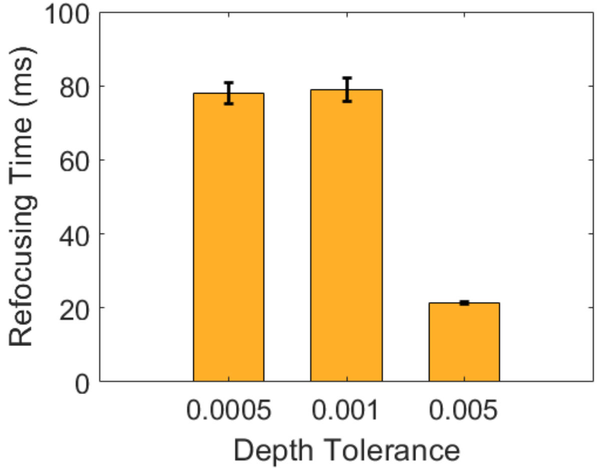


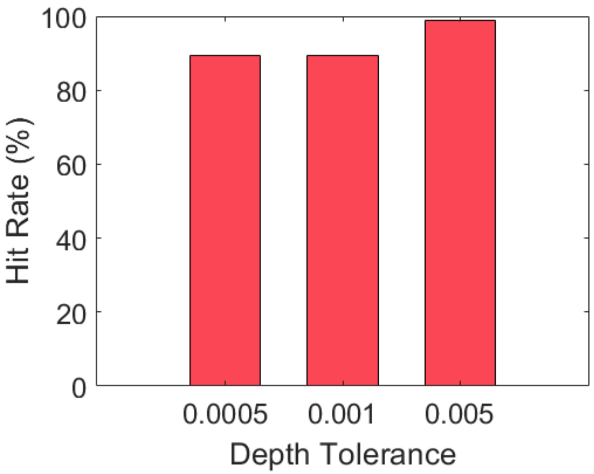


- Hit rate grows as N grows
- The hit rate is high because the depth map is simple
- Average data of 3 runs

Different Depth Tolerance (ε)

- Refocusing time grows as ε drops
- The first two bars are similar because the hit rate are the same
- Average data of 3 runs





- Hit rate drops as ε drops
- The hit rate of the first two bars is small enough to make the difference
- Average data of 3 runs

User Study

10 users
(7 males, 3 females)

Q How much do you like the images? (1~5)

Proposed System

Auto-Refocus VR
with eye gaze

Avg. MOS: **2.8**

Stdev: 0.876

>

P-value: 0.0383

Baseline System

Central sub-aperture
image

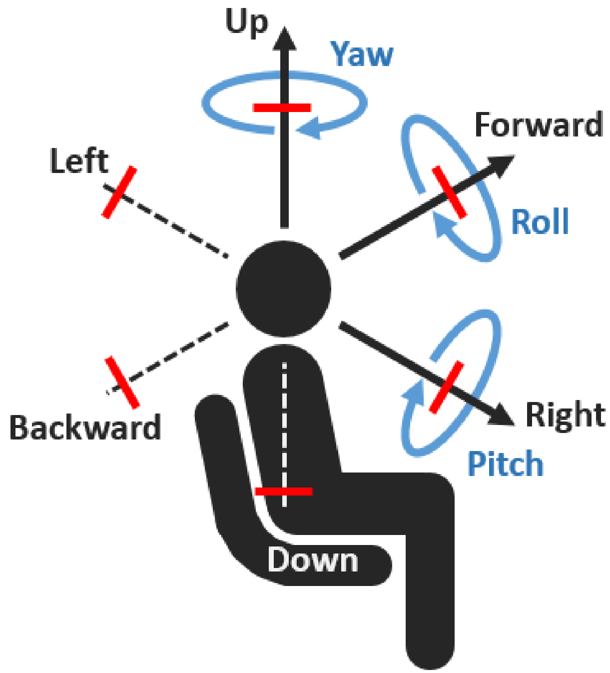
Avg. MOS: **2.3**

Stdev: 0.789

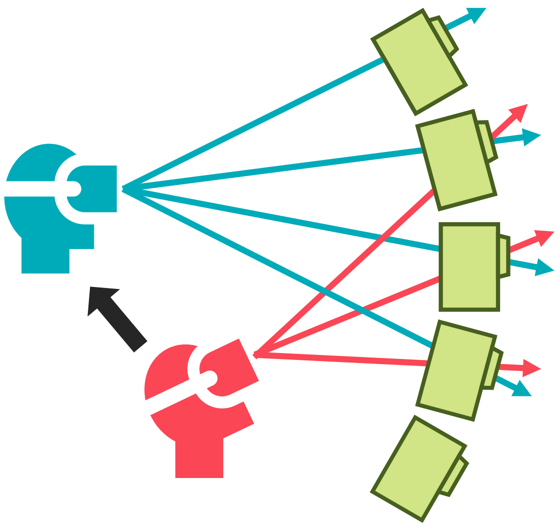
04

3DoF+ VR System

3DoF+ VR



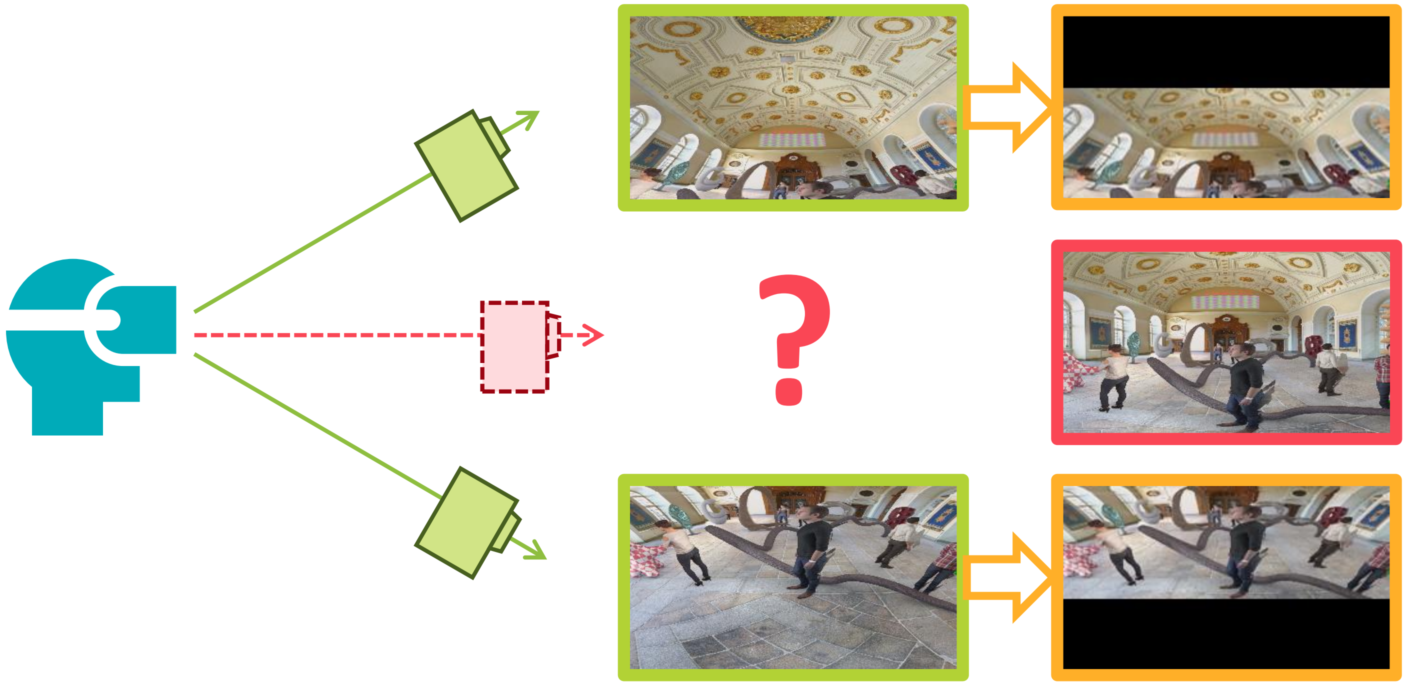
Allow viewpoint changes in a certain scale



Camera array provide information from different viewpoints

Q How to get scenes from different viewpoint?

View Synthesis [7]



- Use **reference view** to generate the **target view** based on their parameter
- **3D view warping** ~~blen~~**ding**
- Heavy computation and time-consuming

S. E. Chen and L. Williams. View interpolation for image synthesis. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH ’93), pages 279–288. ACM Press, 1993.

Why View Selection?

A View synthesis algorithm's complexity is highly relied on the **number of used reference views**

We want to choose only **relevant views** to the synthesis process

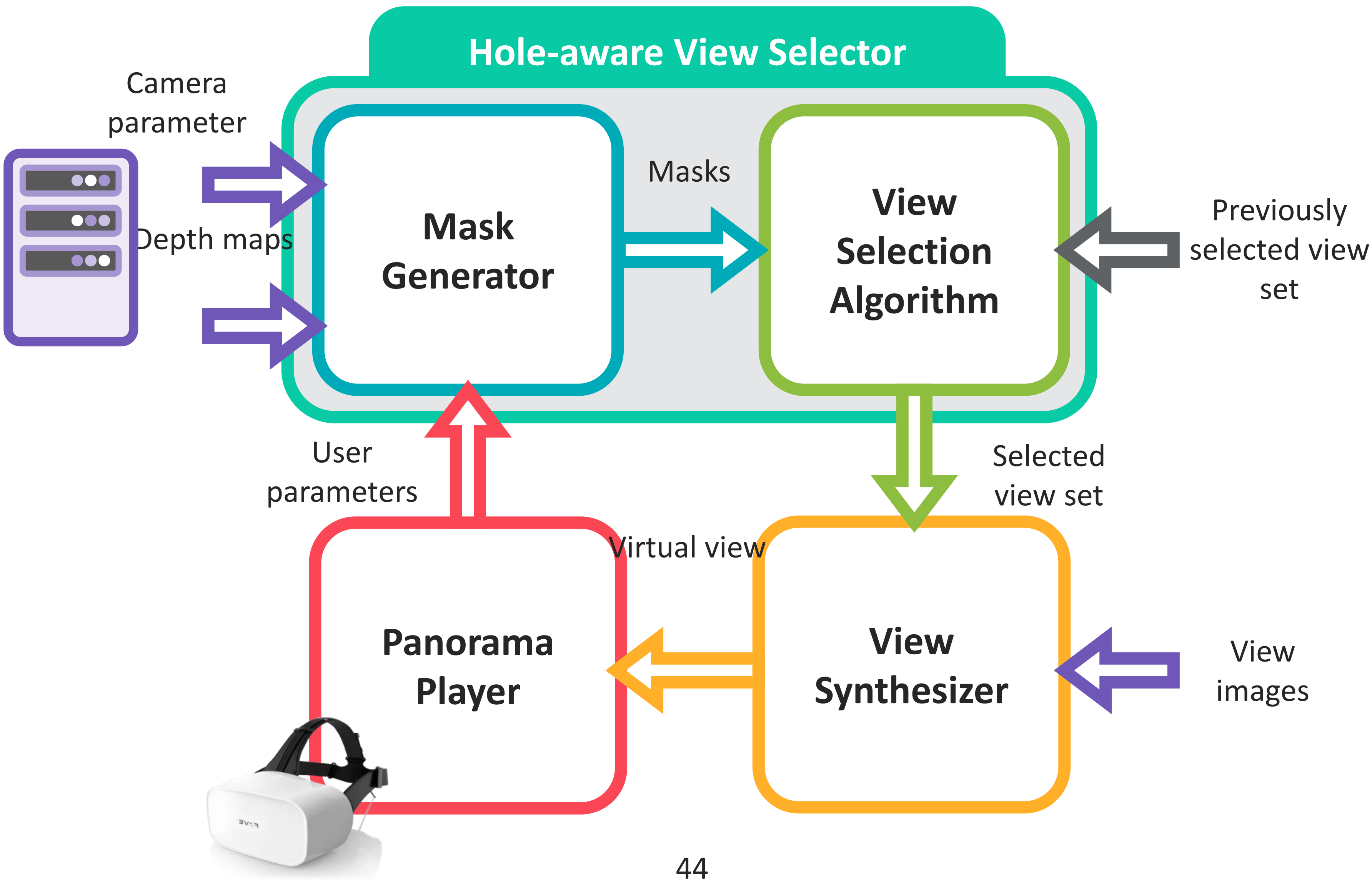
B Only select with geometry relationship may fail due to **the lack of space information**

We need to select the views based on the space information they have

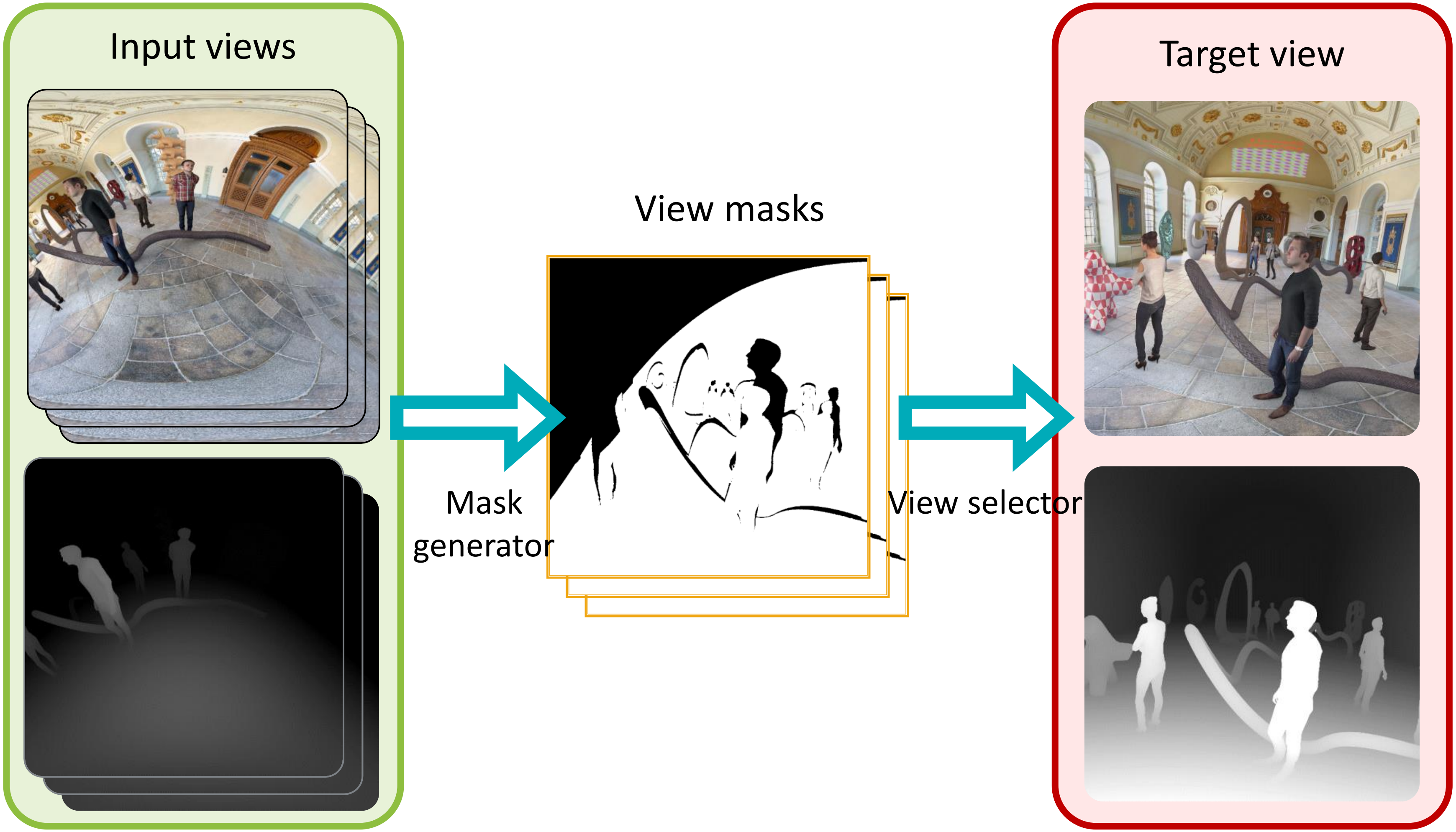
It's important to find an **effective and efficient view selection** method!

Demo Video

System Architecture

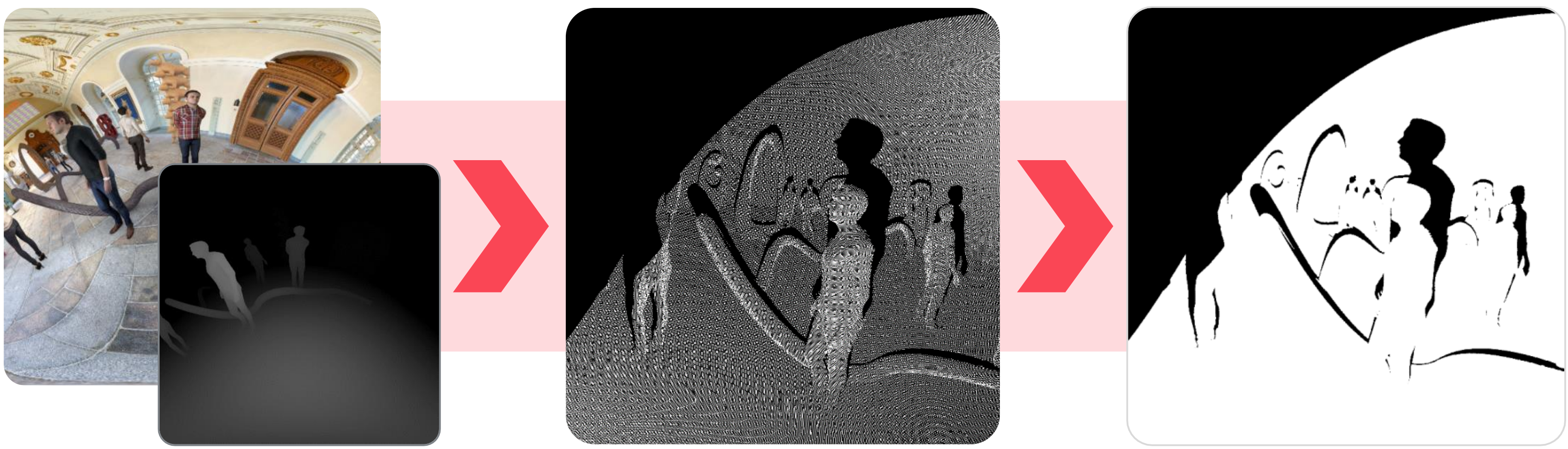


Hole-Aware View Selector



Mask Generator

- To better leverage the 3D information of each reference view
- Binary mask:
  - **1** if the pixel is covered by the reference view, **0** otherwise
- Generation process
  1. 3D warping
  2. Hole filling



3D Warping [8]

> A technique that is used for target viewpoint synthesis based on the camera parameters

1. Unproject the 3D coordinates for each pixel from the view depth map

2. Apply affine transform to the coordinates to change the coordinate system

3. Project the transformed 3D coordinate back to the 2D image

S. Avidan and A. Shashua. Novel view synthesis in tensor space. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1034–1040, June 1997.

Hole Filling

Two kinds of holes!

1. Caused by **view coverage limitation**
2. Caused by **point cloud discontinuity**

What we want to fill!

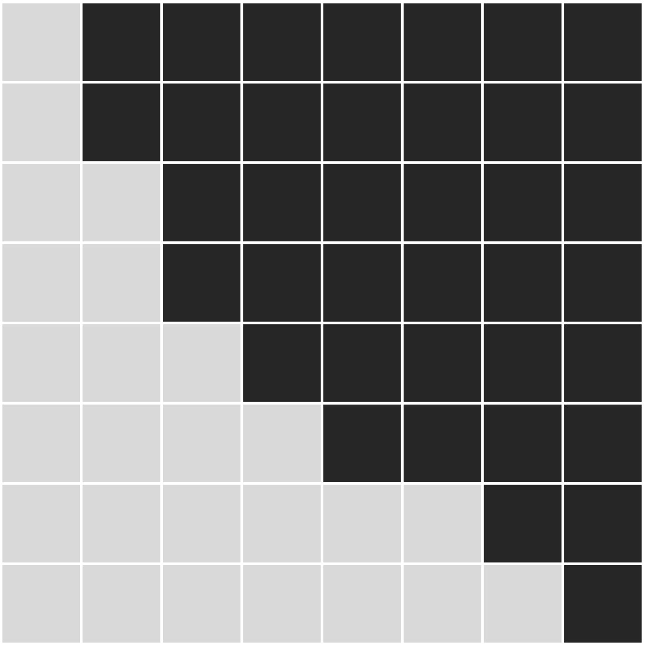
How to fill the holes?

- Apply **convolution** to calculate the **coverage density** around each pixel
- Apply **binarization** to each pixel with the corresponding density

Hole Filling

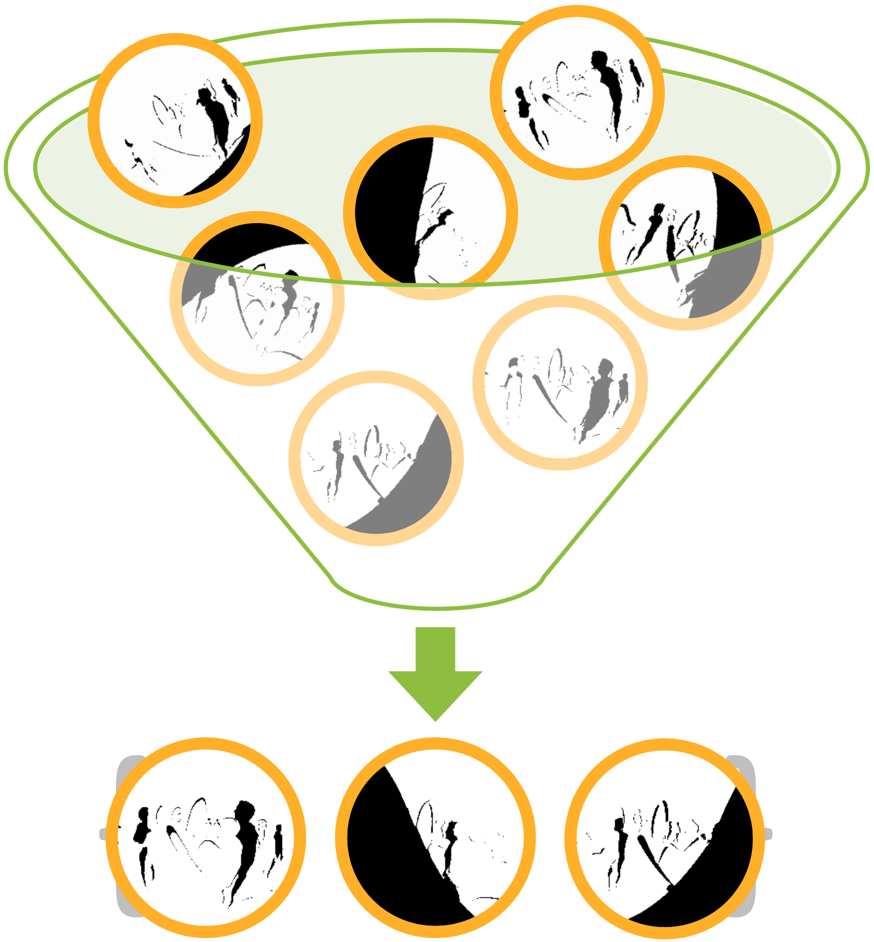
Notation	Description
k	Size of the kernel of the convolution
d	Coverage density of a pixel
τ	Threshold for the hole filling

- Let $k = 3, \tau = \frac{k^2}{2}$
- 2D kernel size: (k, k)
- All scalars in the kernel are 1 (summation)
- Fill up a hole if its coverage density d is bigger than τ



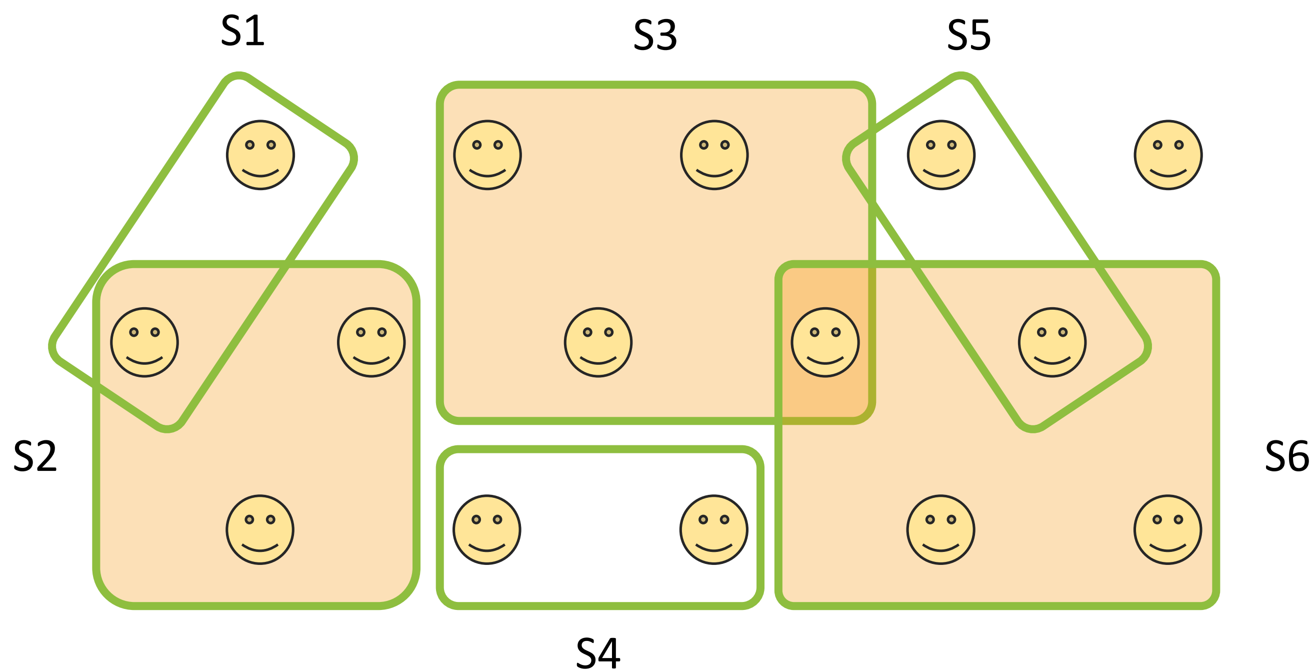
View Selection Algorithm

“Find the view set with limited members that leads to the best synthesis result”



Maximum Coverage Problem [9]

Universe => all elements



Cover as many elements as possible within certain number of sets

Optimal set: **{S2, S3, S6}**

V. Vazirani. Approximation algorithms. Springer, Berlin New York, 2001.

Problem Formulation

Notation	Description
$\mathbf{M}$	Collection of the covered pixels in the masks
$\mathbf{S}$	Set of the selected views
C	Image used for union in each selection stage
$\mathbf{T}$	Union results of image C and all mask in $\mathbf{M}$
r	Coverage scores of $\mathbf{T}$
k	Maximum size of $\mathbf{S}$

Objective Function

$$\max \left| \bigcup_{s \in \mathbf{S}} \mathbf{M}_s \right|, s.t. |\mathbf{S}| \leq k$$

Greedy Solution [10]

Classic MCP solution

Best polynomial time algorithm unless **P=NP**

- Initialize a canvas C with all pixels set to 0
- In each of k stages, get union of $\mathbf{M}_c$ and all masks $\mathbf{M}$
- Find the union result that contains the largest number of not yet covered pixels

Complexity:

$O(|\mathbf{M}| \times k)$

Approx. ratio:

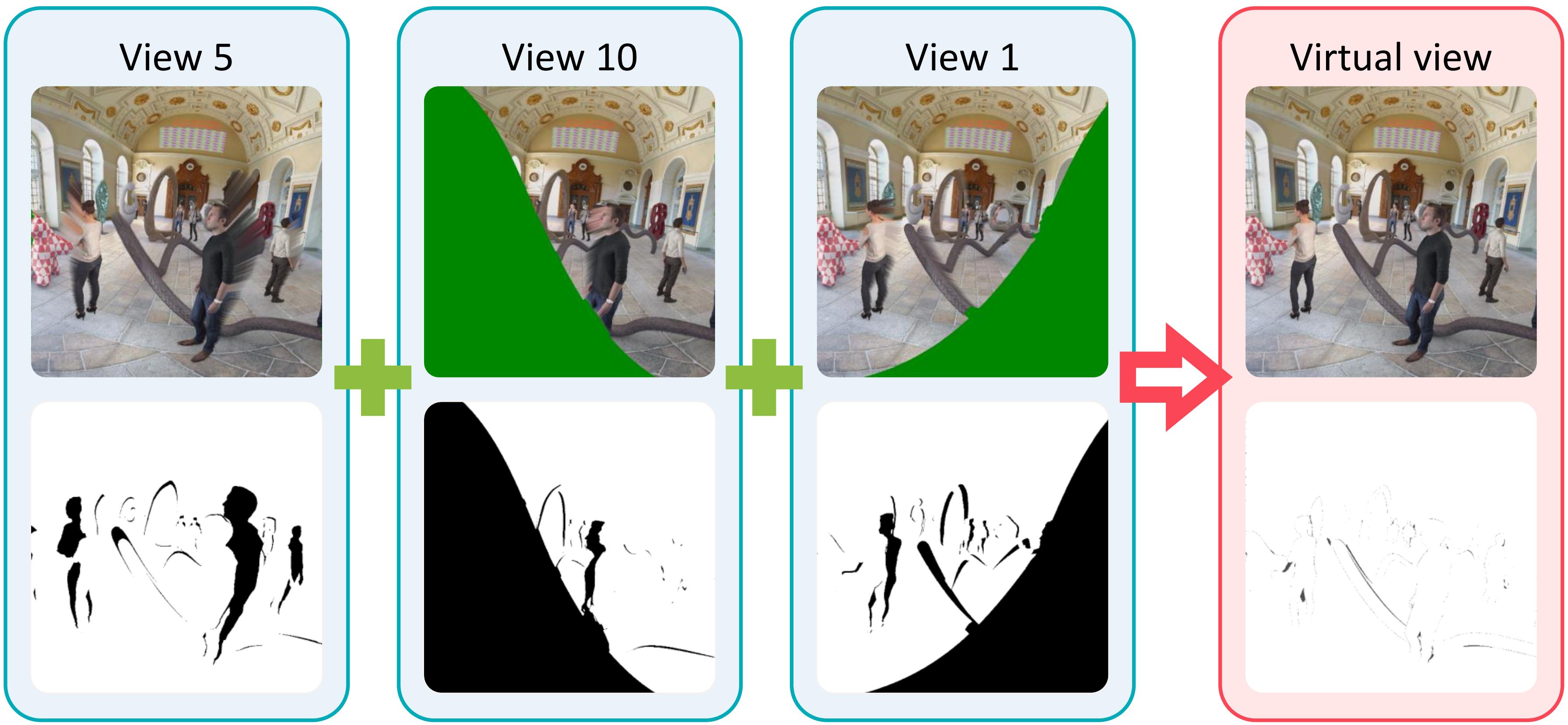
1 – 1 / e ~0.632

Pseudo code

$C \leftarrow$ canvas filled with value-0 pixels in the same size of views
for i from 0 to $k-1$ **do**
    for n from 0 to $|\mathbf{M}|-1$ **do**
        $t_n \leftarrow \mathbf{M}_n \cup C$
        $r_n \leftarrow$ number of value-1 pixels in t_n
    $idx \leftarrow$ index of the max element in r
    $S_i \leftarrow idx$
    $C \leftarrow t_{idx}$
return $\mathbf{S}$

A. A. Ageev and M. I. Sviridenko. Approximation algorithms for maximum coverage and max cut with given sizes of parts. In Integer Programming and Combinatorial Optimization, pages 17–30. Springer Berlin Heidelberg, 1999.

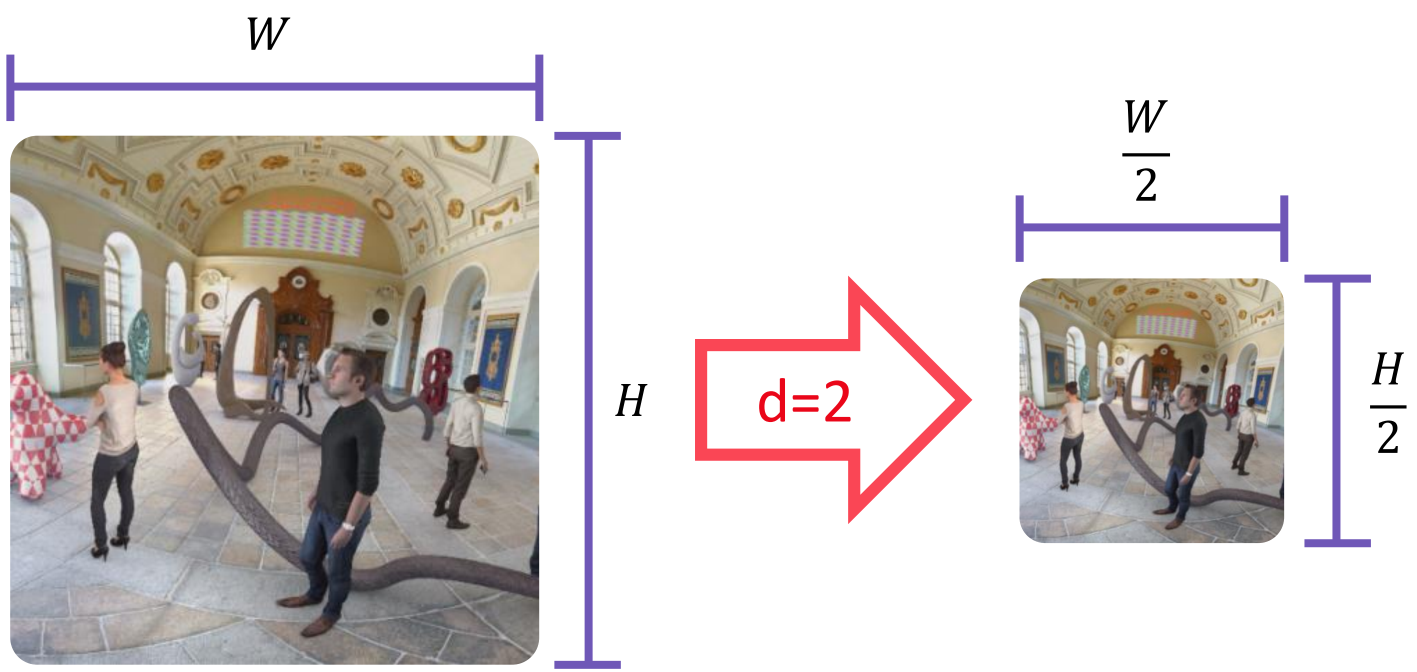
Example



92.56% 54.82% 67.83% 99.37%

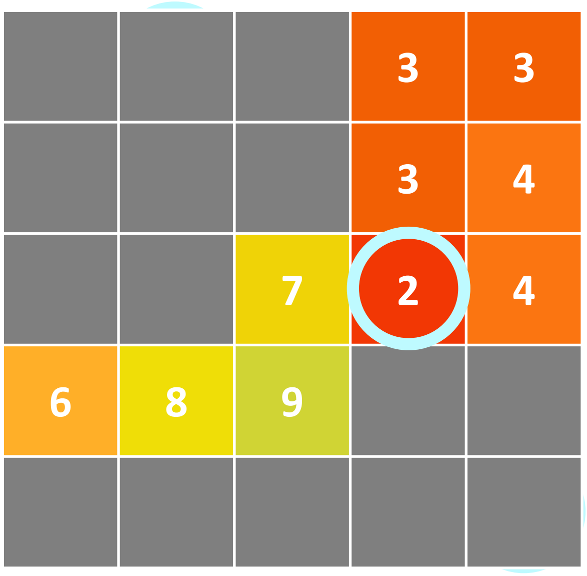
Down-Sampling

- Down-sample the used reference views to reduce the number of processed pixels
- d: determine ratio the view is down-sampled



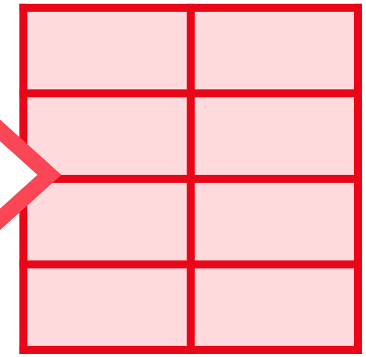
Other Solutions

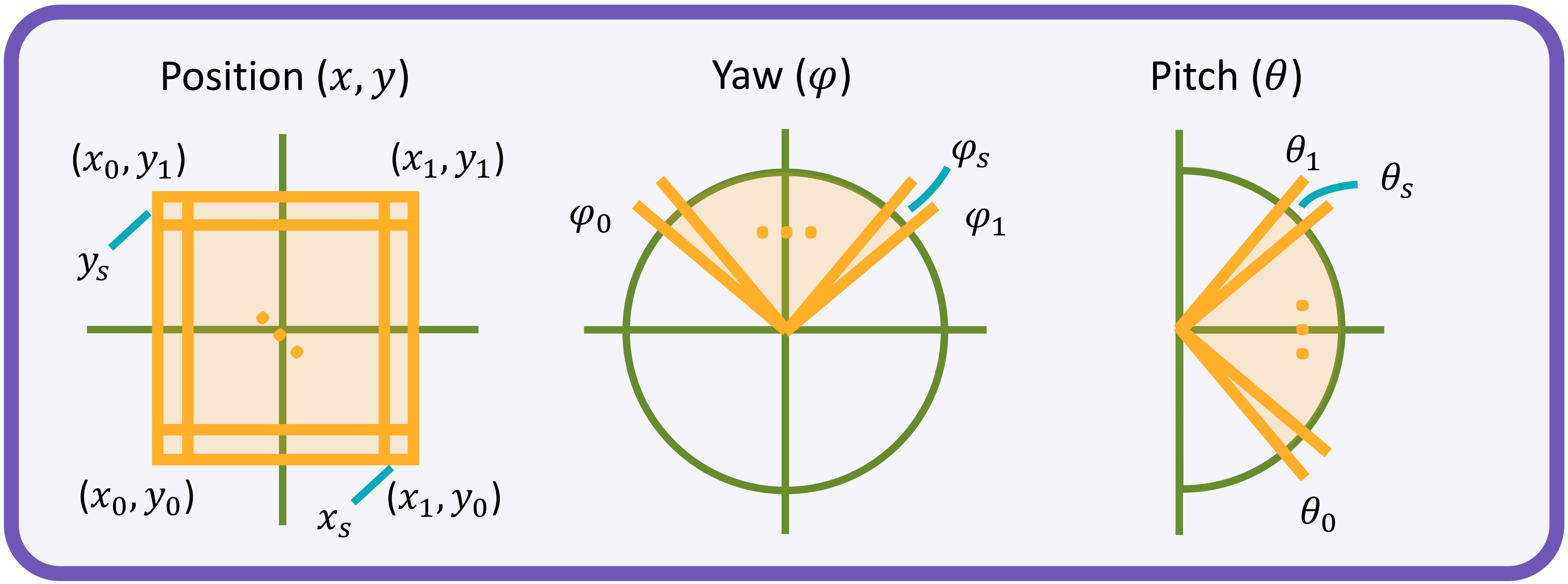
- Pixel Importance
  - Number of views covering a pixel, the smaller the more important
- Select the view that covers the most important pixel
  - **Multiple pixels with the same importance**: select the one that is farthest from the previous selected pixel
  - **Multiple views cover a pixel**: choose the one with max coverage



Offline View Selection

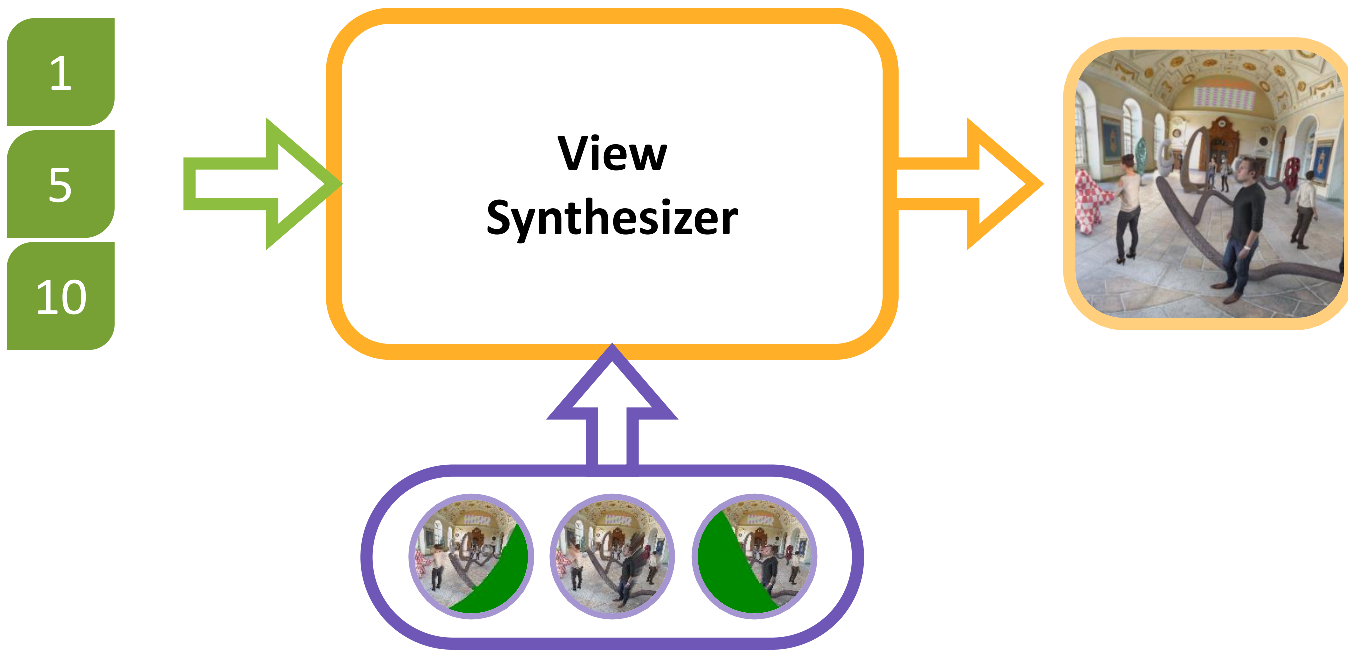
- Cache desired view sets for different viewpoints
- Use the cached set when the user's viewpoint is similar to the viewpoint





View Synthesizer

- Synthesize a virtual view based on the provided view set
- Use MPEG Reference View Synthesizer (RVS) [11]
  - Multi reference views for multi virtual view synthesis
  - Implemented in OpenGL pipeline



B. Kroon. Reference View Synthesizer (RVS) manual. International Organization for Standardization Meeting Document ISO/IEC JTC1/SC29/WG11 MPEG/N18068, 2018. Meeting held at Macau SAR CN.

Implementation

Hole-aware View Selector

View Synthesizer

Viewport Player

Experiments

Performance under different environments

- Compare the results with different algorithms
- Measure synthesis quality and processing latency

System
Performance

Offline vs Online view selection

Test the effectiveness of the cached view set

- Test a viewpoint with the cached view set and the on-the-fly selected view set
- Analyze the result with different solutions

Performance Evaluations

All-View

- The best possible synthesis result for a viewpoint
- Synthesized using all reference views

Geometry-Based

- Dziembowski et al. [12]
- Based on the geometry relationship, like position, rotation
- Baseline in our experiments

Proposed

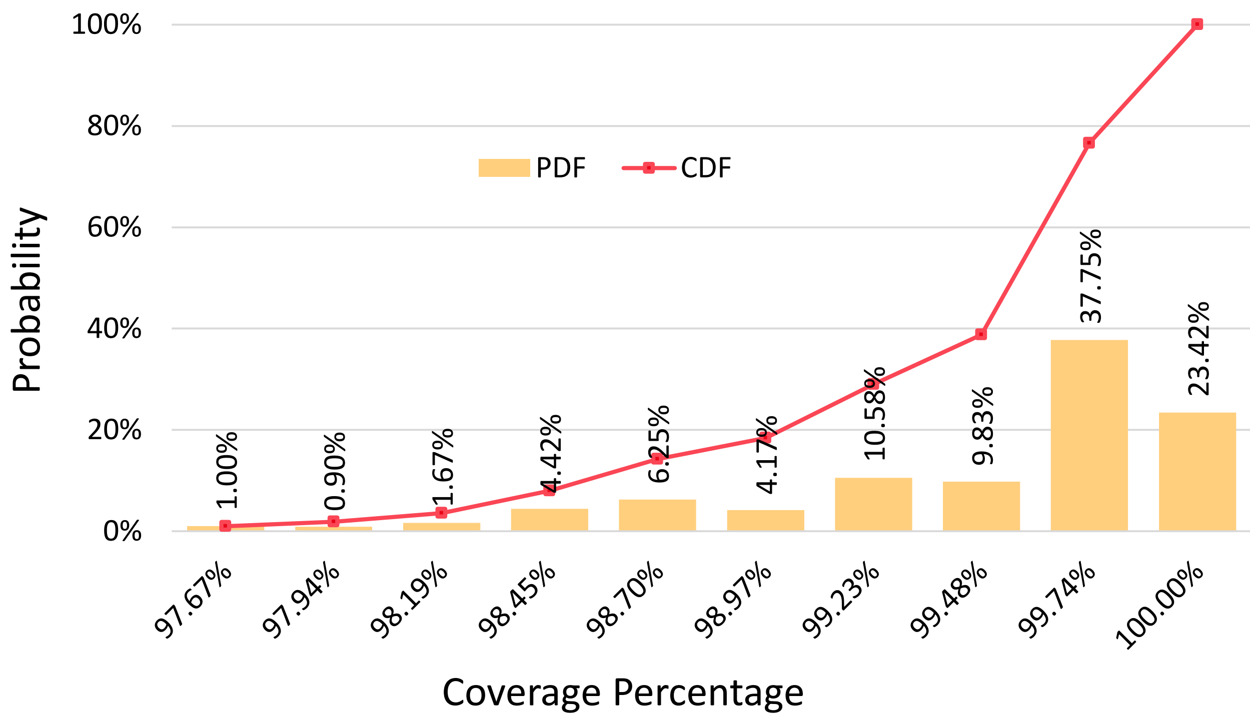
- **Hole-aware** view selection
- **Pixel-importance**-based view selection

Metrics

1. Latency
2. Coverage percentage
3. PSNR
4. SSIM

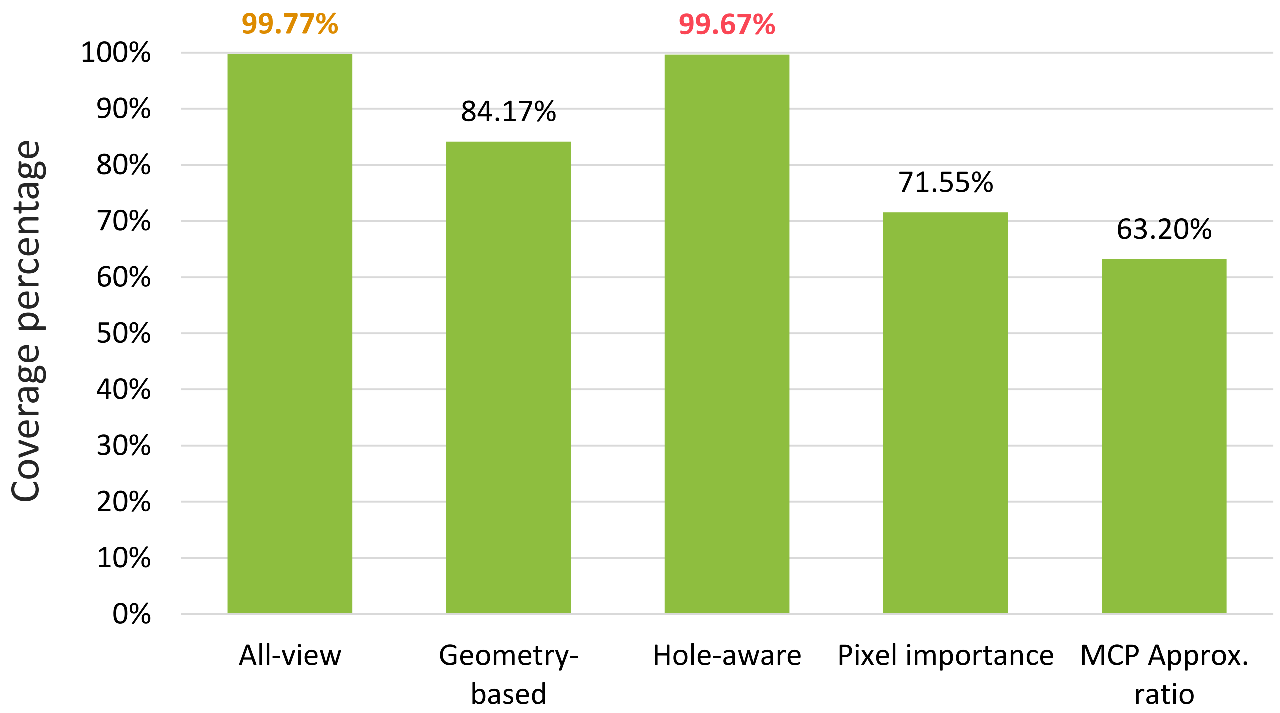
A. Dziembowski, J. Samelak, and M. Doma´nski. View selection for virtual view synthesis in free navigation systems. In 2018 International Conference on Signals and Electronic Systems (ICSES), pages 83–87, Sept. 2018.

Coverage Percentage



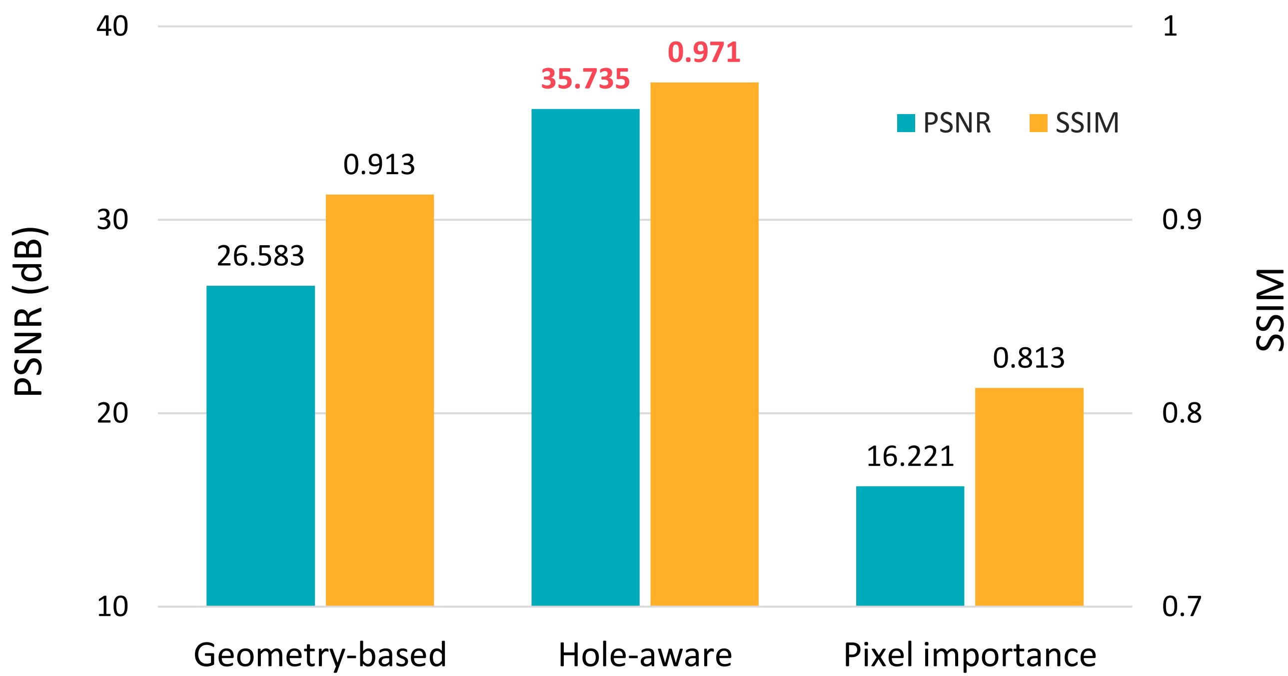
- The trace has **1200** frames
- Over 80% of view synthesis results reach **99%** of coverage percentage
- The worst case still has over 97% of coverage

Algorithm Comparison



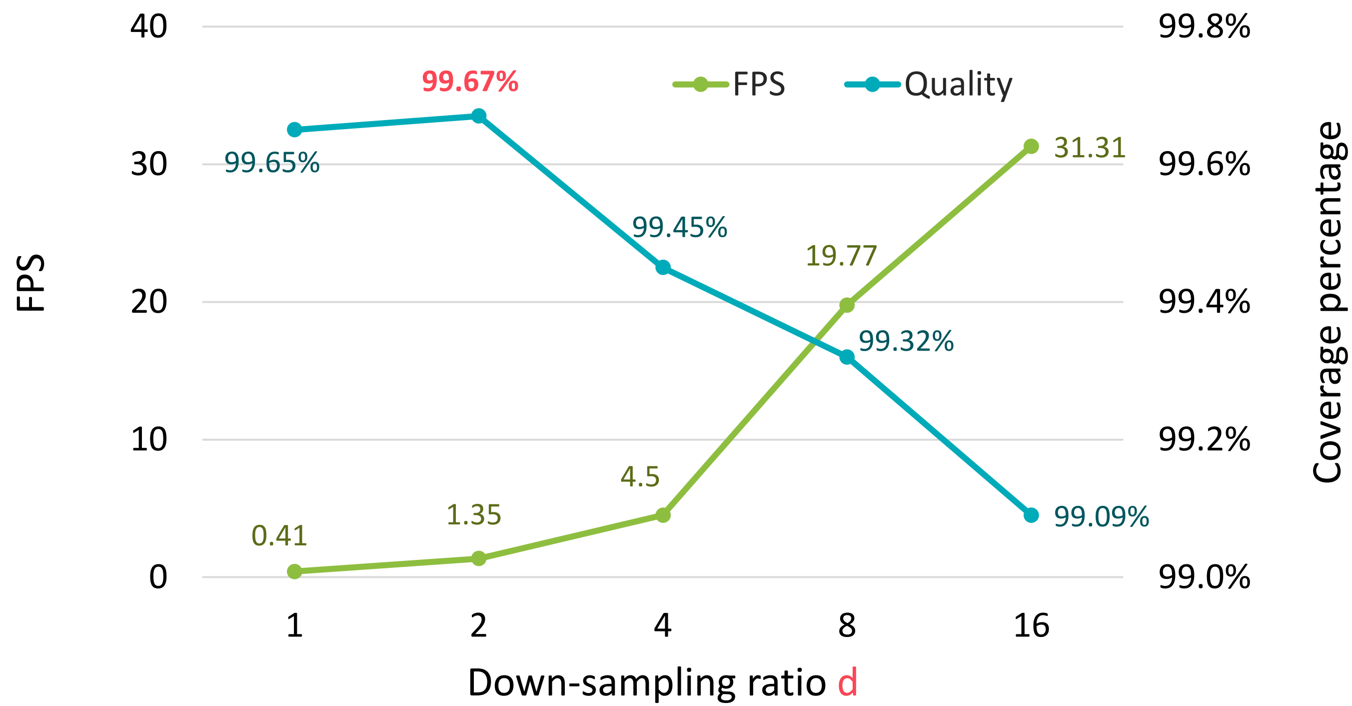
- The coverage percentage using hole-aware algorithm is only **0.1%** lower than the ideal one
- Pixel-importance algorithm is poorly performed
  - The views covering the important pixels **don't necessarily have high coverage**

PSNR & SSIM



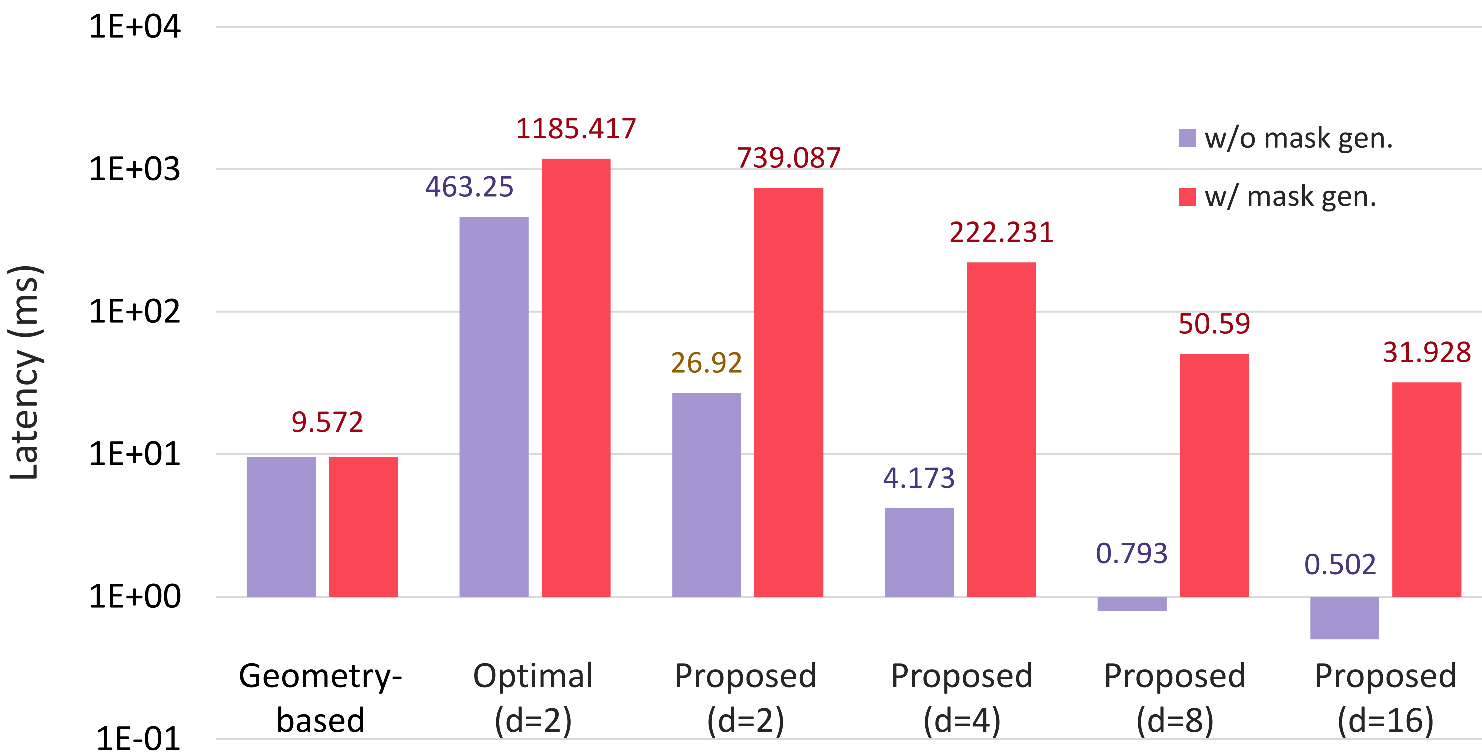
- The results are obtained by comparing to the all-view result
- Our algorithm still holds the highest score

Down-sampling Performance



- Increase d leads to **lower processing latency as well as synthesis quality**
- Real-time process is realized when **d is above 8**

Average Process Latency



- Most of the latency belongs to the mask generation process
- Optimal solution takes about **18** times of processing latency than the proposed solution

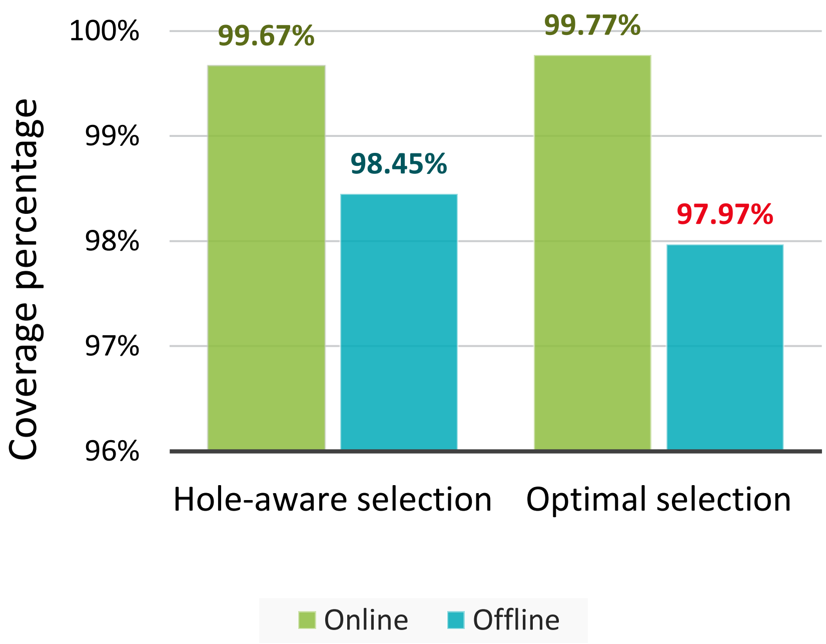
Offline vs Online View Selection

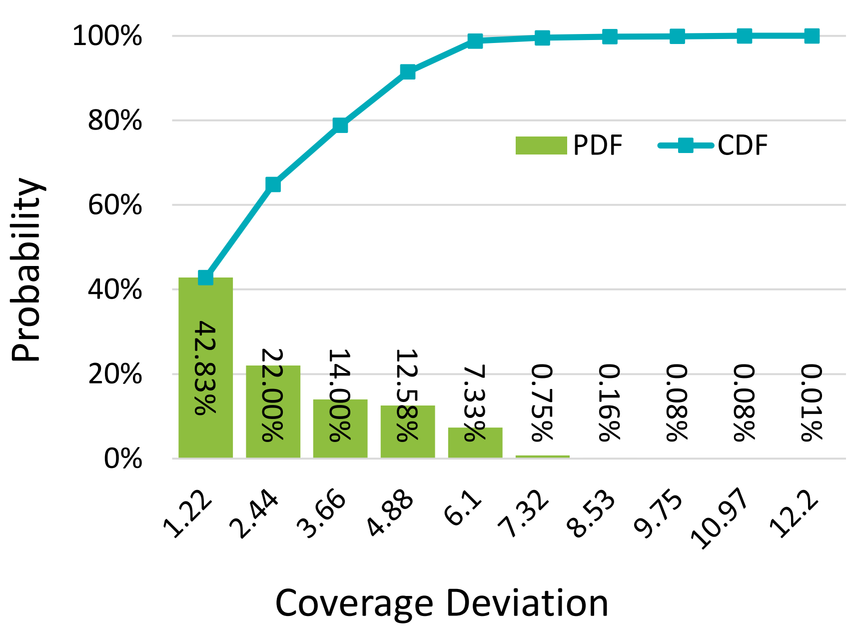
- Use the real user trace (1200 frames) for viewpoint evaluations
- Test with two solutions: **optimal** and **hole-aware**

	Start	End	Step
x	-0.3	0.3	0.1
y	-0.3	0.3	0.1
θ	-50	50	10
φ	-50	50	10

Total 5929 cached viewpoints

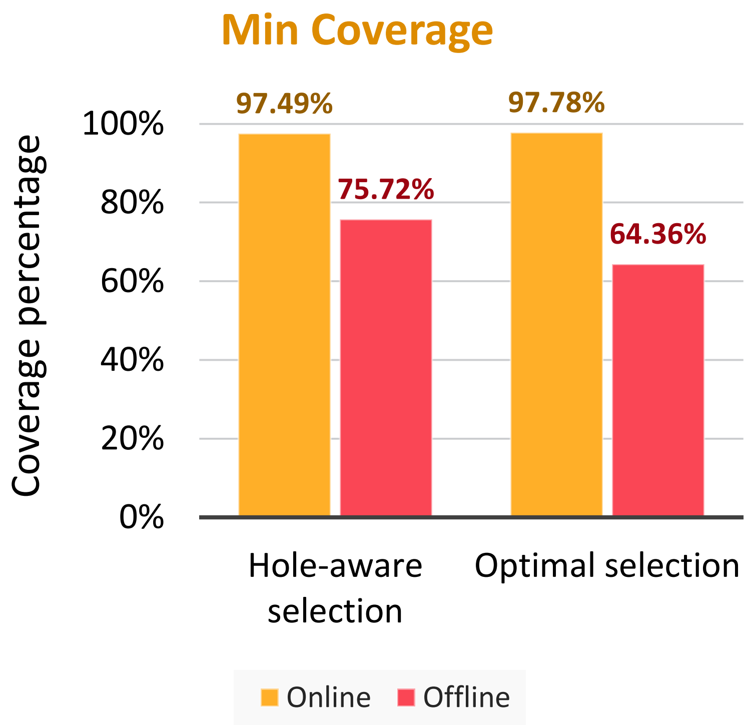
Quality Comparison

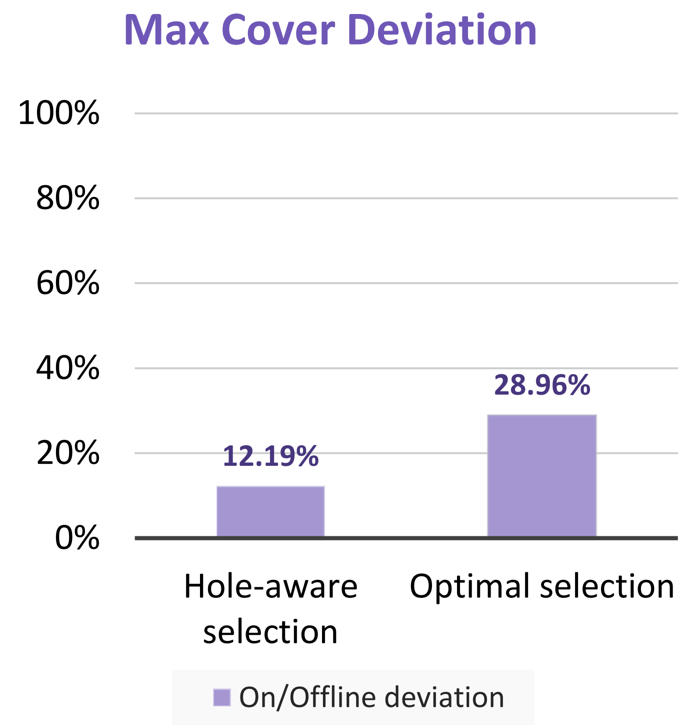




- Online selection is **always better** than offline view selection
  - Viewpoint deviation
- Offline selection of optimal selection leads to **lower** result coverage than that of hole-aware selection
  - Viewpoint specific selection

Viewpoint Specific Selection





- The optimal solution gets the specific set for a certain viewpoint, which leads to **less suitability** to other similar viewpoints
- The proposed solution has better suitability for similar viewpoints

Conclusions

1. We build and evaluate two advanced HMD VR systems with different properties of LF

2. The proposed optimization methods for the refocusing process latency by up to 200 times in average

3. The proposed view selection algorithm can reach view coverage by only 0.1% lower than the optimal solution while being 18 times faster in average

4. The systems help in researches in future VR development

Future Works

Integrate the two proposed systems for further improvement of user experiences

Utilize GPU devices for increasing the system performance

Expand the scale of LF to fully support the 6DoF VR experience

Capitalizing Light-Field Technology in Head-Mounted Virtual Reality

Any Questions?

yuming.lai.8332@gmail.com

Thanks for listening

References

1. J. Moss, J. Scisco, and E. Muth. Simulator sickness during head mounted display (HMD) of real world video captured scenes. Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 52(19):1631–1634, September 2008.

2. M. Levoy and P. Hanrahan. Light field rendering. In Proc. of ACM International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'96), pages 31–42, New Orleans, USA, August 1996.

3. G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu. Light field image processing: An overview. IEEE Journal of Selected Topics in Signal Processing, 11(7):926–954, Oct 2017.

4. X. Wang, L. Chen, S. Zhao, and S. Lei. From OMAF for 3DoF VR to MPEG-I Media Format for 3DoF+, Windowed 6DoF and 6DoF VR. ISO/IEC JTC1/SC29/WG11 MPEG2017/M41197, 2017. Meeting held at Torino, Italy.

5. Y. Lai and C. Hsu. Refocusing supports of panorama light-field images in head-mounted virtual reality. In Proceedings of the 3rd International Workshop on Multimedia Alternate Realities, AltMM'18, pages 15–20. ACM, 2018.

6. R. Ng, M. Levoy, M. Br´edif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Stanford Tech Report CTSR 2005-02, 2005.

References

7 S. E. Chen and L. Williams. View interpolation for image synthesis. In Proceedings of the 20th annual conference on Computer graphics and interactive techniques (SIGGRAPH ’93), pages 279–288. ACM Press, 1993.

8 S. Avidan and A. Shashua. Novel view synthesis in tensor space. In Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 1034–1040, June 1997.

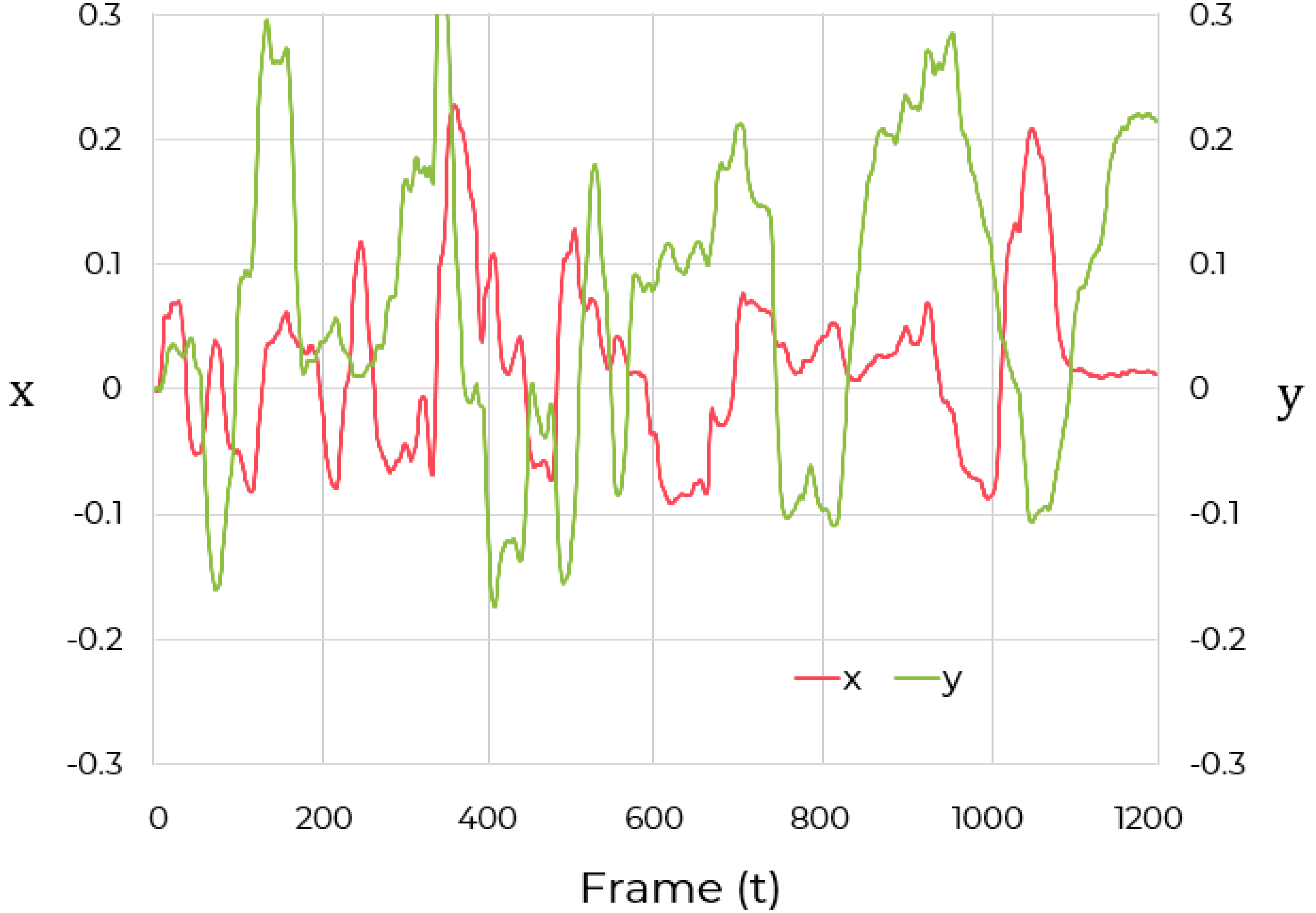
9 V. Vazirani. Approximation algorithms. Springer, Berlin New York, 2001.

10 A. A. Ageev and M. I. Sviridenko. Approximation algorithms for maximum coverage and max cut with given sizes of parts. In Integer Programming and Combinatorial Optimization, pages 17–30. Springer Berlin Heidelberg, 1999.

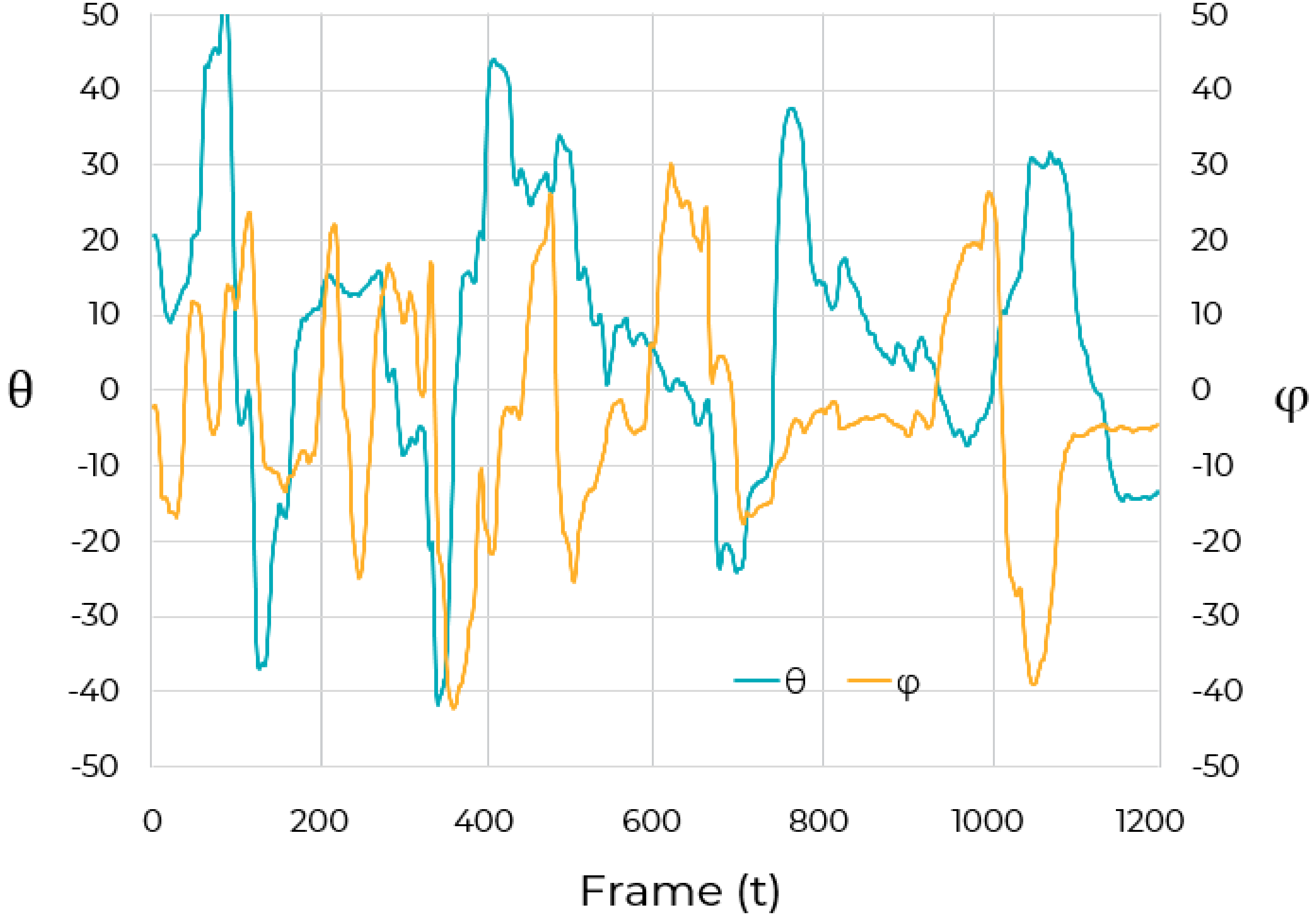
11 B. Kroon. Reference View Synthesizer (RVS) manual. International Organization for Standardization Meeting Document ISO/IEC JTC1/SC29/WG11 MPEG/N18068, 2018. Meeting held at Macau SAR CN.

12 A. Dziembowski, J. Samelak, and M. Doma´nski. View selection for virtual view synthesis in free navigation systems. In 2018 International Conference on Signals and Electronic Systems (ICSES), pages 83–87, Sept. 2018.

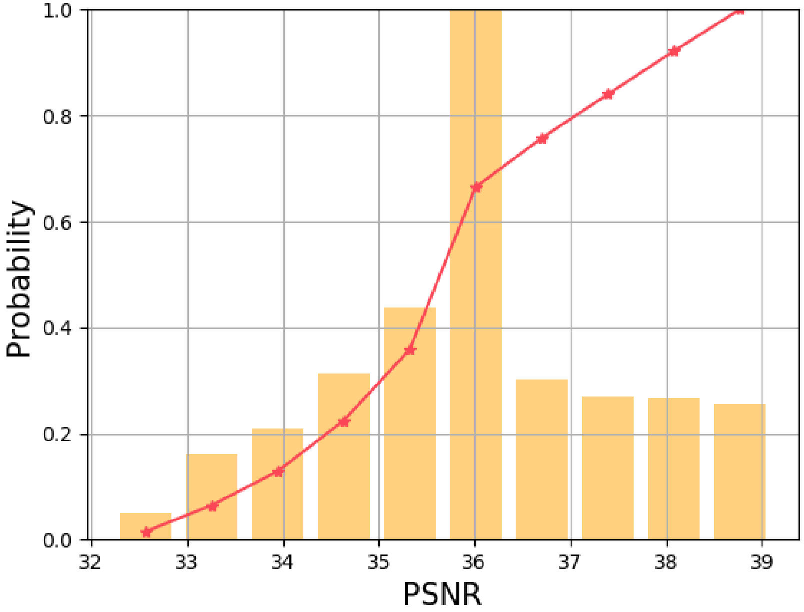
User Trace Trajectory (position)



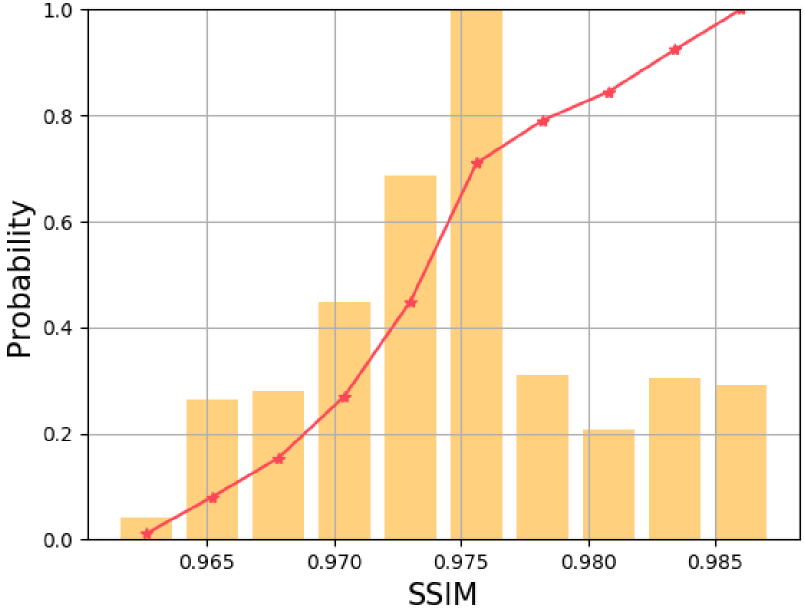
User Trace Trajectory (rotation)



PSNR & SSIM



Probability
1.0
0.8
0.6
0.4
0.2
0.0
32
33
34
35
36
37
38
39
PSNR



Probability
1.0
0.8
0.6
0.4
0.2
0.0
0.965
0.970
0.975
0.980
0.985
SSIM

Performance Supplements

		Optimal	Geometry-based	Hole-aware	Pixel-importance	Approx. ratio
Quality	PSNR (dB)	n/a	24.583	35.735	16.221	n/a
	SSIM	n/a	0.913	0.971	0.813	n/a
	Coverage (%)	99.77	84.165	99.67	71.55	63.2

		Geometry-based	Optimal (d=2)	Hole-aware			
				d=2	d=4	d=8	d=16
Latency	Mask generation (ms)	0	722.167	722.167	218.058	59.797	31.426
	View selection (ms)	9.572	463.25	26.92	4.173	0.793	0.502
	Total (ms)	9.572	1185.417	739.087	222.231	50.59	31.928