High-resolution 360° Video Foveated Stitching for Real-time VR

Wei-Tse Lee* Hsin-I Chen* Ming-Shiuan Chen
I-Chao Shen Bing-Yu Chen

National Taiwan University

PACIFIC GRAPHICS

gaming

medication

Cinema and concert

360 panorama content capturing

Challenge

- Producing 360° video from multiple cameras is still challenging.
- Performance of commercial software **VideoStitch[1]** and **Kolor[2]**
  - **Far from real-time :** 0.07 seconds per frame to generate 1K panoramic video from 6 separated cameras of 2k resolution.
  - Can not be used in real-time scenario
  - Low resolution quality makes users feel unreal in the virtual world.
- Transmitting huge data size with unstable latencies brings viewing quality degradation.

[1] VideoStitch, https://www.orah.co/software/videostitch-studio/
[2] Kolor, https://www.kolor.com

Goal

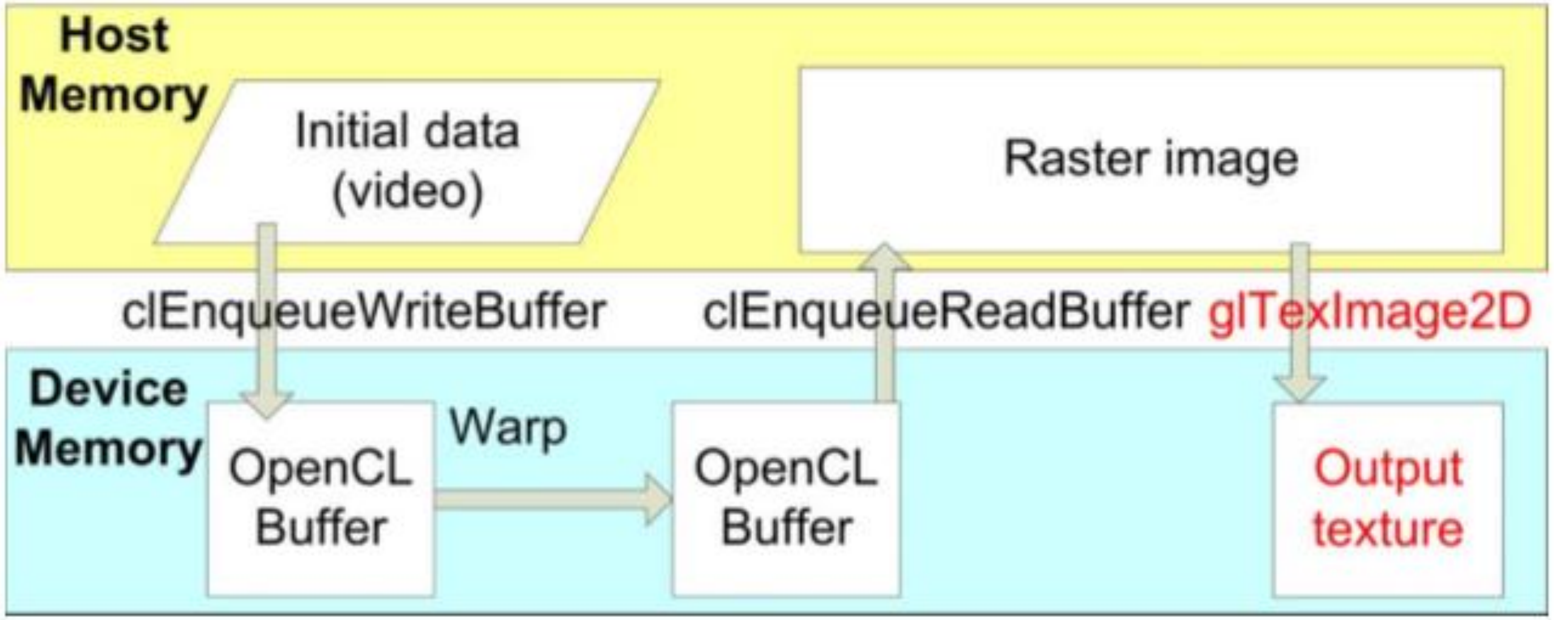
- **capture and generate 4K 360° panorama video in real-time**
  - Reduced computational complexities -> increase the performance
  - Introduces least amounts of perceptual artifacts.

High Quality Panoramic Video

Panoramic Video from Unstructured Camera Arrays, EG 15

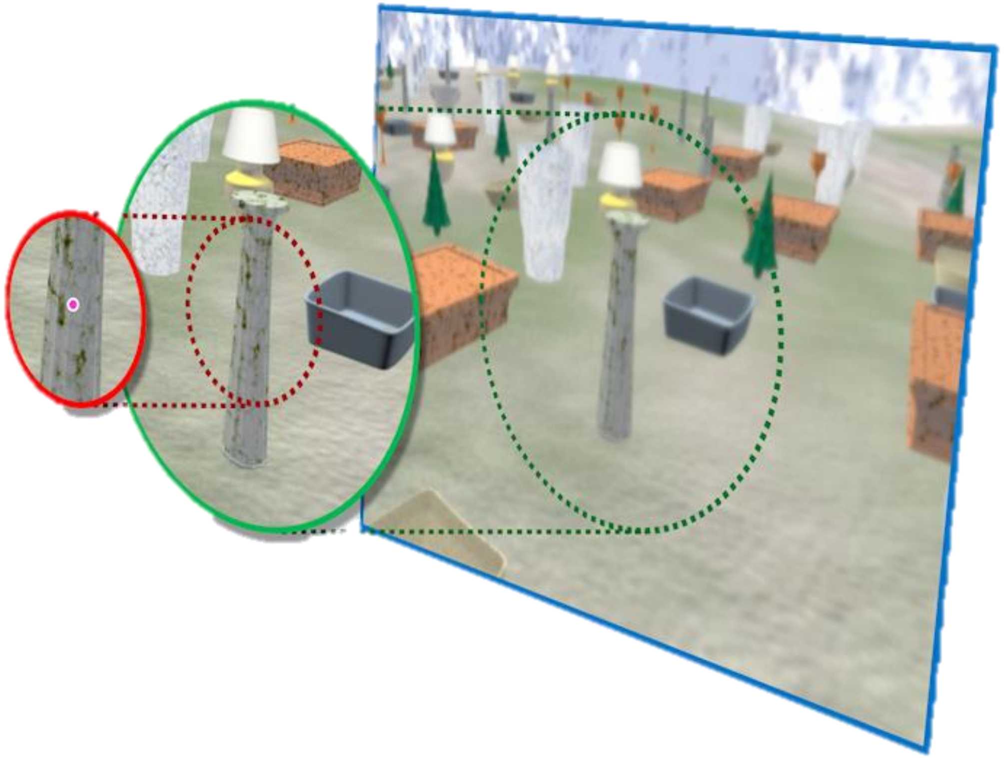
A 360-degree panoramic video system design, VLSI-DAT 14

Fast-stitching Panoramic Video



Stitching Videos Streamed by Mobile Phones in Real-time, MM 09
An effective video stitching method, ICCDA 10
GPU parallel computing of spherical panorama video stitching, ICPADS 12

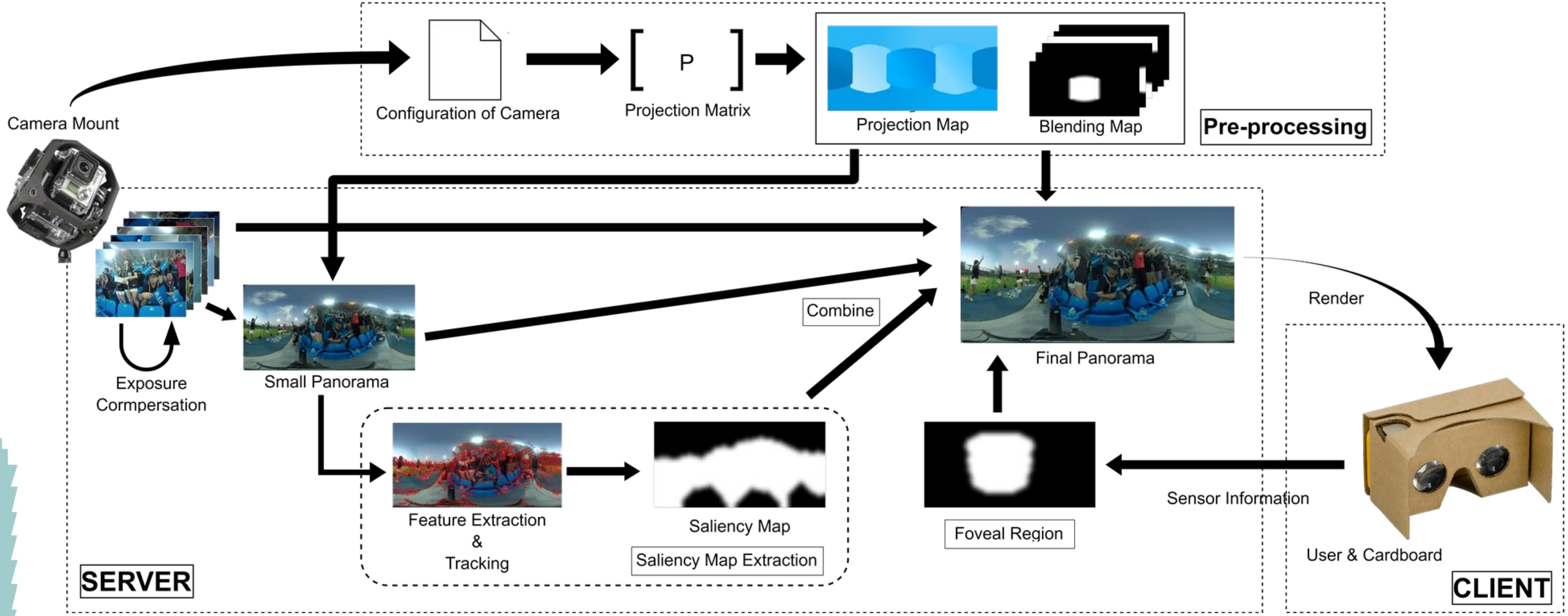
Perceptually Lossless Rendering



Foveated 3d graphics
SIGGRAPH Asia 12

Towards Perceptually Lossless Rendering: Latency Aware Foveated Rendering in Unreal Engine 4
CVMP 15

System Overview



Stitching formula

- We calculate the output panorama F_t at time t

$$F_t(x,y) = \sum_{i \in N} M_i(x,y) * E_{i,t}(x,y) * B_i(x,y)$$

- Projection Map

$$M_i(x,y) = I_i(x', y'), \text{where } i \in N$$

Stitching formula

- **We calculate the output panorama F_t at time t**

$$F_t(x,y) = \sum_{i \in N} \underline{M_i(x,y)} * E_{i,t}(x,y) * \underline{B_i(x,y)}$$

- **Blending map**
  - take L1 distance to the image center as the weight

Stitching formula

- **We calculate the output panorama F_t at time t**

$$F_t(x,y) = \sum_{i \in N} \underline{M_i(x,y)} * \underline{E_{i,t}(x,y)} * \underline{B_i(x,y)}$$

- **Blending map**
  - take L1 distance to the image center as the weight
- **Exposure compensation**
  - Use the method from [Brown and Lowe 2007]

Stitching formula

- We calculate the output panorama F_t at time t

$$F_t(x,y) = \sum_{i \in N} M_i(x,y) * E_{i,t}(x,y) * B_i(x,y)$$

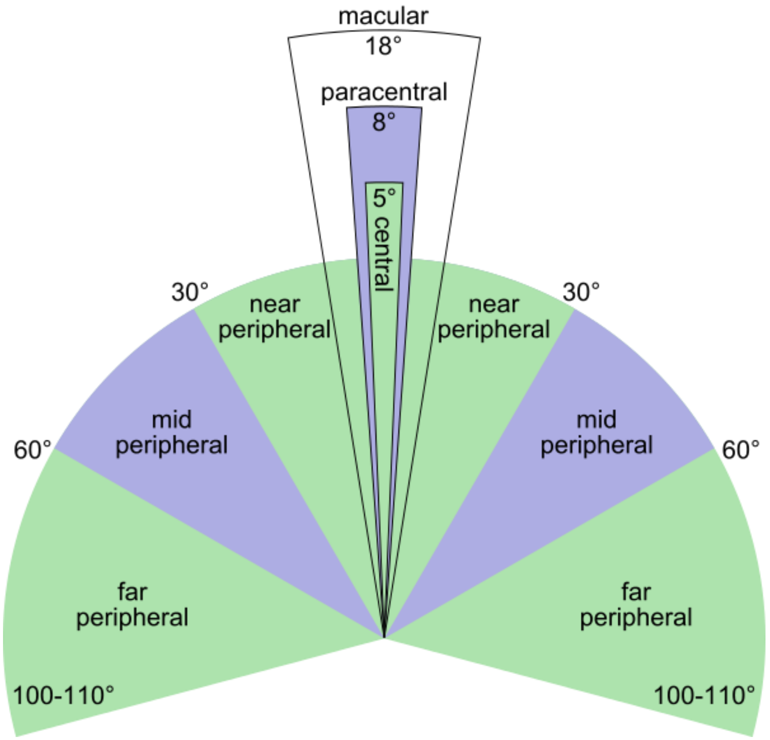
- Blending map
  - take L1 distance to the image center as the weight
- Exposure compensation
  - Use the method from [Brown and Lowe 2017]

Gaze-contingent framework

- **With the increasing use of 4K-8K UHD displays and the push towards higher pixel densities for head-mounted displays**
  - The content rendering is too computational heavy.
- **Exploiting properties of the human visual system (HVS)**
  - Equipped with eye tracking device or device to approximate it.
  - Foveated rendering technique [Guenter et al. 2012]

Foveated rendering in a nutshell

- Technique to reduce the rendering workload by greatly reducing the image quality in the peripheral vision .



Foveated rendering

- **Challenge**
  - Sensitive to system latency [1].
  - When gaze location changes rapidly, even short delays may result in visible artifacts which make the gaze-contingent rendering unfavorable.
- **To compensate**
  - Increase rendered foveal region diameter.
  - The true foveal field-of-view is always contained within the rendered foveal region.

[1] Direct measurement of the system latency of gaze-contingent displays, DR Saunders and RL Woods, Behavior Research Methods 46, 2 (2014)

Latency-aware foveal region diameter

- We use the formula to measure the size of foveal diameter [1]

$$F_\phi = 2\rho_{pixel} d_u tan(L_{tot} S_{max} + \frac{\alpha}{2}) + 2b_w + c$$

L_{tot} : average tracking latency in milliseconds
S_{max}: estimated maximum saccadic speed
ρ_{pixel}: pixel density of the screen
d_u: distance between user and the screen
α: the angle subtended by the fovea which is around 5-degree

[1] SWAFFORD, N. T., COSKER, D., AND MITCHELL, K. 2015. Latency aware foveated rendering in unreal engine 4. In Proceedings of the 12th European Conference on Visual Media Production, 17:1–17:1.

Latency estimation

(1) Network : Client -> Server
(2) Stitching & Blending
(3) Network : Server -> Client
(4) Render : Cardboard app
(5) Screen : Scan out

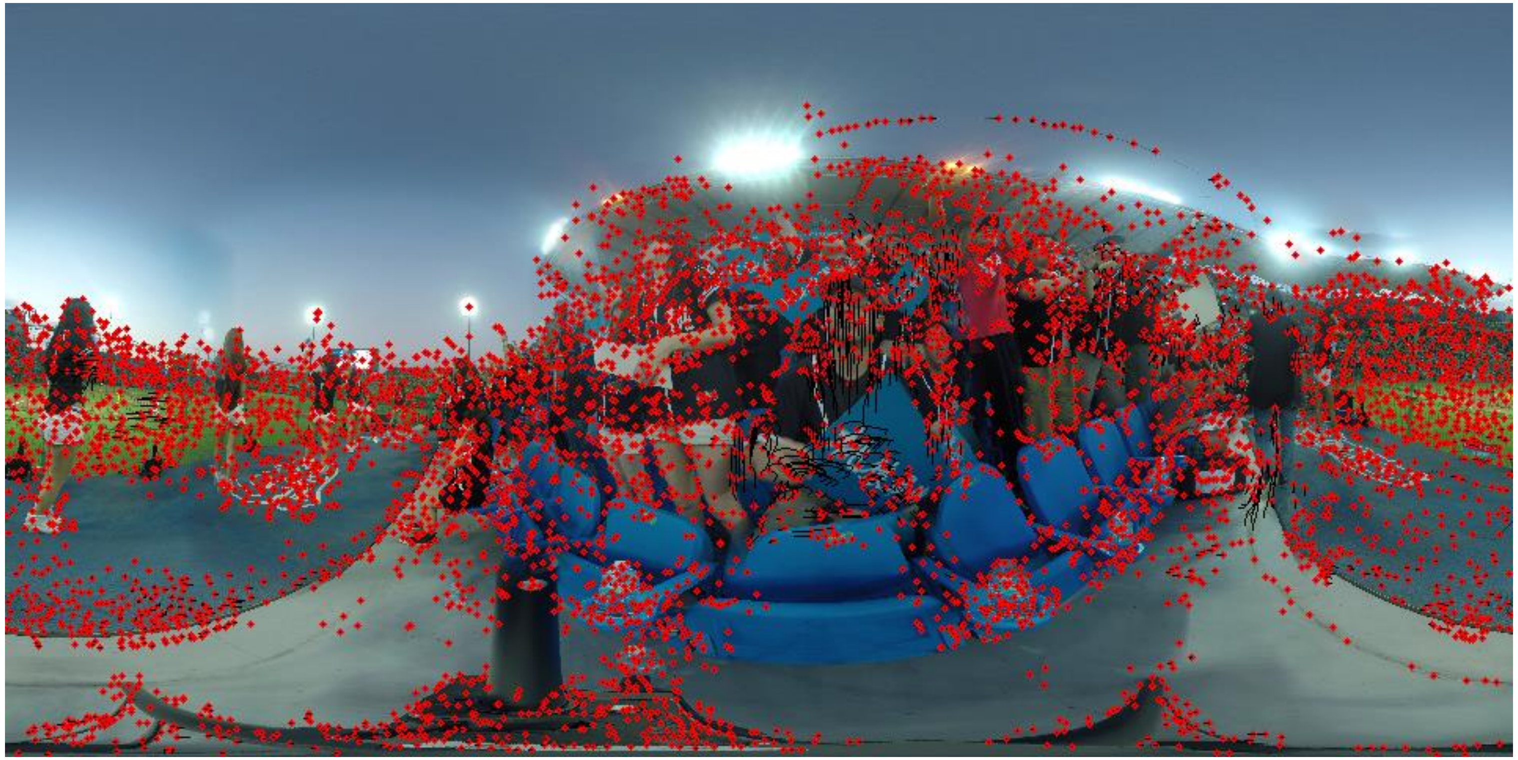
Our system: **61.8** ms on average

Saliency-aware Level of Detail

Thresholding formula:

$$S_t(g) = \begin{cases} 1, & \text{if } \text{THRESH}(f_t(g)/S_g, \epsilon_f) \\ 0, & \text{otherwise} \end{cases}$$

Low resolution panorama



Feature Extraction & Tracking

Saliency-aware Level of Detail

From p
(Fov

Low Resolution Panorama

Low Resolution

Gaussian Blurring

Triangle Mesh

High Resolution

Experiment and User study

- **Evaluation of perceptually loss:**
- User study (13 males, 8 females, 10 videos)

- **Evaluation of system performance:**
- **Simulation** from sensor data collected in user study

Hardware

- Video data were captured using 6 **GoPro Hero4** cameras (*2704x1520*)
- **Server side**:
  - quad-core Intel i7-3770 CPU @3.40 GHz
  - 24 GB RAM
  - GTX 980 GPU
- **Client side:**
  - Sony XPeria Z

User Study Setting

- Generate video offline
  - Without acuity map estimation, only saliency map.
  - We collect gaze data at the same time.
- Display : Google cardboard + Sony XPeria Z
  - **1920 x 960 (highest resolution of android phone)**

User Study Setting

- **We use 2 sequences (seq1 and seq2)**
  - generated 5 configurations for each sequence
- **For each users, ask he (she) to view this 10 cases in random order.**
  - Score quality from 1 – 10.
  - 10 indicates highest quality, 1 indicates lowest quality.
  - recruited 21 users (13 males and 8 females)
- Evaluate the effectiveness of parameter ε_f

User study results

Case	Avg Score for Seq 1	Avg Score for Seq 2
High resolution	**6.68**	**6.11**
Low resolution	**2.68**	**3.05**
Our method (ε_f = 0.03)	**5.95**	**5.90**
Our method (ε_f = 0.04)	**6.26**	**5.95**
Our method (ε_f = 0.06)	**5.05**	**5.58**

we use 1920 x 960 as high resolution, 960 x 480 as low resolution

Performance evaluation

Resolution	Seq1 FPS (CPU / GPU)	Seq2 FPS (CPU / GPU)
2160 x 1080	7.30 / 23.76	7.42 / 23.87
4320 x 2160	3.47 / 20.25	3.48 / 20.32

We compare the system performance under CPU / GPU

Future work

- Better alignment for better projection approximation with parallex removal.
- Implement framework on high-performance VR devices.
- Further acceleration.
- Stable foveal region detection, with additional sensors.

Conclusion

- We propose a *gaze-contingent framework*
  - Foveated stitching technique based on foveated rendering technique saliency-aware level-of-detail.
  - Real-time system based on GPU implementation.

- Such techniques could be used in several VR applications such as live game streaming and view sharing.


ANY QUESTIONS
DO YOU HAVE?
memegenerator.net