Video Streaming for Foveated High-resolution Rendering

Master's thesis, Marc Aurel Kastner
Supervised by M. Stengel, Prof. M. Magnor

Bandwidth limits


Source: Flickr
Grzegorz Rogala

- Panoramic scenes create enormous FOV
- Video data continuously increases
  - Resolution: 2K → 4K → 8K → …?
  - Frame rate: 25 fps → 60 fps → …?
- Unable to process *full* high resolution frames

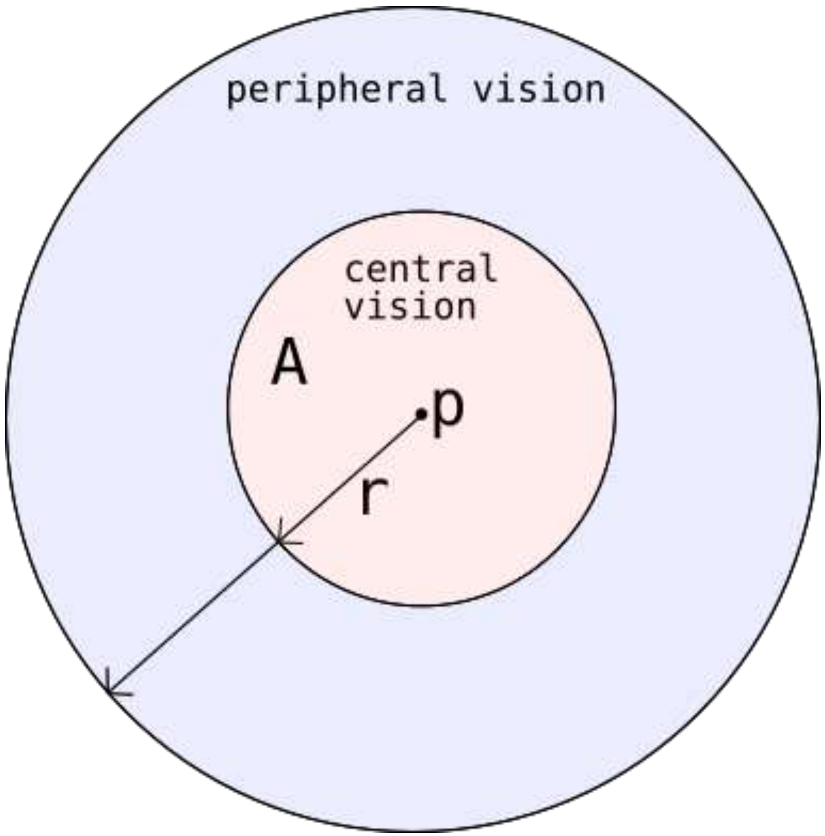
→ Need for high resolution with low bandwidth

Foveated imaging

Source: Wikipedia

Background: Visual acuity

- Simplified model:
  - Aubert / Foerster (1857)
  - Linear fall-off until 20°
  - Then, strong drop
- Still commonly used
  - Conservative
  - Simplicity



Related Work

Foveated MPEG
(Geisler et al. 1996)

- Static / Passive
  - Saliency estimation
  - Molding acuity into data
- e.g. sport & news clips
- Prone to error
  - Users might look in different directions
  - Not well optimized to one users' view

Foveated 3D Graphics
(Gunther et al. 2012)

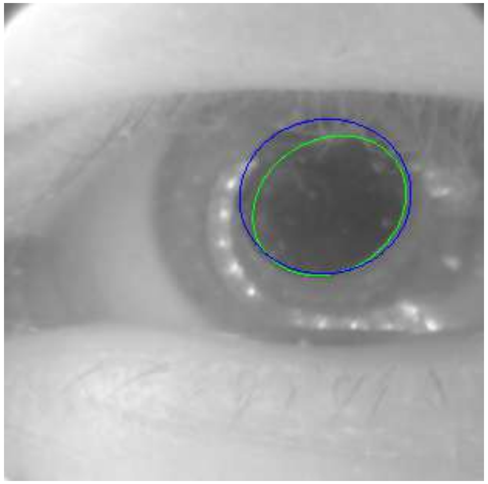
- Dynamic
  - Pupil Tracking
  - No data modifications
- e.g. rasterization, raytracing
- Adapts to users' eye

→ Make this for videos

Pupil tracking in HMDs

- Recent VR often use Head mounted Displays

- HMDs allow pupil tracking
  - Active gaze estimation
  - Knowledge about users' field of view



Idea

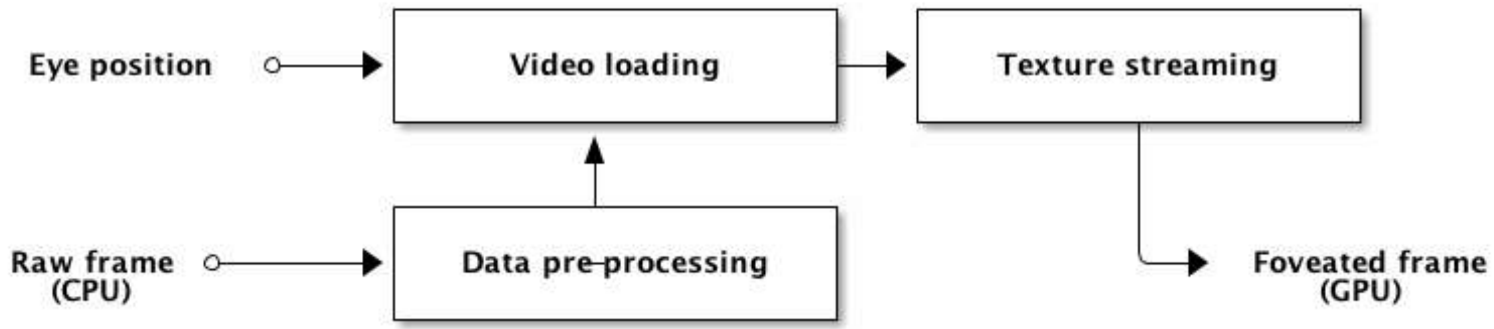
- Decrease bandwidth
  - Use gaze estimation
  - Filter resolutions simulating acuity
  - Optimize streaming
- Restraints
  - Dynamic
  - Performance
  - Perception

Source: Wikipedia

Table of contents

Design

- Three main pillars
  - Pre-processing
  - Video loading
  - Texture streaming

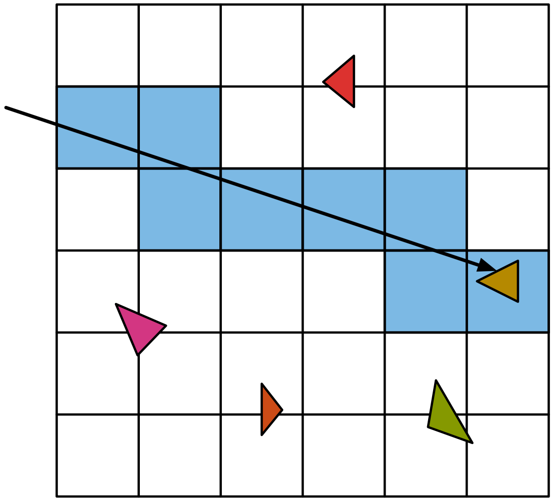


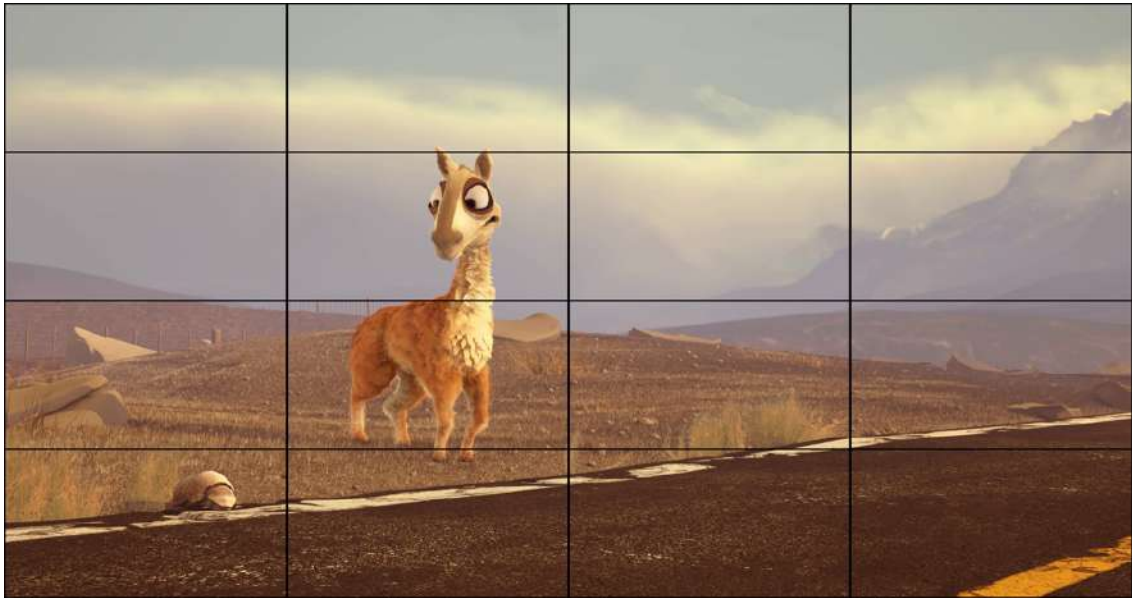
Pre-processing

- Video formats don't allow region or pixel access
- Impractical to decode *full* frames
  - Access smaller videos
  - Have multiple quality versions
- No relation to eye position
  - Dynamic

Background: Uniform grid

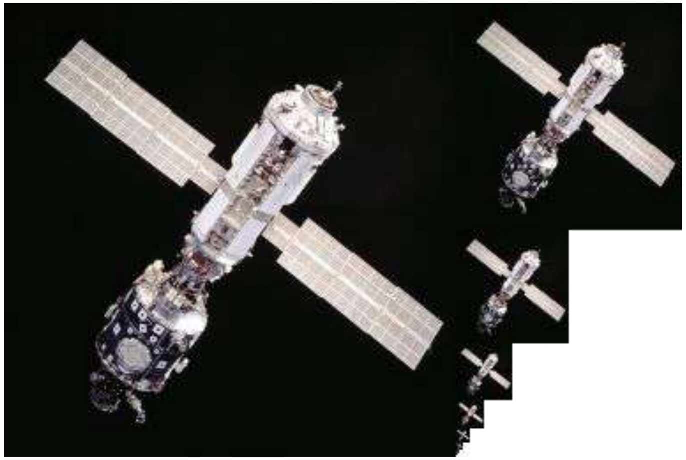
- Uniformly distributed
- Spatial sub-division
- Used for
  - Ray tracing
  - Video tiling





Background: Mip mapping

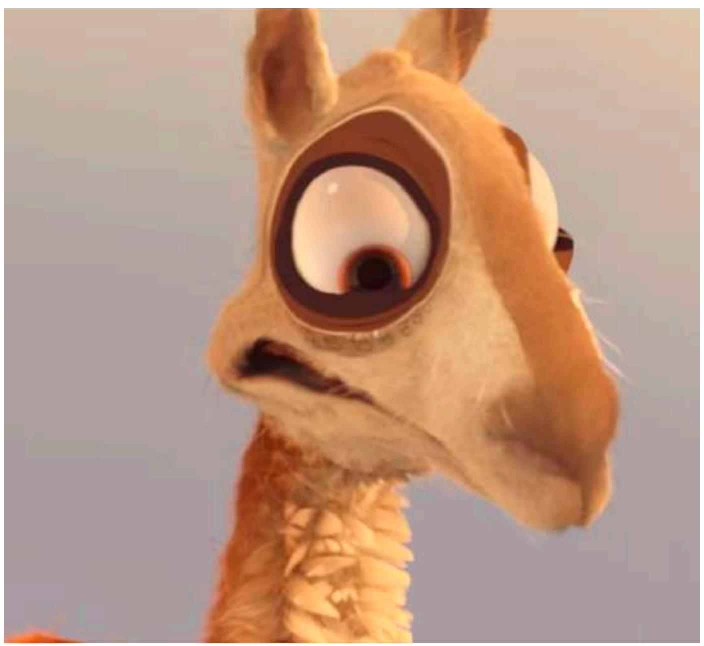
- Idea
  - Pre-process scaled down variants of textures
  - Save different resolution levels
  - Reduce stress on GPU, simplify filtering, avoid moiré patterns, …
- Often used in video games

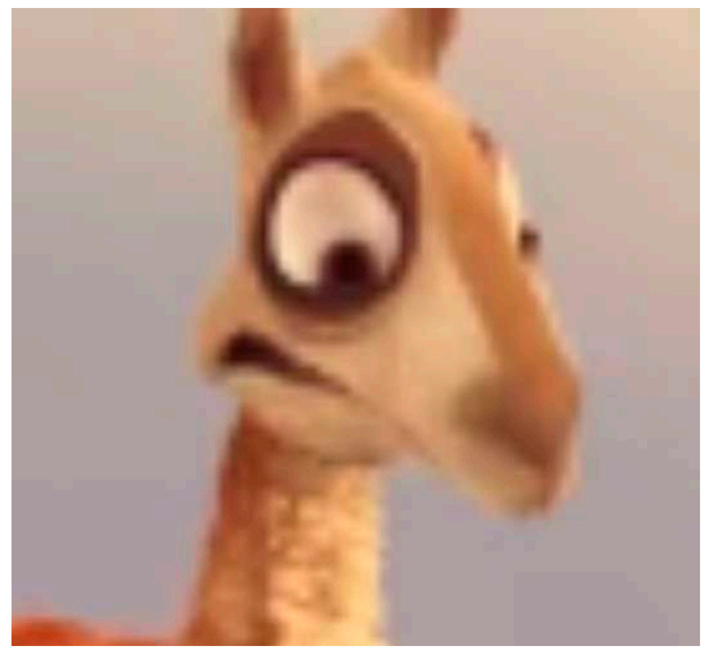


Source: Wikipedia

Pre-processing

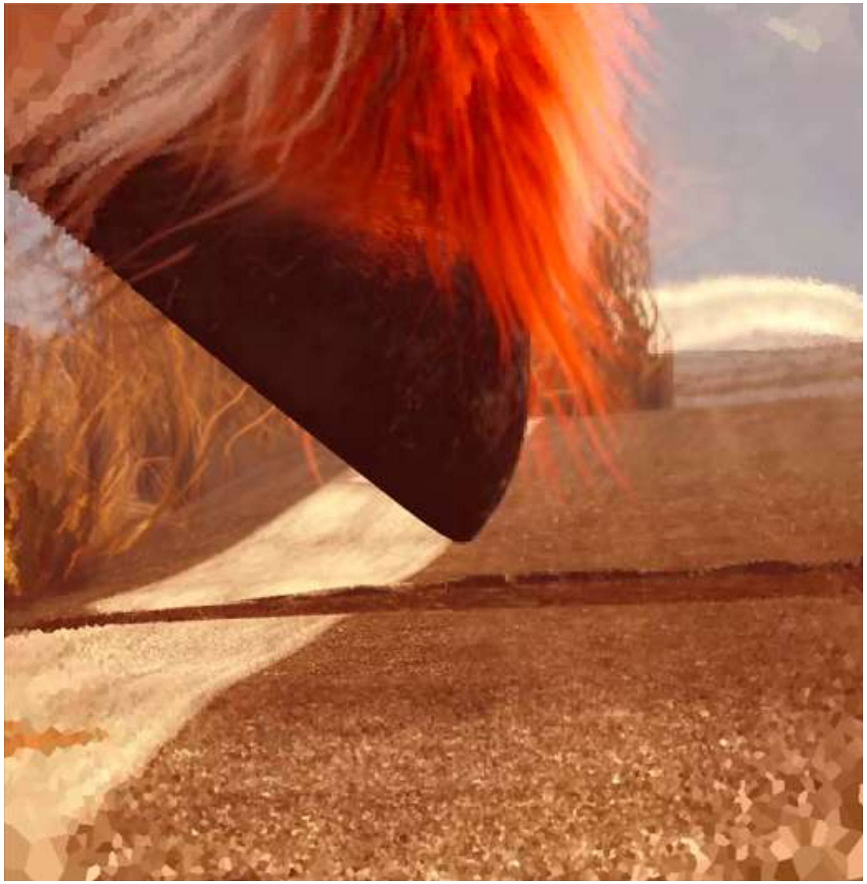
- Idea
  - Use video tiling
  - Create multiple mip-map levels per video tile
- Allow region-based loading





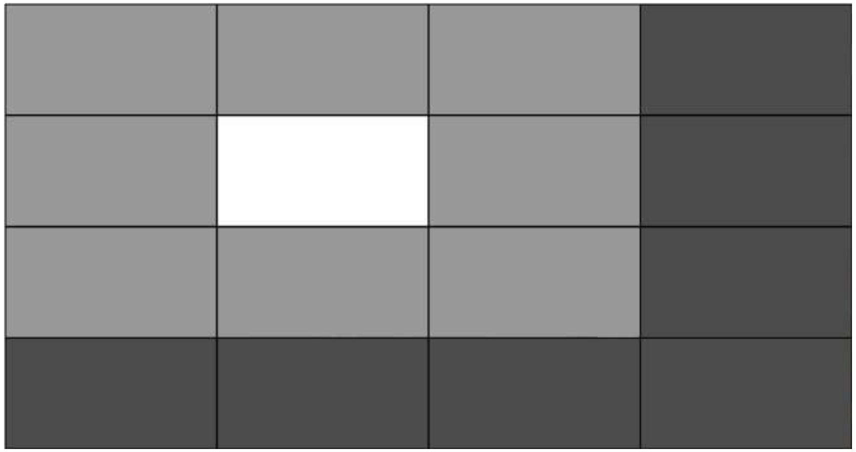
First approach

- Wide-screen videos
  - Arbitrary codec
- Acuity distribution: Hybrid!
  - Grid-based loading
  - Sampling-based streaming



Grid-based video loading

- Depending on eye position
  - Select suitable resolution
  - Threshold mip-map levels with distance
- Grid 4x4, 3x Mip-map
  - 48 video files
  - Load frames as needed



Multi-threaded video loading

- Seeking is major bottleneck
  - Every video file is open in parallel
  - An eye movement triggers wall of seeking

- Solution
  - Multi-threading model
  - Sometimes use *wrong* mip-map levels
    - Didyk et al. (2010): Retina takes 60 ms to adapt
  - Allow system to keep up

Sampling-based acuity

- Two masks
  1. Acuity mask
  2. Interpolation mask

- Allows minimum bandwidth
- Linear reconstruction with look up tables

ID	X	Y
0	15	12
1	46	56
2	89	96
…	…	…

Acuity mask

0	0	1	1
0	2	3	1
4	5	6	7
4	4	7	7

Interpolation mask

Frame reconstruction

ID	X	Y
0	15	12
1	46	56
2	89	96
…	…	…

Streamed to GPU

0	1	2	3	4	5	6	7

every frame

+

0	0	1	1
0	2	3	1
4	5	6	7
4	4	7	7

once

Shading

0	0	1	1
0	2	3	1
4	5	6	7
4	4	7	7

Second approach

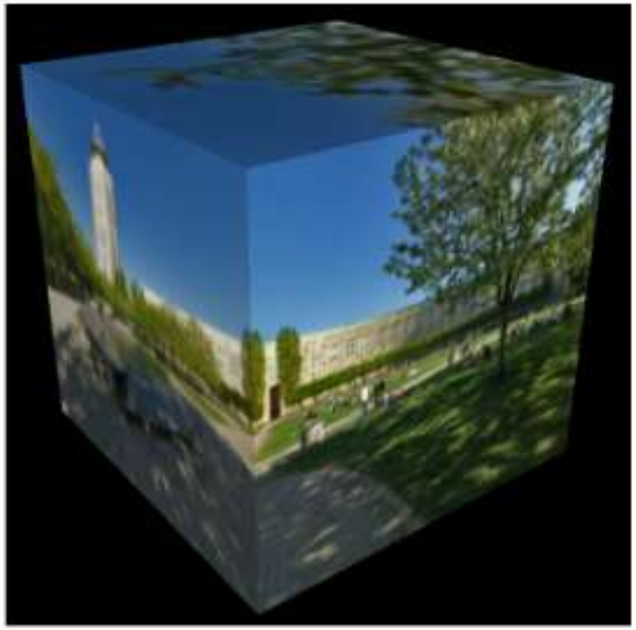
- 360 degree panorama
  - Pictures per frame
  - JPEG/PNG

- Sampling may introduce visible flickering

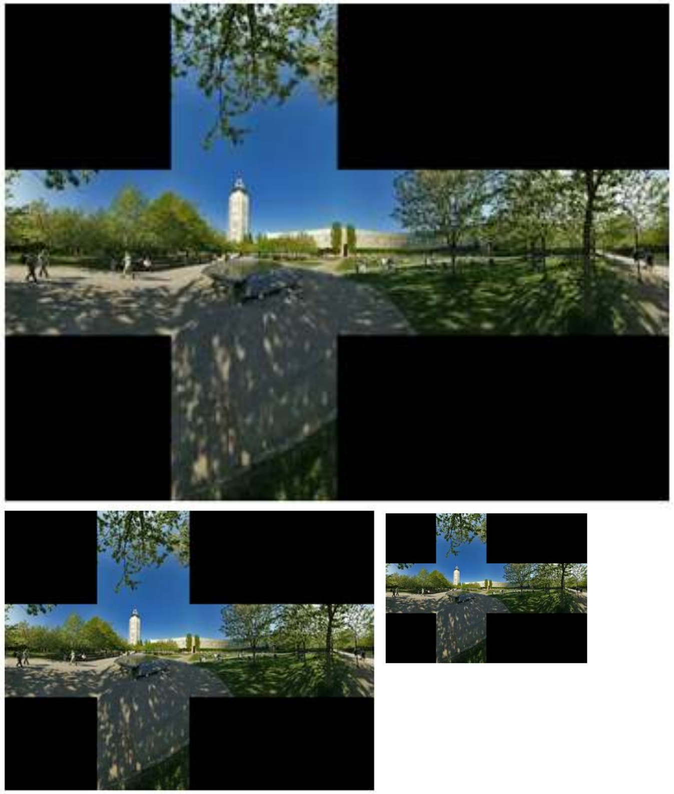
- Acuity distribution
  - Fully grid-based
  - Radial blur in post-process reduces peripheral frequencies

Background: Cube mapping

- Six single textures for 360° surface
- GPU: One cube map per mip-map
  - One Grid per cube side



Source: David J. Eck



Source: Emil Persson

Acuity distribution

- For every frame:
  - Calculate view vector
  - Decide *mip-map*
- Update only relevant region per frame
- Acuity data as look up table

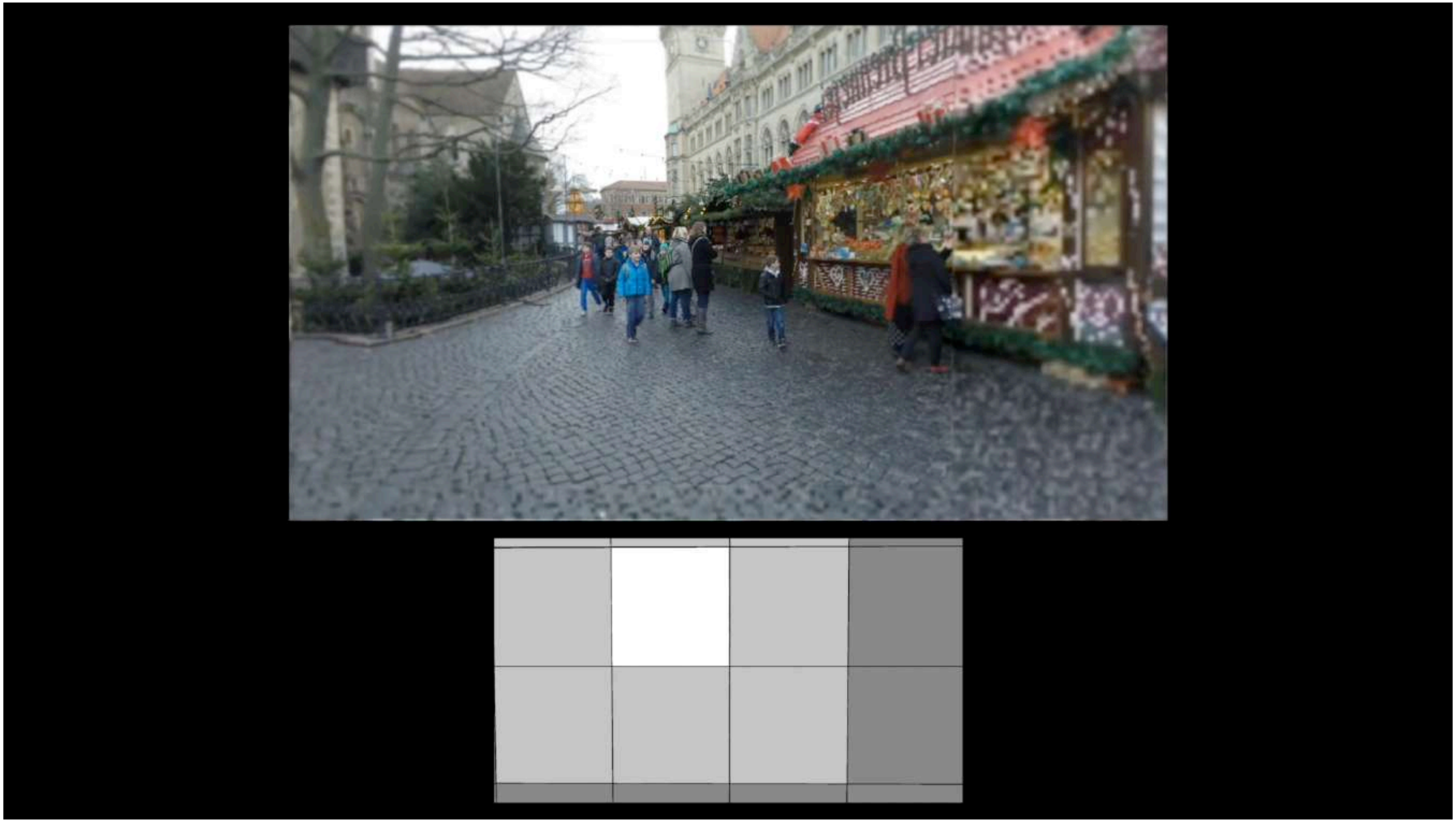
+

+

=

Evaluation: Demo video 1

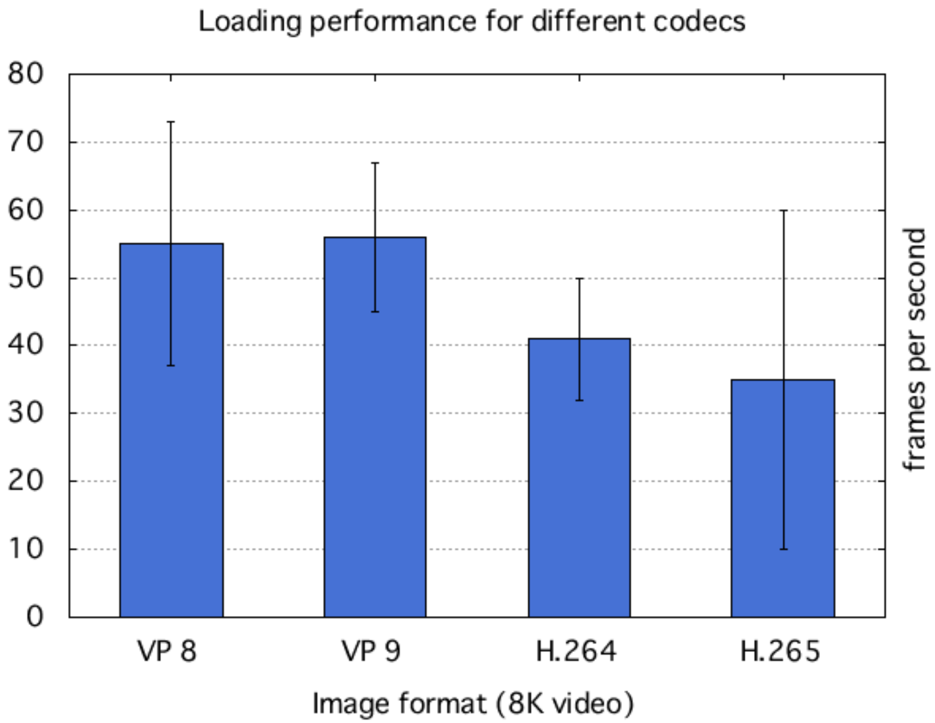
Evaluation: Demo video 2



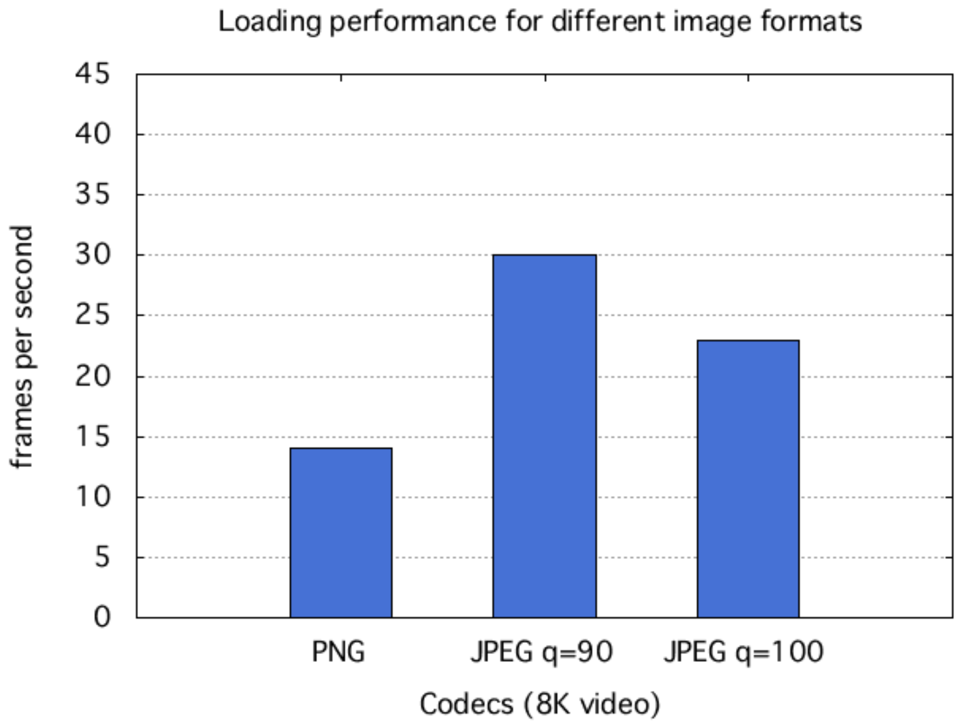
Evaluation Codecs

Technische Universität Braunschweig

Foveated Video Approach 1

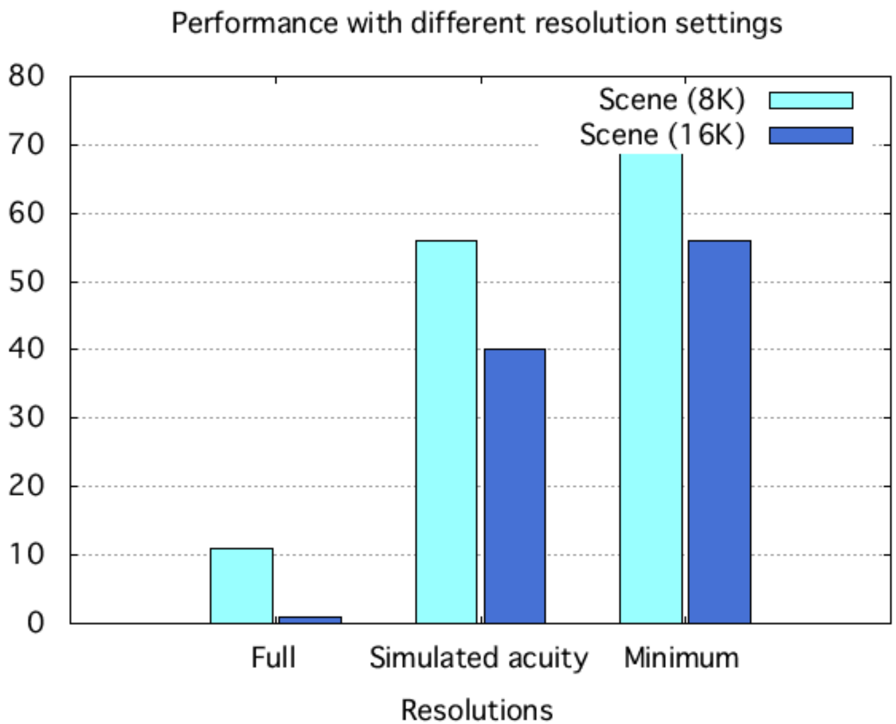


Foveated Video Approach 2

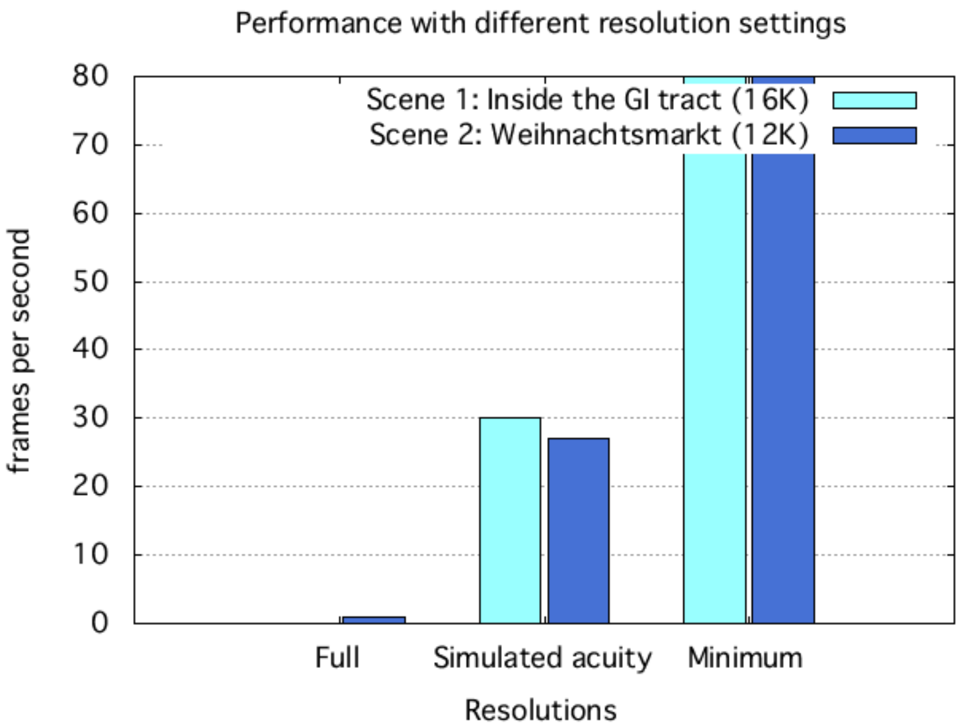


Evaluation Performance

Foveated Video Approach 1



Foveated Video Approach 2



Evaluation Bandwidth

Wide-screen videos	Transferred pixels	Percentage
Sampling-based acuity (1080p acuity mask)	690.926	**2.08%**
Minimum resolution (smallest mip-map 240p)	129.600	**0.39%**
Full resolution (8K wide-screen)	3.177.600	**100.00%**

Panoramic videos	Transferred pixels	Percentage
Grid-based acuity (best case)	1.773.229	**2.89%**
Grid-based acuity (worst case)	2.789.232	**7.74%**
Minimum resolution (smallest mip-map 120p)	212.064	**0.59%**
Full resolution (12K panoramic)	36.000.000	**100.00%**

Conclusion

- Exploit human visual system
- Replace bandwidth with storage
  - Pre-processing is cheaper than bandwidth
- No alternative: Common video codecs very slow >4K
- Potential for various future apps (VR, mobile, internet)

Thank you for your attention.

http://graphics.tu-bs.de

Appendix

Bibliography (excerpt)

- Slide 5

  - Wilson S. Geisler et al. “Implementation of a foveated image coding system for image bandwidth reduction”. 1996.
  - Brian Guenter et al. “Foveated 3D Graphics”. In: ACM Trans. Graph. 31.6 (Nov. 2012), 164:1–164:10. issn: 0730-0301.

- Slide 16

  - Piotr Didyk et al. “Apparent display resolution enhancement for moving images”. In: ACM Transactions on Graphics (TOG). Vol. 29. 4. ACM. 2010, p. 113.

- Full bibliography in thesis p. 69-76

Sources / Licenses

- Slide 2
  - Grzegorz Rogala, Creative Commons https://www.flickr.com/photos/grzegorz_rogala/5097827282/
- Slide 3
  - JeffPerry, Public domain https://en.wikipedia.org/wiki/Foveated_imaging
- Slide 12
  - Mulad, Creative Commons https://en.wikipedia.org/wiki/Mipmap
- Slide 20
  - Emil Persson, Creative Commons http://www.humus.name/index.php?page=Textures
  - David J. Eck, Creative Commons http://math.hws.edu/graphicsbook/c5/s3.html
- Evaluated scenes
  - Appendix Slide 34
- Everything else created by myself or ICG

Evaluation

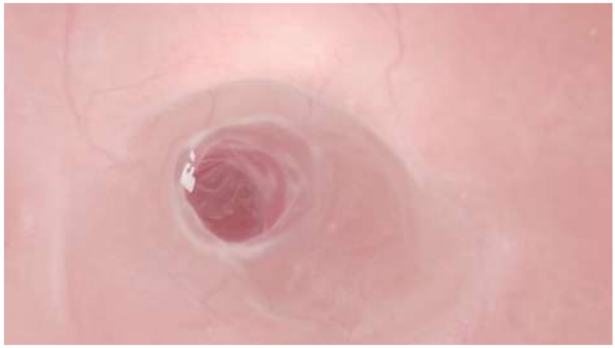
- Hardware
  - MacBook Pro
  - 2,5 GHz Intel Core i7 Haswell CPU
  - NVIDIA GeForce GT 750M 2048 MB
  - 16 GB DDR3 RAM
  - SSD
- Software
  - SDL 2.0
  - OpenCV 3.1
  - FFmpeg 2.8.6
  - STB_Image 2.8
  - PIL 3.0

  - Mac OS X 10.11.3
  - C++ ’14 (Main)
  - Python 3.5.1 (Pre-Proc)

Evaluation Scenes

- Scene 1: Caminandes
  - Wide Screen
  - Original 4K, upscaled 8K/16K
  - Blender Foundation (Creative Com.)
  - http://www.caminandes.com

- Scene 2: Inside the GI tract
  - Panoramic
  - Original 4K, upscaled 16K
  - HybridMedical (Creative Commons)
  - https://www.youtube.com/watch?v=upSnH436ya8



- Scene 3: Christmas market
  - Panoramic
  - Original 12K
  - Captured with GoPro (Matthias)

Evaluation

- Grid
  - 8x8 for widescreen 16K
  - 4x4 everything else
- Mip-map levels: 3
  - Full resolution
  - Mid resolution = 1 / 4 times full
  - Low resolution = 1 / 16 times full

Evaluation Video RAM

Approach	Data	Percentage
Acuity mask	2.76 mb	0.69
Interpolation mask	44.00 mb	11.06
Total used by approach	46.76 mb	11.75
Total used by classic video	398.00 mb	100.00

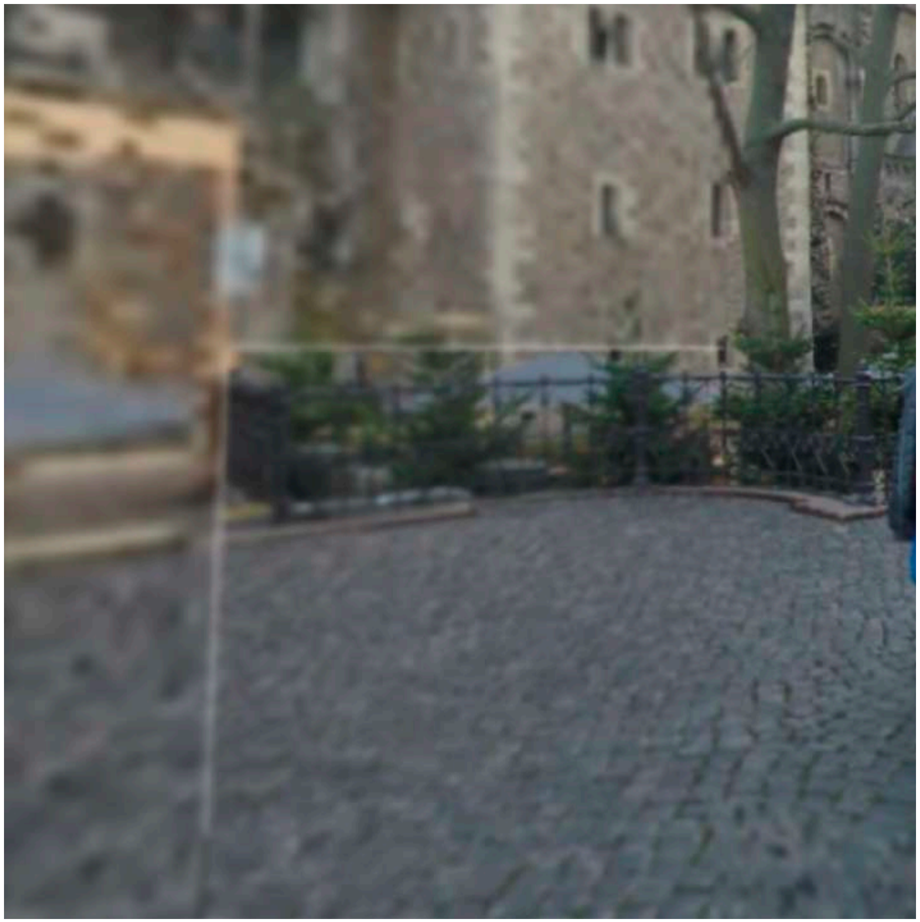
Approach	Data	Percentage
Cube map (high resolution)	1152.0 mb	0.69
Cube map (mid resolution)	72.0 mb	11.06
Cube map (low resolution)	4.5 mb	11.75
Total used by appoach	1228.5 mb	106.64
Total used by classic video	1152.0 mb	100.00

Evaluation Pre-processing

Approach	Time spent	Storage used
Framework 1: VP 8	3 h 58 min	3237 mb
Framework 1: VP 9	8 h 17 min	2294 mb
Framework 1: h.284	0 h 36 min	1923 mb
Framework 1: h.285	1 h 30 min	104 mb
Framework 2: PNG	50 min	1608 mb
Framework 2: JPEG q=90	17 min	816 mb
Framework 2: JPEG q=100	21 min	2500 mb

Approach	Mip-map 0	Mip-map 1	Mip-map 2
Framework 1 (Scene 1)	1885 MB	365 mb	46 mb
Framework 2 (Scene 2)	721 MB	81 mb	14 mb
Framework 2 (Scene 3)	3520 MB	463 mb	92 mb

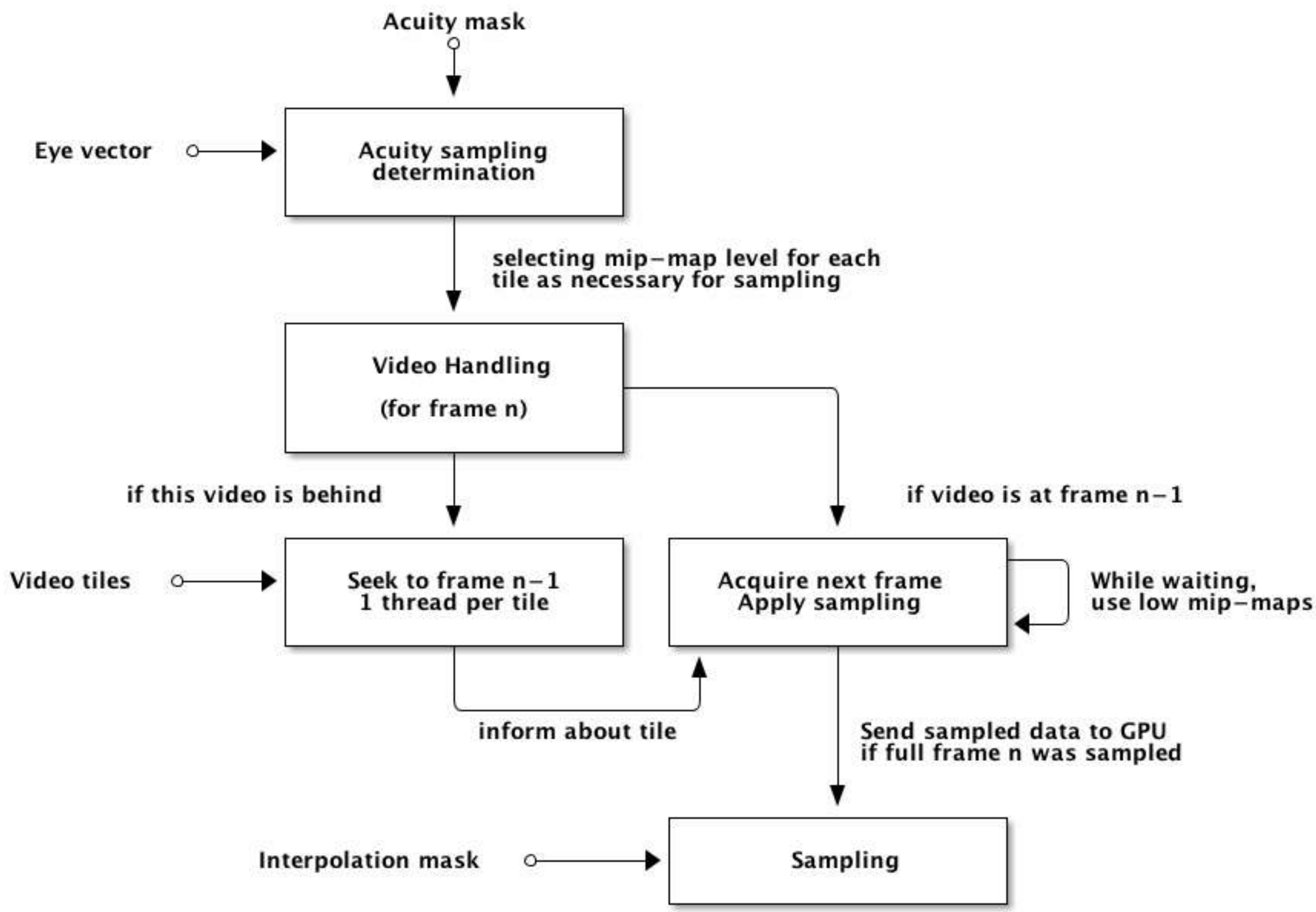
Evaluation Perception



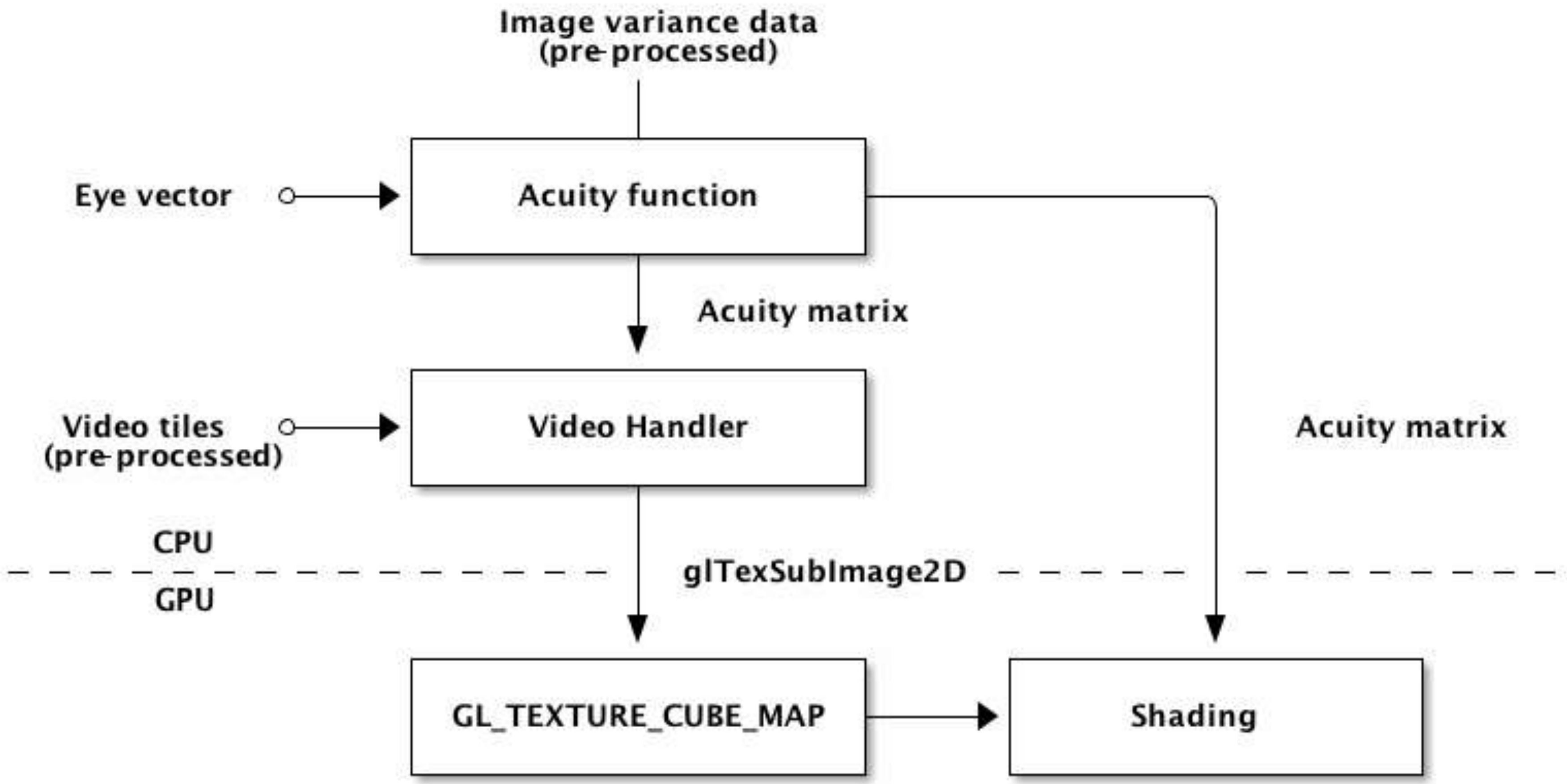
Perception optimization

- Saliency-based approaches
- Post-processing
  - Radial blur filter / Gauss blur
- Data
  - Bigger Grid size
  - Higher mip-map levels
  - Less distance between mip-map levels

Flowchart Multithread Model



Flowchart Panoramic Streaming



Acuity calculations

```
side, axis calculateCubeside(vec3 eye):
    switch eye:
        eye.x >  0.5: return +X, X
        eye.x <= 0.5: return -X, X
        eye.y >  0.5: return +Y, Y
        eye.y <= 0.5: return -Y, Y
        eye.z >  0.5: return +Z, Z
        eye.z <= 0.5: return -Z, Z
```

```
vec2 calculateGridPosition(vec3):
    side, axis = calculateCubeside(eye)
    otheraxis1 = (getAxisUnequal(eye, axis).a + 0.75) * 2
    otheraxis2 = (getAxisUnequal(eye, axis).b + 0.75) * 2
    return (otheraxis1 / 4.0, otheraxis2 / 4.0)
```

Acuity calculations

```
void selectMipmap(vec3 eye):
    setquality(ALL, lowest)
    // Grid-based acuity
    vec2 fovealCell = calculateGridPosition(eye)
    setquality(fovealCell, maximum)
    foreach(neighborCell : adjacent to fovealCell):
        setquality(neighborCell, halfway)
    // Variance-based acuity
    foreach(noiseCell : cell near maximum(color variance)):
        setquality(noiseCell, halfway)
```

Saliency-based approach

- Idea
  - Add analysis to pre-processing
    - Color variance (RGB/HSV/LAB)
    - Hard edges (FFT)
    - Temporal differences
  - Set mip-map level not only based on visual acuity but also above analysis
  - For example:
    - use mid resolution instead of low resolution if salient feature in tile

HMD System

- Current implementation: ”Debug”
  - Eye position = Mouse pointer
  - No stereoscopic rendering
- Port to HMD: No obstacles
  - Code is standard C++ ‘14
  - All cross-platform libraries
  - Stereoscopic rendering presumably no major influence on performance

Limitations

- Hardware
  - GPU least busy component
  - Due to parallel video access SSD necessary
- Pre-processing necessary
  - Mostly time consuming
  - Overhead in storage
- Grid

Future work

- Perceptual user study to verify results
- Bandwidth allows new applications
  - Mobile devices
  - Remote streaming
    - Internet/Cloud?

- Approach
  - Other video codecs
  - Container format, storage optimizations
  - Add saliency-based approaches (variances, edges)