



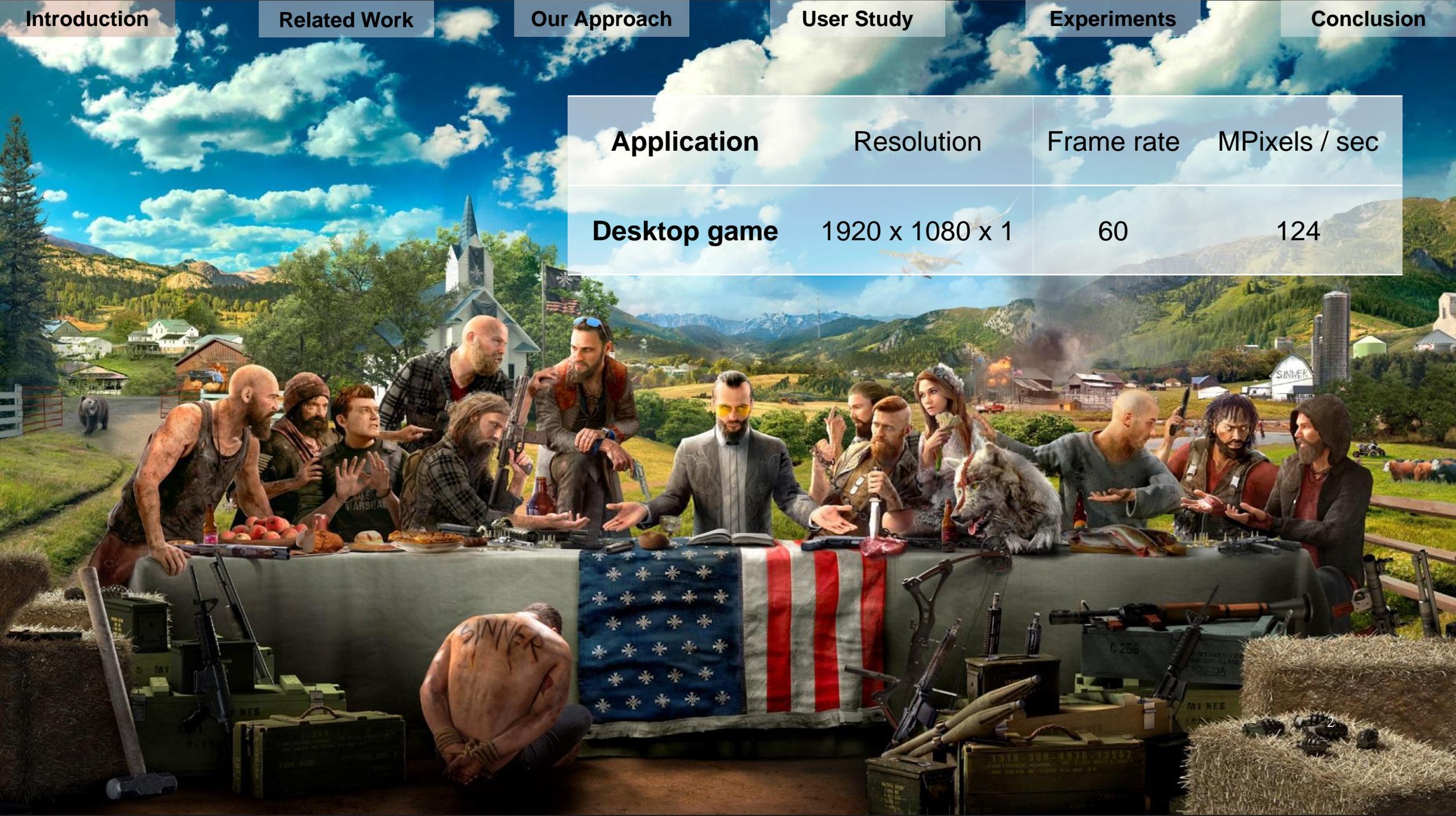
Kernel Foveated Rendering

Xiaoxu Meng, Ruofei Du, Matthias Zwicker and Amitabh Varshney

Augmentarium | UMIACS

University of Maryland, College Park

ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2018



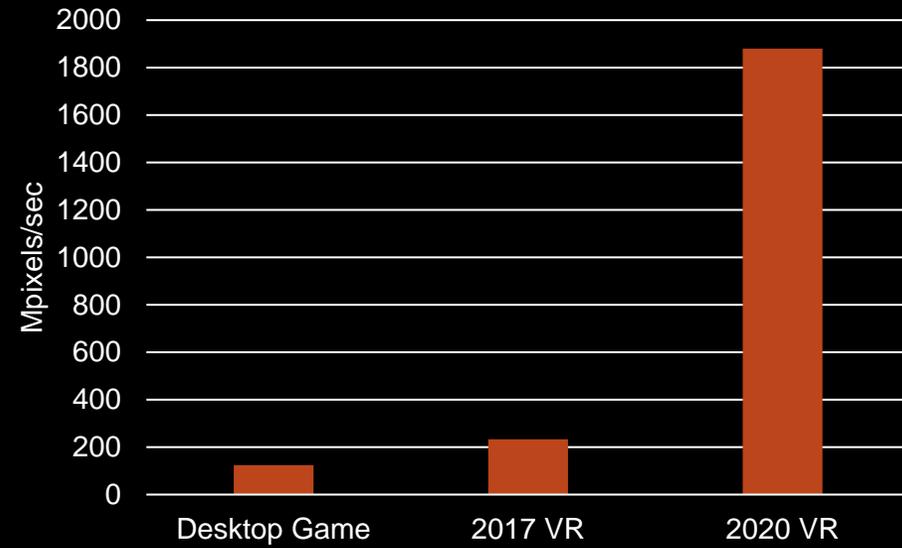
Application	Resolution	Frame rate	MPixels / sec
Desktop game	1920 x 1080 x 1	60	124



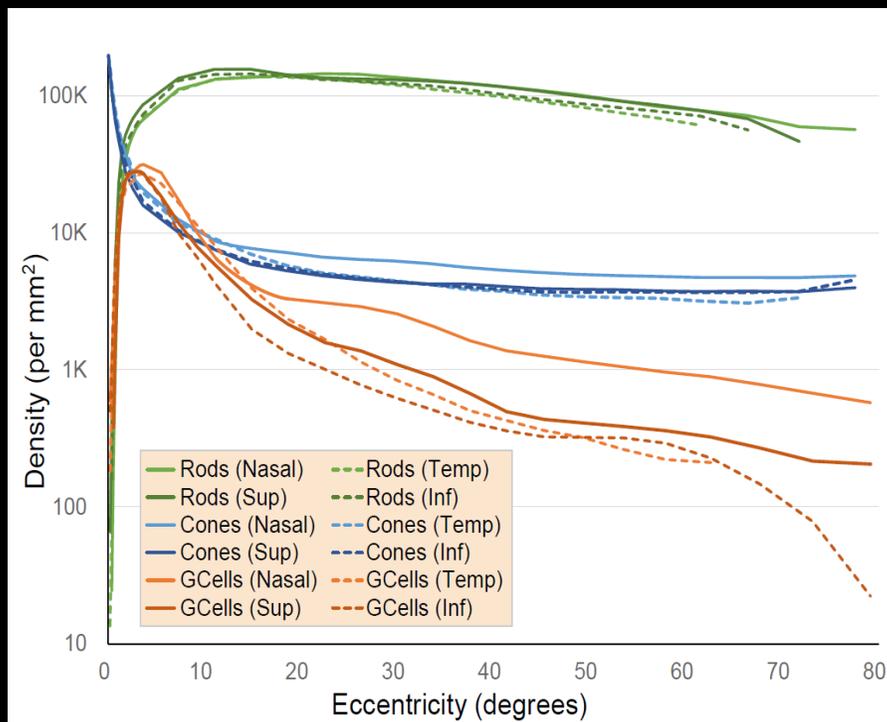
Application	Resolution	Frame rate	MPixels / sec
Desktop game	1920 x 1080 x 1	60	124
2018 VR (HTC Vive PRO)	1440 x 1600 x 2	90	414



Application	Resolution	Frame rate	MPixels / sec
Desktop game	1920 x 1080 x 1	60	124
2018 VR (HTC Vive PRO)	1440 x 1600 x 2	90	414
2020 VR *	4000 x 4000 x 2	90	2,880

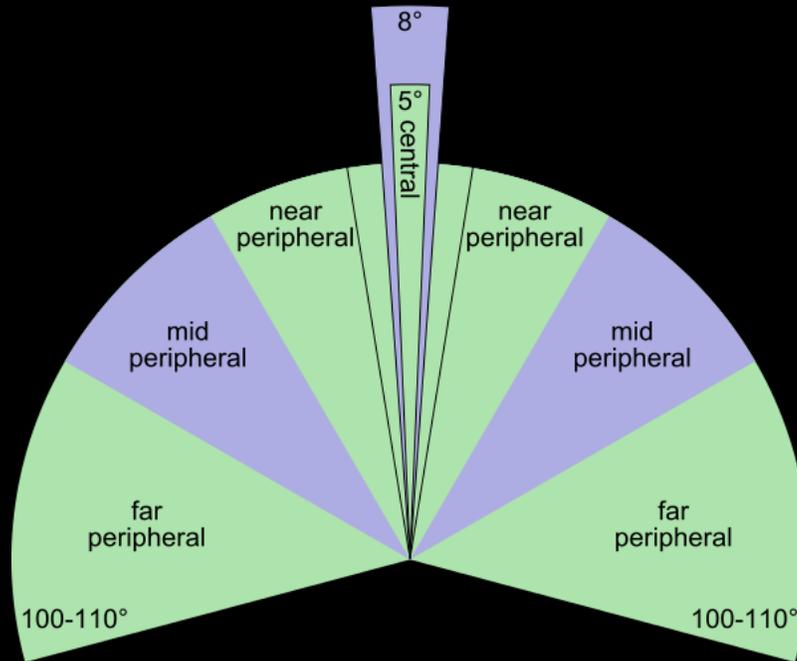


- Virtual reality is a challenging workload



fovea:
the center of the retina
corresponds to the center of the vision field

- Virtual reality is a challenging workload
- Most VR pixels are peripheral



foveal region:

the human eye detects significant detail

peripheral region:

the human eye detects little high fidelity detail

- Virtual reality is a challenging workload
- Most VR pixels are peripheral



foveal region:

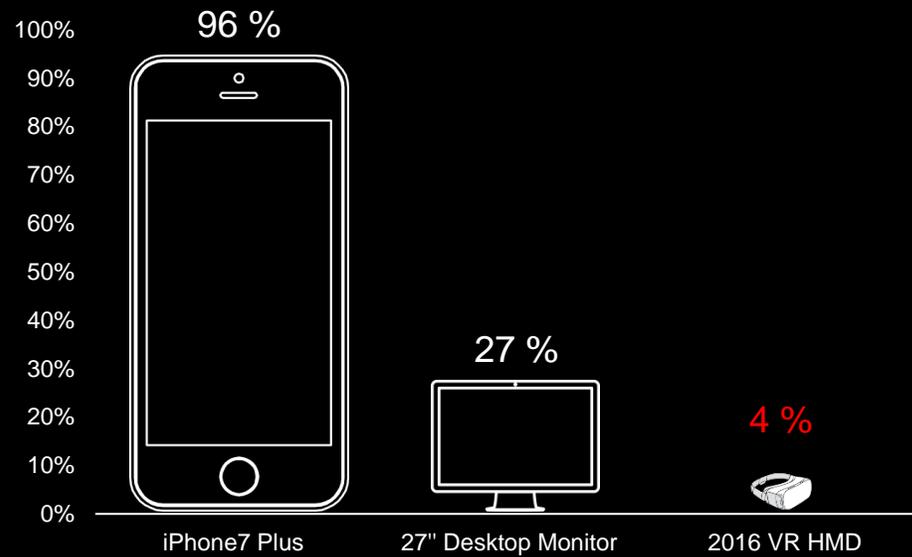
the human eye detects significant detail

peripheral region:

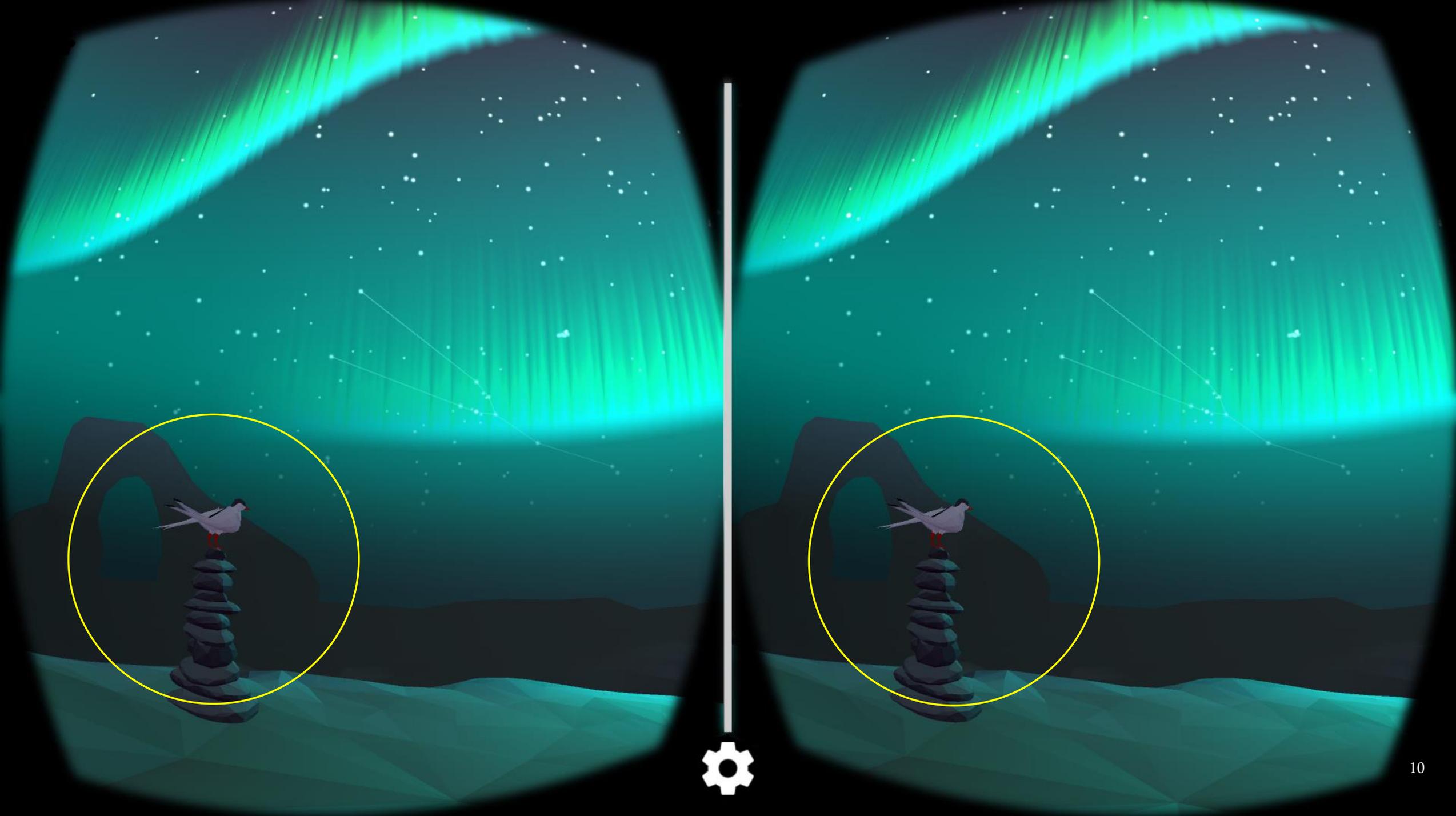
the human eye detects little high fidelity detail

- Virtual reality is a challenging workload
- Most VR pixels are peripheral

Percentage of the foveal pixels



- Virtual reality is a challenging workload
- Most VR pixels are peripheral



Foveated Rendering





- Virtual reality is a challenging workload
- Most VR pixels are peripheral
- Eye tracking technology available

Foveate

Brian Guenter Mark Finch S
Mi

Abstract

exploit the falloff of acuity in the visual periphery to accelerate graphics computation by a factor of 5-6 on a desktop HD display (20x1080). Our method tracks the user's gaze point and renders image layers around it at progressively higher angular sizes. The three layers are then magnified to display at the target sampling rate. The three layers are then magnified to display at the target sampling rate. The three layers are then magnified to display at the target sampling rate. The three layers are then magnified to display at the target sampling rate.

We performed a user study to validate these results. It includes two levels of foveation quality: a more conservative one in which users reported foveated rendering quality as equivalent to or better than non-foveated when directly shown both, and a more aggressive one in which users were unable to correctly label as increasing or decreasing a short quality progression relative to a high-foveated reference. Based on this user study, we obtain a value for the model of 1.32-1.65 arc minutes per degree of eccentricity. This allows us to predict two future advantages of foveated rendering: (1) bigger savings with larger, sharper displays than currently (e.g. 100 times speedup at a field of view of 70 degrees), and (2) a roughly linear resolution matching foveal acuity, and (2) a roughly linear resolution matching foveal acuity, and (2) a roughly linear resolution matching foveal acuity, and (2) a roughly linear resolution matching foveal acuity.

Keywords: antialiasing, eccentricity, minimum angle of resolution (MAR), multiresolution gaze-contingent display (MGCD).

Links: DL PDF WEB VIDEO

1 Introduction

We see 135° vertically and 160° horizontally, but sense detail only within a 5° central circle. This tiny portion of the visual field projects to the retinal region called the fovea, packed with color cone receptors. The angular distance from the central gaze direction is called eccentricity. Acuity drops rapidly as eccentricity increases due to reduced receptor density in the retina, reduced optical nerve "bandwidth" and other factors.

¹A smaller region of 1° diameter, called the foveola, is often the site of foveal vision.

High Performance Graphics (2014)
Jonathan Ragan-Kelley and Ingo Wald (Editors)

Coarse Pixel Shading

K. Vaidyanathan¹, M. Salvati¹, R. Toth¹, T. Foley¹, T. Akenine-Olson¹, J. Munkberg¹, J. Hasselgren¹, M. Sugihara¹, P. Claiborn¹, T. Scahill¹

¹Intel Corporation, ²Lund University

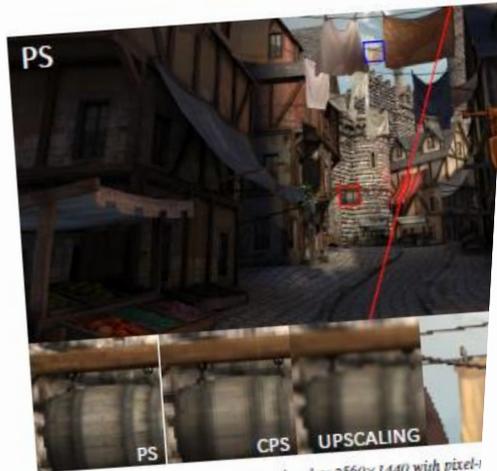


Figure 1: The CITADEL scene, rendered at 2560x1440 with pixel-level Coarse Pixel Shading (CPS) on the right, using a coarse pixel size of 2x2. CPS almost has no perceptible differences on a high pixel density display, with a structural rendering at 1280x720 and upscaled exhibits blurring at silhouette edge.

Abstract

We present a novel architecture for flexible control of shading rates for various applications. We quantize shading rates to a finite set of screen-aligned grid cells. This pipeline compared to alternative approaches. Our architecture provides control of the shading rate, which enables efficient shading in many displays, foveated rendering, and adaptive shading for multiple rates in a single pass, which allows the user to control their frequency content.

Categories and Subject Descriptors (according to ACM CCS): Graphics processors

© The Eurographics Association 2014

DOI: 10.1111/cgf.12956
Eurographics Symposium on Rendering 2016
E. Eisemann and E. Fiume
(Guest Editors)

Adaptive Gaze-Con

Michael Stengel¹, Steffen

¹TU Braunschweig

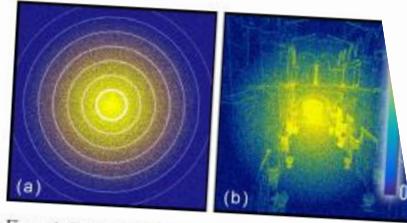


Figure 1: Gaze-contingent Rendering Pipeline. Incorporating a perceptually-adaptive sampling pattern (b). Sparse shading (c) each fragment at a fraction of the original shading costs. The result shows reduced detail in the periphery (flowers inset).

Abstract

With ever-increasing display resolution for wide field-of-view displays—shading has become the major computational cost in rendering. In contrast to previous approaches we design a scheme that incorporates multiple aspects of the human visual system (e.g. material or lighting properties), and brightness adaptation, to shade the image's perceptually relevant fragments while preserving detail in the fovea. Our approach does not impose any restrictions on the rendering pipeline. We validate our approach in experiments to validate scene- and task-independence of our approach, reduced by 50% to 80%. Our algorithm scales favorably with increasing resolution for head-mounted displays and wide-field-of-view projection.

Categories and Subject Descriptors (according to ACM CCS): I.3.1 Realism—Virtual Reality I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism

1. Introduction

Modern rasterization algorithms can generate photo-realistic images. The computational cost for creating such images is mainly governed by the cost induced by shading computations. Shading has become the limiting factor in real-time rendering with ever-increasing display resolution, especially for wide field-of-view (FOV) displays such as head-mounted displays (HMD) or wide-screen projection systems.

© 2016 The Author(s)
Computer Graphics Forum © 2016 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

Towards Foveated Rendering for Gaze-Tracked Virtual Reality

Anjul Patney*

Marco Salvati

Joochwan Kim
David Luebke

Anton Kaplanyan
Aaron Lefohn

Chris Wyman

Nir Benty

NVIDIA

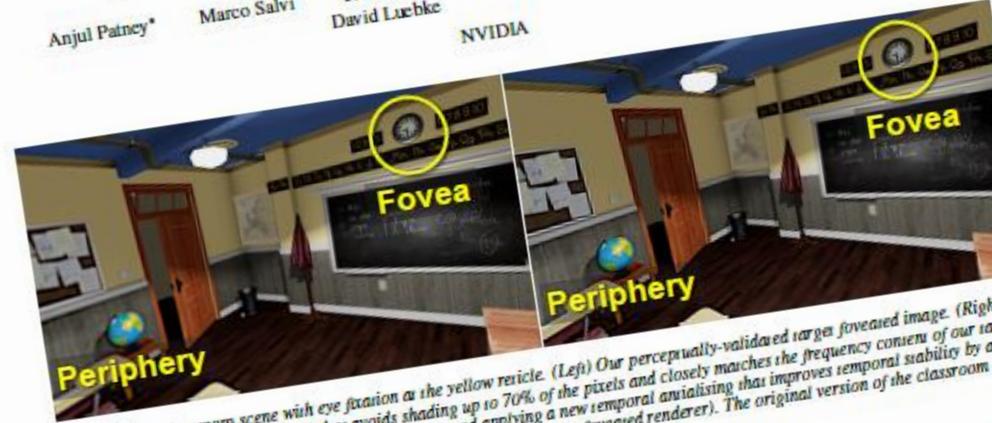


Figure 1: Our classroom scene with eye fixation at the yellow circle. (Left) Our perceptually-validated target foveated image. (Right) Our proposed foveated rendering system that avoids shading up to 70% of the pixels and closely matches the frequency content of our target image using pre-filtered shading terms, contrast preservation, and applying a new temporal antialiasing that improves temporal stability by a factor of 1.5. (Right) The original version of the classroom scene with eye fixation at the yellow circle. (Right) Our proposed foveated rendering system that avoids shading up to 70% of the pixels and closely matches the frequency content of our target image using pre-filtered shading terms, contrast preservation, and applying a new temporal antialiasing that improves temporal stability by a factor of 1.5.

Abstract

Foveated rendering synthesizes images with progressively less detail outside the eye fixation region, potentially unlocking significant speedups for wide field-of-view displays, such as head mounted displays, where target framerate and resolution is increasing faster than the performance of traditional real-time renderers.

To study and improve potential gains, we designed a foveated rendering system to evaluate the perceptual abilities of human peripheral vision when viewing today's displays. We determined that filtering peripheral regions reduces contrast, inducing a sense of tunnel vision. When applying a postprocess contrast enhancement, subjects tolerated up to 2x larger blur radius before detecting differences from a non-foveated ground truth. After verifying these insights on both desktop and head mounted displays augmented with high-speed gaze-tracking, we designed a perceptual target image to strive for when engineering a production foveated renderer.

Given our perceptual target, we designed a practical foveated rendering system that reduces number of shades by up to 70% and allows coarsened shading up to 30° closer to the fovea than Guenter et al. [2012] without introducing perceivable aliasing or blur. We filter both pre- and post-shading to address aliasing- and saccade-induced temporal antialiasing algorithm, and use contrast enhancement to help recover peripheral details that are resolvable by our eye but degraded by filtering.

We validate our system by performing another user study. Frequency analysis shows our system closely matches our perceptual target. Measurements of temporal stability show we obtain quality similar to temporally filtered non-foveated renderings.

*apatney@nvidia.com
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components owned by others than the author(s) must be acknowledged. Request permission to reproduce all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components owned by others than the author(s) must be acknowledged. Request permission to reproduce all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components owned by others than the author(s) must be acknowledged.

Keywords: foveated rendering, gaze-tracking, perceptual reality
Concepts: Computing methodologies → Graphics systems; Human-computer interaction → Virtual reality; Perception; Virtual reality

1 Introduction

Even with tremendous advances in graphics hardware, the demand for real-time rendering systems has increased. Adoption of realistic lighting and physically based shading [Kajiya and Van Dam 1985; Humphreys 2010; Hill et al. 2015] has amplified the demand for high-quality rendering. Virtual reality (VR) has increased display resolution and refresh rates. In addition, the trend toward rendering on mobile devices such as phones, tablets, and portable gaming consoles motivates the goal of achieving the highest possible quality using minimal computation.

As a result, algorithms that imperceptibly reduce detail in the peripheral regions of the visual field are becoming more important. Interestingly, human visual acuity decreases between the retina center (the fovea) and the periphery. This is due to the larger size of photoreceptors in the periphery and the larger size of photoreceptors in the periphery. This is due to the larger size of photoreceptors in the periphery and the larger size of photoreceptors in the periphery.

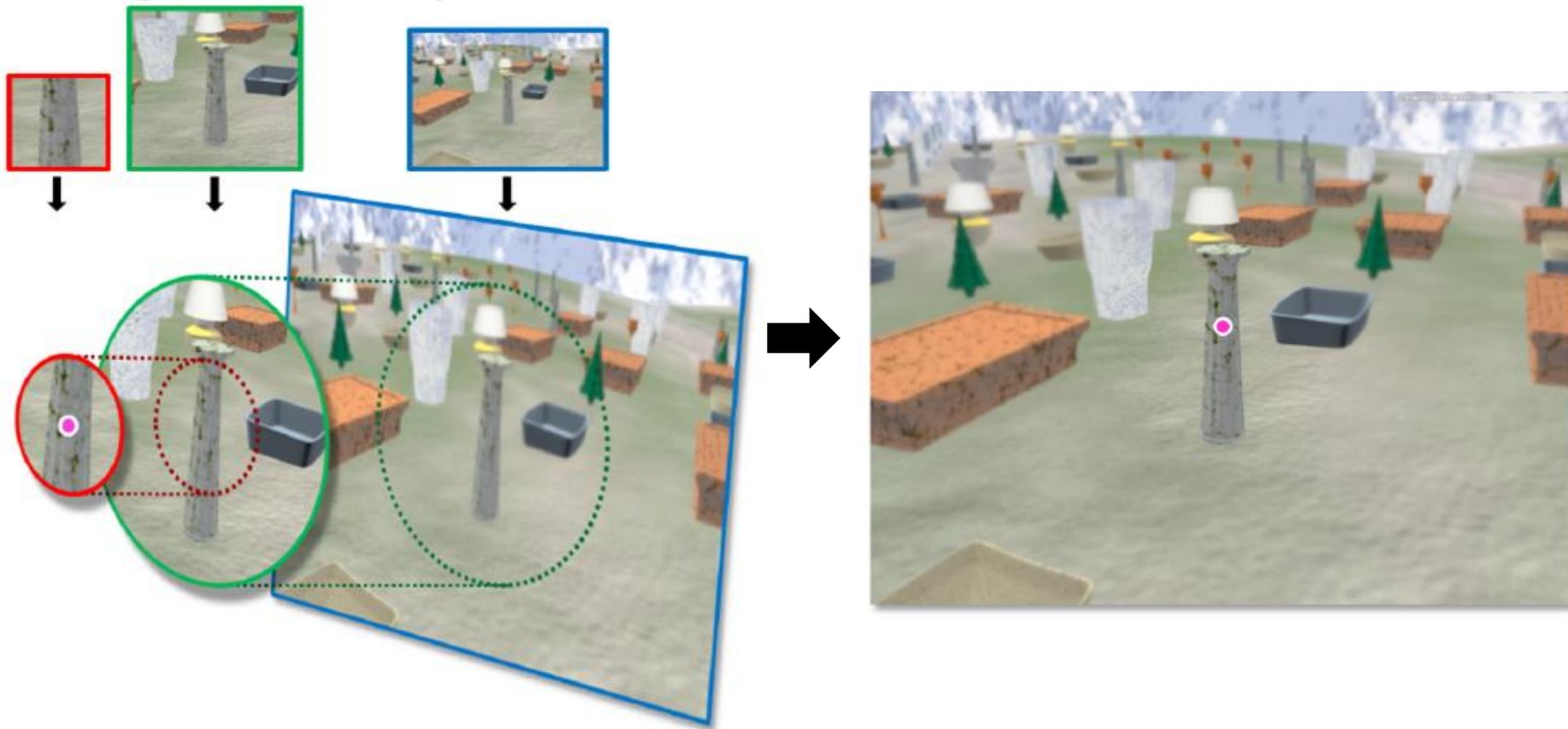
This work owned by others than the author(s) must be acknowledged. Request permission to reproduce all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components owned by others than the author(s) must be acknowledged. Request permission to reproduce all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components owned by others than the author(s) must be acknowledged.

Multi-Pass Foveated Rendering [Guenter et al. 2012]

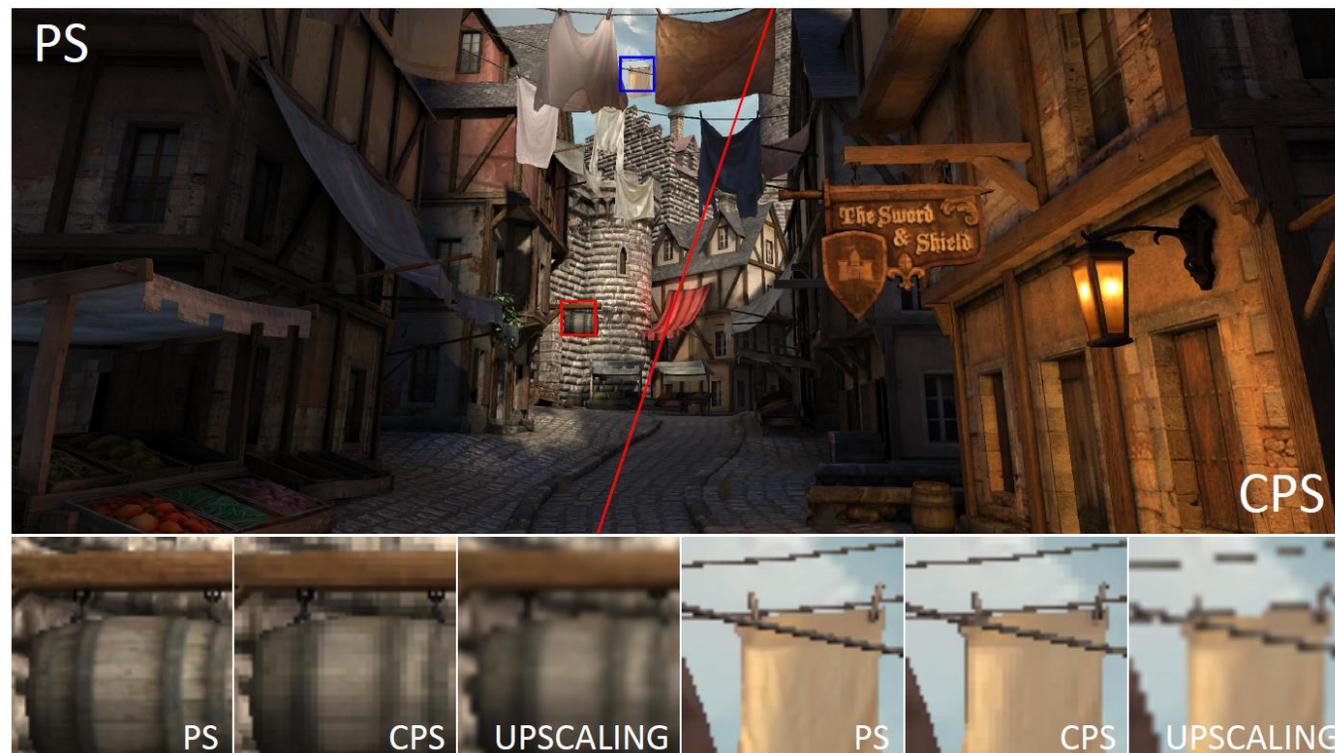
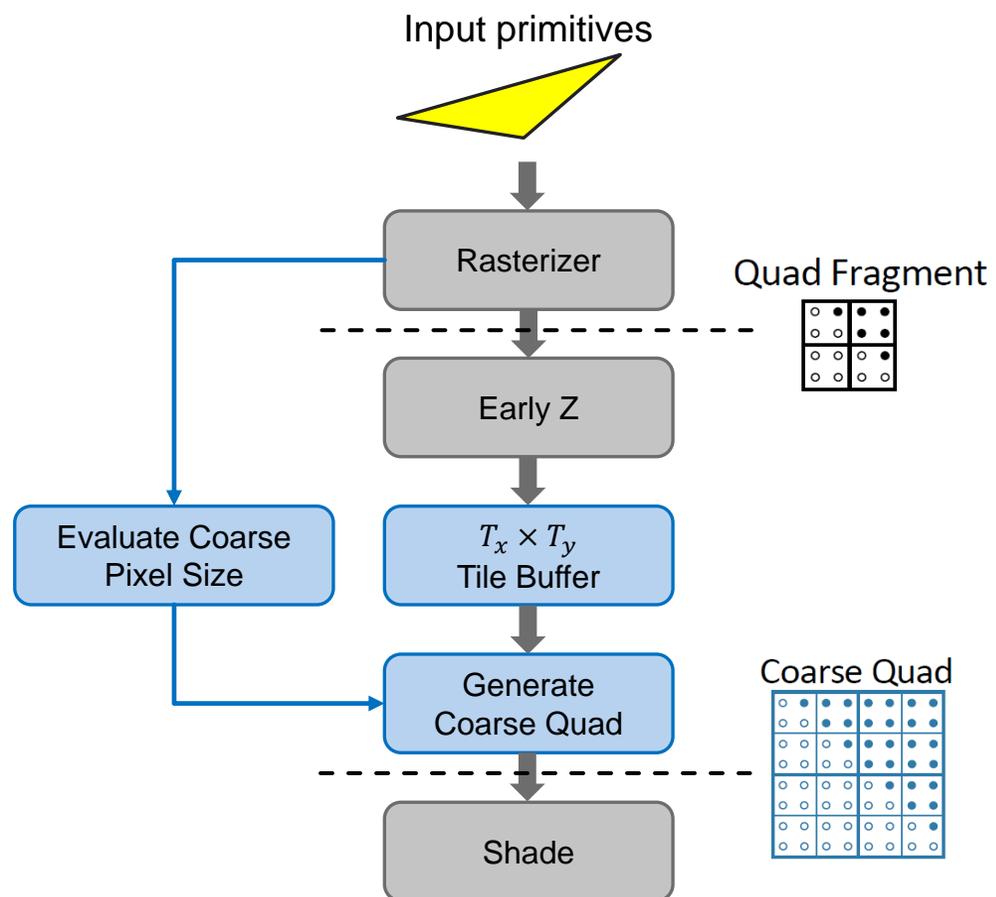
Full Resolution

$\frac{1}{2}$ Resolution

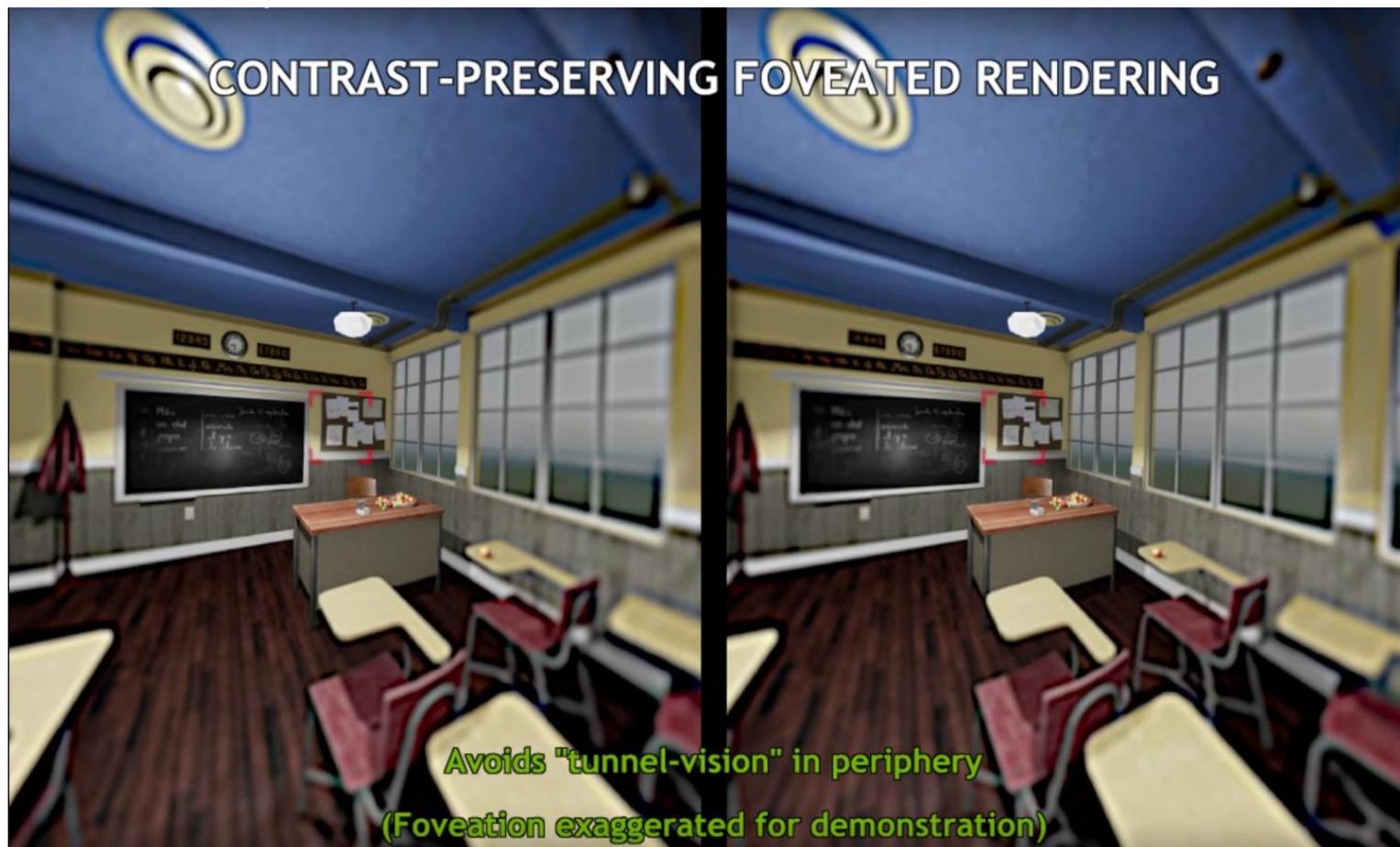
$\frac{1}{4}$ Resolution



Coarse Pixel Shading (CPS) [Vaidyanathan et al. 2014]

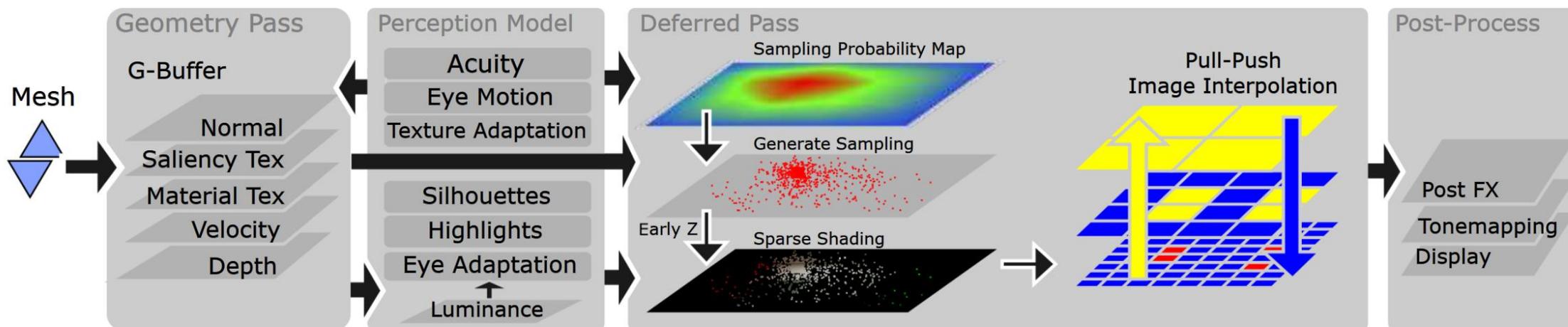


CPS with TAA & Contrast Preservation [Patney et al. 2016]



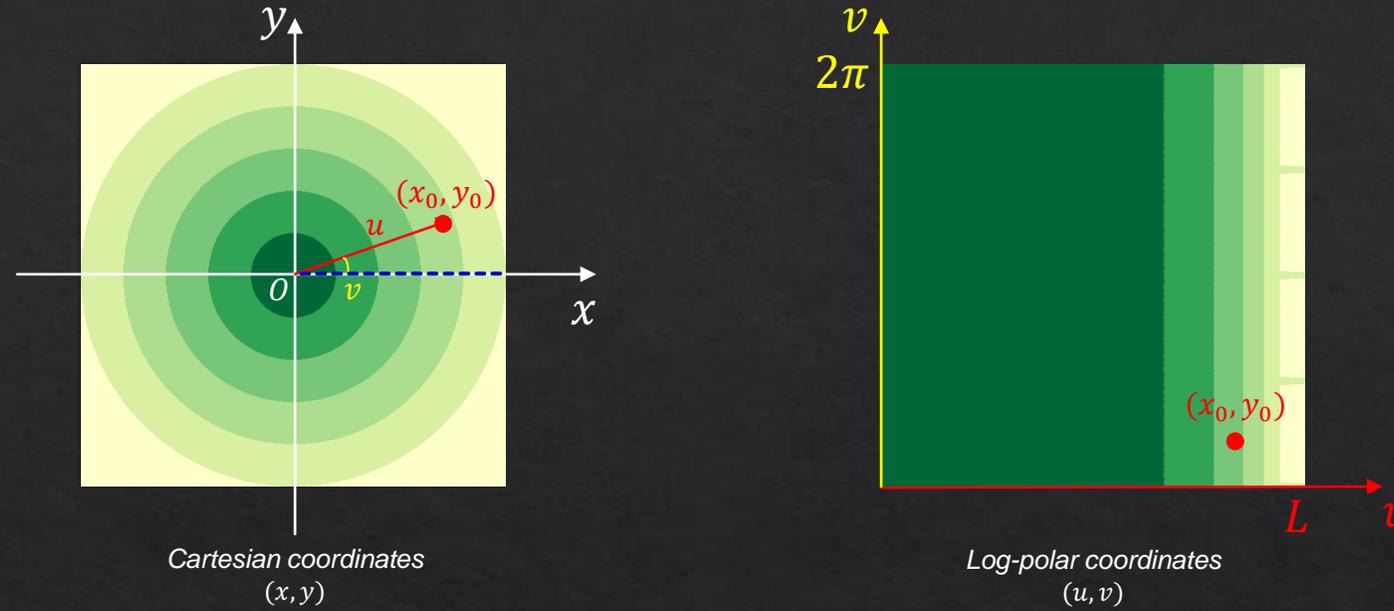
Can we change the resolution gradually?

Perceptual Foveated Rendering [Stengel et al. 2016]



Is there a foveated rendering approach
without
the expensive pixel interpolation?

Log-polar mapping [Araujo and Dias 1996]



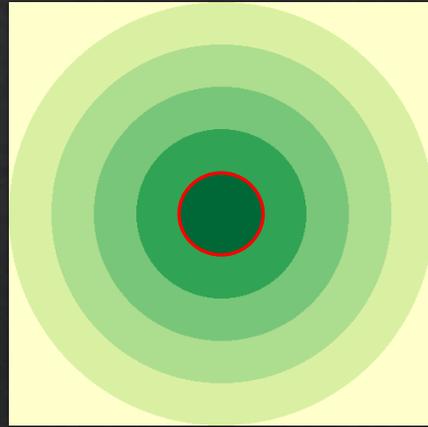
Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

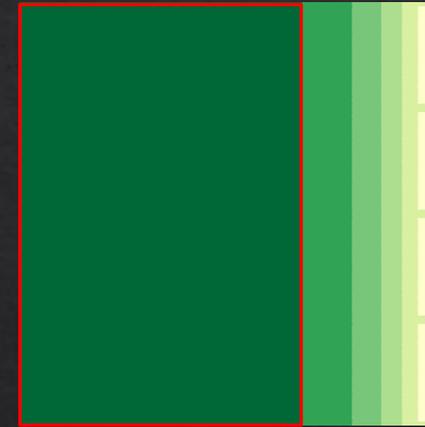
$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)

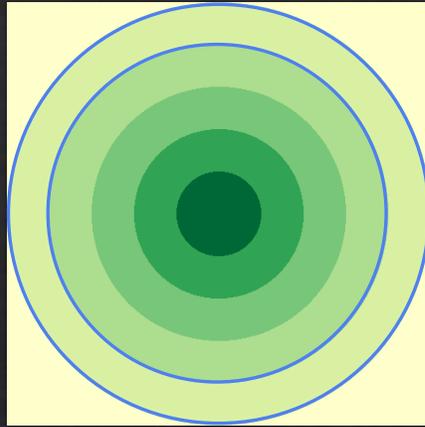
Log-polar Mapping

$$u = \frac{\log \sqrt{x^2 + y^2}}{L} \cdot w$$

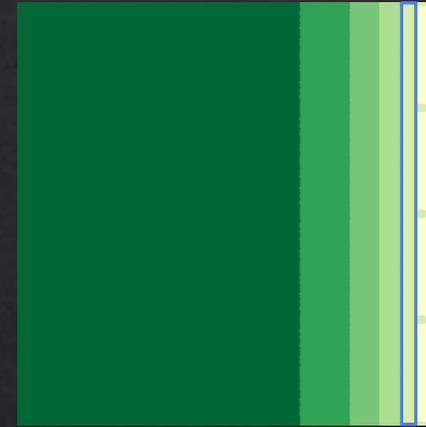
$$v = \frac{(\arctan \frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log \sqrt{W^2 + H^2}$

Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
 (x, y)



Log-polar coordinates
 (u, v)

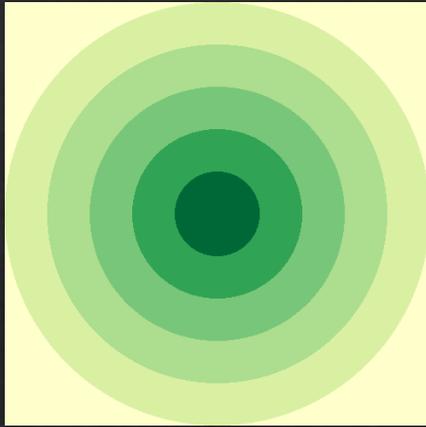
Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

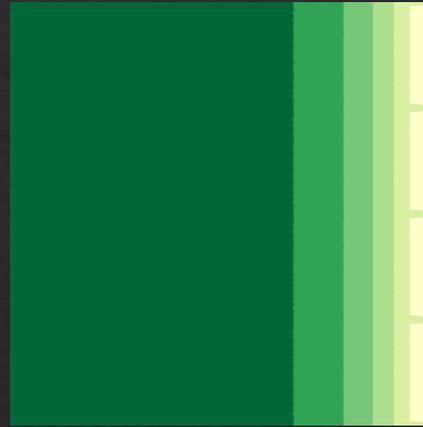
$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

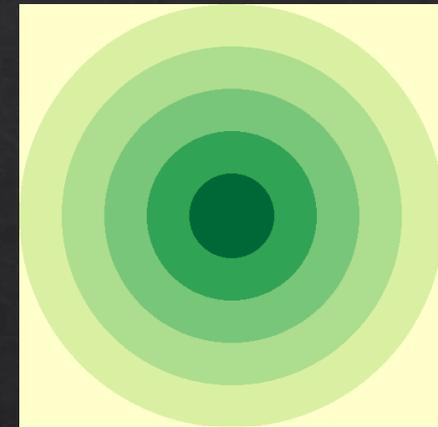
Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

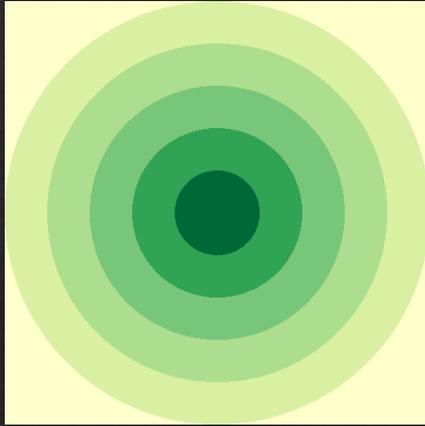
$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

$$x = e^{L\frac{u}{w}} \cos\left(v \cdot \frac{2\pi}{h}\right)$$

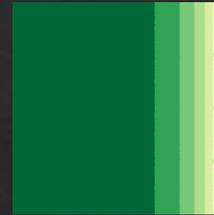
$$y = e^{L\frac{u}{w}} \sin\left(v \cdot \frac{2\pi}{h}\right)$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

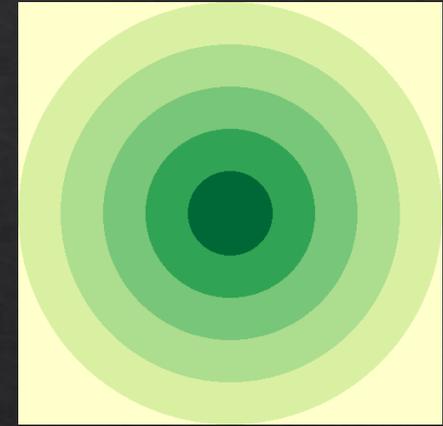
Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

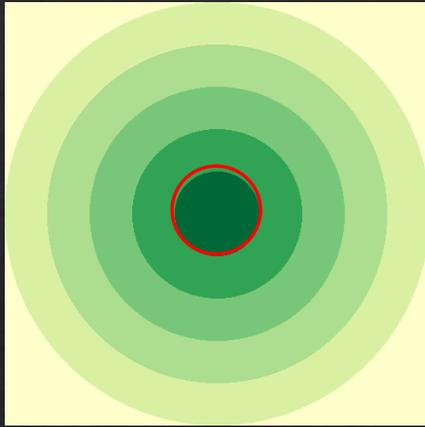
$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

$$x = e^{L\frac{u}{w}} \cos\left(v \cdot \frac{2\pi}{h}\right)$$

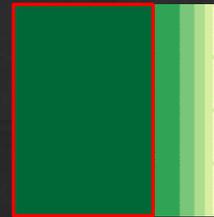
$$y = e^{L\frac{u}{w}} \sin\left(v \cdot \frac{2\pi}{h}\right)$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

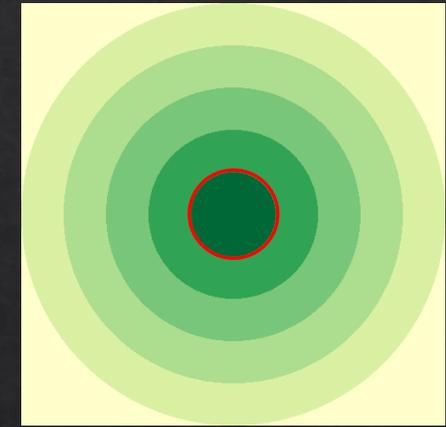
Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

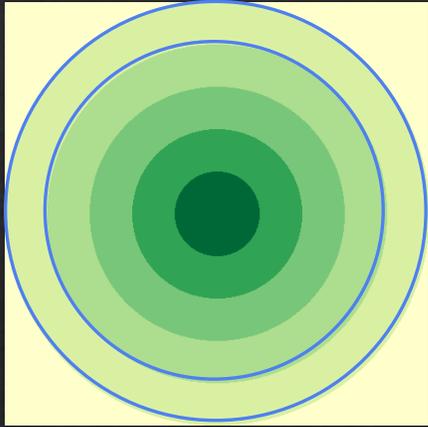
$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

$$x = e^{L\frac{u}{w}} \cos\left(v \cdot \frac{2\pi}{h}\right)$$

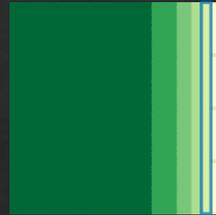
$$y = e^{L\frac{u}{w}} \sin\left(v \cdot \frac{2\pi}{h}\right)$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

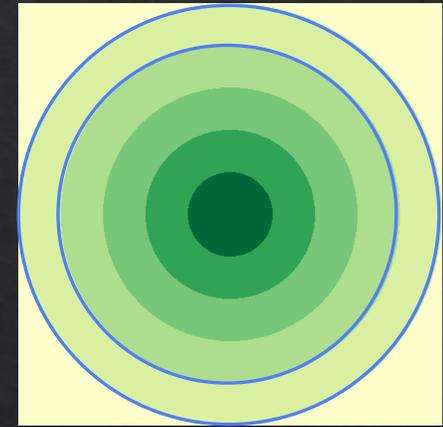
Log-polar mapping [Araujo and Dias 1996]



Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

$$x = e^{L\frac{u}{w}} \cos\left(v \cdot \frac{2\pi}{h}\right)$$

$$y = e^{L\frac{u}{w}} \sin\left(v \cdot \frac{2\pi}{h}\right)$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$

Log-polar Mapping for 2D Image [Antonelli et al. 2015]



MxM cartesian image

Log-polar
mapping



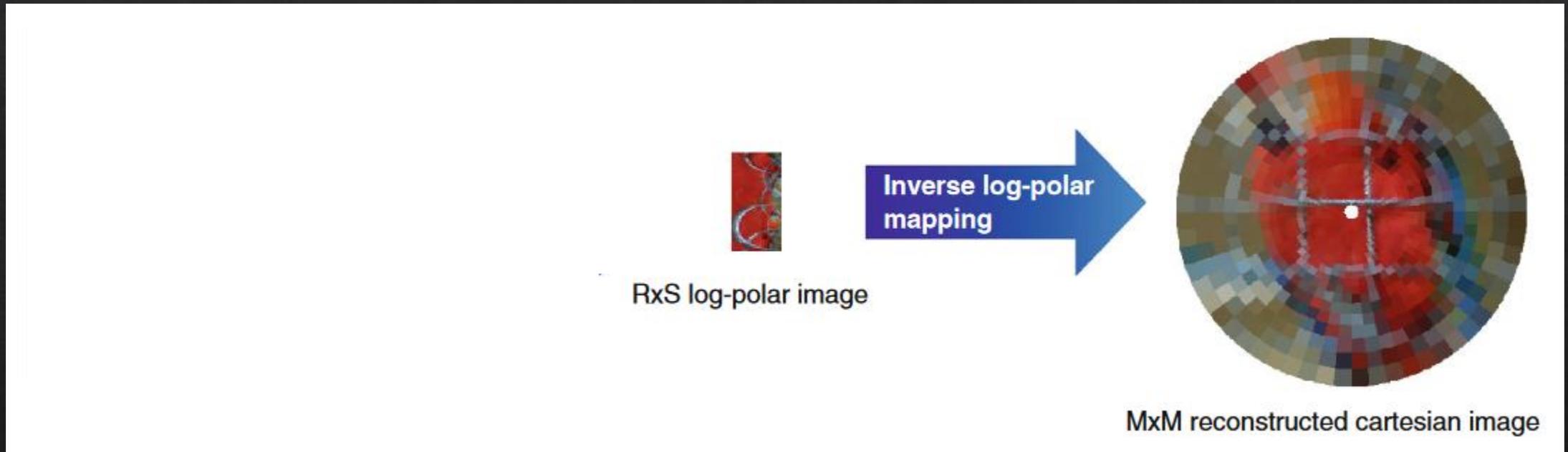
RxS log-polar image

Inverse log-polar
mapping



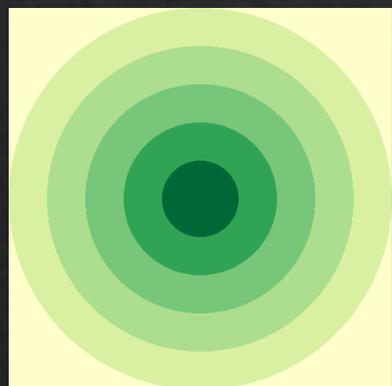
MxM reconstructed cartesian image

Log-polar Mapping for 2D Image



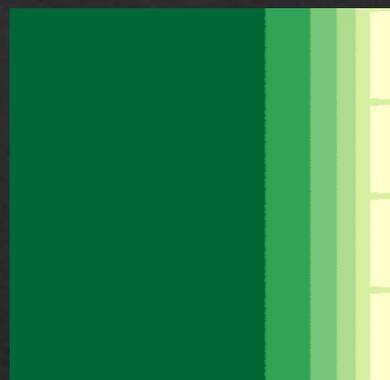
Our Approach

Kernel Log-polar Mapping

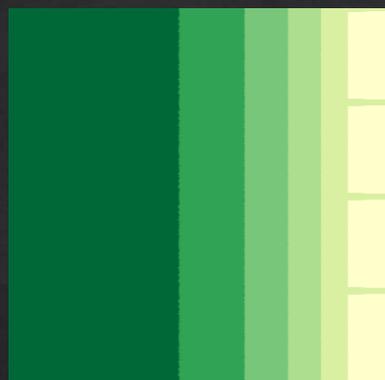


$$u = K^{-1} \left(\frac{\log \sqrt{x^2 + y^2}}{L} \right) \cdot w$$

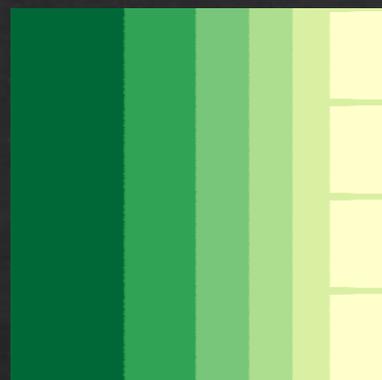
range: [0,1]



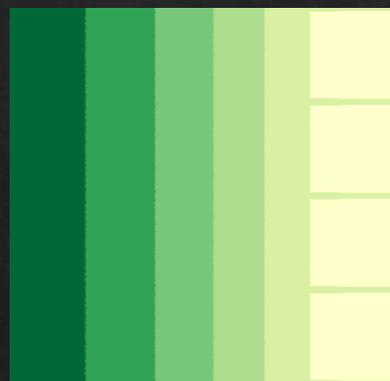
$$K(x) = x$$



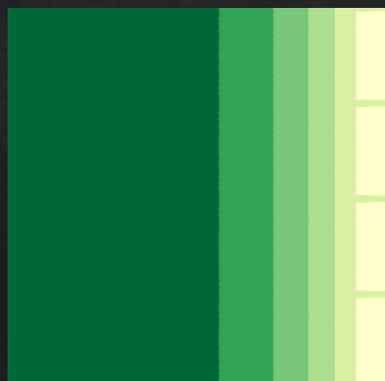
$$K(x) = x^2$$



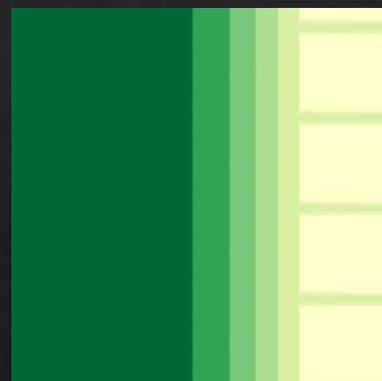
$$K(x) = x^3$$



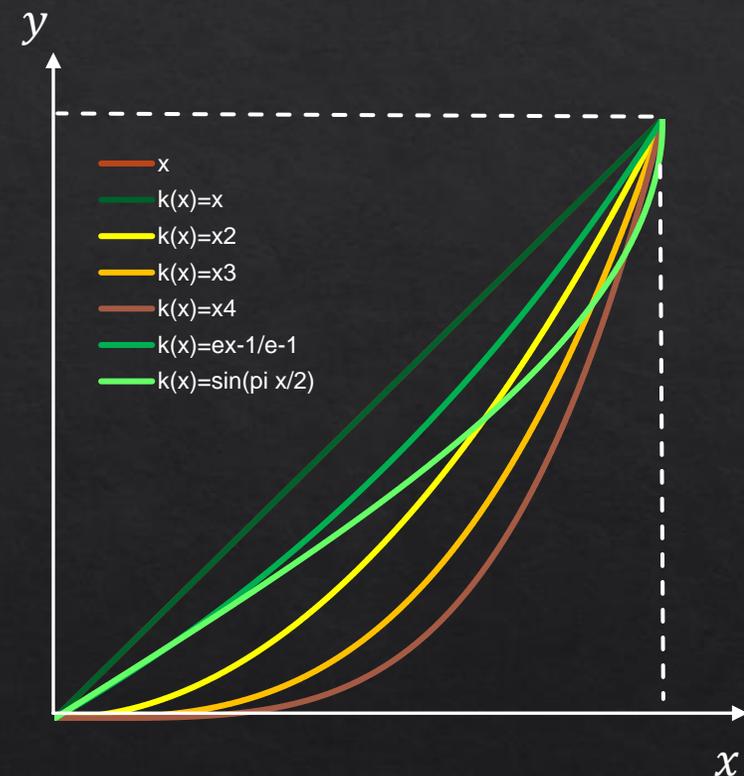
$$K(x) = x^4$$



$$K(x) = \frac{e^x - 1}{e - 1}$$

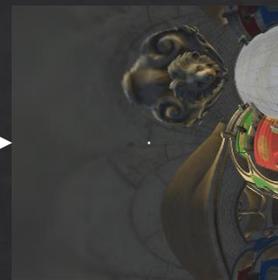


$$K(x) = \sin\left(\frac{\pi}{2} x\right)$$

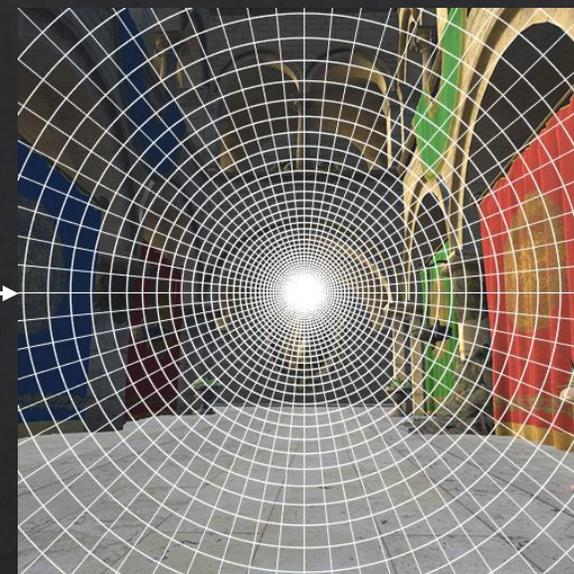




Cartesian coordinates
(x, y)



Log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Log-polar Mapping

$$u = \frac{\log\sqrt{x^2 + y^2}}{L} \cdot w$$

$$v = \frac{(\arctan\frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi)}{2\pi} \cdot h$$

$$x = e^{L\frac{u}{w}} \cos\left(v \cdot \frac{2\pi}{h}\right)$$

$$y = e^{L\frac{u}{w}} \sin\left(v \cdot \frac{2\pi}{h}\right)$$

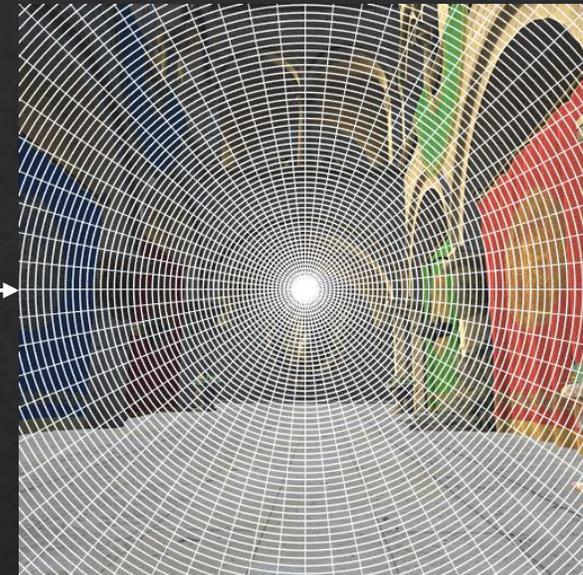
- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log\sqrt{W^2 + H^2}$



Cartesian coordinates
(x, y)



Kernel log-polar coordinates
(u, v)



Cartesian coordinates
(x, y)

Kernel Log-polar Mapping

$$u = K^{-1} \left(\frac{\log \sqrt{x^2 + y^2}}{L} \right) \cdot w$$

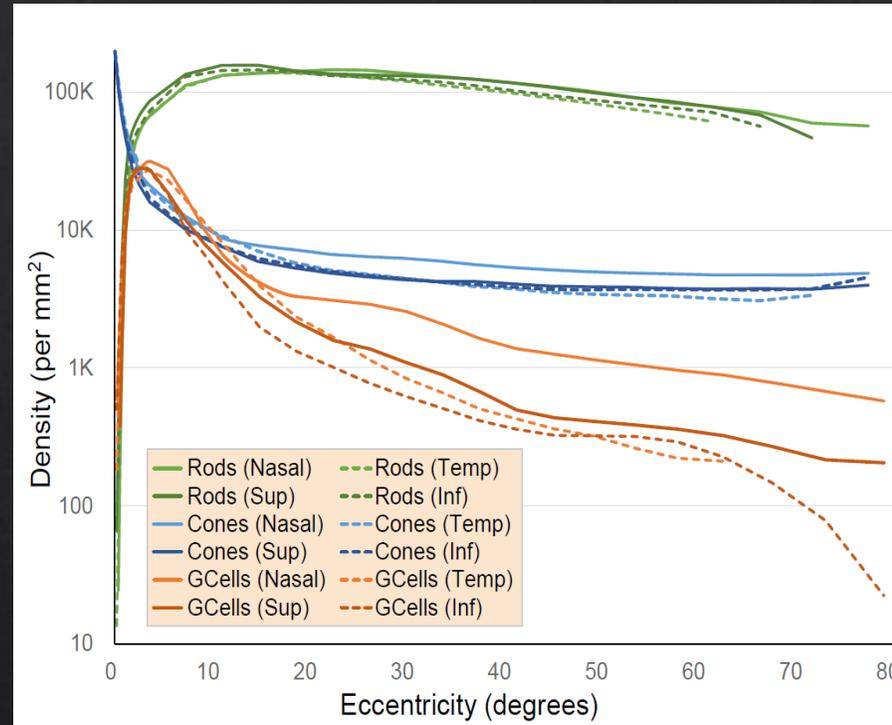
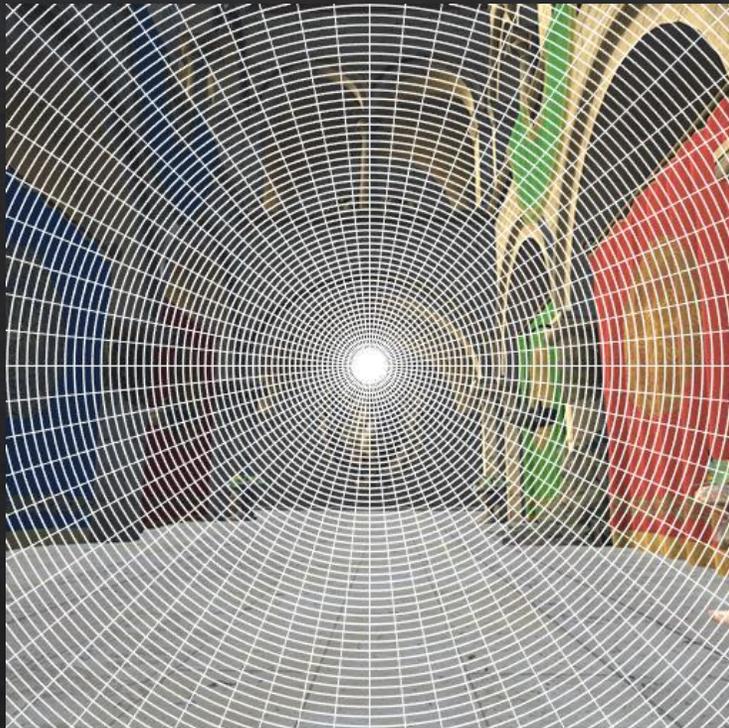
$$v = \frac{\left(\arctan \frac{y}{x} + \mathbf{1}[y < 0] \cdot 2\pi \right)}{2\pi} \cdot h$$

$$x = e^{L \cdot K \left(\frac{u}{w} \right)} \cos \left(v \cdot \frac{2\pi}{h} \right)$$

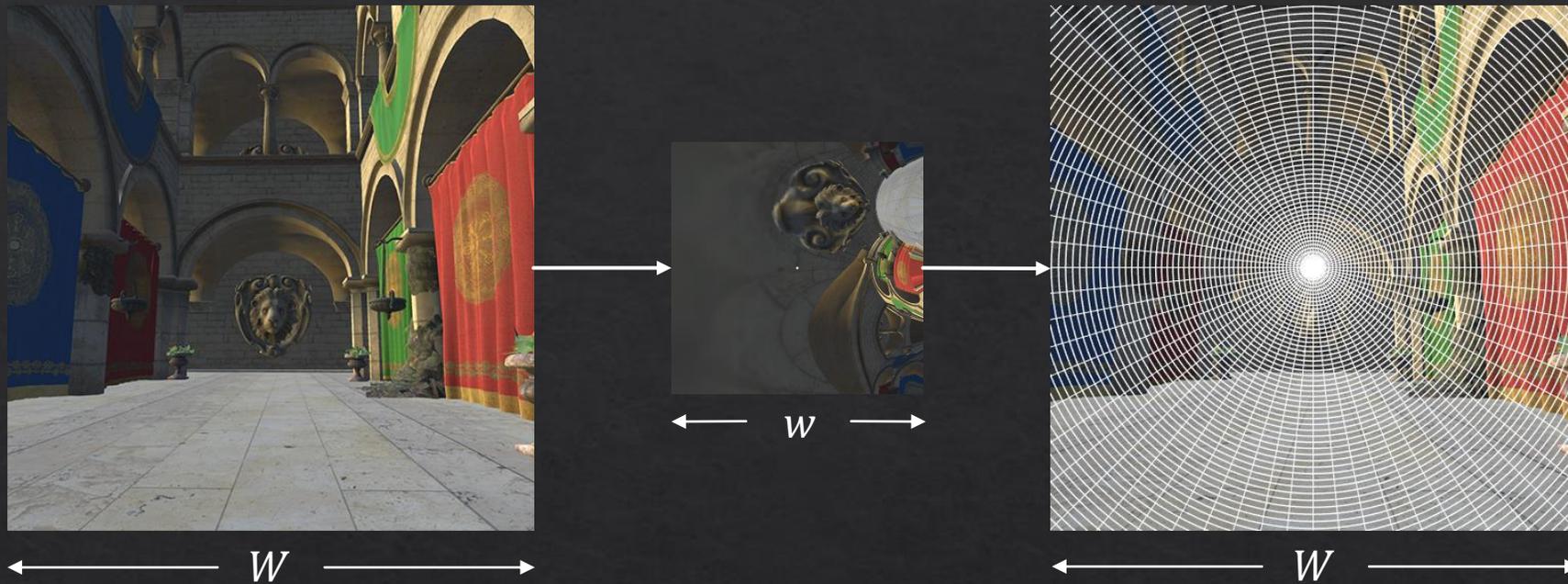
$$y = e^{L \cdot K \left(\frac{u}{w} \right)} \sin \left(v \cdot \frac{2\pi}{h} \right)$$

- W : screen width H : screen height w : buffer width h : buffer height
- $\mathbf{1}[y < 0] = \begin{cases} 1 & y < 0 \\ 0 & y > 0 \end{cases}$
- $L = \log \sqrt{W^2 + H^2}$
- $K(x) = \sum_{i=0}^{\infty} \beta_i x^i$, where $\sum_{i=0}^{\infty} \beta_i = 1$

Kernel Foveated Rendering



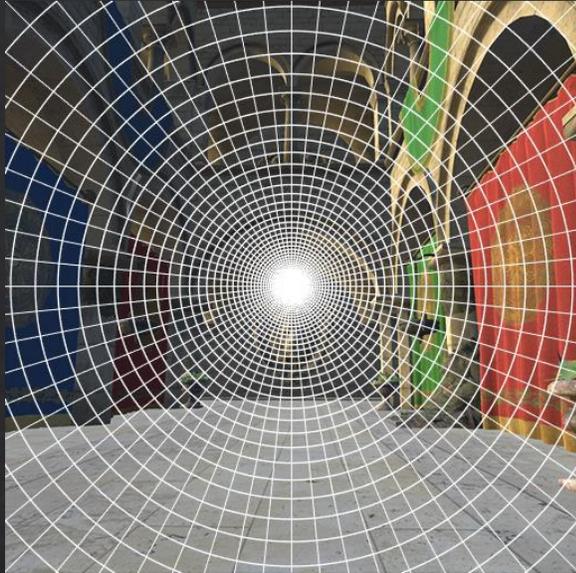
Distribution of pixels $\xrightarrow{\text{mimic}}$ *Distribution of photoreceptors in the human retina*



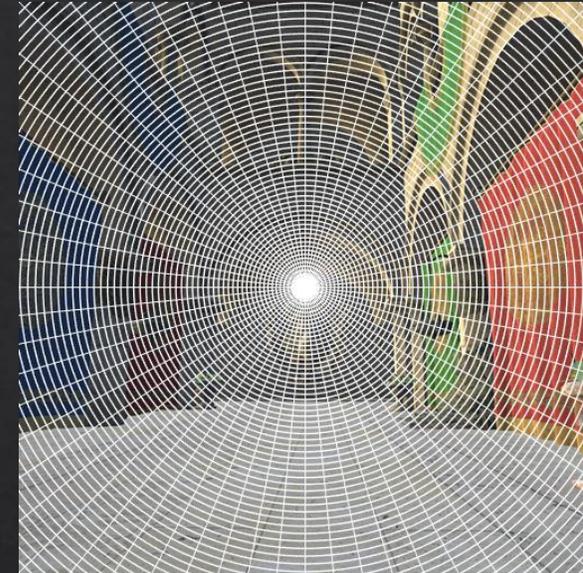
Kernel log-polar Mapping

- Define buffer parameter σ

$$\sigma = \frac{W}{w}$$



Result of log-polar
($K(x) = x$)



Result of kernel log-polar
($K(x) = x^4$)

Kernel log-polar Mapping

- Define buffer parameter σ

$$\sigma = \frac{W}{w}$$

- Define kernel function parameter α

$$K(x) = x^\alpha$$

Buffer parameter

$$\sigma$$

Original Frame



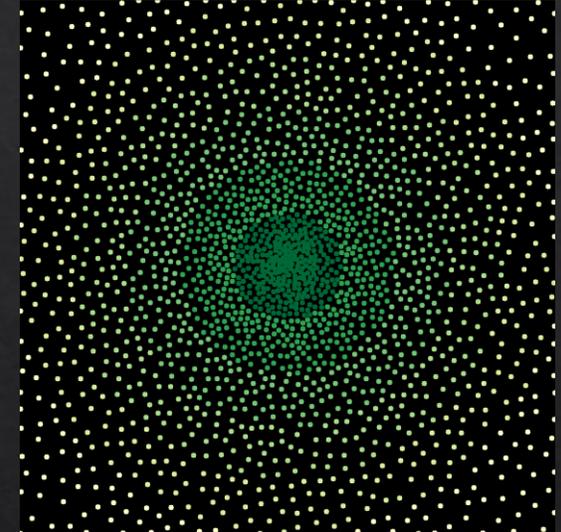
Buffer



Screen



Sample Map



$$\sigma = 1.2$$

Original Frame



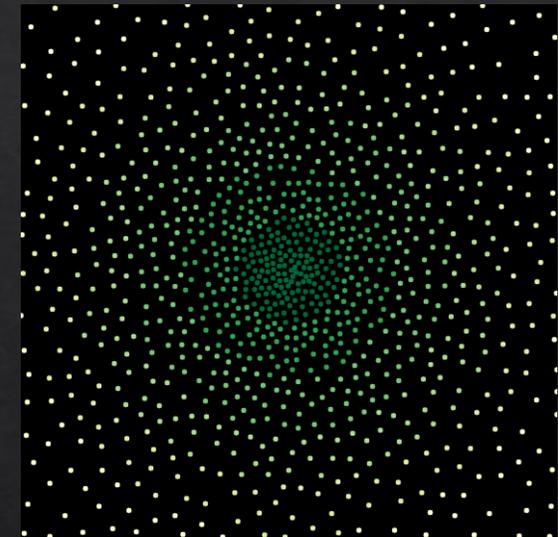
Buffer



Screen



Sample Map



$$\sigma = 1.7$$

Original Frame



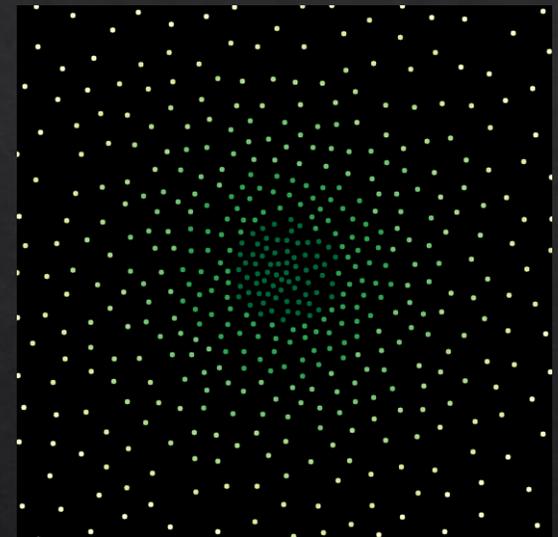
Buffer



Screen

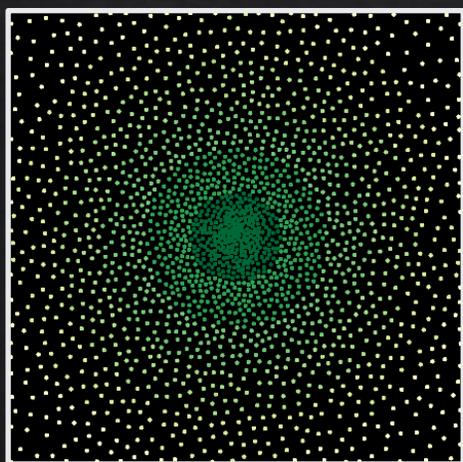
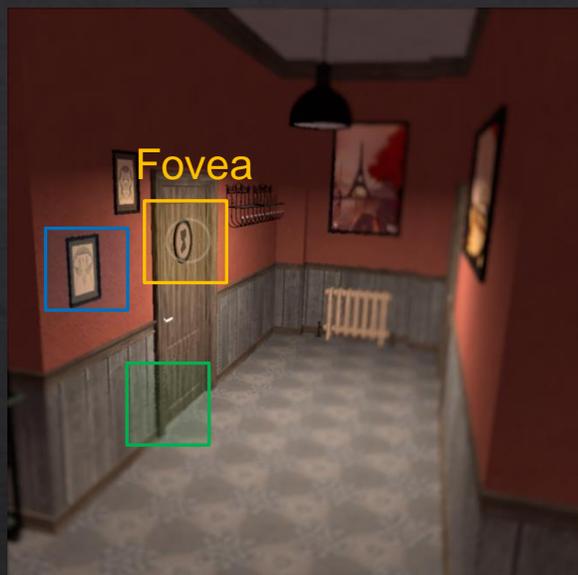


Sample Map

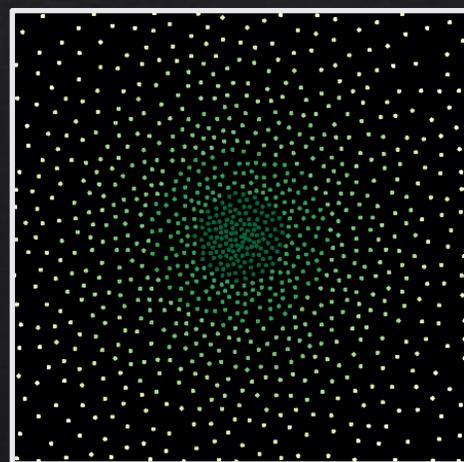


$$\sigma = 2.4$$

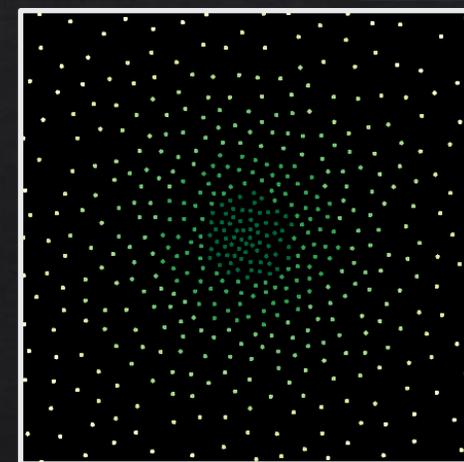
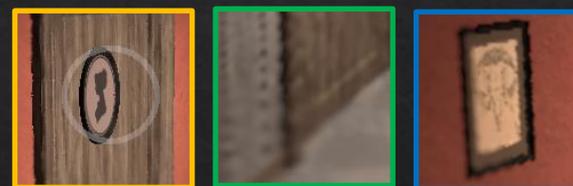
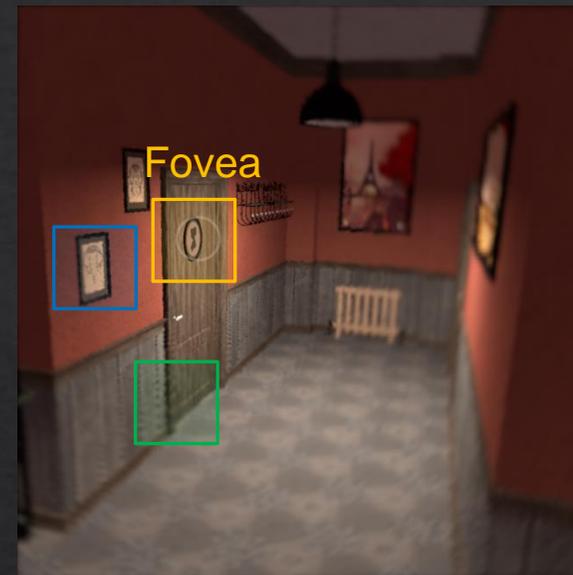
$\sigma = 1.2$



$\sigma = 1.7$



$\sigma = 2.4$



kernel function parameter
 α

Original Frame



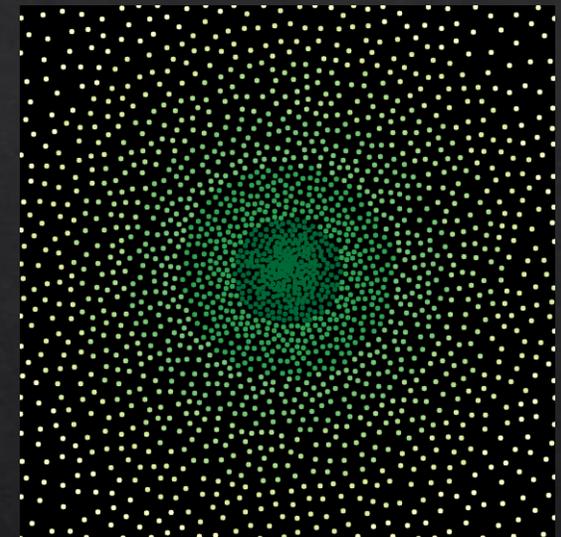
Buffer



Screen



Sample Map



$$\alpha = 1$$

Original Frame



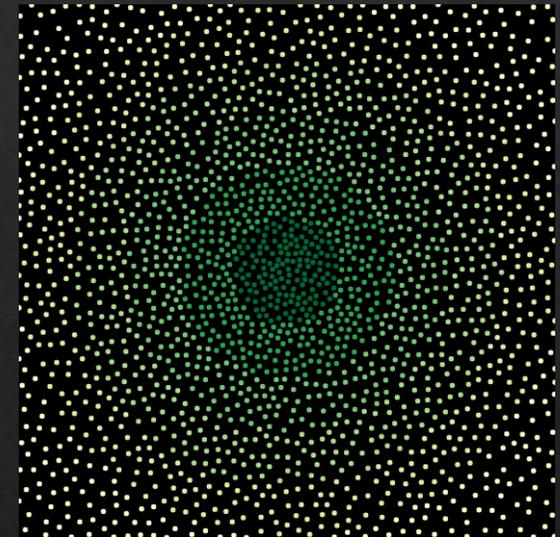
Buffer



Screen



Sample Map



$$\alpha = 4$$

Original Frame



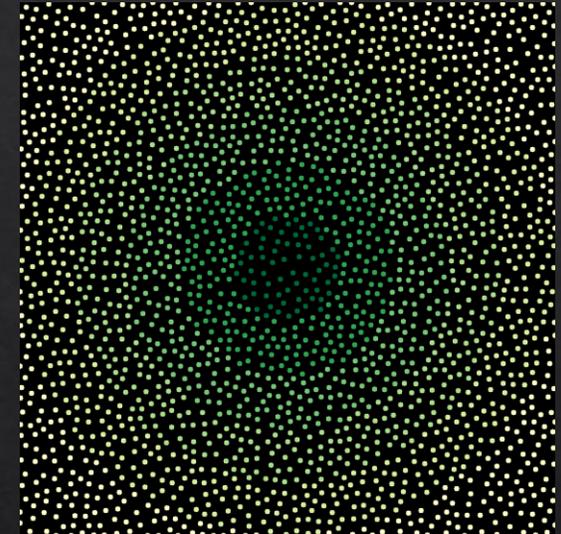
Buffer



Screen

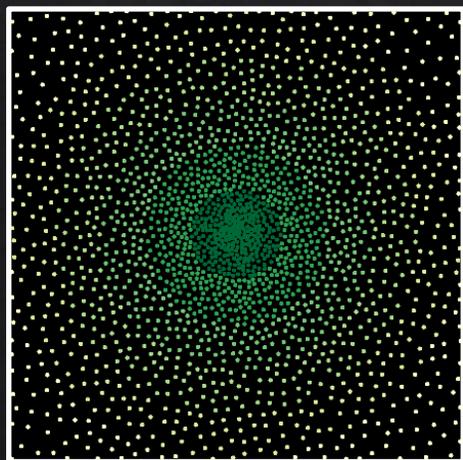
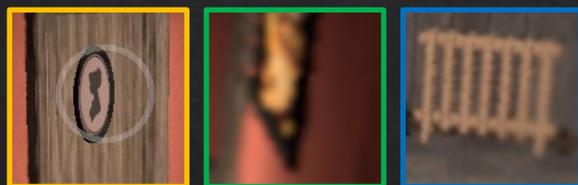


Sample Map

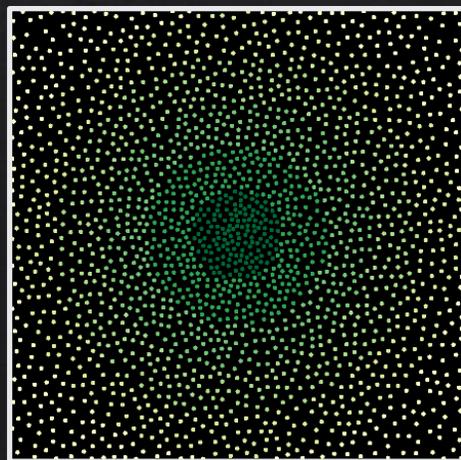


$$\alpha = 6$$

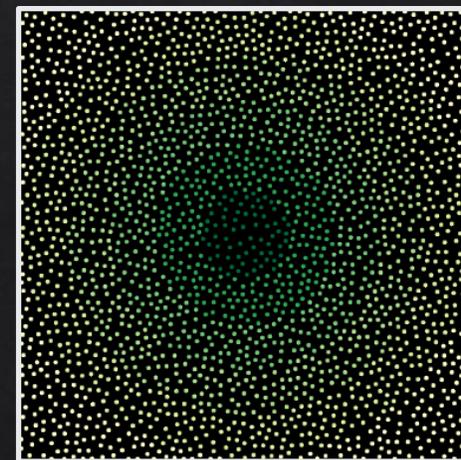
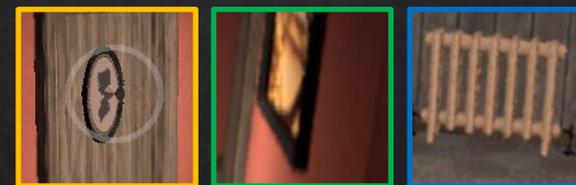
$\alpha = 1$



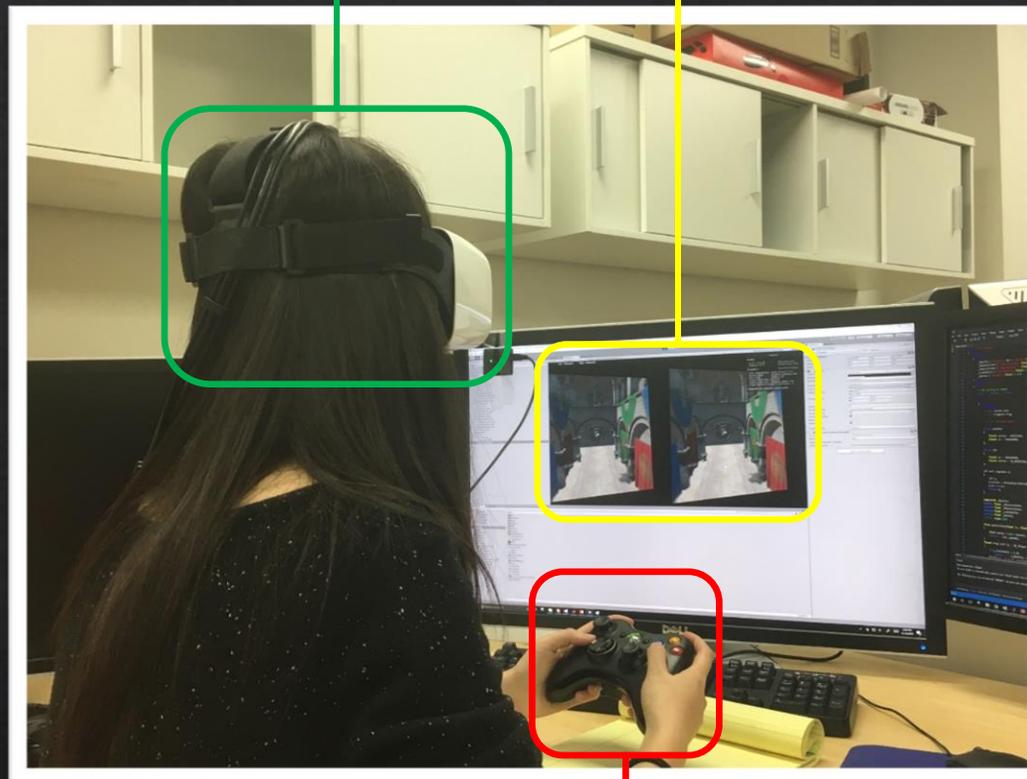
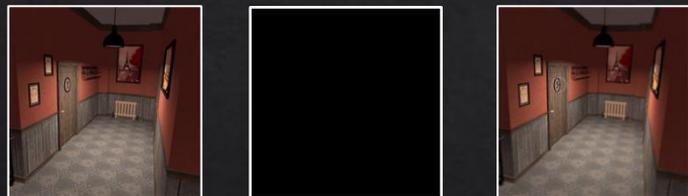
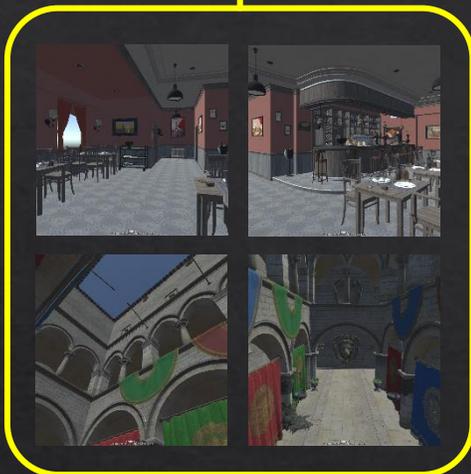
$\alpha = 4$



$\alpha = 6$



User Study

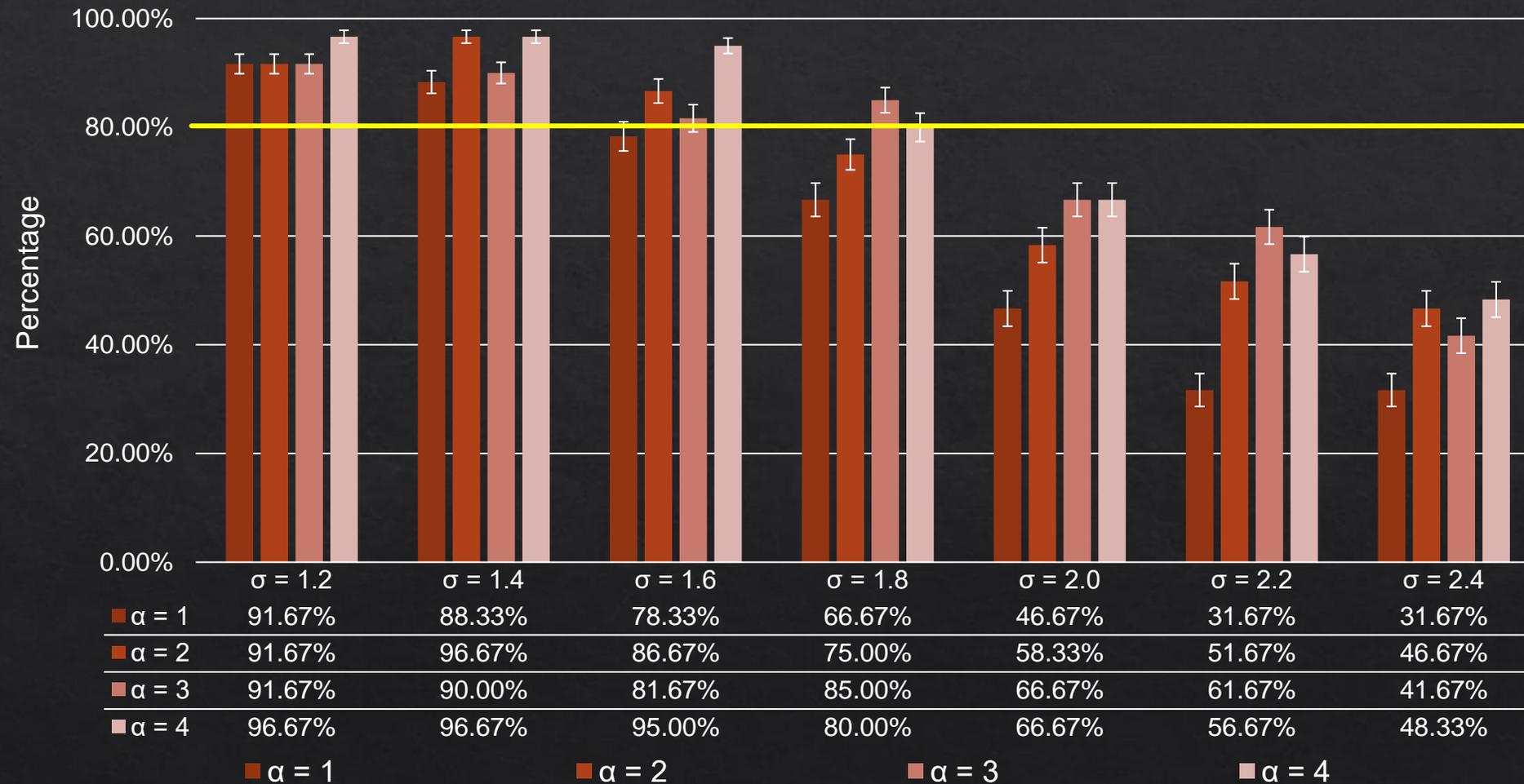


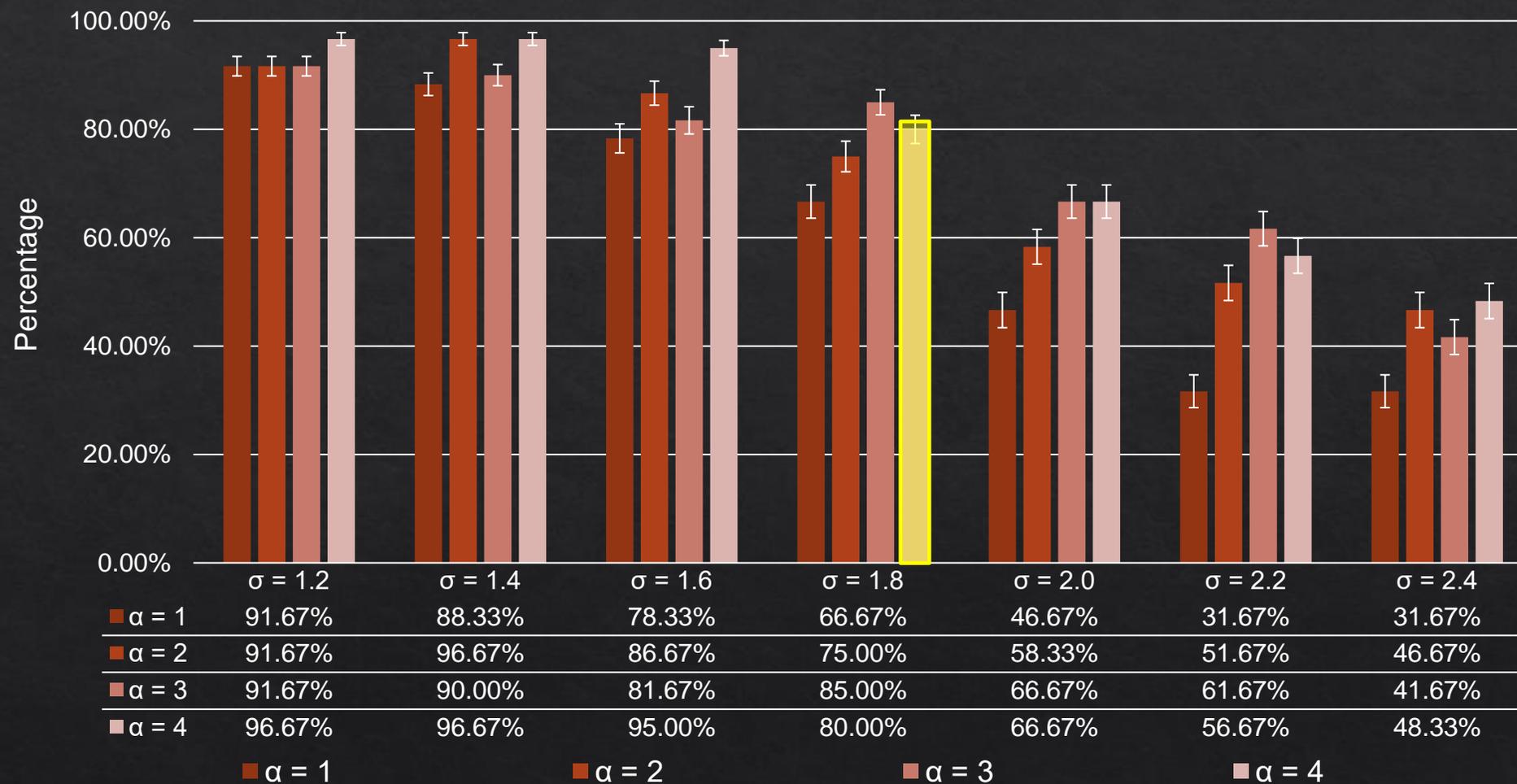
Resolution: 2560×1440
Field of view: up to 100 degrees



$\sigma \in [1.2, 2.4]$
 $\alpha \in [1, 4]$

step size: 0.2
step size: 1.0

Identical percentage under different α and σ 

Identical percentage under different α and σ 

G-buffer



World position

Bit tangent

Normal

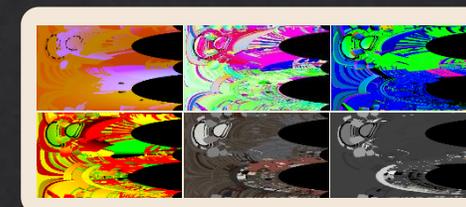


Texture coordinates

Albedo map

Roughness, ambient, and refraction maps

Kernel log-polar transformation



LP-buffer

($\sigma = 3.0$)

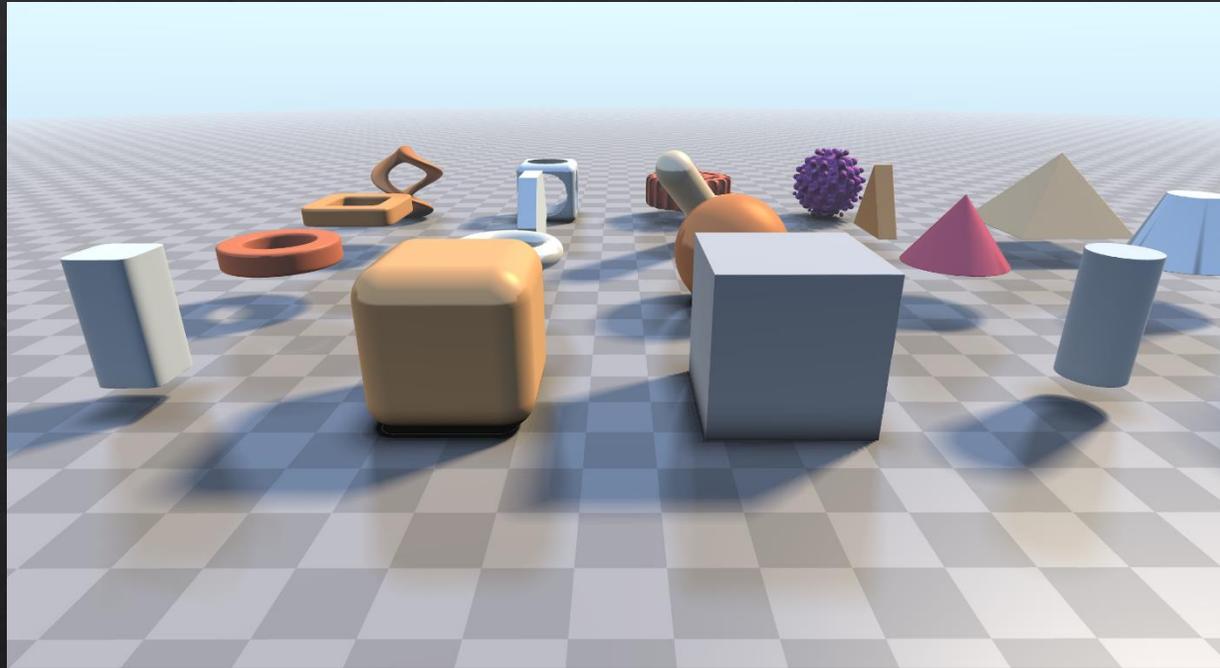
Shading & internal anti-aliasing



Inverse kernel log-polar transformation & post anti-aliasing



Screen



original ray-marching scene
10 FPS



foveated ray-marching scene ($\sigma = 1.8$, $\alpha = 4$)
30 FPS



original 3D geometries
31 FPS



foveated 3D geometries ($\sigma = 1.8$, $\alpha = 4$)
67 FPS

Scene	3D Textured Meshes			Ray Casting		
Resolution	Ground Truth	Foveated	Speed up	Ground Truth	Foveated	Speed up
1920 × 1080	55 FPS	110 FPS	2.0X	20 FPS	57 FPS	2.9X
2560 × 1440	31 FPS	67 FPS	2.2X	10 FPS	30 FPS	3.0X
3840 × 2160	8 FPS	23 FPS	2.8X	5 FPS	16 FPS	3.2X

Summary

- Kernel log-polar transformation for 3D graphics
 - Deferred Shading
 - Parameterize with kernel parameter α and buffer parameter σ
- User study
 - Determine parameters to maximize perceptual realism and minimize computation
- Experiment
 - 2.8X – 3.2X speedup



Ground Truth



Kernel Foveated Rendering

Thanks!



video



paper

Kernel Foveated Rendering

Xiaoxu Meng, Ruofei Du, Matthias Zwicker and Amitabh Varshney

Augmentarium | UMIACS

University of Maryland, College Park

ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games 2018

FOVE Headset



- ◇ DISPLAY
 - ◇ WQHD OLED (2560 X 1440)
 - ◇ Frame rate: 70fps
 - ◇ Field of view: Up to 100 degrees
- ◇ EYE TRACKING SENSORS
 - ◇ Infrared eye tracking system x 2
 - ◇ Tracking accuracy: less than 1 degree
 - ◇ Frame rate: 120fps

User Study: Significance

σ^2 –value	1.2	1.4	1.6	1.8	2.0	2.2	2.4
Cochran's Q value	1.72	5.76	8.20	8.25	7.49	14.27	5.48
p-value	0.631	0.122	0.042	0.041	0.058	0.002	0.139

Two-level Anti-aliasing

G-buffer



World position

Bit tangent

Normal

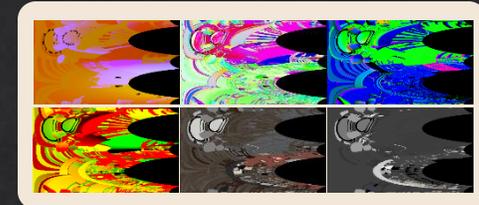


Texture coordinates

Albedo map

Roughness, ambient, and refraction maps

Kernel log-polar transformation



LP-buffer
($\sigma = 3.0$)

Shading & internal anti-aliasing

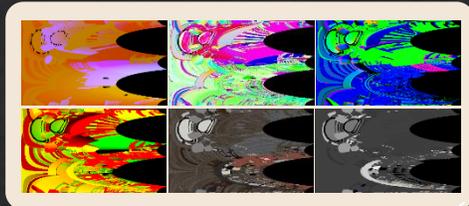


Inverse kernel log-polar transformation & post anti-aliasing



Screen

Two-level Anti-aliasing



*Shading &
internal anti-aliasing*



*Inverse kernel
log-polar transformation
& post anti-aliasing*



Non-uniform Gaussian Blur

Kernel size increase from left
(fovea) to right (periphery)



Non-uniform Gaussian Blur

Kernel size increase from
fovea to periphery

Video & Paper



video



paper