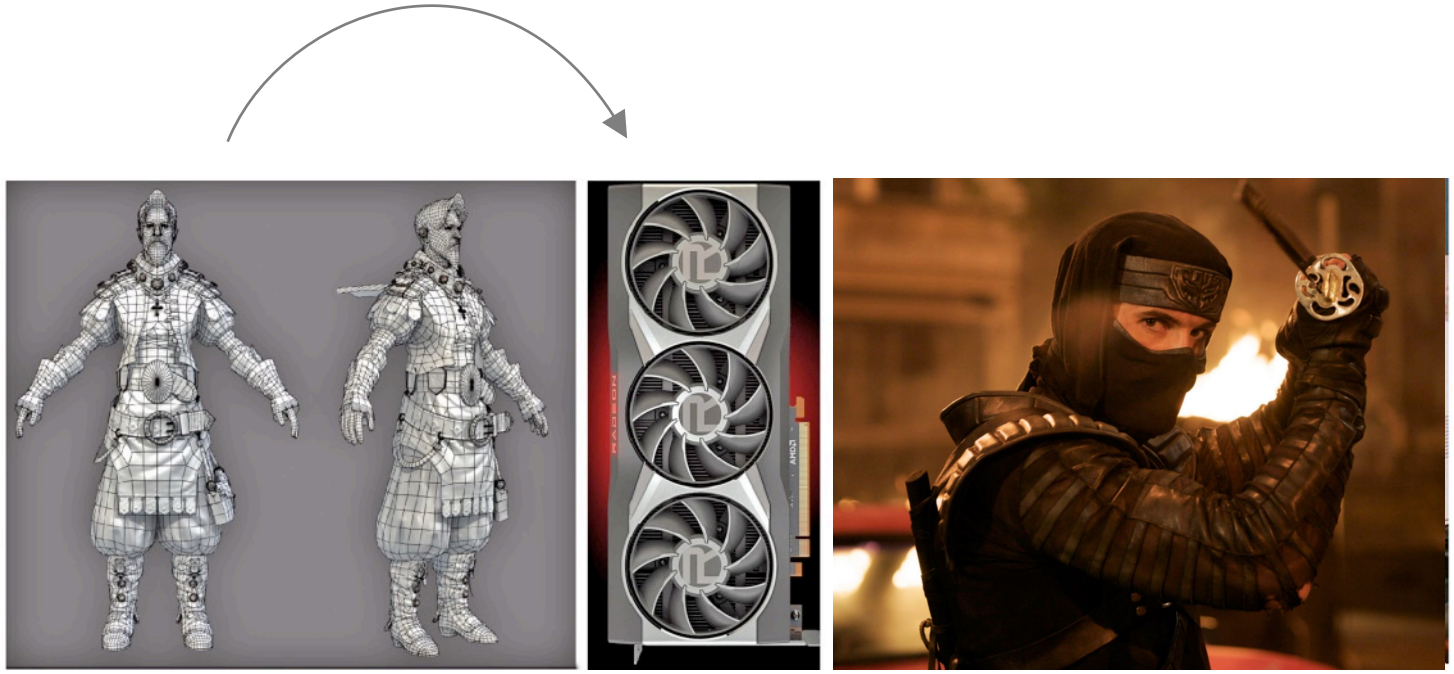
Perceptually-inspired VR Image Synthesis

Tobias Ritschel

University College London

CVMP 2021, 6/7th of December 2021, London / UK

$$\arg\min_{\mathcal{R}} |\mathcal{R}(C) - I|_2$$



Content C

Rendering $\mathcal{R}$

Reference I

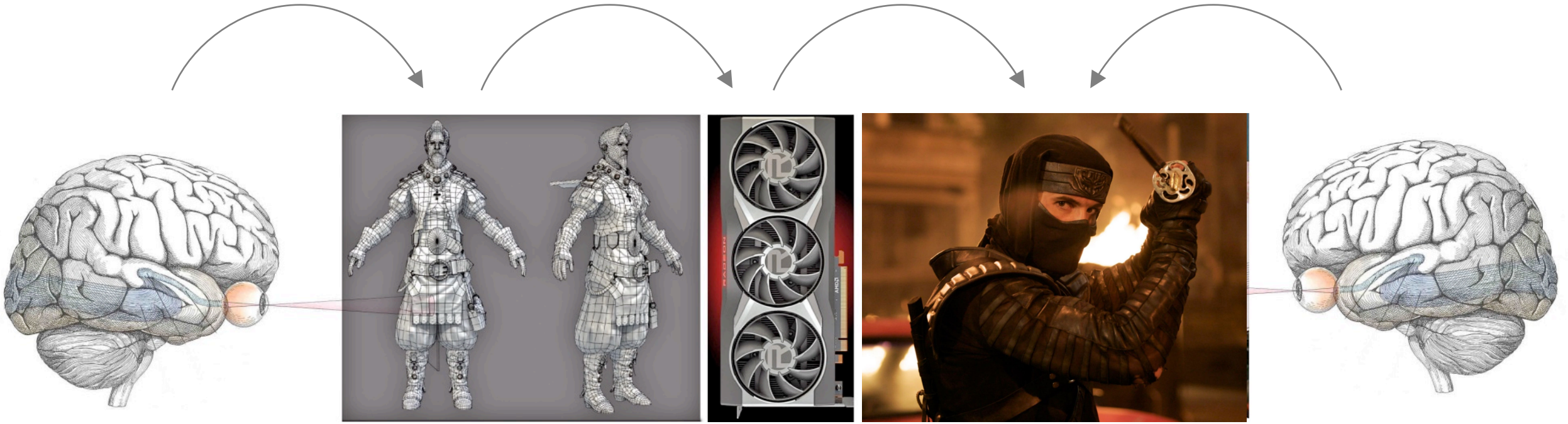
Human $\arg\min$

```
54 def intersectSphere(rayPositions, rayDirections, position, radius, tMin, tMax):
55   q = position - rayPositions
56   vDotQ = jnp.sum(rayDirections * q, axis = -1)
57   squareDiffs = jnp.sum(q * q, axis = -1) - radius * radius
58   discrim = vDotQ * vDotQ - squareDiffs
59
60   t0 = jnp.where(discrim > 0, vDotQ - jnp.sqrt(discrim), float('nan'))
61   t1 = jnp.where(discrim > 0, vDotQ + jnp.sqrt(discrim), float('nan'))
62   t0 = jnp.where(t0 < tMin, float('nan'), t0)
63   t0 = jnp.where(t0 > tMax, float('nan'), t0)
64   t1 = jnp.where(t1 < tMin, float('nan'), t1)
65   t1 = jnp.where(t1 > tMax, float('nan'), t1)
66
```

Machine $\arg\min$

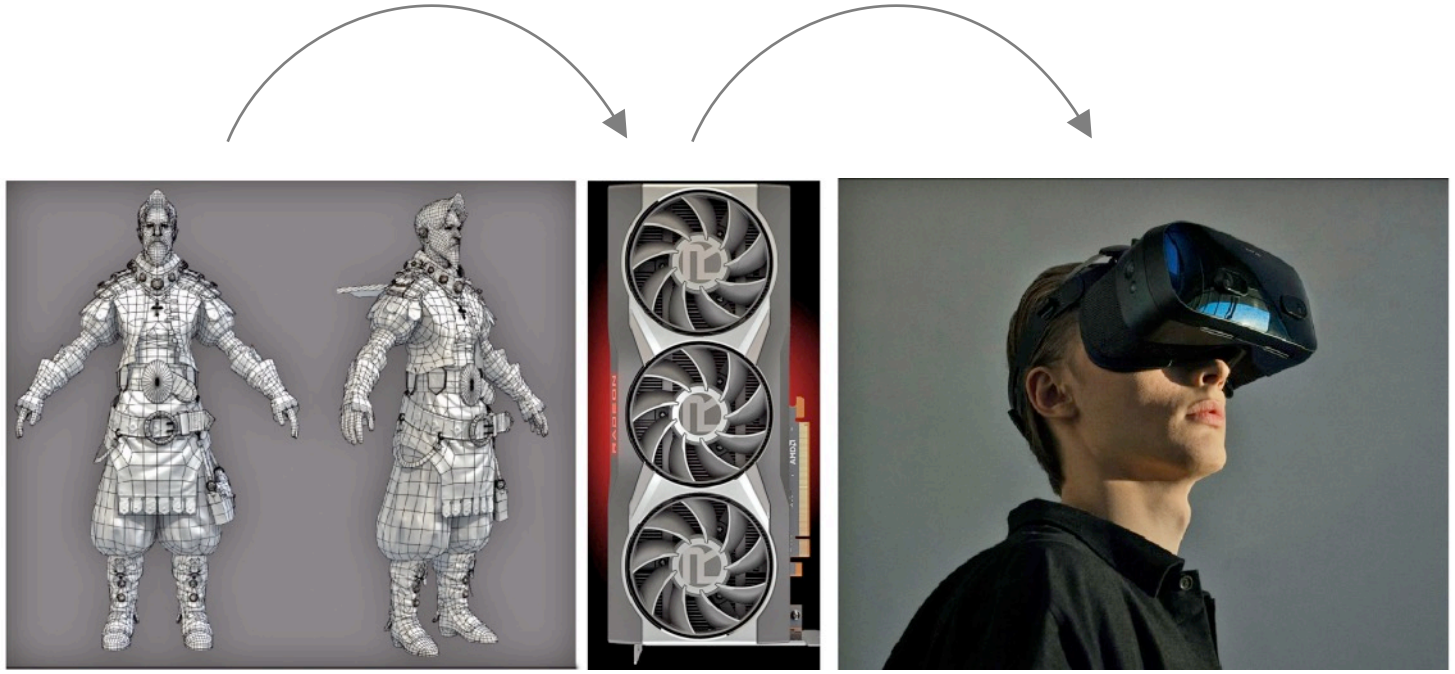
```
55 @jit
56 def update(params, x, y, opt_state):
57     value, grads = value_and_grad(loss)(params, x, y)
58     opt_state = opt_update(0, grads, opt_state)
59     return get_params(opt_state), opt_state, value
60
```

$$\arg\min_{\mathcal{R}} |\mathcal{P}(\mathcal{R}(C)) - \mathcal{P}(I)|_2$$



Perception	Content	Rendering	Reference	Perception
$\mathcal{P}$	C	$\mathcal{R}$	I	$\mathcal{P}$

$$\arg\min_{\mathcal{R}} |\mathcal{D}(\mathcal{R}(C)) - I|_2$$

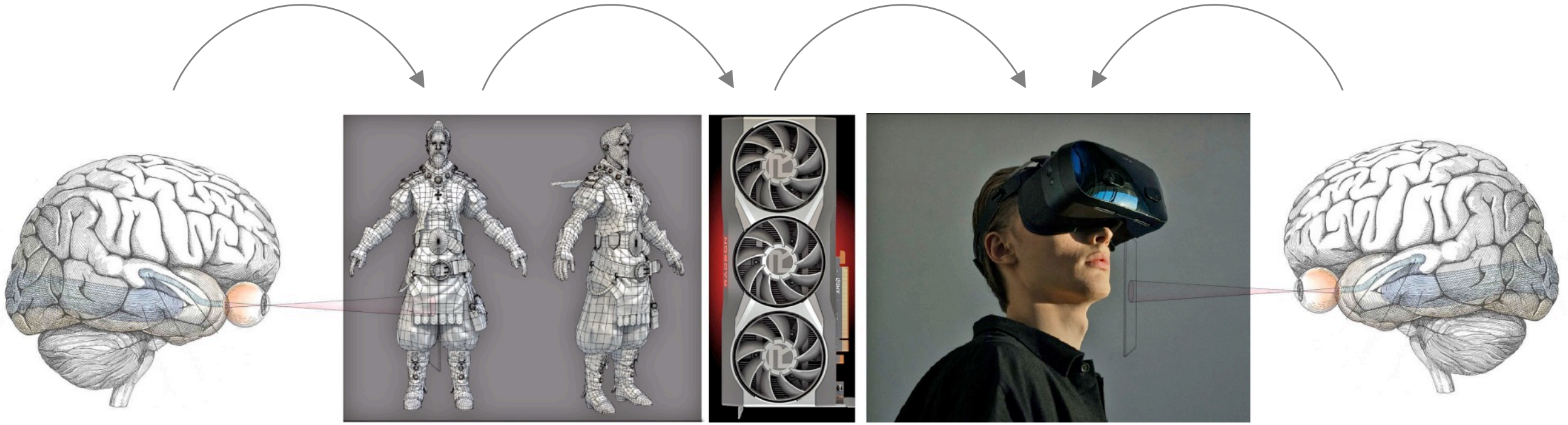


Content C

Rendering $\mathcal{R}$

Display $\mathcal{D}$

$$\arg\min_{\mathcal{R}} |\mathcal{P}(\mathcal{D}(\mathcal{R}(C))) - \mathcal{P}(I)|_2$$

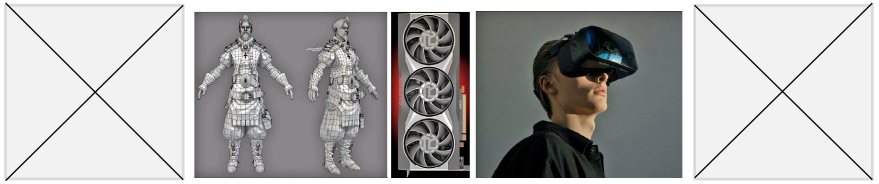


Perception	Content	Rendering	Display	Perception
$\mathcal{P}$	C	$\mathcal{R}$	$\mathcal{D}$	$\mathcal{P}$

1

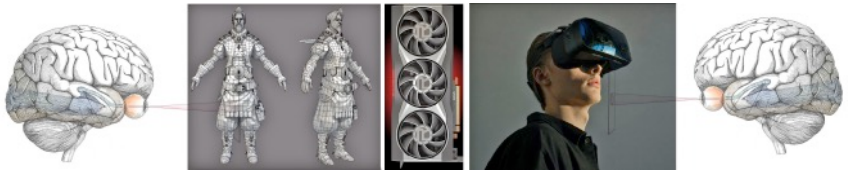
Laplacian Kernel Splatting for Efficient Depth-of-field and Motion Blur Synthesis or Reconstruction
T Leimkühler, H-P Seidel, T Ritschel
ACM Trans. Graph. (Proc. SIGGRAPH 2018) 37(3)

2



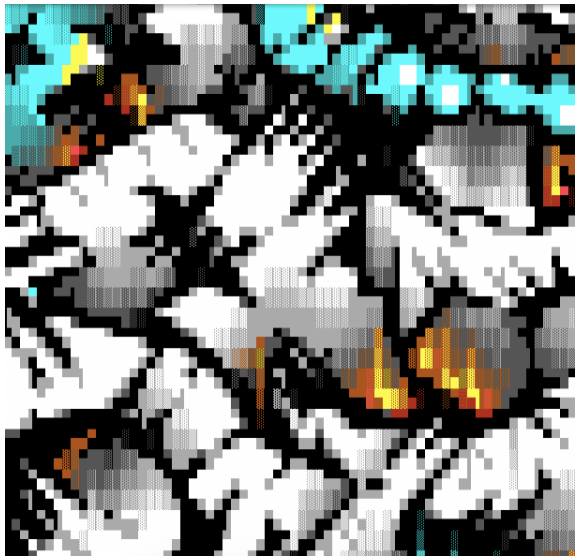
Perceptual Rasterization for Head-mounted Display Image Synthesis
S Friston, T Ritschel, A Steed
ACM Trans. Graph. (Proc. SIGGRAPH 2019) 38(3)

3



Beyond Blur: Real-time Metamers for Foveated Rendering
D Walton, R Kuffner-dos Anjos, S Friston, D Swapp, A Steed, T Ritschel
ACM Trans Graph (Proc. SIGGRAPH 2021) 40(3)

Pixels and me



Early 90ies

BSS / ANSI Art

Black Maiden, Fire, Mean Scheme
(US, Germany)

Late 90ies

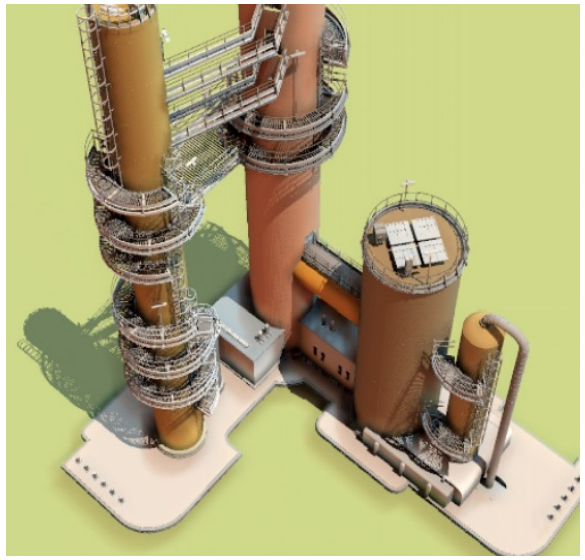
Demoscene

Kolor
(Germany)

Early 2000s

Production tools

Mudbox, Autodesk
(US)



Since 2005

Research

Max Planck, CNRS, UCL
(Germany, France, UK)

1

Laplacian Kernel Splatting for Efficient Depth-of-field and Motion Blur Synthesis or Reconstruction
T Leimkühler, H-P Seidel, T Ritschel
ACM Trans. Graph. (Proc. SIGGRAPH 2018) 37(3)

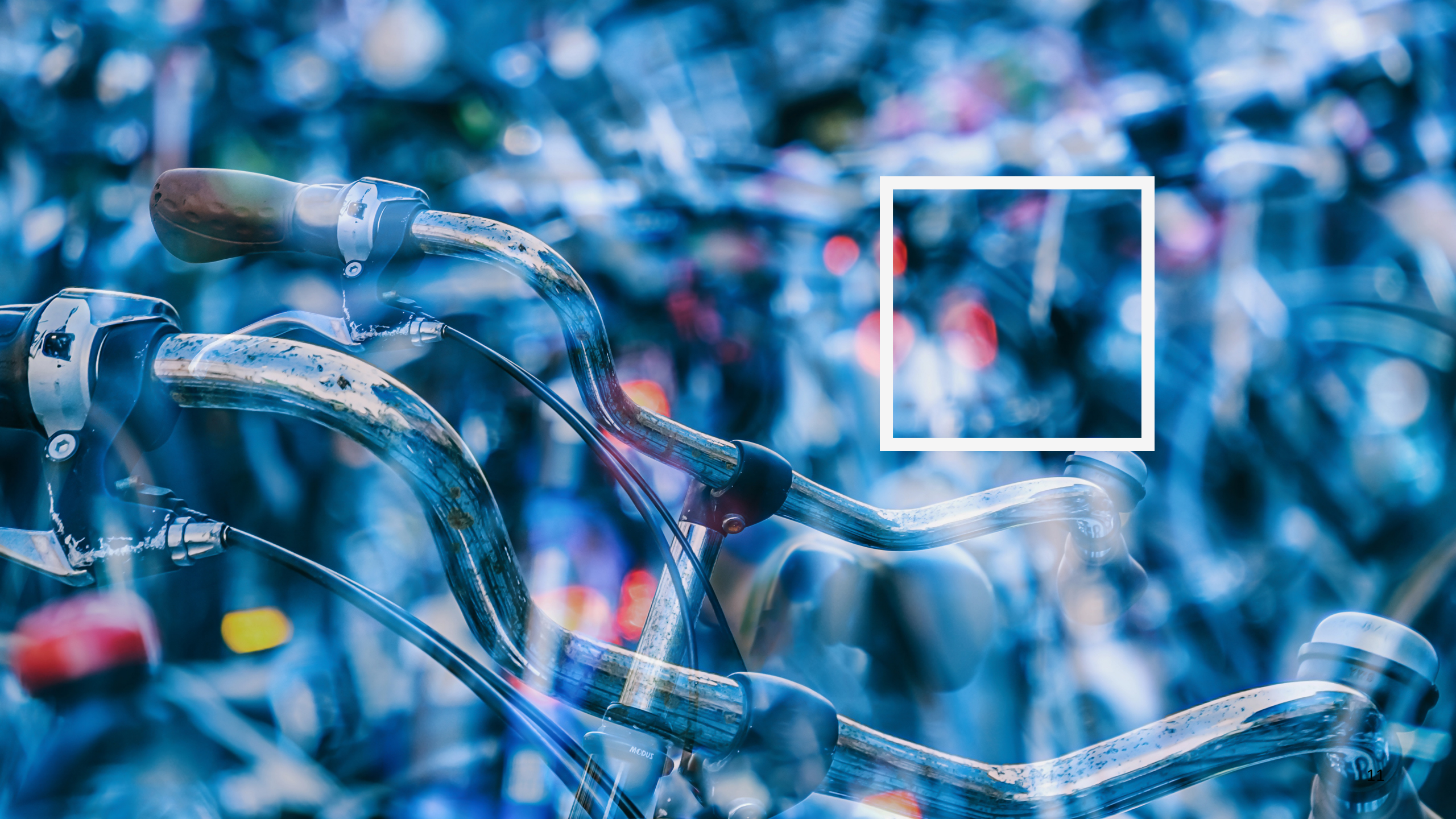
模拟景深和运动模糊的组合是影响合成图像电影质量的一个重要因素，但计算可能需要很长时间。绘制每个像素的点扩散函数 (PSF) 是一种通用方法，可以提供高质量，但需要大量的计算时间。我们分两步加速：在预处理中，我们优化所有可能的 PSF 的拉普拉斯稀疏表示，我们称之为 Spreadlet。在运行时，spreadlet 可以有效地展开为图像的拉普拉斯算子。积分该图像会产生最终结果。我们的方法忠实地扩展到强烈的运动和大的失焦区域，并且在速度和质量上与离线和交互式方法相比具有优势。它既适用于从针孔合成，也适用于从随机图像重建，无论是否分层。

2

Perceptual Rasterization for Head-mounted Displa
S Friston, T Ritschel, A Steed
ACM Trans. Graph. (Proc. SIGGRAPH 2019) 38(3)

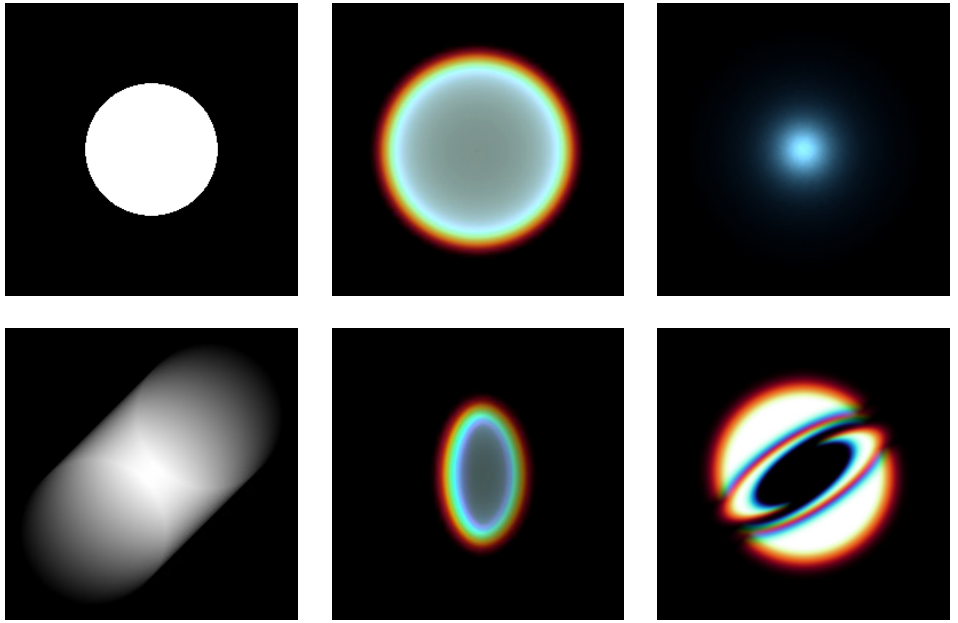
3

Beyond Blur: Real-time Metamers for Foveated Rendering
D Walton, R Kuffner-dos Anjos, S Friston, D Swapp, A Steed, T Ritschel
ACM Trans Graph (Proc. SIGGRAPH 2021) 40(3)



Goals

- General point spread functions
  - Arbitrary combinations of different effects
  - Physical models
- Interactive performance



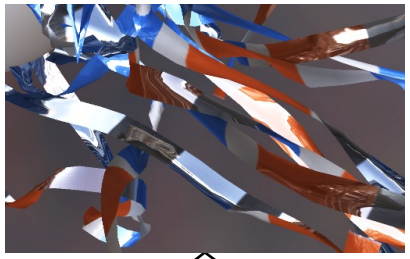
Distribution effects in Computer graphics

Stochastic (Raytracer)

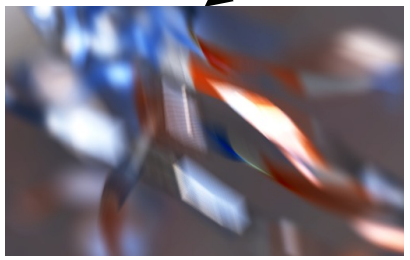
Convergence

Reconstruction

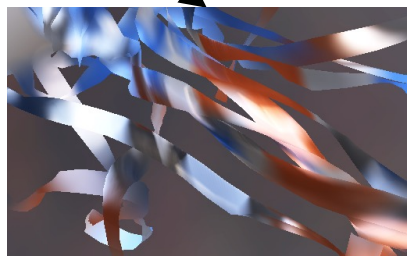
Pinhole (OpenGL)



Splatting



Filtering



Slow

Fast

Splatting vs. Filtering

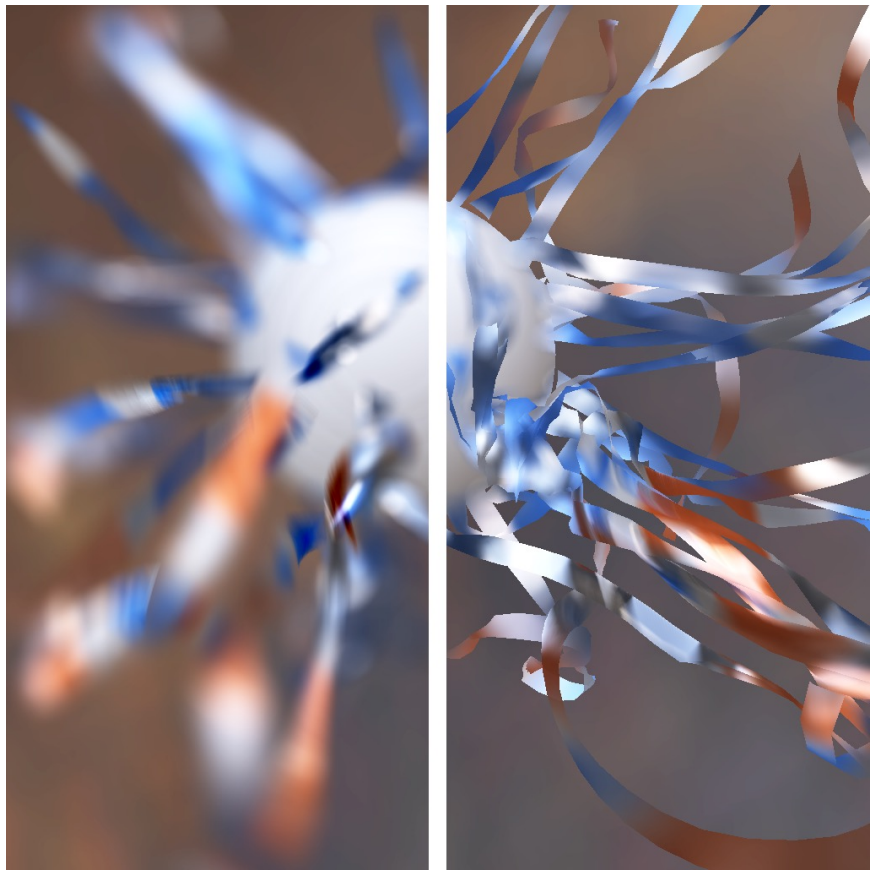
Splatting

Quality:

Accurate and versatile

Speed:

Scattering



Filtering

Quality:

Crude approximation, Problems at silhouettes

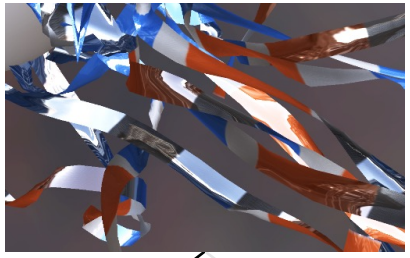
Speed:

Gathering

Distribution effects in Computer graphics

Stochastic

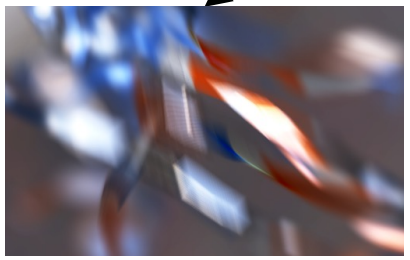
Pinhole



Convergence

Reconstruction

Splatting



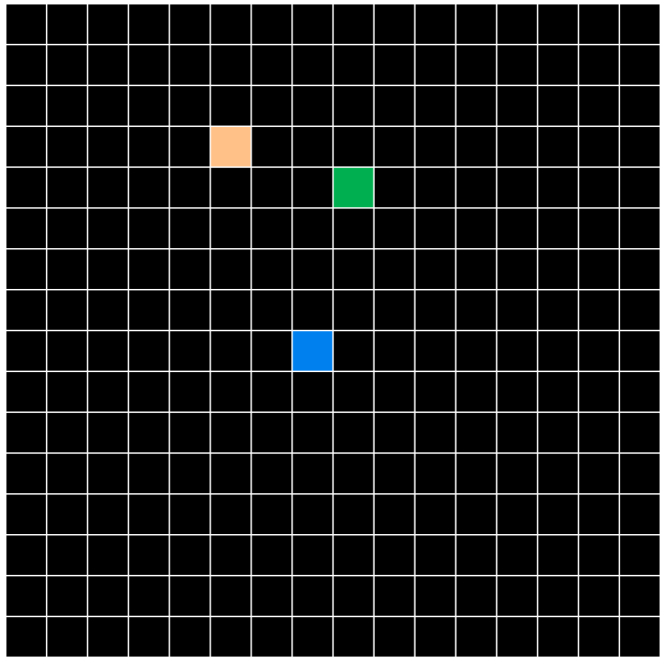
Filtering

Slow

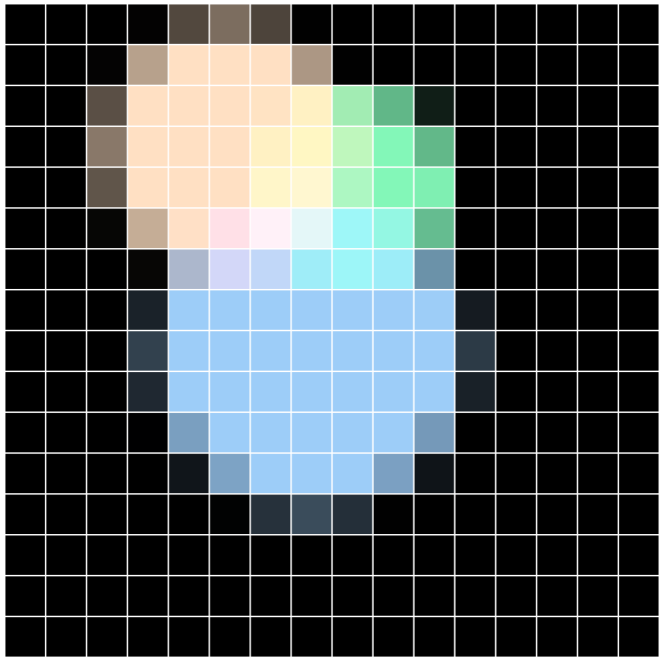
Fast

Kernel Splatting Problem

- Kernel of each input pixel can affect a large image area!

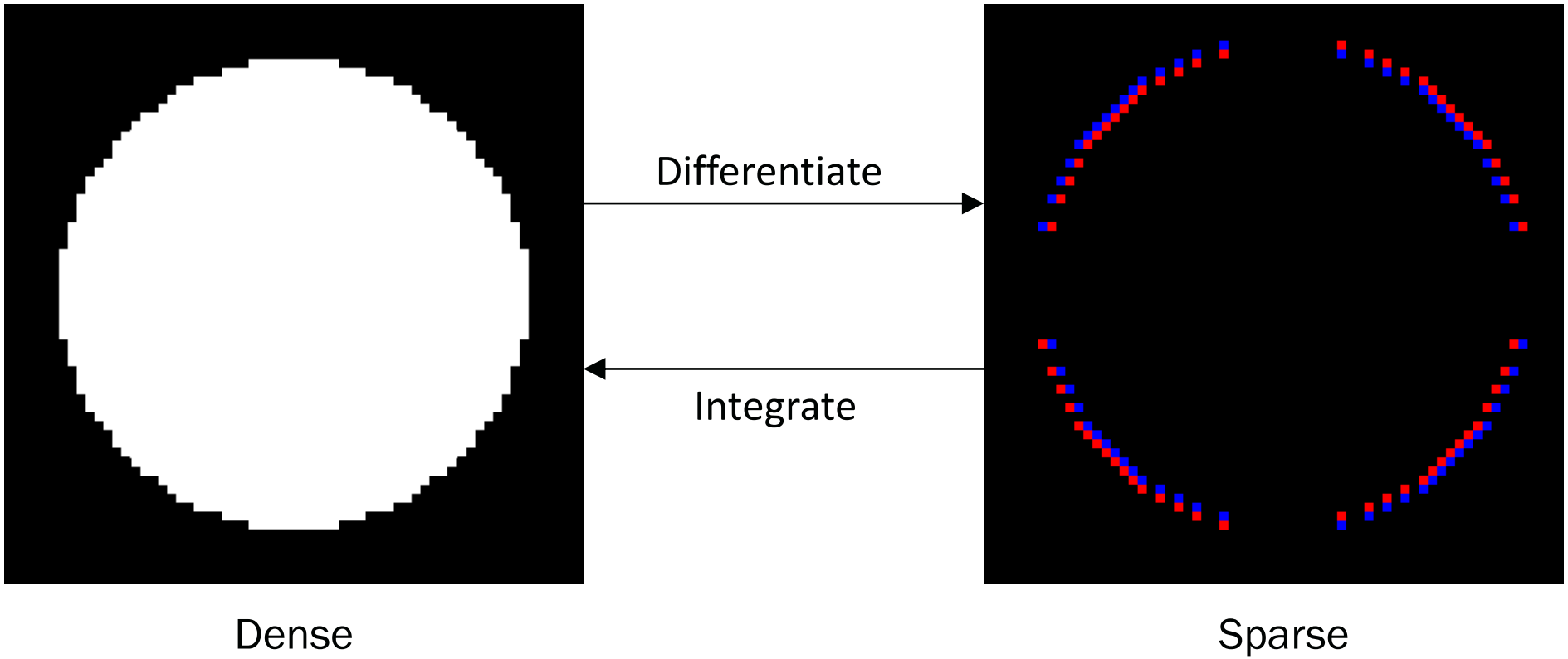


Pinhole Input



Splatting Output

Differential Kernel, Solution



Differential rasterization

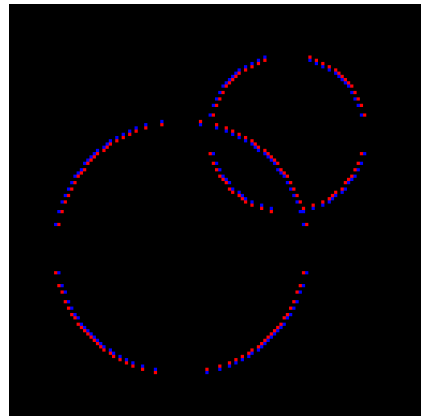
Additive Splatting

Image

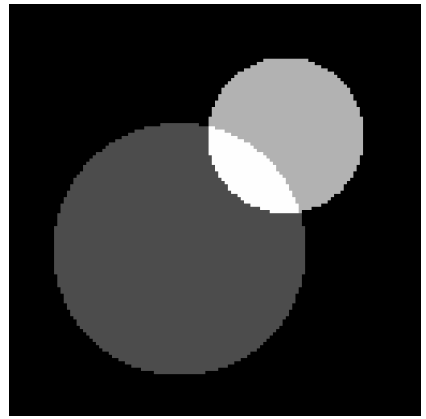
$$\text{splat}(\boldsymbol{f}, \boldsymbol{I})$$

Spatially-varying Kernel

1. Splat sparse differential kernels



2. Integrate image

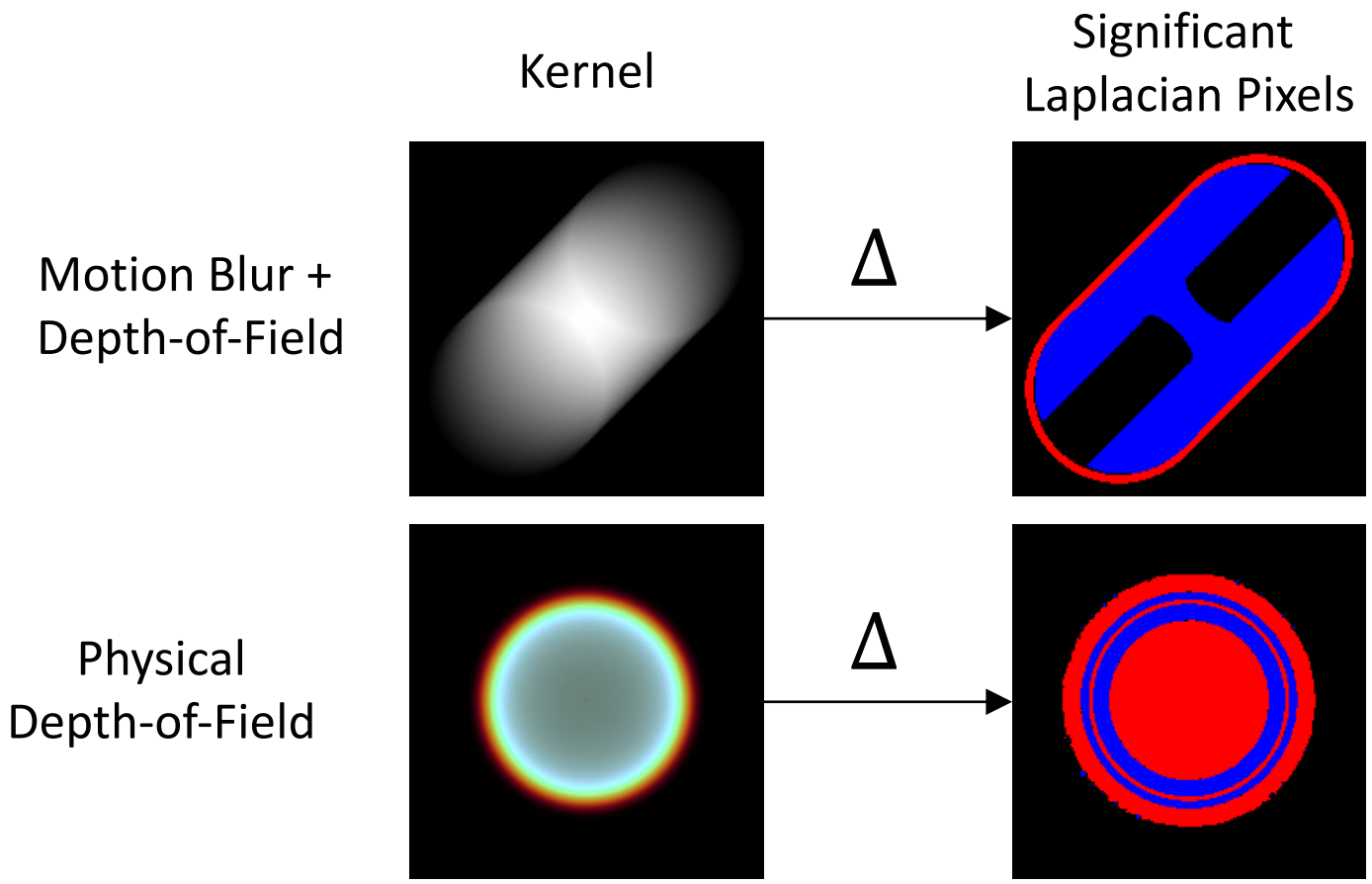


Which Differential?

- Differential operator of choice: Laplacian
- Why?!

$$\Delta f(x,y) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

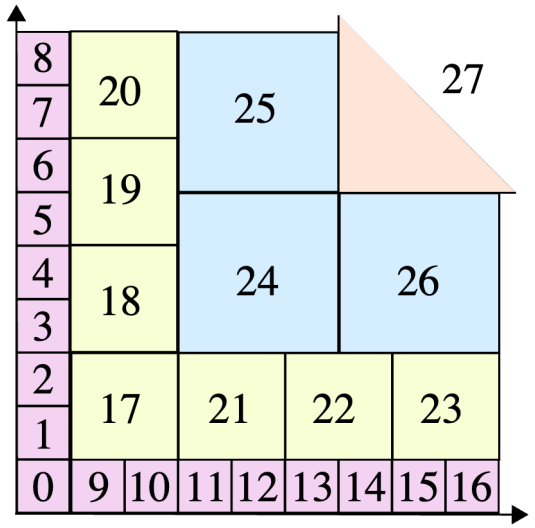
Most Kernels have dense Laplacians



Our solution:

Precompute a sparse approximation of the Laplacian

Pre-computation Overview

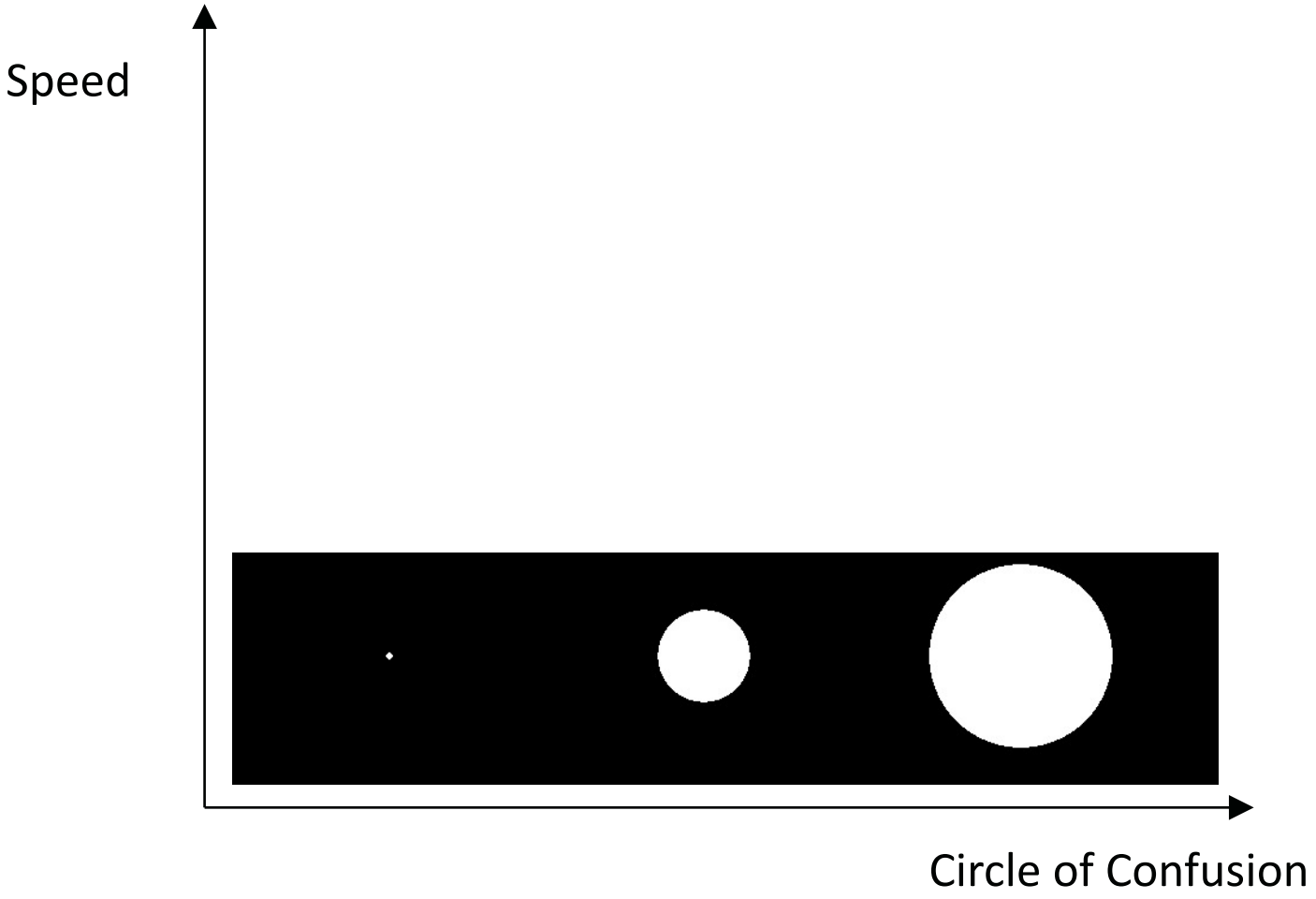


1. Parametrization & Sampling

2. Kernel Generation

3. Sparsification

Parametrization



- Angular sampling not required!

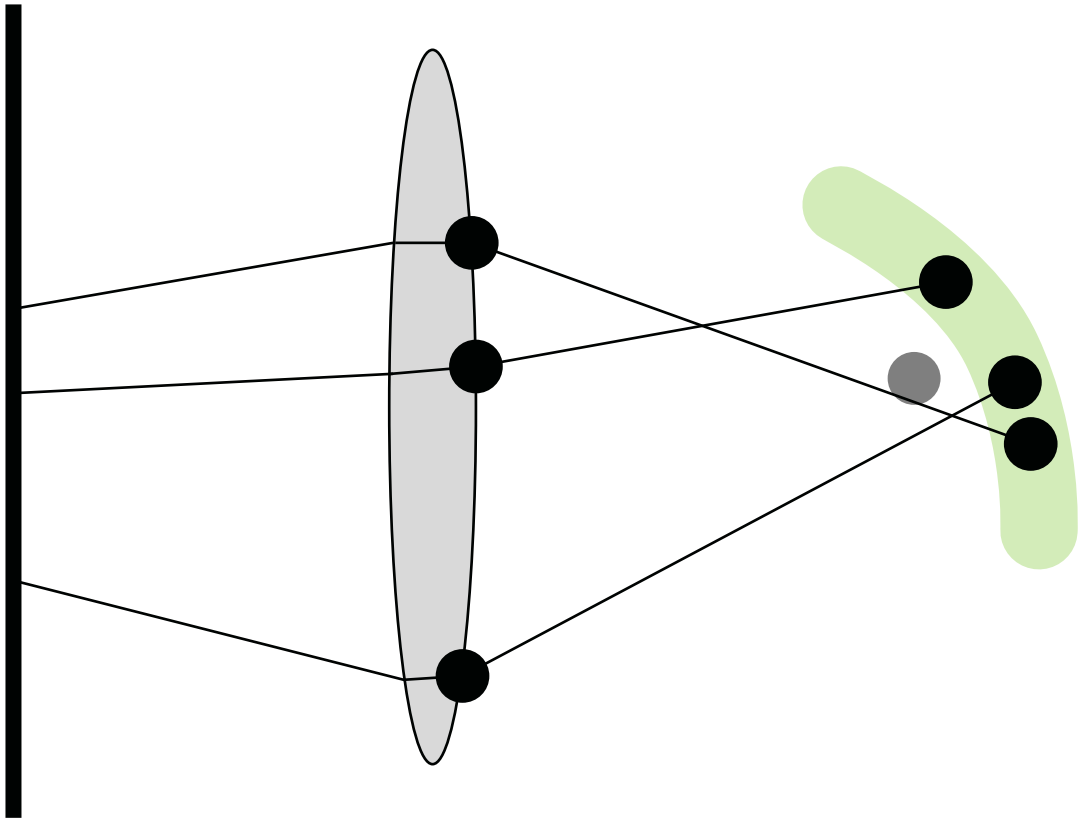
Pre-computation Overview

1. Parametrization & Sampling

2. Kernel Generation

3. Sparsification

Kernel generation



- Select kernel parameters
  - Depth
  - Eccentricity
  - Motion
- Sample integrand: Light tracing
  - Time
  - Lens

Pre-computation Overview

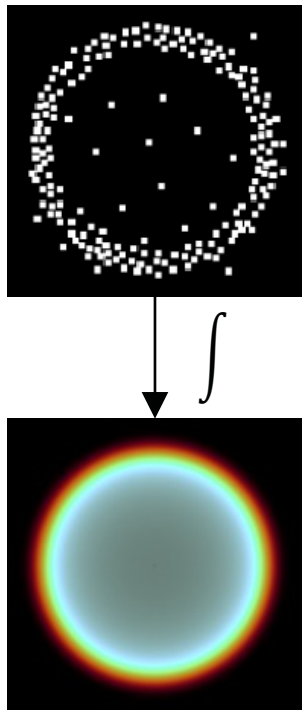
1. Parametrization & Sampling

2. Kernel Generation

3. Sparsification

Sparsification

- Objective: Find a sparse set of points that yields the desired kernel after integration



$$\min \int_{\Omega} \left\| f(x) - \int_{\Omega} \sum_{i=0}^{n} \mathrm{splat}(\boldsymbol{p}_i, \boldsymbol{\Delta \bar{f}}_i)\, dx \right\| dx$$

Kernel

Reconstruction

n : Number of Points

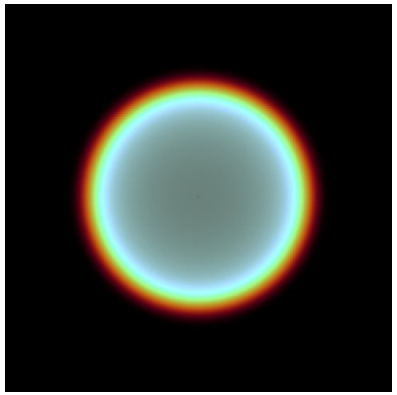
$\boldsymbol{p}$: Point Positions

$\boldsymbol{\Delta \bar{f}}$: Point Values

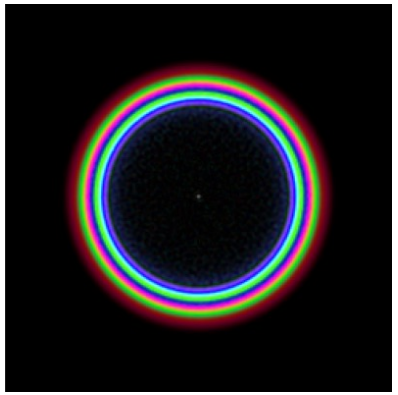
Sparsification 稀疏化

Determines
Number of Points

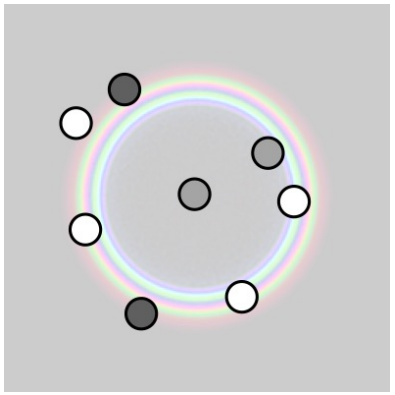
Determines
Point Positions and Values



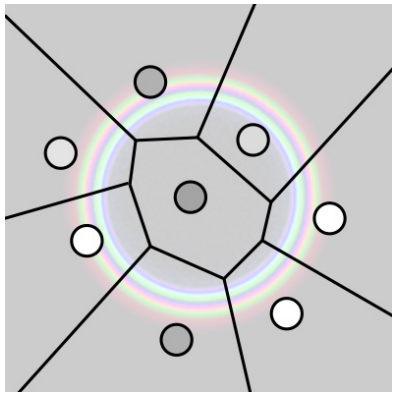
Kernel



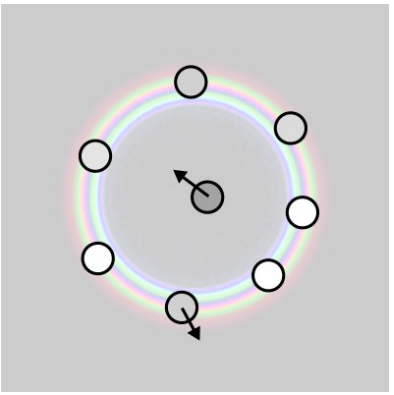
Laplacian
Image Filter



Dart
Throwing

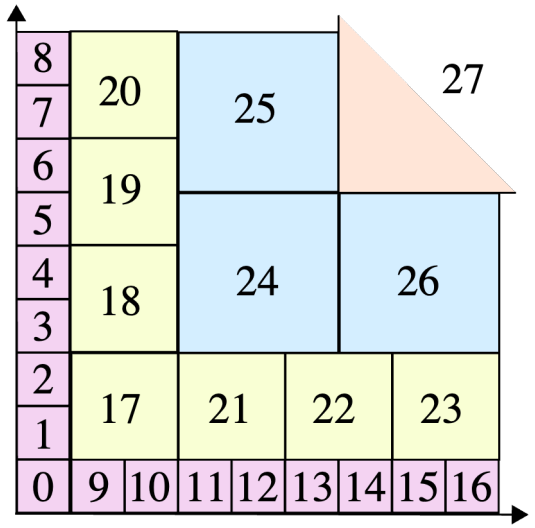


Lloyd
Relaxation

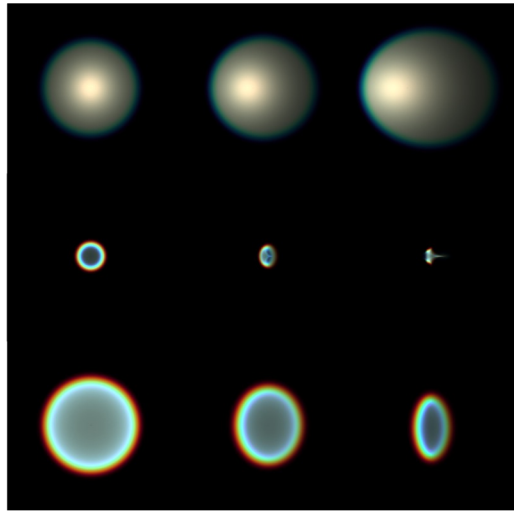


Simulated
Annealing

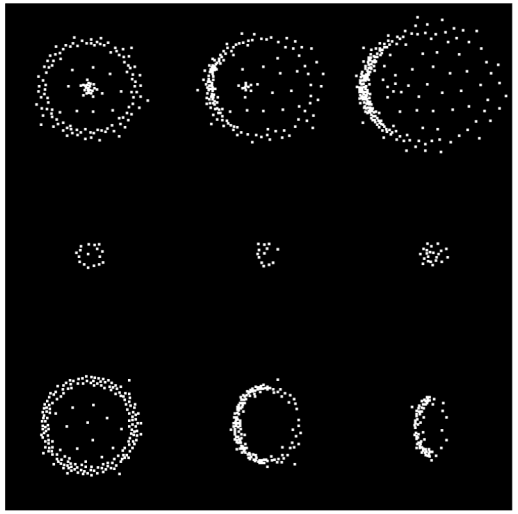
Pre-computation Overview



1. Parametrization & Sampling

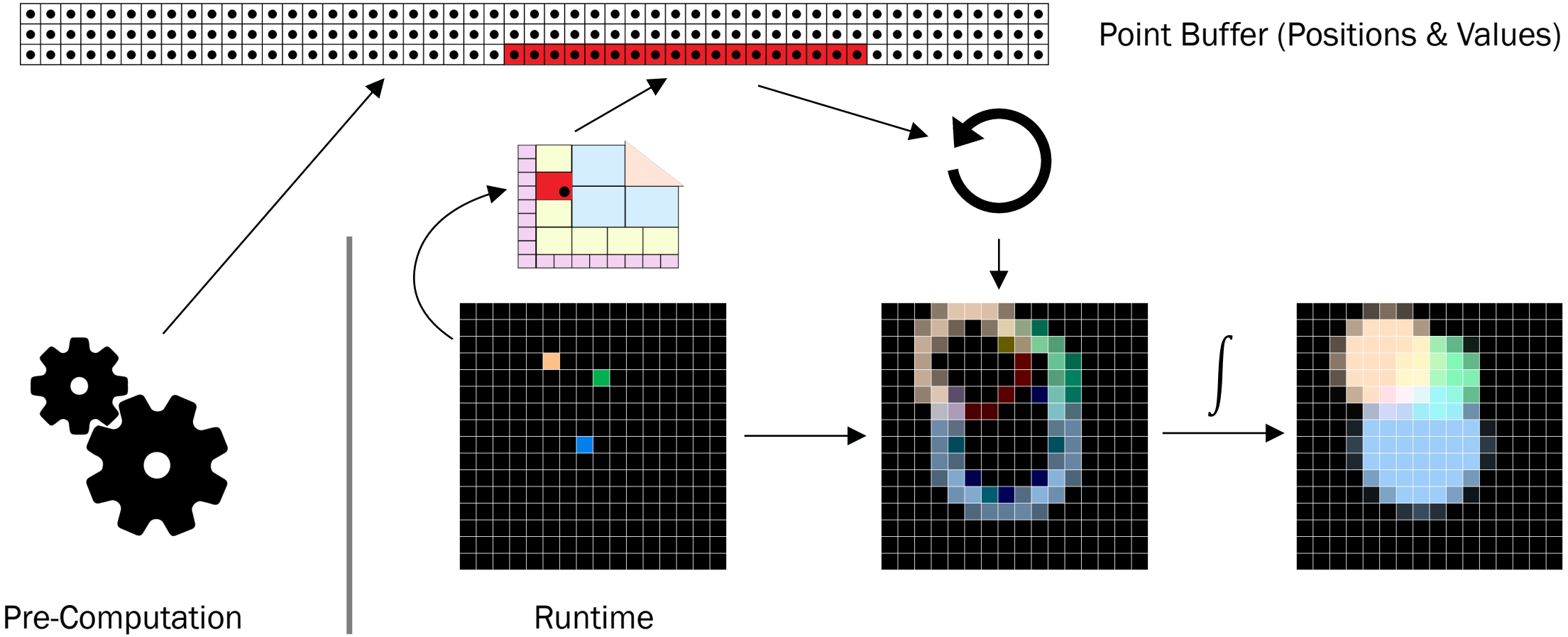


2. Kernel Generation



3. Sparsification

Pipeline



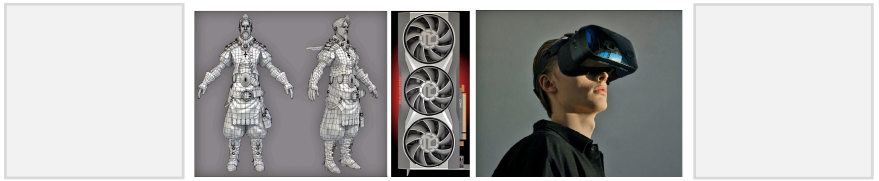

ø163ms/frame (on GTX 980 Ti)


Aperture
Focal Plane
Shutter
Time

1

Laplacian Kernel Splatting for Efficient Depth-of-field and Motion Blur Synthesis or Reconstruction
T Leimkühler, H-P Seidel, T Ritschel
ACM Trans. Graph. (Proc. SIGGRAPH 2018) 37(3)

2

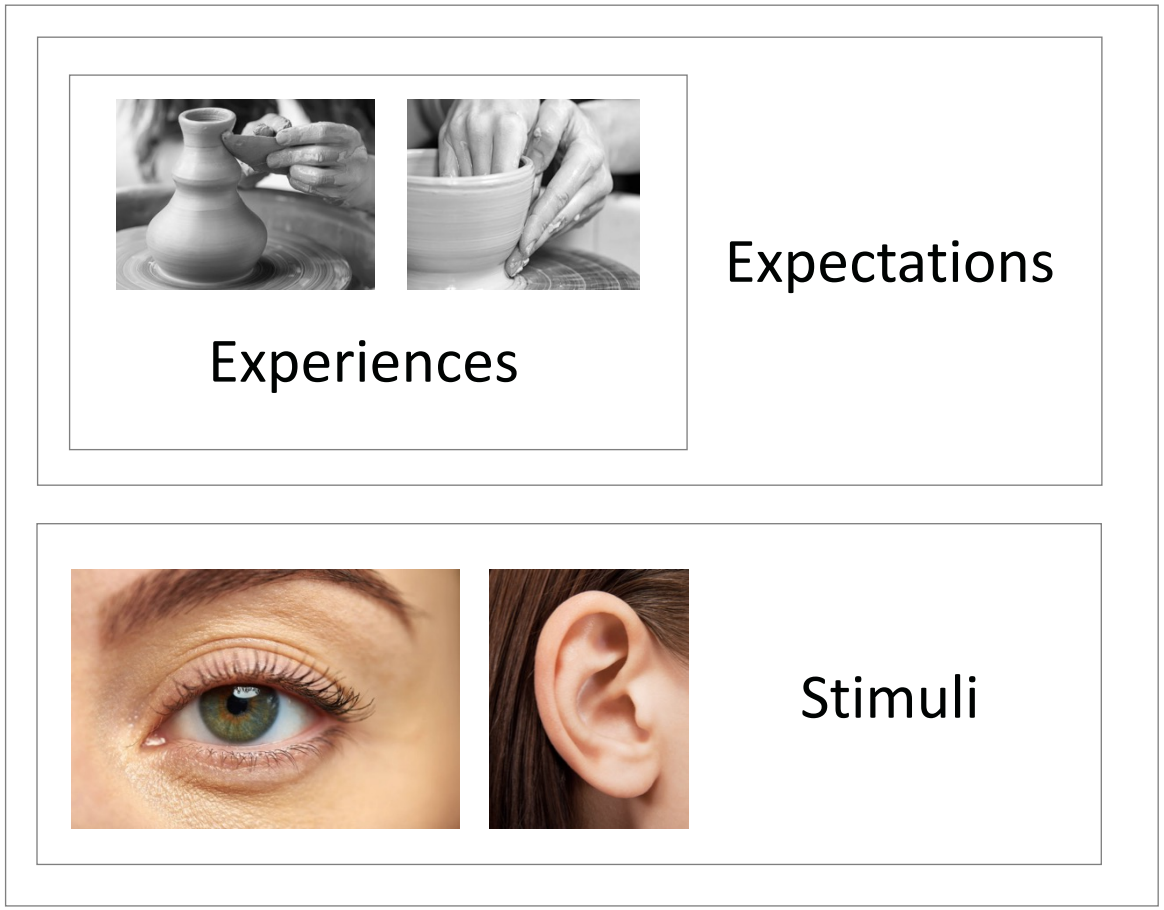


Perceptual Rasterization for Head-mounted Display Image Synthesis
S Friston, T Ritschel, A Steed
ACM Trans. Graph. (Proc. SIGGRAPH 2019) 38(3)

3

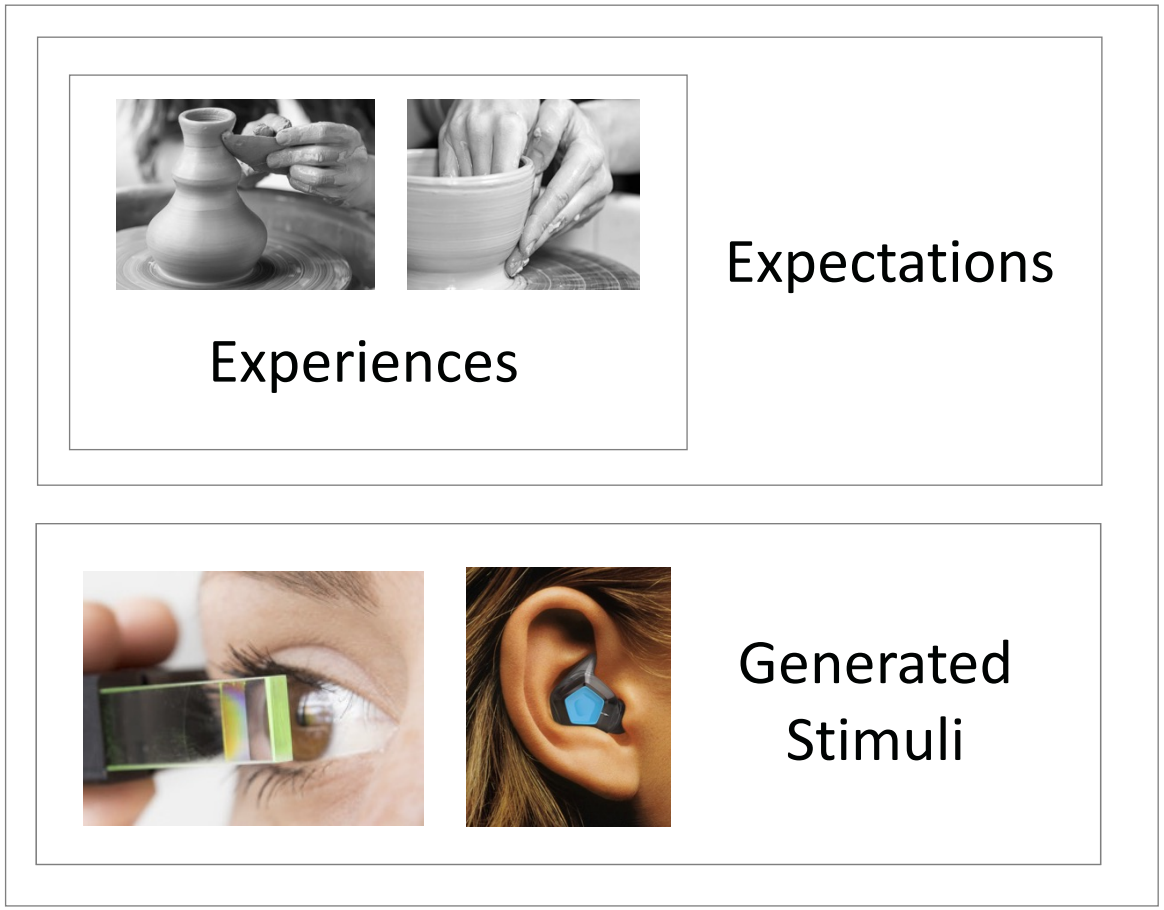
Beyond Blur: Real-time Metamers for Foveated Rendering
D Walton, R Kuffner-dos Anjos, S Friston, D Swapp, A Steed, T Ritschel
ACM Trans Graph (Proc. SIGGRAPH 2021) 40(3)

Perceiving the Real World




World

Perceiving Virtual Reality

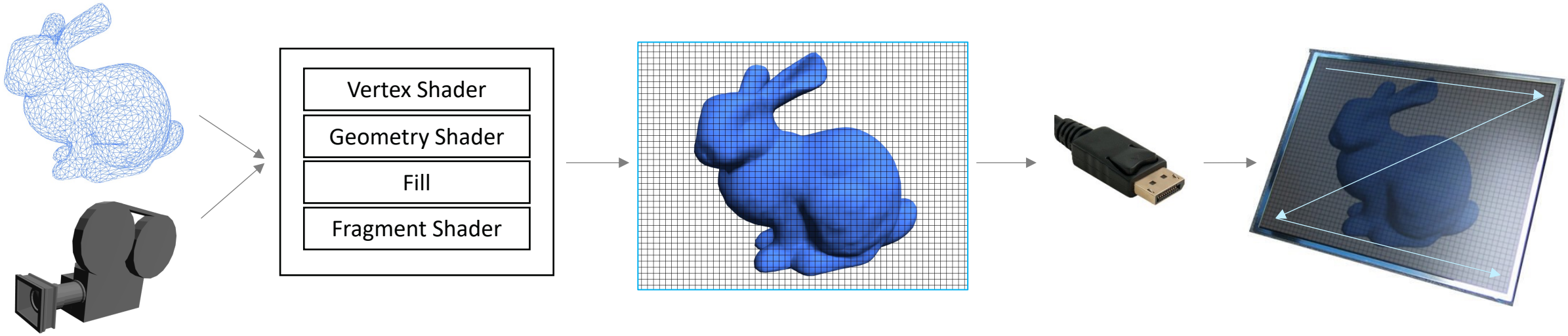


World

Perceiving Virtual Reality: Resolution

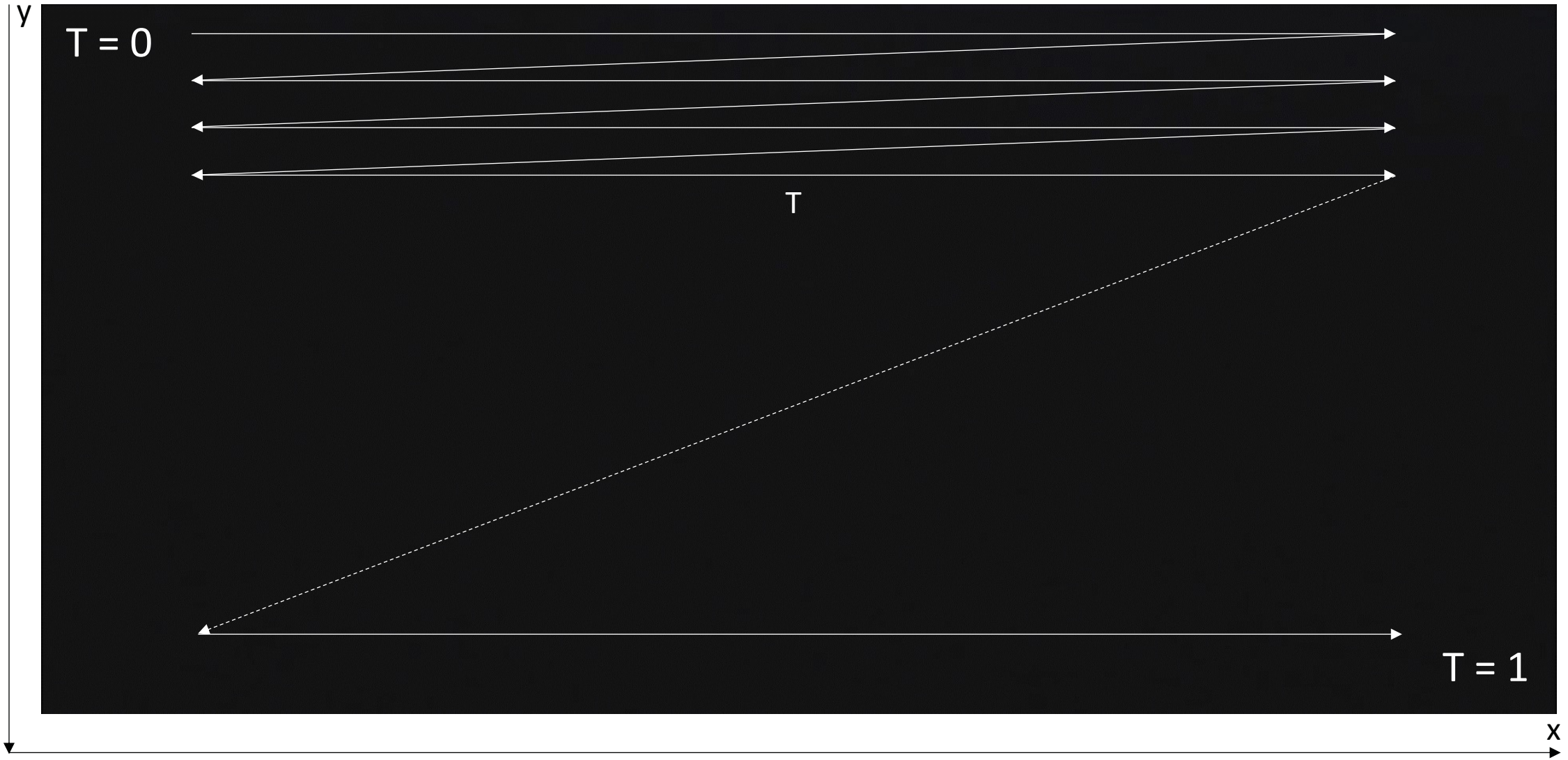
Perceiving Virtual Reality: Latency

Background: Current Graphics Pipelines

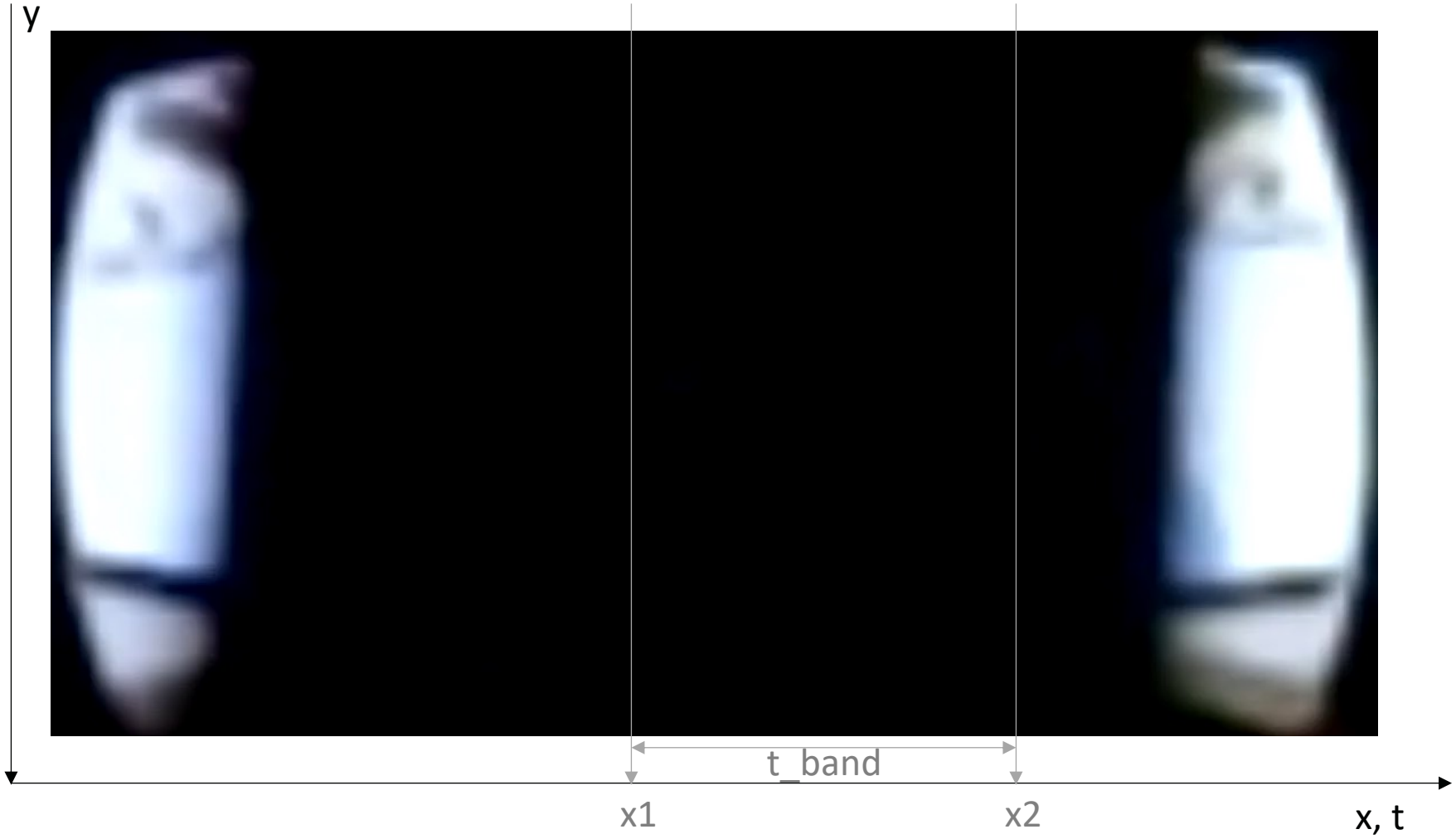


Draw Calls	Projection	Render Texture	Transmission	Scanout
Fixed Time	Linear	Uniform	Sequential	Sequential

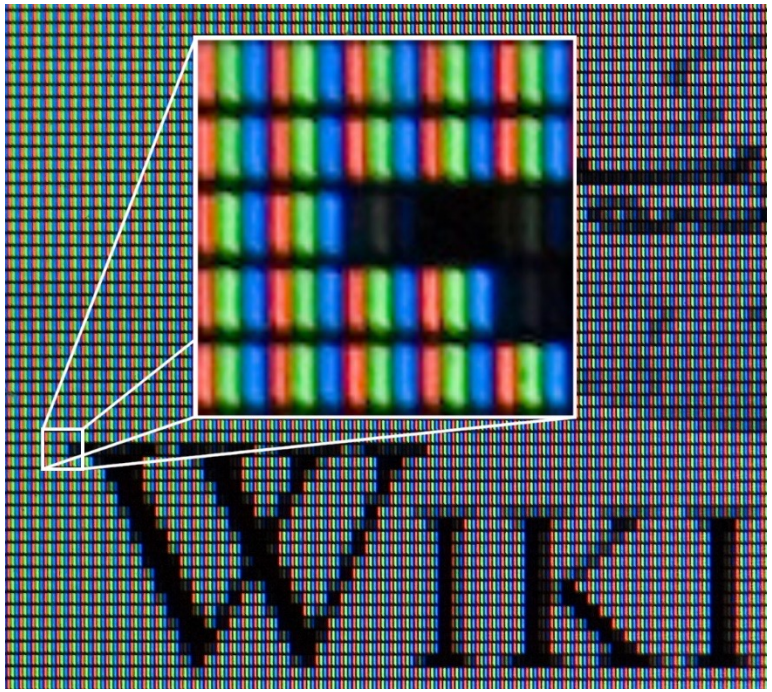
Background: Sequential Scan-outs

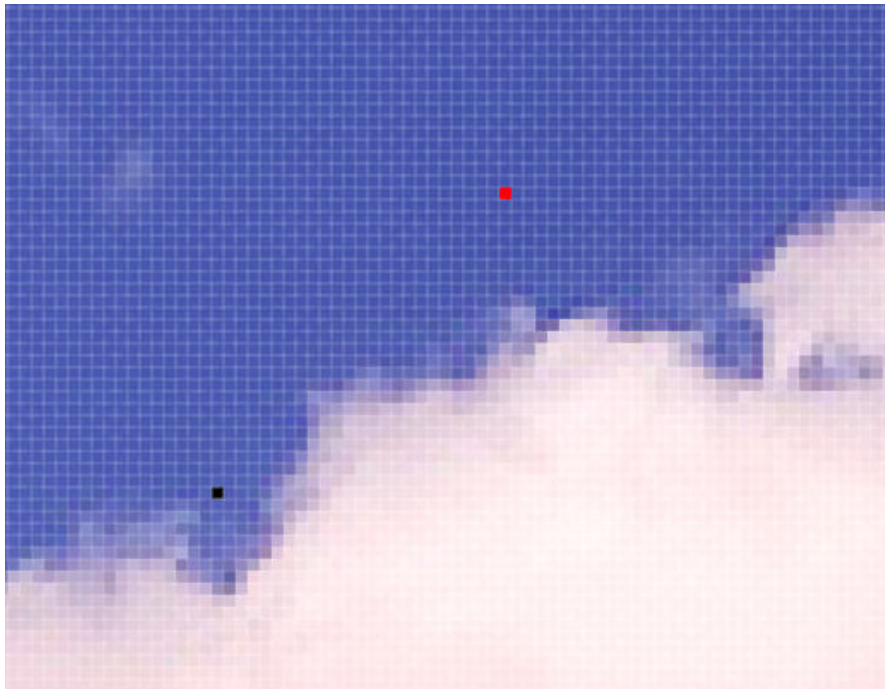


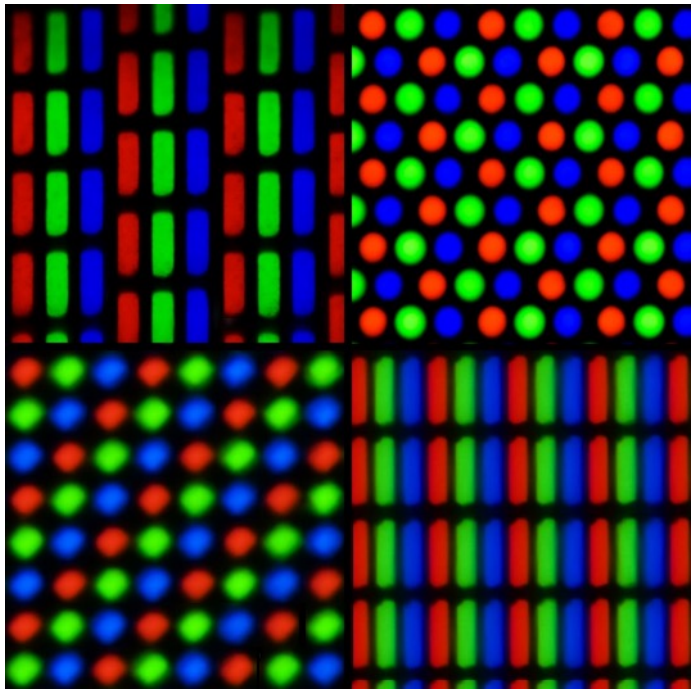
Techniques : Rolling Displays



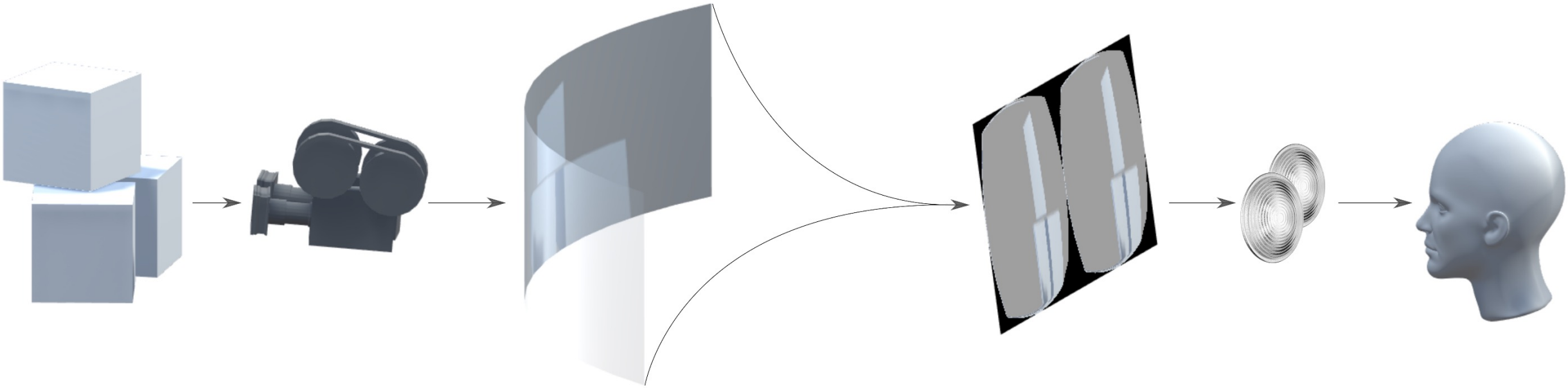
Background: Uniform Sampling



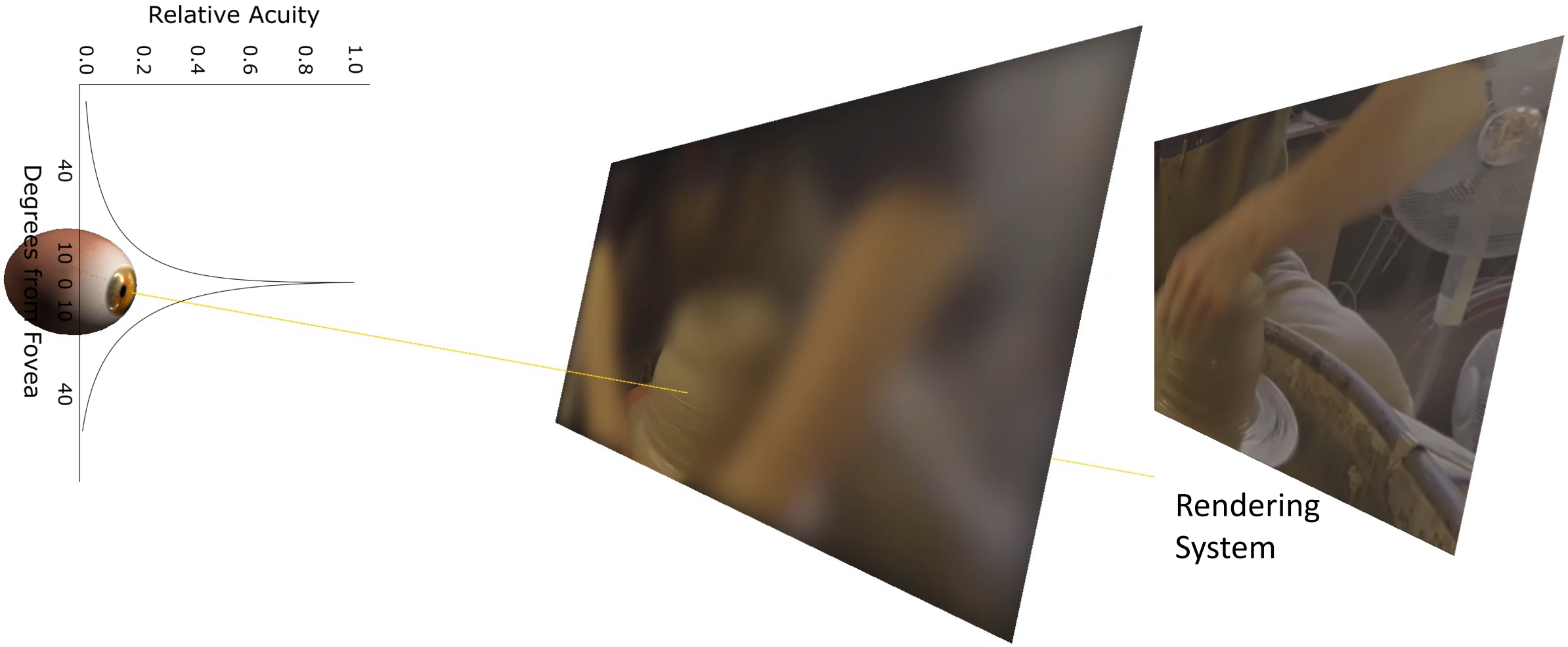




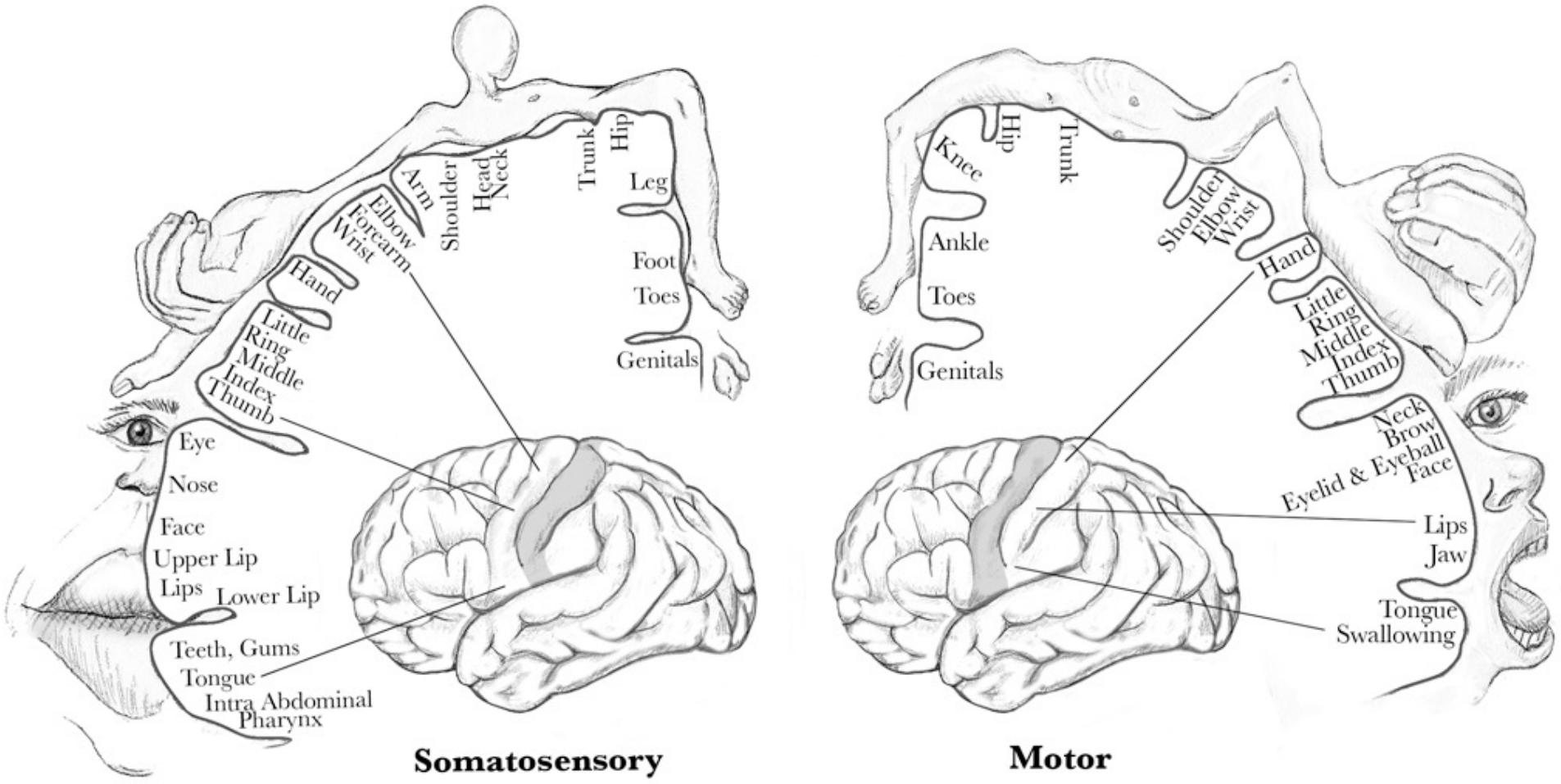
Background: Lens Distortion



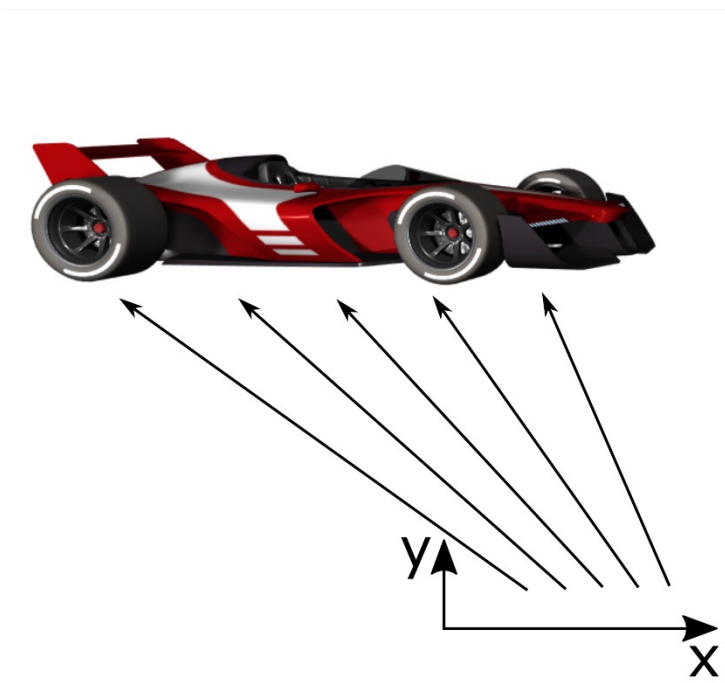
Techniques: Foveated Rendering



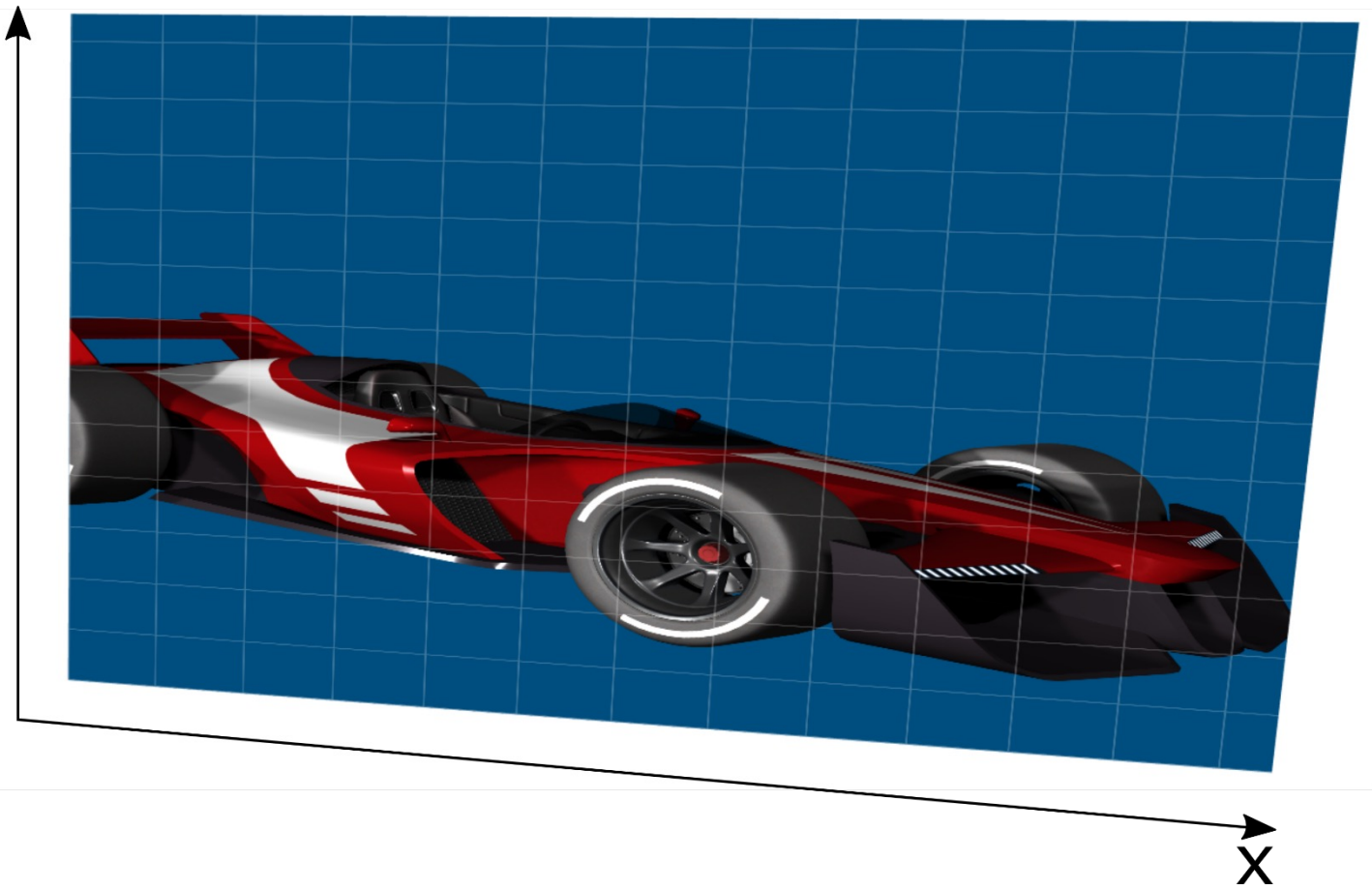
Cortical Magnification



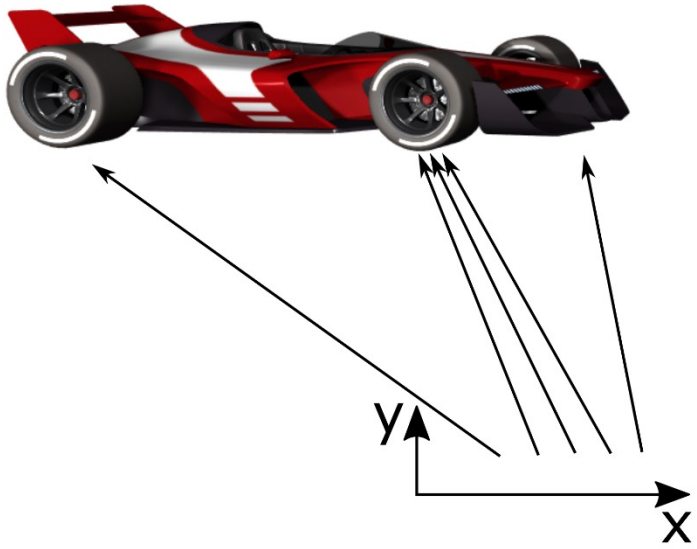
Our Technique: Overview



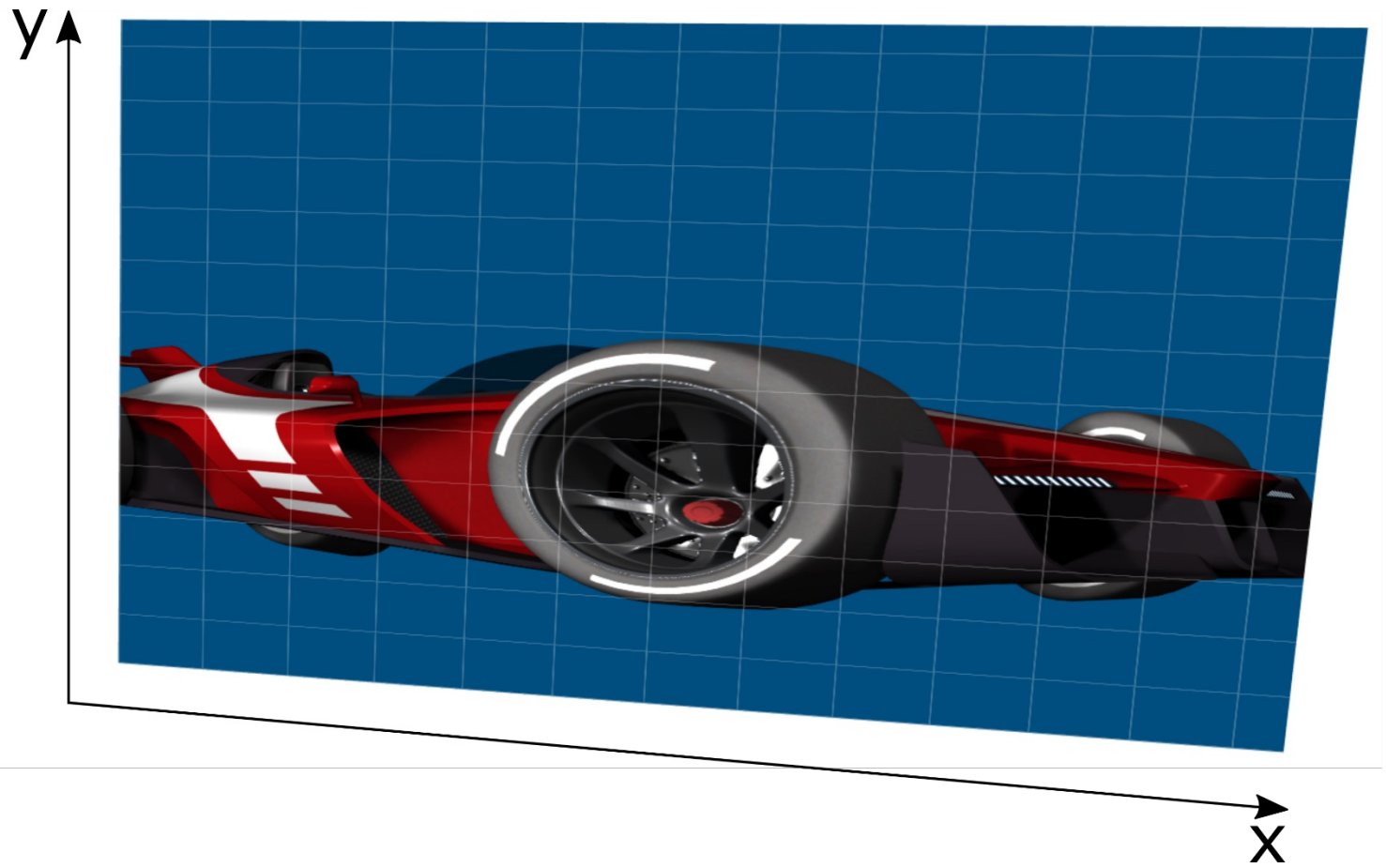
$ray(x, y, 0)$



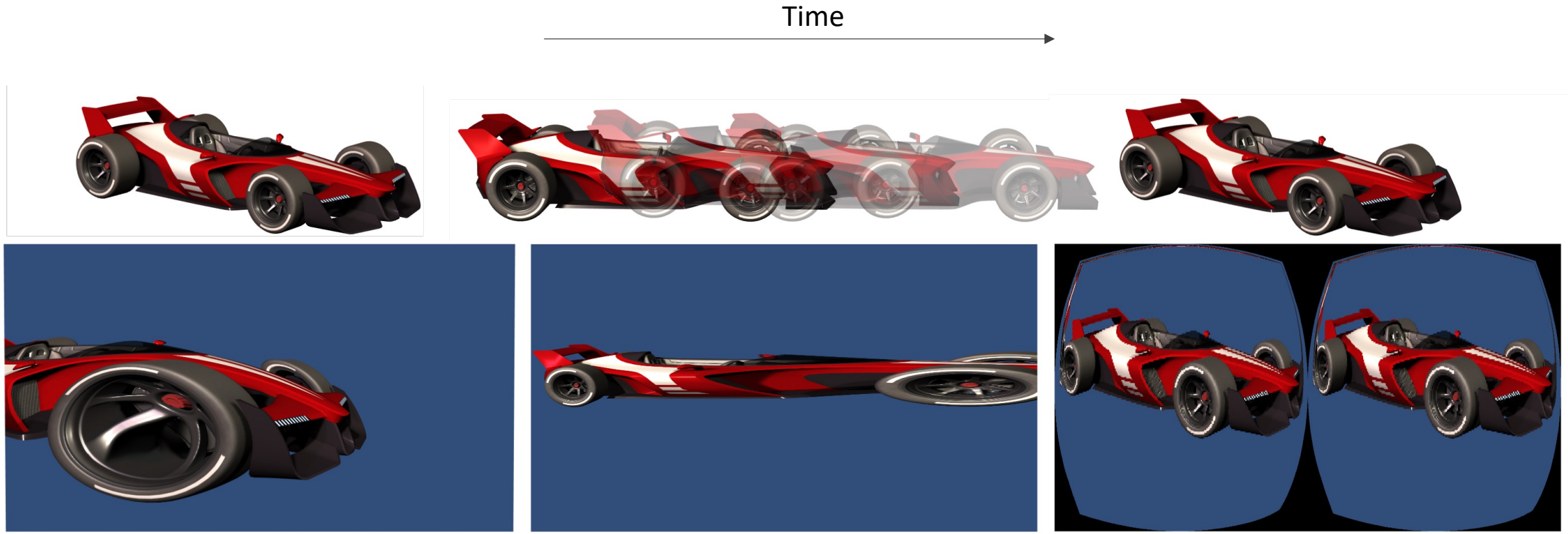
Our Technique: Overview



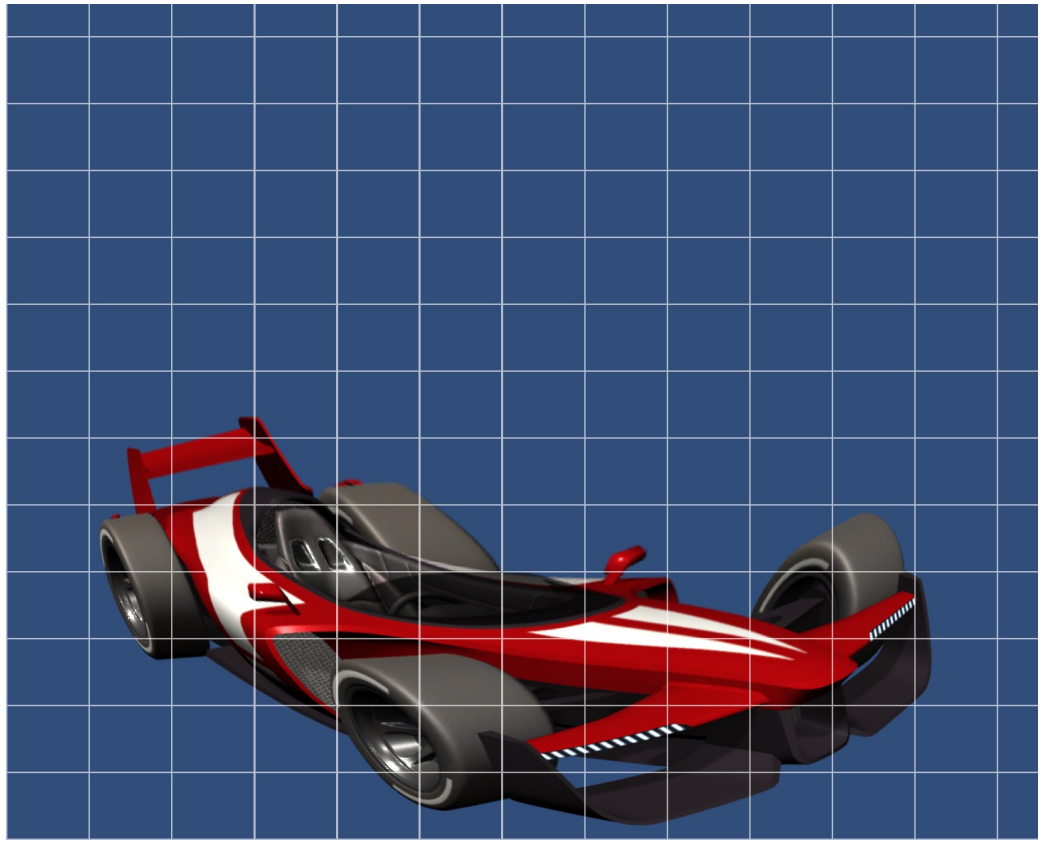
$$ray(f_x(x), f_y(y), f_t(t)))$$



Our Technique: Non-linear Framebuffer

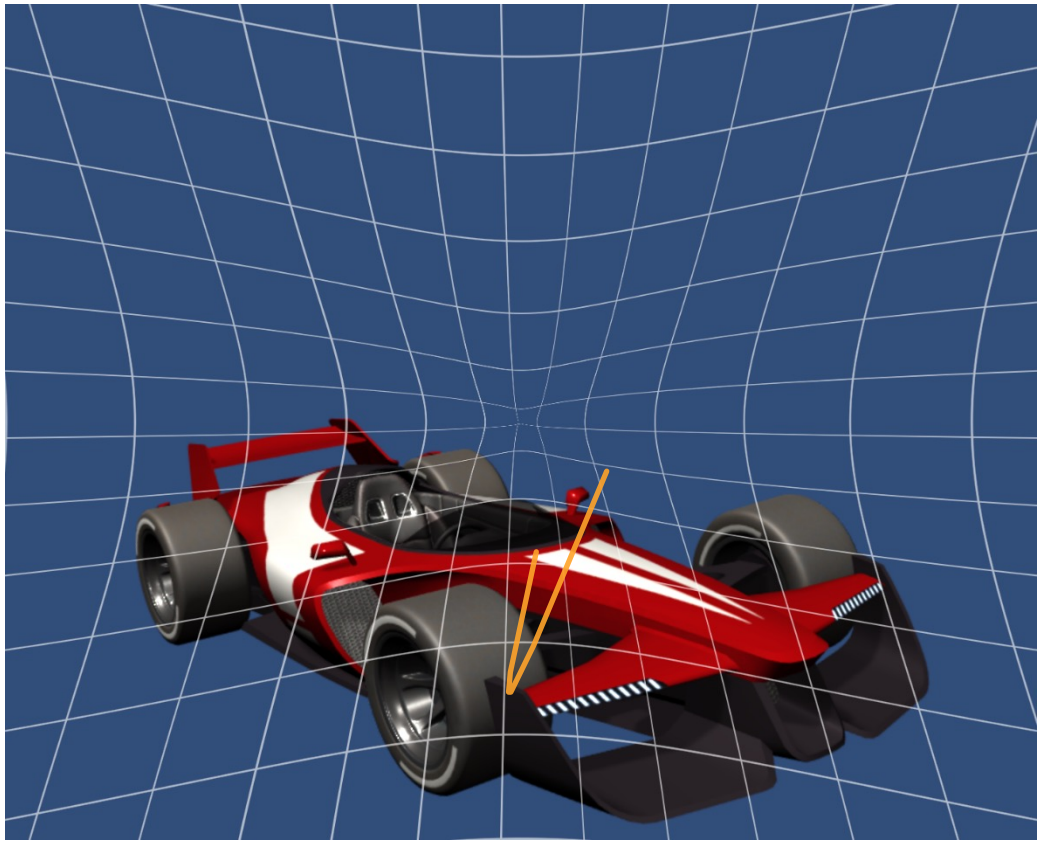


Our Technique: Unsampling



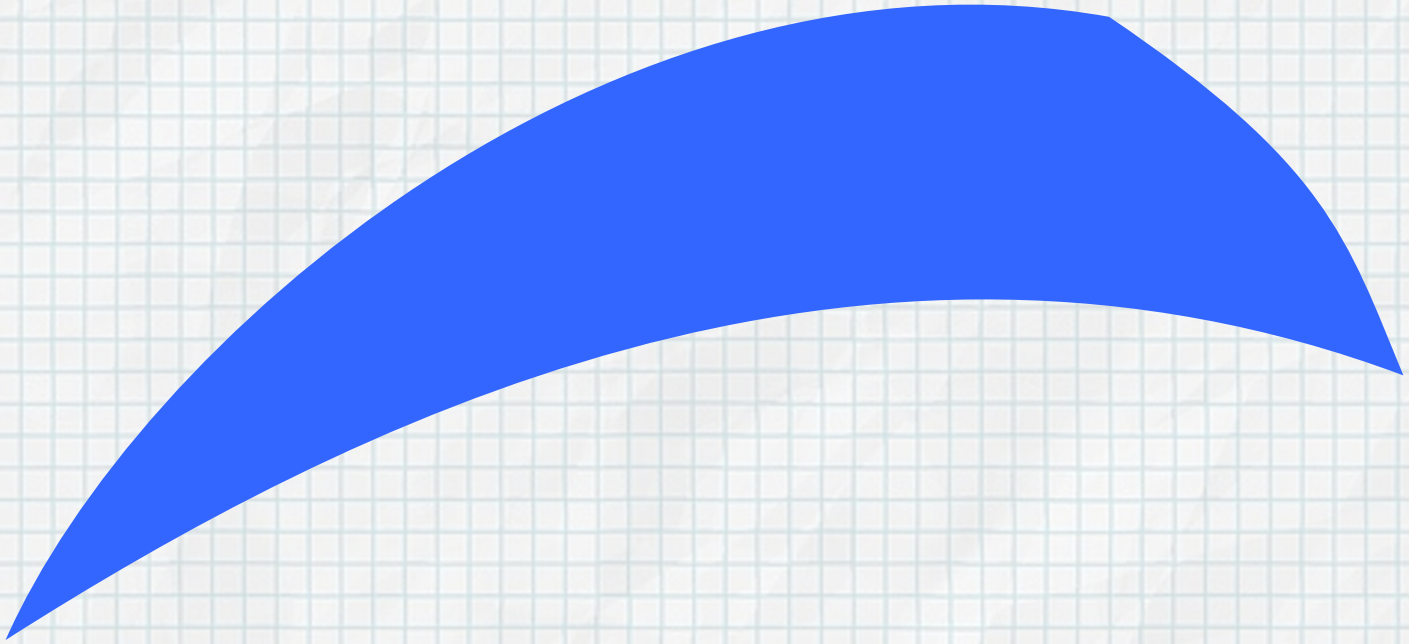
Foveal Buffer

Unfoveate



Traditional Display

Our Technique: Non-Linear Rasterization



Ray-tracing vs. Rasterization

Ray Tracing

```
for every pixel create a ray r
 for every primitive p
  test pair r,p for intersection
  if hit
   if hit closest
    store
```

Rasterization

```
for every primitive p
 for every pixel create a ray r
   test pair r,p for intersection
   if hit
    if hit closest
     store
```

Ray-tracing vs. Rasterization

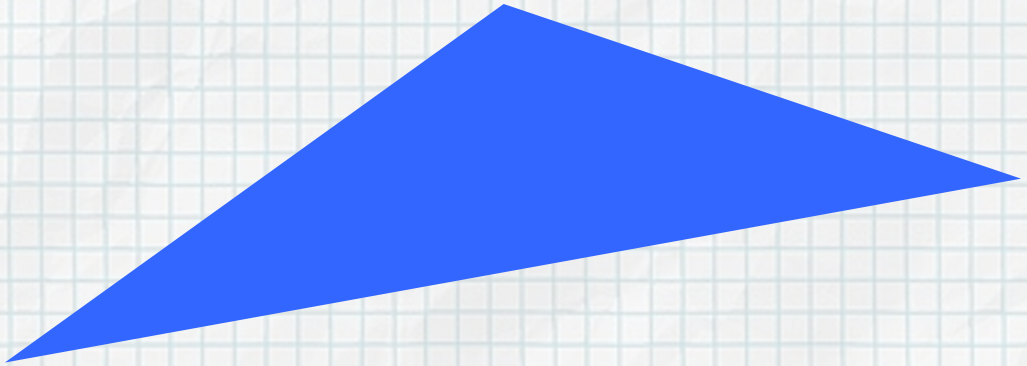
Ray Tracing

```
for every pixel create a ray r
 for every primitive p
  test pair r,p for intersection
  if hit
   if hit closest
    store
```

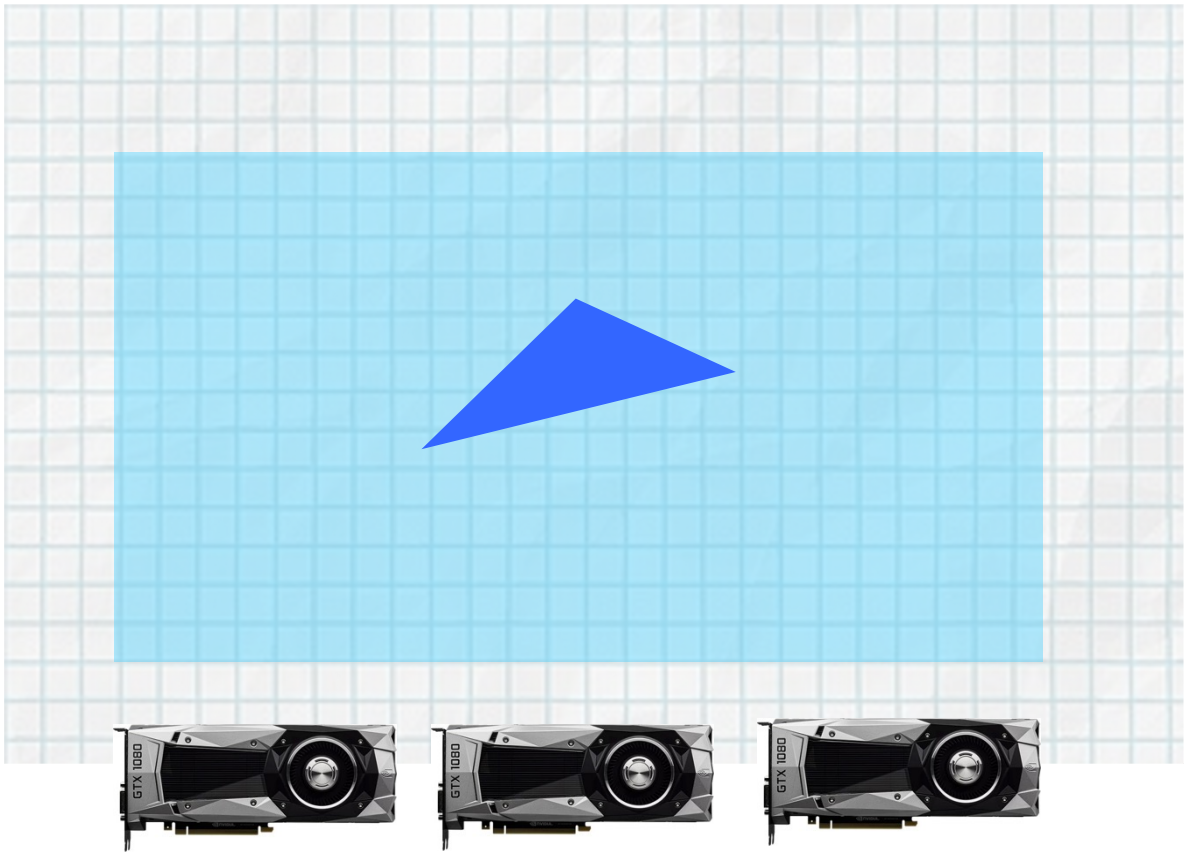
Rasterization

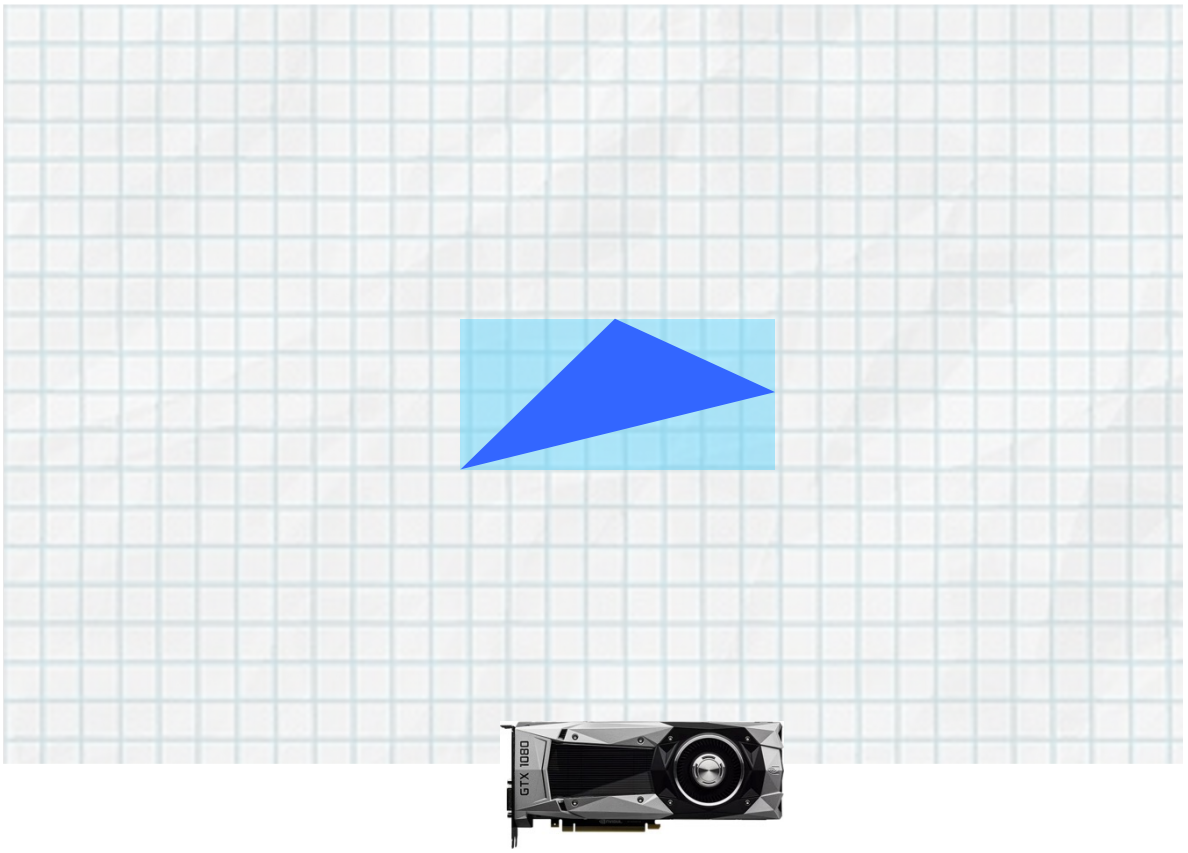
```
for every primitive p
 for some pixel create a ray r
   test pair r,p for intersection
   if hit
    if hit closest
     store
```

Bounding:



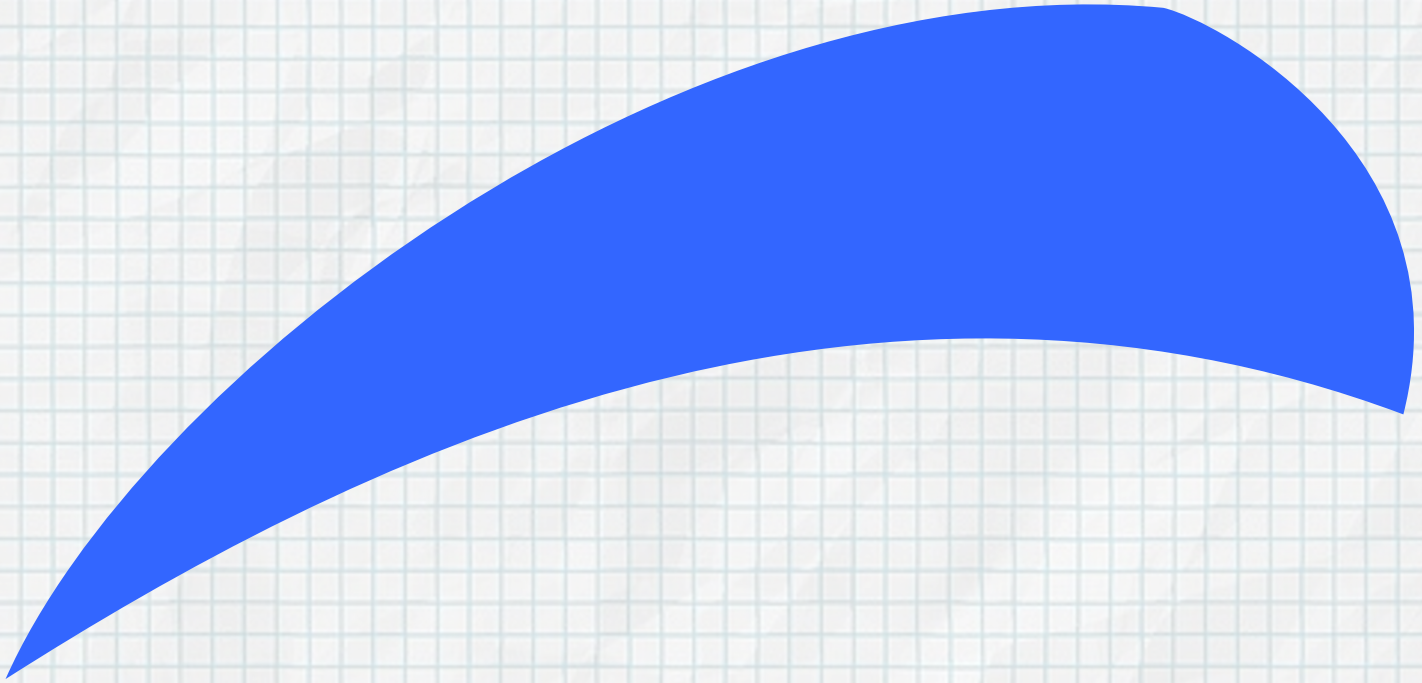
Tightness is currency



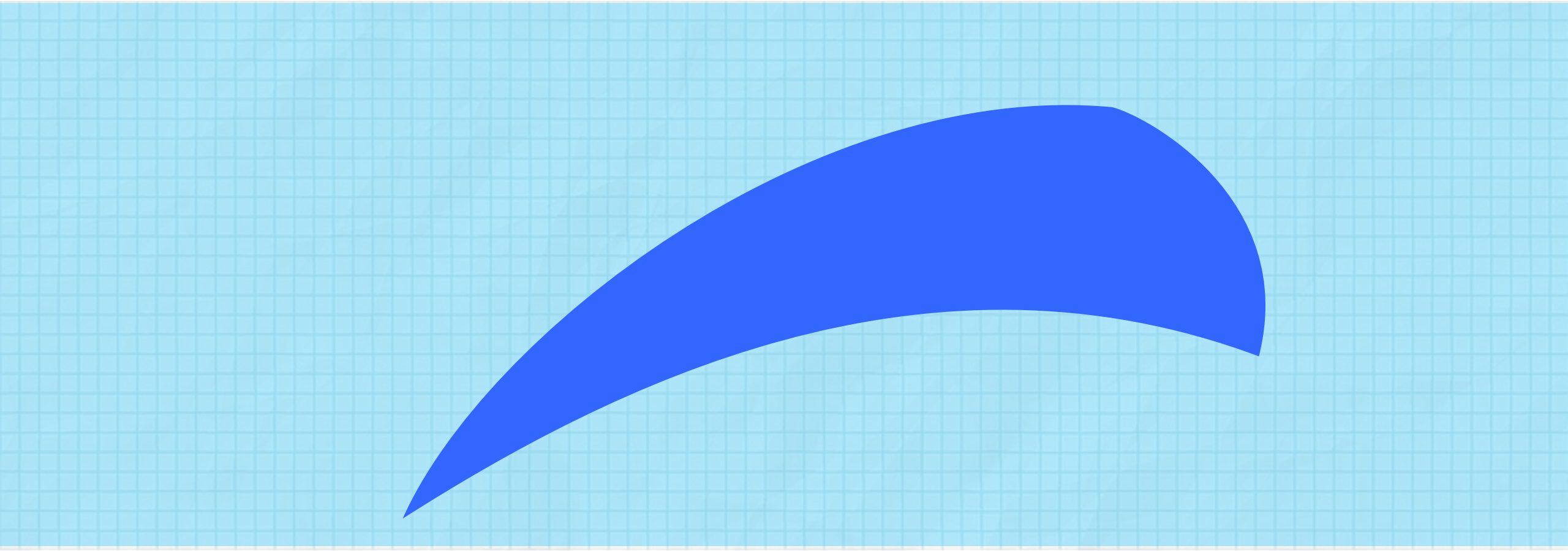


$$$ > $

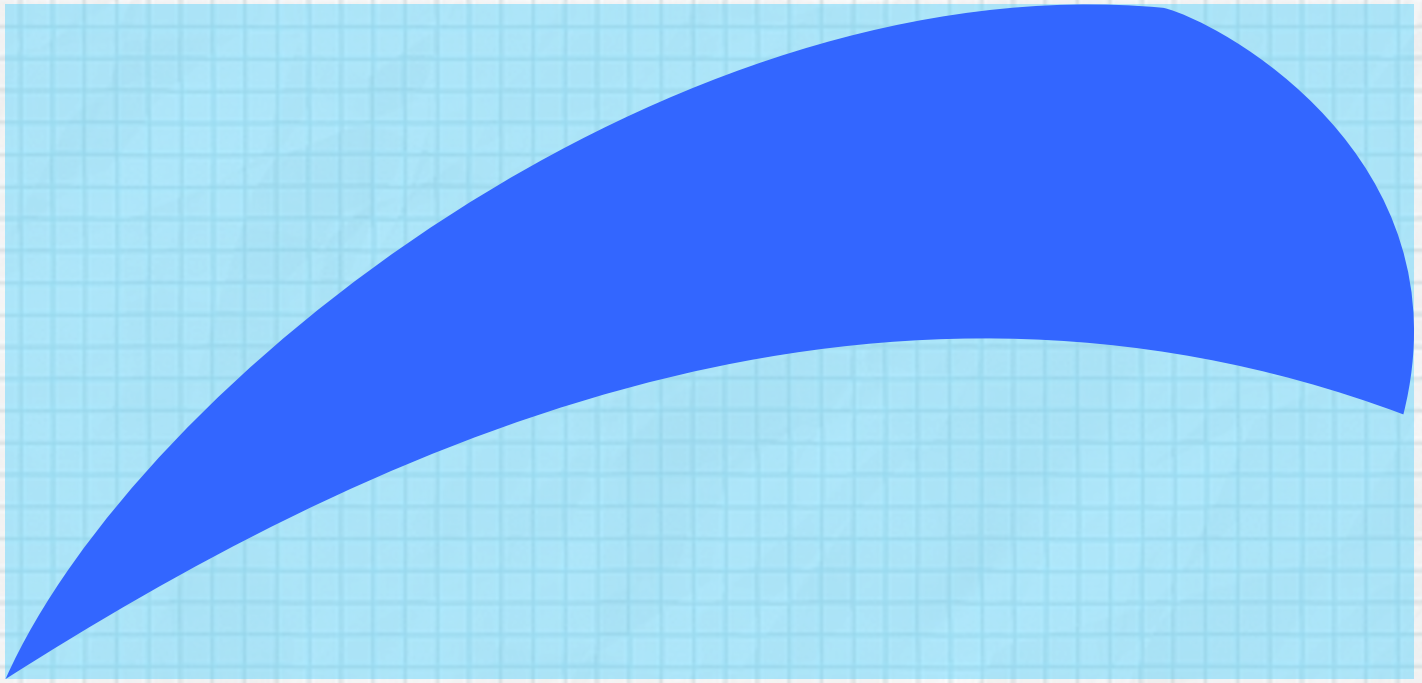
Bounding: Curved primitive



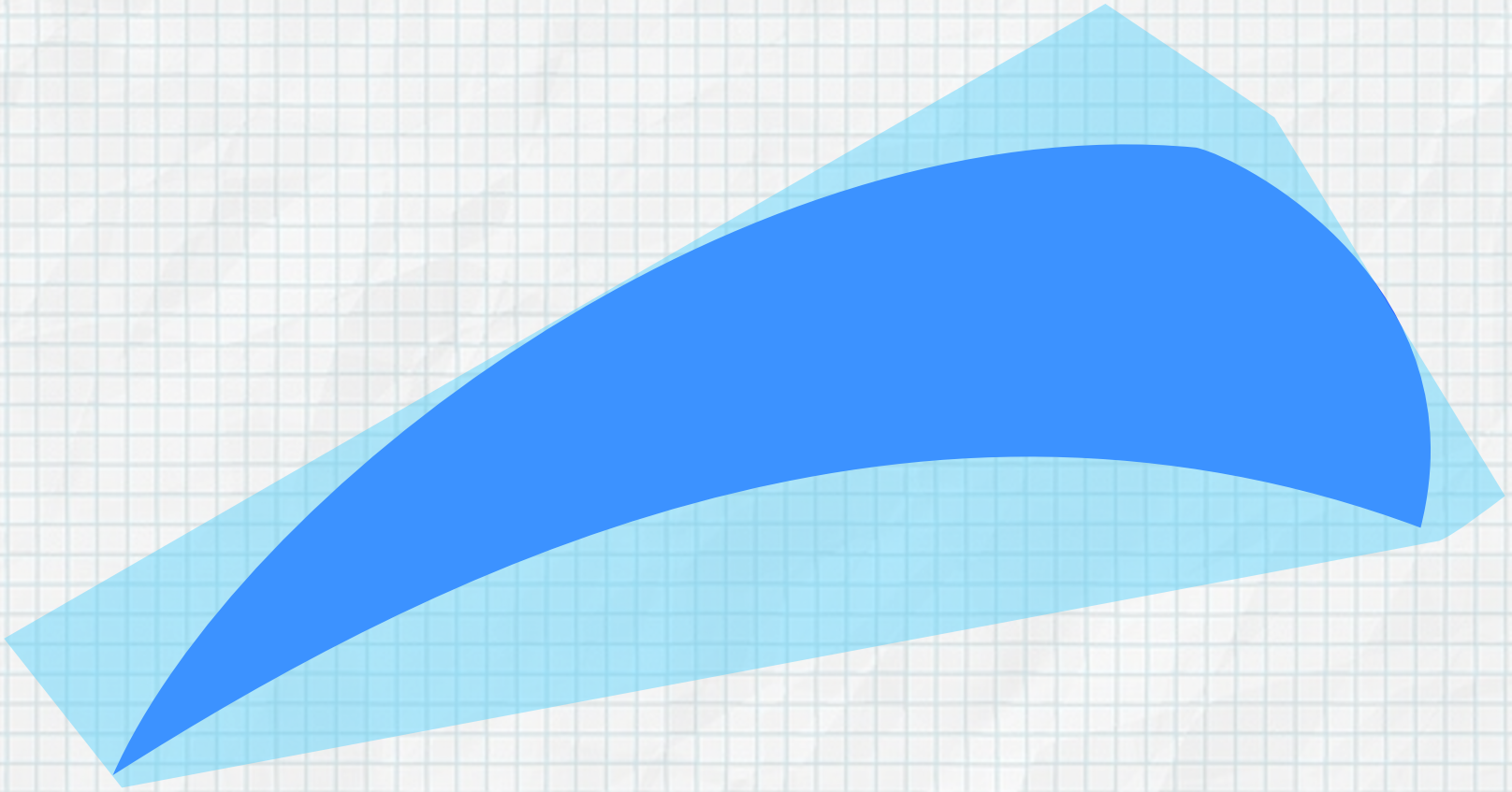
Bounding: Whole screen



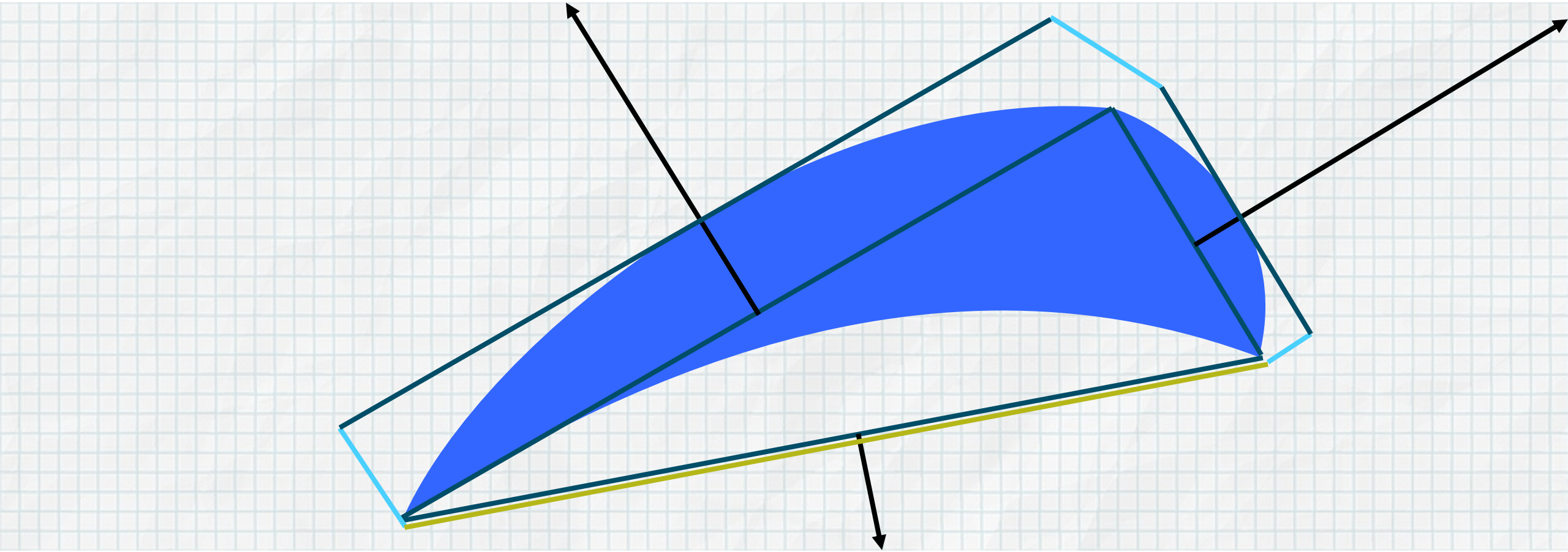
Bounding: Bounding box always works



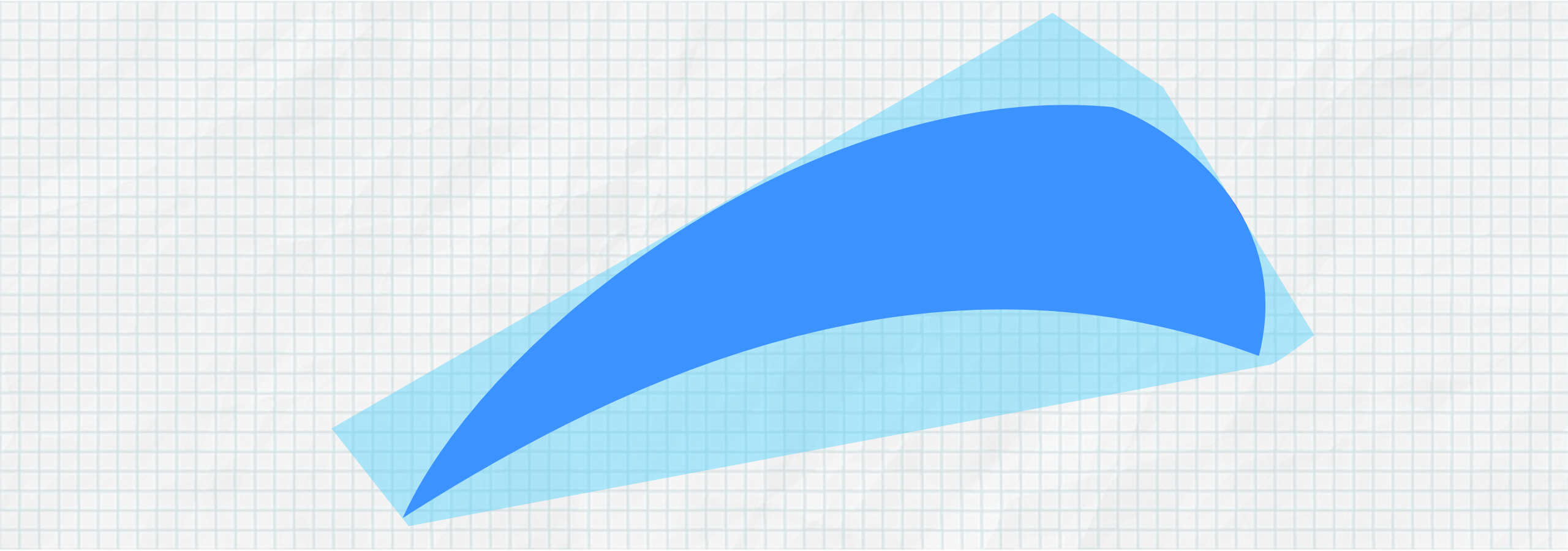
Bounding: Tight perceptual bounds



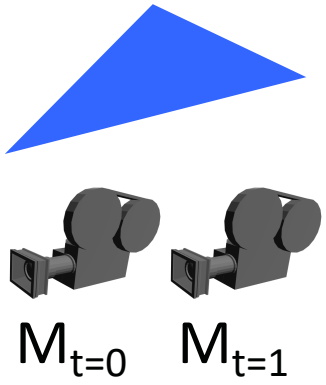
Bounding: Construction

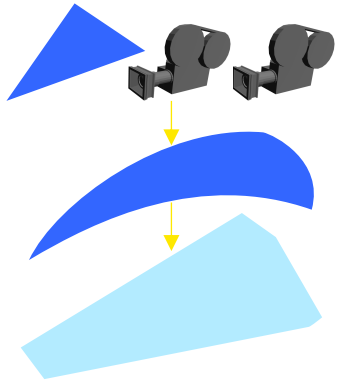


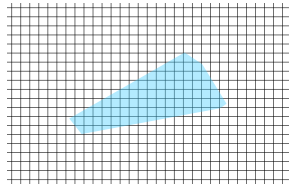
Bounding: Construction

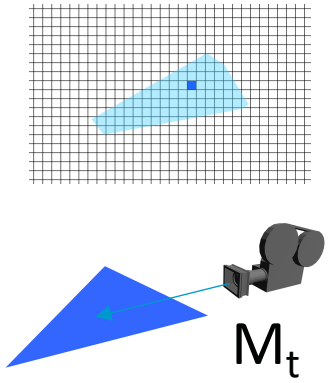


Implementation









Draw Call Submission

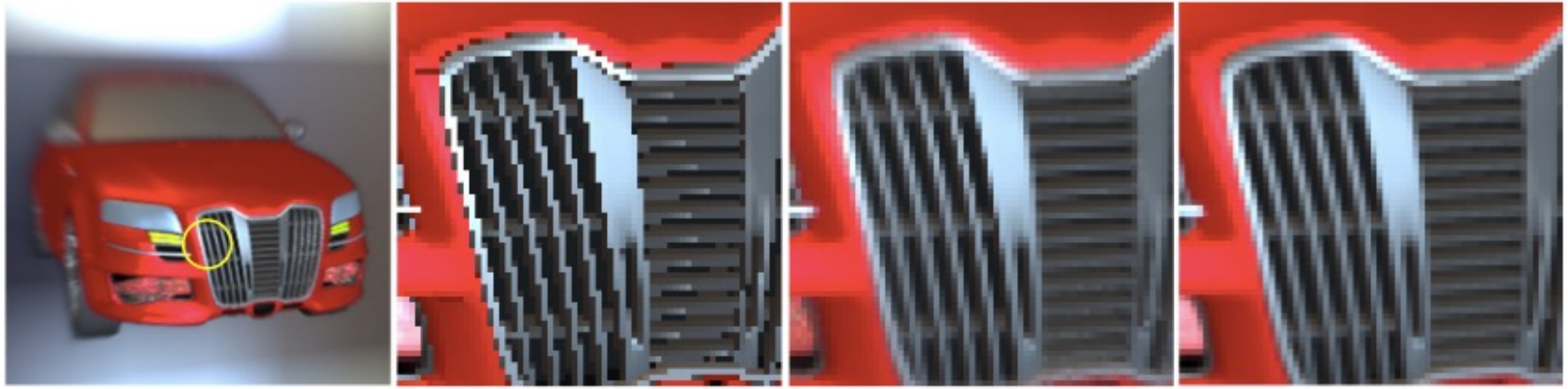
Vertex Shader

Geometry Shader

Fill Stage

Fragment Shader

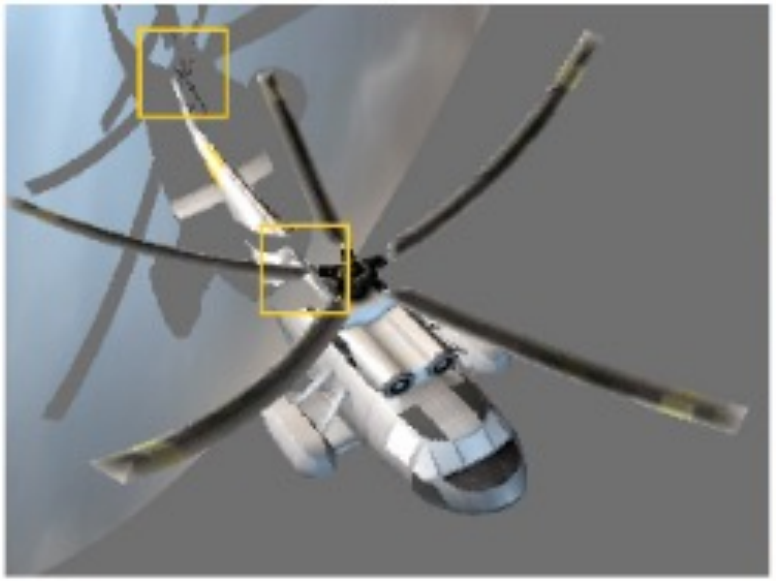
Result: Foveation 焦点



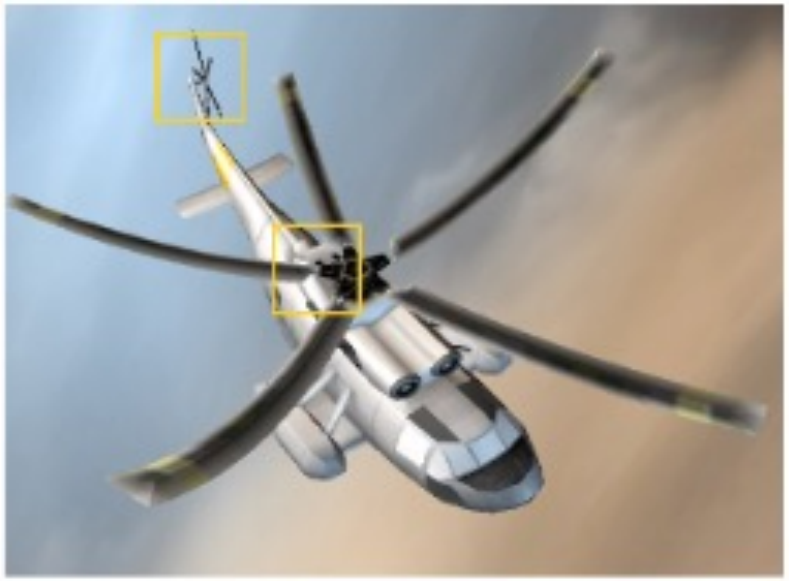
Foveal Region of Image | Common | Ours | Ray-traced Reference

Result: Rolling

Common



Warping



Ours

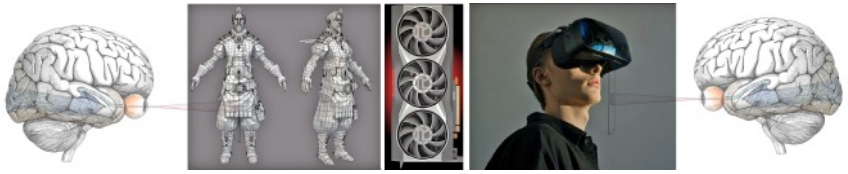
1

Laplacian Kernel Splatting for Efficient Depth-of-field and Motion Blur Synthesis or Reconstruction
T Leimkühler, H-P Seidel, T Ritschel
ACM Trans. Graph. (Proc. SIGGRAPH 2018) 37(3)

2

Perceptual Rasterization for Head-mounted Display Image Synthesis
S Friston, T Ritschel, A Steed
ACM Trans. Graph. (Proc. SIGGRAPH 2019) 38(3)

3



Beyond Blur: Real-time Metamers for Foveated Rendering
D Walton, R Kuffner-dos Anjos, S Friston, D Swapp, A Steed, T Ritschel
ACM Trans Graph (Proc. SIGGRAPH 2021) 40(3)

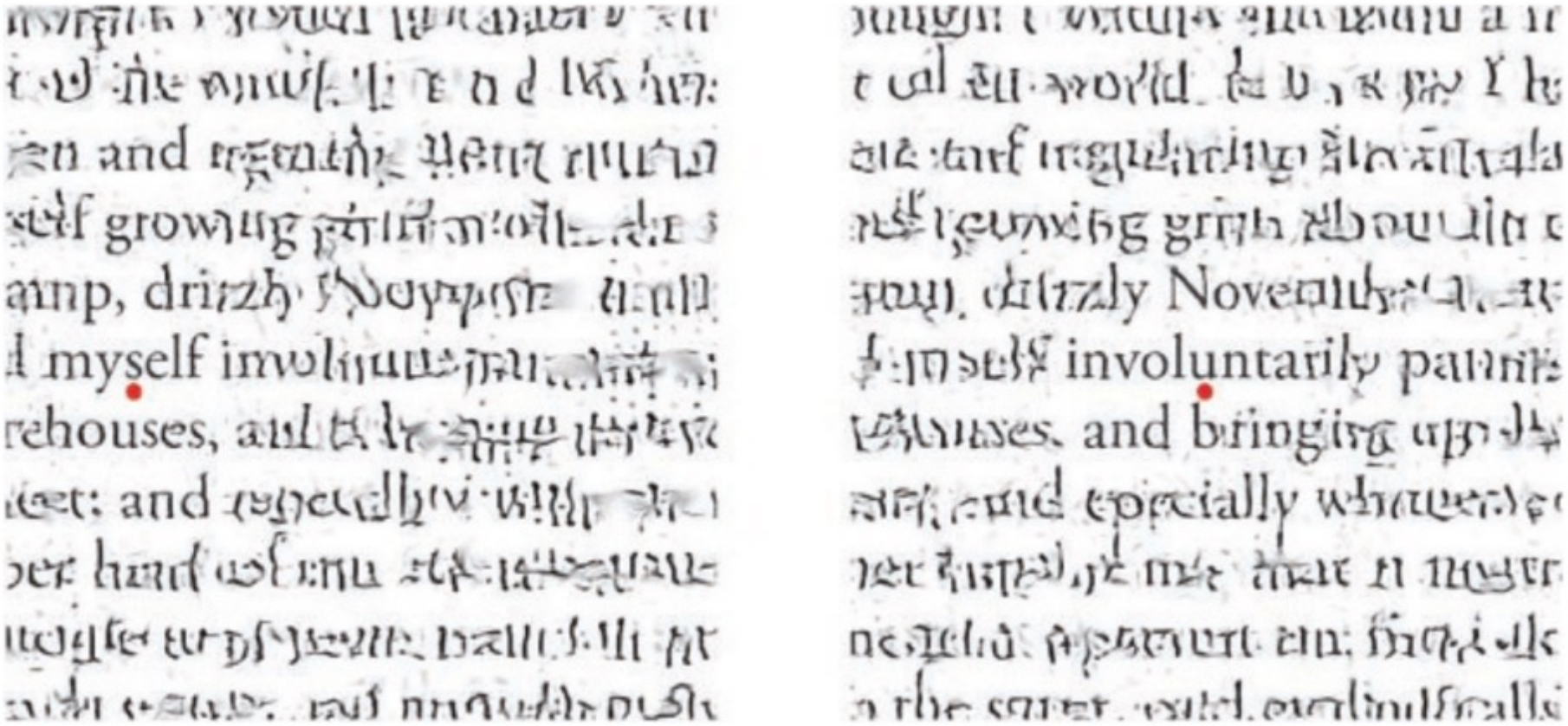
Blur

Beyond blur

What are metamers?

$$M_x := \{x' | f(x') = f(x)\}$$

What are image metamers?



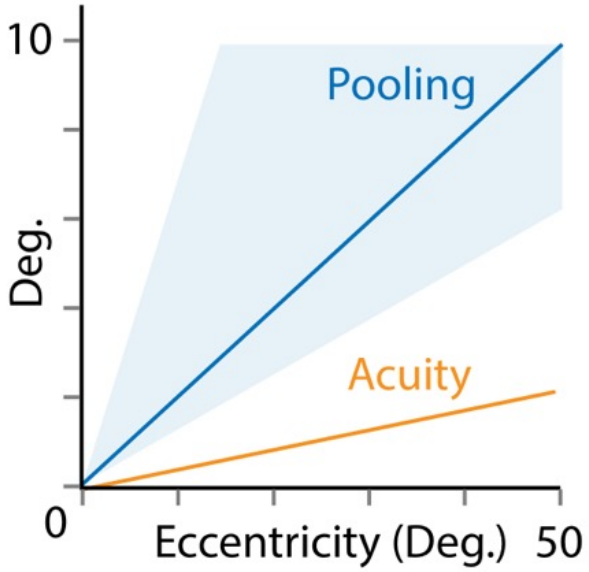
Metamers of the Ventral Stream, Freeman & Simoncelli, *Nature* 2011

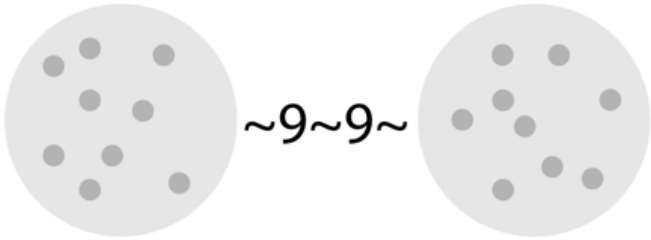
Our goal

Freeman & Simoncelli, 6 hours

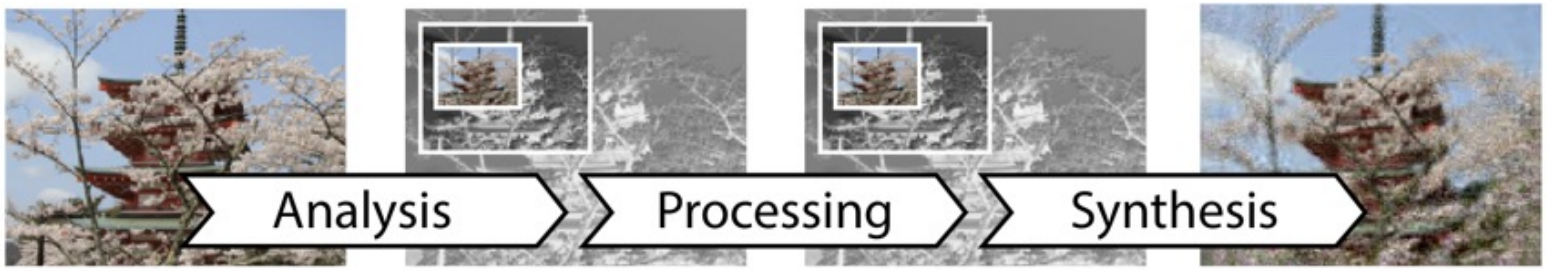
Ours, 10 ms

Why do image metamers exist?





Our real-time model

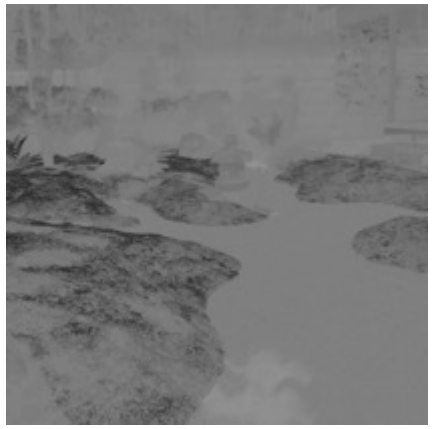


Analysis: Overview

- RGB to YCbCr
- Construct Steerable Pyramid
- Compute local 1st and 2nd moments

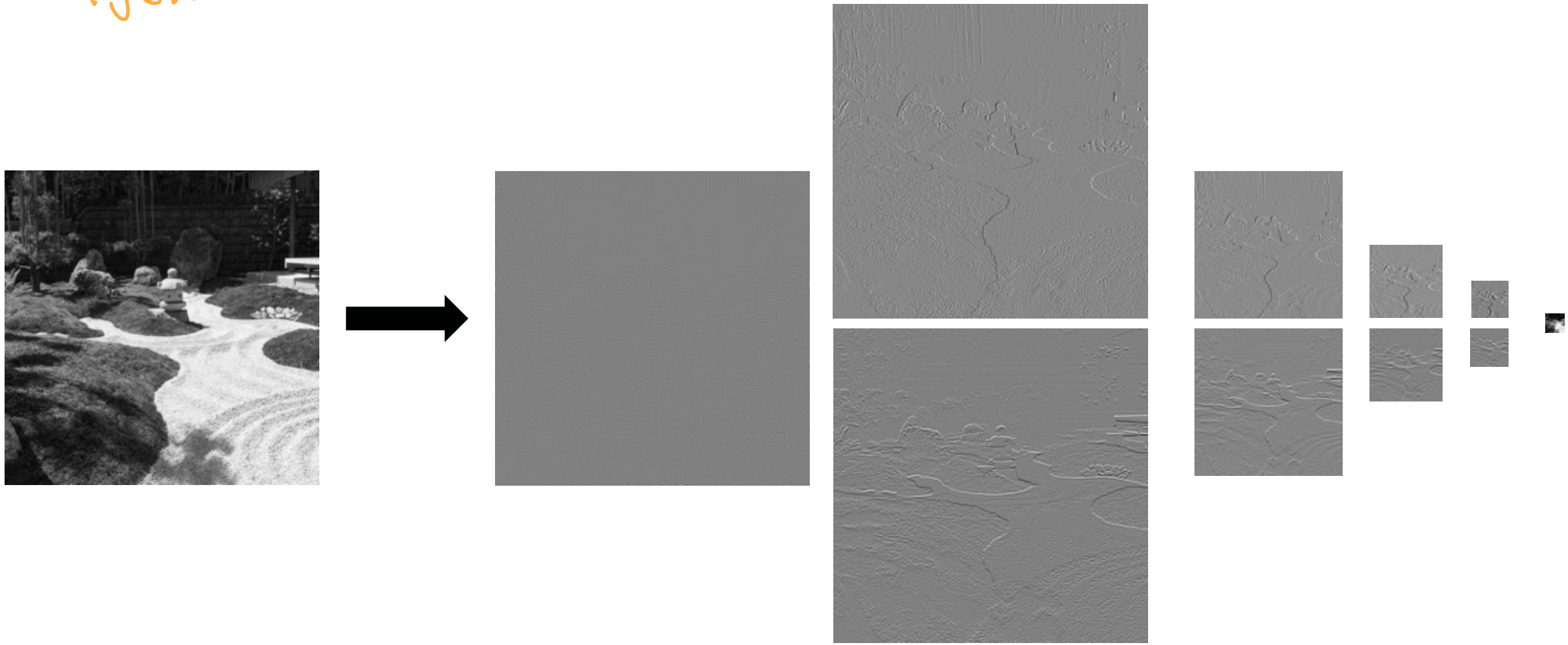
average

variance

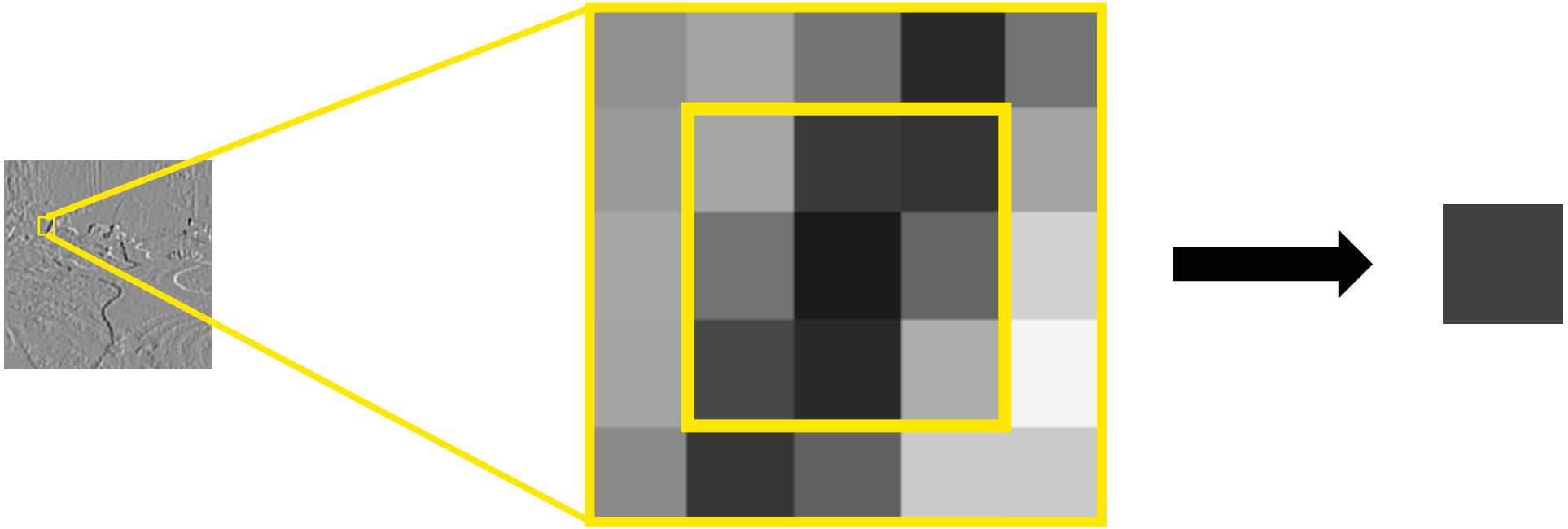


Steerable pyramid

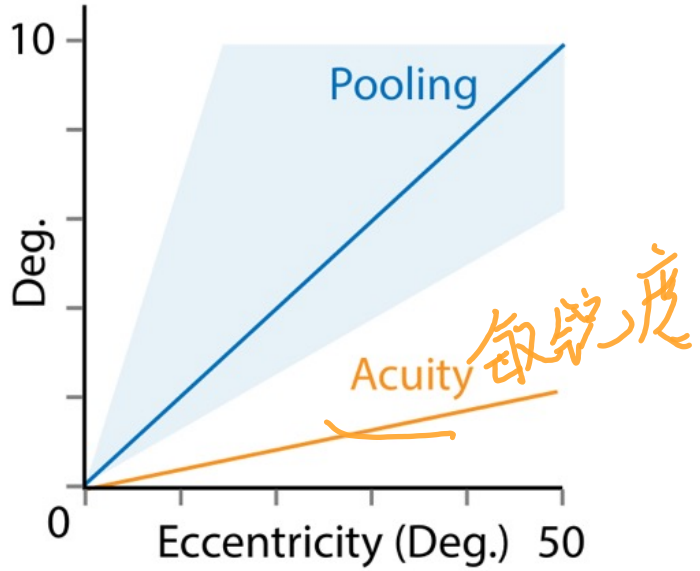
可操纵

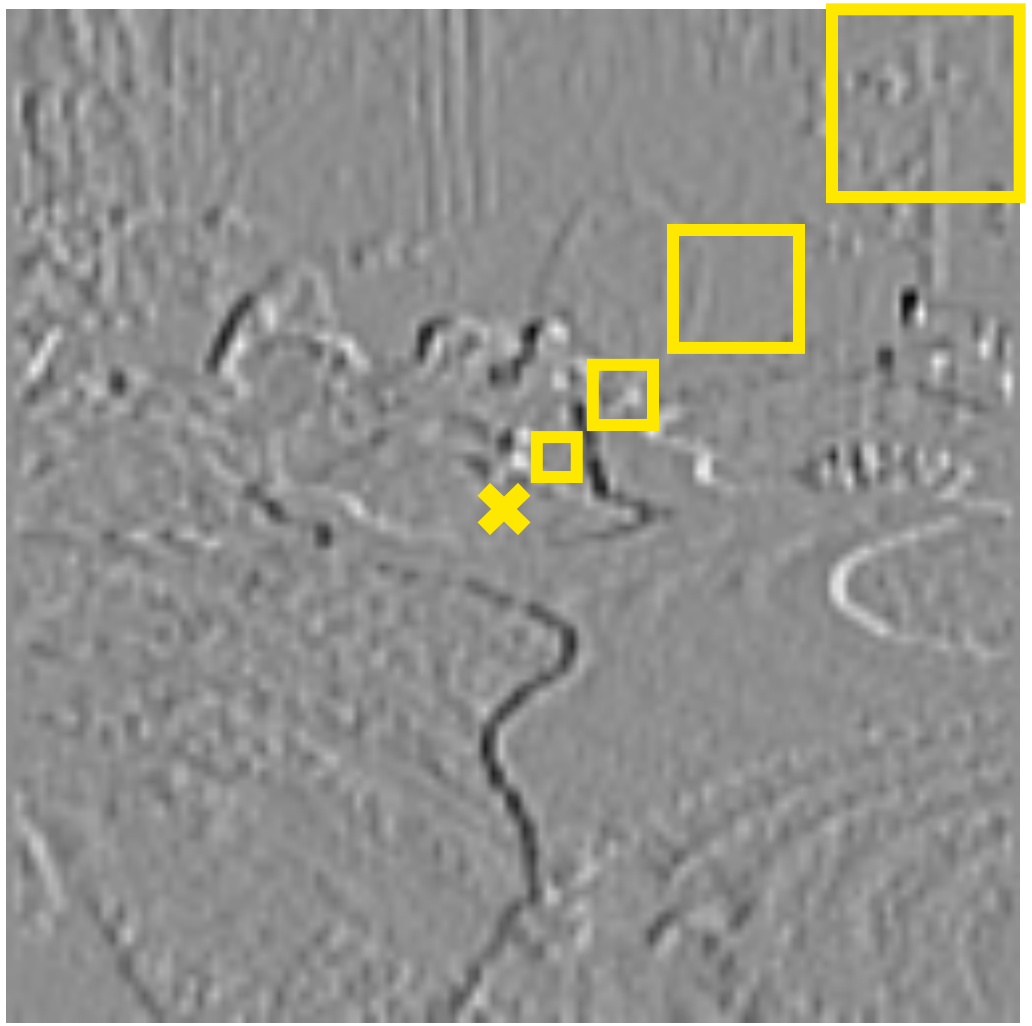


Computing Local Moments



Computing Local Moments



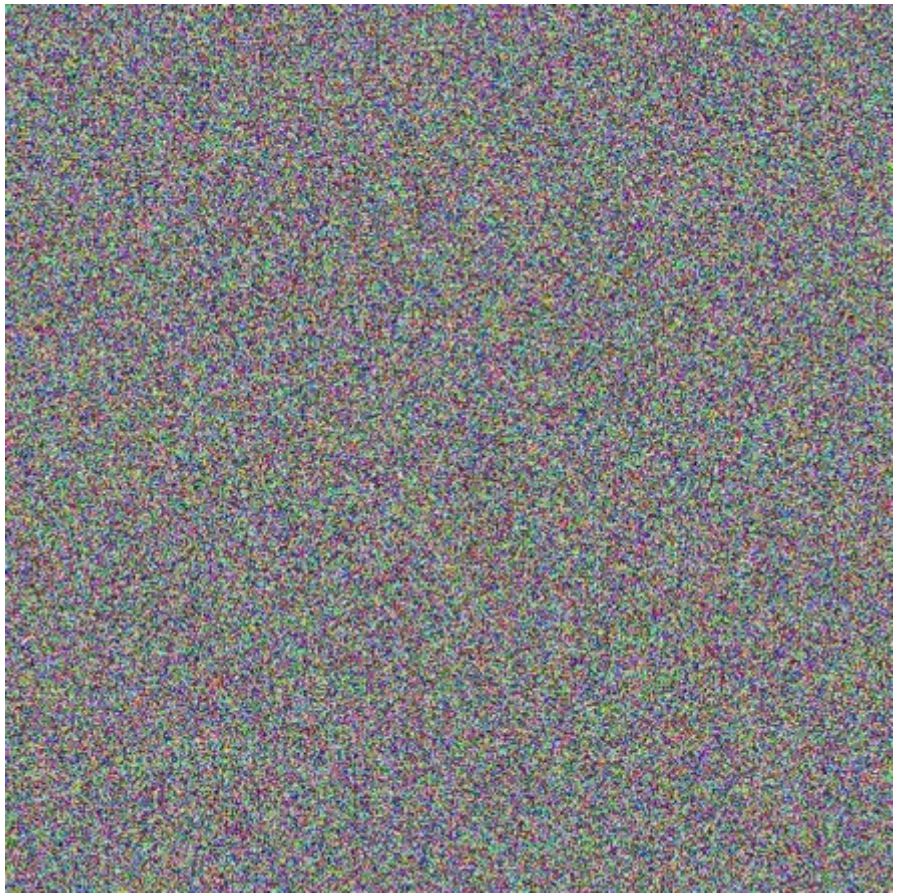


Computing Local Moments

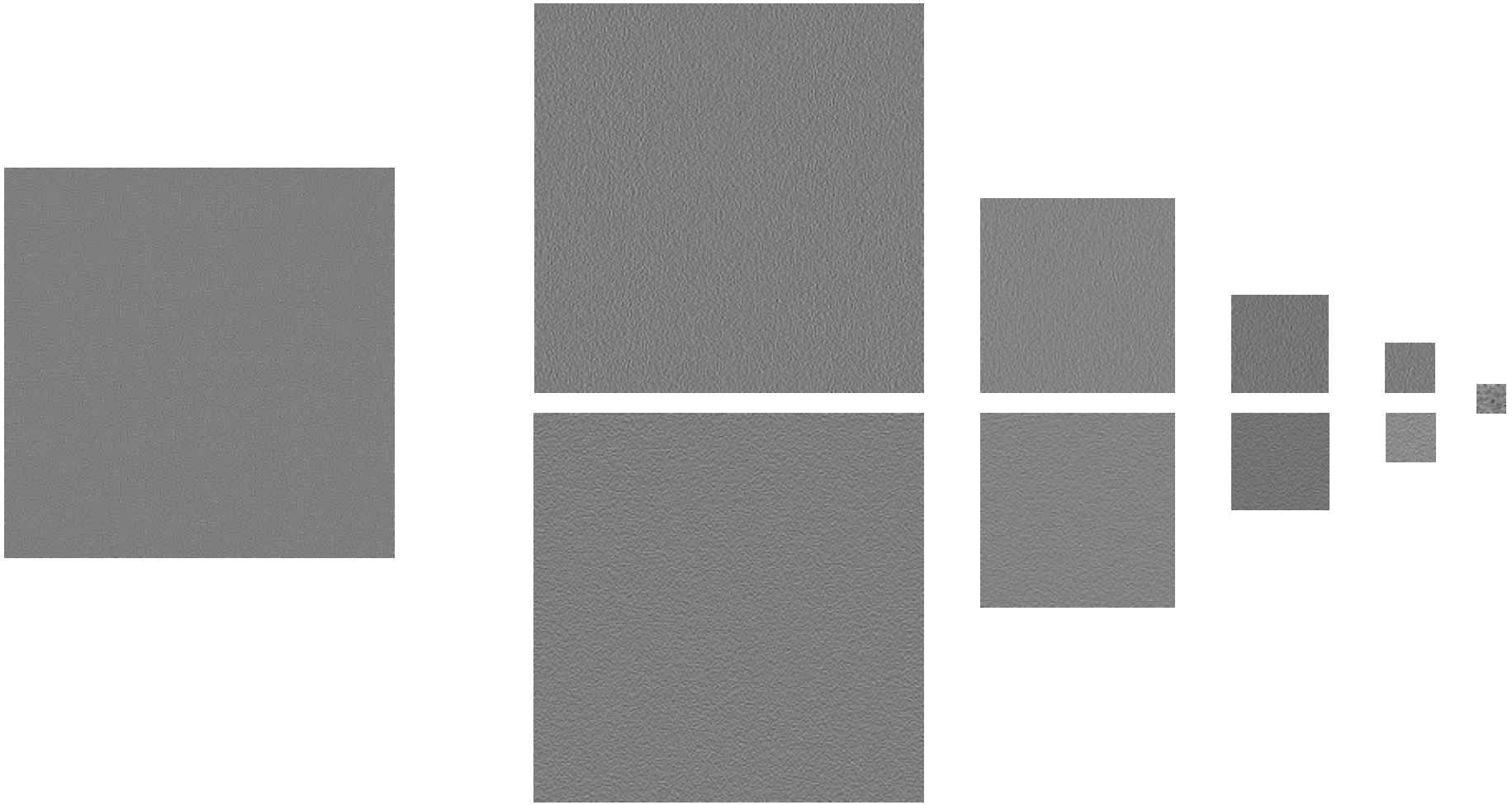
$$Var(x) = E(x^2) - E(x)^2$$

```
l_mean = conv(l)
l_var = conv(l*l) - l_mean*l_mean
```

Synthesis: Goal



Synthesis: normalised Noise Pyramid

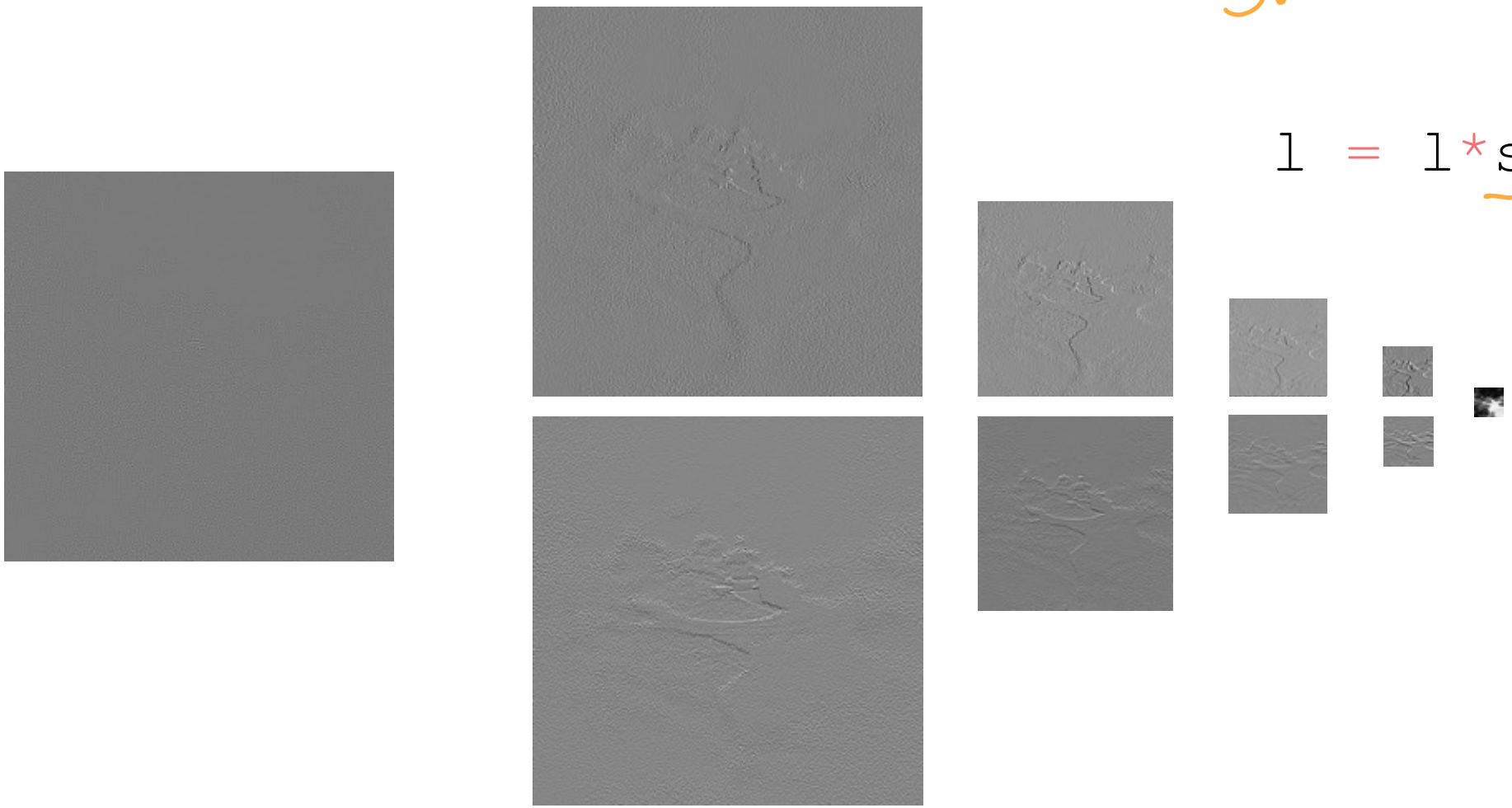


$$\mu = 0$$
$$\sigma^2 = 1$$

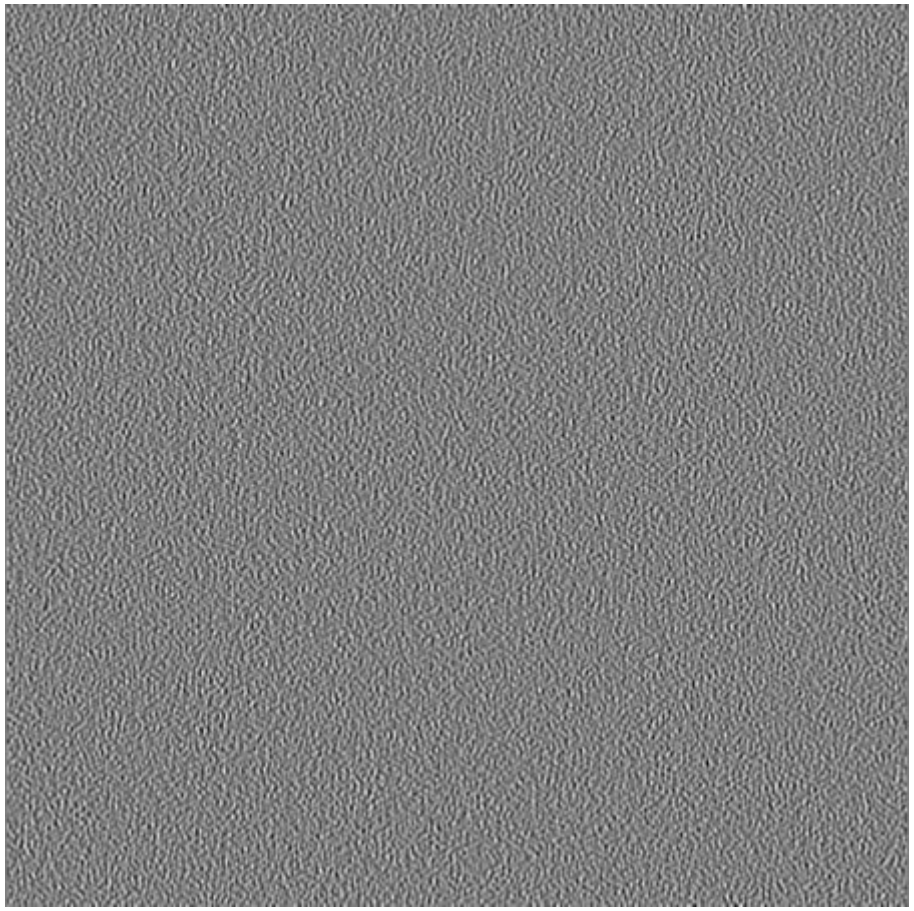
Synthesis: matching Noise Pyramid

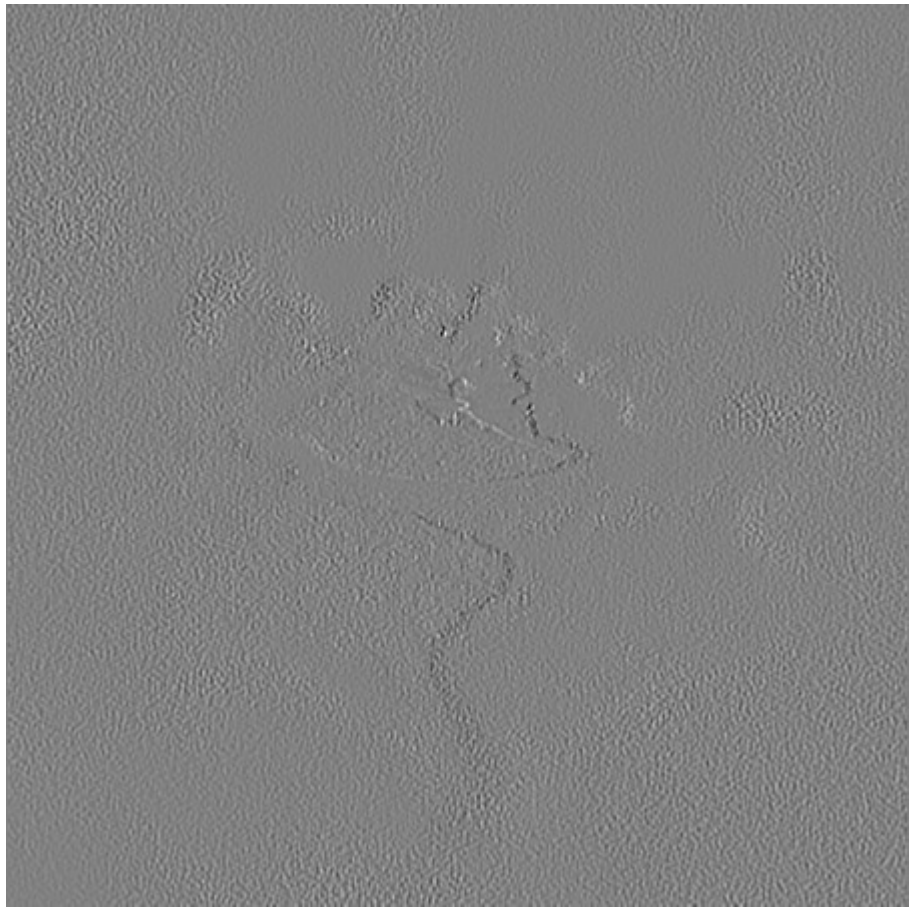
standard deviation

```
l = l*stdevs + means
```

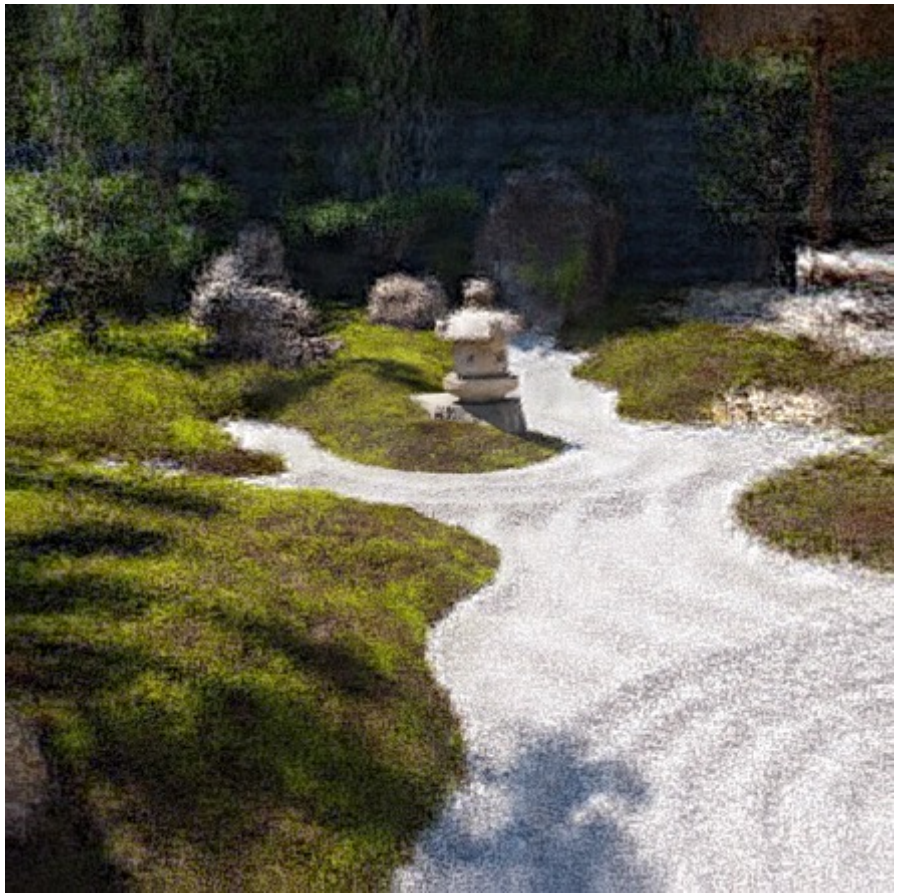


Before & after matching



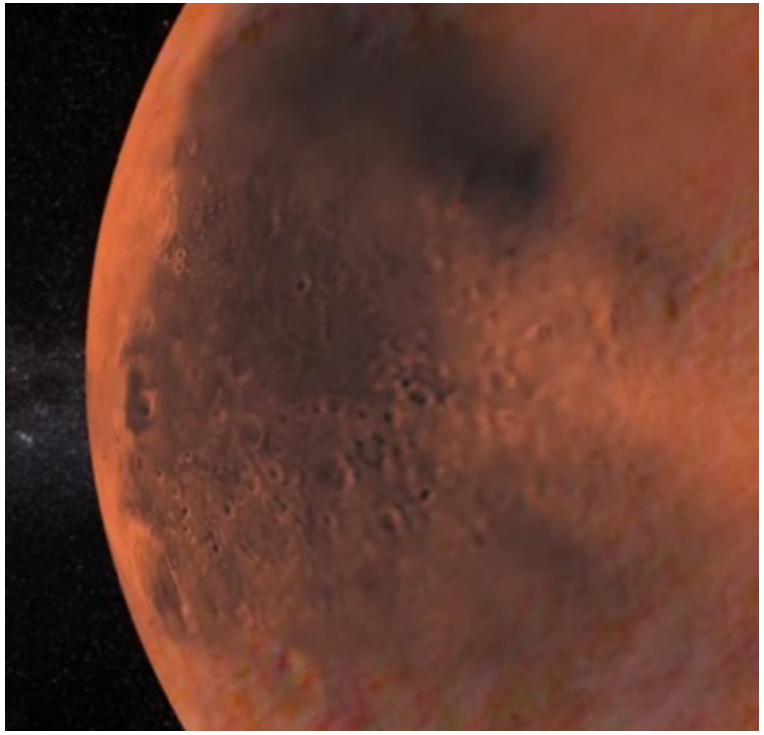


Synthesis: Result



Applications

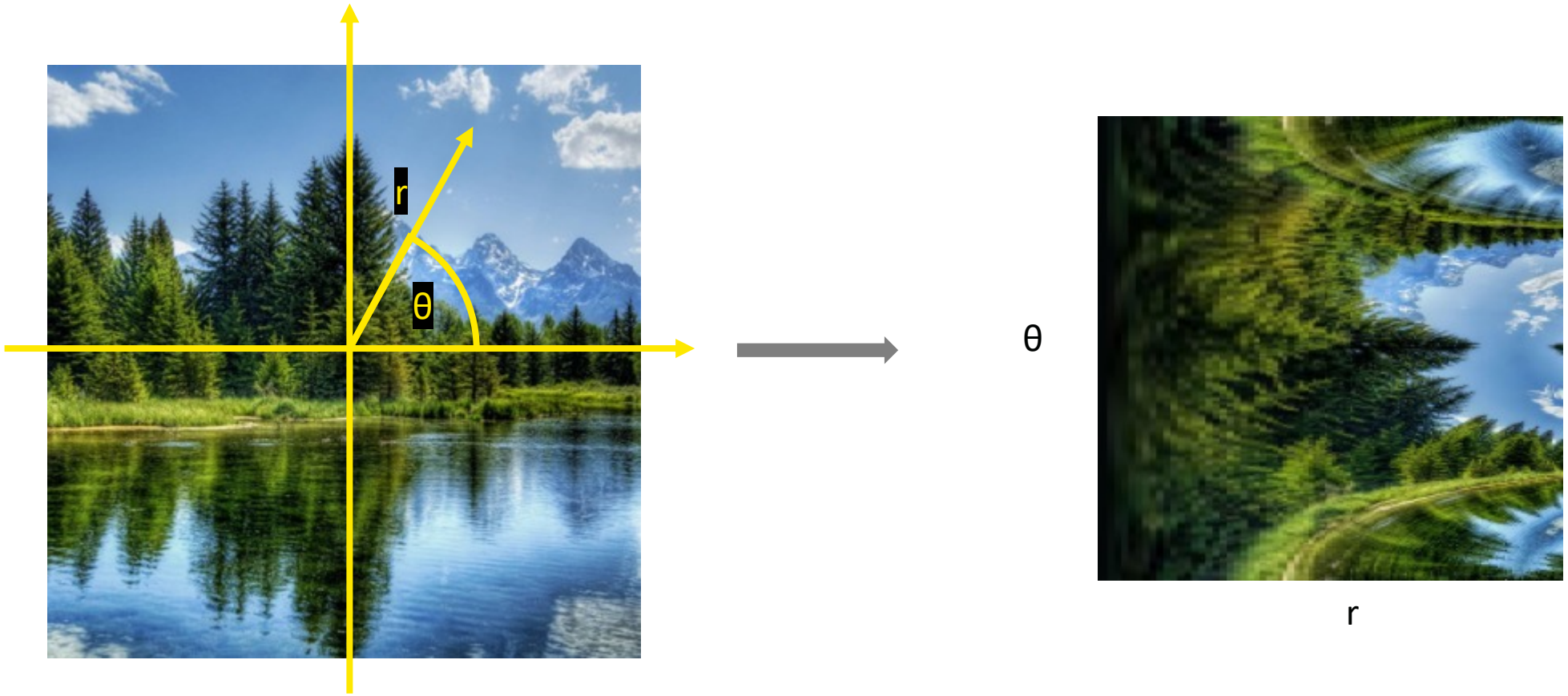
Compression



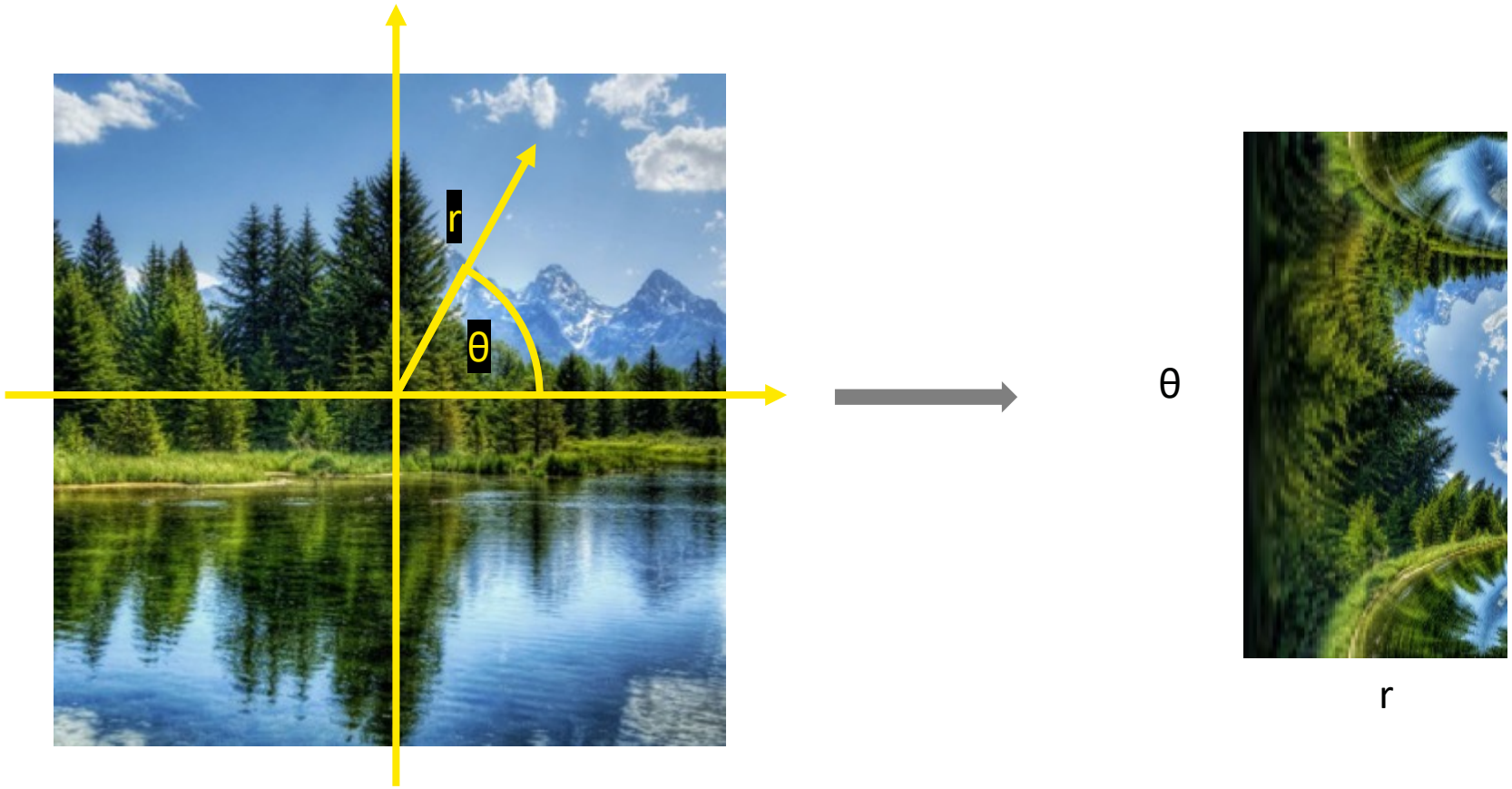
MIP Mapping

Denoising

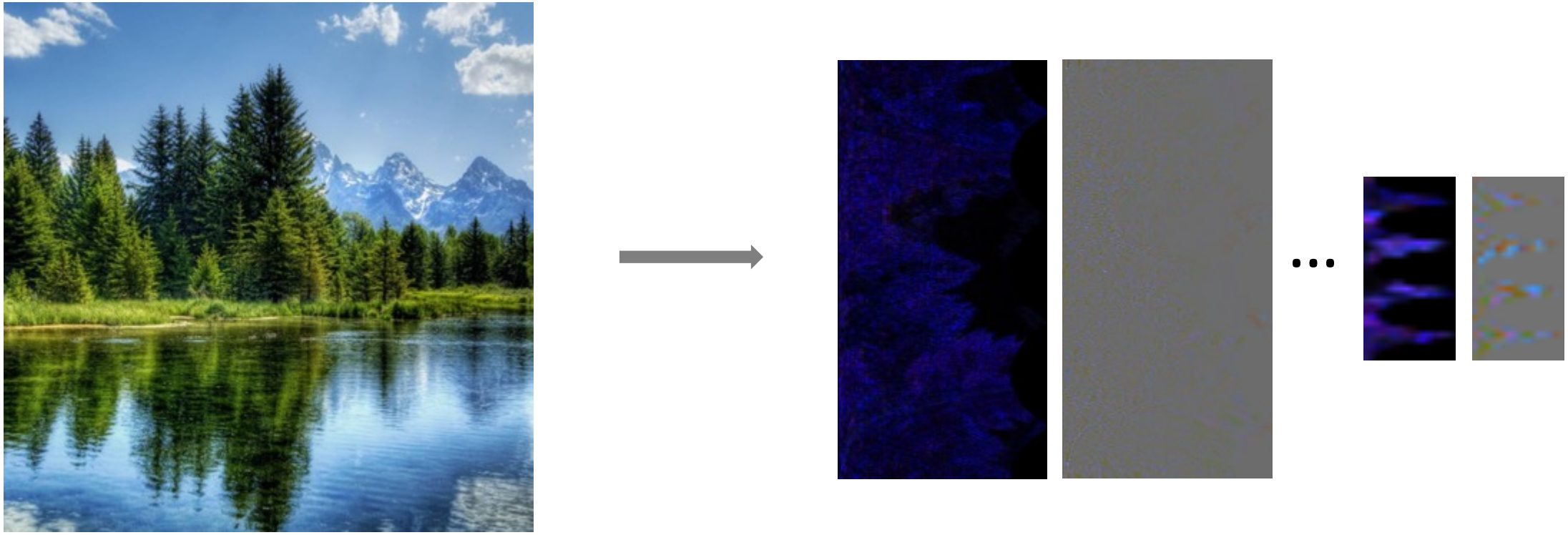
Compression



Compression



Compression

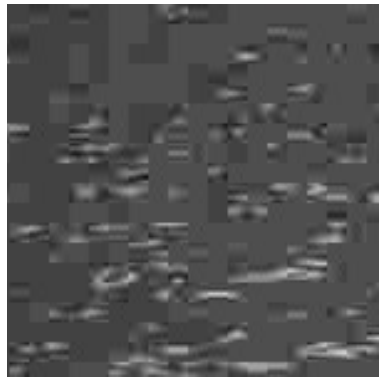


Compression: Results

Ours (40KB)

JPEG (40KB)

JPEG-of-Blur (40KB)



Further Applications

Light fields

Warping / inpainting

Holography

1

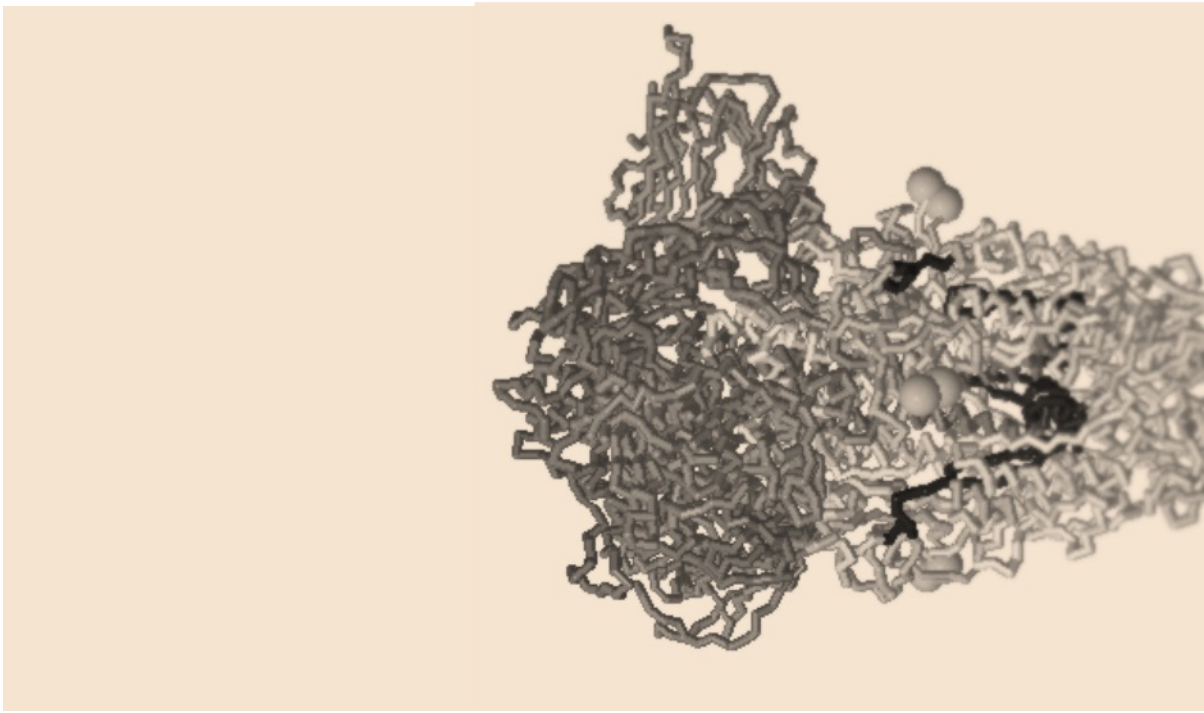
Laplacian Kernel Splatting for Efficient Depth-of-field and Motion Blur Synthesis or Reconstruction
T Leimkühler, H-P Seidel, T Ritschel
ACM Trans. Graph. (Proc. SIGGRAPH 2018) 37(3)

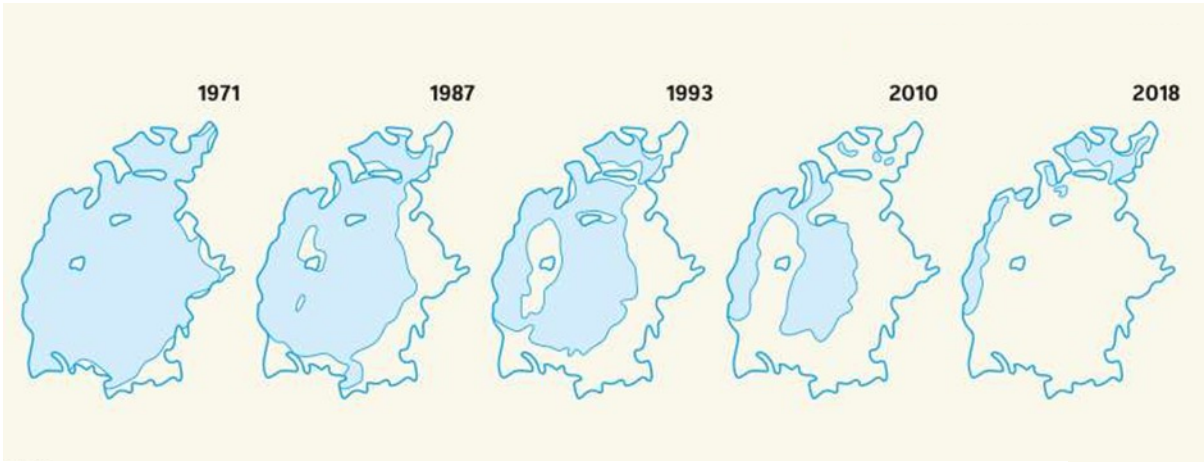
2

Perceptual Rasterization for Head-mounted Display Image Synthesis
S Friston, T Ritschel, A Steed
ACM Trans. Graph. (Proc. SIGGRAPH 2019) 38(3)

3

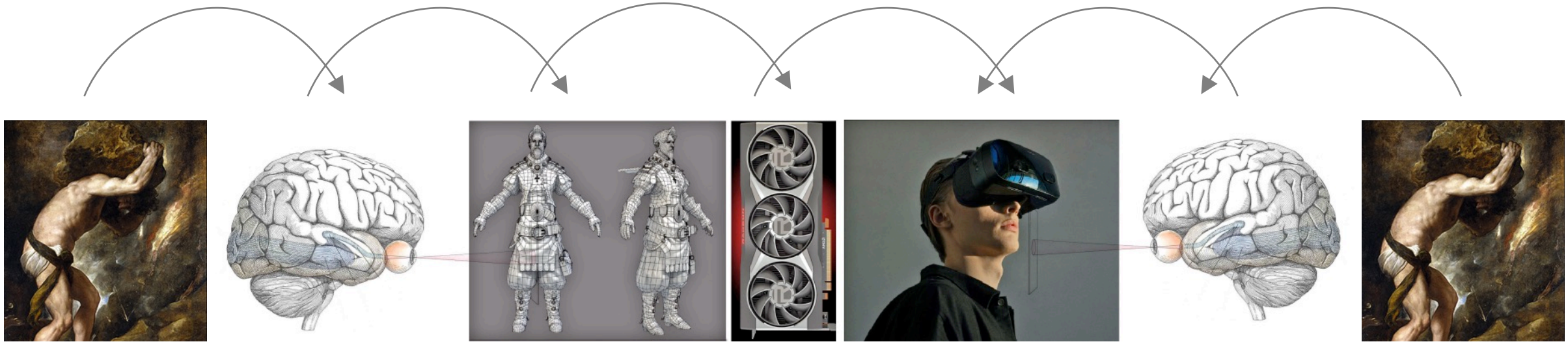
Beyond Blur: Real-time Metamers for Foveated Rendering
D Walton, R Kuffner-dos Anjos, S Friston, D Swapp, A Steed, T Ritschel
ACM Trans Graph (Proc. SIGGRAPH 2021) 40(3)





Sisyphus, *Titian* (1548—49)
SARS Cov-19 spike protein, *SimMod* (2021)
Aral lake, *Westermann Weltatlas* (2020)

$$\arg\min_{\mathcal{R}} |\mathcal{T}(\mathcal{P}(\mathcal{D}(\mathcal{R}(C)))) - \mathcal{T}(\mathcal{P}(I))|_2$$



Task	Perception	Content	Rendering	Display	Perception	Task
$\mathcal{T}$	$\mathcal{P}$	C	$\mathcal{R}$	$\mathcal{D}$	$\mathcal{P}$	$\mathcal{T}$

Thanks to ..

- .. the attendees
- .. the organizers for the first real conference I attended in more than two years
- .. Rafal and Christian and the CVMP committee for inviting me
- .. all my collaborators on these projects *(in order of appearance)*

David Walton (UCL), Rafael Kuffner dos Anjos (UCL), Sebastian Friston (UCL), Anthony Steed (UCL), Thomas Leimkühler (MPI, Hans-Peter Seidel (MPI), Kaan Aksti (UCL), David Swapp (UCL)

Shameless plug

- We have two post-doc positions available on the related projects
- One closes tomorrow
- Contact Tobias

Contact:

t.ritschel@ucl.ac.uk
www.ucl.ac.uk/~ucactri