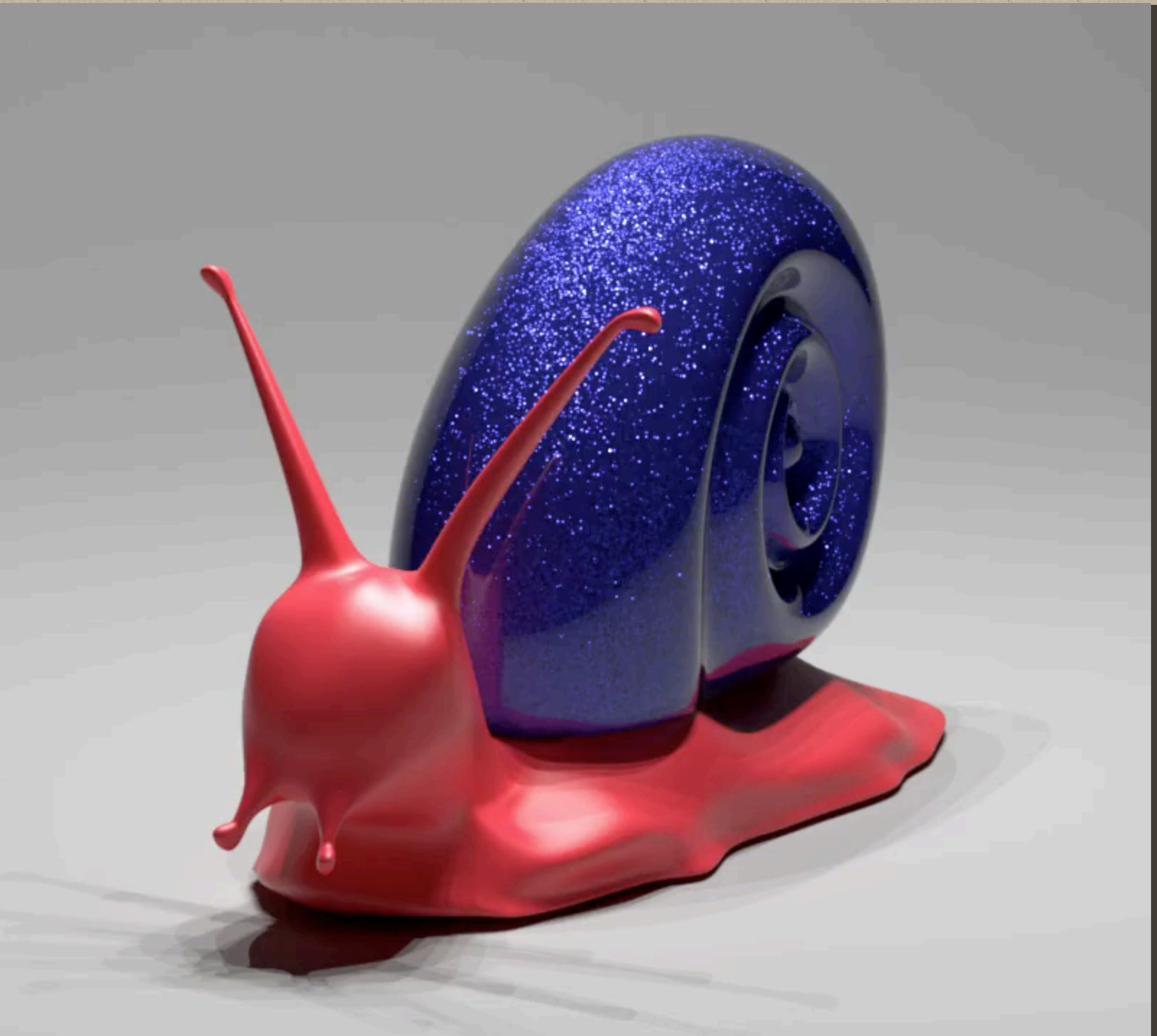
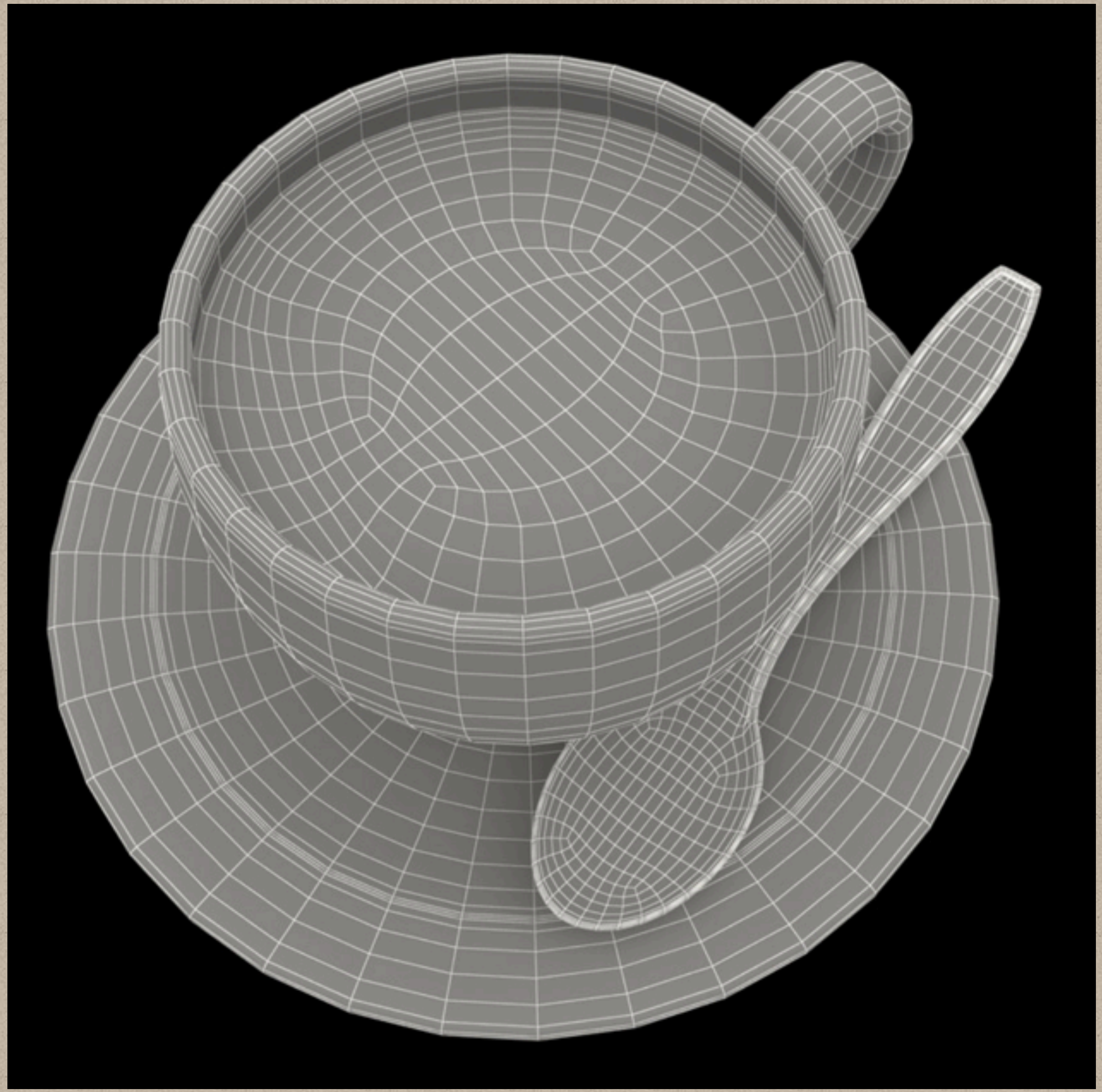
Ling-Qi's
FACULTY TALK
AT PURDUE UNIVERSITY



Physically-based Modeling and Rendering of Complex Visual Appearance

Ling-Qi Yan
University of California, Berkeley

What is Rendering?



3D scene (meshes, lights, etc.)

Calculating light -> eye

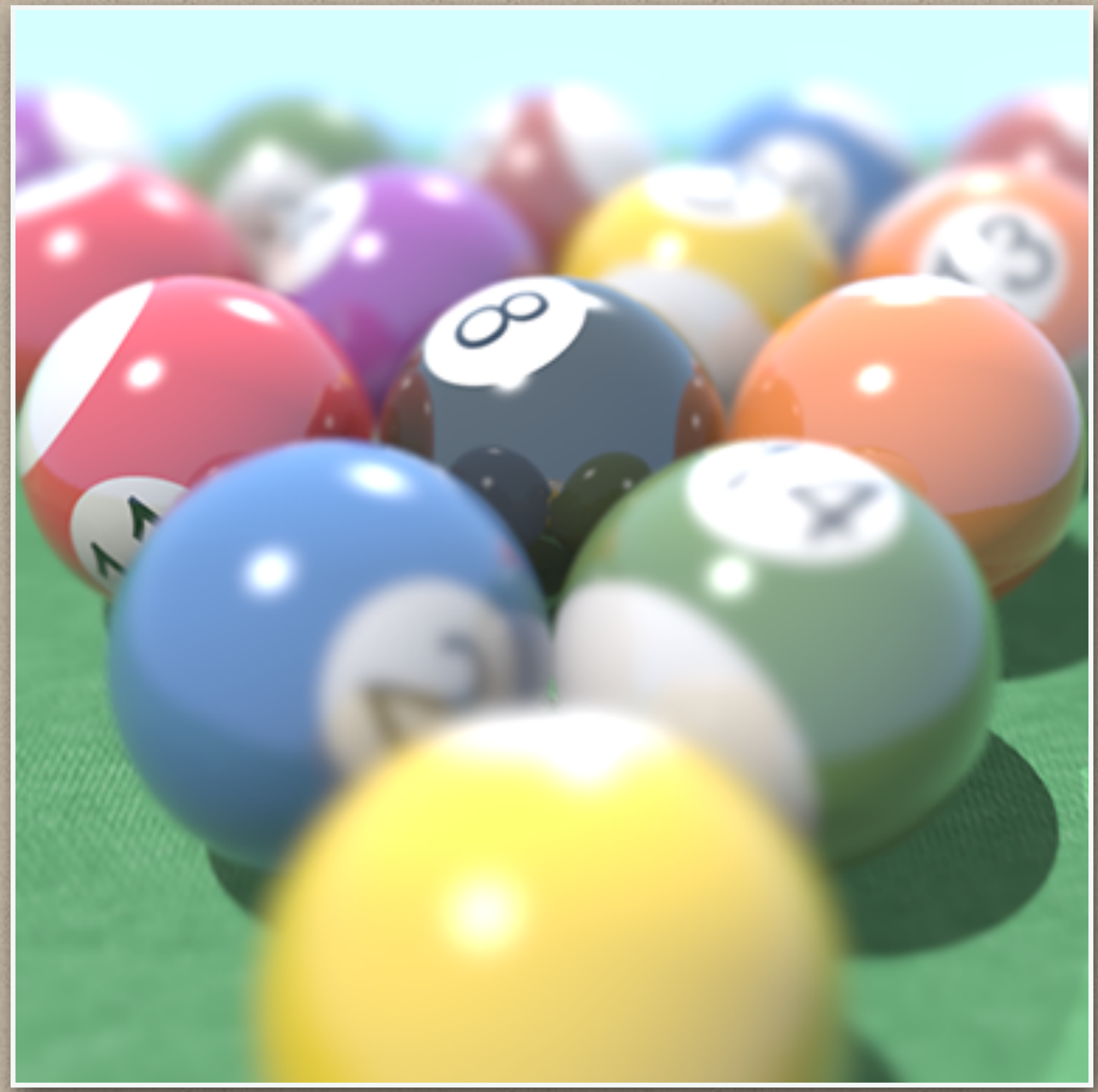
Image

Outline

- My research (Computer Graphics - Rendering)
  - High level goal: **realism** and **speed**

detailed rendering

detailed appearance modeling



real-time ray tracing

- Future research directions

Part I: Detailed Rendering

detailed rendering

detailed appearance modeling

real-time ray tracing

Rendering is powerful today

Car rendered in NVIDIA Iray

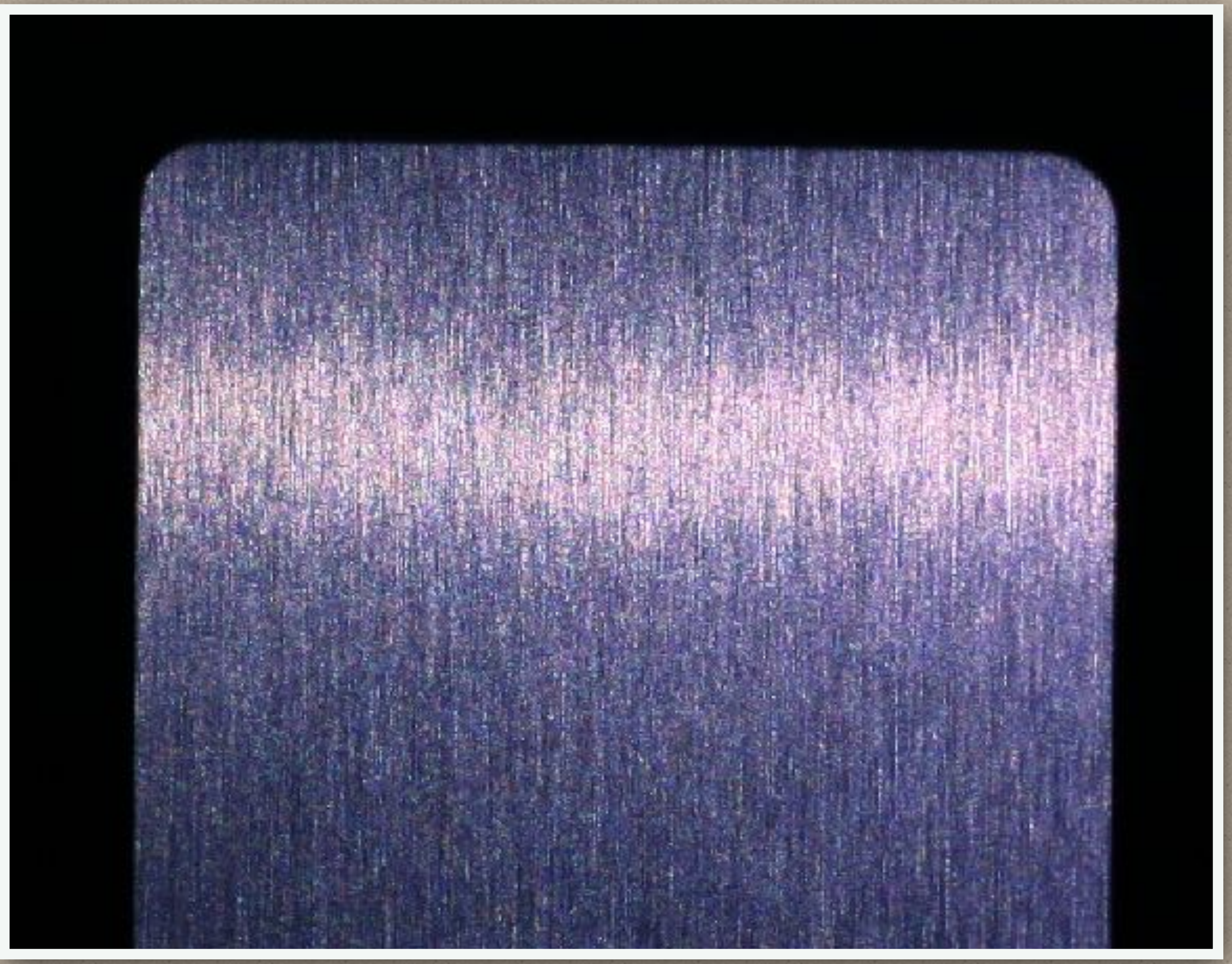
Mouse rendered in Autodesk 3DS Max

Real world is more complicated

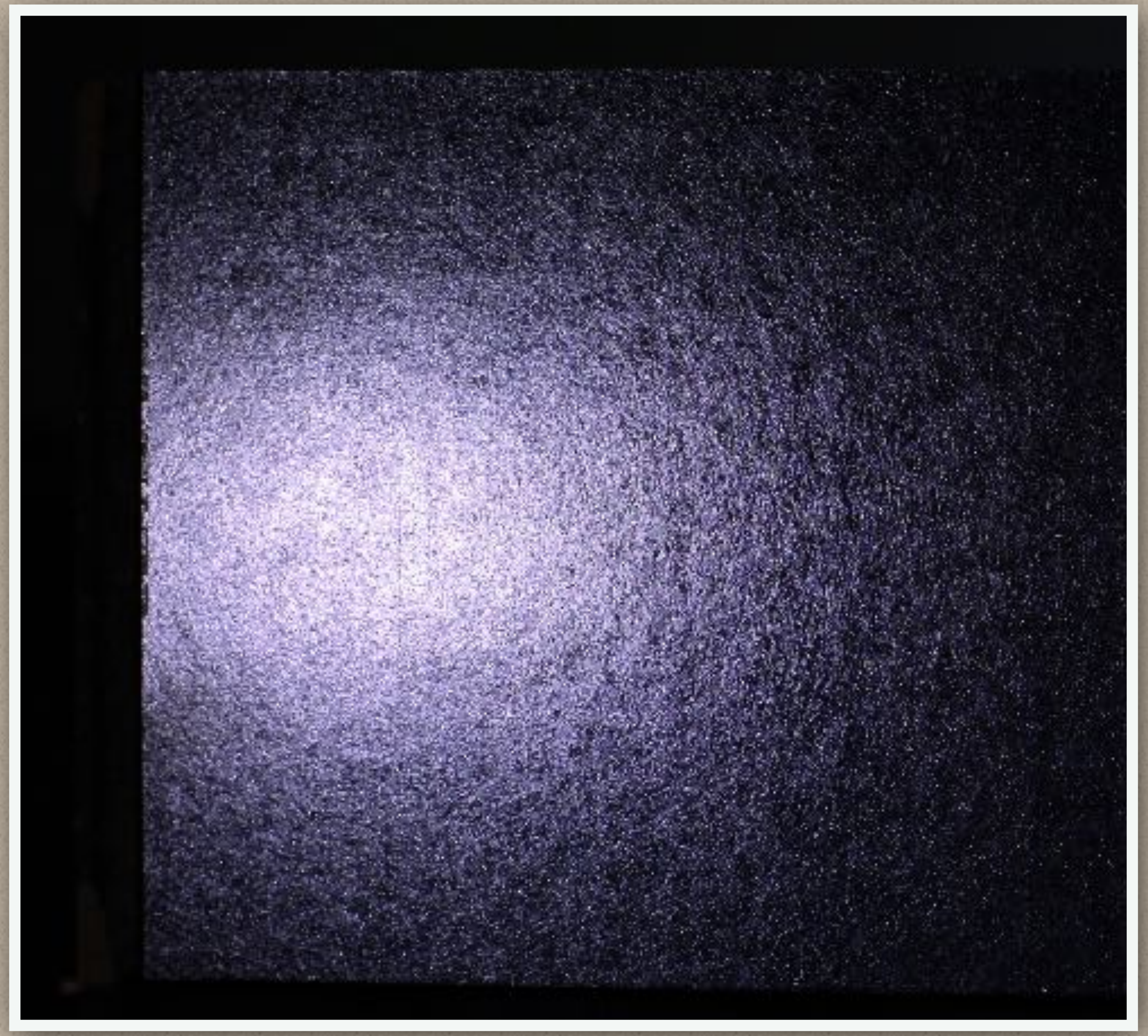
Real photograph of a car

Real video of a mouse

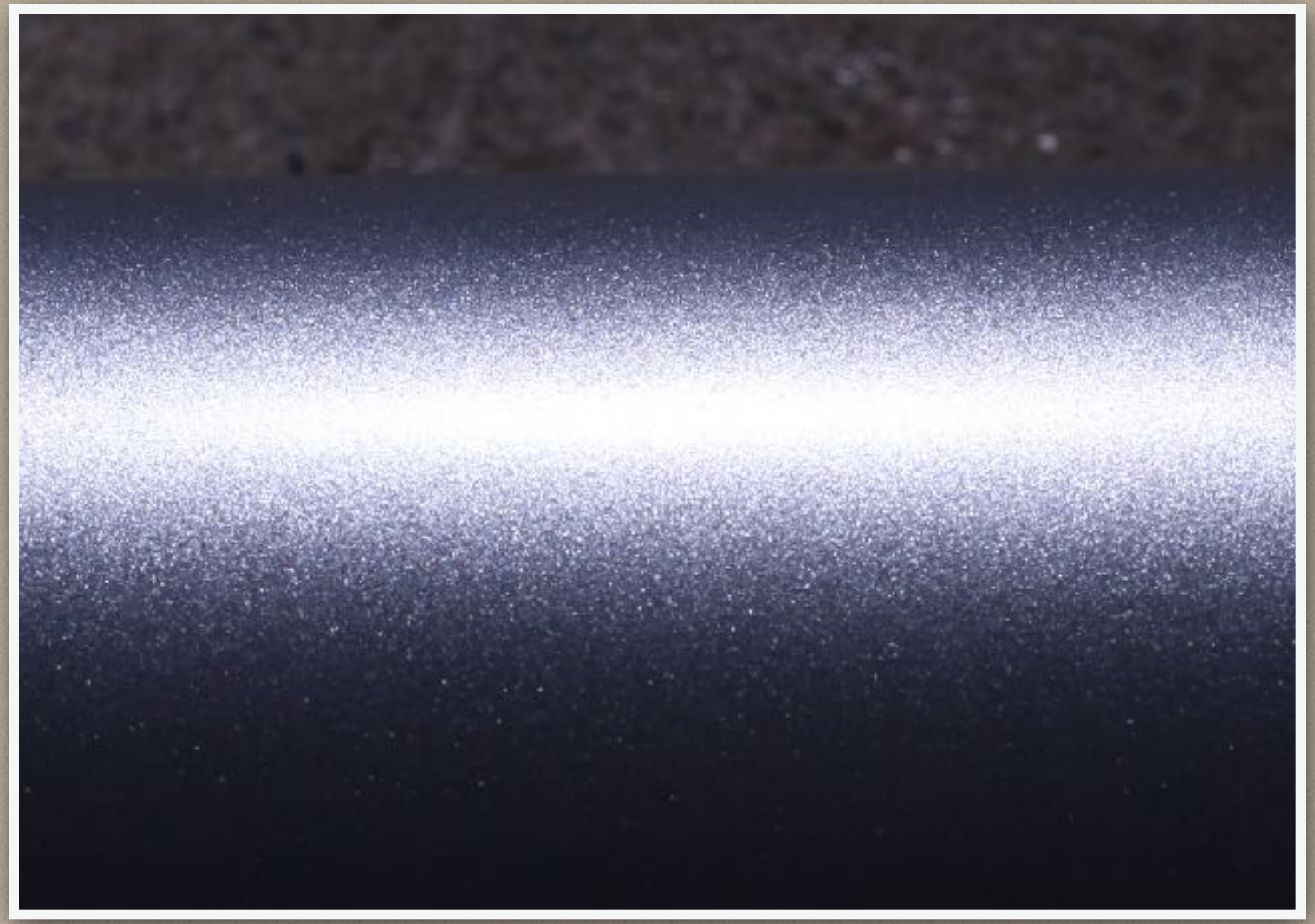
More real world photos



brushed metal



laminate



powder coating

Why details?

traditional
model

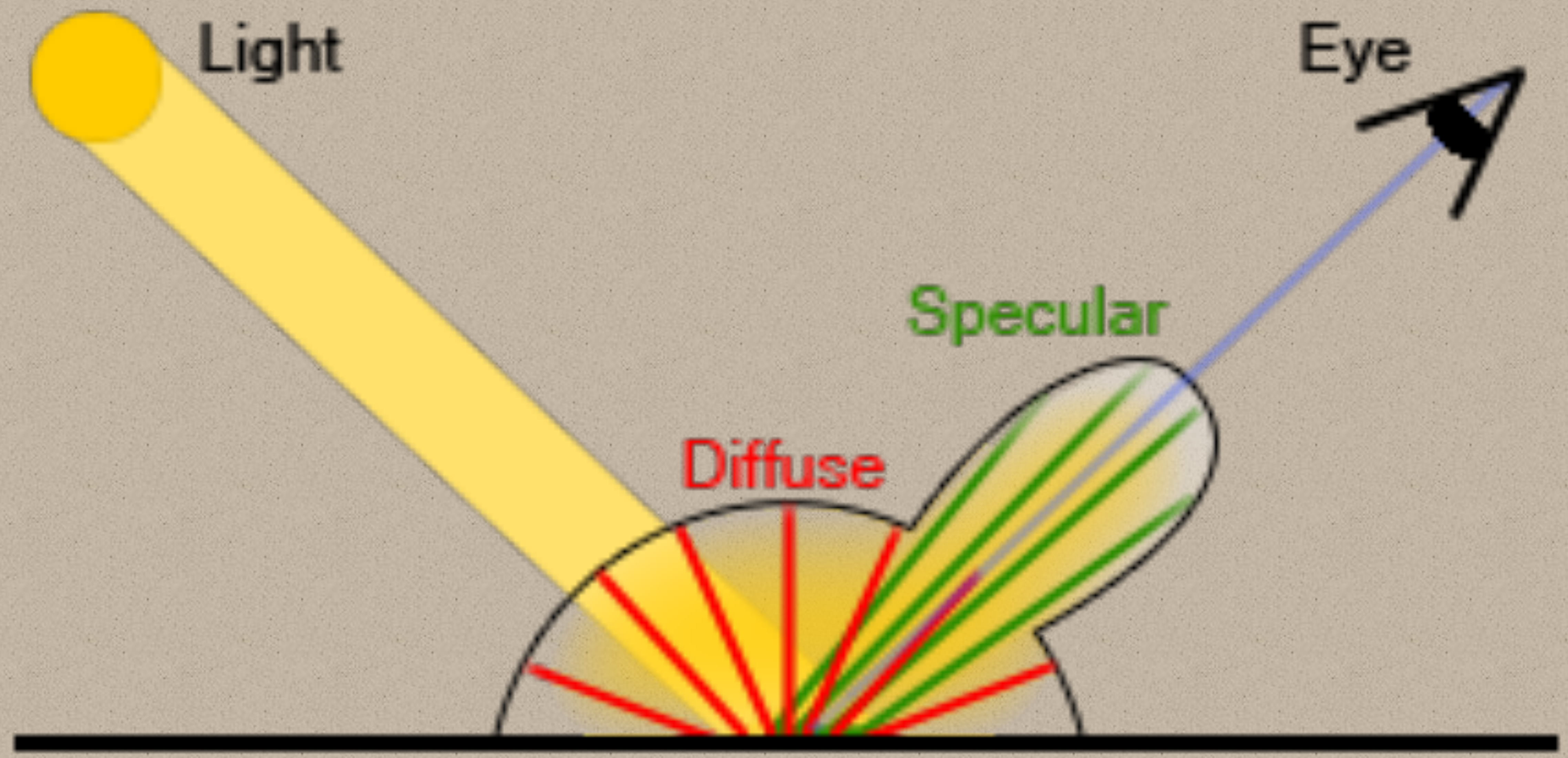
Why details?

our
model

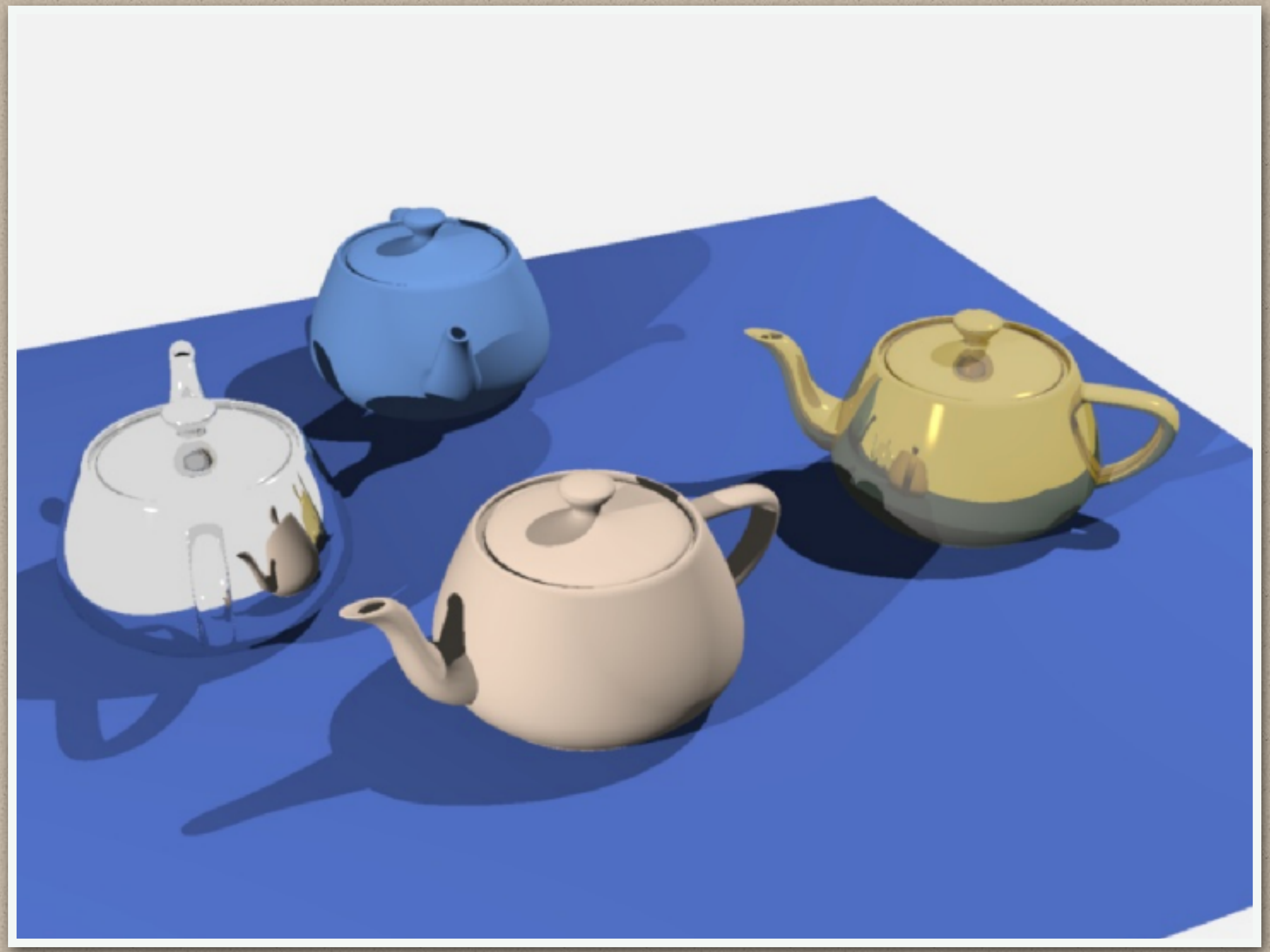
Why details?

our
model

Recap: traditional BRDF*



A typical BRDF (Phong model)

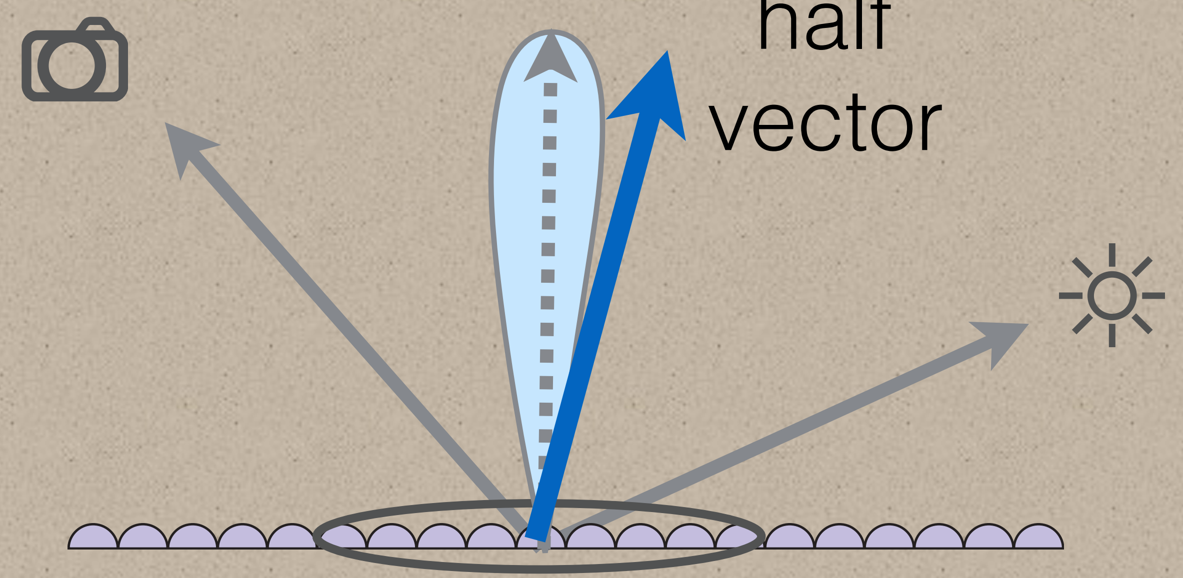


unpleasing, artificial results

*: Bidirectional Reflectance Distribution Function

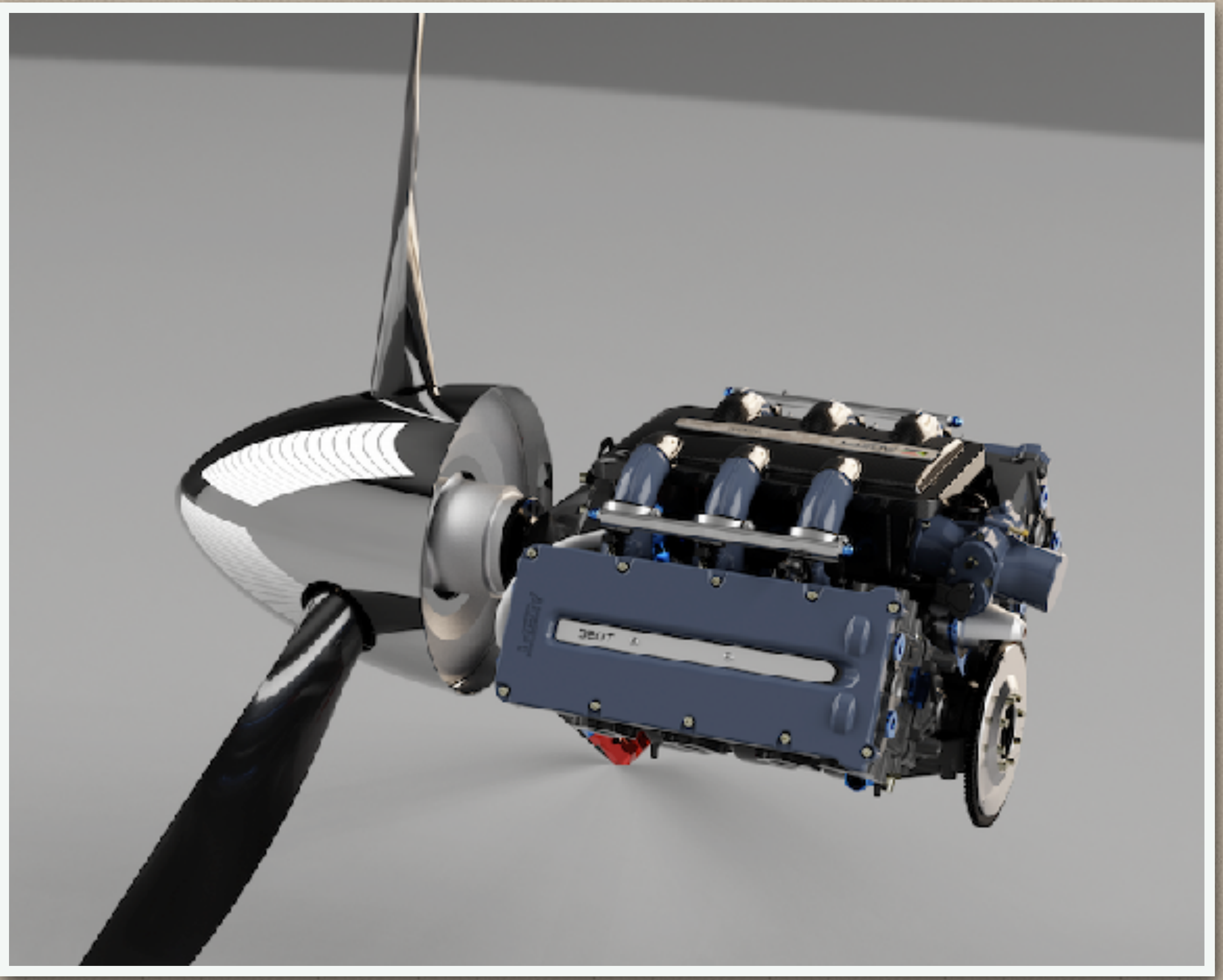
Advanced model:

Microfacet BRDF



Surface = Specular microfacets + Random normals

$$f(\mathbf{i},\mathbf{o}) = \frac{\mathbf{F}(\mathbf{i},\mathbf{h})\mathbf{G}(\mathbf{i},\mathbf{o},\mathbf{h})\mathbf{D}(\mathbf{h})}{4(\mathbf{n},\mathbf{i})(\mathbf{n},\mathbf{o})}$$

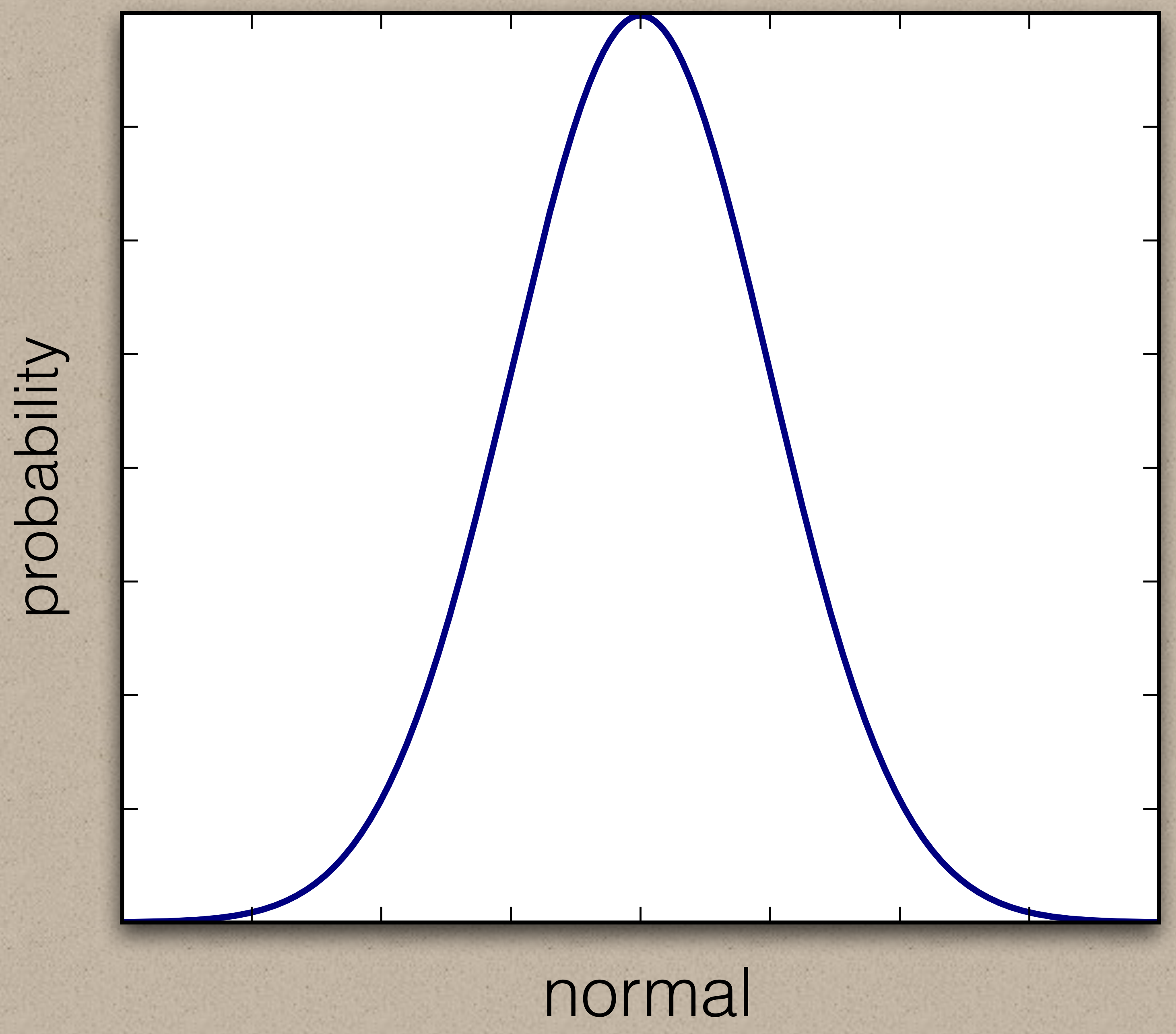


pleasing results

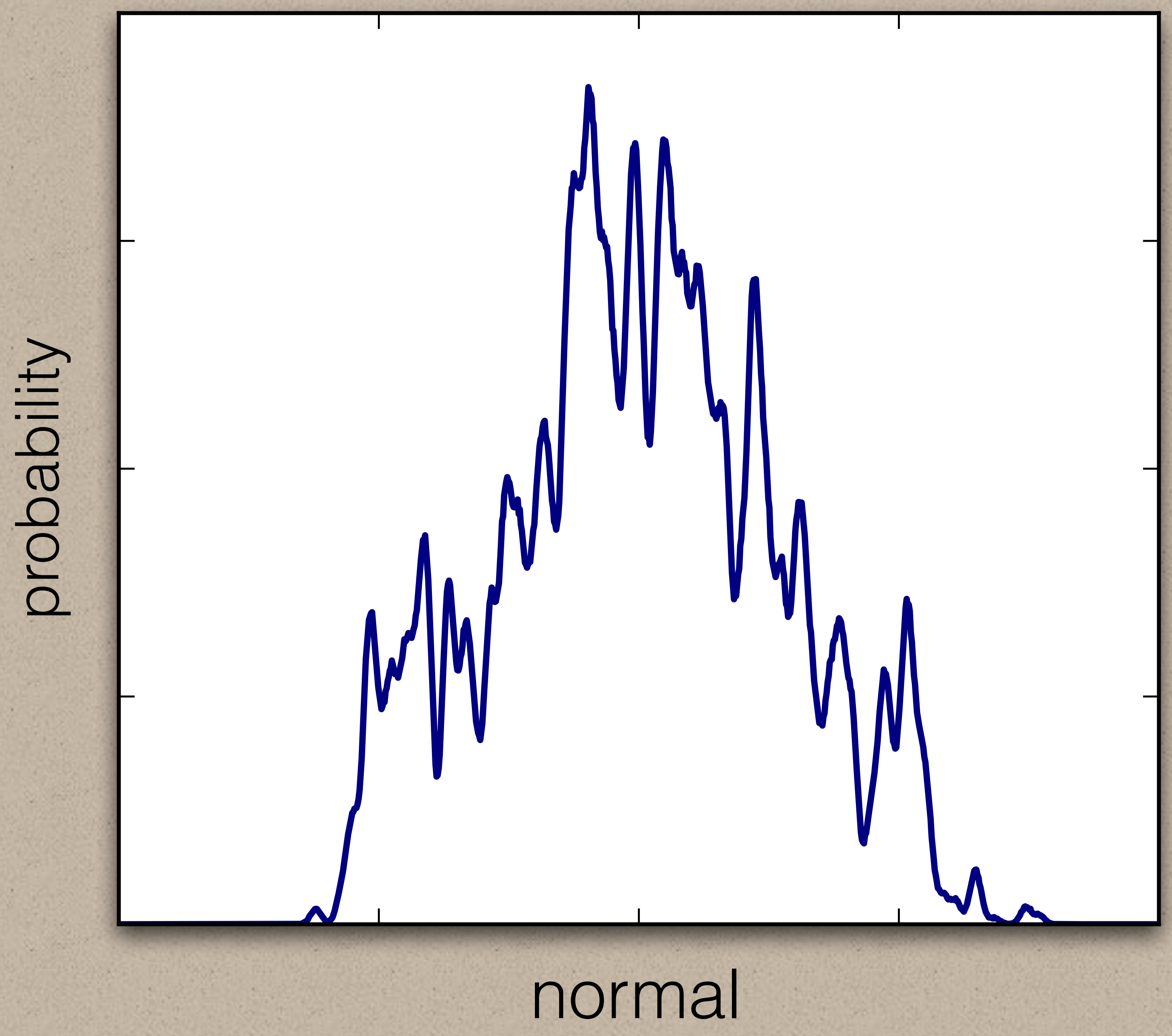
Normal Distribution: ideal vs. real

Surface = Specular microfacets +
Random normals

Normal Distribution Function (NDF)

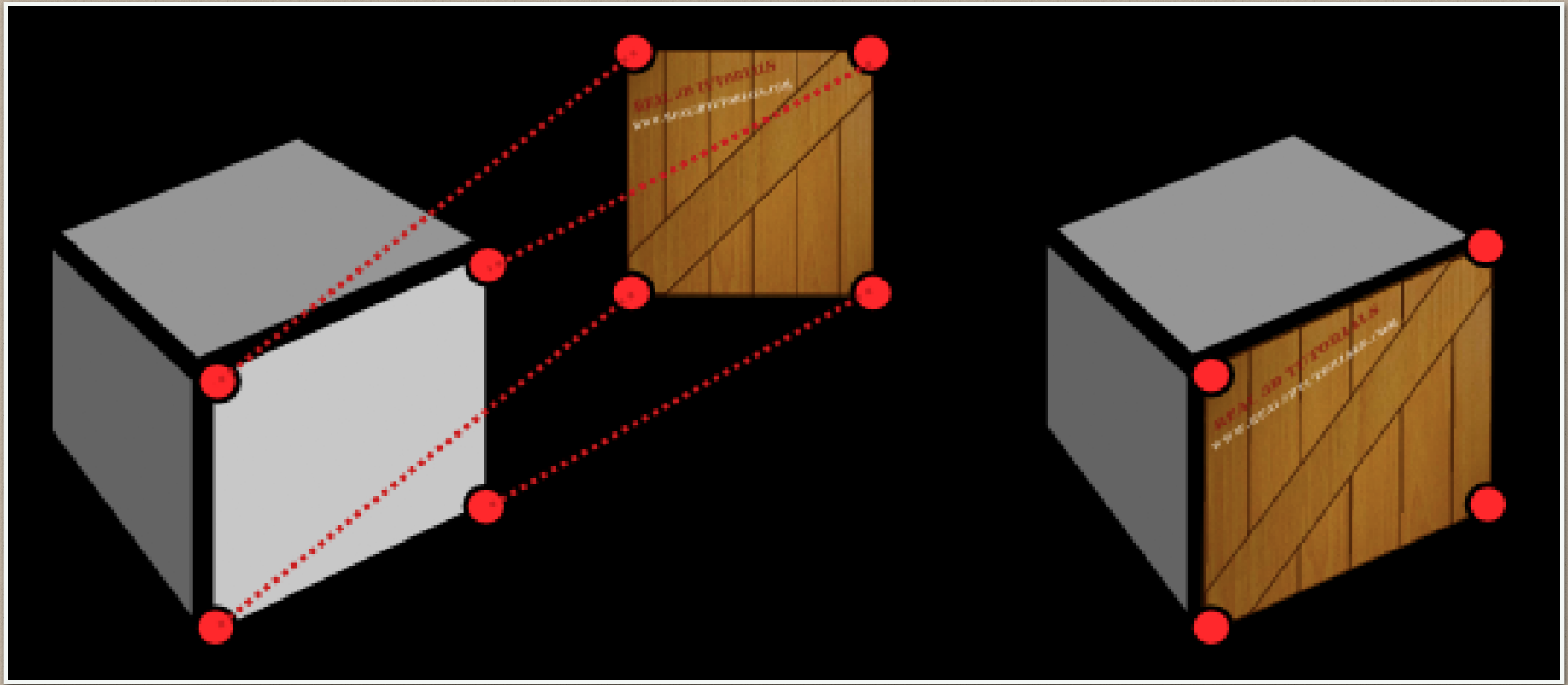


What we have

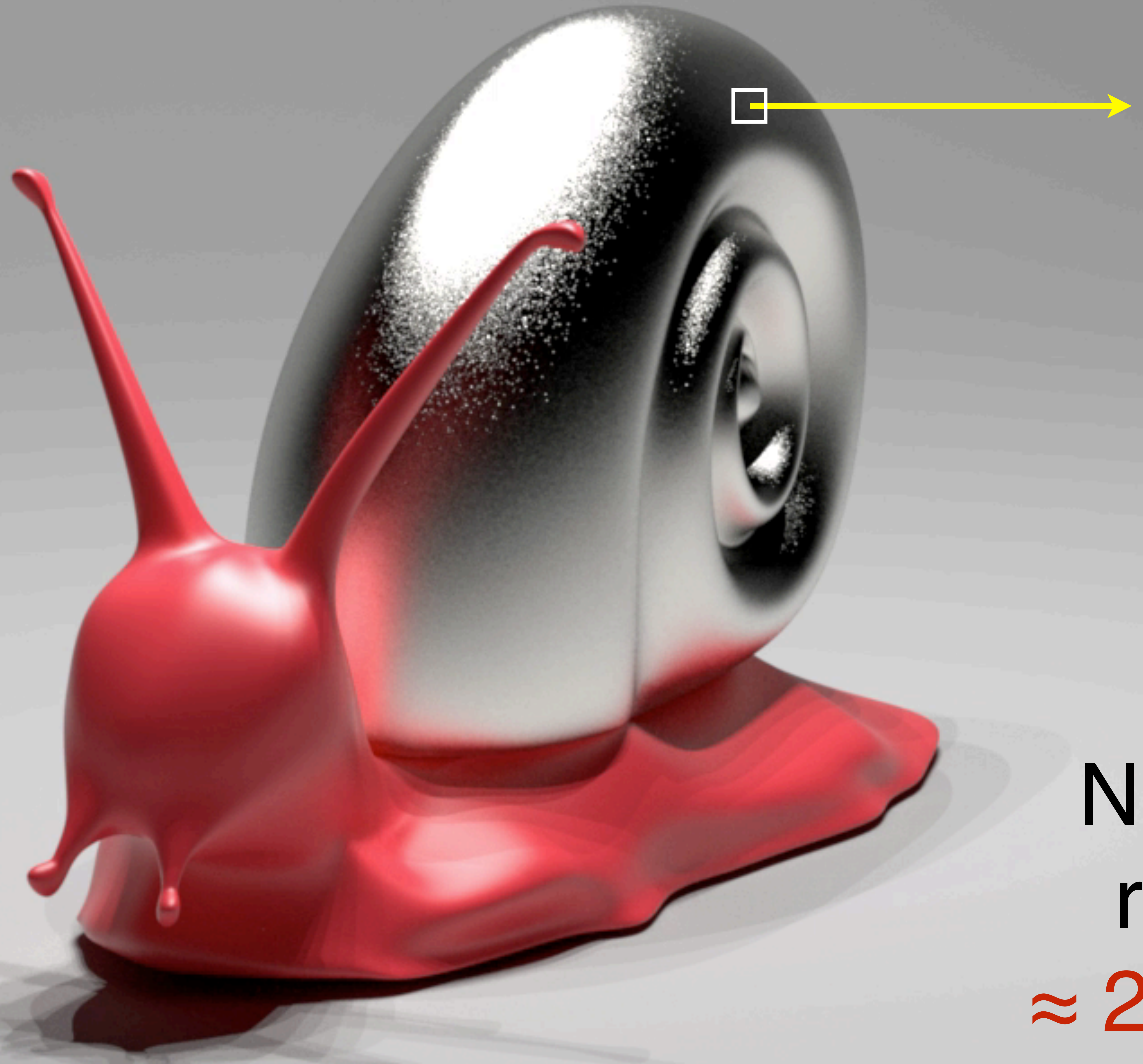


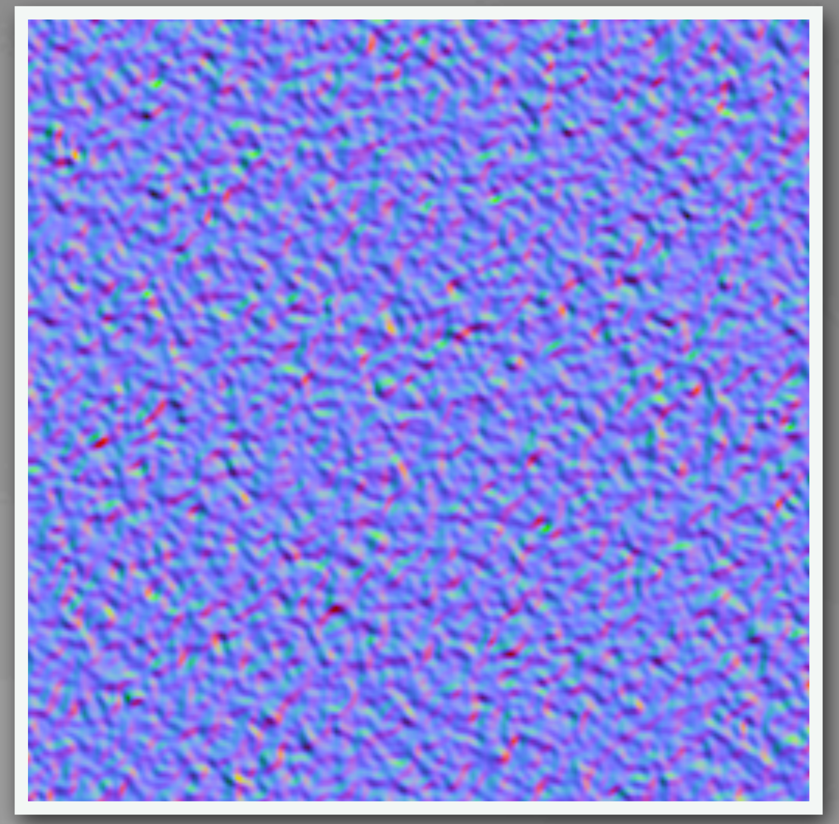
What we expect

Recap: texture mapping



Define the details



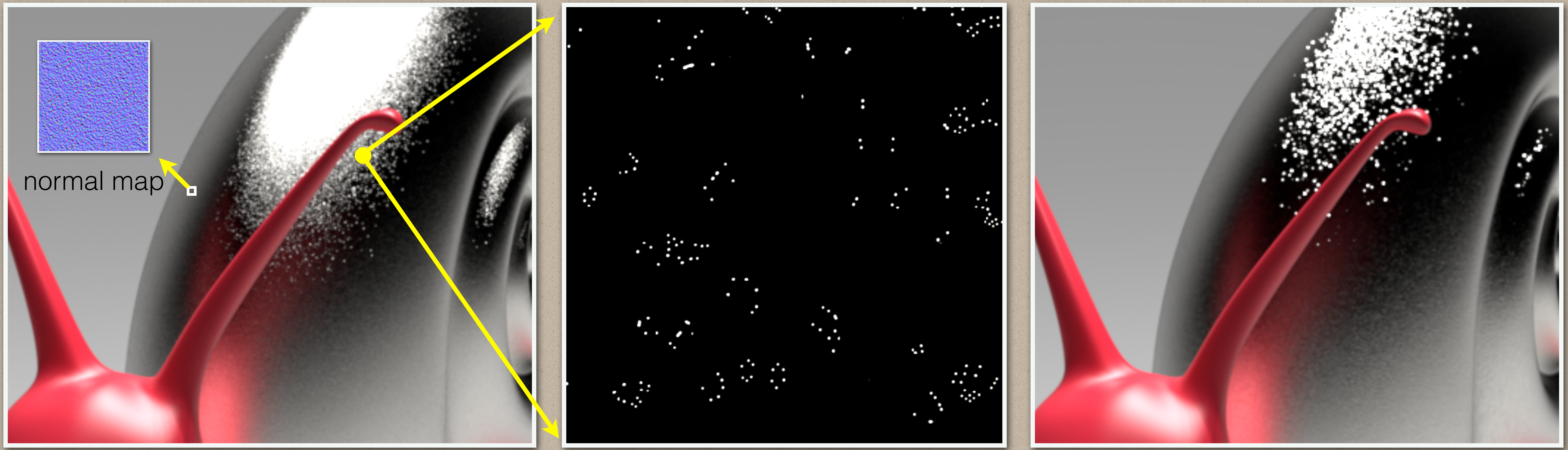


isotropic noise normal map

Normal map resolution:

≈ 200K x 200K

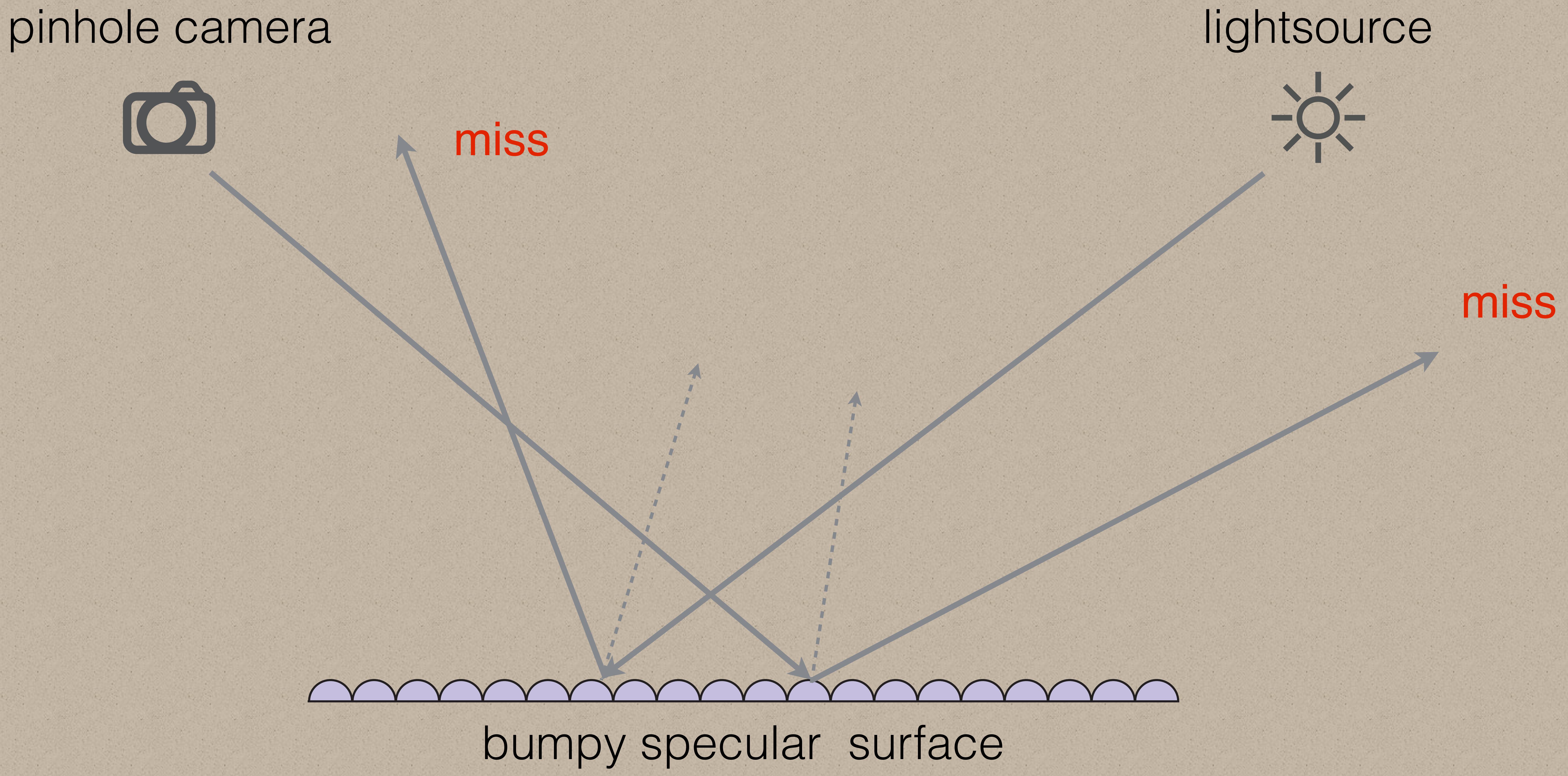
Rendering? Too difficult!



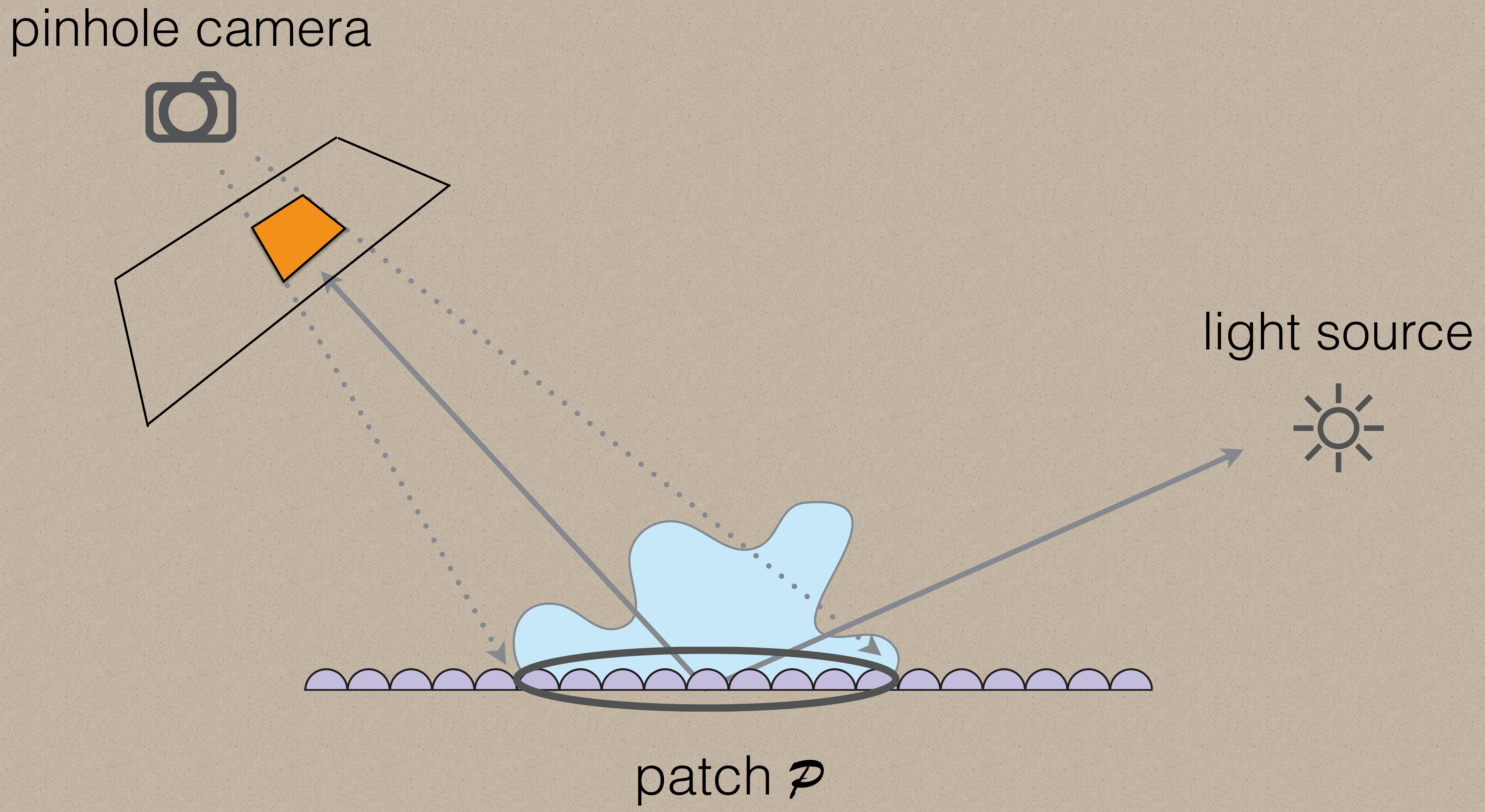
our result | zoom of a single pixel | naive sampling (2 hours)

(≫ 21.3 days to converge)

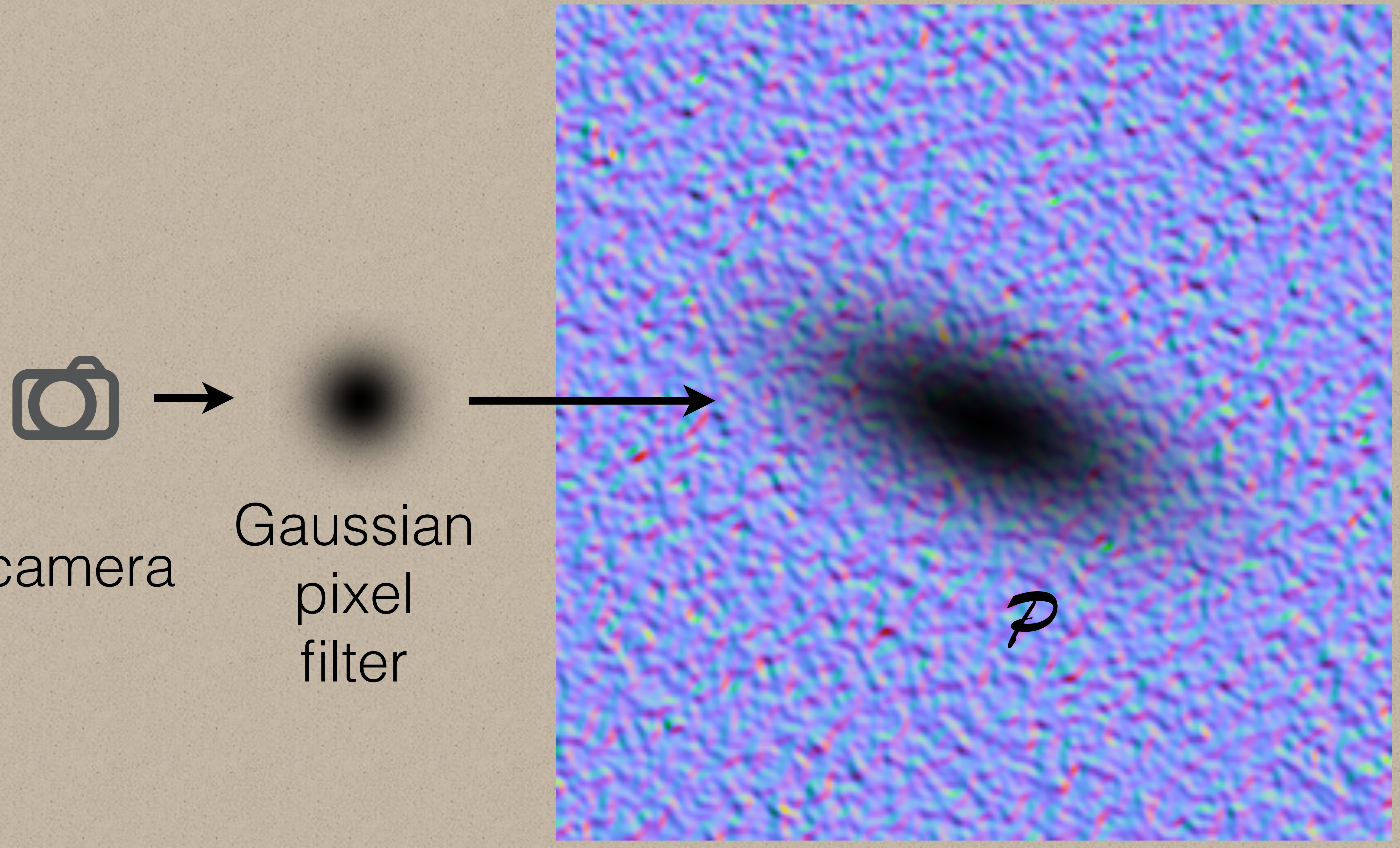
Difficult path sampling problem



Solution: BRDF over a pixel



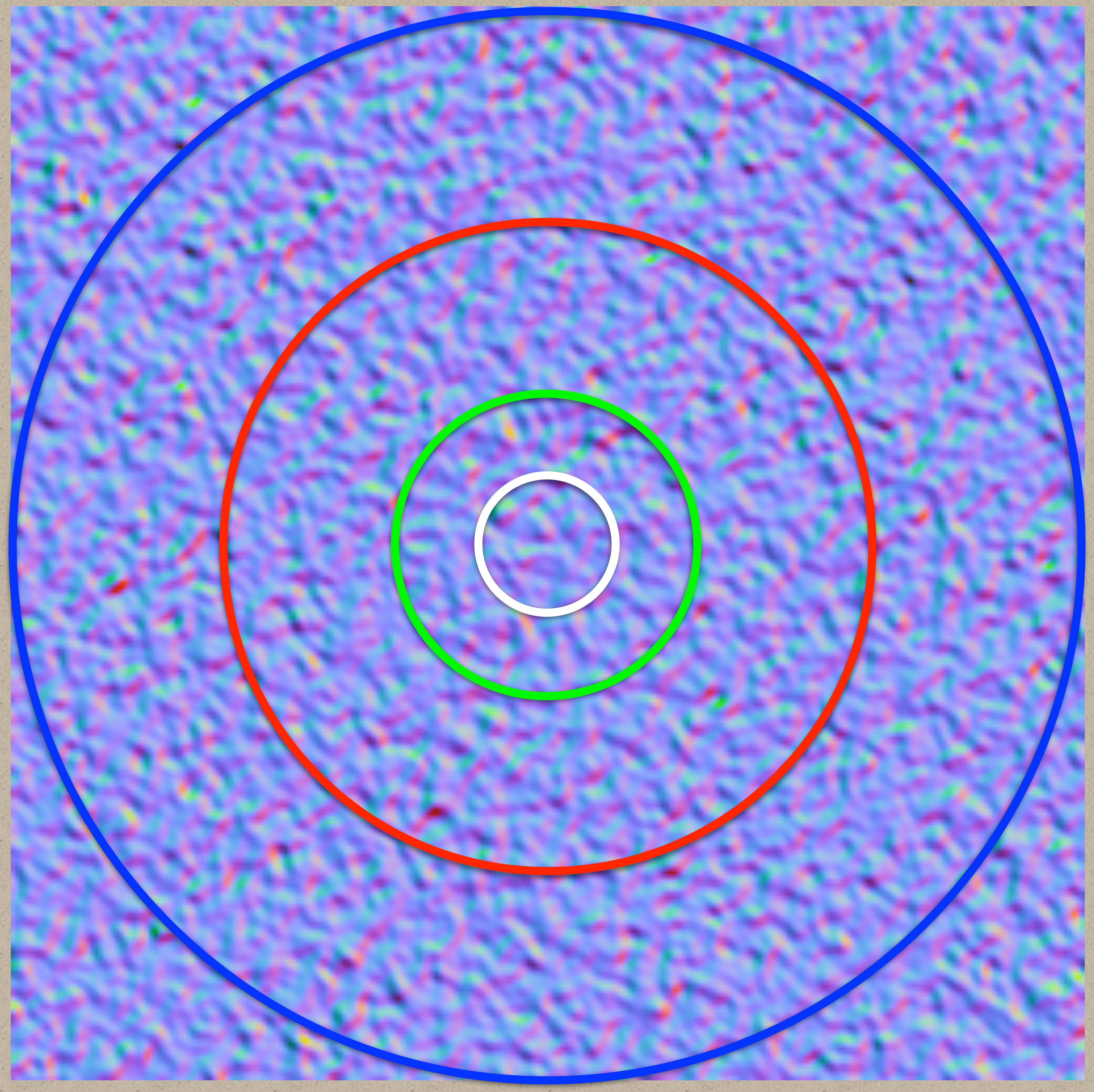
NDF over a pixel



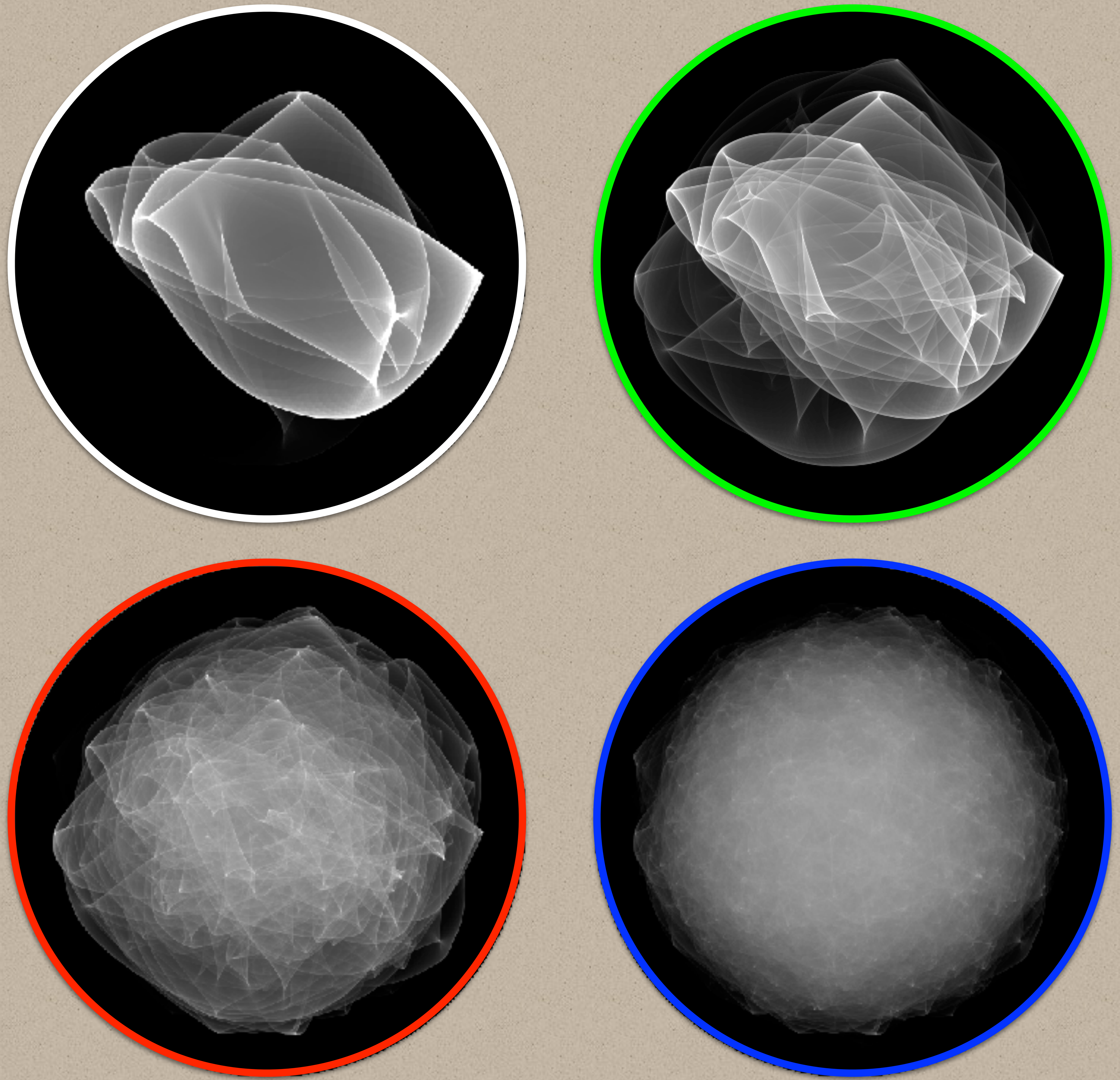
normal-mapped
surface

$\mathcal{P}$-NDF:
distribution of normals over $\mathcal{P}$

NDFs have sharp features

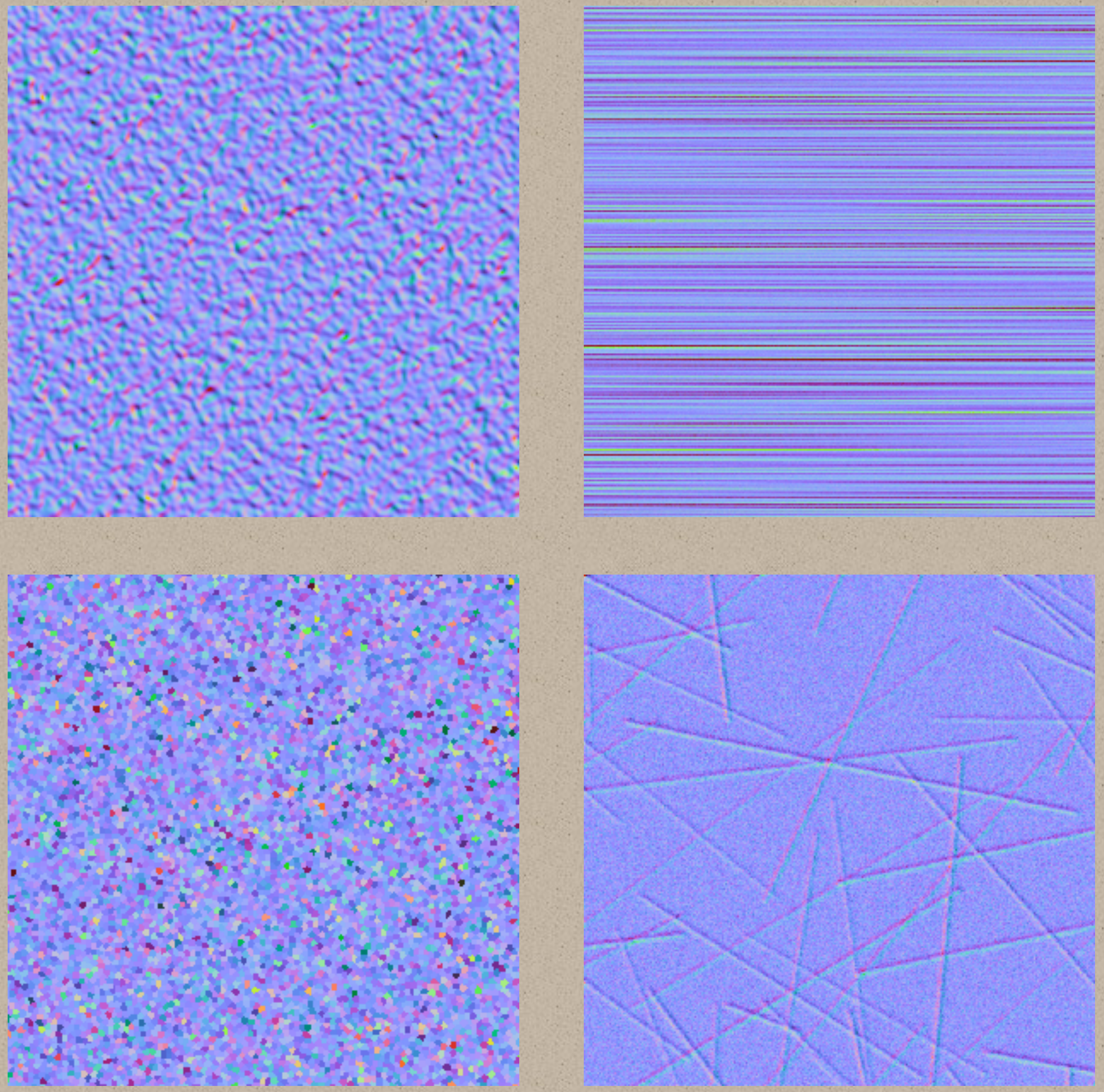


normal map

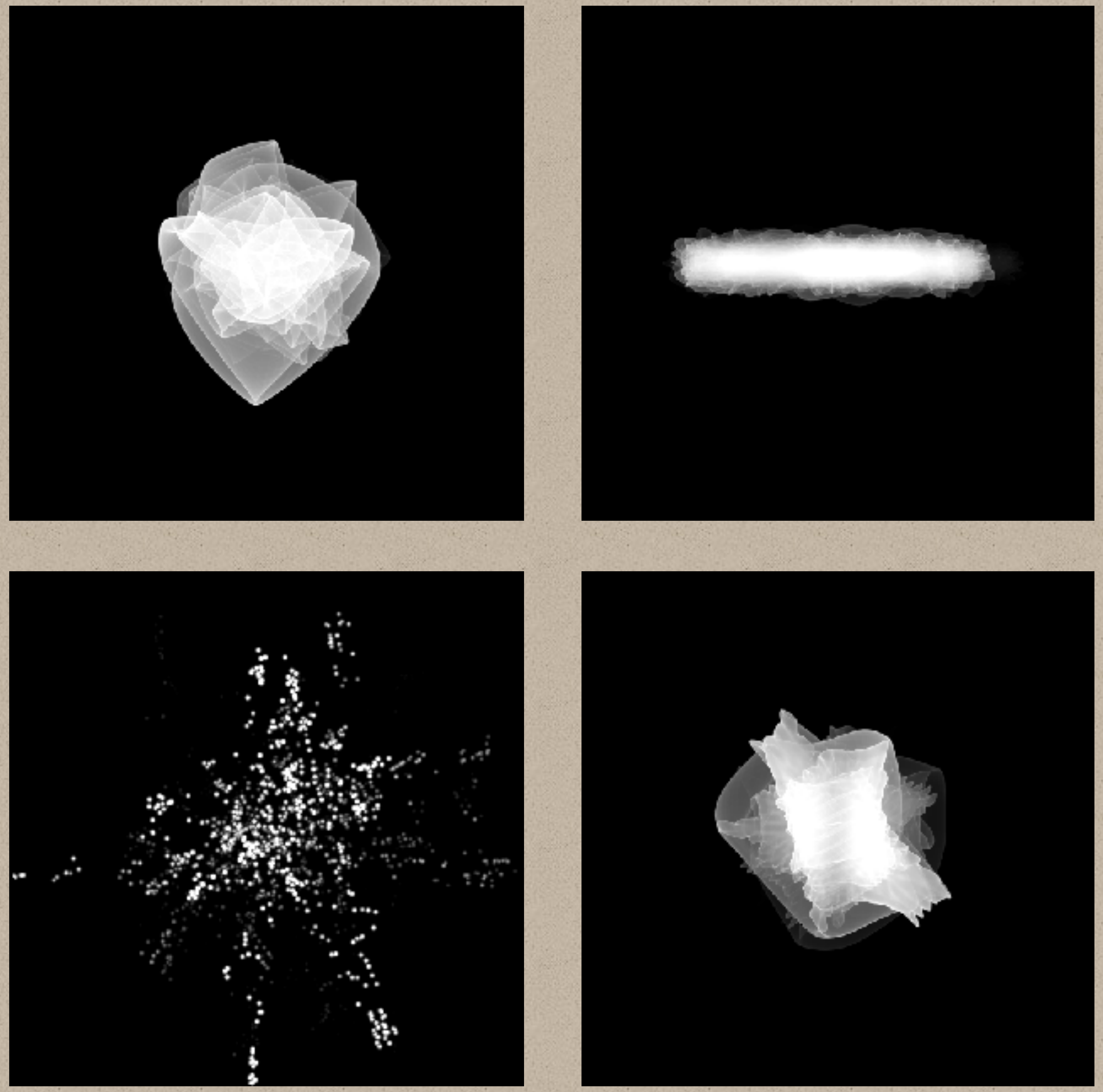


$\mathcal{P}$-NDFs

NDF shapes



normal maps



$\mathcal{P}$-NDFs

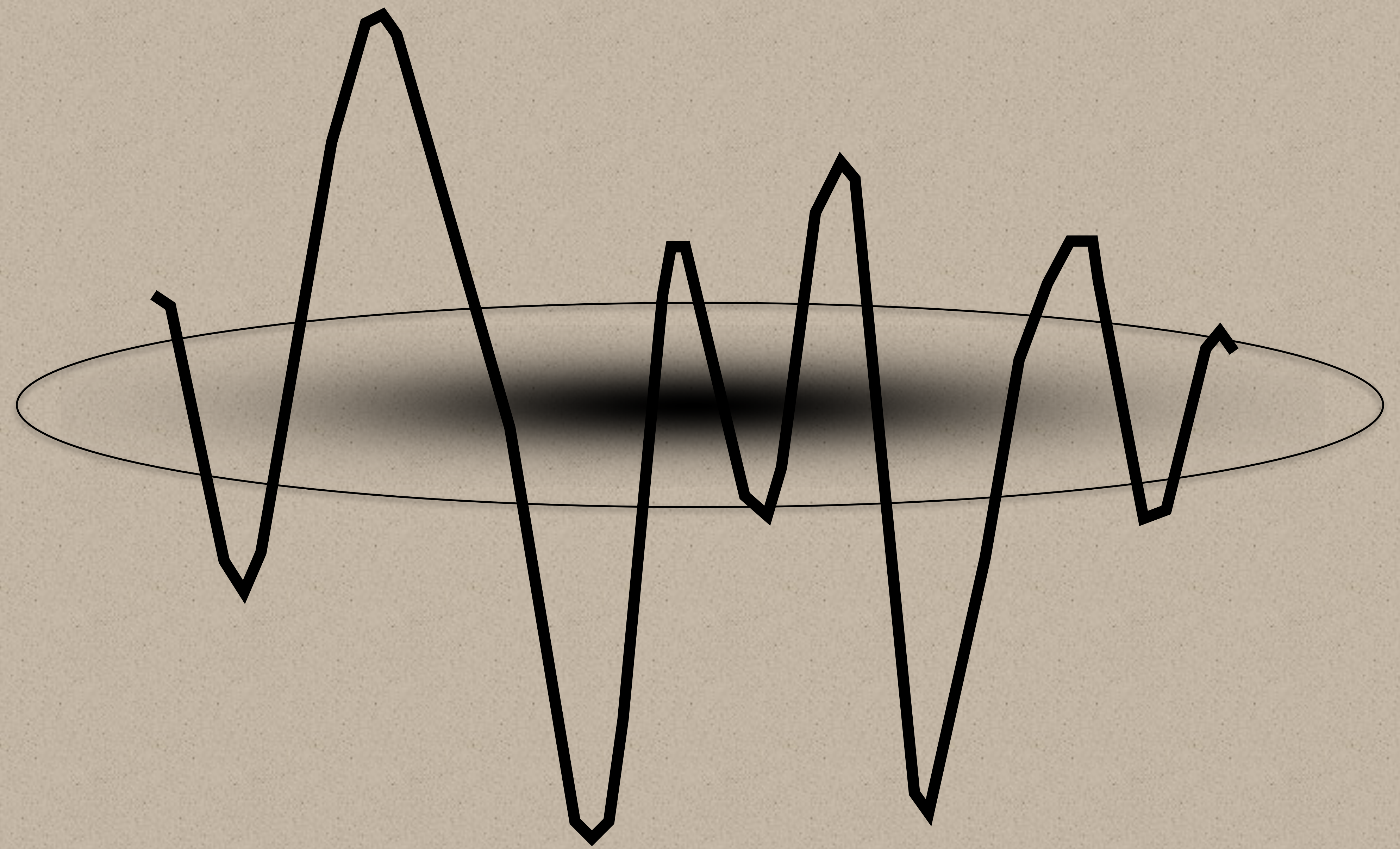
Converting NDF to BRDF

microfacet BRDF

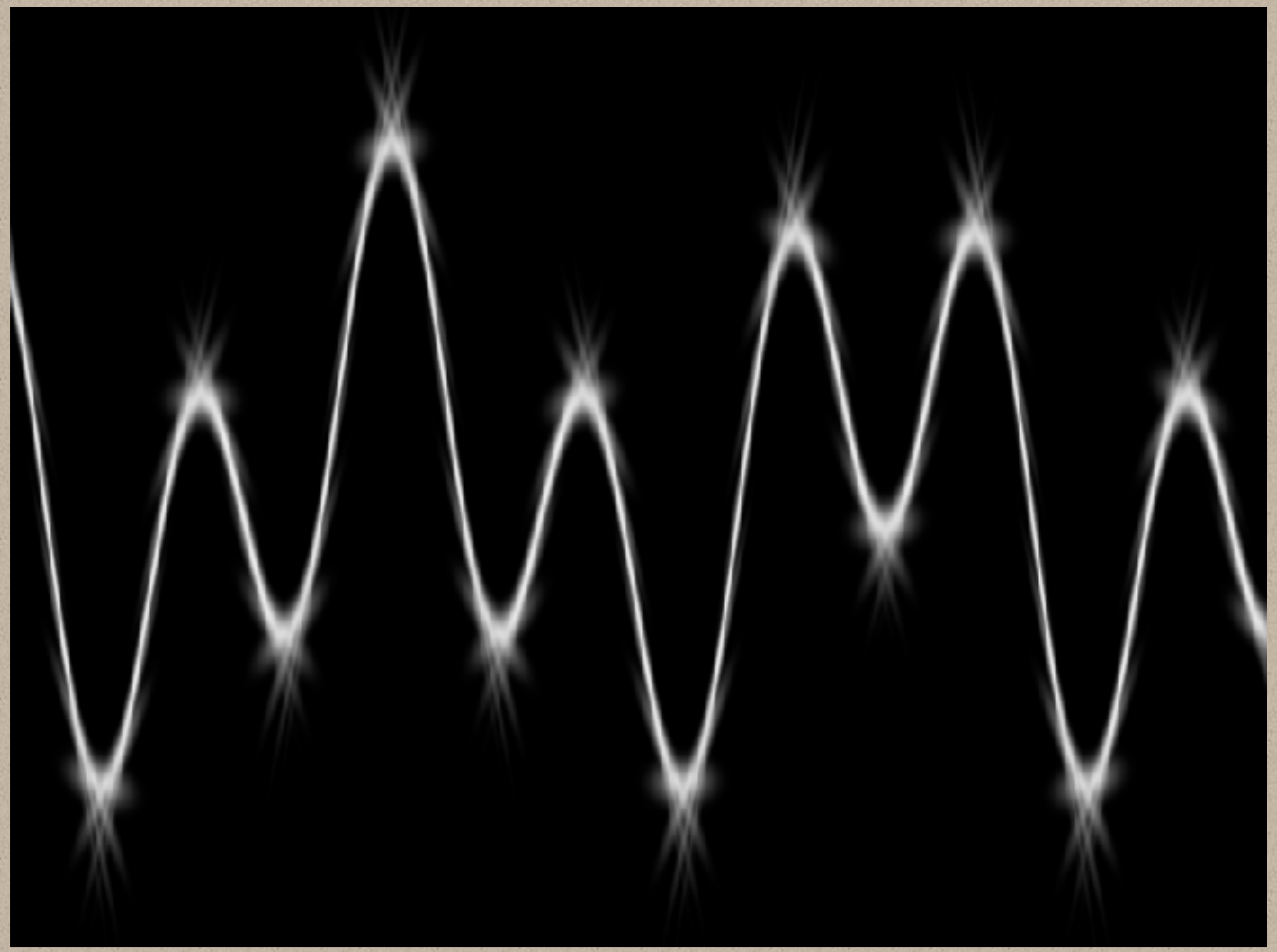
accurate NDF

$$f_{\mathcal{P}}(\mathbf{i}, \mathbf{o}) = \frac{\mathbf{F}\ \mathbf{G}\ \boxed{\mathbf{D}_{\mathcal{P}}(\mathbf{h})}}{4 \cos\theta_{\mathbf{i}} \cos\theta_{\mathbf{o}}}$$

Two different solutions to NDFs



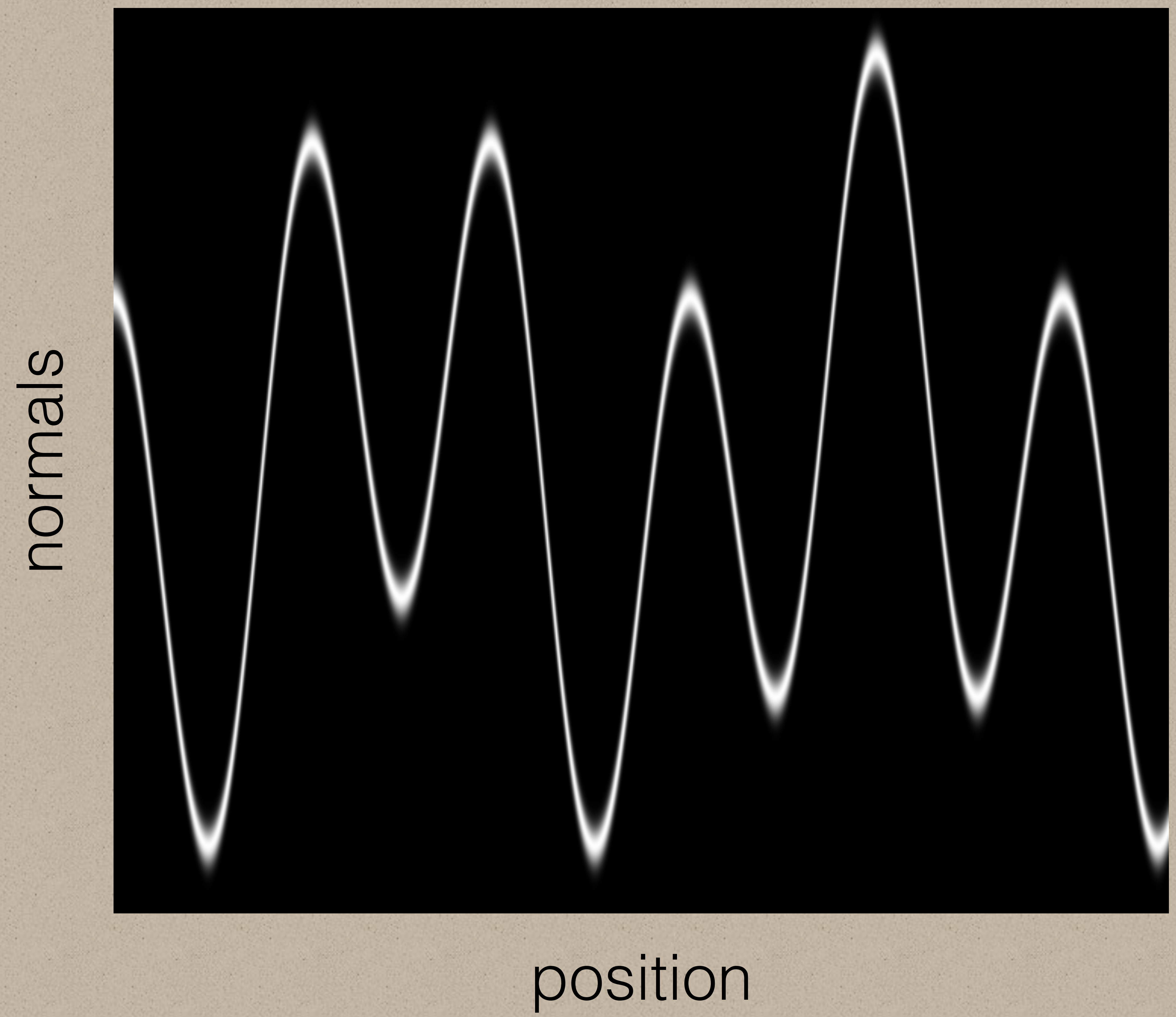
Yan et al. [2014]



Yan et al. [2016]
(100x faster)

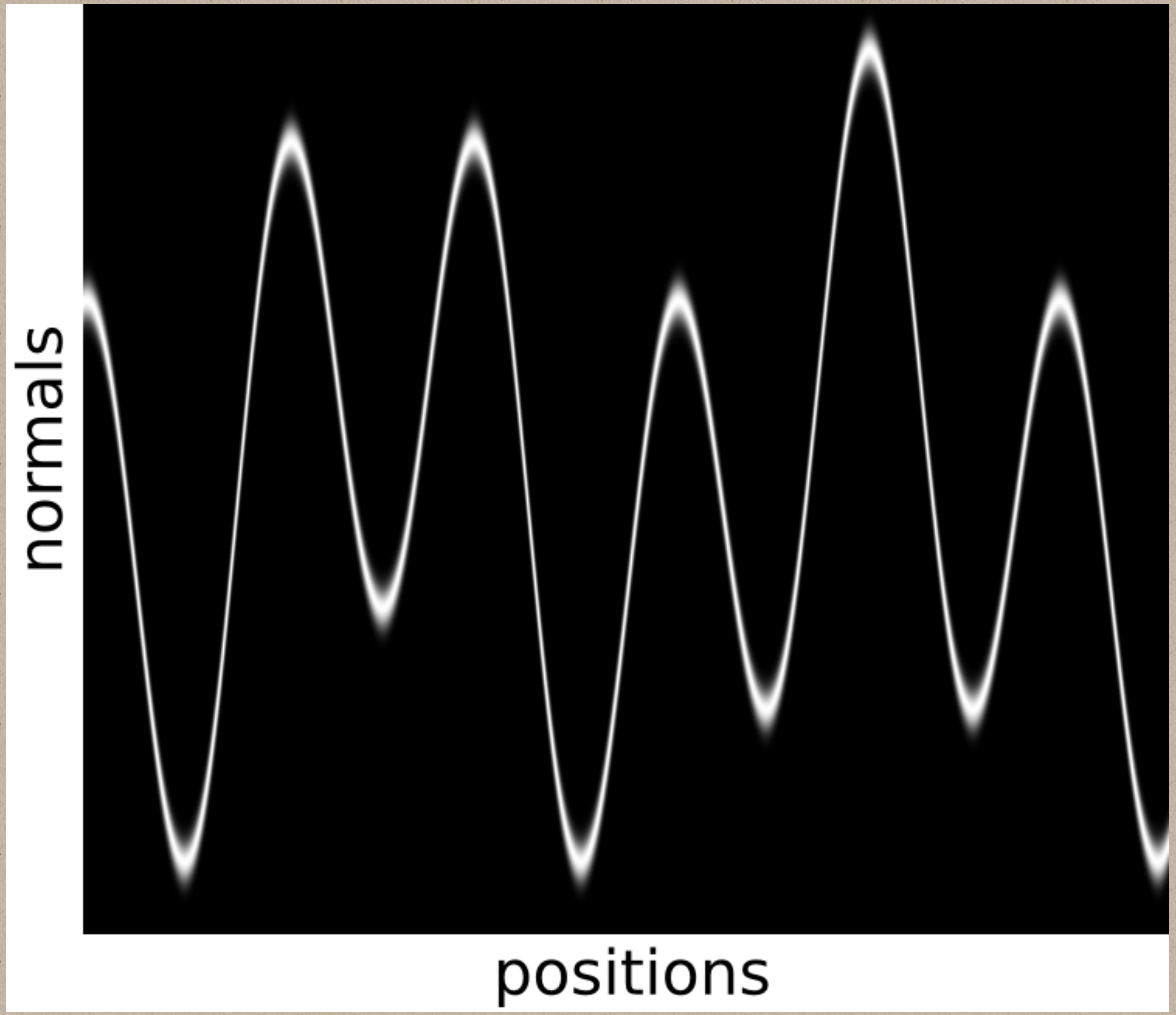
Position-normal distribution

- Key idea
  - Assume 1D normal map
  - Plot position-normal curve
  - **Treat the plot as a 2D image**
  - Fit using 2D Gaussians

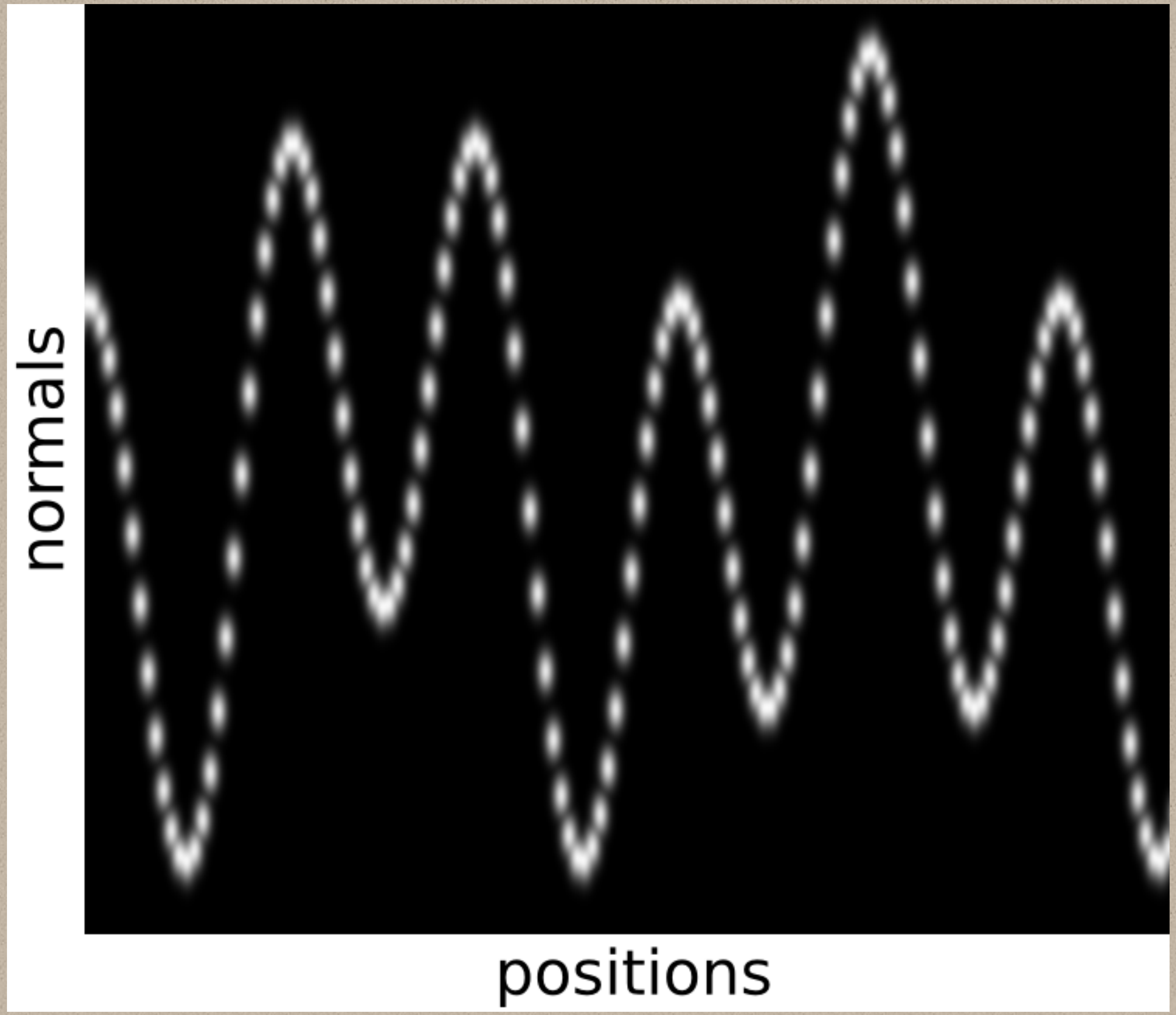


position-normal distribution

Approximating position-normal distribution

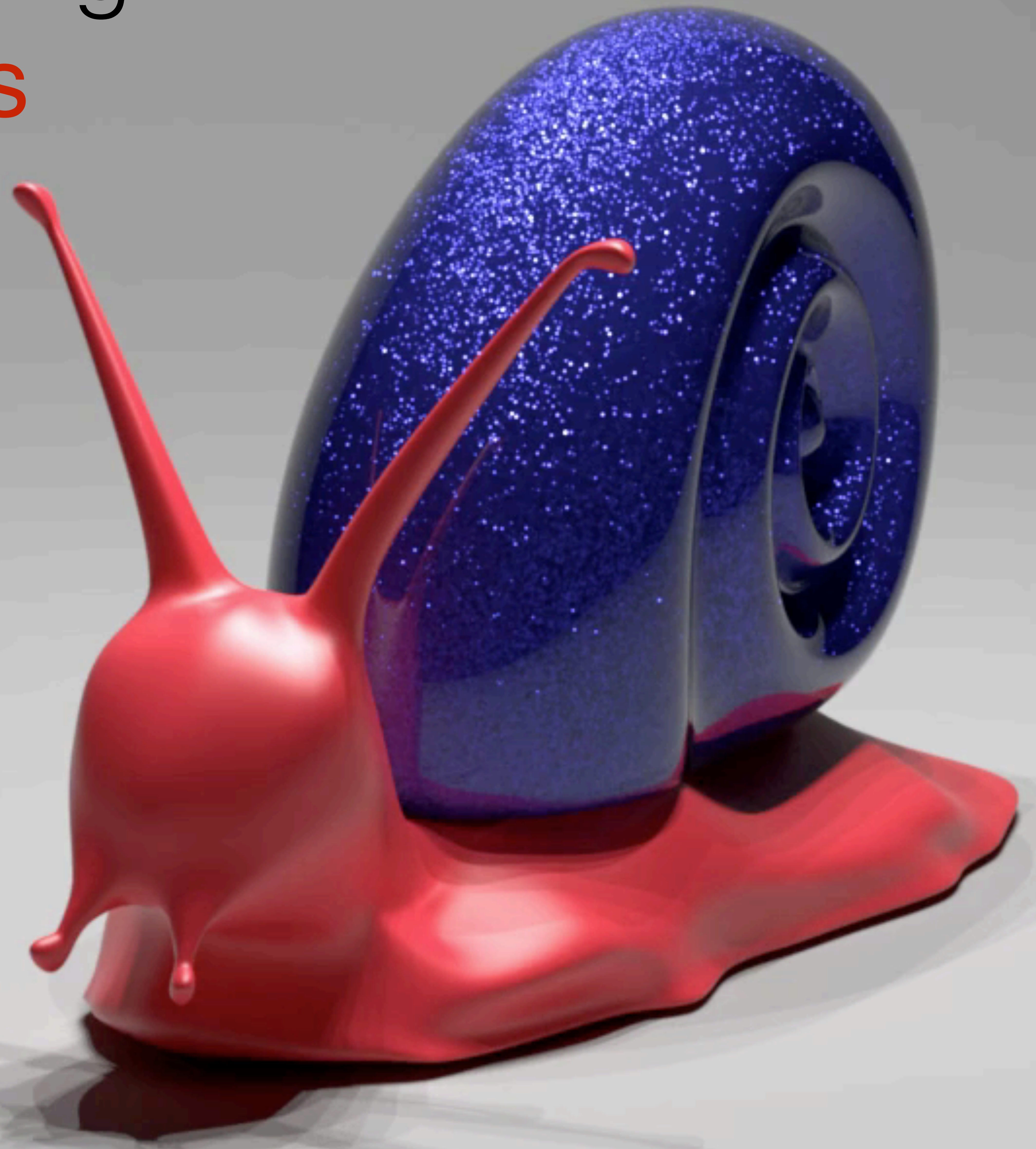


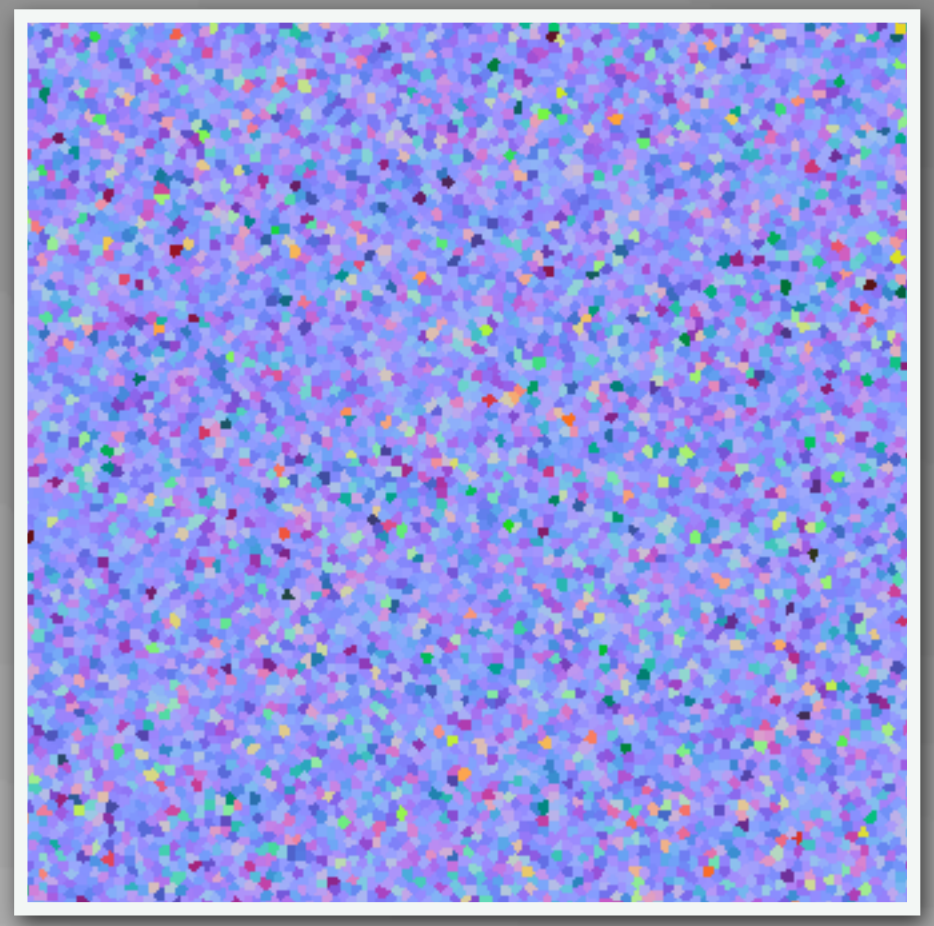
position-normal distribution



approx using 160 flat elements
(axis-aligned Gaussians)

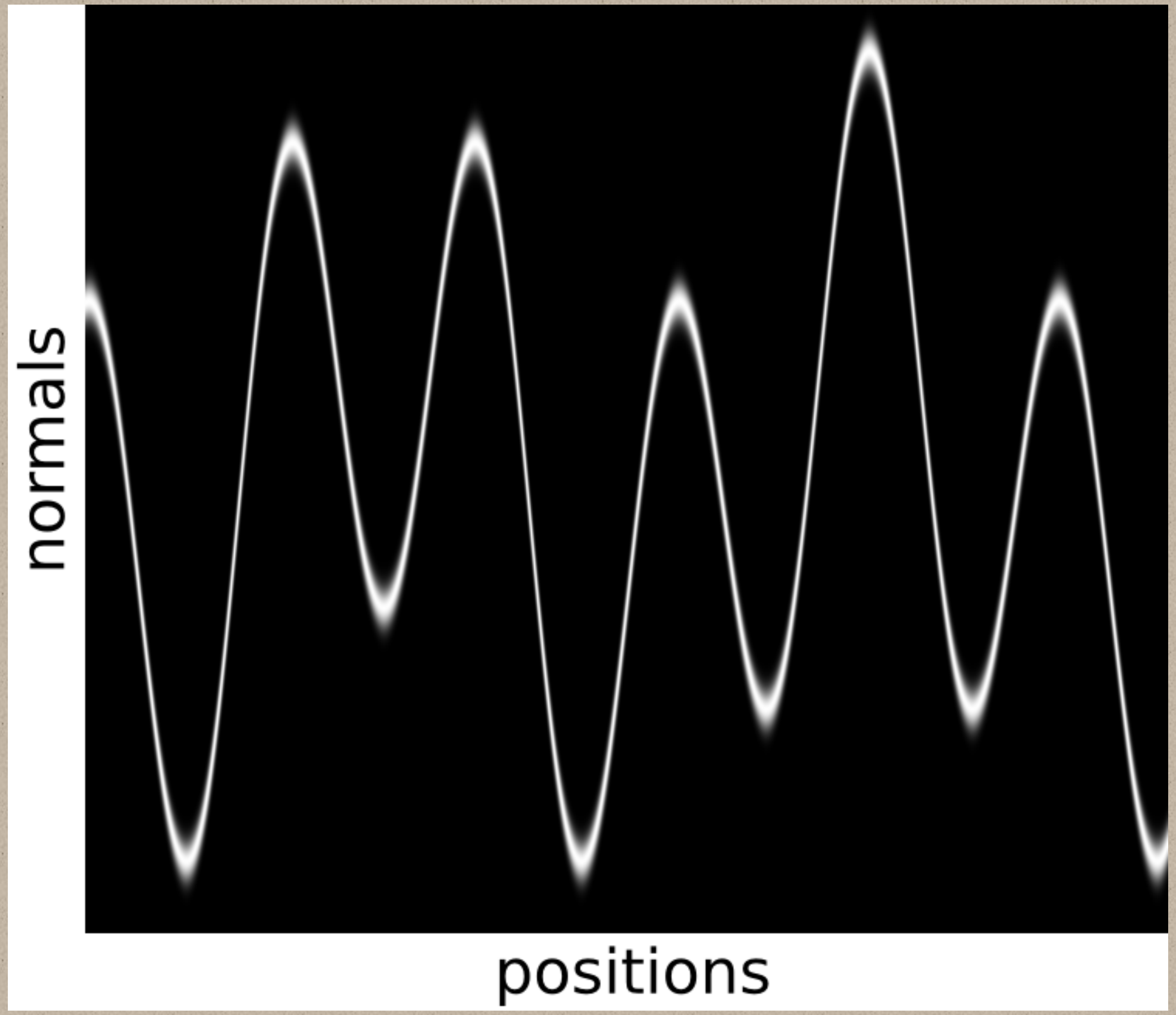
Application using flat elements



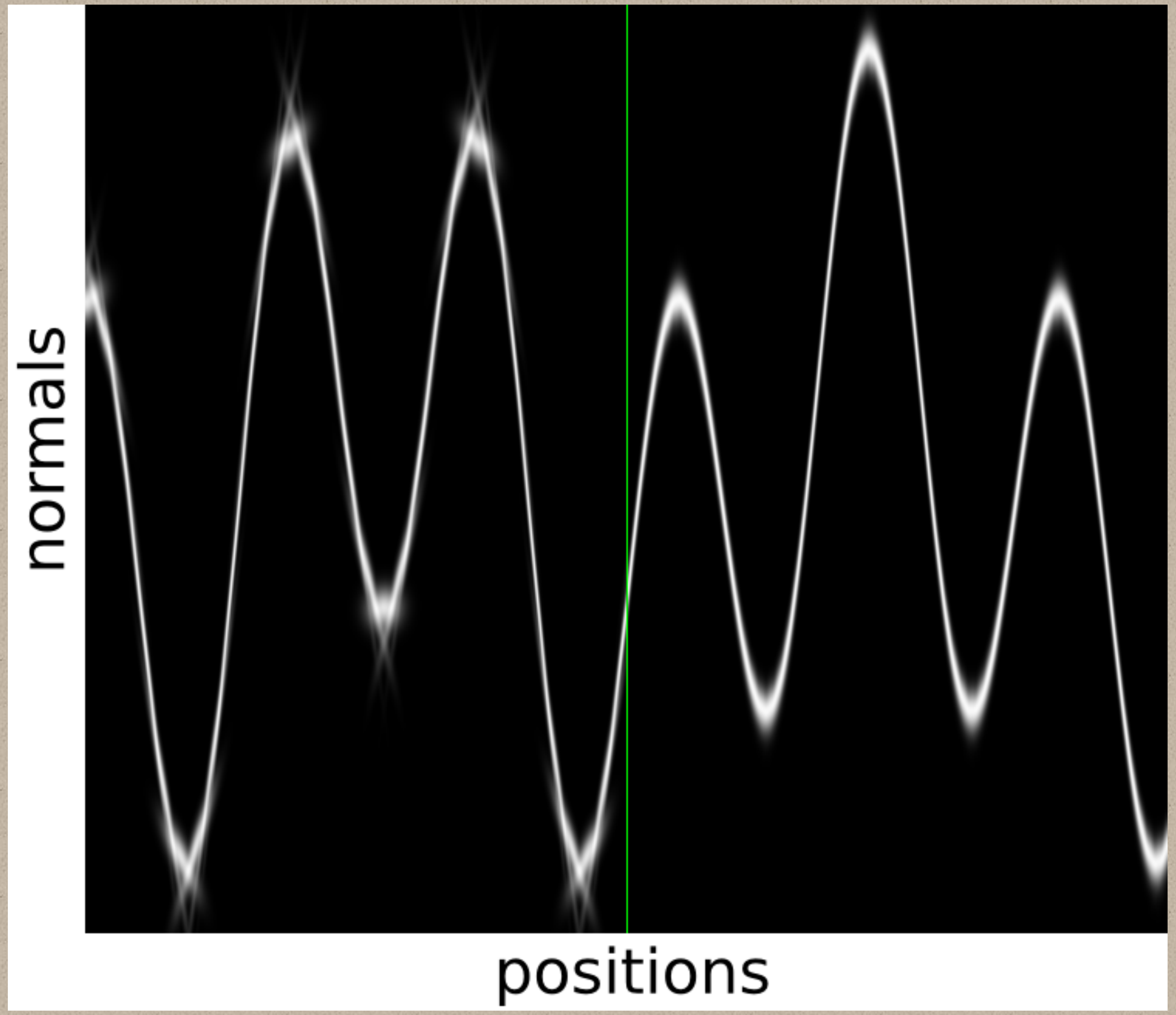


Metallic flakes

Approximating position-normal distribution



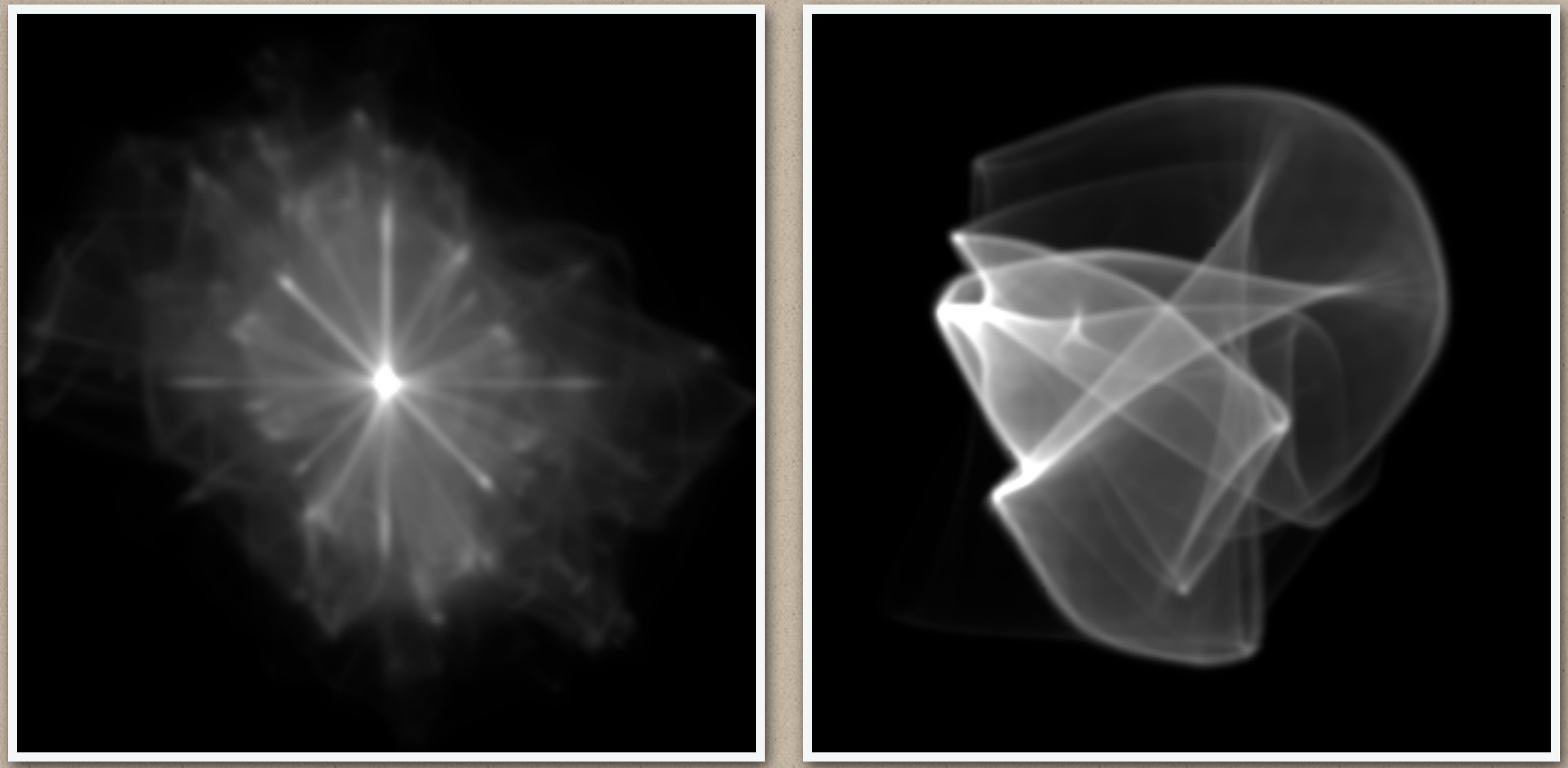
‘vertically’ blurred normal map

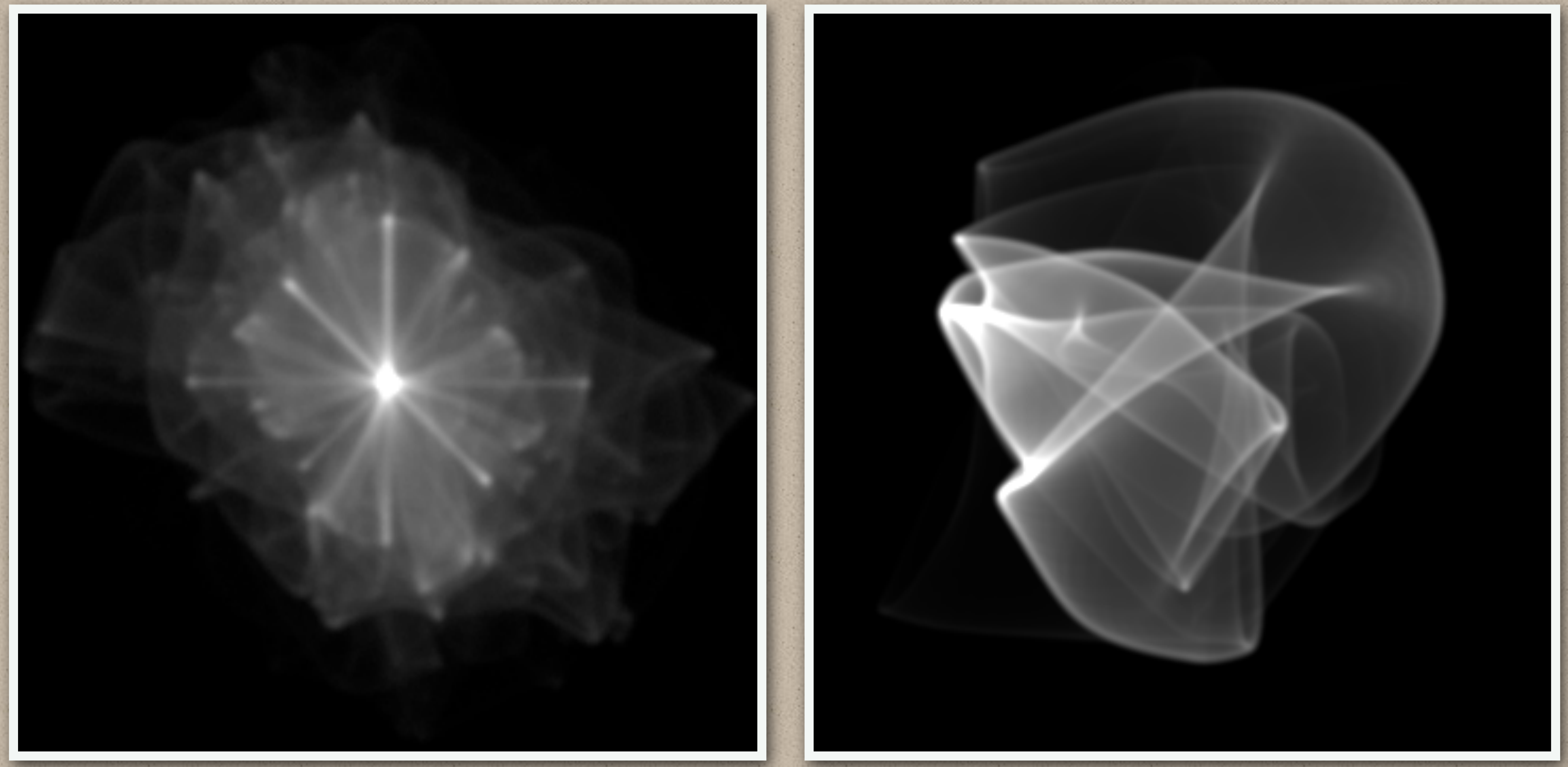


approx using 80/160 curved elements
(arbitrary Gaussians)


Application using
curved elements
Ocean waves

Ground truth $\mathcal{P}$-NDF comparison

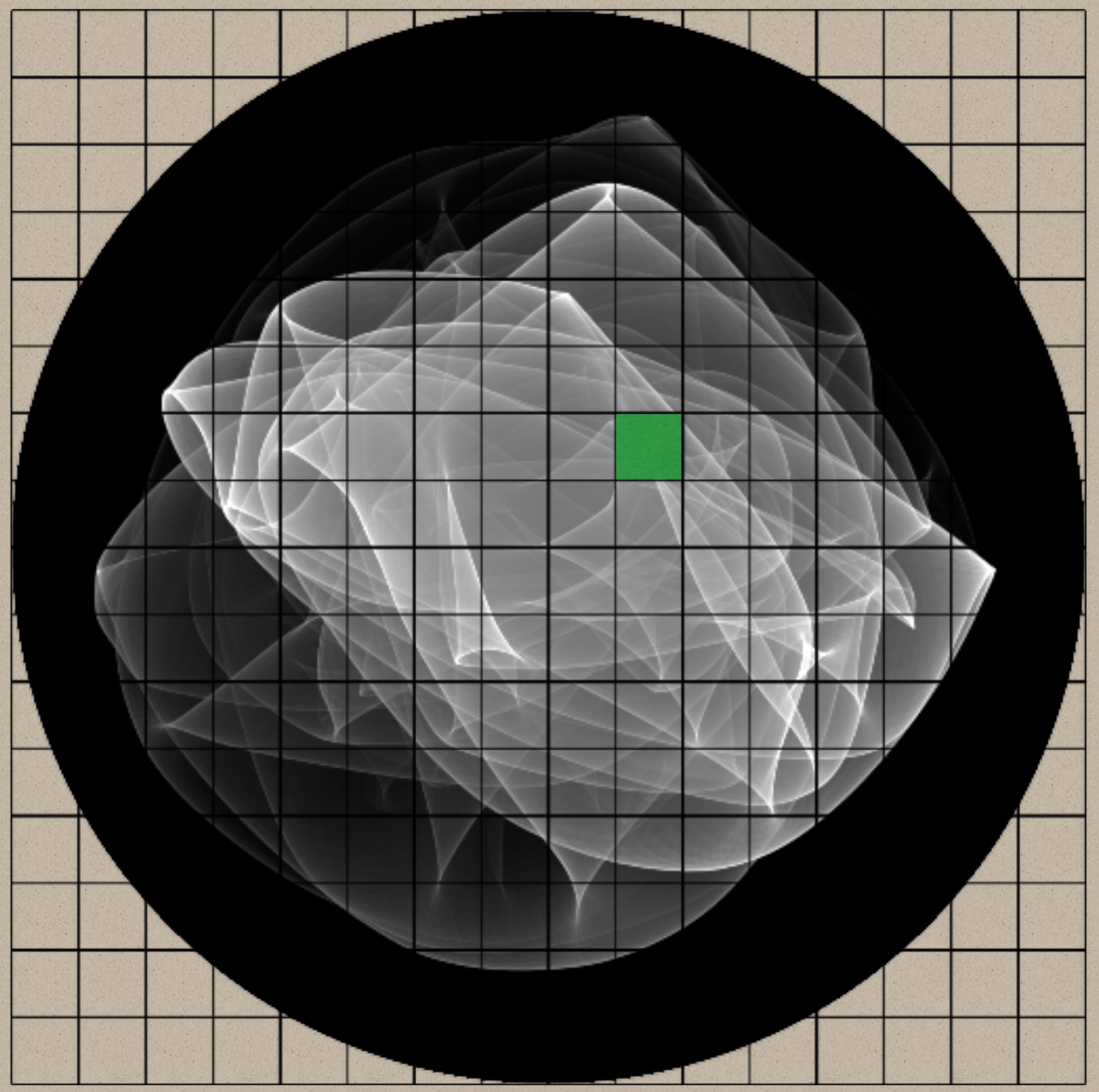


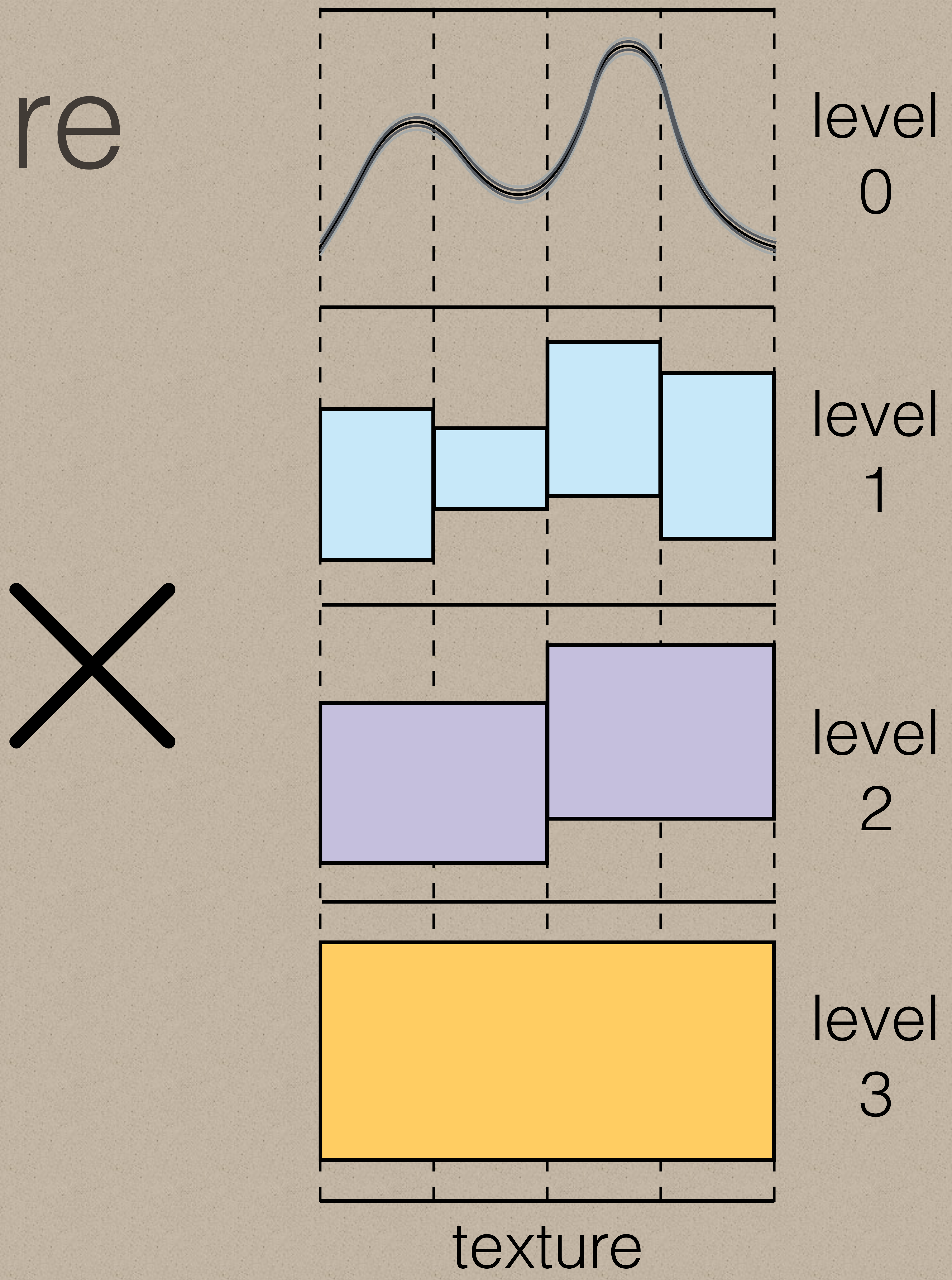


our method
(point-wise evaluation)

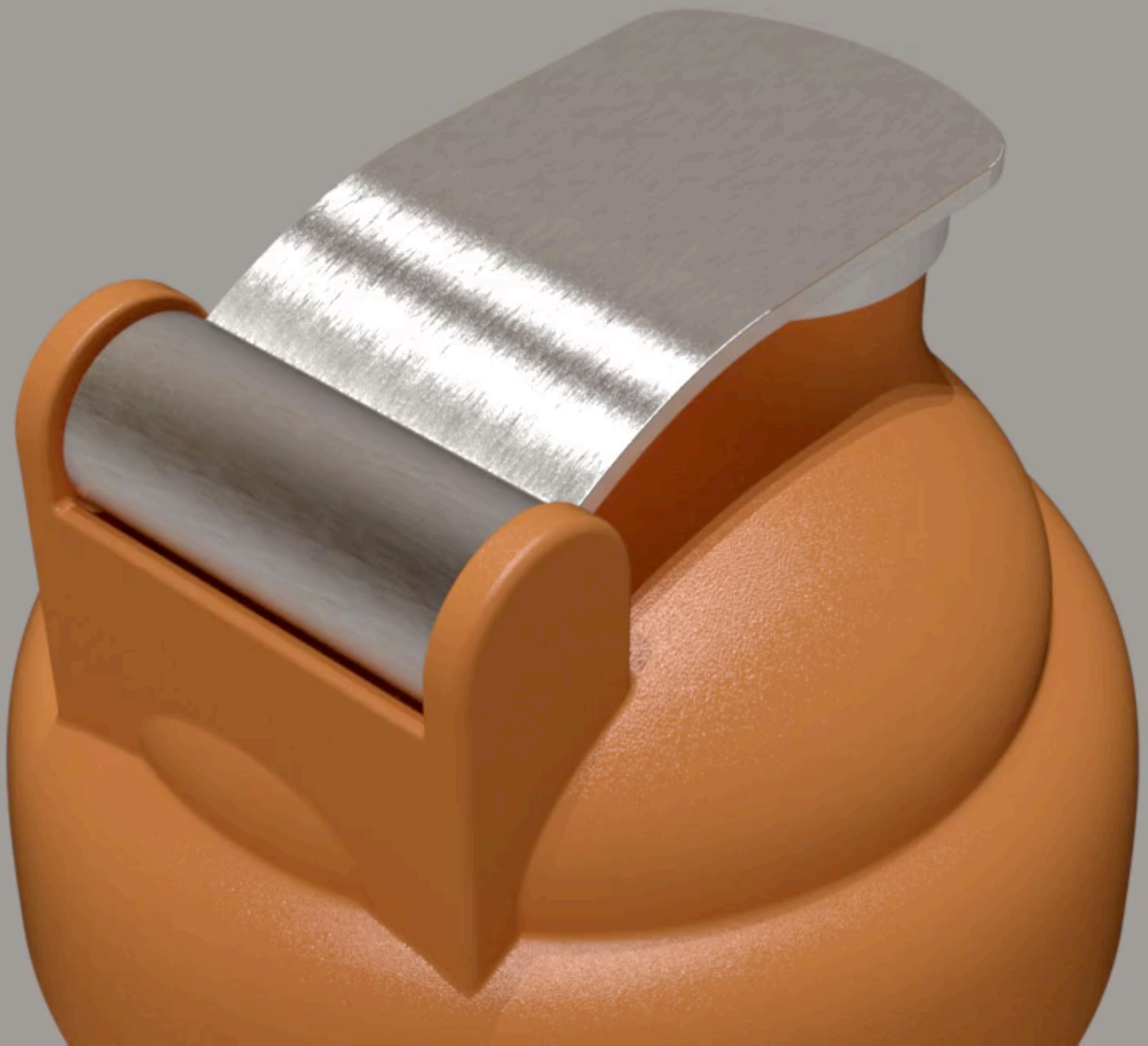
binning (histogram)
100 million samples

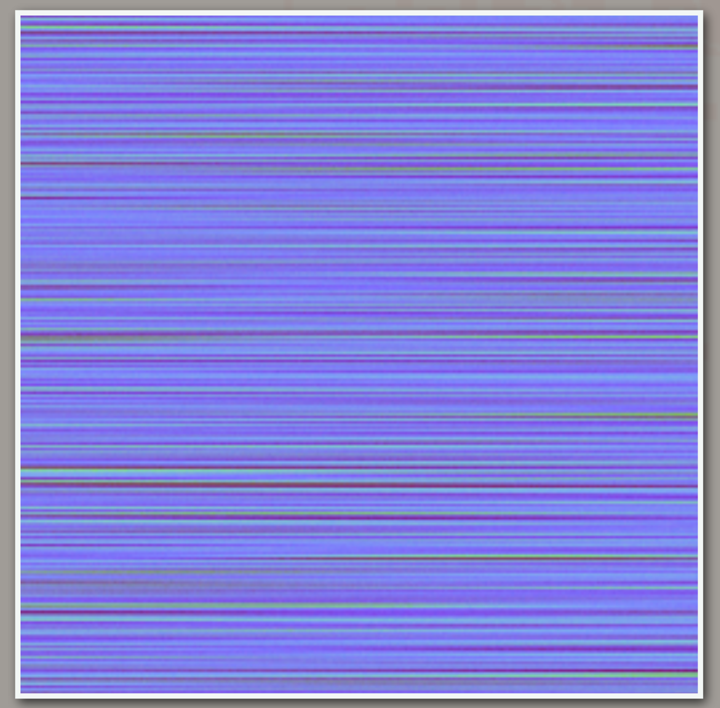
Acceleration structure



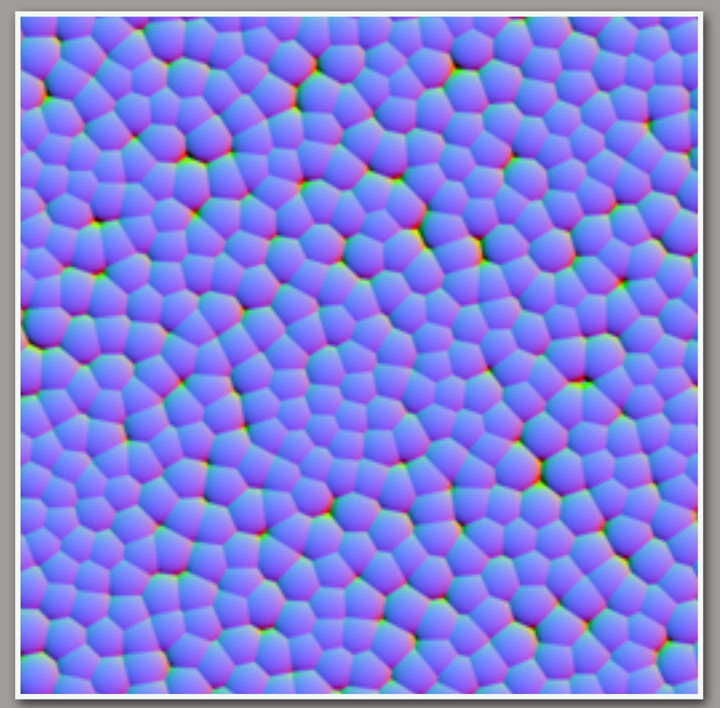


Blender



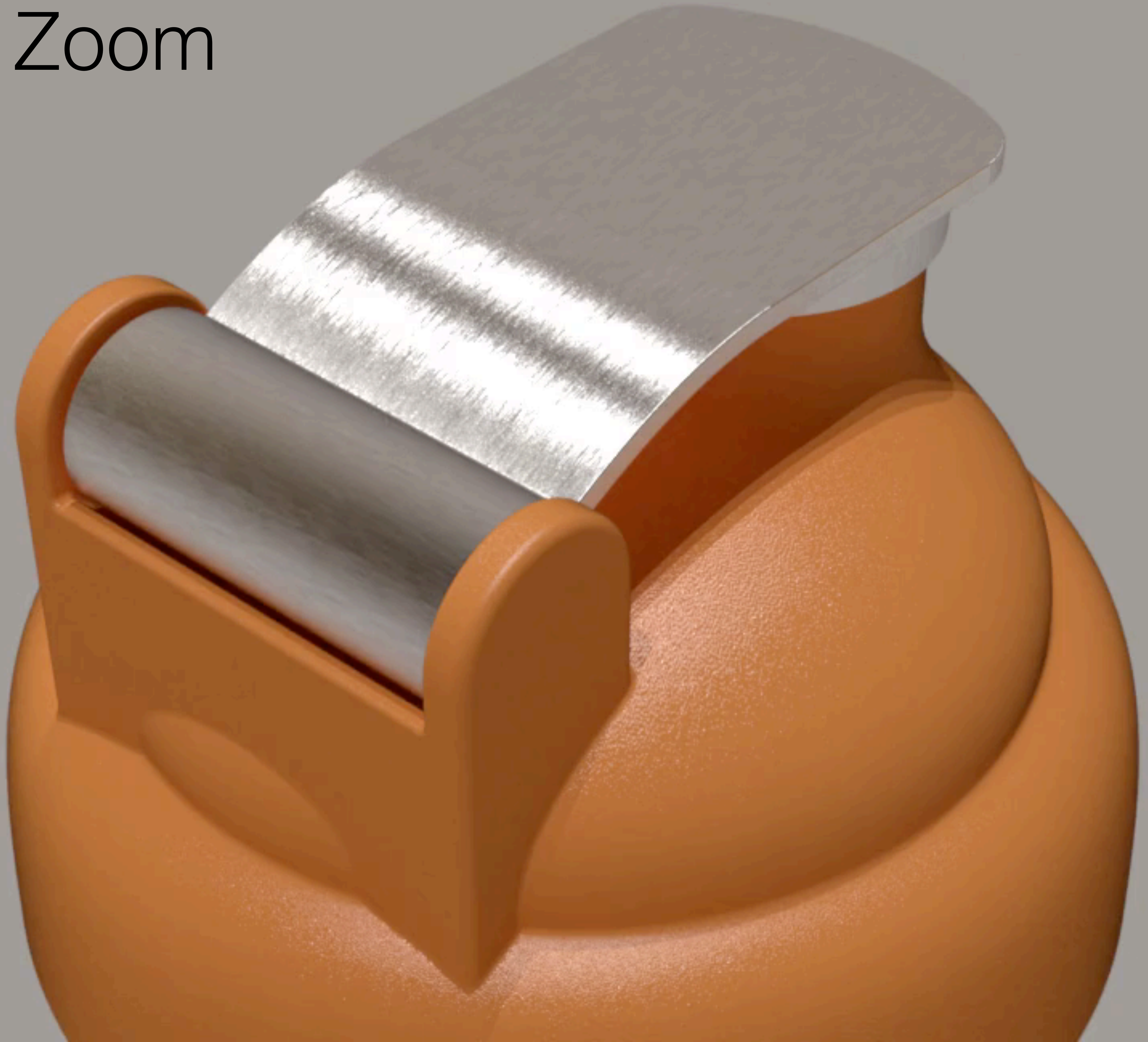


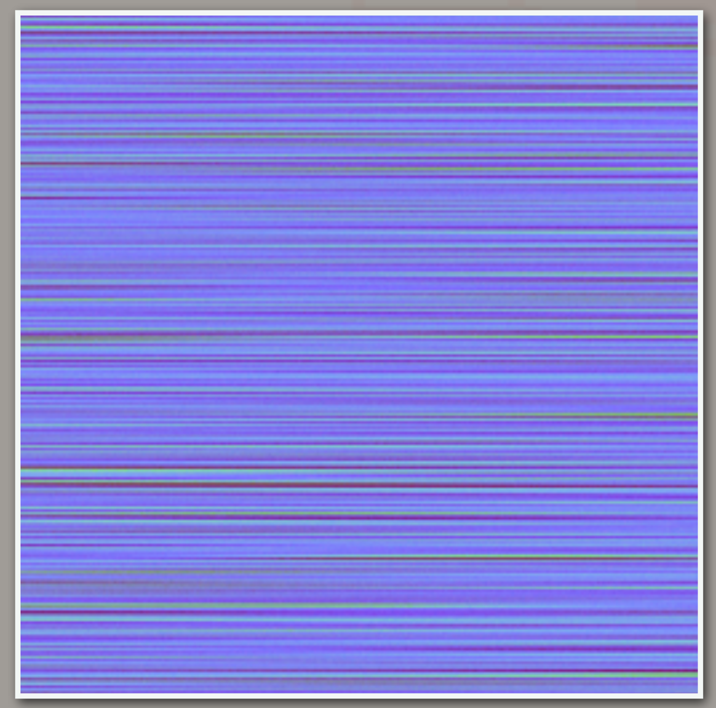
brushed metal



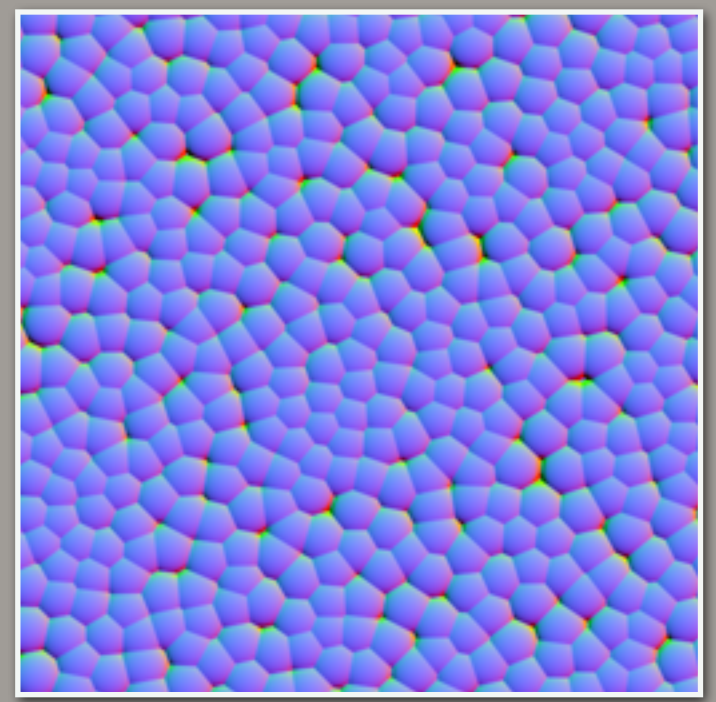
ellipsoid bumps

Blender: Zoom





brushed metal



ellipsoid bumps

Cutlery

scratched metal

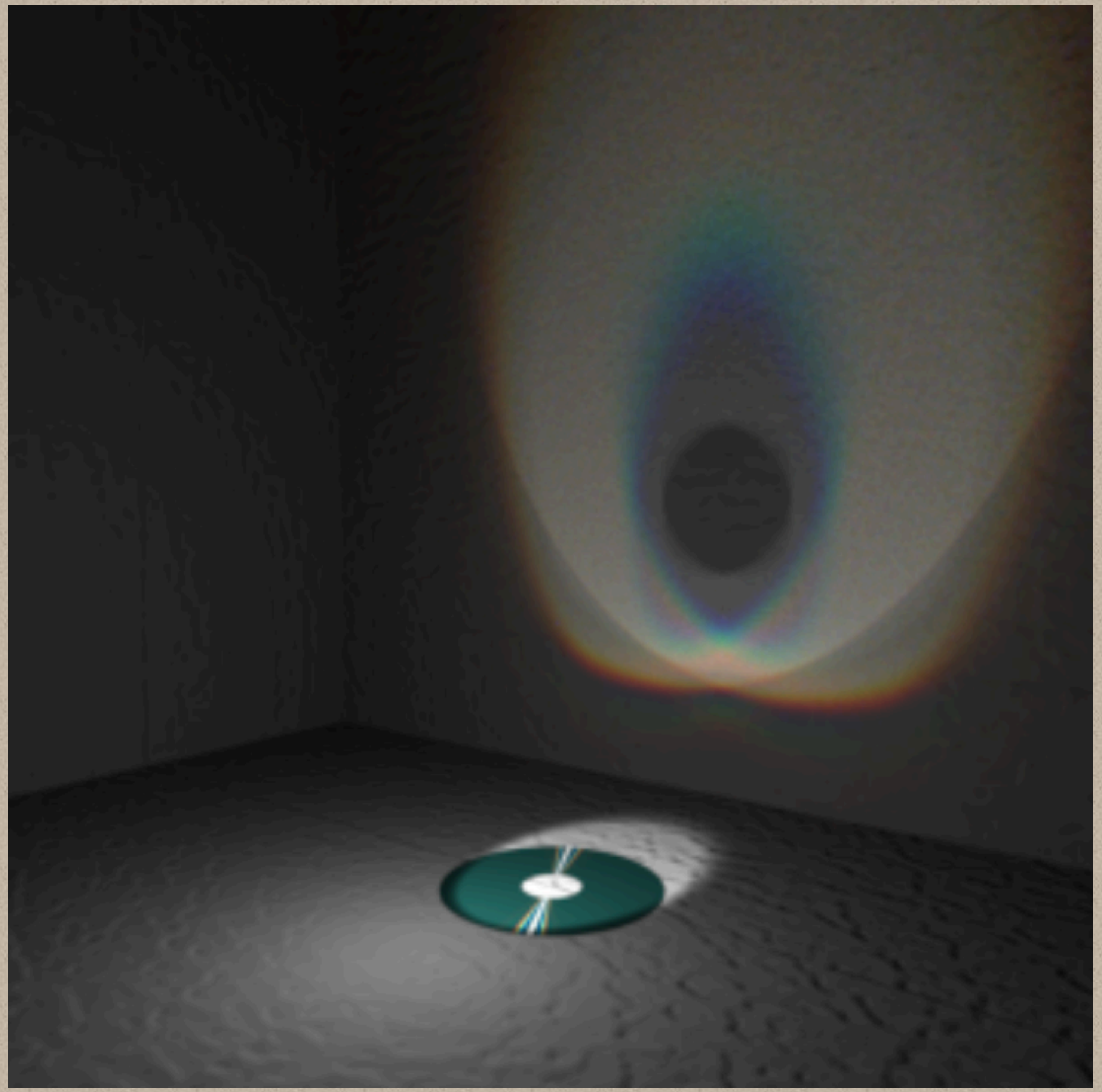
Car (metallic paint + scratches)

Frame render times (Yan et al. [2016])

	Kettle (microfacet BRDF)	Kettle	Car door	Sofa & floor
Time (minutes)	1.8	2.6 (only 1.4x slower!)	6.8	7.6

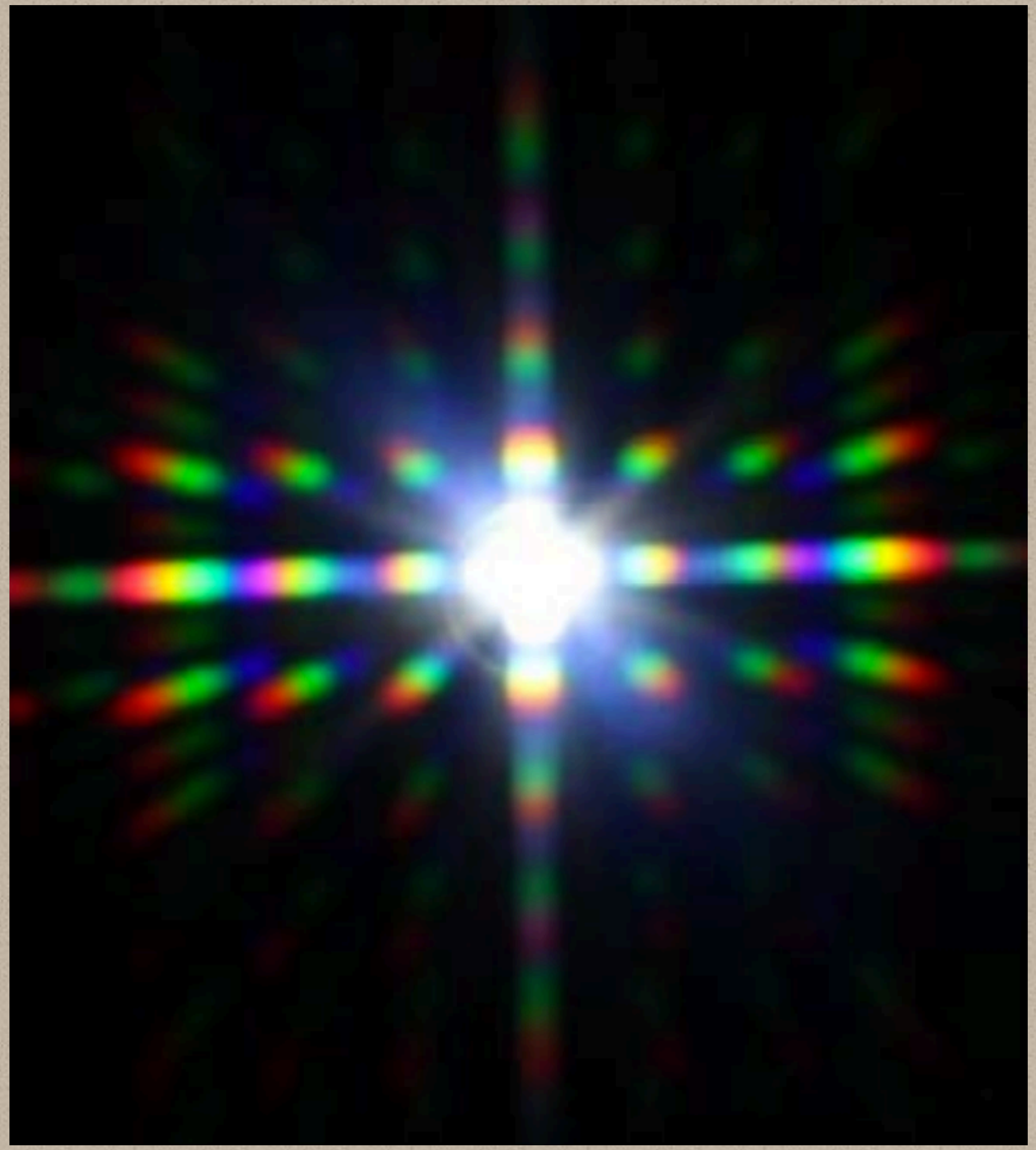
C++, Mitsuba, 36-core Amazon EC2 machine

Recent Trend: Wave Optics



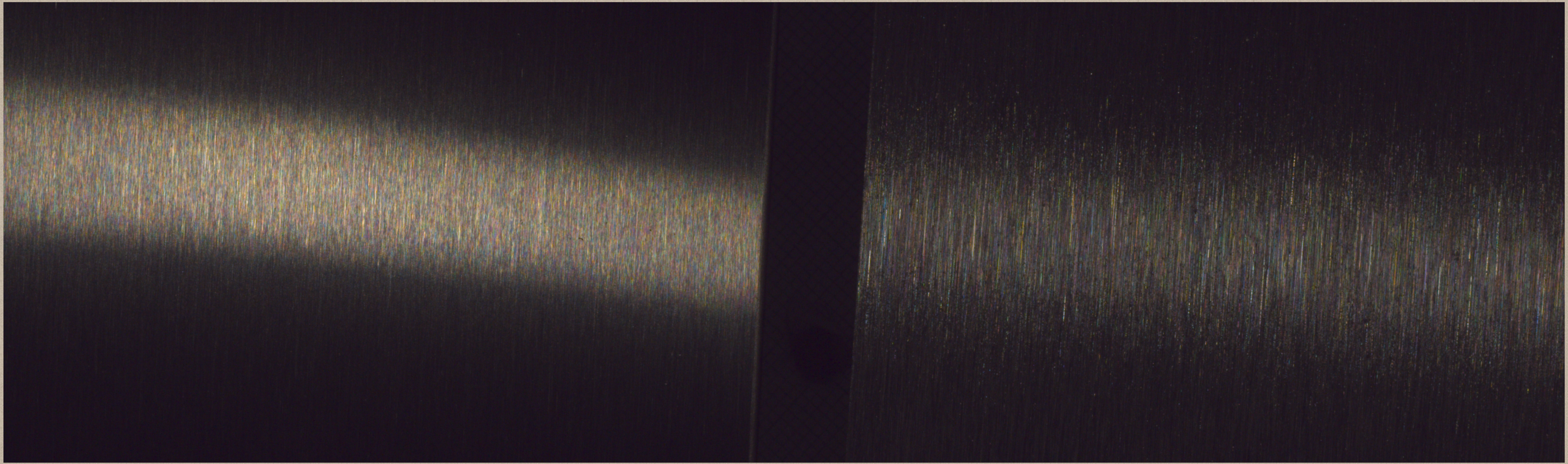
laser CD

metallic filming



phone screen

Observations



photos of scratched metal

Observations

photo of a Macbook

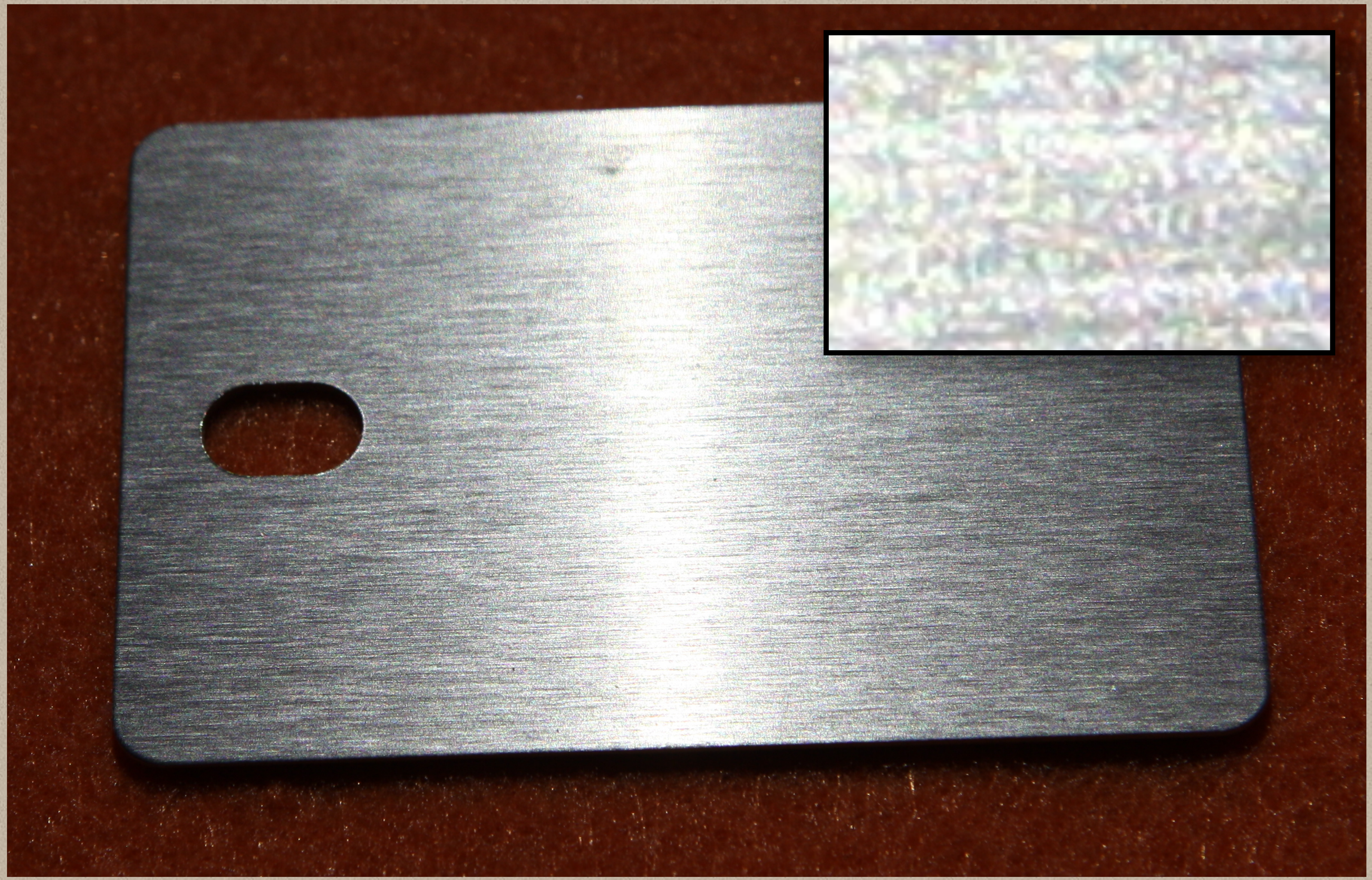
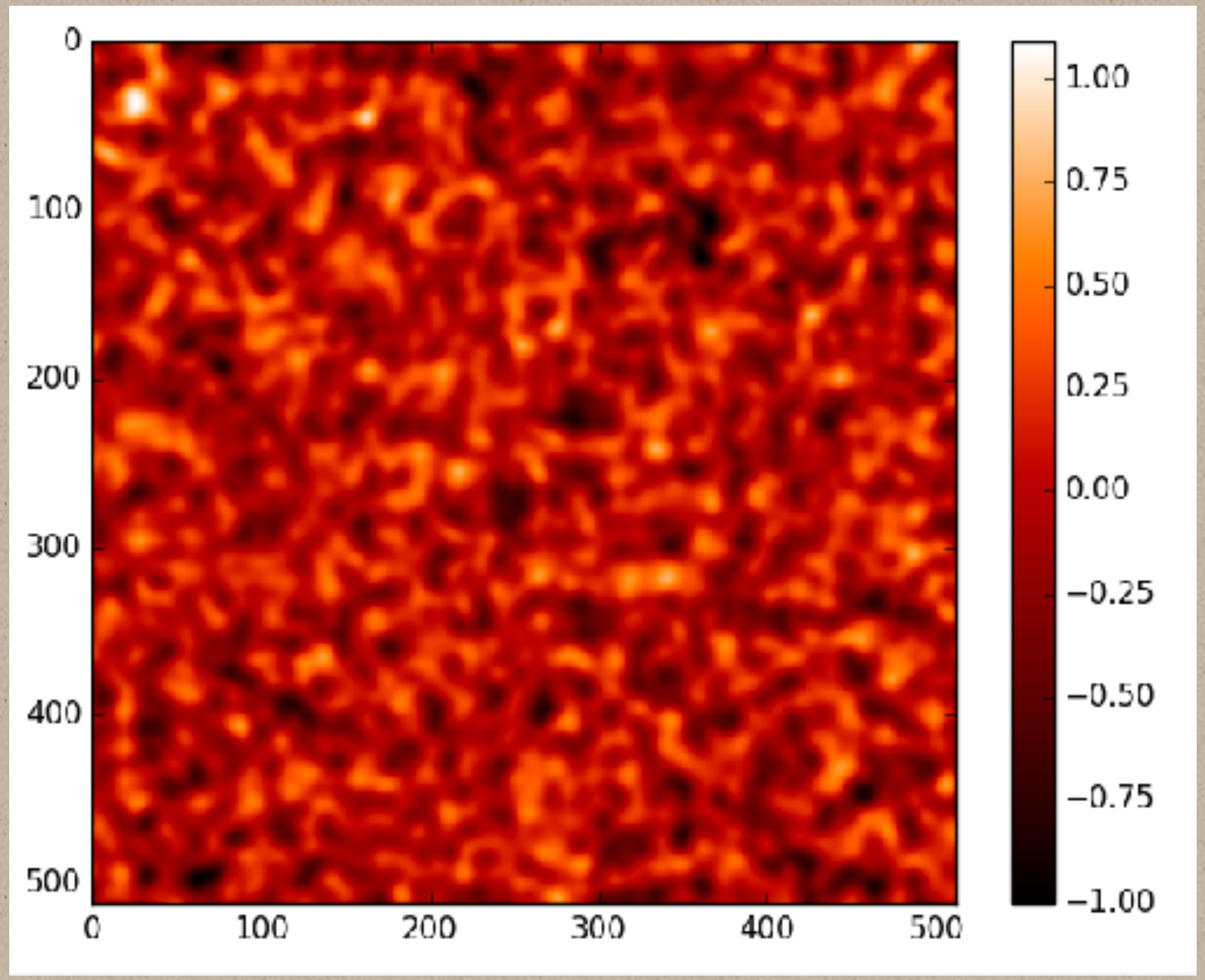


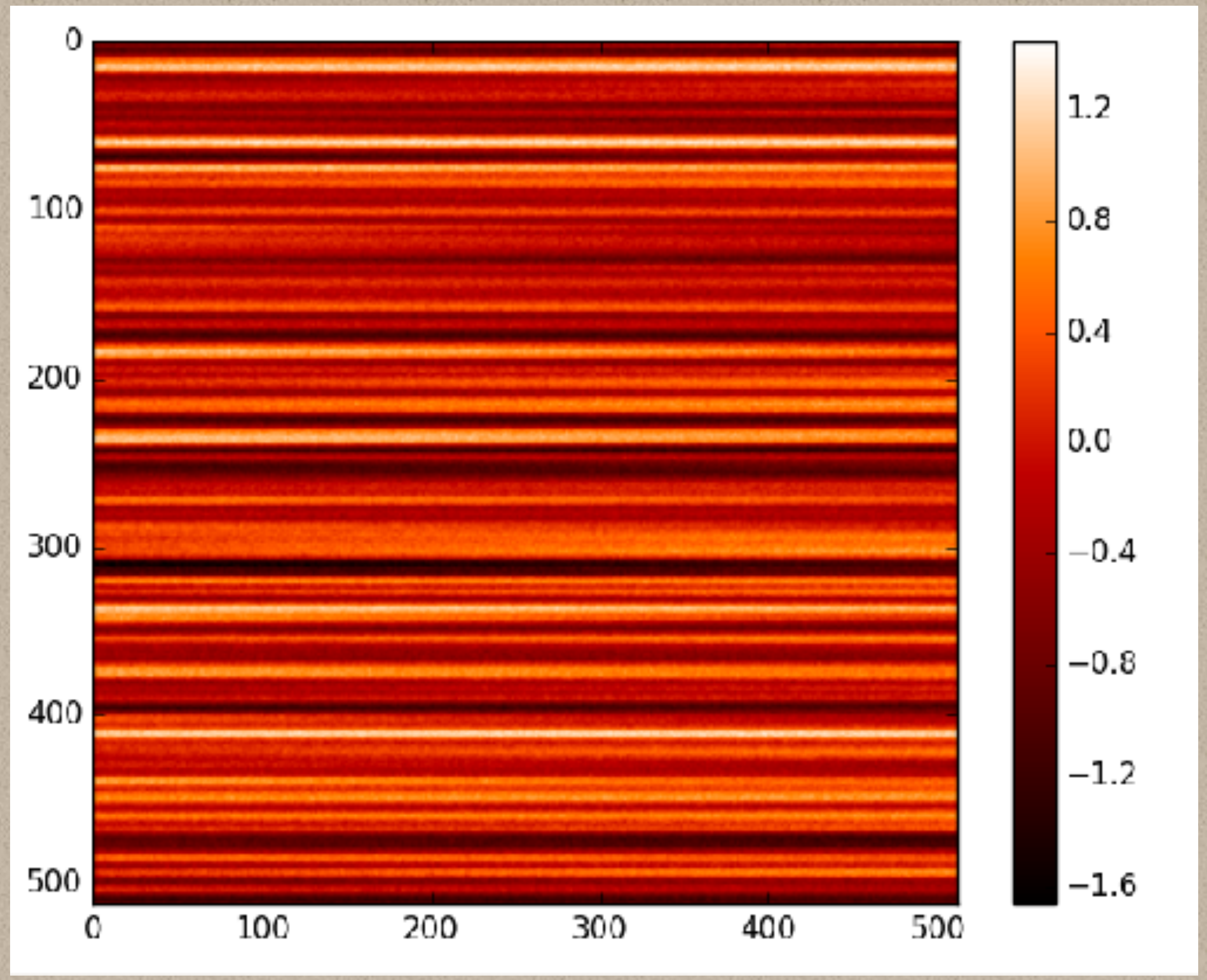
photo of an aluminum patch

Latest Work on Wave Optics (submitted)

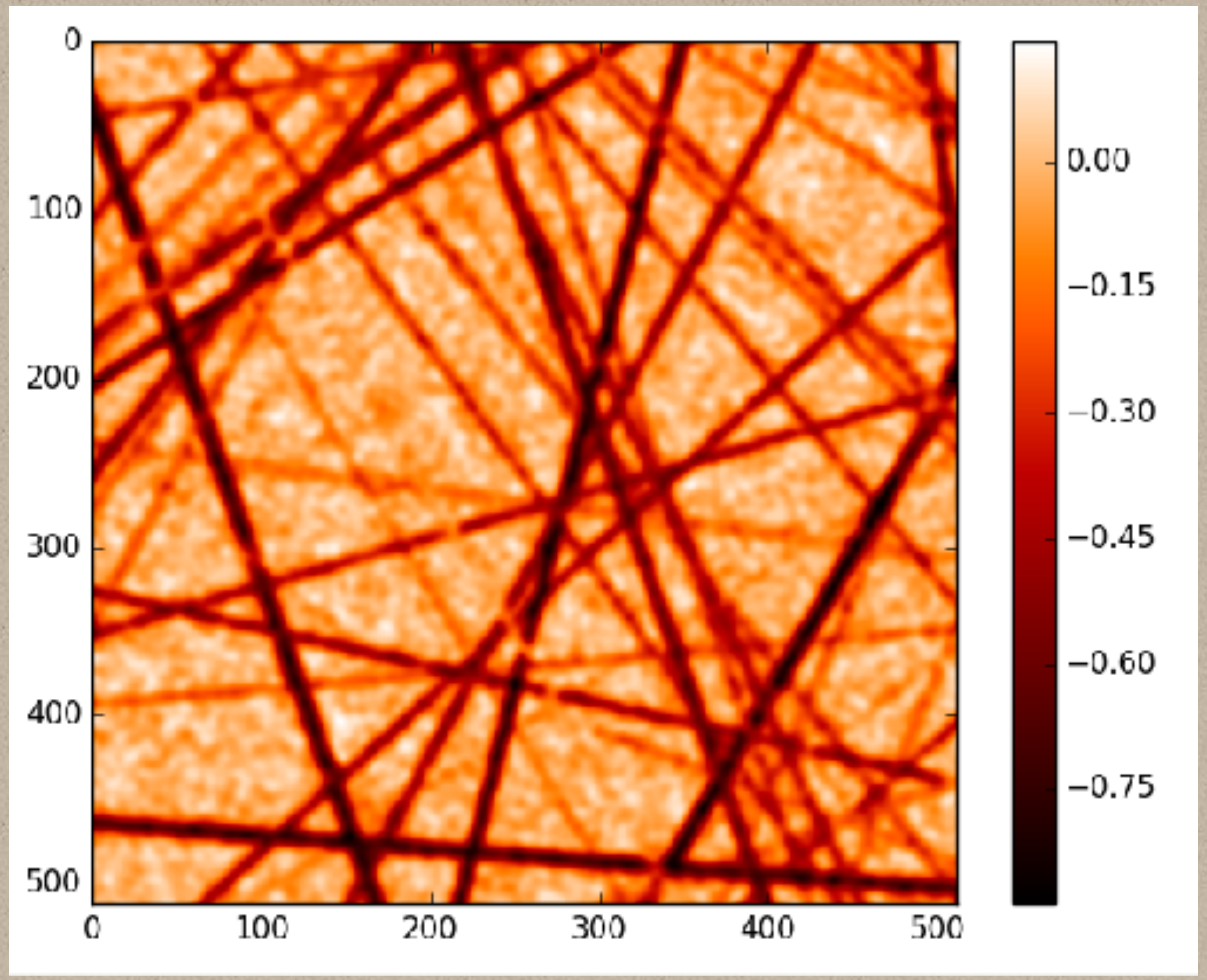
Heightfields



isotropic

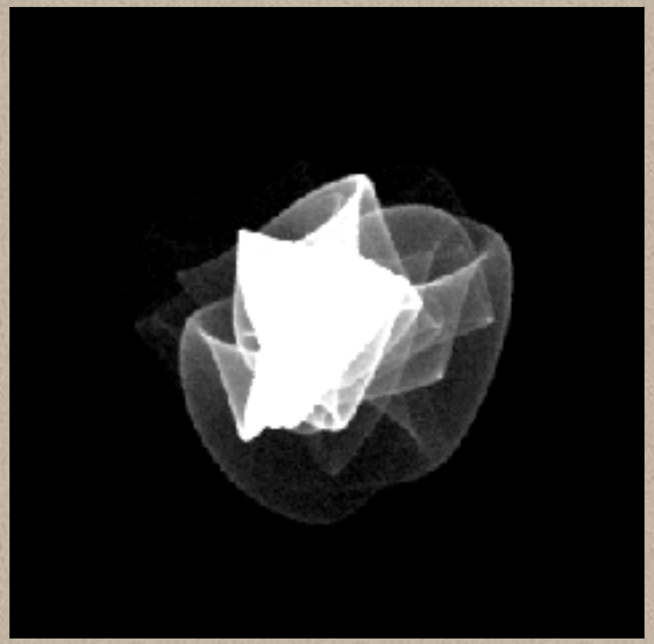


brushed

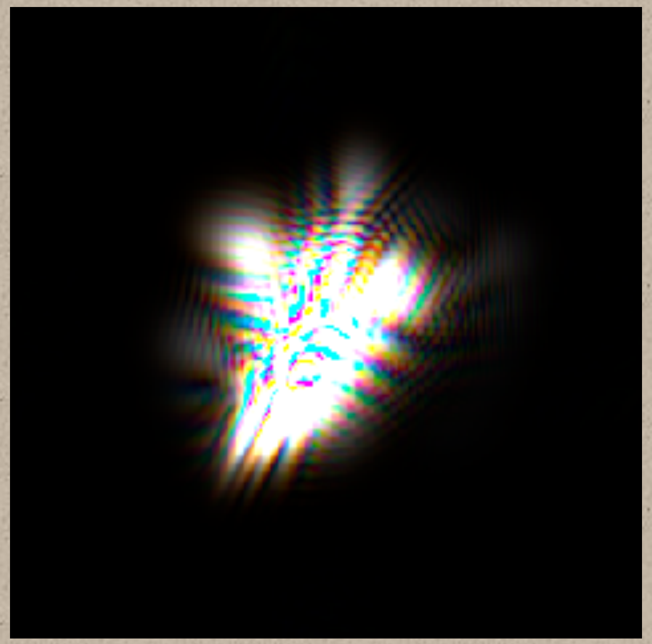


scratched

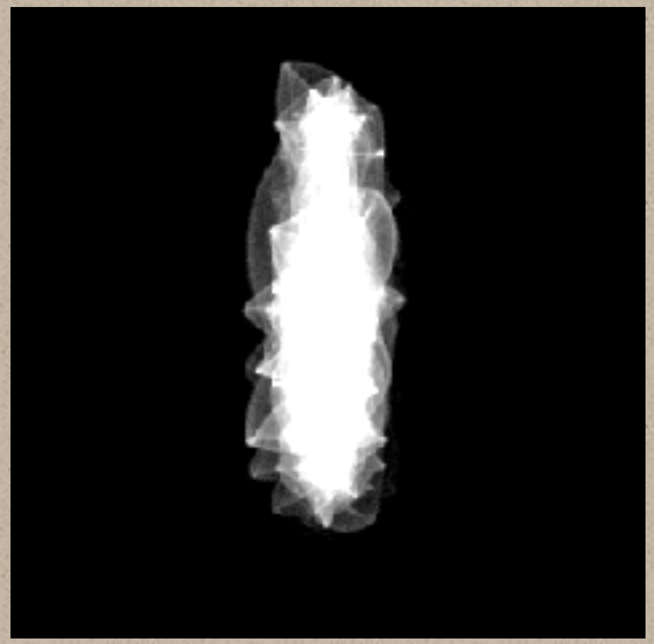
BRDFs



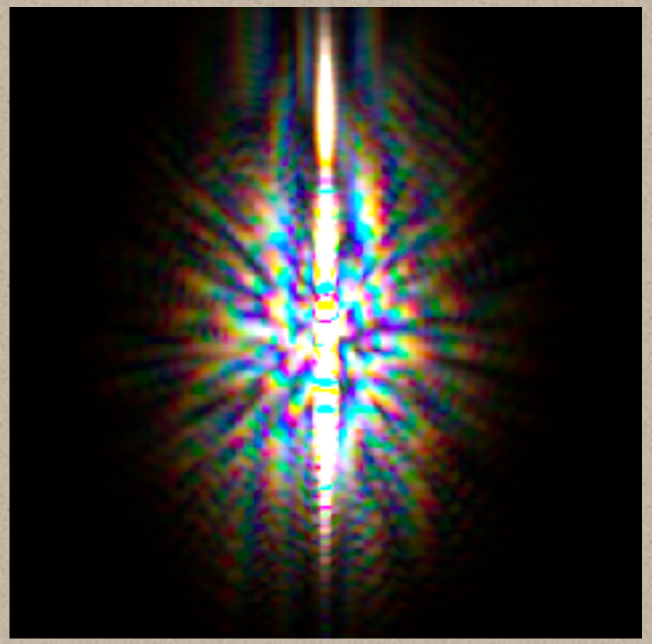
geometric



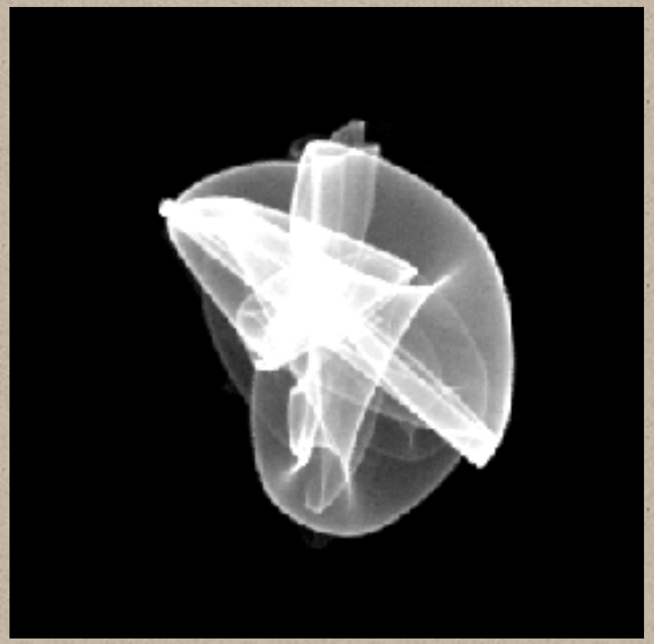
wave



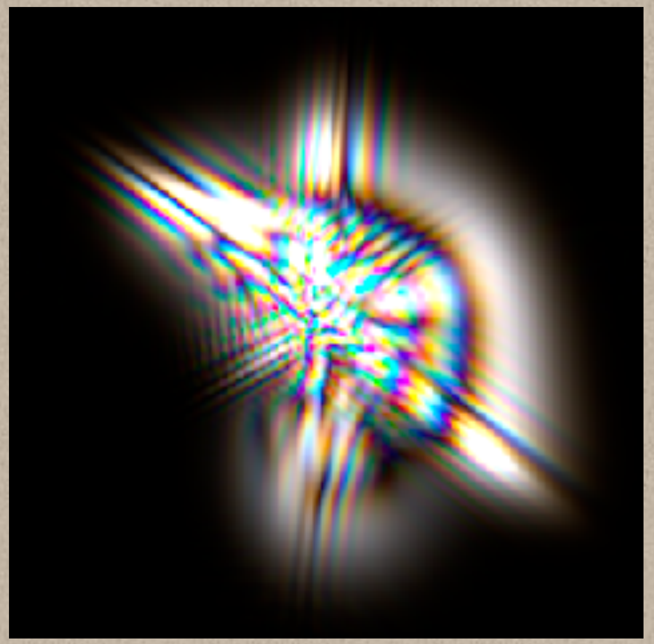
geometric



wave



geometric



wave

Rendered using wave optics

Wave optics

Part II: Detailed Appearance Modeling

detailed rendering

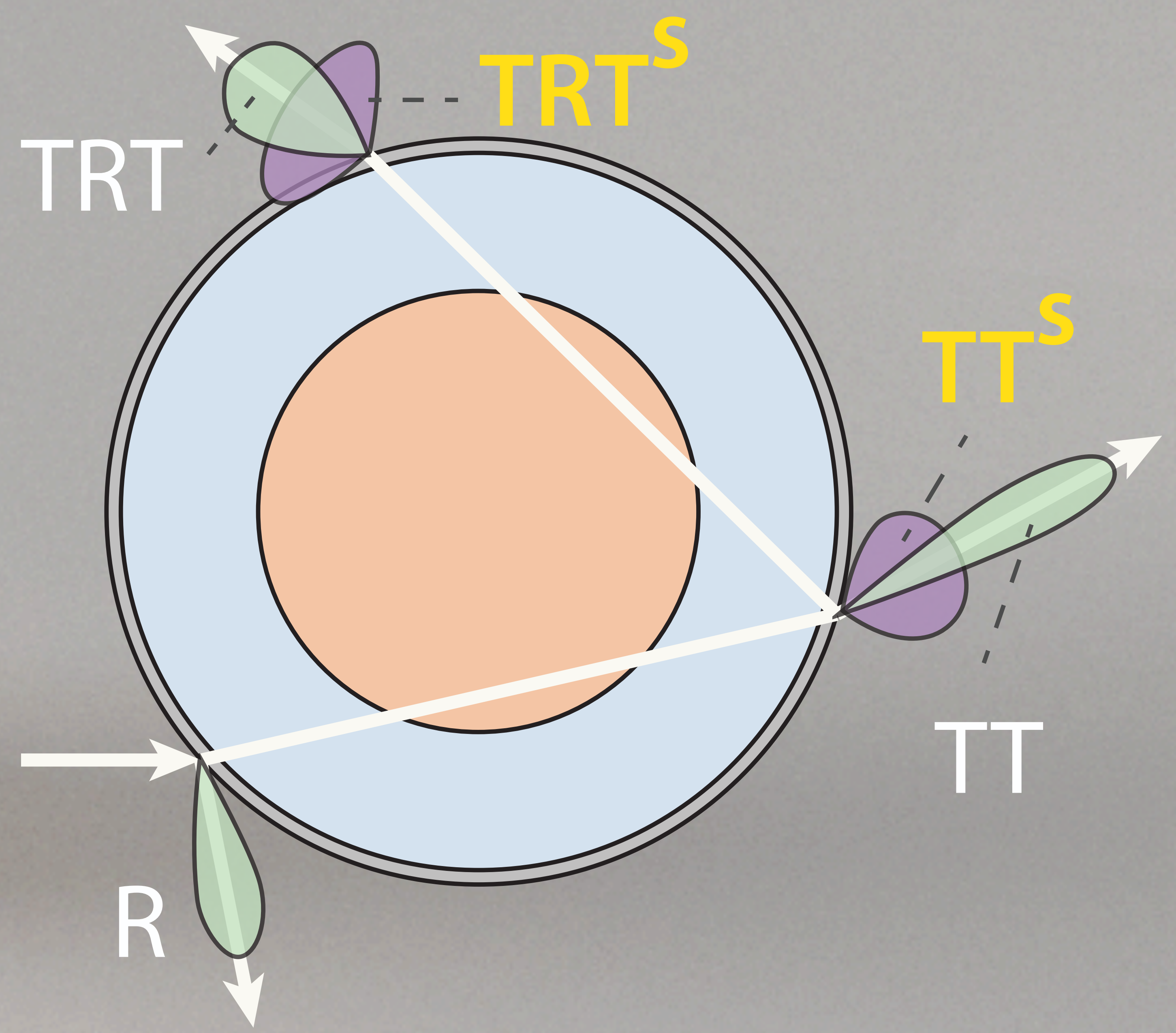
detailed appearance modeling

real-time ray tracing

Overview

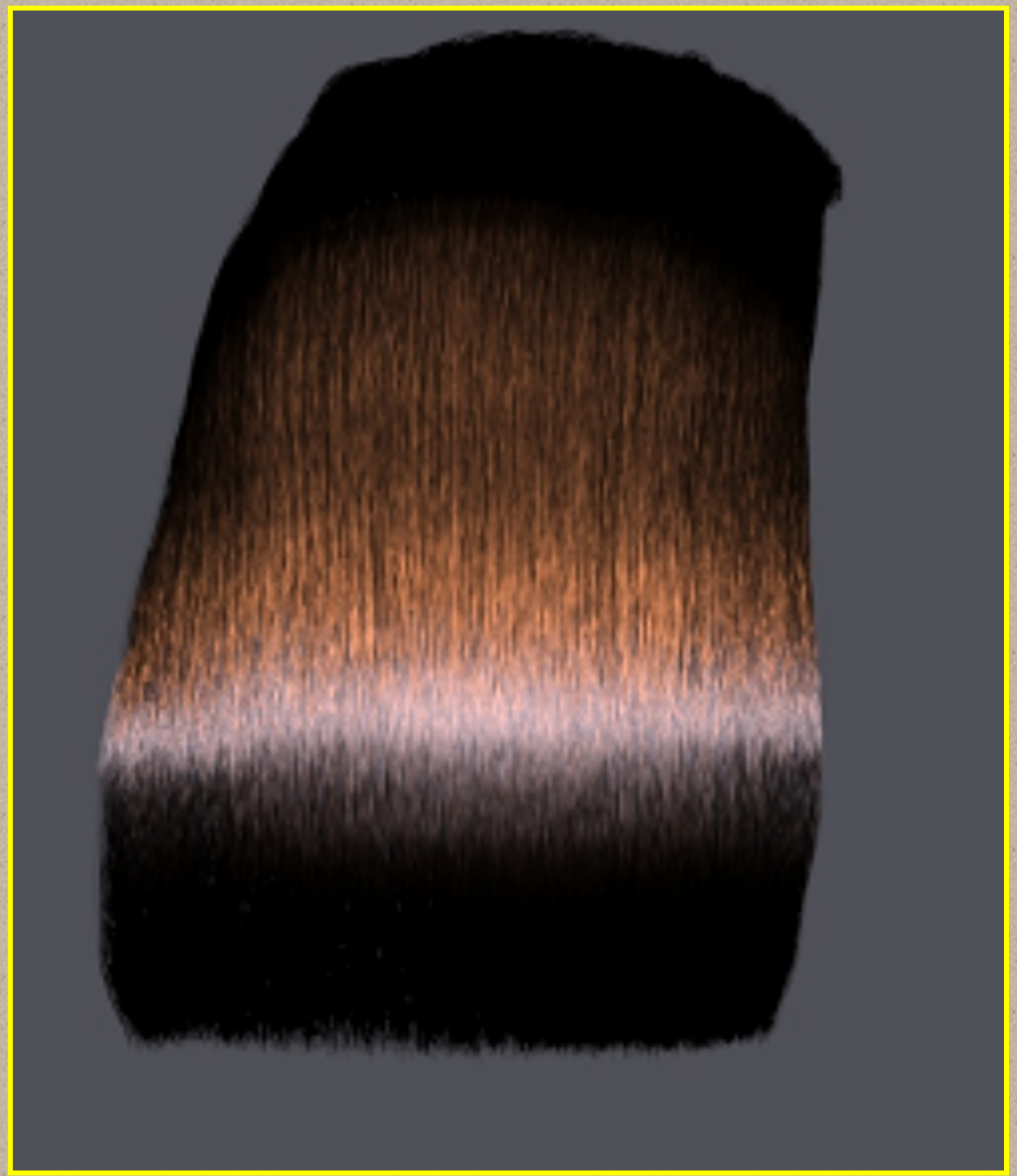
- Fur **appearance** model
- Assuming
  - Individual fur fibers (no concept of surface)

- Figuring out
  - how the light interacts each fur fiber

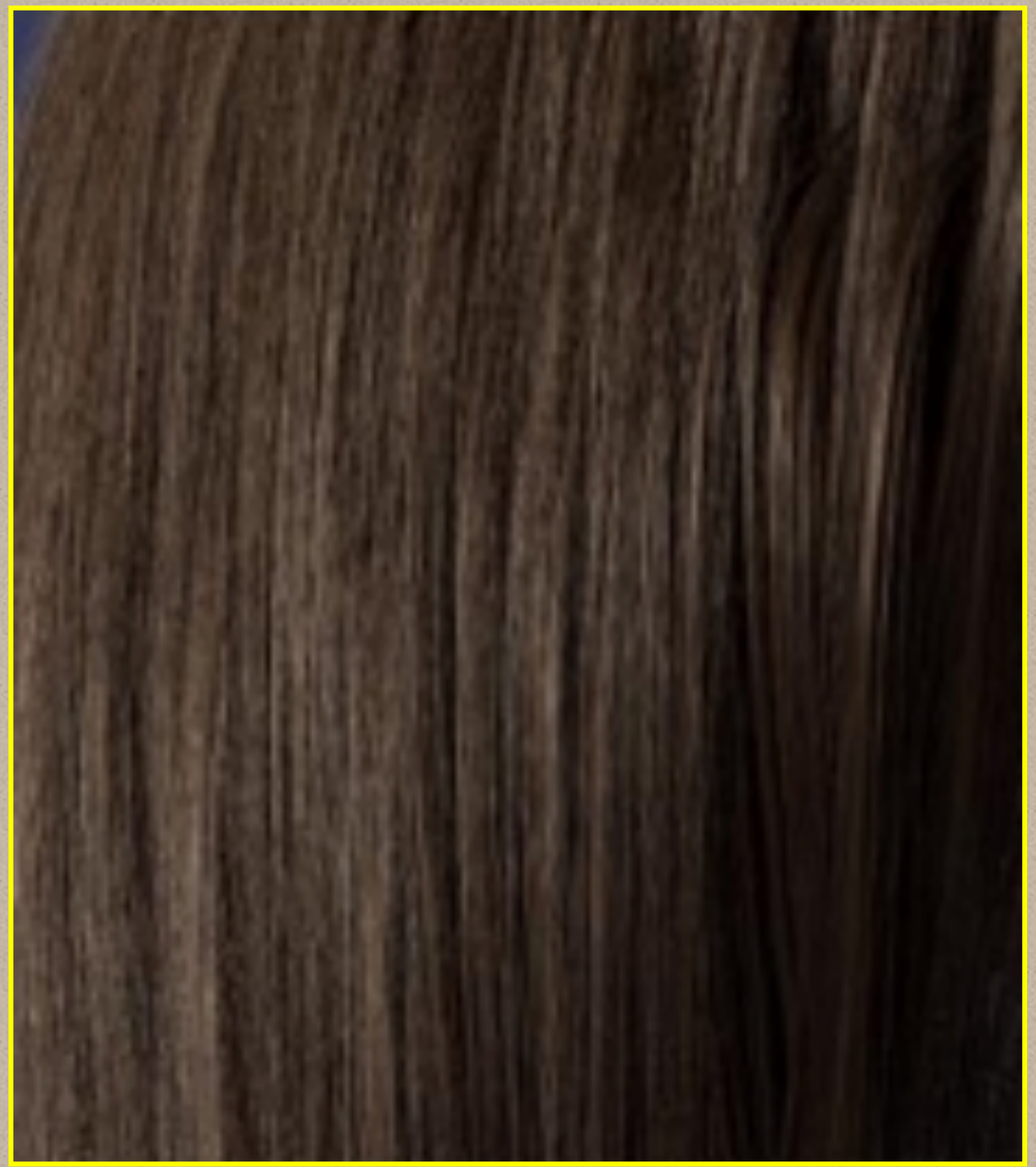


Hair Reflectance Models

- Actively developing



[Marschner 03]



[Zinke 07]

[d'Eon 11]

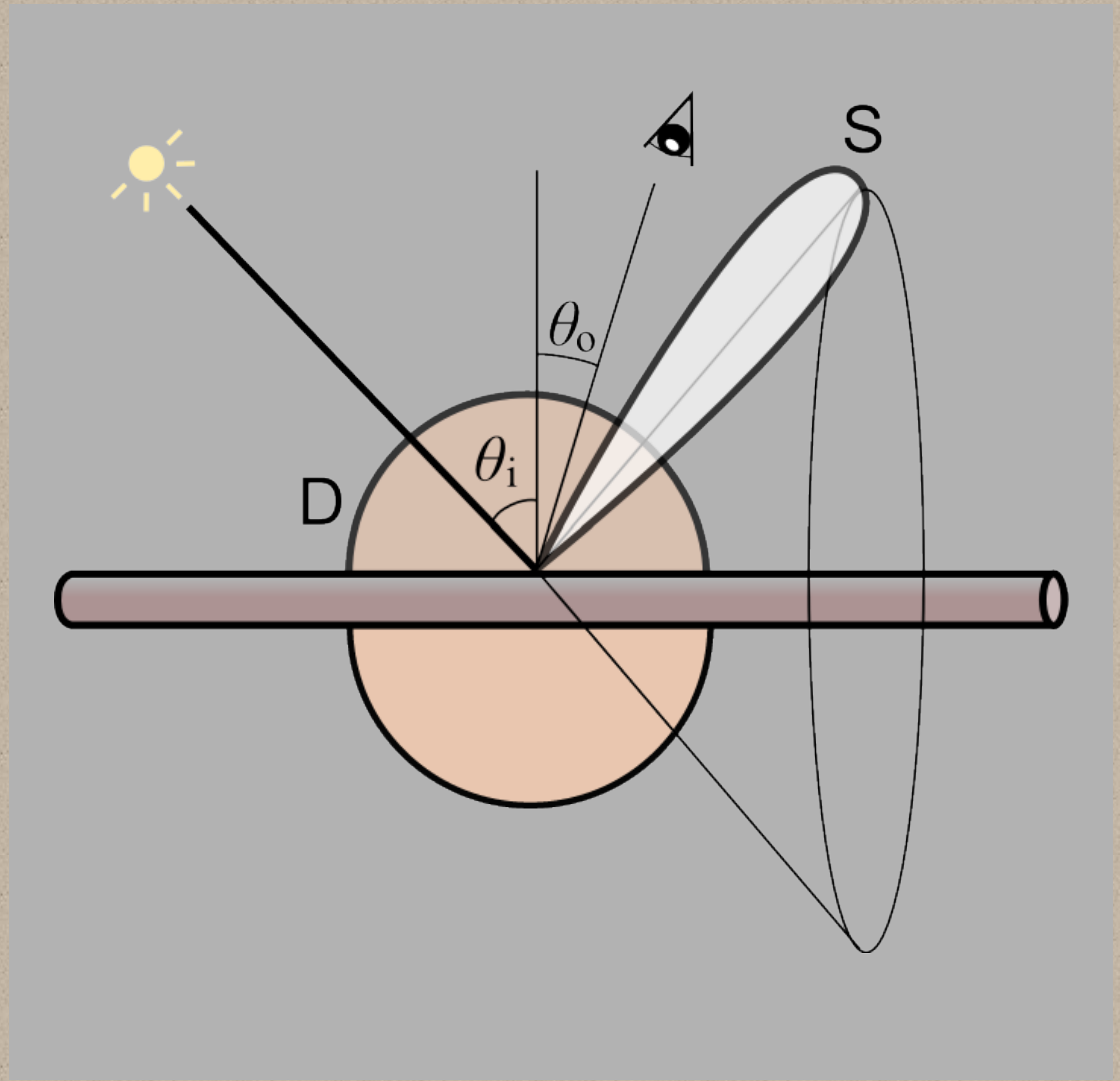
[Chiang 16]

Fur Reflectance — Kajiya-Kay Model

- Too simple to be realistic

[Kajiya & Kay 89]

[Goldman 97]



The Kajiya-Kay Model

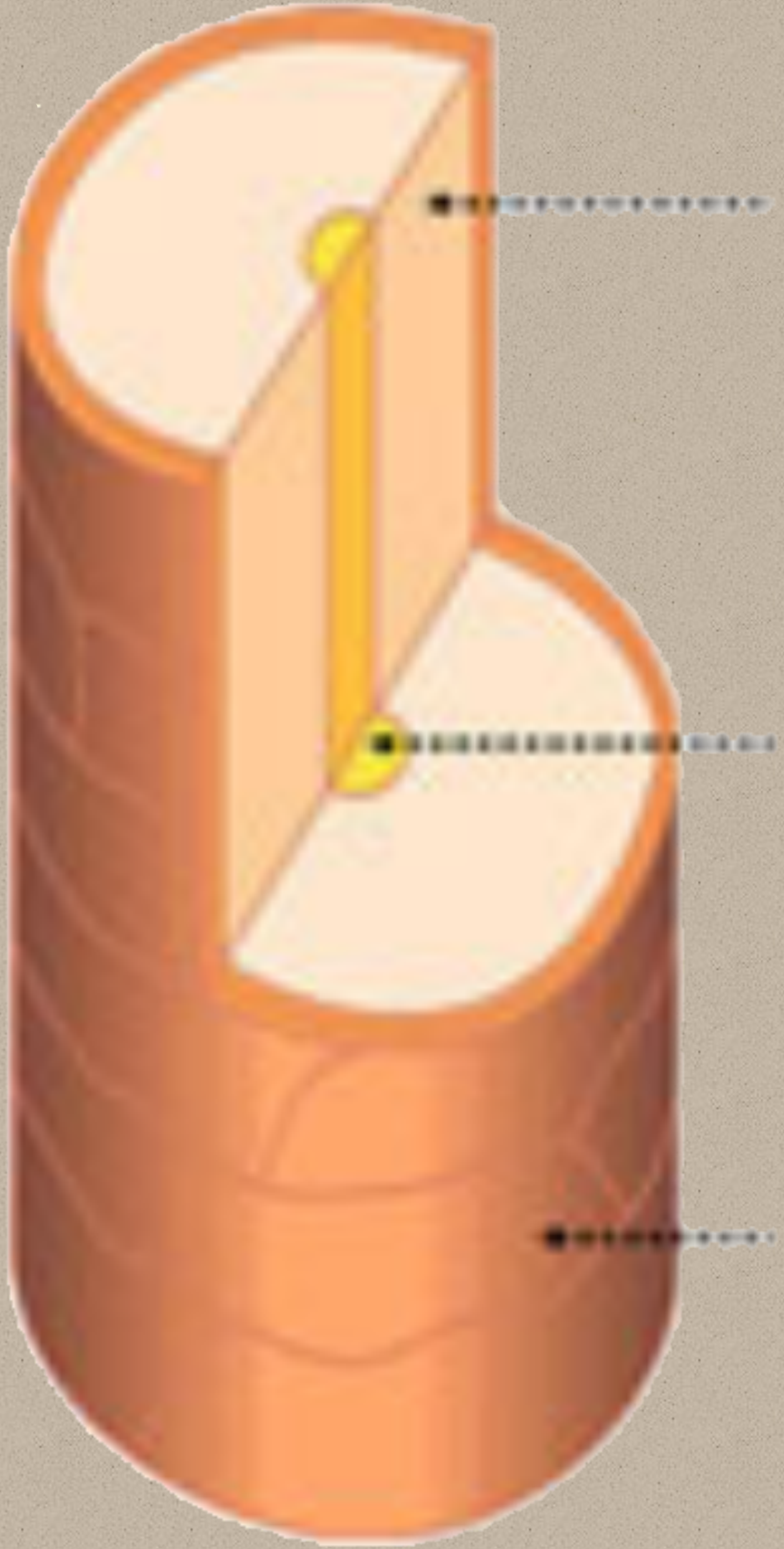
Fur Reflectance — As Human Hair

- Cannot represent diffusive and saturated appearance

Rendered as human hair
[Marschner 03]

Rendered as animal fur
[Yan 15]

Main Difference — Medulla



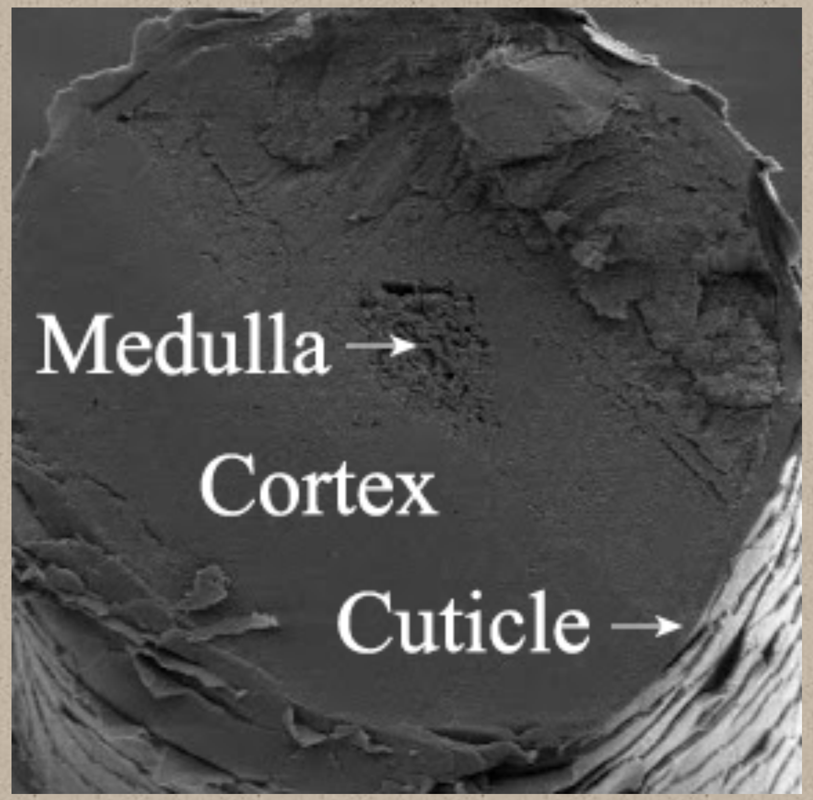
Cortex

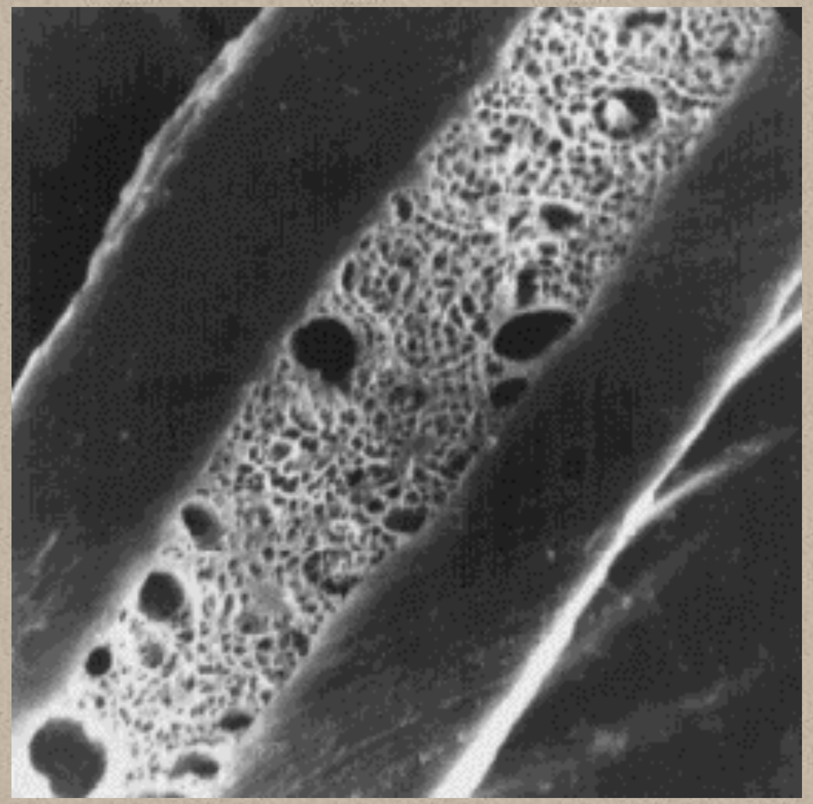
- Contains pigments
- Absorbs light

Medulla

- Complex structure
- **Scatters light**

Cuticle





Microscopic images
(Top: human, Bottom: Cougar)

Importance of Medulla

Increasing medulla size

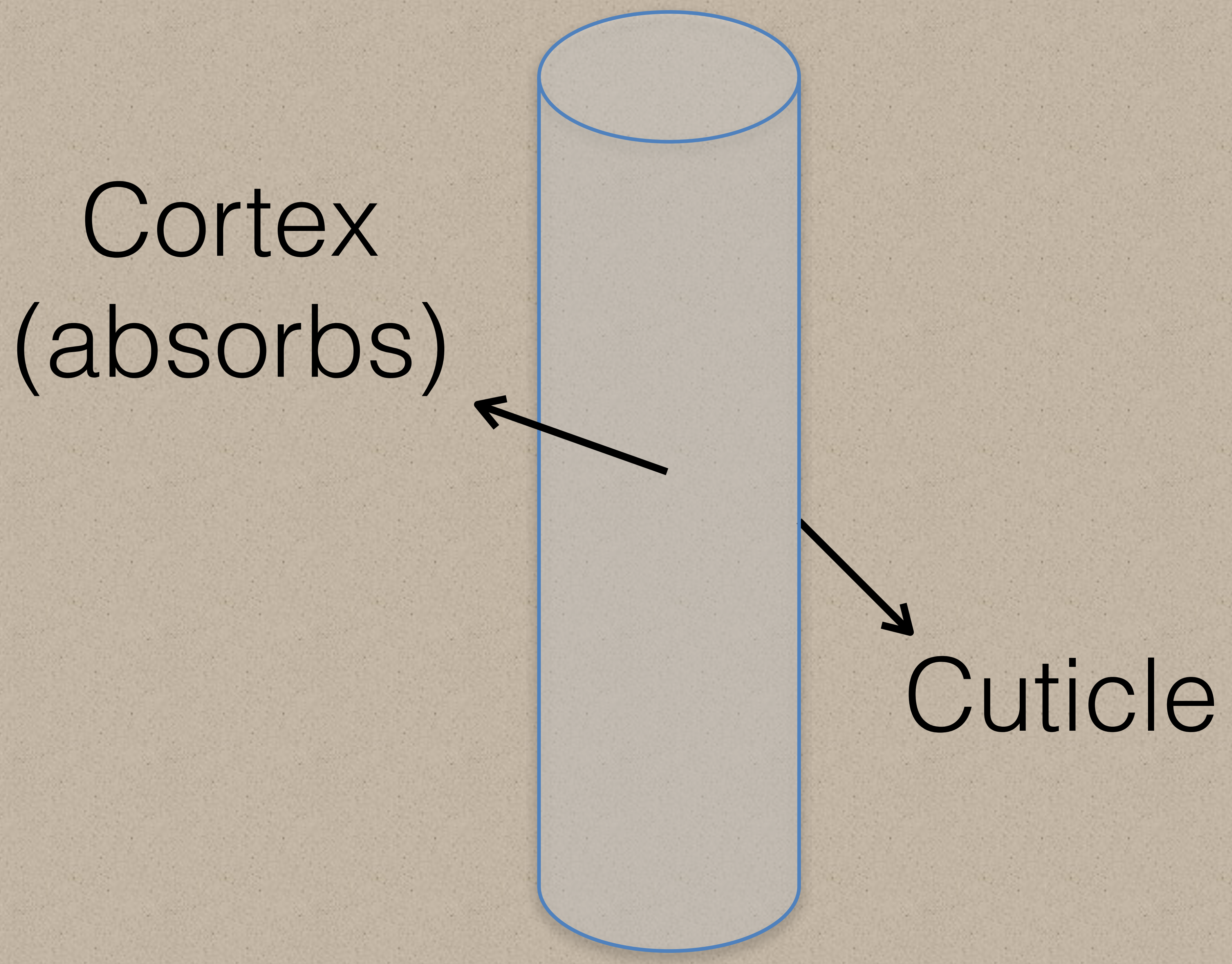
Importance of Medulla

Without medulla

With medulla (15%)

Hair Reflectance Model

- Dielectric cylinder

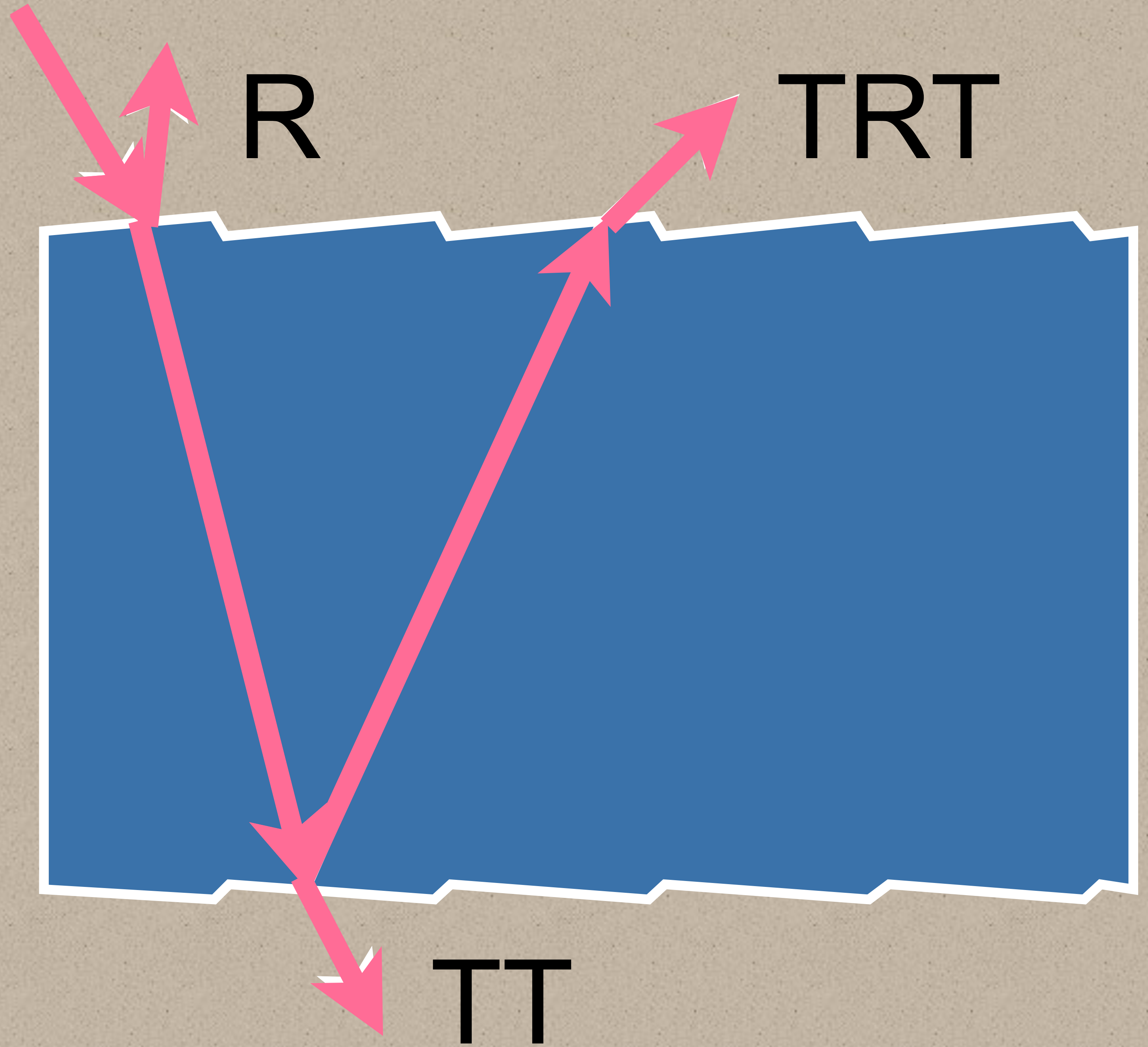


[Marschner 03]

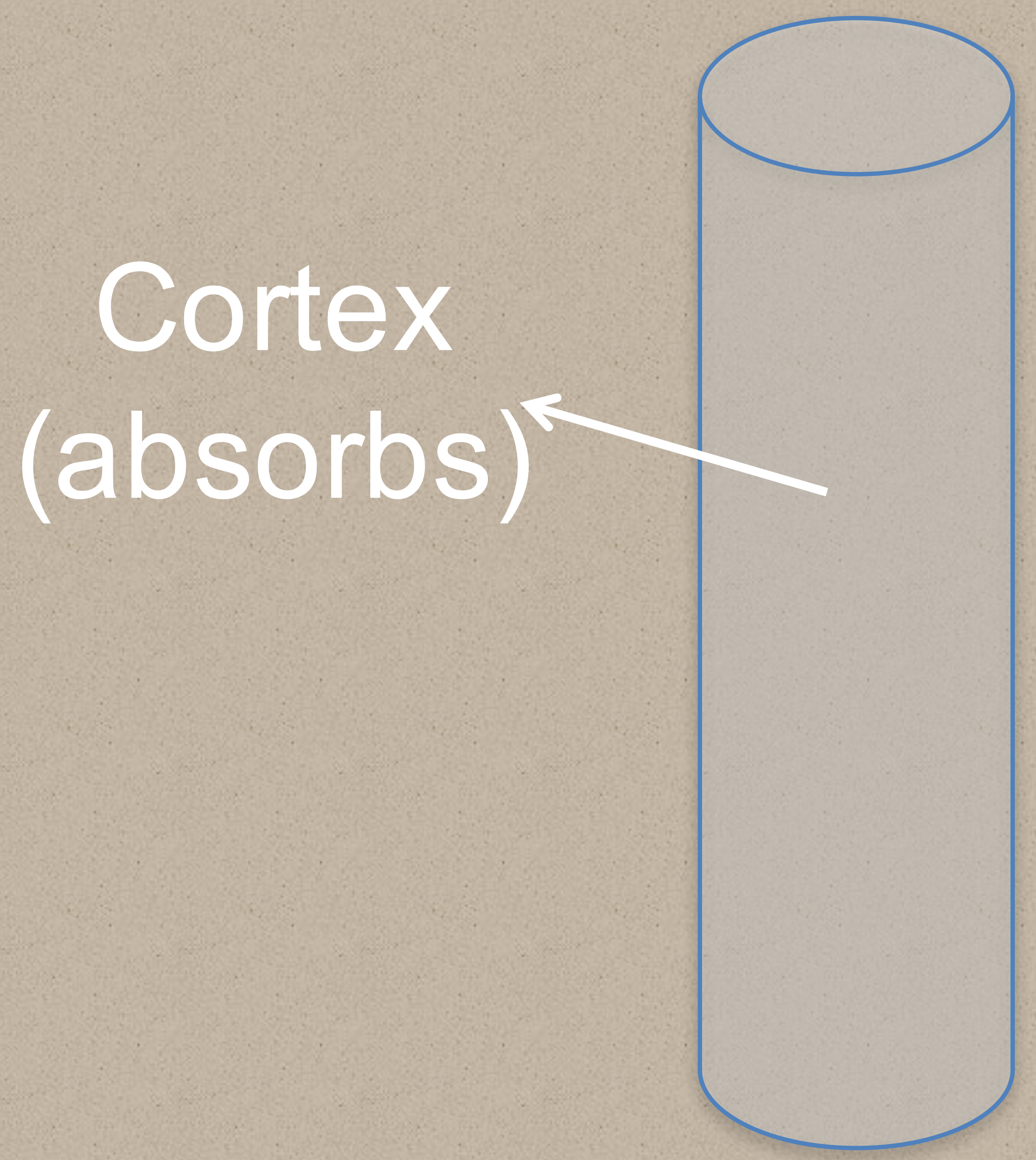
- 3 types of light interactions:

  R, **TT**, **TRT**

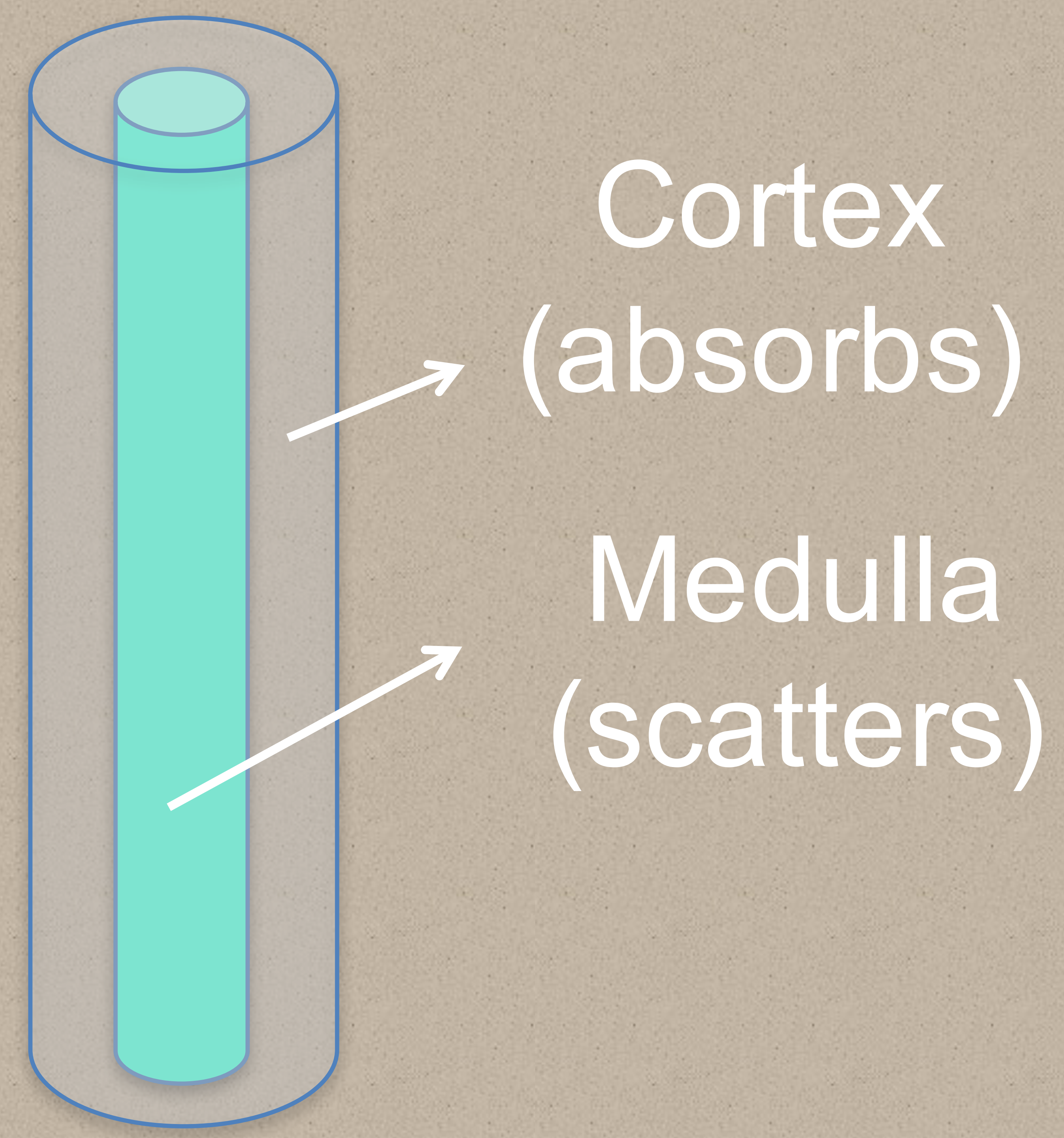
  (R: reflection, T: transmission)



Fur Reflectance Model [Yan 15, 17]



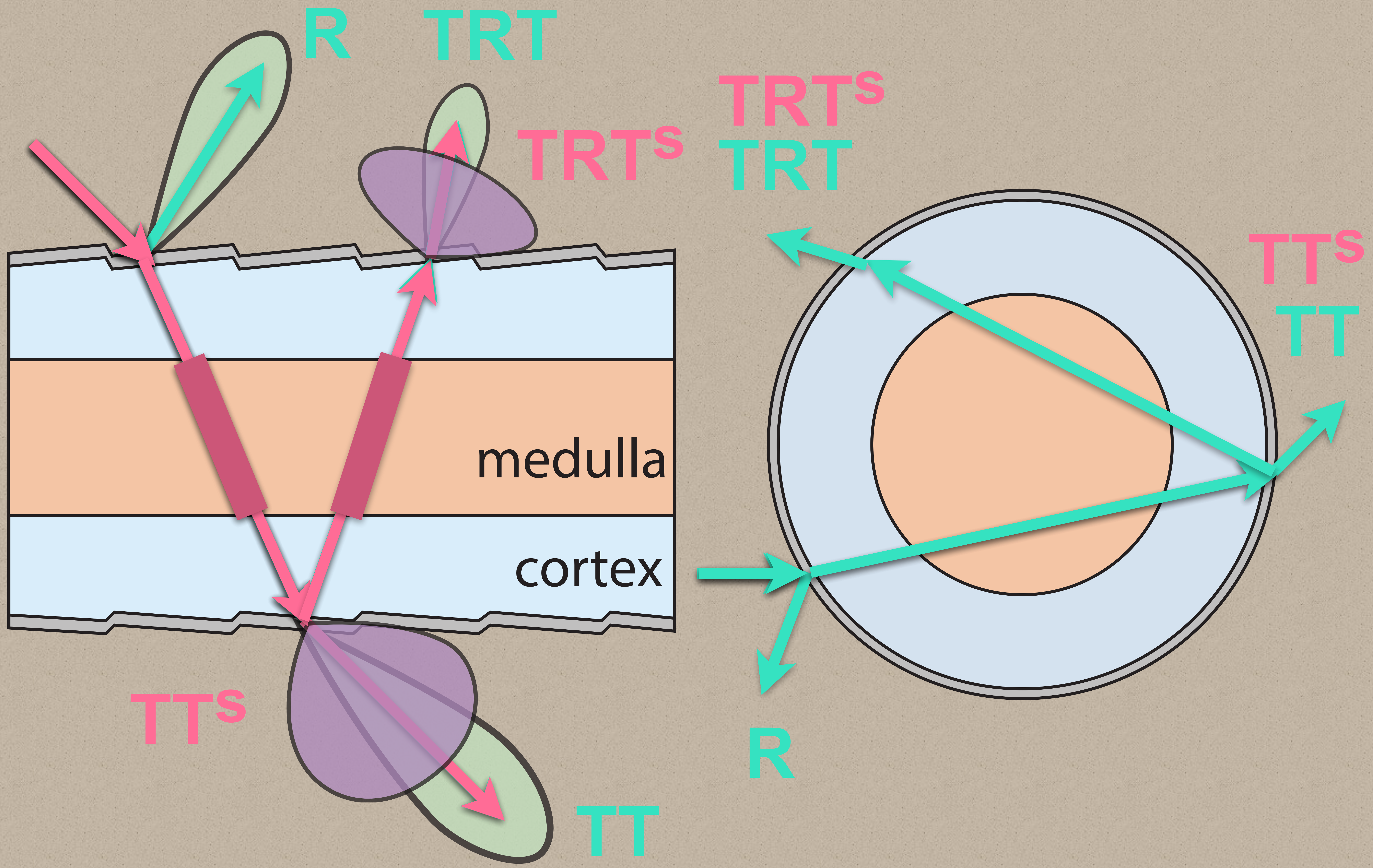
Hair model
[Marschner 03]



Double cylinder model
[Yan 15, 17]

Fur Reflectance Model — Lobes

- Unscattered lobes
  - R
  - TT
  - TRT
- Scattered lobes
  - TT^s
  - TRT^s



Visualizing Lobes

=

All R TT TRT TT^s TRT^s


1024 SPP
14.1 min / frame

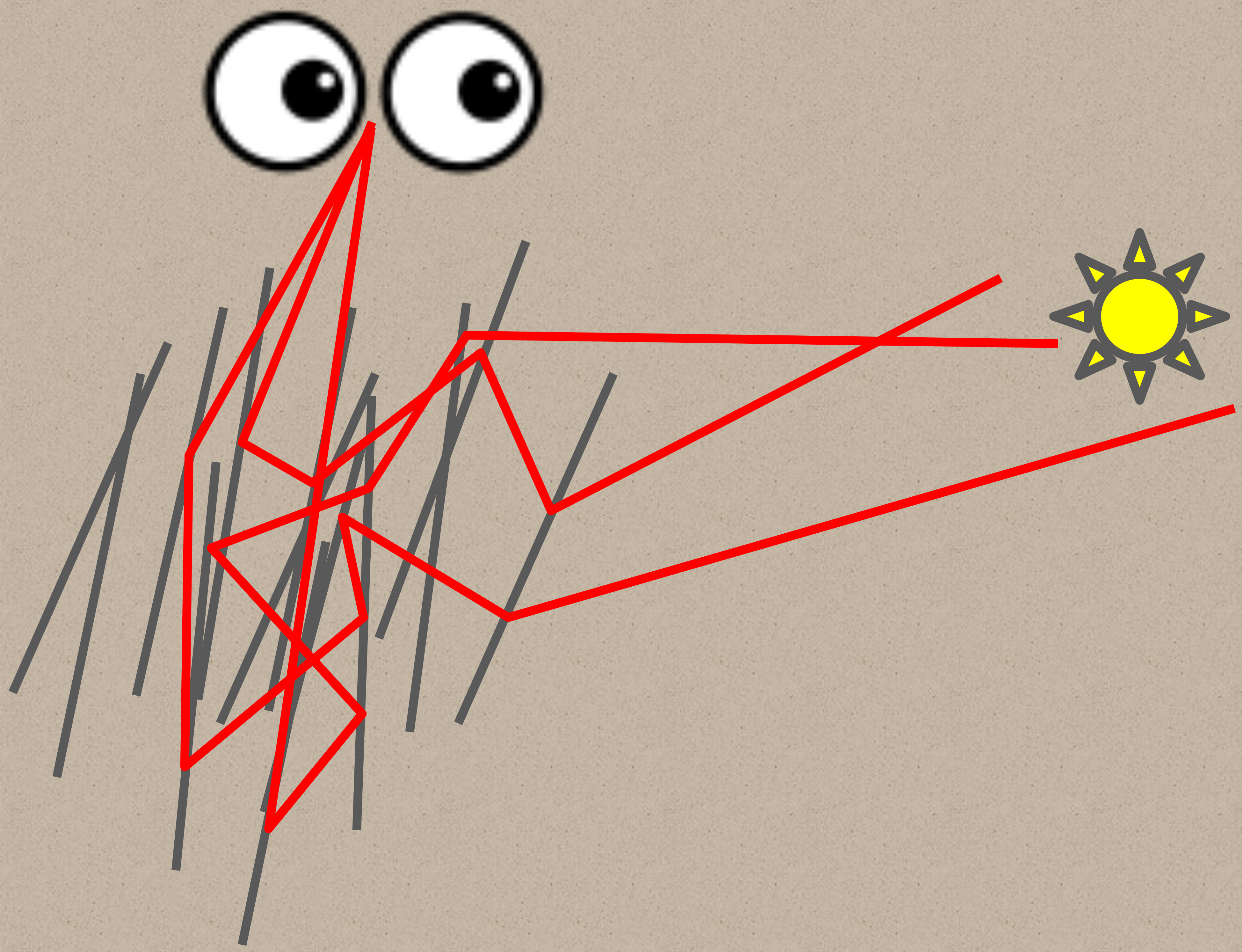
Applied in *War for the Planet of the Apes,* by Weta Digital

Further speed up

- Render using ray tracing
  - Simulating light bouncing multiple times

    light -> fur fiber #1 -> fur fiber #2 -> …

    -> fur fiber #100 -> … -> eye
  - Slow
- Can we avoid tracing multiple bounces?



Motivation

Very similar!

Between Physical Systems

- Use a Neural Network
  - 2 hidden layers
  - 10 nodes per layer
  - fully connected

fur fibers' properties

- thickness of cuticle
- size of medulla
- scattering coeff
- absorption coeff
- …

→ **Neural Network** →

clouds' properties

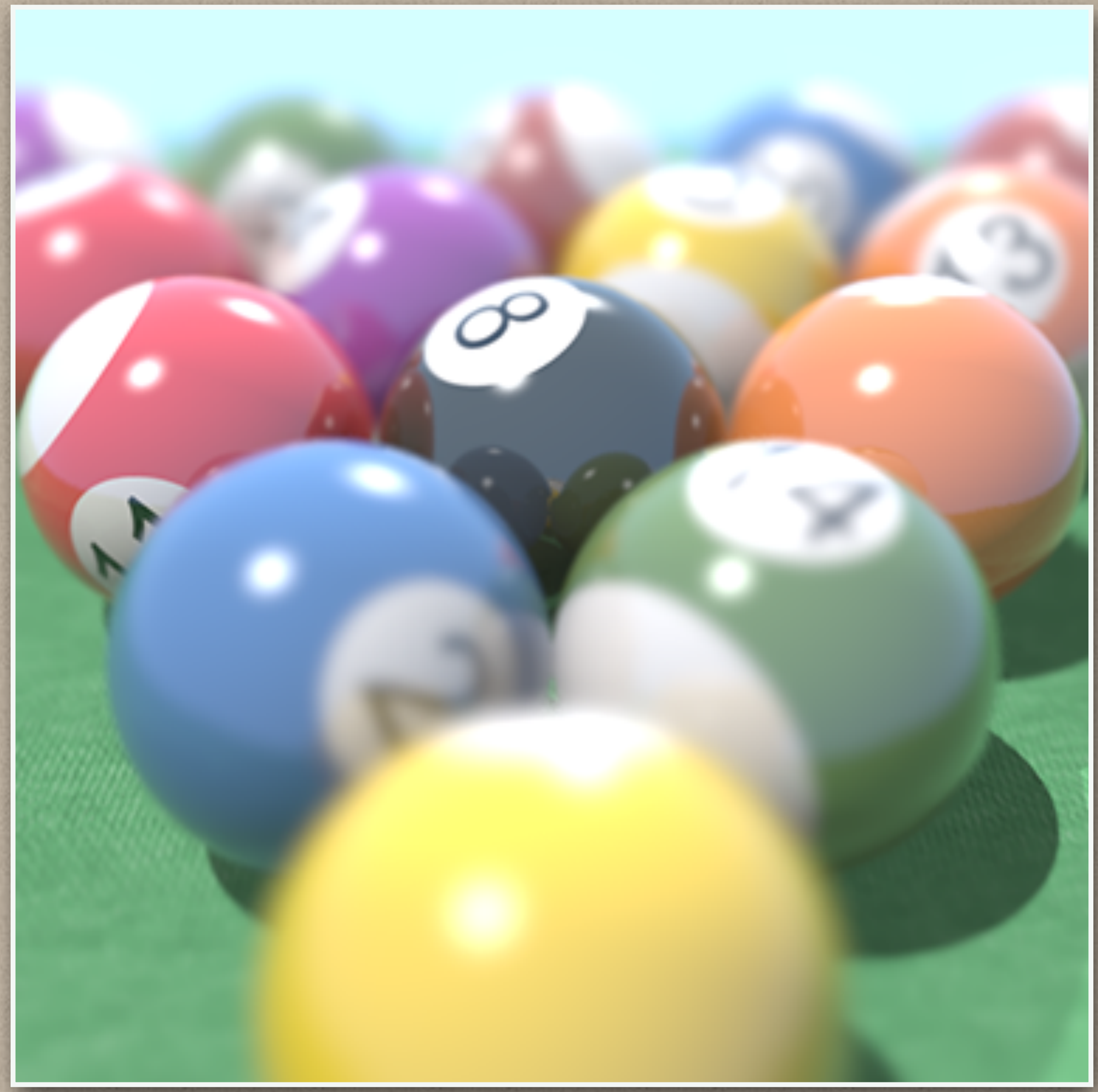
- density
- scattering coeff
- absorption coeff


Ours

Part III: Real-time Ray Tracing

detailed rendering

detailed appearance modeling



real-time ray tracing

Motivation: Ray Tracing vs. Rasterization

- Ray tracing: slow / noisy
- Rasterization: fast, less realistic

Ray tracing + Filtering

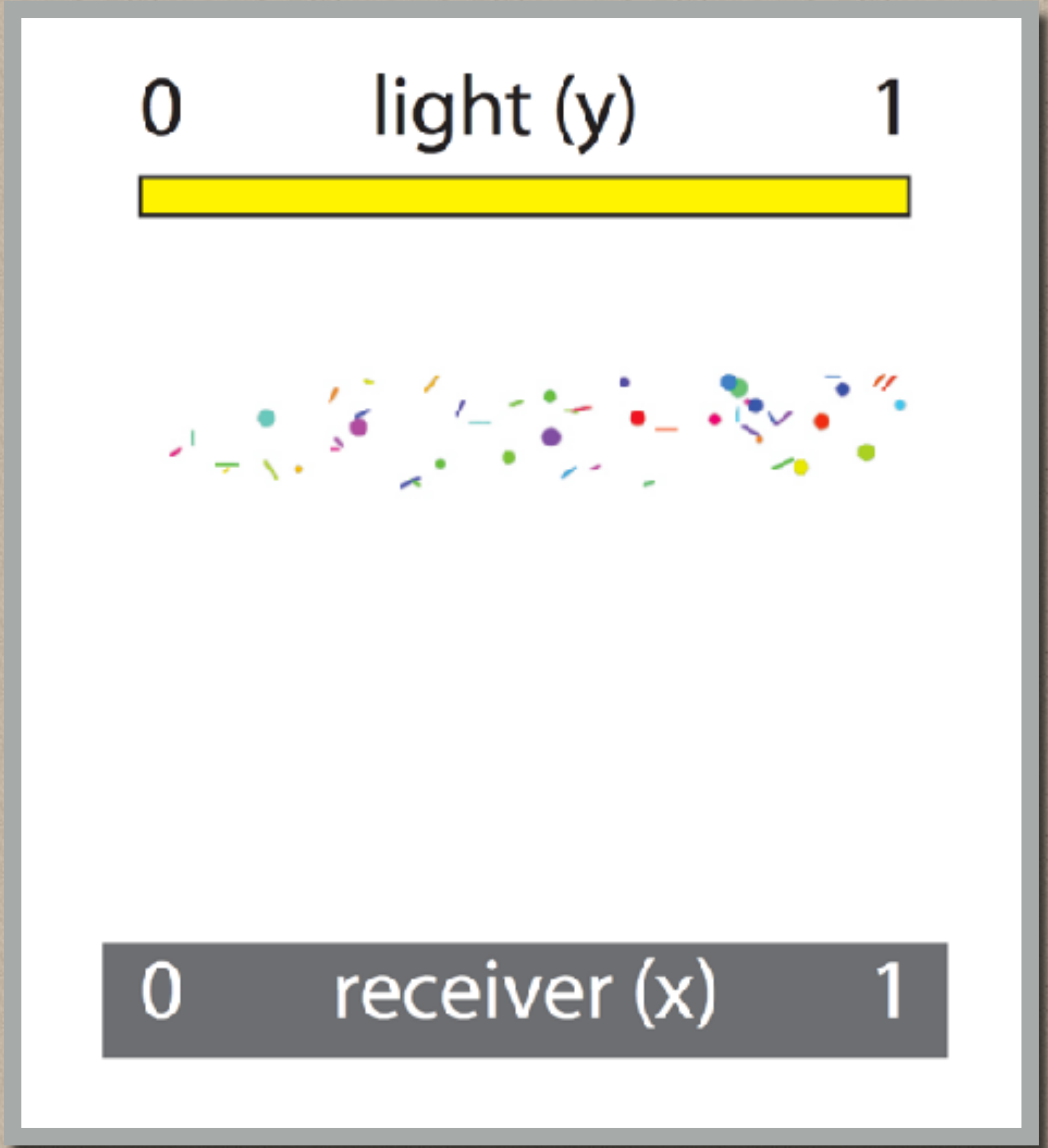
Interactive ray tracing on GPU

Car interactively rendered using NVIDIA OptiX

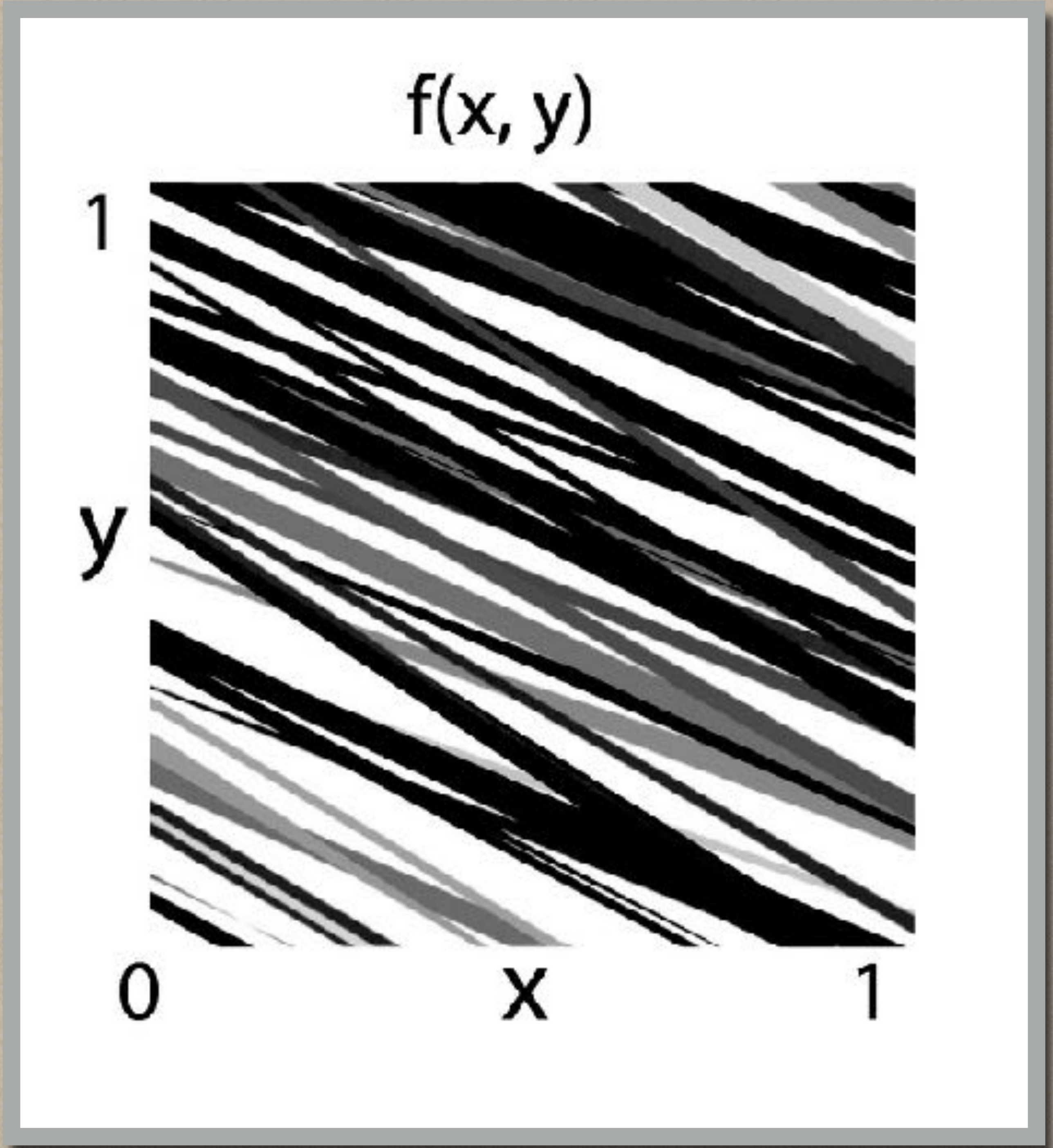
Pixar's real-time previewer

Background

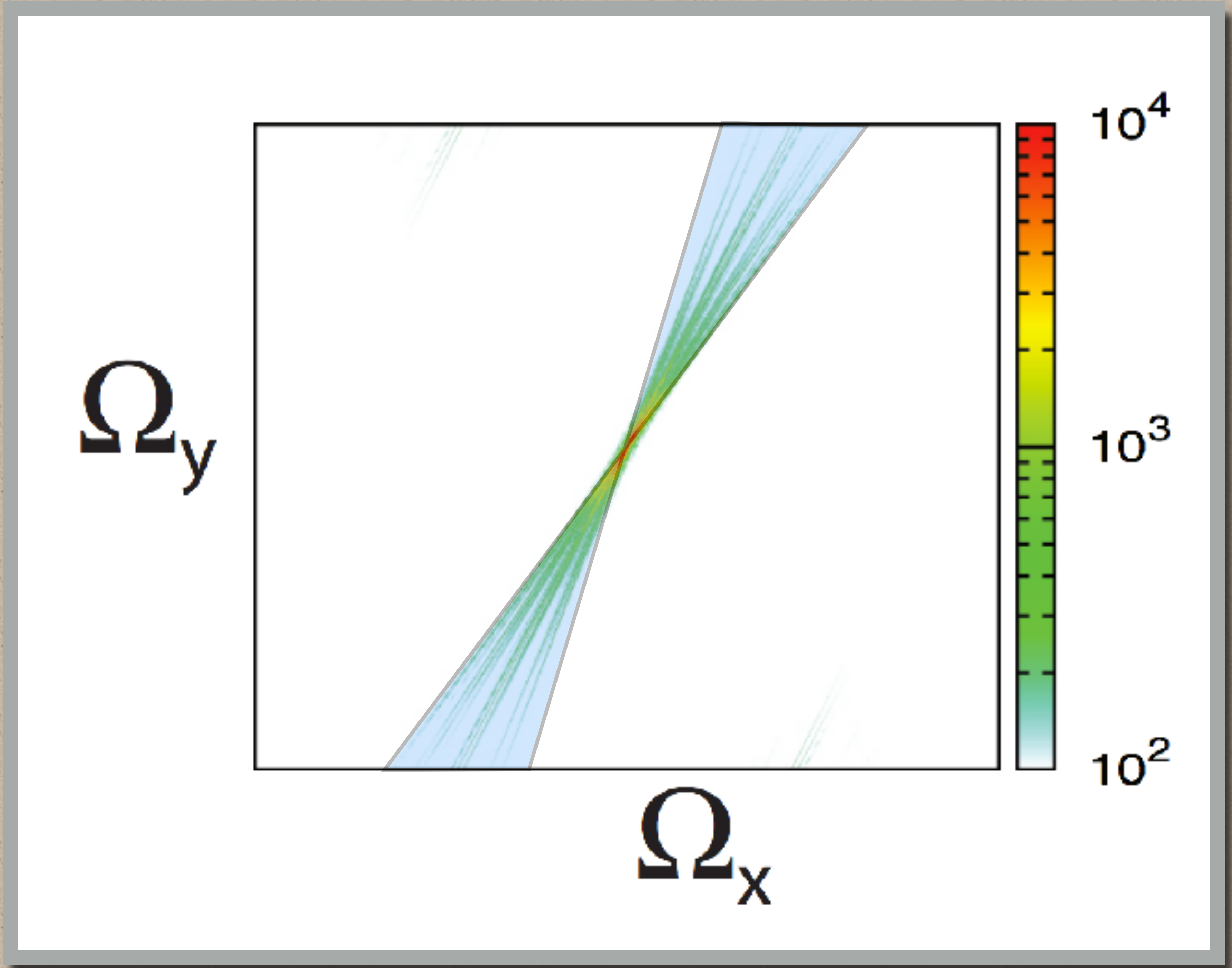
- Frequency analysis on soft shadows (flatland)



scene in flatland

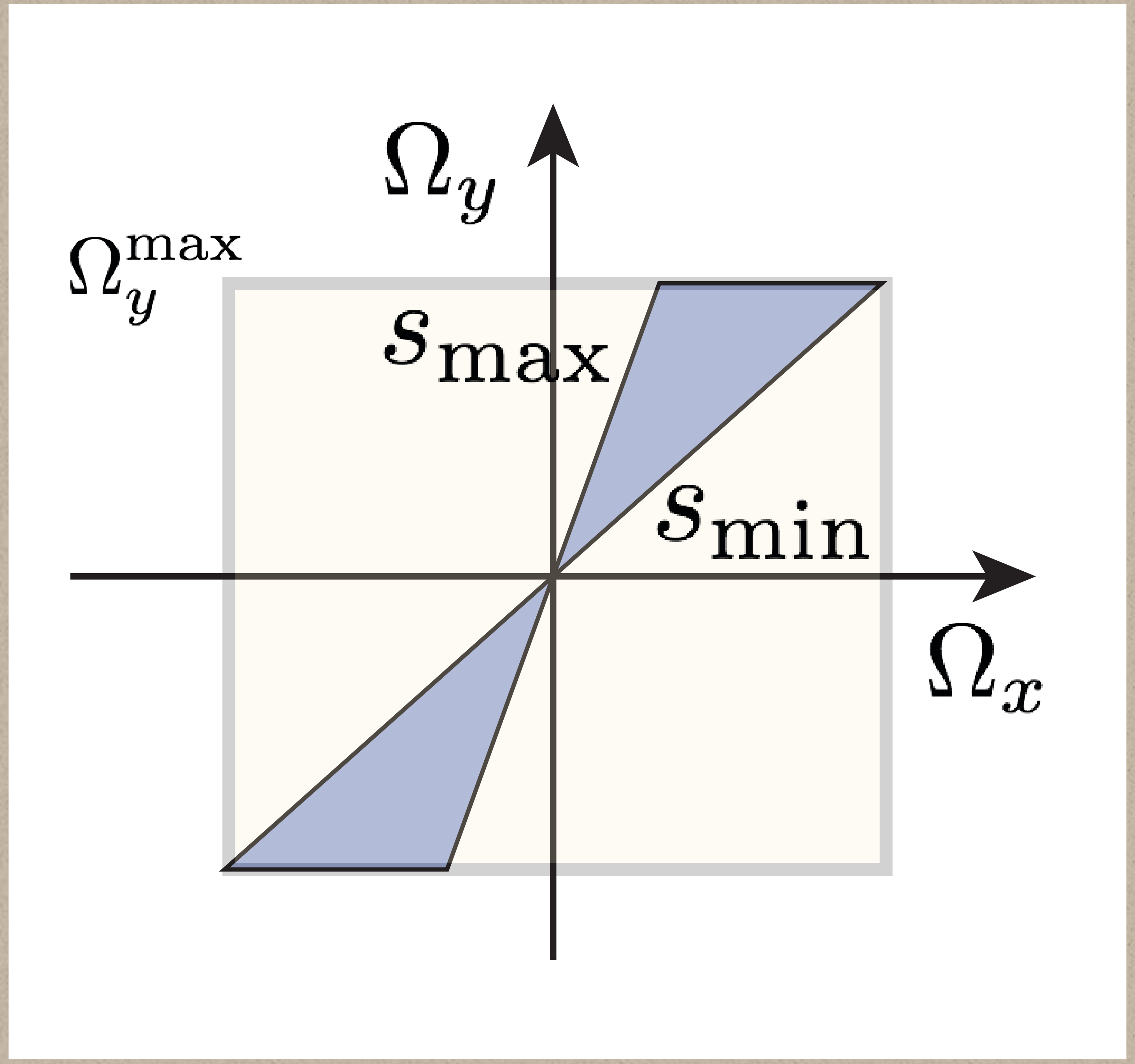


visibility function

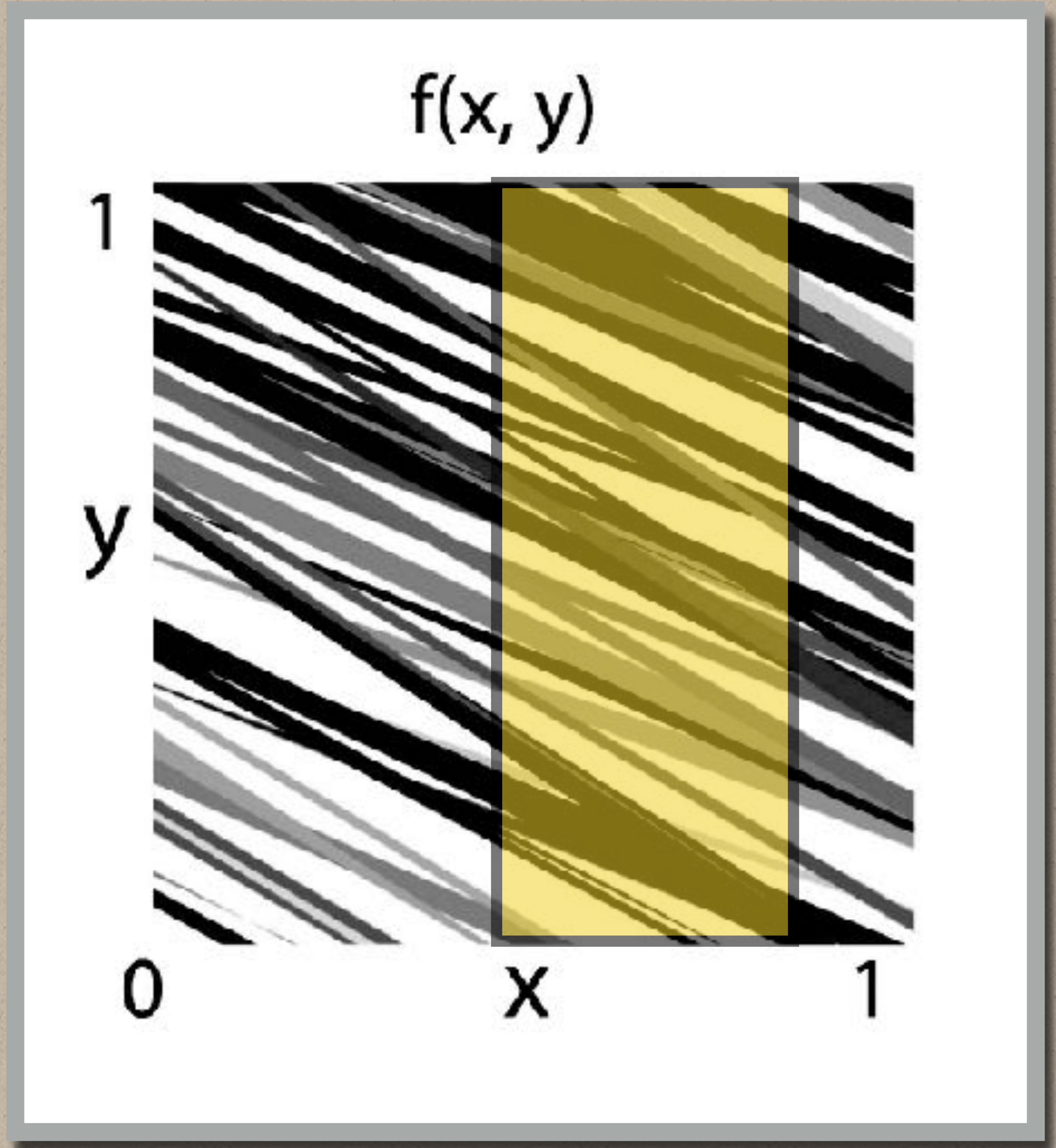


Fourier spectrum

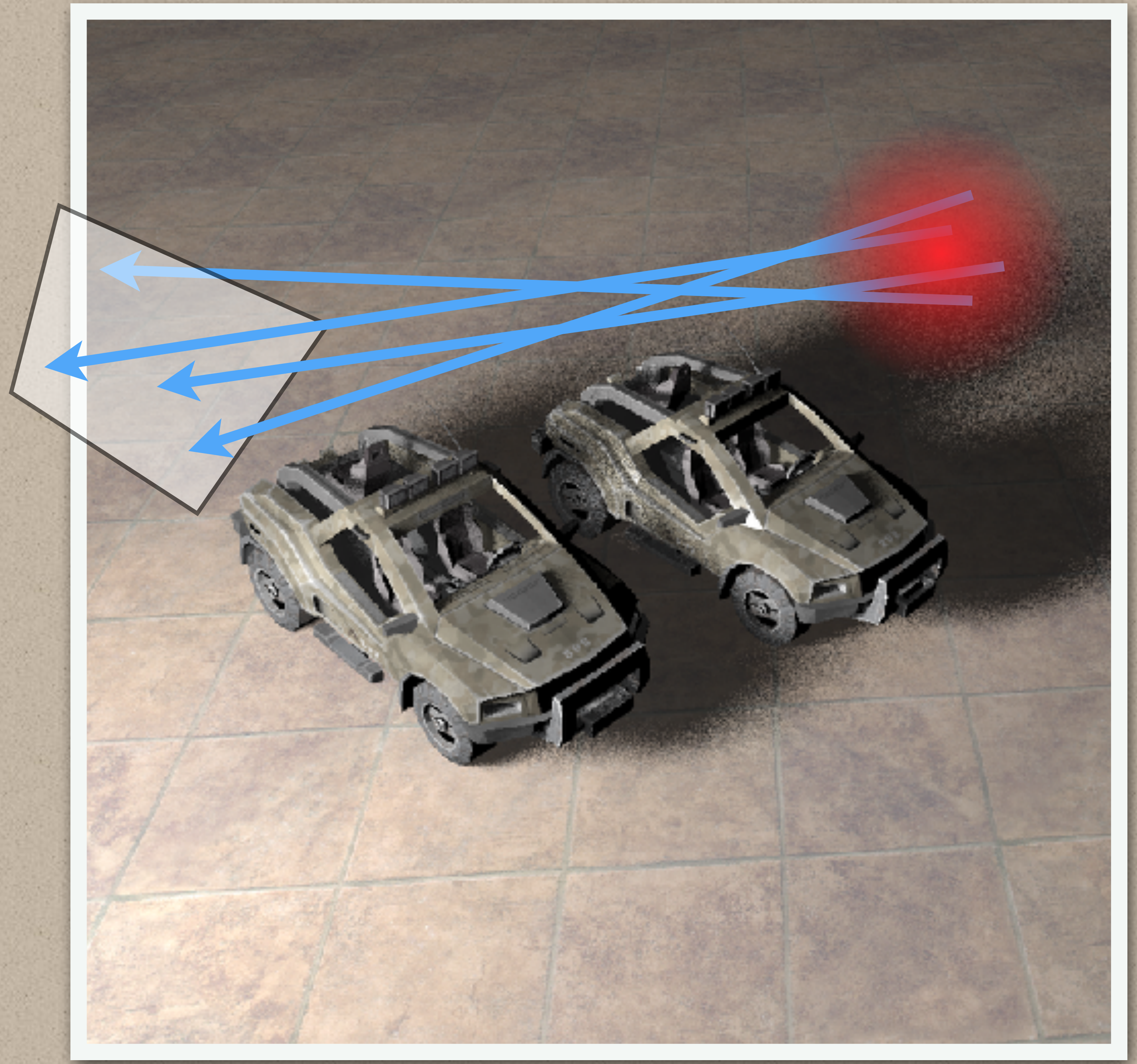
Axis-aligned filtering



Fourier domain
(flatland)



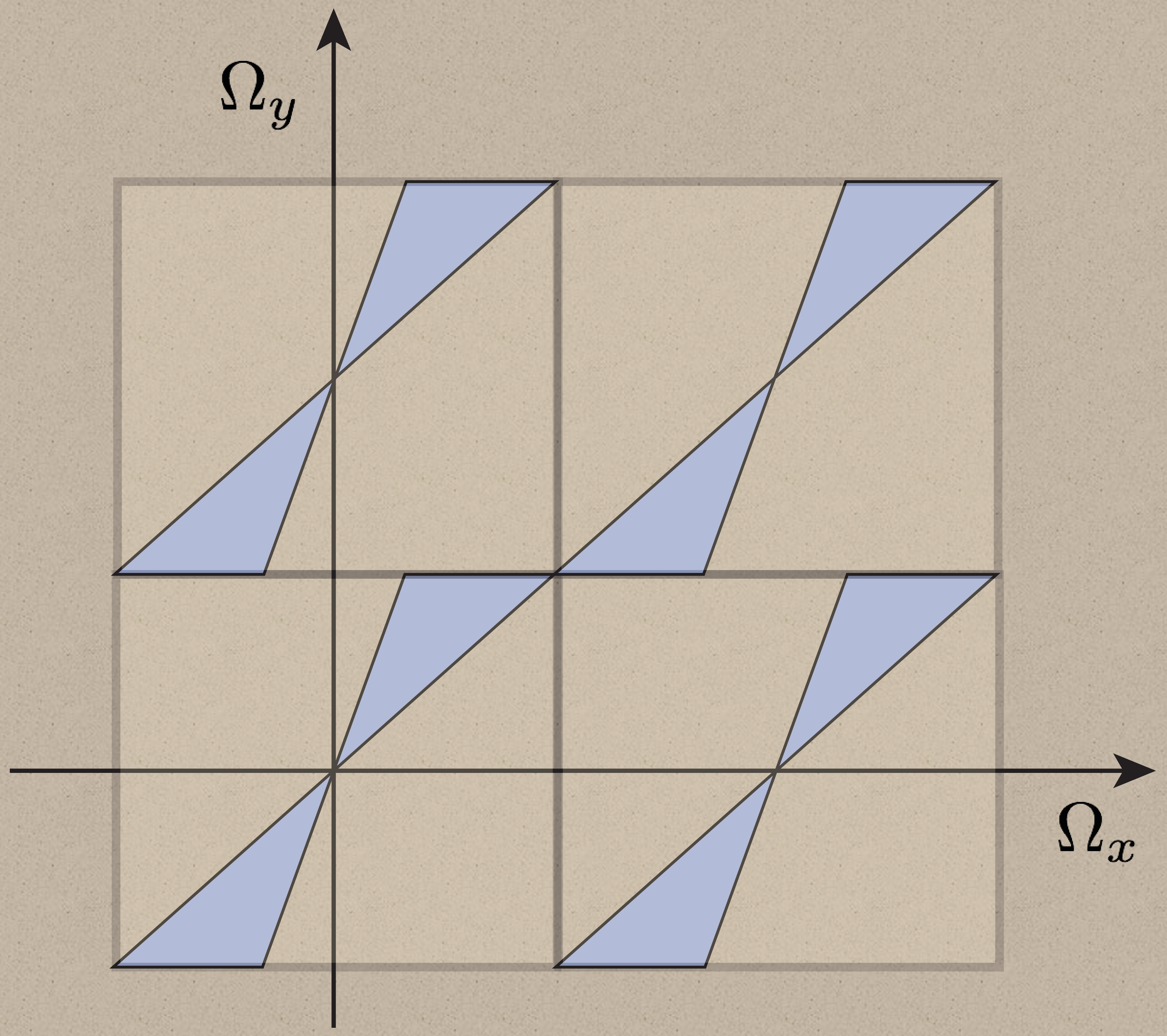
Primal domain
(flatland — 2D filter)



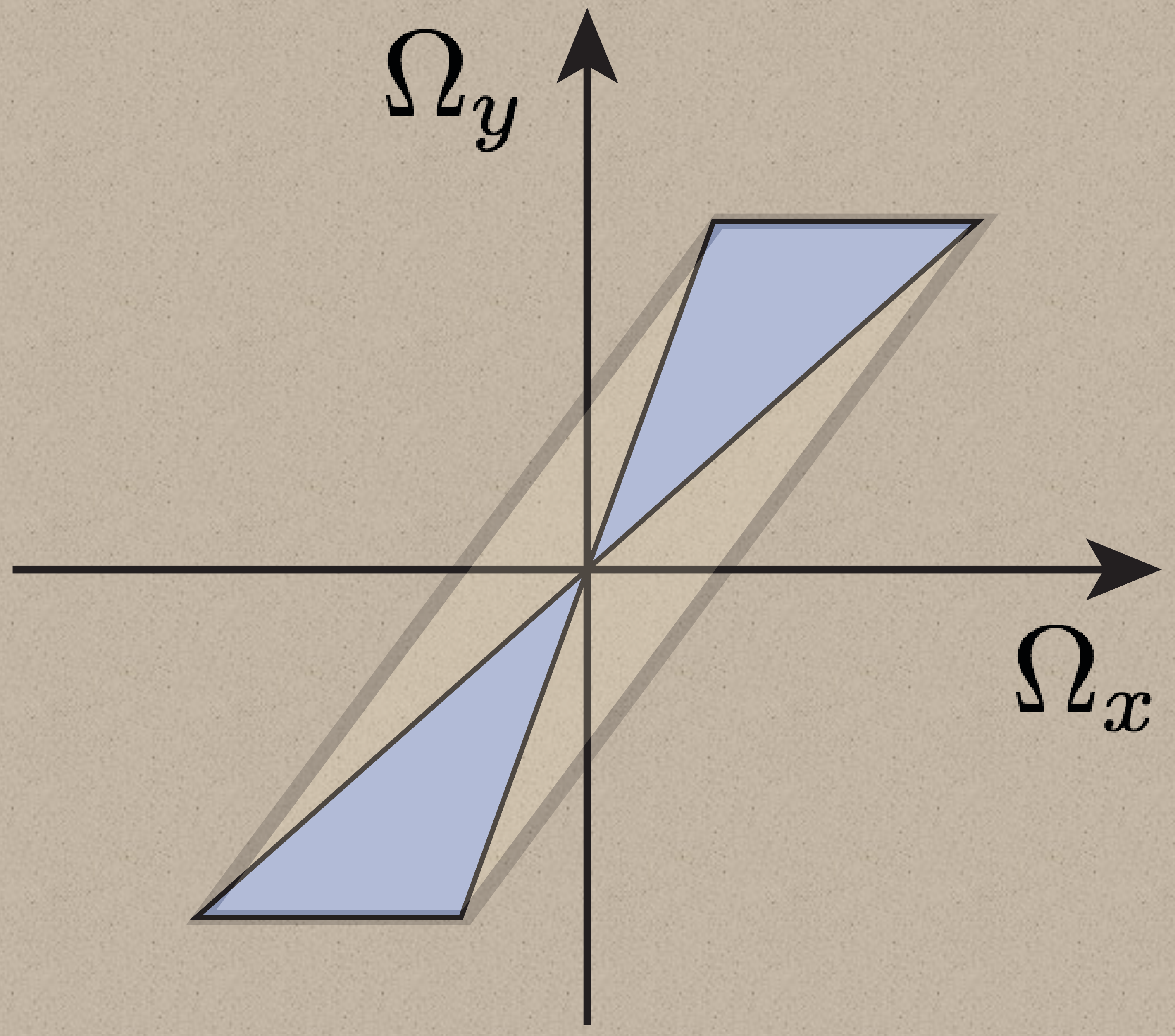
Primal domain
(scene — 4D filter)

Axis-aligned filtering

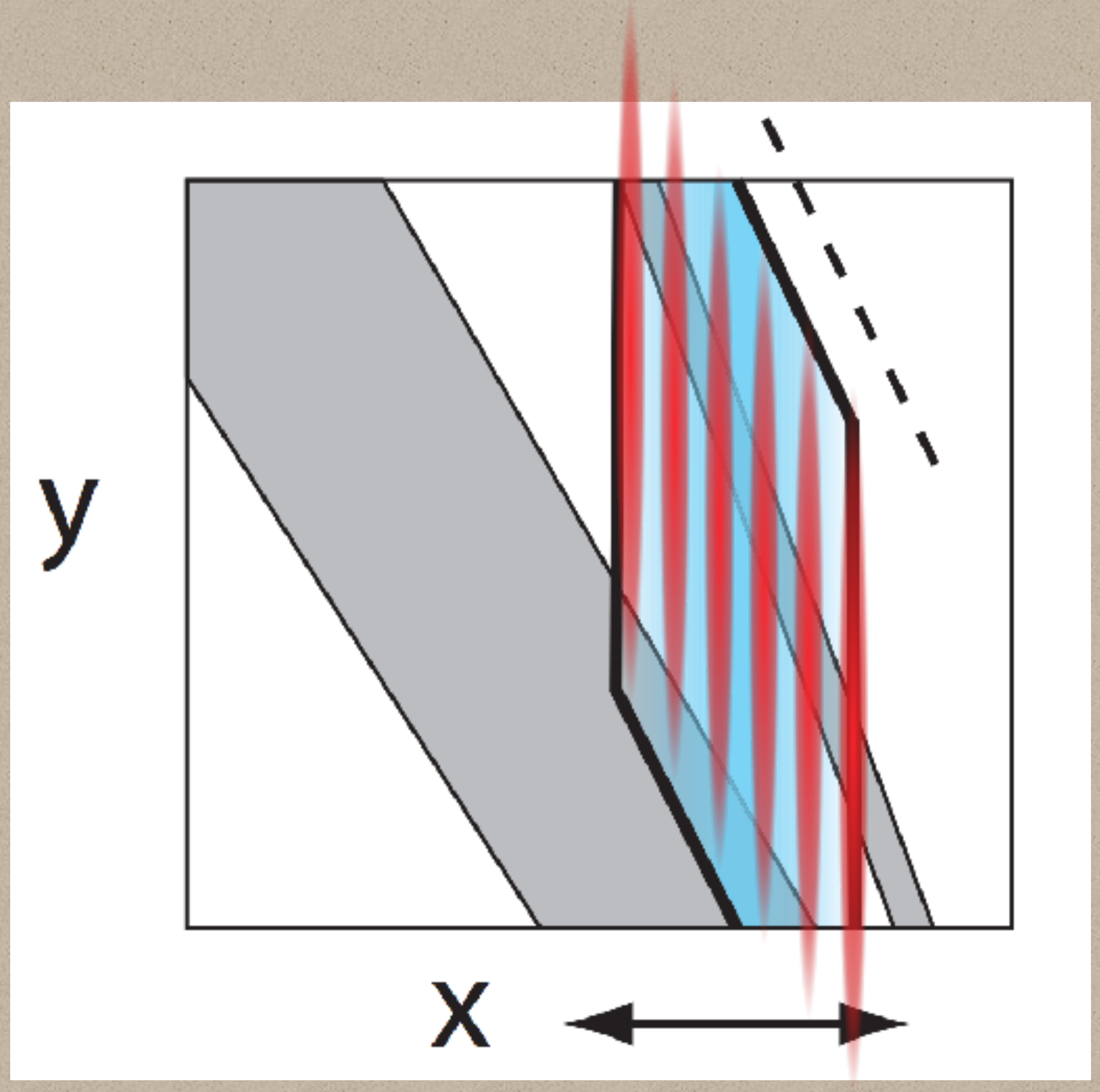
- Fourier: axis-aligned shape
  - Primal: separable filter
- Fourier: not compact
  - Primal: need more samples



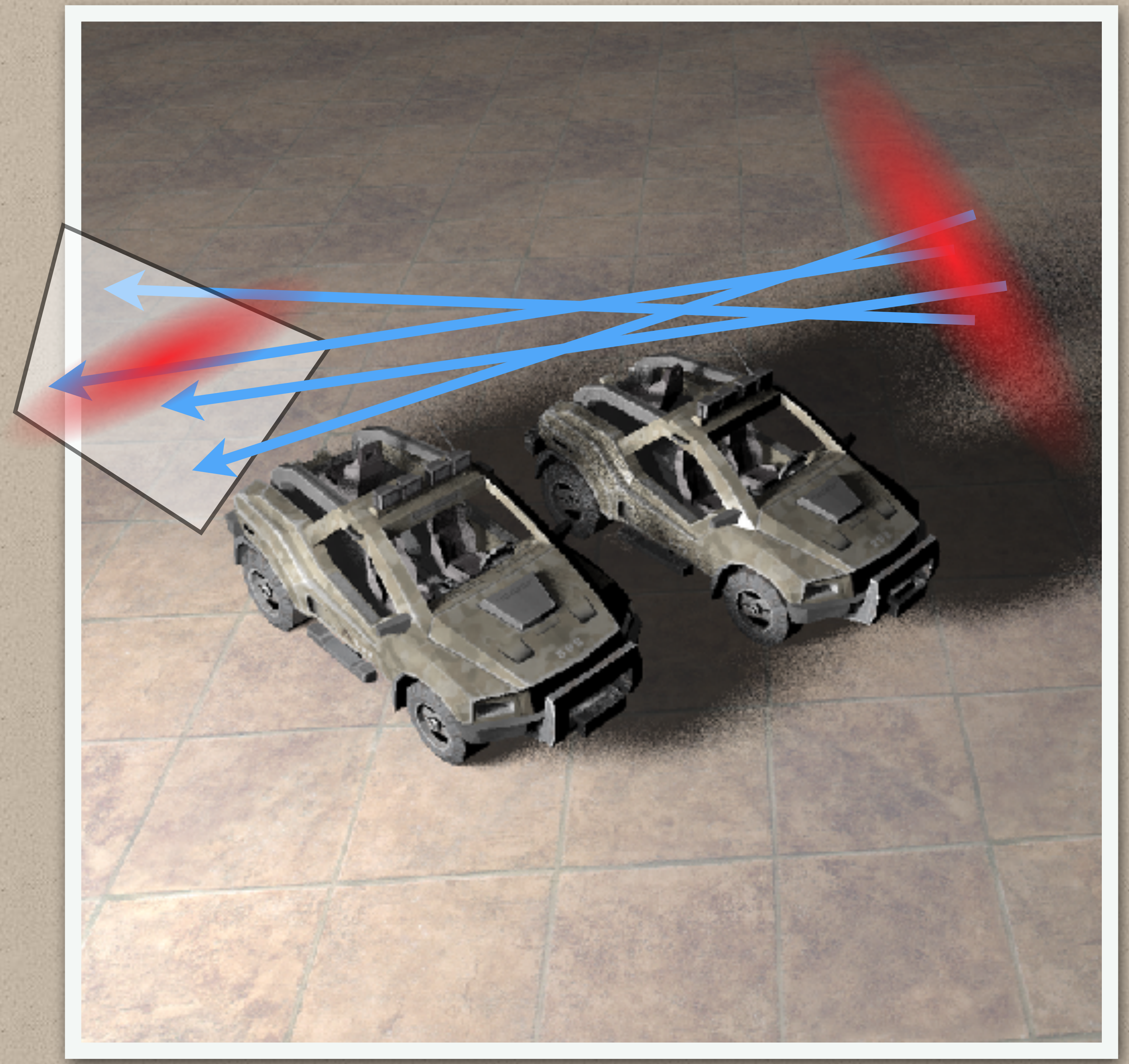
Sheared filtering



Fourier domain

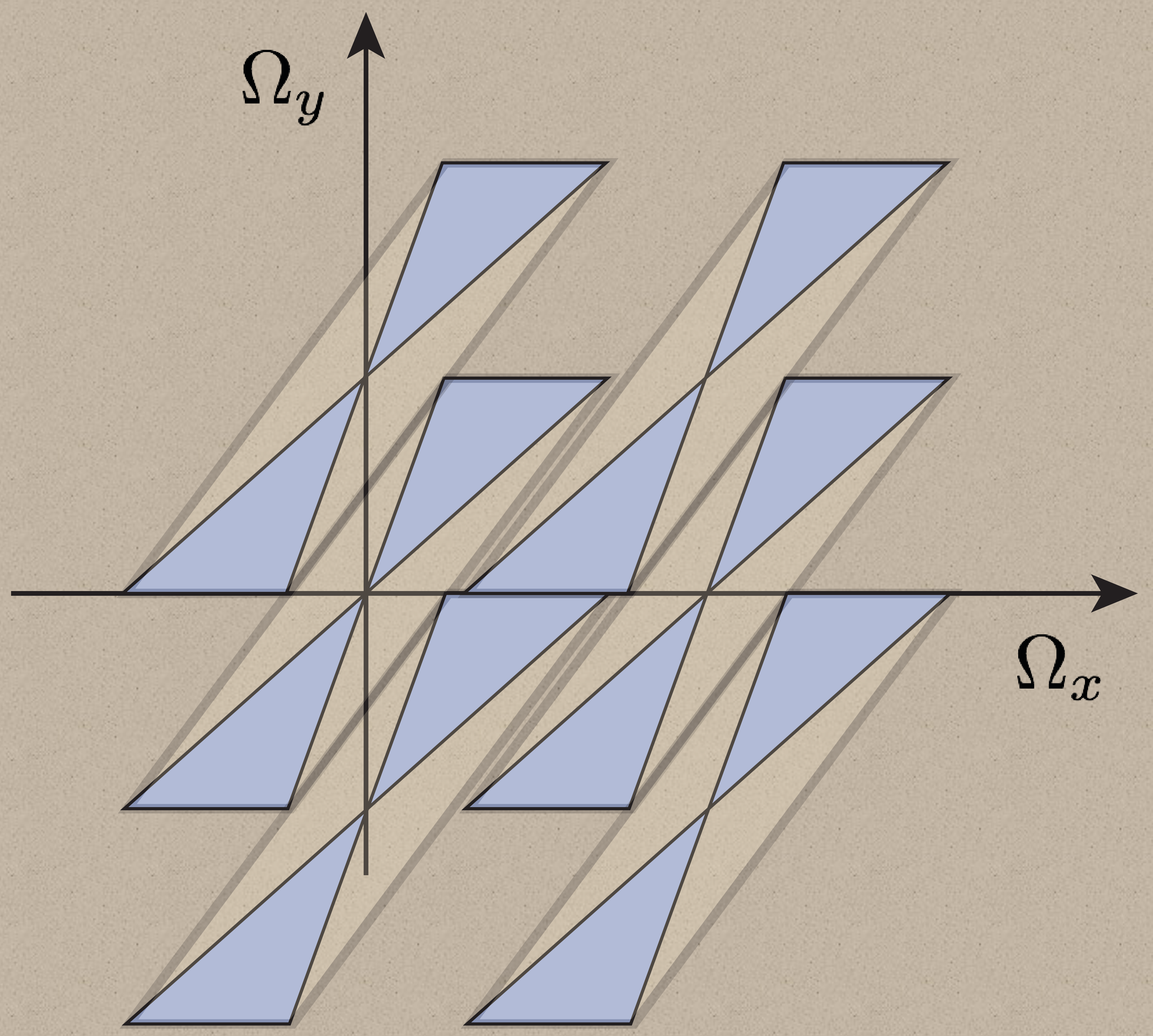


Primal domain
(flatland — 2D **sheared** filter)



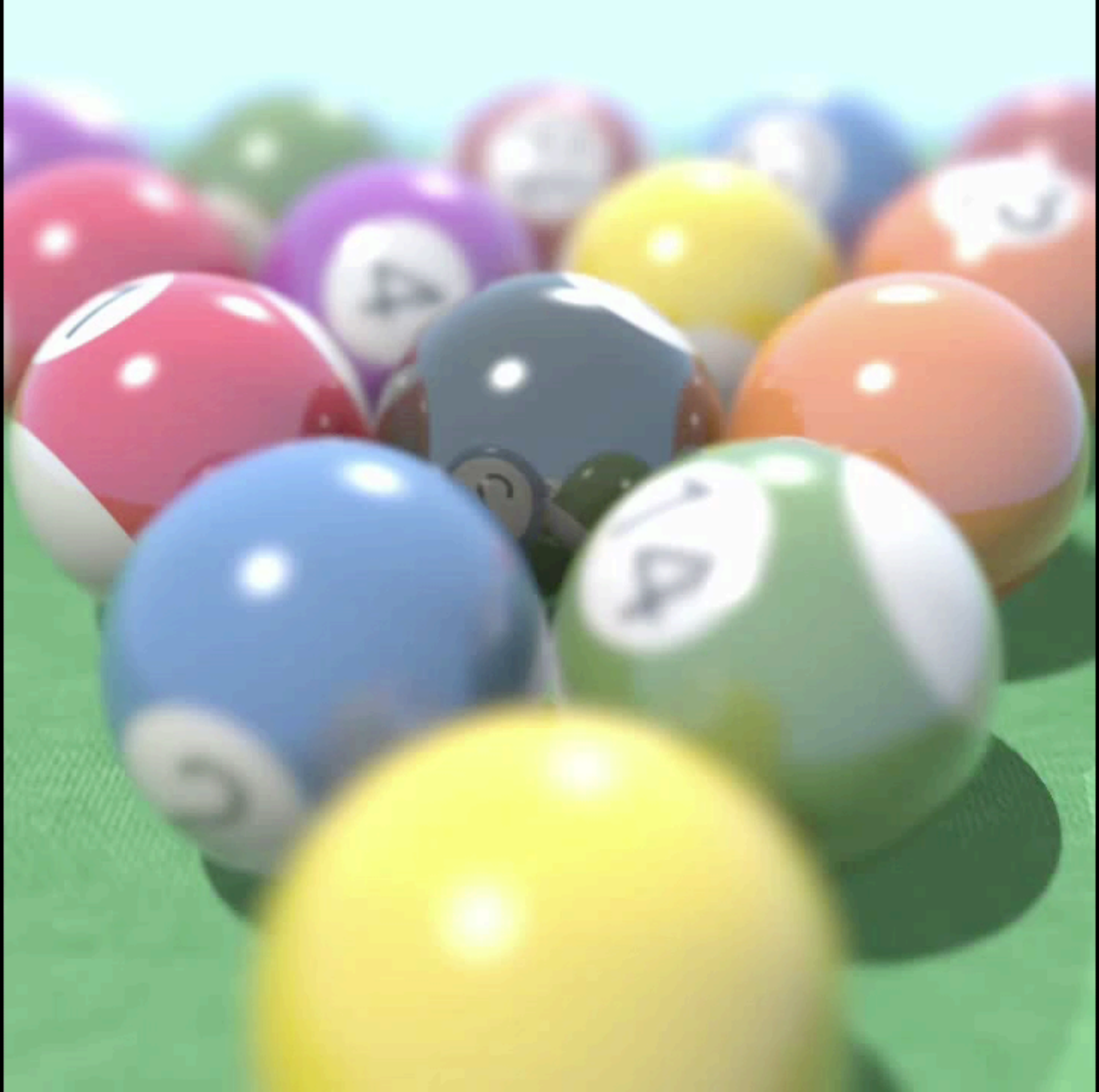
Primal domain
(scene — 4D **sheared** filter)

Sheared filtering



- Fourier: compactly bounded
  - Primal: few sampled required
- Fourier: sheared shape
  - Primal: hard to separate filter


17 FPS (59.03 ms), 1spp
Video Capture
H.264
Codec
60
Video FPS
Codec Options
Capture UI (?)
Use Time-Range
Time Range
0.000
Start Time
23.000
End Time
Start Recording
Cancel
Global Controls
0.000
Time
1.000
Time Scale
Reset
Play
Stop
Screen Capture
Video Capture
Load scene
Save scene
Load model
Load filter stack
Save filter stack
Renderer
Path Tracer
Renderer
1
Subsample
Increase Indirect Roughness
0.050
MIS thresho
2
Bounces
Rasterize primary rays
1
Spp
Pixel subsampling
Direct lighting
16
Sample loo
Visualize buffers
Accumulate samples
2.120
Exposure
Pixel peep
FilterStack
Enable
Reset
[0] Remove Outliers
Remove Outliers
Filter
Cut
Debug Output
Enable
Clamping
Method
Replace outlier with Median
7
Filter Si
FilterStack/Filter_0Average
FilterStack/Filter_0Clamping
[1] DAF



Future Research Directions

- Detailed rendering from **volumetric** microstructures
  - Glints from snow
  - Visible dust in 'god ray'

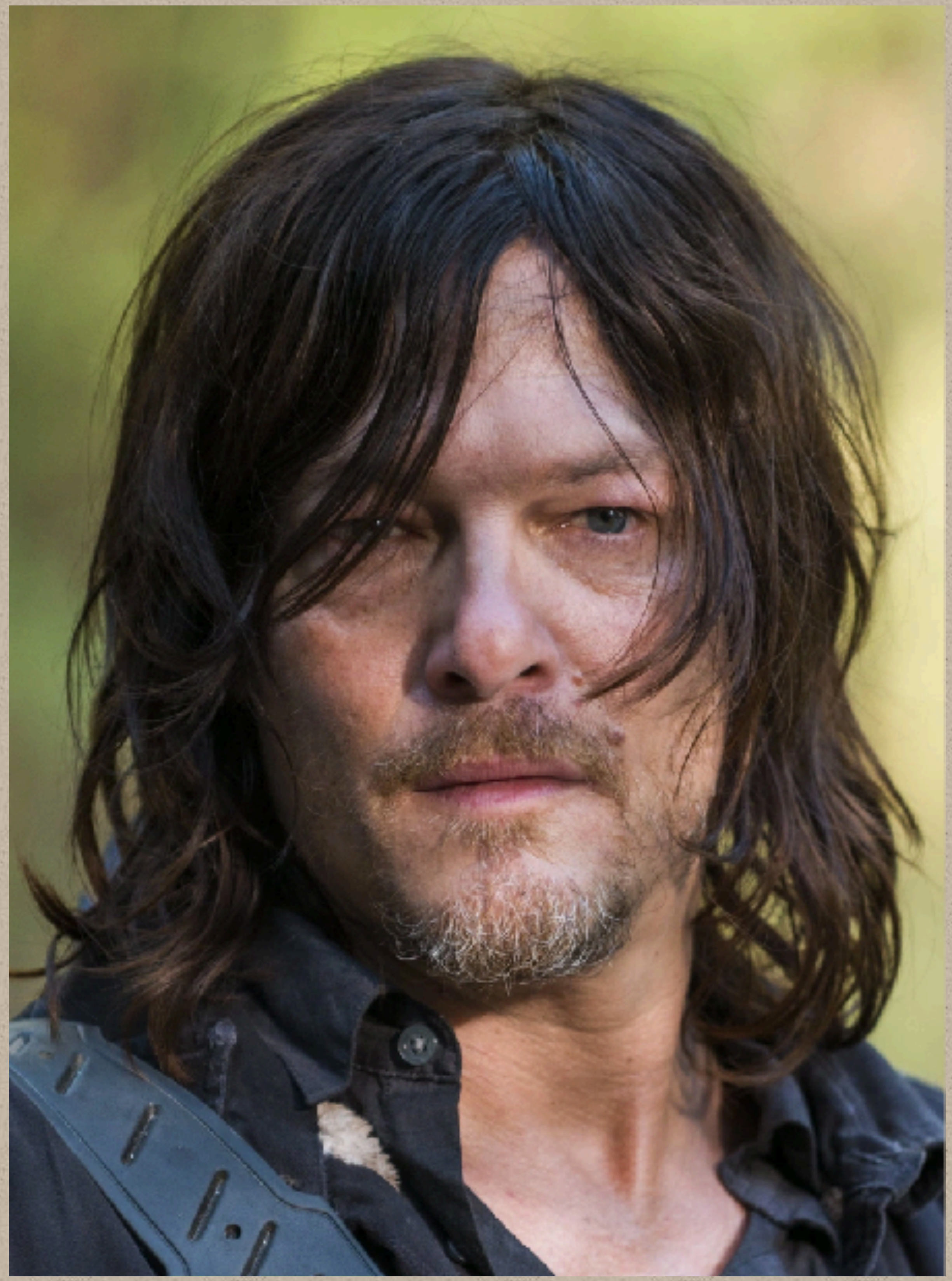
Future Research Directions

- Difficult visual effects
  - Especially multiple bounces of light
  - e.g. frosted glass and elevator inside

Future Research Directions

- More complex appearance modeling
  - Ultimate goal: **indistinguishable** from real photos

Norman Reedus in *Death Stranding*
(PS4 game, CG)



Norman Reedus
in *The Walking Dead*
(TV series, real photo)

Future Research Directions

- More complex appearance modeling
  - Ultimate goal: **indistinguishable** from real photos

photo

rendered

Future Research Directions

- Further speed up
  - Ultimate goal: **real-time**

Zafari, an animation rendered completely using Unreal game engine

Future Research Directions

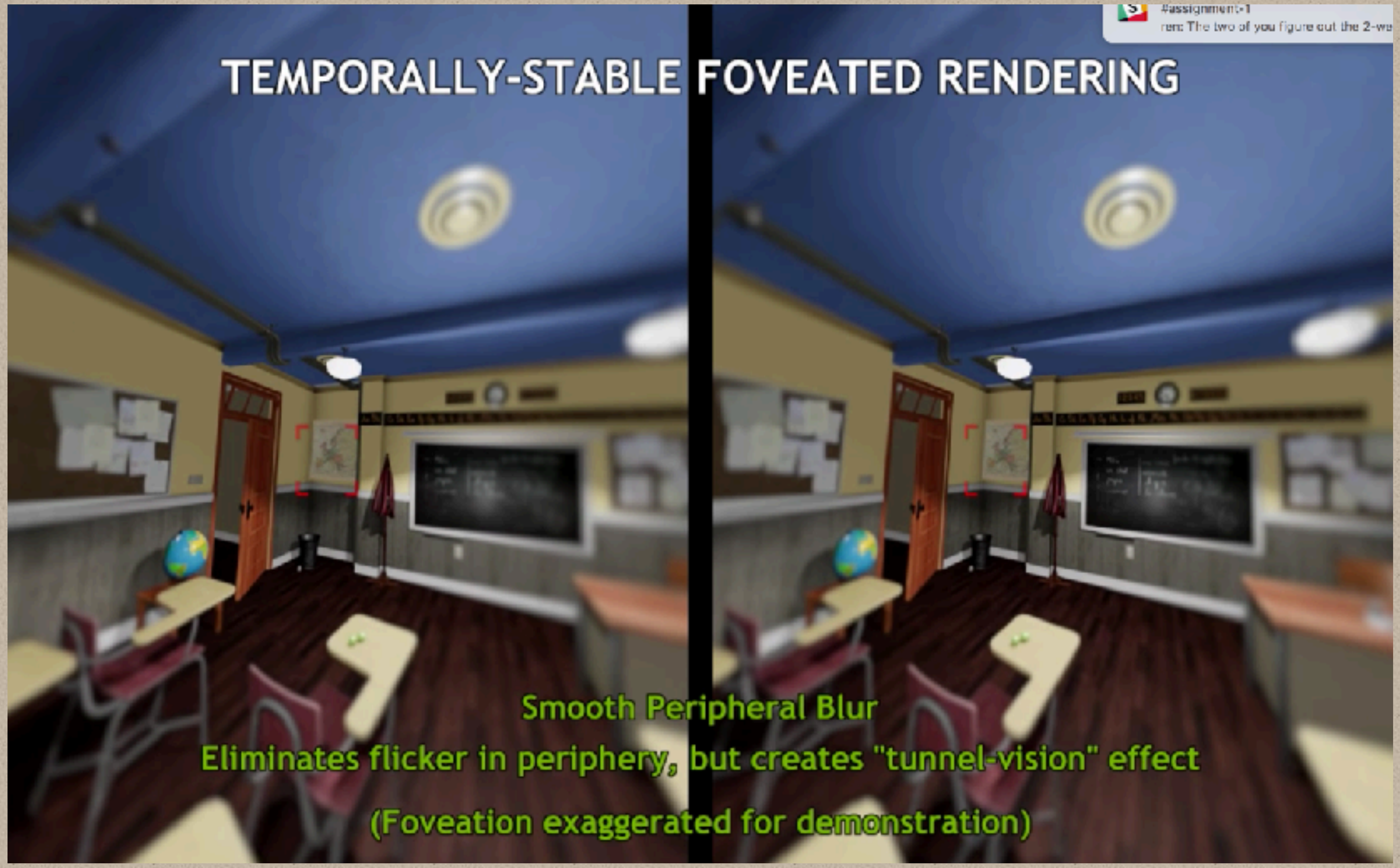
- Exploiting Machine Learning for Computer Graphics
  - Differentiable rendering for Computer Vision

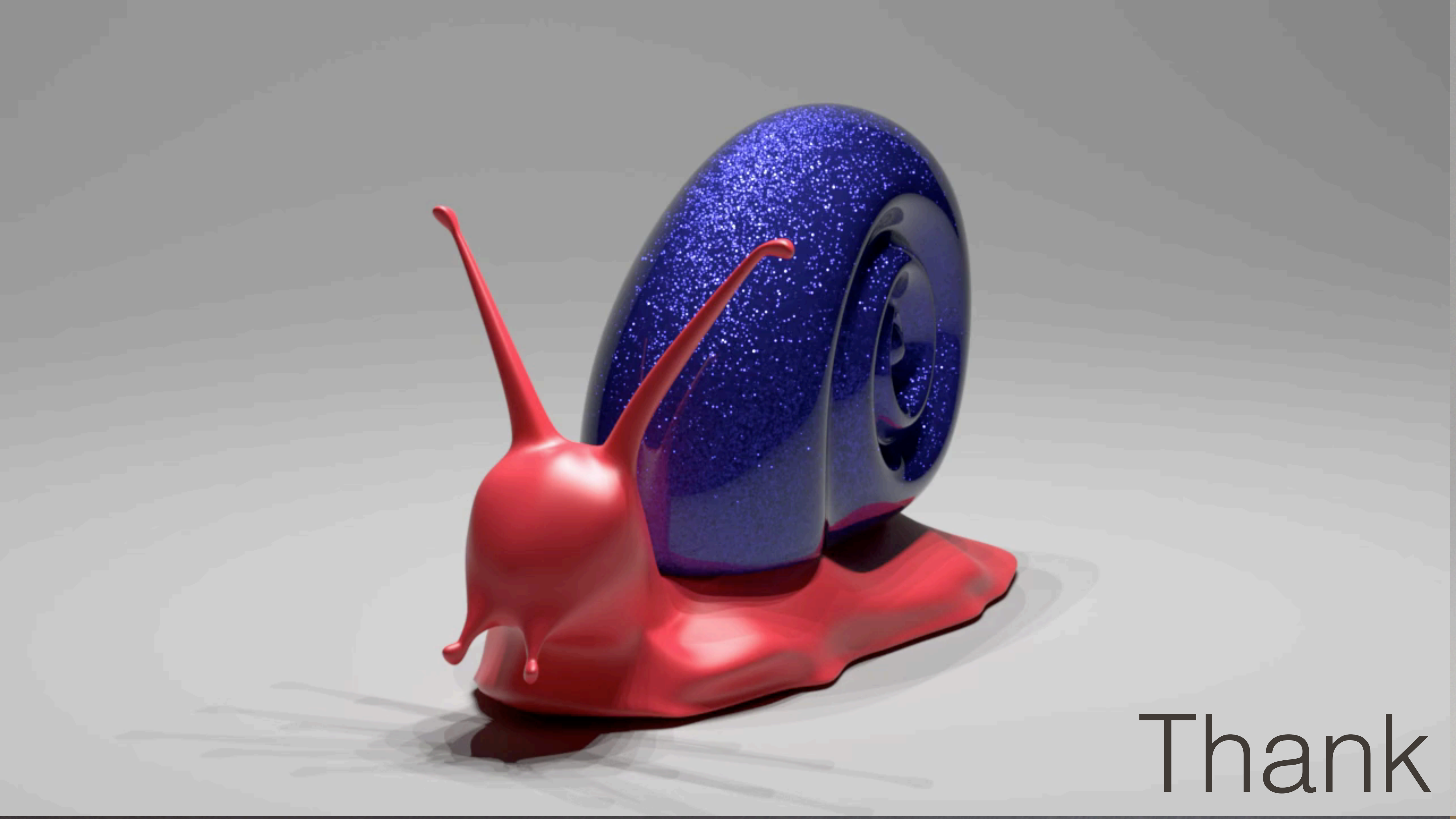
Future Research Directions

- Exploiting Machine Learning for Computer Graphics
  - Neural network as geometry representation

Future Research Directions

- Virtual Reality (VR) / Augmented Reality (AR)
  - VR = equipment + rendering
  - AR = Computer Vision + VR





Thank you!