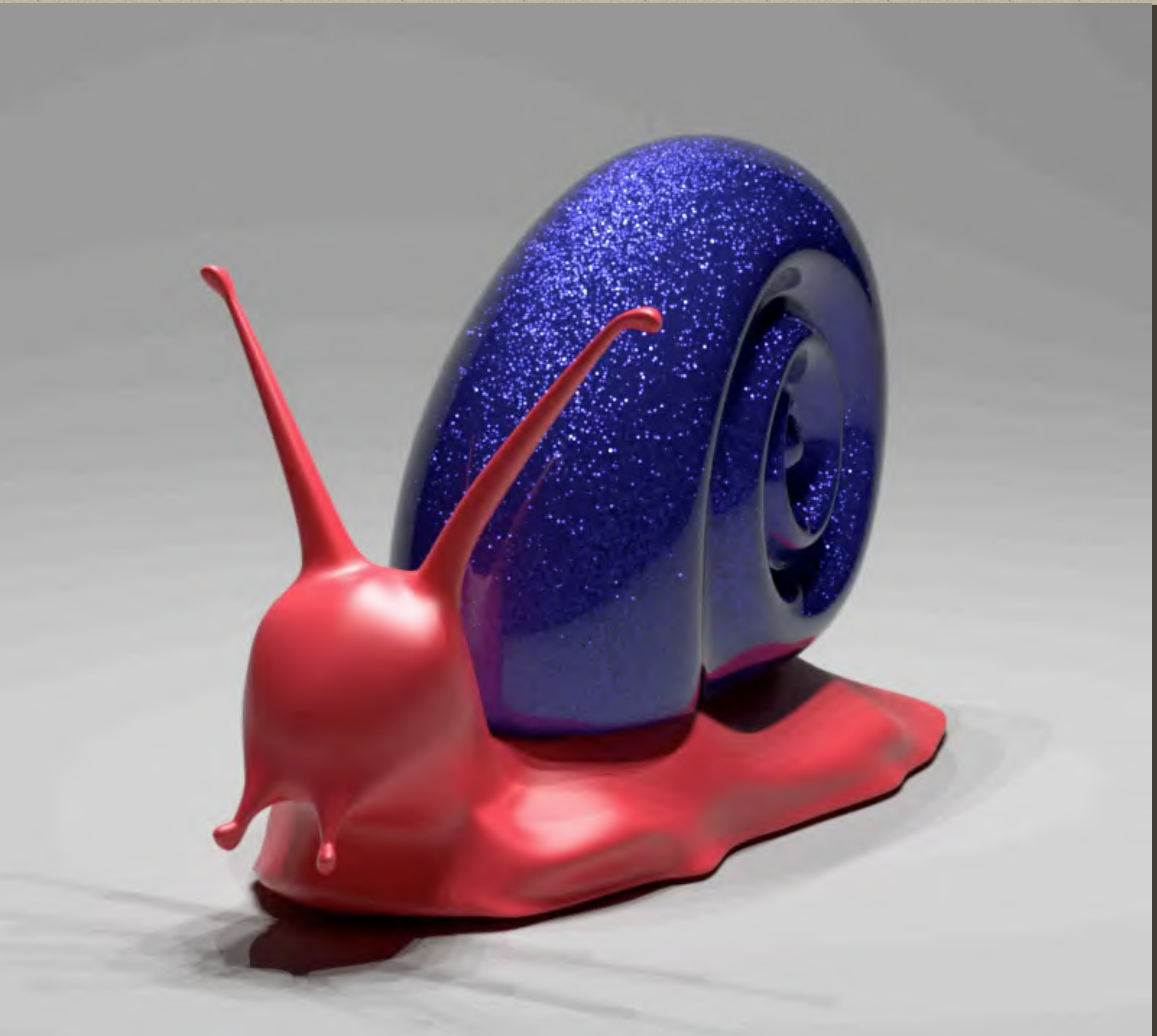
Lingqi's

GRAPHICS TALK

AT PEKING UNIVERSITY



Next Generation Rendering: Photorealism and Speed

Lingqi Yan (闫令琪)

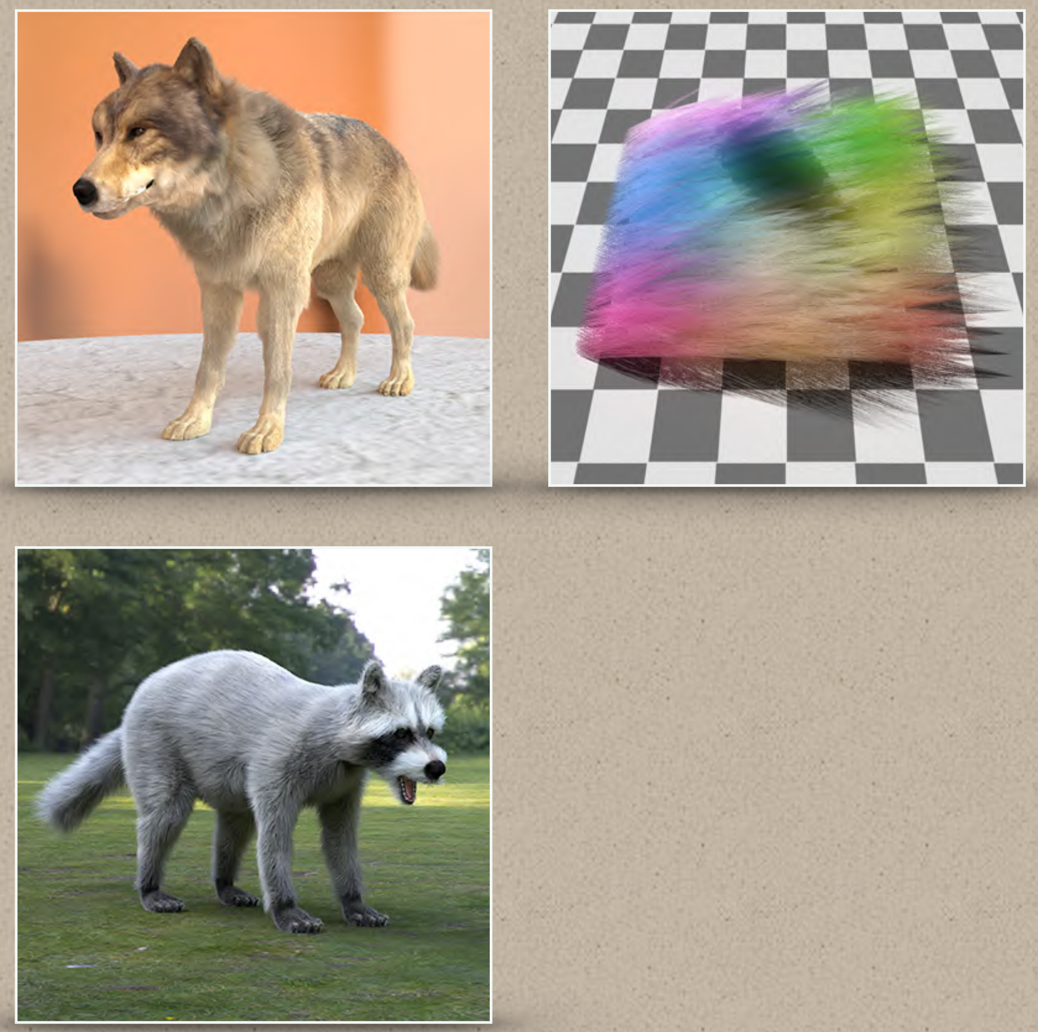
Assistant Professor

University of California, Santa Barbara

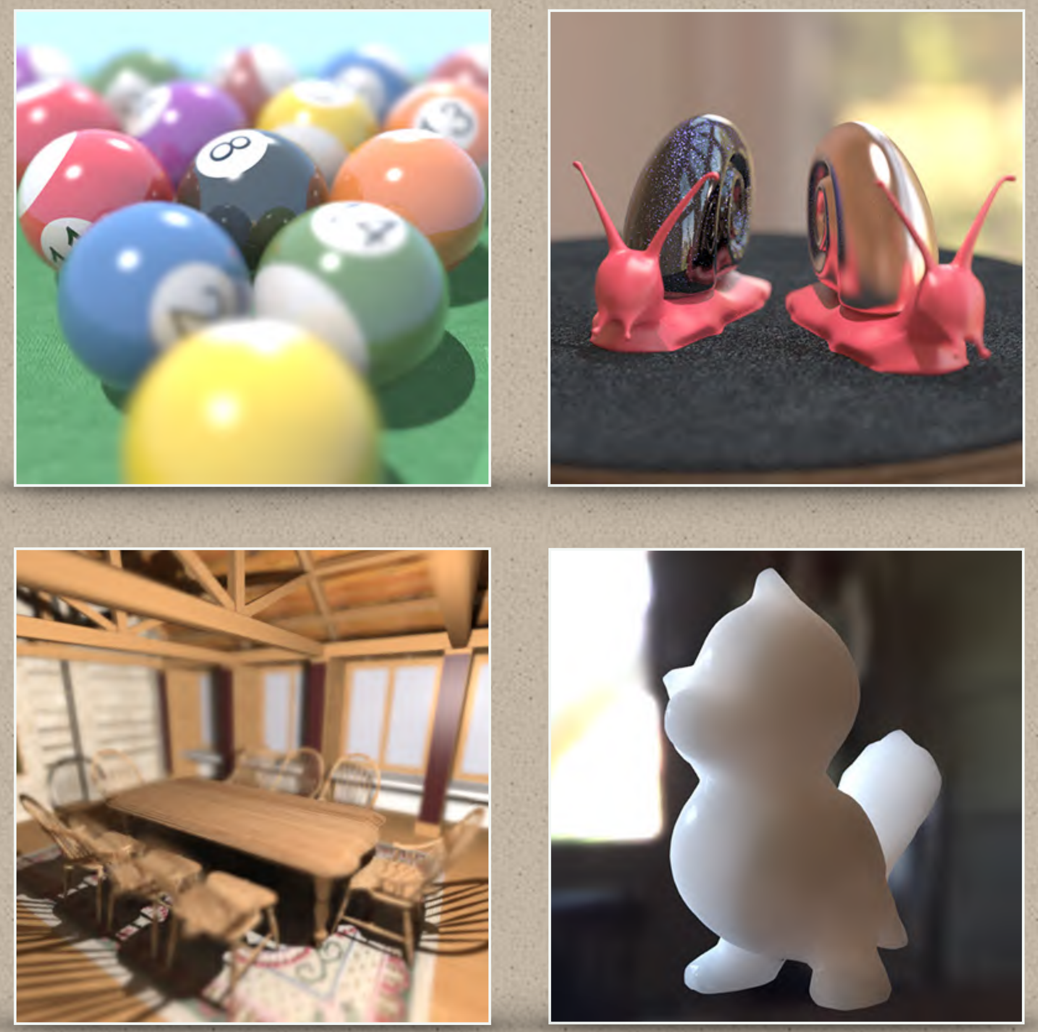
About Me

- Lingqi Yan
  - Tsinghua (B.E. 2013) — UC Berkeley (Ph.D. 2018) — UC Santa Barbara (AP)
  - Research: **Rendering**!

Detailed Rendering
[SIG 14, 14, 16, 18]

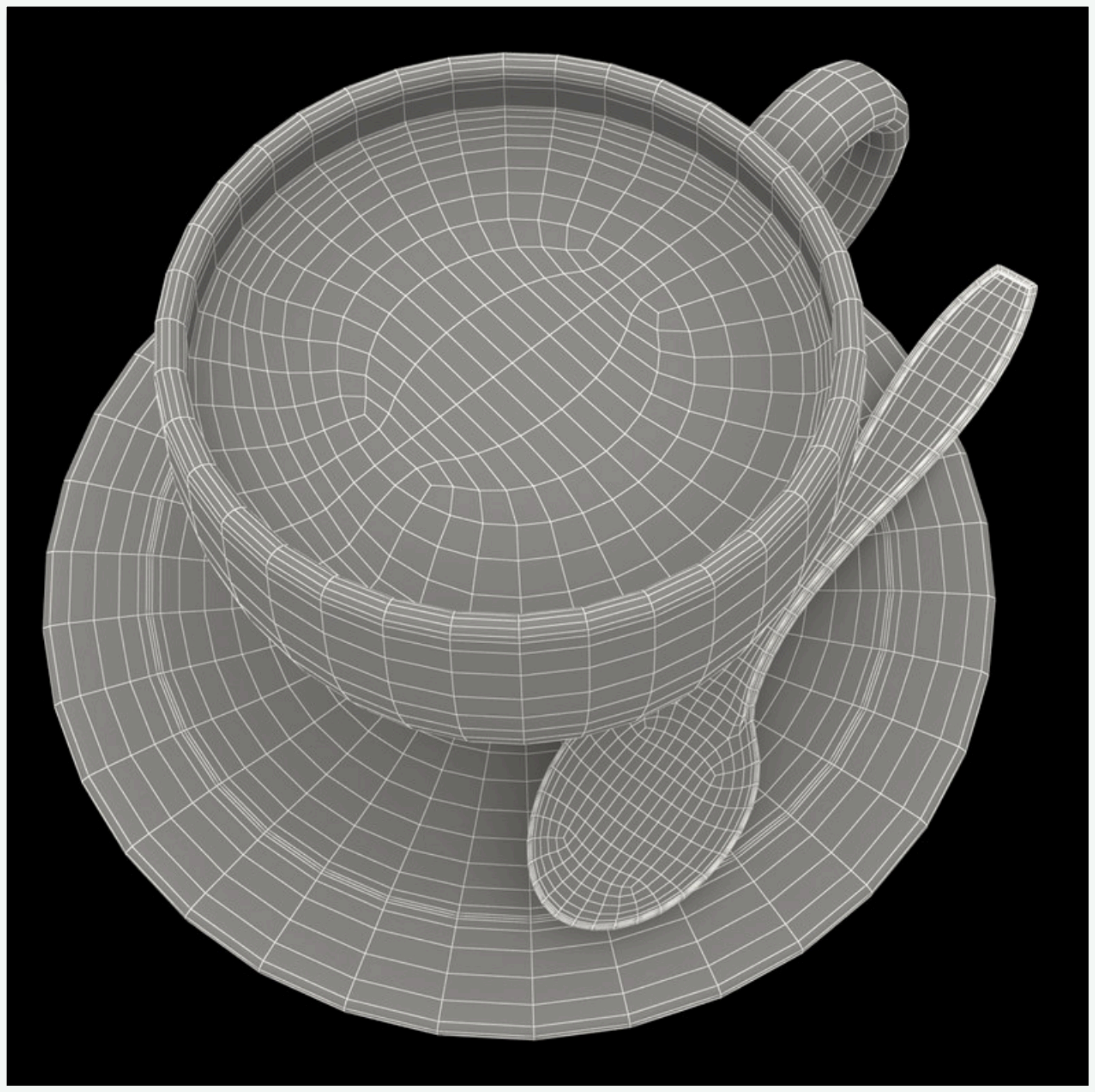


Appearance Modeling
[SIG 15, 17, SIG Asia 17]



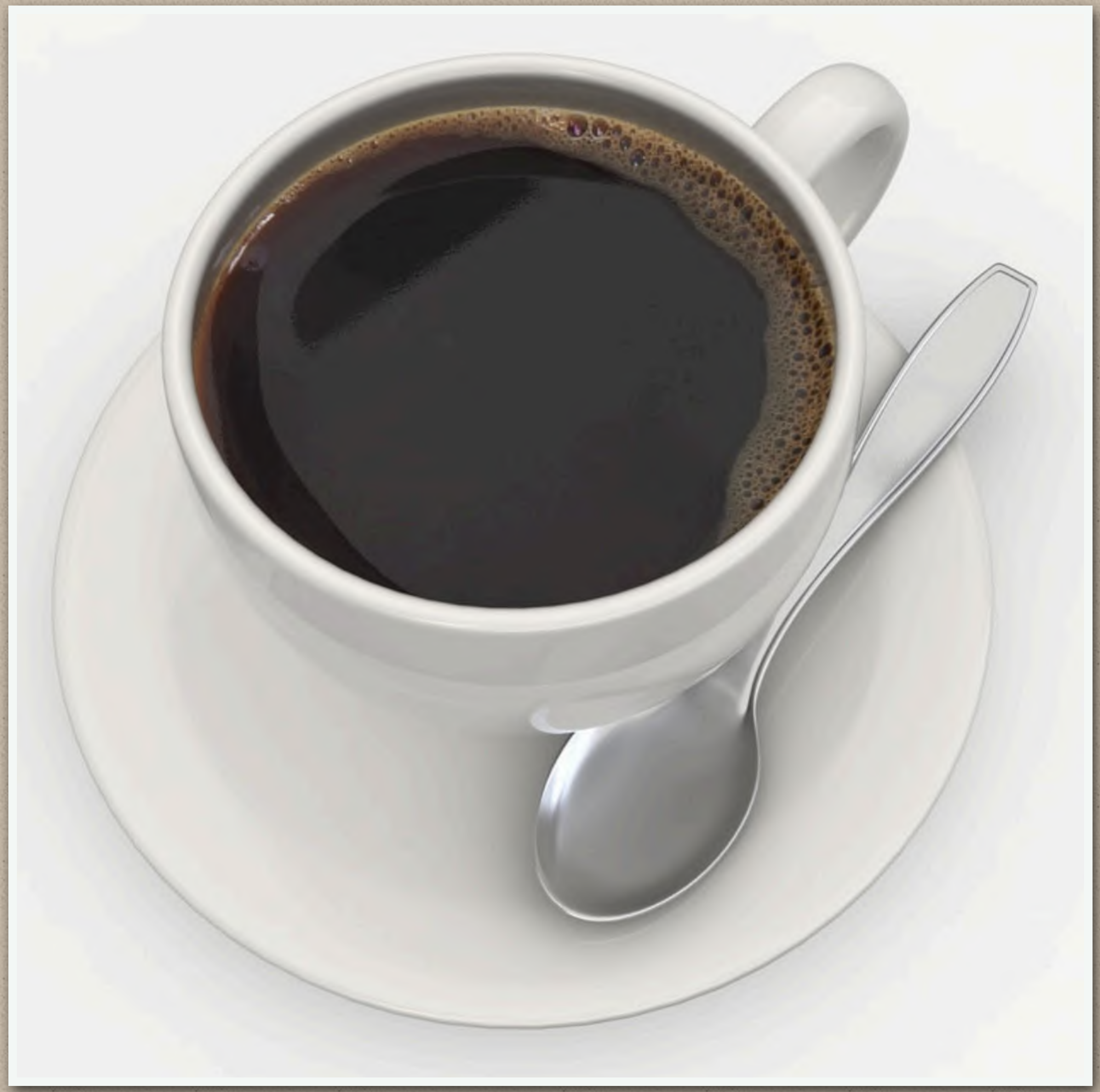
RT / off-line Light transport
[SIG 15, 16, EGSR 17, PG 12]

What is Rendering?



3D scene (meshes, lights, etc.)

Calculating light -> eye



Image

Rendering is everywhere

video games
(Super Mario
Odyssey)

product design
(Amazon
Echo Dot)

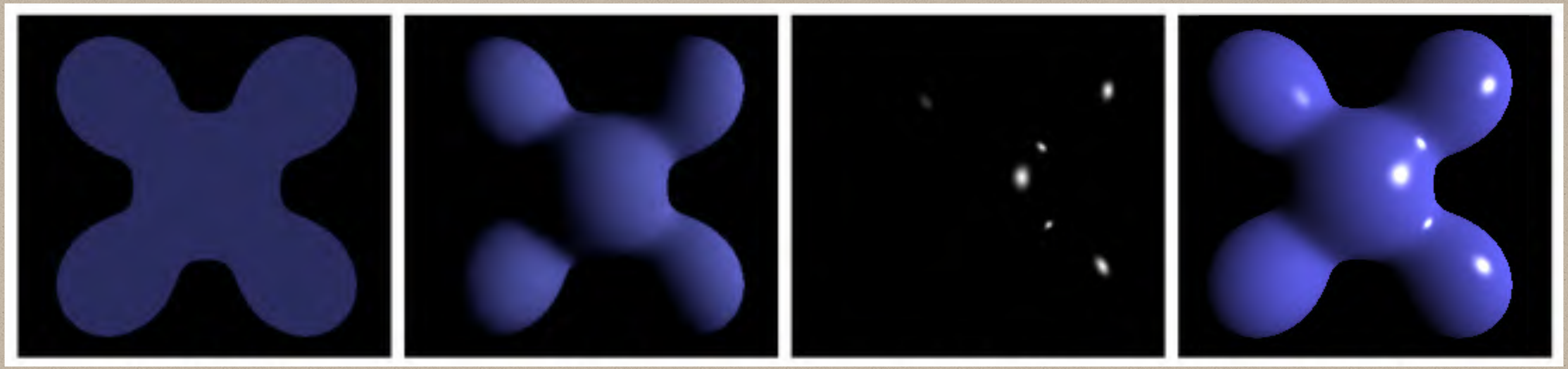
predictive rendering
(Avametric)

animations /
movies
(Zootopia)

Part 0: State of the Art Rendering

- ~~Light transport (will not be covered today)~~
  - vertex connection & merging, Metropolis, etc.
- Appearance modeling
  - Microfacet model, anisotropy
- Material capture
- Complex materials
- …

Simple Appearance Models

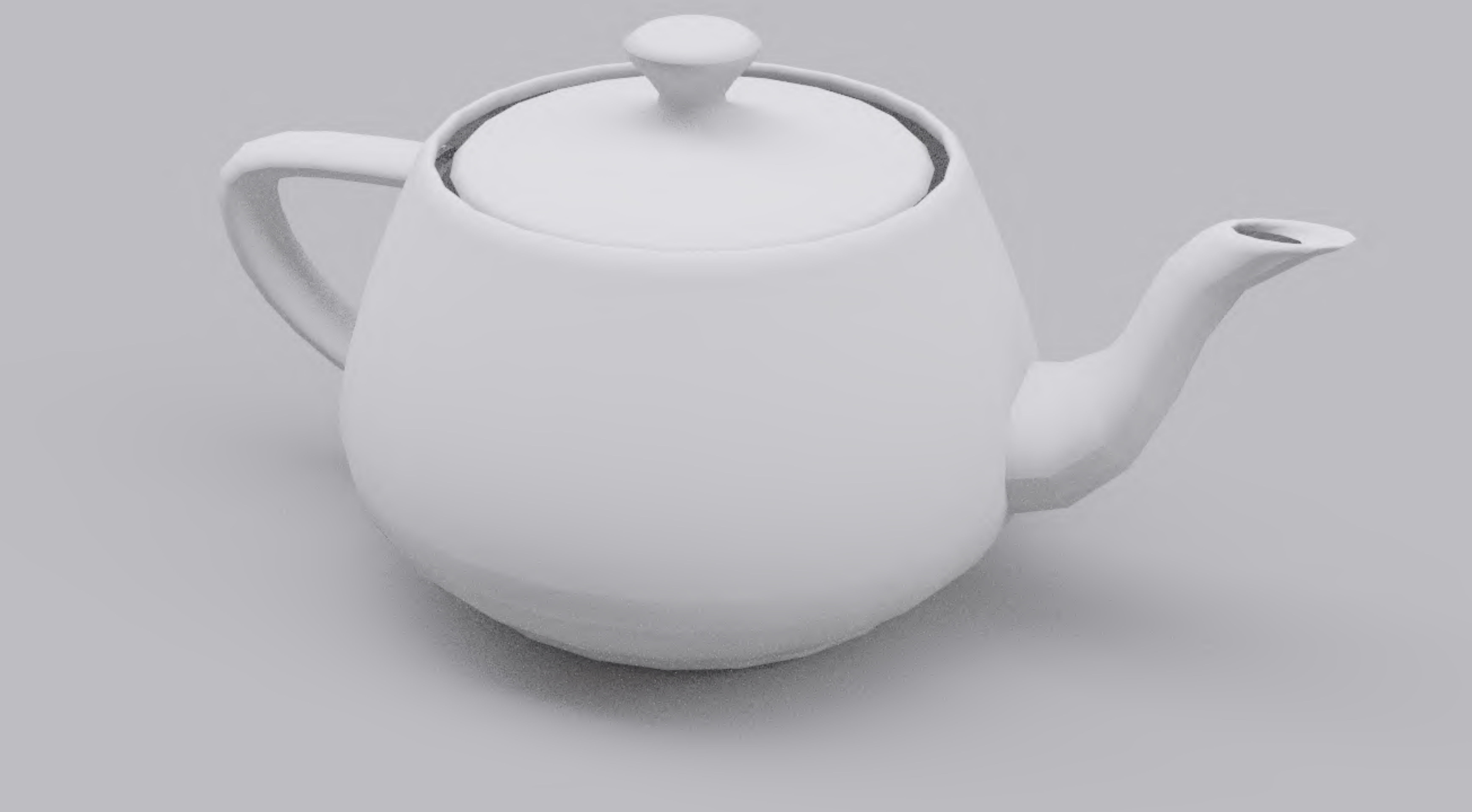


Ambient + Diffuse + Specular = **Blinn-Phong Model**

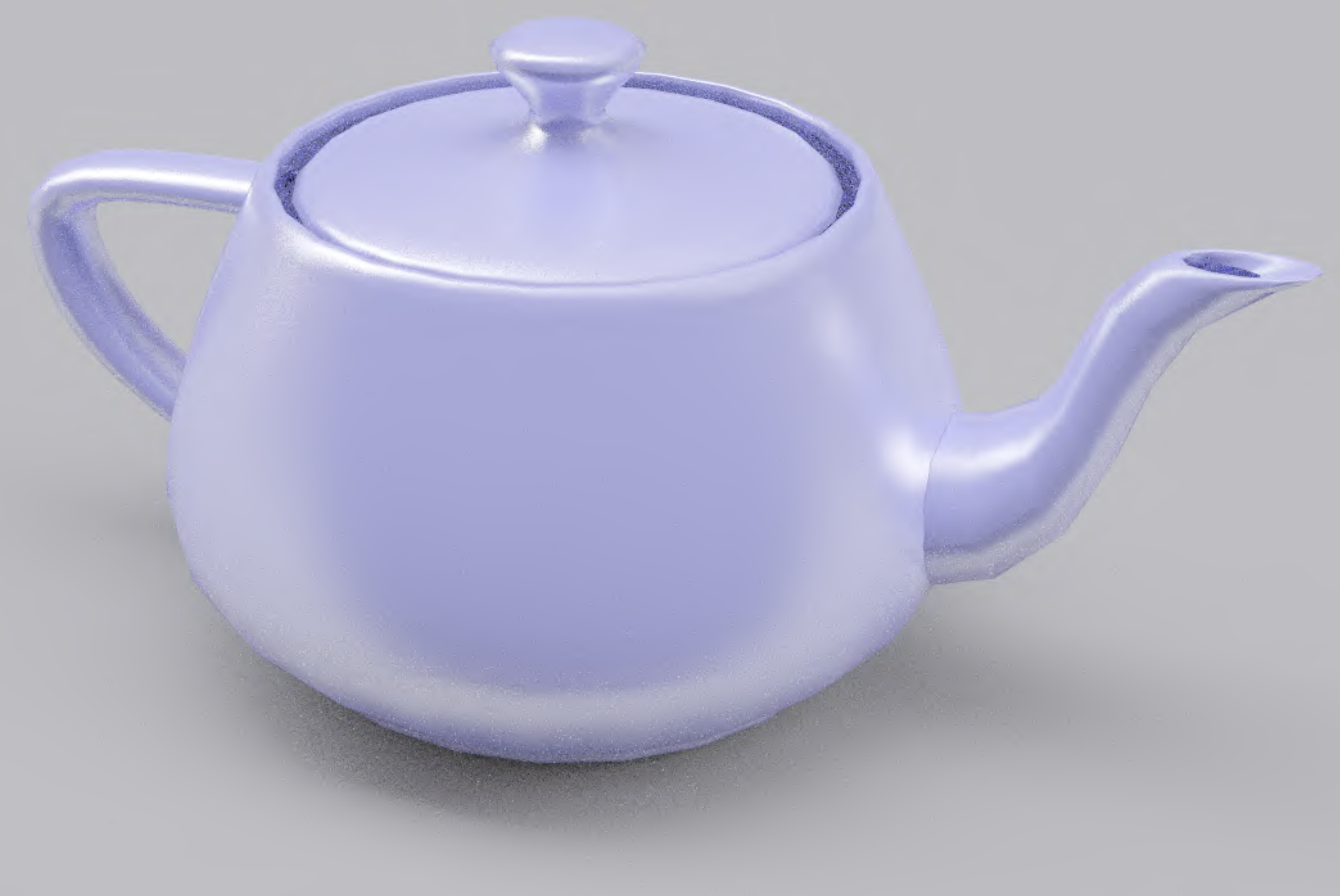
$$
\begin{aligned}
L &= L_a + L_d + L_s \\
&= k_a\, I_a + k_d\, (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s\, (I/r^2) \max(0, \mathbf{n} \cdot \mathbf{h})^p
\end{aligned}
$$

More Types of Appearance

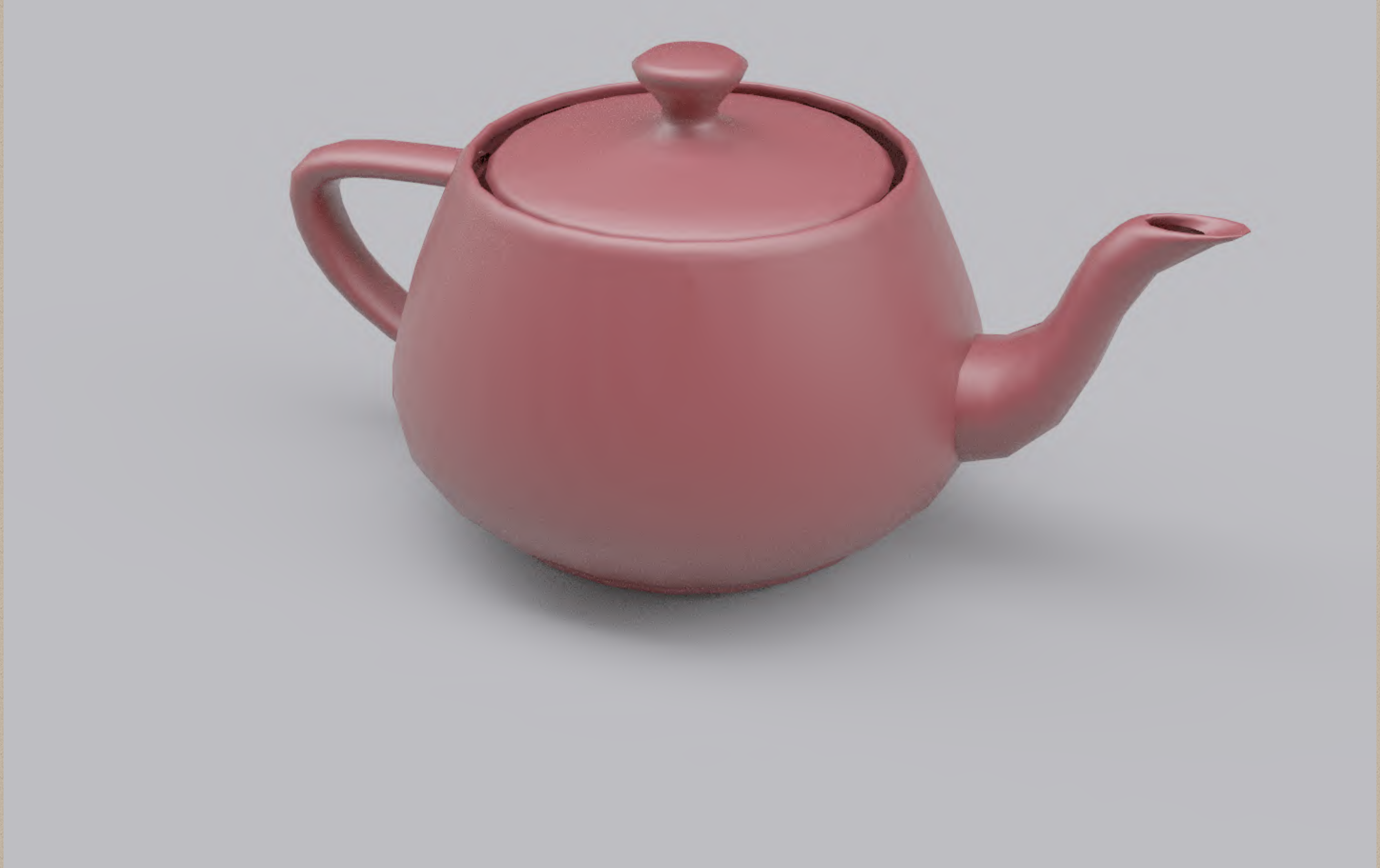
Materials: Diffuse



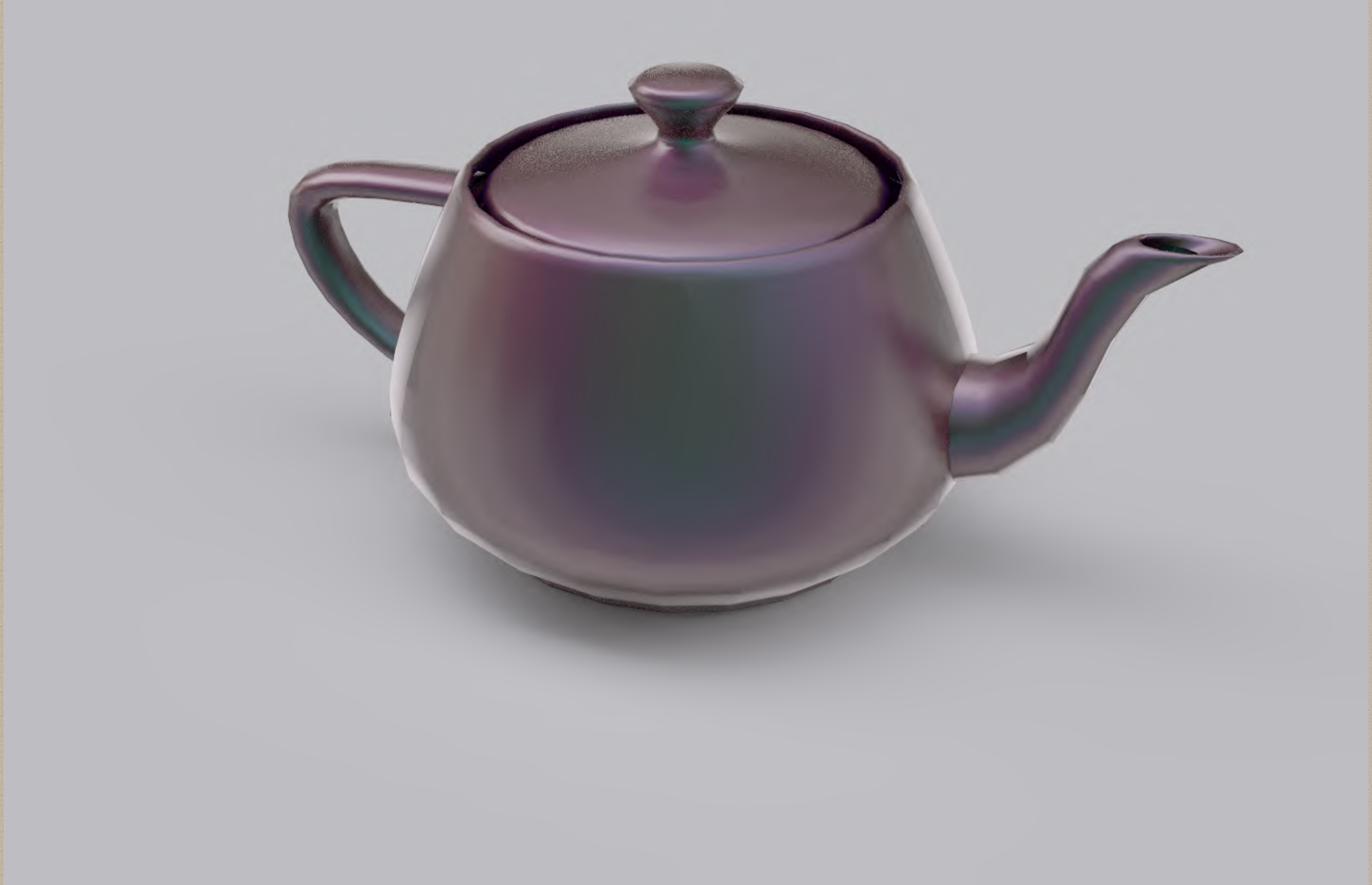
Materials: Plastic



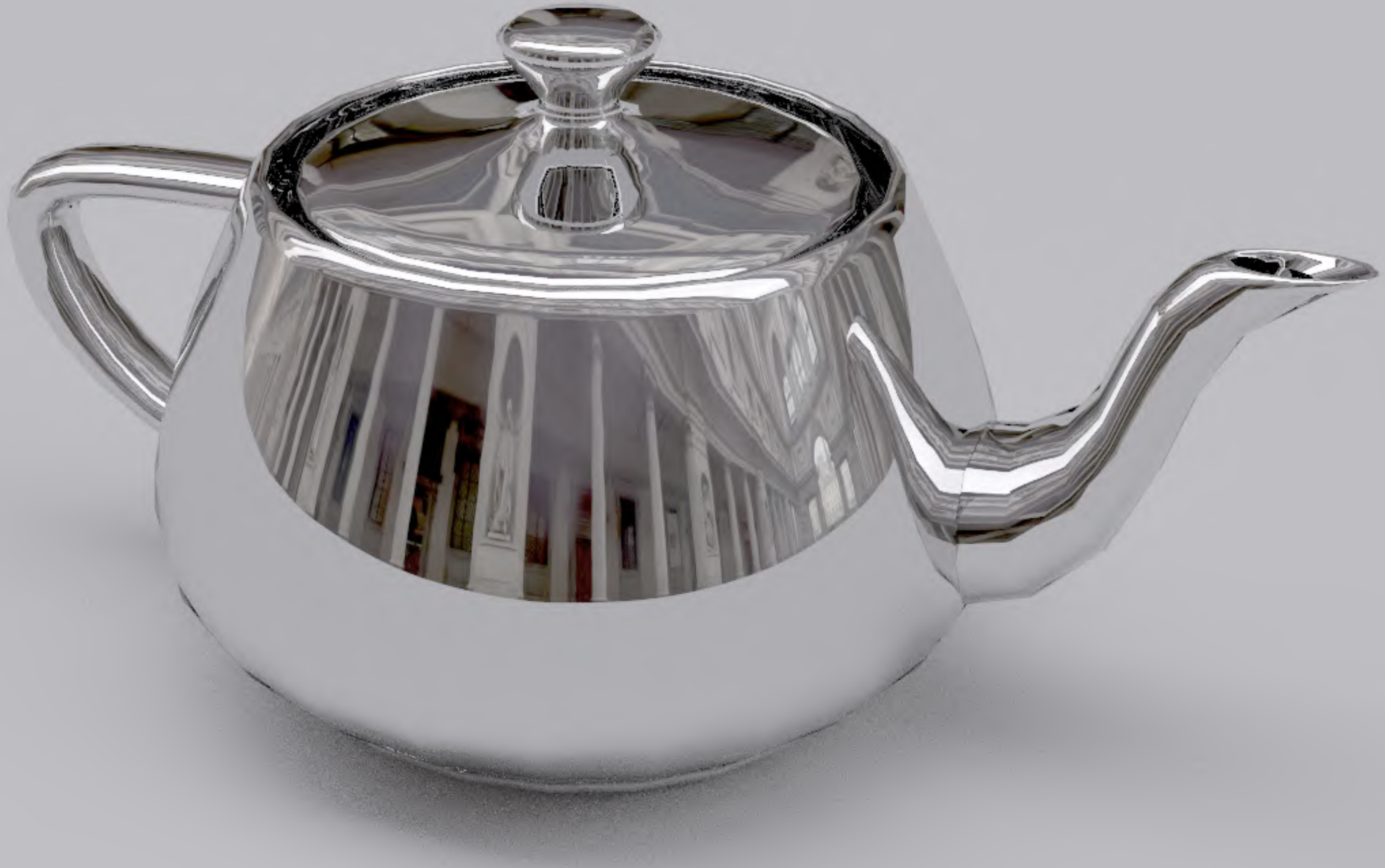
Materials: Red Semi-Gloss Paint



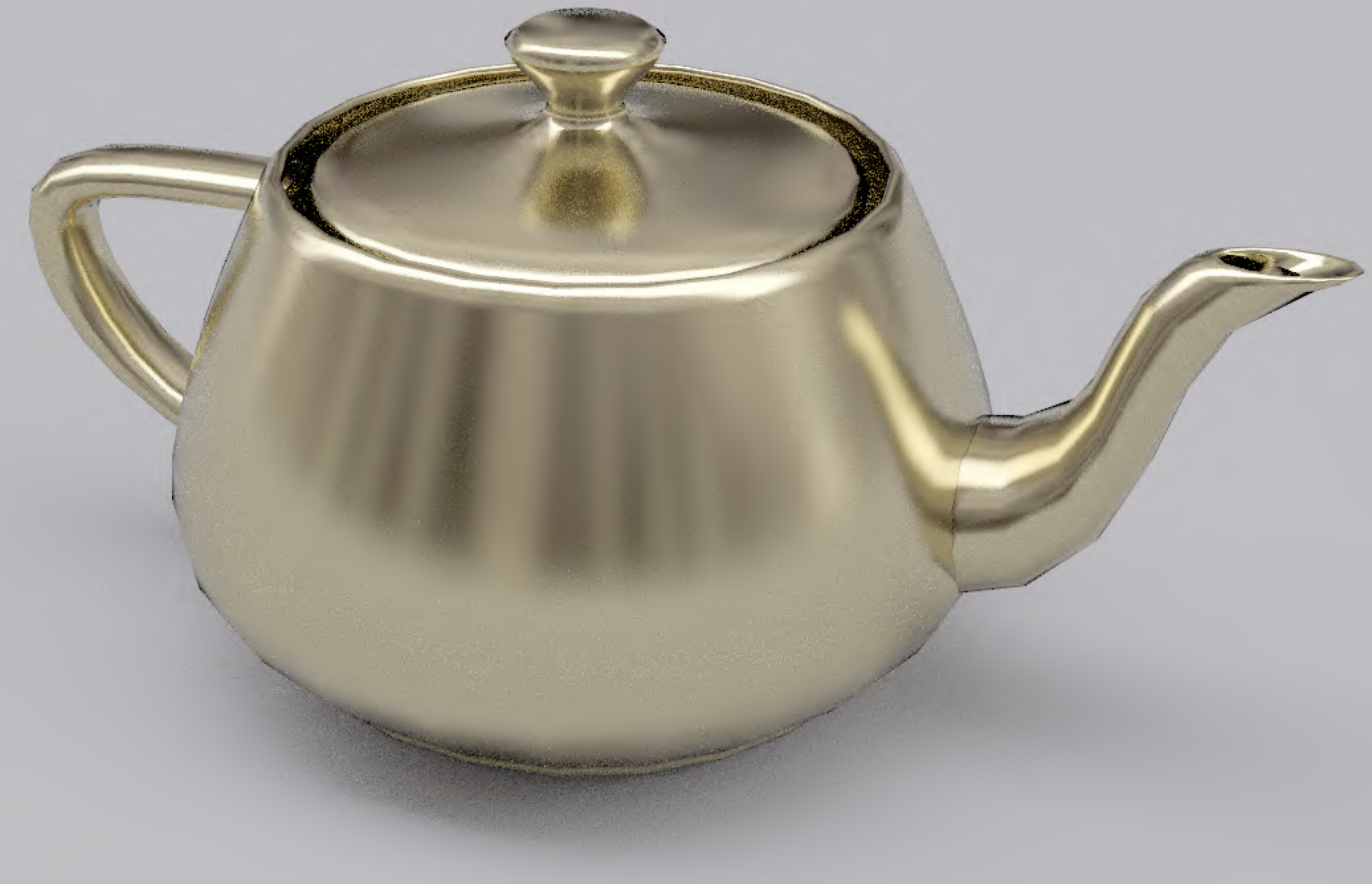
Materials: Ford Mystic Lacquer Paint



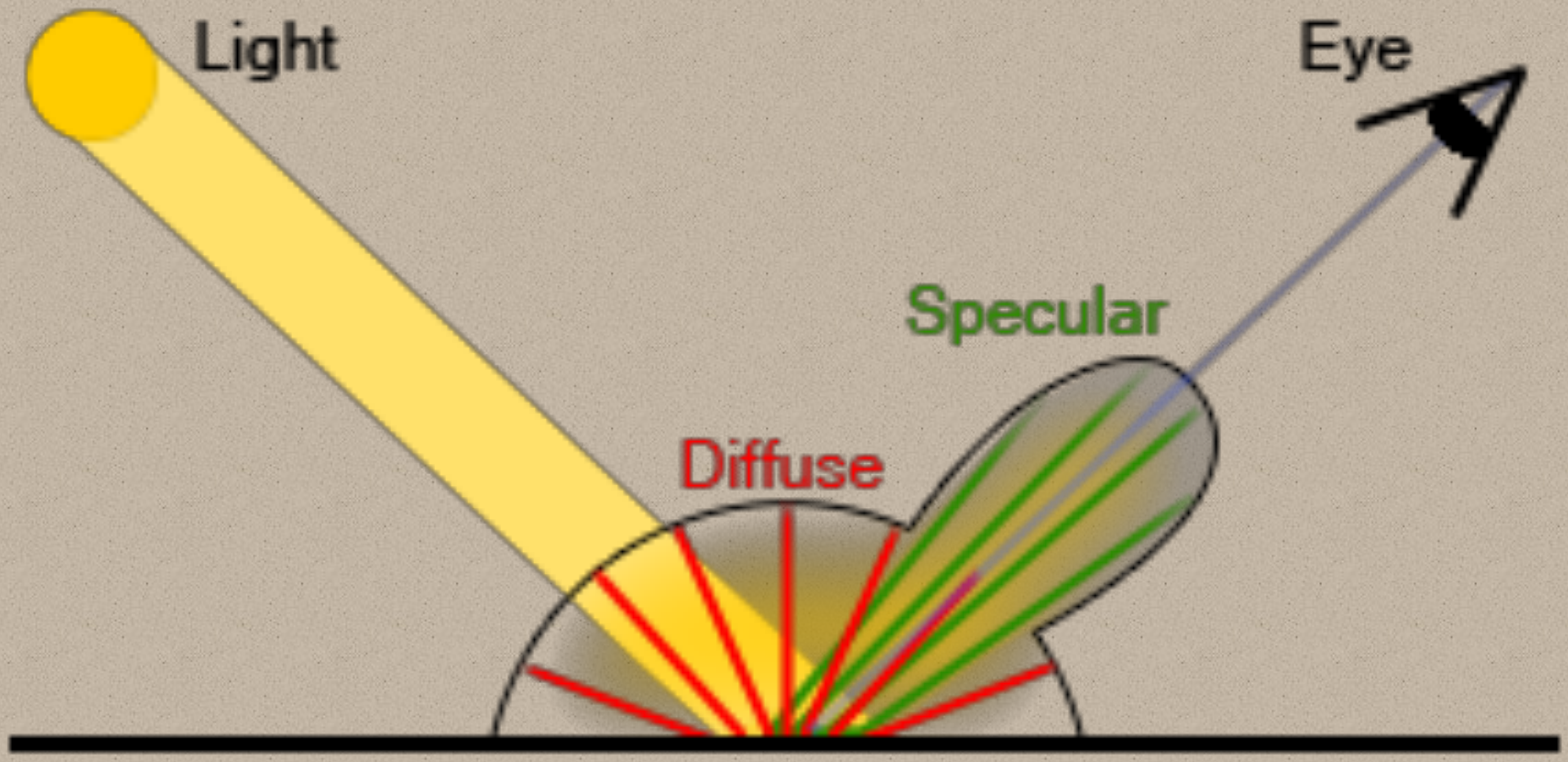
Materials: Mirror



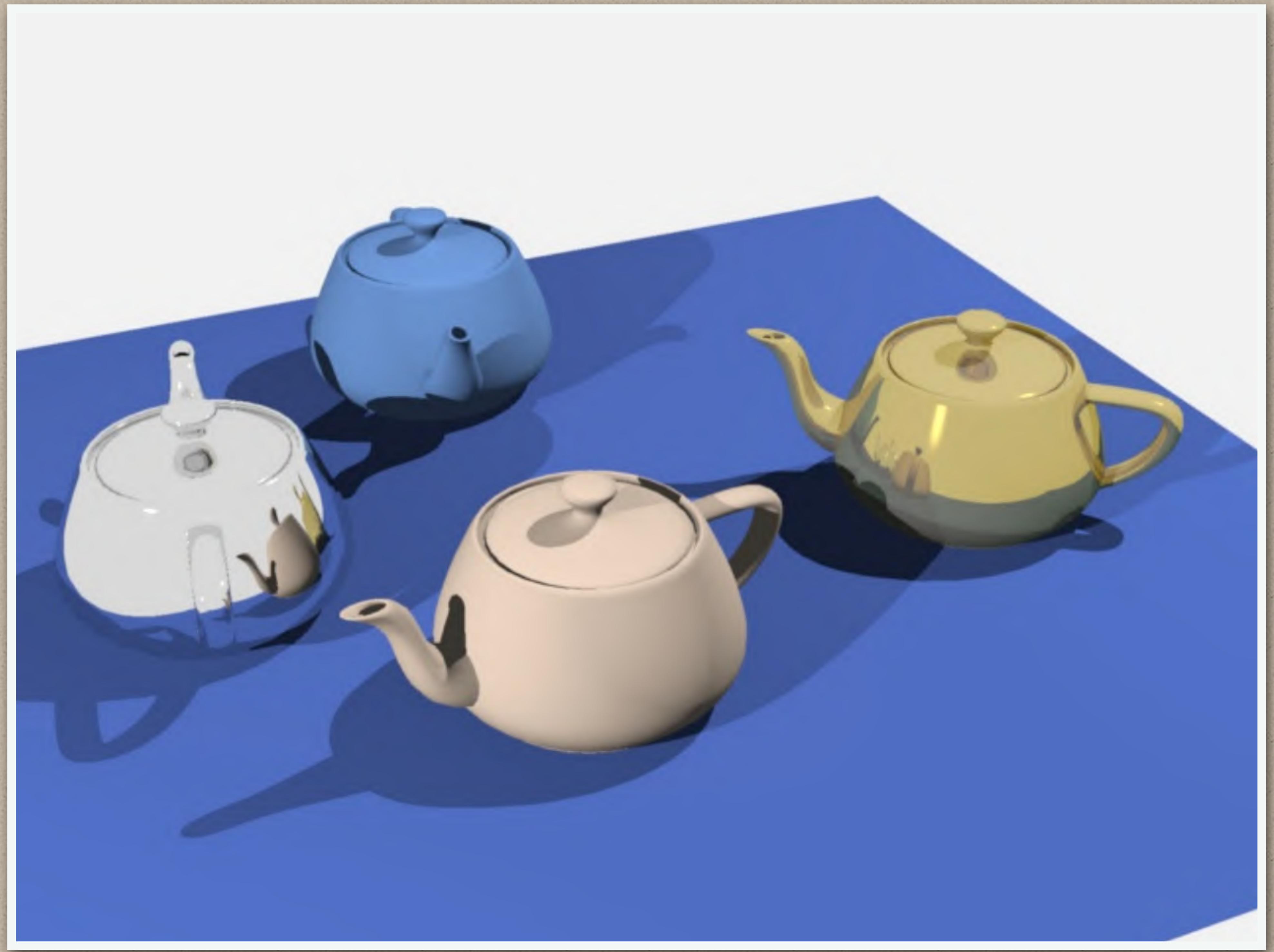
Materials: Gold



Describing Reflections: BRDF



Traditional BRDF (Phong model)



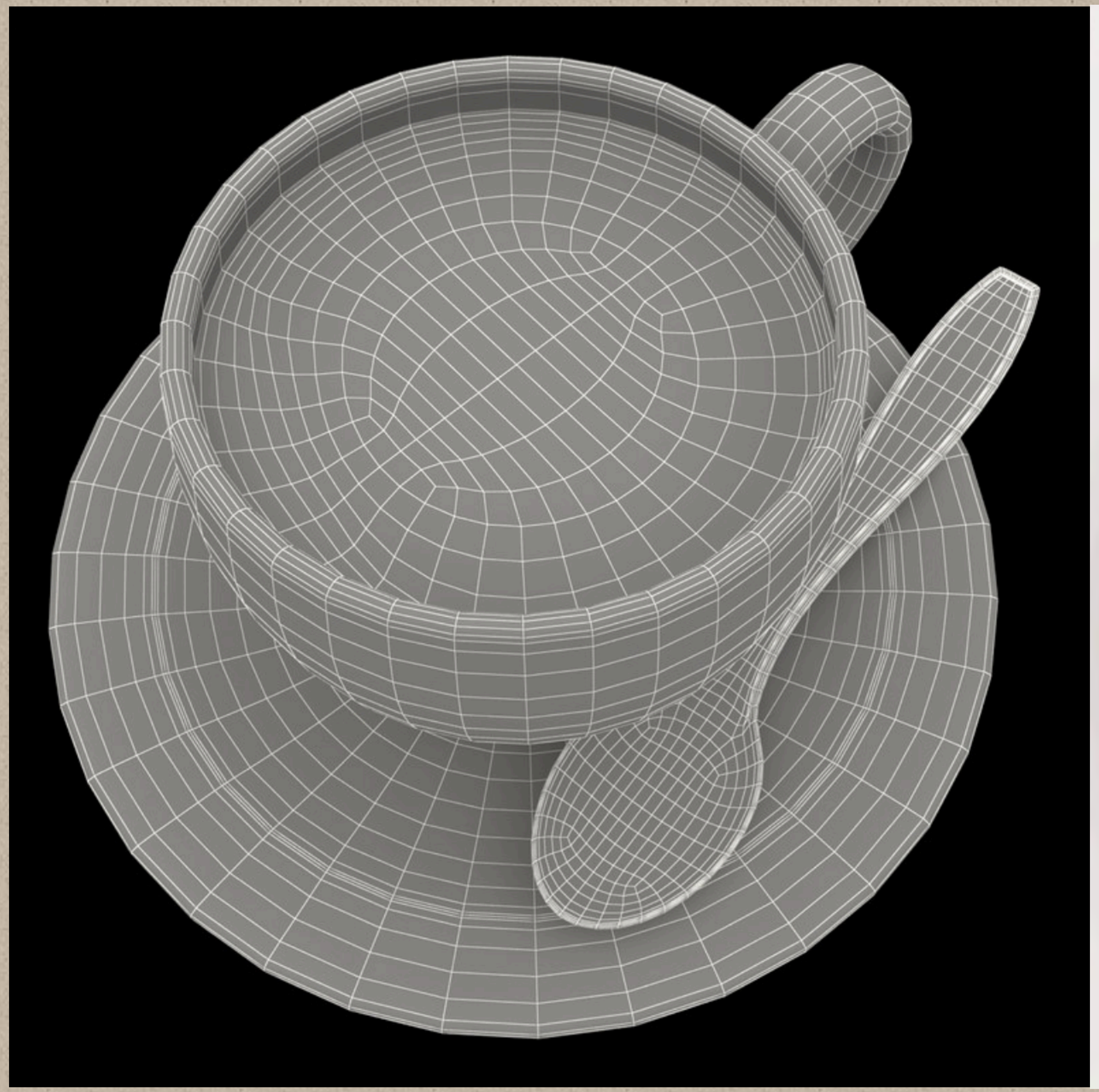
unrealistic, artificial results

*: Bidirectional Reflectance Distribution Function

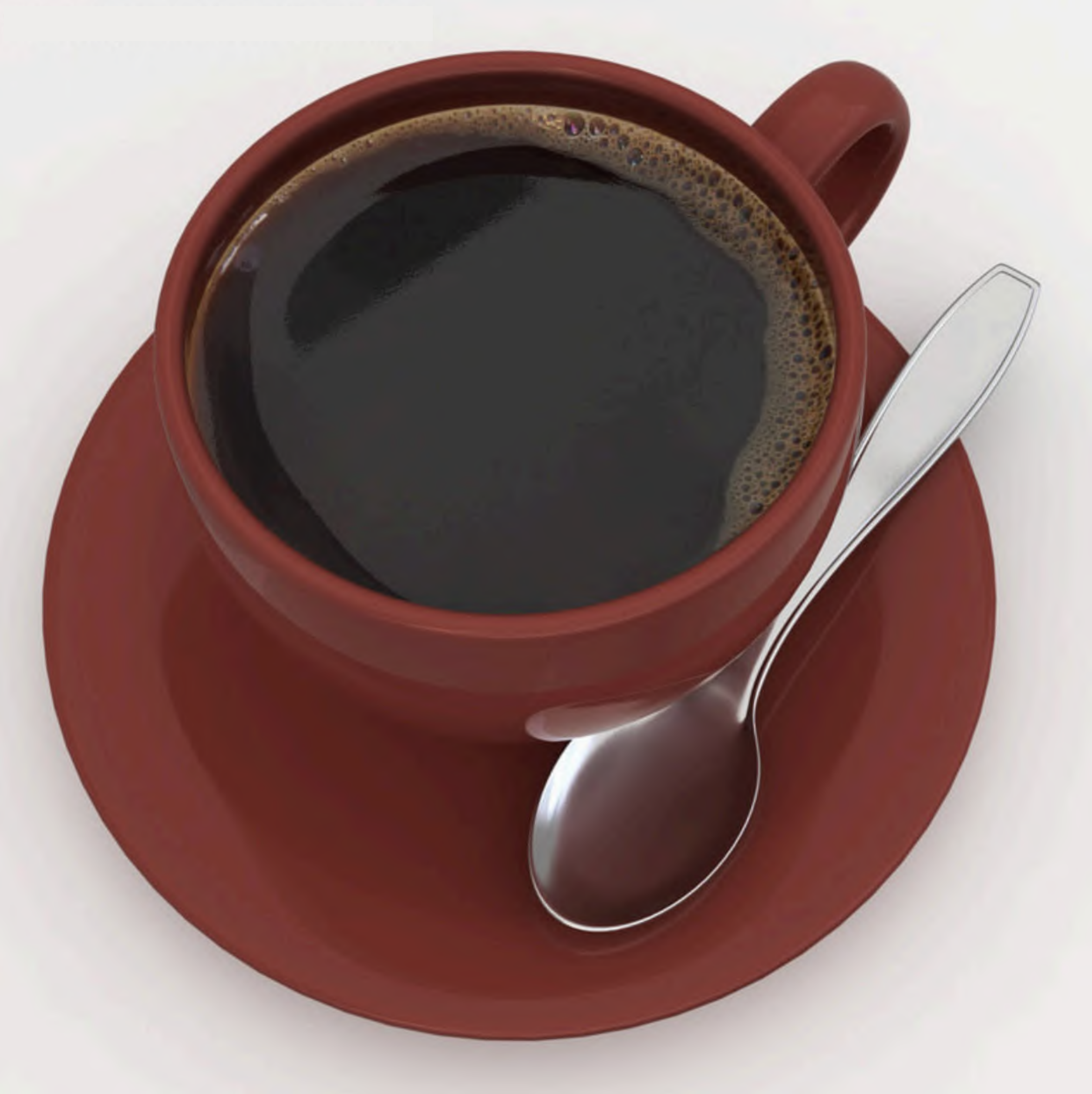
The Appearance of Natural Materials

[Courtesy of Prof. Henrik Wann Jensen, UCSD]

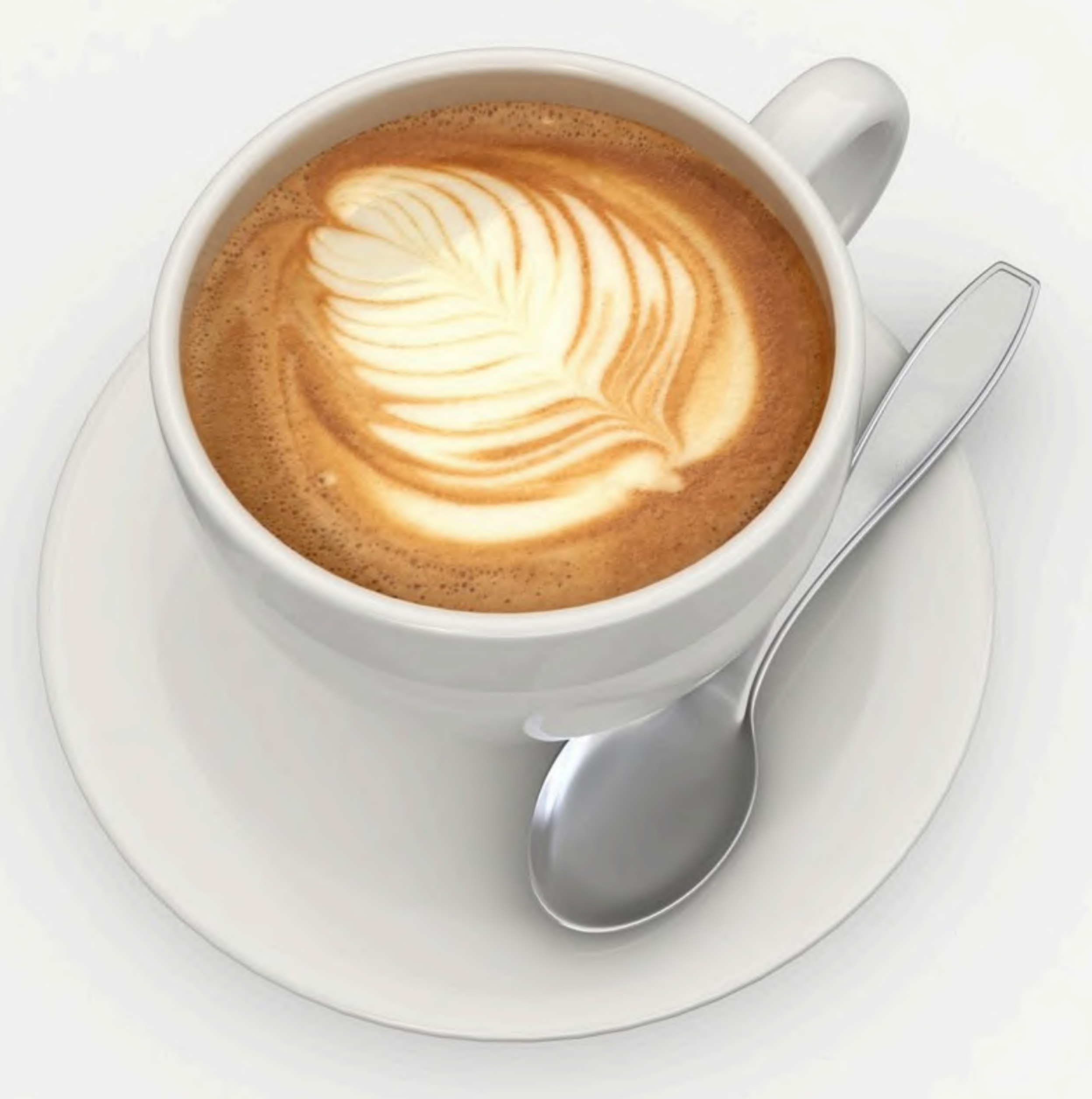
What is Material in Computer Graphics?



3D coffee mug model



Rendered

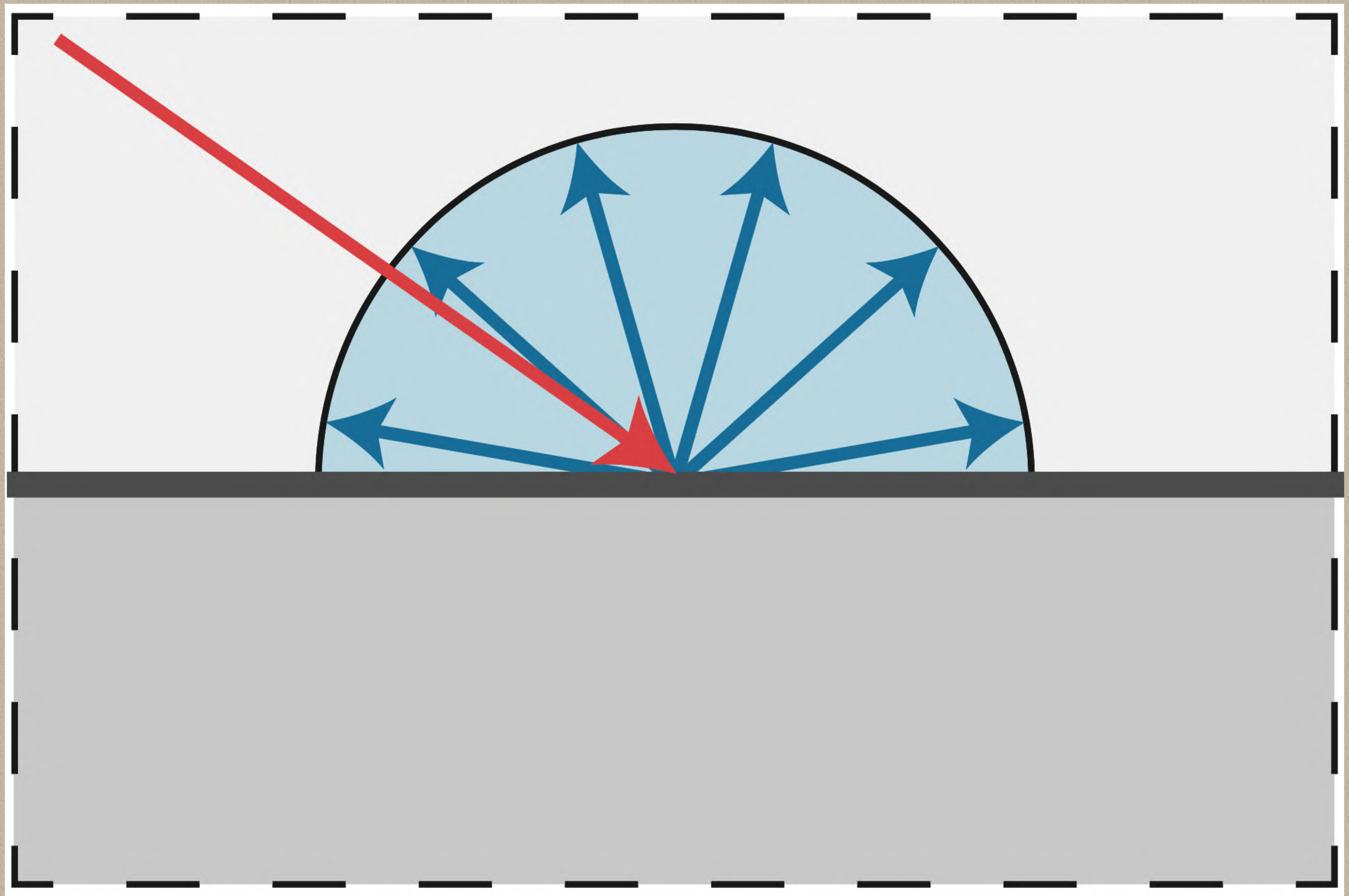


Rendered

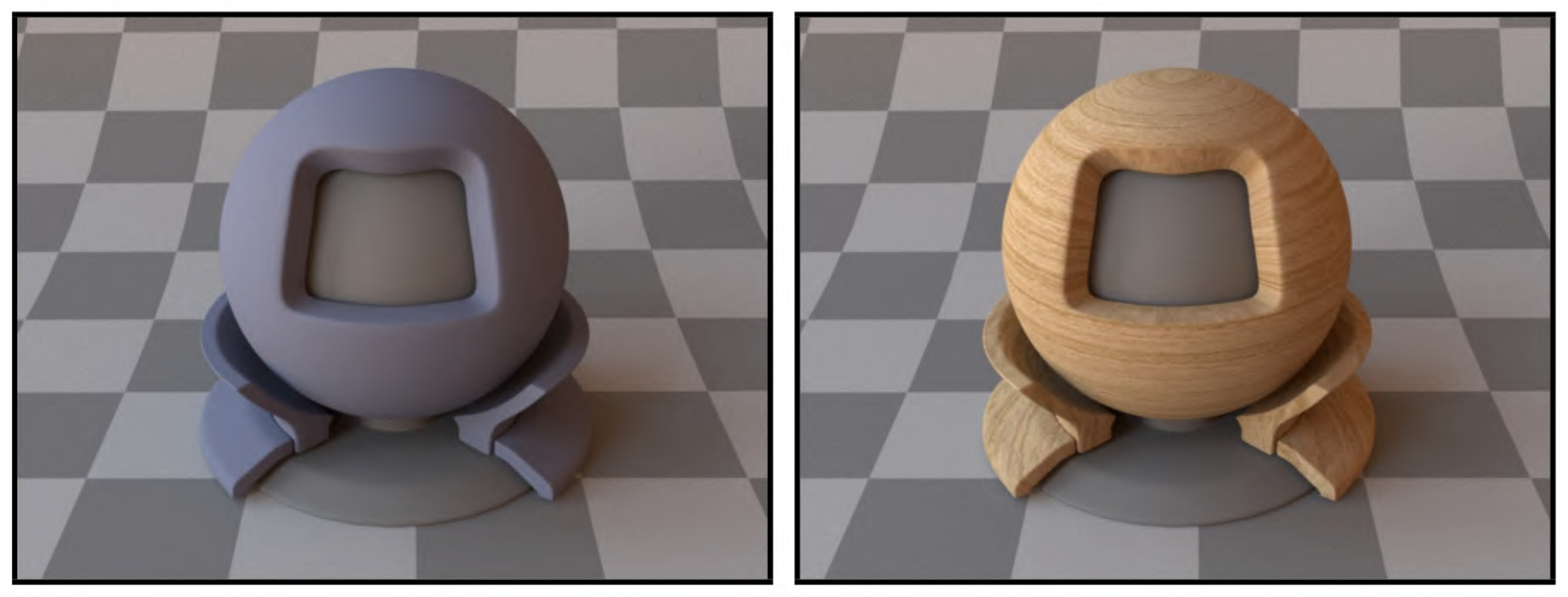
[From TurboSquid, created by artist 3dror]

Material == BRDF

What is this material?

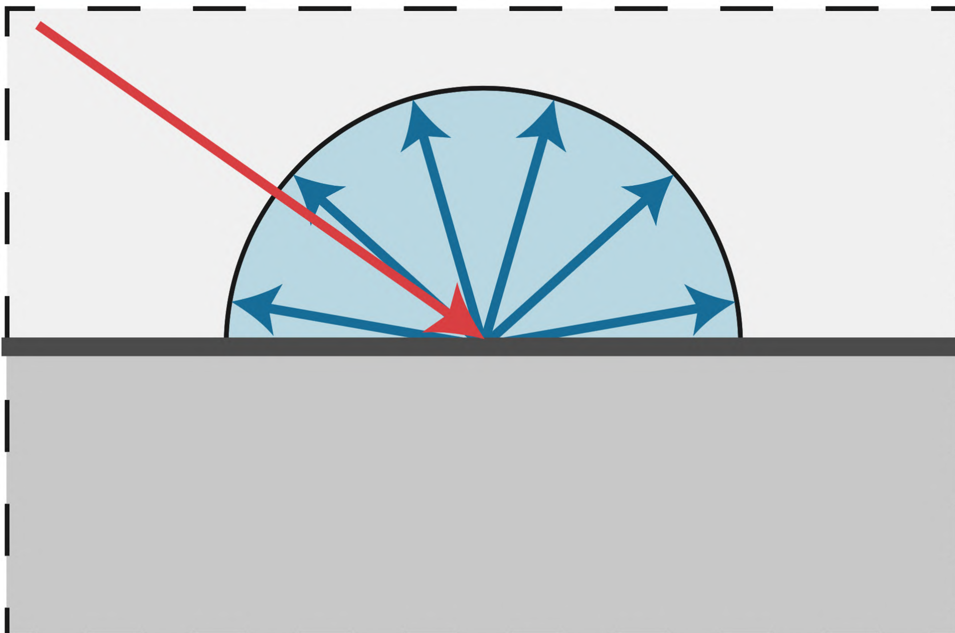


Diffuse / Lambertian Material (BRDF)



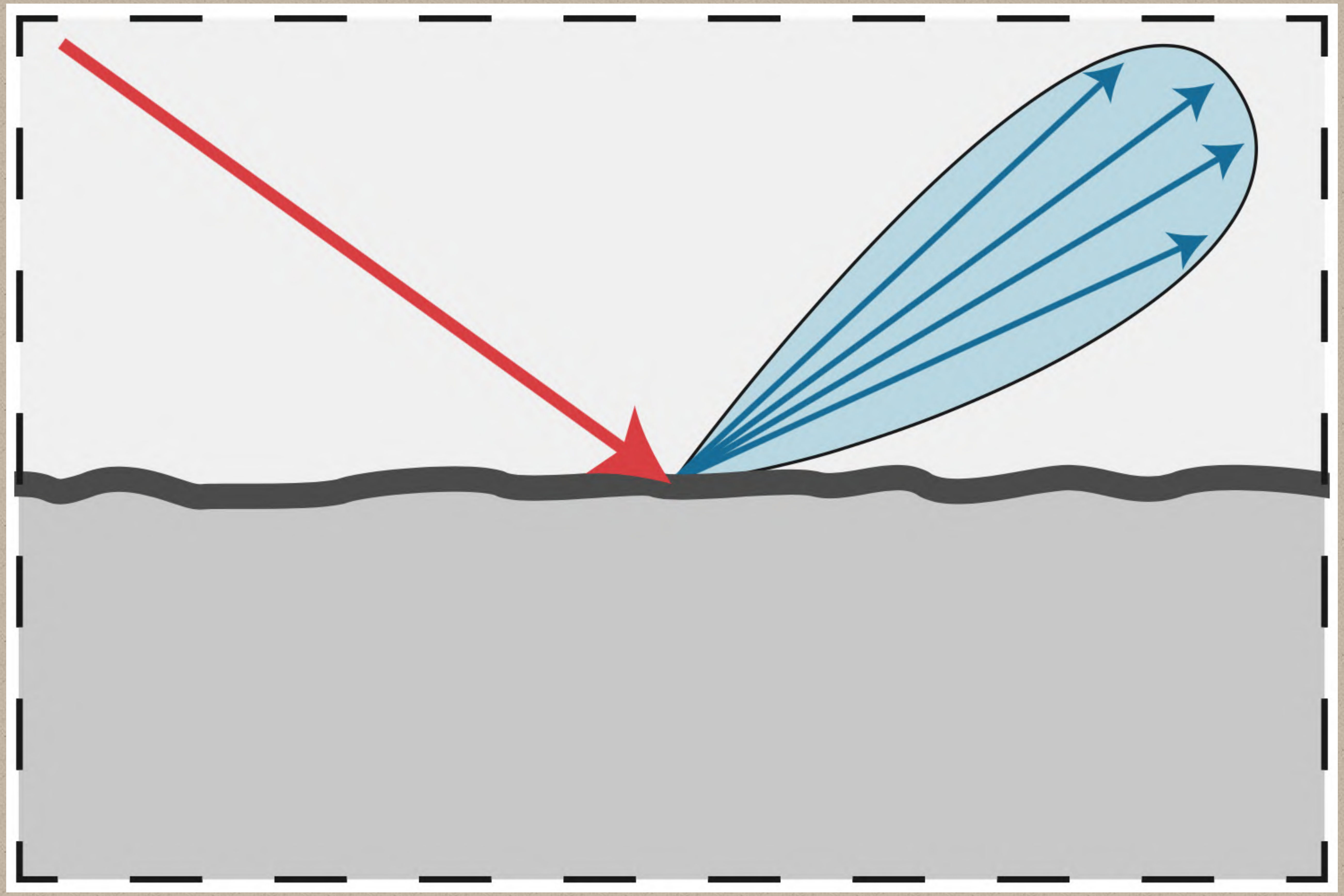
Uniform colored diffuse BRDF

Textured diffuse BRDF

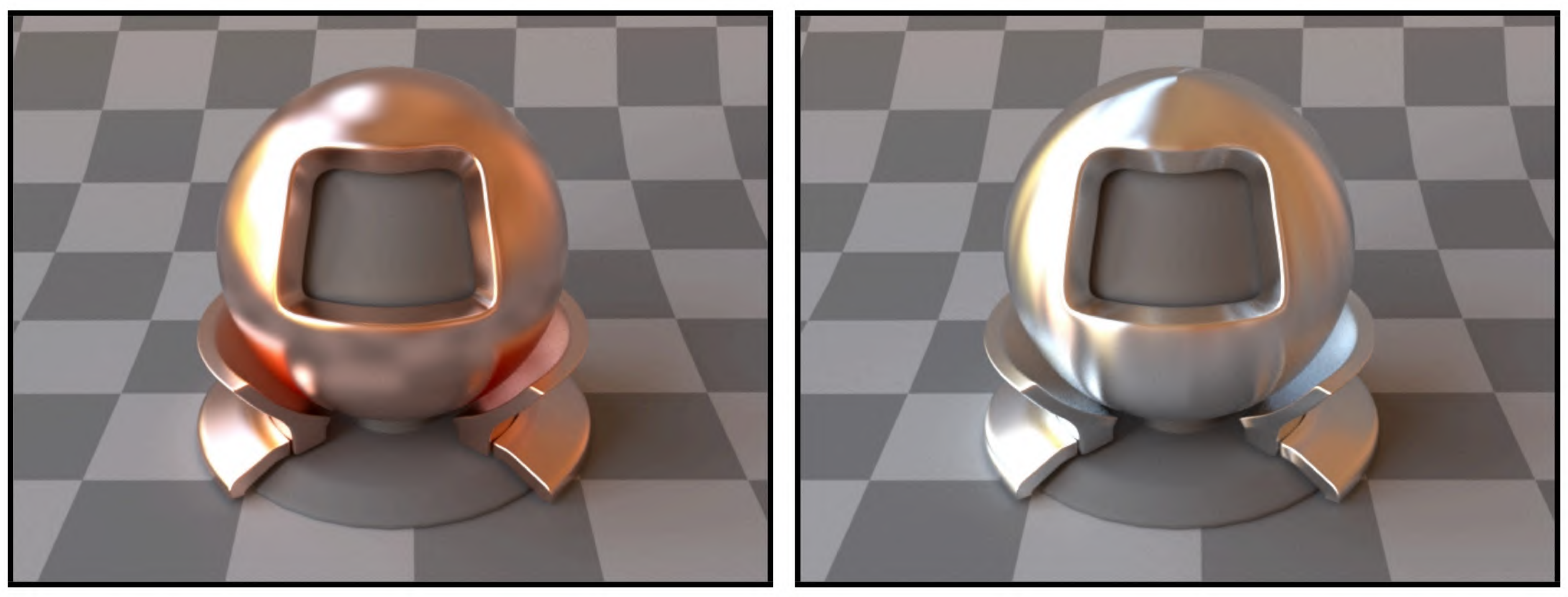


[Mitsuba renderer, Wenzel Jakob, 2010]

What is this material?

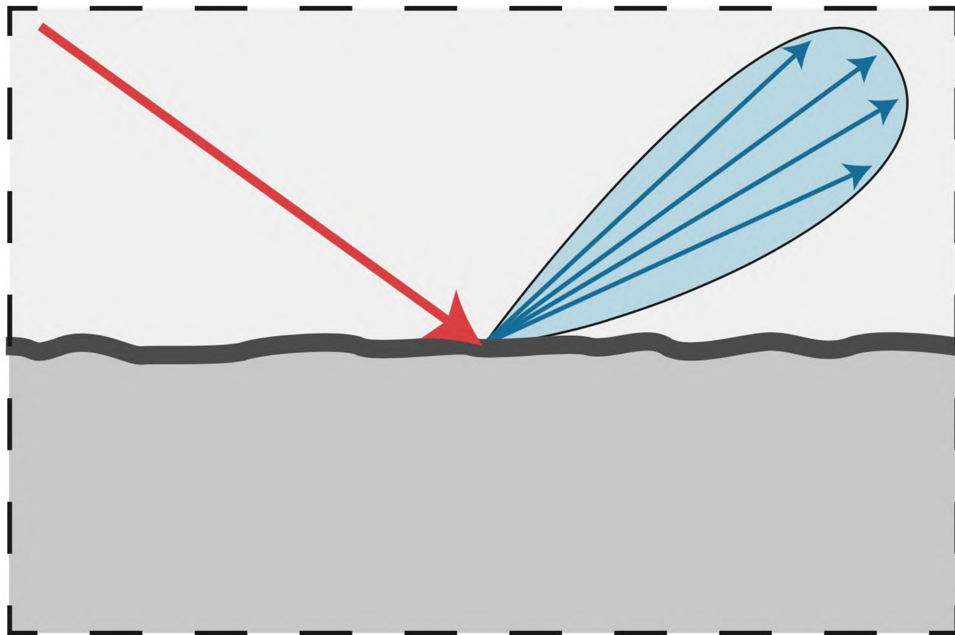


Glossy material (BRDF)



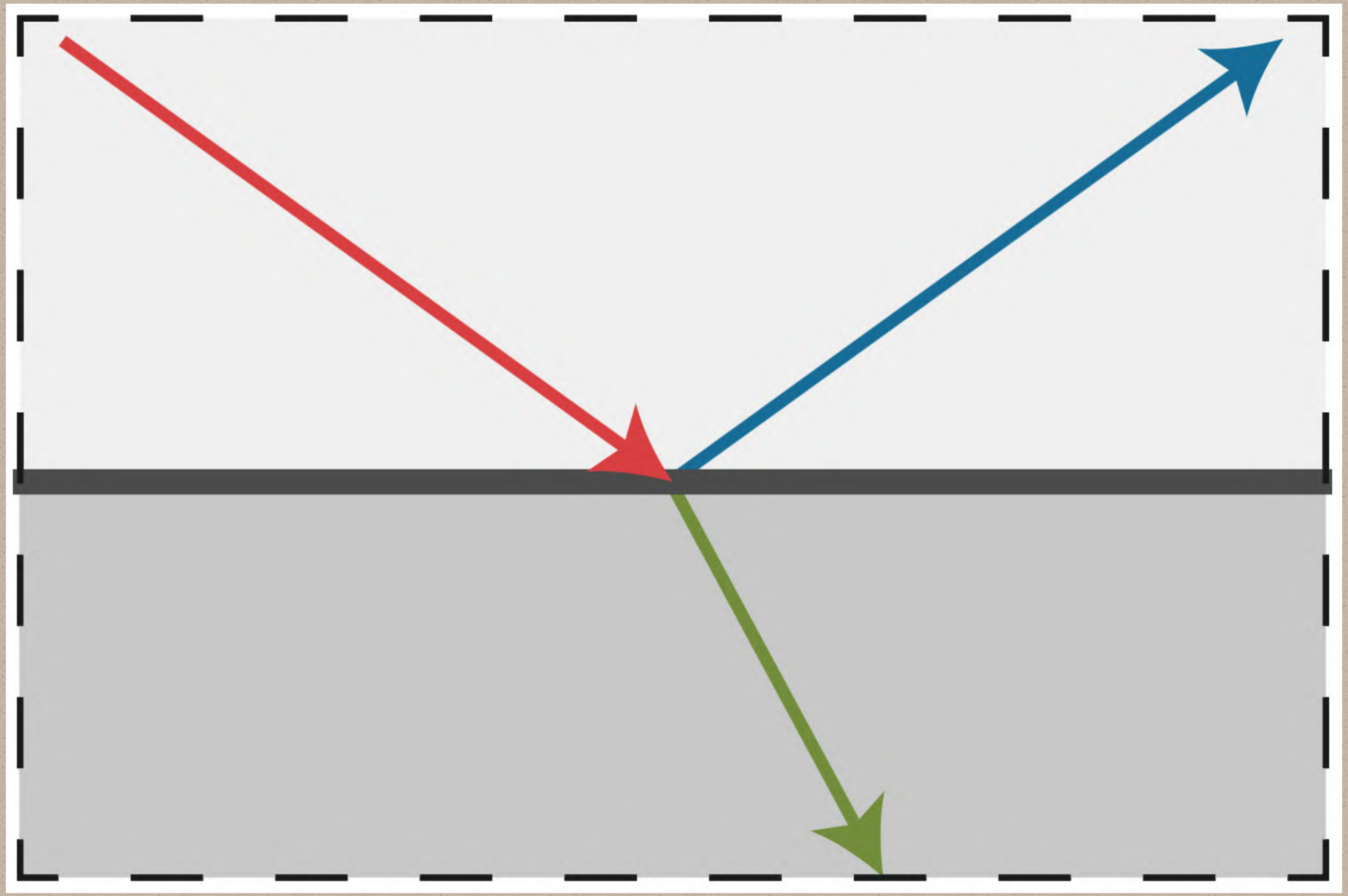
Copper

Aluminum

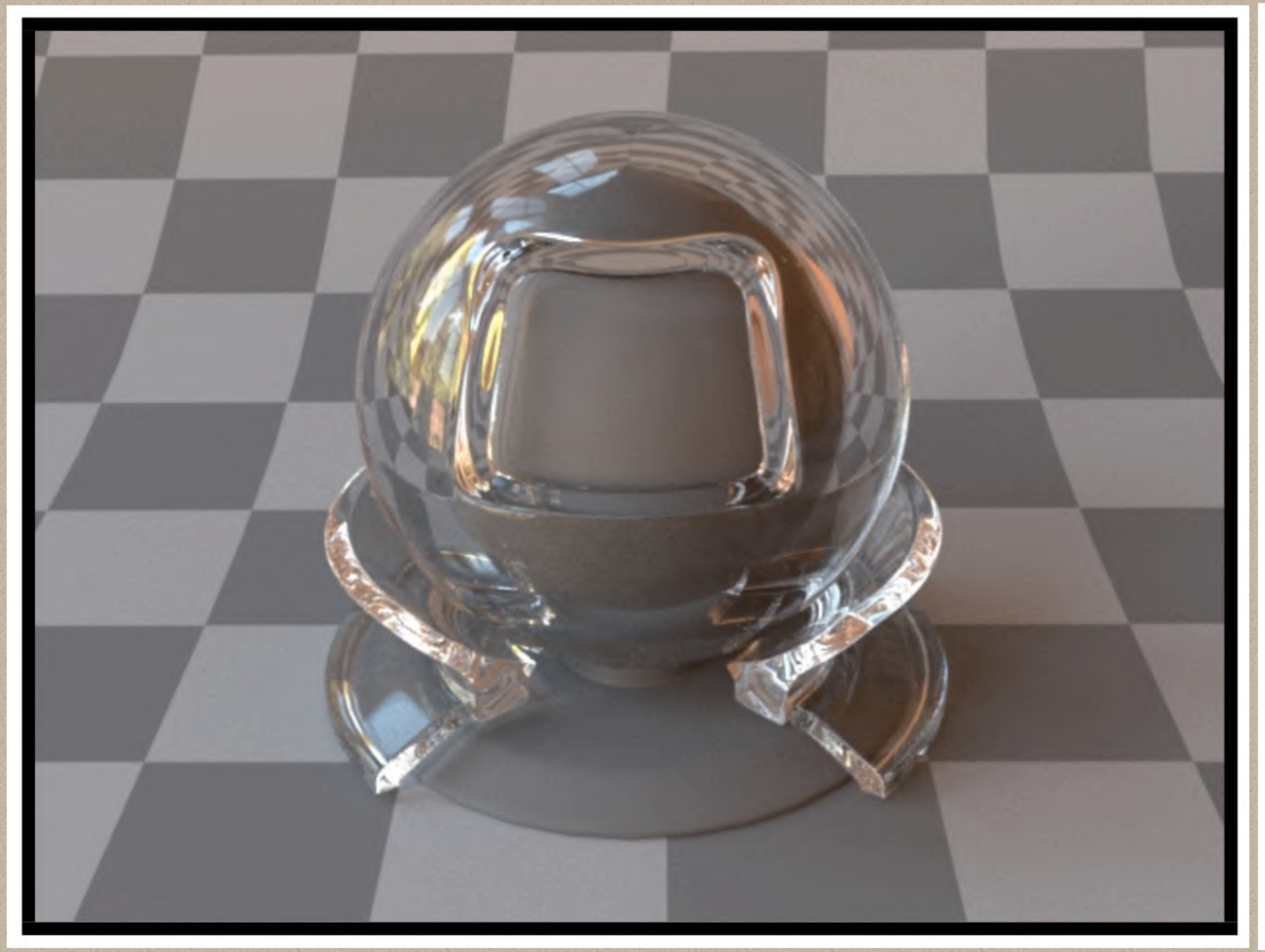


[Mitsuba renderer, Wenzel Jakob, 2010]

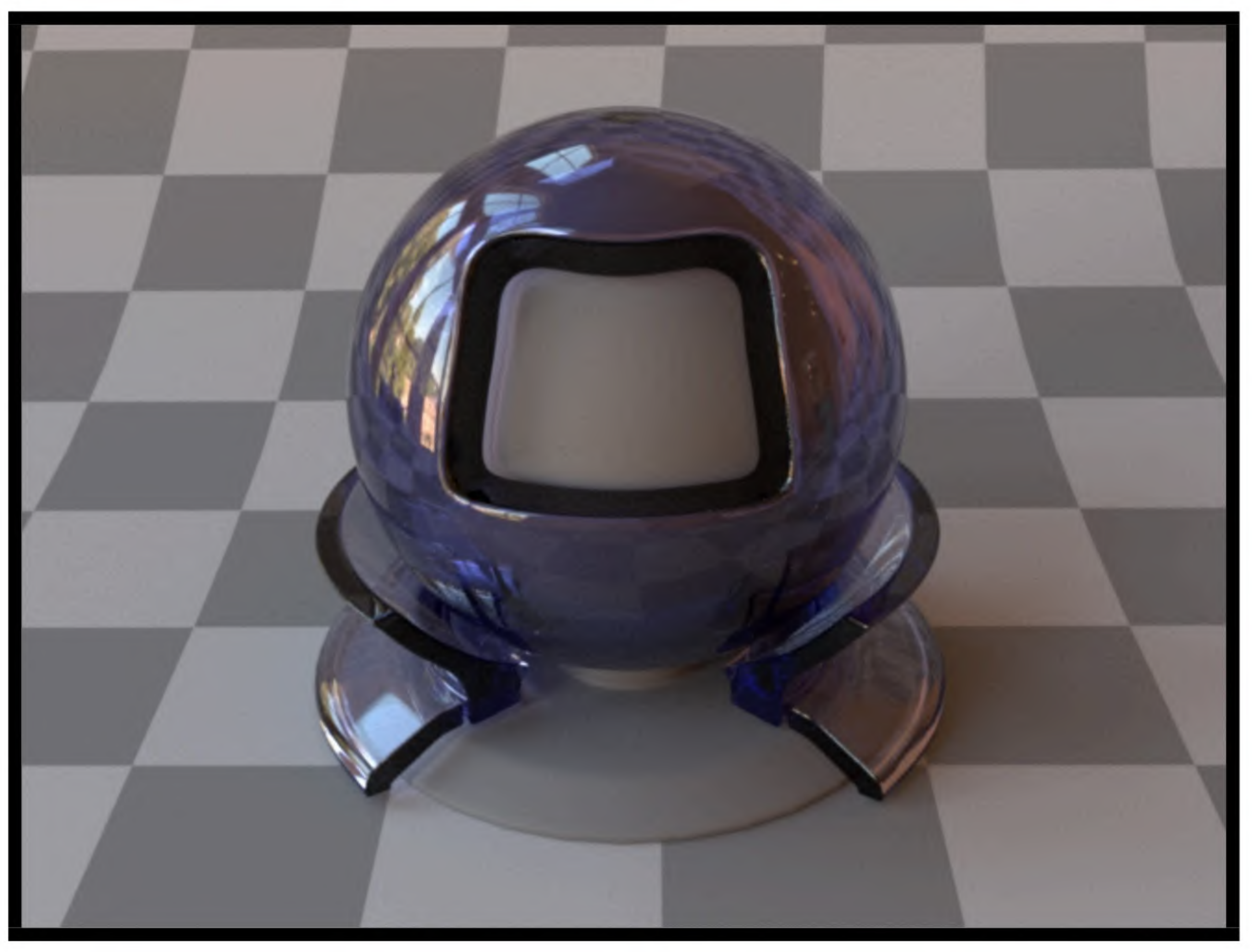
What is this material?



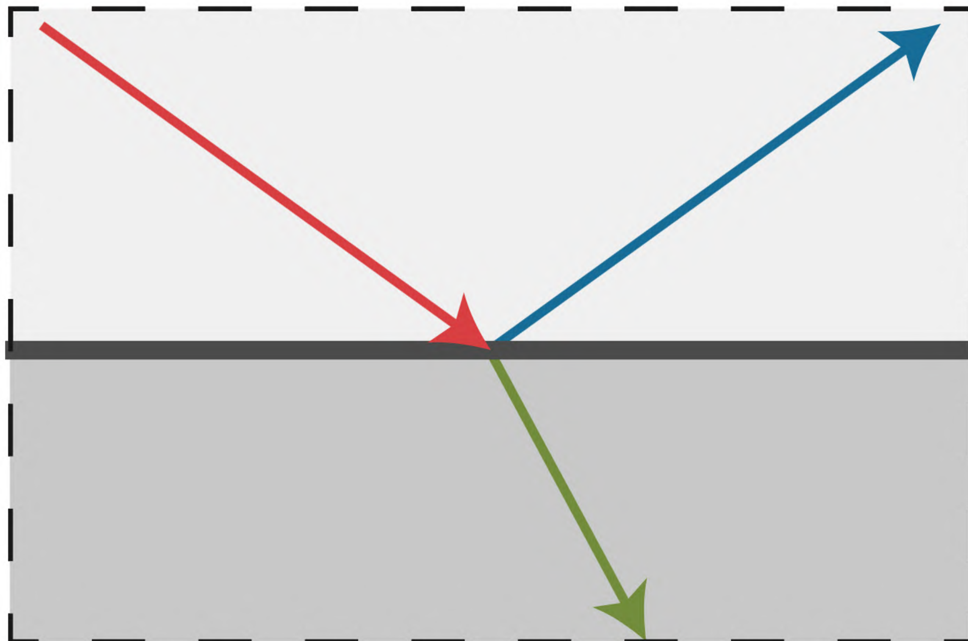
Ideal reflective / refractive material (BSDF*)



Air <-> water interface



Air <-> glass interface
(with absorption)



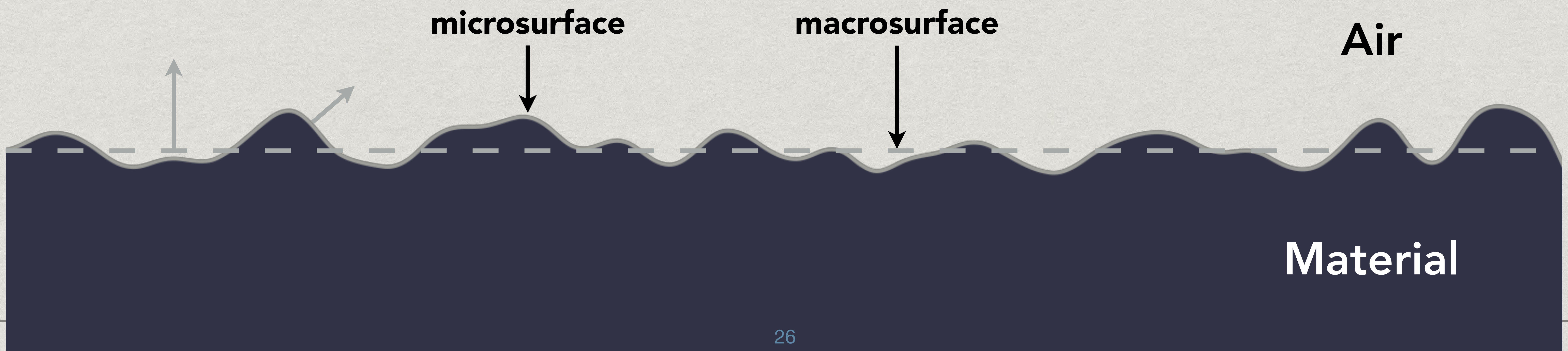
[Mitsuba renderer, Wenzel Jakob, 2010]

Microfacet BRDF

Microfacet model in the real world

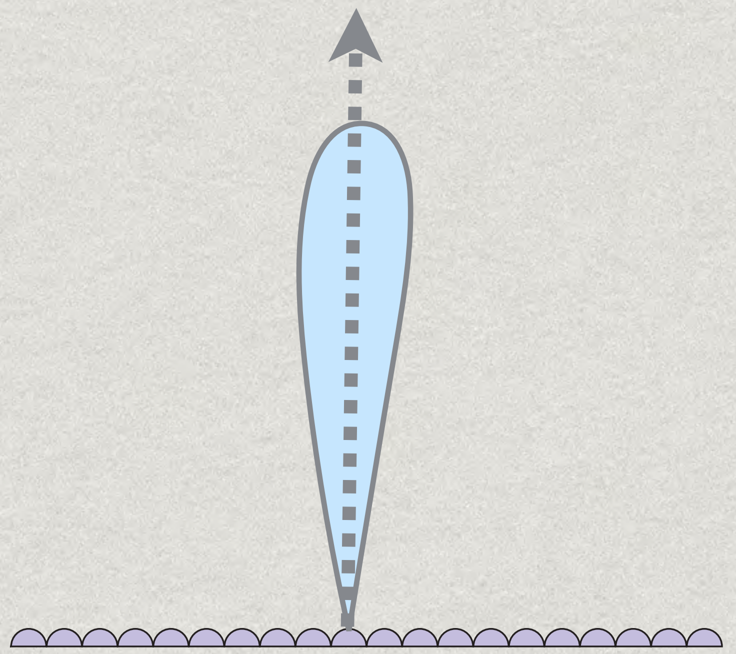
Microfacet Theory

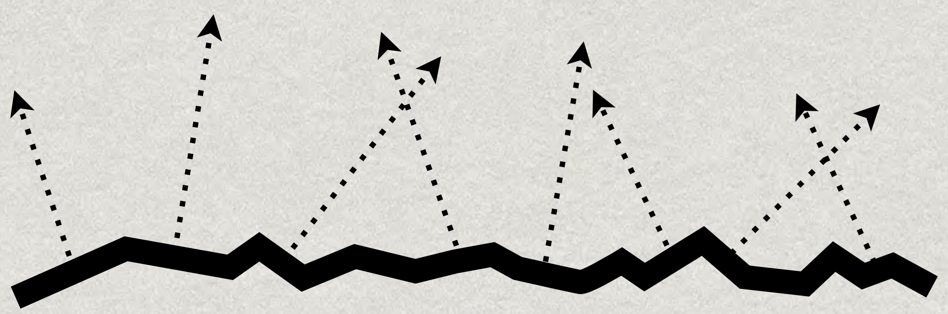
- Rough surface
  - Macroscale: flat & rough
  - Microscale: bumpy & specular
- Individual elements of surface act like mirrors
  - Known as Microfacets
  - Each microfacet has its own normal



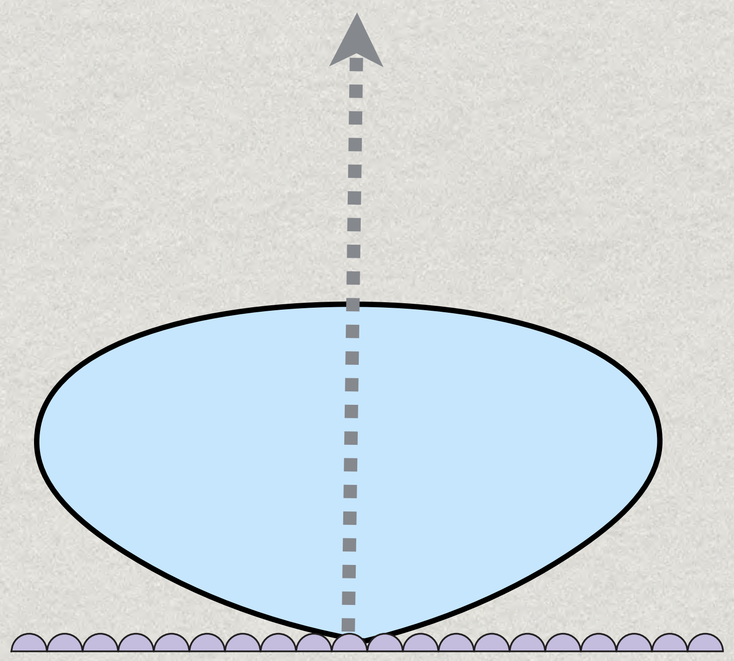
Microfacet BRDF

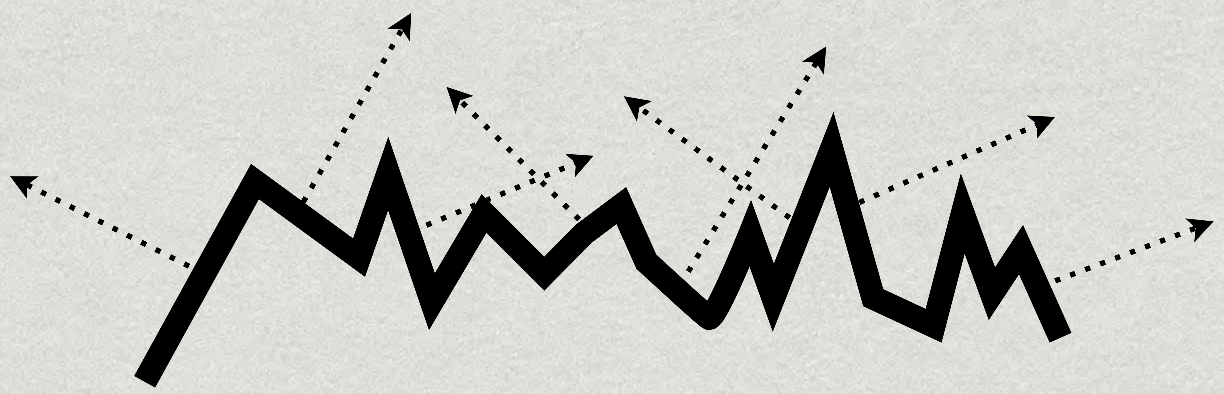
- Key: the distribution of microfacets' normals
  - Concentrated <==> glossy

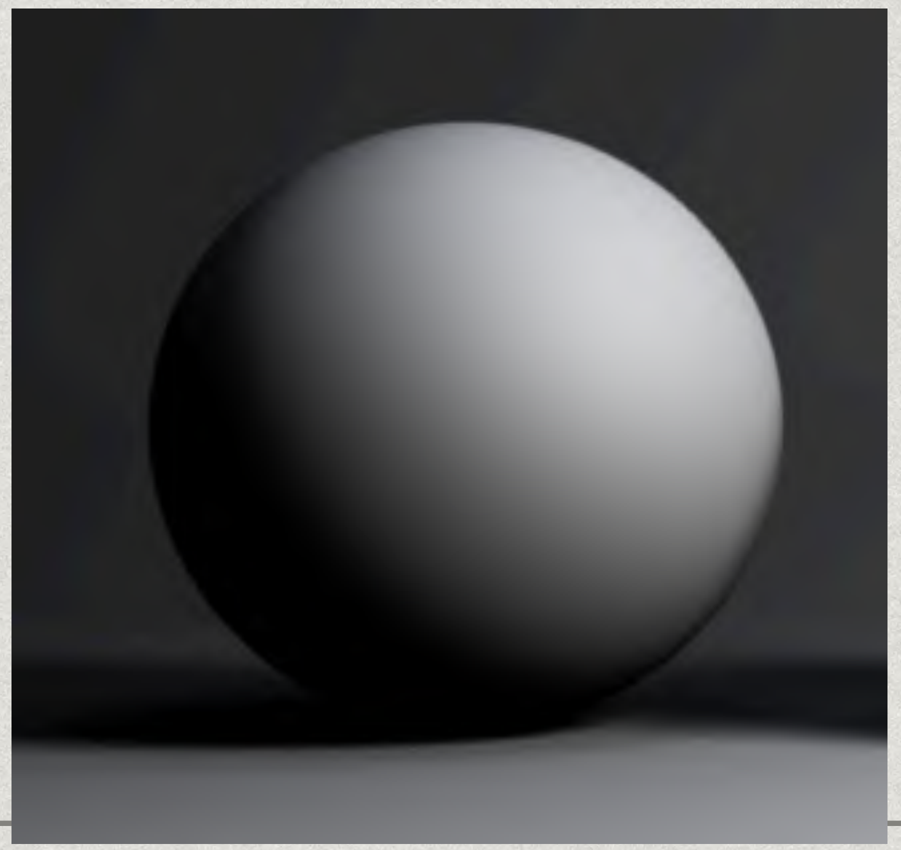




  - Spread <==> diffuse

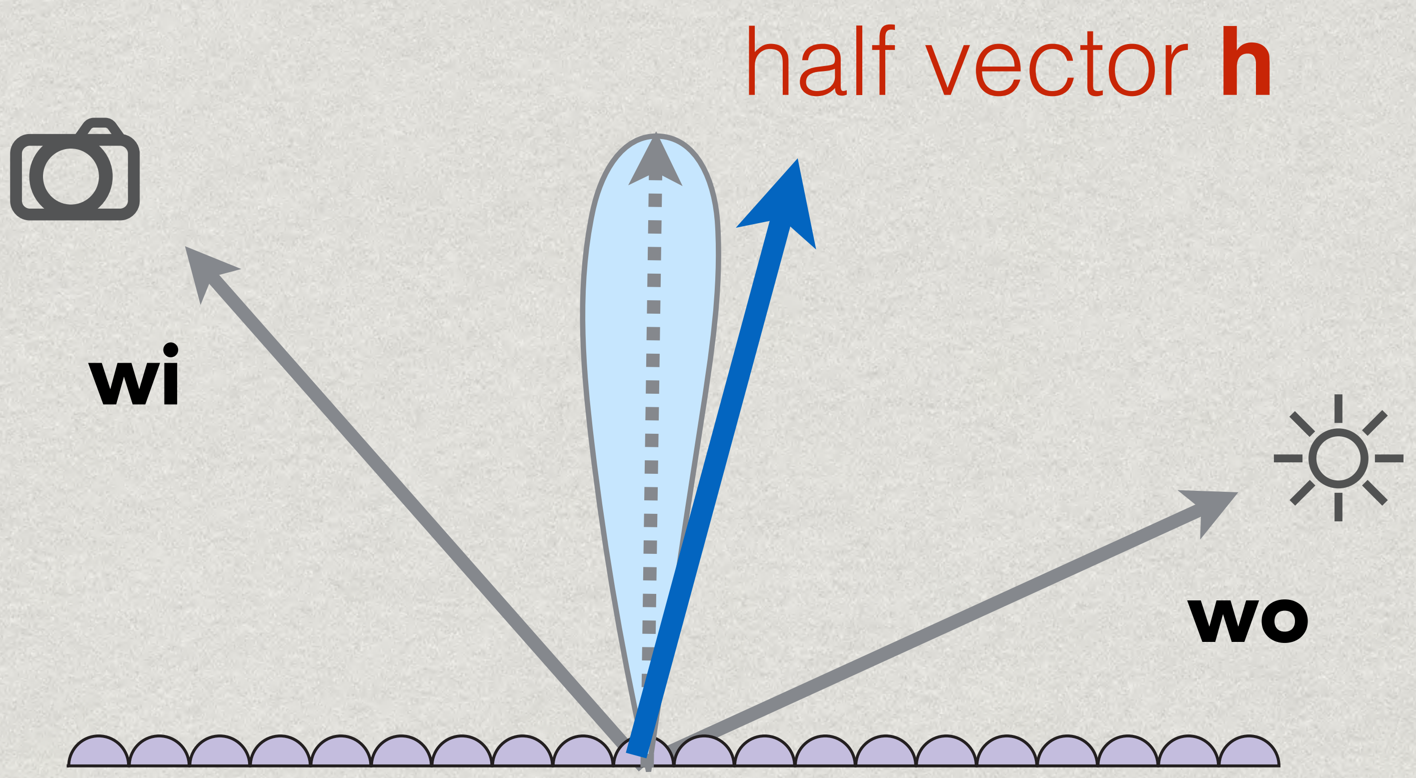


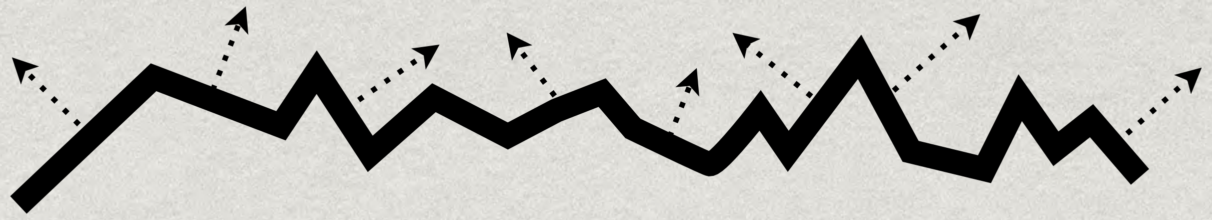




Microfacet BRDF

- What kind of microfacets reflect wi to wo?
(hint: microfacets are mirrors)

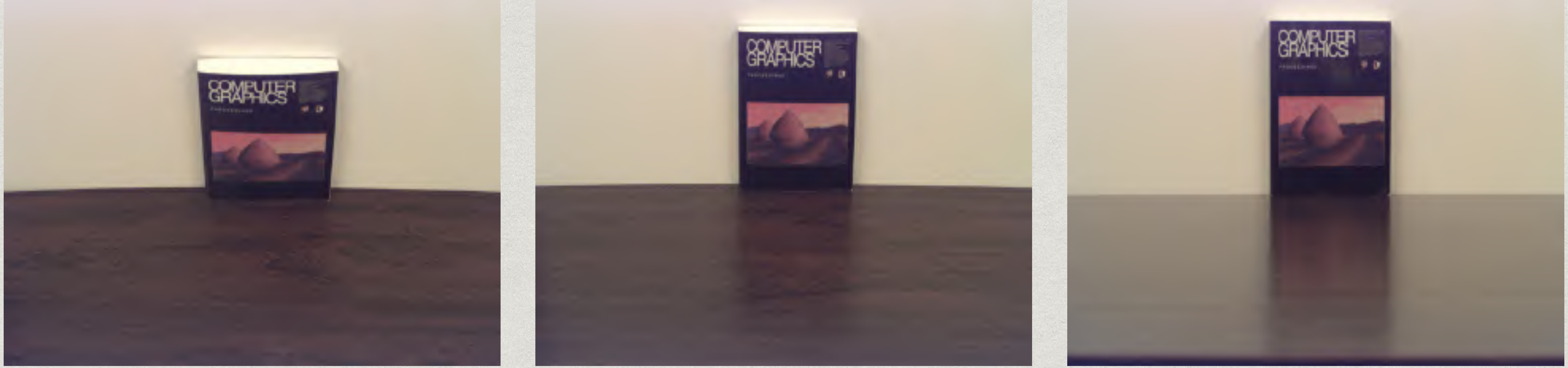




$$f(\mathbf{i},\mathbf{o}) = \frac{\overset{\text{Fresnel term}}{\boxed{\mathbf{F}(\mathbf{i},\mathbf{h})}}\,\overset{\text{shadowing-masking term}}{\boxed{\mathbf{G}(\mathbf{i},\mathbf{o},\mathbf{h})}}\,\overset{\textbf{distribution of normals}}{\boxed{\mathbf{D}(\mathbf{h})}}}{4(\mathbf{n},\mathbf{i})(\mathbf{n},\mathbf{o})}$$

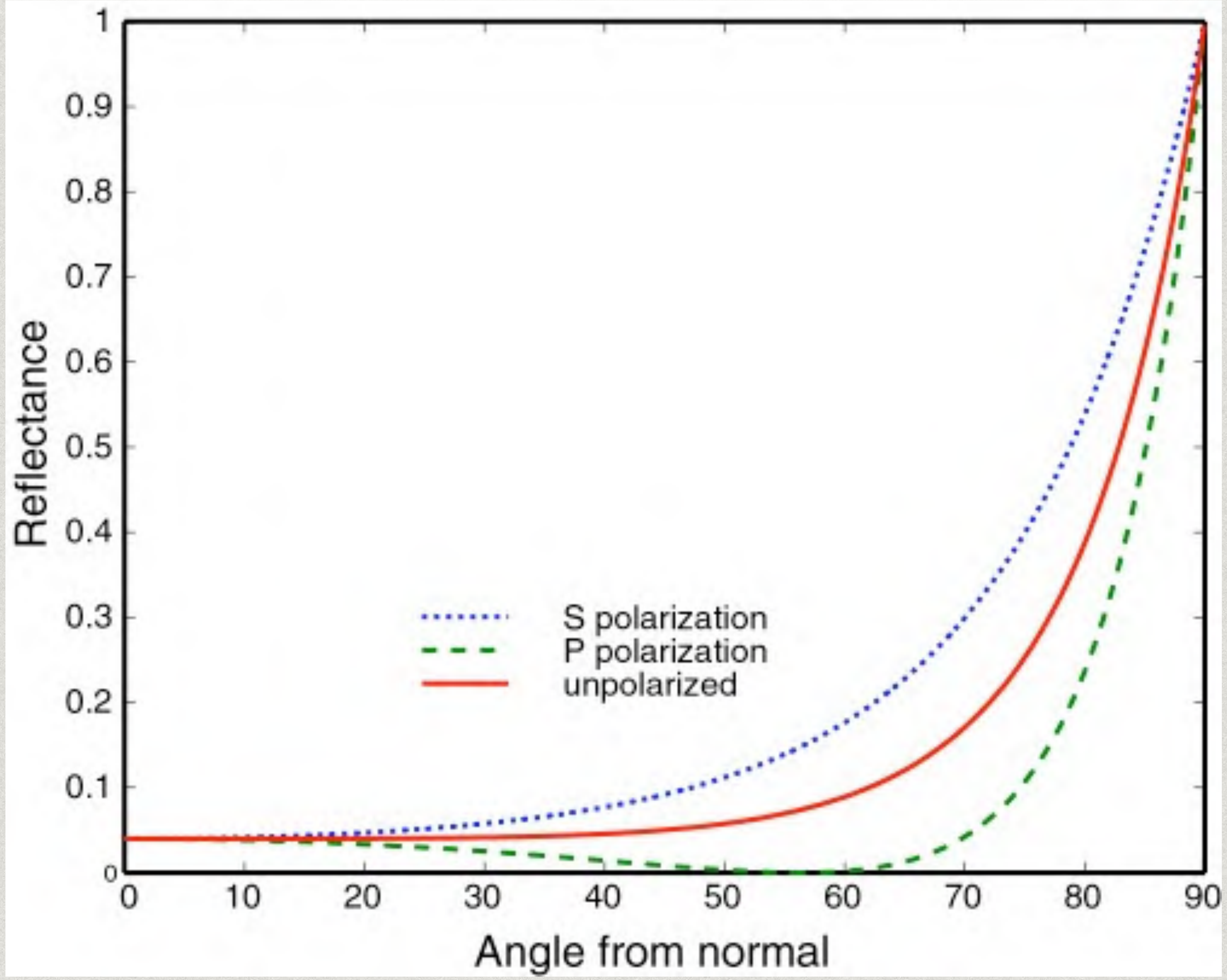
Fresnel Reflection / Term

Reflectance depends on incident angle (and polarization of light)

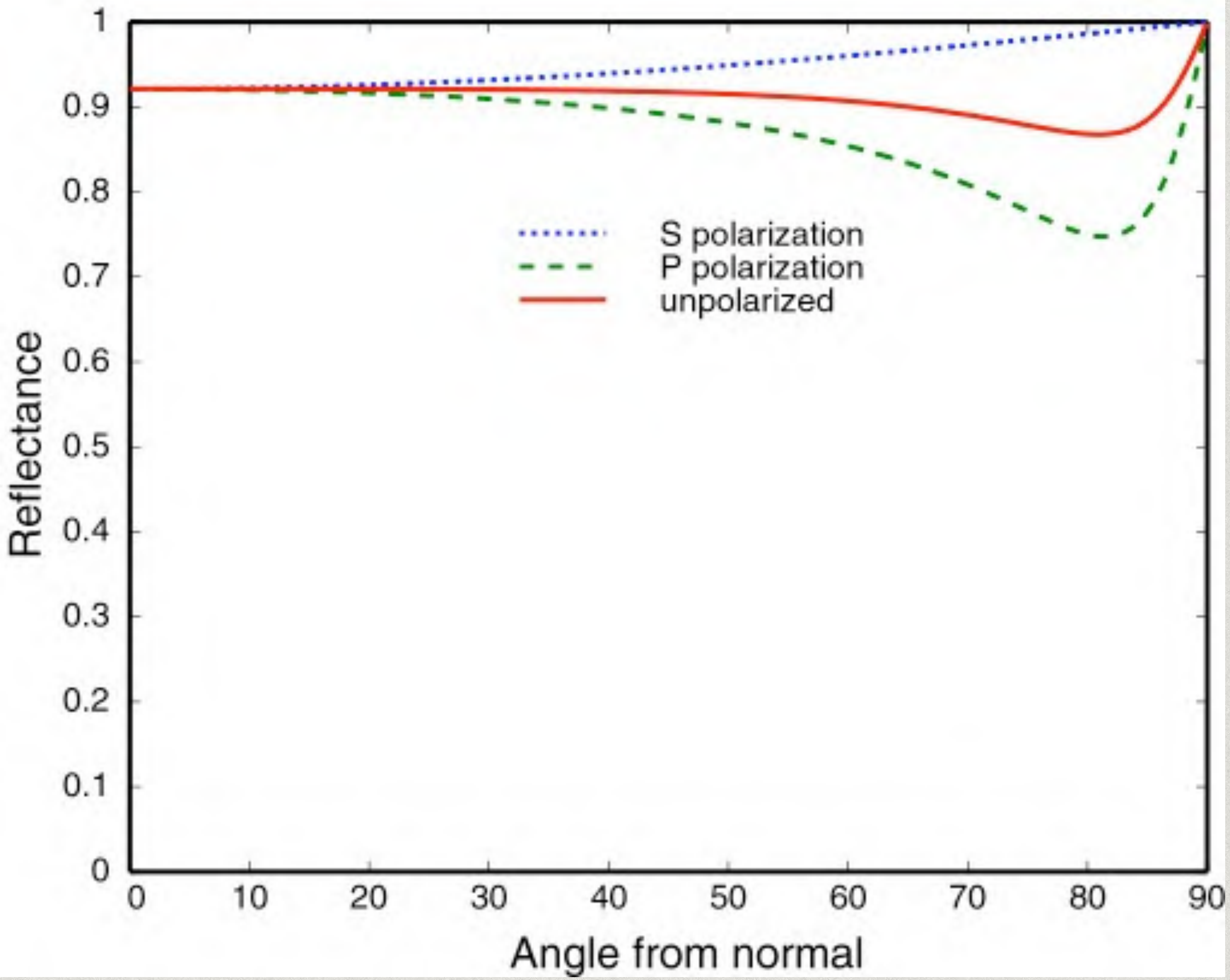


This example: reflectance increases with grazing angle

Fresnel Term (Dielectric, $\eta = 1.5$)



Fresnel Term (Conductor)

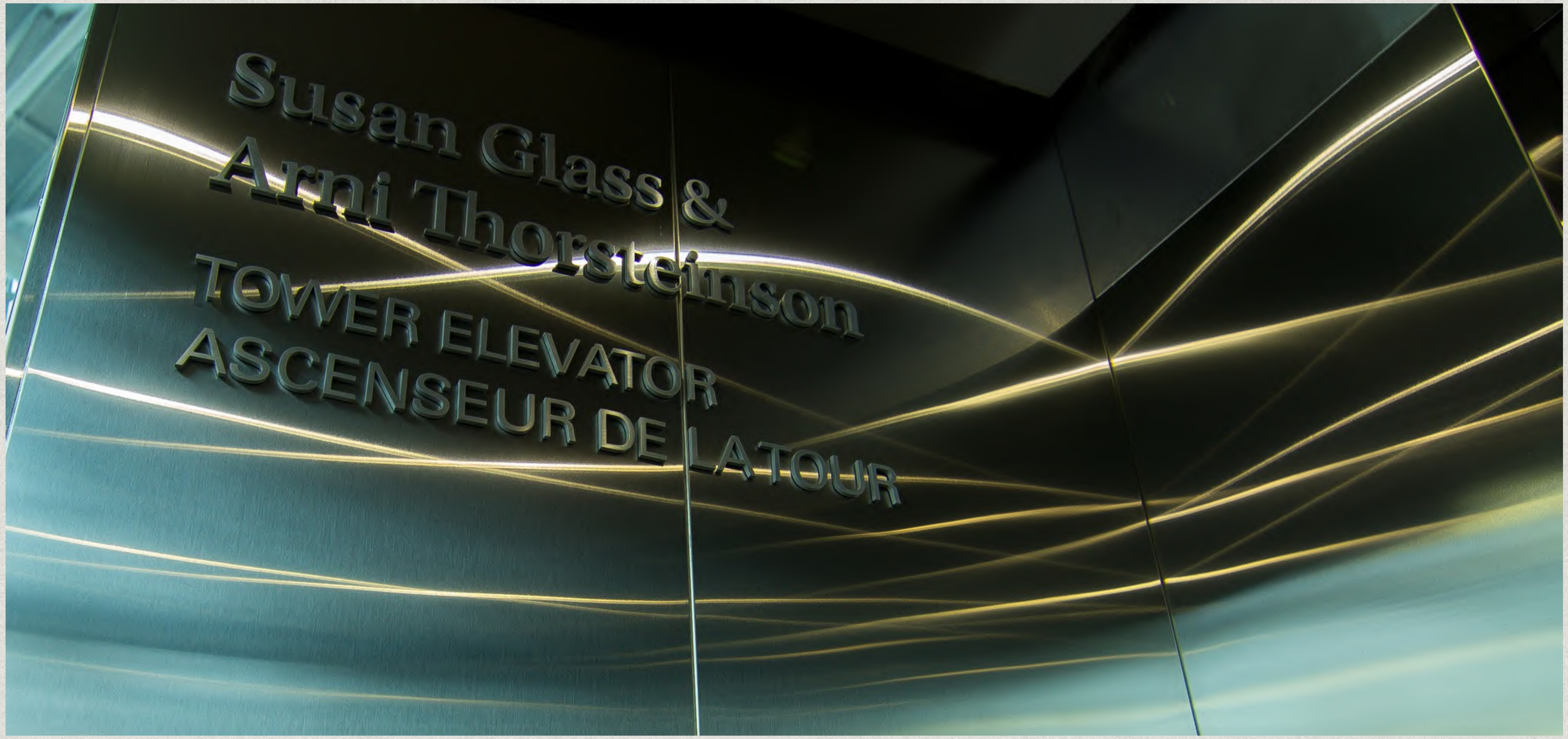


Microfacet BRDF: Examples

[Autodesk Fusion 360]

Isotropic / Anisotropic Materials (BRDFs)

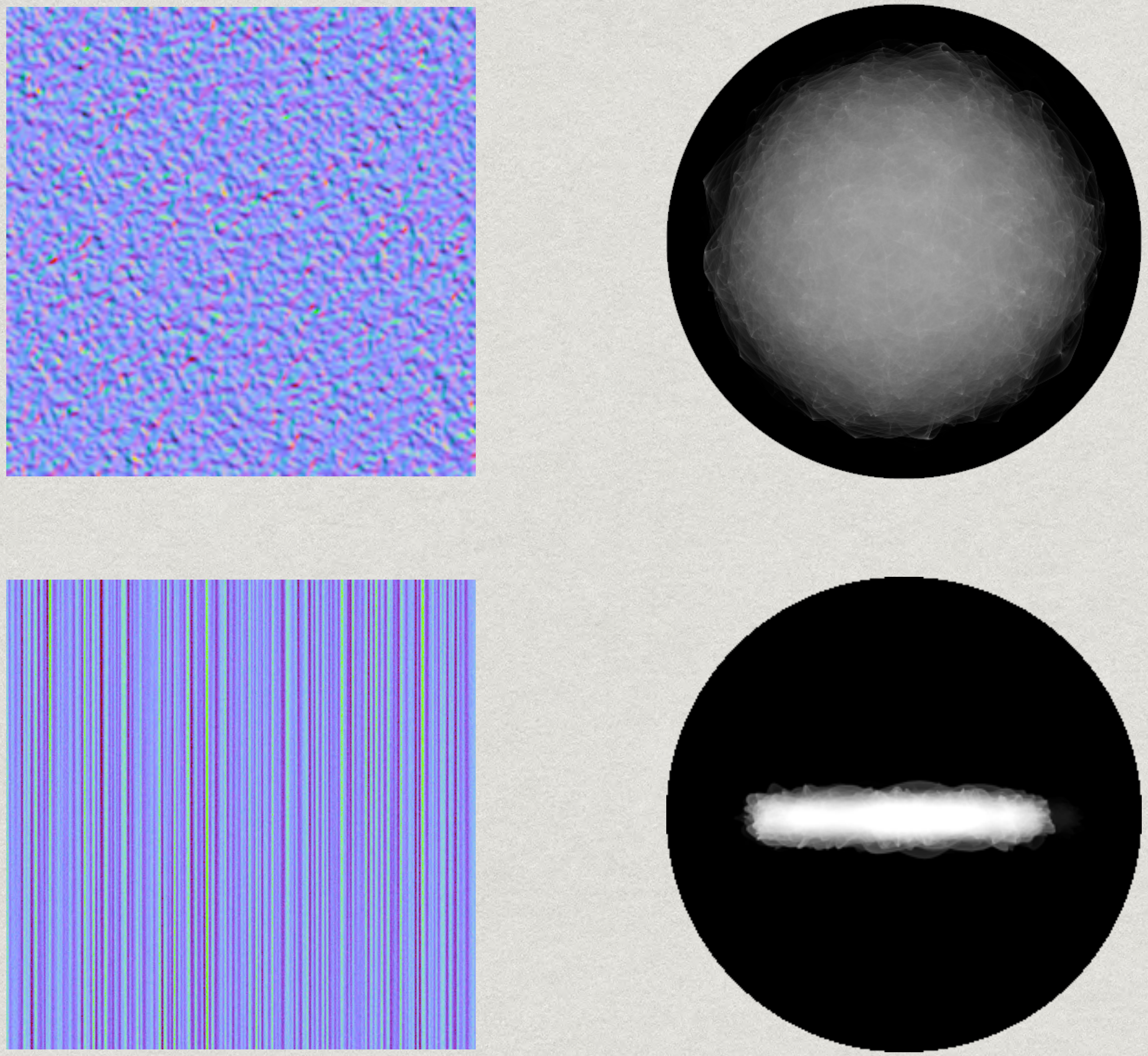
- So far, Point light + Metal = Round / Elliptical highlight
- What can we see inside an elevator?



Inside an elevator

Isotropic / Anisotropic Materials (BRDFs)

- Key: directionality of underlying surface



Isotropic

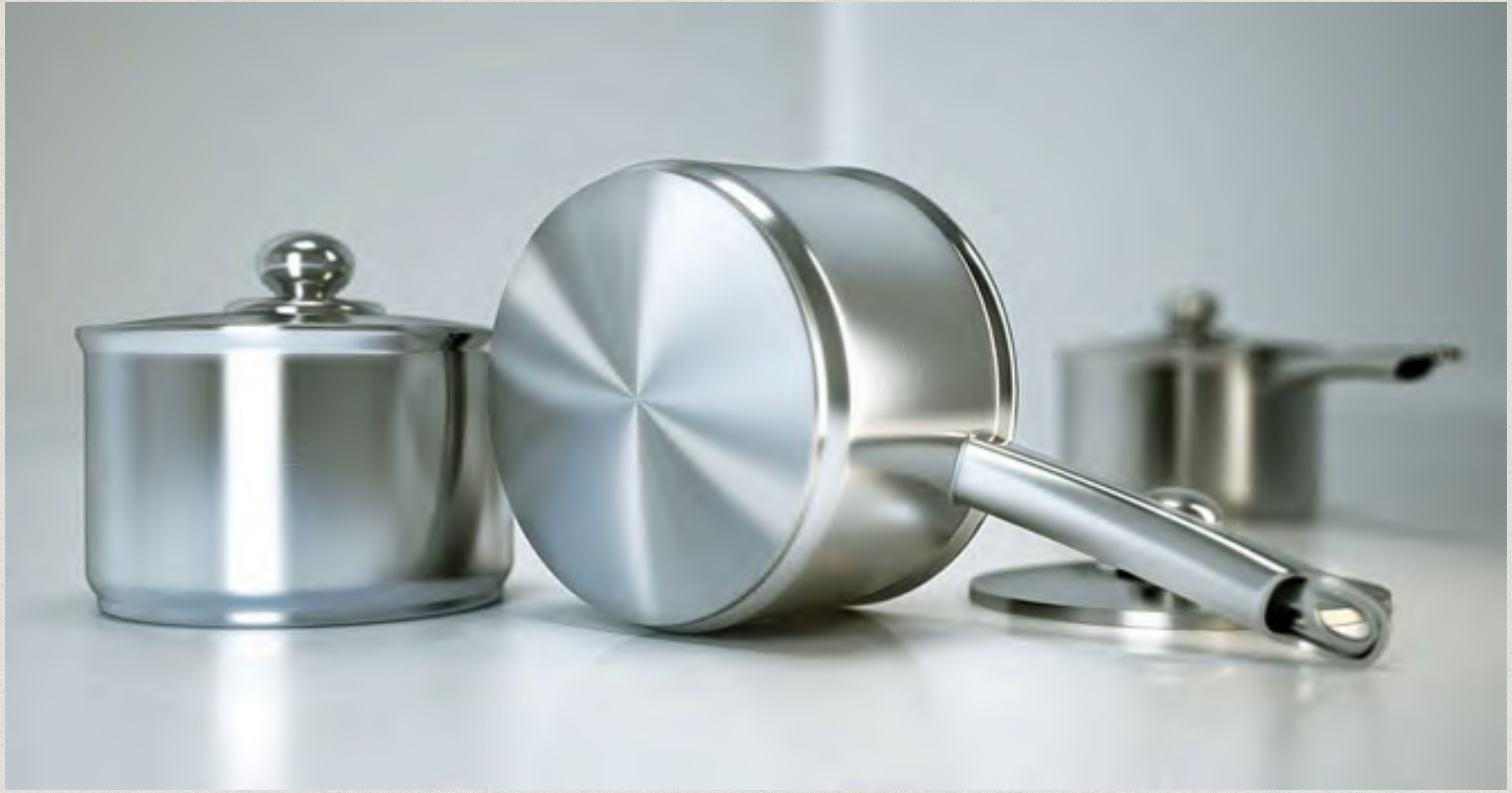
Anisotropic

Surface (normals)

BRDF (fix wi, vary wo)

Anisotropic BRDF: Brushed Metal

- How is the pan brushed?



[VRay renderer]

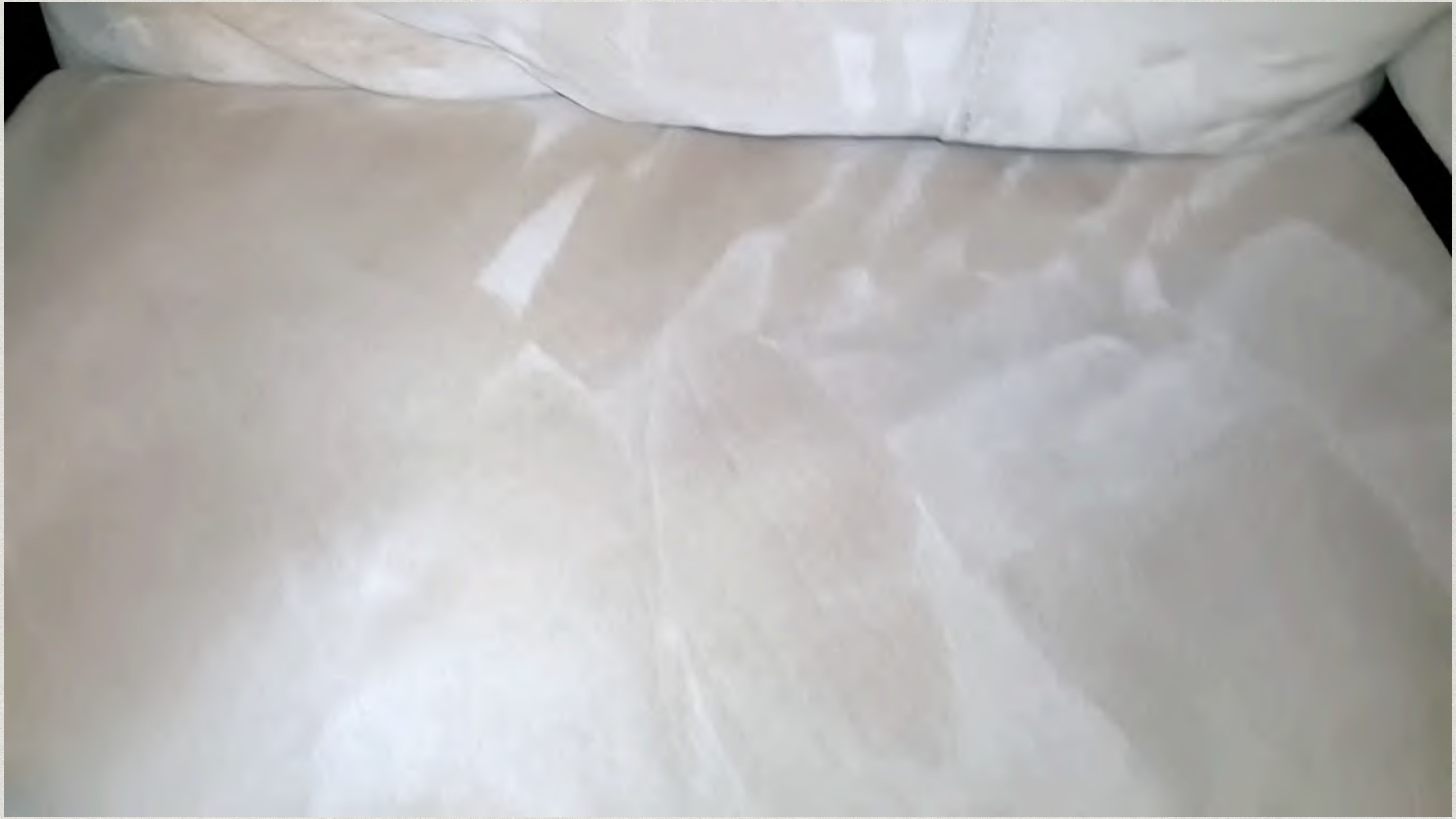
Anisotropic BRDF: Nylon

[Westin et al. 1992]

Anisotropic BRDF: Velvet

[Westin et al. 1992]

Anisotropic BRDF: Velvet

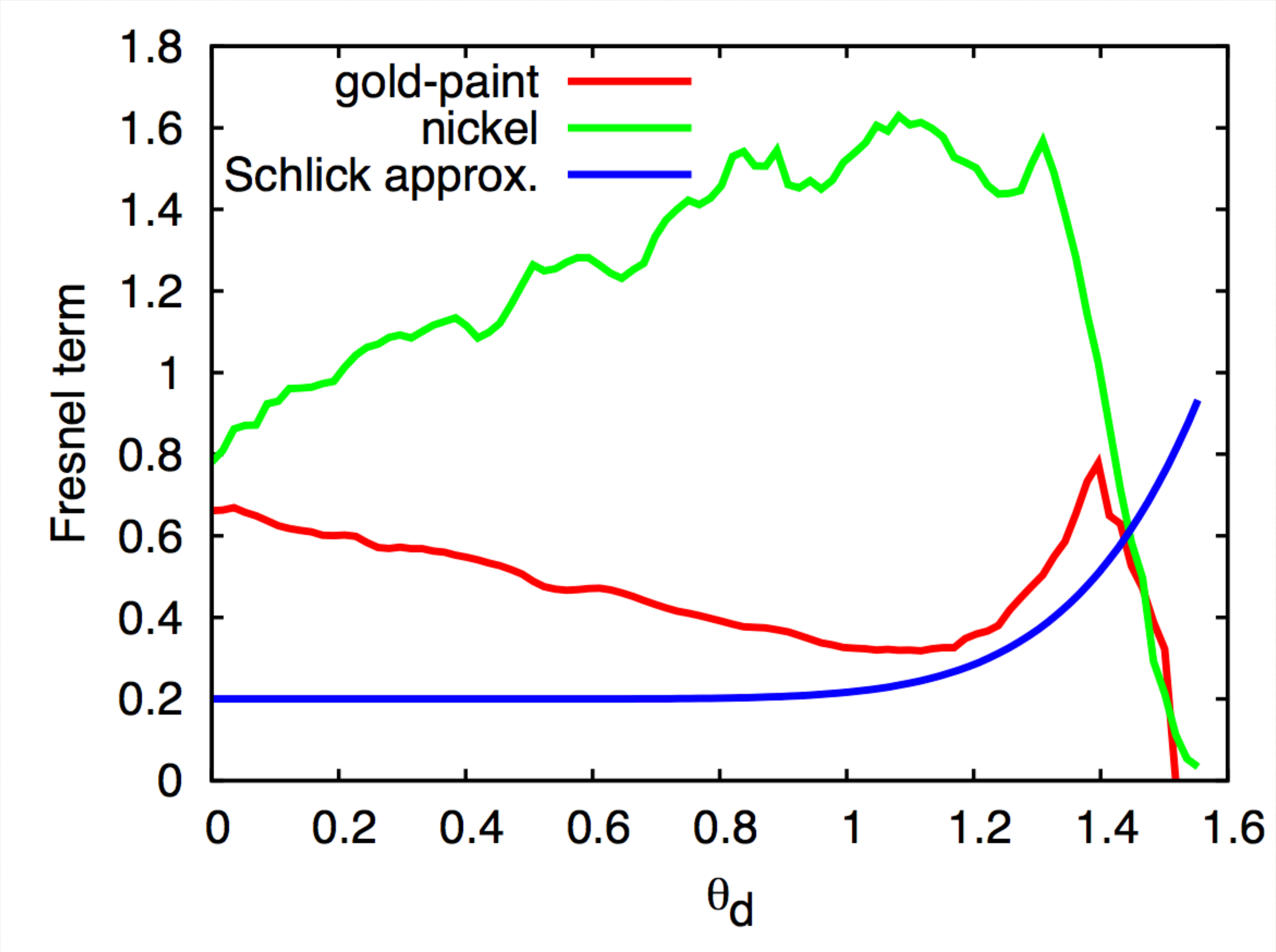


Material Capture

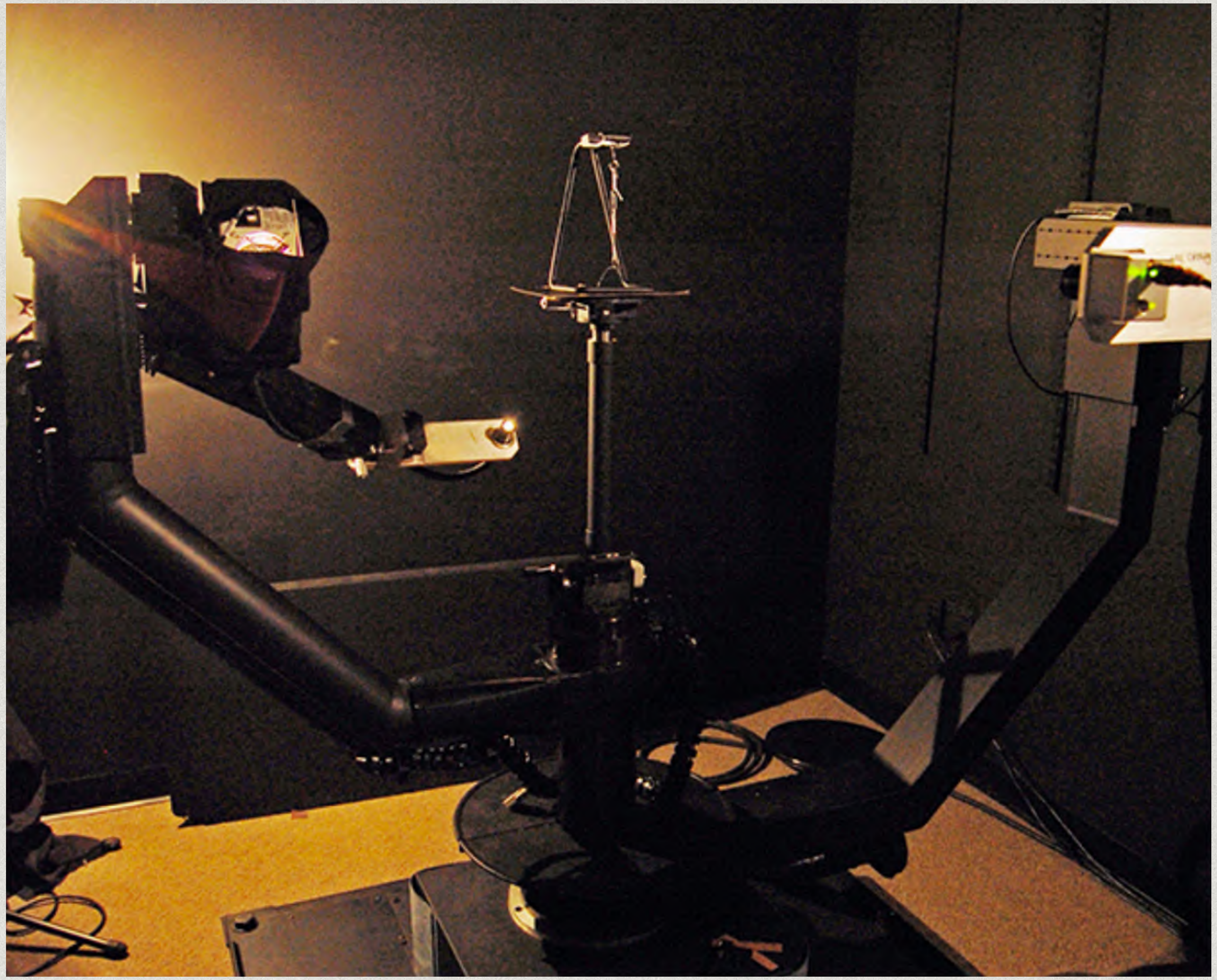
(BRDF measurement)

Measuring BRDFs: Motivation

- Avoid need to develop / derive models
    - Automatically includes all of the scattering effects present
- Can accurately render with real-world materials
    - Useful for product design, special effects, ...
- Theory vs. practice:



Measuring BRDFs: gonioreflectometer

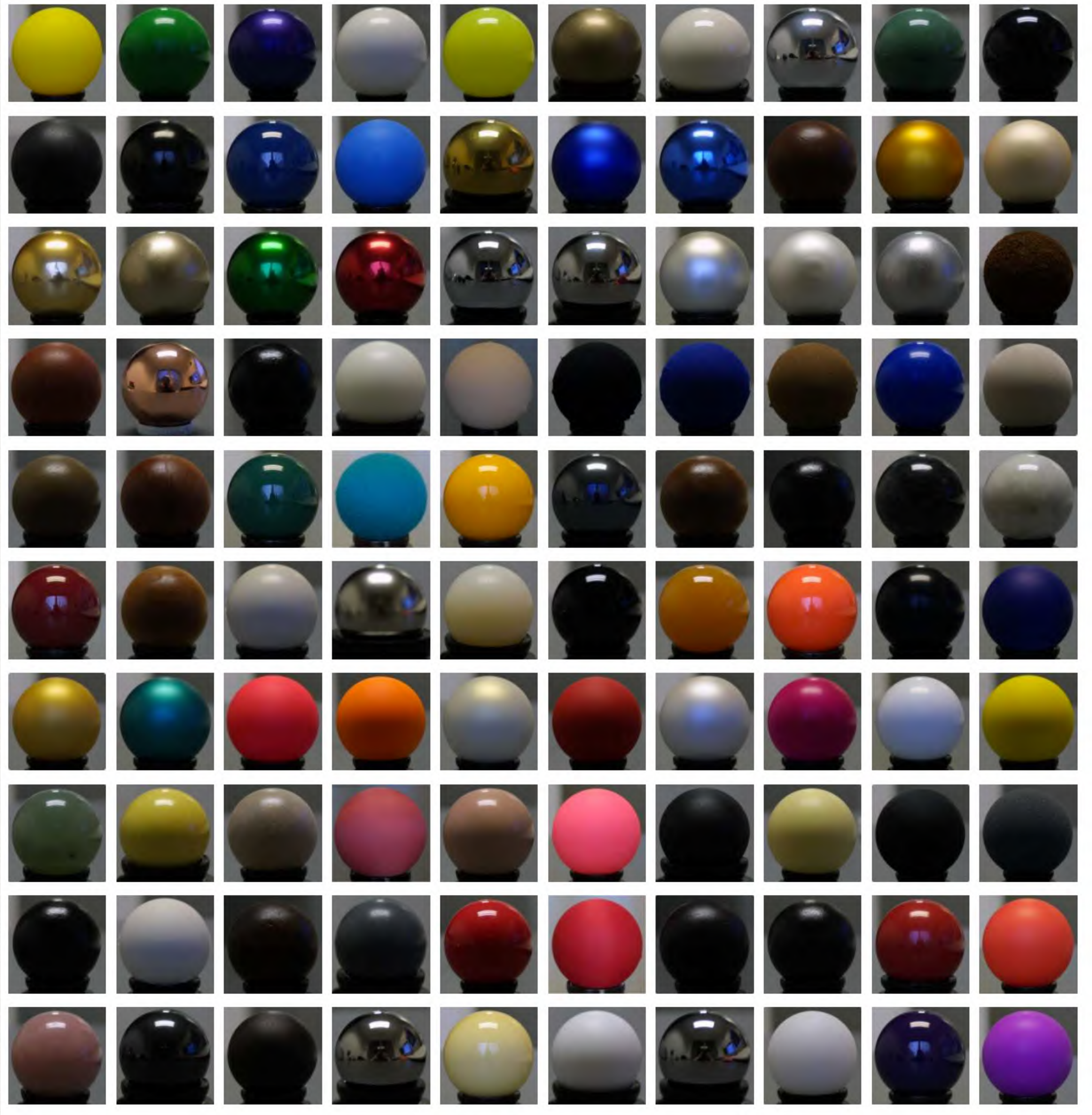


Spherical gantry at UC San Diego

Tabular Representation

$$(\theta_i, \theta_o, |\phi_i - \phi_o|)$$

- Store regularly-spaced samples in
  - Better: reparameterize angles to better match specularities
- Generally need to resample measured values to table
- Very high storage requirements



MERL BRDF Database
[Matusik et al. 2004]
90*90*180 measurements

Advanced materials

- Participating media
- Translucent material (BSSRDF)
- Cloth
- Granular material
- Procedural appearance
- …

Participating Media

Participating Media: Fog

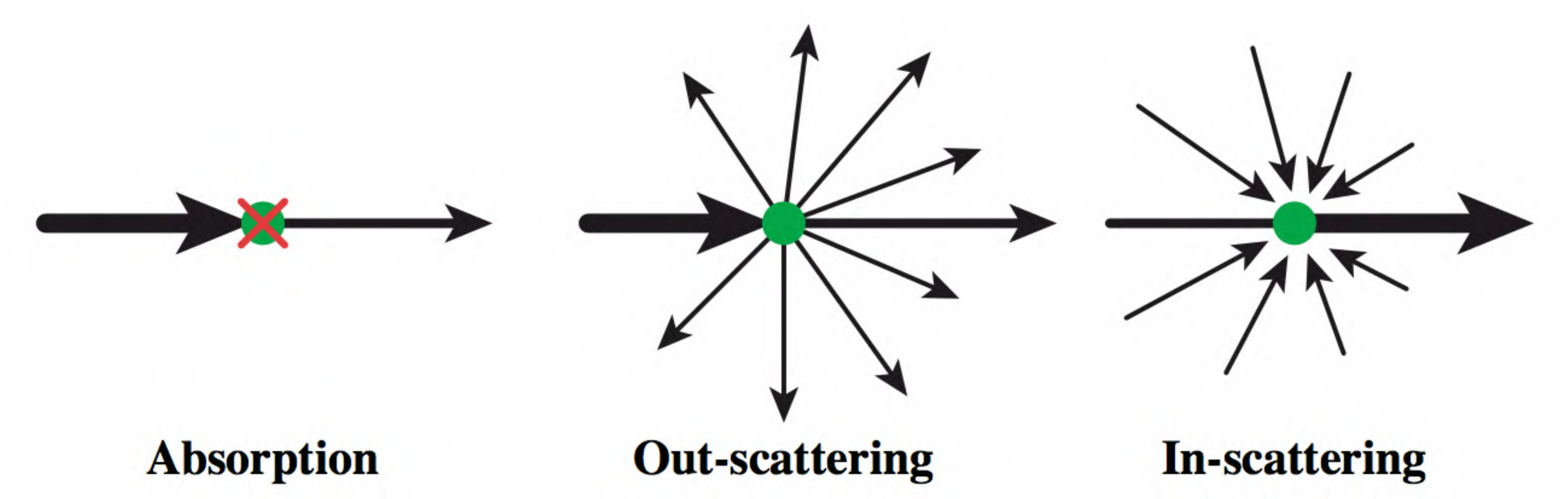
[Novák et al. 2012]

Participating Media: Cloud

[by thephotographer0]

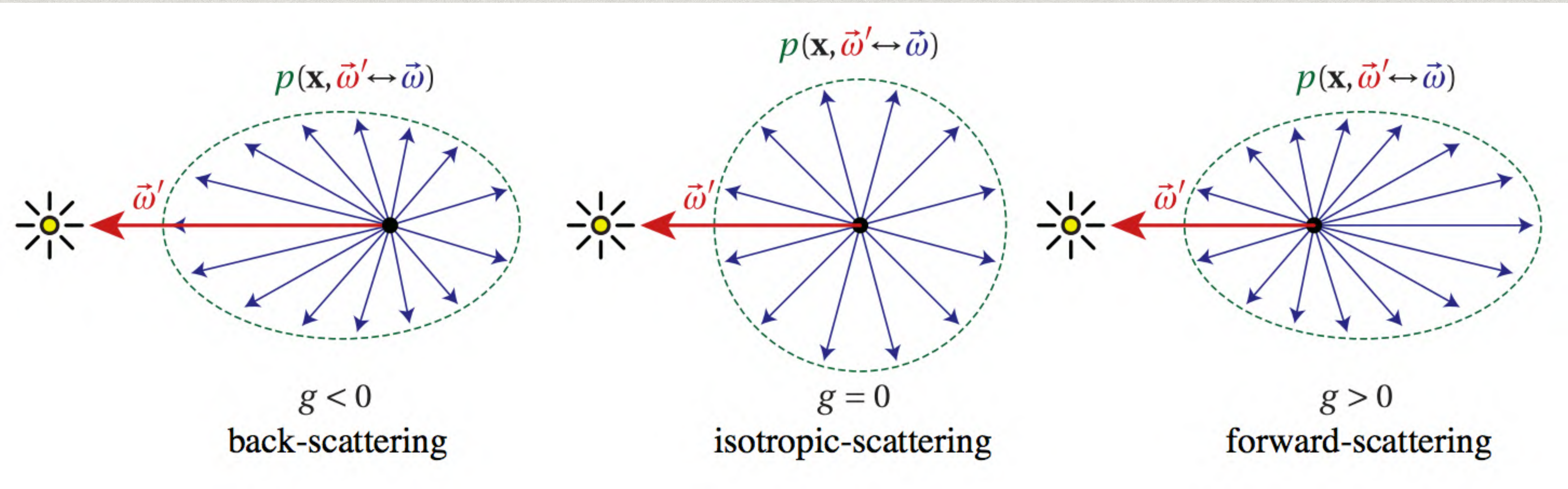
Participating Media

- At any point as light travels through a participating medium, it can be (partially) absorbed and scattered.



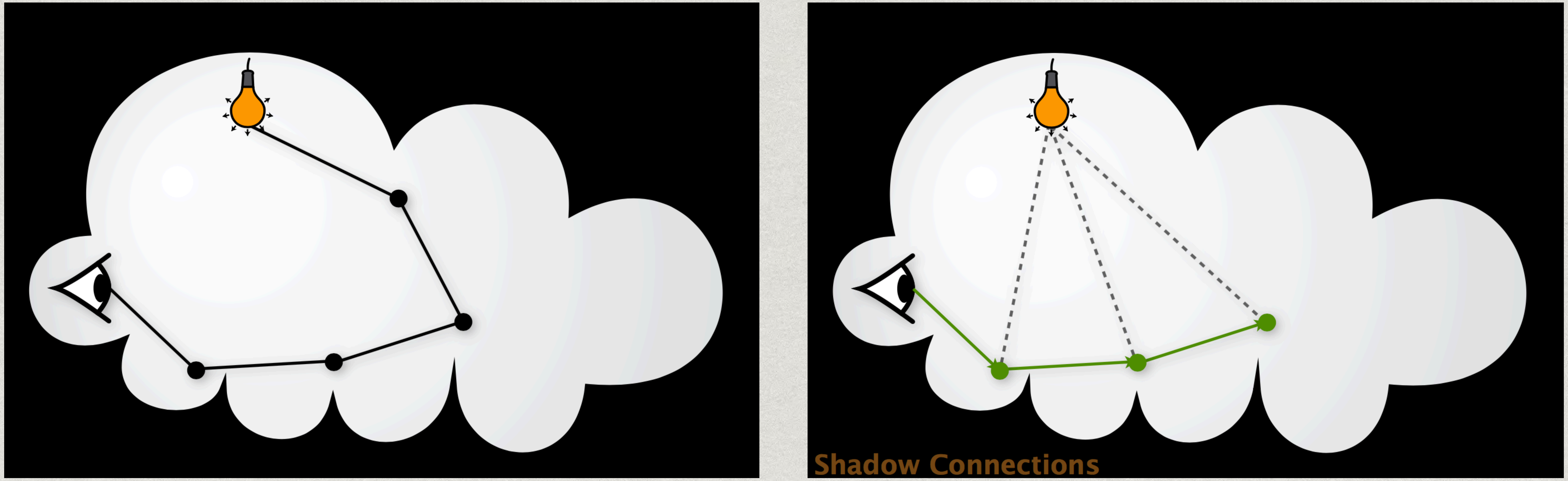
Participating Media

- Use Phase Function to describe the angular distribution of light scattering at any point x within participating media.


$p(\mathbf{x}, \vec{\omega}' \leftrightarrow \vec{\omega})$
$\vec{\omega}'$
$g < 0$
back-scattering
$p(\mathbf{x}, \vec{\omega}' \leftrightarrow \vec{\omega})$
$\vec{\omega}'$
$g = 0$
isotropic-scattering
$p(\mathbf{x}, \vec{\omega}' \leftrightarrow \vec{\omega})$
$\vec{\omega}'$
$g > 0$
forward-scattering

Participating Media: Rendering

- Randomly choose a direction to bounce
- Randomly choose a distance to go straight
- At each ‘shading point’, connect to the light



[Derek Nowrouzezahrai]

Participating Media: Application

[Big Hero 6, 2014 Disney]

Participating Media: Application

[Assassin's Creed Syndicate. 2015 Ubisoft]

Participating Media: Demo

[Stomakhin et al. 2014]

Translucent Material

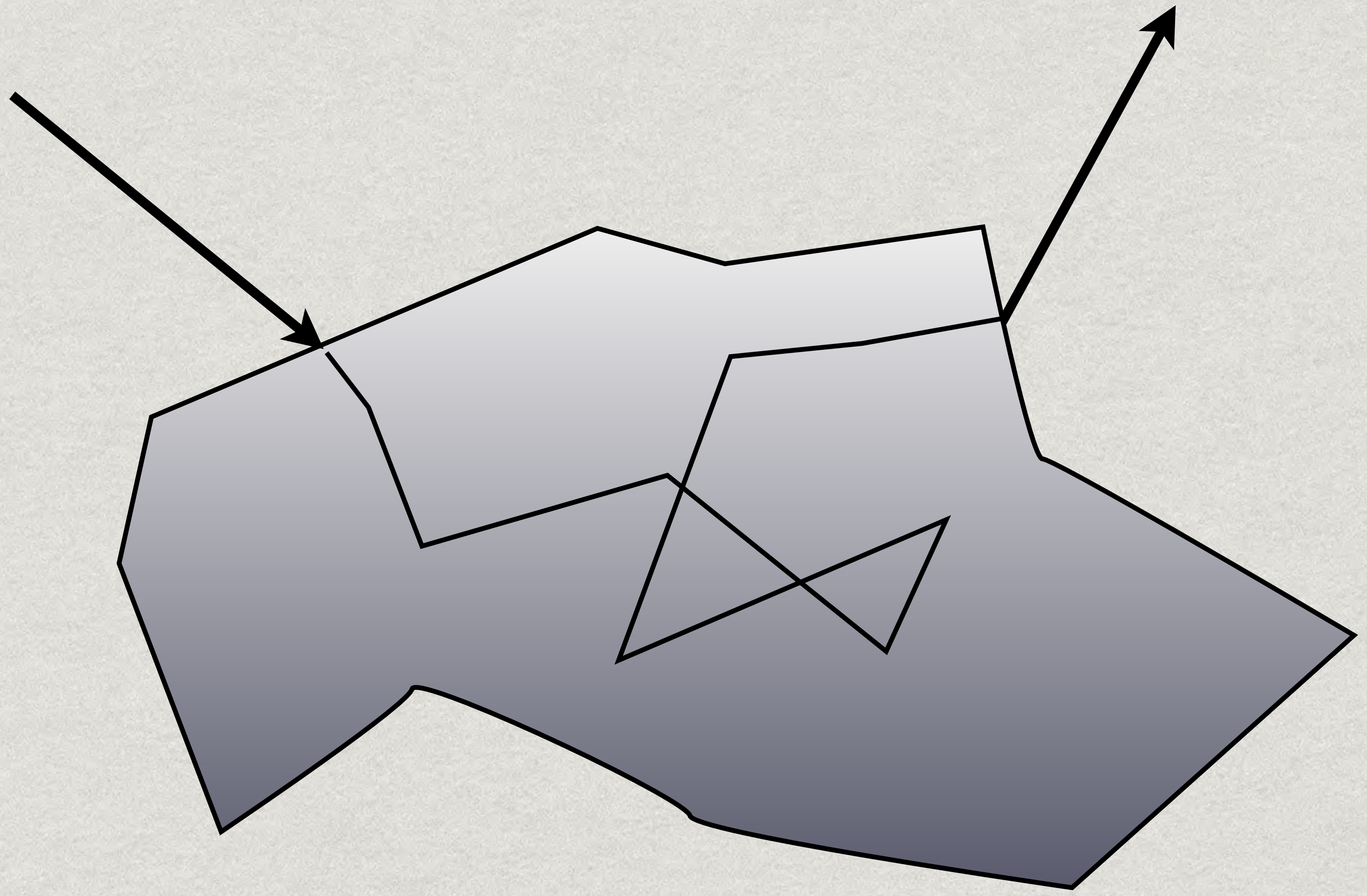
(a specific participating medium)

Translucent Material: Jade

Translucent Material: Jellyfish

Subsurface Scattering

- Visual characteristics of many surfaces caused by light exiting at different points than it enters
  - Violates a fundamental assumption of the BRDF



  - Different from transparent

[Jensen et al 2001]

[Donner et al 2008]

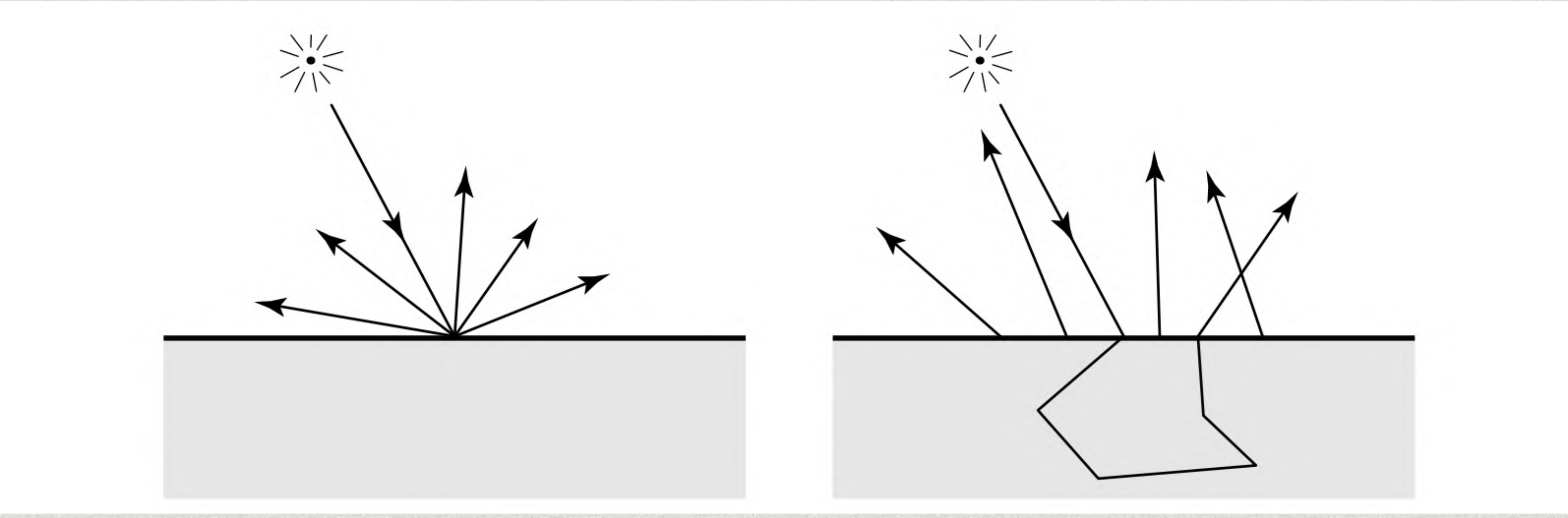
Scattering Functions

- BSSRDF: generalization of BRDF; exitant radiance at one point due to incident differential irradiance at another point:

$$S(x_i, \omega_i, x_o, \omega_o)$$

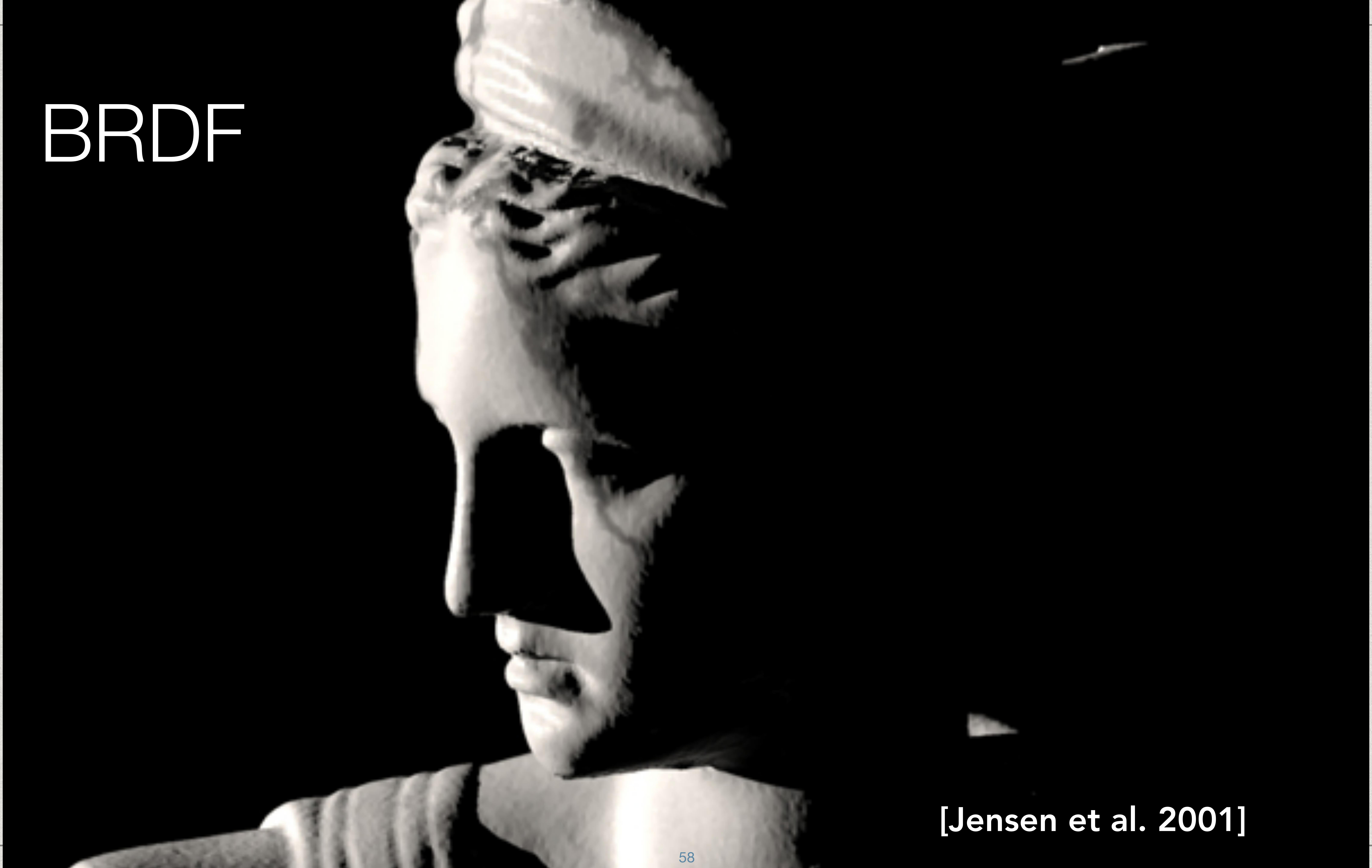
- Generalization of rendering equation: integrating over all points on the surface and all directions (!)

$$L(x_o, \omega_o) = \int_A \int_{H^2} S(x_i, \omega_i, x_o, \omega_o)\, L_i(x_i, \omega_i) \cos\theta_i \,\mathrm{d}\omega_i \,\mathrm{d}A$$



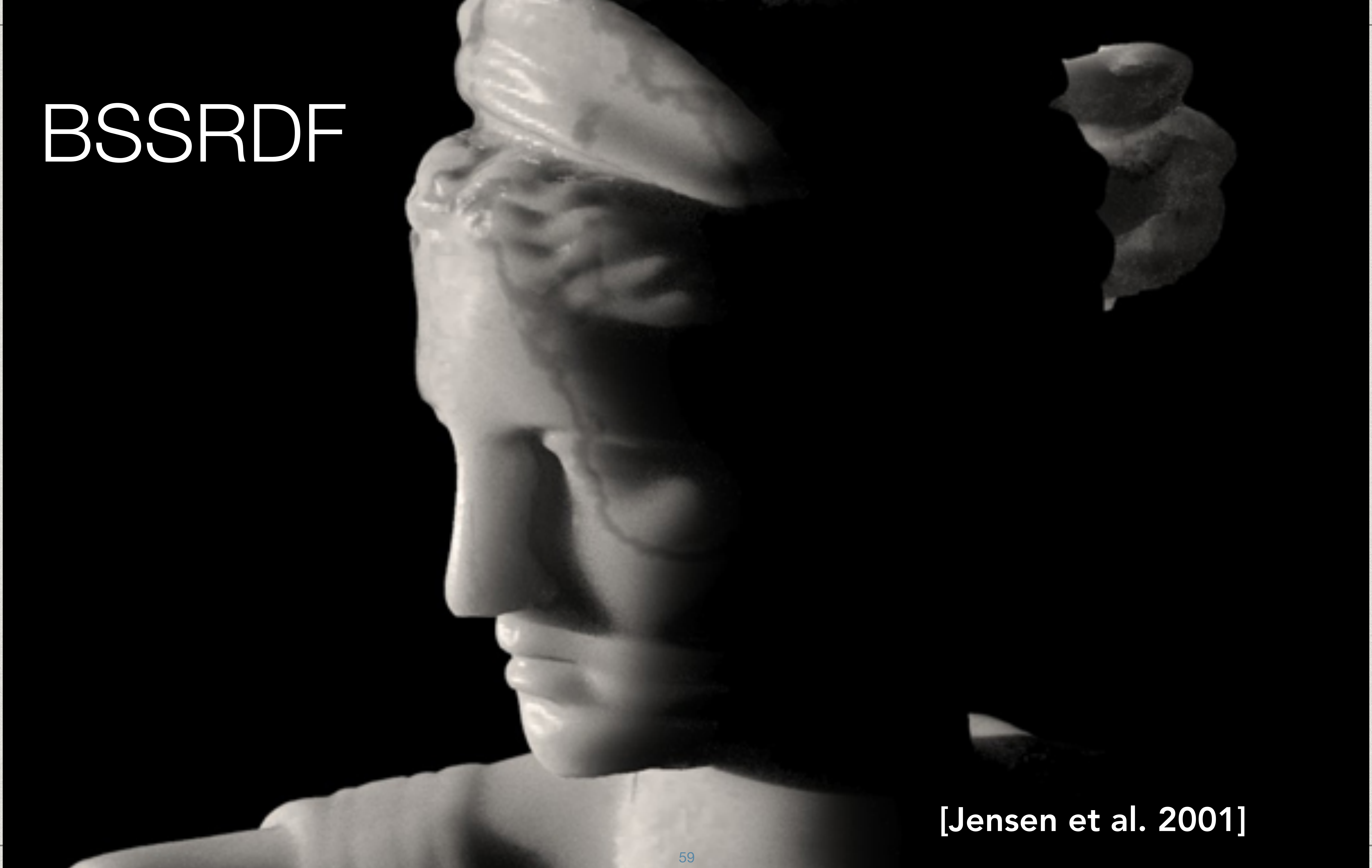
[Jensen et al. 2001]

BRDF



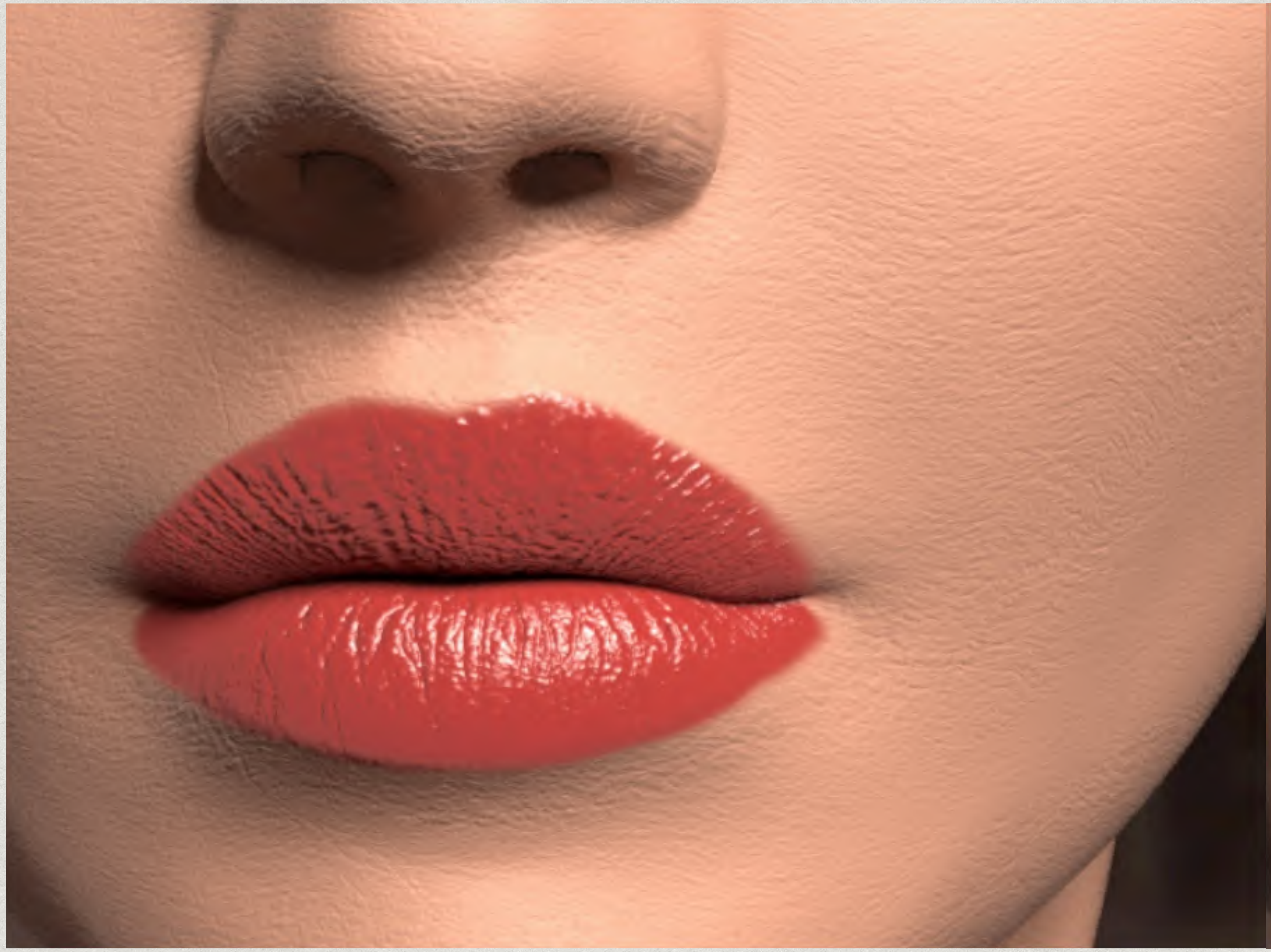
[Jensen et al. 2001]

BSSRDF

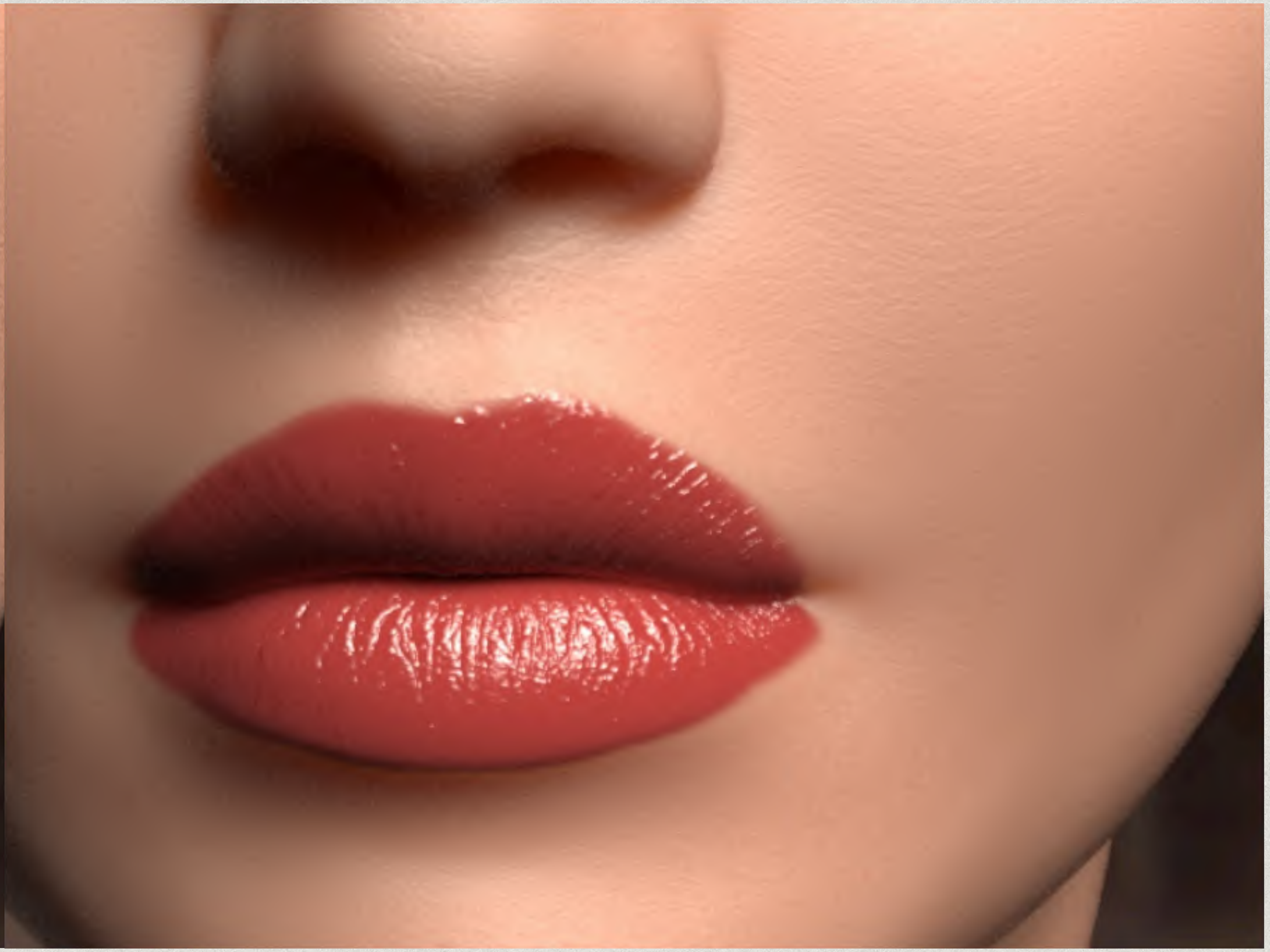


[Jensen et al. 2001]

BRDF vs BSSRDF



BRDF

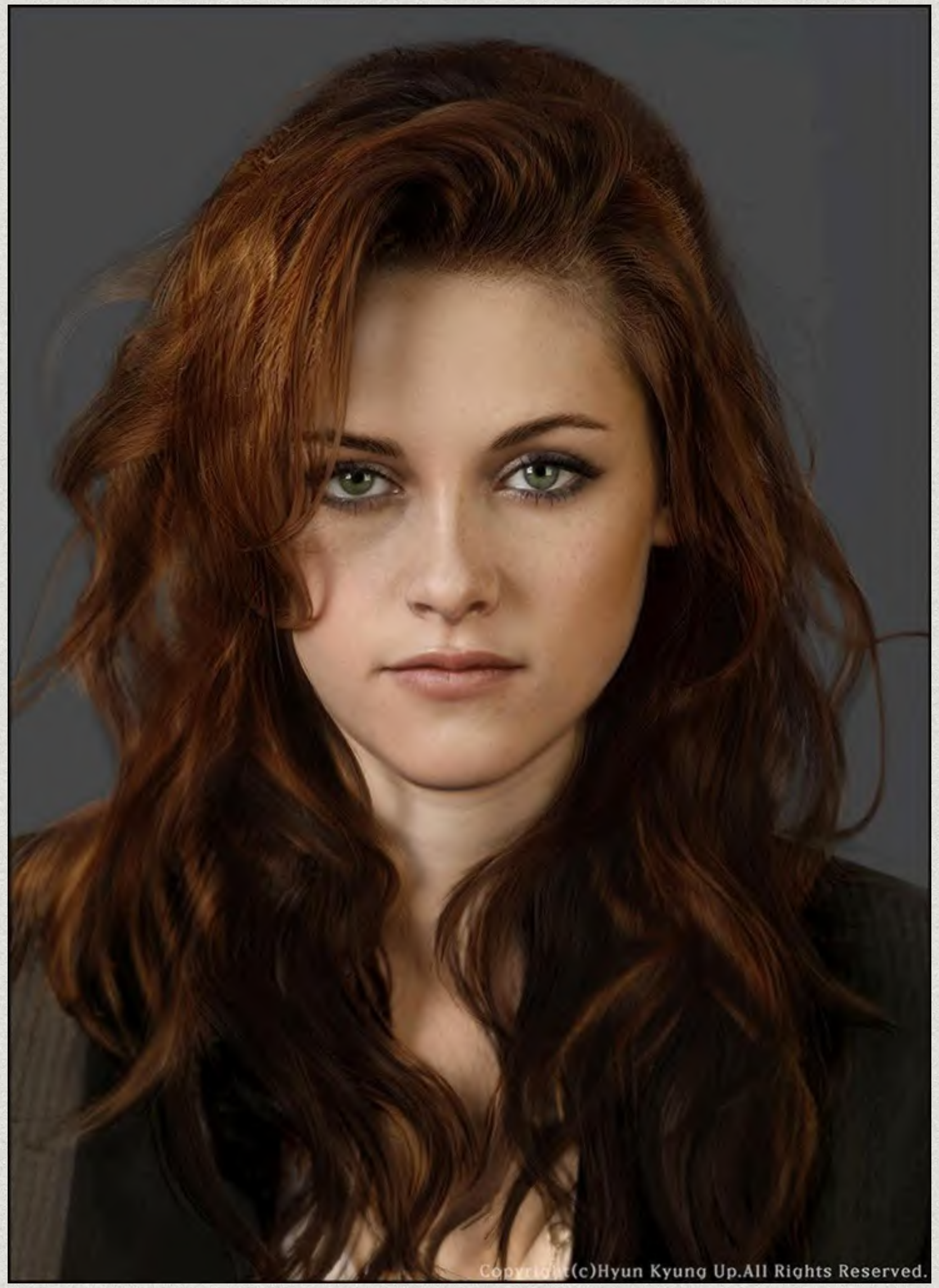


BSSRDF

[Jensen et al. 2001]

BSSRDF: Application

[Artist: Teruyuki and Yuka]



[Artist: Hyun Kyung]

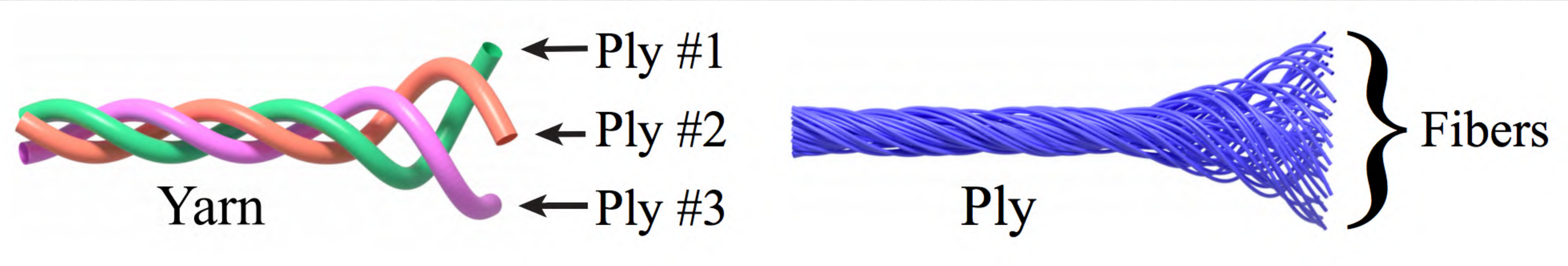
[Artist: Dan Roarty]

https://cgelves.com/10-most-realistic-human-3d-models-that-will-wow-you/

Cloth

Cloth

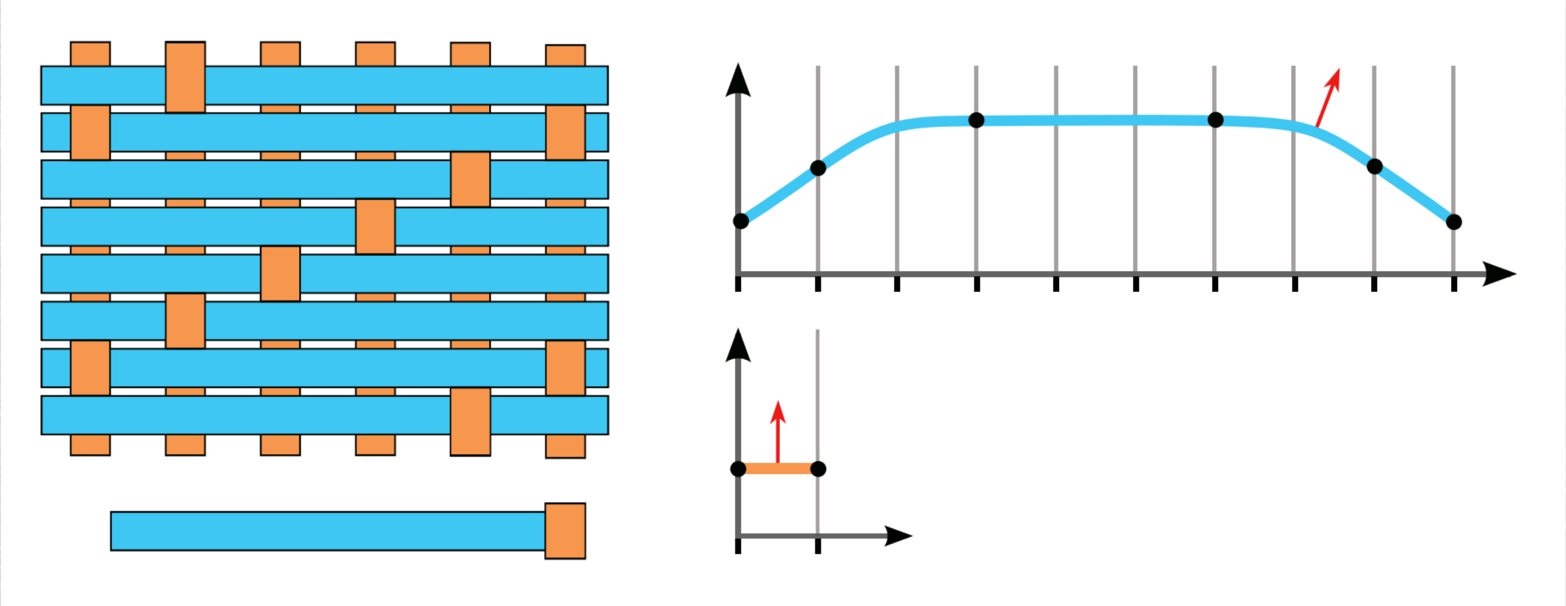
- A collection of twisted fibers!
- Two levels of twist

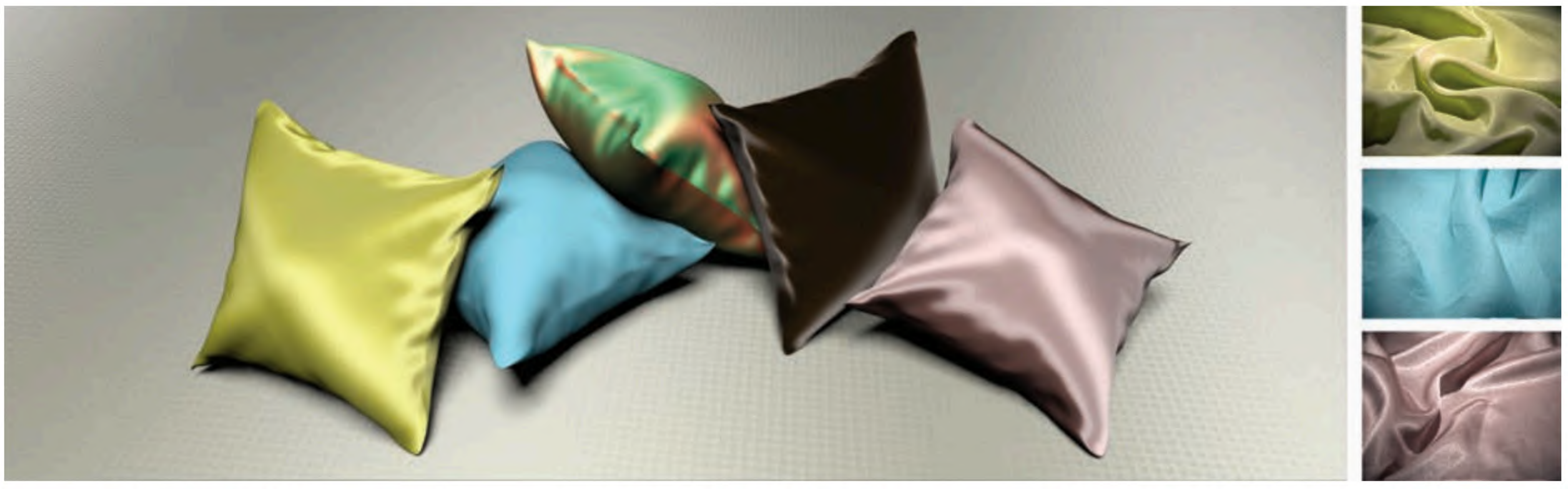


- Woven or knitted

Cloth: Render as Surface

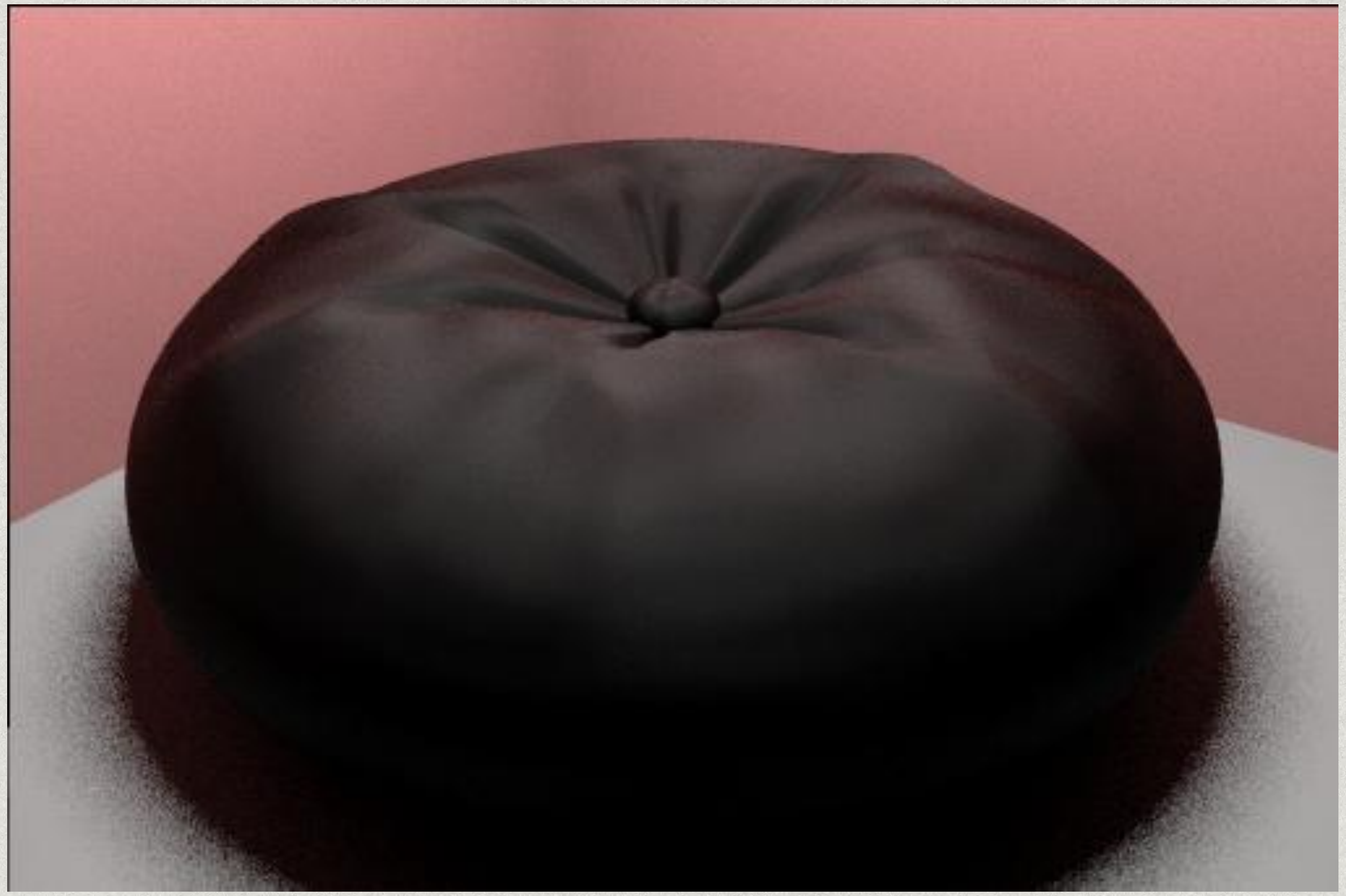
- Given the weaving pattern, calculate the overall behavior
- Render using a BRDF

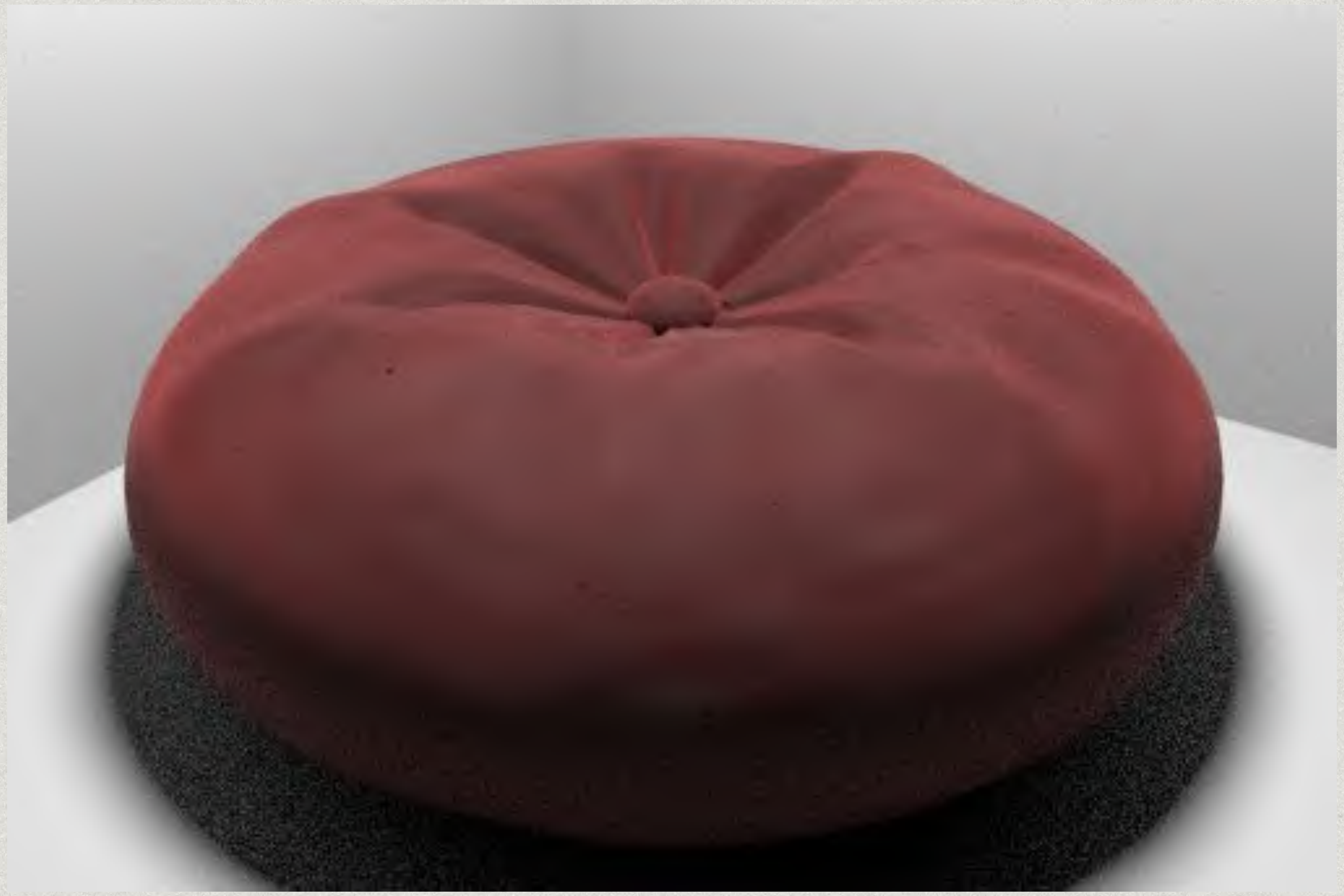


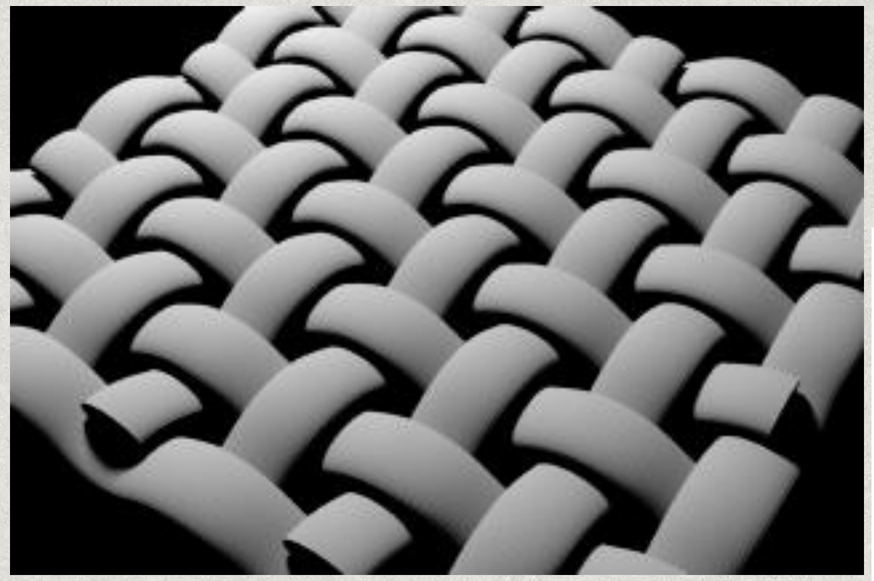


[Sadeghi et al. 2013]

Render as Surface — Limitation



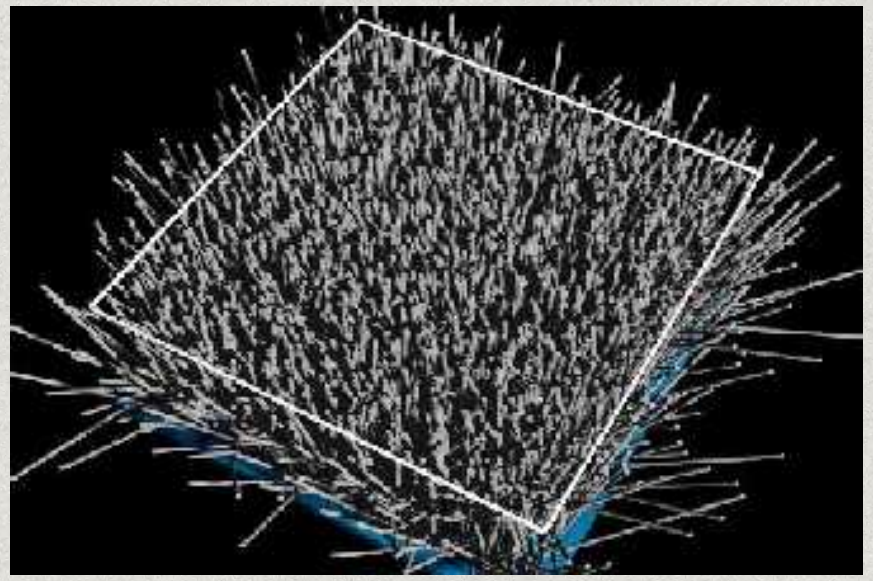




✓

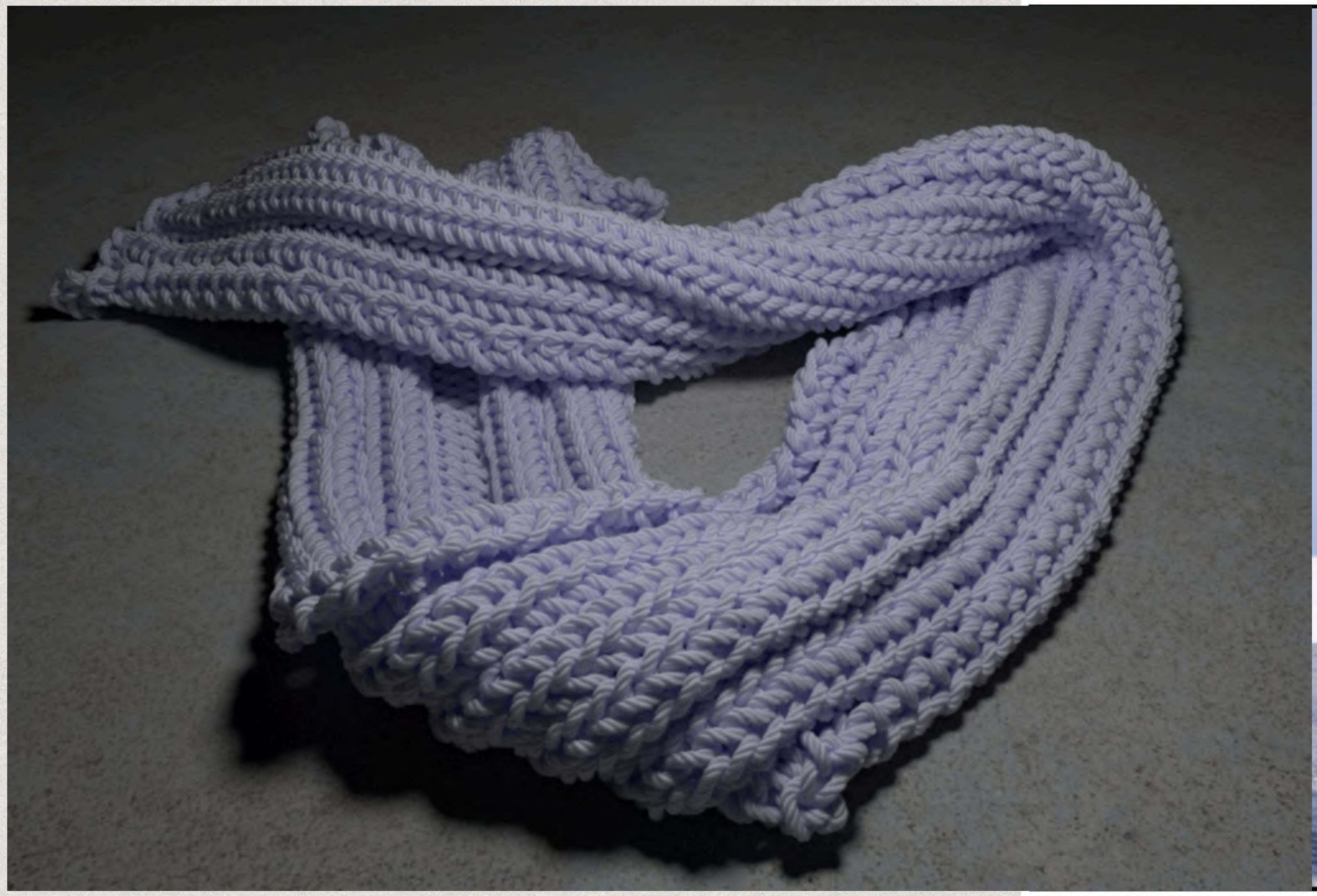
[Westin et al. 1992]

✗

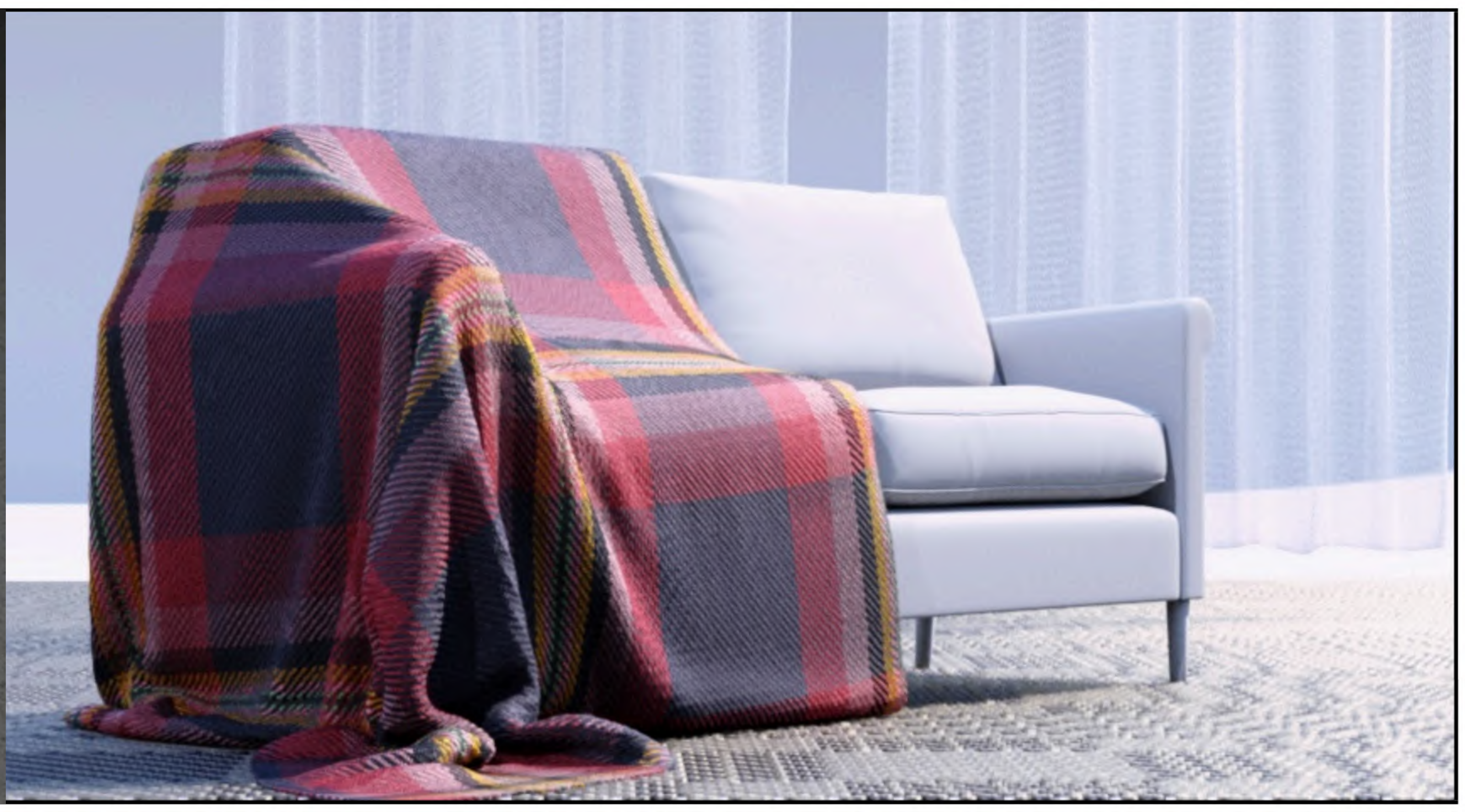


Cloth: Render as Participating Media

- Properties of individual fibers & their distribution -> scattering parameters
- Render as a participating medium



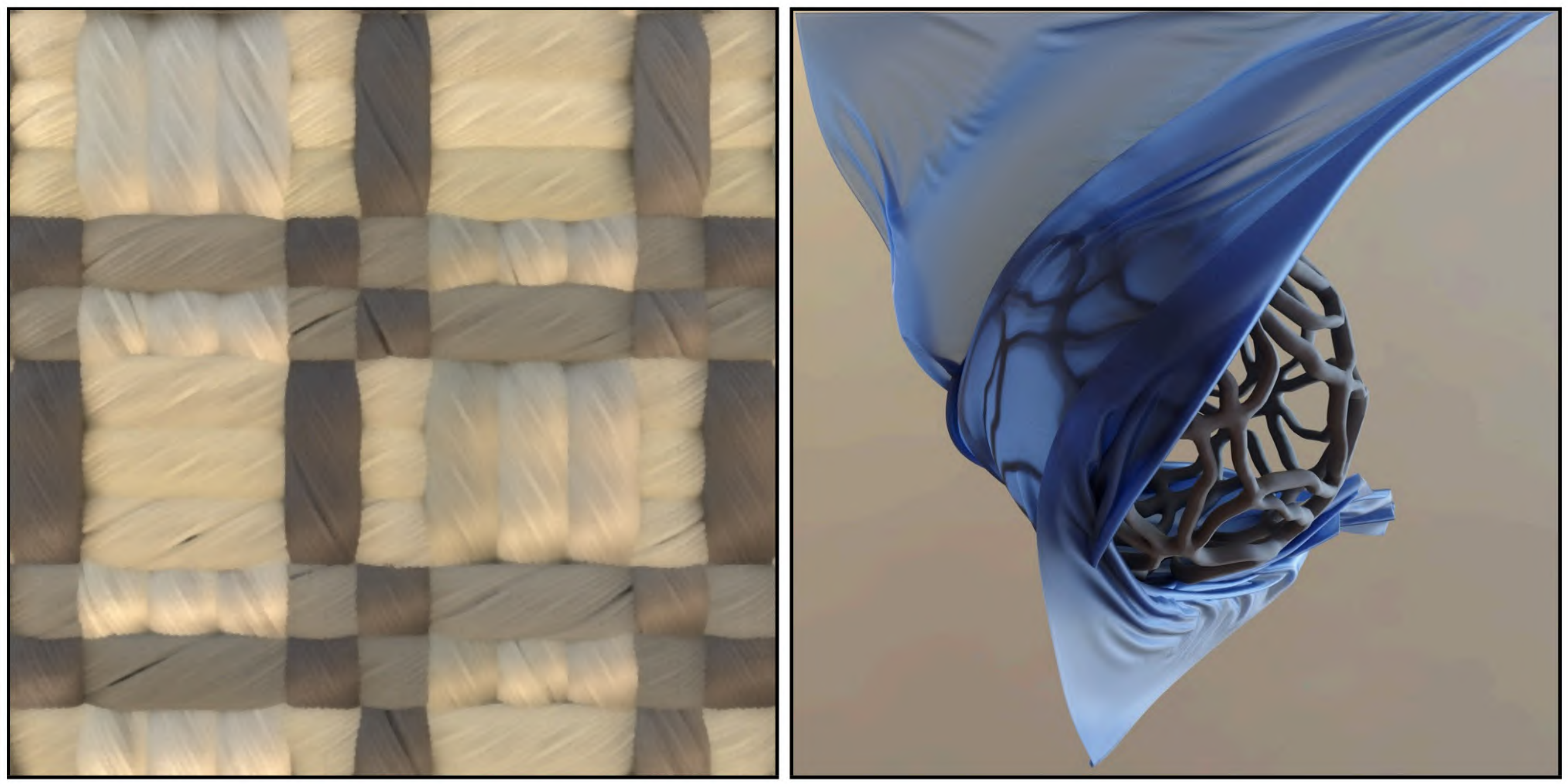
[Jakob et al. 2010]



[Schroder et al. 2011]

Cloth: Render as Actual Fibers

- Render every fiber explicitly!



[Kai Schroder]

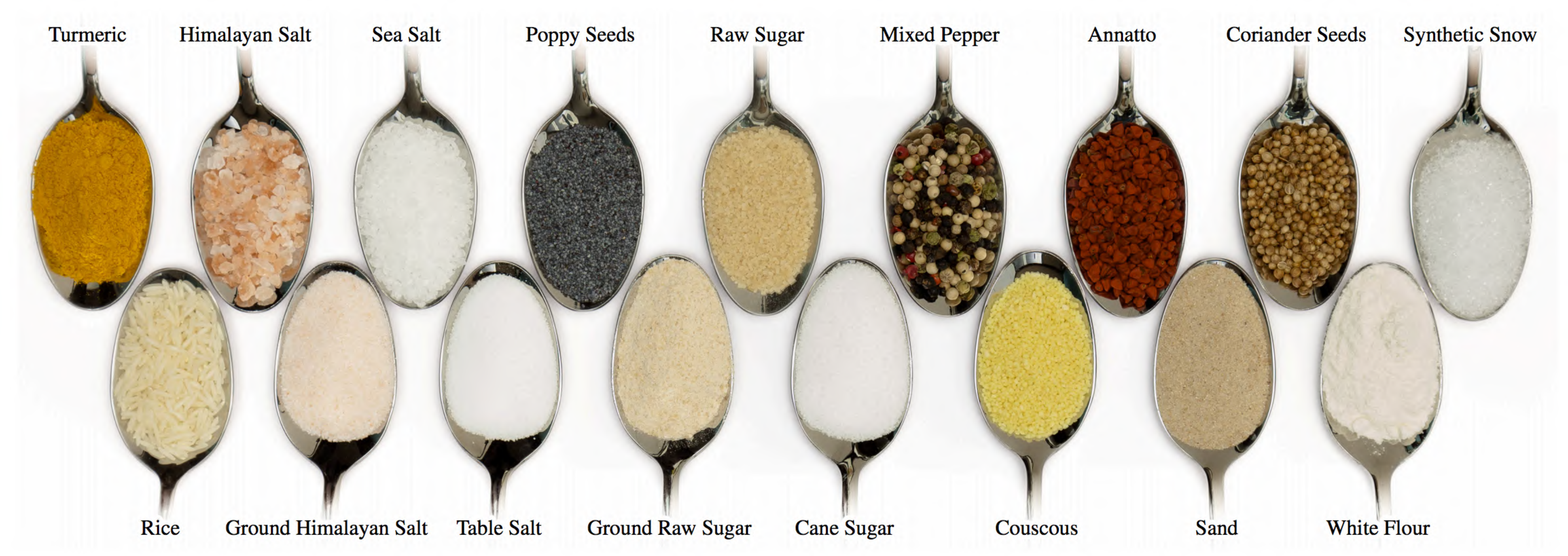
Cloth: Application

[The BFG. 2016 Disney]

Granular Material

Granular Material

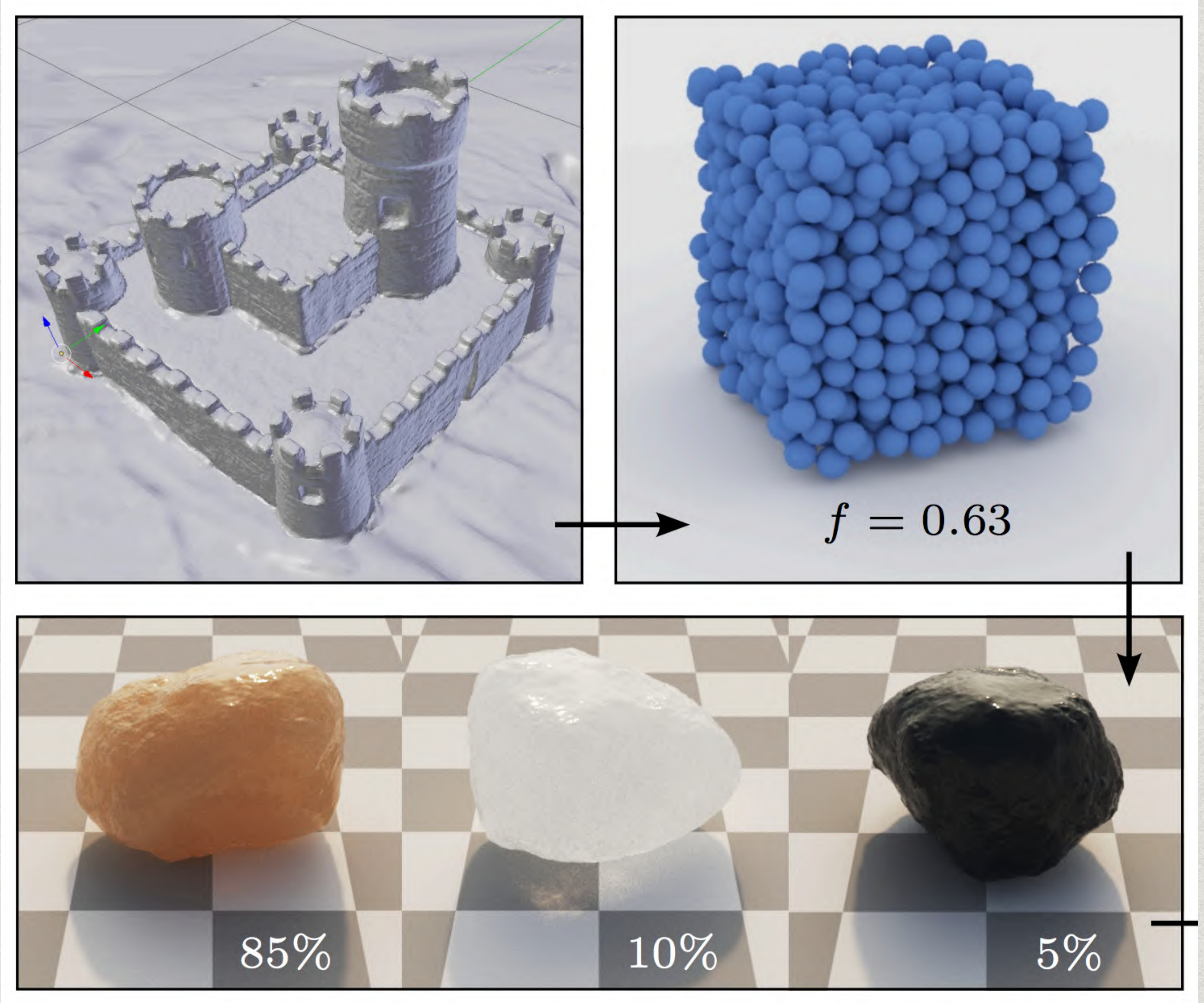
- What is granular material?

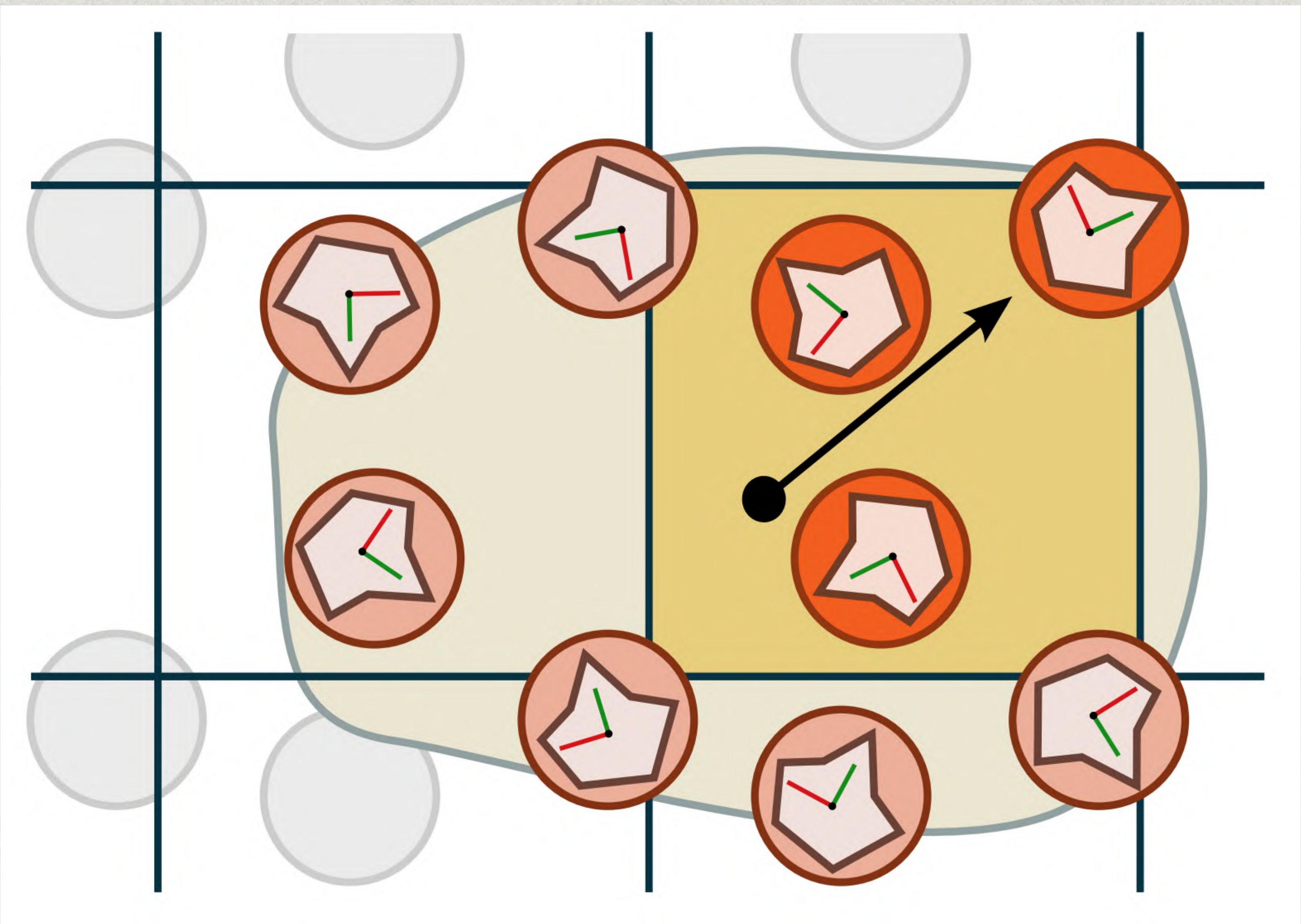


[Meng et al. 2015]

Granular Material

- Can we avoid explicit modeling of all granules?
  - Yes with procedural definition.





[Meng et al. 2015]

Granular Material

[Meng et al. 2015]

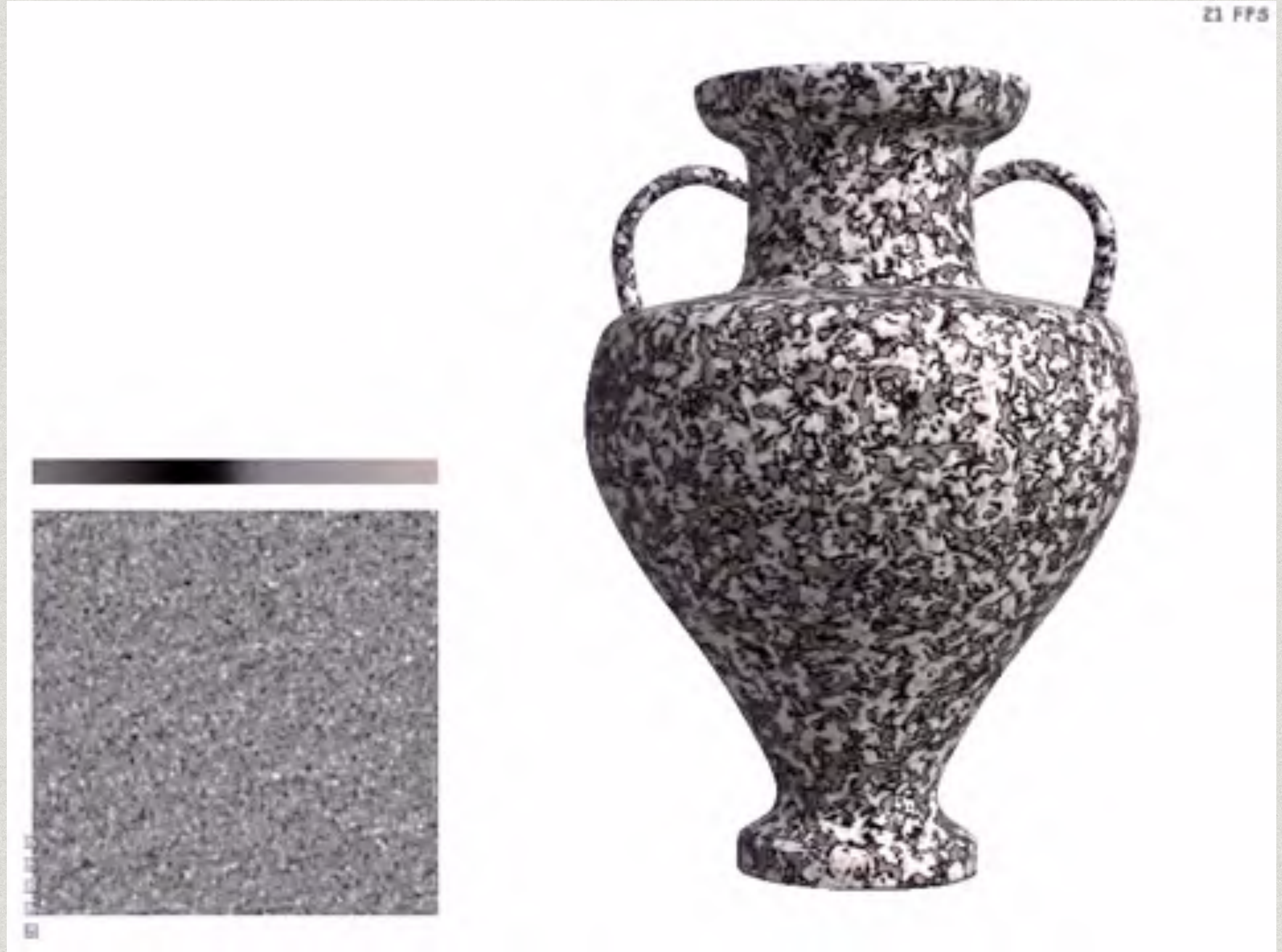
Granular Material: Application

[Piper. 2016 Pixar]

Procedural Material

Procedural Model

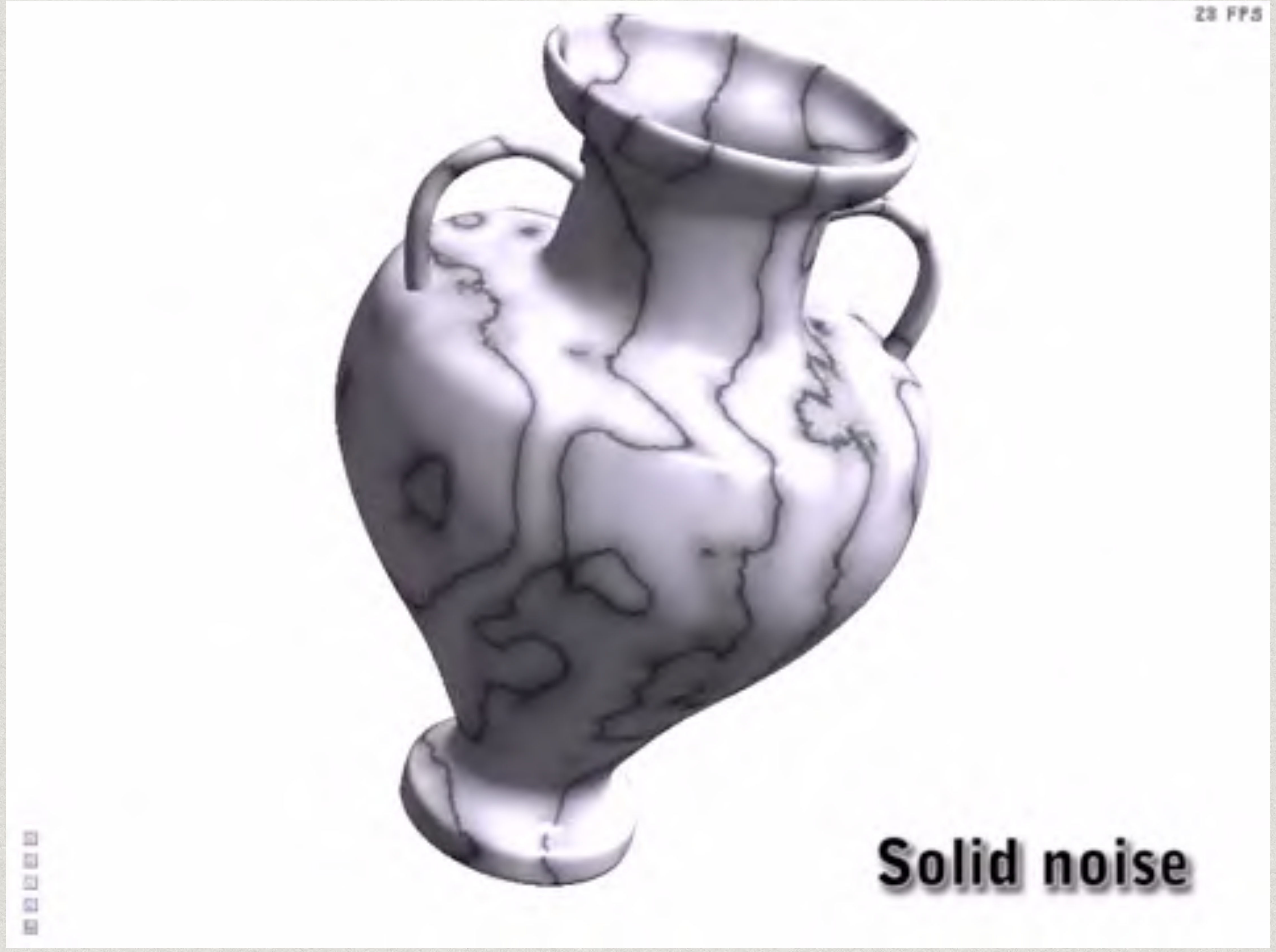
- Can we define details without textures?
  - Yes! Compute a noise function on the fly.



[Lagae et al. 2009]

Procedural Model

- Can we define details without textures?
  - Yes! Compute a noise function on the fly.
  - 3D noise -> internal structure if cut or broken



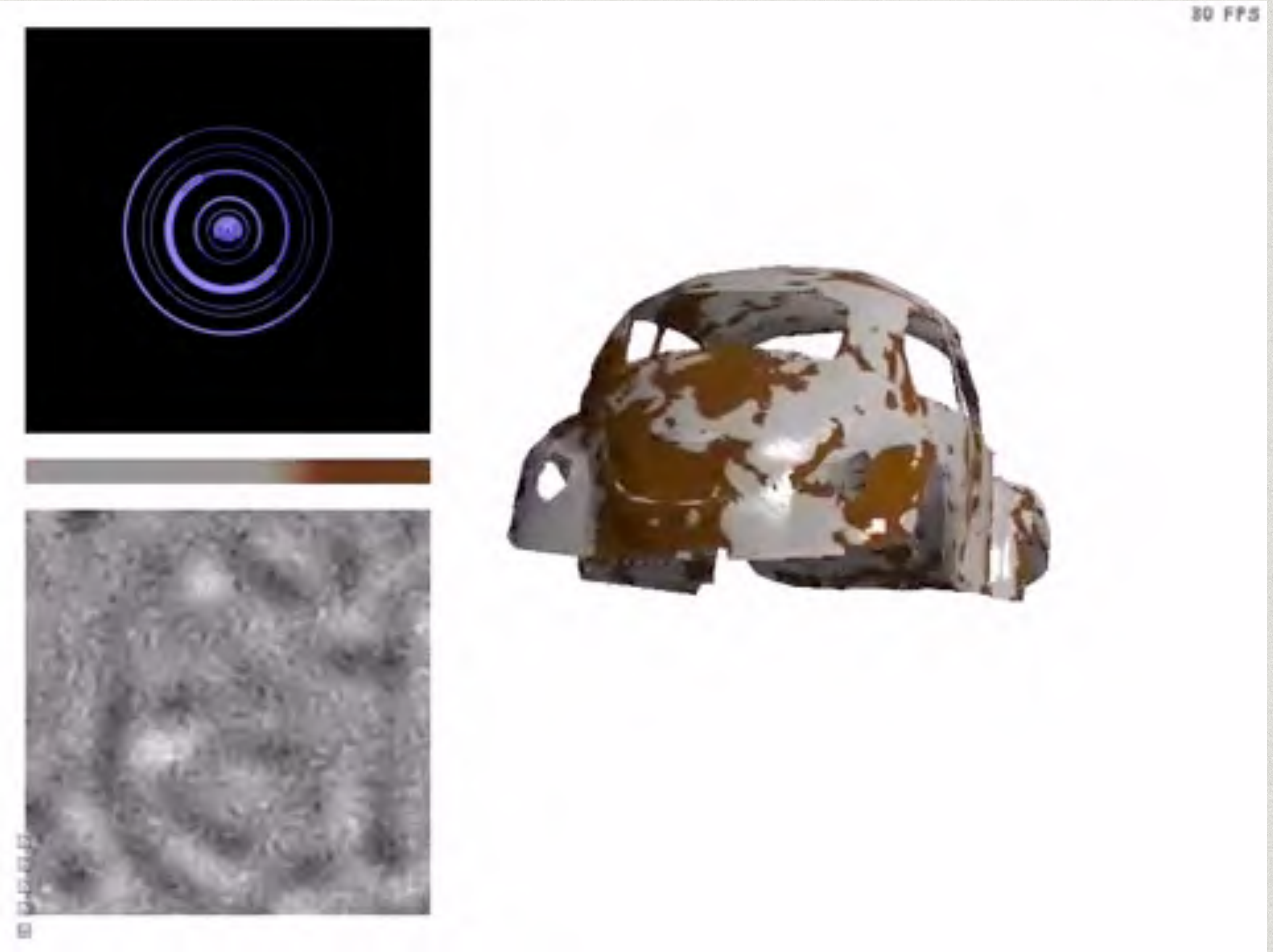
[Lagae et al. 2009]

Procedural Model

- Can we define details without textures?
  - Yes! Compute a noise function on the fly.
  - Thresholding
    (noise -> binary noise)

Example:

```
if noise(x, y, z) > threshold:
    reflectance = 1
else:
    reflectance = 0
```



[Lagae et al. 2009]

Procedural Model

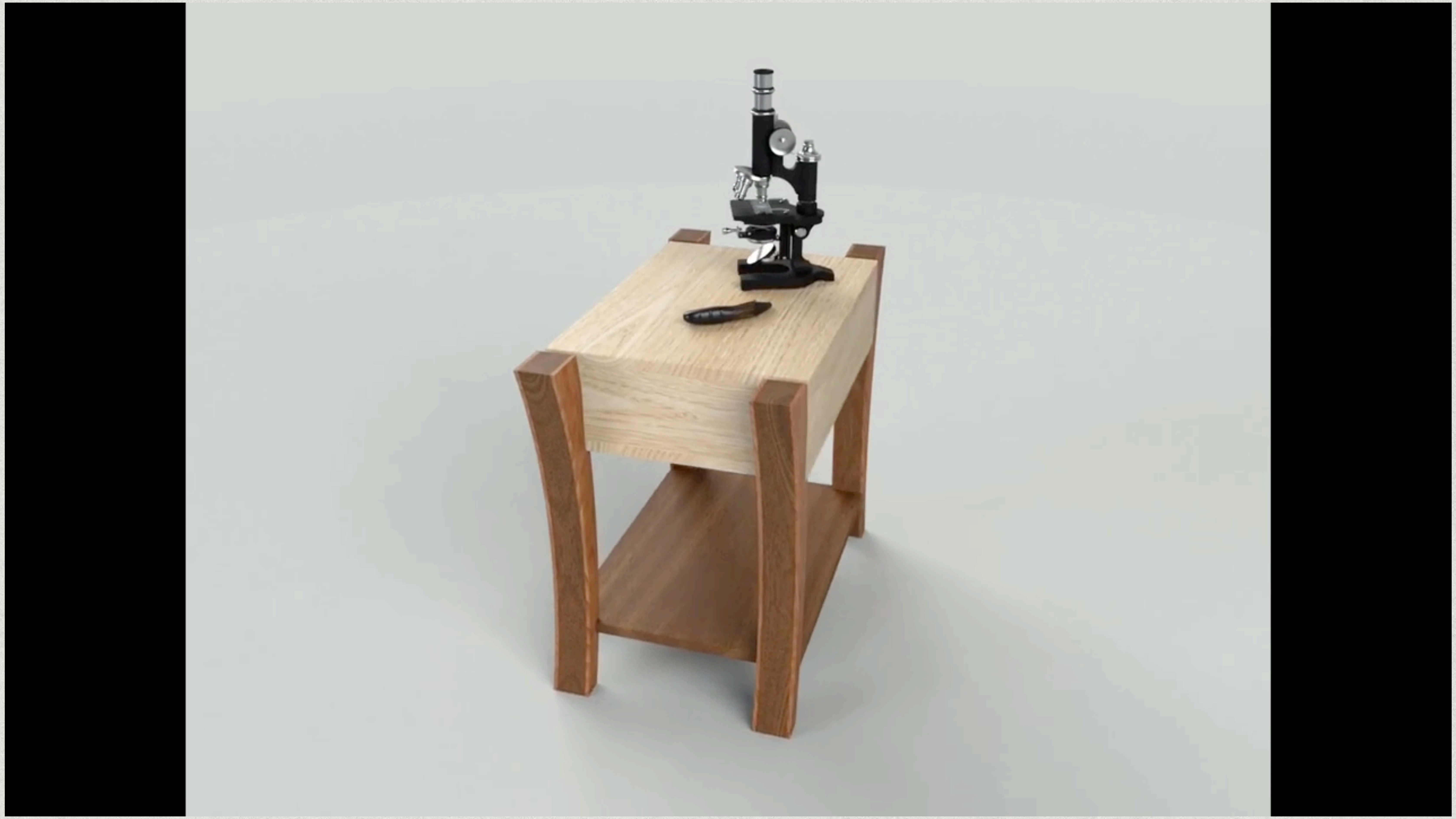
- Complex noise functions can be very powerful.

Procedural Model

- Complex noise functions can be very powerful.

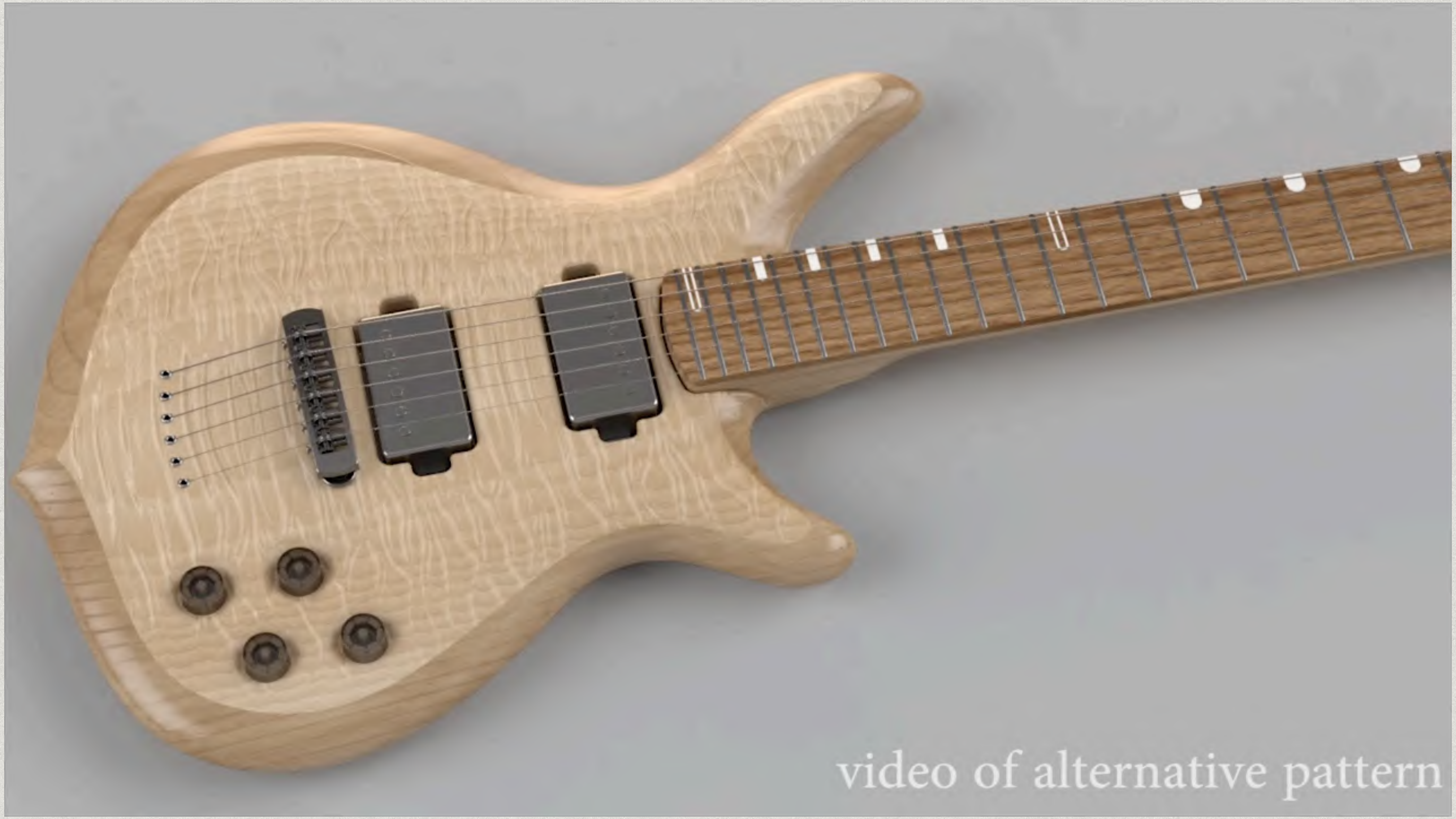
Procedural Model

- Complex noise functions can be very powerful.



Procedural Model

- Complex noise functions can be very powerful.



Beyond State of the Art

- Next generation rendering / my research
  - High level goal: **realism** and **speed**

detailed rendering

appearance modeling

real-time ray tracing

Part I: Detailed Rendering

detailed rendering

appearance modeling

real-time ray tracing

Rendering is powerful today

Car rendered in NVIDIA Iray

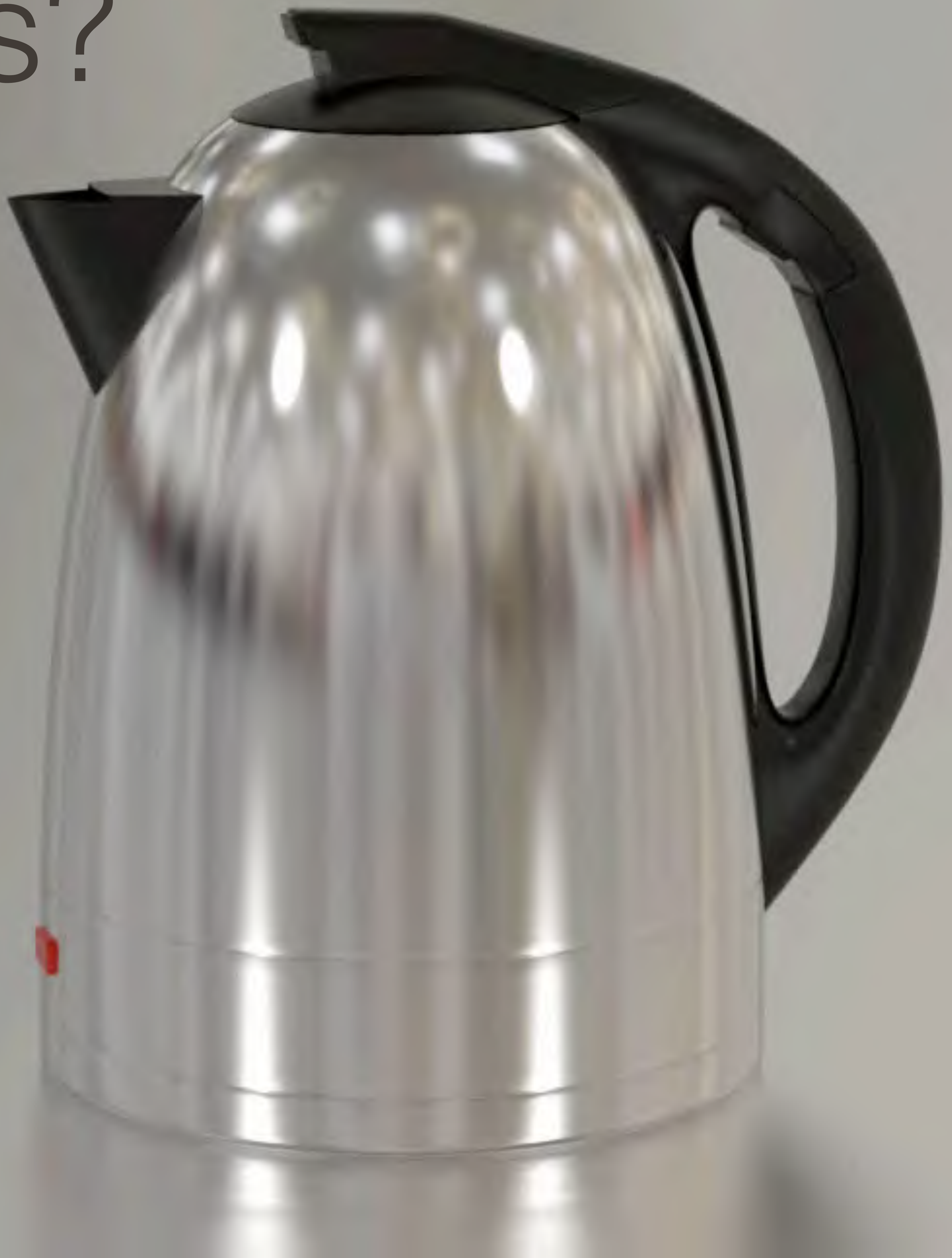
Mouse rendered in
Autodesk 3DS Max

Real world is more complicated

Real photograph of a car

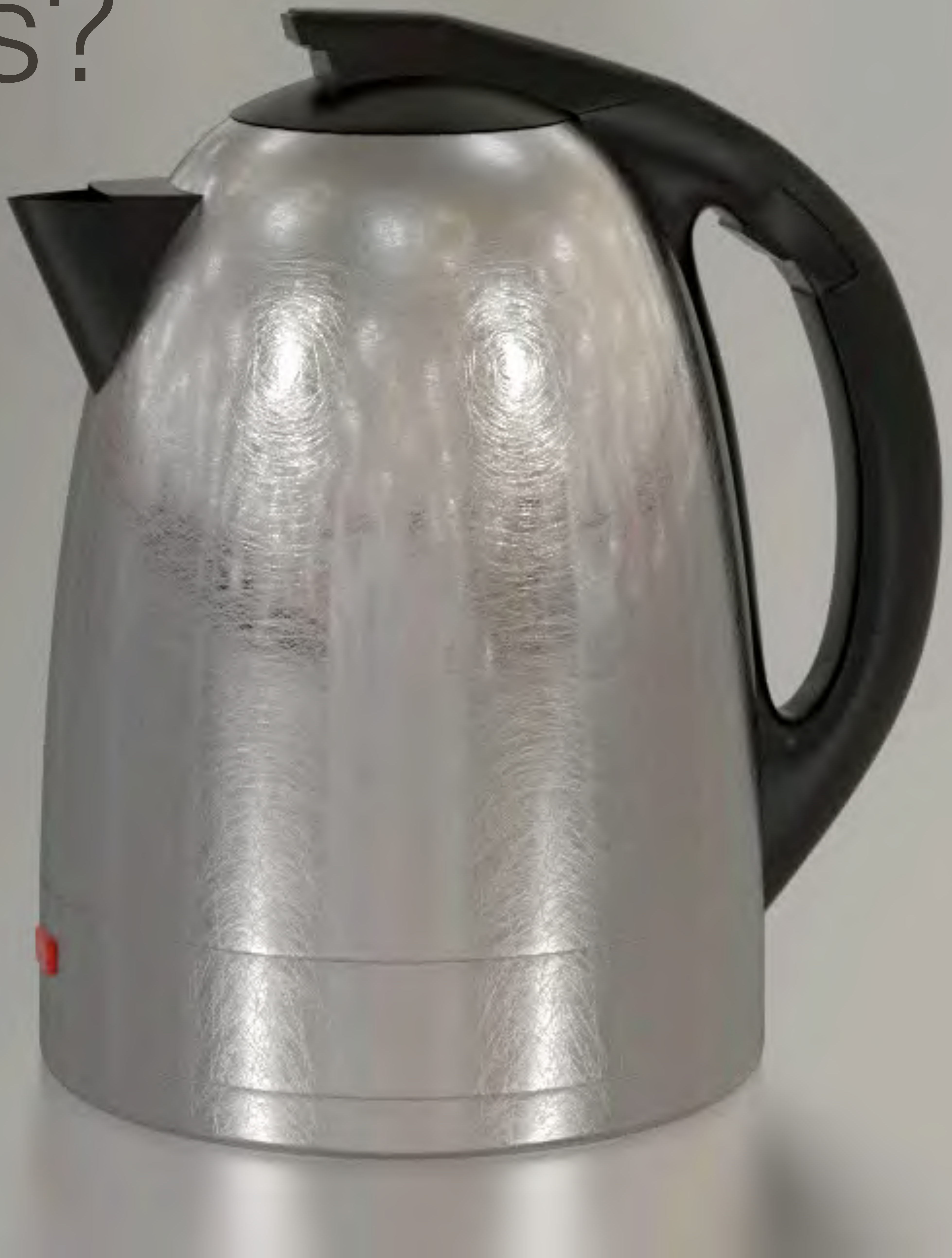
Real video of a mouse

Why details?



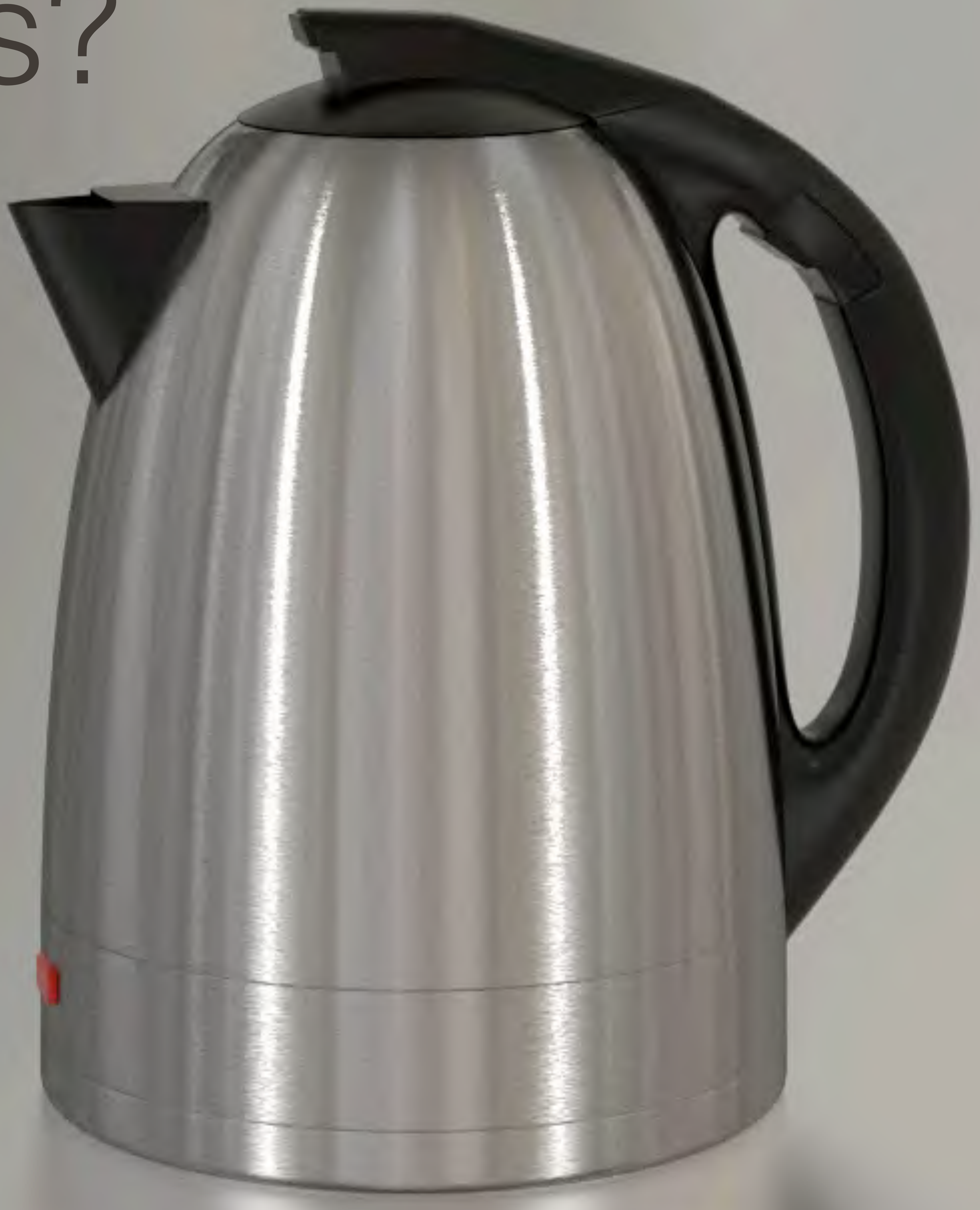
traditional
model

Why details?



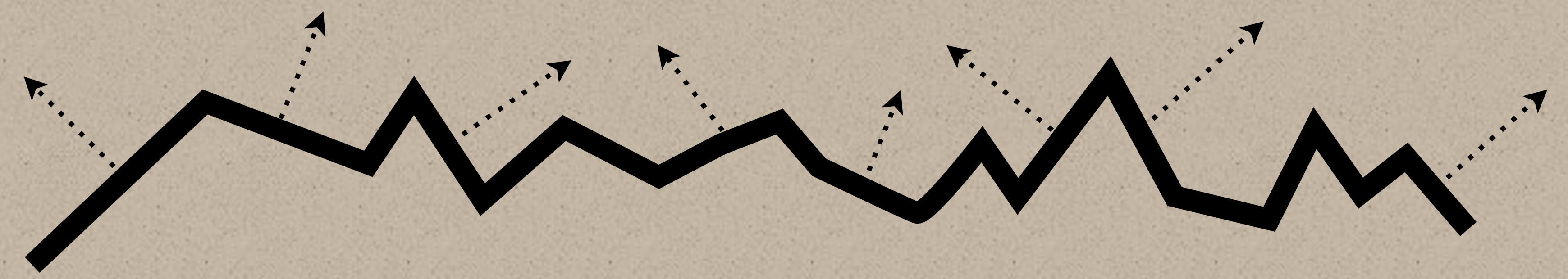
our model

Why details?

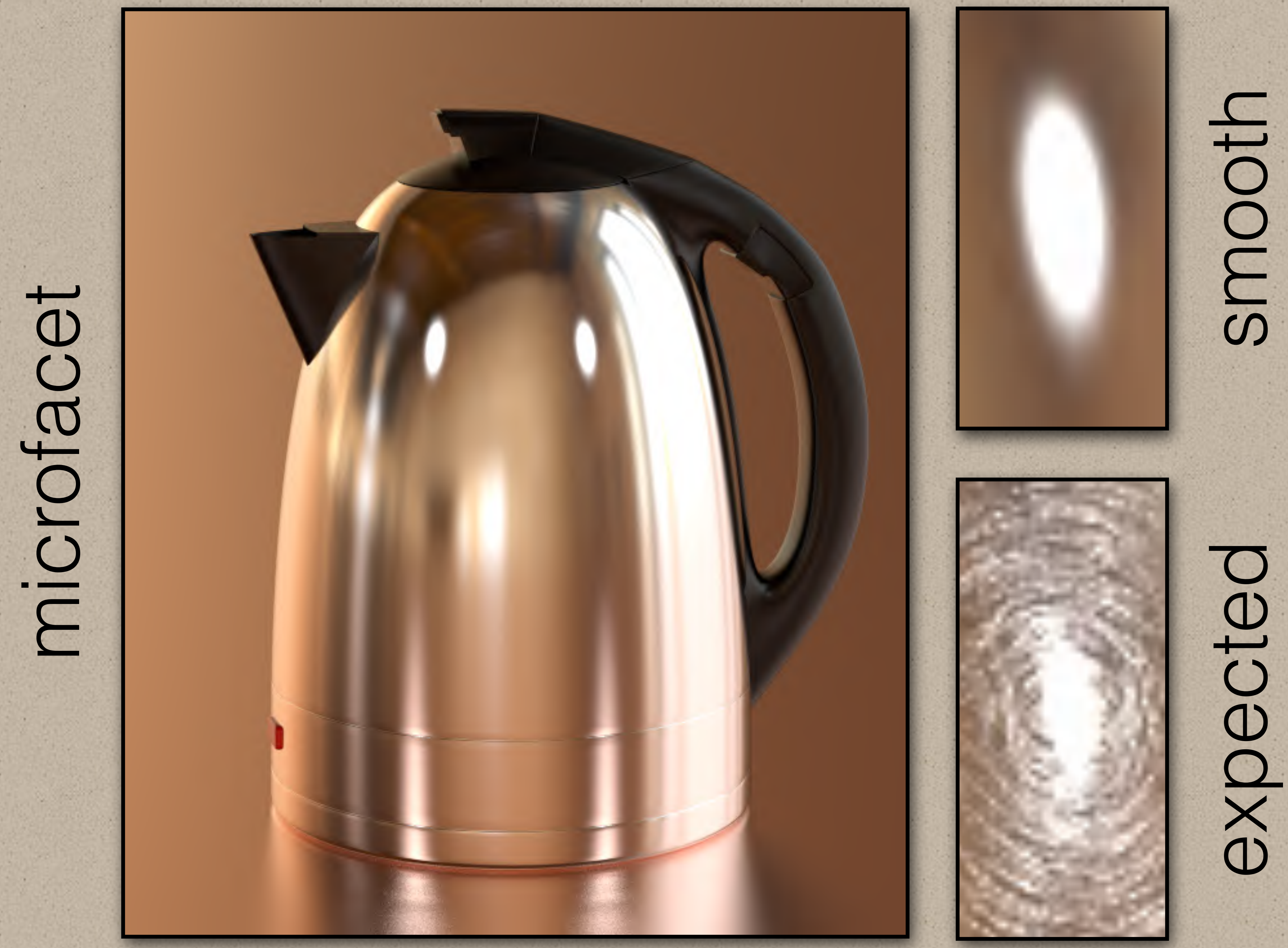


our model

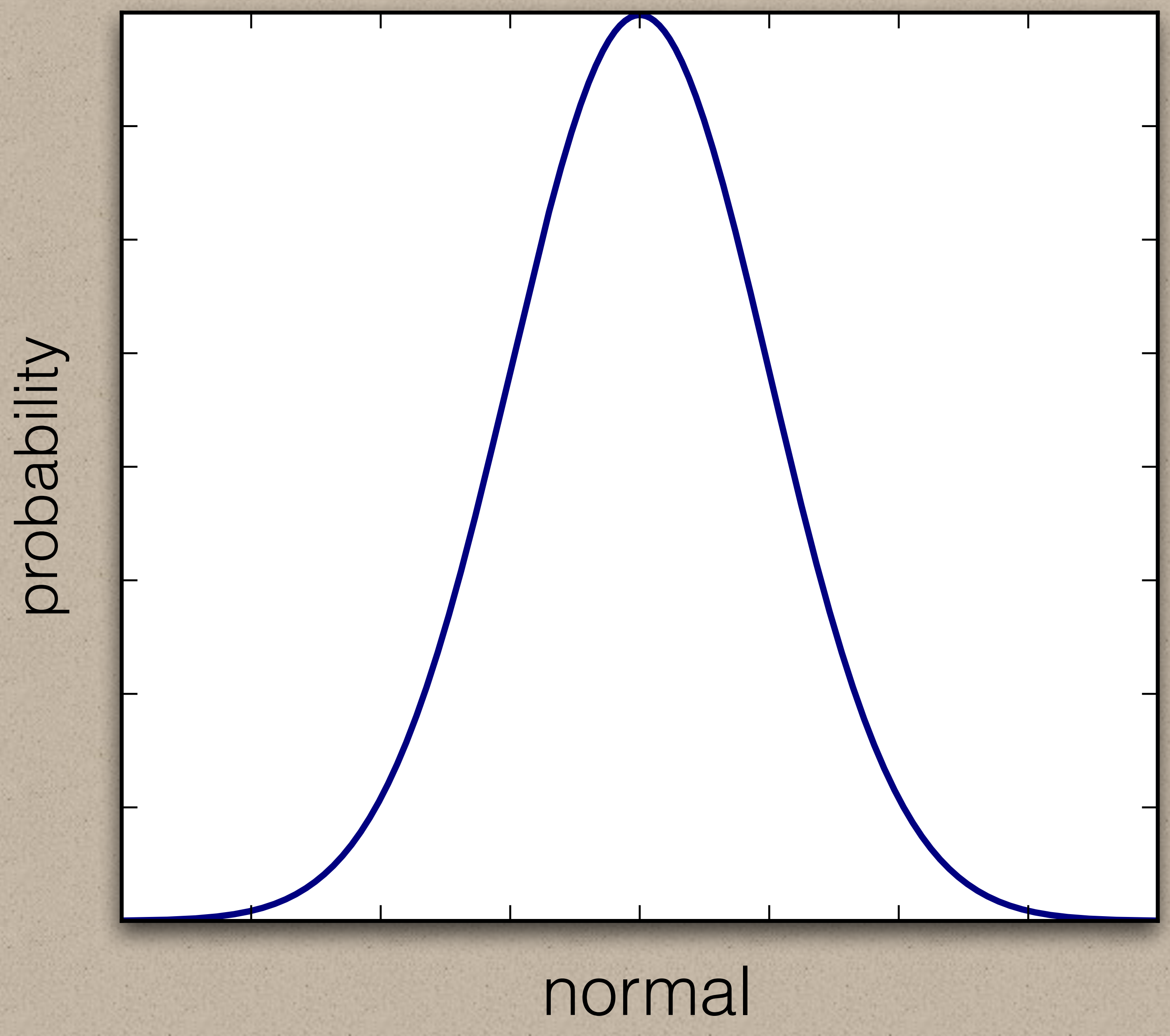
Normal Distribution: ideal vs. real



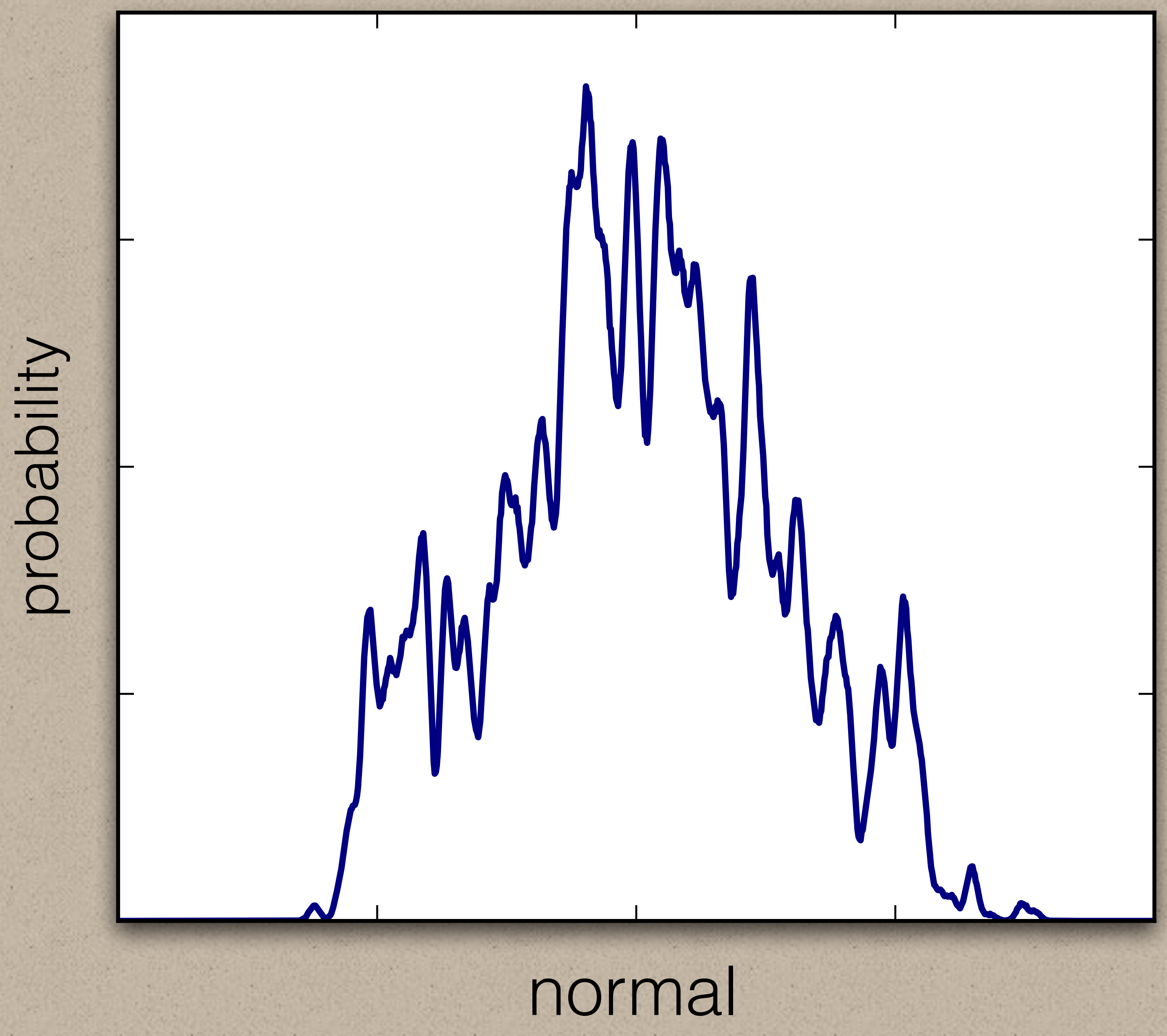
Surface = Specular microfacets +
Random normals



Normal Distribution Function (NDF)

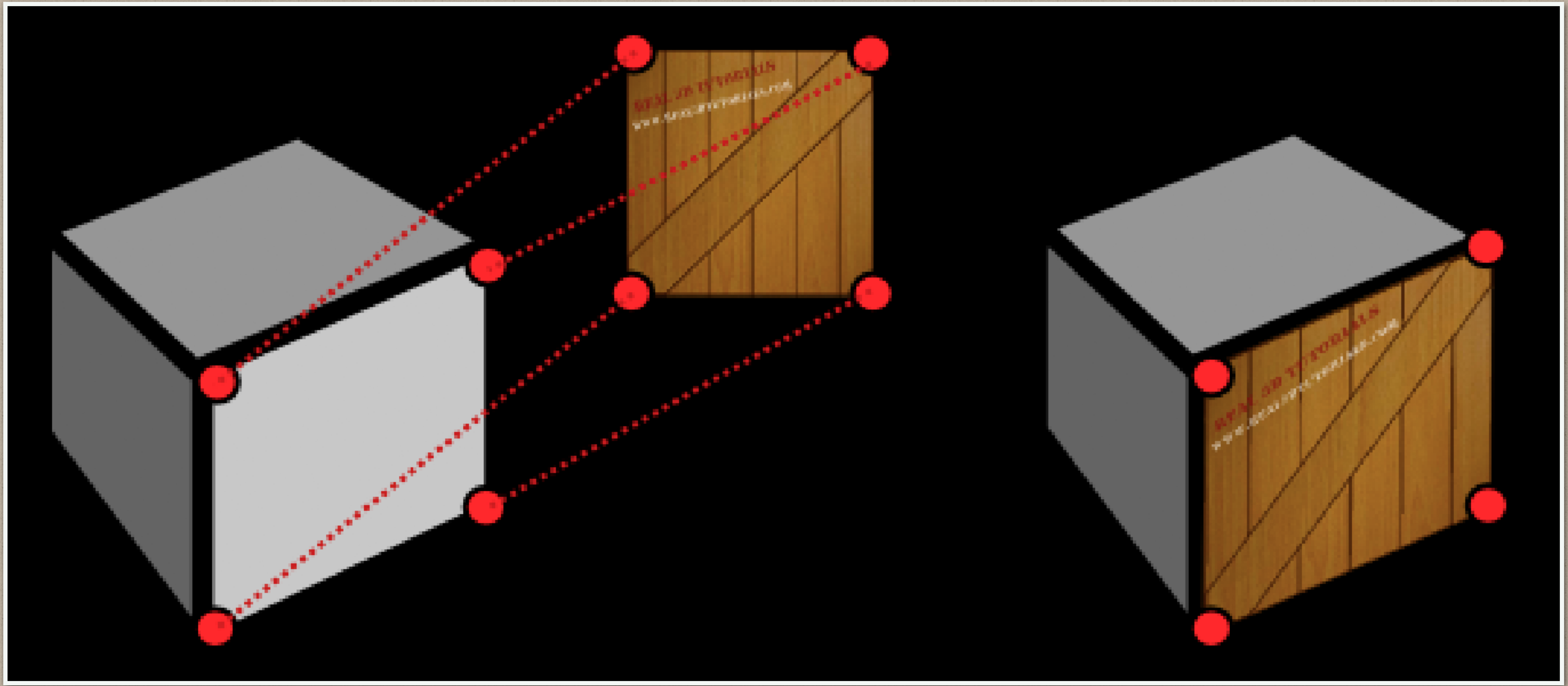


What we have previously

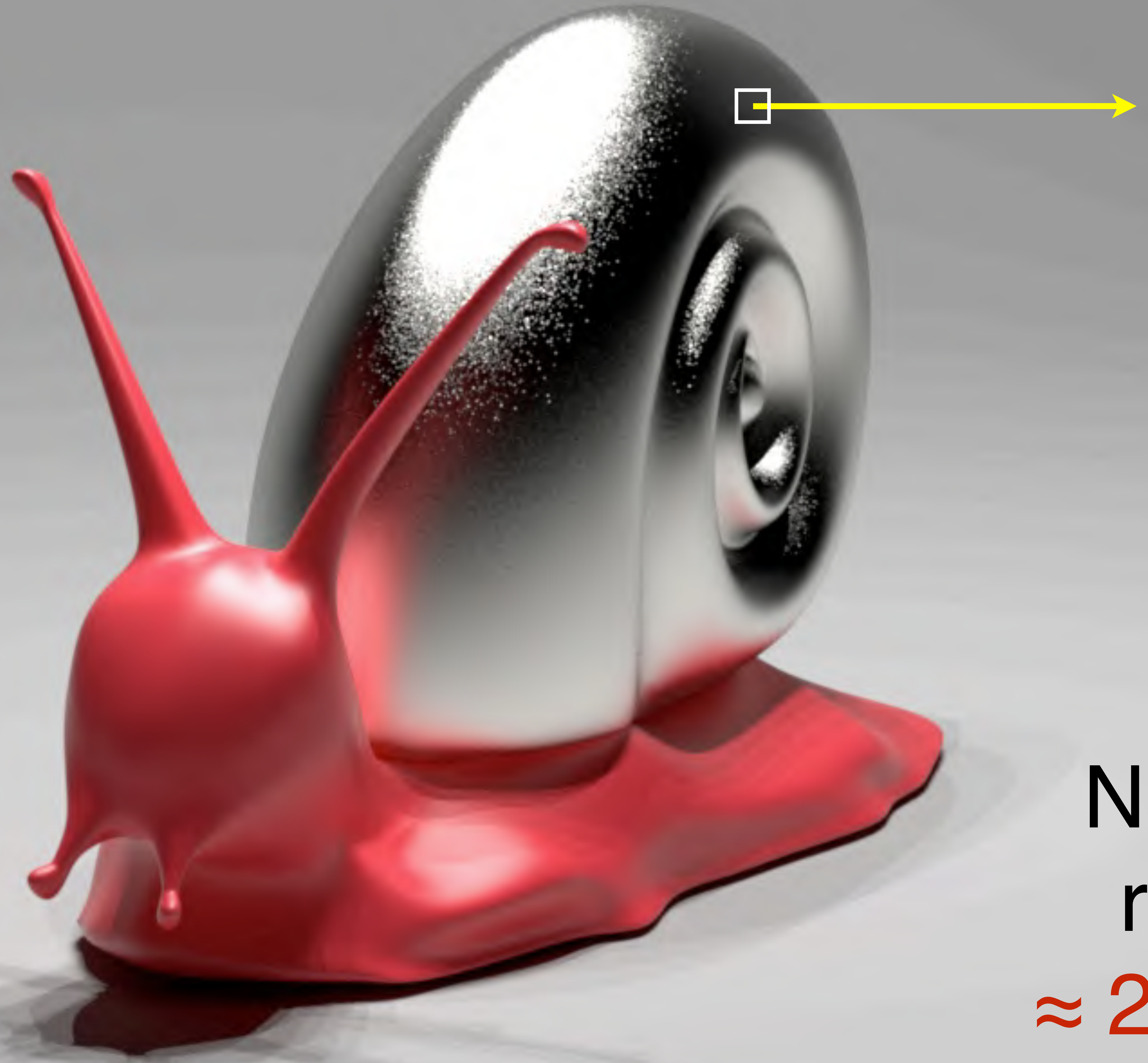


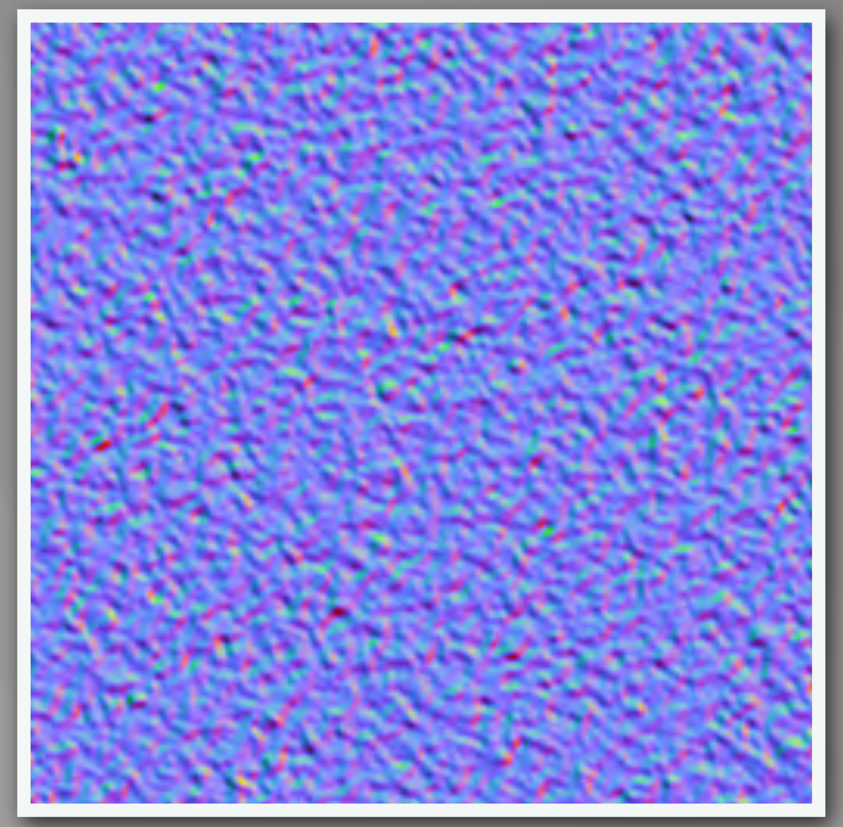
What we expect

Recap: texture mapping



Define the details

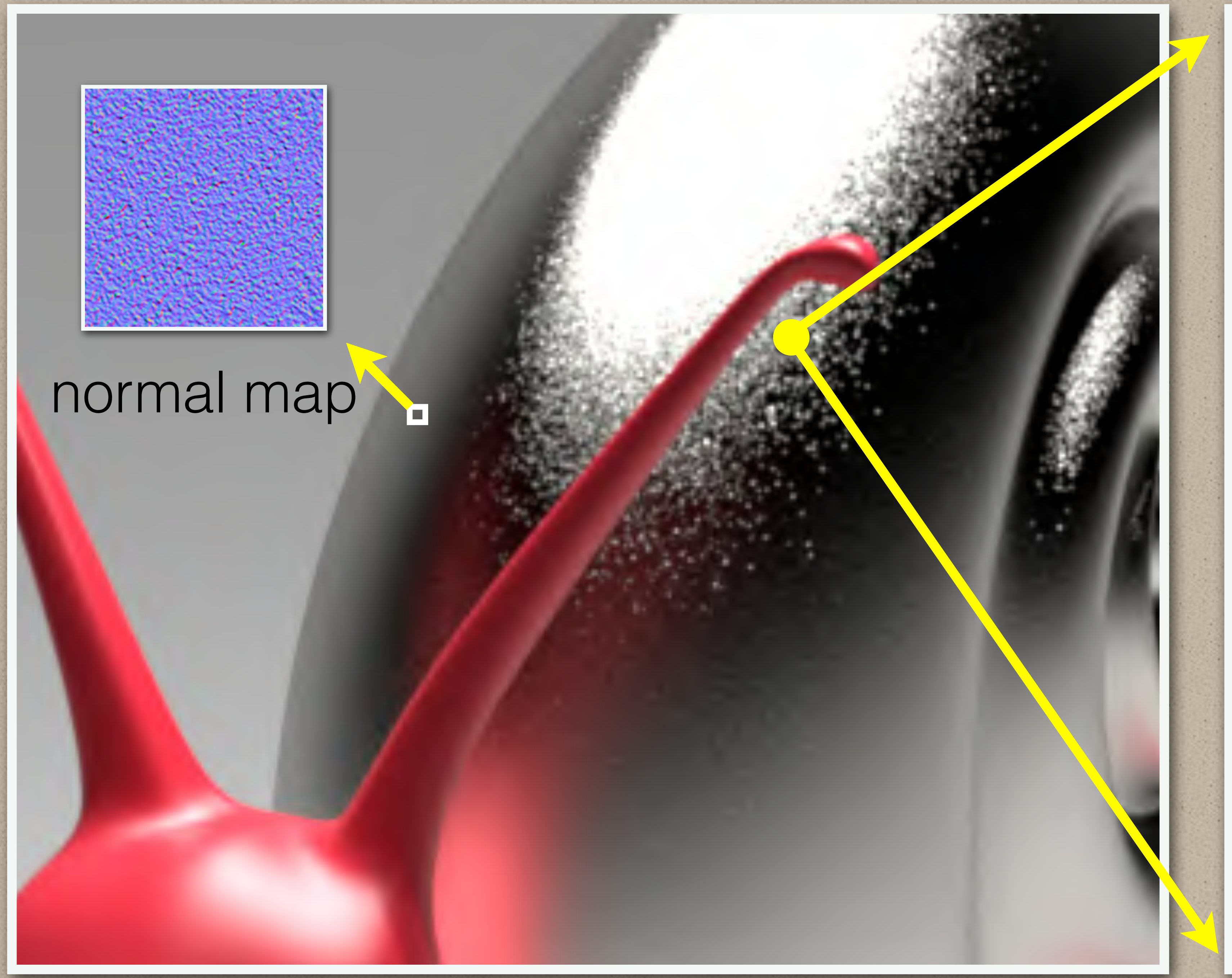




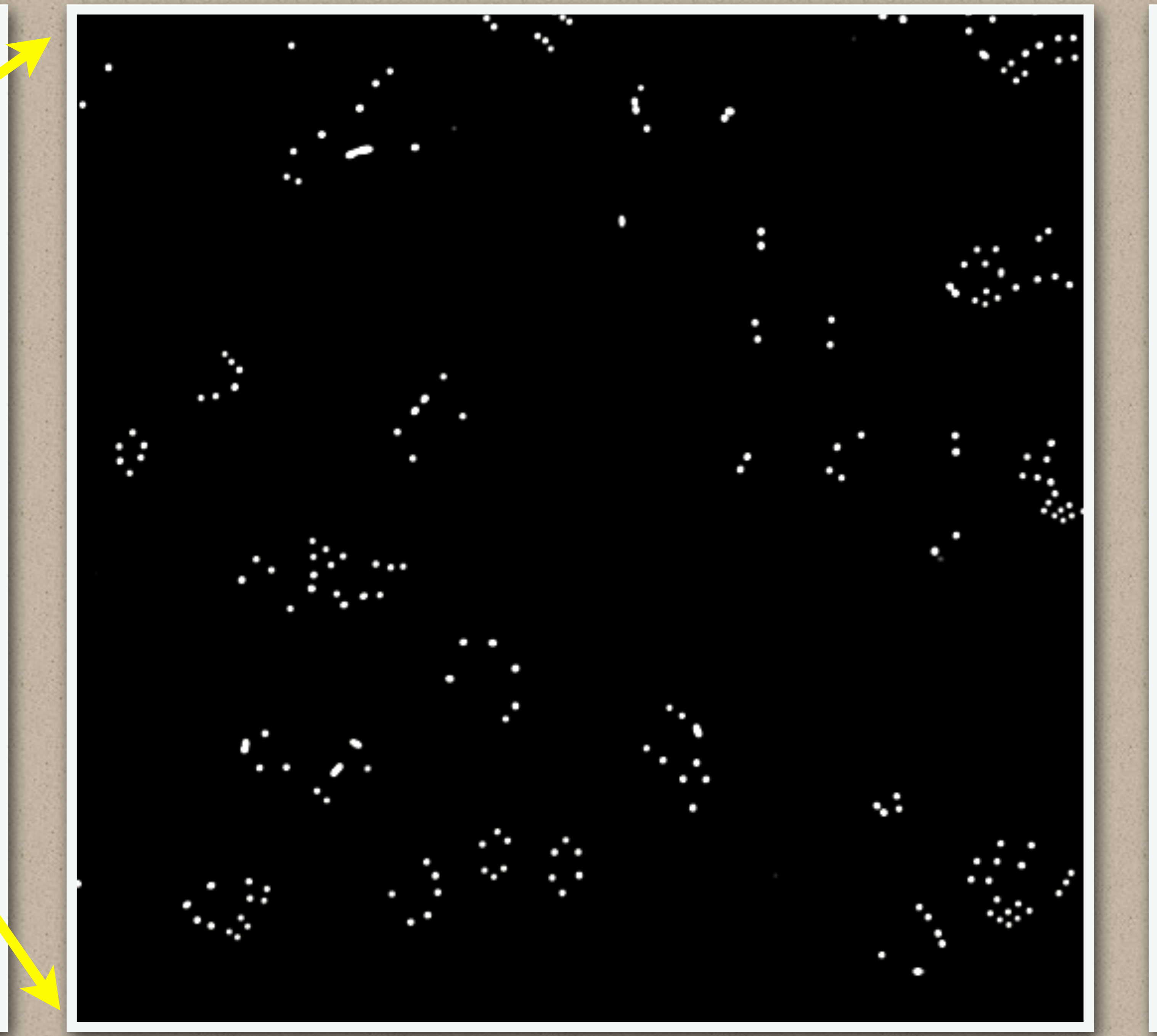
isotropic noise normal map

Normal map resolution:
≈ 200K x 200K

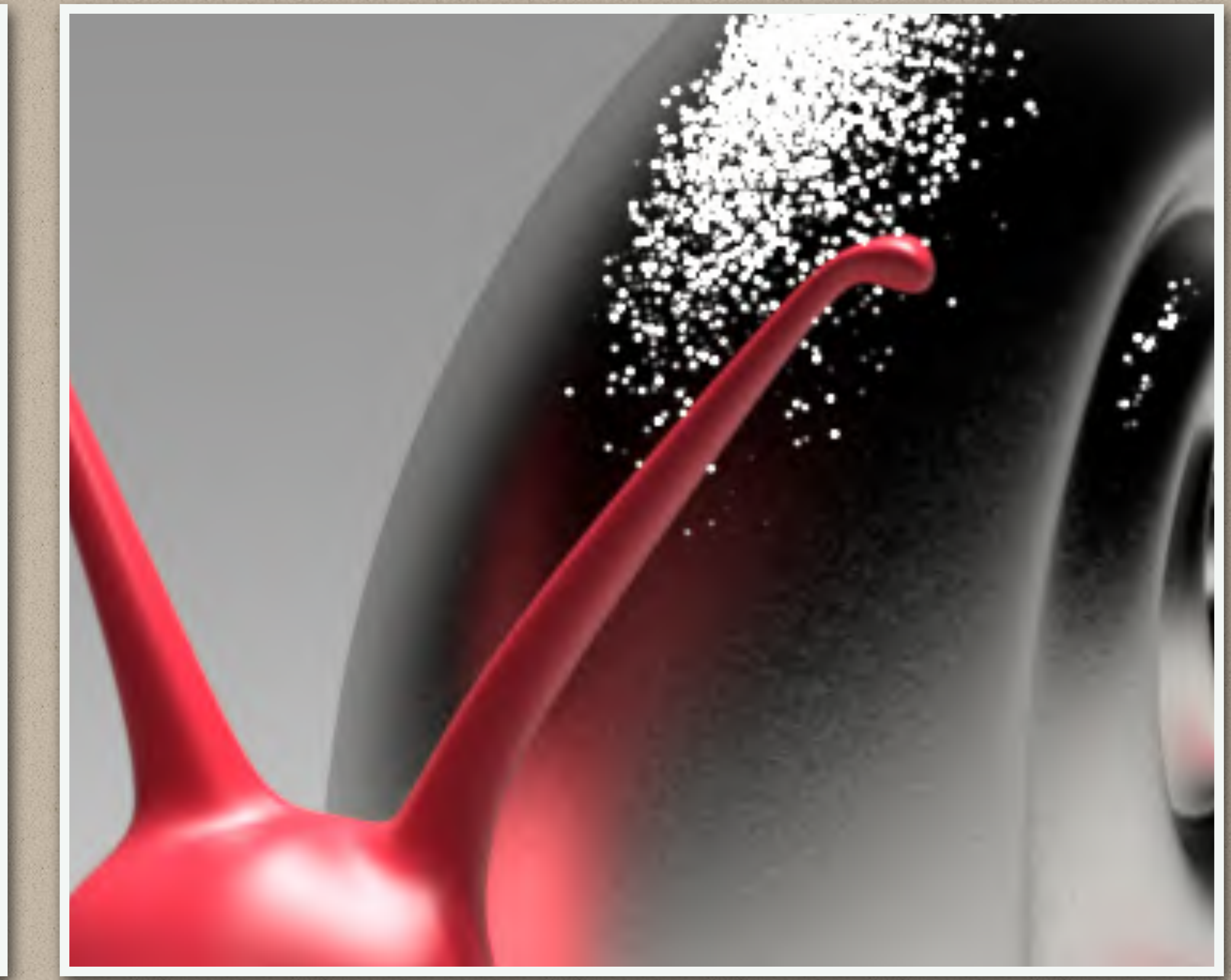
Rendering? Too difficult!



our result



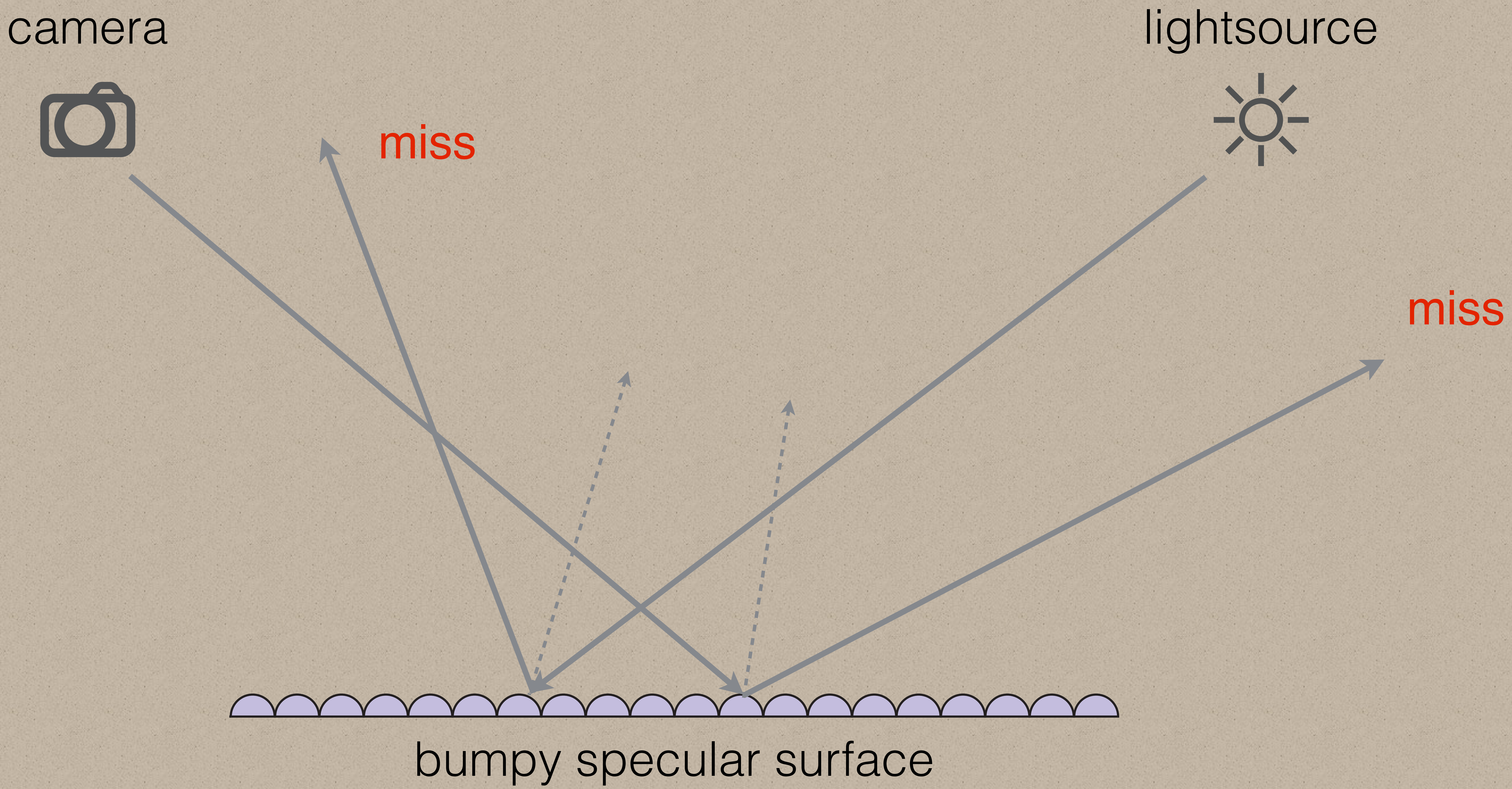
zoom of a single pixel



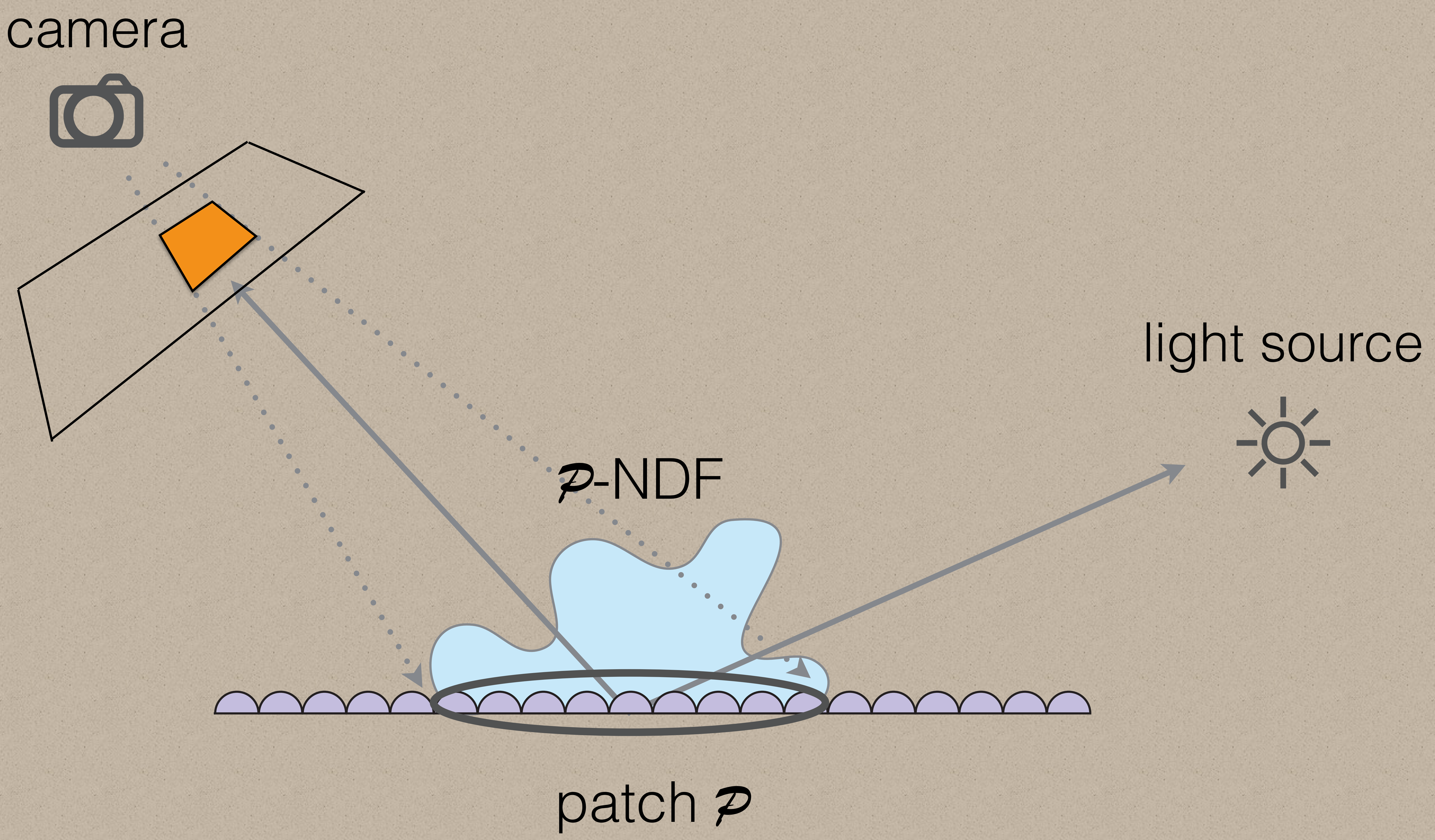
naive sampling (2 hours)

(≫ 21.3 days to converge)

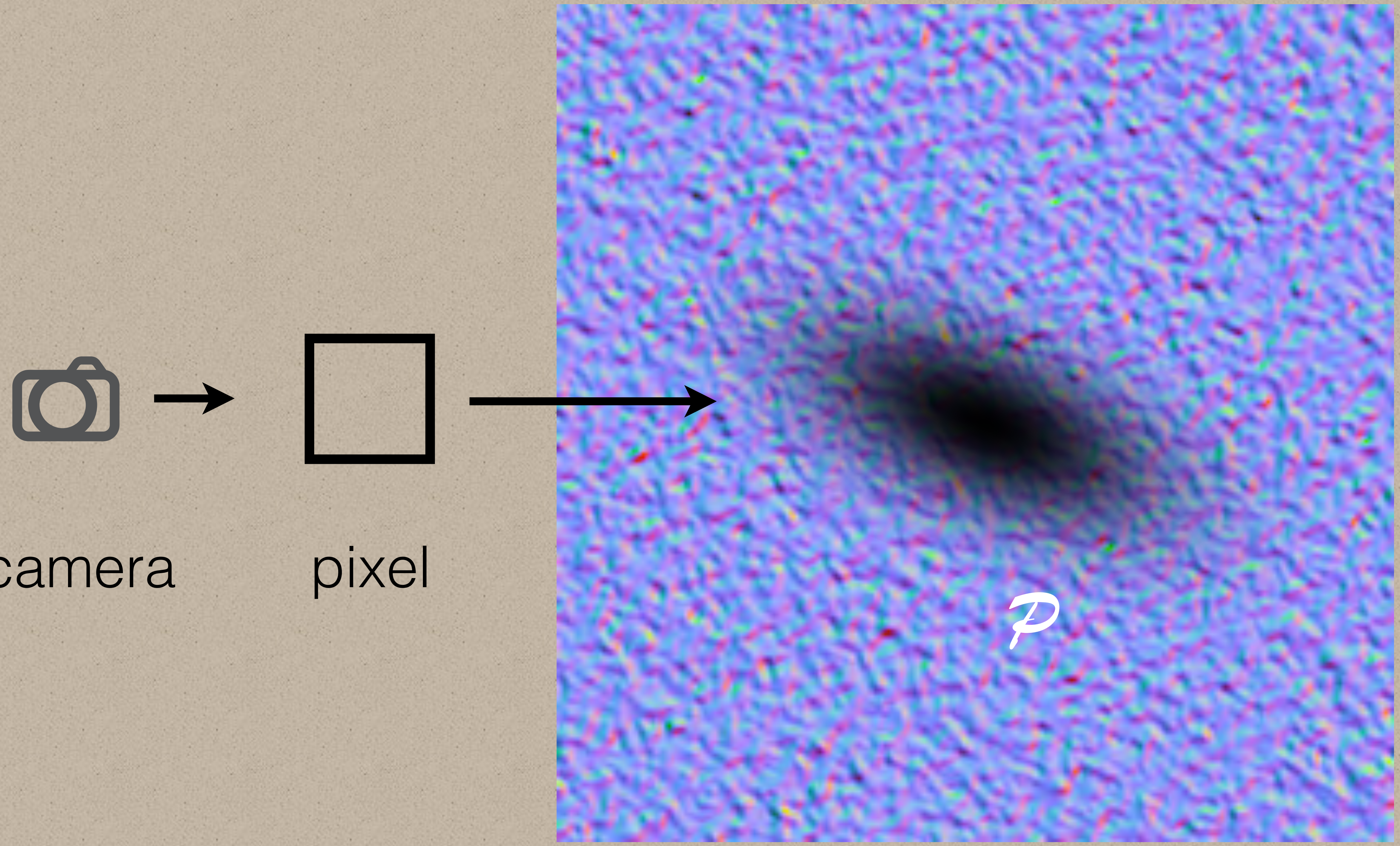
Difficult ray sampling problem



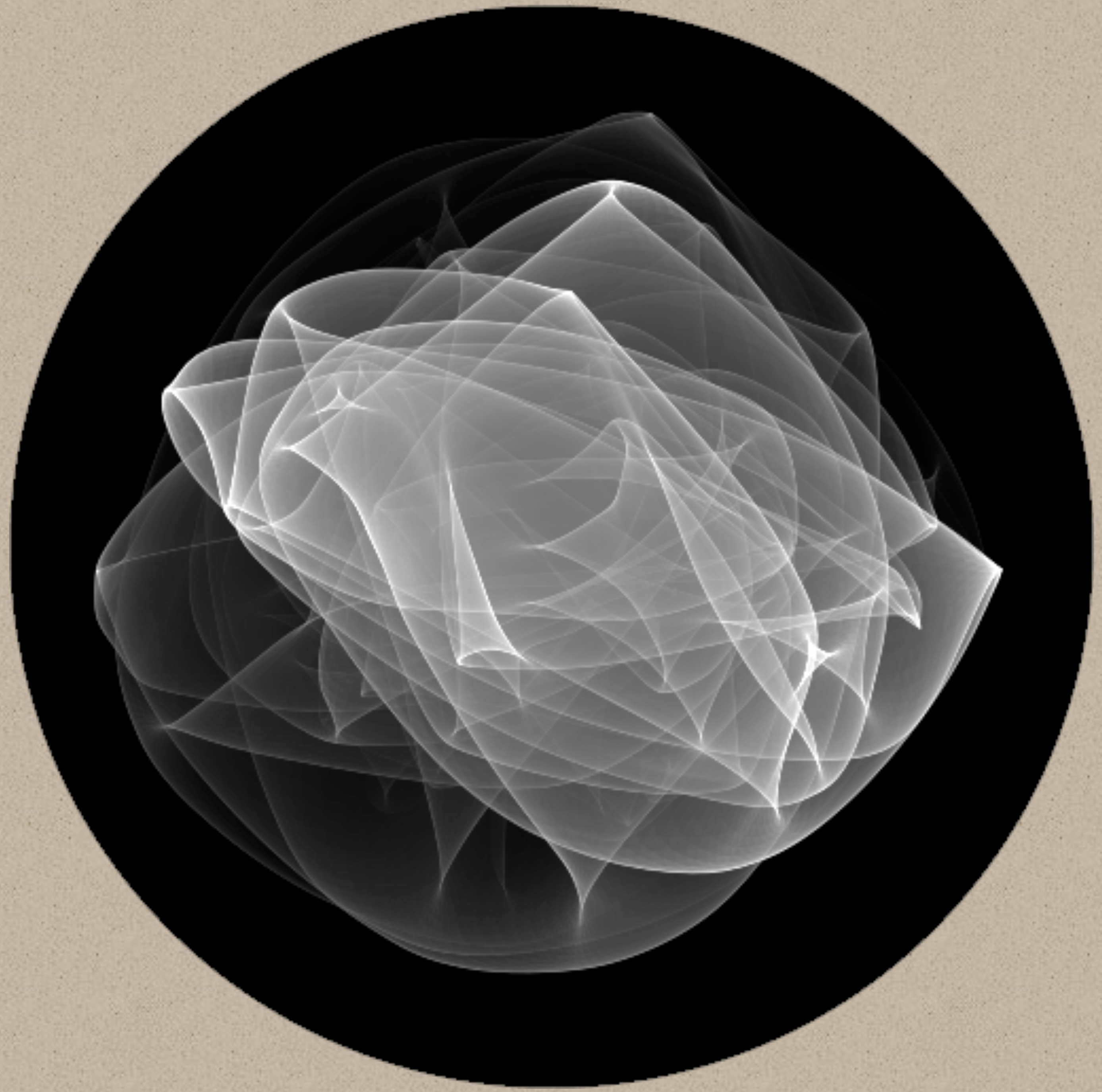
Solution: BRDF over a pixel



$\mathcal{P}$-NDF over a pixel

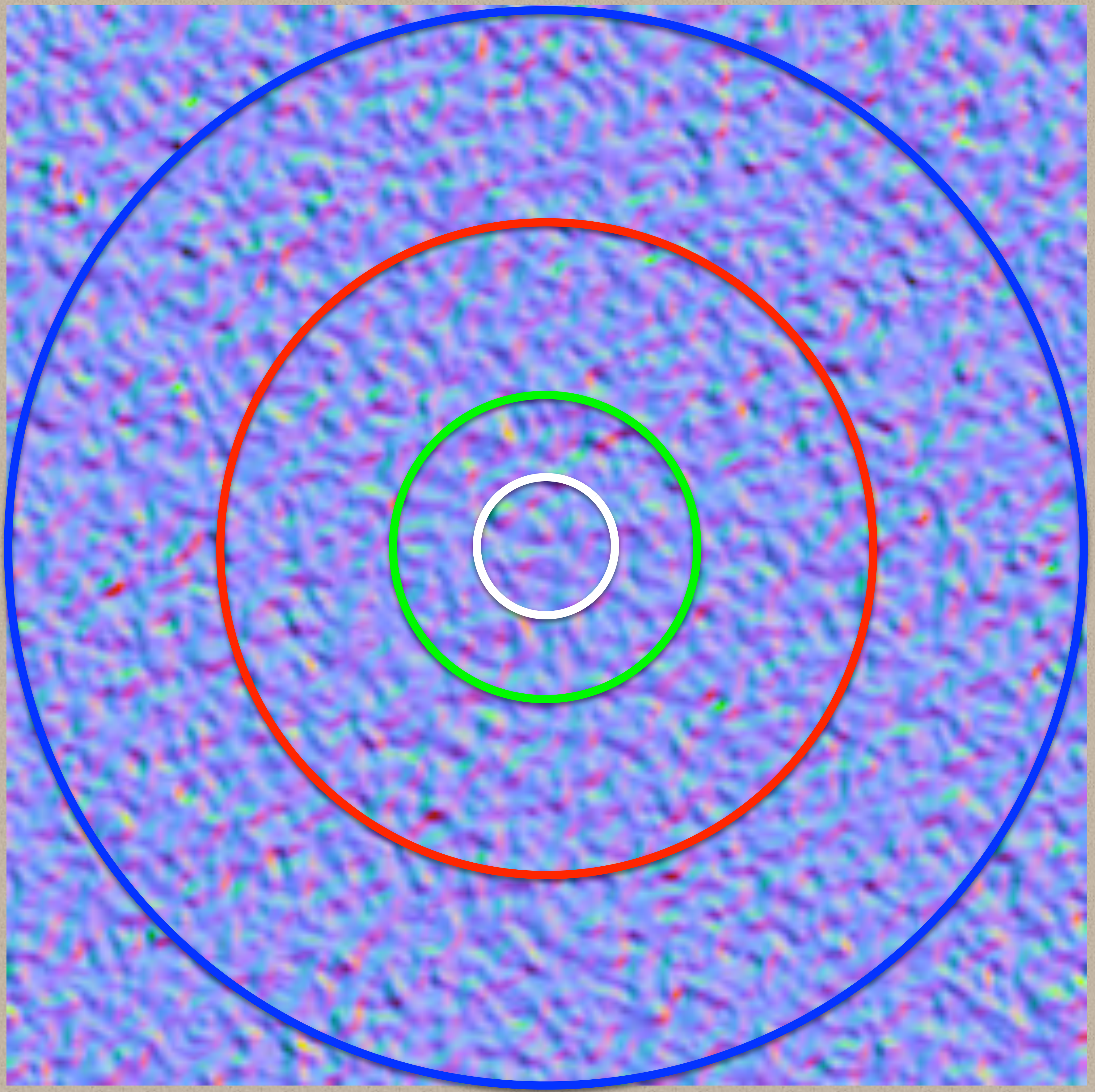


normal-mapped
surface

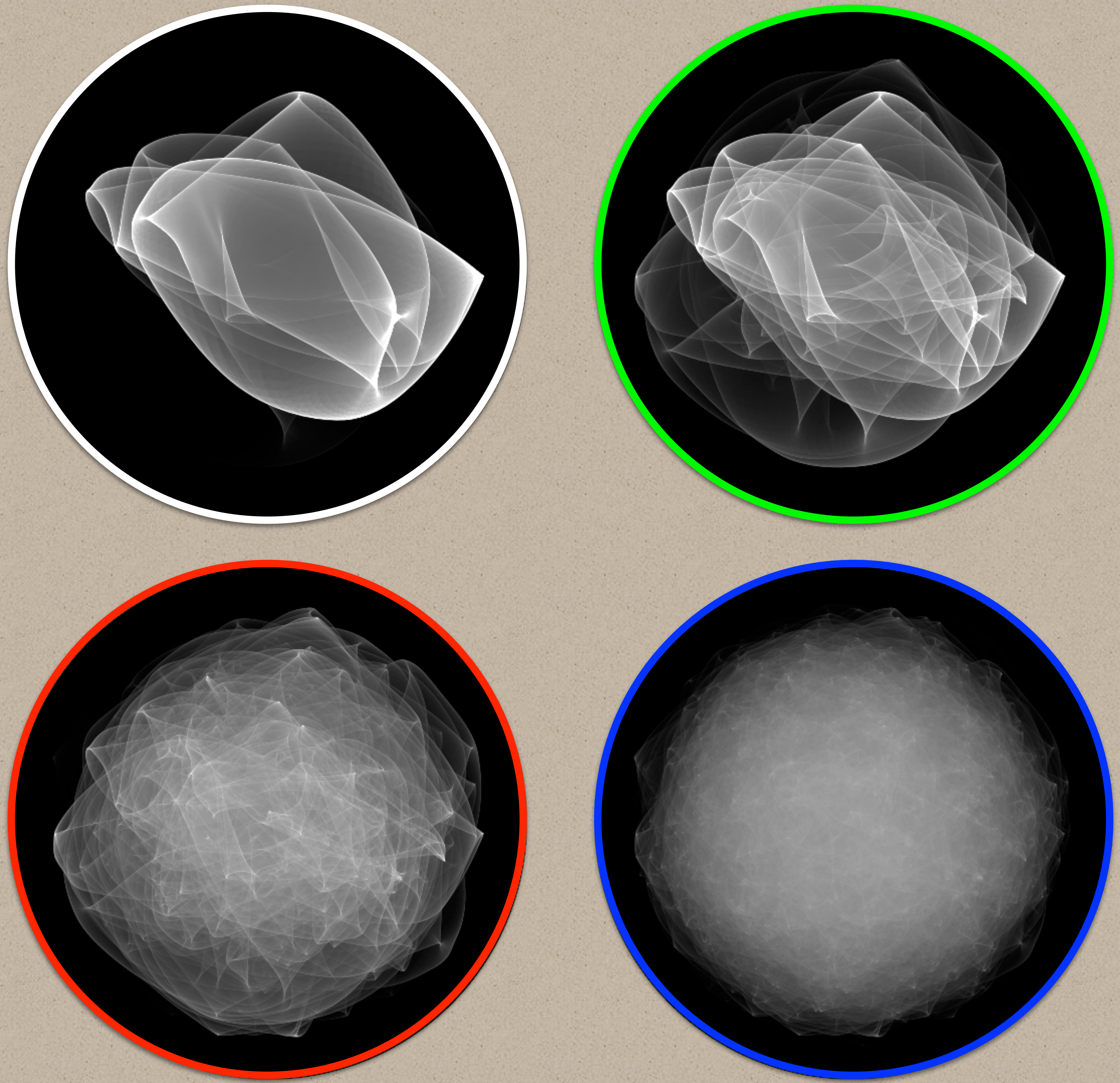


$\mathcal{P}$-NDF:
distribution of normals over $\mathcal{P}$

$\mathcal{P}$-NDFs have sharp features

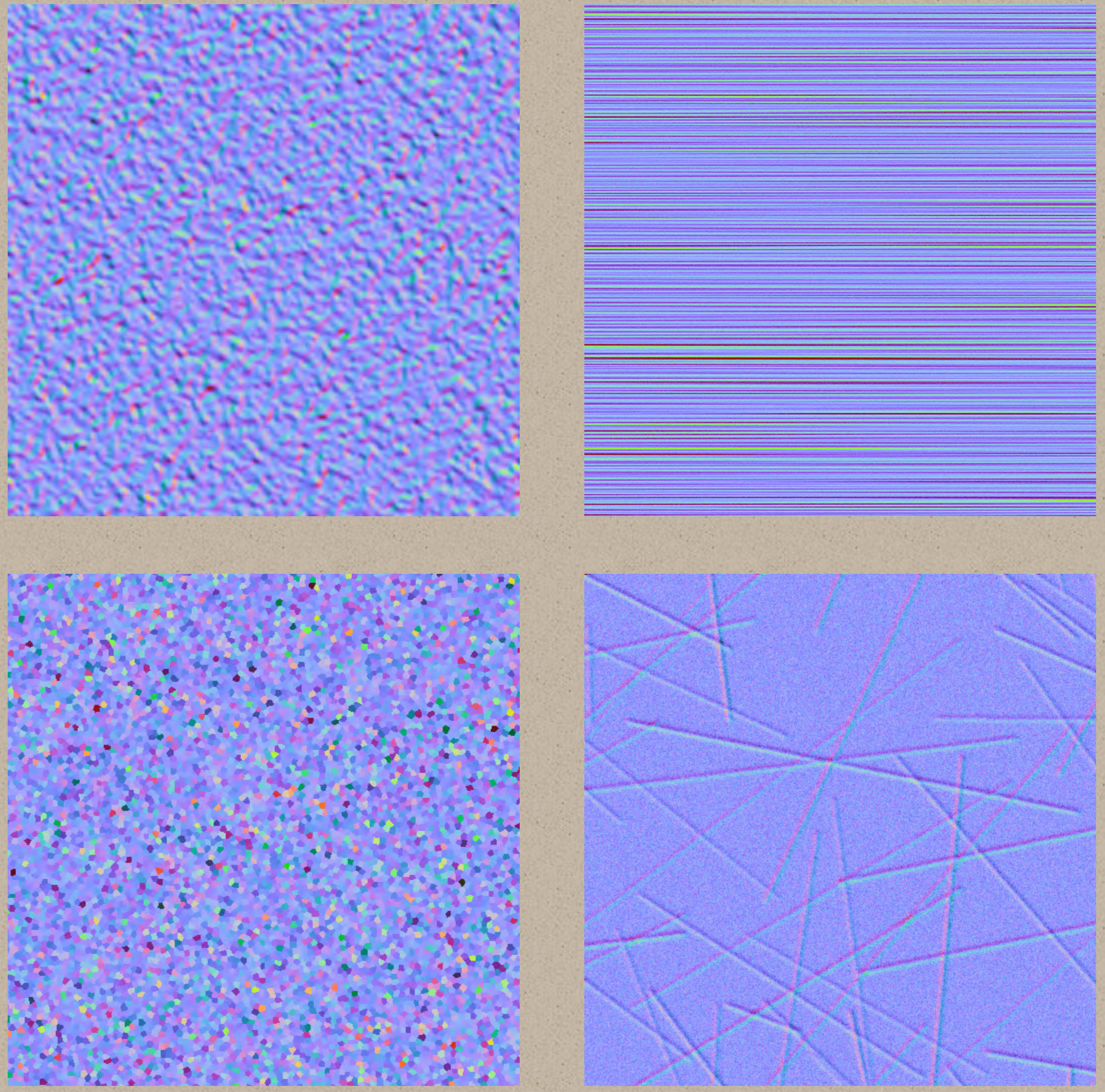


normal map

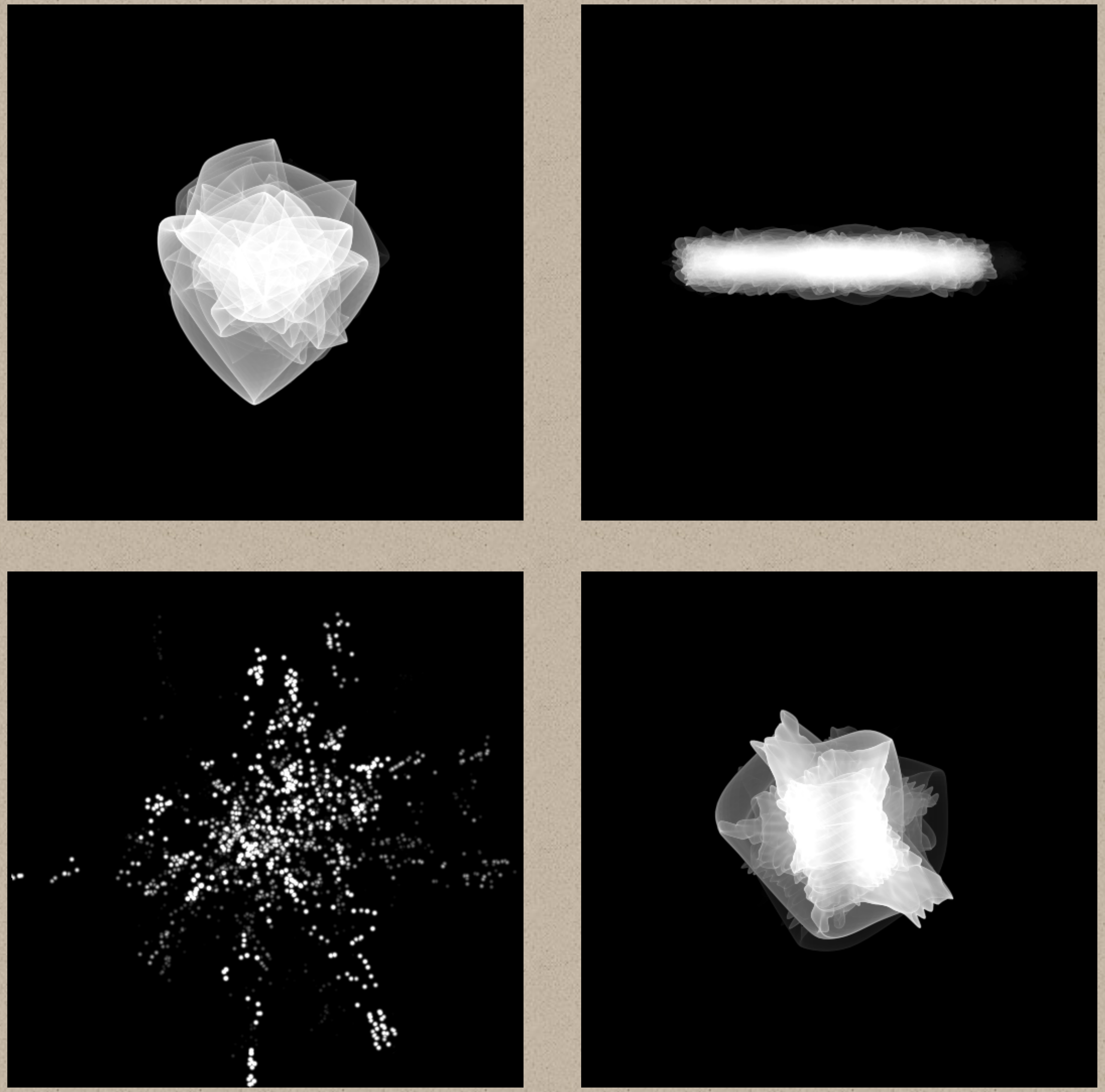


$\mathcal{P}$-NDFs

$\mathcal{P}$-NDF shapes

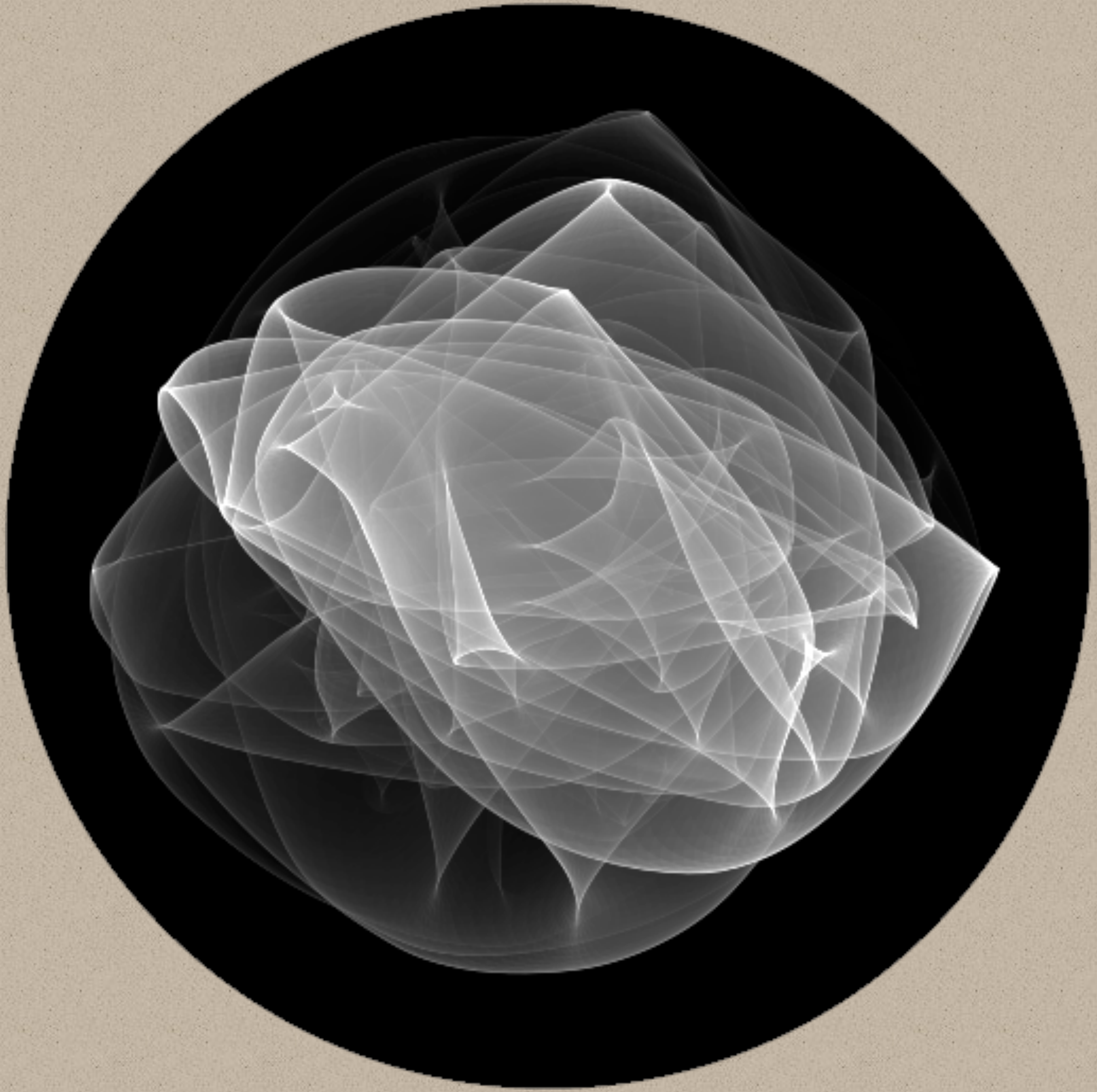


normal maps



$\mathcal{P}$-NDFs

Converting $\mathcal{P}$-NDF to BRDF



microfacet BRDF

accurate $\mathcal{P}$-NDF

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

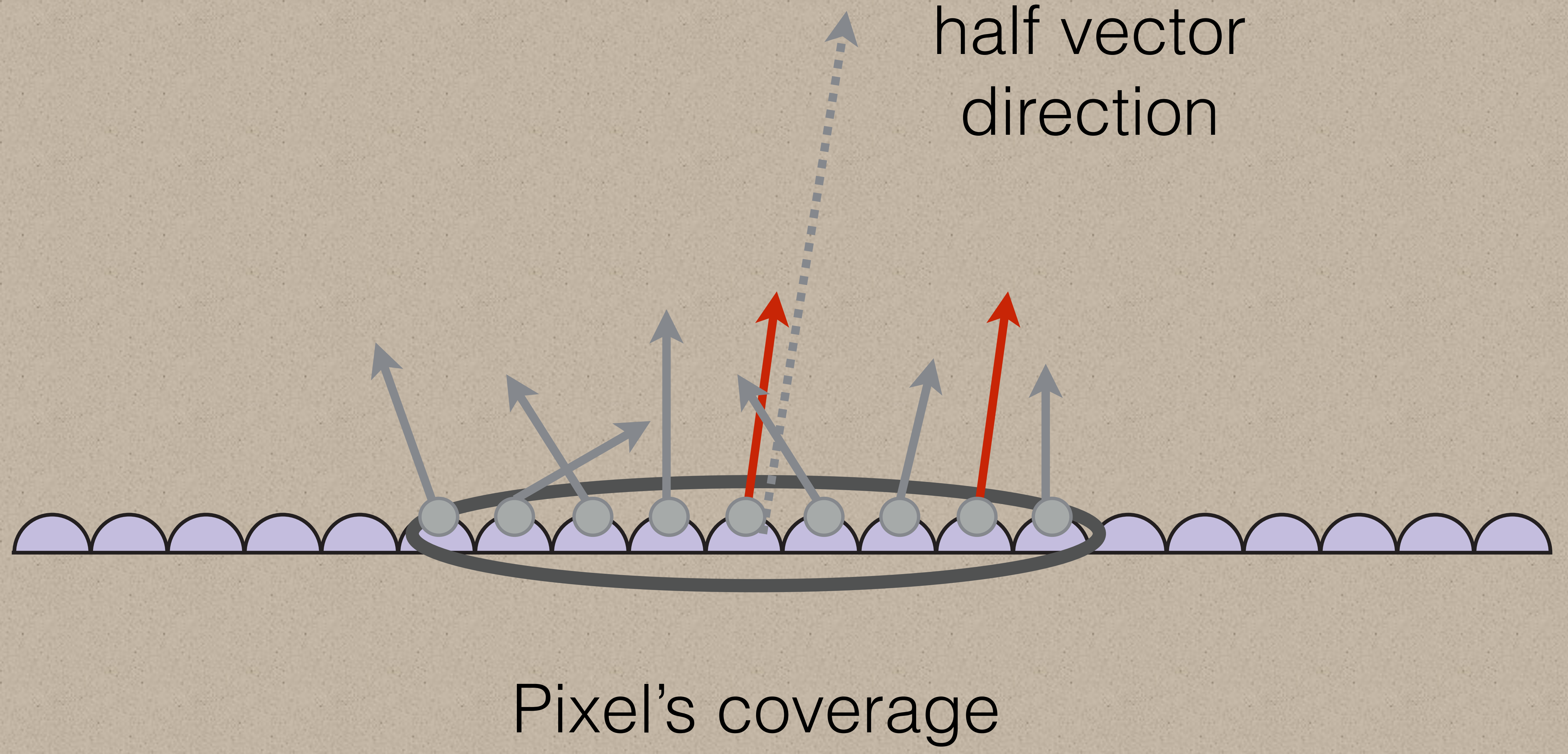
$\mathcal{P}$-NDF evaluation with perfect specularity

$$D(s) = \int_{-\infty}^{\infty} \underline{G_p(u)}\,\underline{\delta(n(u) - s)}\,du = \sum_i \frac{G_p(u_i)}{\boxed{|n'(u_i)|}},$$

can be zero

$\mathcal{P}$-NDF

sum over the roots of $n(u) = s$

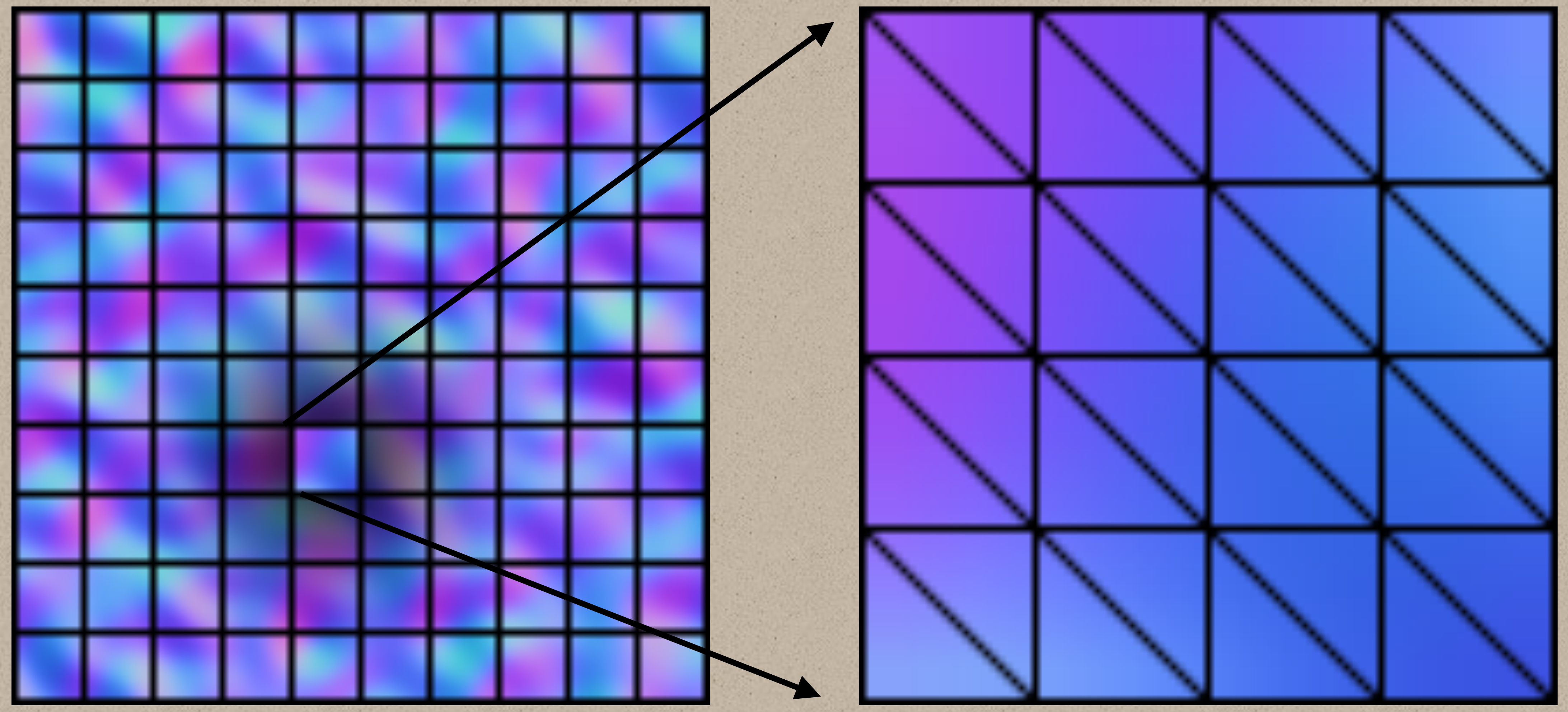


Better way: Intrinsic roughness

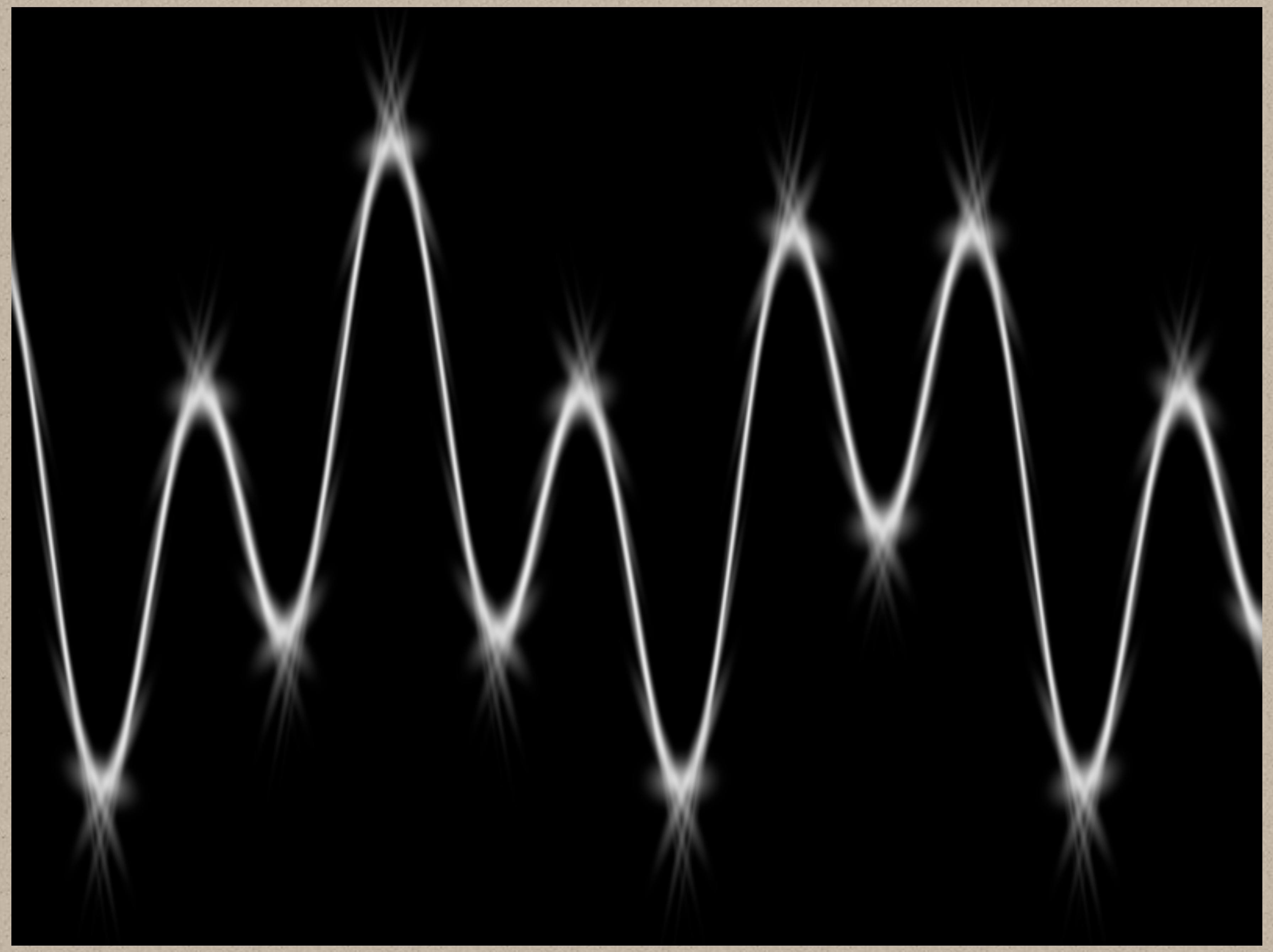
$$D(s) = \underbrace{\int_{-1}^{1} G_r(s - s')}_{\text{roughness kernel (sigma = 0.001 radians)}} \overbrace{\int_{-\infty}^{\infty} G_p(u)\delta(n(u) - s')du}^{\text{perfectly specular NDF}} \boxed{ds'}$$

$$= \int_{-\infty}^{\infty} G_p(u) G_r(n(u) - s)\, \mathrm{d}u$$

Two different solutions to NDFs



Yan et al. [2014]

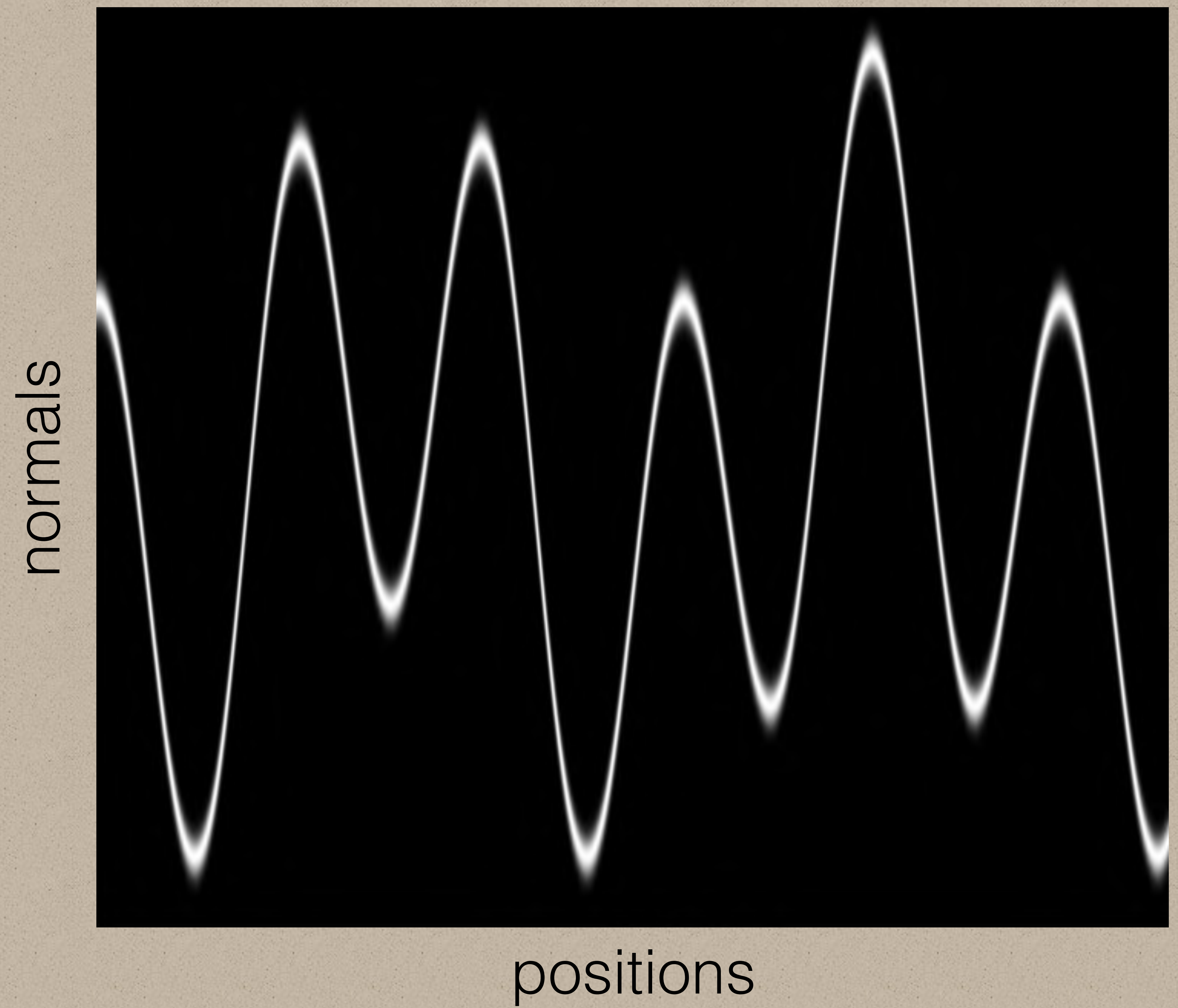


Yan et al. [2016]
(100x faster)

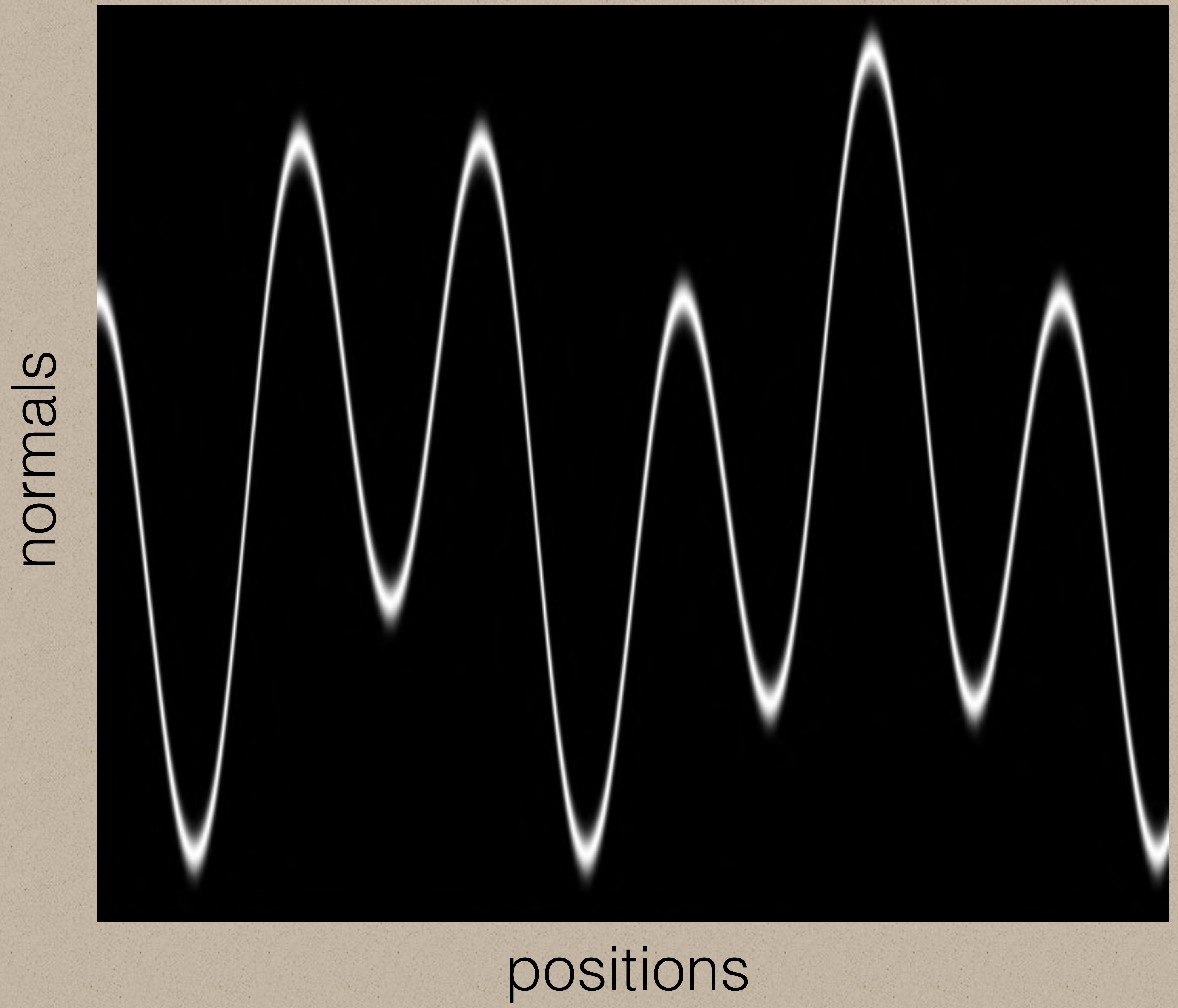
$$D(s) = \int_{\infty}^{\infty} G_p(u) \boxed{G_r(n(u) - s)} \, \mathrm{d}u$$

‘vertically’ blurred
graph of normal map

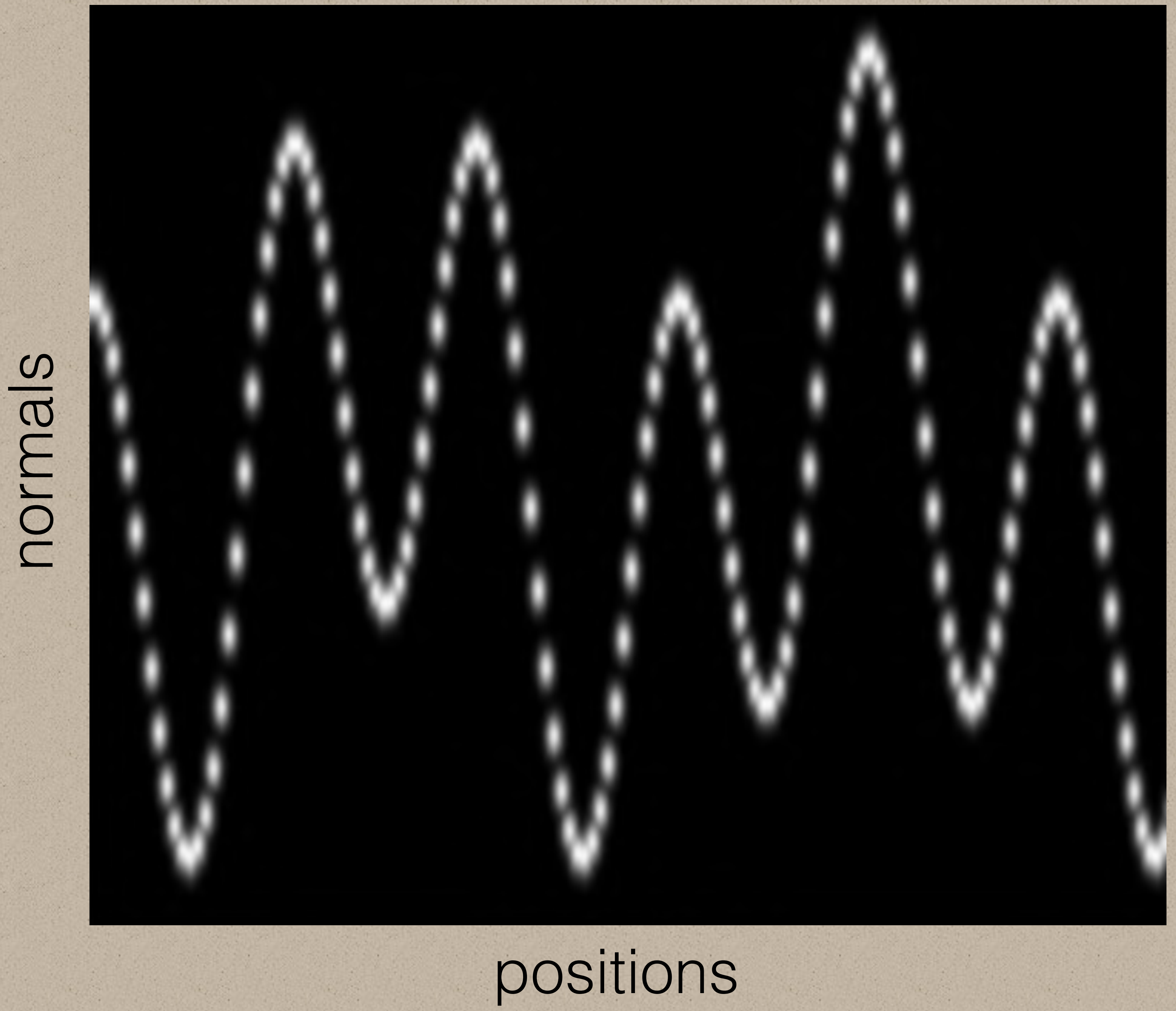
position-normal
distribution



Approximating position-normal distribution

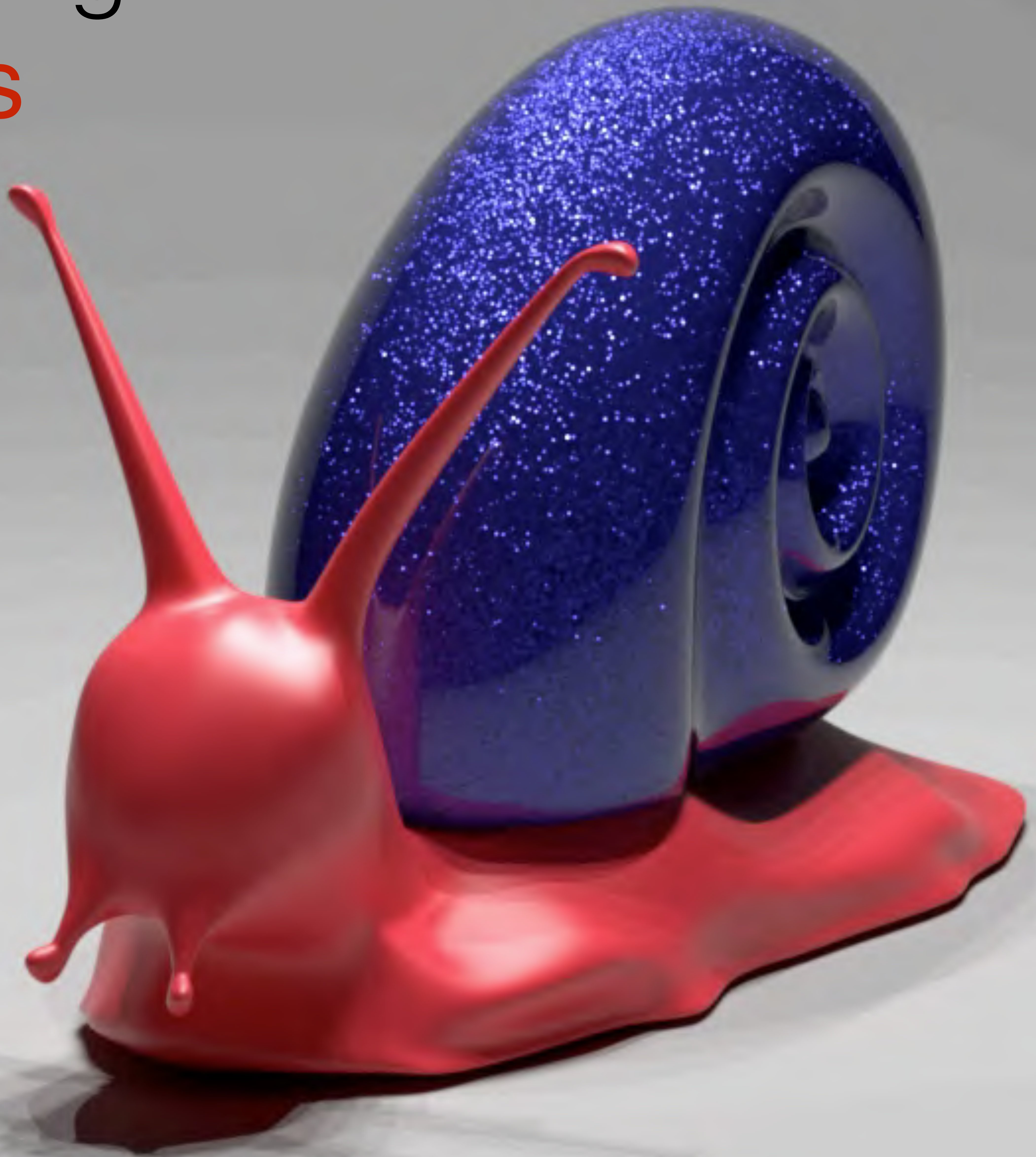


position-normal distribution

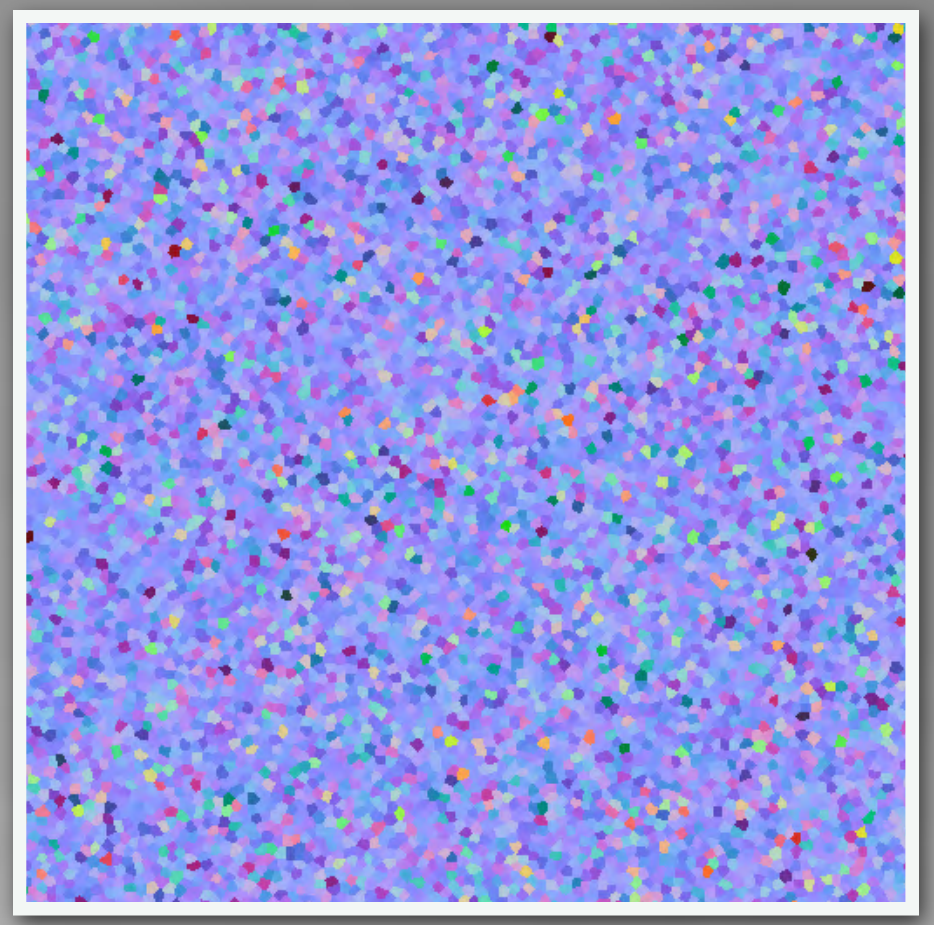


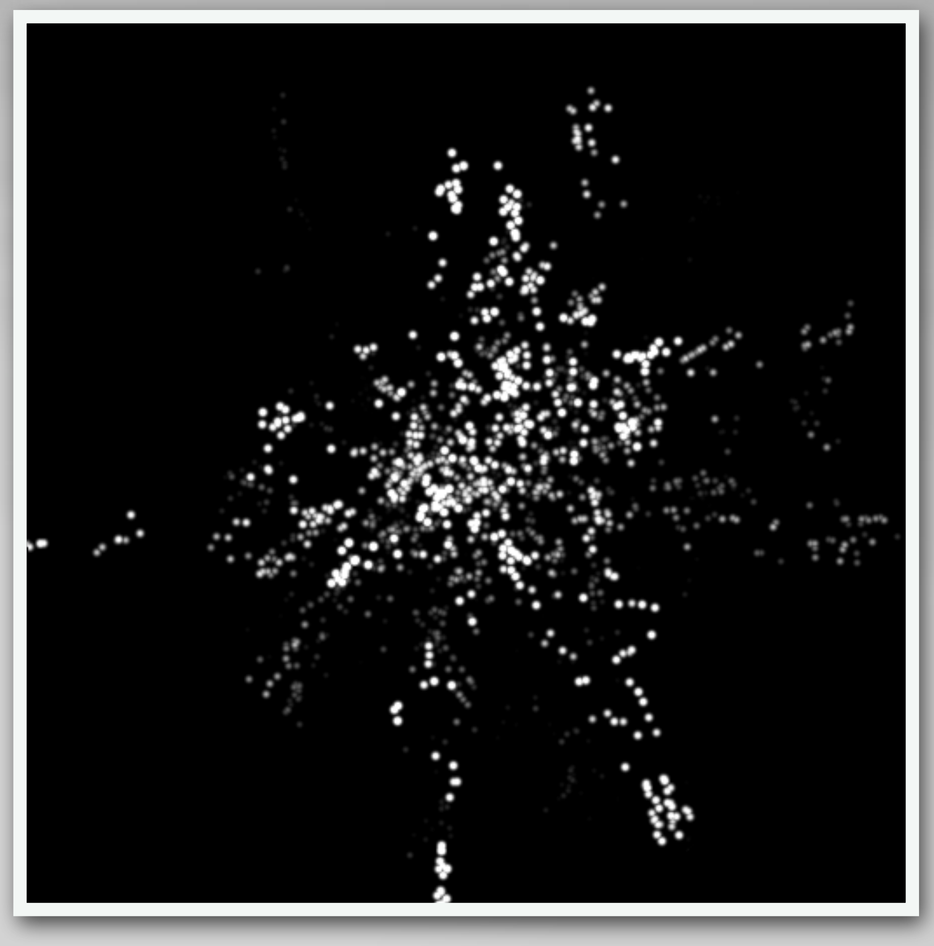
approx using 160 flat elements
(axis-aligned Gaussians)

Application using flat elements



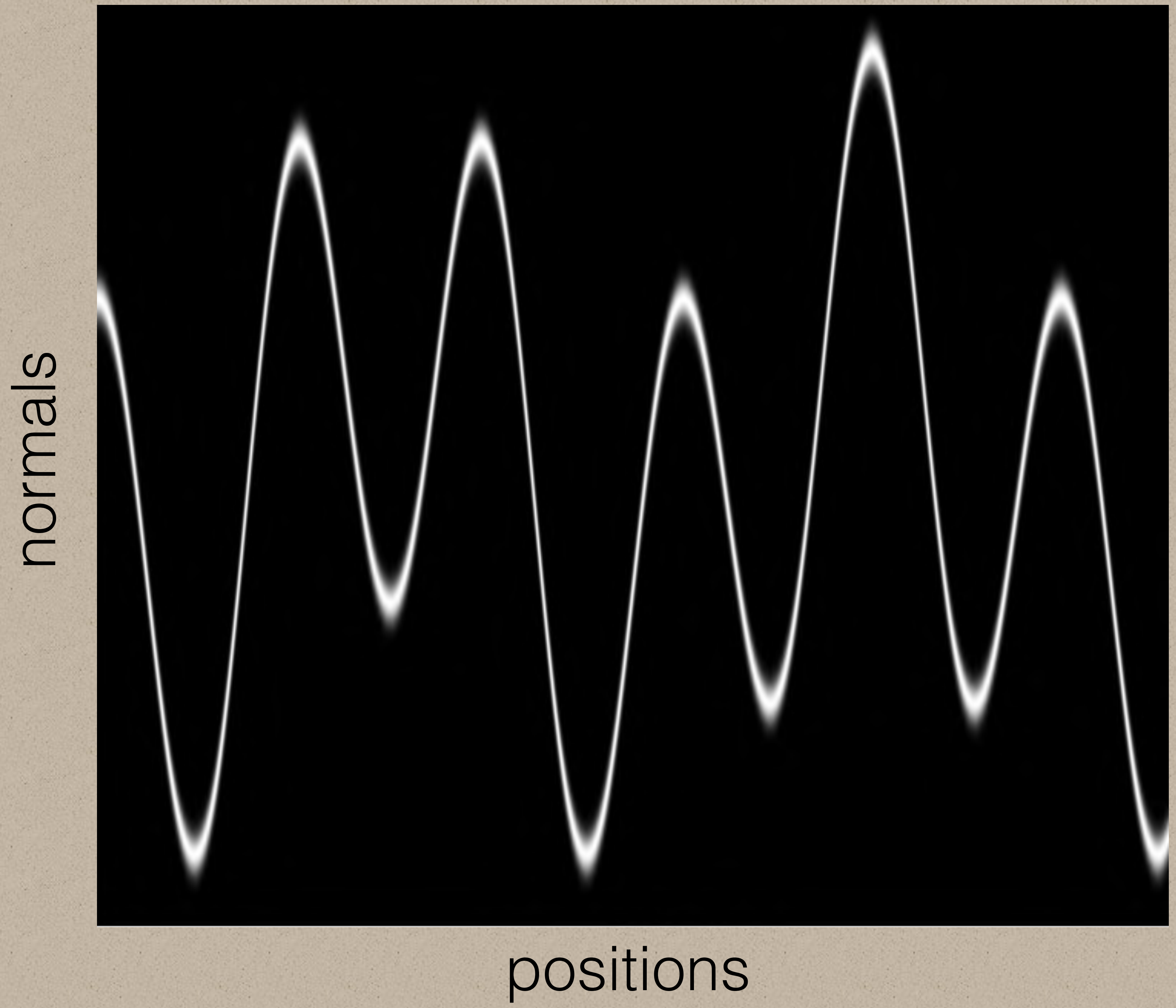
SIGGRAPH 2014
technical paper trailer



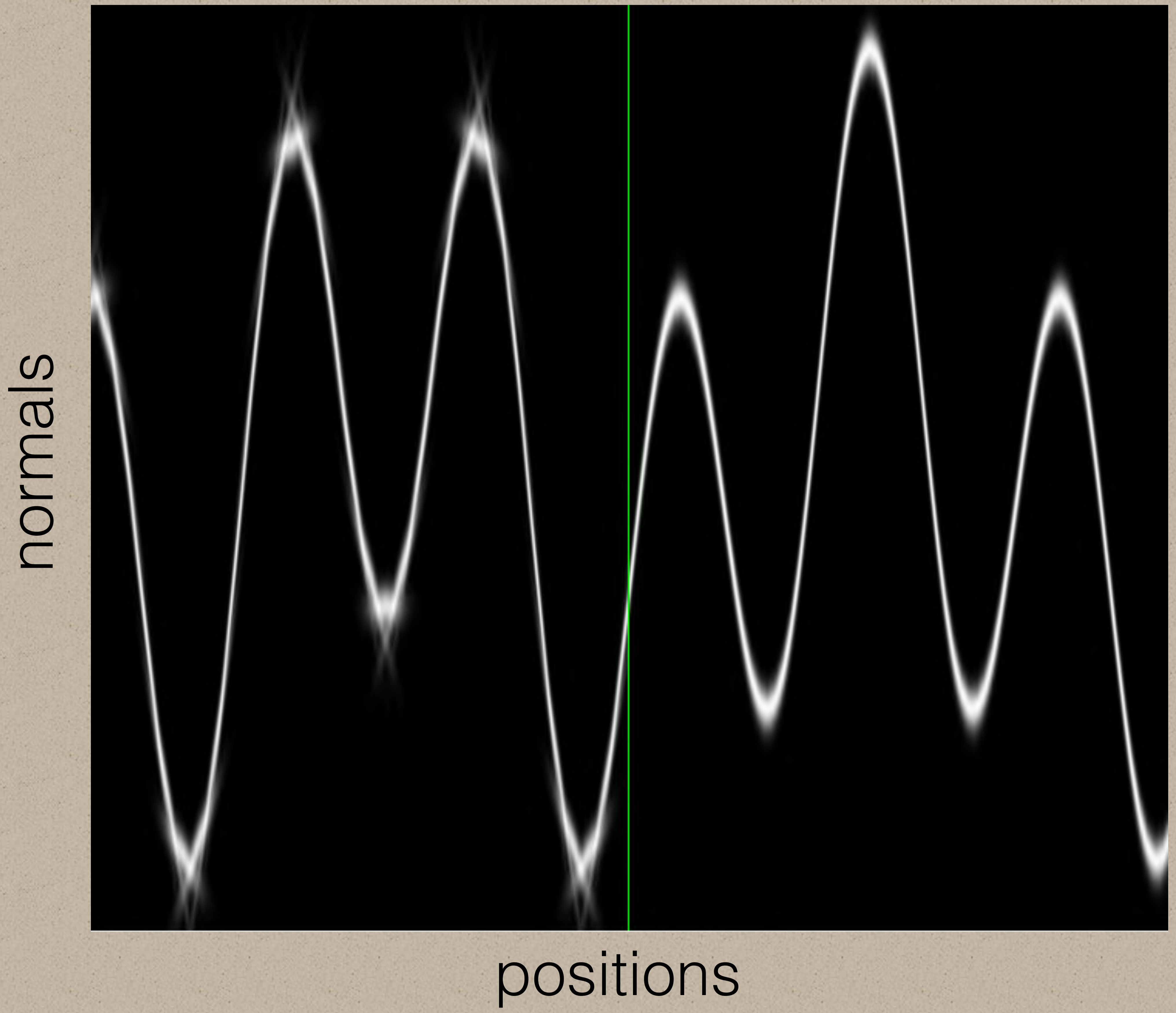


Metallic flakes

Approximating position-normal distribution



‘vertically’ blurred normal map



approx using 80/160 curved elements
(arbitrary Gaussians)

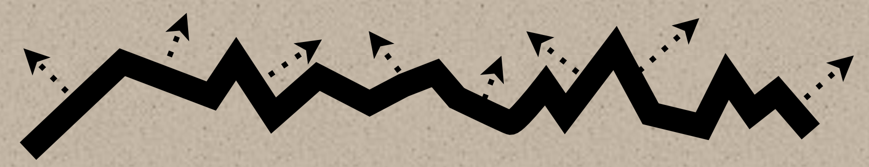
$\mathcal{P}$-NDF evaluation cont.

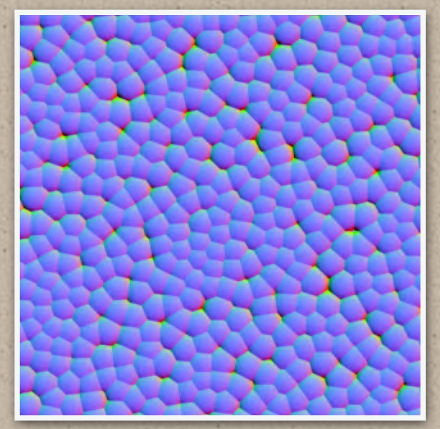
$$D(s) = \int_{\infty}^{\infty} G_p(u) G_r(n(u) - s)\,\mathrm{d}u$$

position-normal distribution

$$= \sum_{i=1}^{m} \int_{-\infty}^{\infty} \underline{G_p(u)}\,\underline{G_i(u, s)}\,\mathrm{d}u$$

1D	1D slice of 2D
2D	2D slice of 4D

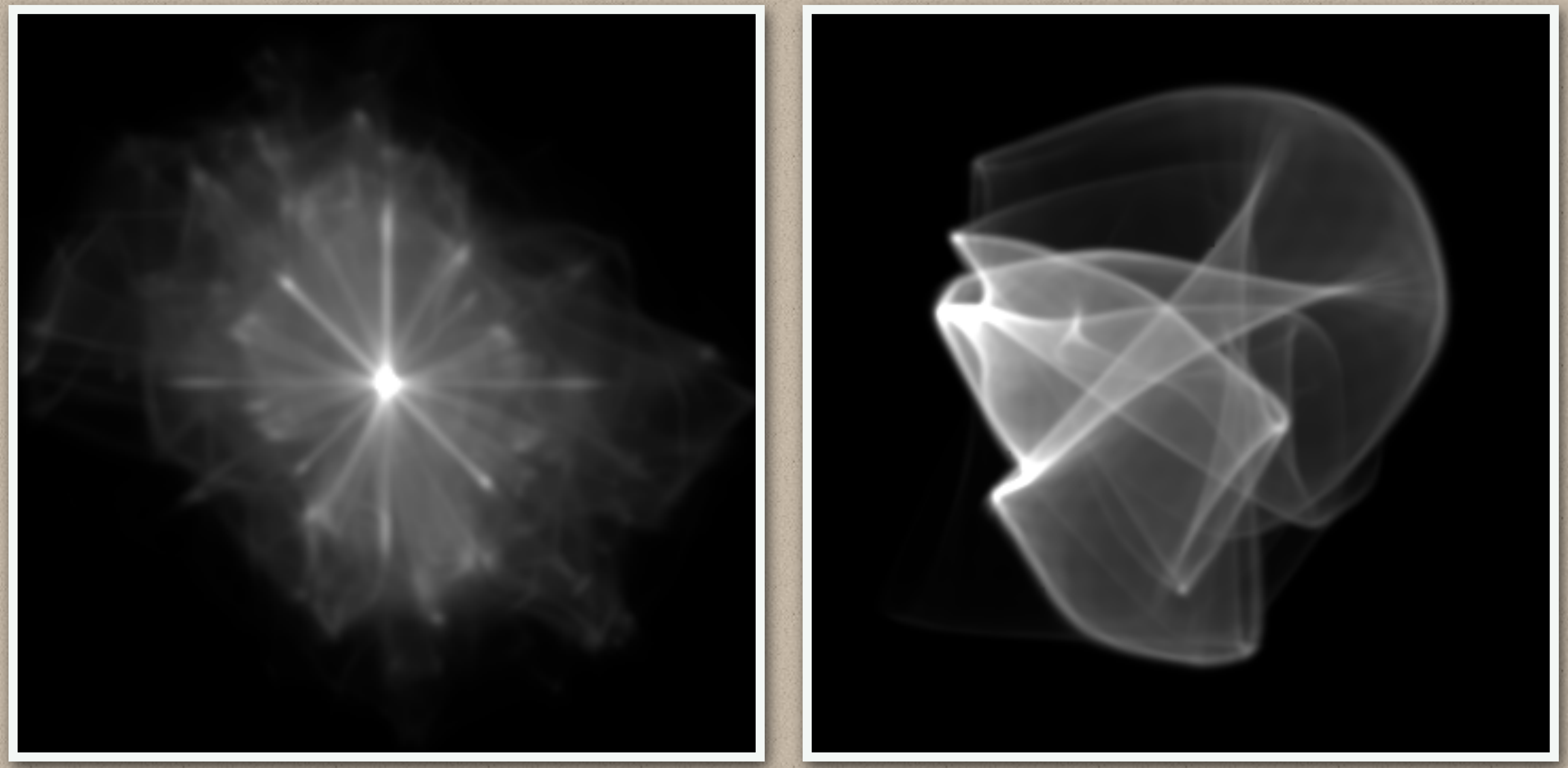




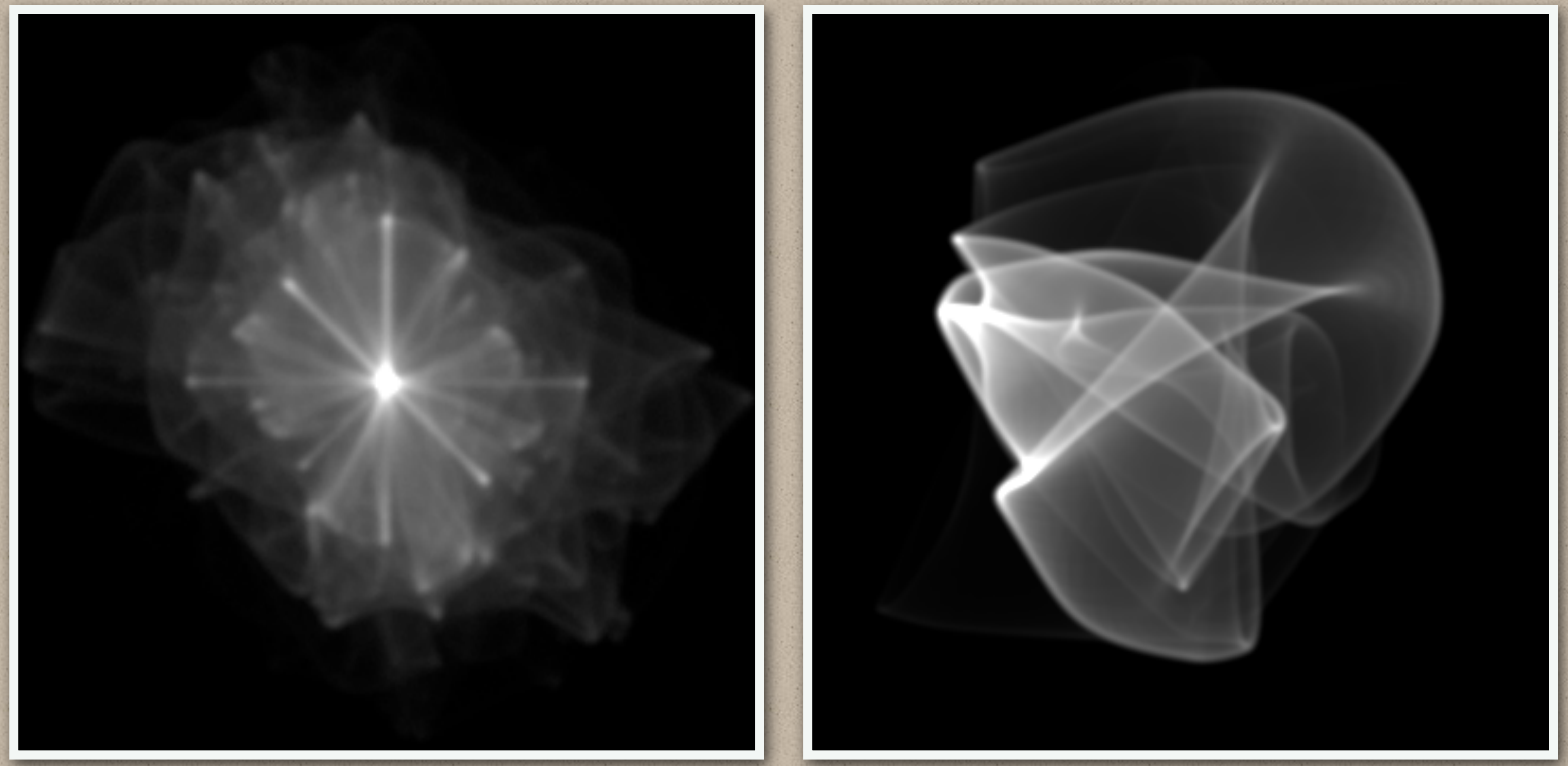
Application using curved elements

Ocean waves

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

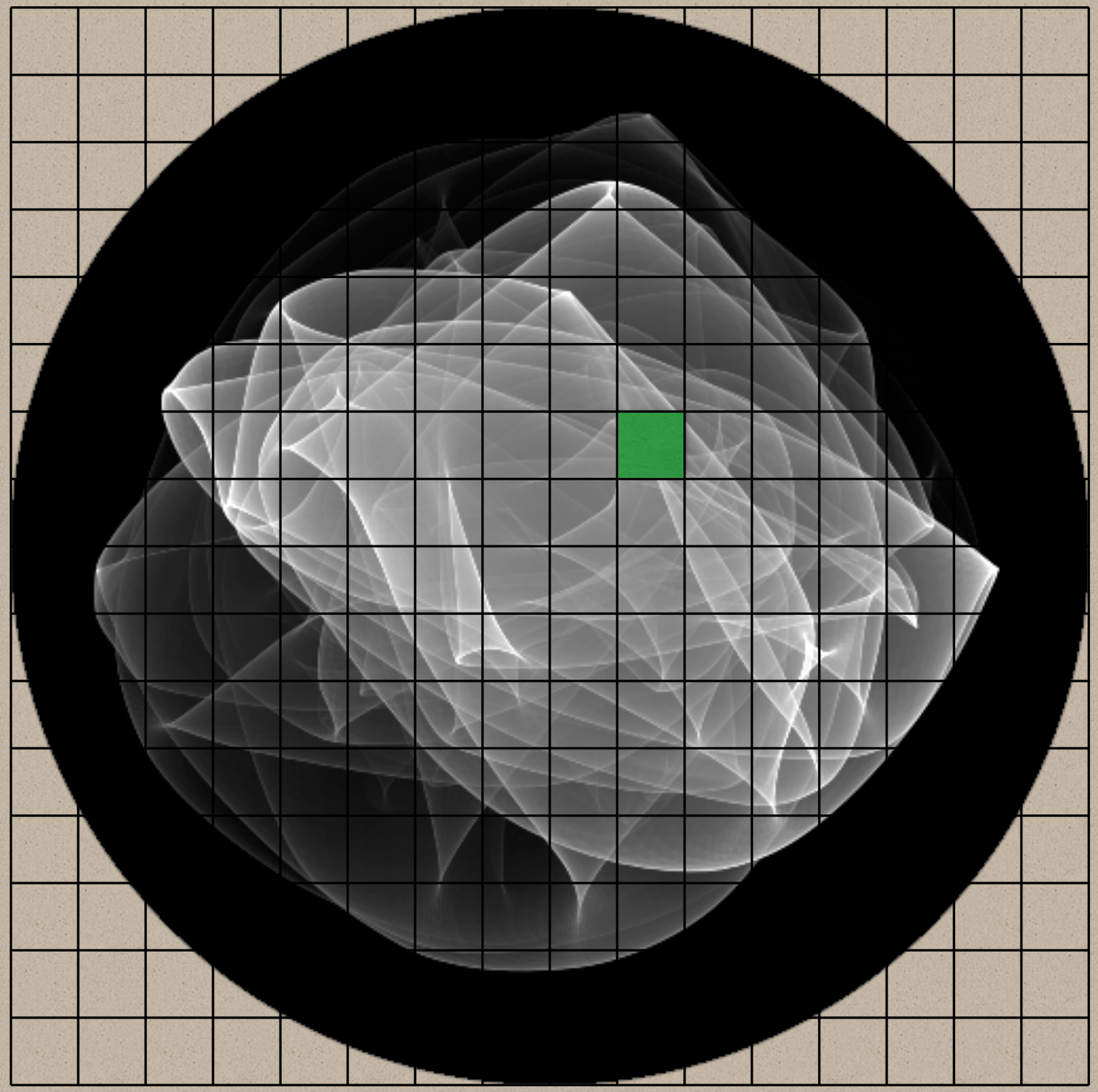


our method
(point-wise evaluation)

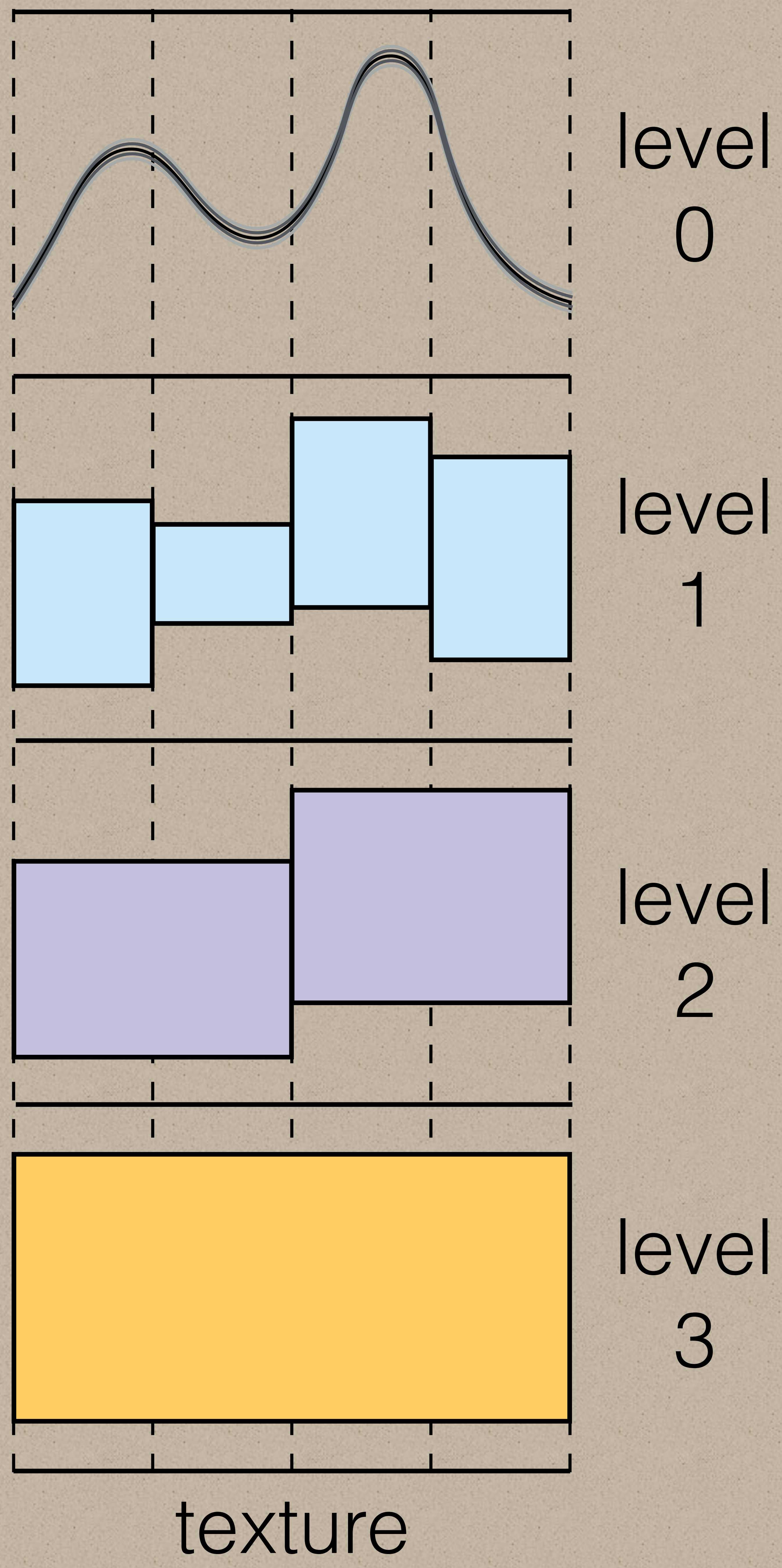


binning (histogram)
100 million samples

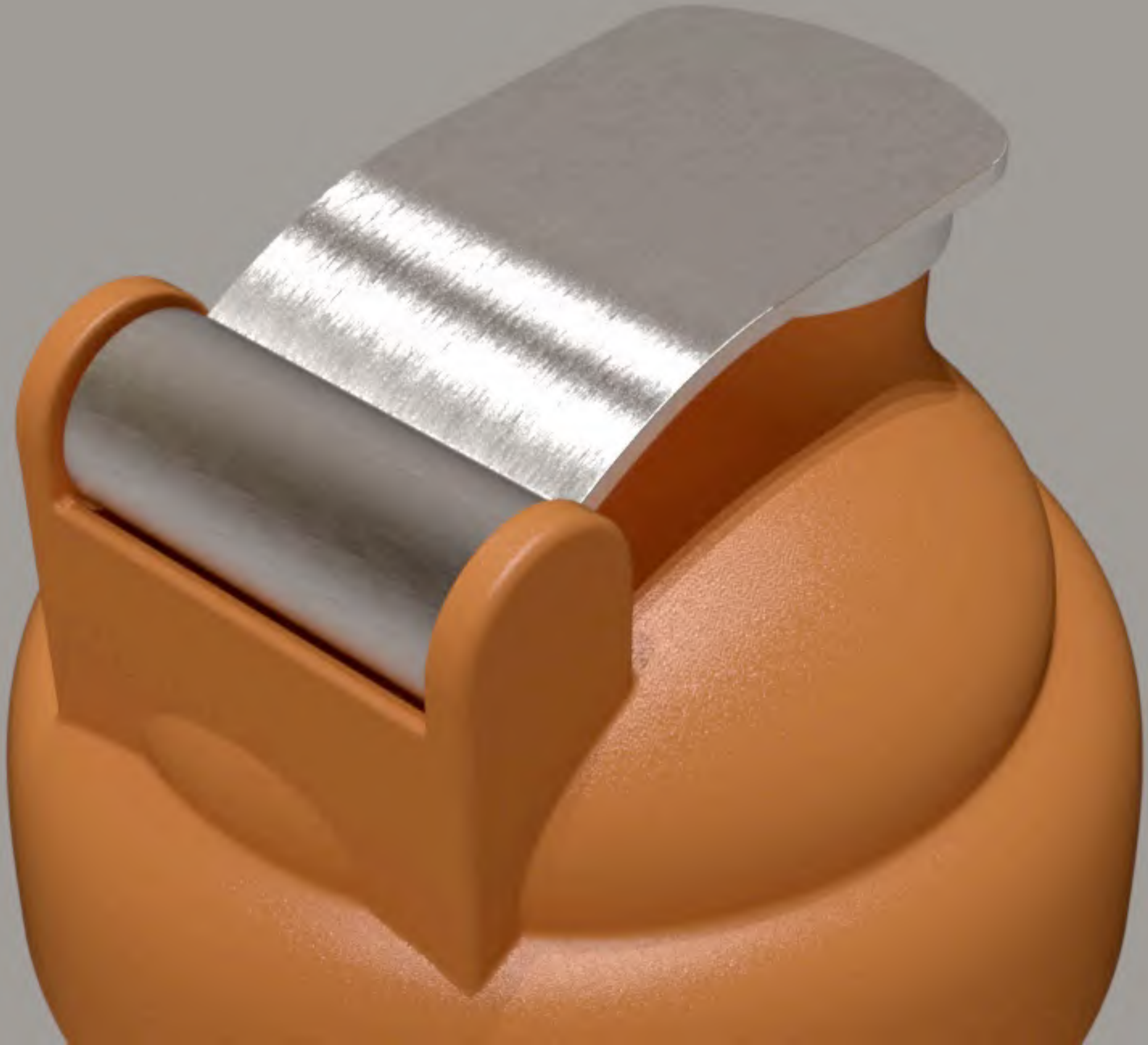
Acceleration structure

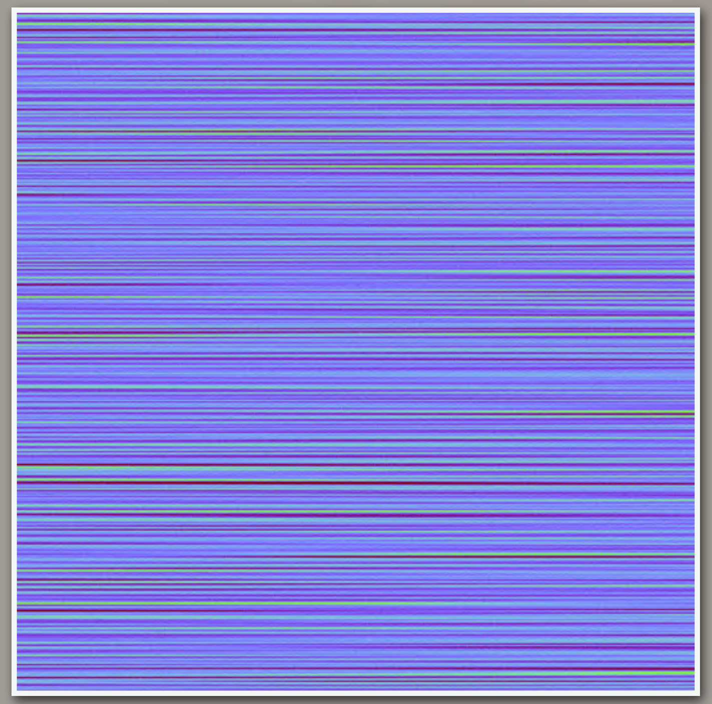


×

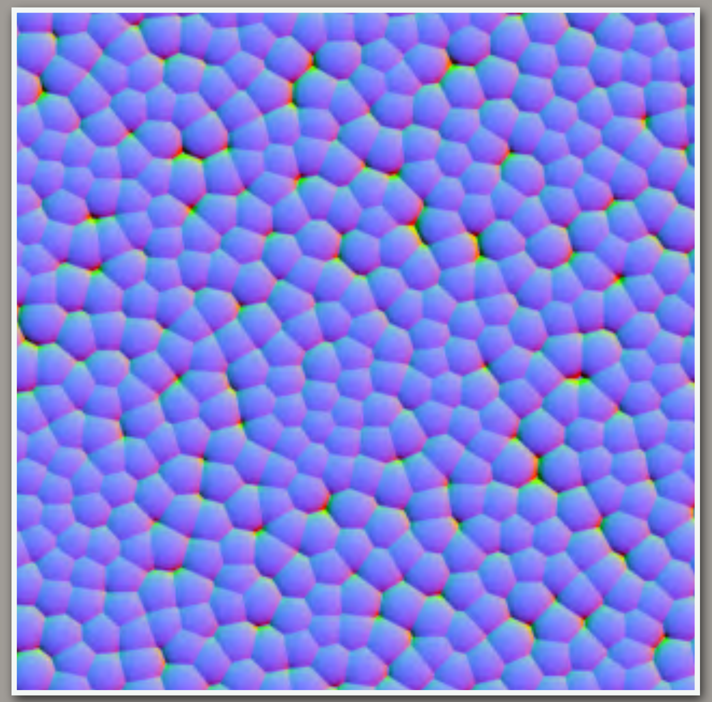


Blender



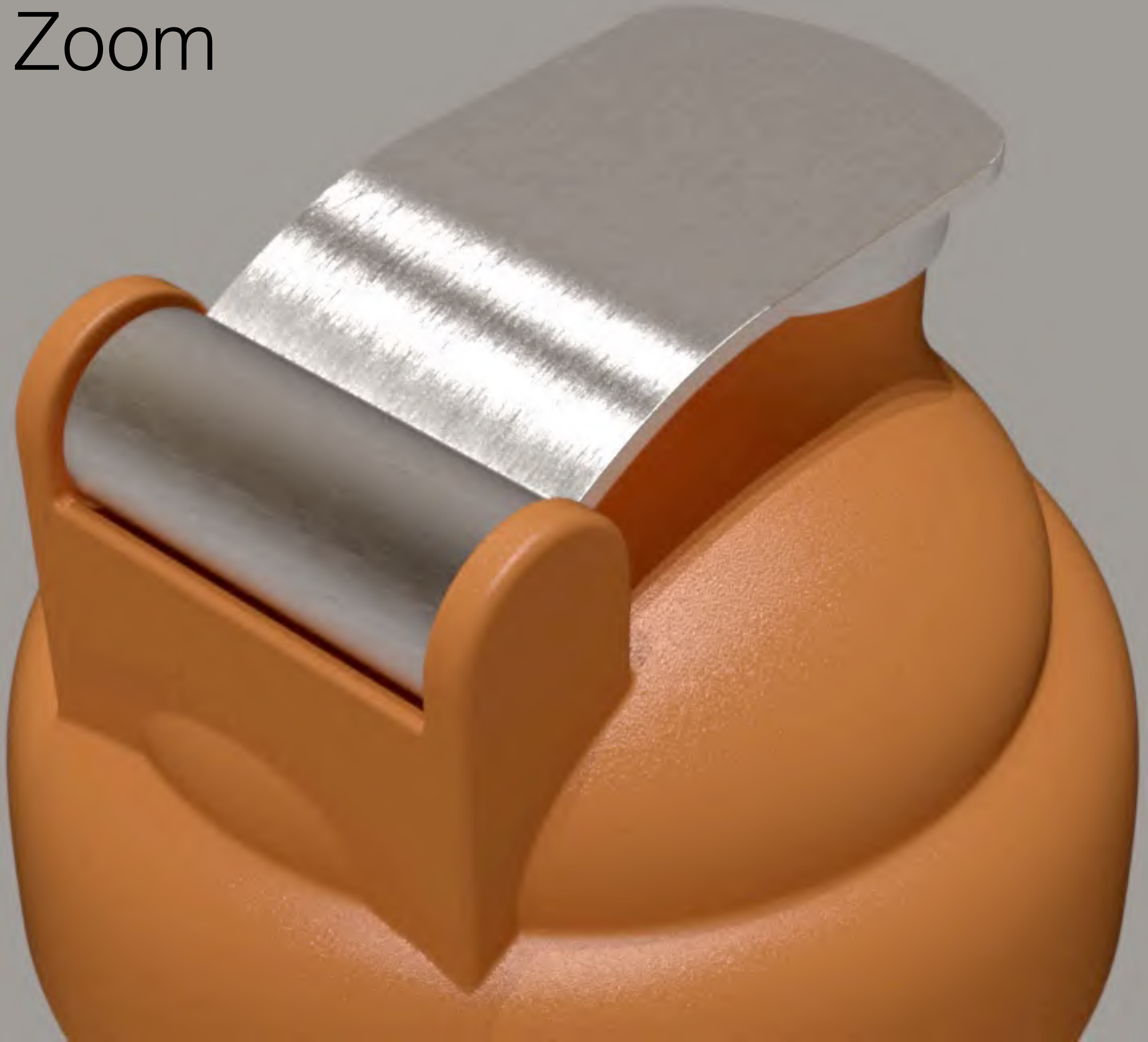


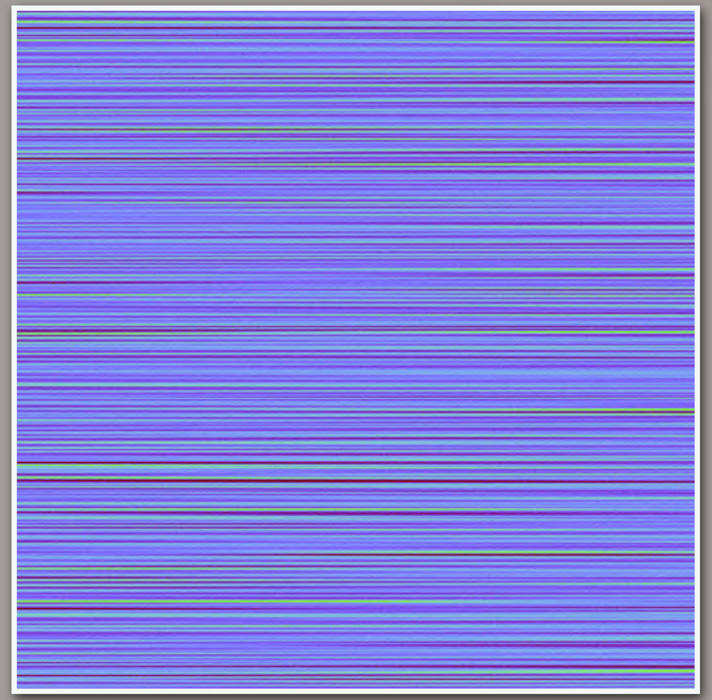
brushed metal



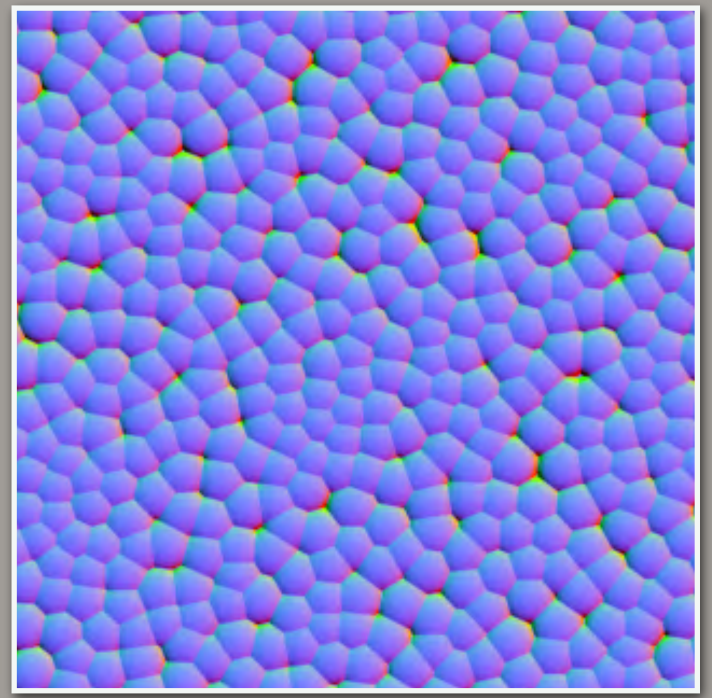
ellipsoid bumps

Blender: Zoom





brushed metal



ellipsoid bumps

Car (metallic paint + scratches)

scratched

flakes

Frame Render Times (Yan et al. [2016])

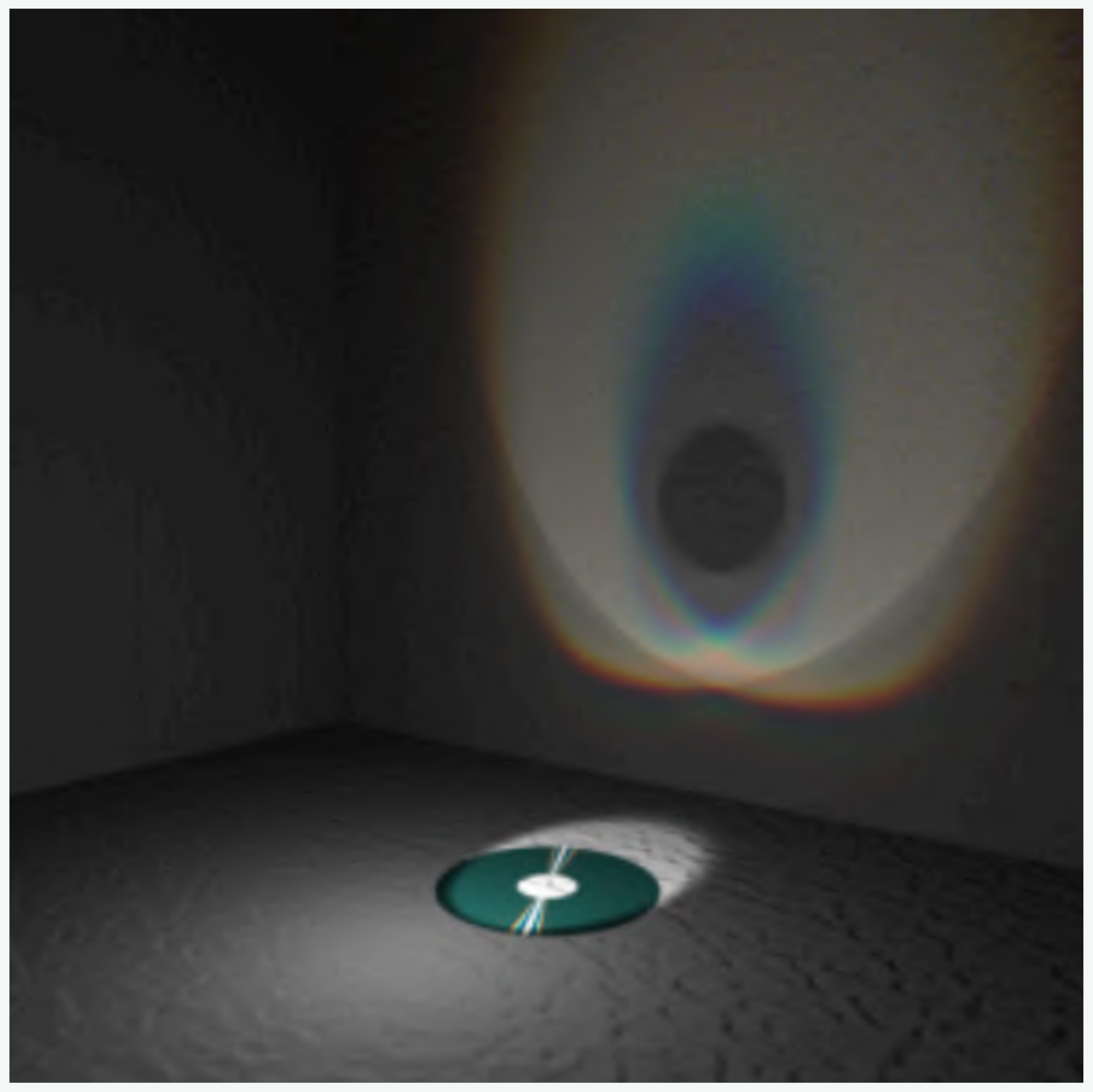
	Kettle (microfacet BRDF)	Kettle	Car door	Sofa & floor
Time (minutes)	1.8	2.6 (only 1.4x slower!)	6.8	7.6

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

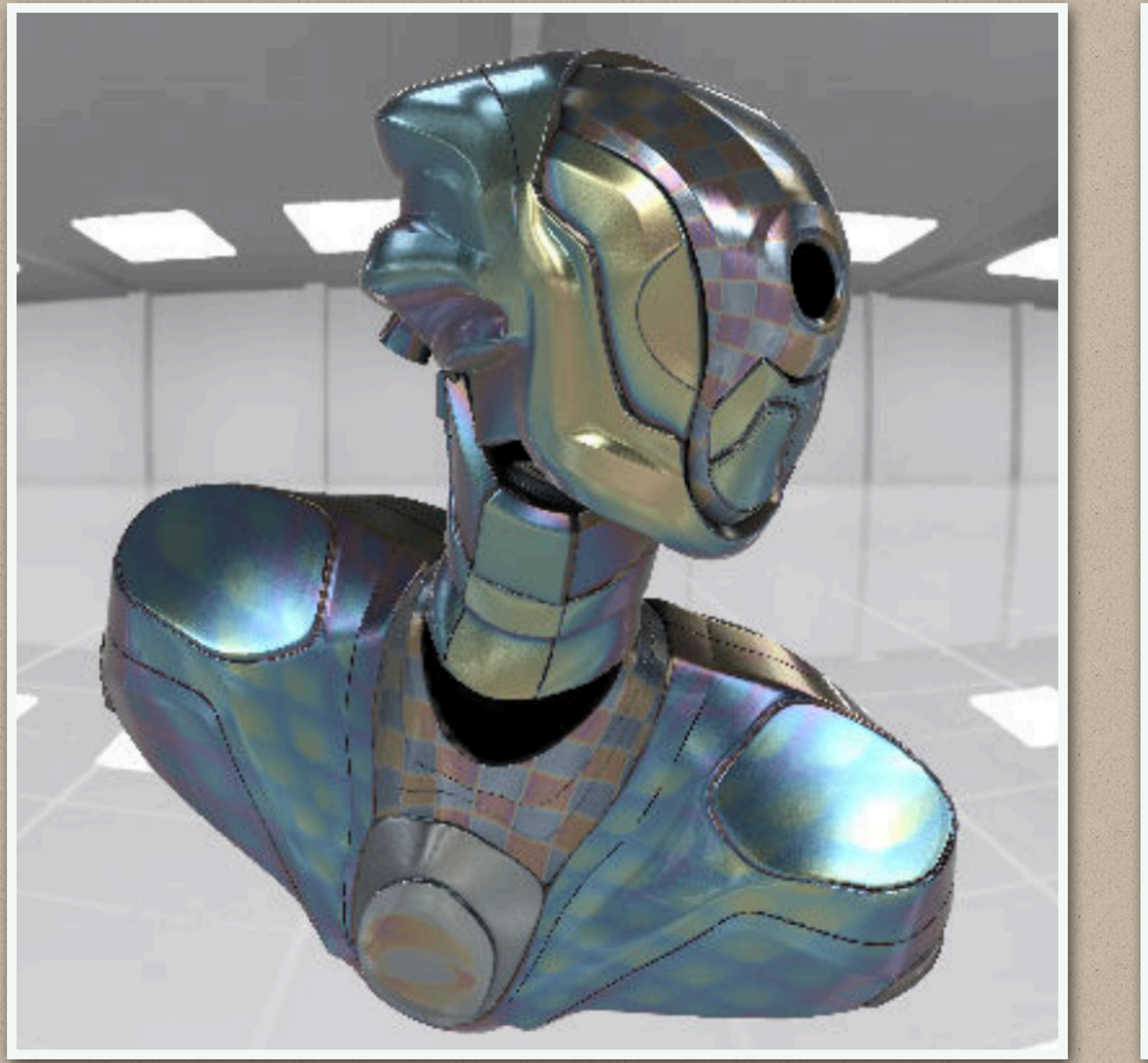
Rise of the Tomb Raider

2016 video game by Square Enix

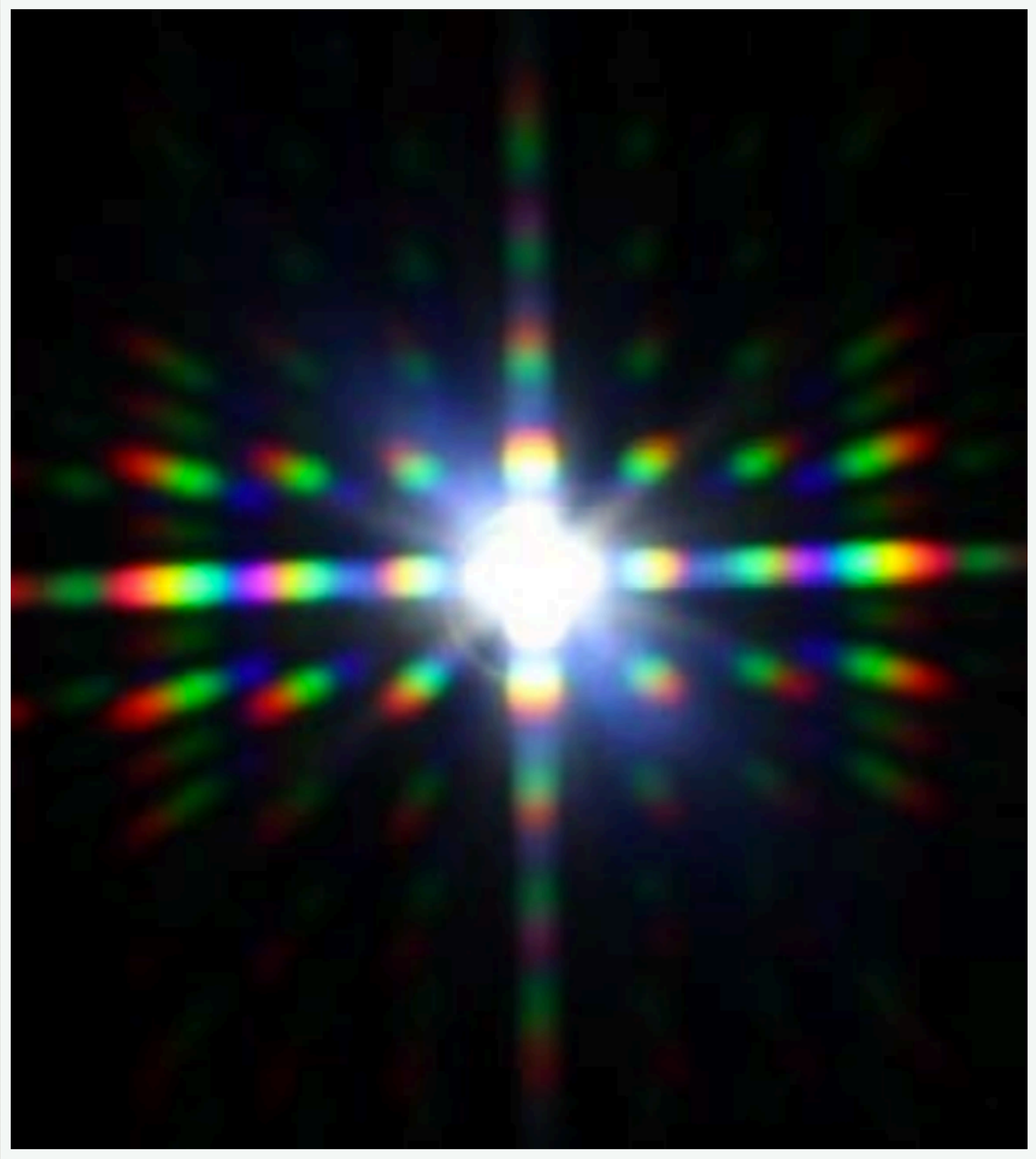
Recent Trend: Wave Optics



compact disk (CD)
[Cuypers 11]

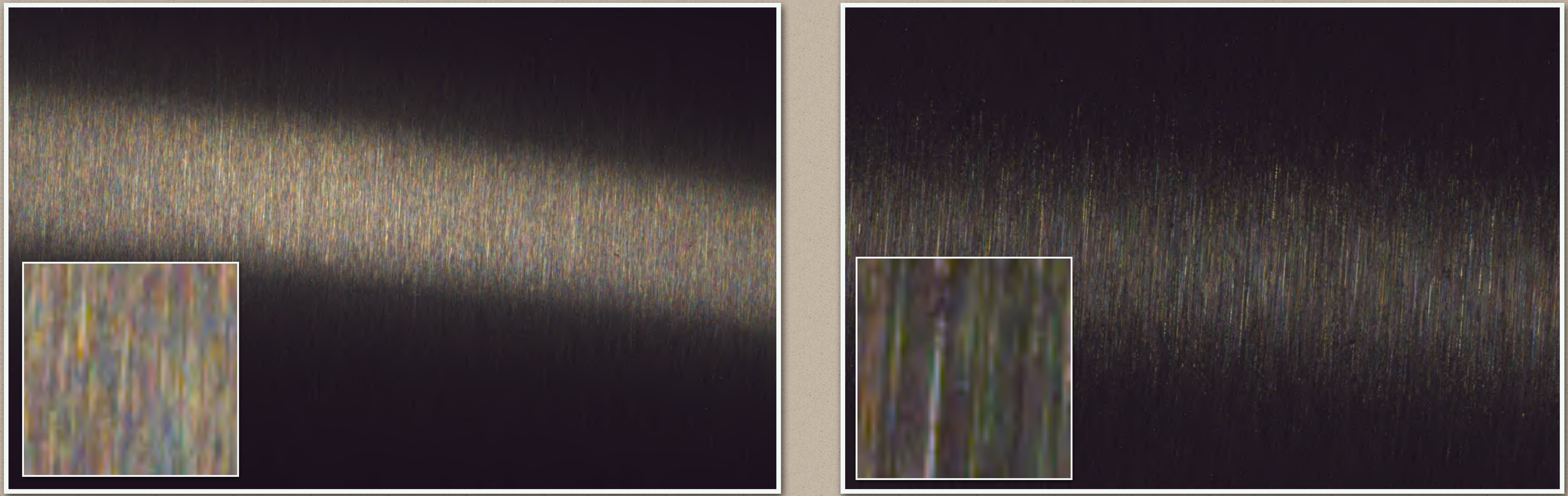


metallic film
[Laurent 17]



phone screen
[Toisoul 17]

Observations



photos of scratched metal

Observations

photo of a Macbook

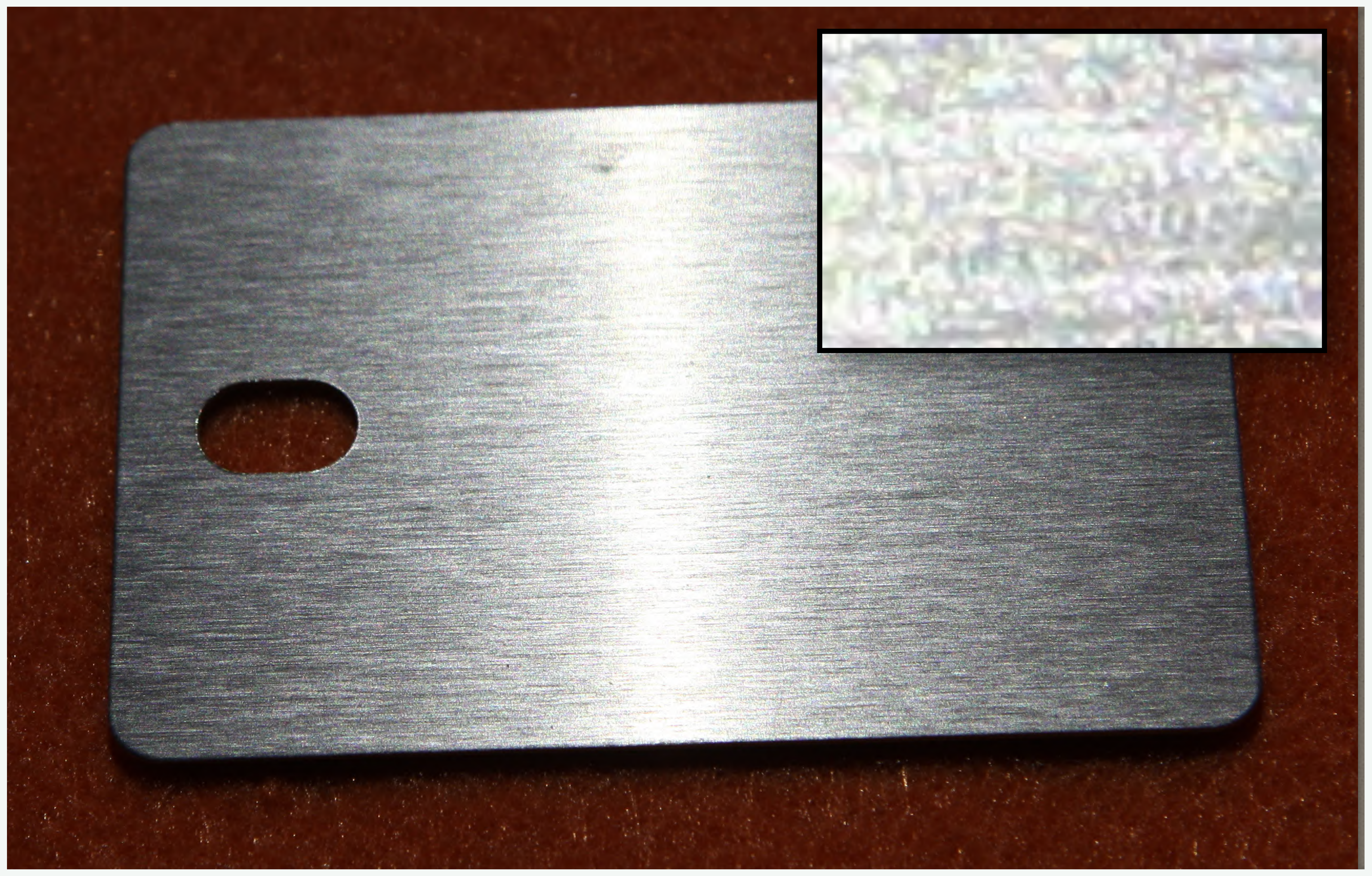
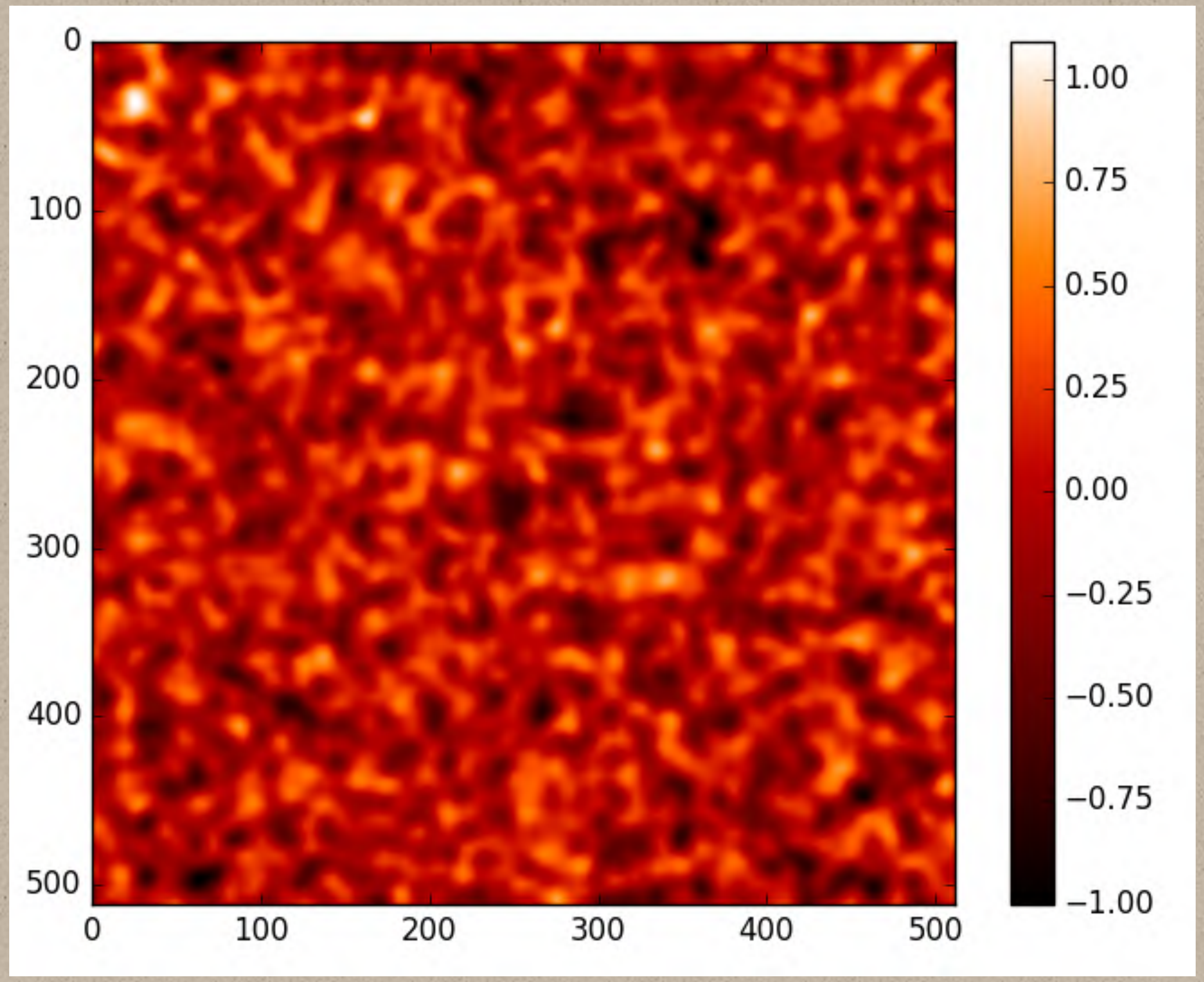
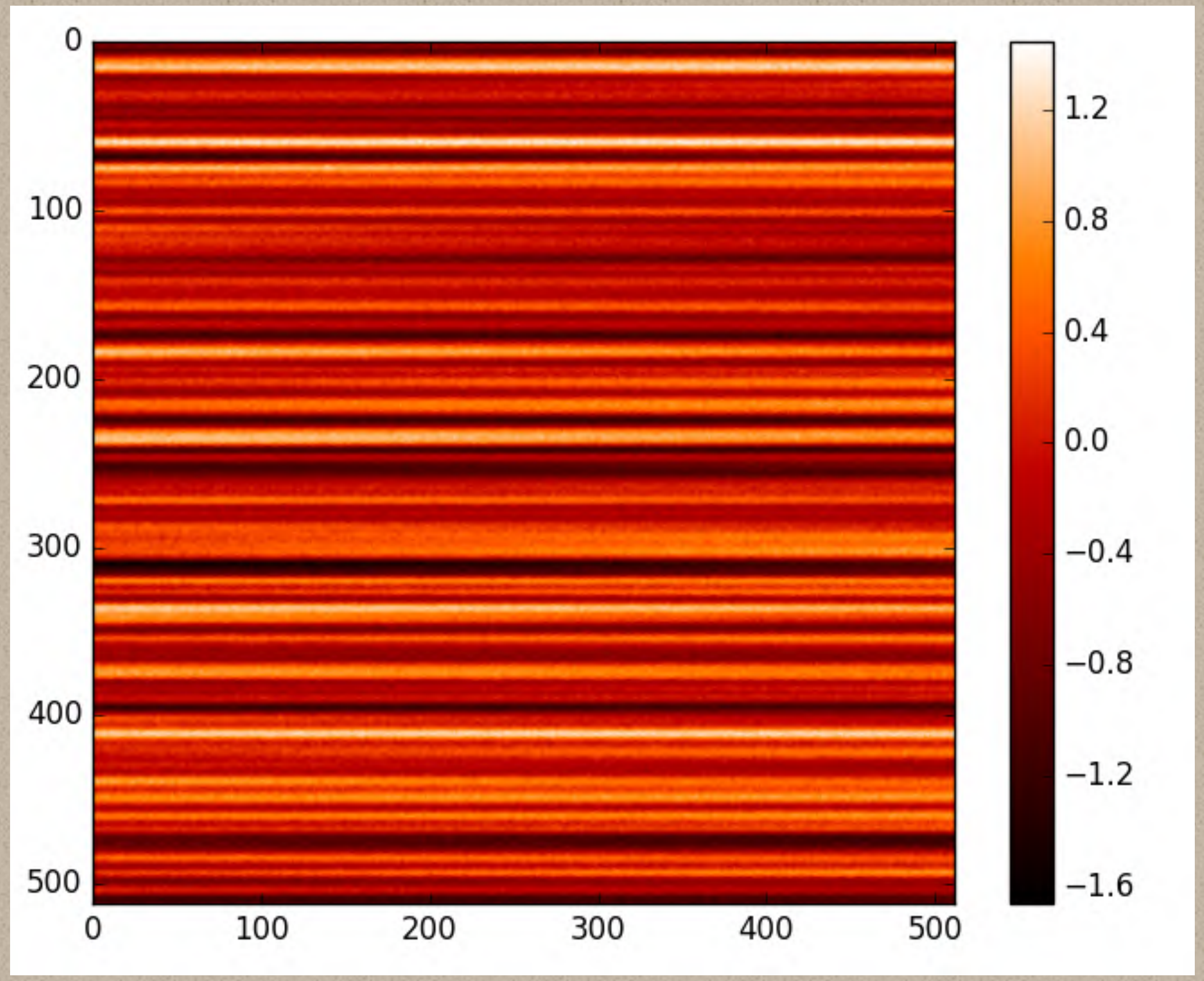


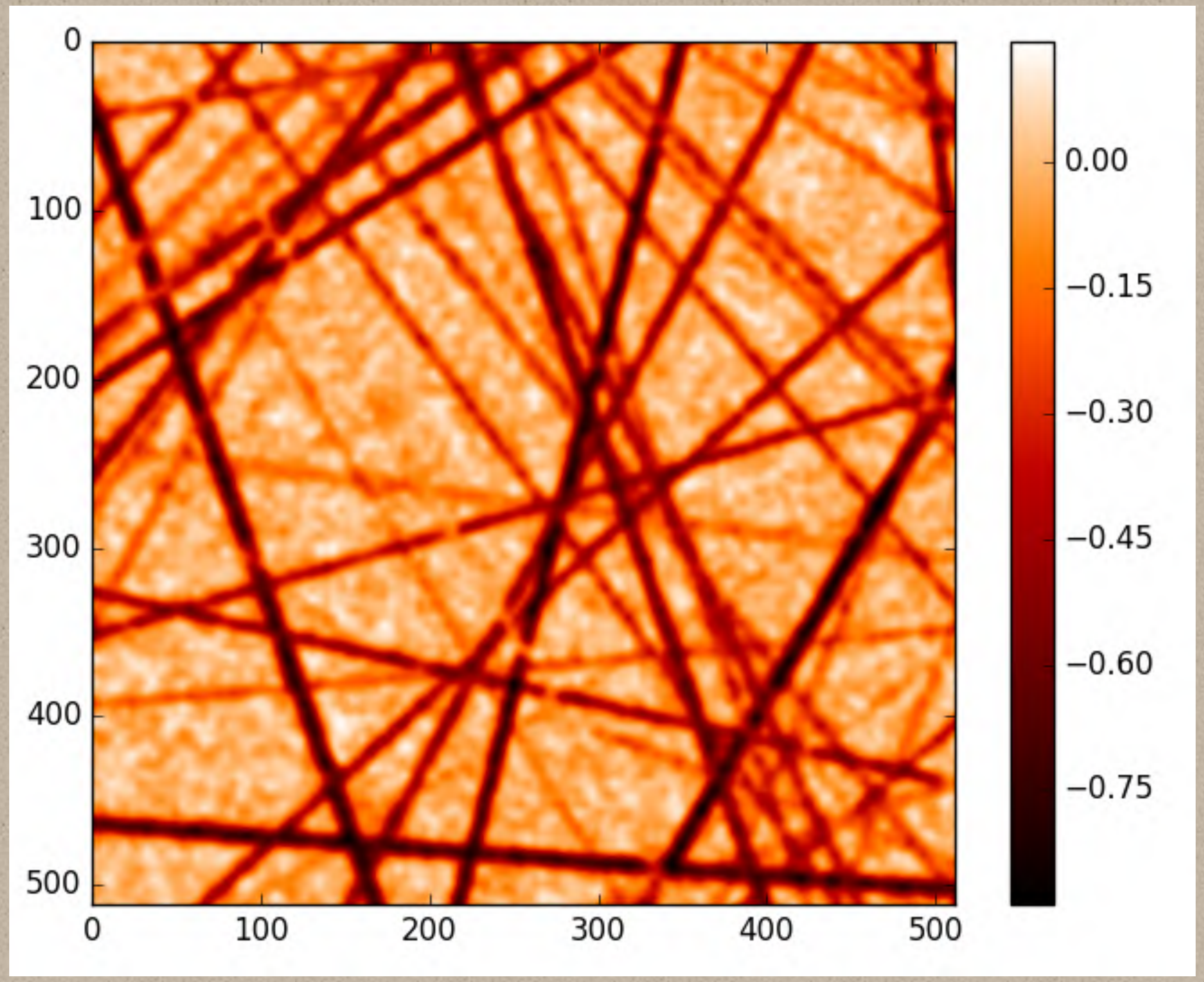
photo of an aluminum patch

Latest Work on Wave Optics (accepted)

Heightfields





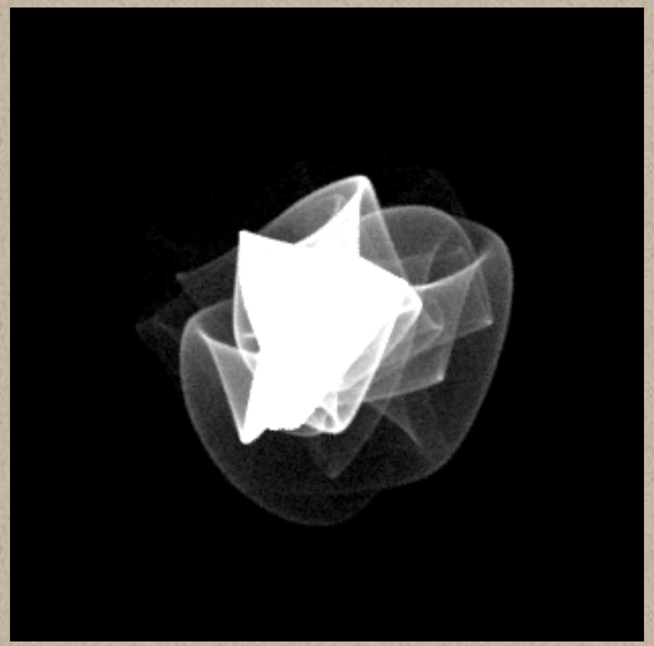


isotropic

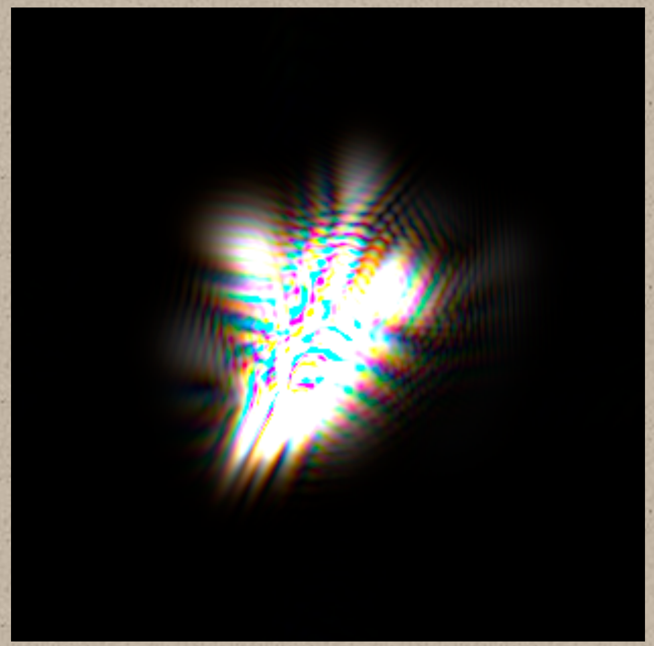
brushed

scratched

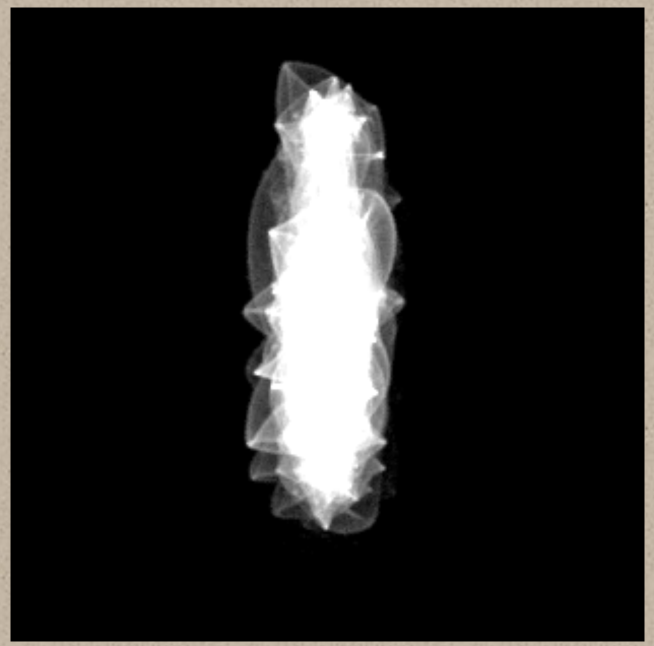
BRDFs



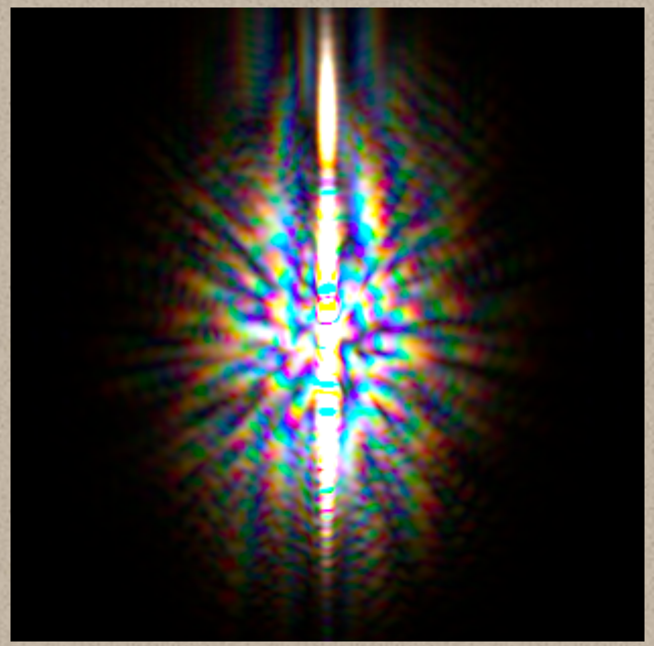
geometric



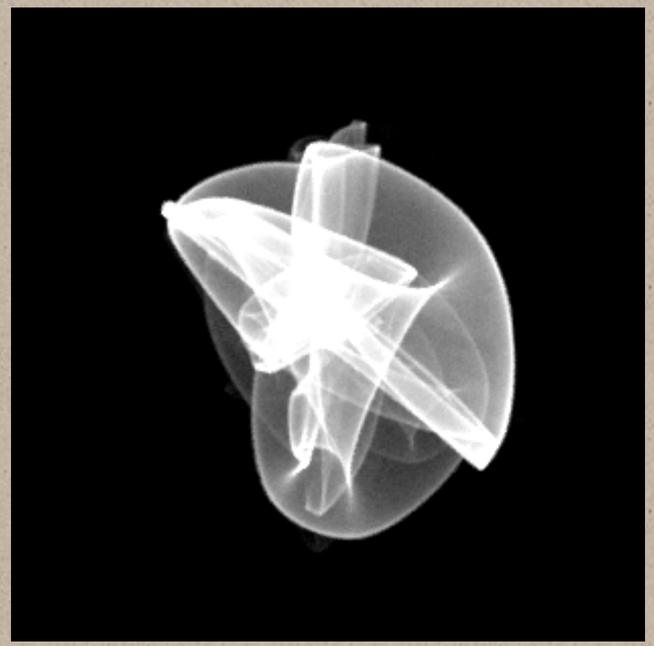
wave



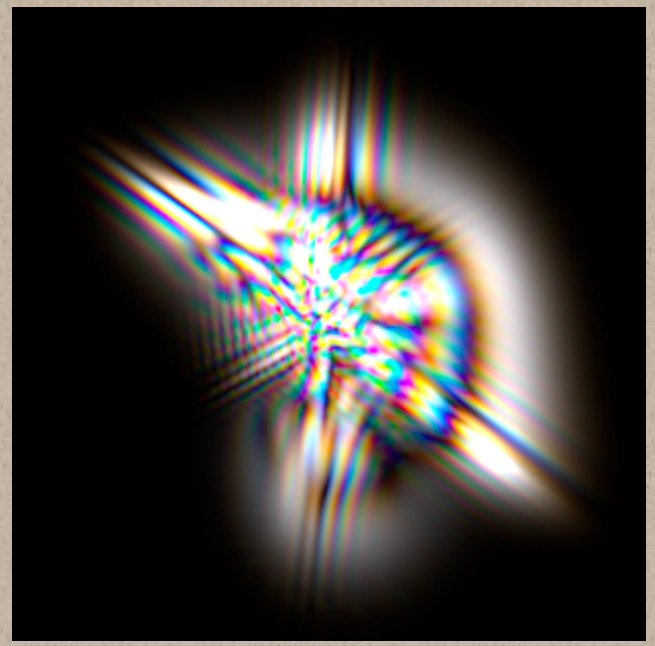
geometric



wave



geometric



wave

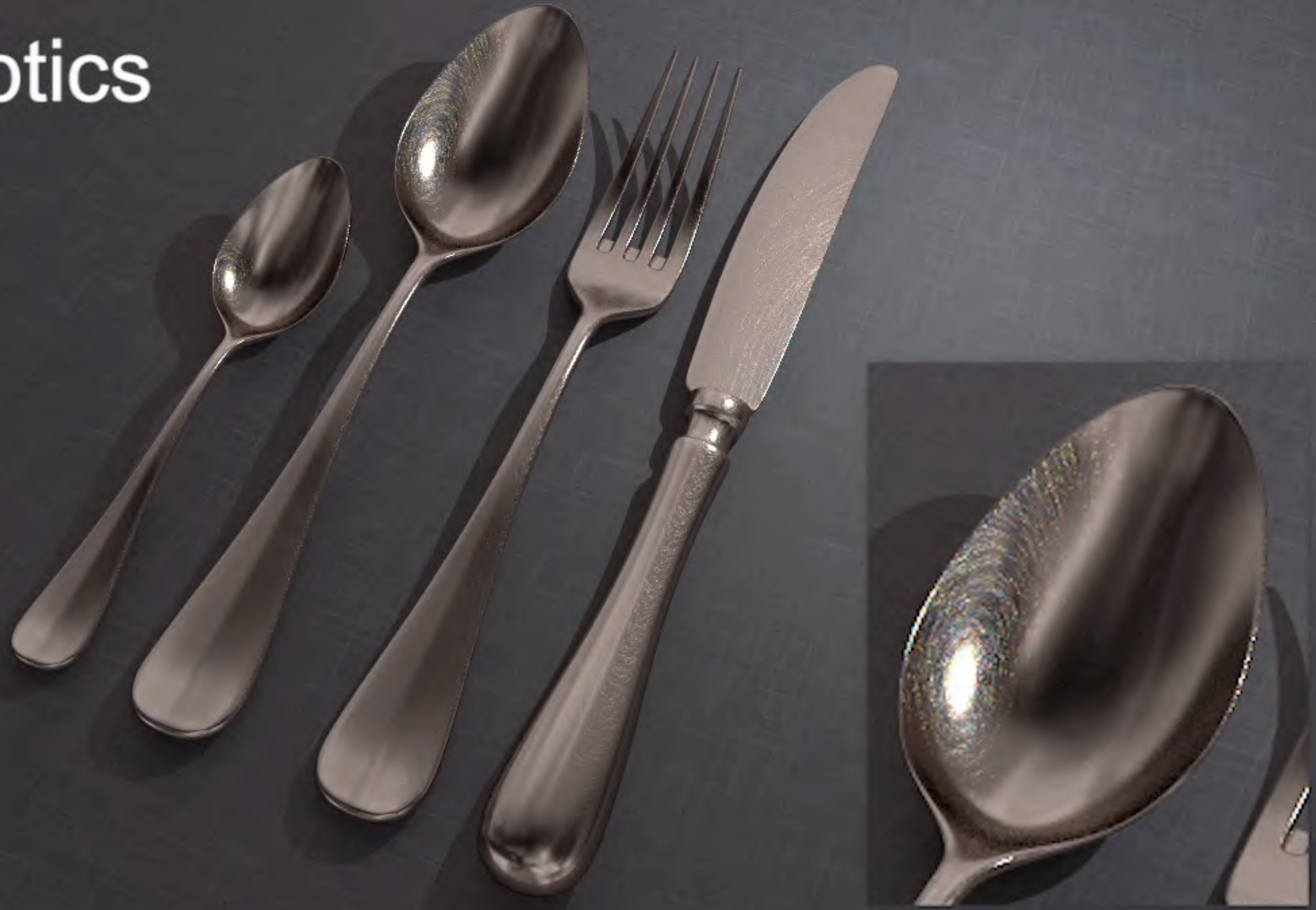
` 1 2 3 4 5 6 7 8 9 0 - = delete
tab Q W E R T Y U I O P { [}] | \
caps lock A S D F G H J K L : ; " ' enter return
shift Z X C V B N M < , > . ? / shift
fn control alt option ⌘ command ⌘ command alt option ▲ ◀ ▼ ▶

Rendered using wave optics

fn
control
option
command

fn
control
option
⌘
command

Wave optics



Part II: Detailed Appearance Modeling

detailed rendering

detailed appearance modeling

real-time ray tracing

The Appearance of Natural Materials

Hair Reflectance Models

- Actively developing

[Marschner 03]

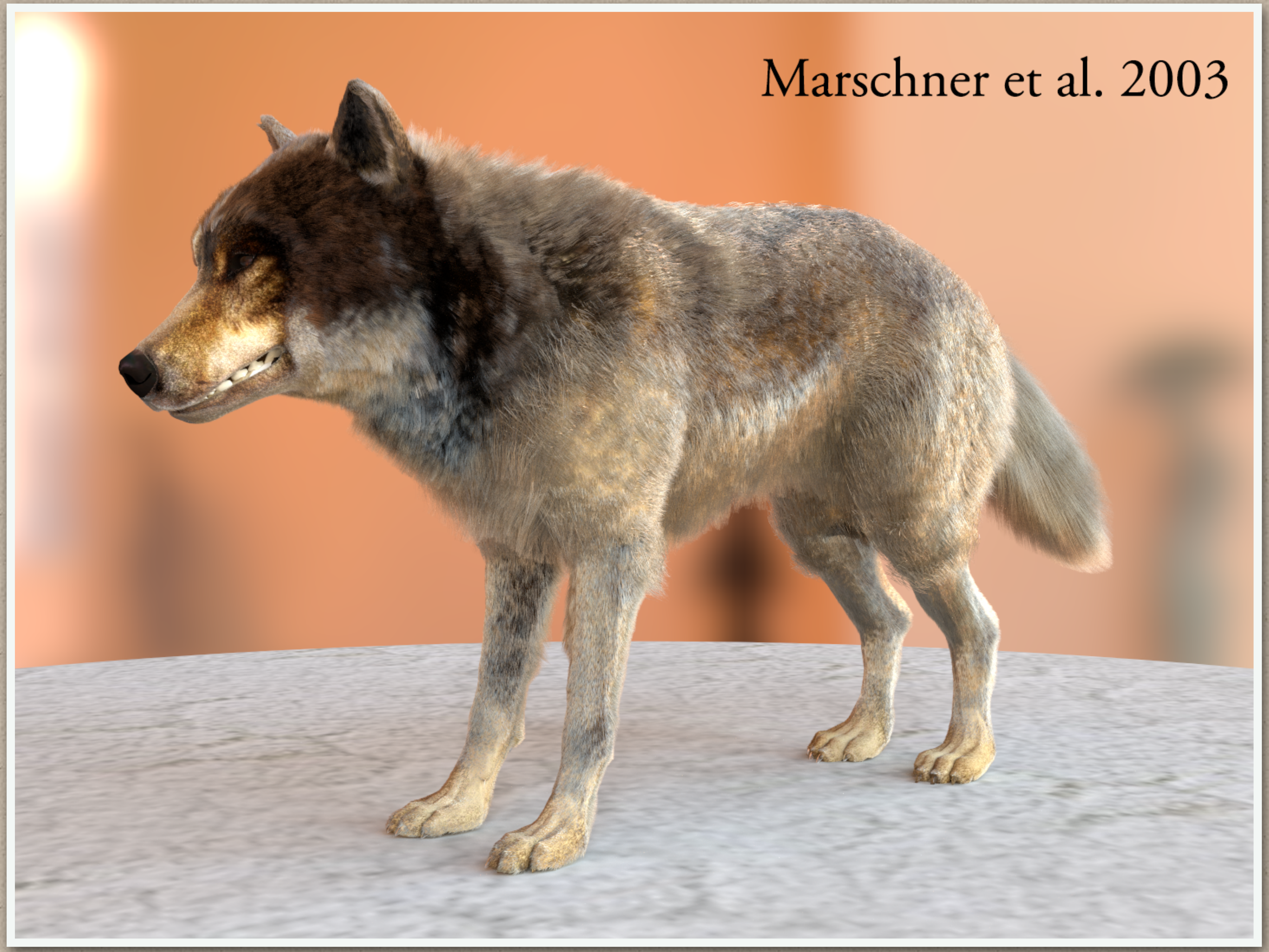
[Zinke 07]

[d'Eon 11]

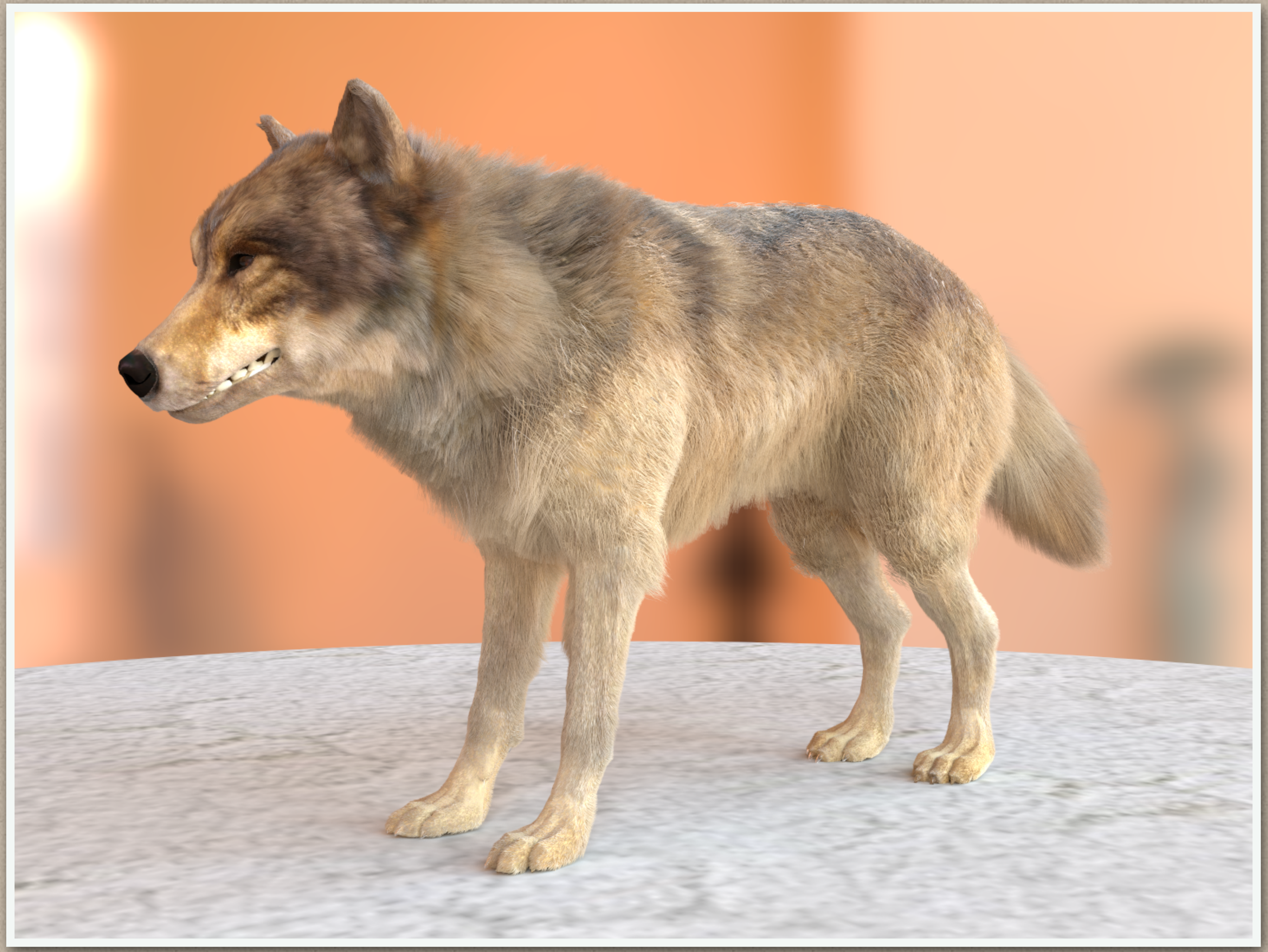
[Chiang 16]

Fur Reflectance — As Human Hair

- Cannot represent diffusive and saturated appearance

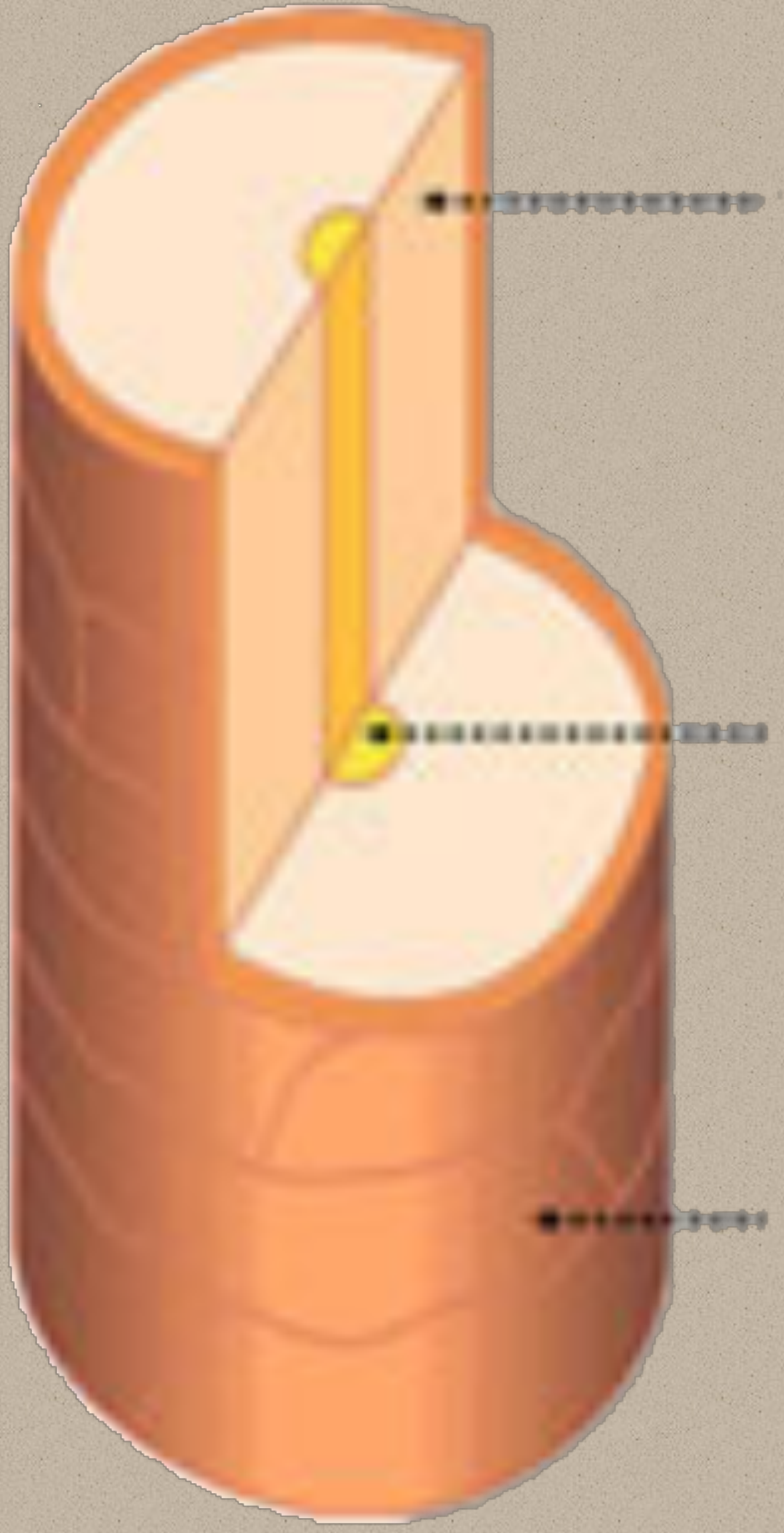


Rendered as human hair
[Marschner 03]



Rendered as animal fur
[Yan 15]

Main Difference — Medulla



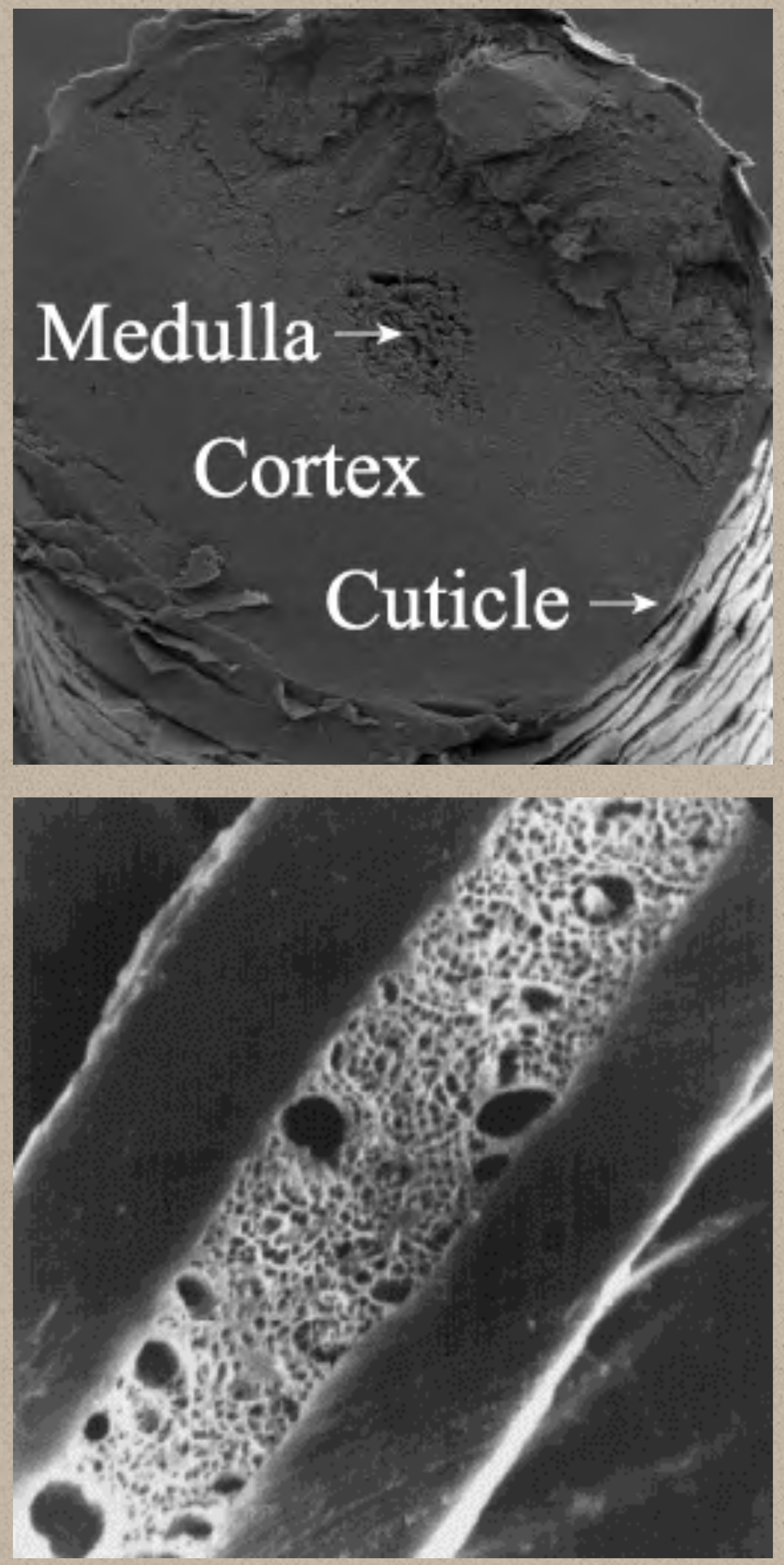
Cortex

- Contains pigments
- Absorbs light

Medulla

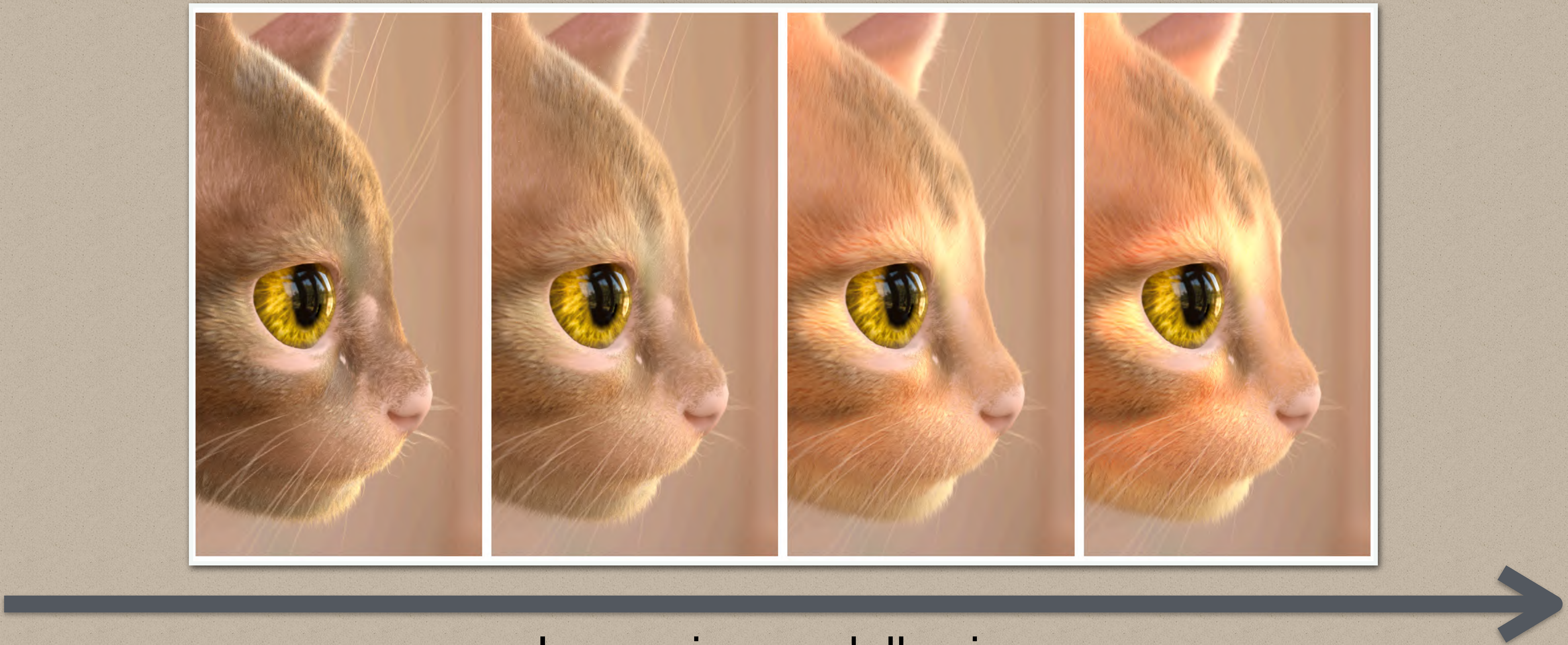
- Complex structure
- **Scatters light**

Cuticle



Microscopic images
(Top: human, Bottom: Cougar)

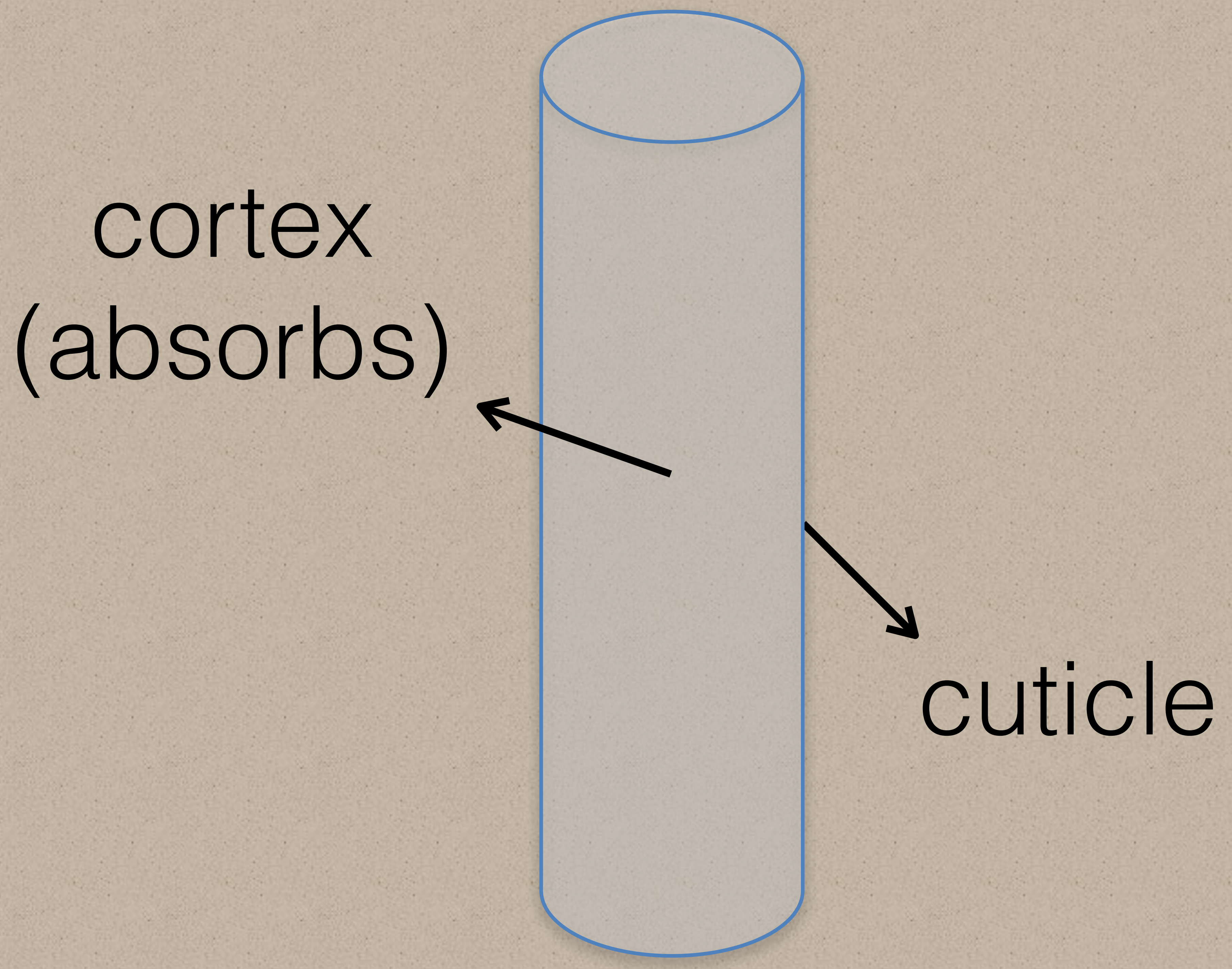
Importance of Medulla



Increasing medulla size

Hair Reflectance Model

- Glass-like cylinder

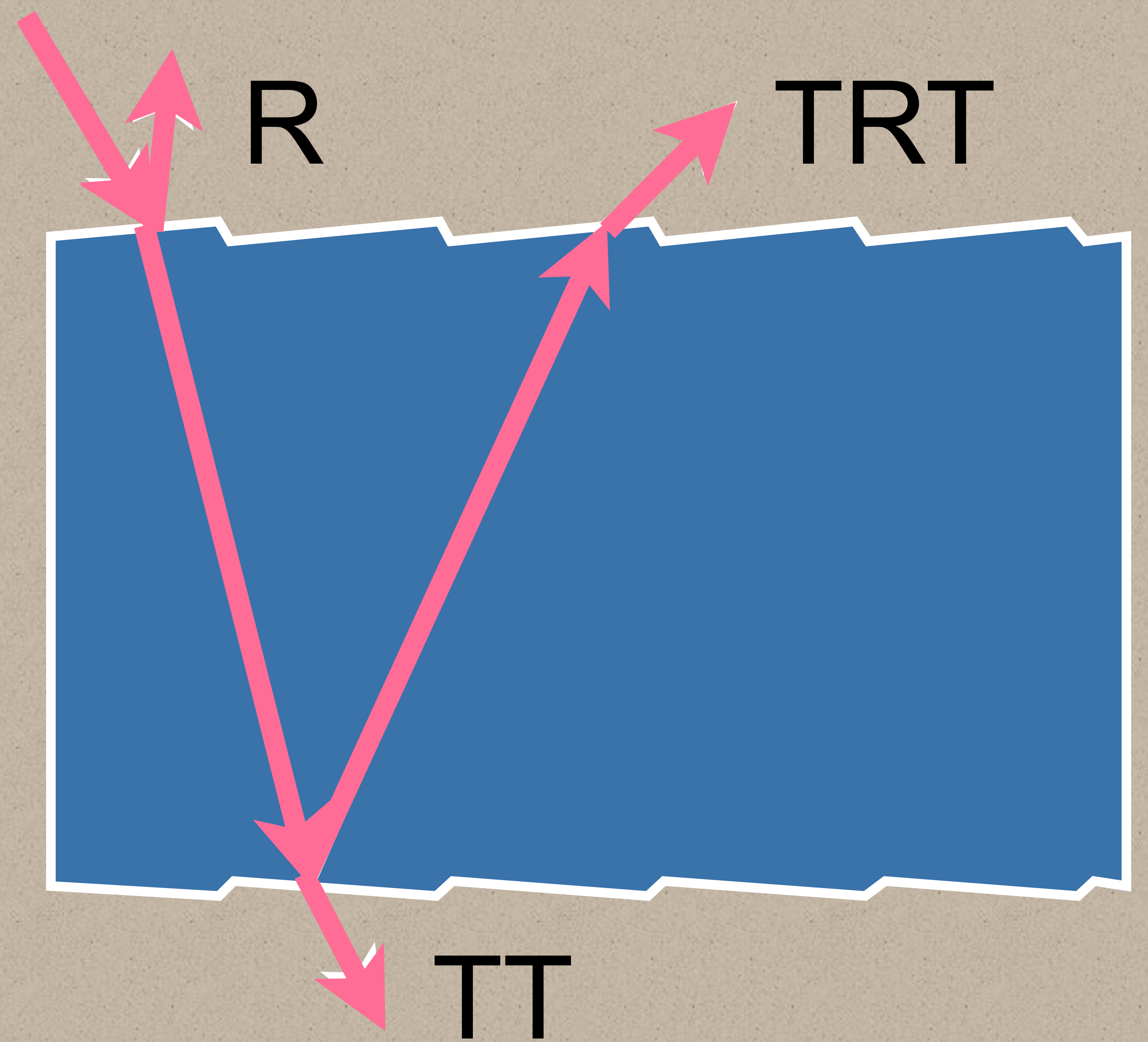


[Marschner 03]

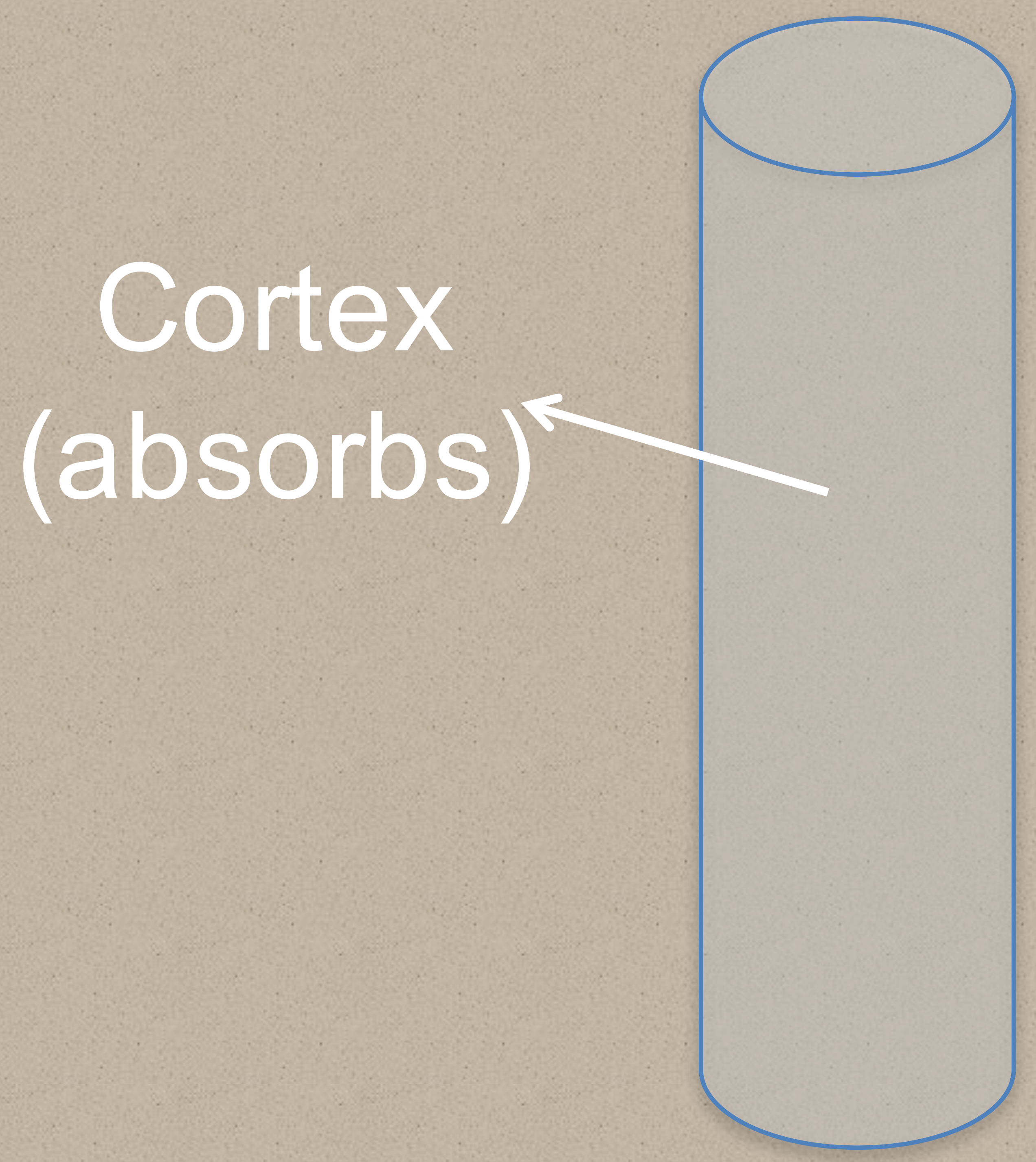
- 3 types of light interactions:

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

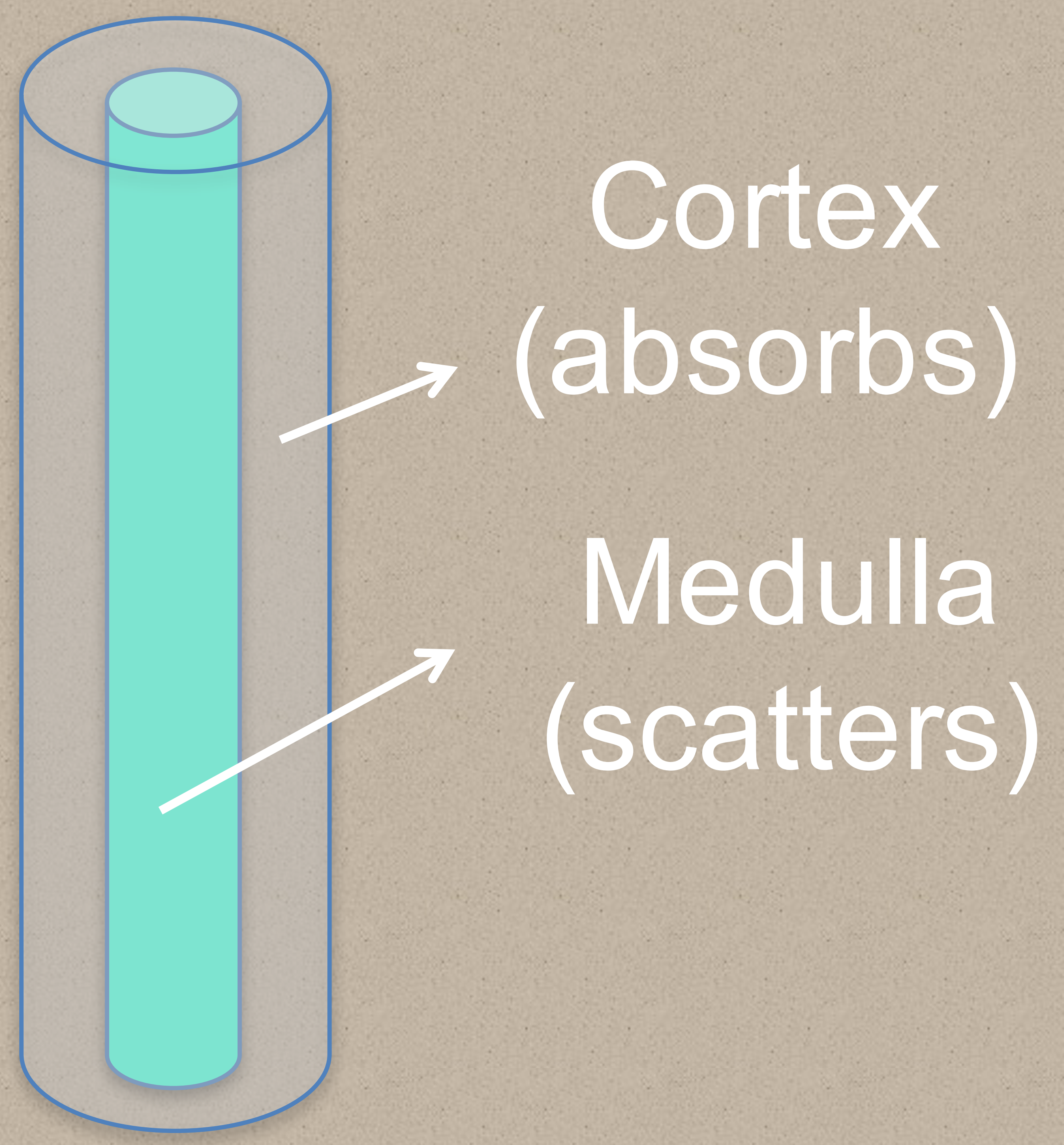
(R: reflection, T: transmission)



Fur Reflectance Model [Yan 15, 17]



Hair model
[Marschner 03]

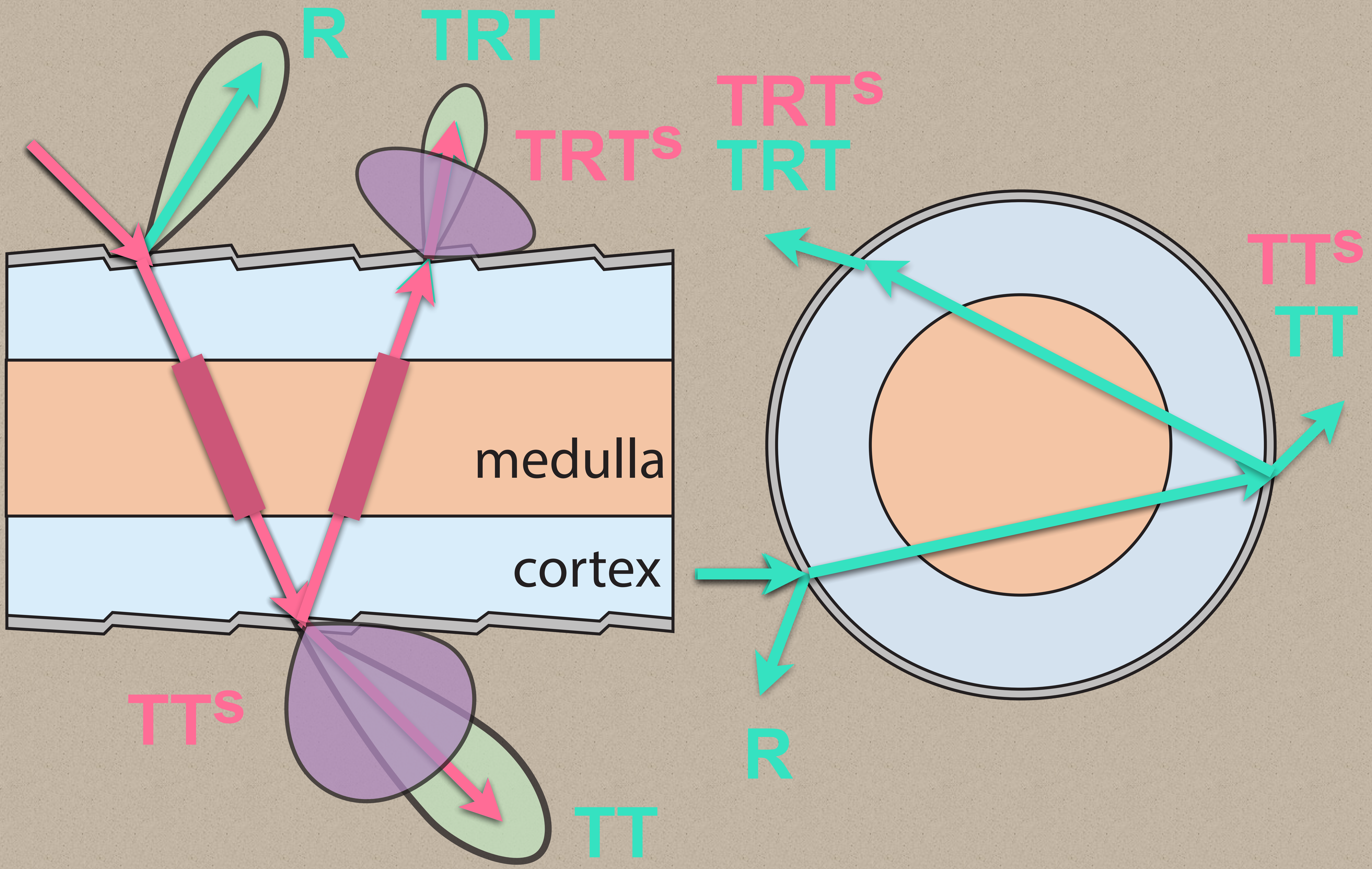


Double cylinder model
[Yan 15, 17]

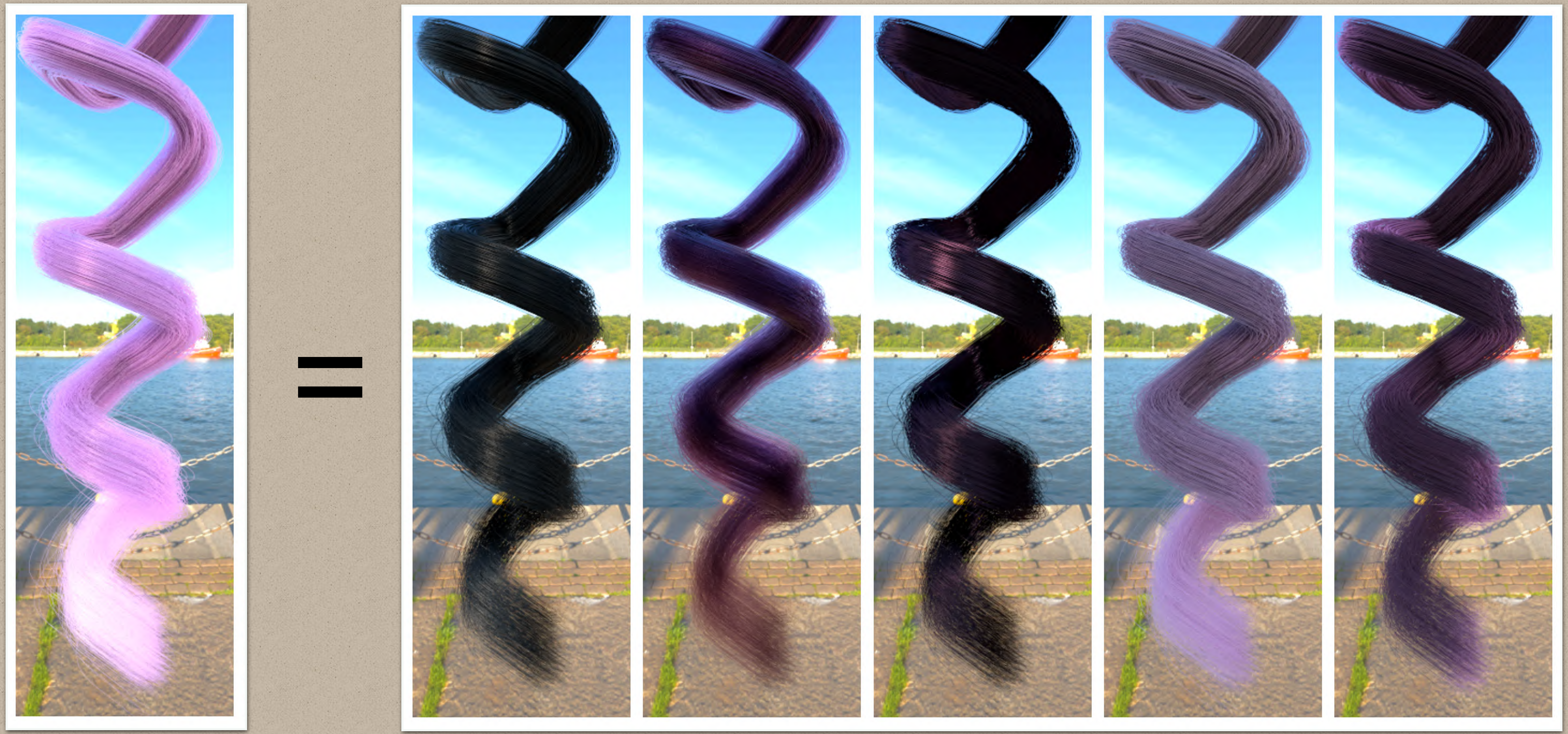
Fur Reflectance Model — Lobes

- Unscattered lobes
  - R
  - TT
  - TRT

- Scattered lobes
  - TT^s
  - TRT^s

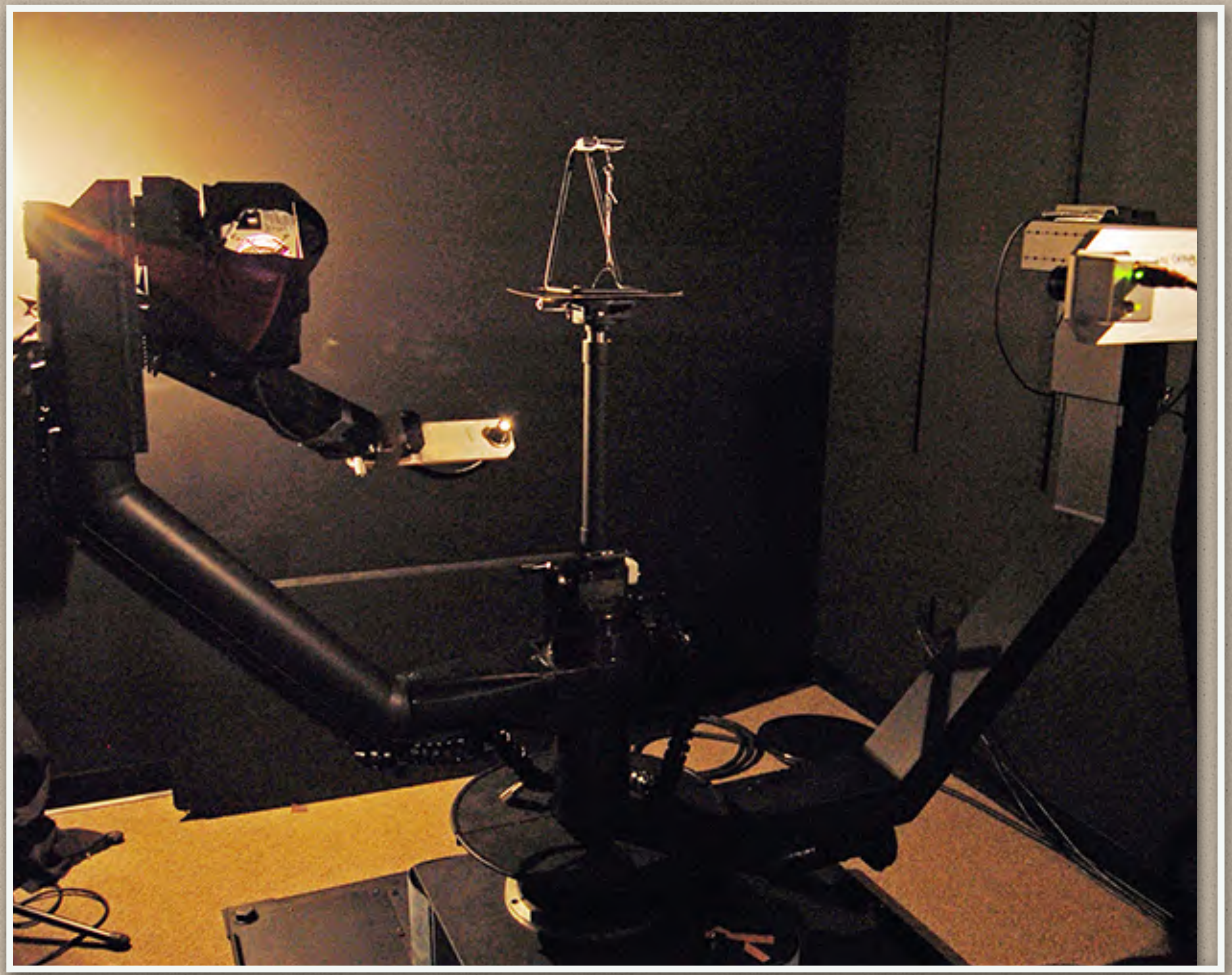


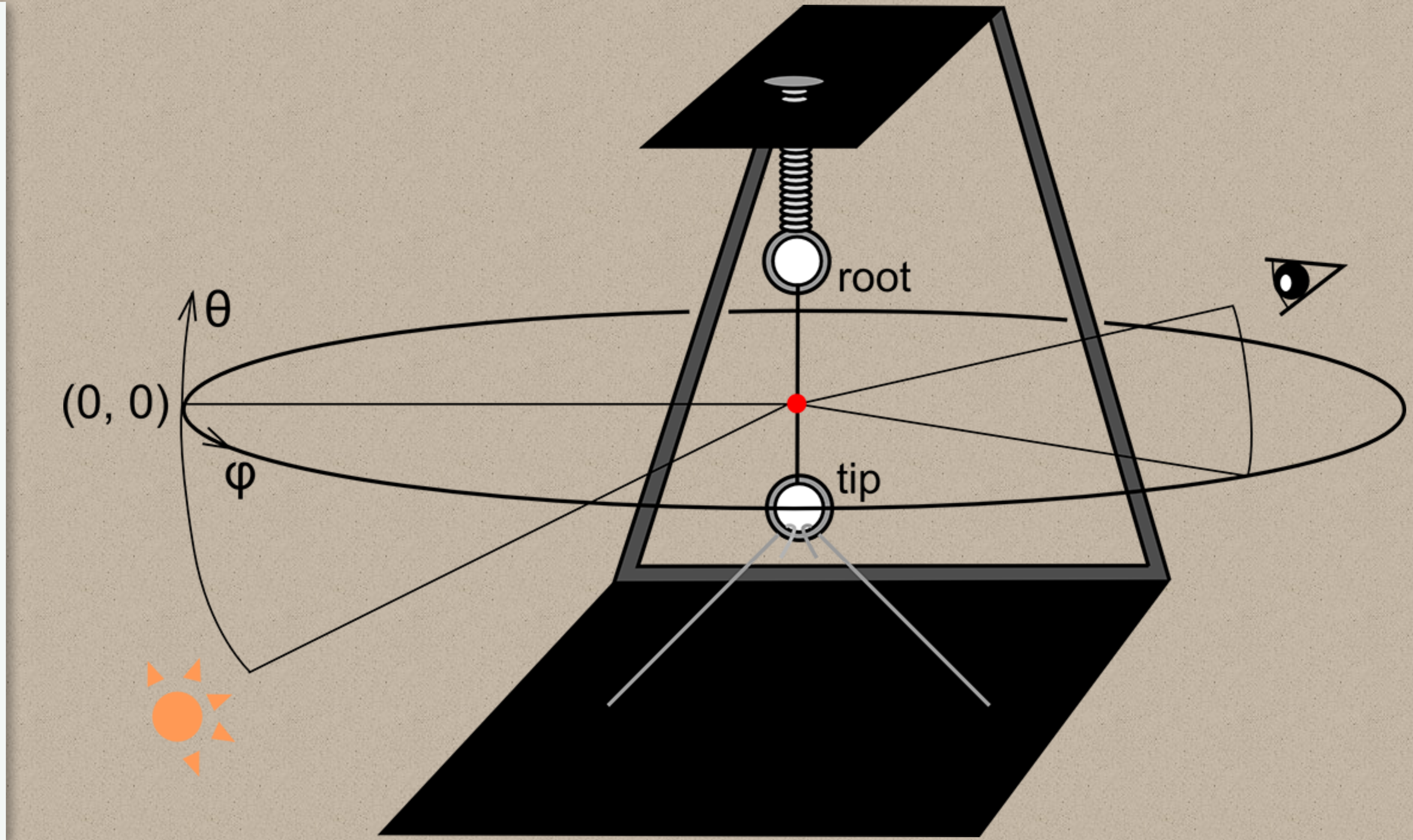
Visualizing Lobes



All = R TT TRT **TT^{s}** **TRT^{s}**

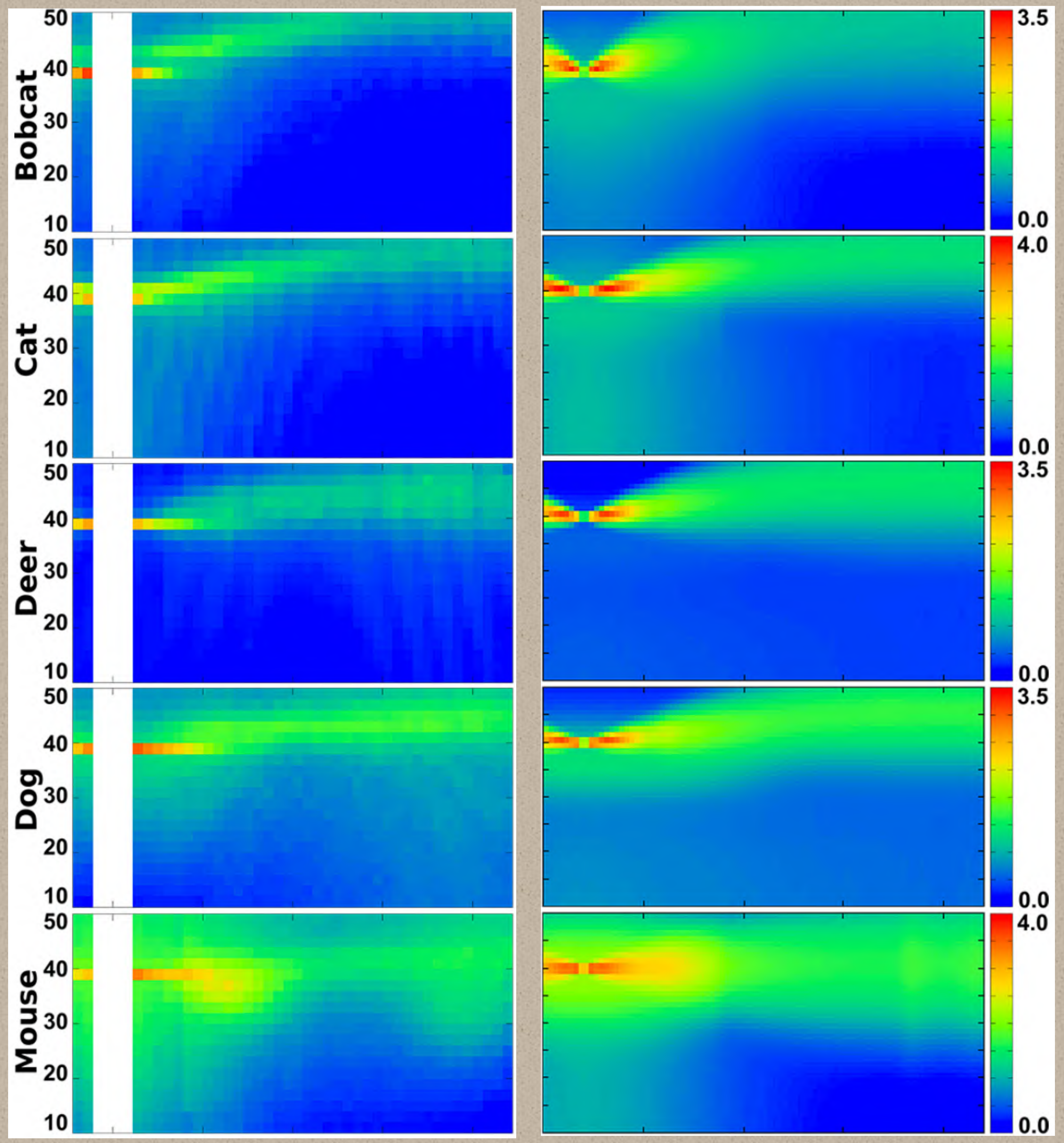
Measurement — Spherical Gantry



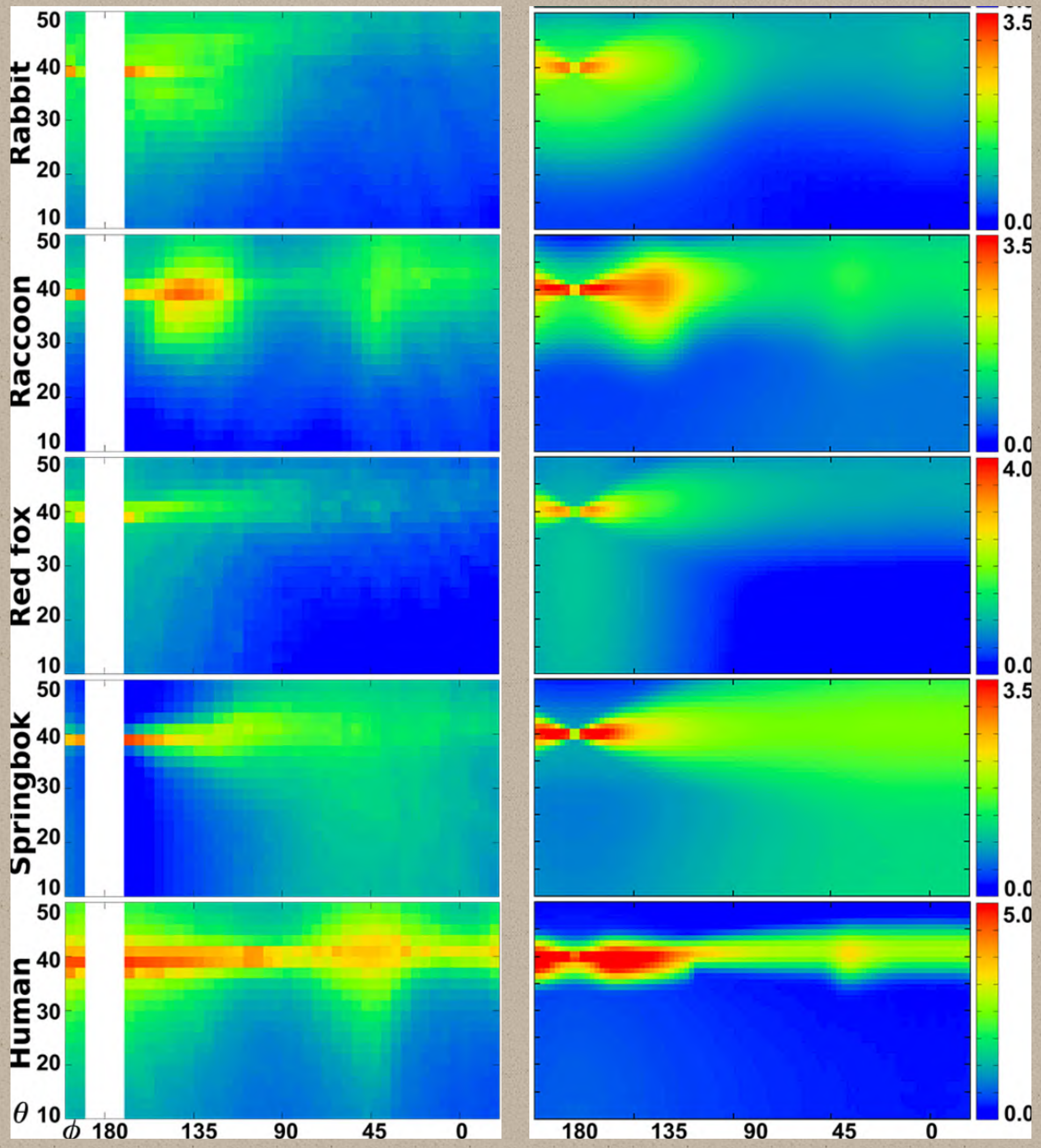


Validation

Measured Ours



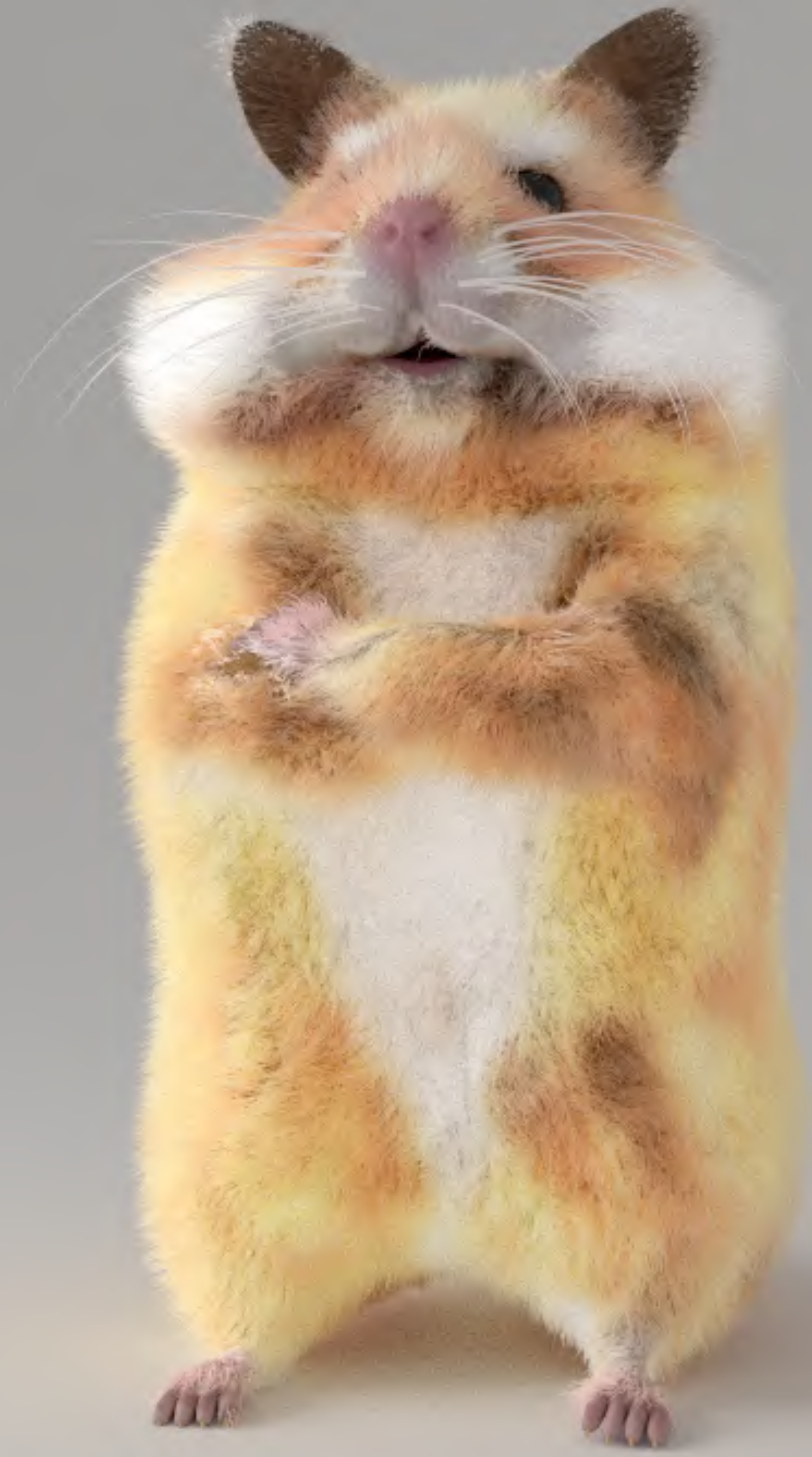
Measured Ours



600,000 fur fibers

1024 samples / pixel

36.9 min / frame



SIGGRAPH 2017

technical paper trailer

[Yan et al. 15, 17]

260,000 fur fibers
1024 samples / pixel
14.1 min / frame

[Yan et al.17]

War for the Planet of the Apes, 2017 movie

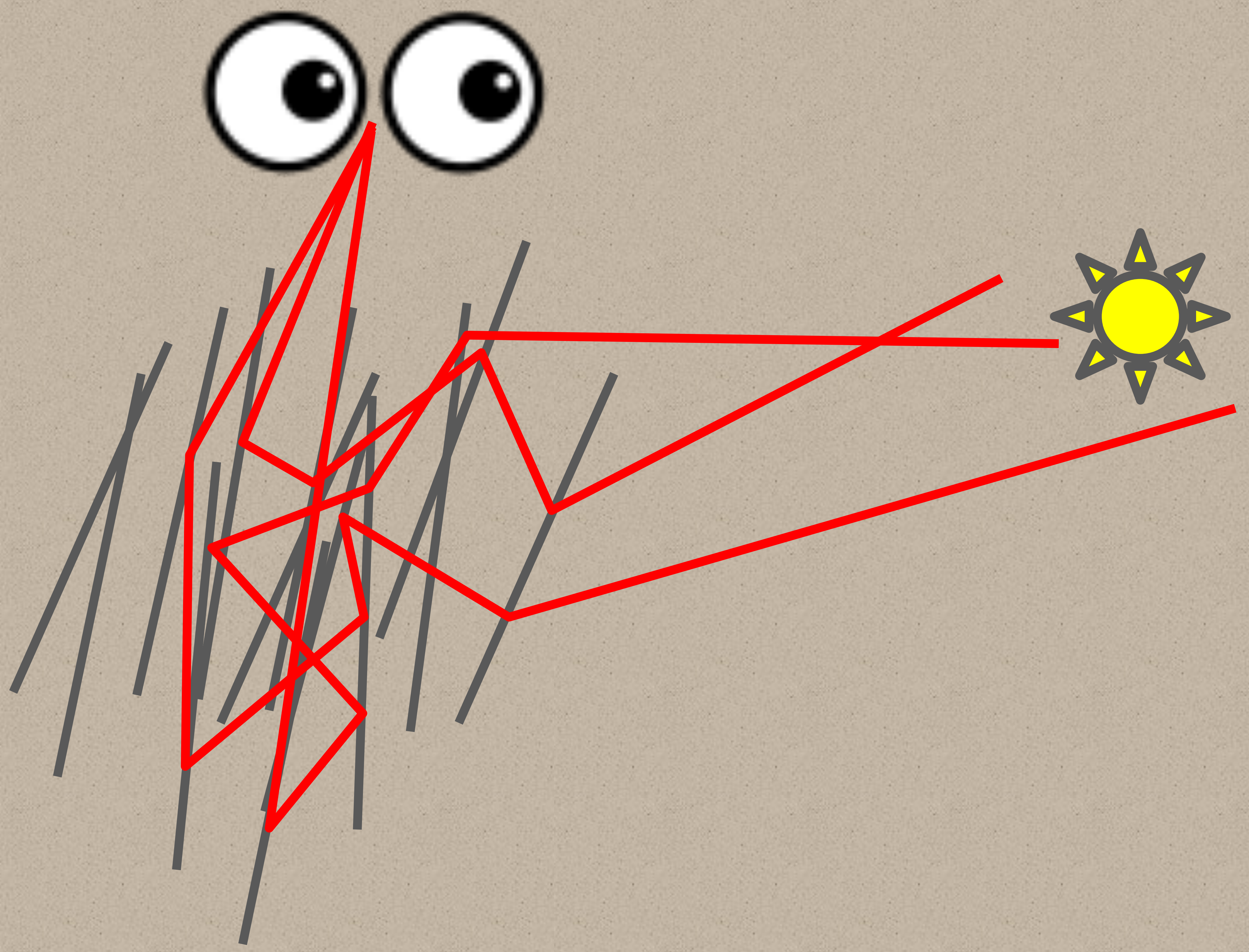
2018 Oscar Nominee for Best Visual Effects

Further speed up [Yan 17]

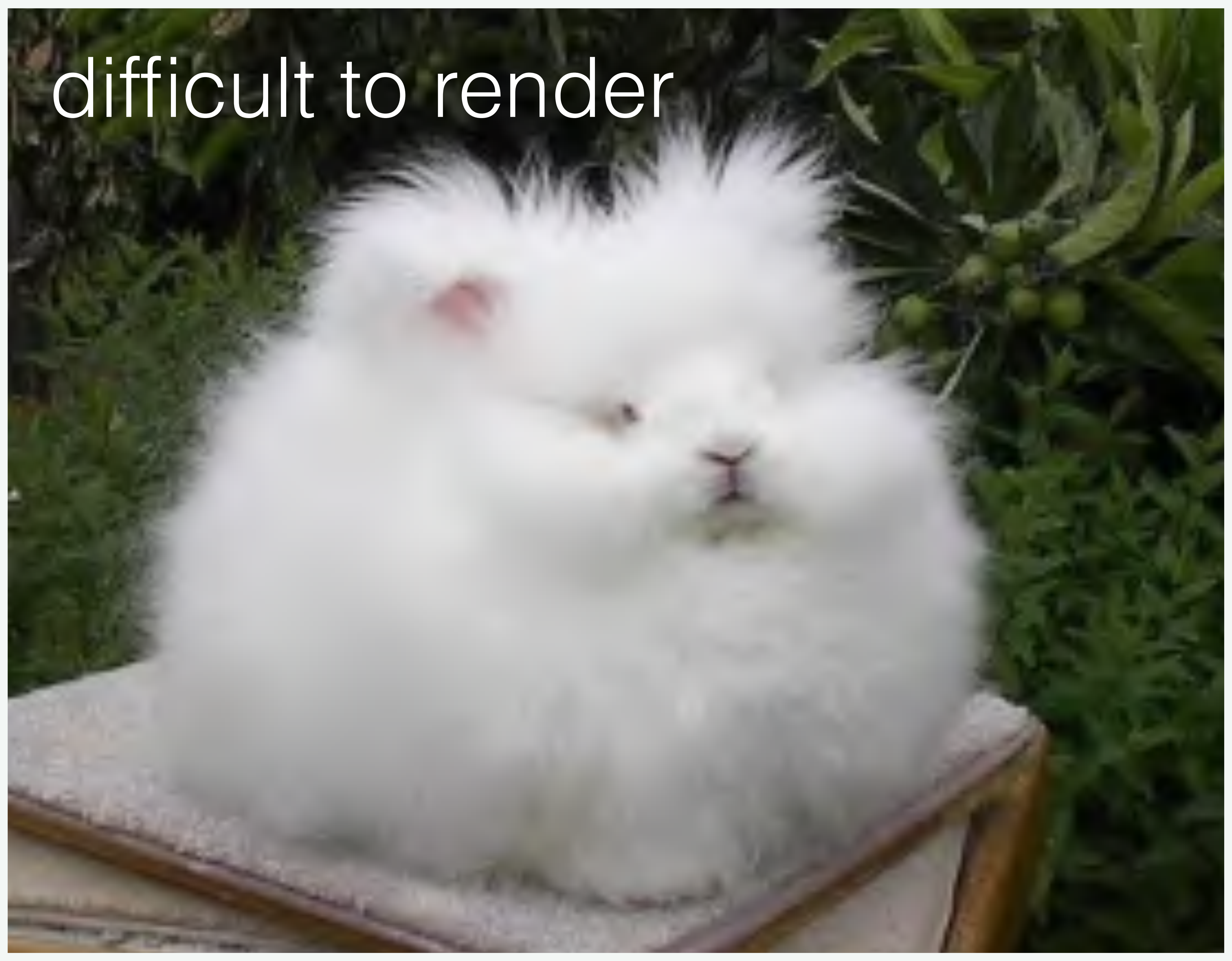
- Render using ray tracing
  - Simulating light bouncing multiple times

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

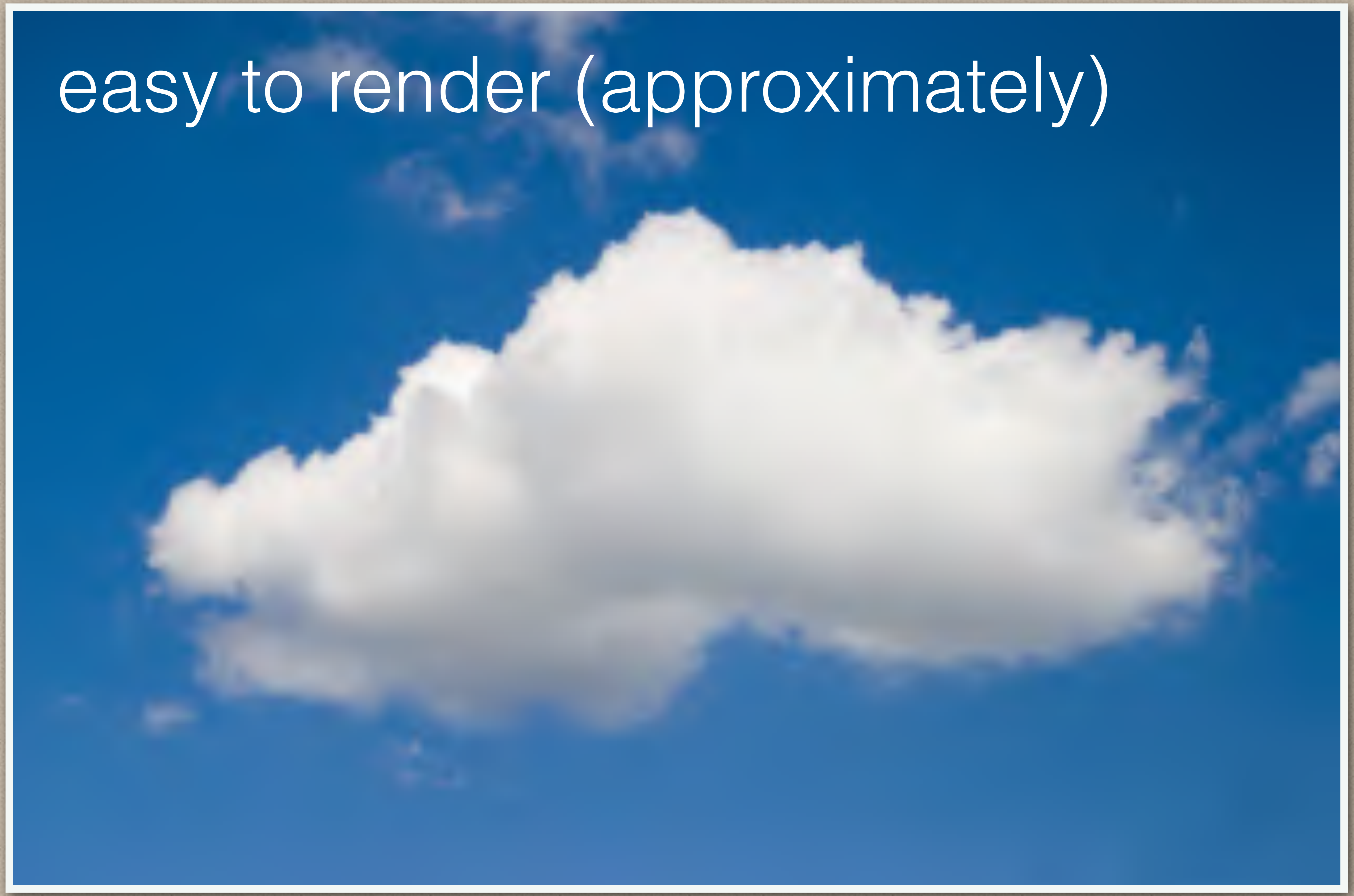
    -> fur fiber #100 -> … -> eye
  - Slow
- Can we avoid tracing multiple bounces?



Motivation



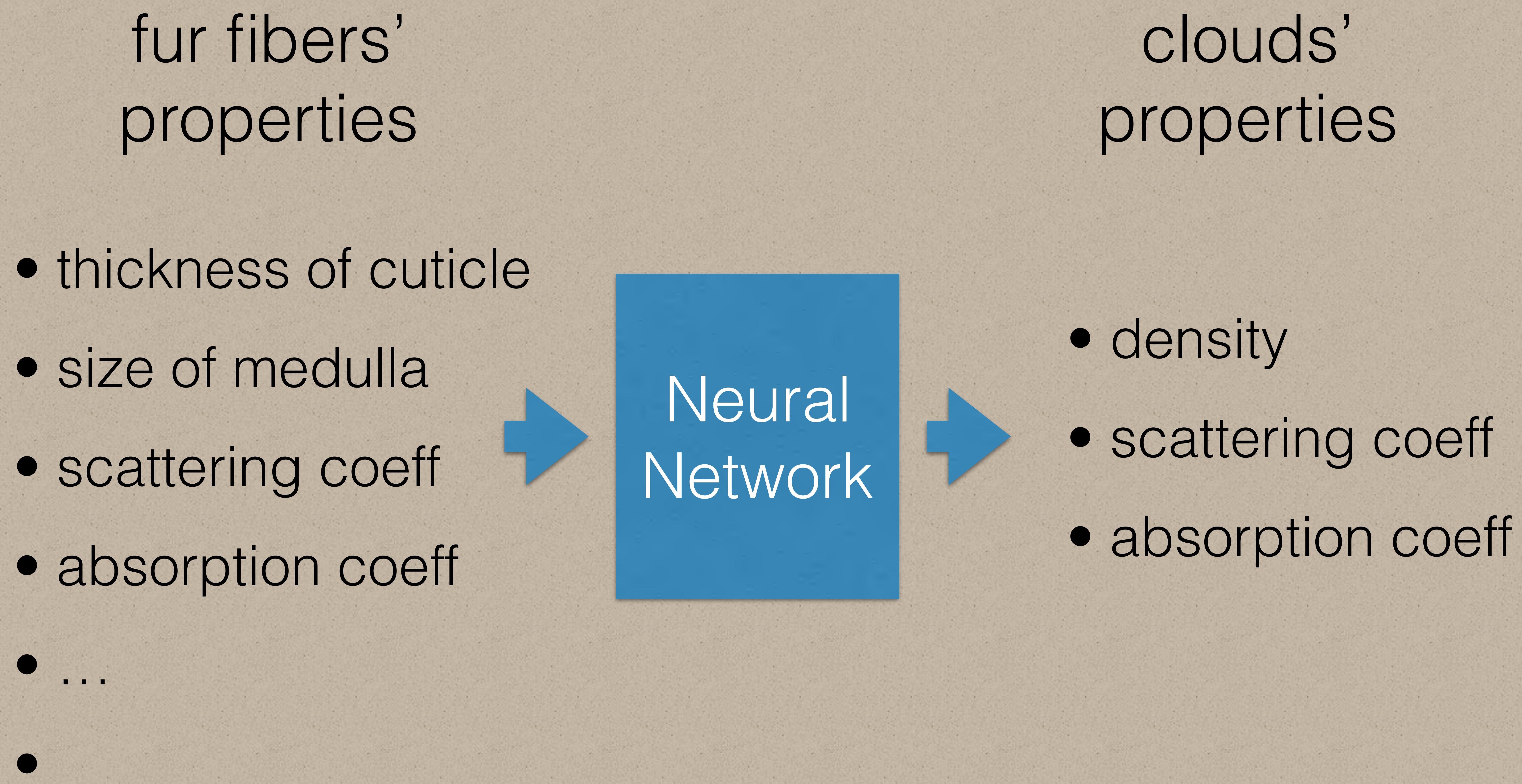
world's fuzziest bunny



Very similar!

Between Physical Systems

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



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

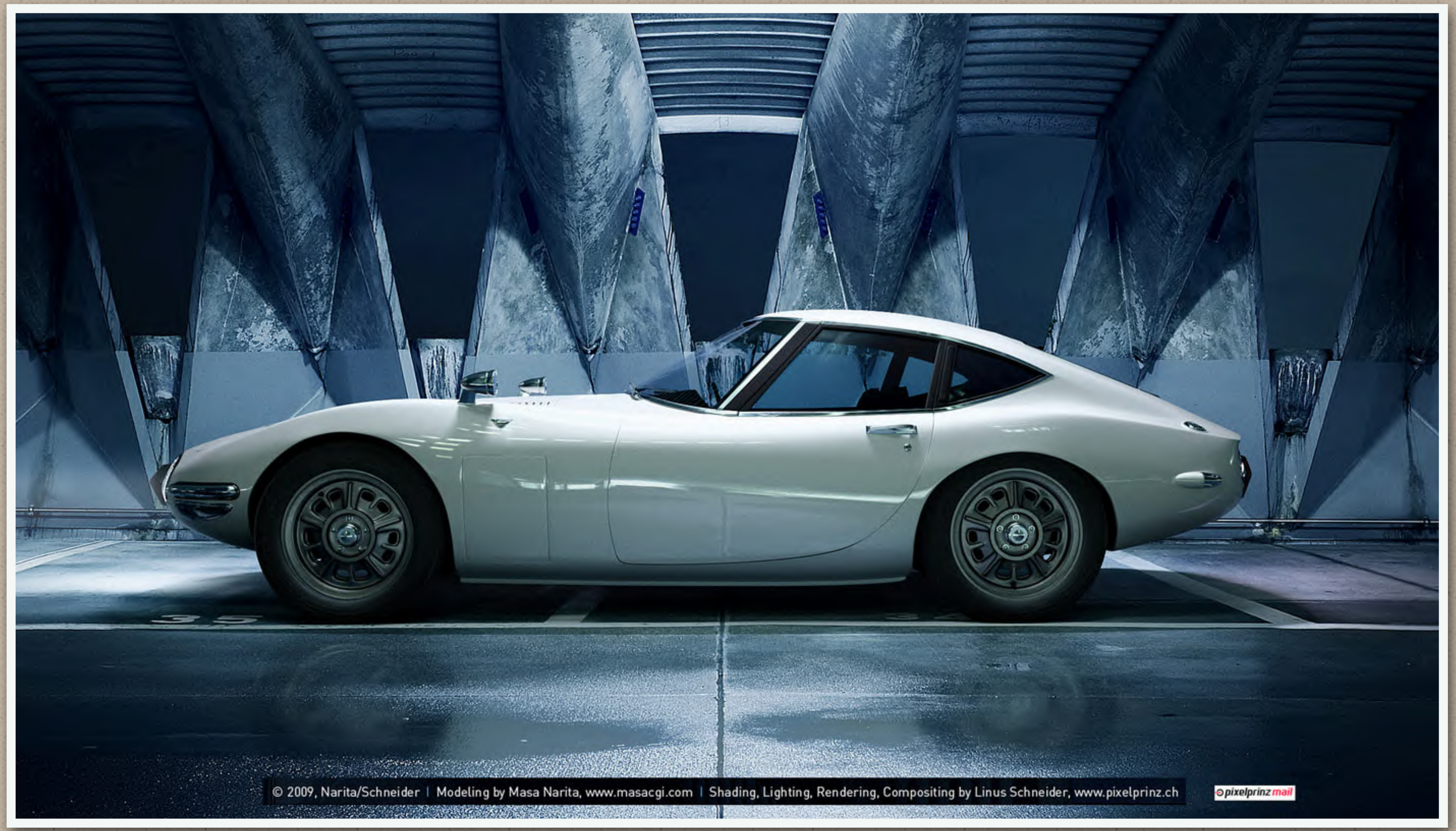
detailed rendering

detailed appearance modeling

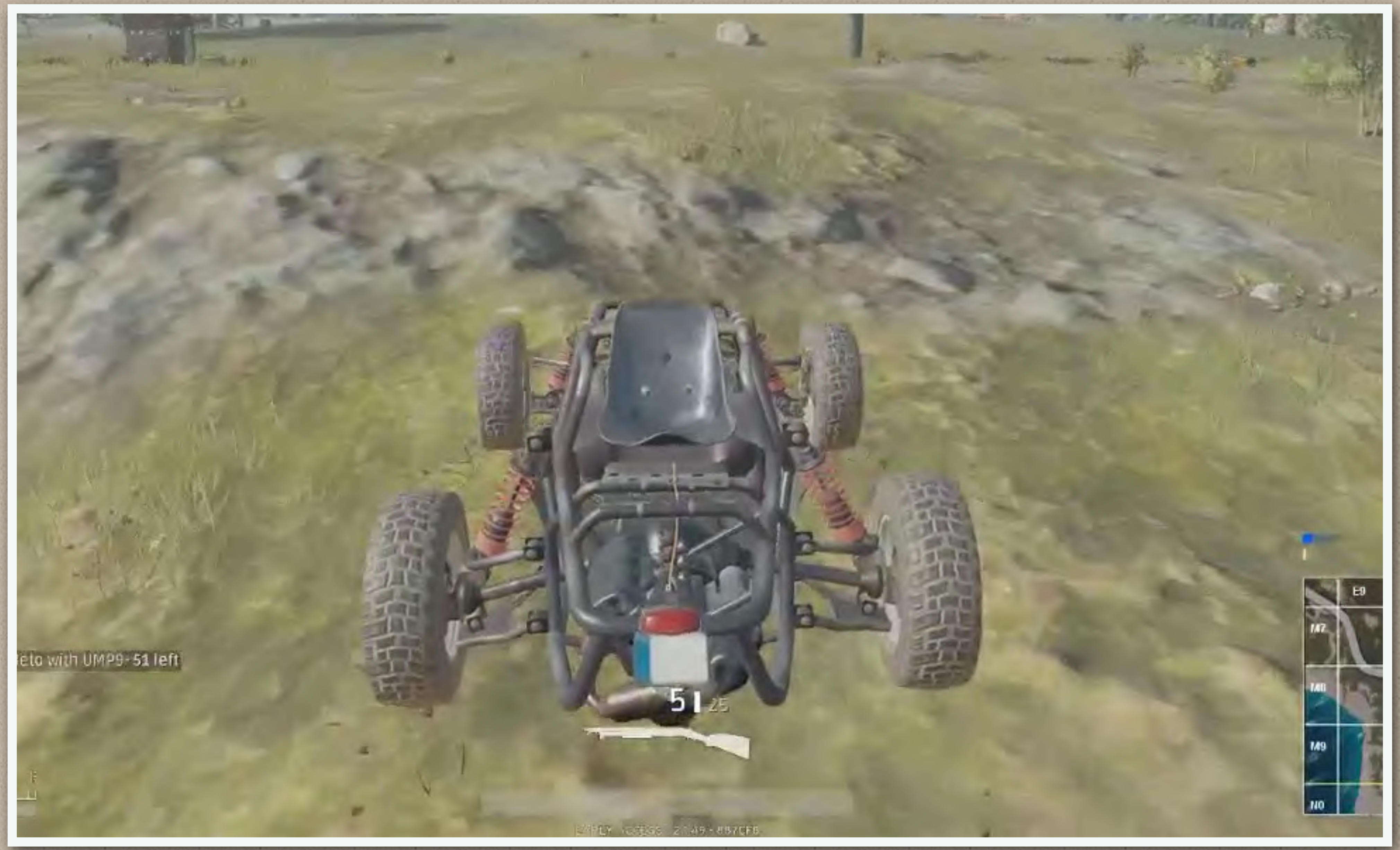
real-time ray tracing

Motivation: Ray Tracing vs. Rasterization

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


© 2009, Narita/Schneider | Modeling by Masa Narita, www.masacgi.com | Shading, Lighting, Rendering, Compositing by Linus Schneider, www.pixelprinz.ch
pixelprinz mail

Toyota 2000GT, from TurboSquid
(source for professional 3D models)



Buggy, from PlayerUnknown's
Battlegrounds (PC game)

Interactive Ray Tracing on GPU

Car interactively rendered using NVIDIA OptiX

Pixar's real-time previewer

Real-Time Ray Tracing (RTRT)

> “Ray tracing is the future
> **and ever will be**.”
>
> — The real-time industry

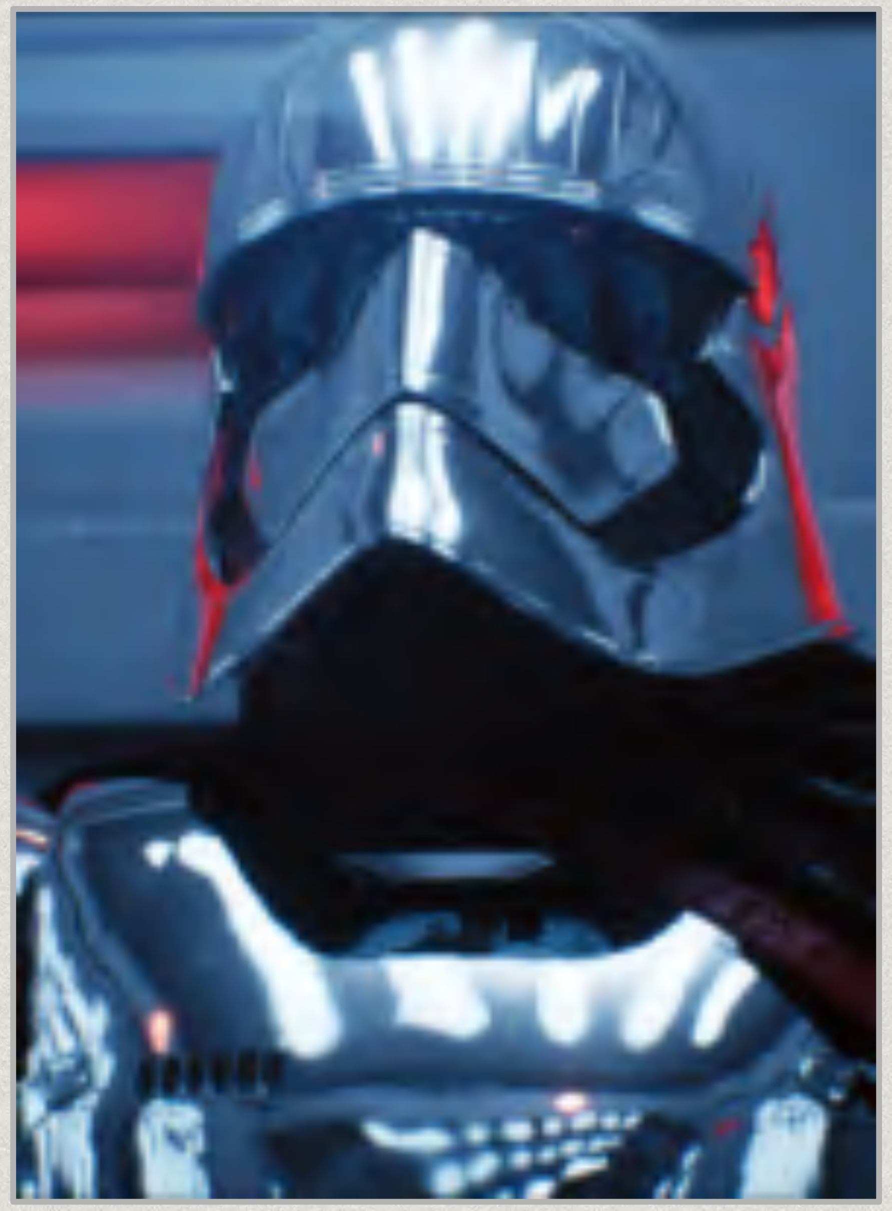
Real-Time Ray Tracing (RTRT) is finally there?

- NVIDIA announced GeForce **RTX** series (Turing architecture) recently
  - Opening a $250 billion market

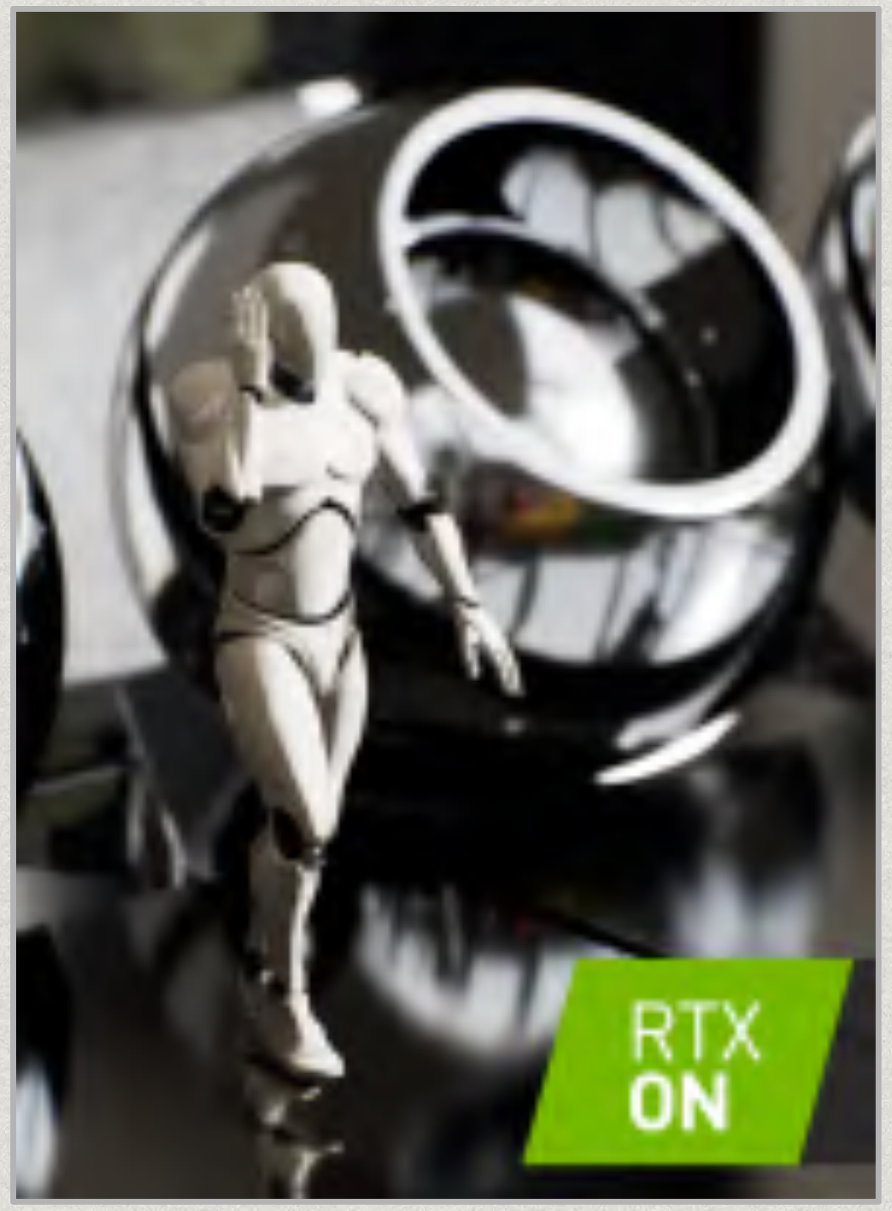
Real-Time Ray Tracing (RTRT) is finally there?

- What does RTX do?

Impressive demos of RTRT



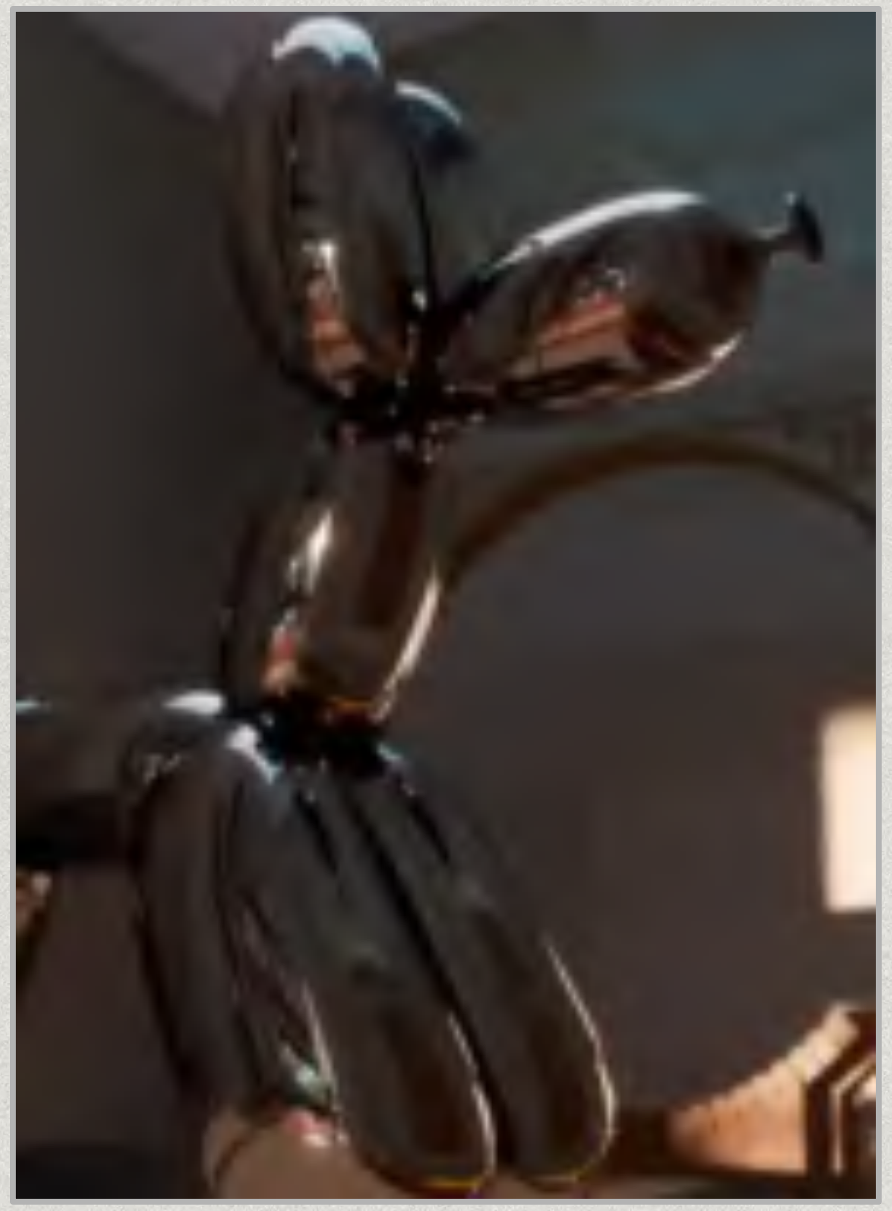
Star Wars Reflections



RTX Demo

Porsche 70 Trailer

SOL



Rosewood Bangkok

Real-Time Ray Tracing (RTRT) is finally there?

- What does RTX actually do?

Advanced **ray traced** effects

Shadows

Reflections & Specular

Ambient Occlusion

Global Illumination

Real-Time Ray Tracing (RTRT) is **getting** there

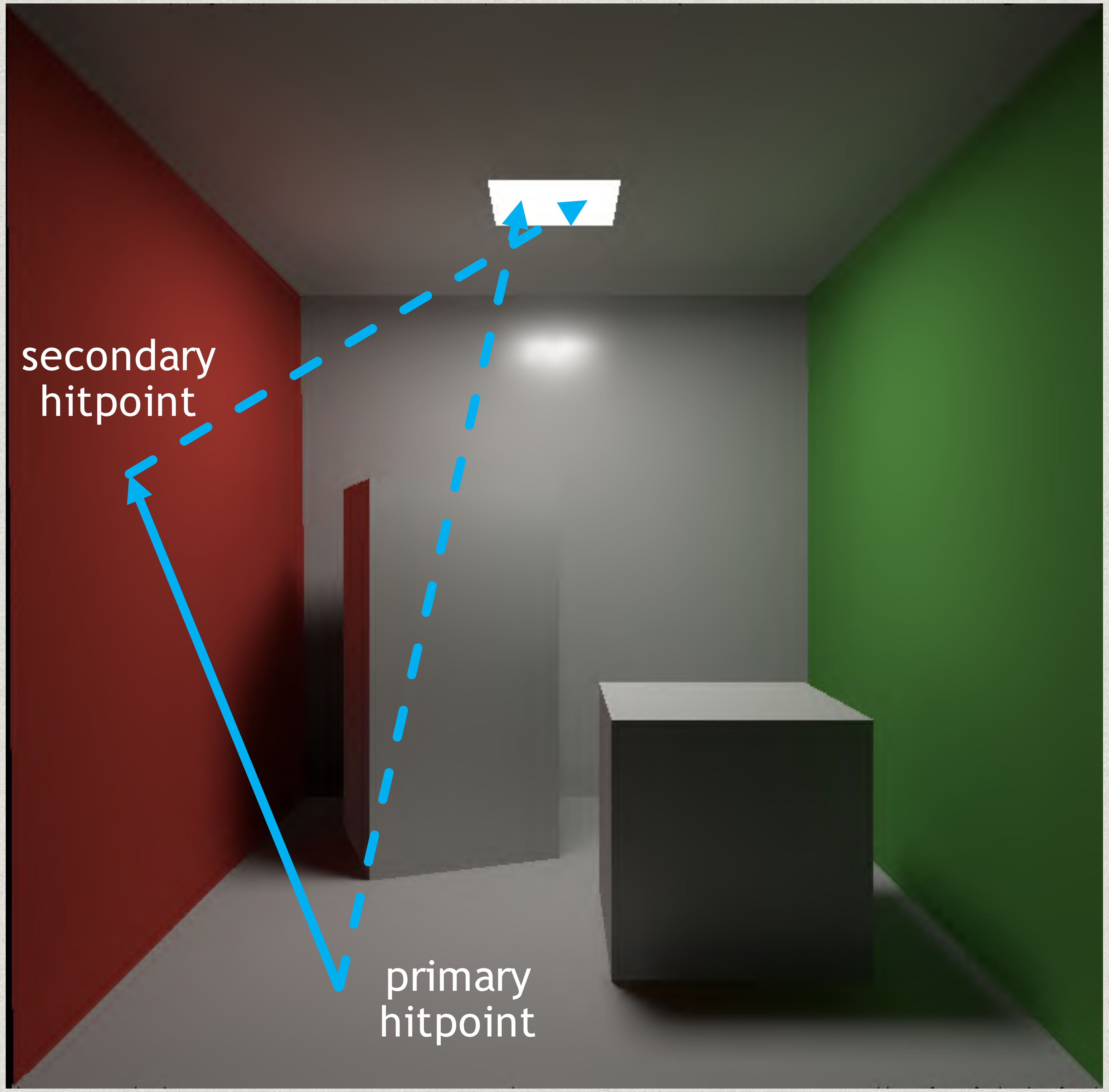
- What does RTX really do?

10 Giga rays per second == **1 sample per pixel**

(for real time applications)

Real-Time Ray Tracing (RTRT) is **getting** there

- 1 SPP == 1 rasterization (primary) +
  1 ray (primary visibility) +
  1 ray (secondary bounce) +
  1 ray (secondary vis.)



Real-Time Ray Tracing (RTRT) is **getting** there

- 1 SPP = Extremely noisy results
- Key technology
  - **Denoising**

Fun image on Twitter

State of the art denoising solution to RTRT

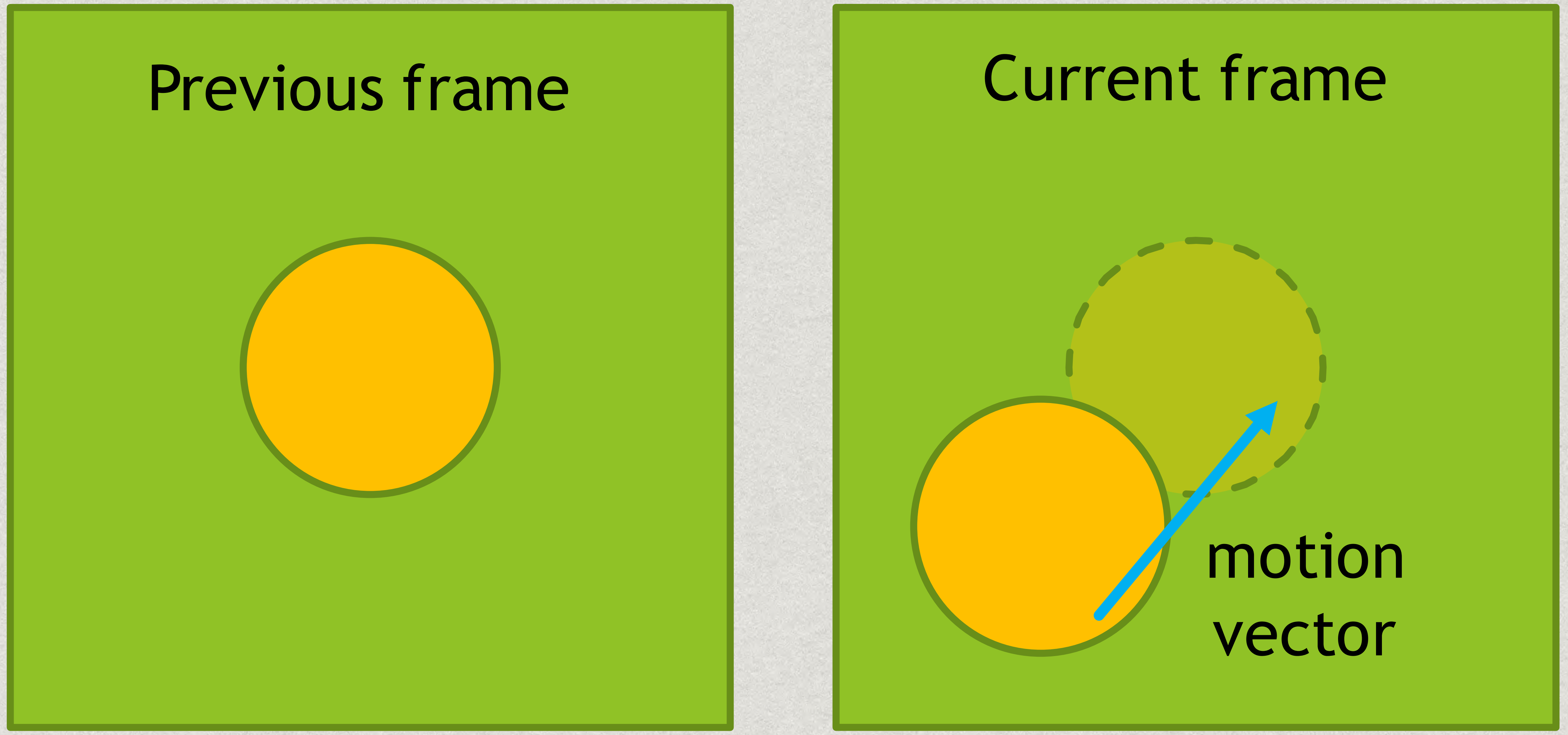
KEEP CALM AND CARRY ON

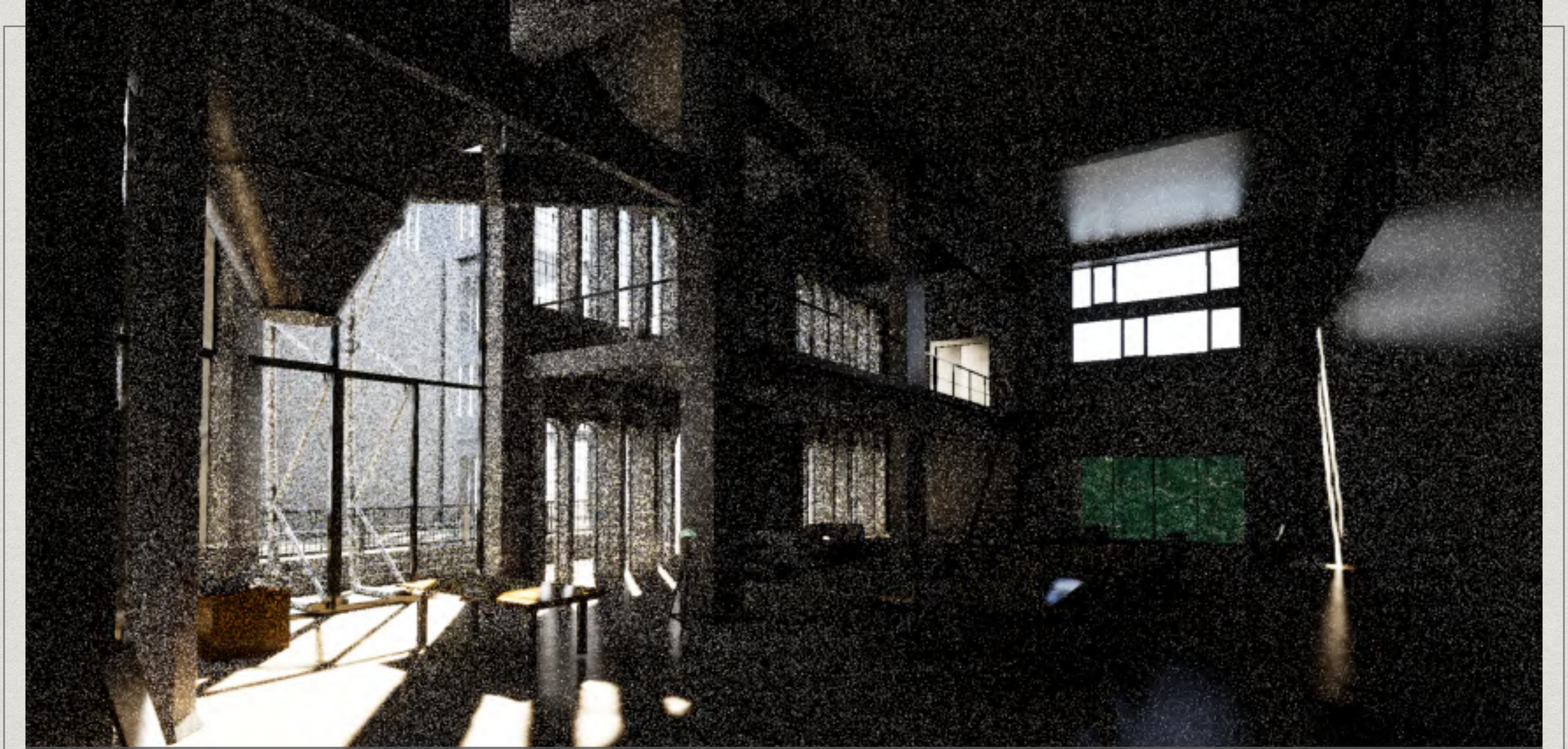
Before we proceed…

- Goals (**with 1 SPP**)
  - Quality (no overblur, no artifacts, keep all details…)
  - Speed (< 2 ms to denoise one frame)
- **Mission impossible**
  - Sheared filtering series (SF, AAF, FSF, MAAF, …)
  - Other offline filtering methods (IPP, BM3D, APR, …)
  - Deep learning series (CNN, Autoencoder, …)

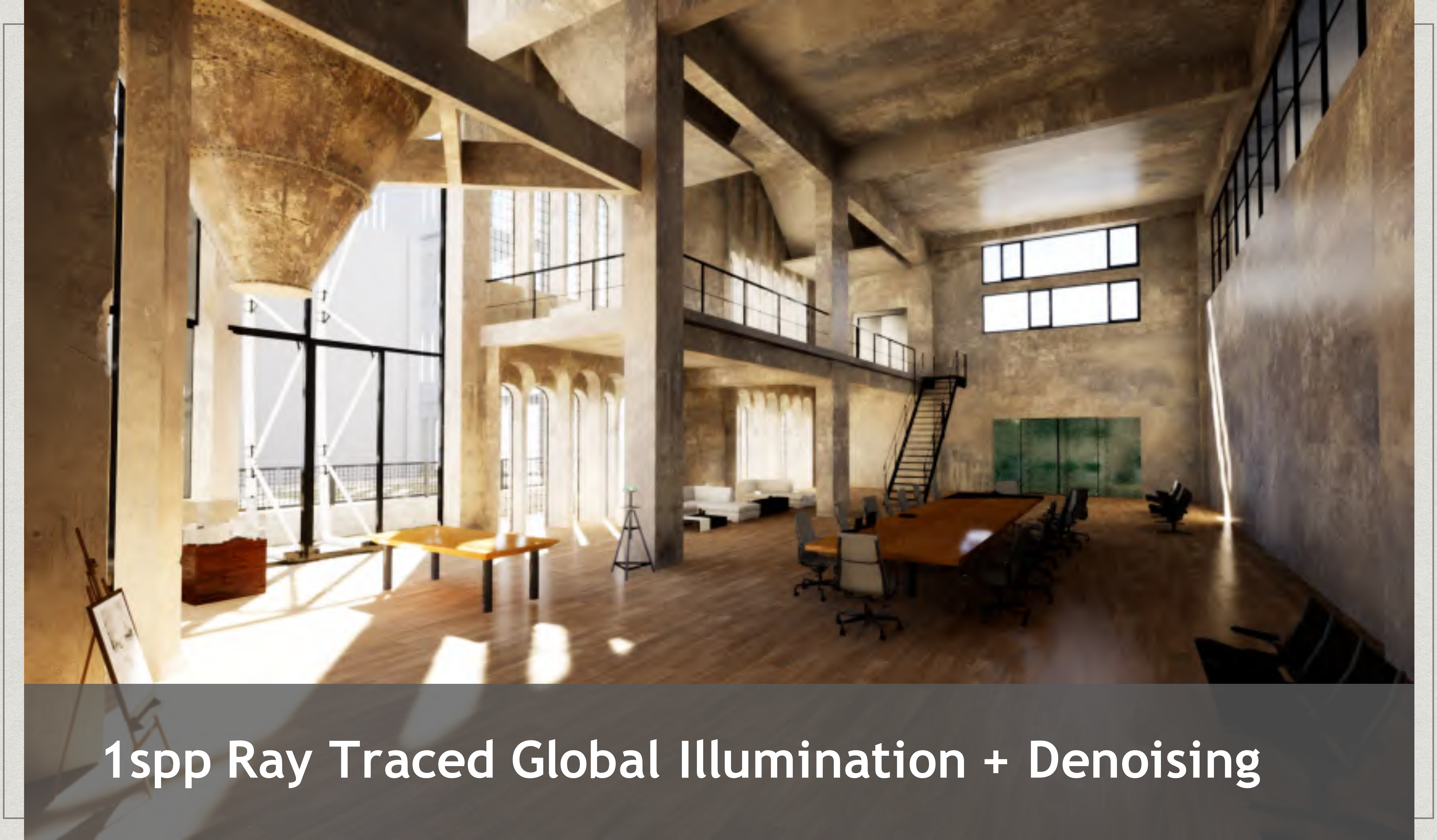
State of the art denoising solution to RTRT

- 3 most important ideas
  - Temporal!
  - **Temporal!!**
  - **Temporal!!!**
- Key idea
  - Suppose the previous frame is denoised and reuse it
  - Use motion vectors to find previous locations
  - Essentially increased SPP
  - Spatial? Small cross-bilateral filter



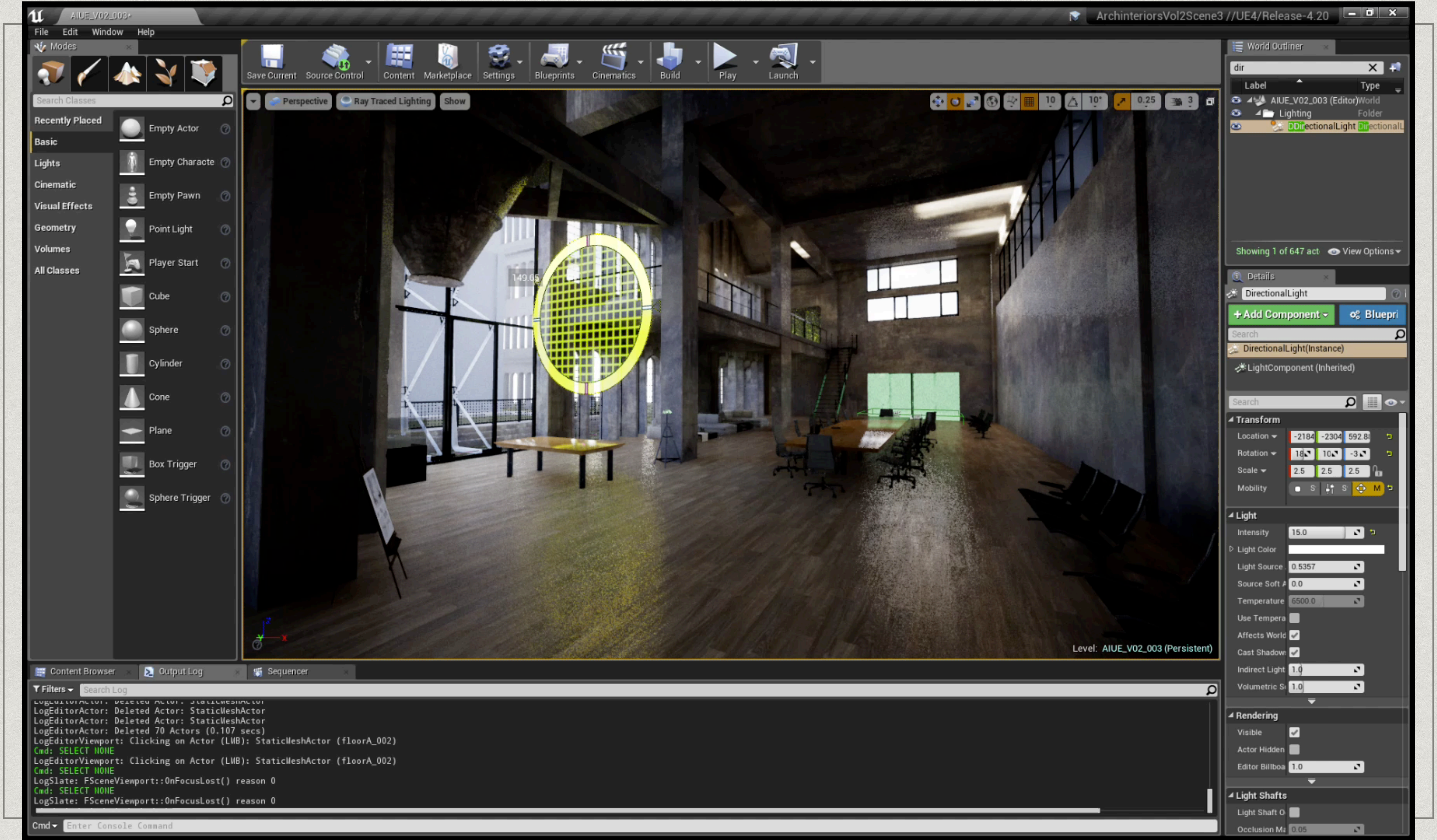


1spp Ray Traced Global Illumination



1spp Ray Traced Global Illumination + Denoising

Ground Truth



World Outliner search: dir

Label	Type
AIUE_V02_003 (Editor)	World
Lighting	Folder
DDirectionalLight	DirectionalL

Details: DirectionalLight

DirectionalLight(Instance)

LightComponent (Inherited)

Transform

Location	-2184	-2304	592.8
Rotation	18	10	-3
Scale	2.5	2.5	2.5
Mobility	S	S	M

Light

Intensity	15.0
Light Color	
Light Source	0.5357
Source Soft A	0.0
Temperature	6500.0
Use Tempera	□
Affects World	☑
Cast Shadow	☑
Indirect Light	1.0
Volumetric S	1.0

Rendering

Visible	☑
Actor Hidden	□
Editor Billboa	1.0

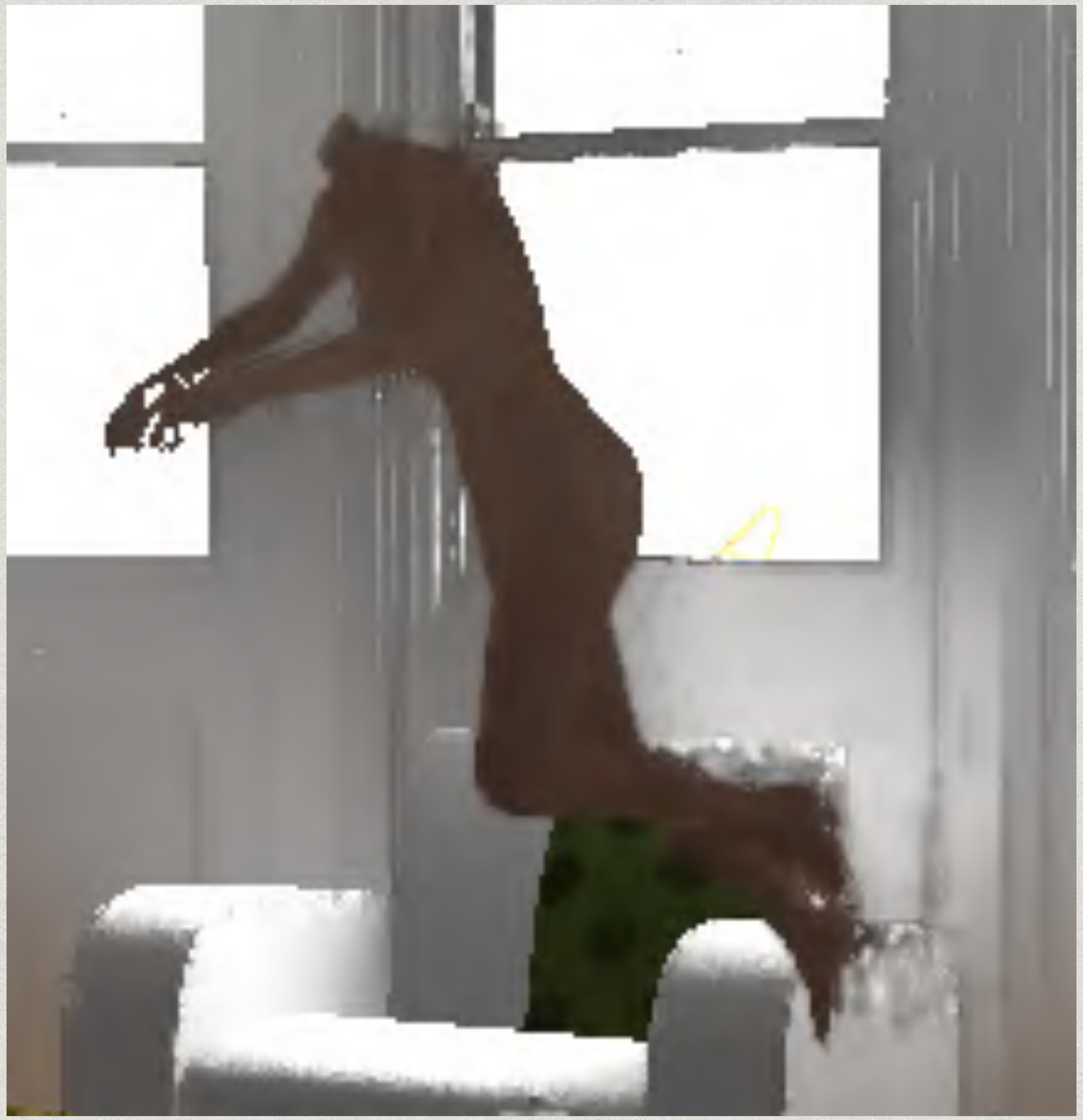
Light Shafts

Light Shaft O	□
Occlusion Ma	0.05

```
LogEditorActor: Deleted Actor: StaticMeshActor
LogEditorActor: Deleted Actor: StaticMeshActor
LogEditorActor: Deleted Actor: StaticMeshActor
LogEditorActor: Deleted 70 Actors (0.107 secs)
LogEditorViewport: Clicking on Actor (LMB): StaticMeshActor (floorA_002)
Cmd: SELECT NONE
LogEditorViewport: Clicking on Actor (LMB): StaticMeshActor (floorA_002)
Cmd: SELECT NONE
LogSlate: FSceneViewport::OnFocusLost() reason 0
Cmd: SELECT NONE
LogSlate: FSceneViewport::OnFocusLost() reason 0
```

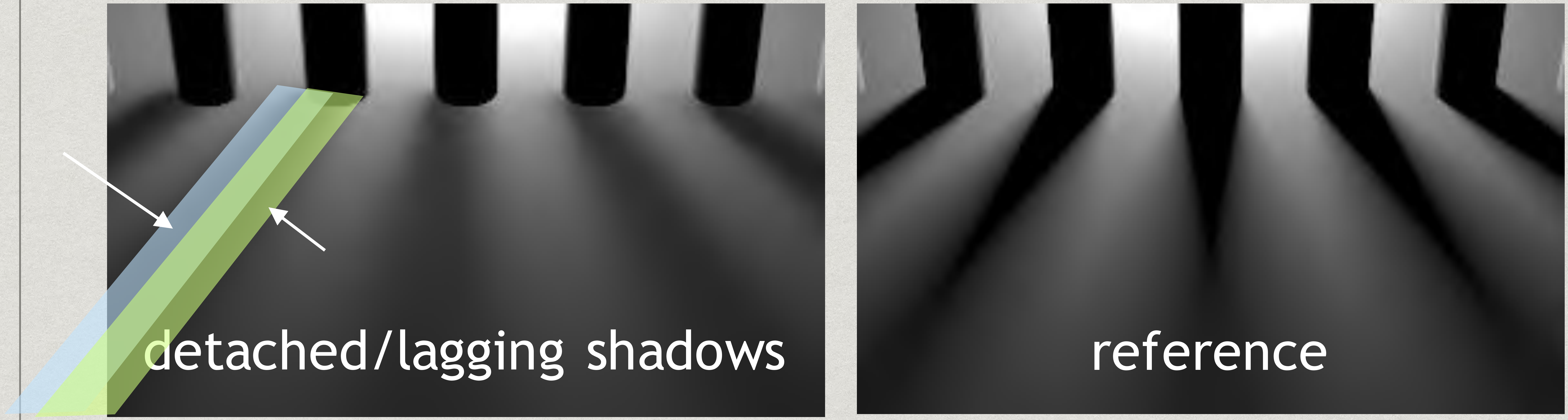
Known issues of temporal denoising

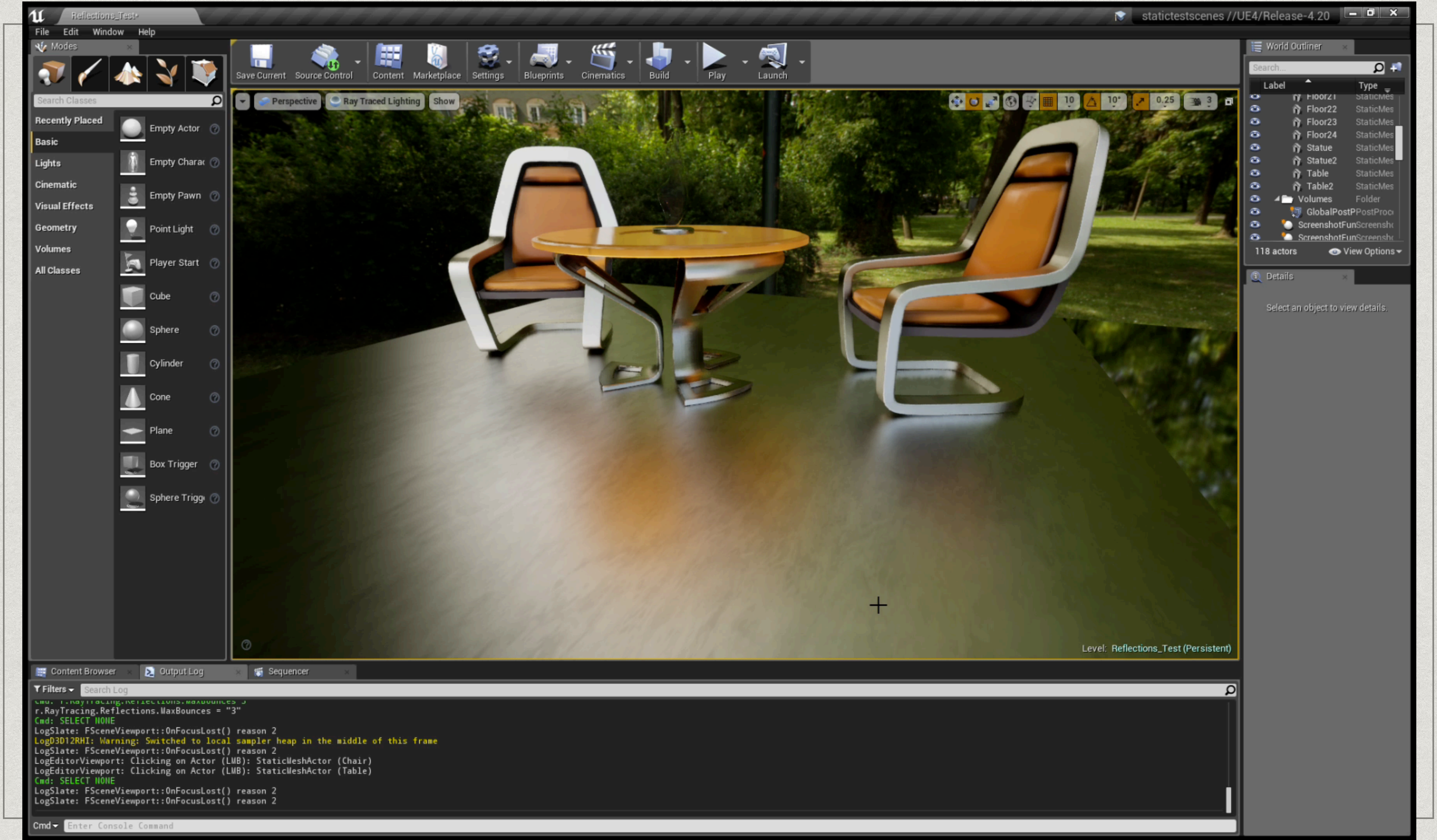
- Occlusion



Known issues of temporal denoising

- Lagging





Label	Type
Floor21	StaticMes
Floor22	StaticMes
Floor23	StaticMes
Floor24	StaticMes
Statue	StaticMes
Statue2	StaticMes
Table	StaticMes
Table2	StaticMes
Volumes	Folder
GlobalPostP	PostProc
ScreenshotFun	Screensh
ScreenshotFun	Screensh

118 actors View Options

Level: Reflections_Test (Persistent)

```
Cmd: r.RayTracing.Reflections.MaxBounces 3
r.RayTracing.Reflections.MaxBounces = "3"
Cmd: SELECT NONE
LogSlate: FSceneViewport::OnFocusLost() reason 2
LogD3D12RHI: Warning: Switched to local sampler heap in the middle of this frame
LogSlate: FSceneViewport::OnFocusLost() reason 2
LogEditorViewport: Clicking on Actor (LMB): StaticMeshActor (Chair)
LogEditorViewport: Clicking on Actor (LMB): StaticMeshActor (Table)
Cmd: SELECT NONE
LogSlate: FSceneViewport::OnFocusLost() reason 2
LogSlate: FSceneViewport::OnFocusLost() reason 2
```

Future Research Directions

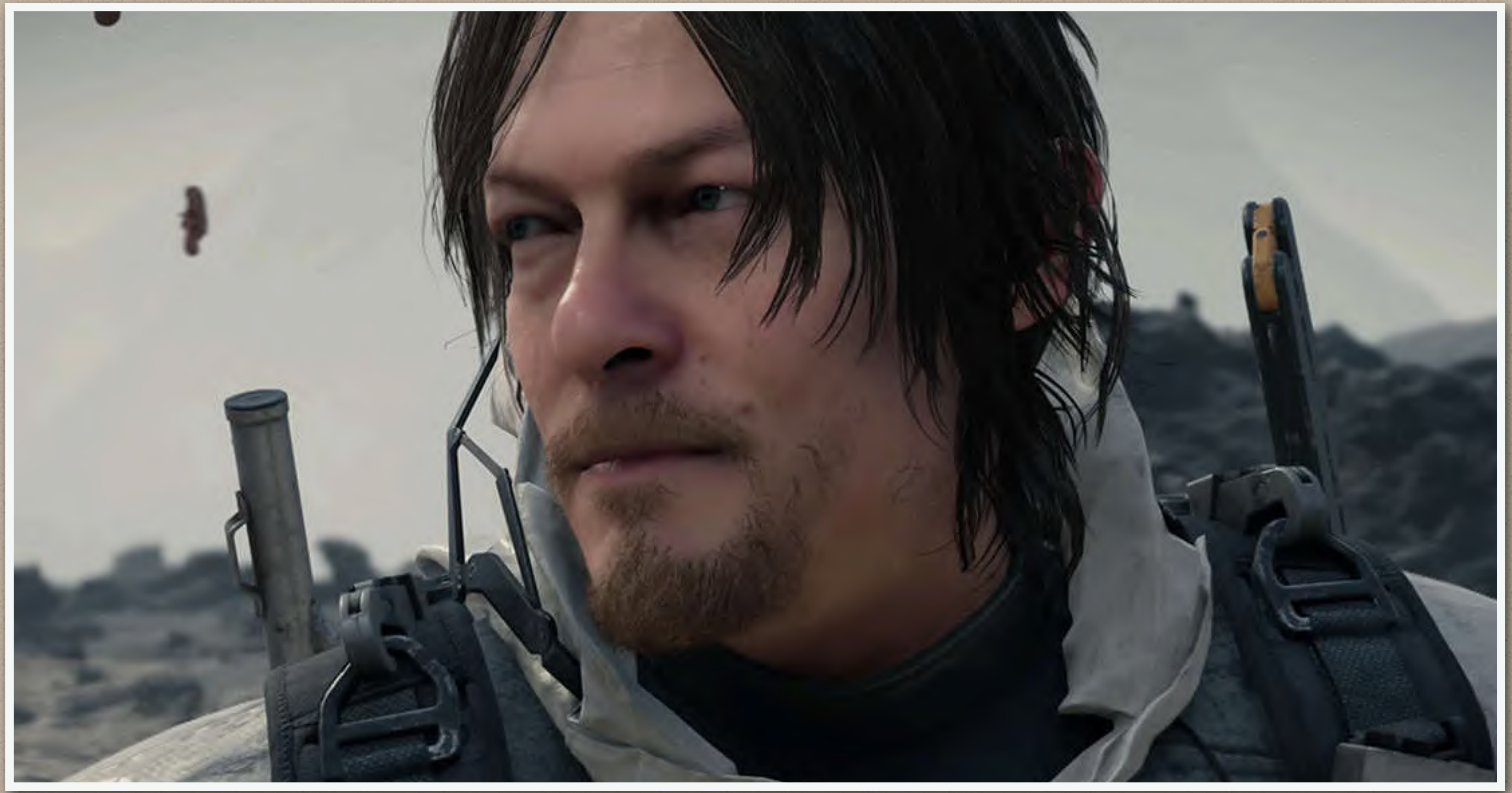
- Challenges in rendering
  - Better realism
  - Further speed up
  - What else?

Future Research Directions

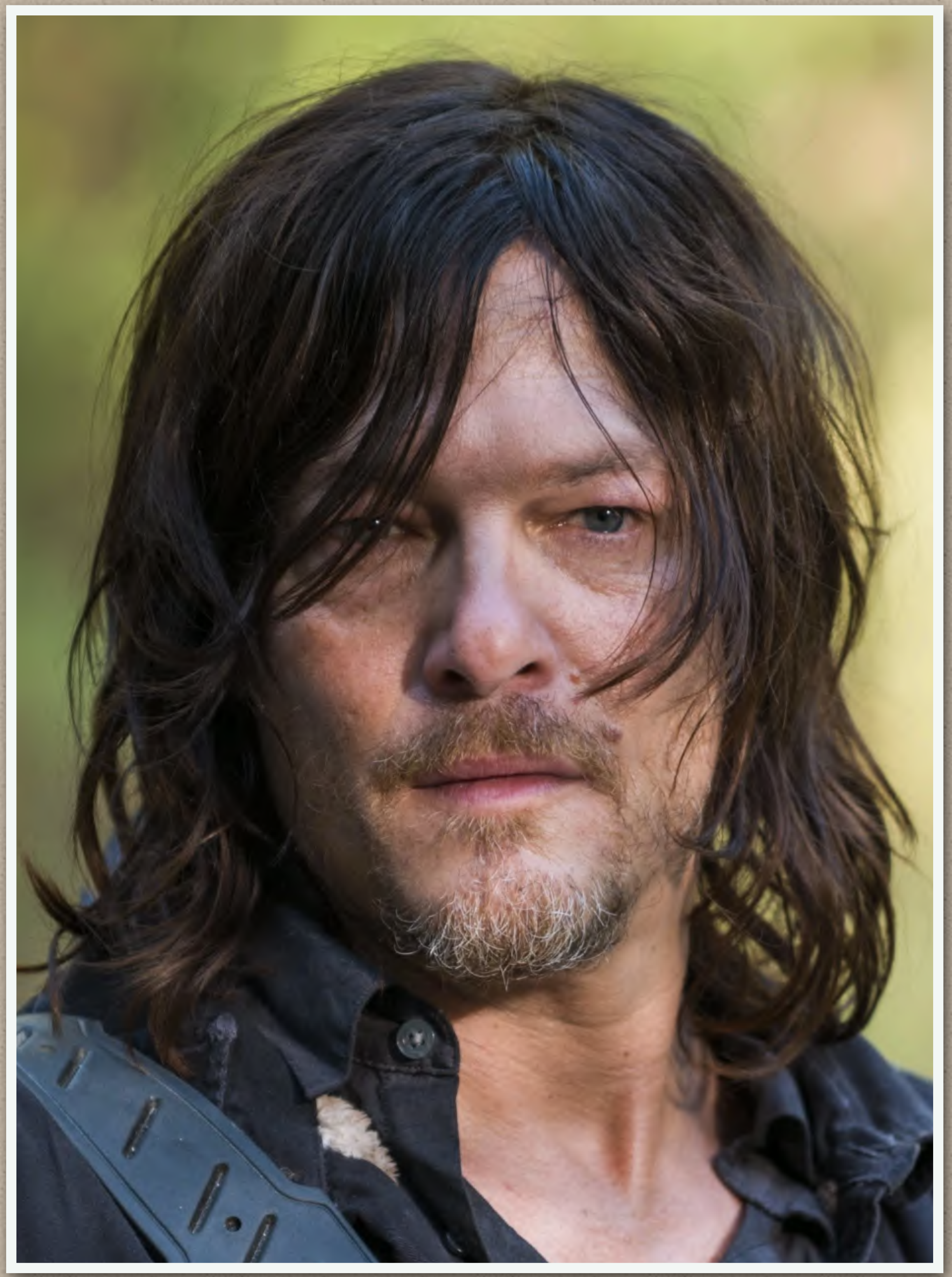
- Detailed rendering from **volumetric** microstructures
  - Glints and halo from snow
  - Visible dust in 'god ray'

Future Research Directions

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



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



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

Future Research Directions

- Realistic rendering is in need
  - Accurate synthetic data
  - 3D repository

Golden Gate Bridge

?

Golden Gate Bridge
(rainy / snowy)

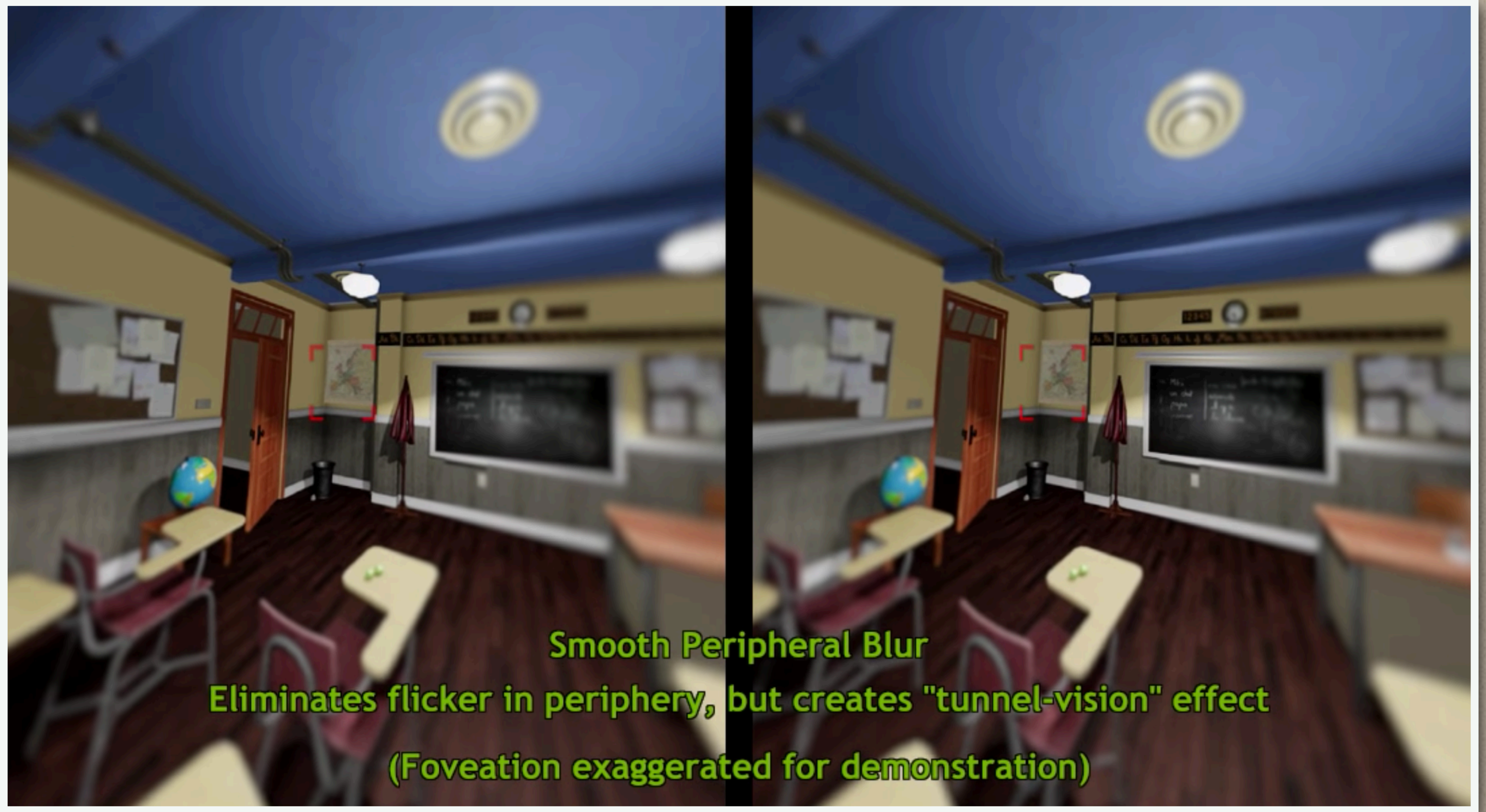
Deep Learning
for
**All Weather
Conditions**

Self-Driving Cars

Future Research Directions

- Virtual Reality (VR) / Augmented Reality (AR)
  - VR = Equipment + CG (CG: Computer Graphics)
  - AR = CV + VR (CV: Computer Vision)

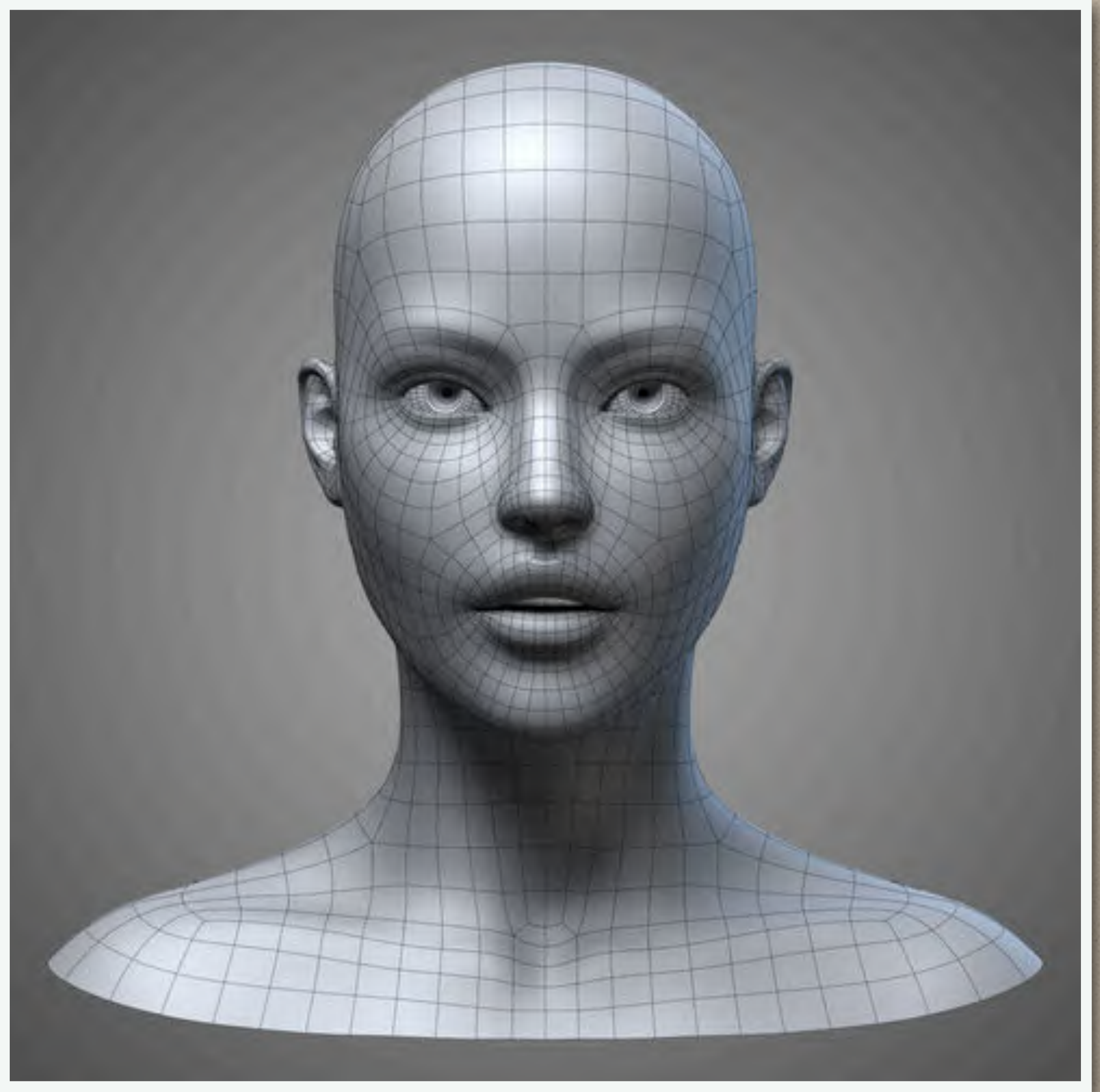
Oculus Rift (VR equipment)


Smooth Peripheral Blur
Eliminates flicker in periphery, but creates "tunnel-vision" effect
(Foveation exaggerated for demonstration)

foveated rendering by NVIDIA

Future Research Directions

- Exploiting Machine Learning for Computer Graphics
  - Differentiable rendering + Computer Vision
  - Does RTRT fit here?

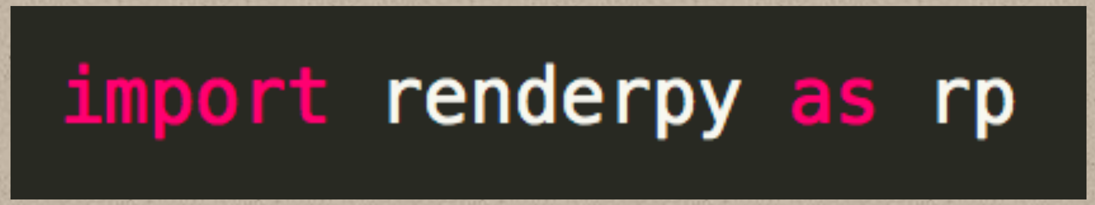


Human bust model from CGTrader
(source for professional 3D models)

Saya, Computer Generated Imagery (CGI)

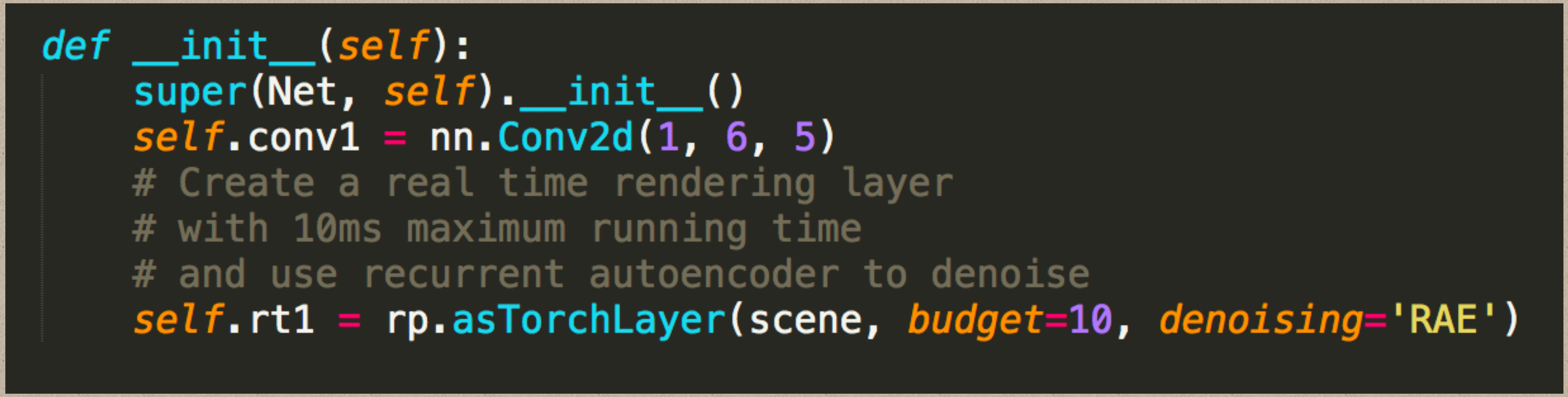
Future Research Directions

- Inter-operability with other communities
  - A black-box RTRT renderer (unlike Mitsuba & PBRT)
  - Think about this



```
import renderpy as rp
```

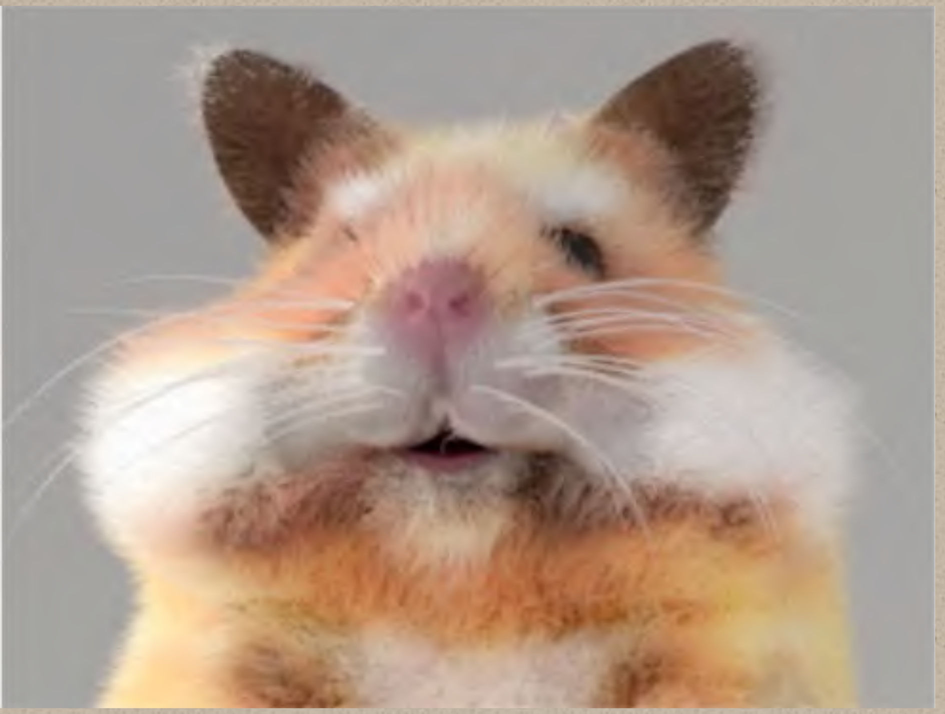
  - And this



```
def __init__(self):
    super(Net, self).__init__()
    self.conv1 = nn.Conv2d(1, 6, 5)
    # Create a real time rendering layer
    # with 10ms maximum running time
    # and use recurrent autoencoder to denoise
    self.rt1 = rp.asTorchLayer(scene, budget=10, denoising='RAE')
```

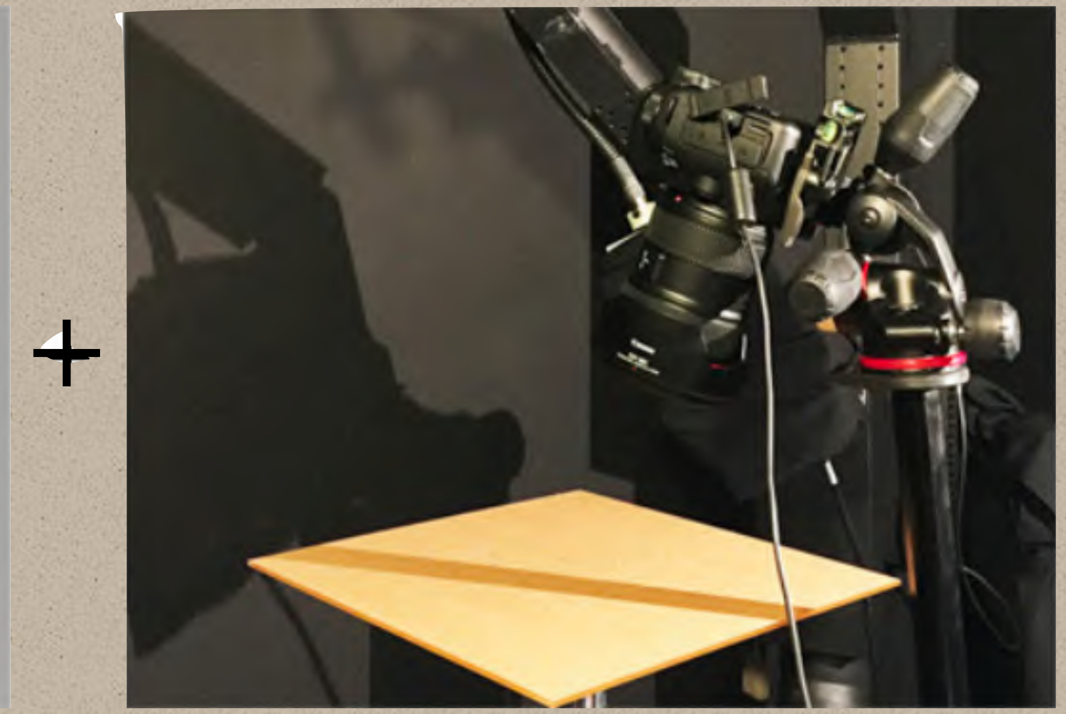
Future Research Directions

- My “rendering equation”



RT / offline
light transport

+

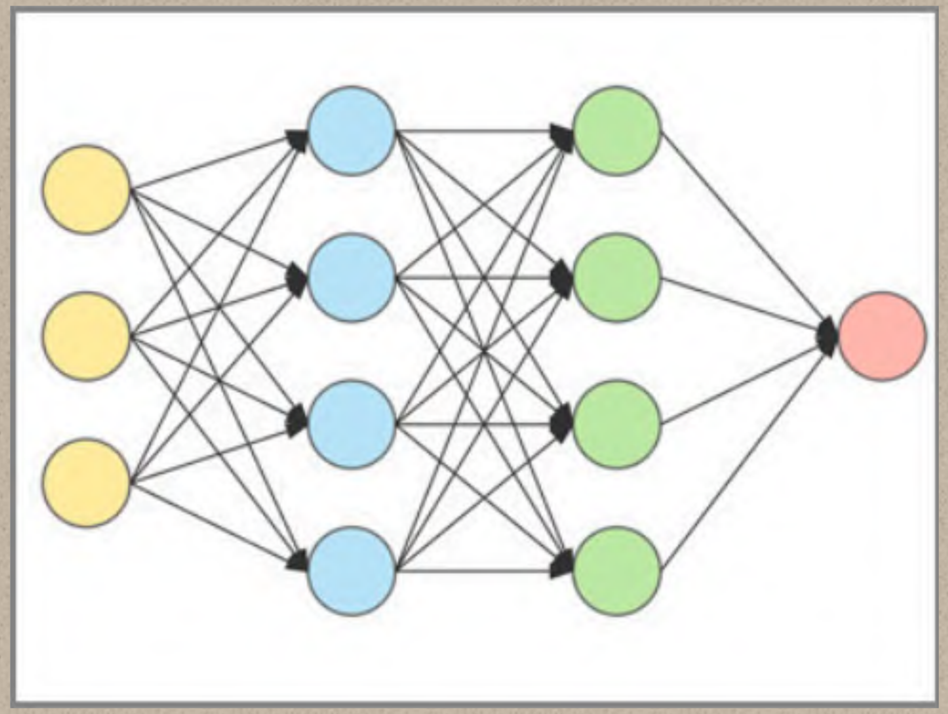


Appearance
Modeling

+

VR / AR

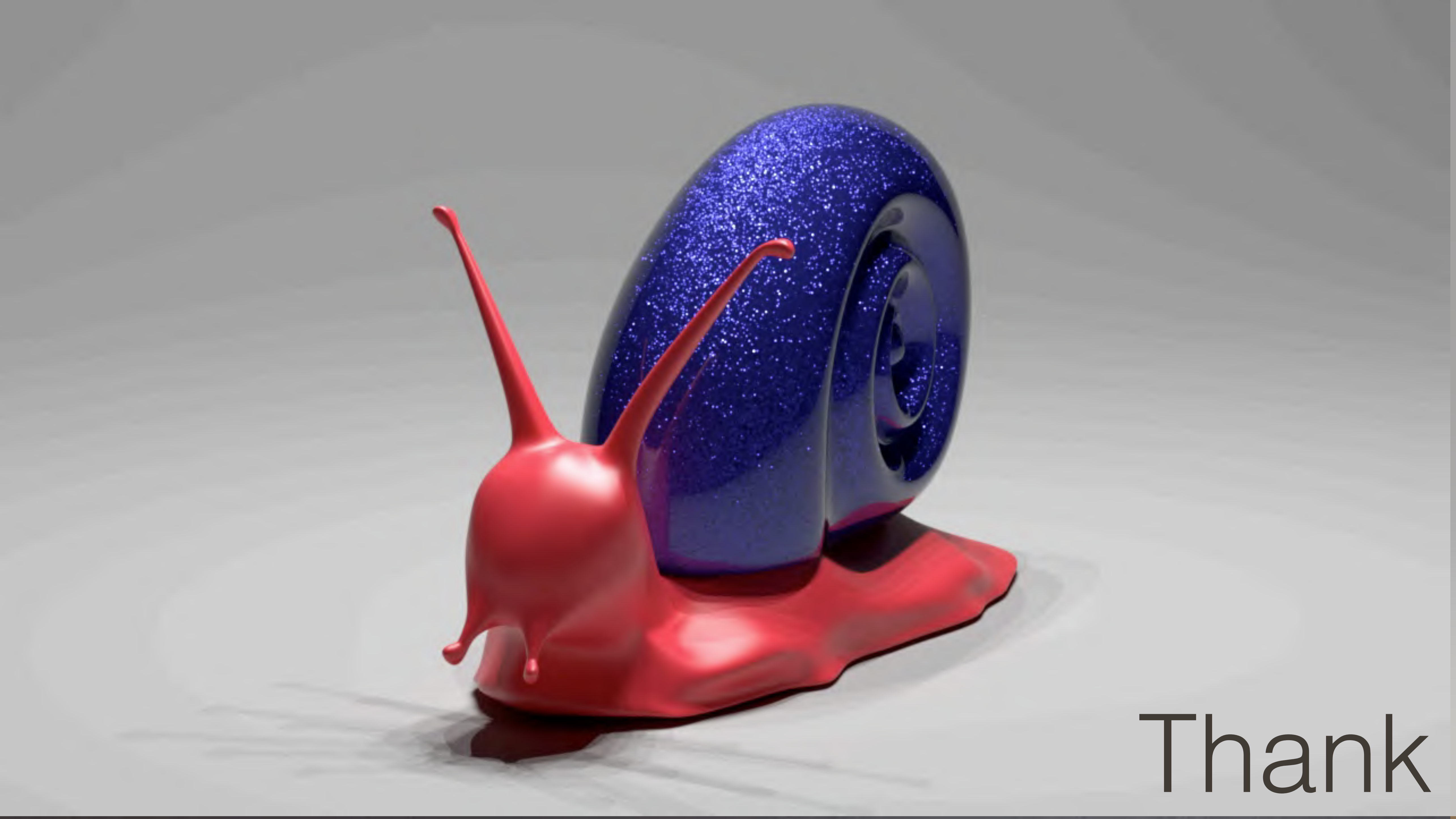
+



Machine
Learning

=

Ultimate
Realism



Thank you!