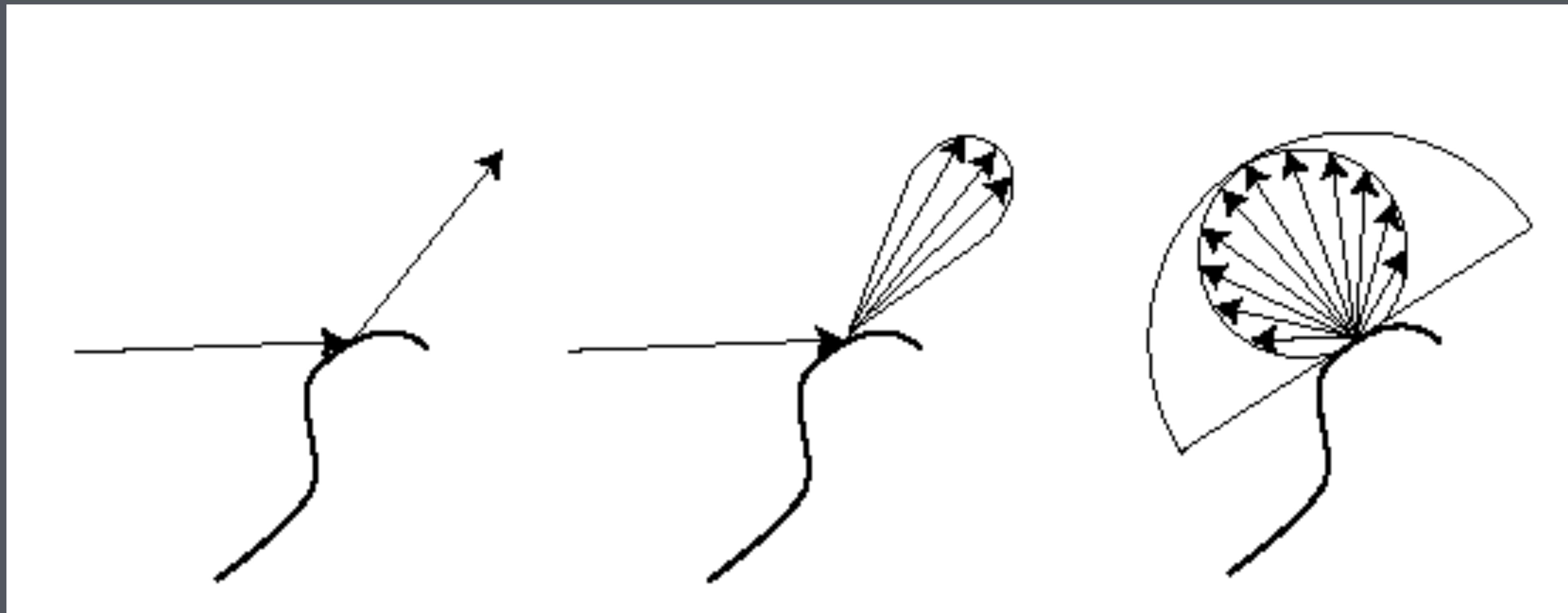


16 Soft Illumination Effects

Steve Marschner
CS5625 Spring 2016

Irradiance environment mapping

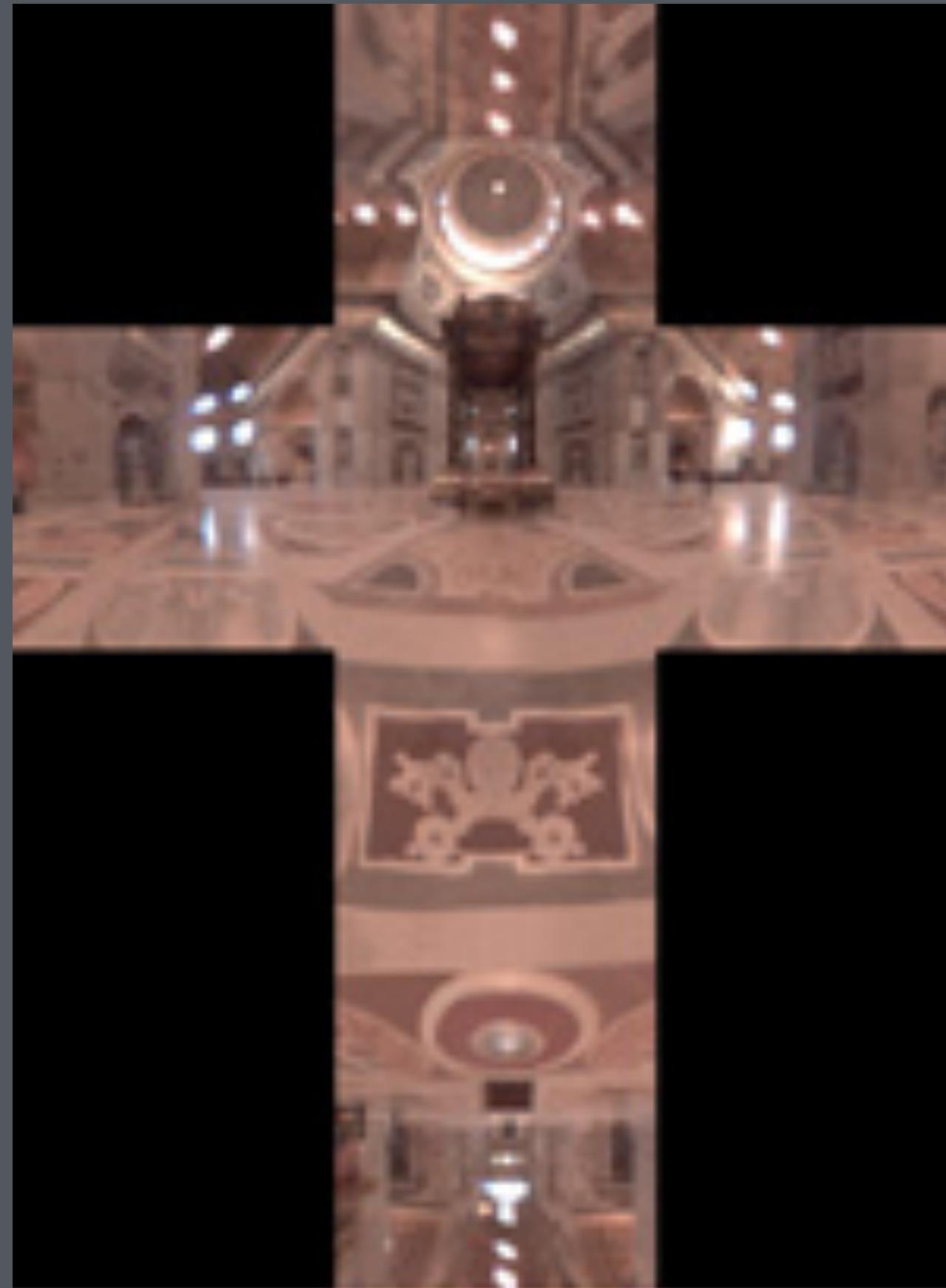


environment map
for specular surface

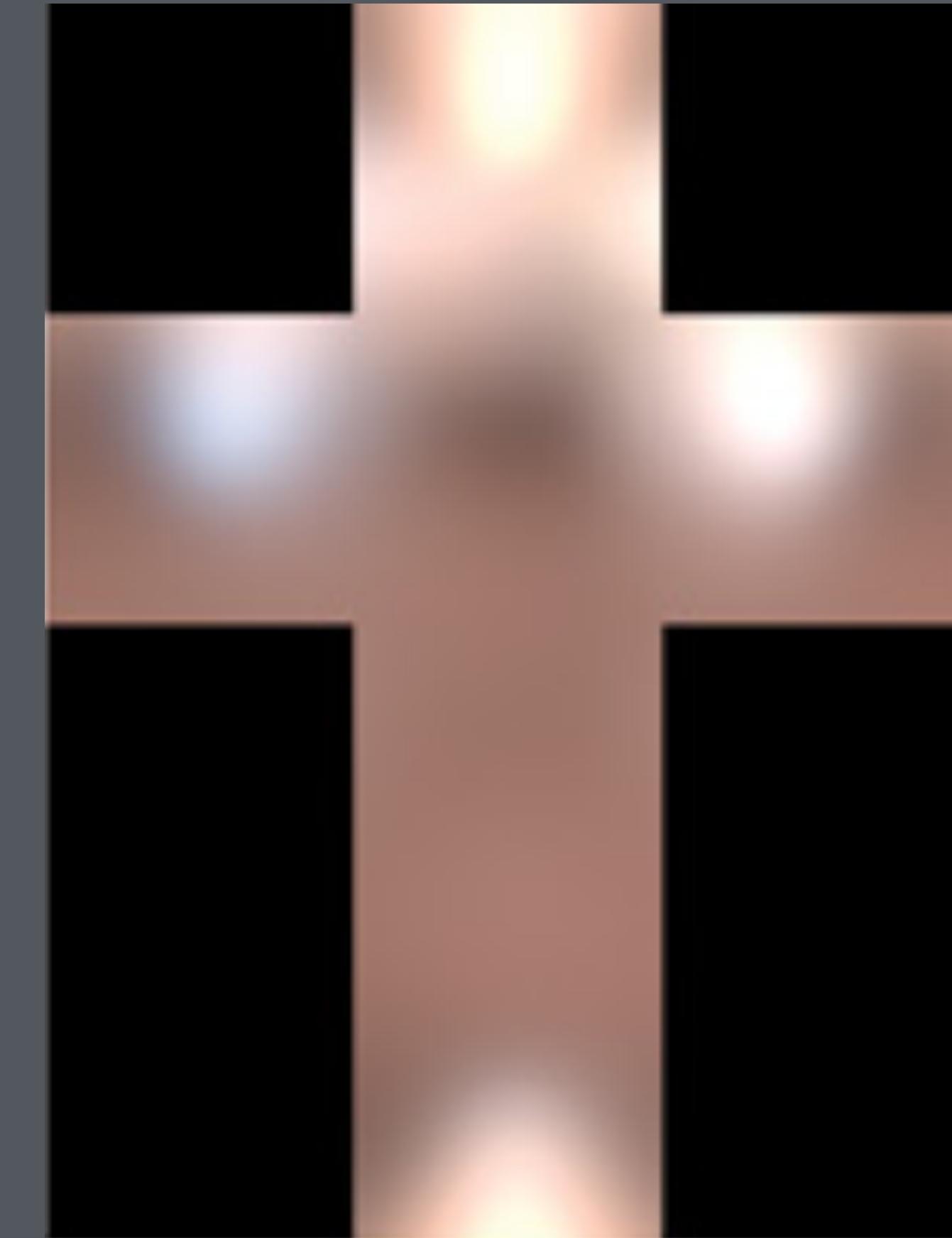
prefiltered map
for glossy surface

prefiltered map
for diffuse surface

Prefiltered environment map

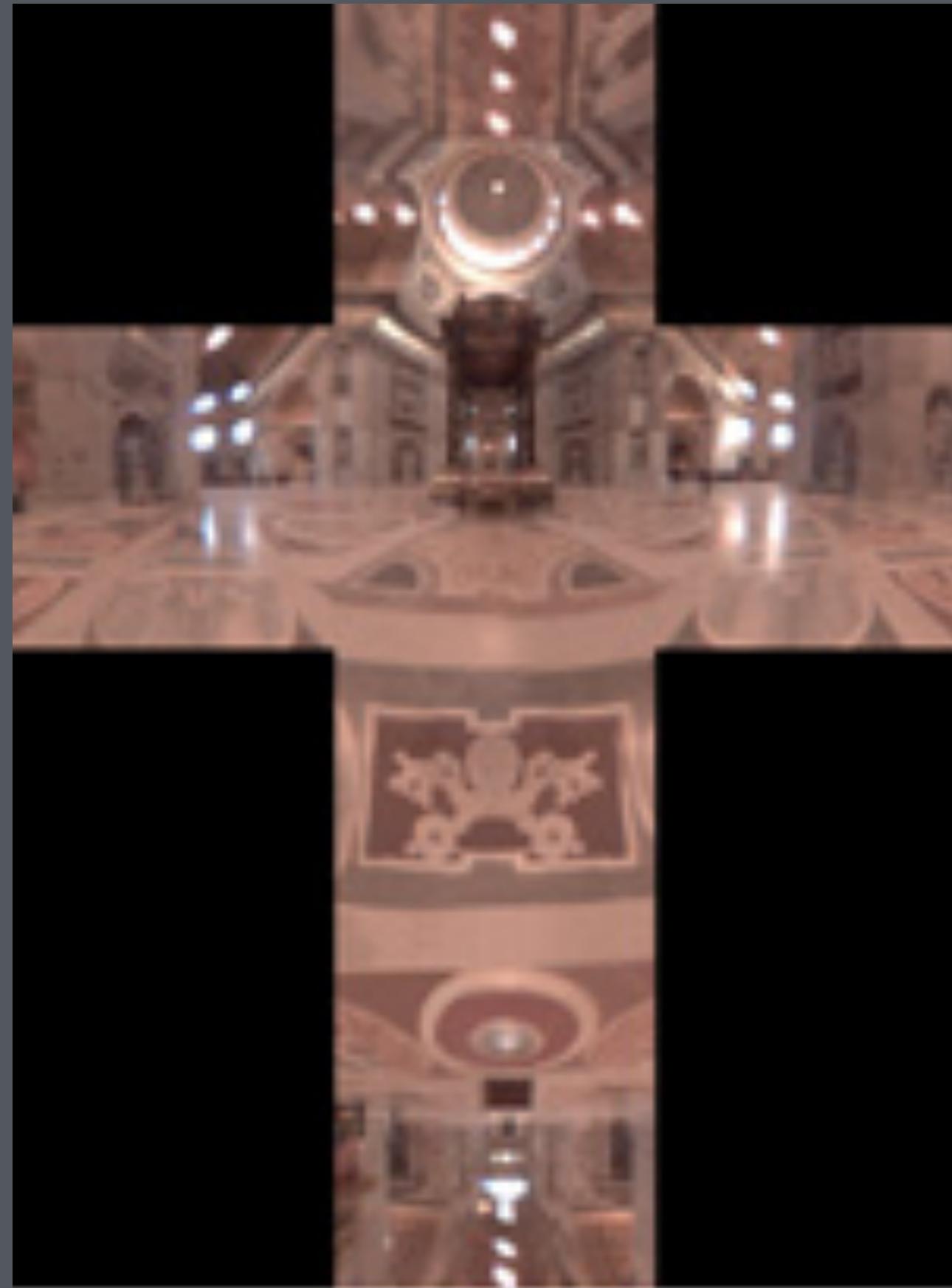


environment map

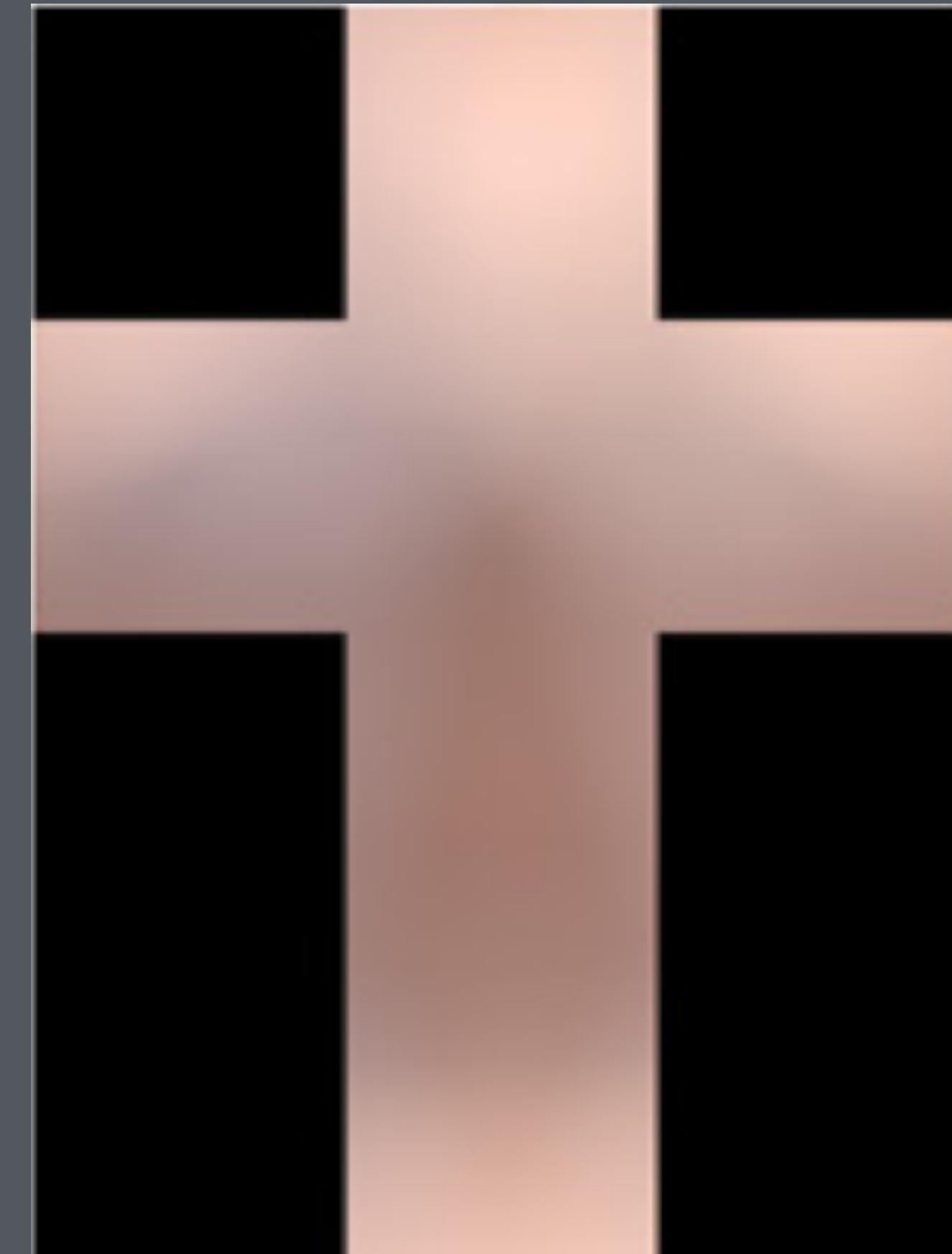


prefiltered for Phong

Irradiance environment map

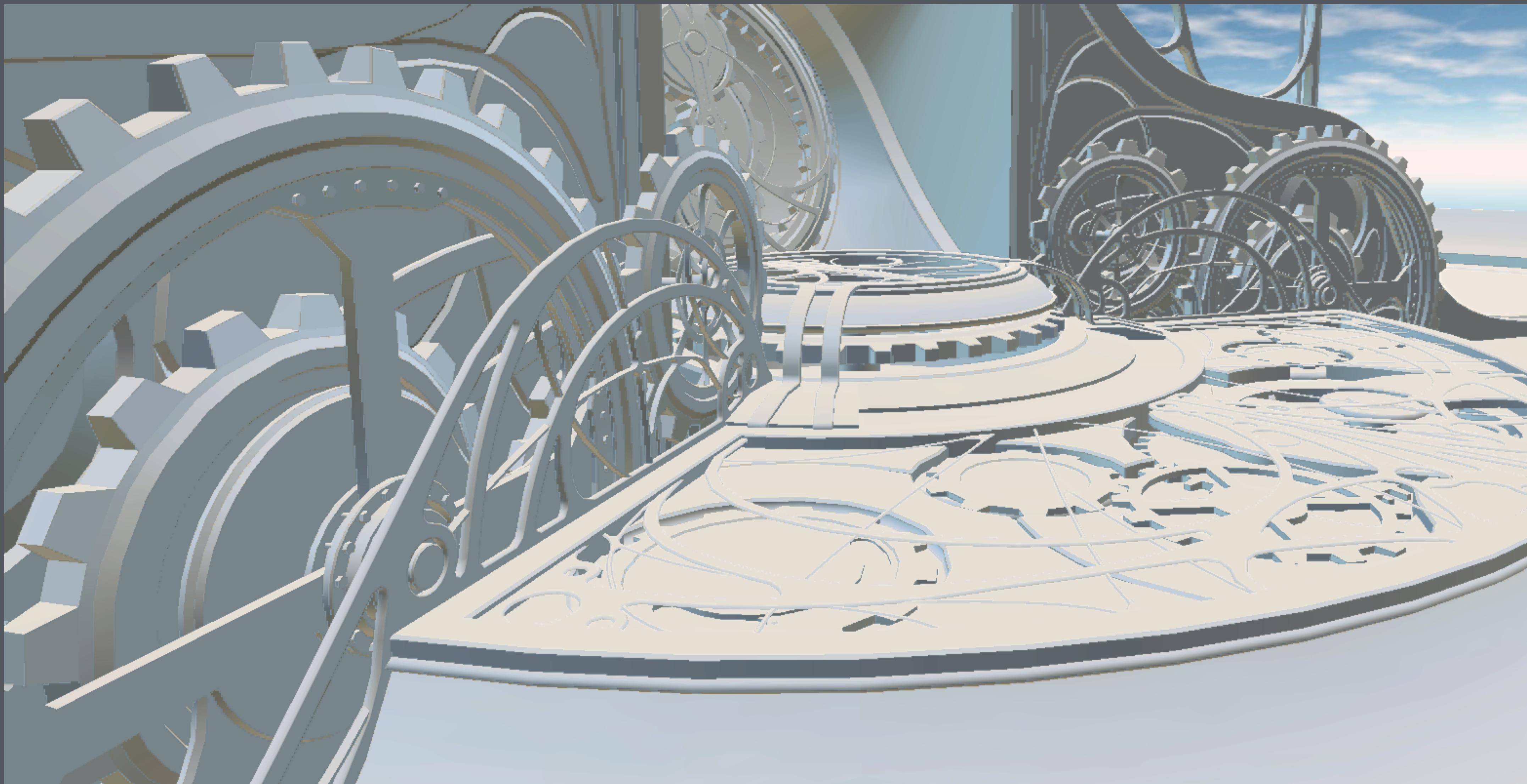


environment map



irradiance map

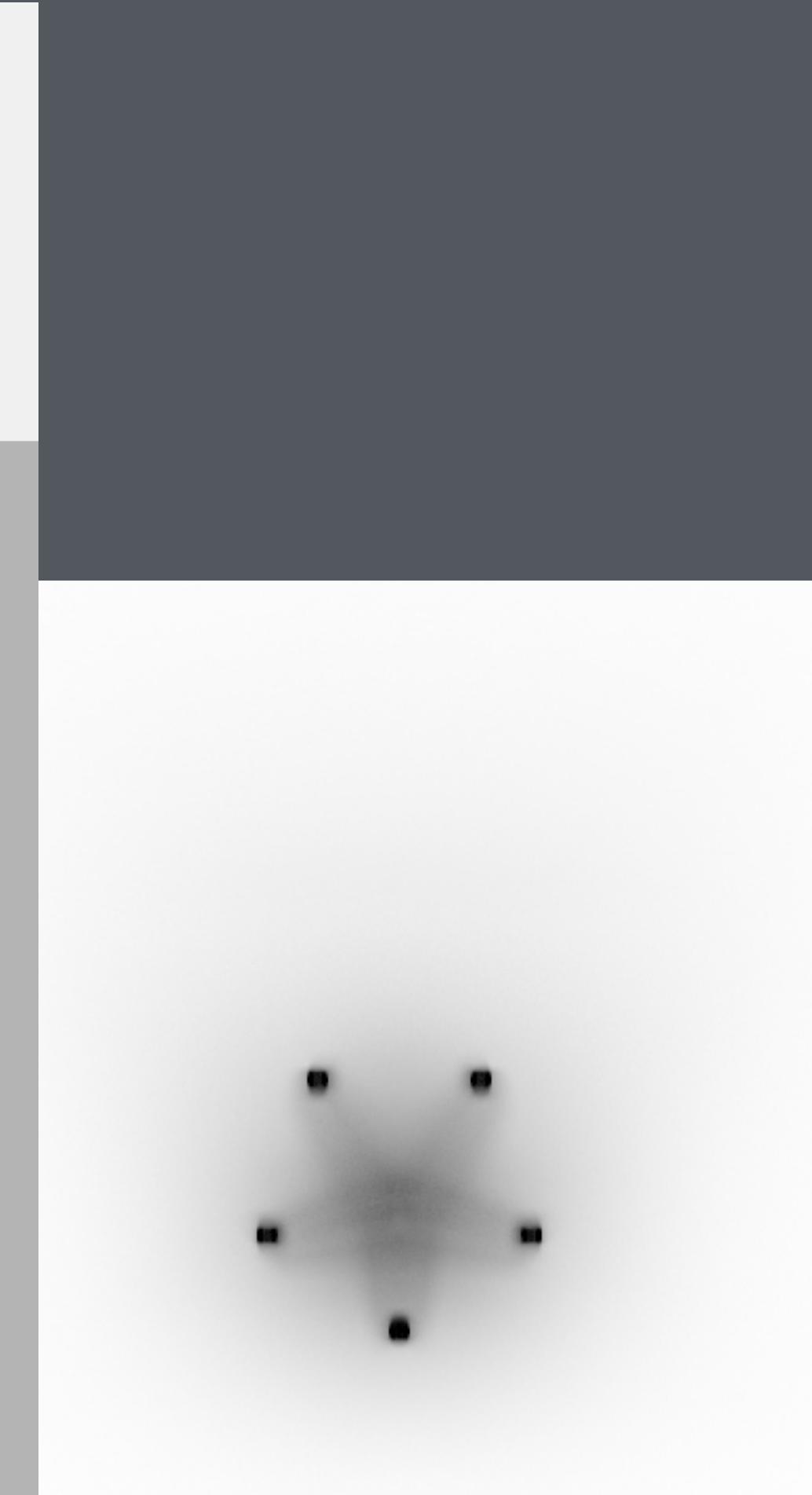
Irradiance map illumination



Shadow baking



Rendering with no shadows,
darker diffuse floor



Irradiance texture computed
using rectangular light

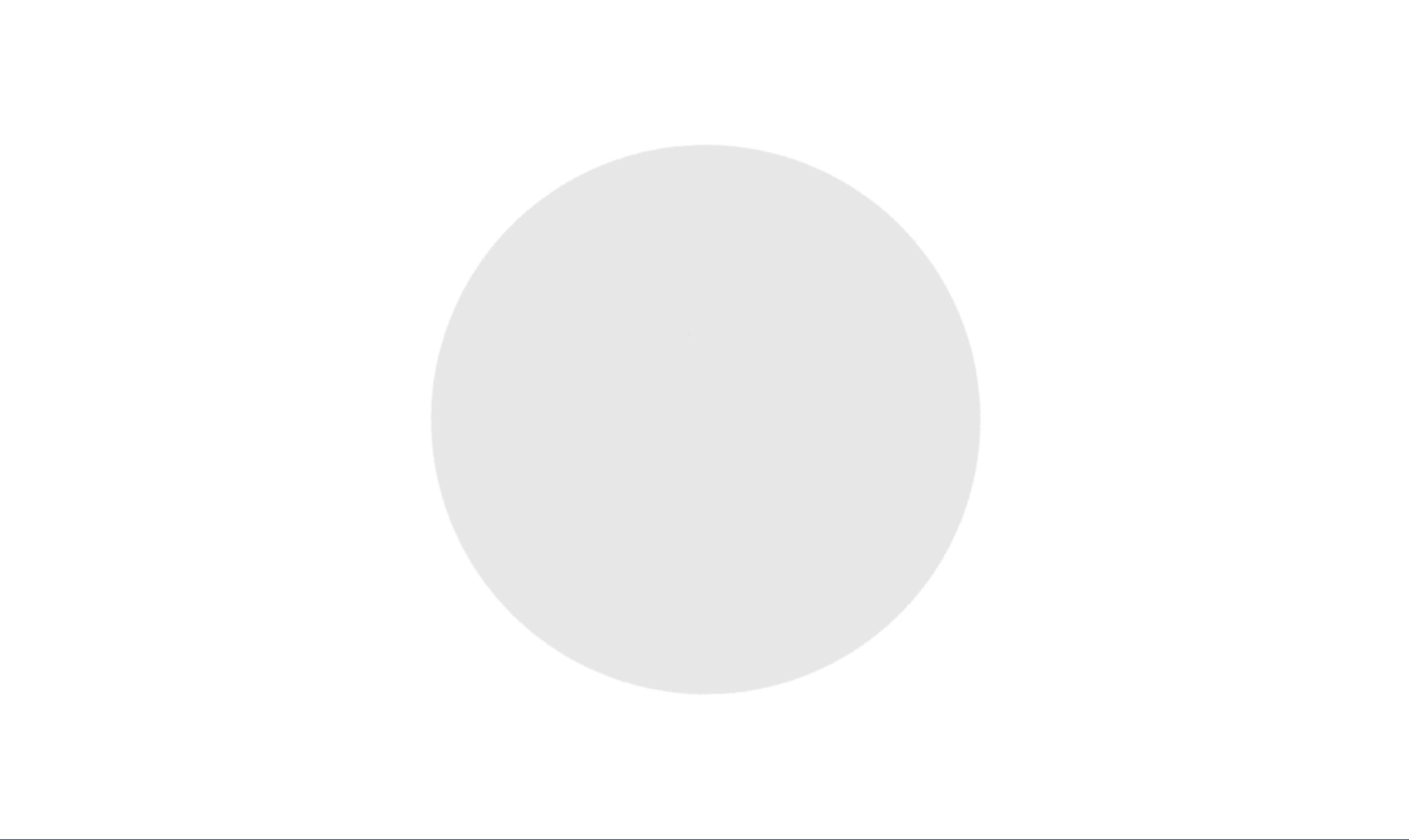


Floor shaded with irradiance
from shadow texture

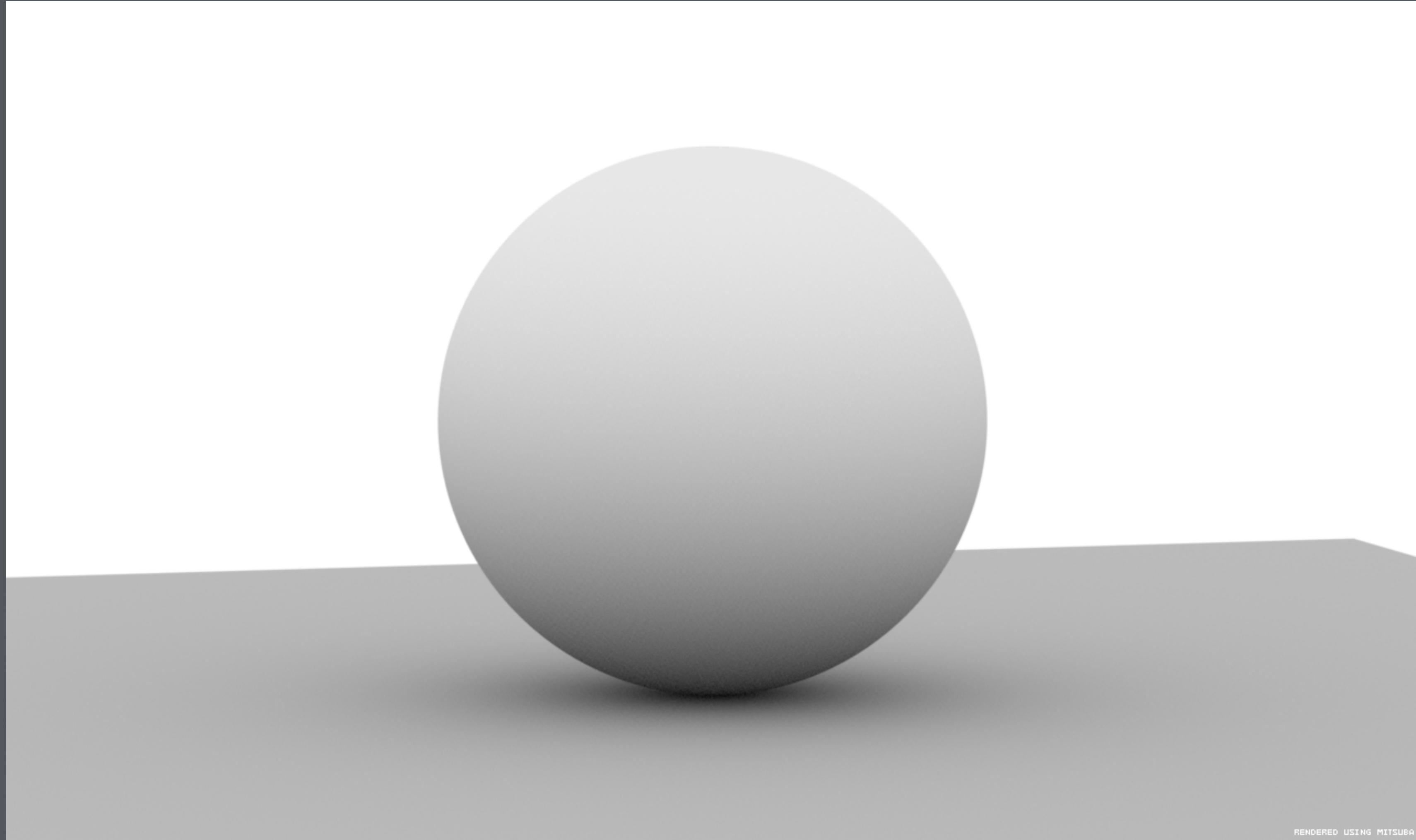
Irradiance map illumination



McGuire et al. HPG '11 [10.1145/2018323.2018327](https://doi.org/10.1145/2018323.2018327)



a convex diffuse object in a constant-radiance environment

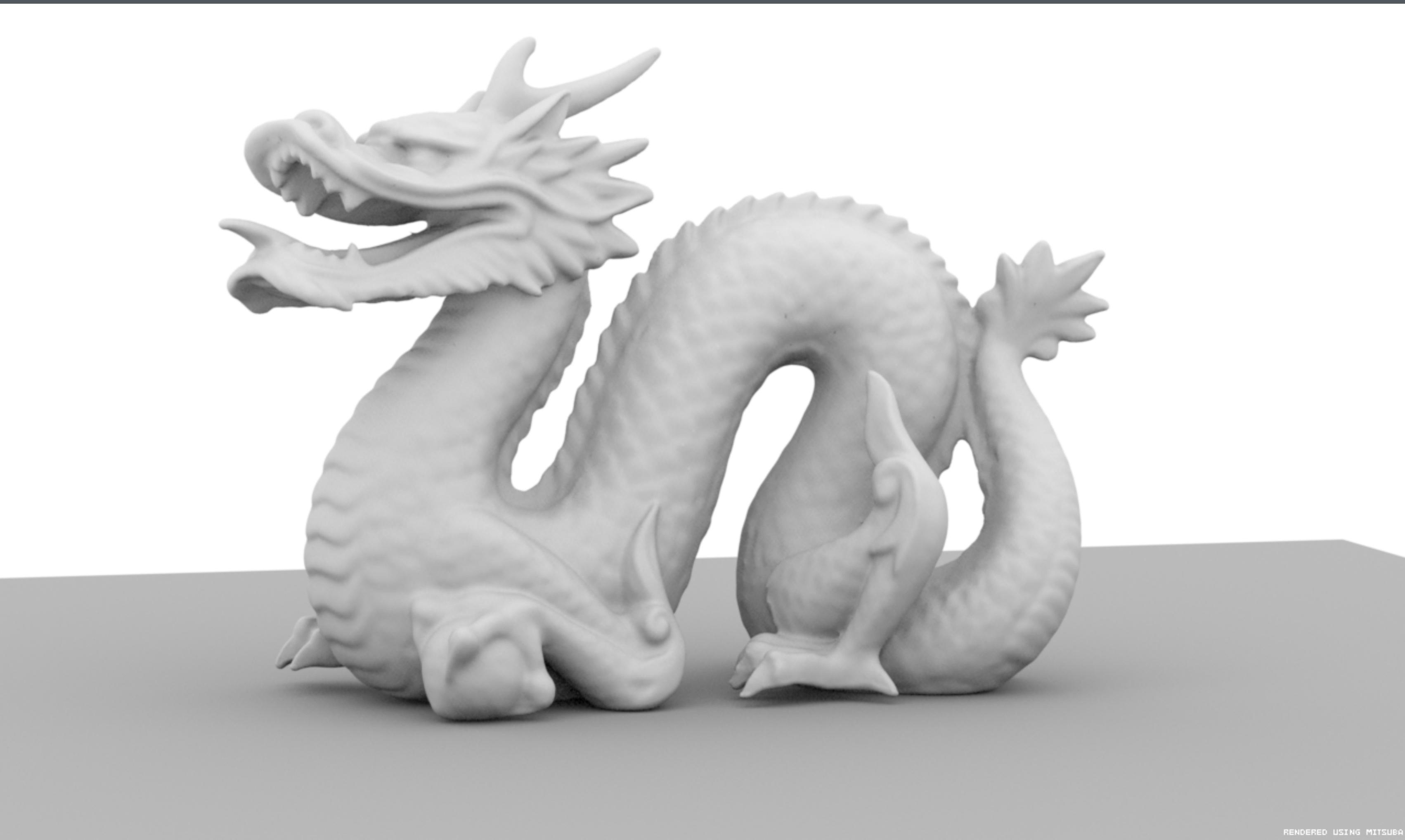


RENDERED USING MITSUEA

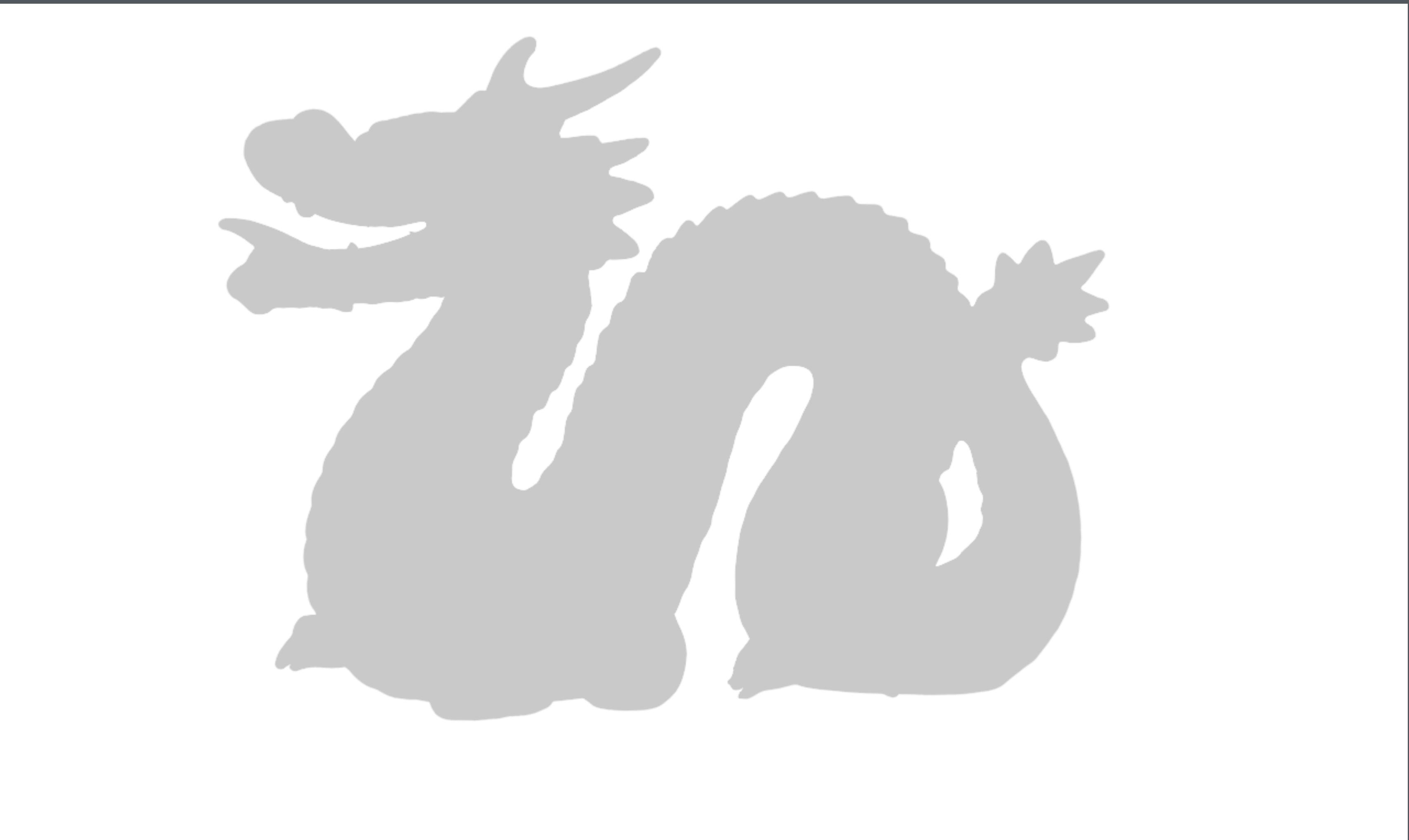
a non-convex diffuse scene under constant-radiance illumination



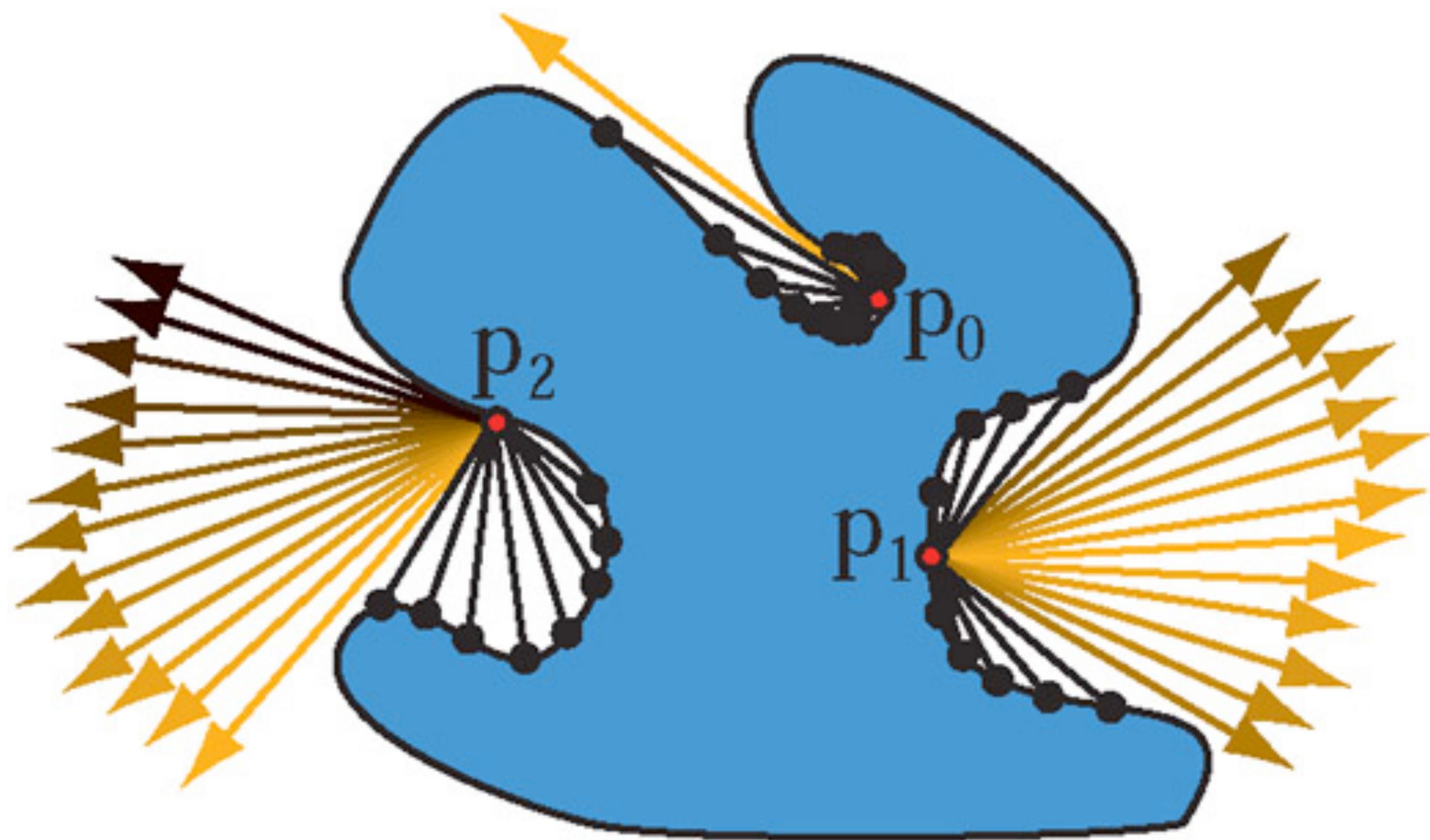
a non-convex diffuse object in a constant-radiance environment



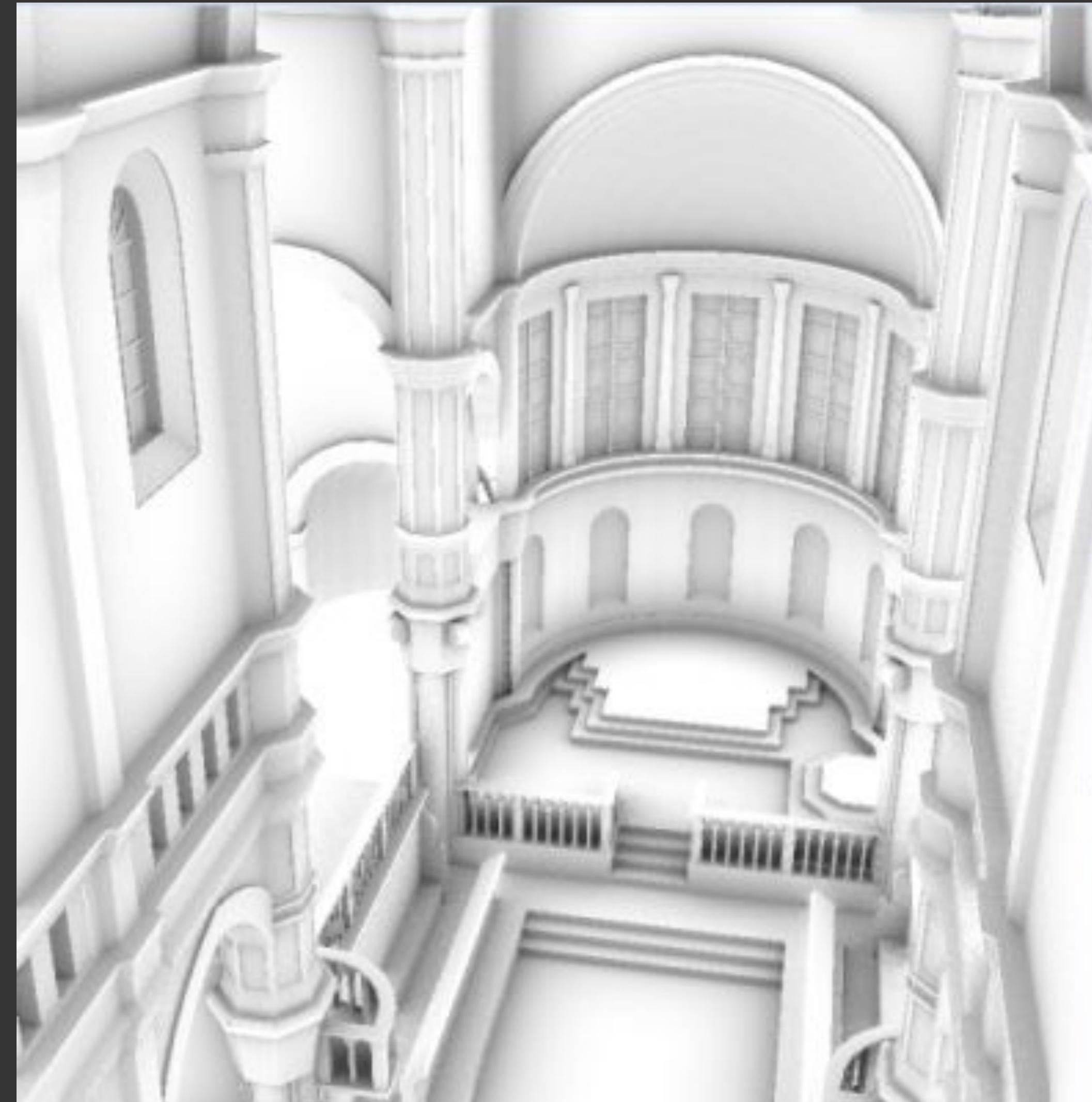
a non-convex diffuse scene under constant-radiance illumination



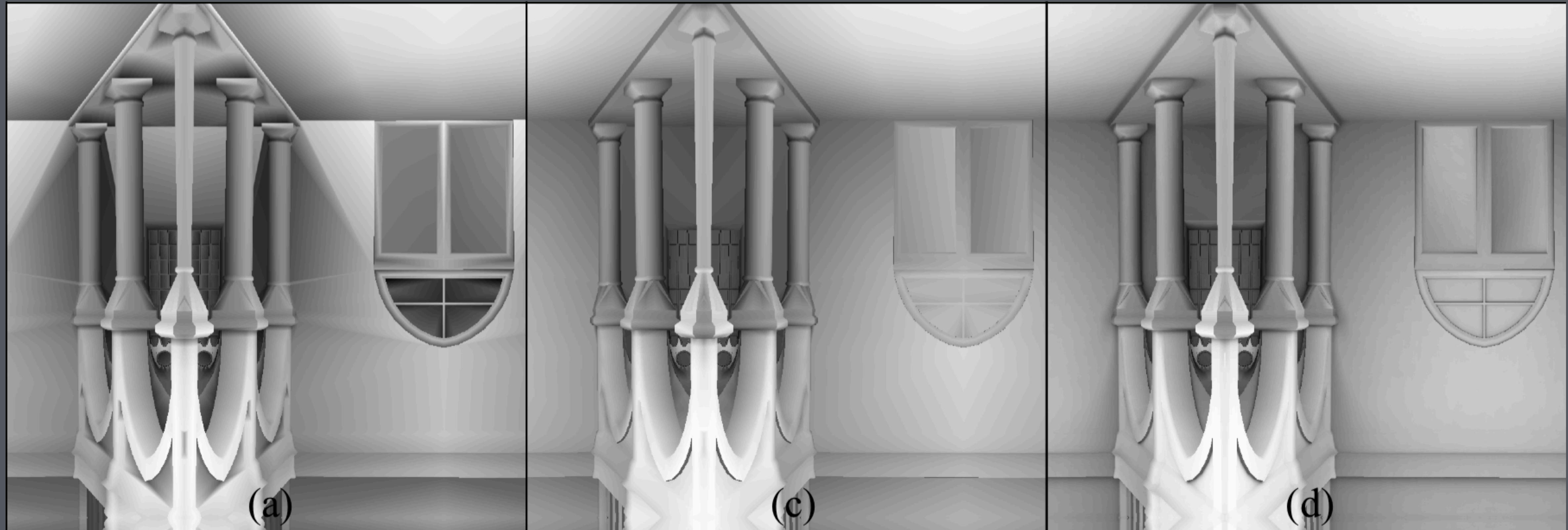
object in a constant-radiance environment with no shadowing



AO Maps



Ray traced vertex AO

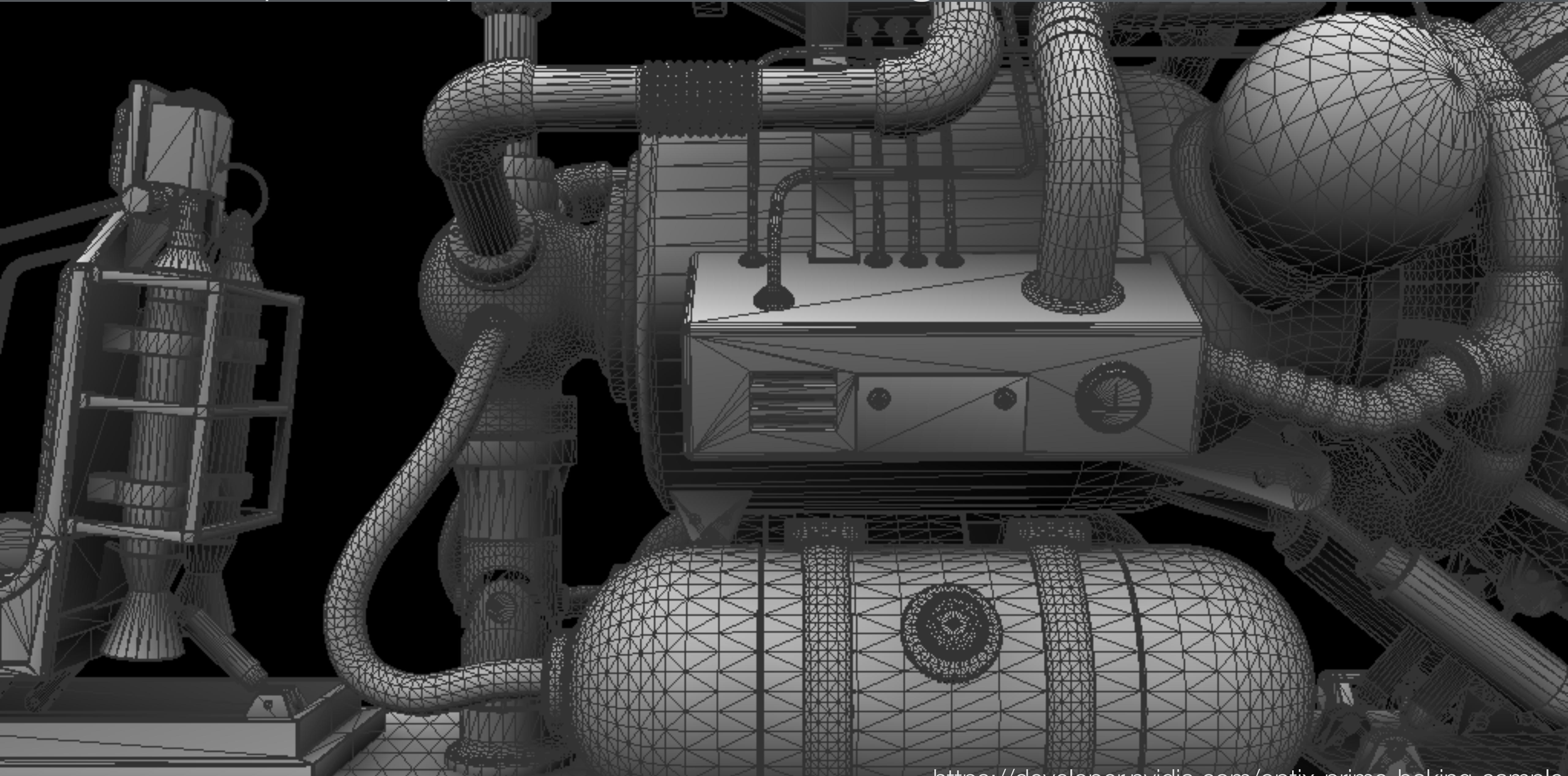


ambient occlusion sampled at vertices,
interpolated as vertex color

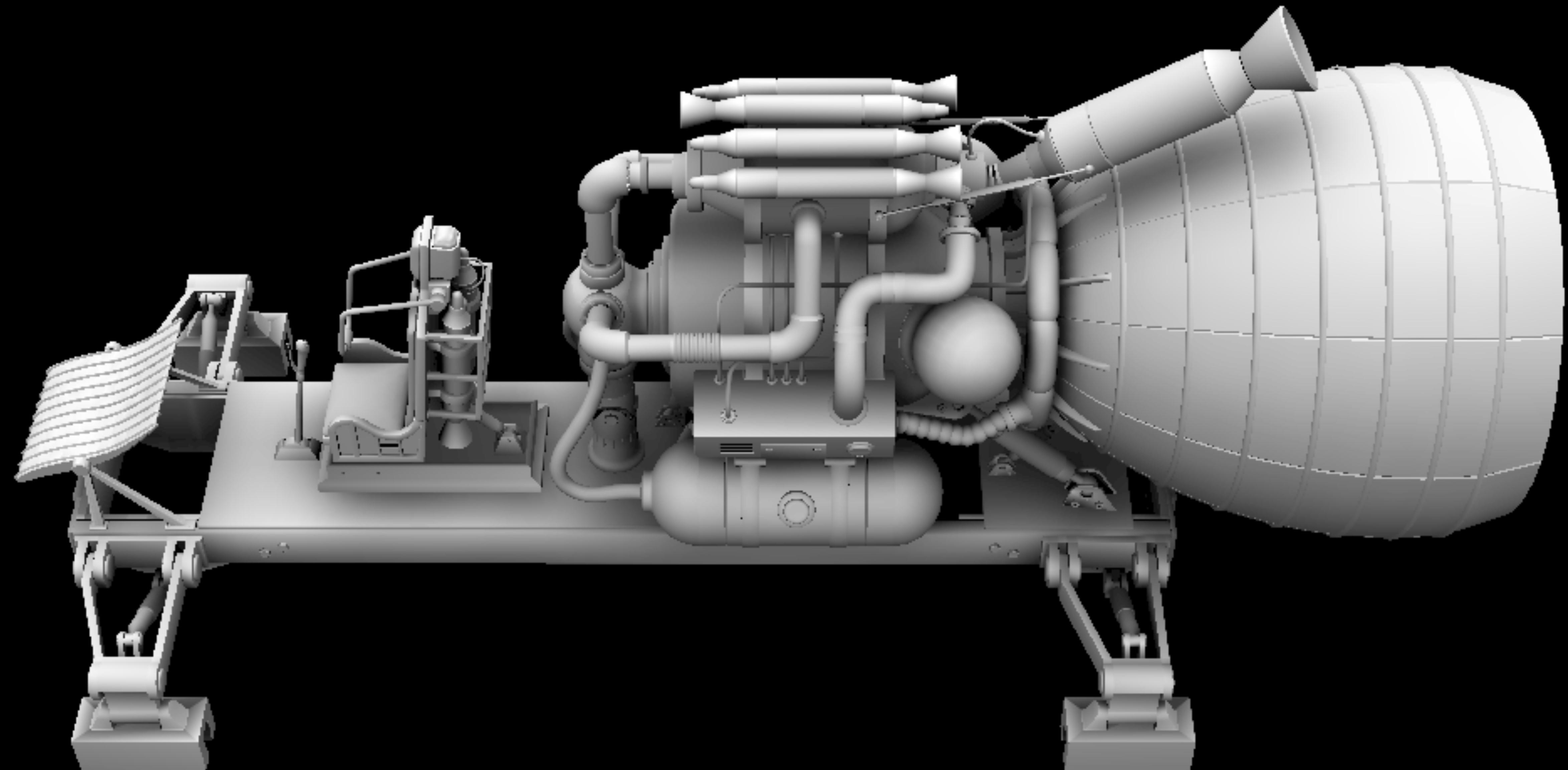
ambient occlusion sampled inside
triangles, vertex values fit to samples,
interpolated as vertex color

ambient occlusion computed at each
pixel (ground truth)

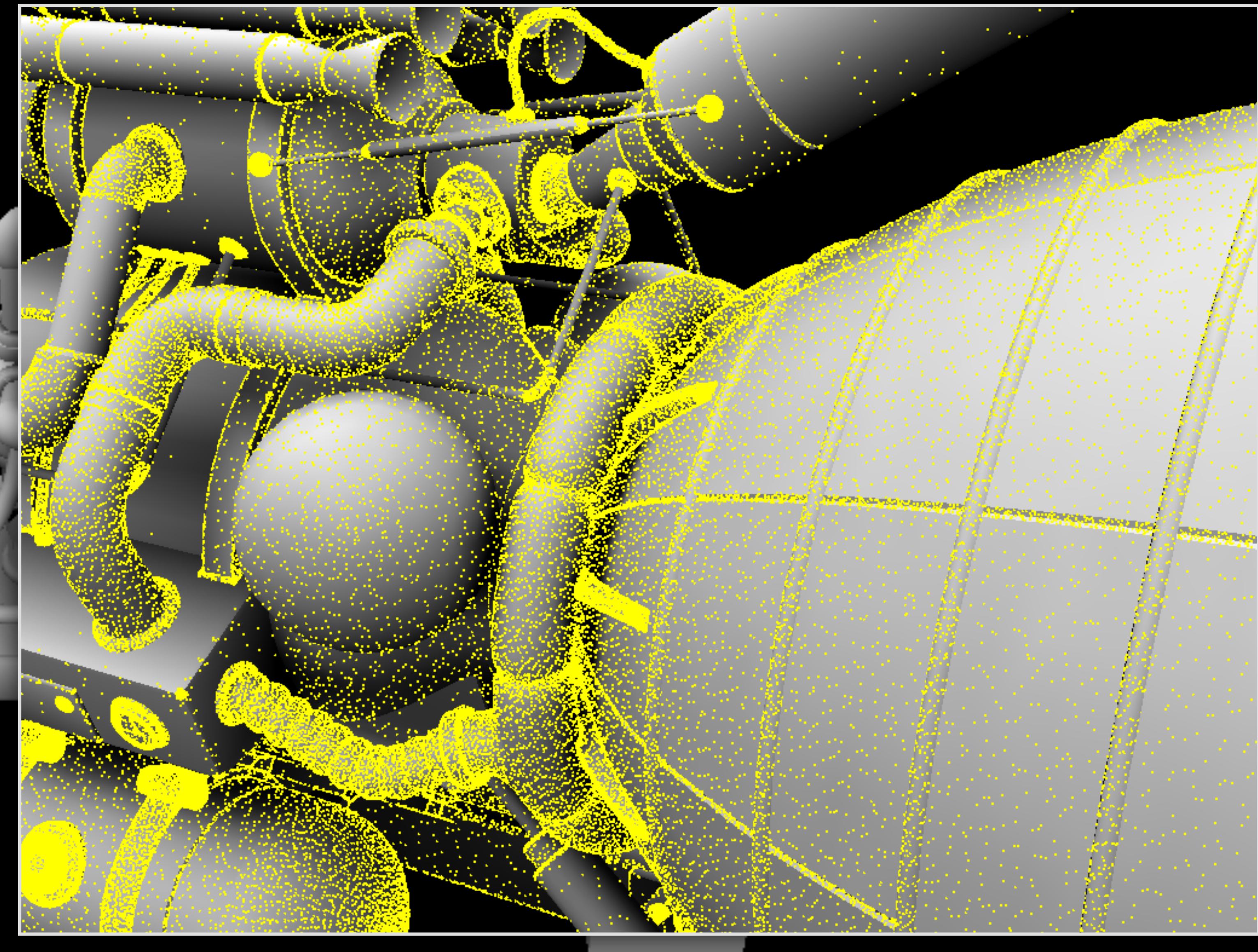
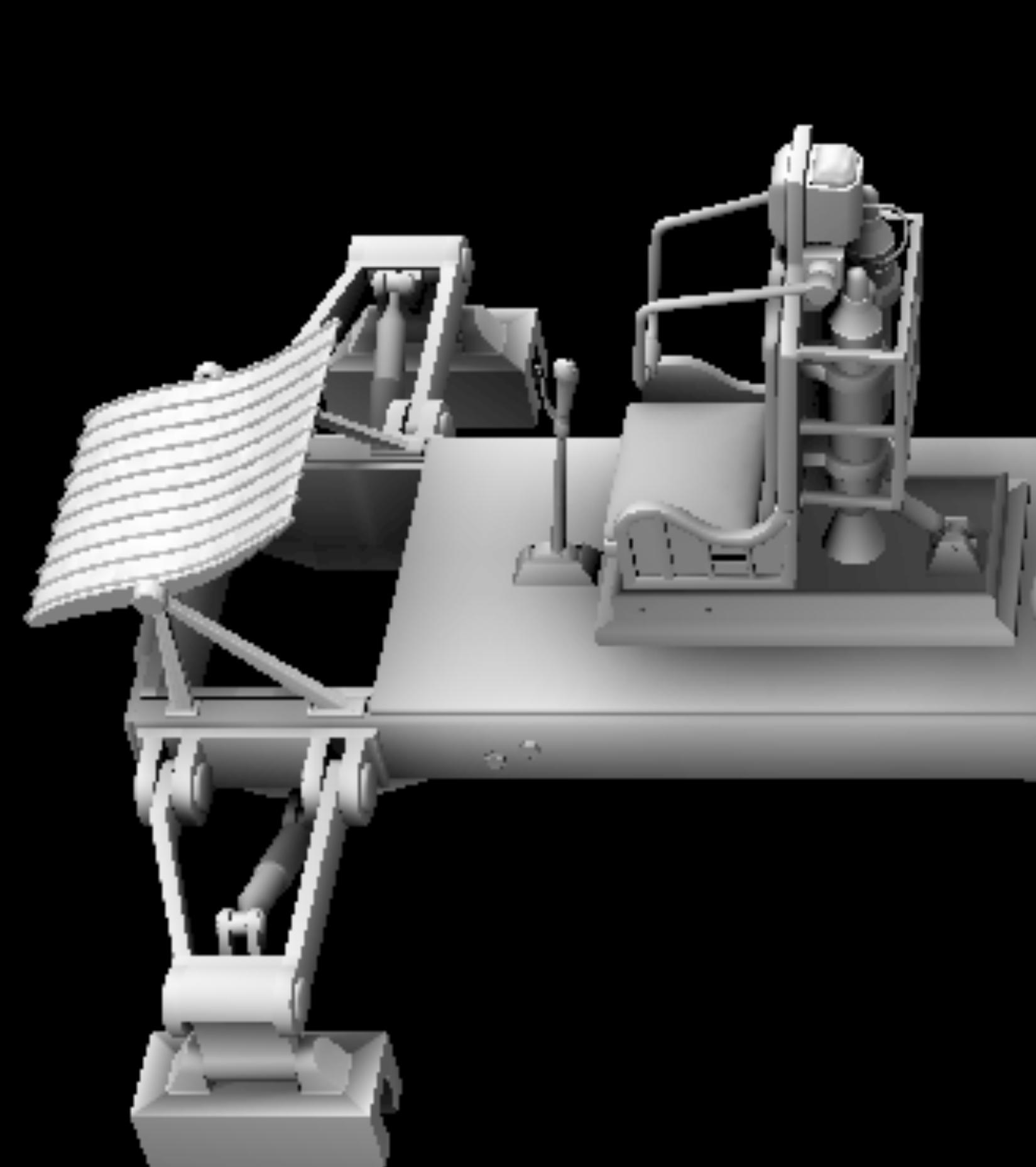
NVIDIA OptiX implementation images



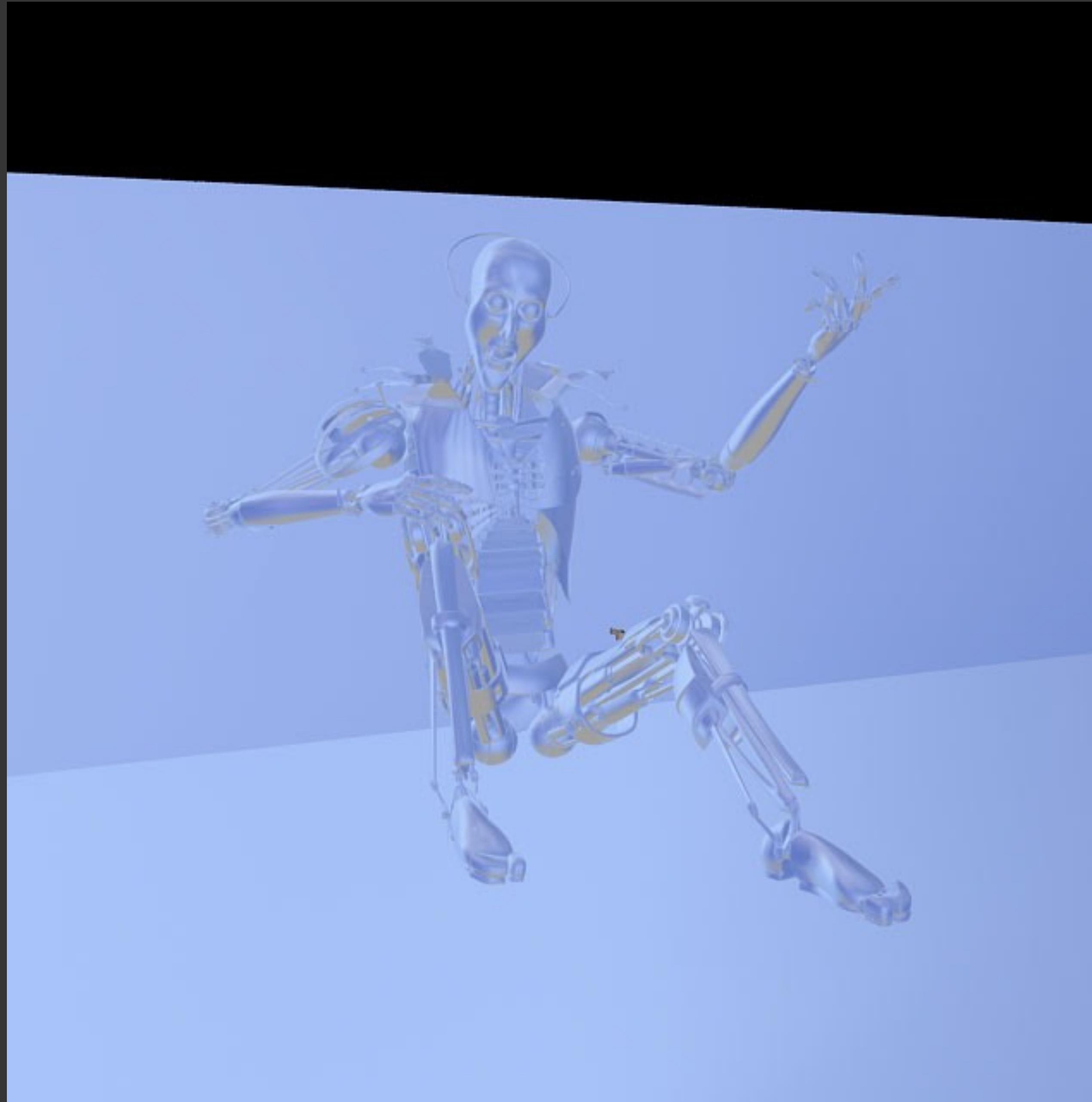
NVIDIA OptiX implementation images



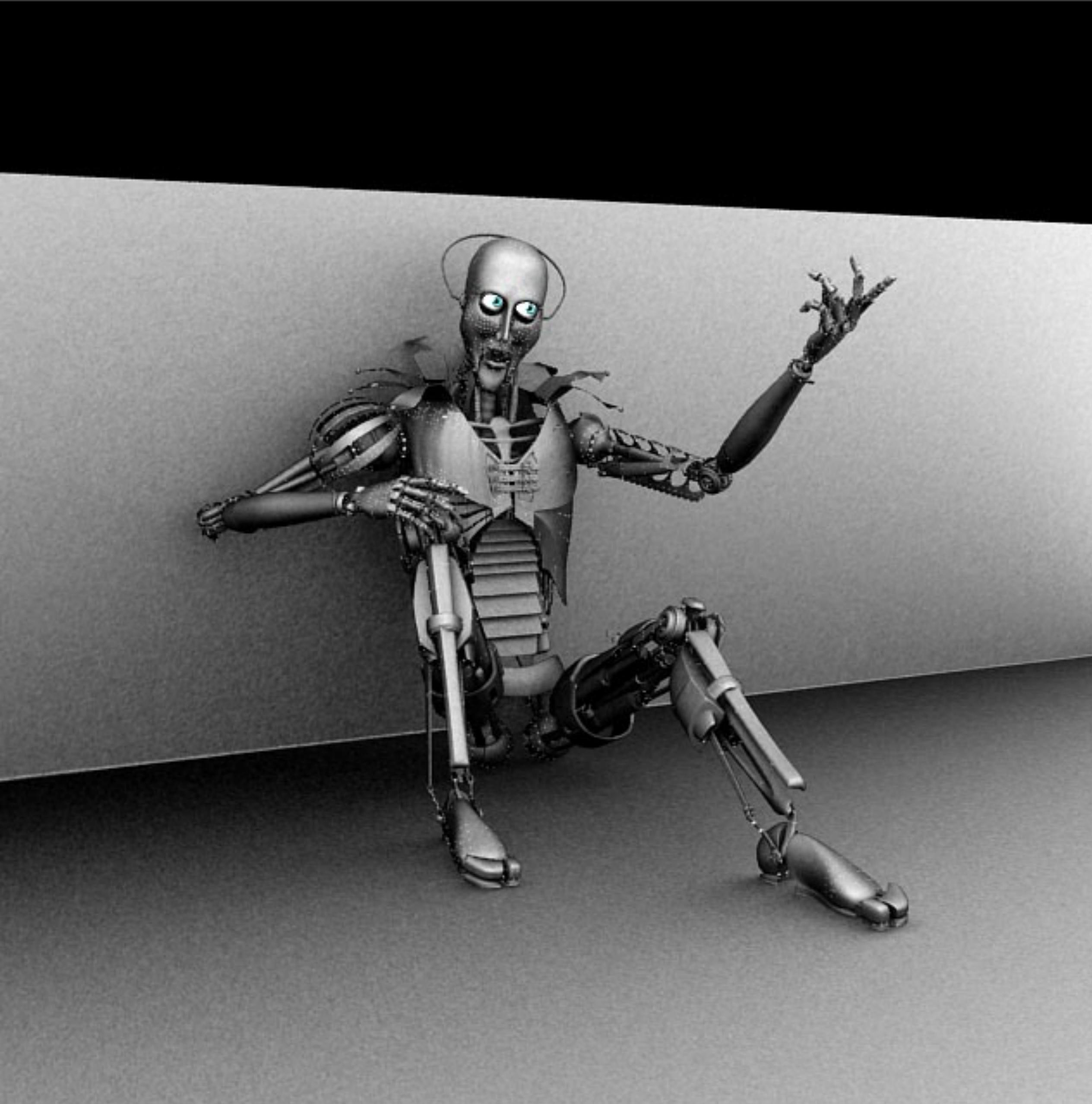
NVIDIA OptiX implementation images



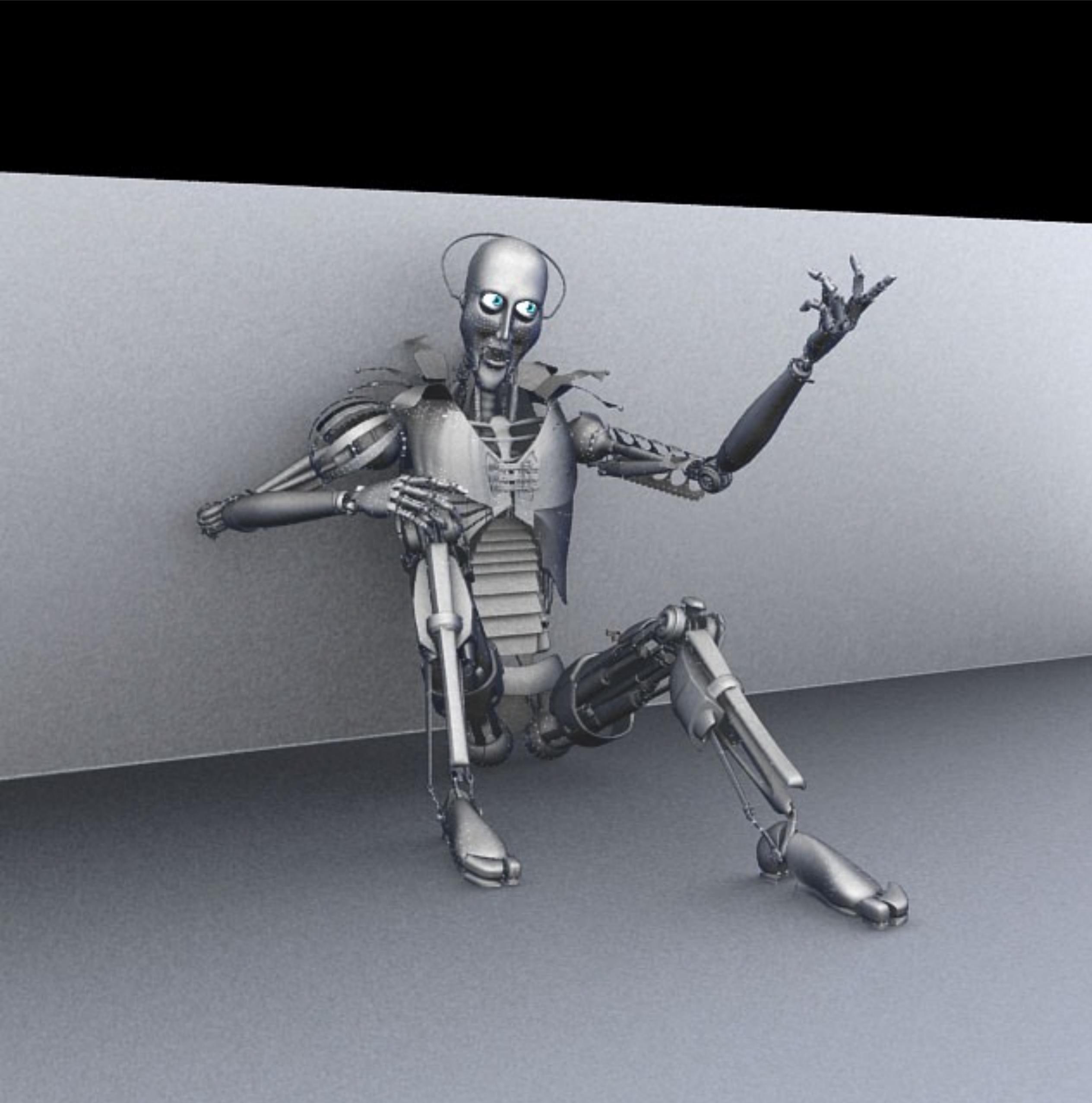
EnvMap

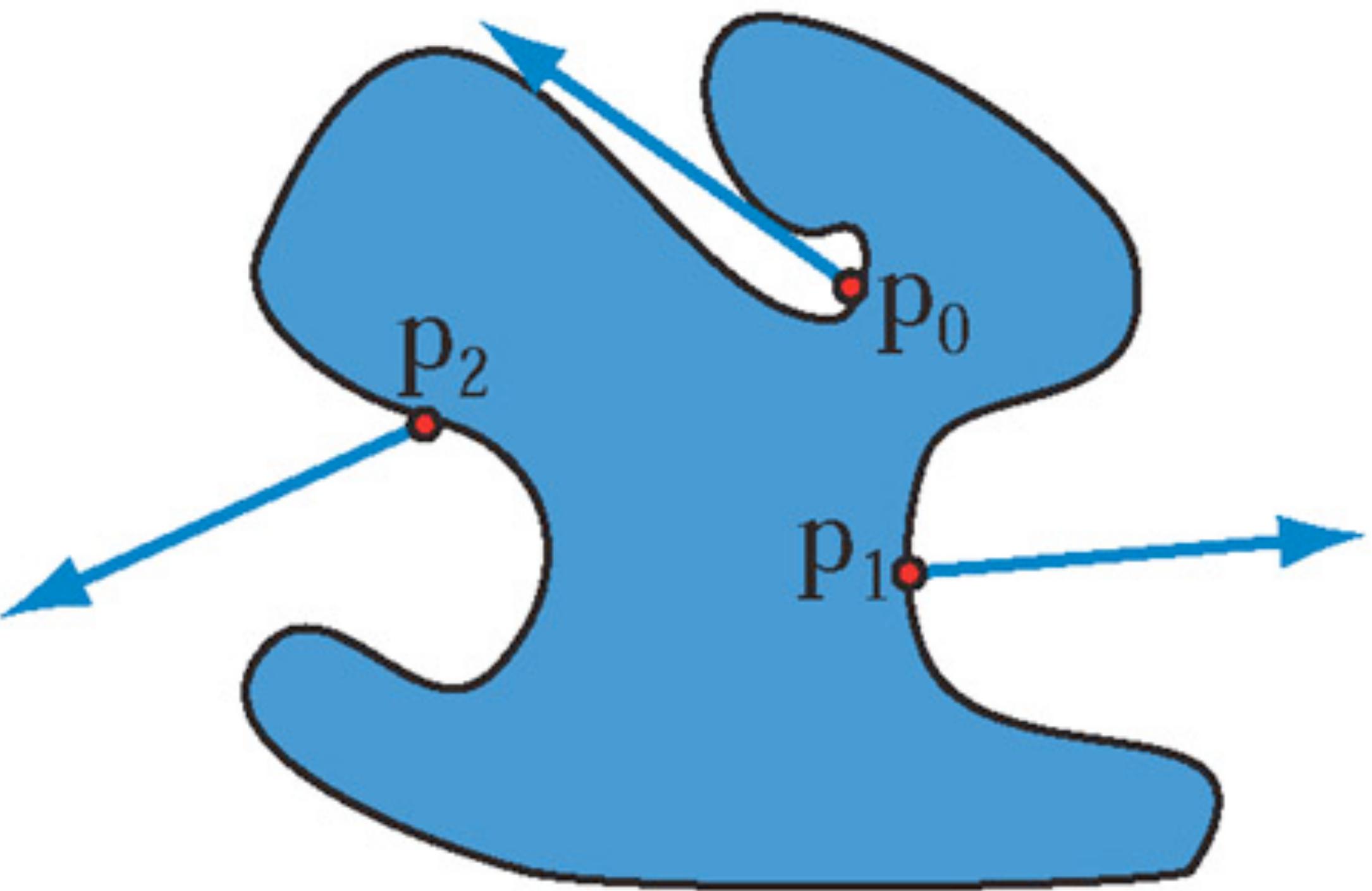
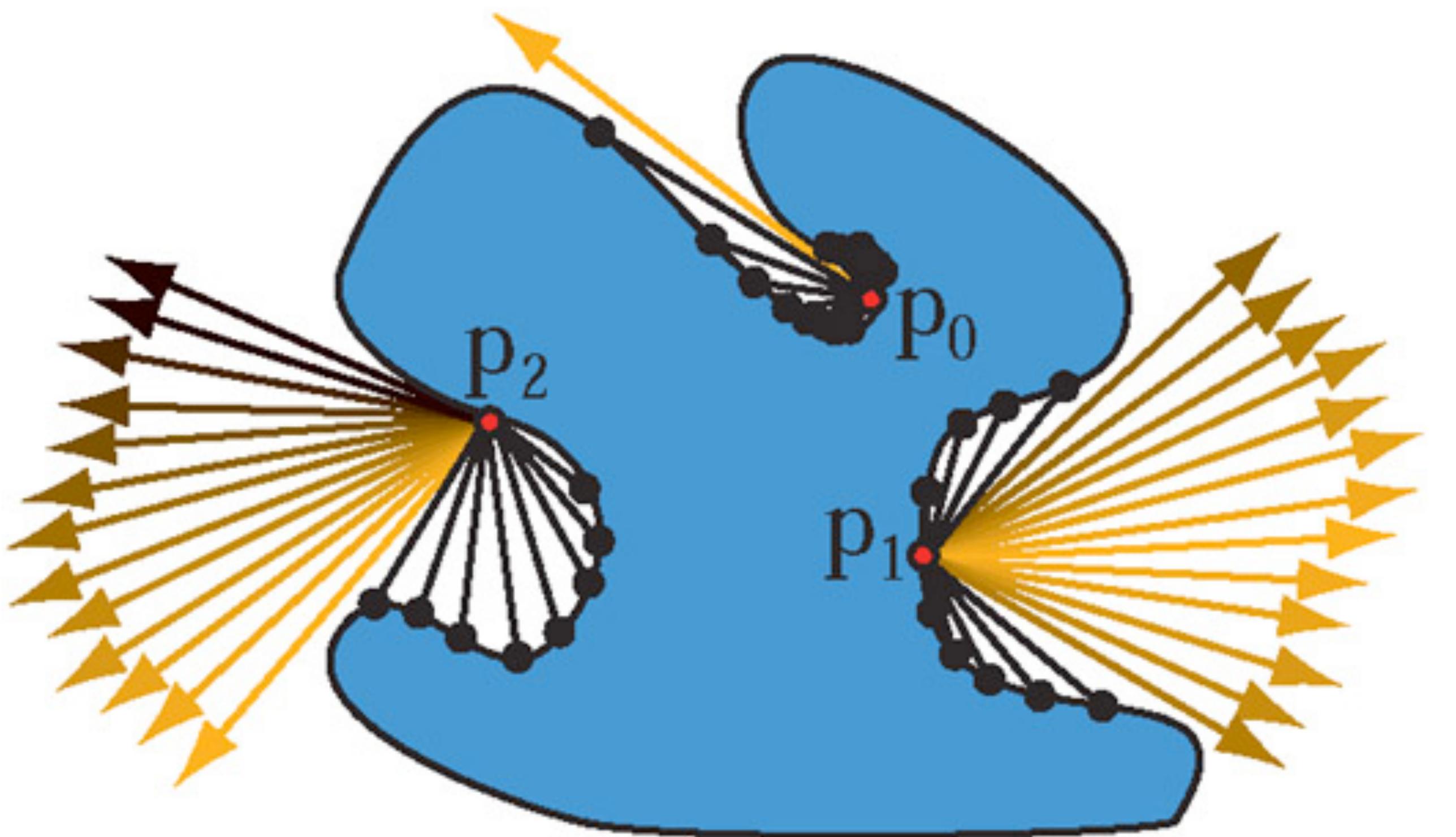


AO



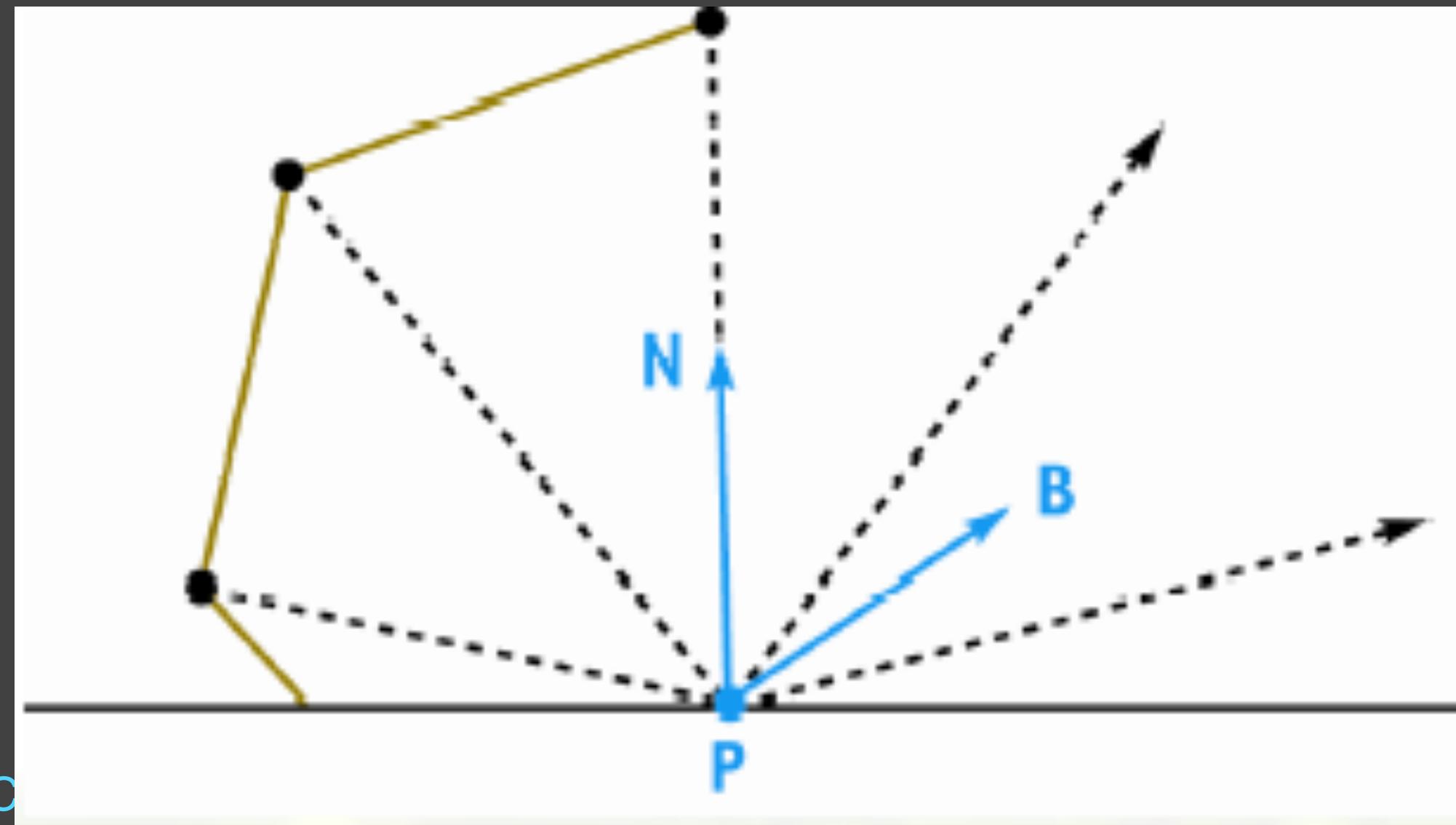
Total





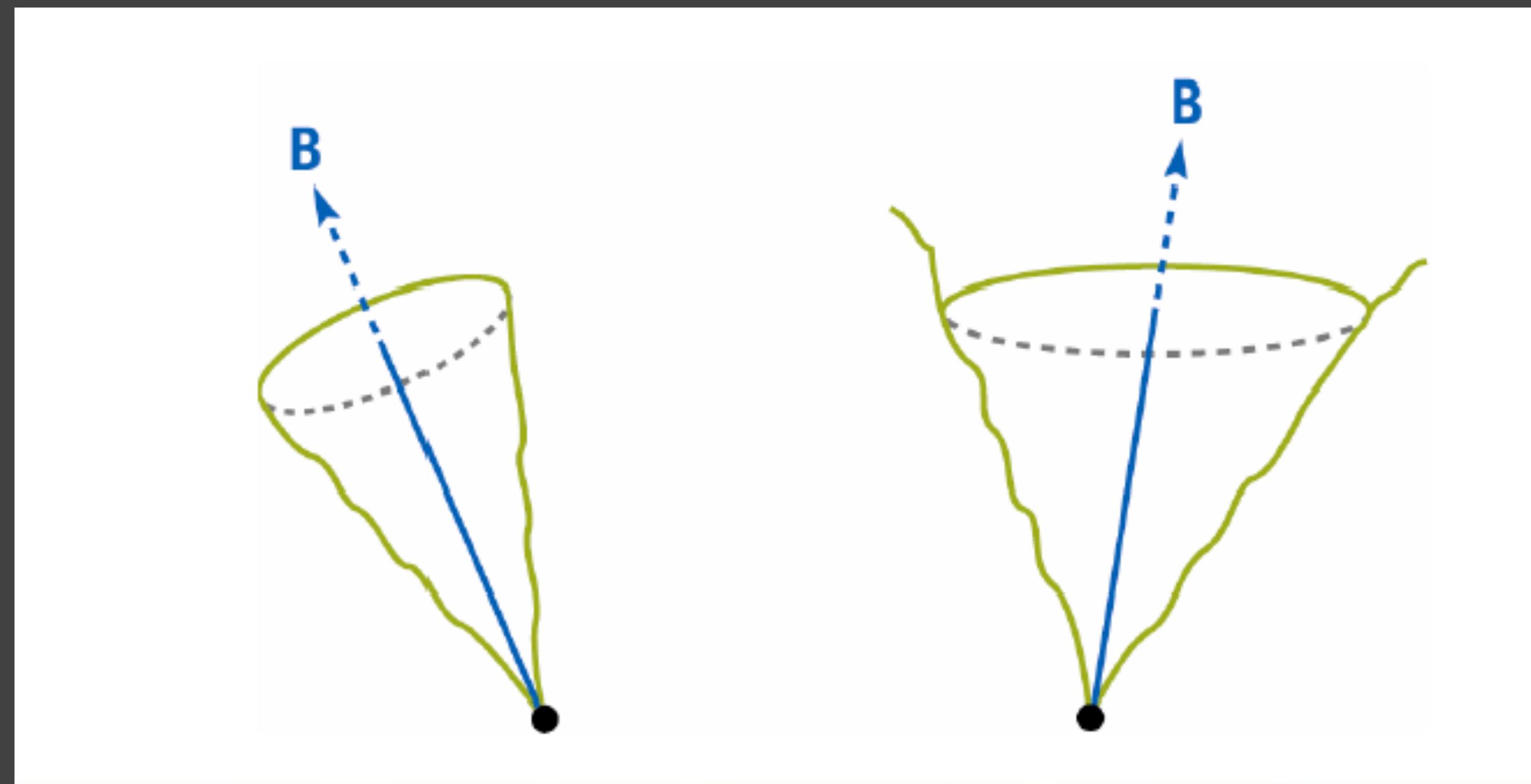
Ambient Occlusion: Improvement

- At each point find
 - Fraction of hemisphere that is occluded
 - Also, average unoccluded direction B
 - (bent normal)
 - Use B for lighting (see later)



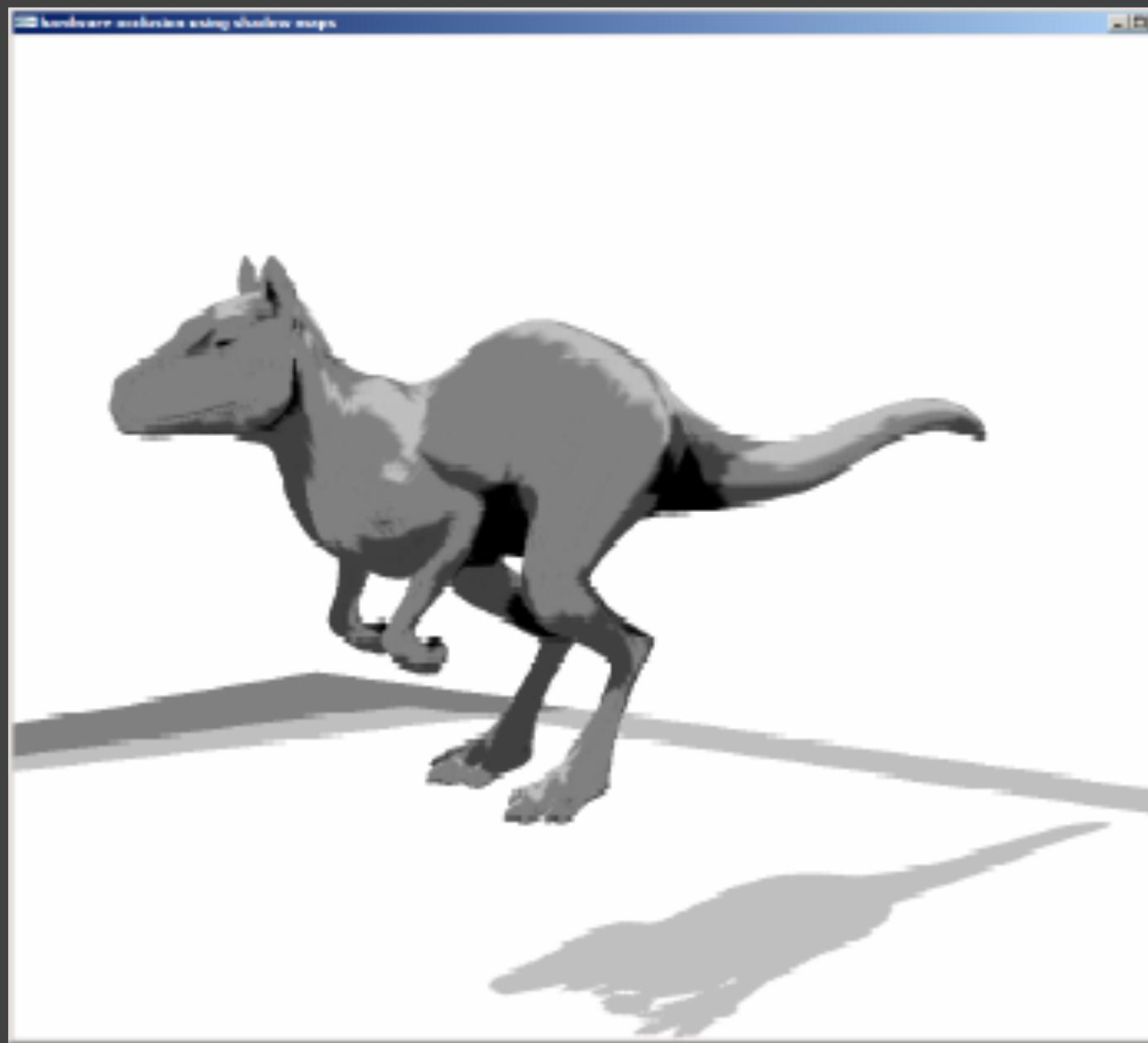
What about B?

- The unoccluded direction gives an idea of where the main illumination is coming from



Computing AO using shadow maps

- Create shadow maps from N point lights on sphere
- Check visibility of point wrt each light and determine occlusion: accumulation buffer



4 samples

slide courtesy of Kavita Bala, Cornell University



32 samples

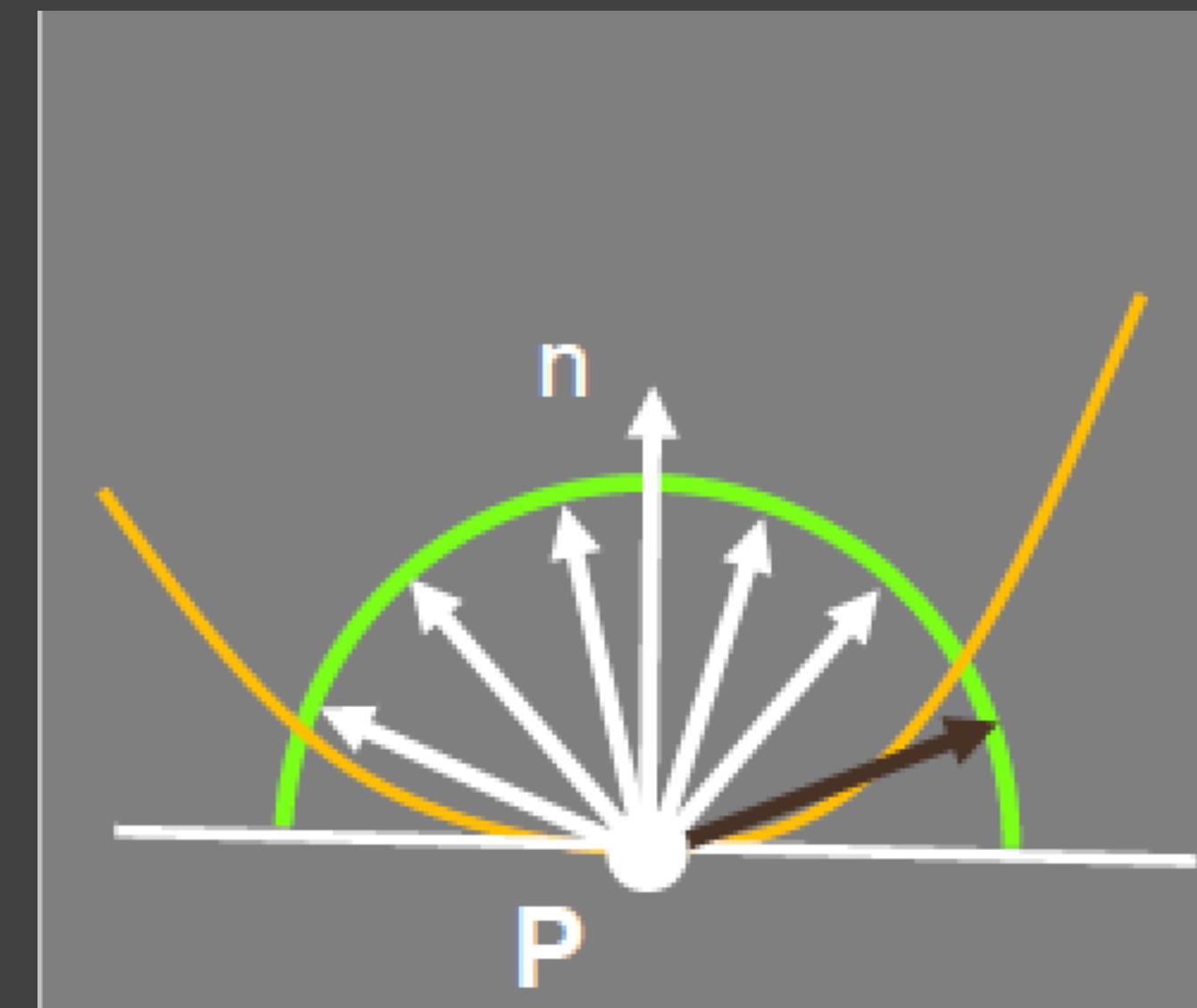
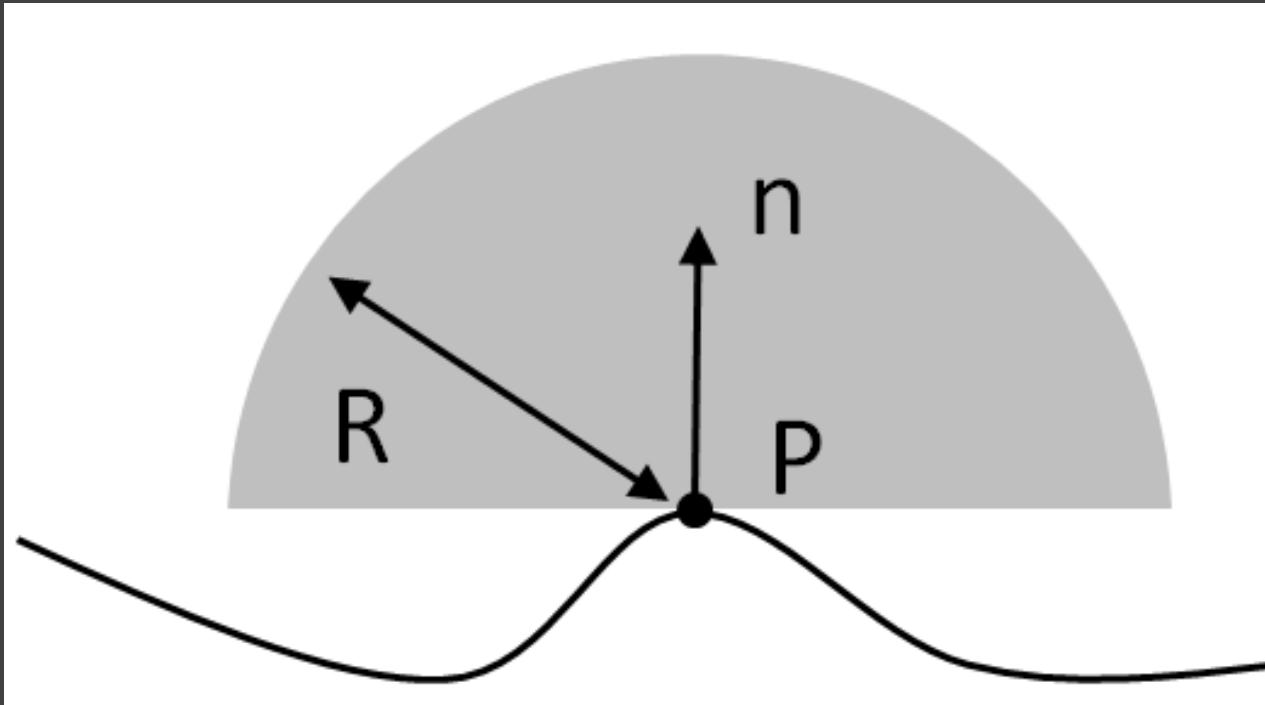
Computing the values: SM



512 samples

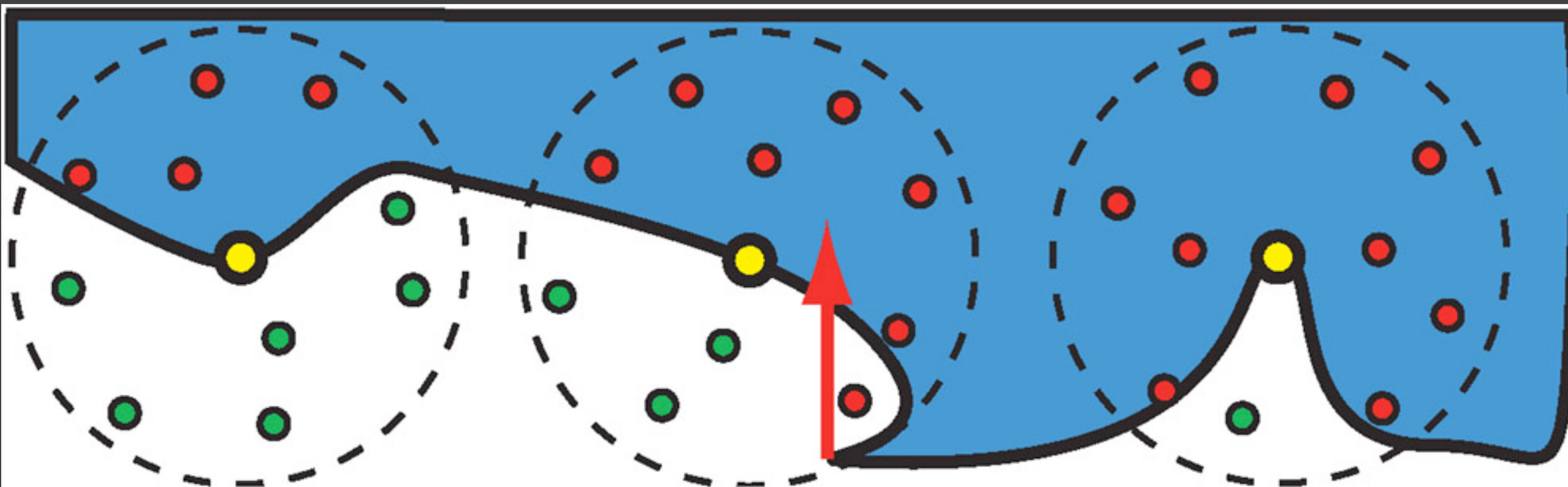
SSAO

- Restrict the hemisphere
 - Why? Think of AO in box
- Typically add a drop-off as you get to the hemisphere boundary



Crytek for Crisis: SSAO

- Take z-buffer
- Consider sphere around a point p





Martin Mittring, Crysis GmbH <http://crytek.com/cryengine/presentations/finding-next-gen-cryengine--2>

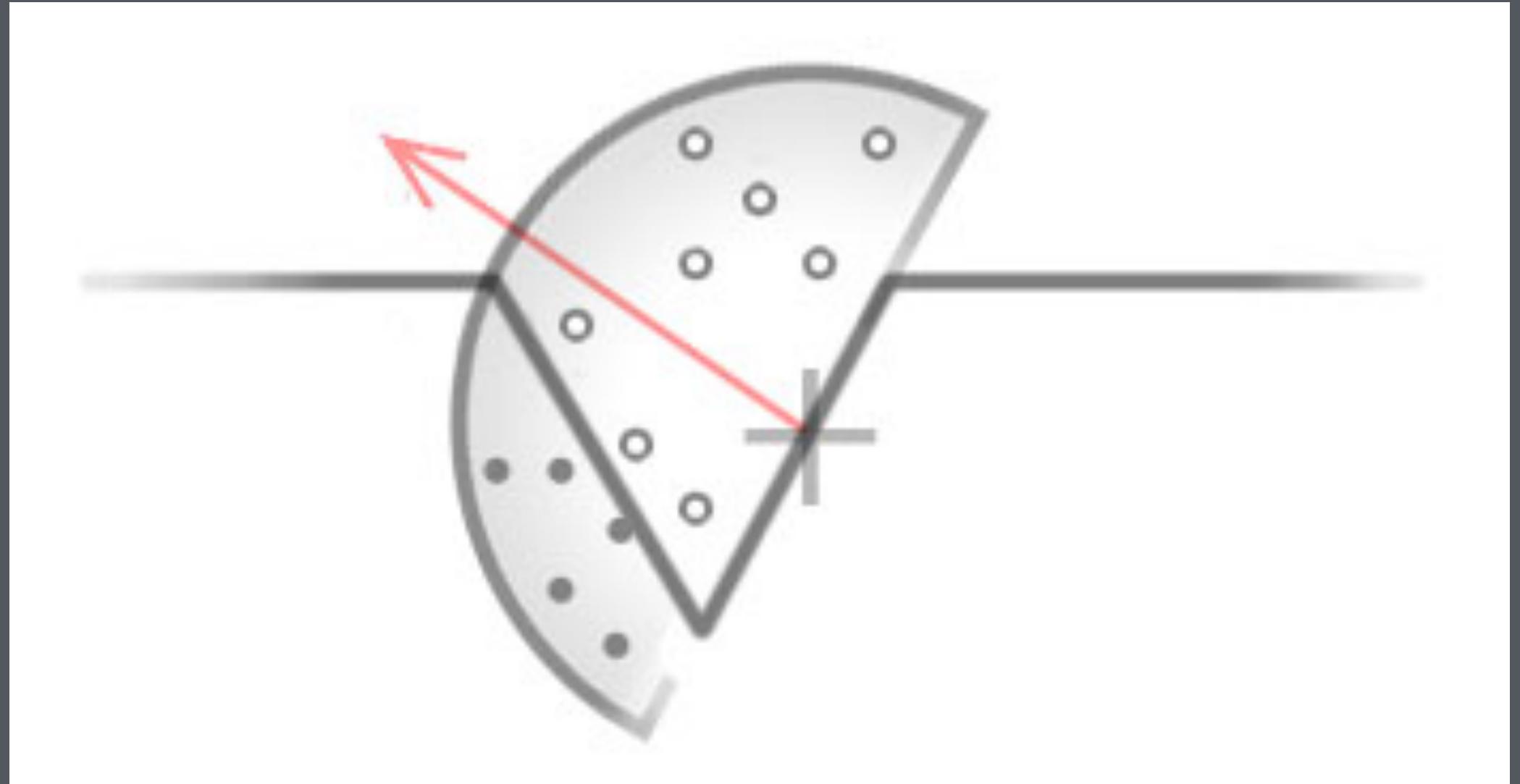
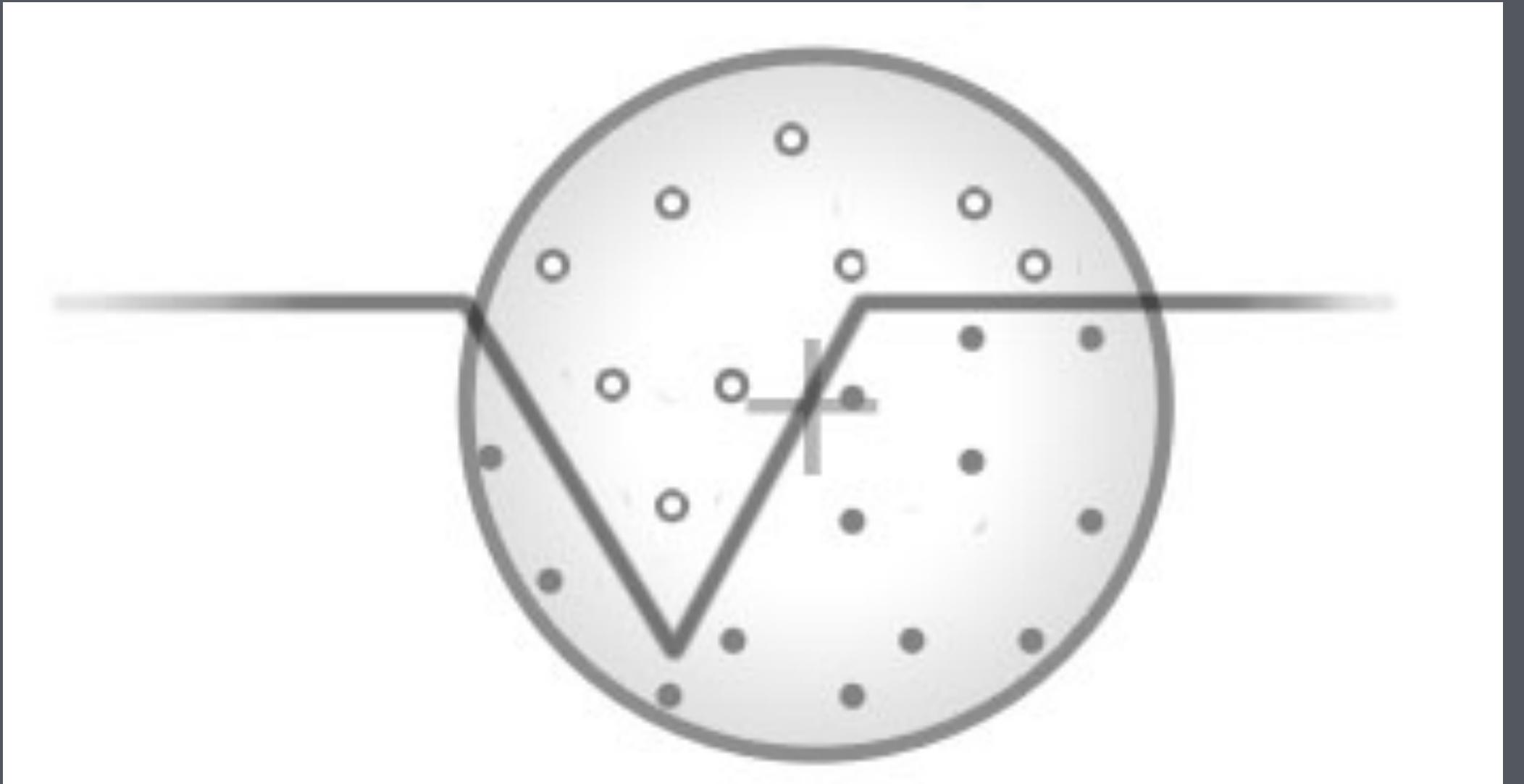


Martin Mittring, Crysis GmbH <http://crytek.com/cryengine/presentations/finding-next-gen-cryengine--2>

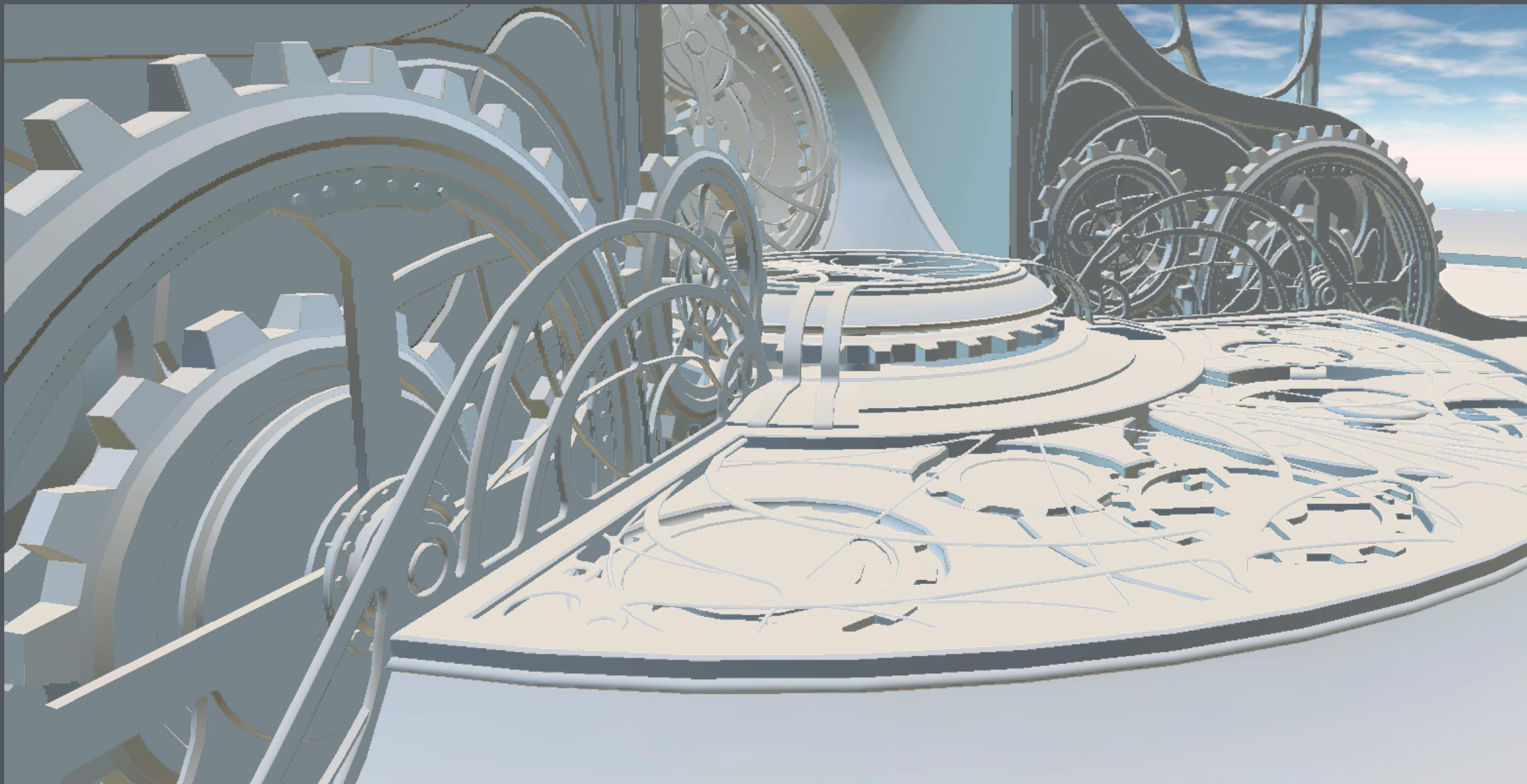


Martin Mittring, Crysis GmbH <http://crytek.com/cryengine/presentations/finding-next-gen-cryengine--2>

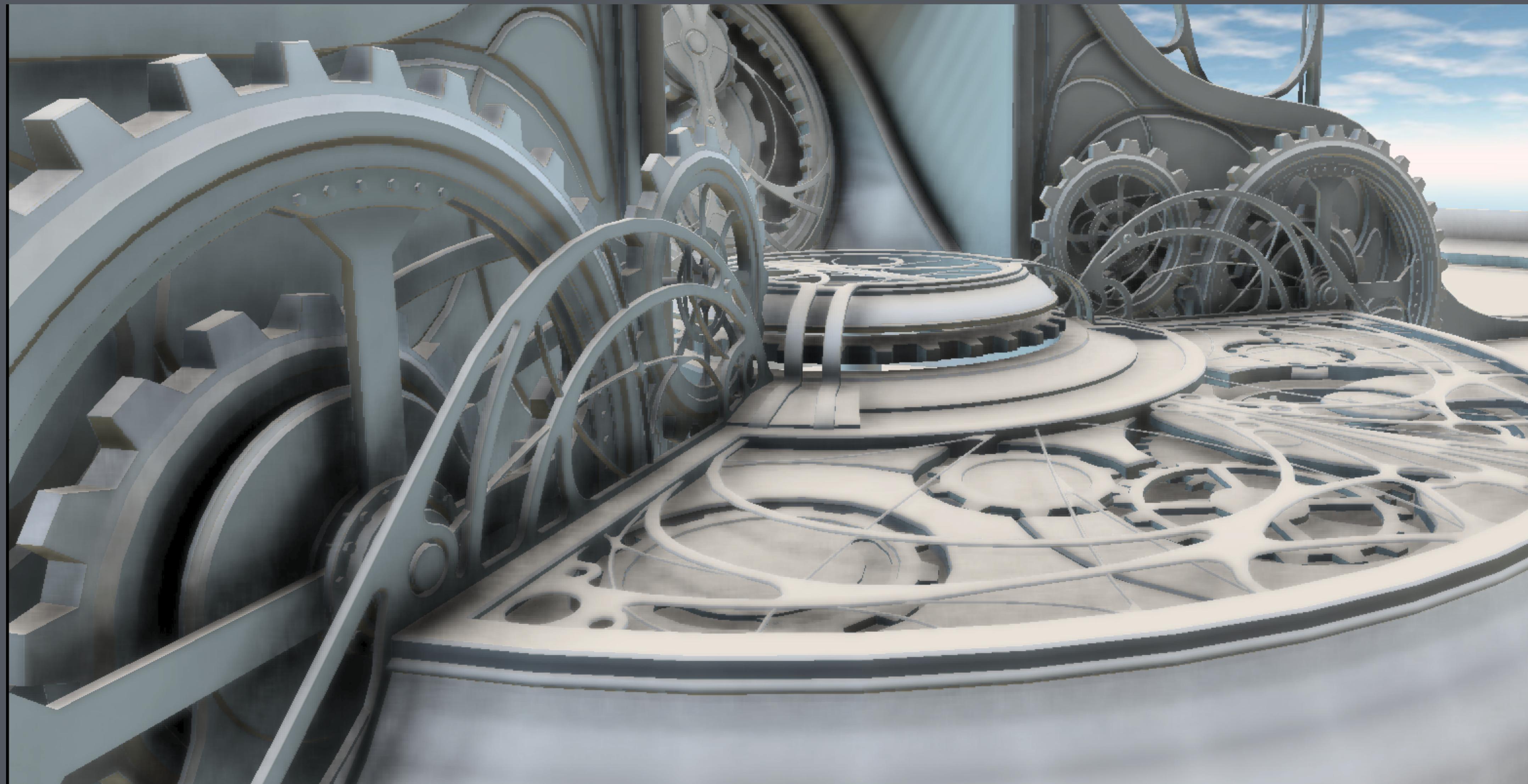
Hemisphere vs. Sphere



Irradiance map + SSAO



Irradiance map + SSAO



Irradiance map + SSAO



McGuire et al. HPG '11 [10.1145/2018323.2018327](https://doi.org/10.1145/2018323.2018327)

Irradiance map + SSAO



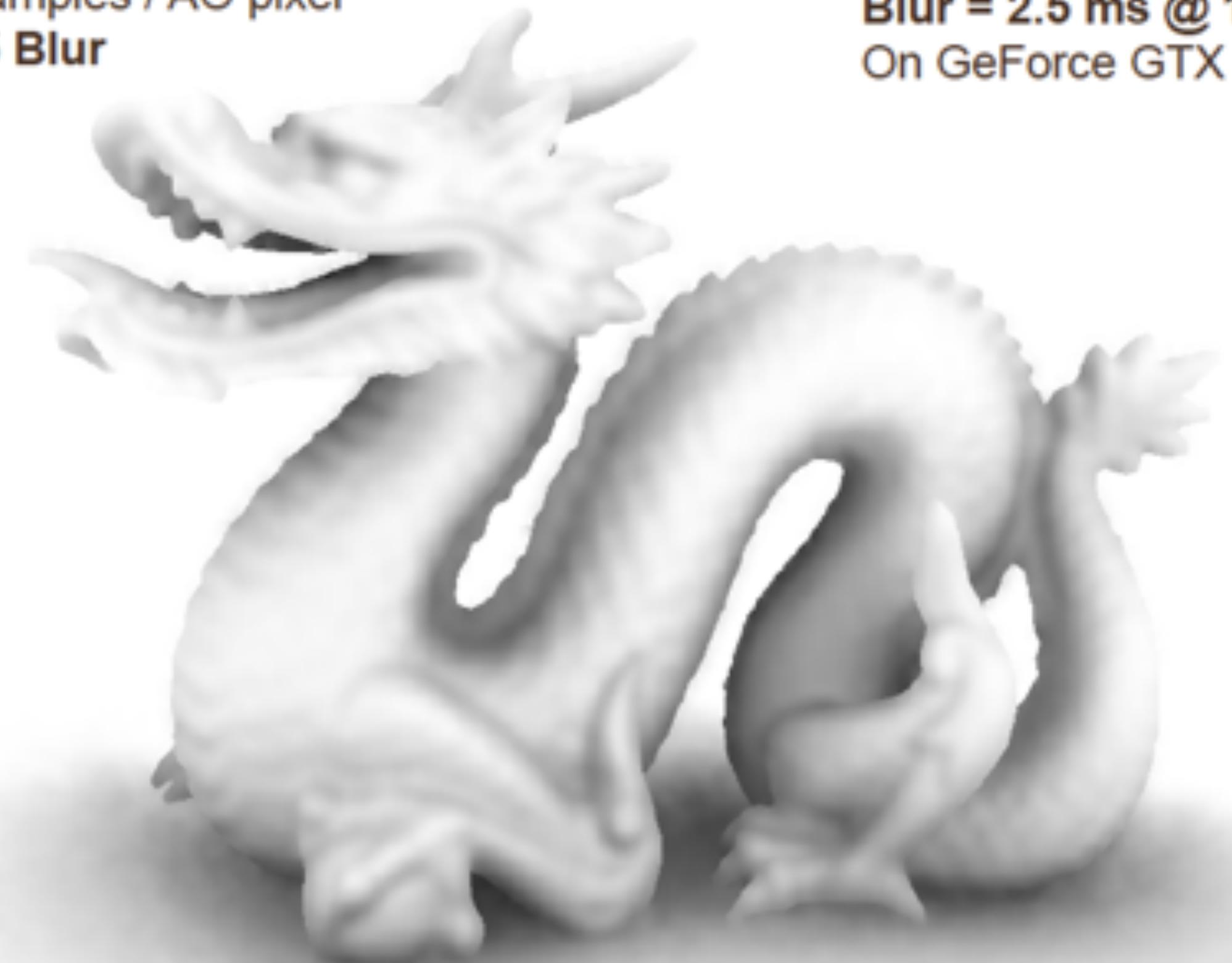
McGuire et al. HPG '11 [10.1145/2018323.2018327](https://doi.org/10.1145/2018323.2018327)

Half-Resolution AO
6x6 samples / AO pixel
No Blur

AO = 3.5 ms @ 800x600
On GeForce GTX 280



Half-Resolution AO
6x6 samples / AO pixel
15x15 Blur

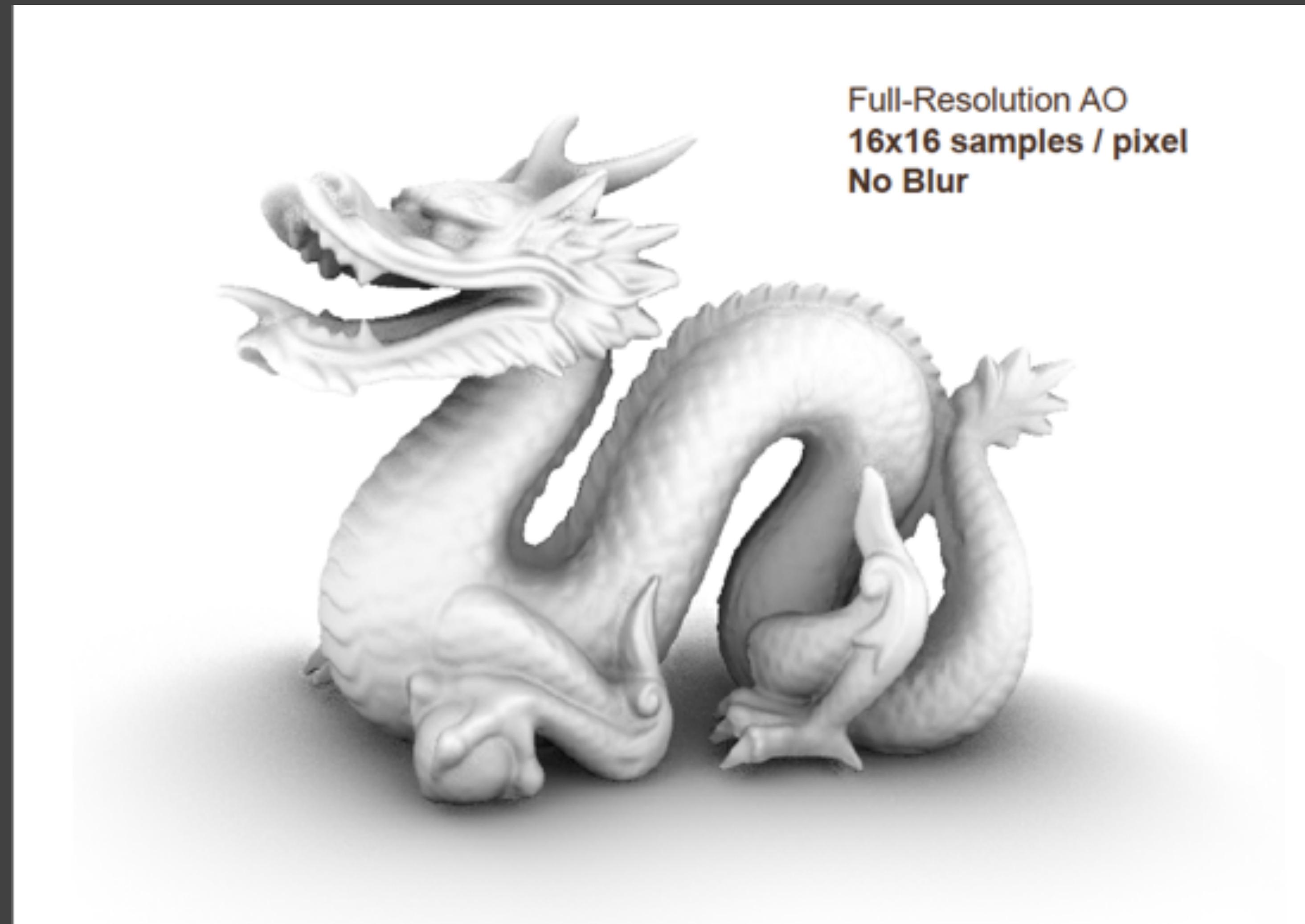


AO = 3.5 ms @ 800x600
Blur = 2.5 ms @ 1600x1200
On GeForce GTX 280

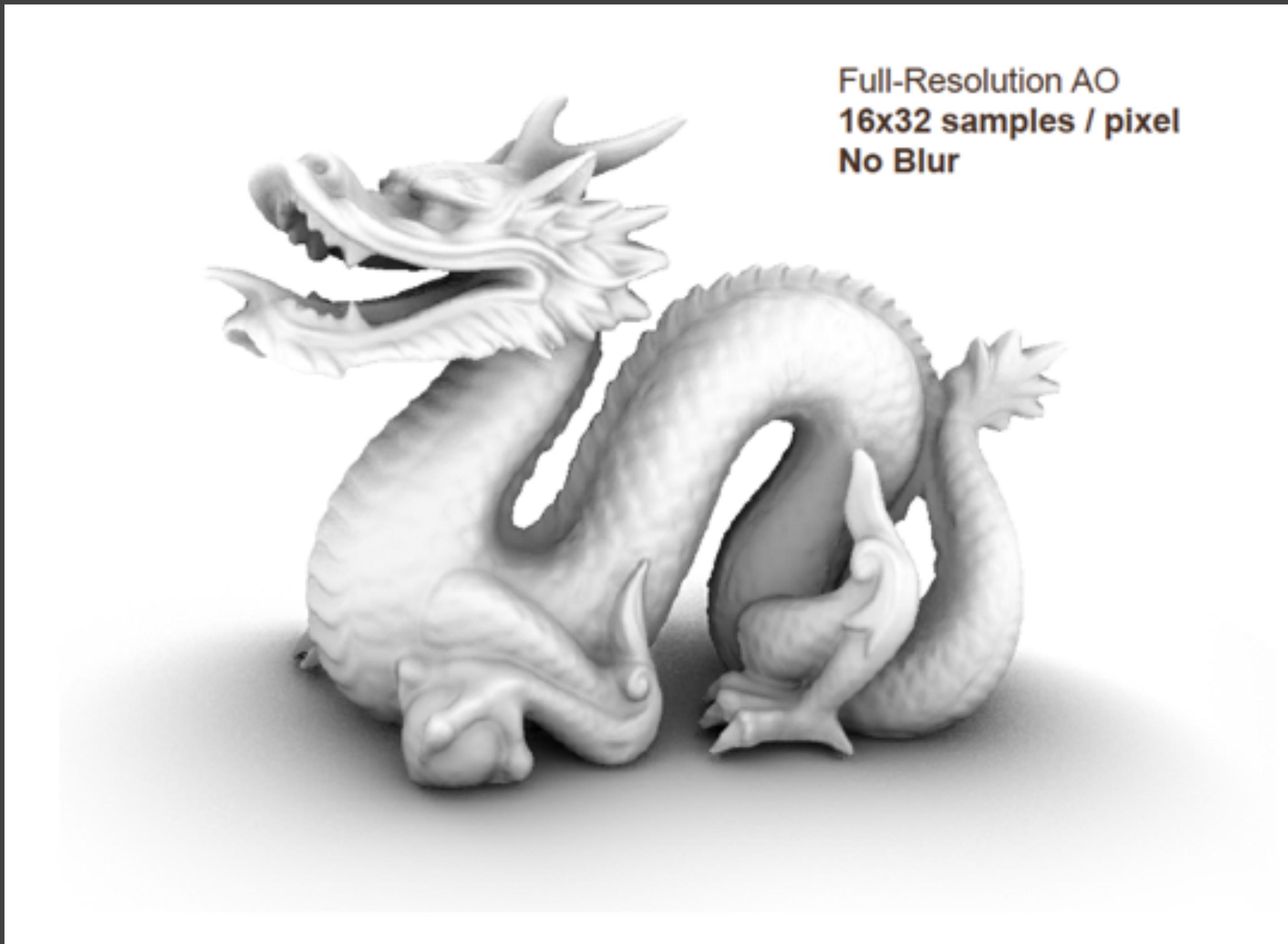
Full-Resolution AO
6x6 samples / AO pixel
15x15 Blur



AO = 30 ms @ 800x600
Blur = 2.5 ms @ 1600x1200
On GeForce GTX 280



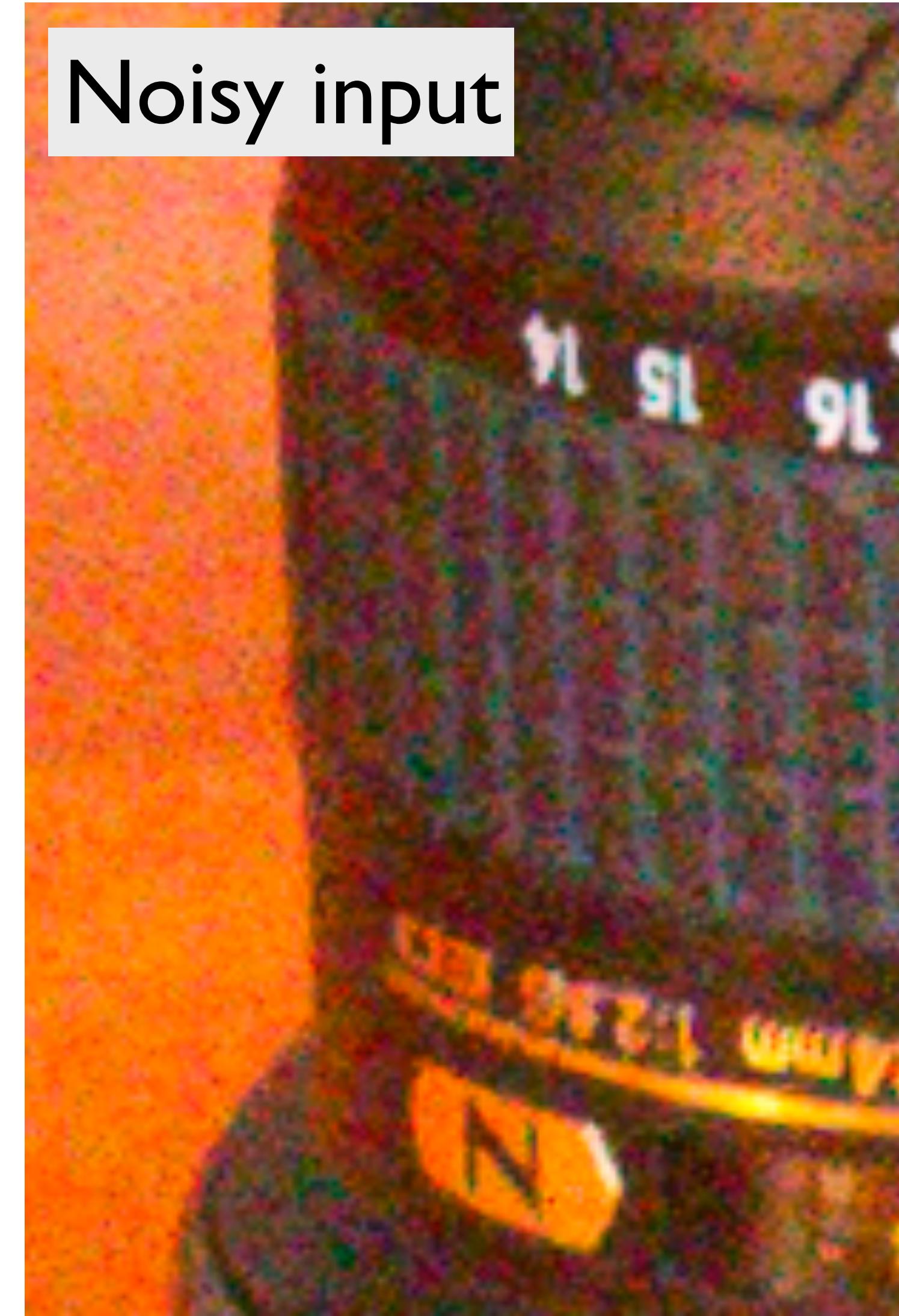
Full-Resolution AO
16x16 samples / pixel
No Blur



Full-Resolution AO
16x32 samples / pixel
No Blur

Denoising from 1 image

- We can't take average over multiple images



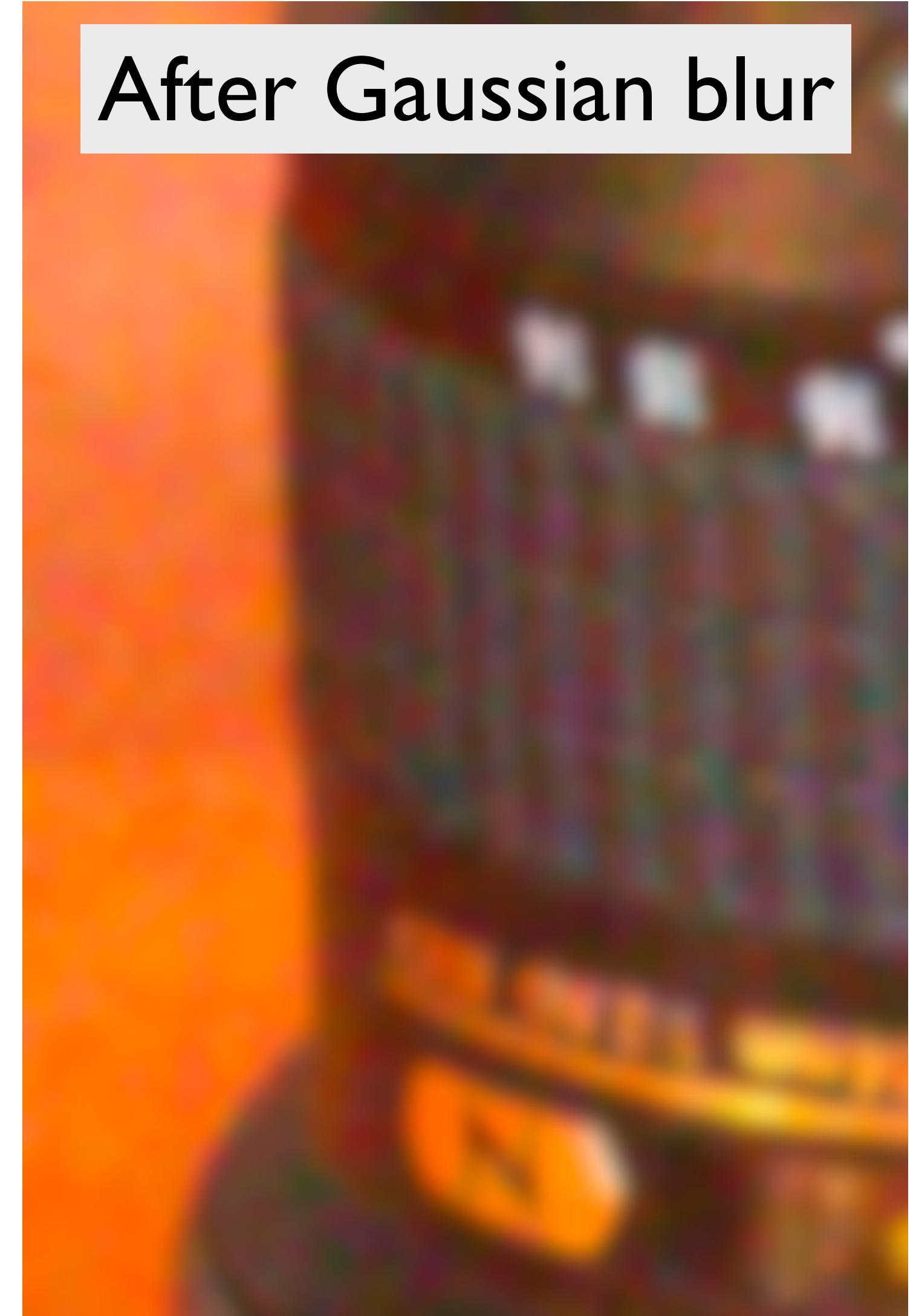
Denoising from 1 image

- We can't take average over multiple images
- Idea 1: take a spatial average
 - Most pixels have roughly the same color as their neighbor
 - Noise looks high frequency => do a low pass
- Here: Gaussian blur



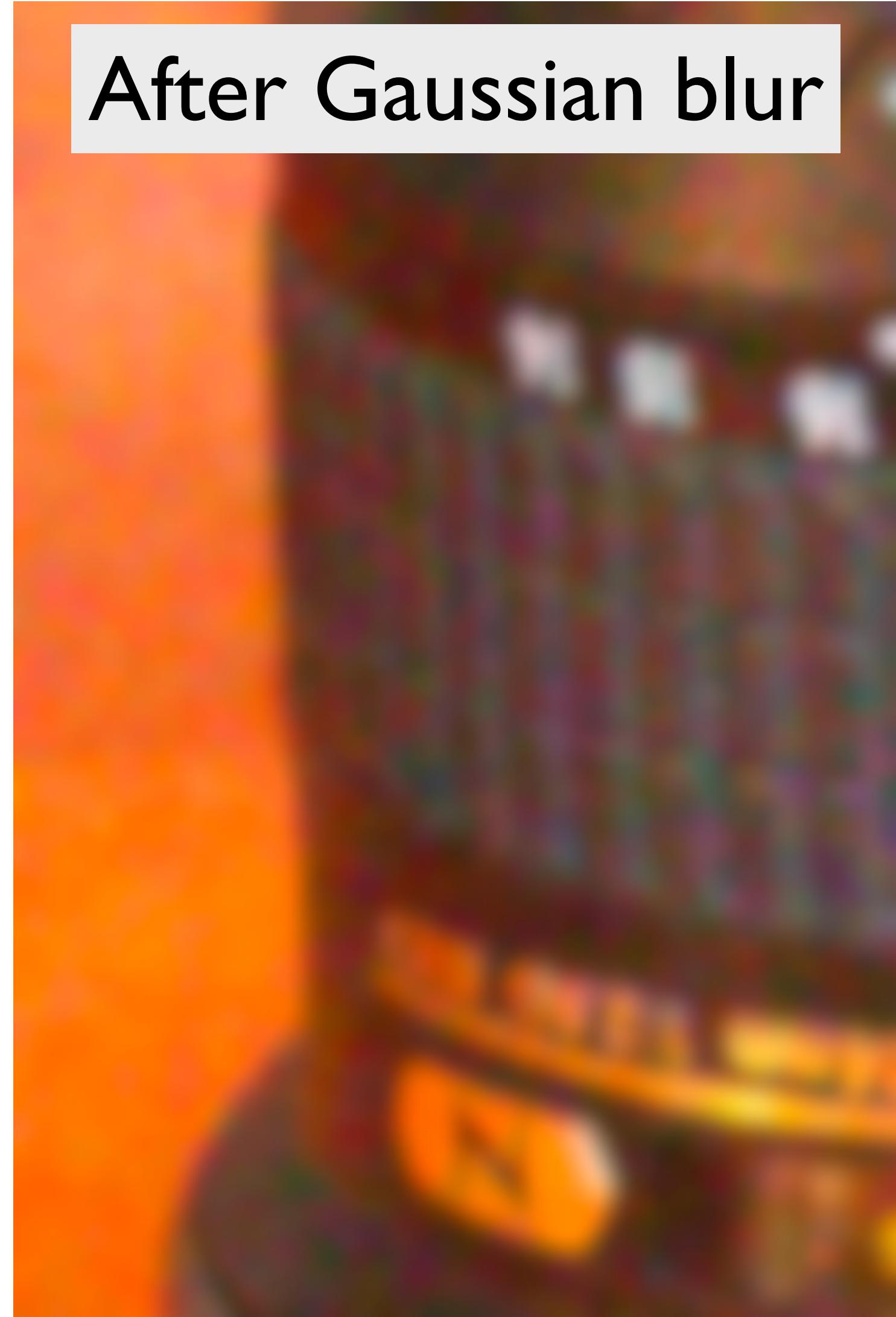
Gaussian blur

- Noise is mostly gone
- But image is blurry
 - duh!



Gaussian blur

- Noise is mostly gone
- But image is blurry
 - duh!
- Question: how to blur/smooth/abstract image, but without destroying important features?

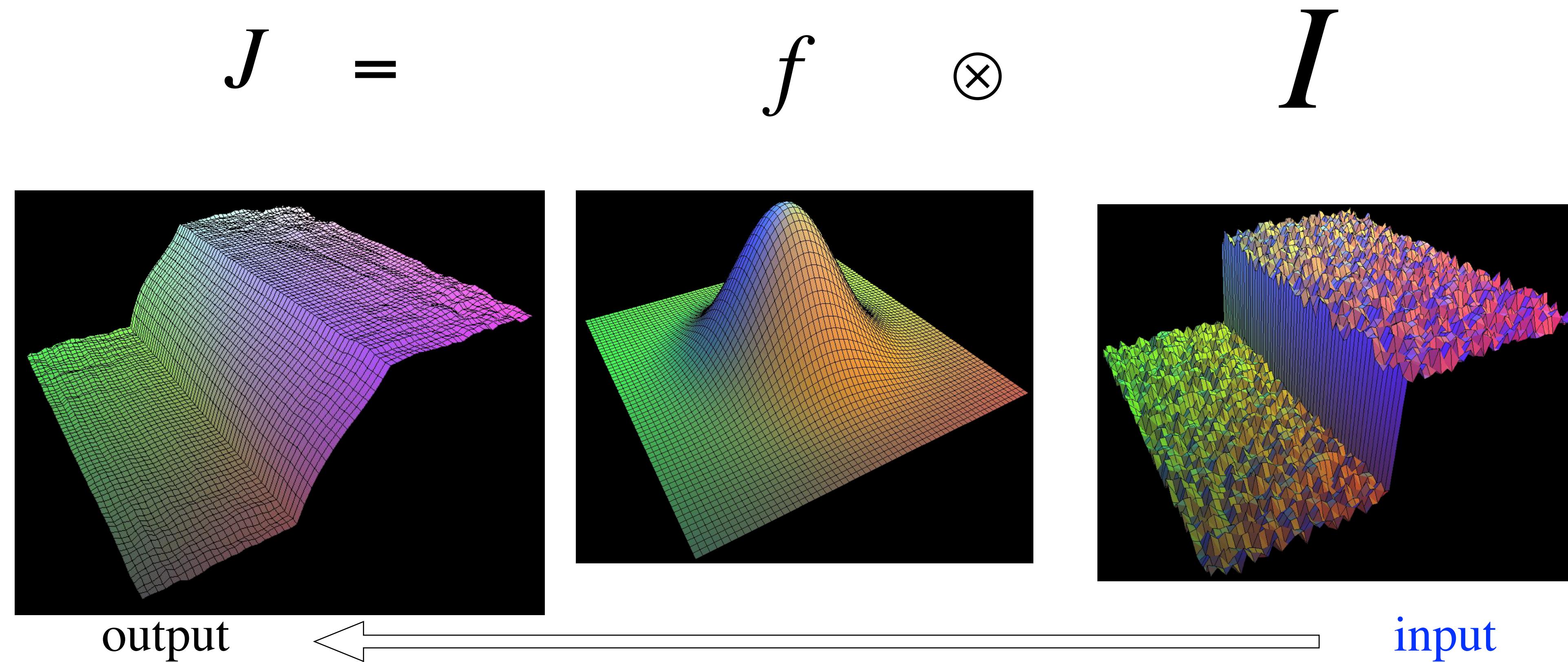


Bilateral filter

- [Tomasi and Manduchi 1998]
 - <http://www.cse.ucsc.edu/~manduchi/Papers/ICCV98.pdf>
- Developed for denoising
- Related to
 - SUSAN filter [Smith and Brady 95]
<http://citeseer.ist.psu.edu/smith95susan.html>
 - Digital-TV [Chan, Osher and Chen 2001]
<http://citeseer.ist.psu.edu/chan01digital.html>
 - sigma filter <http://www.geogr.ku.dk/CHIPS/Manual/f187.htm>
- Full survey: http://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf

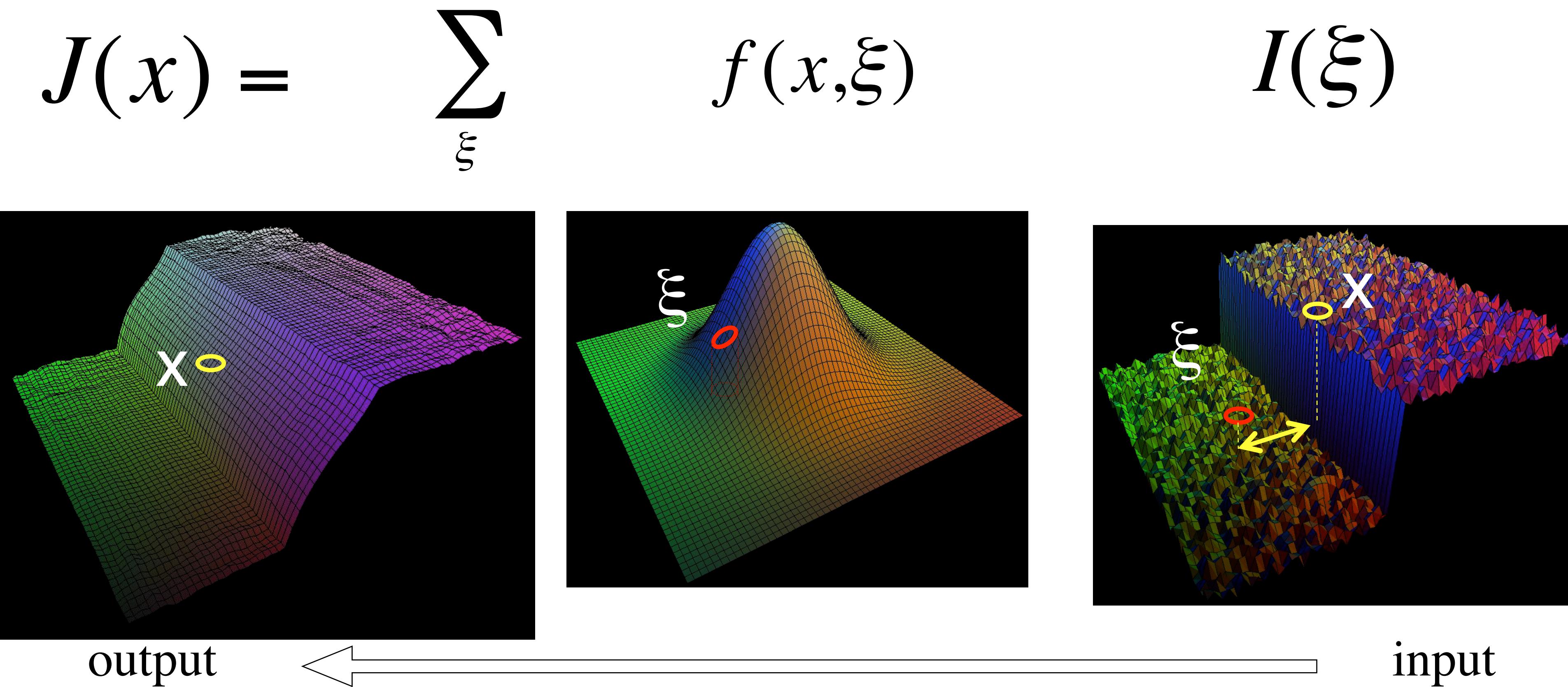
Start with Gaussian filtering

- Here, input is a step function + noise



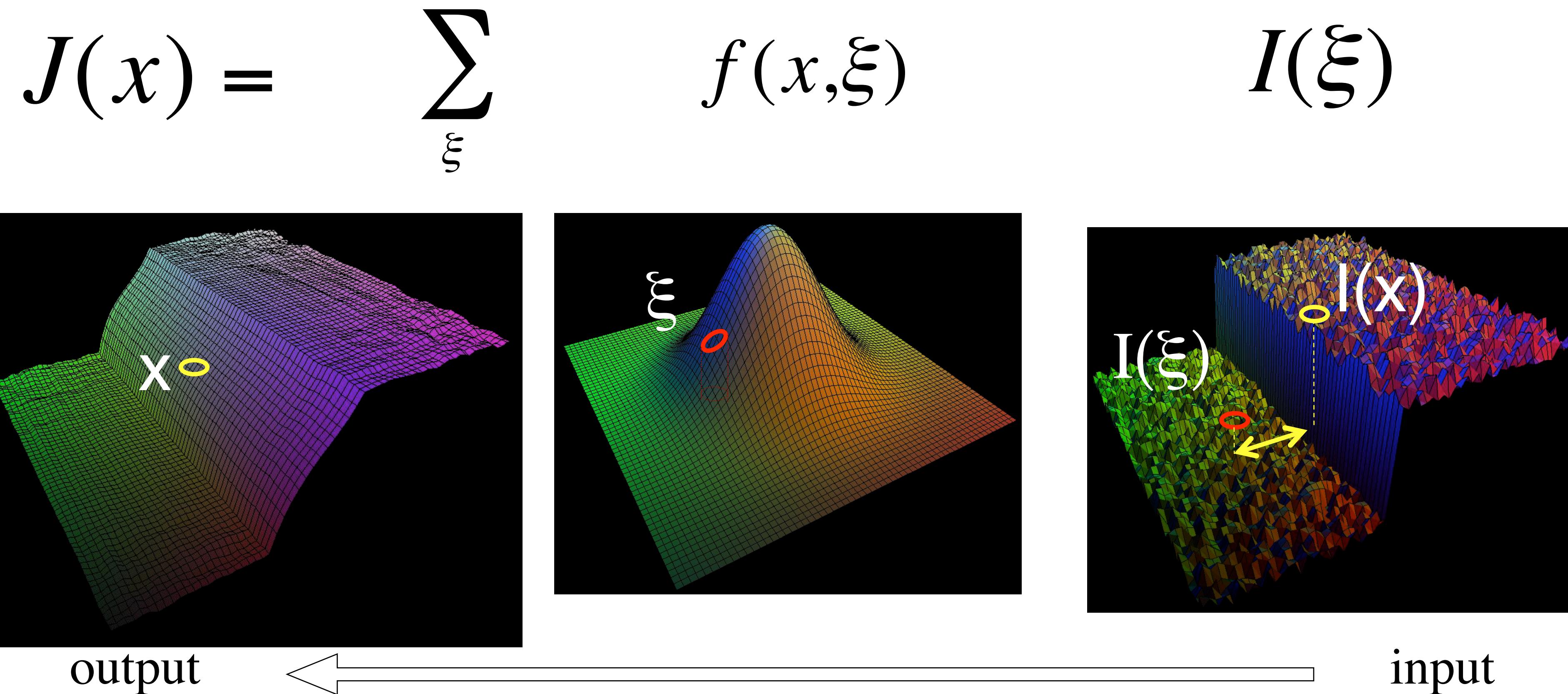
Gaussian filter as weighted average

- Weight of ξ depends on distance to x



The problem of edges

- Here, $I(\xi)$ “pollutes” our estimate $J(x)$
- It is too different

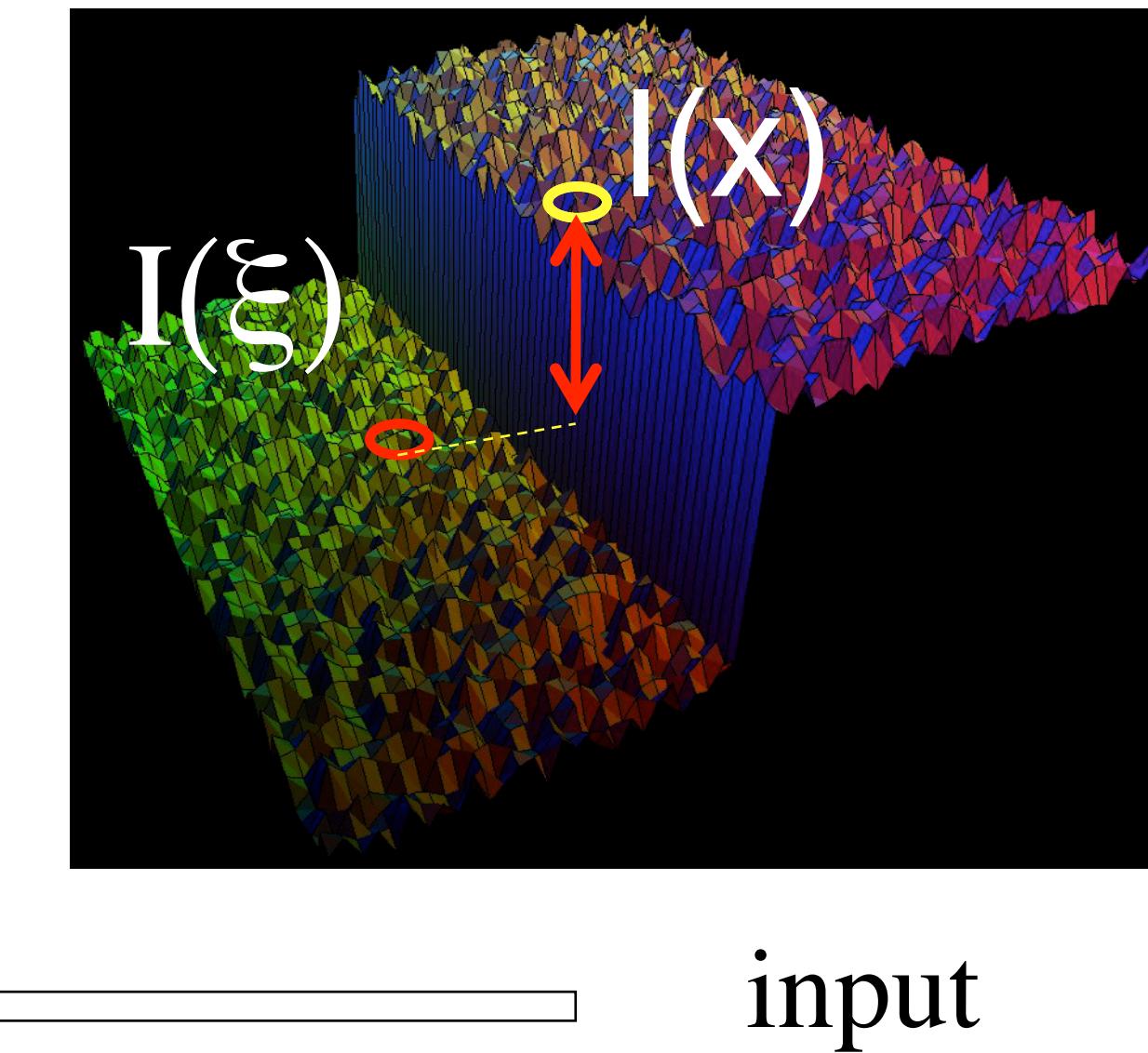
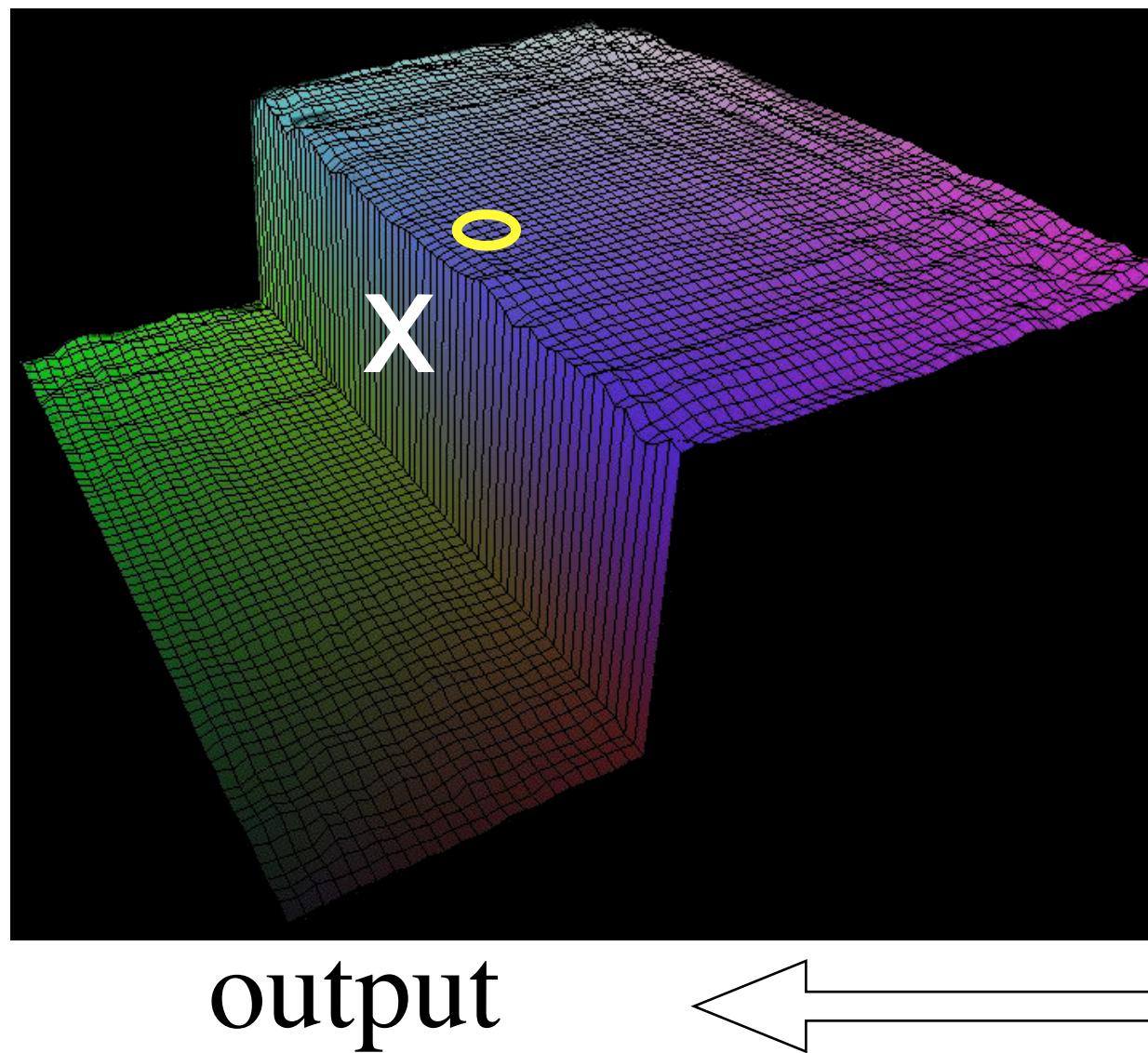


Principle of Bilateral filtering

[Tomasi and Manduchi 1998]

- Penalty **g** on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi) \quad g(I(\xi) - I(x)) \quad I(\xi)$$

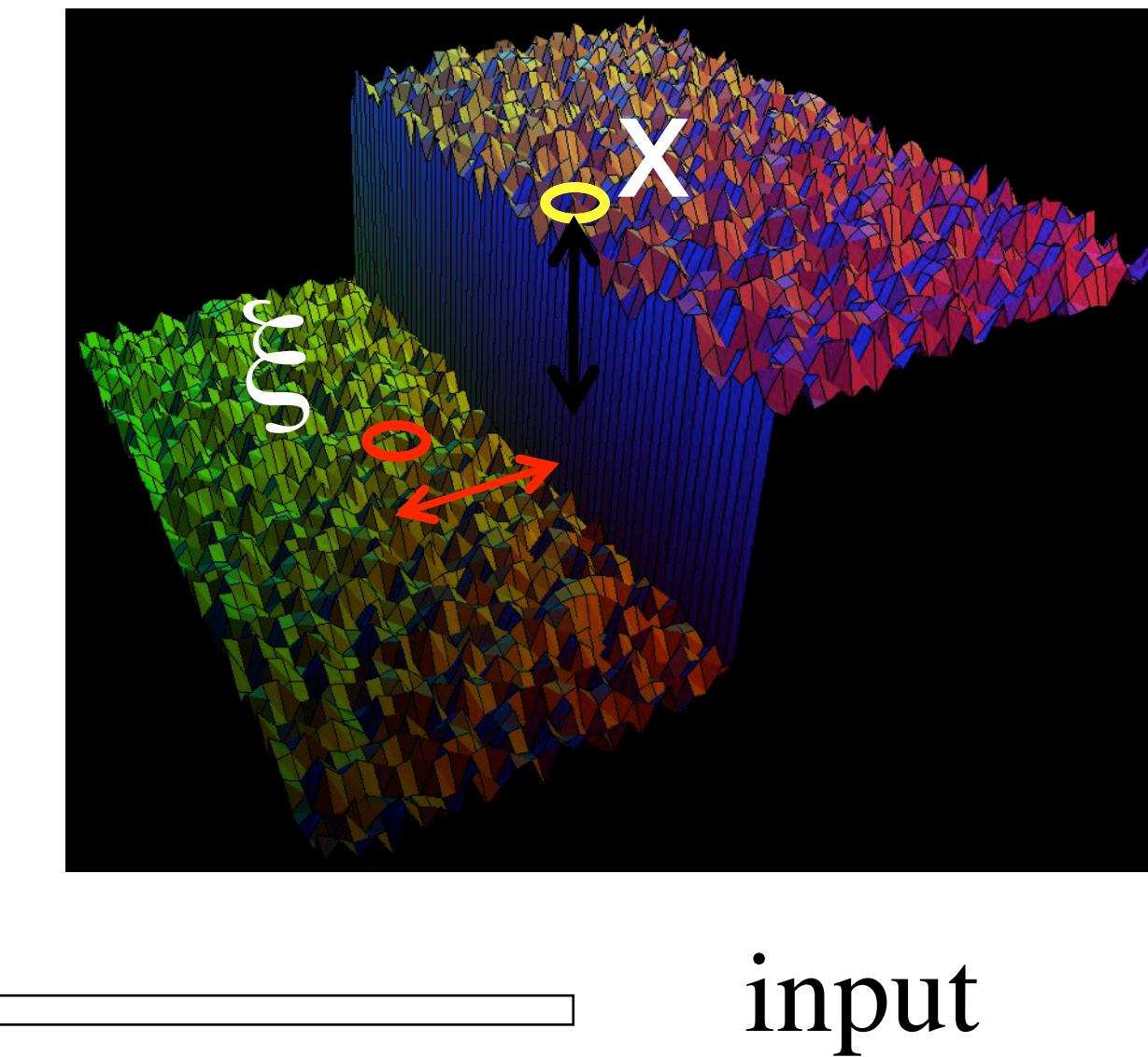
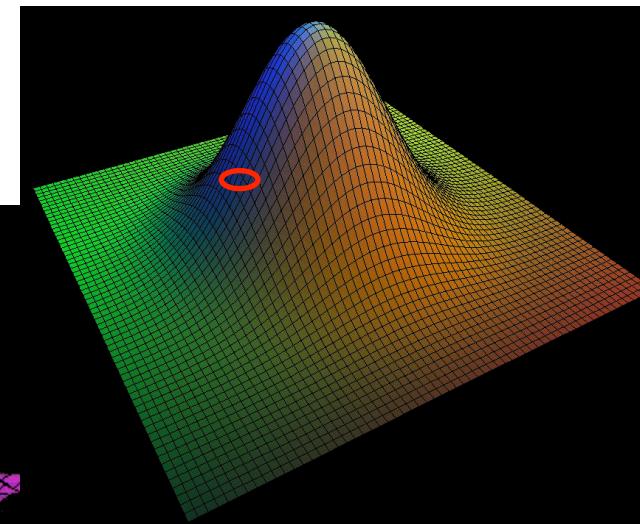
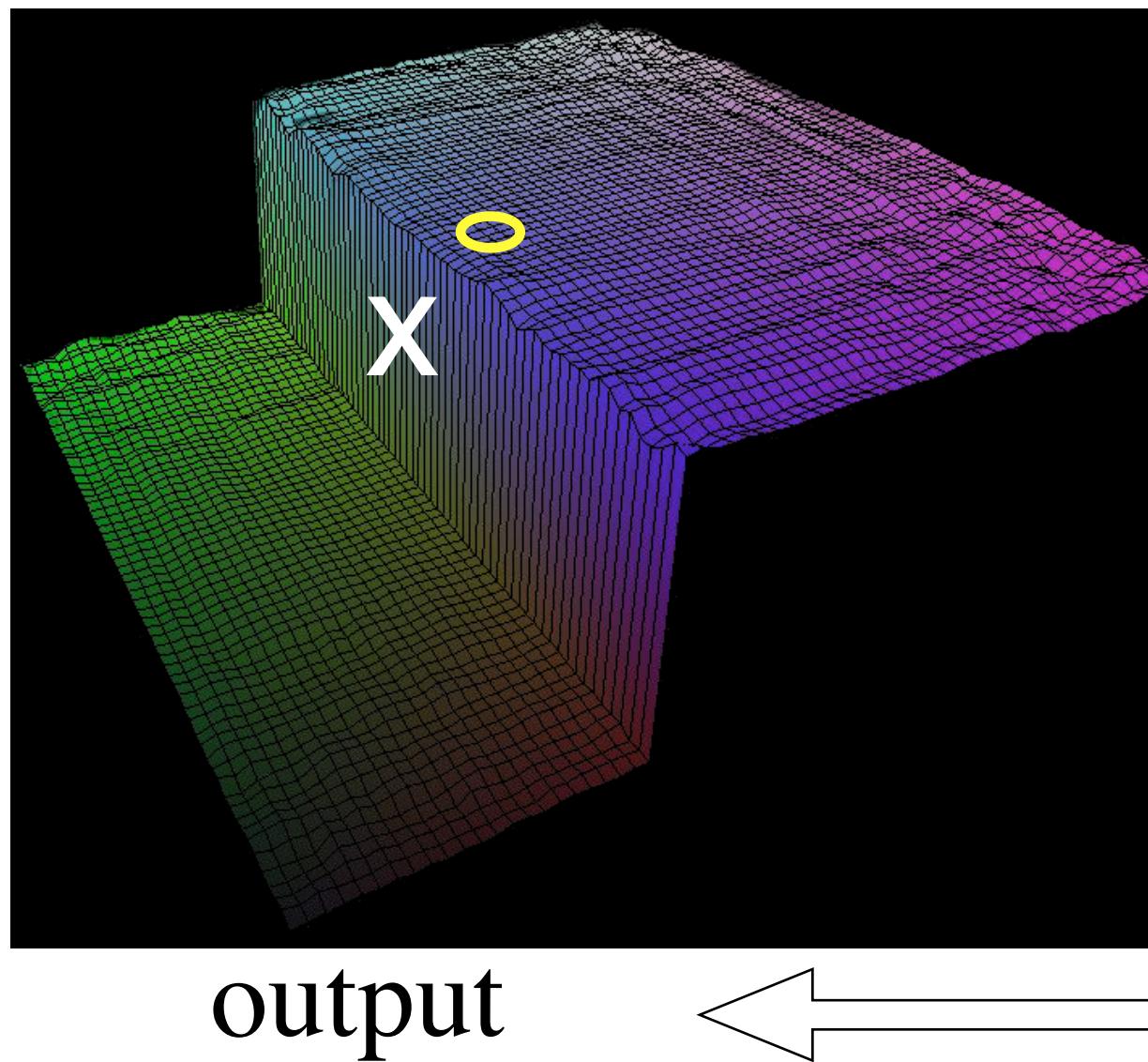


Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian f

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi) g(I(\xi) - I(x)) \quad I(\xi)$$

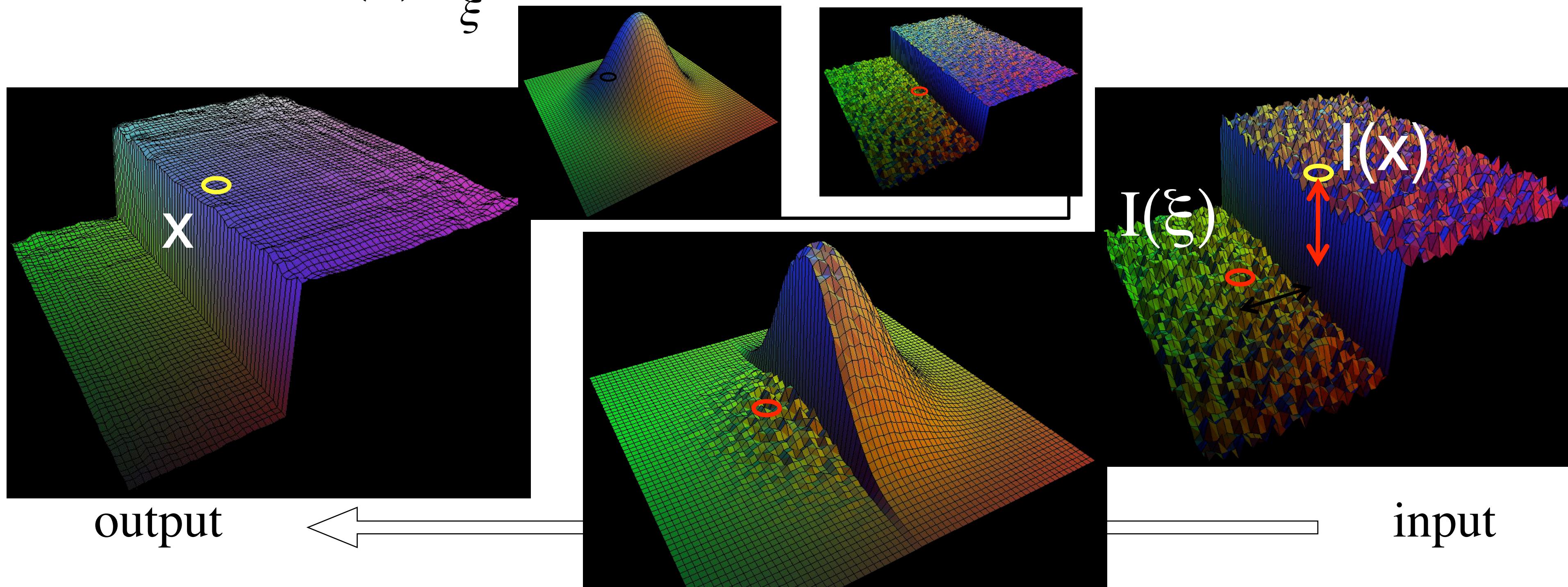


Bilateral filtering

[Tomasi and Manduchi 1998]

- Spatial Gaussian f
- Gaussian g on the intensity difference

$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi) g(I(\xi) - I(x)) I(\xi)$$

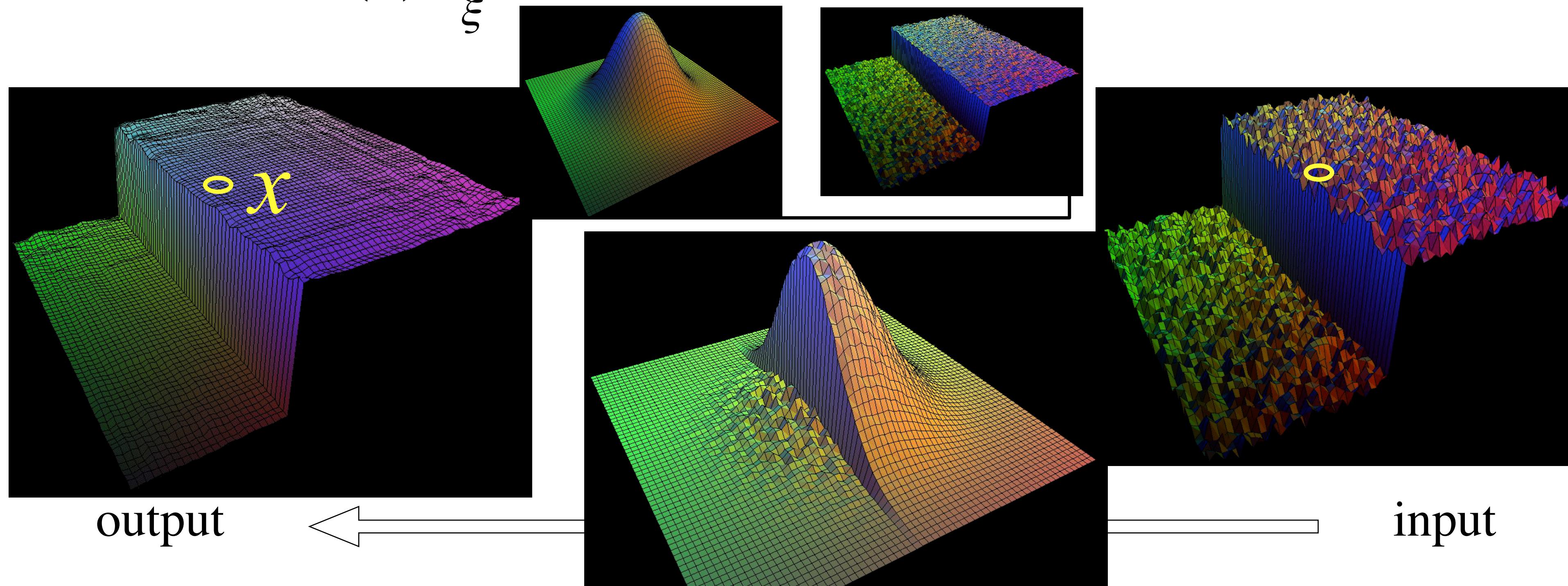


Normalization factor

[Tomasi and Manduchi 1998]

- $k(x) = \sum_{\xi} f(x, \xi) g(I(\xi) - I(x))$

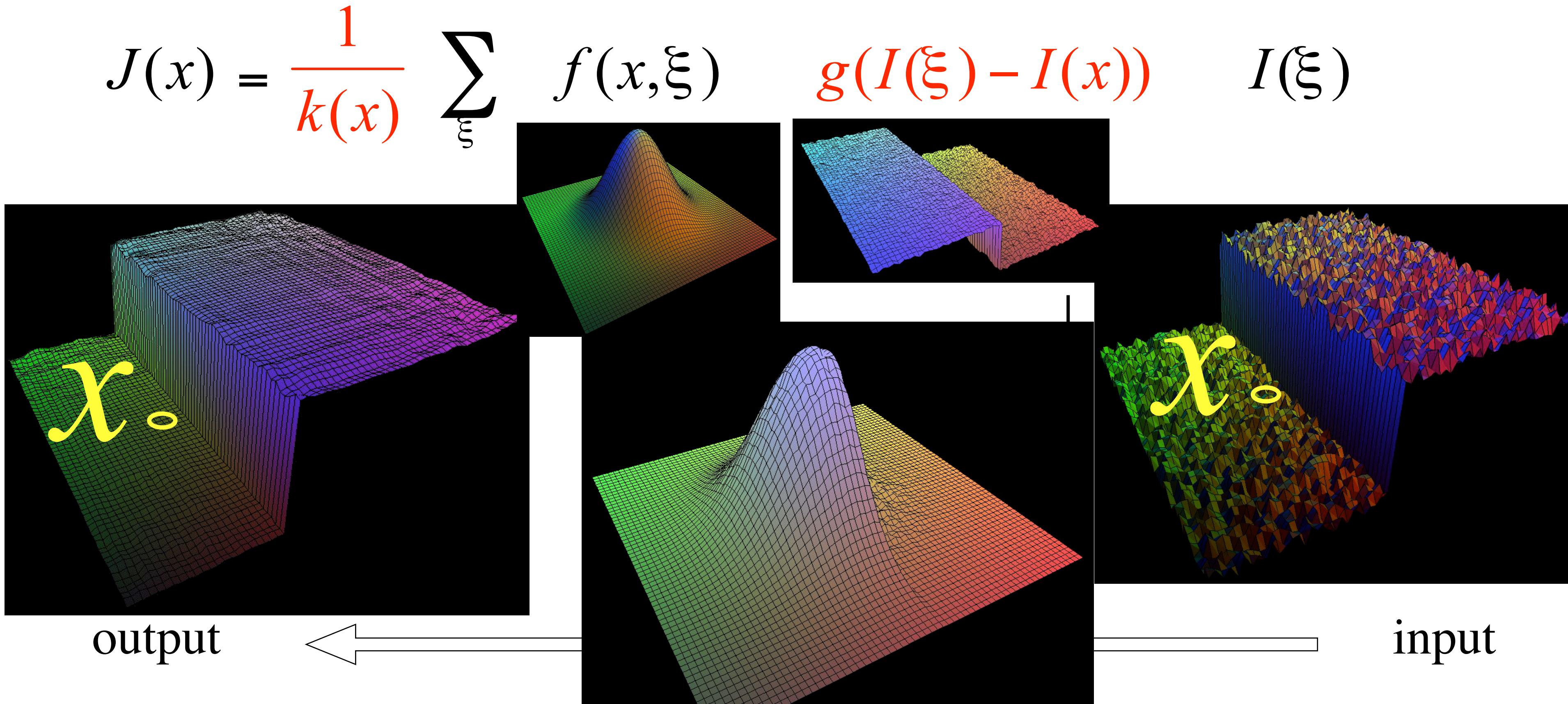
$$J(x) = \frac{1}{k(x)} \sum_{\xi} f(x, \xi) g(I(\xi) - I(x)) I(\xi)$$



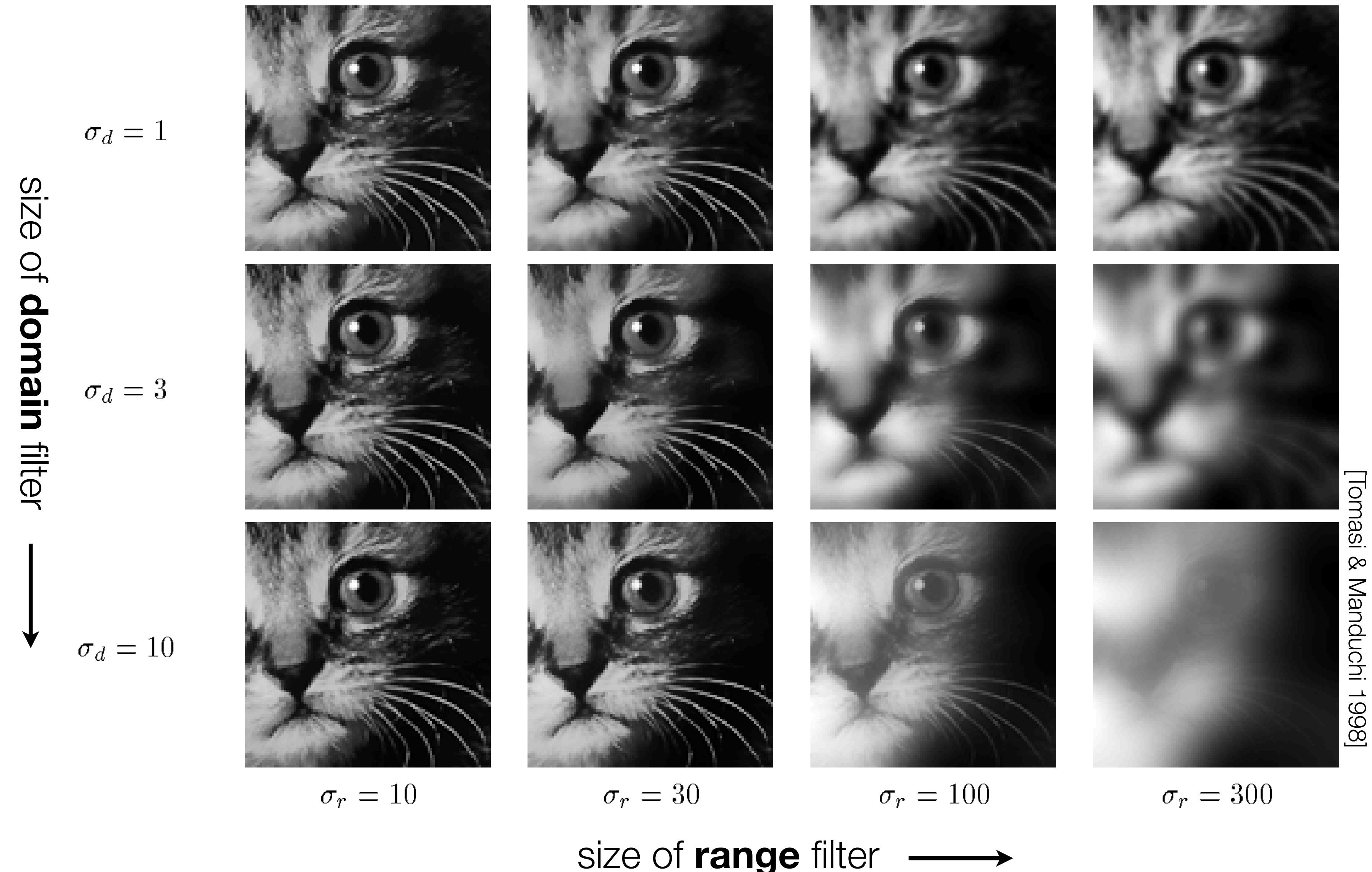
Bilateral filtering is non-linear

[Tomasi and Manduchi 1998]

- The weights are different for each output pixel



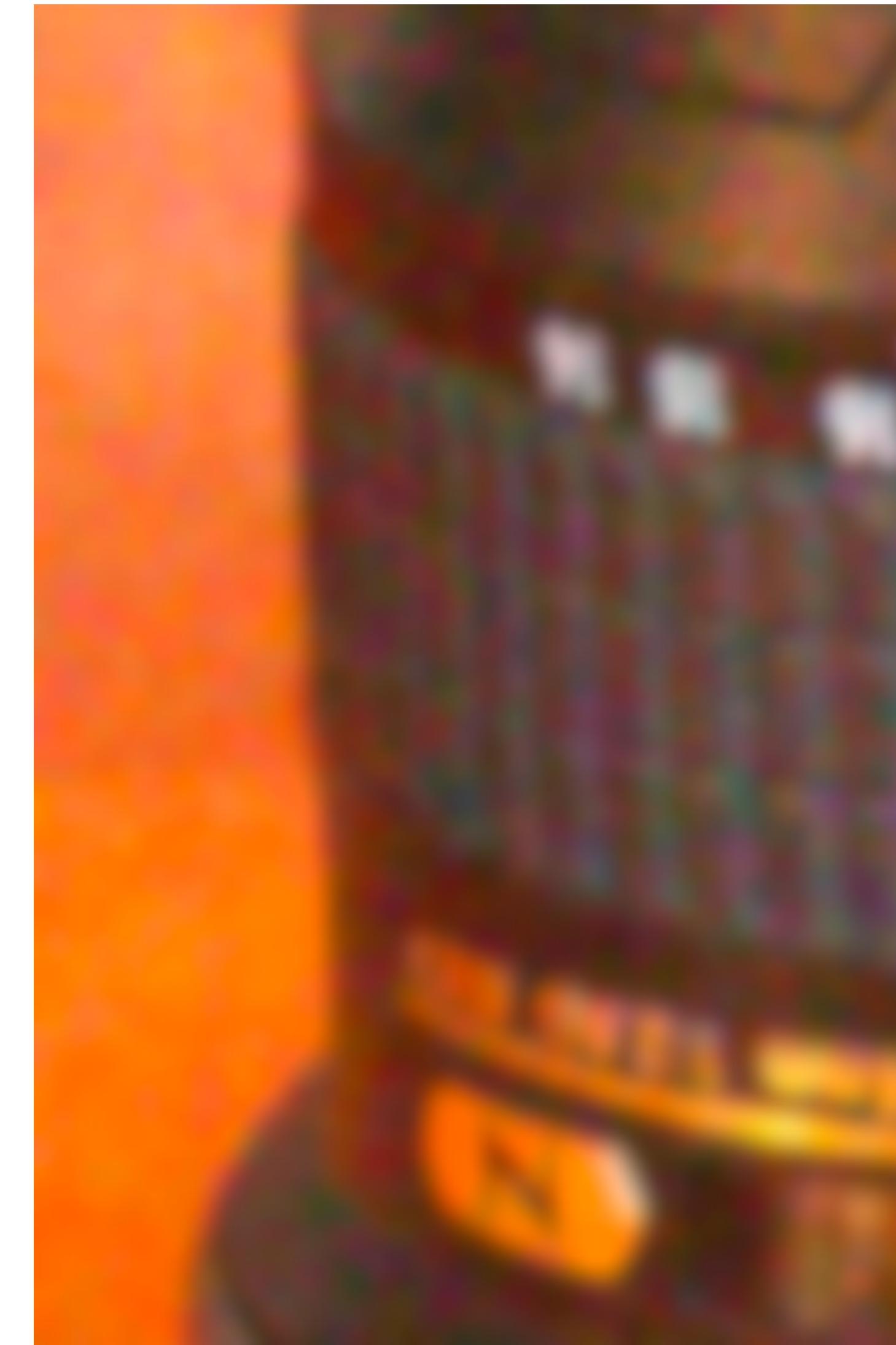
Effects of bilateral filter



Bilateral filter



Noisy input



After gaussian blur

Bilateral filter



Noisy input



After bilateral filter