07 Environment and ambient illumination

Steve Marschner

CS5625 Spring 2022

Real-time environment illumination

Now incident radiance field is stored in a texture, not defined by a polygon

- this makes it easy to do mirror reflections: single cubemap lookup

Two special kinds of BRDFs are convenient

- diffuse BRDFs: irradiance depends only on surface normal (not **v**)
  …leads to **irradiance environment maps**
- rotationally symmetric lobes (e.g. Phong): it's a convolution of the environment map
  …leads to **prefiltered environment maps**

Standard reflection map

For mirror-specular surface, the illumination integral reduces to

$$L_r(\mathbf{v}) = L_i(\mathbf{r}(\mathbf{v}, \mathbf{n}))$$

This is a function only of r

- so store it in a cubemap and look up using **r**(**v**, **n**).

Irradiance environment map

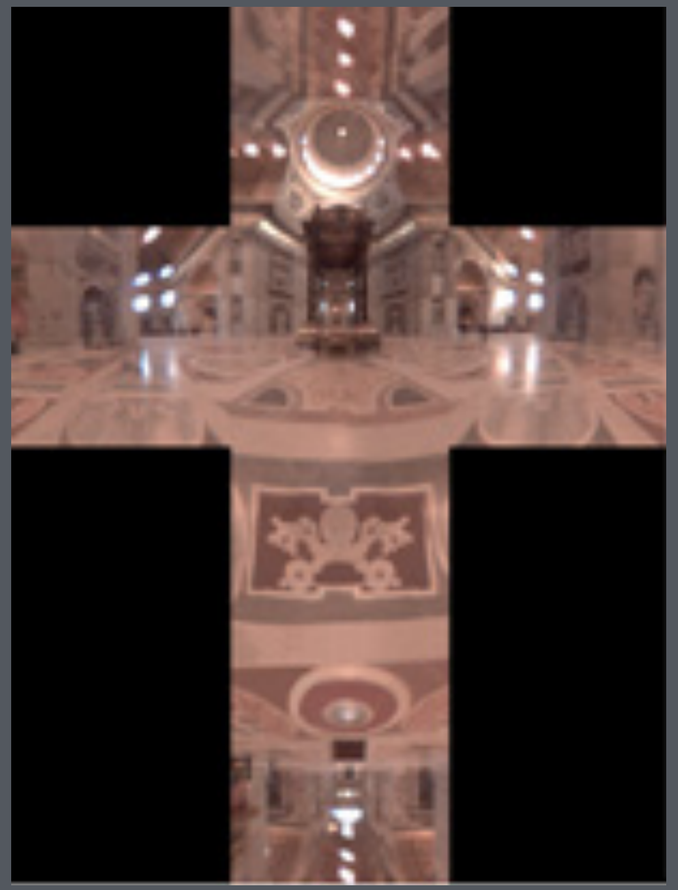
For diffuse surface, illumination integral reduces to

$$L_r = \frac{R}{\pi} \int_{\Omega} L_i(\mathbf{l}) \, (\mathbf{n} \cdot \mathbf{l}) \, d\mathbf{l}$$

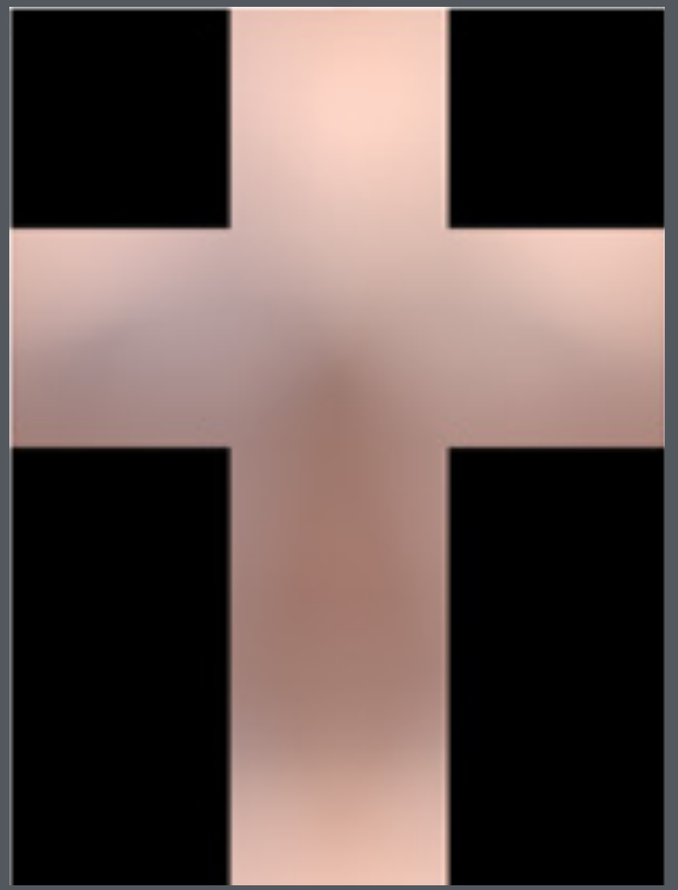
This is a function only of n

- so store it in a cubemap and look up using **n**.

Irradiance environment map



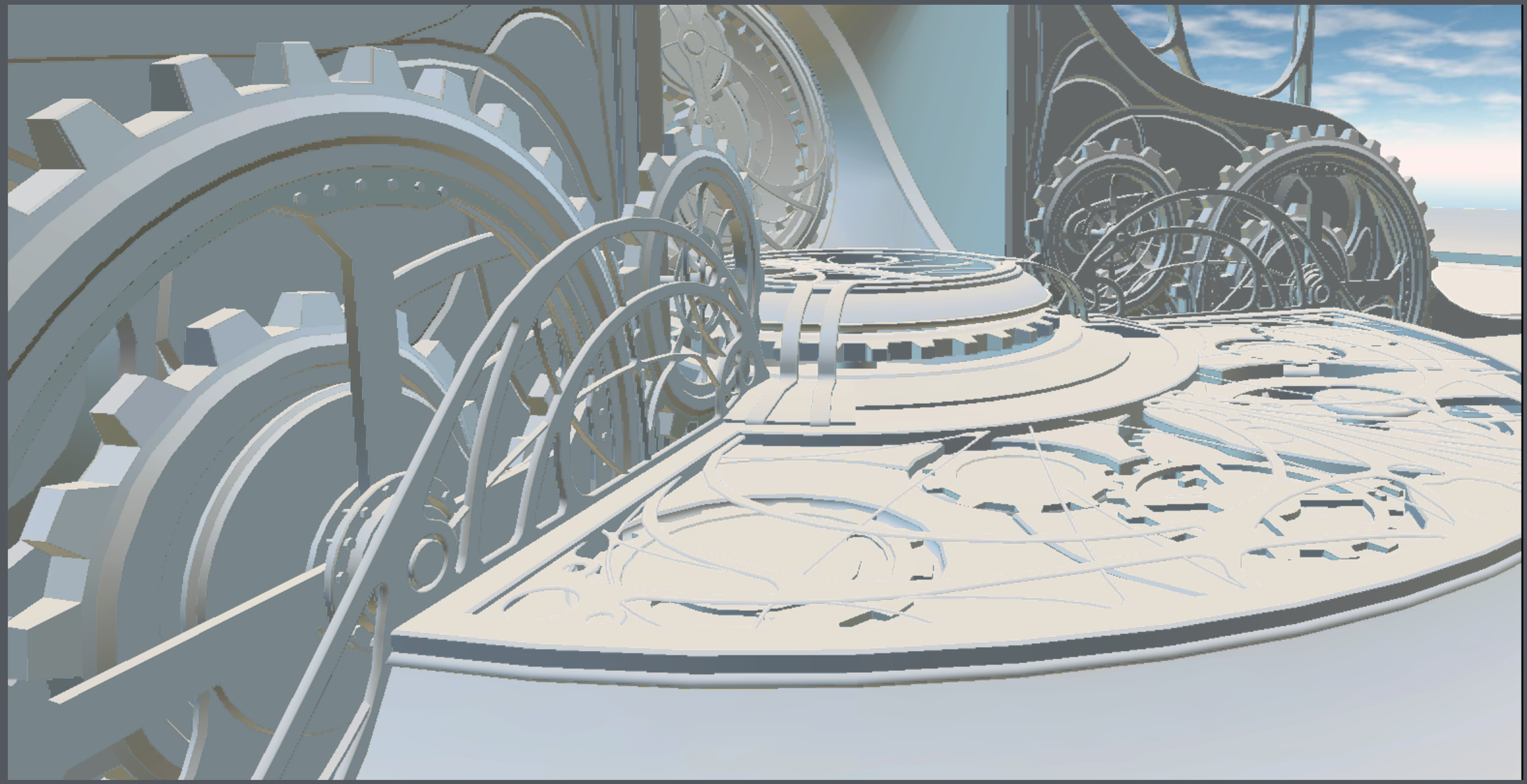
environment map



irradiance map

Gary King in GPU Gems 2 http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter10.html

Irradiance map illumination



McGuire et al. HPG ’11 10.1145/2018323.2018327

Irradiance map illumination

McGuire et al. HPG '11 10.1145/2018323.2018327

Prefiltered environment map

For a general specular surface, approximate the BRDF by a function

$$f_r(\mathbf{v}, \mathrm{l})\,(\mathbf{n} \cdot \mathrm{l}) \approx g_\sigma(\mathrm{l} \cdot \mathbf{r})$$

- where g defines a lobe of width σ that is symmetric around **r**

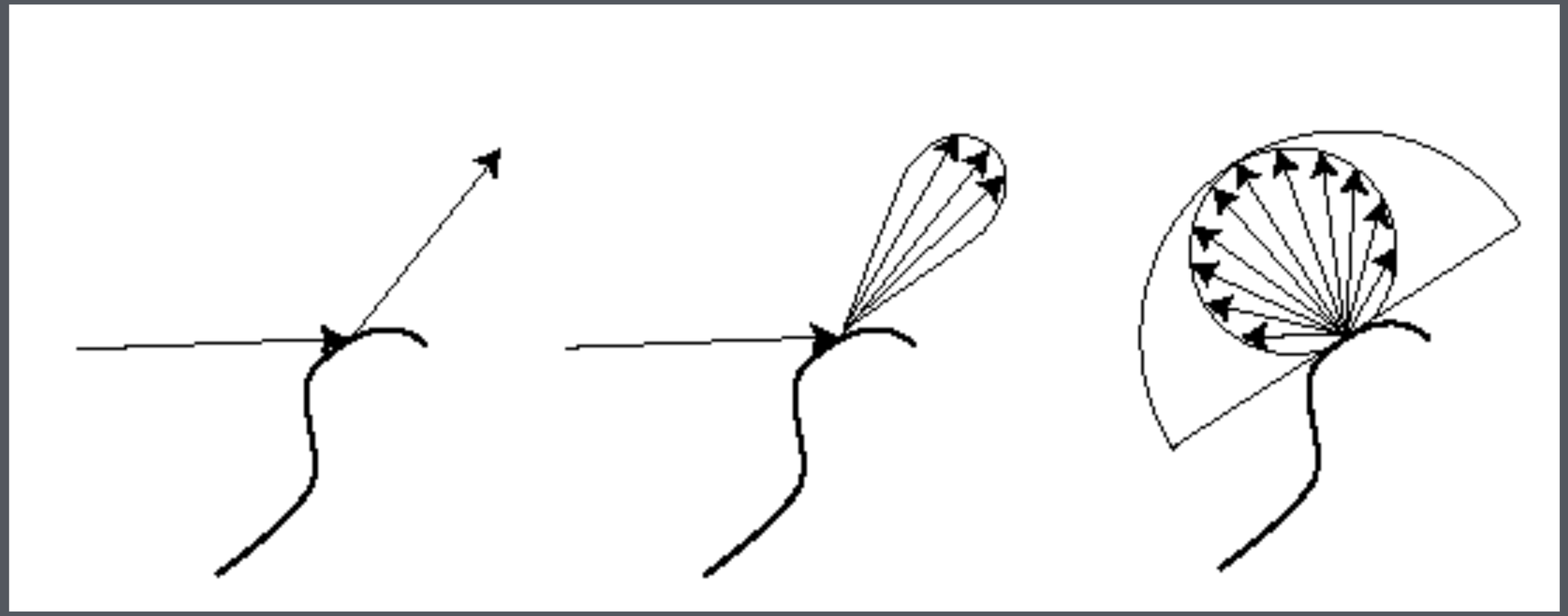
The illumination integral is

$$L_r(\mathbf{v}) \approx \int_\Omega g_\sigma(\mathrm{l} \cdot \mathbf{r})\, L_i(\mathrm{l})\, d\mathrm{l}$$

This depends only on r and the scalar σ

- so store it in an array of cubemaps indexed by σ, and look up using **r**

Irradiance environment mapping



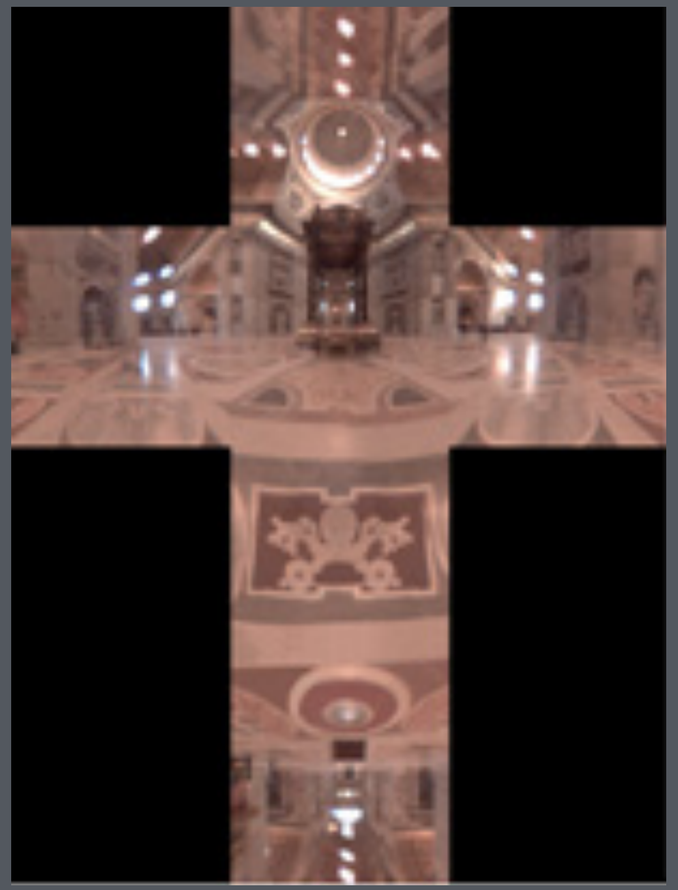
environment map
for specular surface

prefiltered map
for glossy surface

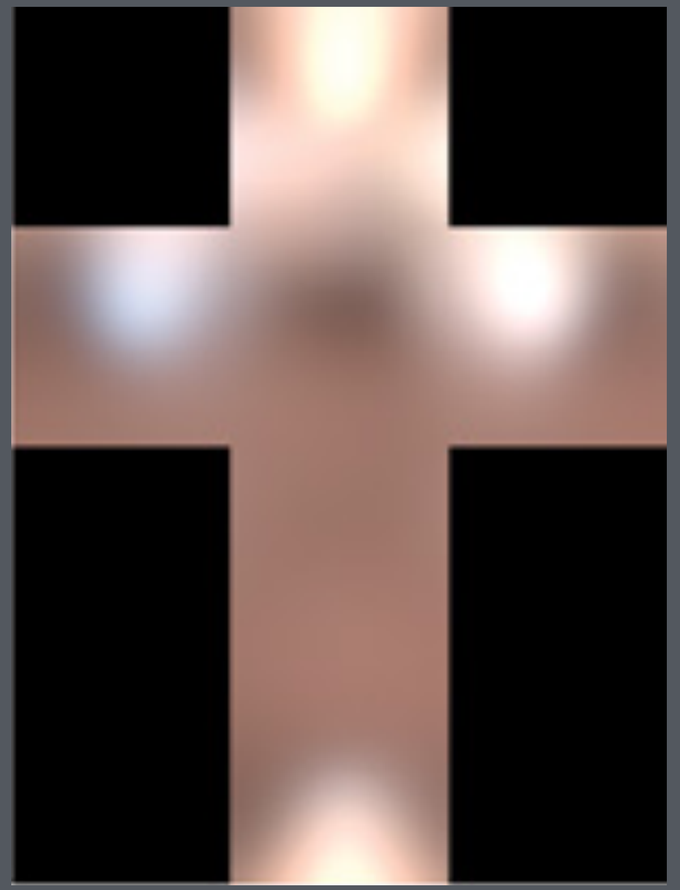
prefiltered map
for diffuse surface

Akenine-Möller et al. *RTR 3e*

Prefiltered environment map



environment map



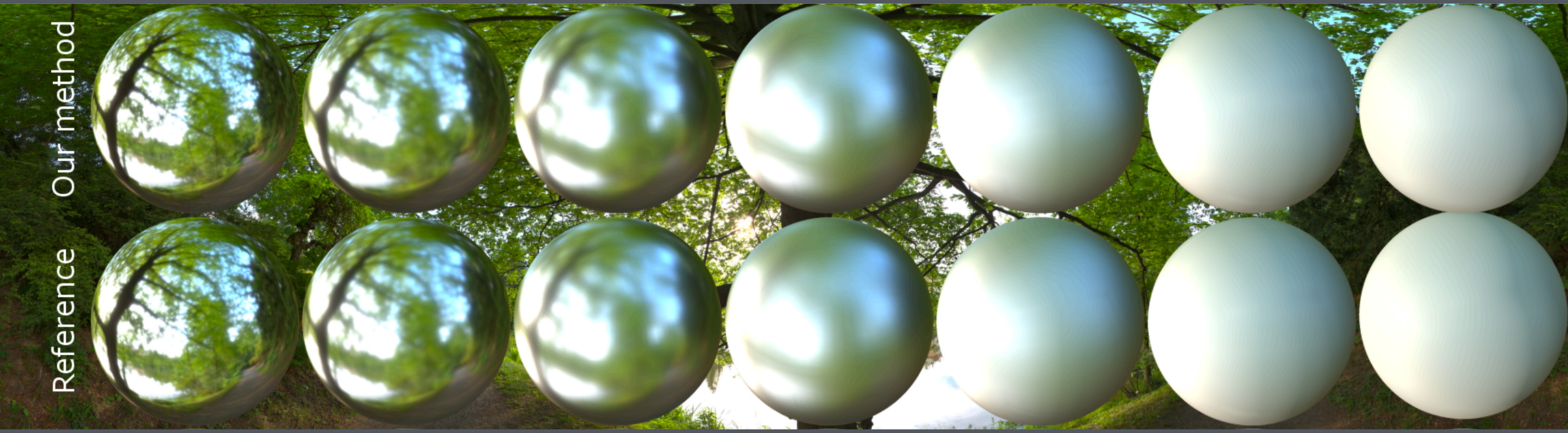
prefiltered for Phong

Gary King in GPU Gems 2 http://http.developer.nvidia.com/GPUGems2/gpugems2_chapter10.html

Prefiltered map variants

Many approaches to approximating the BRDF with symmetric lobes

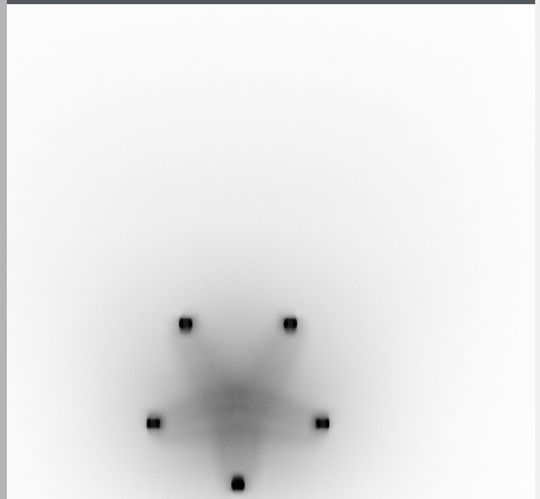
- classic is to use a single lobe, loses stretching at grazing angles
- many techniques use multiple samples, similar to anisotropic texture filtering
- modern thought process is to treat the whole precomputation as approximating more accurate BRDF lobes using multiple samples from arbitrary maps stored in mipmaps



Manson & Sloan 2016

Shadow baking

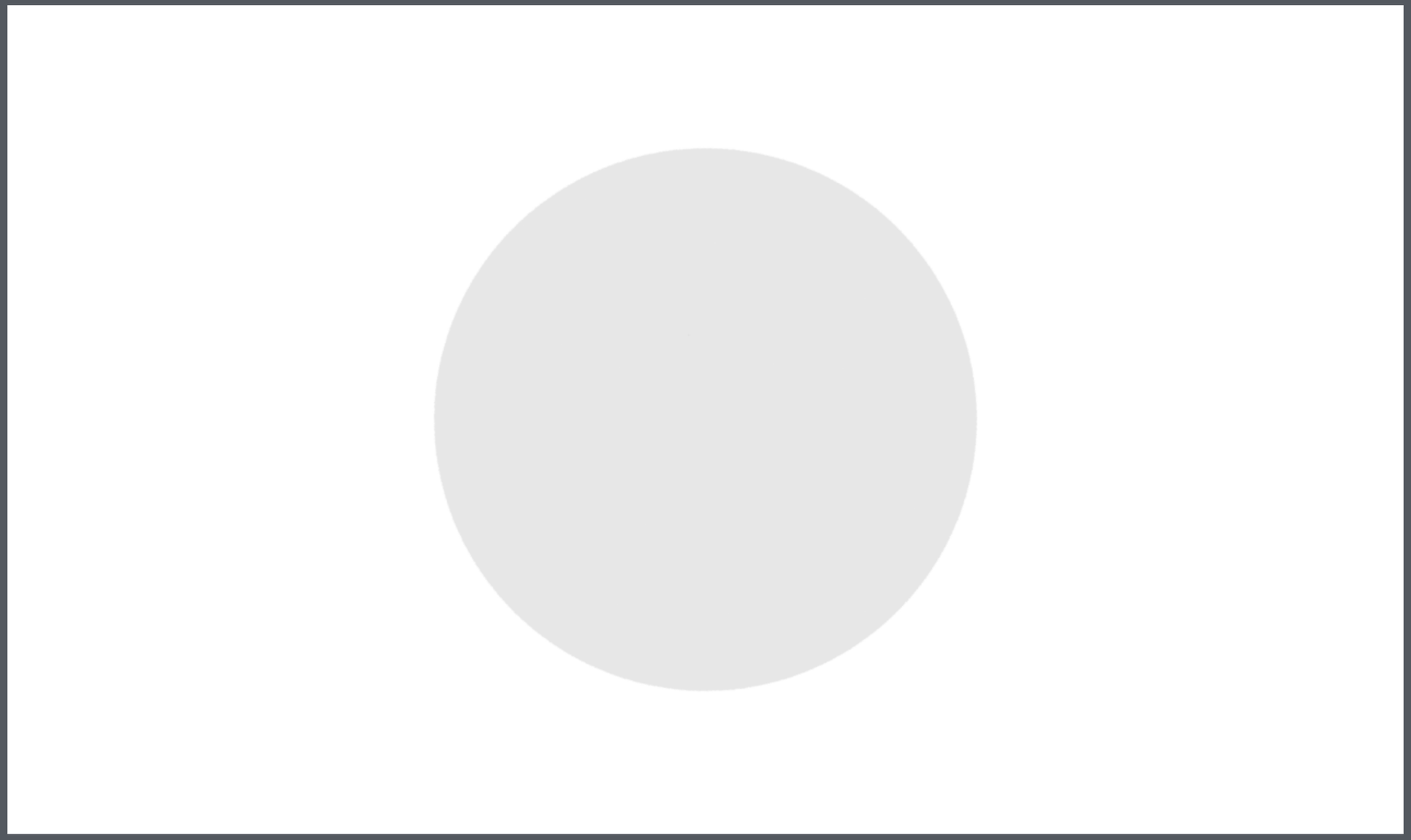
Rendering with no shadows,
darker diffuse floor



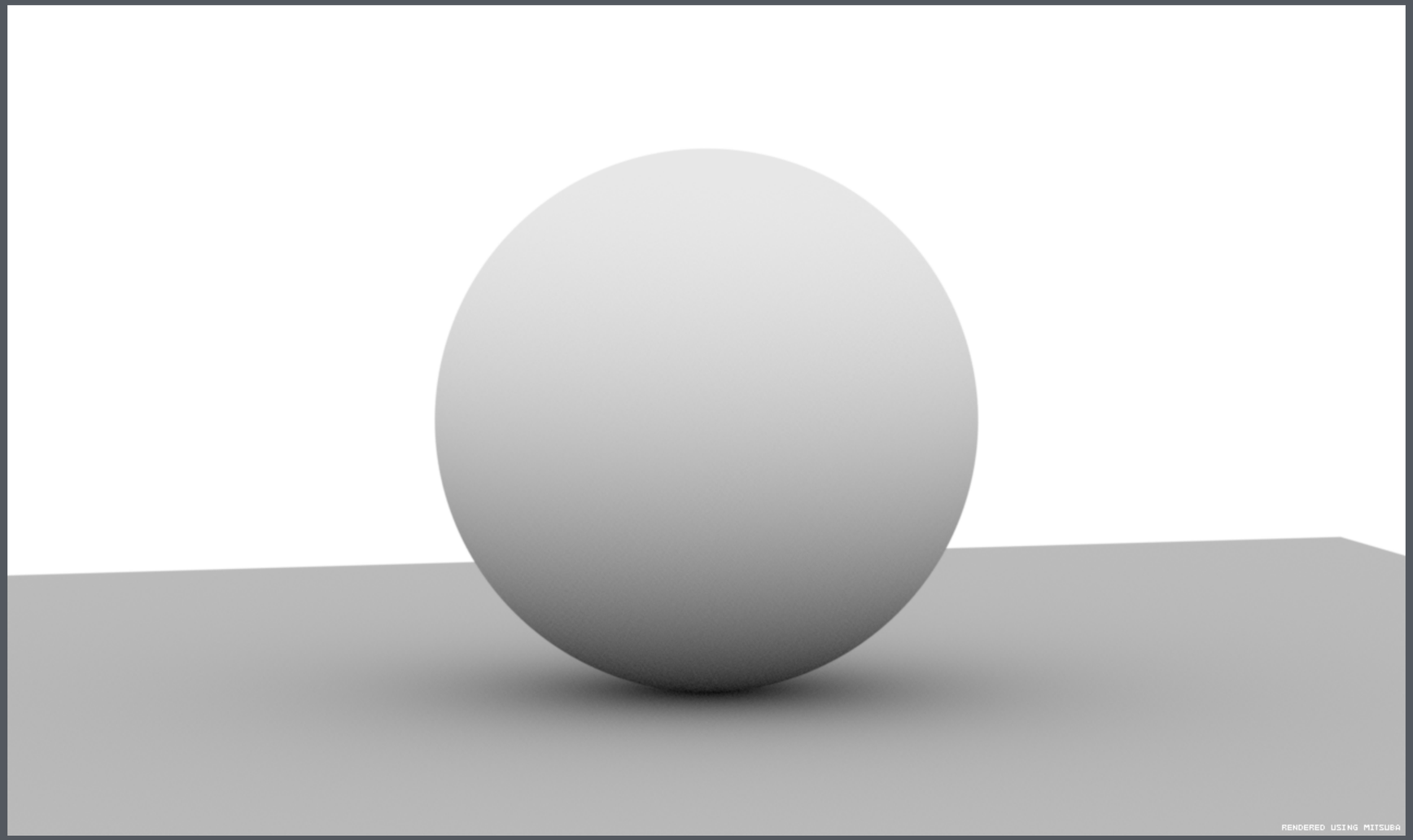
Irradiance texture computed
using rectangular light

Floor shaded with irradiance
from shadow texture

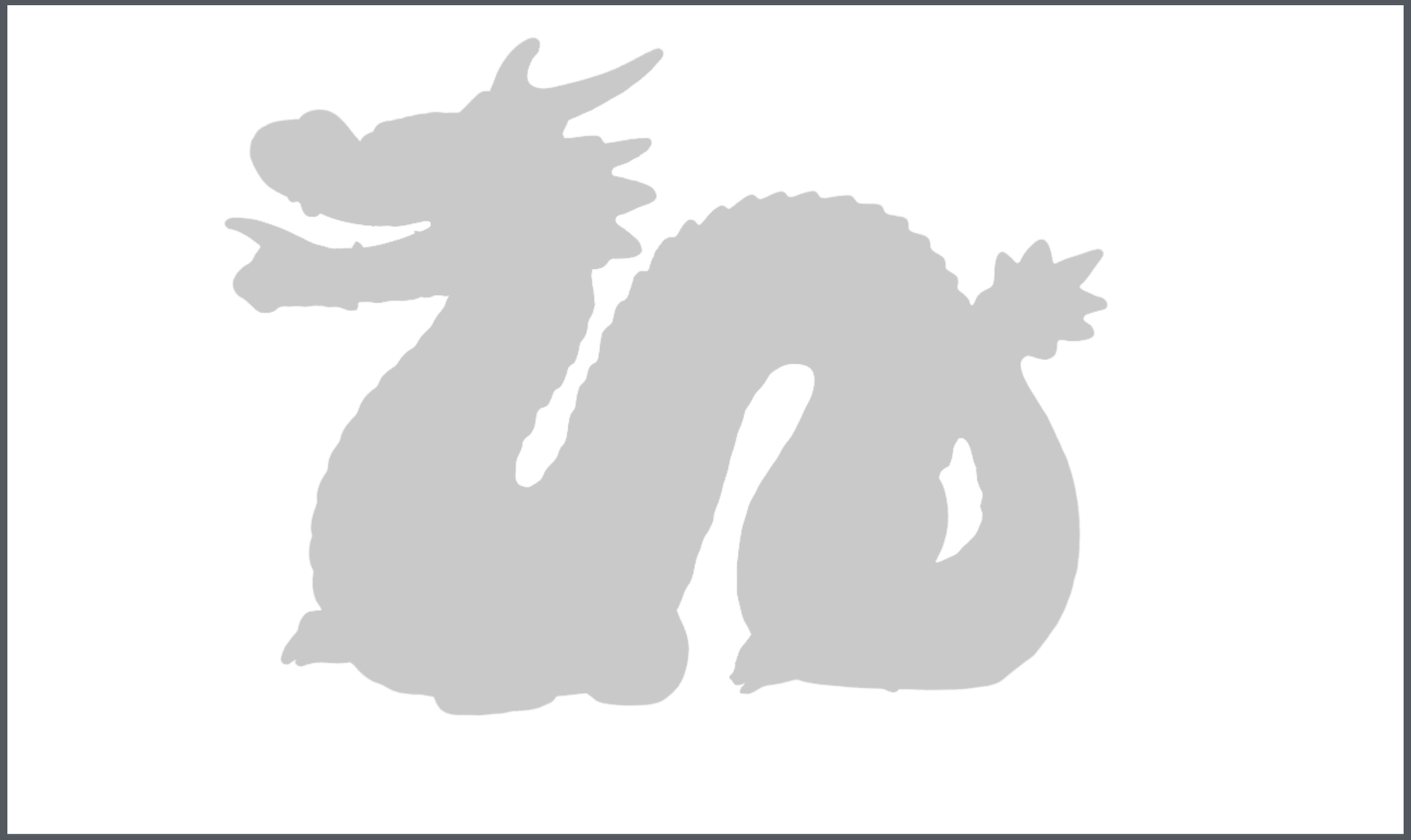
Ambient occlusion



a convex diffuse object in a constant-radiance environment

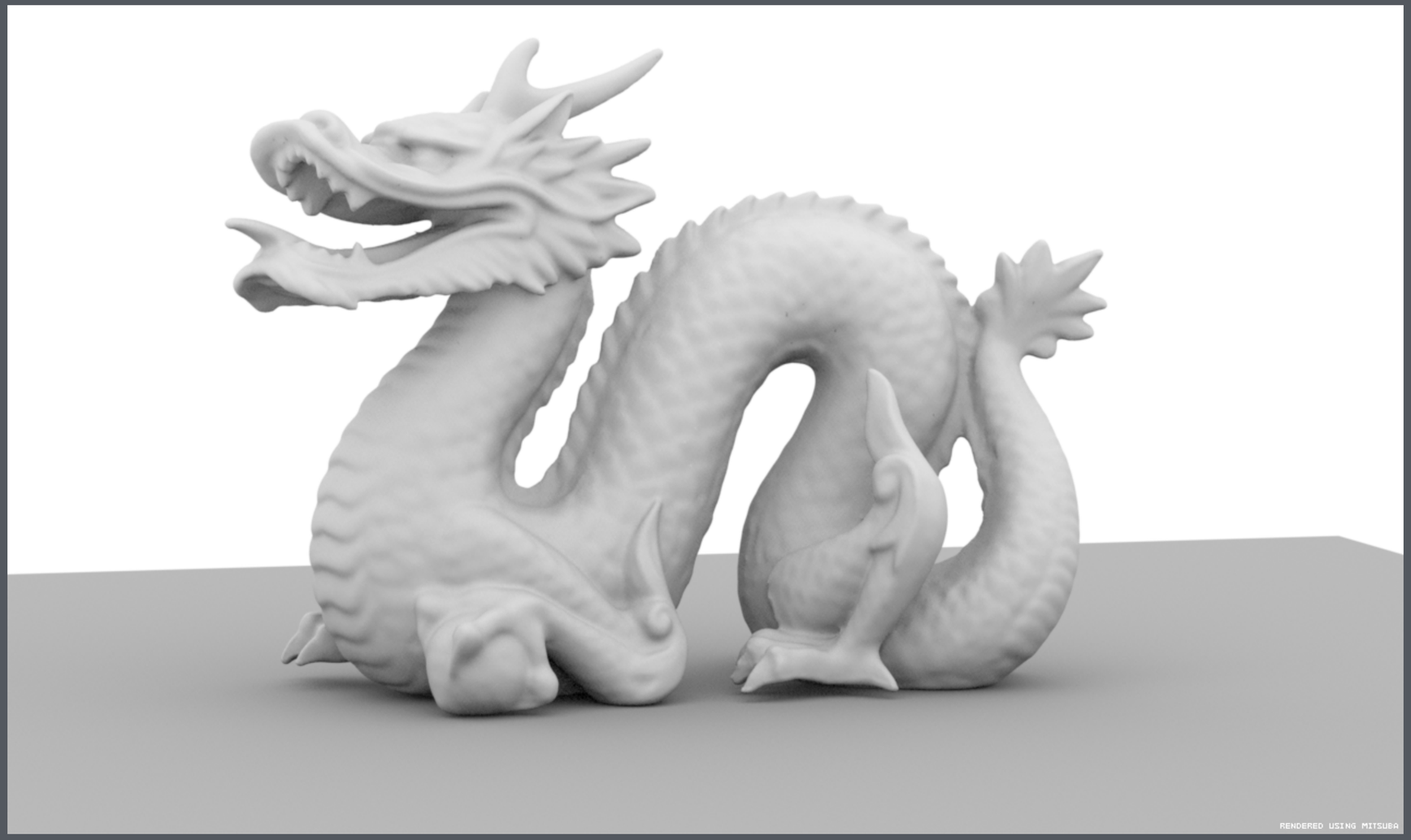


a non-convex diffuse scene under constant-radiance illumination



object in a constant-radiance environment with no shadowing

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



a non-convex diffuse scene under constant-radiance illumination

Ambient occlusion / obscurance

Ambient occlusion

- fraction of (cosine-weighted) rays that can see the sky
- corresponds to direct irradiance under a constant environment map

Ambient obscurance

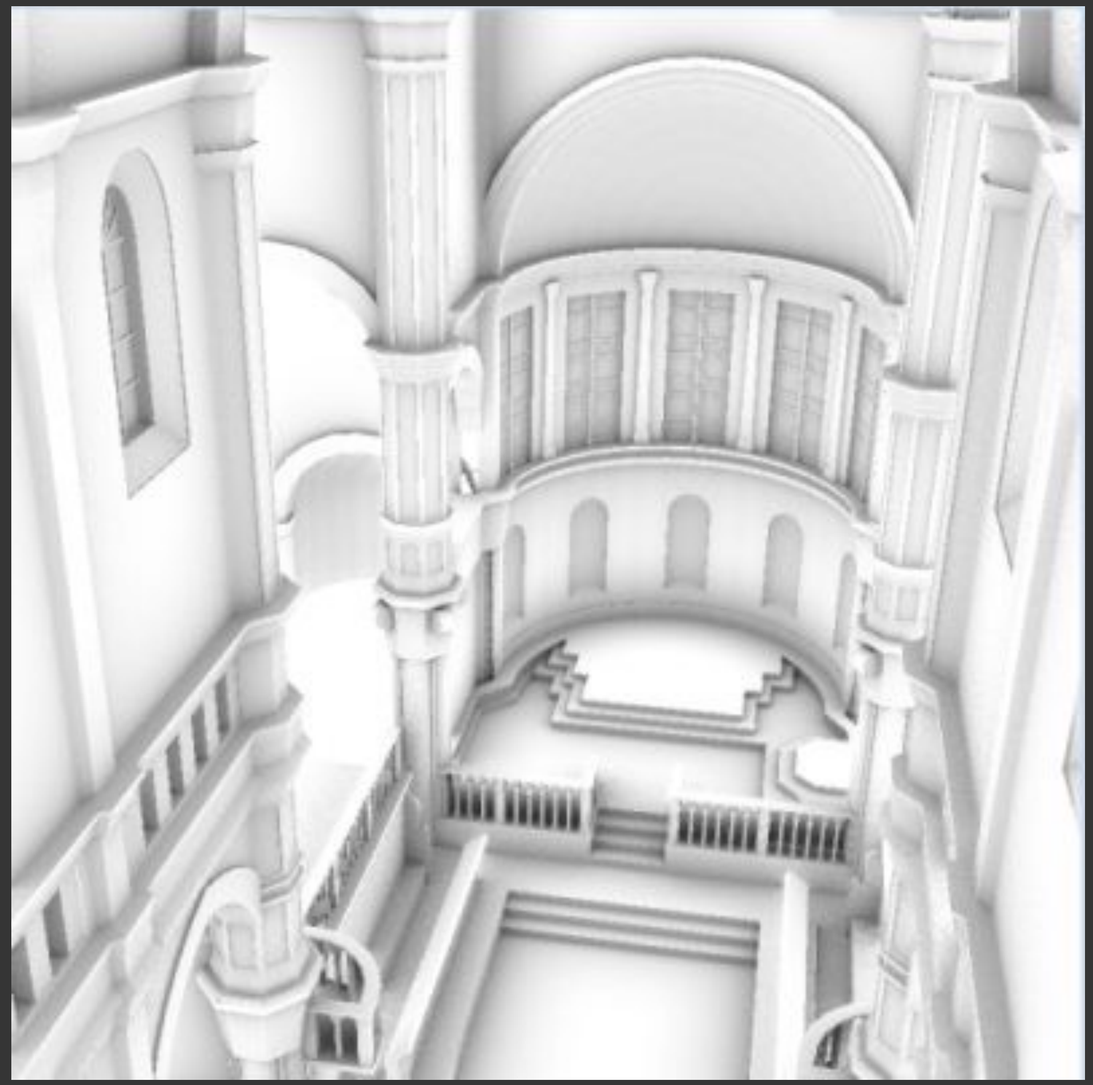
- (cosine-weighted) fraction of rays that can escape to a distance *r*
- useful for interior environments
- often a weighting factor is used leading to equations like

$$A_r(C, \mathbf{n}) = \frac{1}{\pi} \int_{H^2} g(t(C, \omega))(\mathbf{n} \cdot \omega)\, d\omega$$

- where *t*(*C*,ω) is the distance to the first intersection from *C* in the direction ω and *g*(*t*) is a smoothly increasing weighting function; *g*(0) = 0, *g*(*r*) = 1.

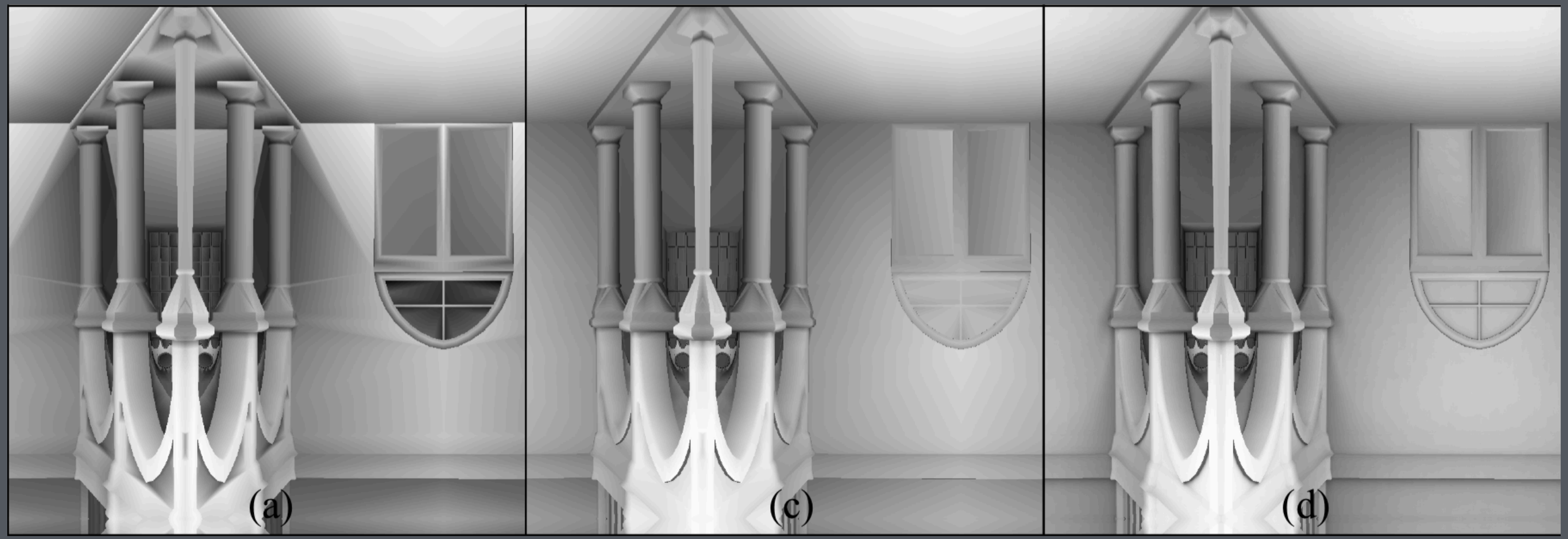
Precomputed ambient occlusion

AO Maps



slide courtesy of Kavita Bala, Cornell University

Ray traced vertex AO



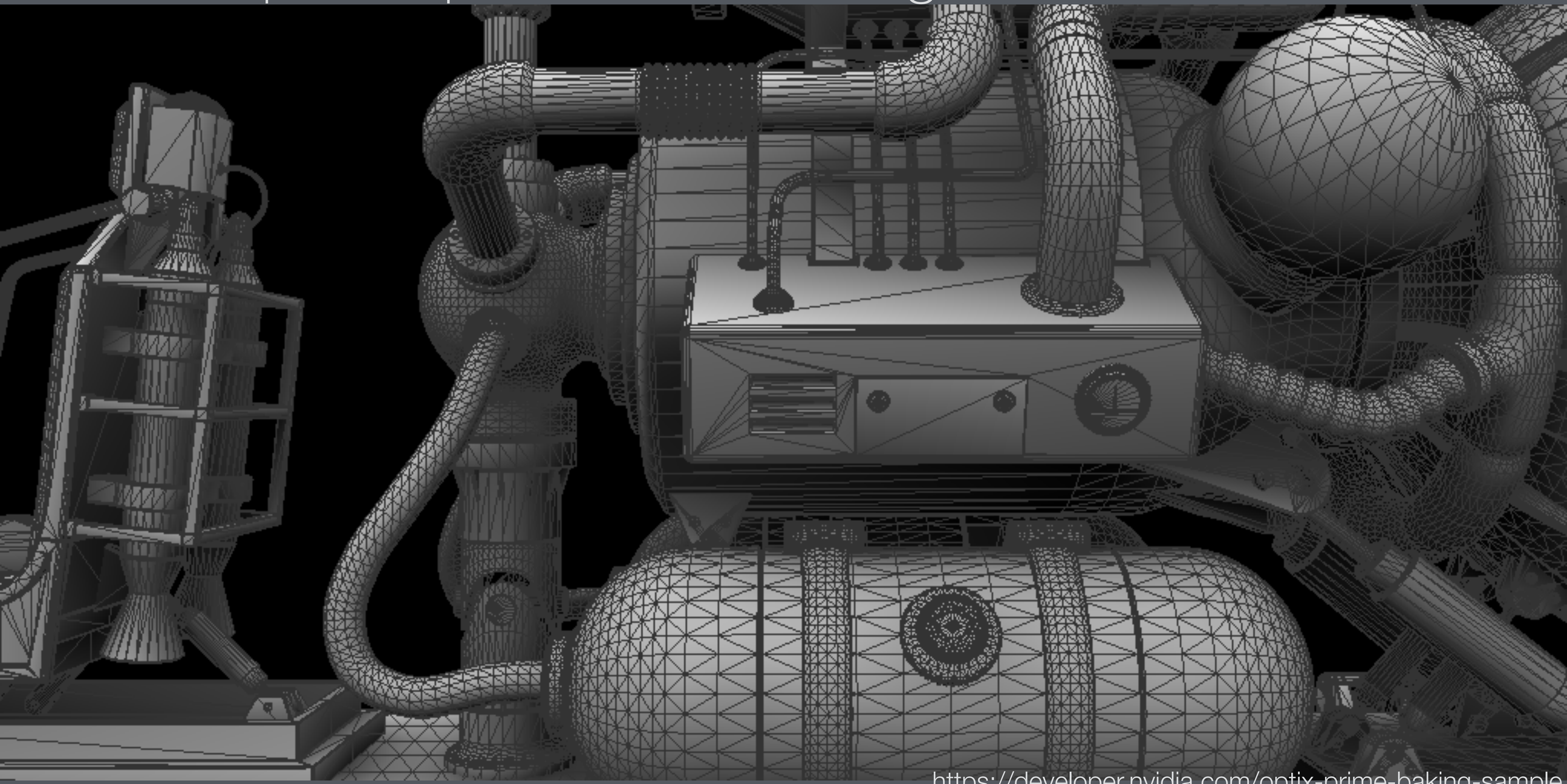
ambient occlusion sampled at vertices, interpolated as vertex color

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

ambient occlusion computed at each pixel (ground truth)

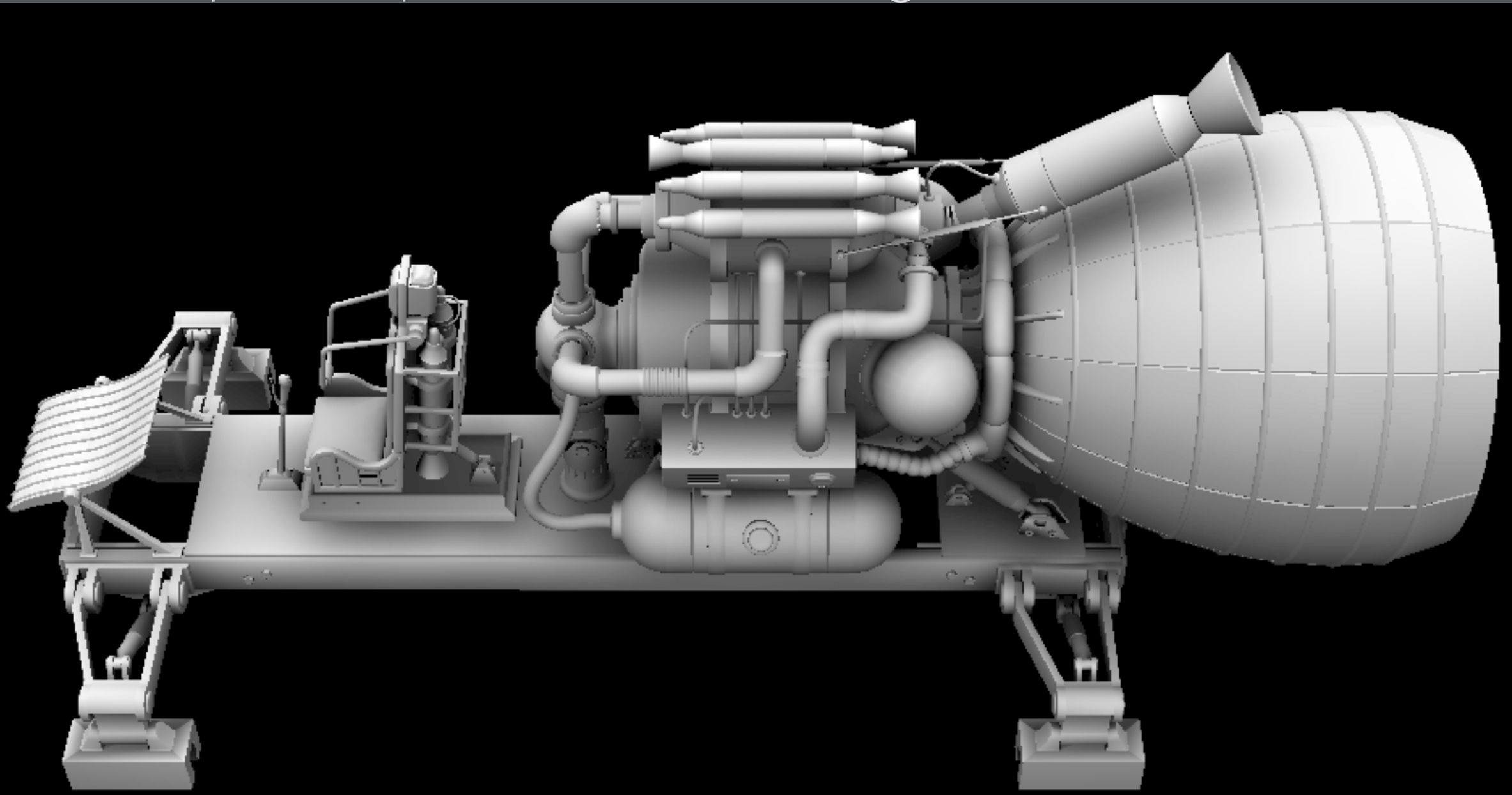
Kavan et al. EGSR 2011

NVIDIA OptiX implementation images



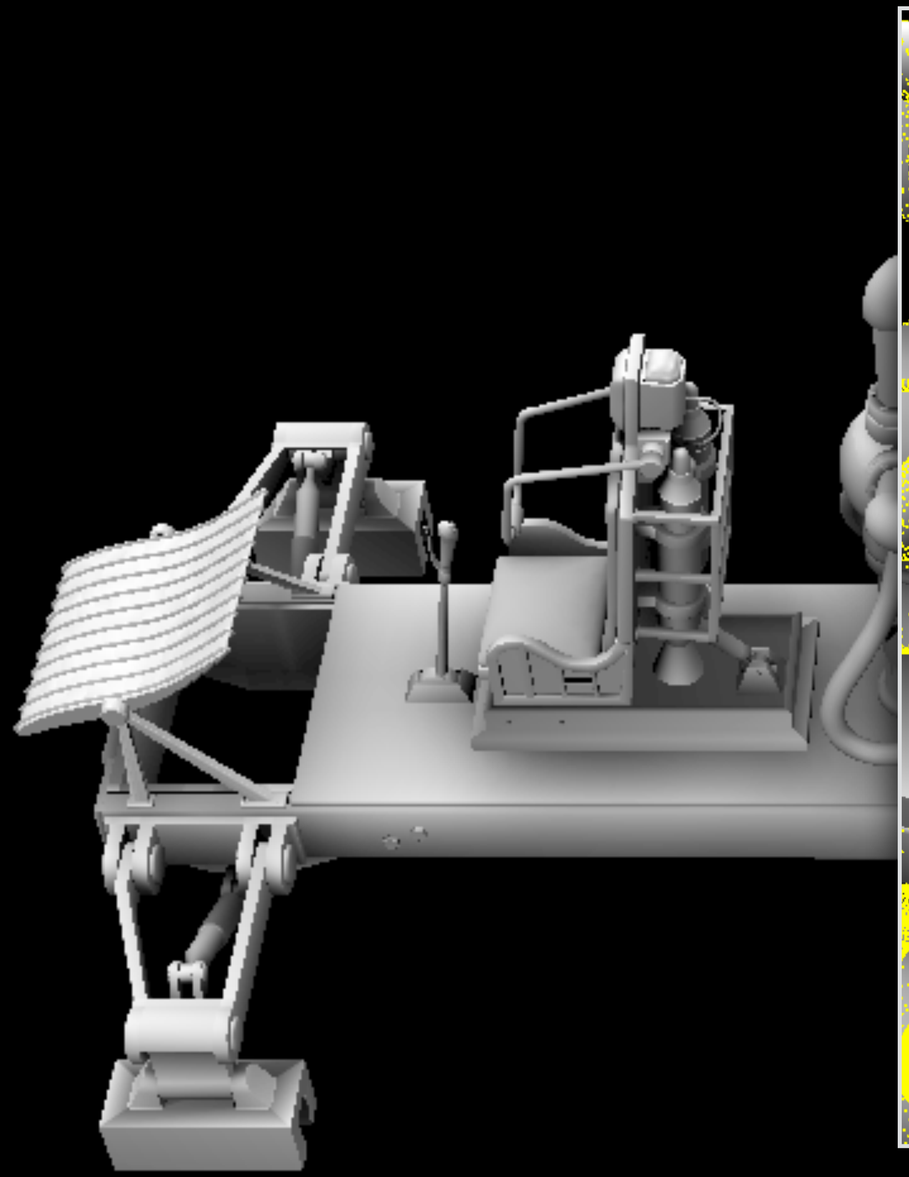
https://developer.nvidia.com/optix-prime-baking-sample

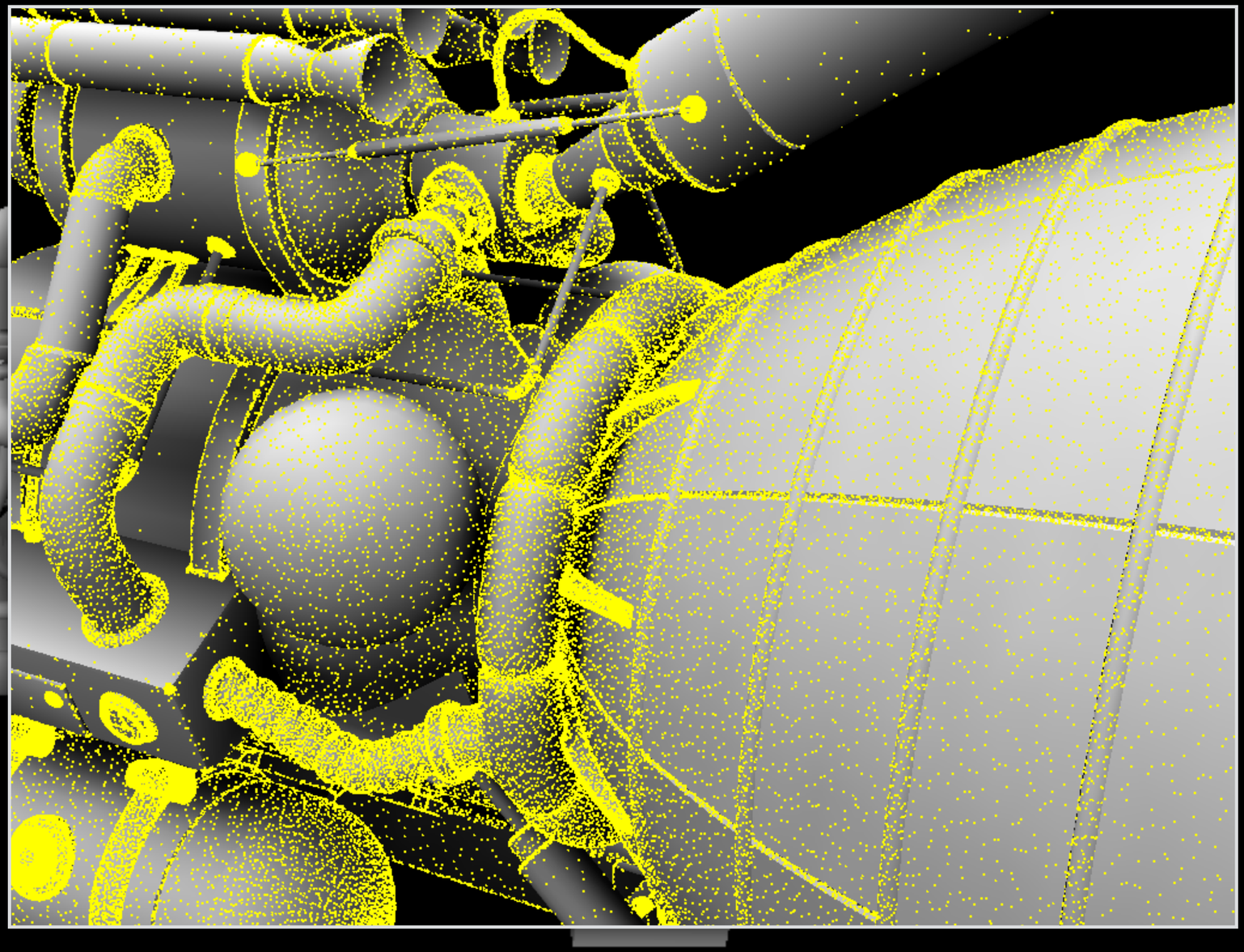
NVIDIA OptiX implementation images



https://developer.nvidia.com/optix-prime-baking-sample

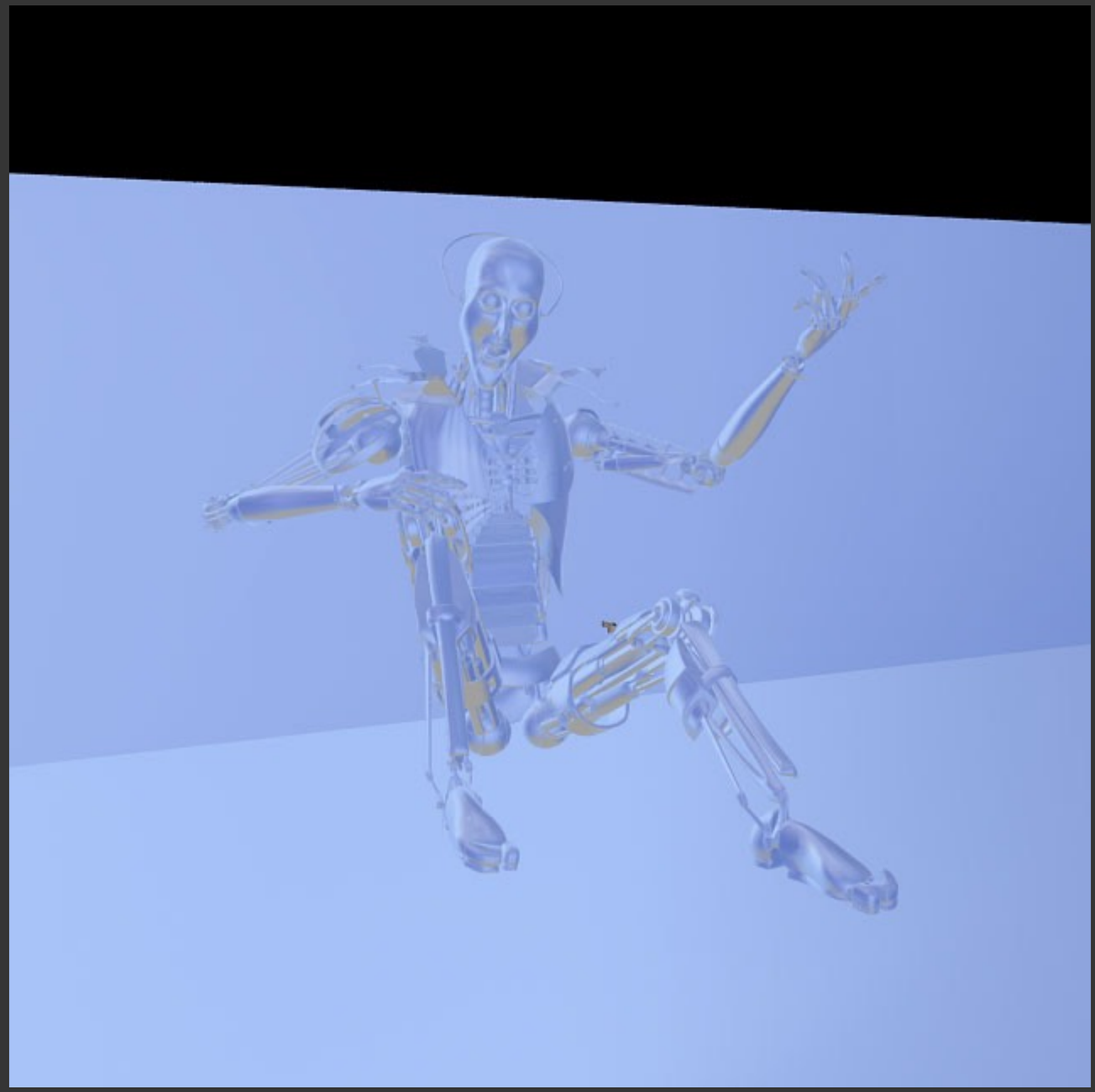
NVIDIA OptiX implementation images





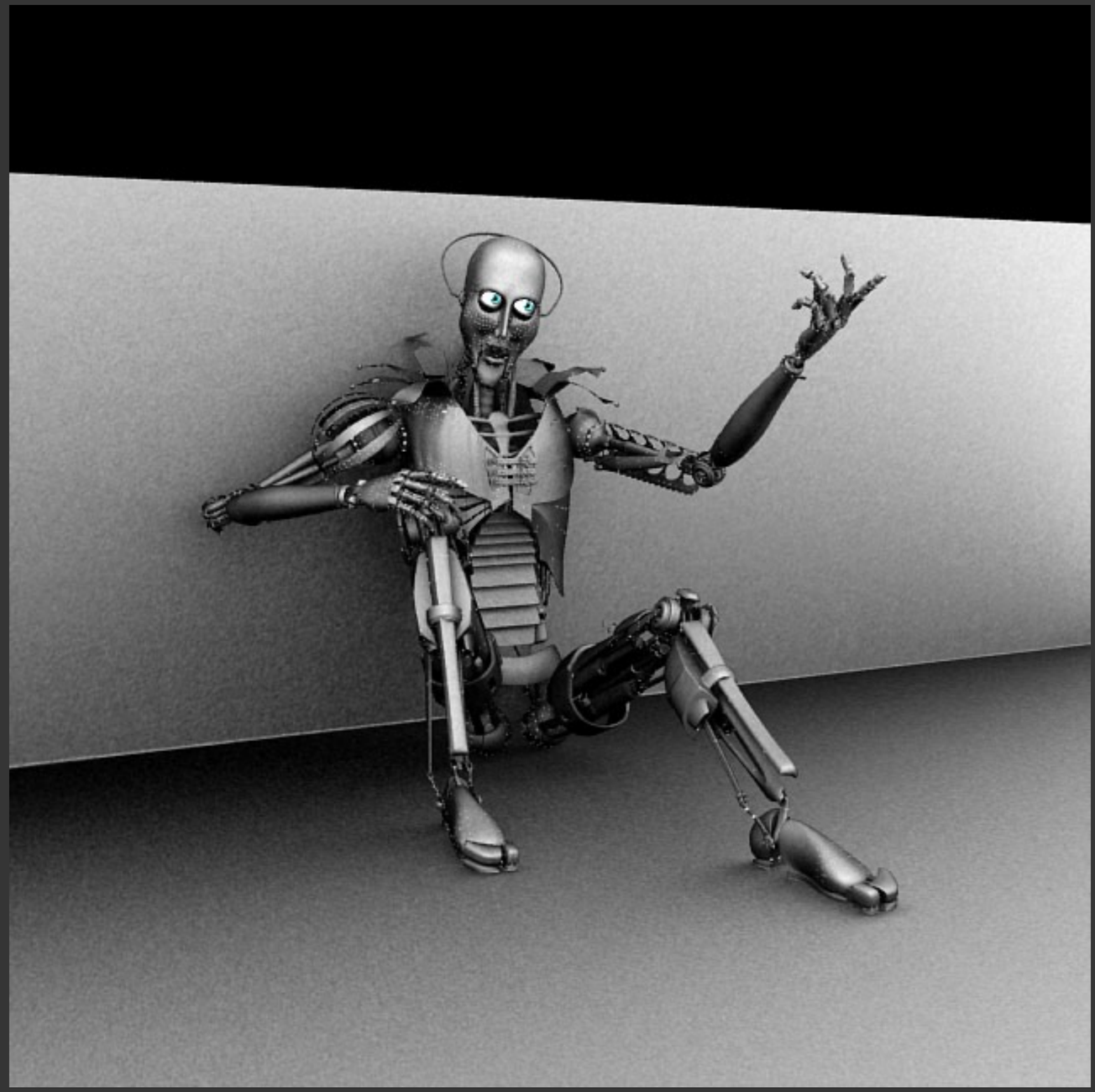
https://developer.nvidia.com/optix-prime-baking-sample

EnvMap



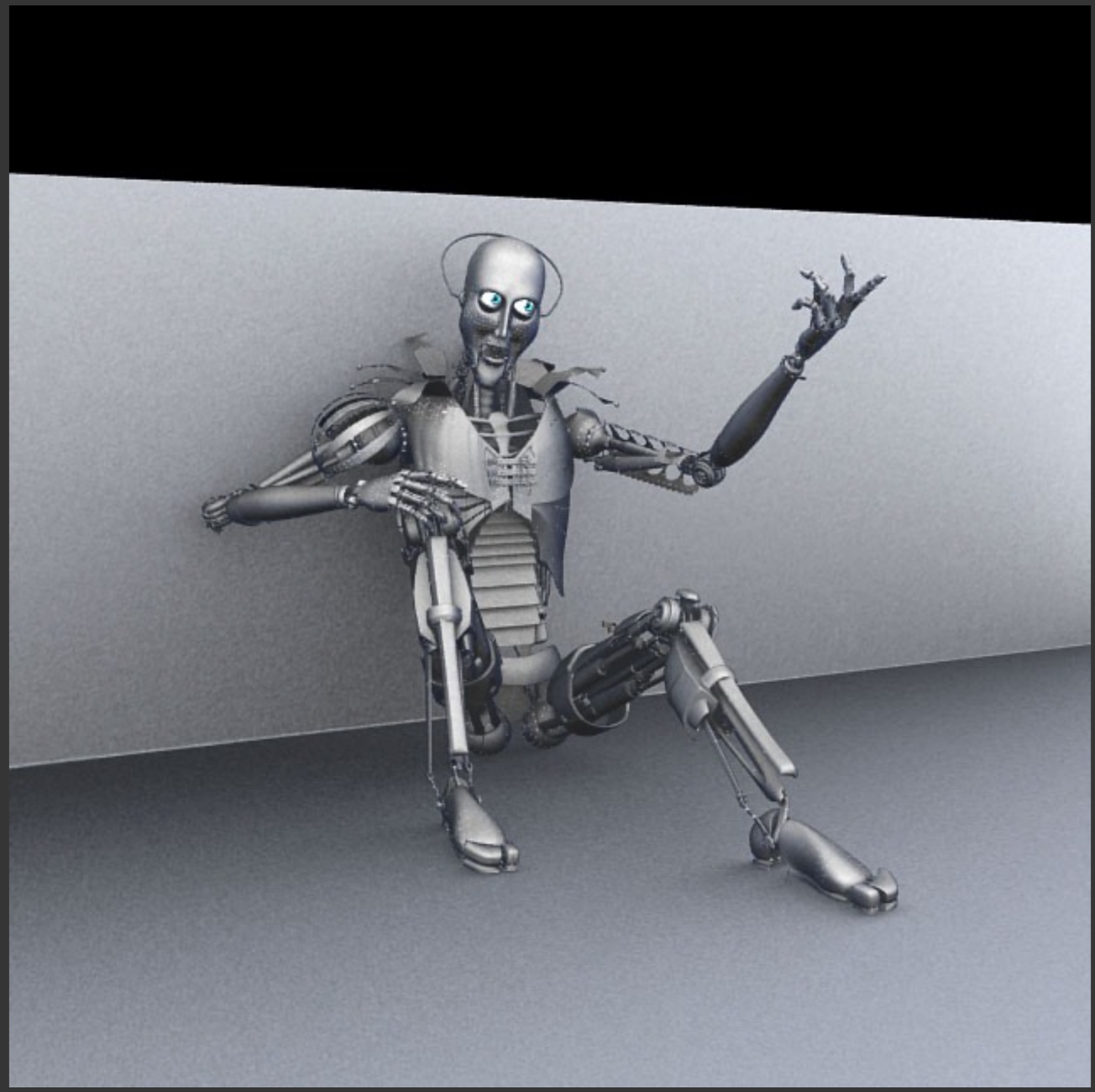
slide courtesy of Kavita Bala, Cornell University

AO



slide courtesy of Kavita Bala, Cornell University

Total



slide courtesy of Kavita Bala, Cornell University

Screen space ambient occlusion

Screen-space Ambient Occlusion (SSAO)

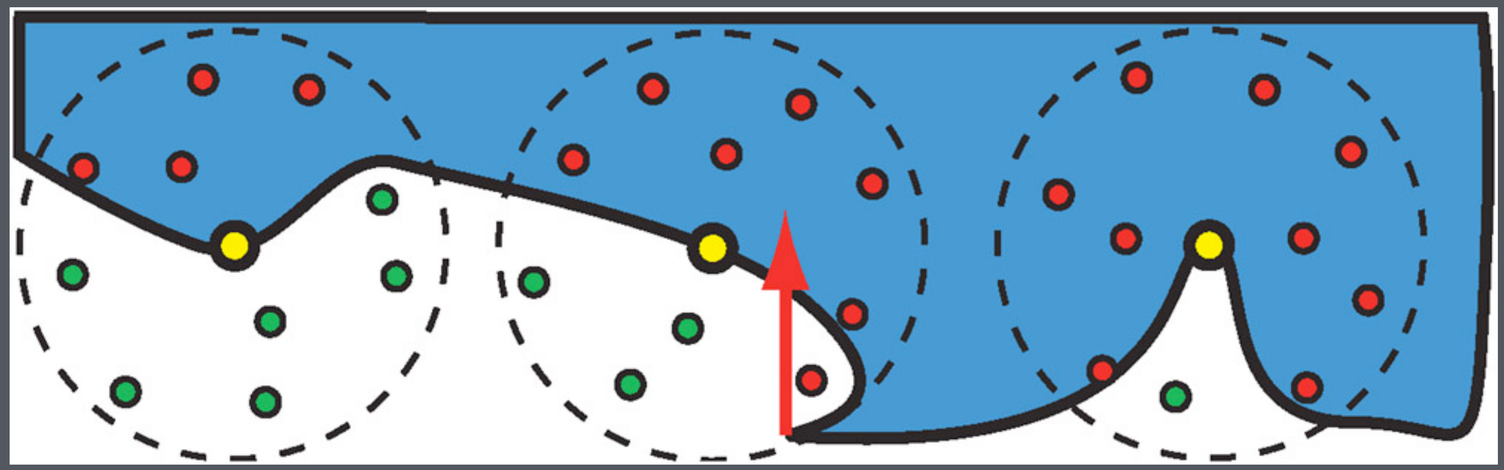
Idea: use z-buffer as a proxy for the local geometry

Redefine problem

- was: fraction of directions leaving point **p** that are unoccluded within distance *r*
- instead: fraction of sphere of radius *r* around point **p** occupied by geometry

Algorithm based on Monte Carlo sampling in this volume

- choose samples at random
- project to screen space
- compare to z buffer; behind = occluded



Akenine-Möller et al. *Real Time Rendering*

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

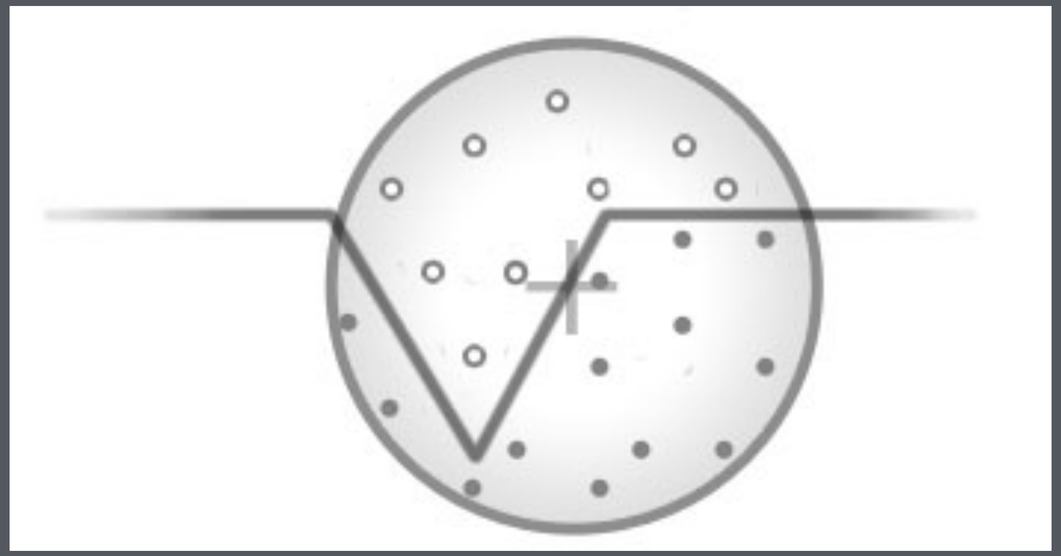
Martin Mittring, Crysis GmBH http://crytek.com/cryengine/presentations/finding-next-gen-cryengine--2

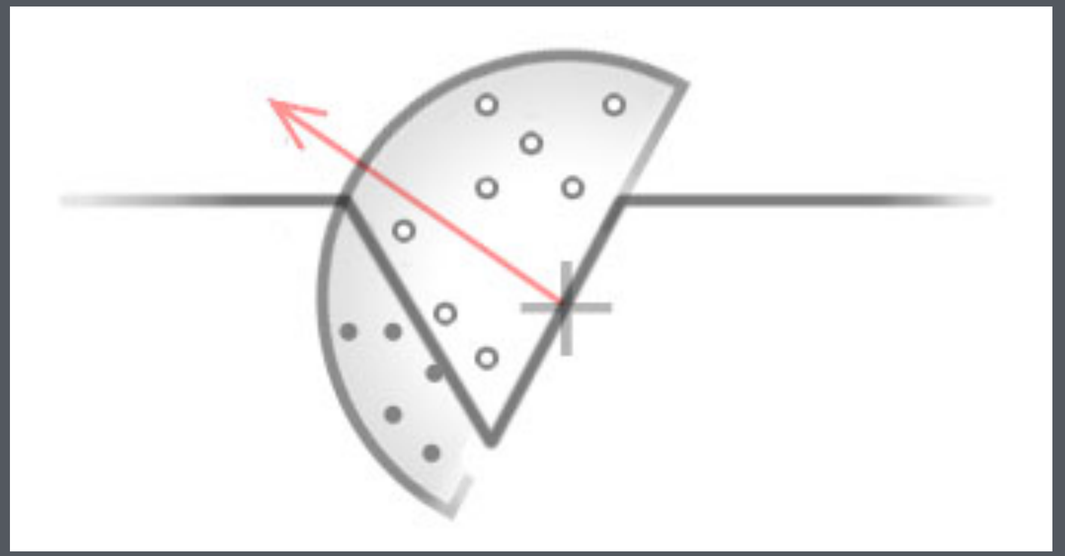
Martin Mittring, Crysis GmBH http://crytek.com/cryengine/presentations/finding-next-gen-cryengine--2

Restricting to hemisphere

Full sphere is affected by geometry behind the surface

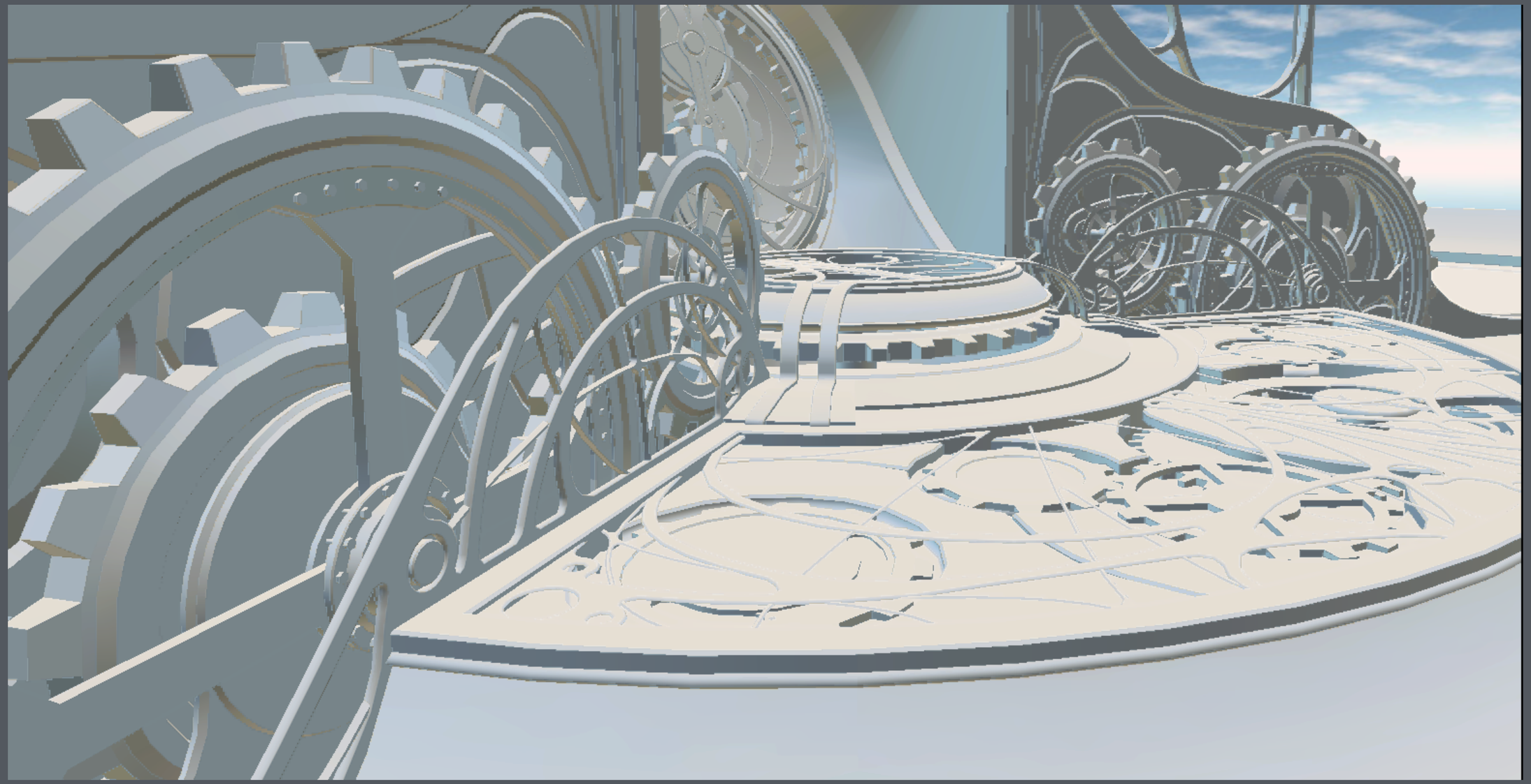
- surface normal may be available (as it is with deferred shading)
- then sample points can be distributed in hemisphere rather than sphere





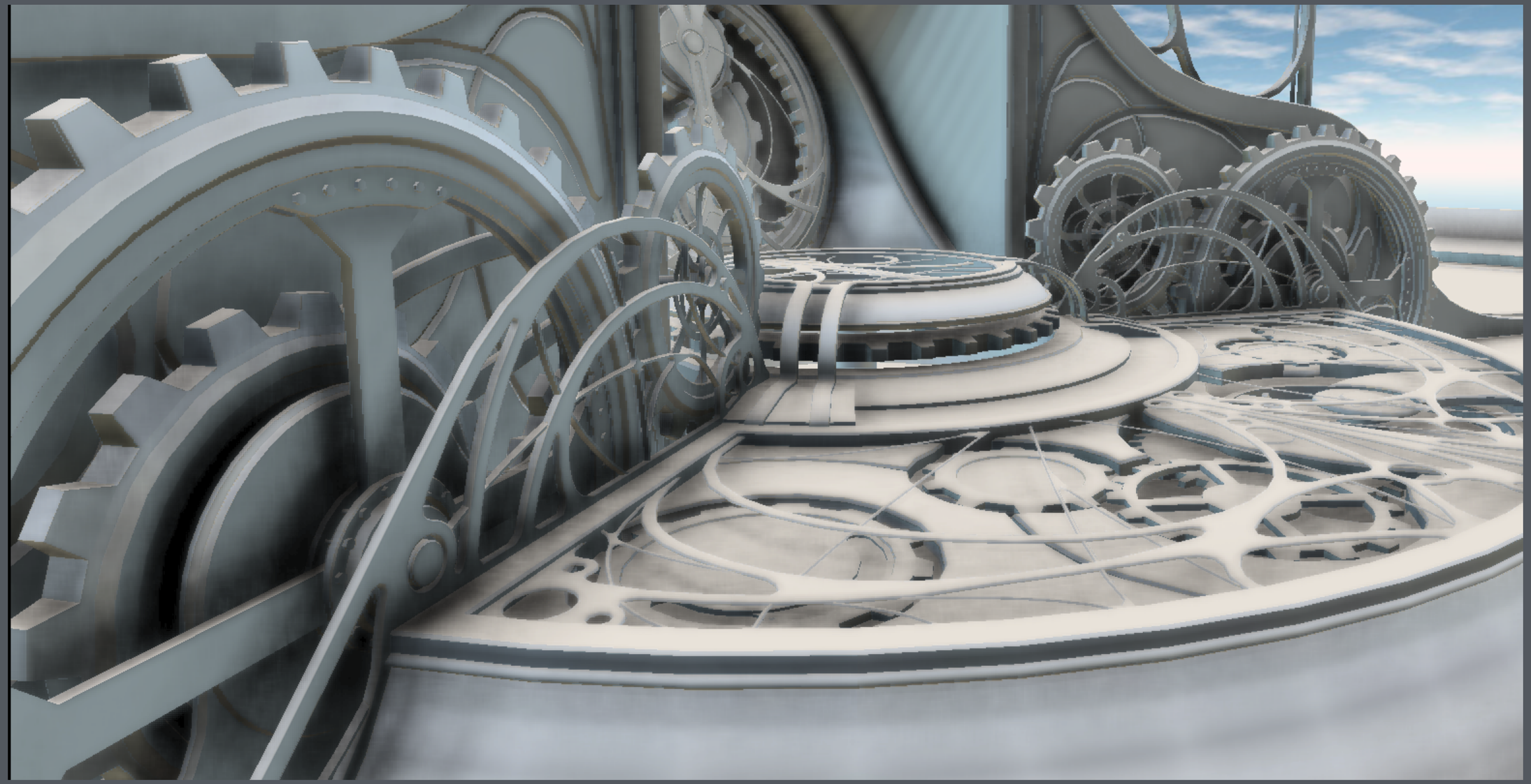
John Chapman http://john-chapman-graphics.blogspot.co.uk/2013/01/ssao-tutorial.html

Irradiance map + SSAO



McGuire et al. HPG ’11 10.1145/2018323.2018327

Irradiance map + SSAO



McGuire et al. HPG ’11 10.1145/2018323.2018327

Irradiance map + SSAO

McGuire et al. HPG '11 10.1145/2018323.2018327

Irradiance map + SSAO

McGuire et al. HPG '11 10.1145/2018323.2018327