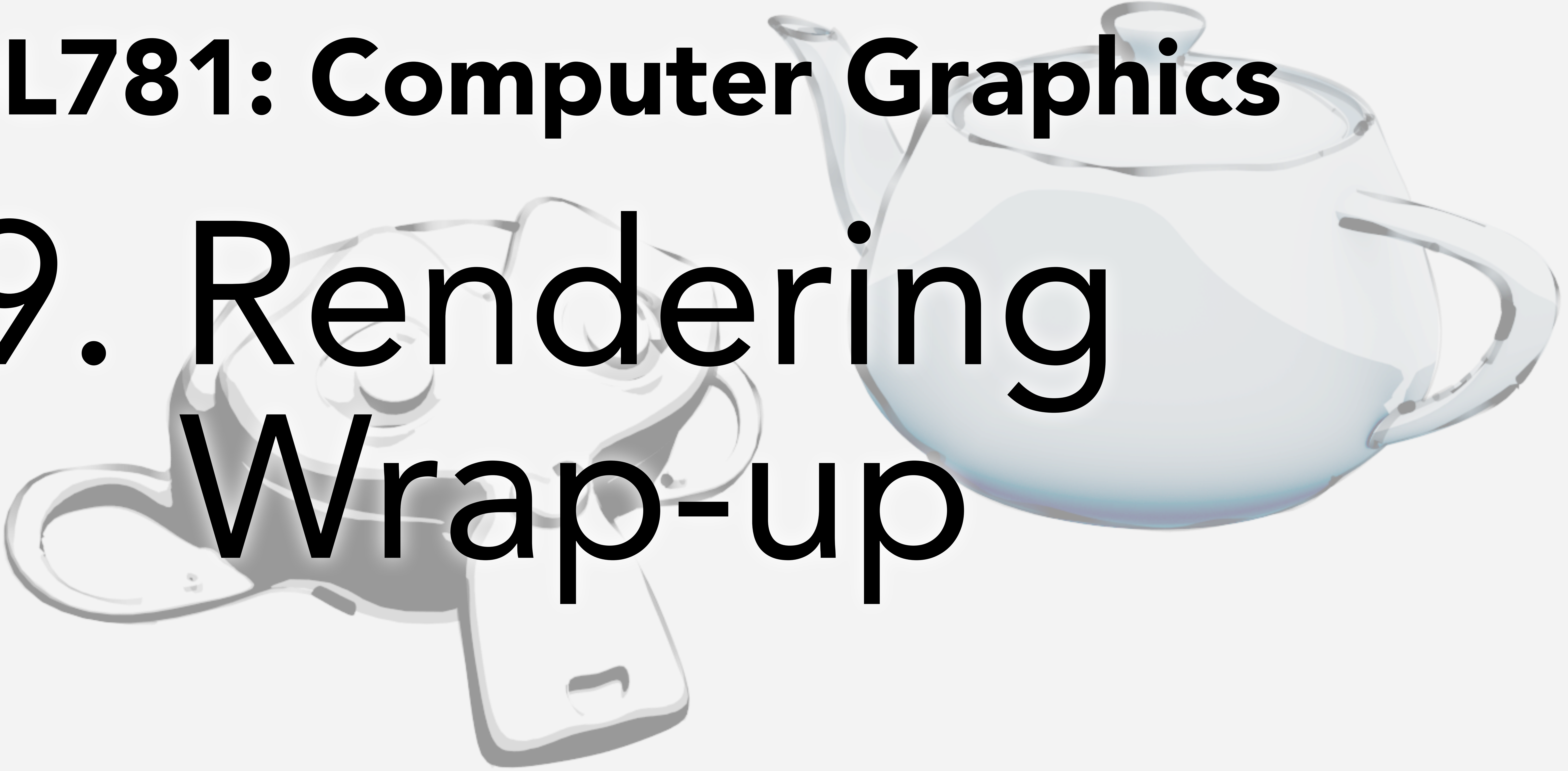
COL781: Computer Graphics

29. Rendering Wrap-up



Michael Collins

Recap: Precomputed radiance transfer (PRT)

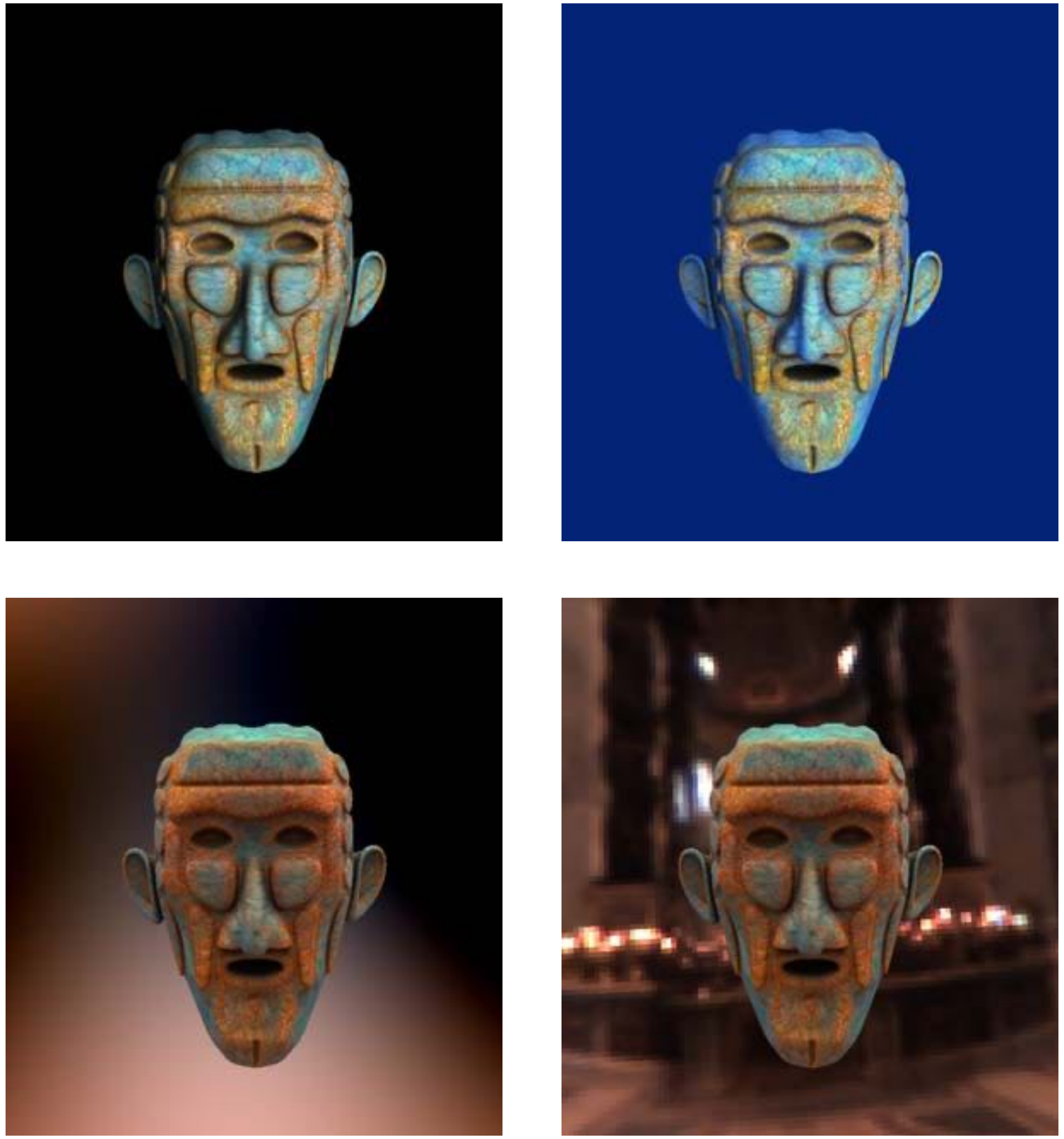
Precompute all the light transport in the scene, assuming…

Fixed:

- Scene geometry
- Materials (let's assume diffuse for simplicity)

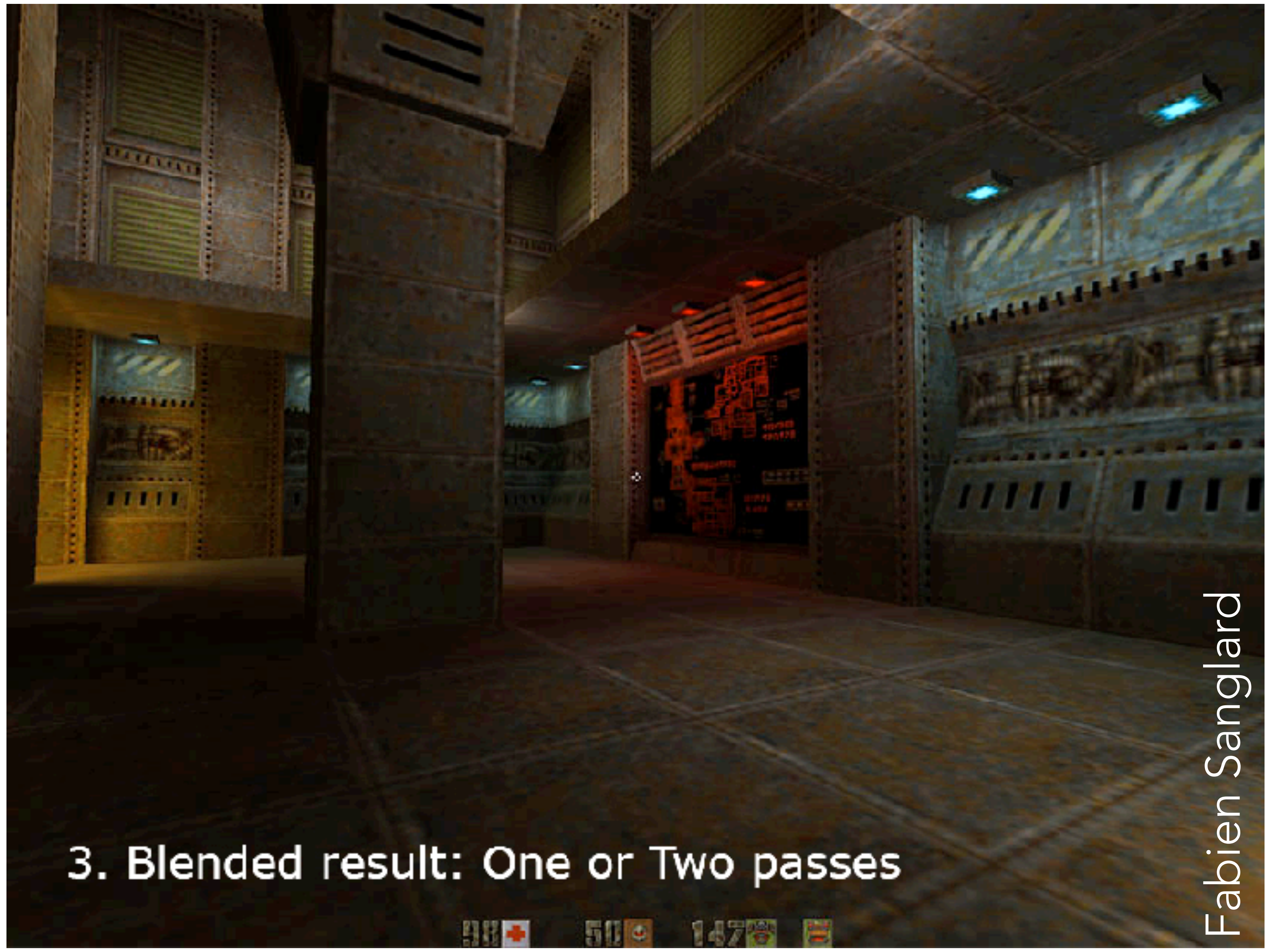
Variable:

- Environment lighting



What if lighting is fixed? Just precompute all global illumination and store: **lightmap**

Lightmaps
in *Quake II*



Fabien Sanglard

At any point **p**, we just need exitant radiance $L_o(\mathbf{p})$ as a function of scene illumination L_{env}

This function is always linear!

Express in some basis $L_{env} = \ell_1 B_1 + \ell_2 B_2 + \cdots$,
then $L_o[L_{env}] = \ell_1 L_o[B_1] + \ell_2 L_o[B_2] + \cdots$

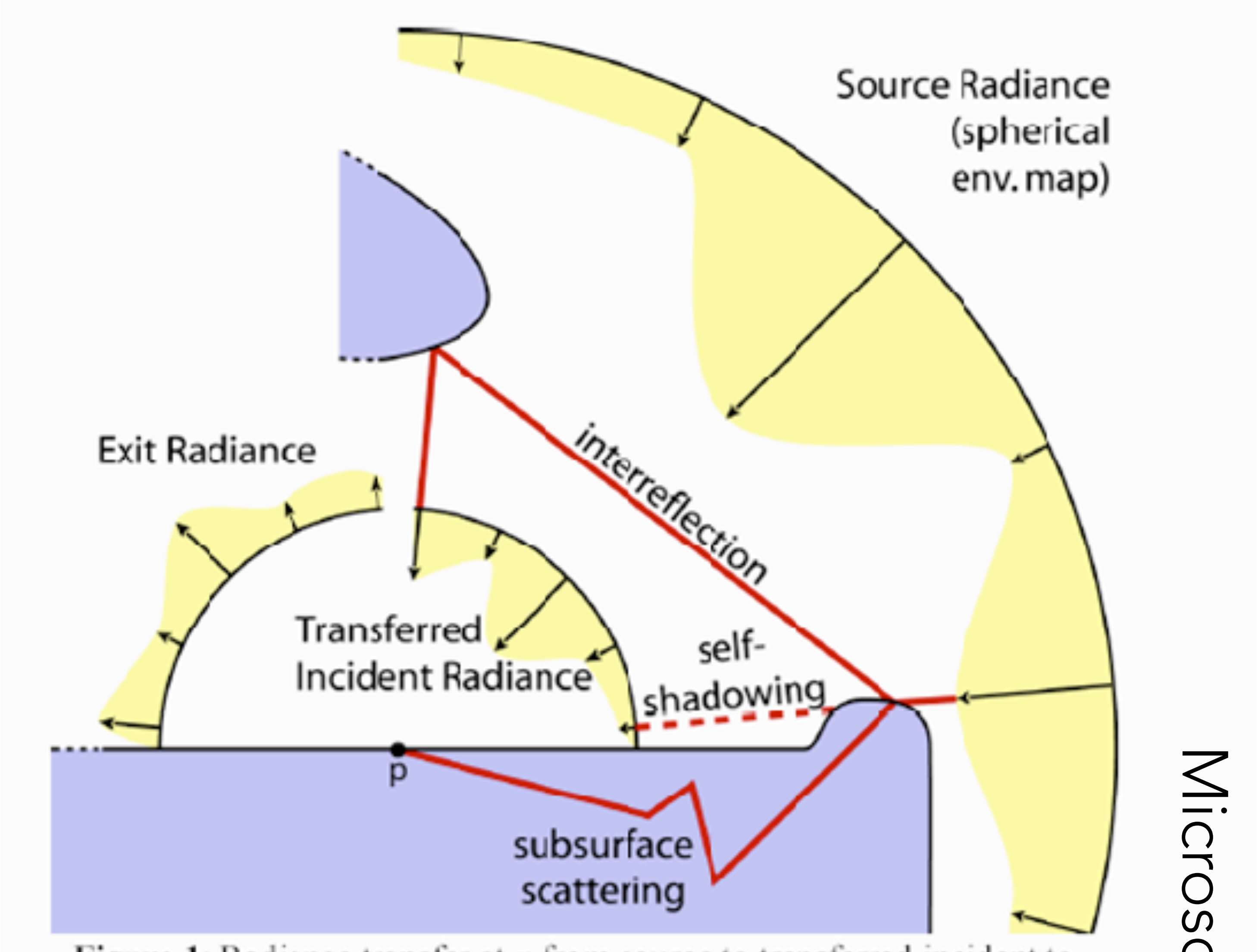
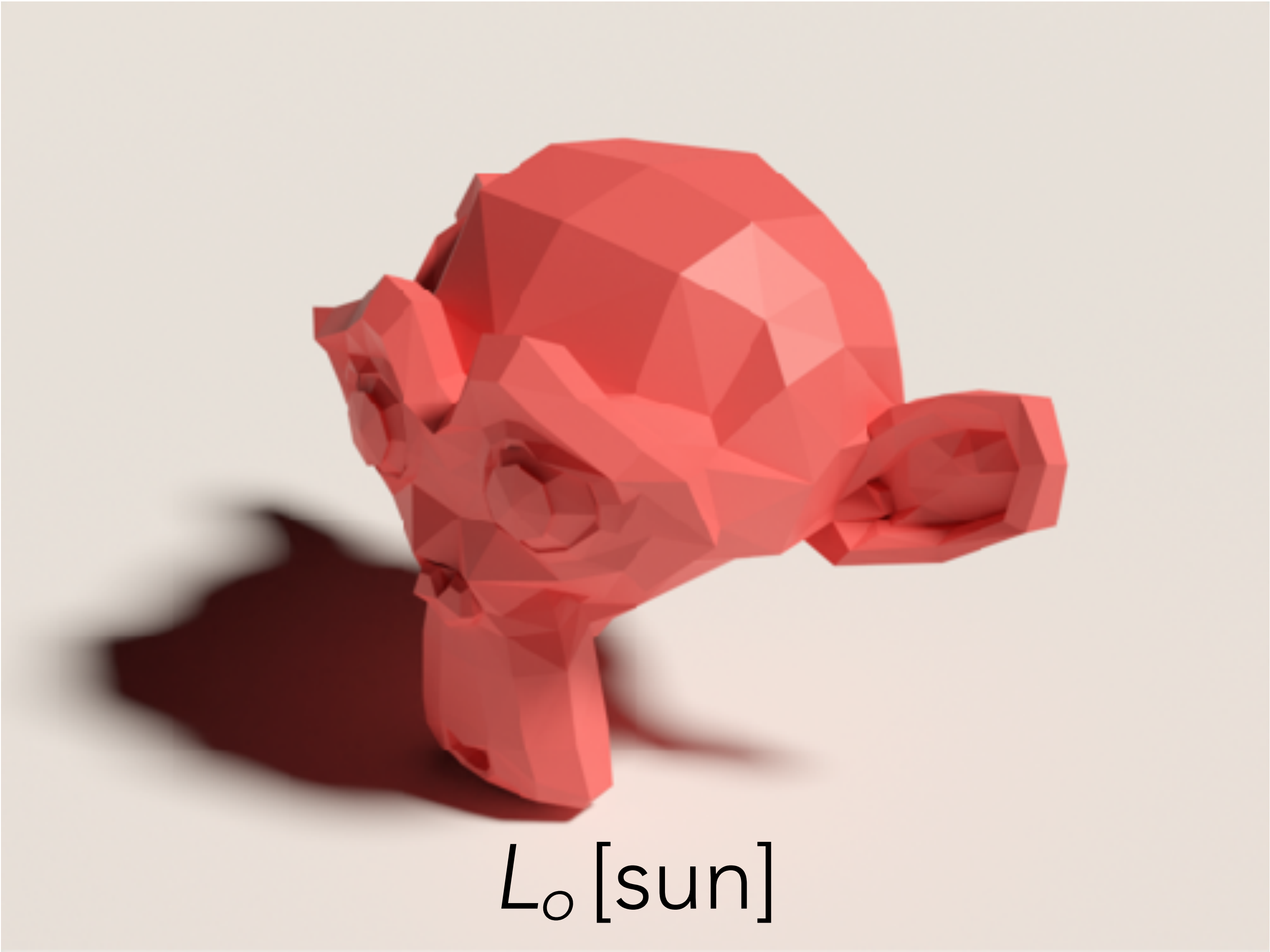
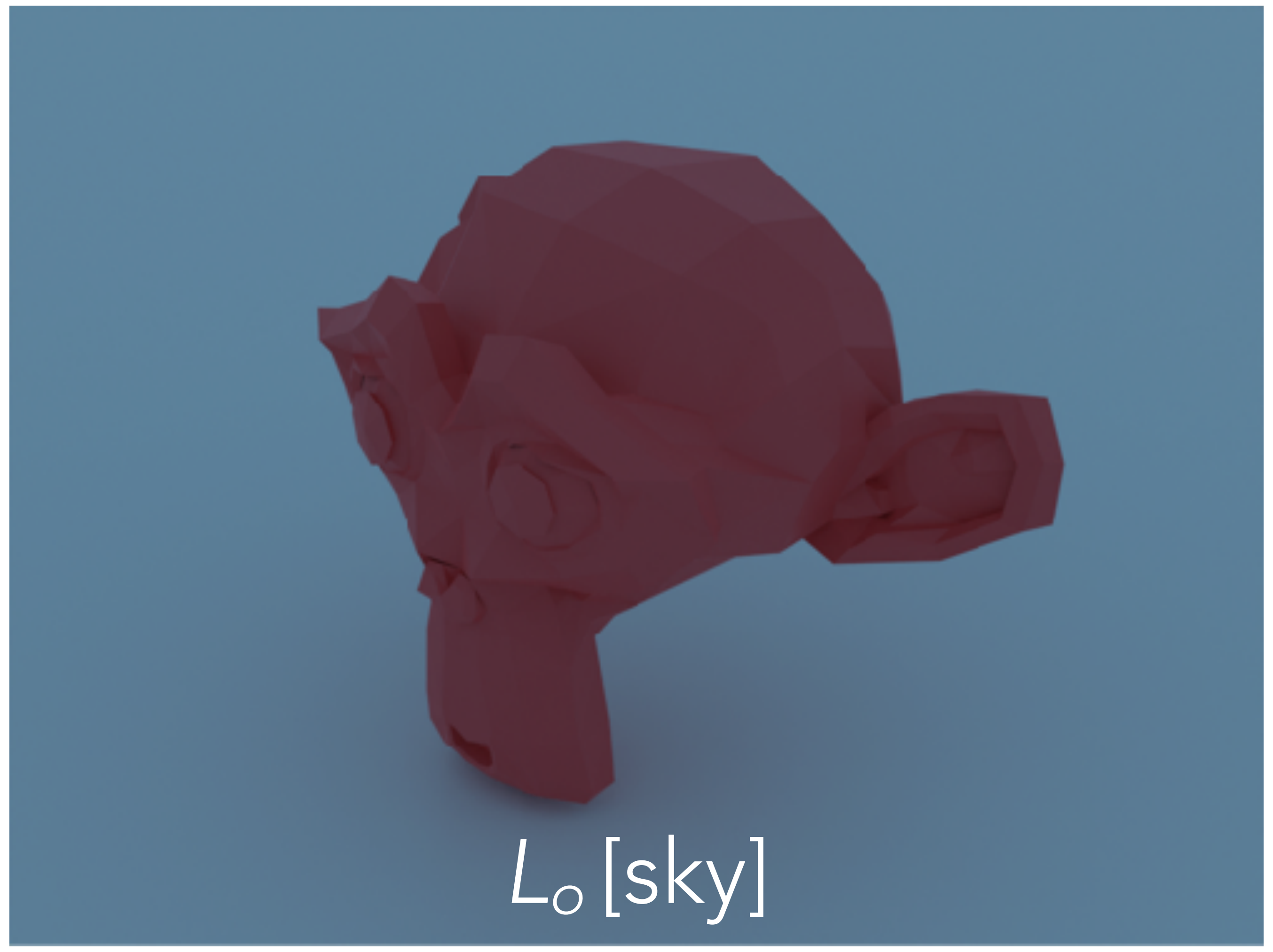


Figure 1: Radiance transfer at p from source to transferred incident to exit.

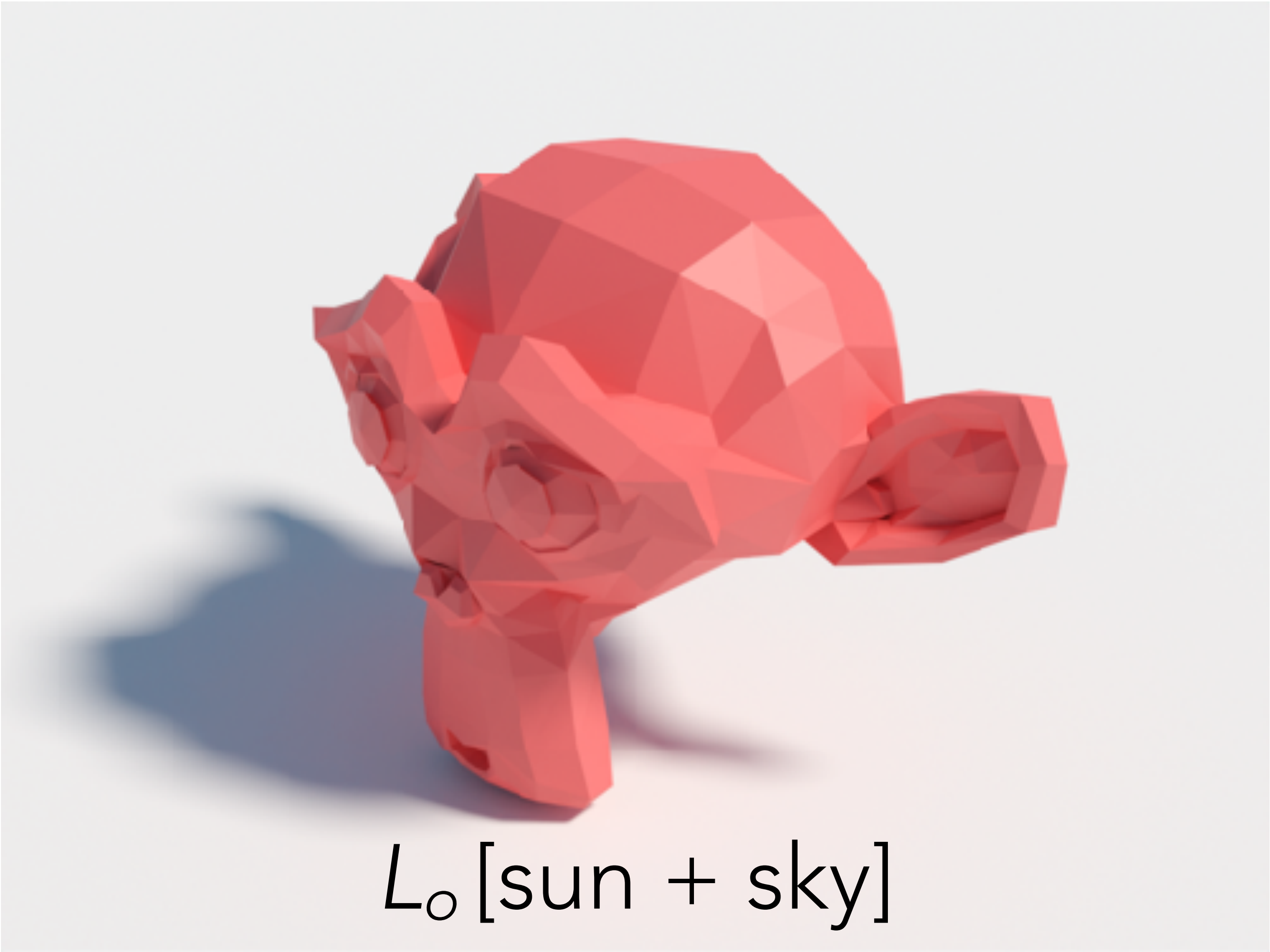
Microsoft



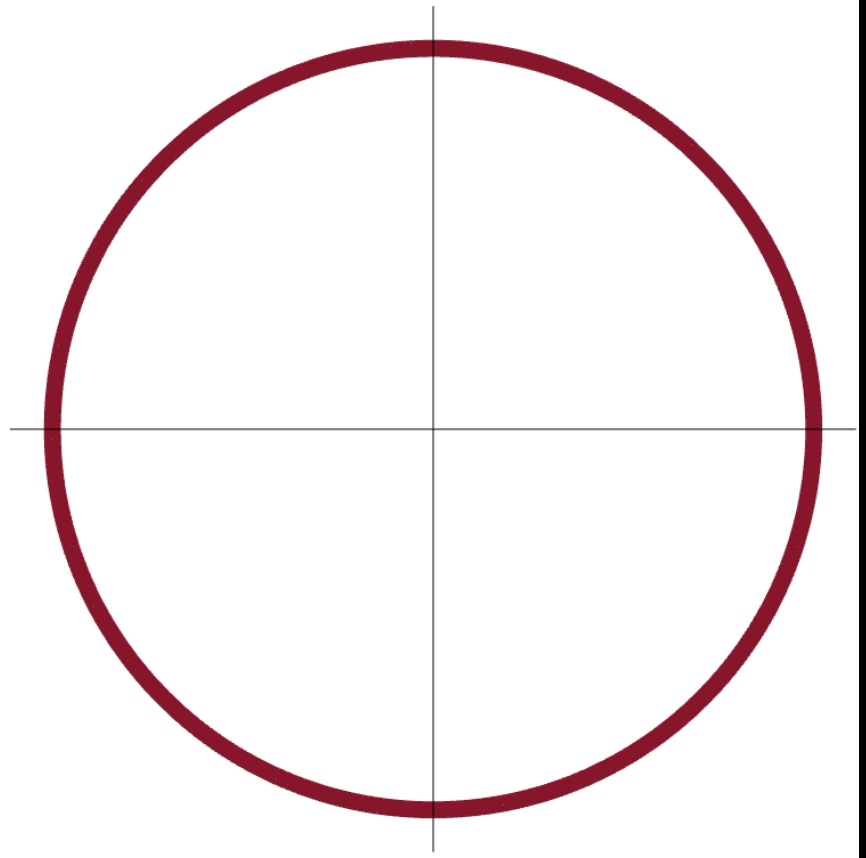
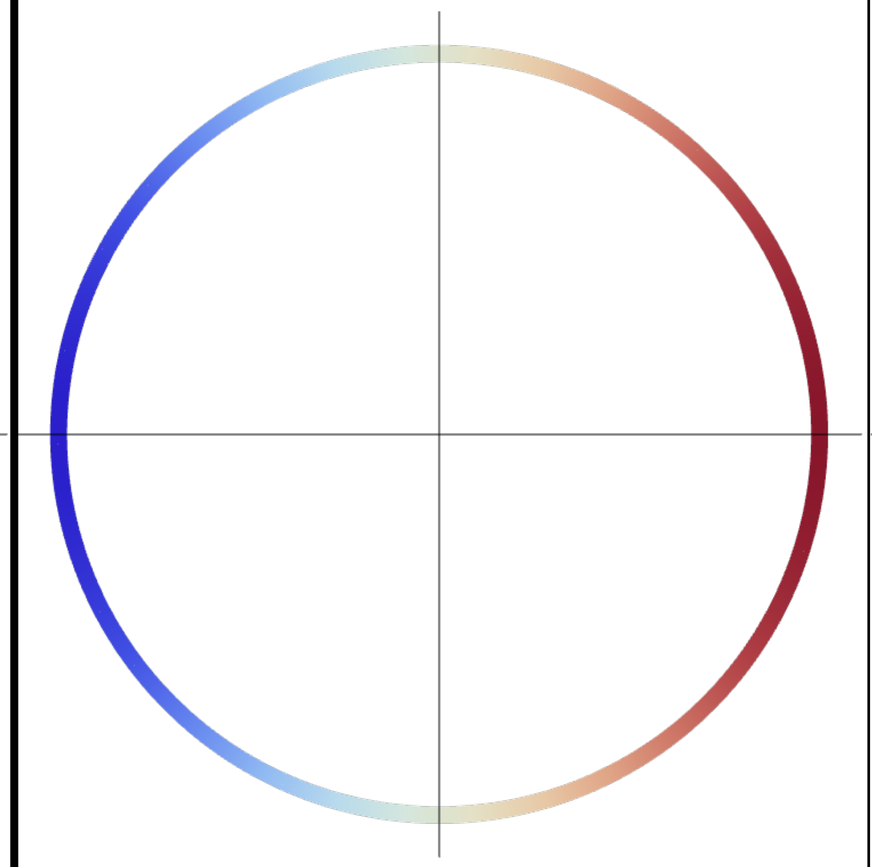
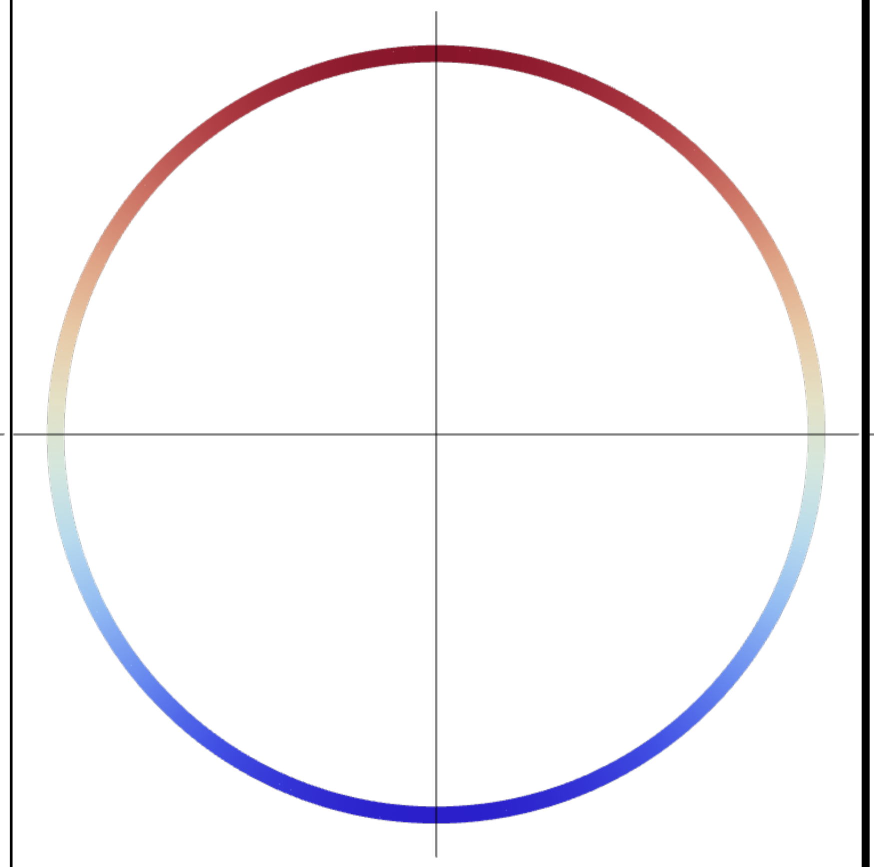
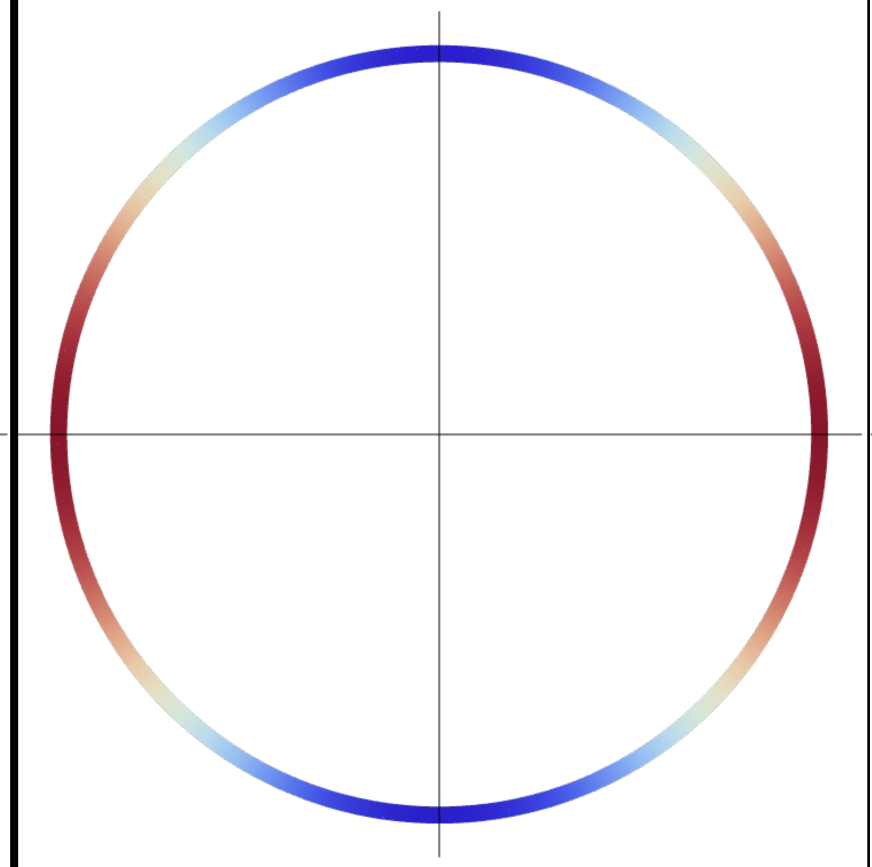
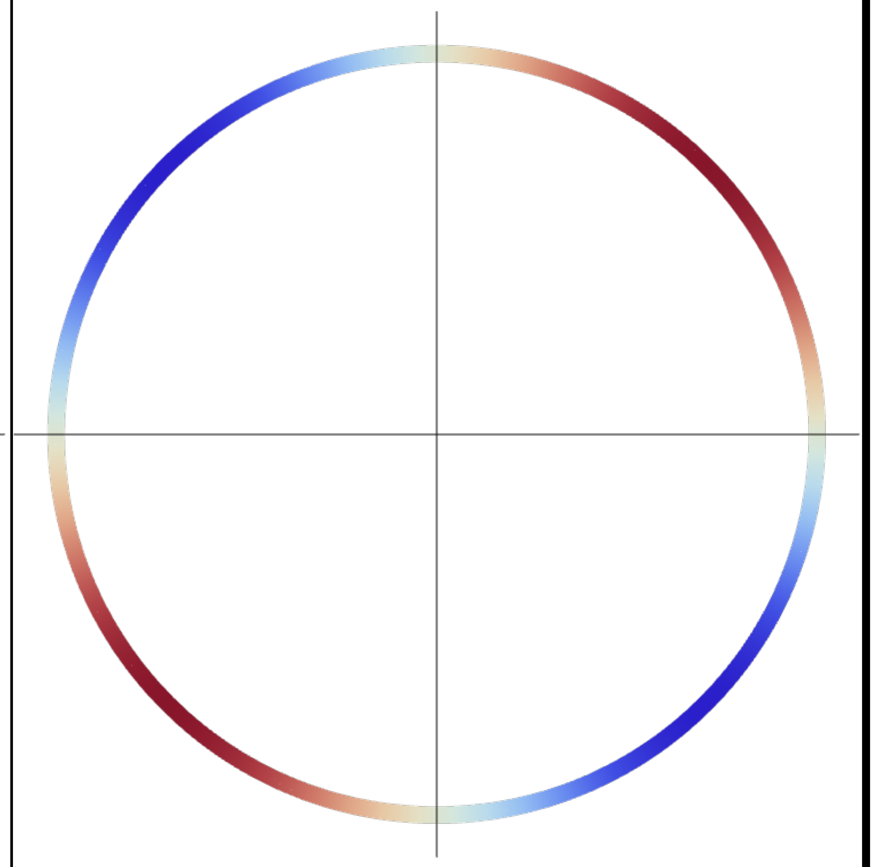
+



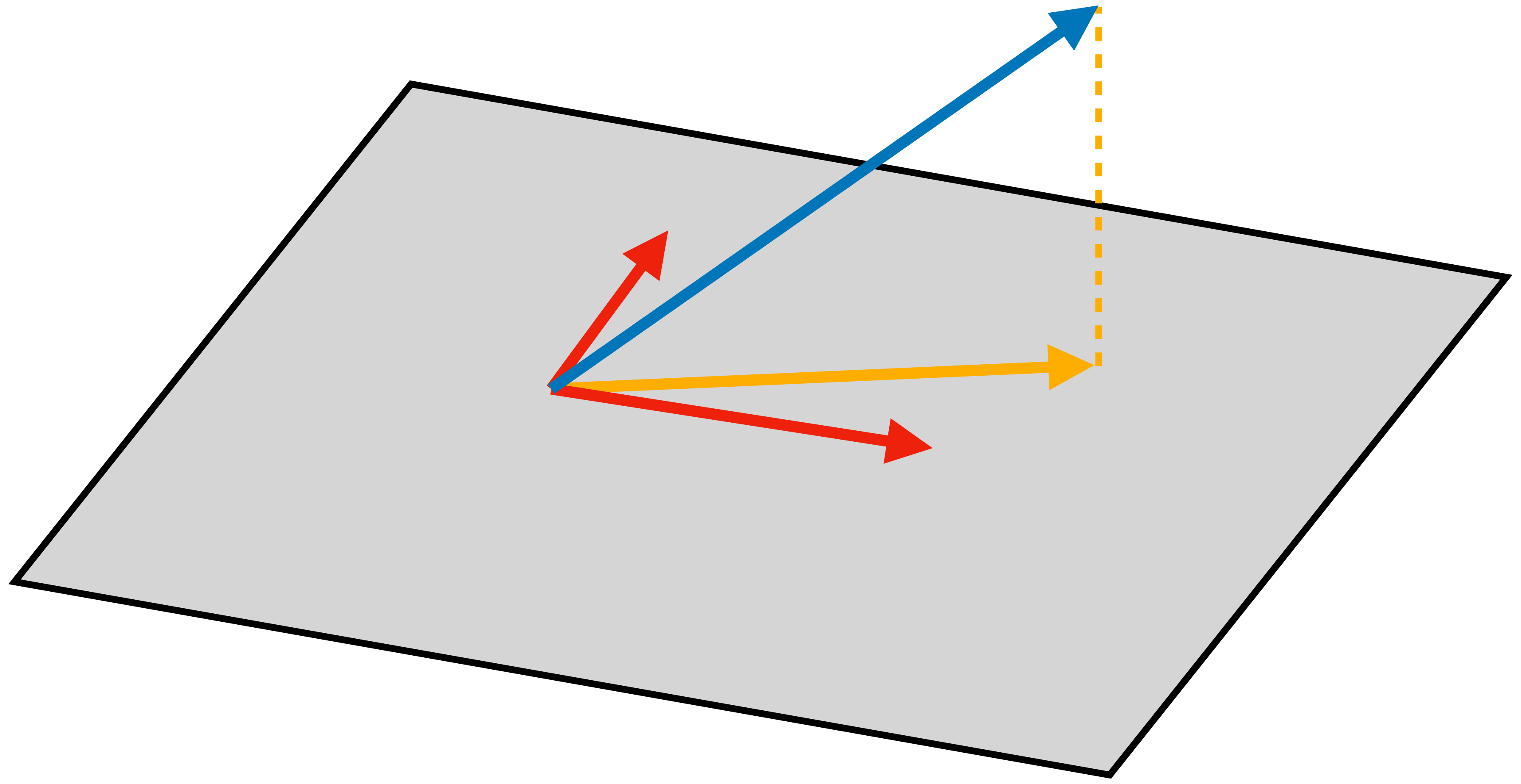
=



Recap: Fourier basis for $\mathbb{S}^1 \to \mathbb{R}$

Frequency 0	Frequency 1		Frequency 2		…
					
1	$\cos(\theta)$	$\sin(\theta)$	$\cos(2\theta)$	$\sin(2\theta)$	
1	x	y	$2x^2 - 1$	$2xy$	

Projection of vector onto orthogonal basis:

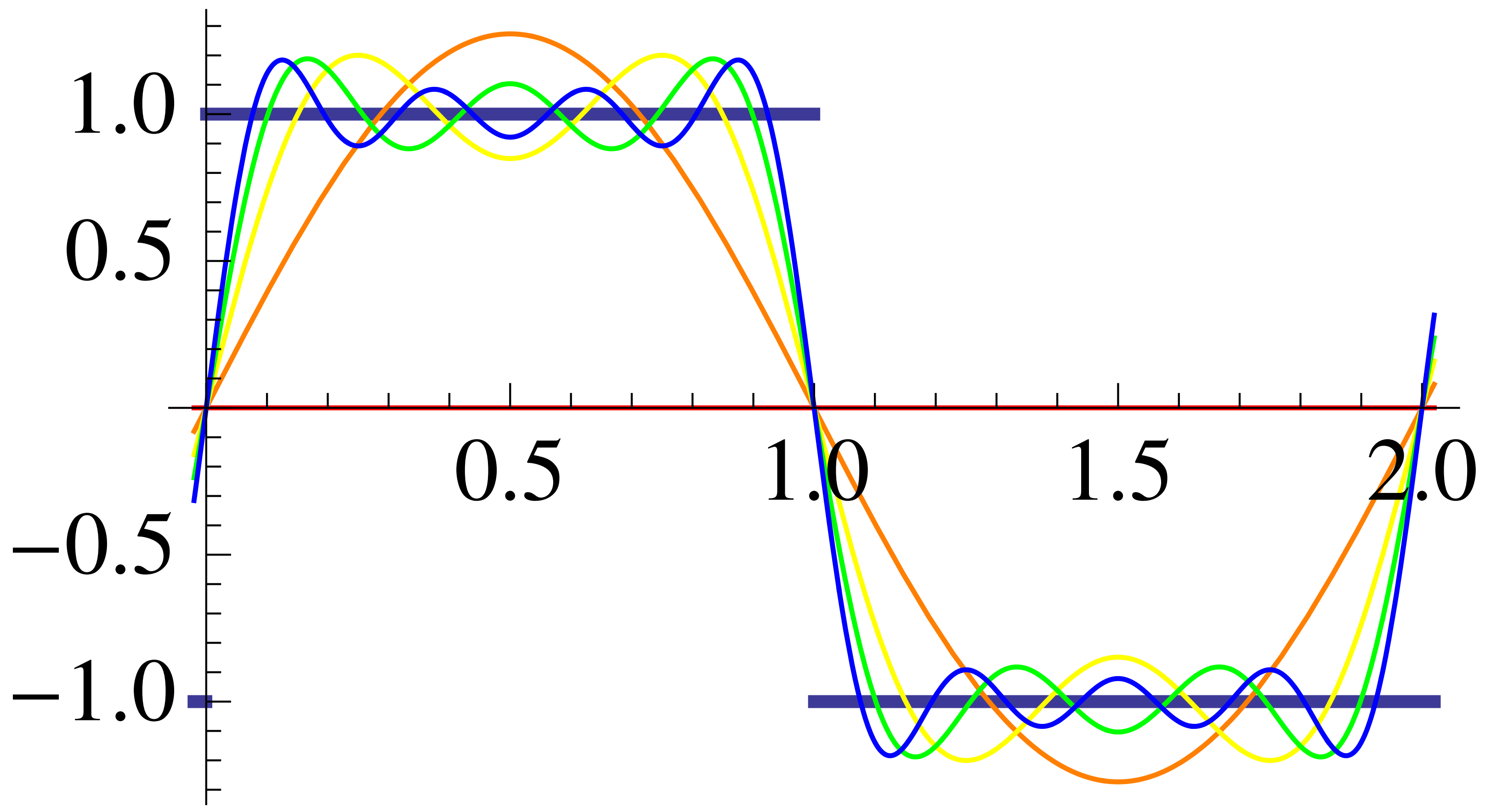


$$\mathbf{v} \approx c_1\,\mathbf{b}_1 + c_2\,\mathbf{b}_2 + \cdots$$
where $c_i = (\mathbf{v}\cdot\mathbf{b}_i)/(\mathbf{b}_i\cdot\mathbf{b}_i)$

Fourier basis is also orthogonal w.r.t. inner product $\langle f, g\rangle = \int_0^{2\pi} f(\theta)\,g(\theta)\,\mathrm{d}\theta$

Any periodic function can be approximated as

$$f(\theta) \approx c_1\,\phi_1(\theta) + c_2\,\phi_2(\theta) + \cdots$$
where $c_i = \dfrac{\langle f, \phi_i\rangle}{\langle \phi_i, \phi_i\rangle}$

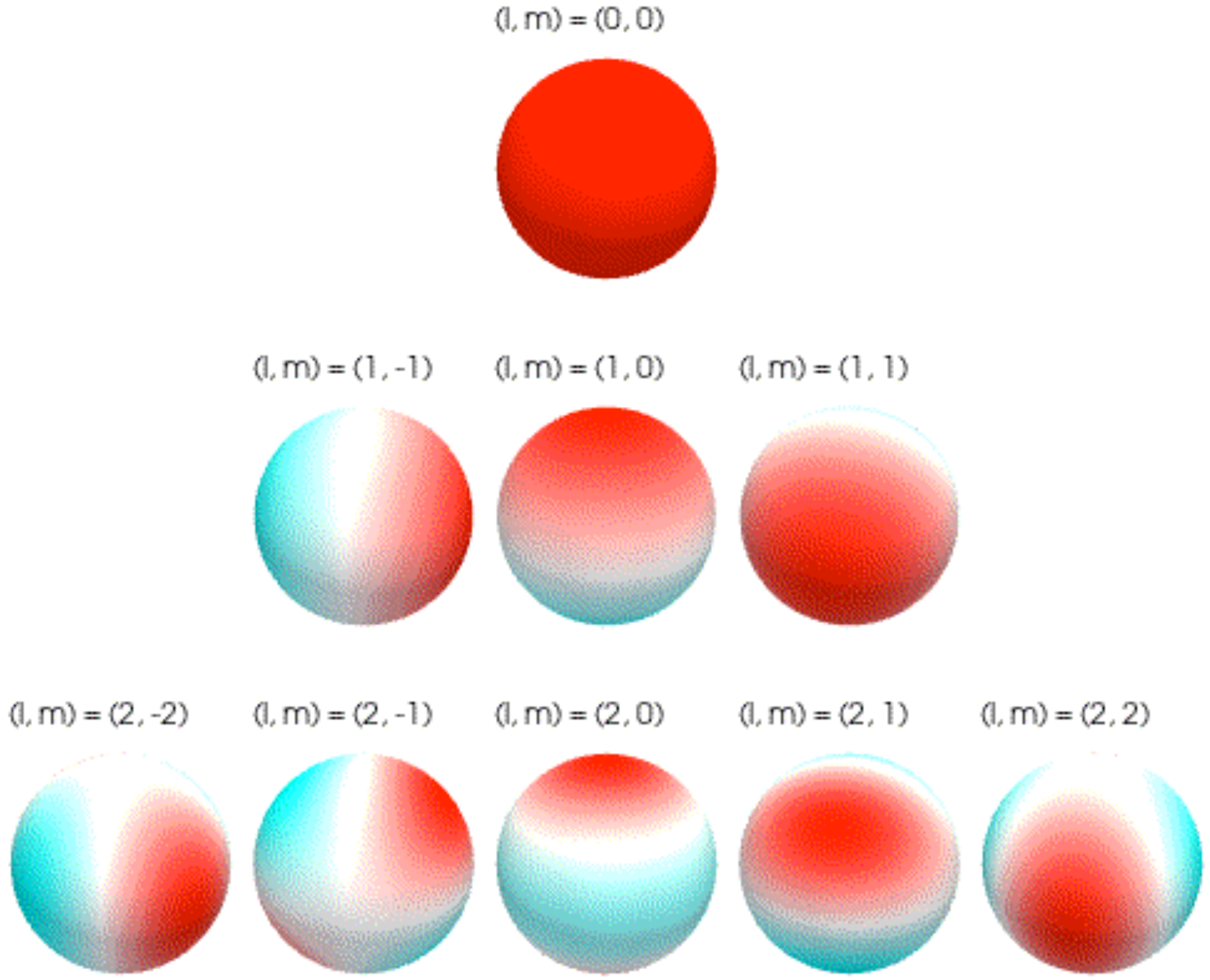


Recap: Spherical harmonics for $\mathbb{S}^2 \to \mathbb{R}$

- Order 0: 1
- Order 1: x, y, z
- Order 2: xy, yz, $3z^2 - 1$, zx, $x^2 - y^2$
- …

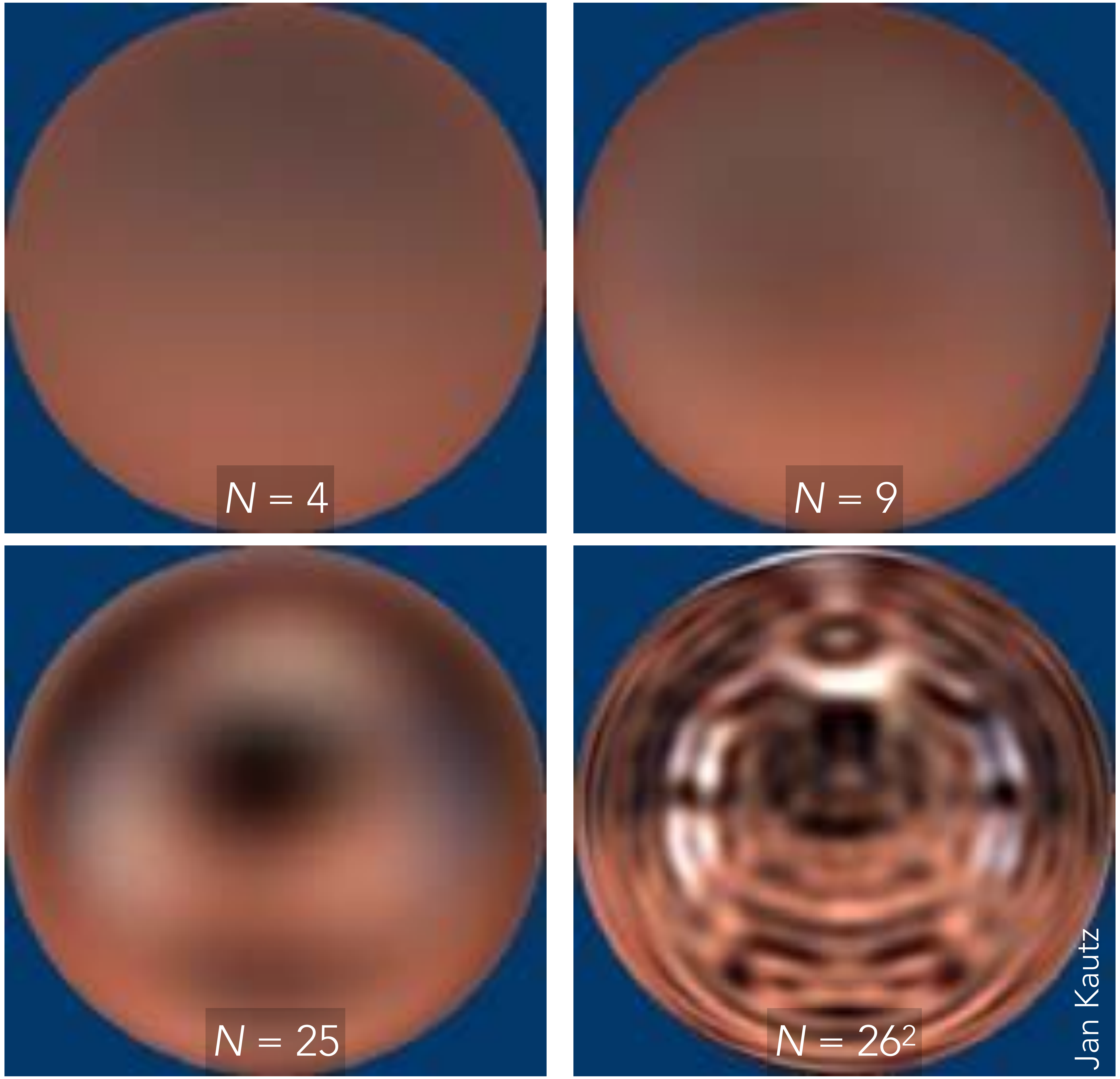
Also orthogonal w.r.t. $\int_{\mathbb{S}^2} f(\boldsymbol{\omega})\, g(\boldsymbol{\omega})\, d\boldsymbol{\omega}$

Thus any spherical function can be approximated as $f(\boldsymbol{\omega}) \approx \sum c_{\ell m} Y_{\ell m}(\boldsymbol{\omega})$



Approximating environment lighting with N spherical harmonics

$$L_{\mathrm{env}}(\boldsymbol{\omega}) = \sum_{i=1}^{N} \ell_i \, B_i(\boldsymbol{\omega})$$



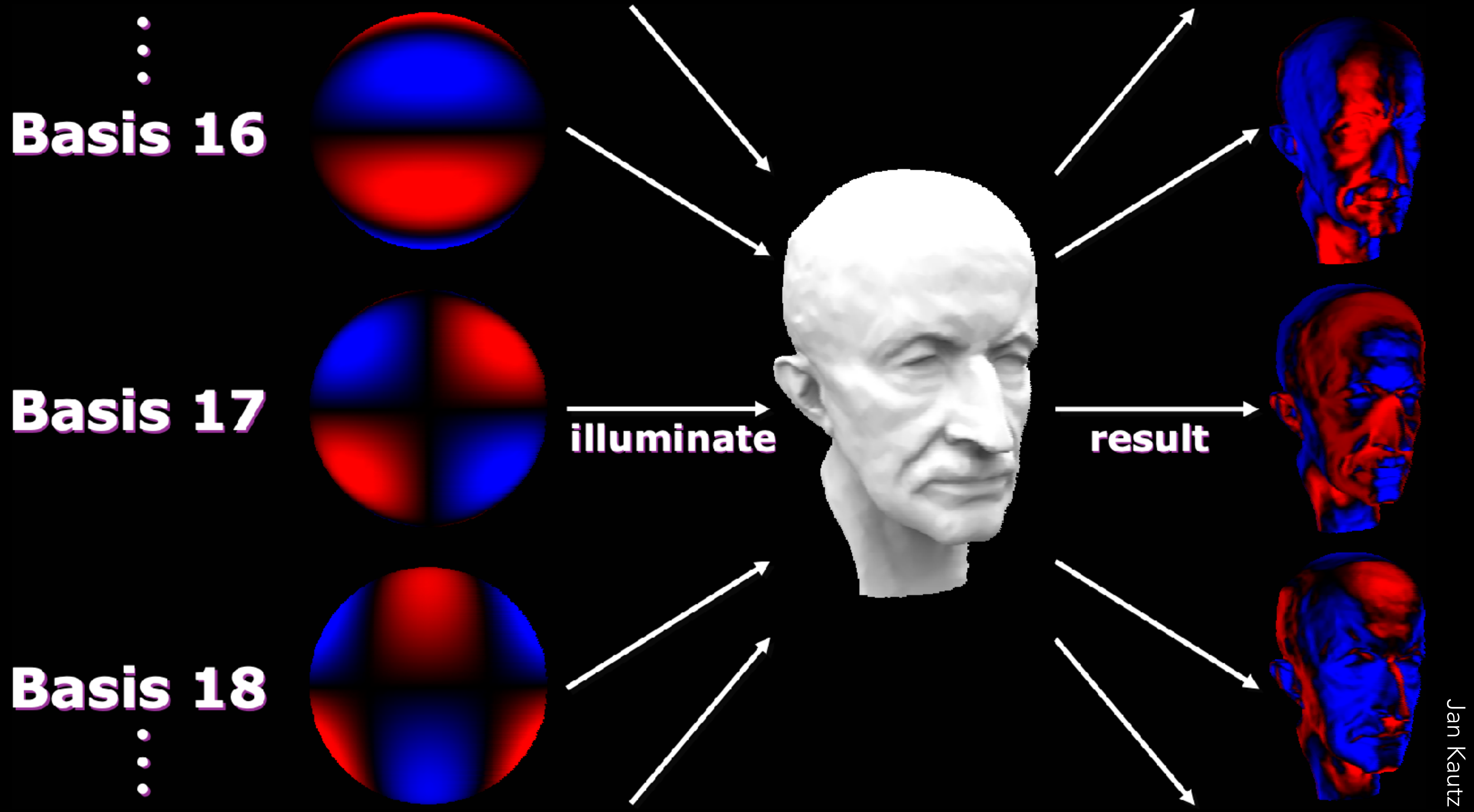
$$L_{\text{env}} = \sum_{i=1}^{N} \ell_i B_i$$

Outgoing radiance at any point is linear in L_{env}, so…

$$L_{\text{o}}[L_{\text{env}}] = \sum_{i=1}^{N} \ell_i L_o[B_i]$$

- Precompute radiance in scene for each lighting basis B_i
- Store as transport vector $\mathbf{t} = (L_o[B_1], L_o[B_2], \ldots)$ at each point
- At run time, just a dot product: $L_{\text{o}}[L_{\text{env}}] = \sum \ell_i t_i$

Can run an arbitrarily complicated global illumination computation here!


Basis 16
Basis 17
Basis 18
illuminate
result
Jan Kautz

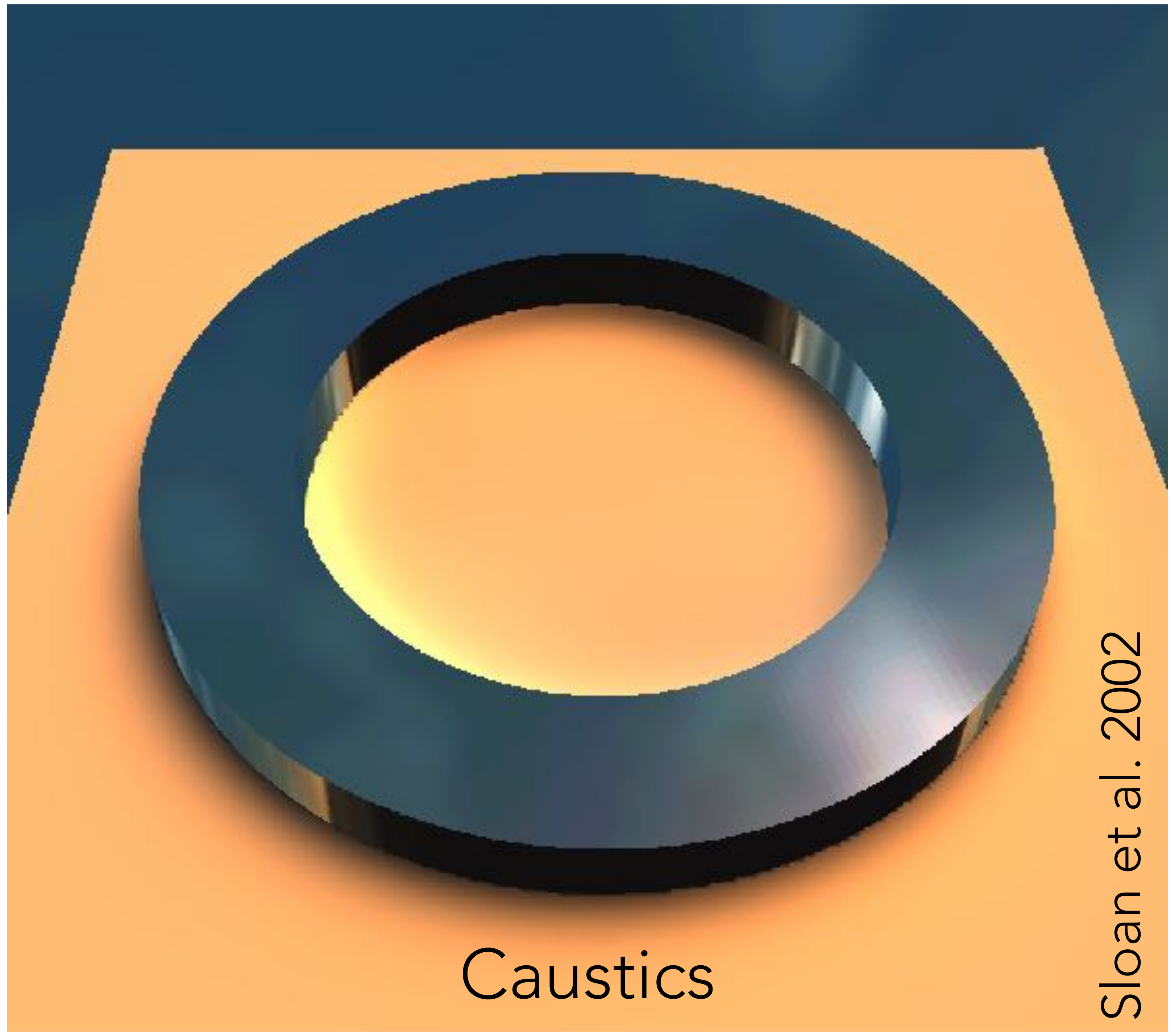

Unshadowed
(irradiance map)

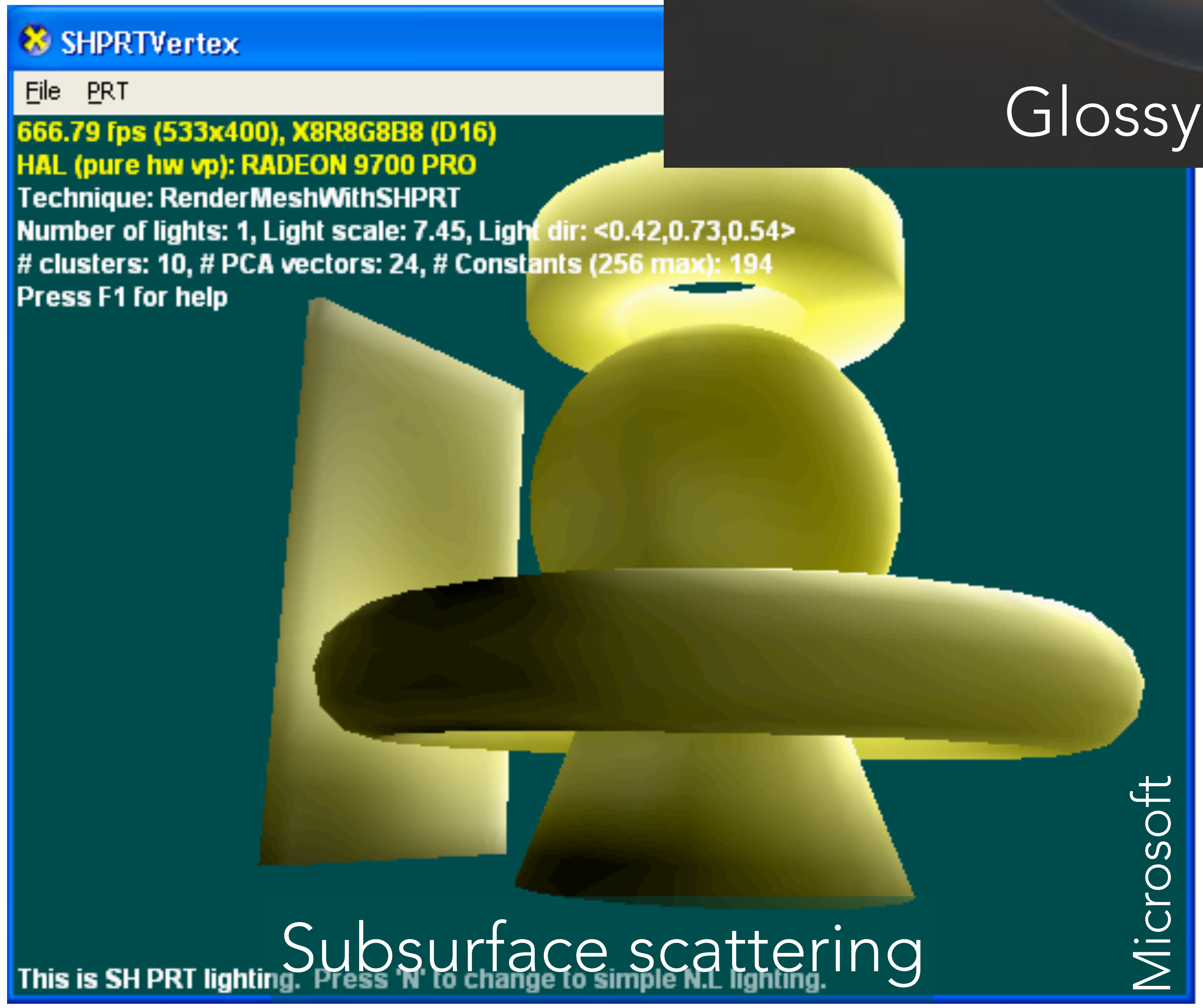

Shadowed
(PRT)


Unshadowed
Shadowed
Interreflected

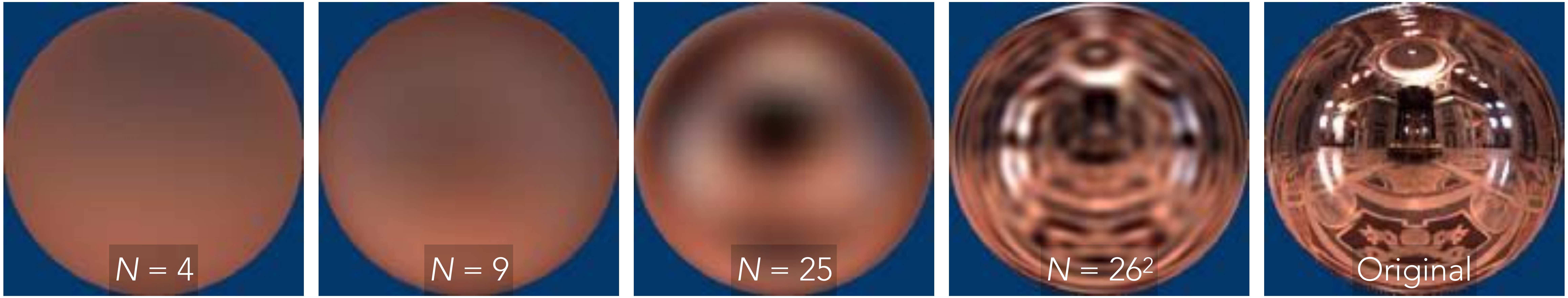
Can be extended to non-diffuse surfaces and arbitrary light transport mechanisms!

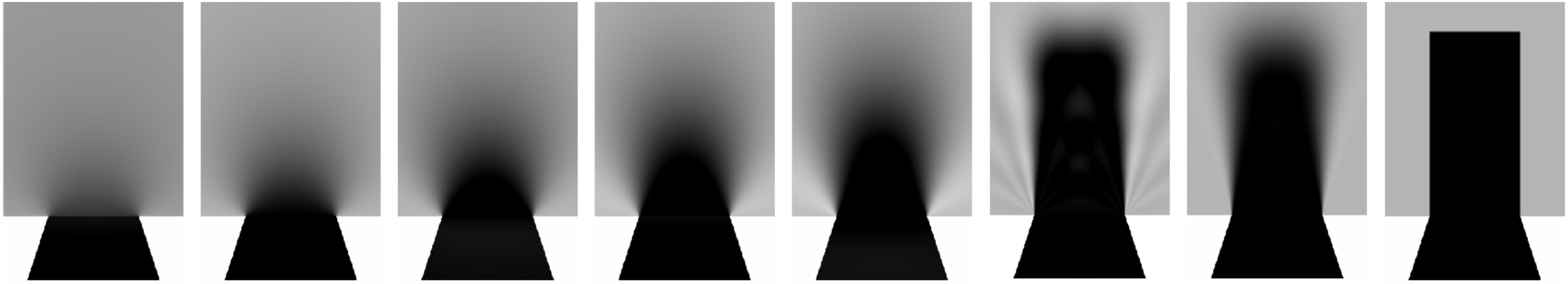
For non-diffuse surfaces, store a transport matrix $\mathbf{T} = [t_{ij}]$ at each point: envmap SH → outgoing SH





Limitation: spherical harmonics are only efficient for low-frequency illumination





(a) linear (n=2) (b) quadratic (n=3) (c) cubic (n=4) (d) quartic* (n=5) (e) quintic (n=6) (f) high-order (n=26) (g) windowed (n=26) (h) ray traced

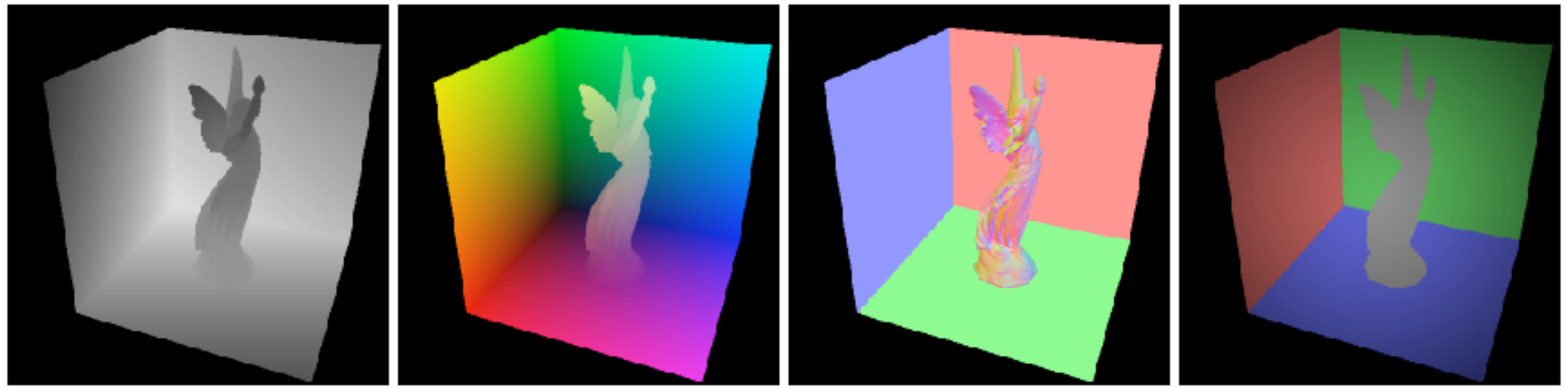
Low-frequency lighting
(spherical harmonics)

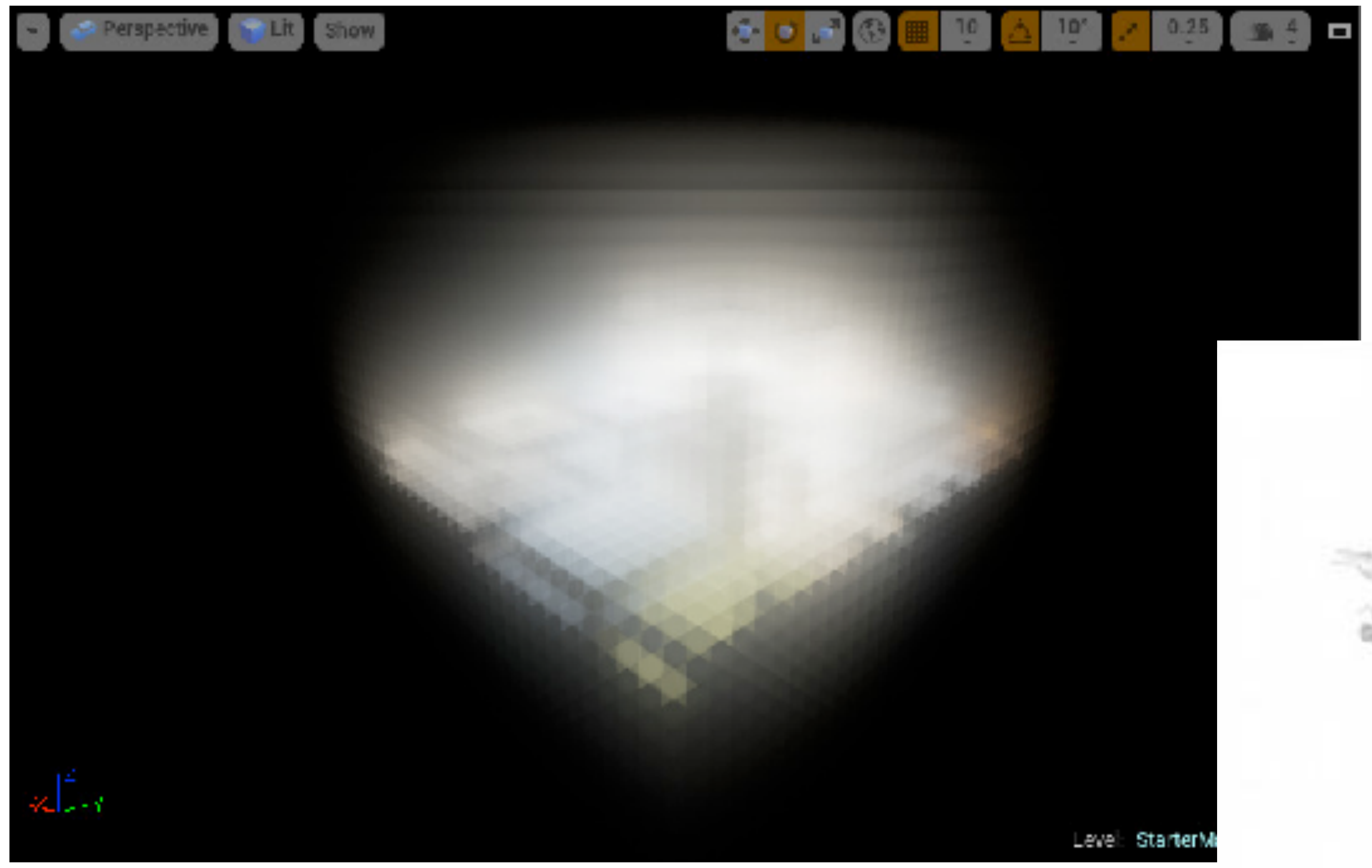
All-frequency lighting
(wavelets)

Ng et al. 2003

Lots more in real-time rendering!

- Reflective shadow maps
- Light propagation volumes
- Screen-space global illumination
- Real-time ray tracing





Real-time ray tracing

Hardware support in recent graphics cards. But what does it actually do?

- 1 sample per pixel
- Only 1 secondary ray (1-bounce indirect illumination)
- Lots of clever denoising!
  - Spatial (using nearby samples)
  - Temporal (using samples from previous frames)

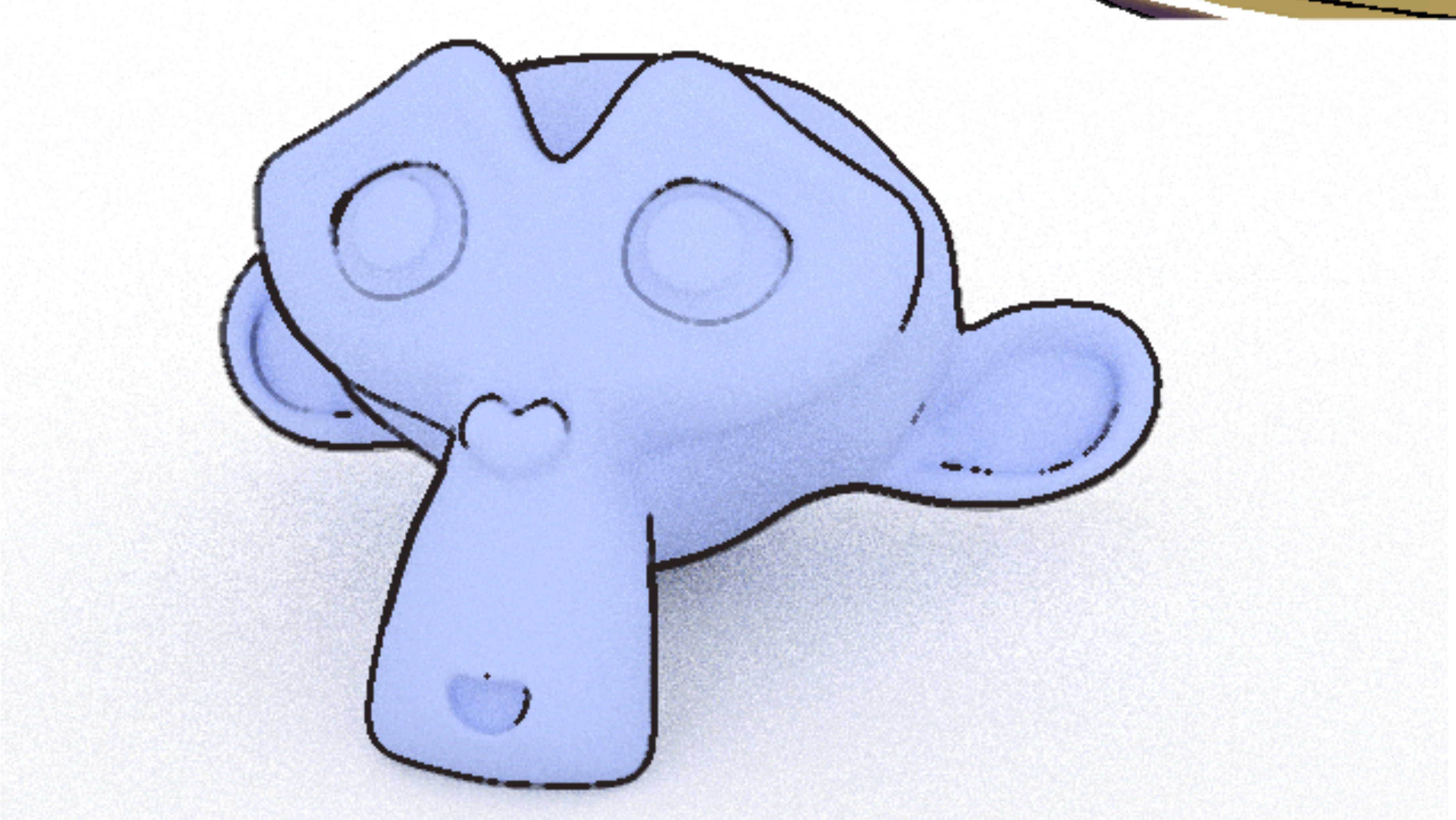
via Lingqi Yan

via Lingqi Yan

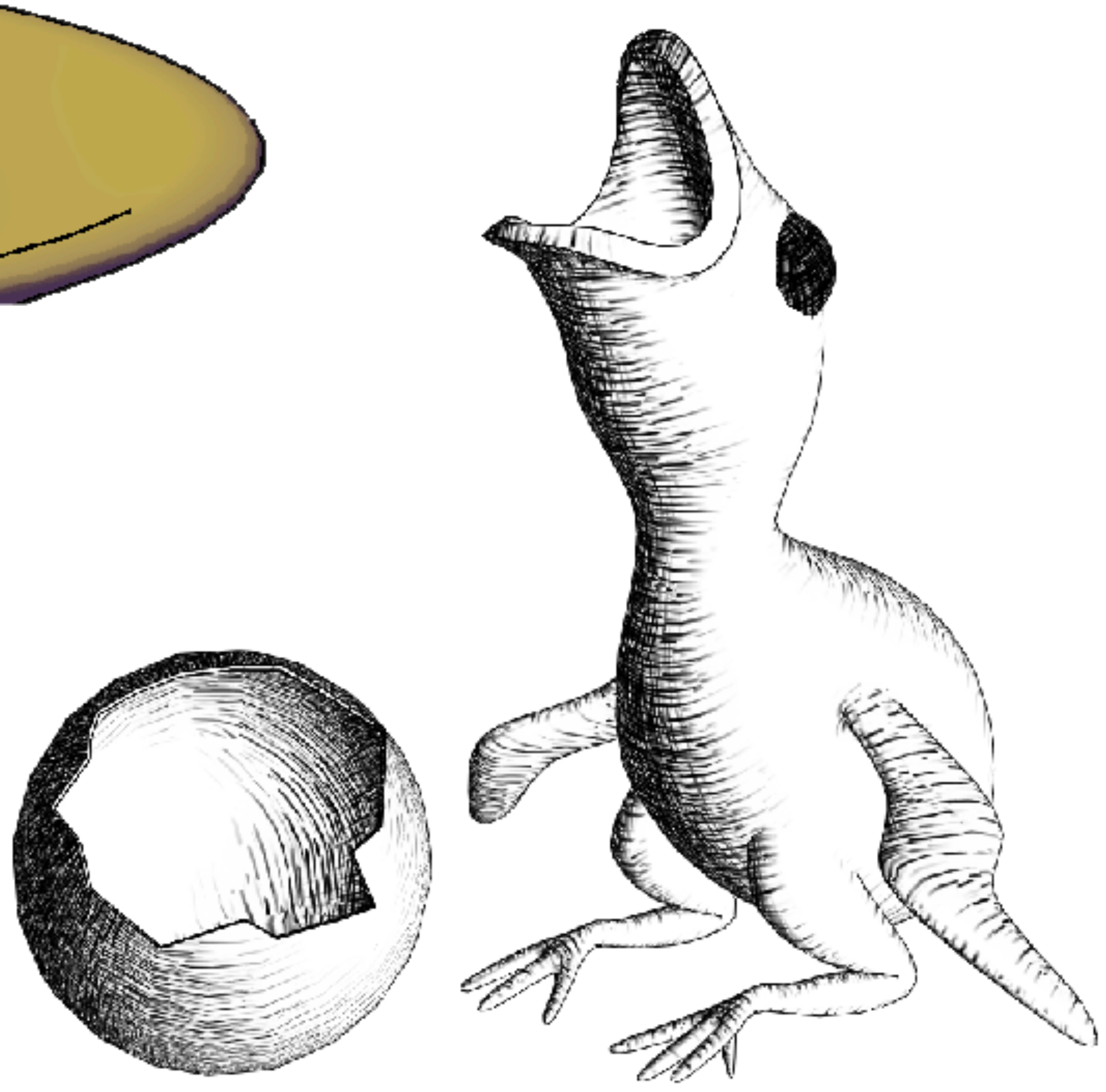
Non-photorealistic rendering

Non-photorealistic rendering

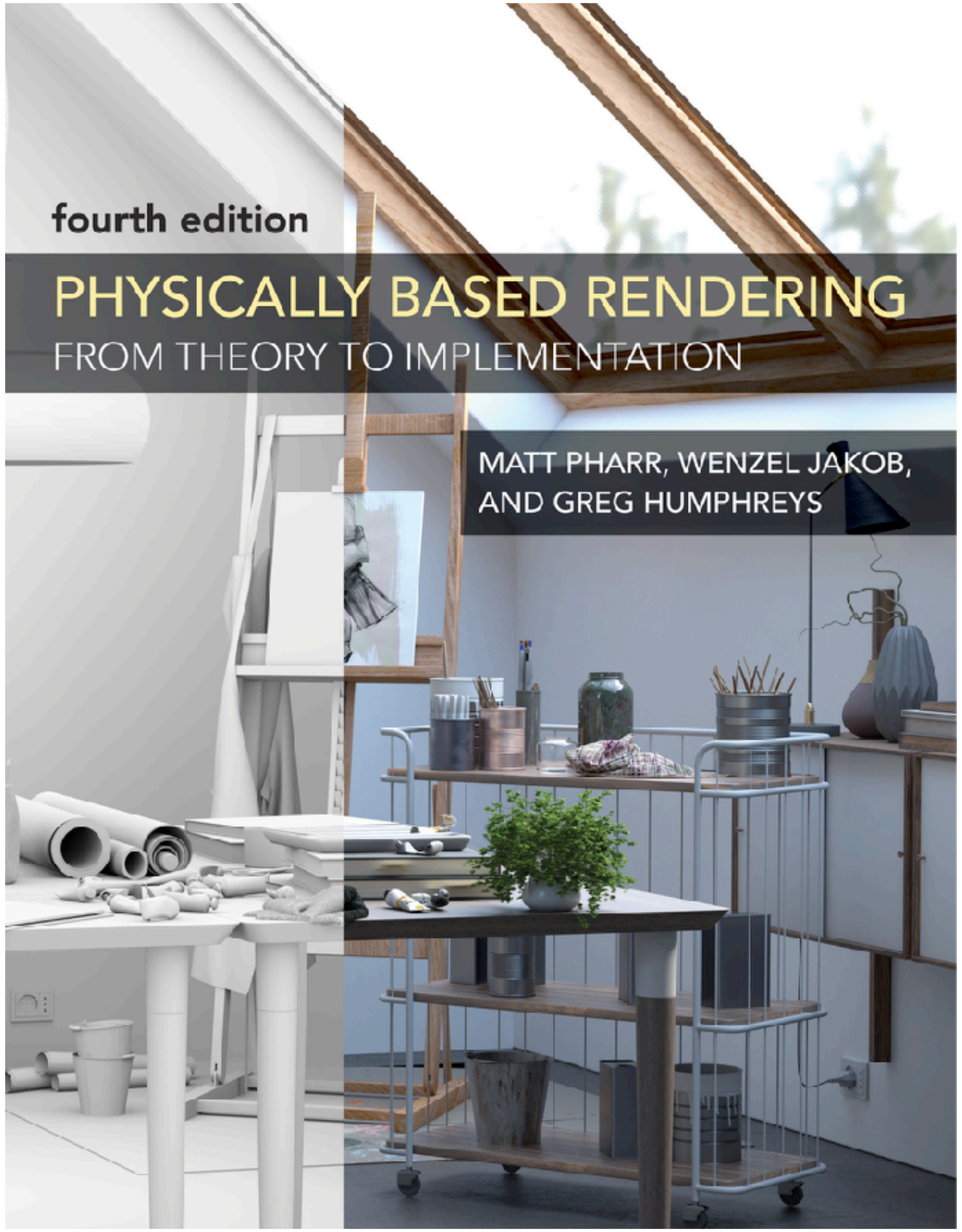
Toon shading

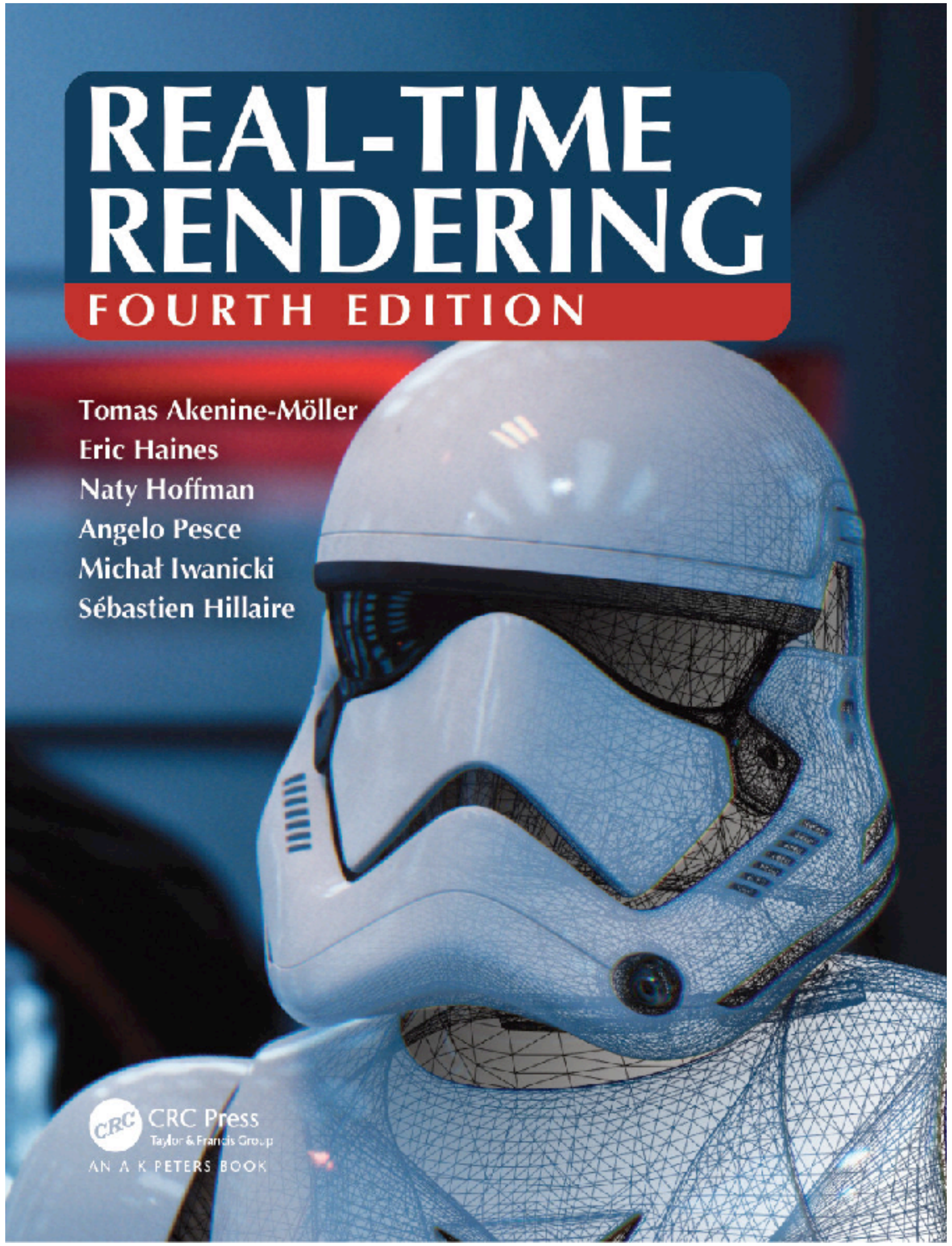


Contour rendering



Stroke rendering


fourth edition
PHYSICALLY BASED RENDERING
FROM THEORY TO IMPLEMENTATION
MATT PHARR, WENZEL JAKOB,
AND GREG HUMPHREYS


REAL-TIME
RENDERING
FOURTH EDITION
Tomas Akenine-Möller
Eric Haines
Naty Hoffman
Angelo Pesce
Michał Iwanicki
Sébastien Hillaire
CRC Press
Taylor & Francis Group
AN A K PETERS BOOK