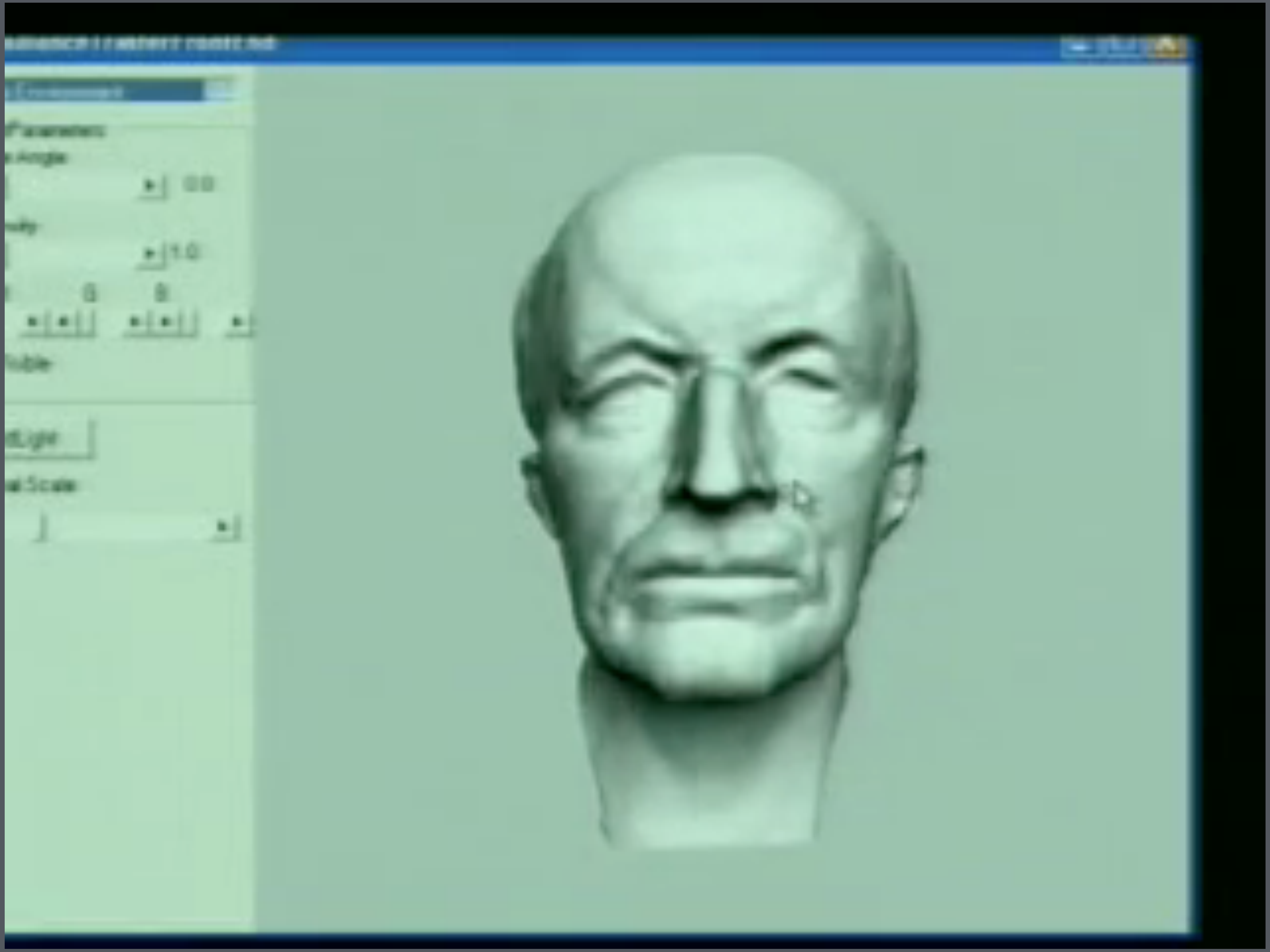
17 Spherical Harmonic Lighting

Steve Marschner

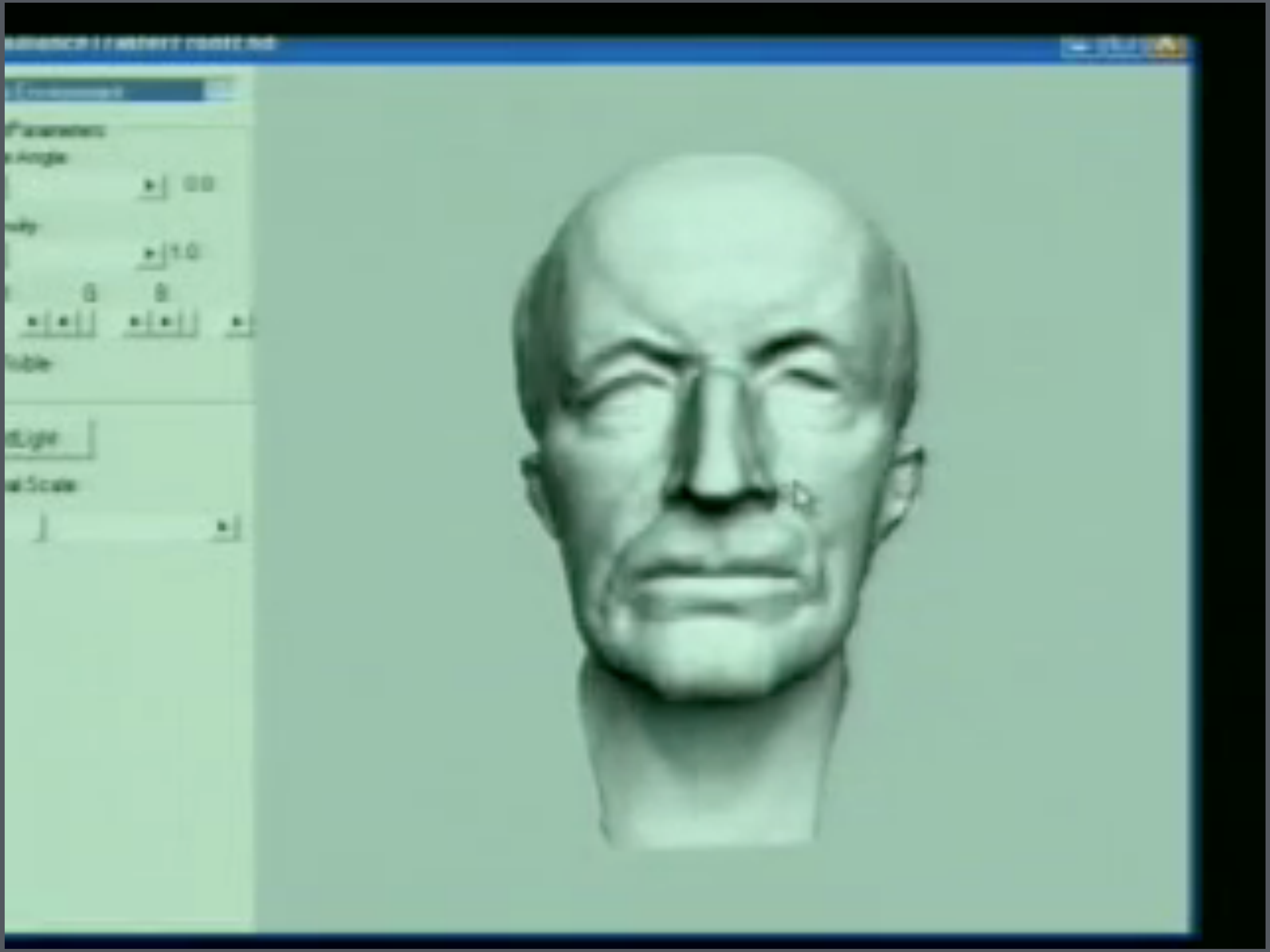
CS5625 Spring 2019

Precomputed Radiance Transfer



Sloan et al. “Precomputed Radiance Transfer” http://dx.doi.org/10.1145/566654.566612

Precomputed Radiance Transfer



Sloan et al. “Precomputed Radiance Transfer” http://dx.doi.org/10.1145/566654.566612

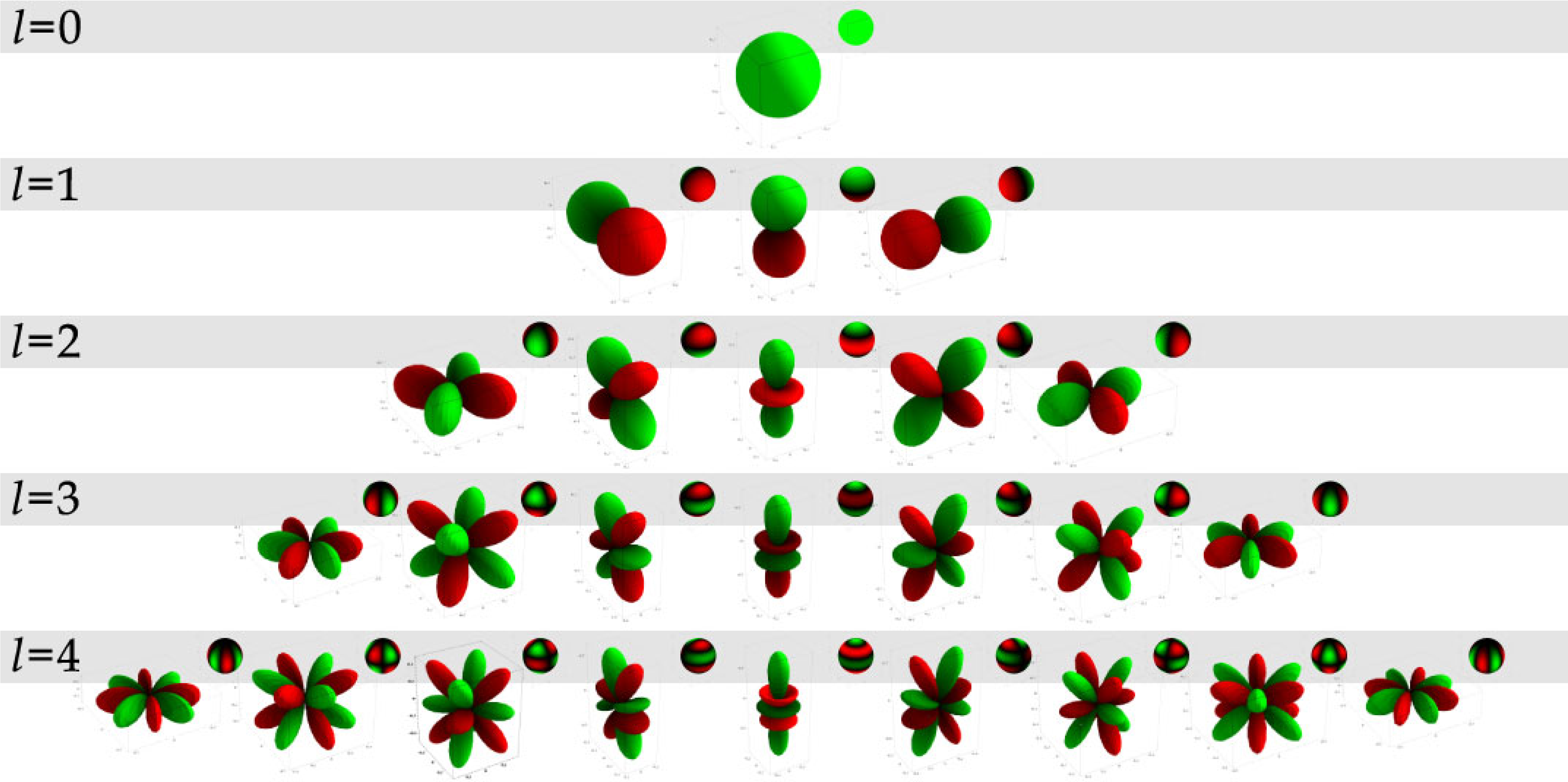


Figure 5. The first 5 SH bands plotted as unsigned spherical functions by distance from the origin and by colour on a unit sphere. Green (light gray) are positive values and red (dark gray) are negative.

Robin Green http://www.cs.columbia.edu/~cs4162/slides/spherical-harmonic-lighting.pdf

Where s are simply locations on the unit sphere. The basis functions are defined as

$$Y_l^m(\theta,\varphi) = K_l^m e^{im\varphi} P_l^{|m|}(\cos\theta), l \in \mathbf{N}, -l \le m \le l$$

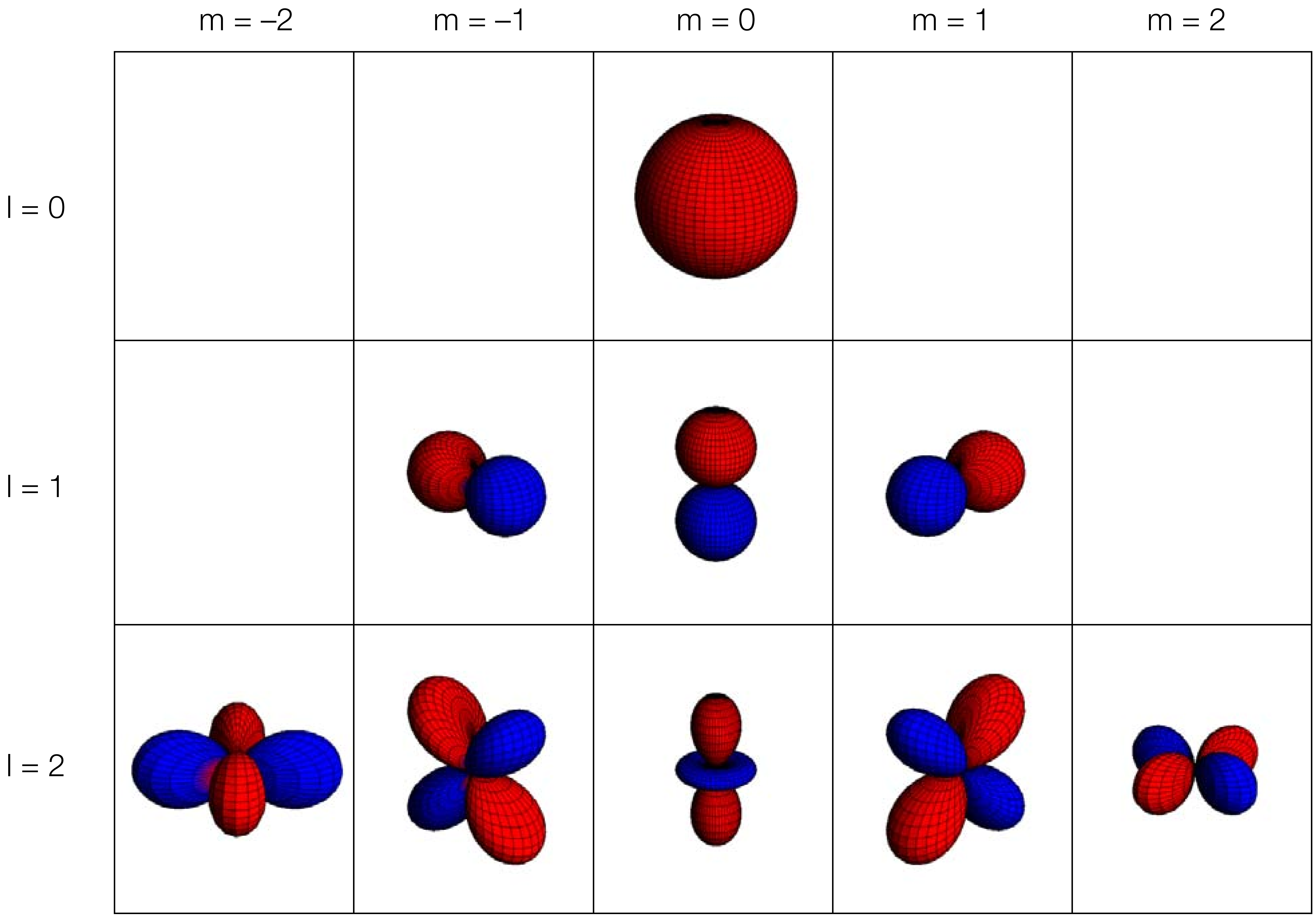
Where P_l^m are the associated Legendre polynomials and K_l^m are the normalization constants

$$K_l^m = \sqrt{\frac{(2l+1)(l-|m|)!}{4\pi(l+|m|)!}}$$

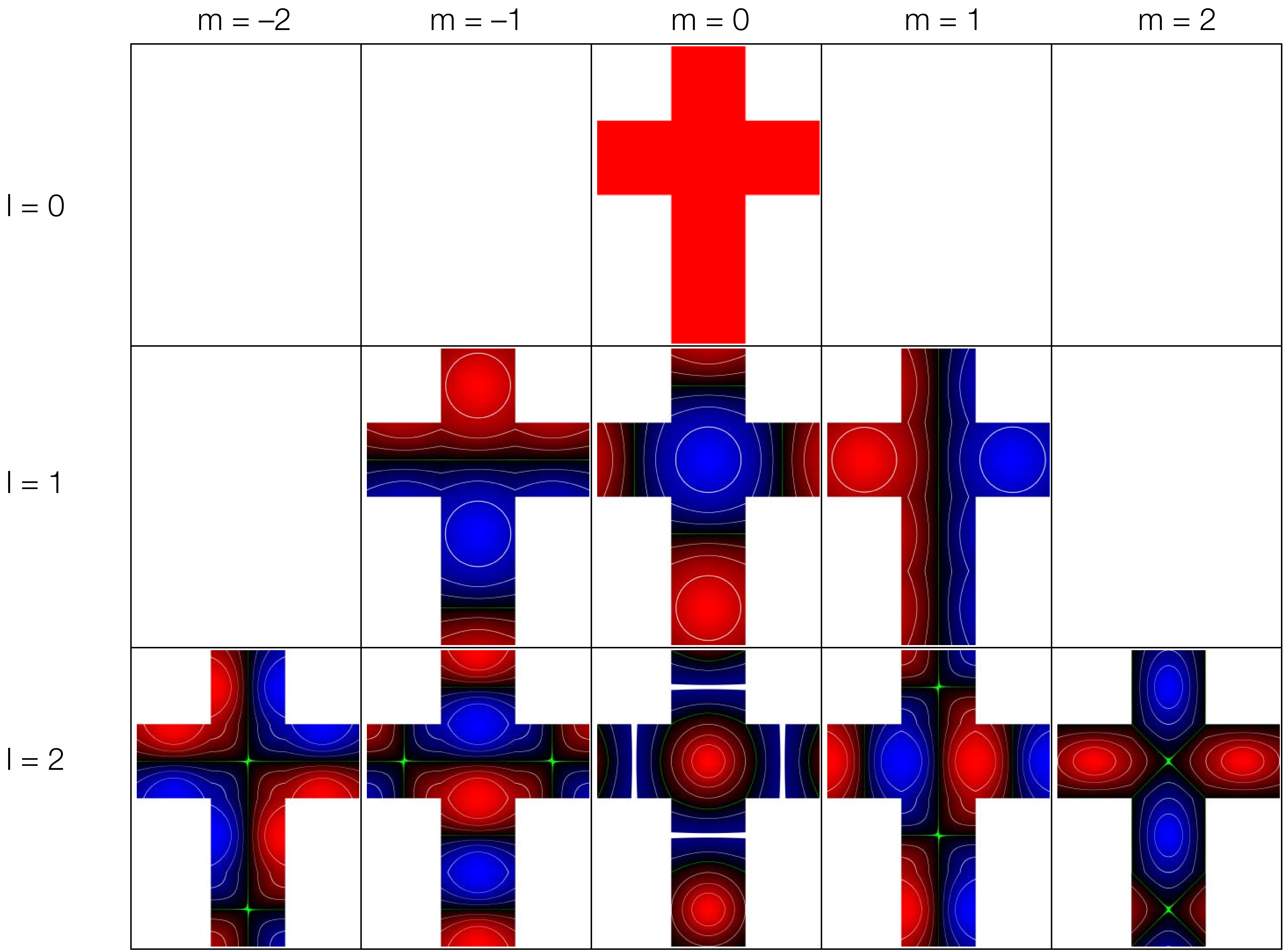
The above definition is for the complex form (most commonly used in the non-graphics literature), a real valued basis is given by the transformation

$$y_l^m = \begin{cases} \sqrt{2}\mathrm{Re}(Y_l^m) & m > 0 \\ \sqrt{2}\mathrm{Im}(Y_l^m) & m < 0 \\ Y_l^0 & m = 0 \end{cases} = \begin{cases} \sqrt{2}K_l^m \cos m\varphi \; P_l^m(\cos\theta) & m > 0 \\ \sqrt{2}K_l^m \sin|m|\varphi \; P_l^{|m|}(\cos\theta) & m < 0 \\ K_l^0 P_l^0(\cos\theta) & m = 0 \end{cases}$$

The index *l* represents the "band". Each band is equivalent to polynomials of that degree (so zero is just a constant function, 1 is linear, etc.) and there are 2*l*+1 functions in a given band.



Peter-Pike Sloan, “Stupid Spherical Harmonics Tricks” http://www.ppsloan.org/publications/



	+Y	
-X	-Z	+x
	-Y	
	+Z	

Peter-Pike Sloan, "Stupid Spherical Harmonics Tricks" http://www.ppsloan.org/publications/

Projection and Reconstruction Because the SH basis is orthonormal the least squares projection of a scalar function *f* defined over **S** is done by simply integrating the function you want to project, $f(s)$, against the basis functions (proof in Appendix A6 Least Squares Projection)

$$f_l^m = \int f(s) y_l^m(s)\, ds$$

These coefficients can be used to reconstruct an approximation of the function *f*

$$\tilde{f}(s) = \sum_{l=0}^{n} \sum_{m=-l}^{l} f_l^m y_l^m(s)$$

What is special about spherical harmonics (for lighting)?

They are polynomials

- which makes them easy to evaluate
- nice recurrence is available (see Sloan)

They are orthogonal

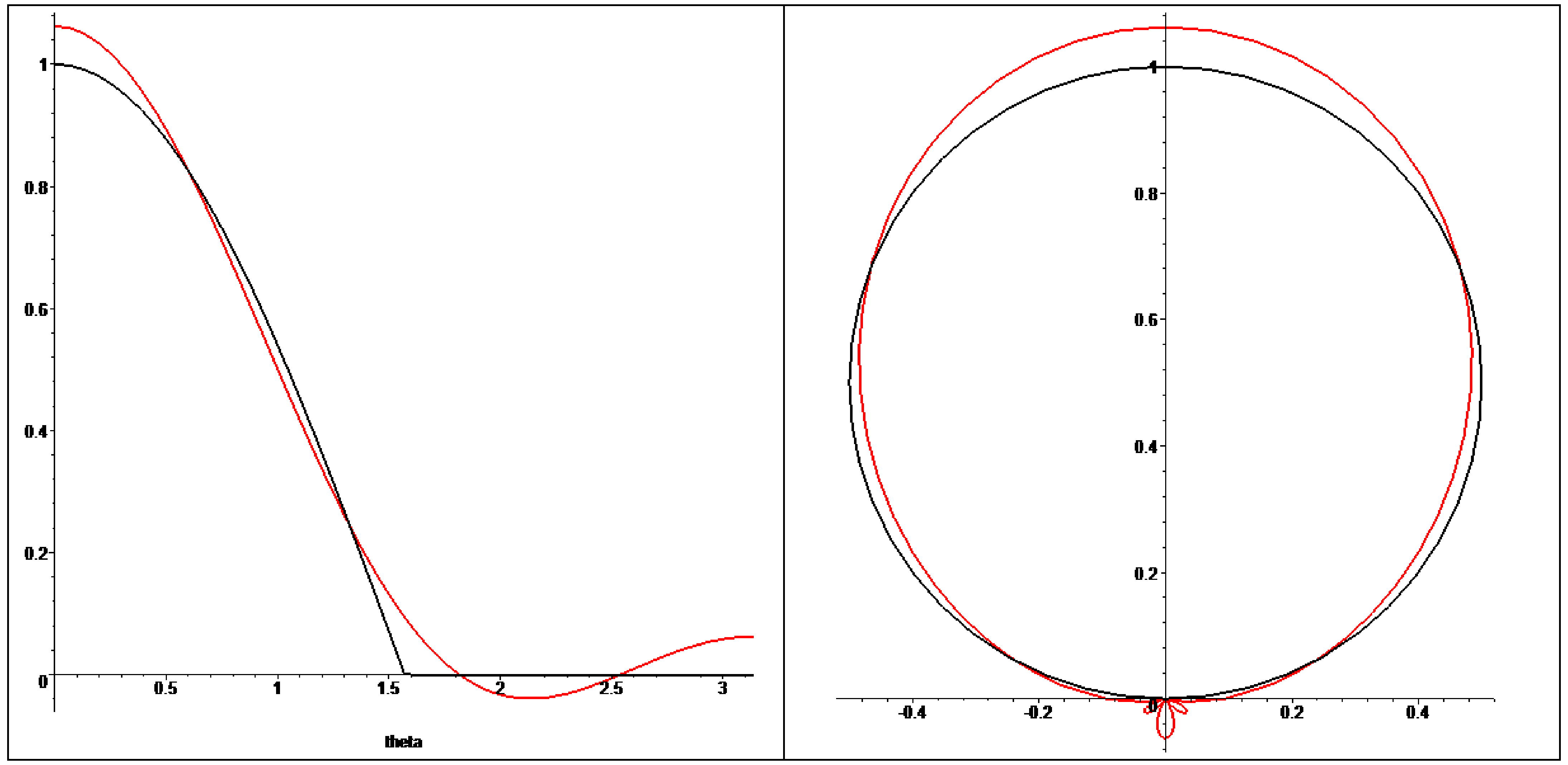
- which makes it easy to compute coefficients
- also makes it easy to compute dot products of functions

They are closed under rotation

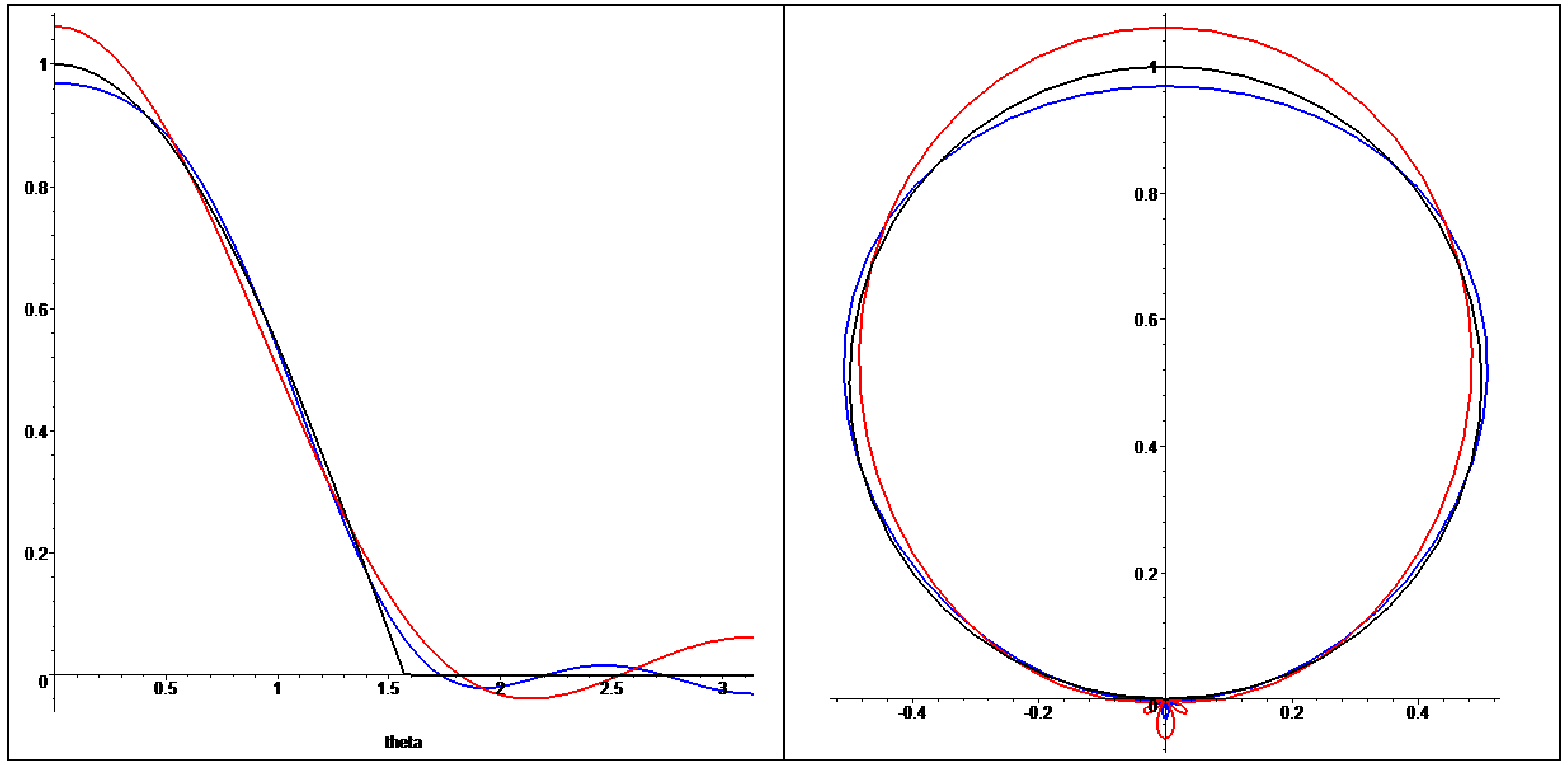
- rotation only rearranges coefficients within bands

They have the convolution–multiplication property

- convolution of functions can be computed by multiplying the coefficients band can be skipped.) Below are images of the clamped cosine kernel and the order 3 SH approximation, the red curve is the SH approximation, the figure on the left is a plot as a function of theta, on the right a polar plot scaled by the absolute value of the function:



Below are plots that also includes the Order 5 projection (blue):

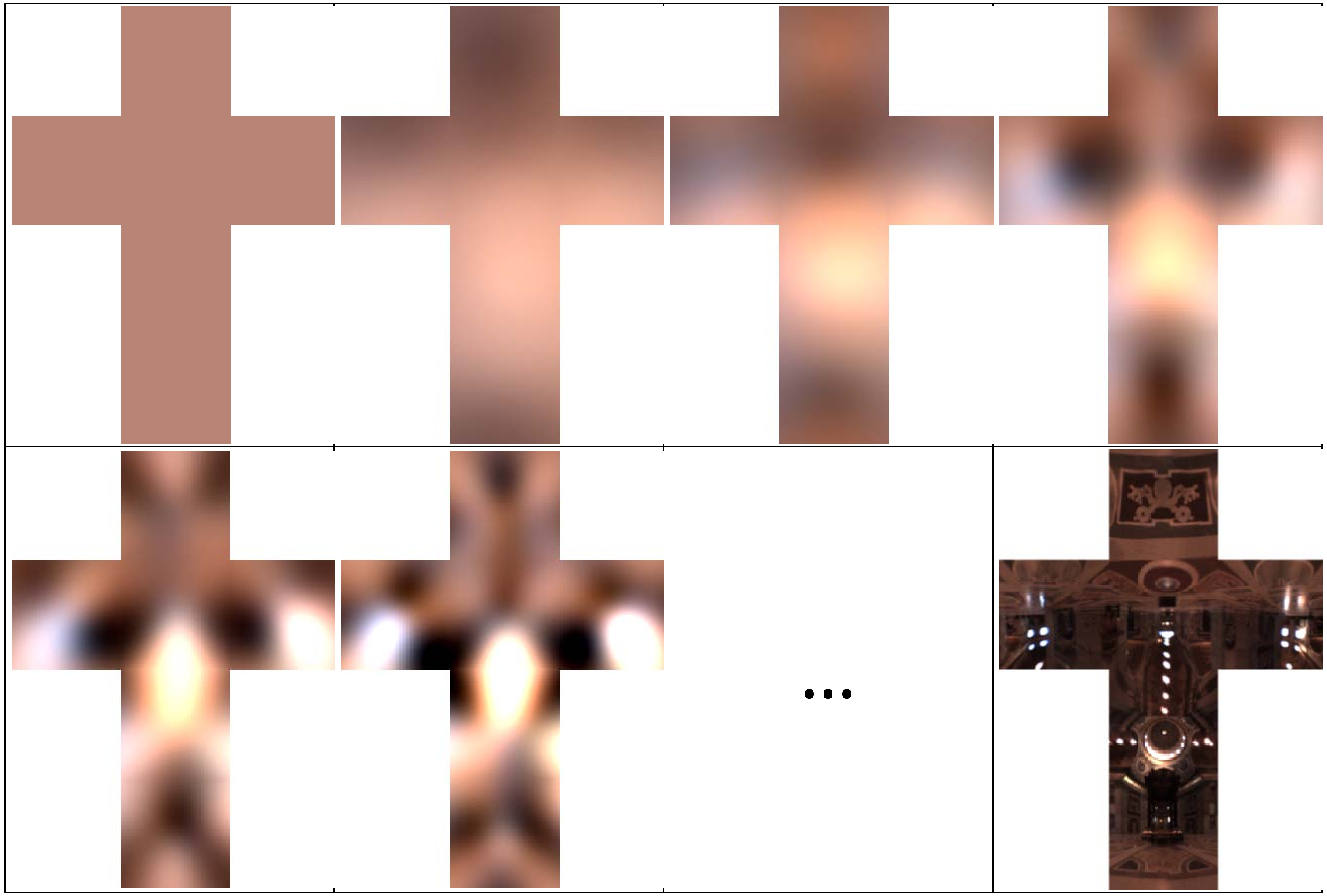


Projection from Cube Maps

To project from a cube map you simply need to integrate the SH basis functions against the cube map. This can be done numerically by evaluate the SH basis functions in the direction of each texel center, weight it by the differential solid angle for that texel and normalize the results. In pseudo-code:

```
float f[],s[];
float fWtSum=0;
Foreach(cube map face)
      Foreach(texel)
            float fTmp = 1 + u^2+v^2;
            float fWt = 4/(sqrt(fTmp)*fTmp);
            EvalSHBasis(texel,s);
            f += t(texel)*fWt*s; // vector
            fWtSum += fWt;
f *= 4*Pi/fWtSum; // area of sphere
```

Below are images of the reconstruction of a HDR light probe into order 1 to 6 Spherical Harmonics. The final image is the light probe that was projected.



Peter-Pike Sloan, “Stupid Spherical Harmonics Tricks” http://www.ppsloan.org/publications/

(a) diffuse, unshadowed (b) diffuse, interreflected (c) glossy, unshadowed (d) glossy, interreflected

Figure 11: Diffuse and Glossy Self-transfer. Unshadowed transfer (a,c) includes no global transport effects. Interreflected transfer (b,d) includes both shadows and interreflections.

Sloan et al., “Precomputed Radiance Transfer…” http://dx.doi.org/10.1145/566654.566612

(a) no interreflections (shadowed transfer) (b) 1-bounce interreflections (c) 2-bounce interreflections

Figure 12: **Interreflections in Self-Transfer.** Top row shows diffuse transfer; bottom row shows glossy transfer. Note the reflections under the knob on the lid and from the spout onto the body. Run-time performance is insensitive to interreflections; only the preprocessed simulation must include additional bounces. Further bounces after the first or second typically provide only subtle change.

Sloan et al., “Precomputed Radiance Transfer…” http://dx.doi.org/10.1145/566654.566612