Real-Time High Quality Rendering

GAMES202, Lingqi Yan, UC Santa Barbara

Lecture 6:
Real-Time Environment Mapping
(Precomputed Radiance Transfer)




Announcements

- No class next week (traveling)
- Some changes to the pace
  - One lecture per week
  - Only on Saturdays (GMT+8)
  - Lecture 7 will be on Apr 17
- Some changes to the order of lectures
  - Will talk about Precomputed Radiance Transfer (PRT) first

Last Lecture

- Distance field soft shadows
- Shading from environment lighting
  - The split sum approximation

Today

- Finishing up
  - Shadow from environment lighting
- Background knowledge
  - Frequency and filtering
  - Basis functions
- Real-time environment lighting (& global illumination)
  - Spherical Harmonics (SH)
  - Prefiltered env. lighting
  - Precomputed Radiance Transfer (PRT)

Shadow from Environment Lighting

- In general, very difficult for real-time rendering
- Different perspectives of view
  - As a many-light problem:
    Cost of SM is linearly to #light
  - As a sampling problem:
    Visibility term V can be arbitrarily complex
    And V cannot be easily separated from the environment

Shadow from Environment Lighting

- Industrial solution
  - Generate one (or a little bit more) shadows from the brightest light sources
- Related research
  - Imperfect shadow maps
  - Light cuts
  - RTRT (might be the ultimate solution)
  - Precomputed radiance transfer

Questions?

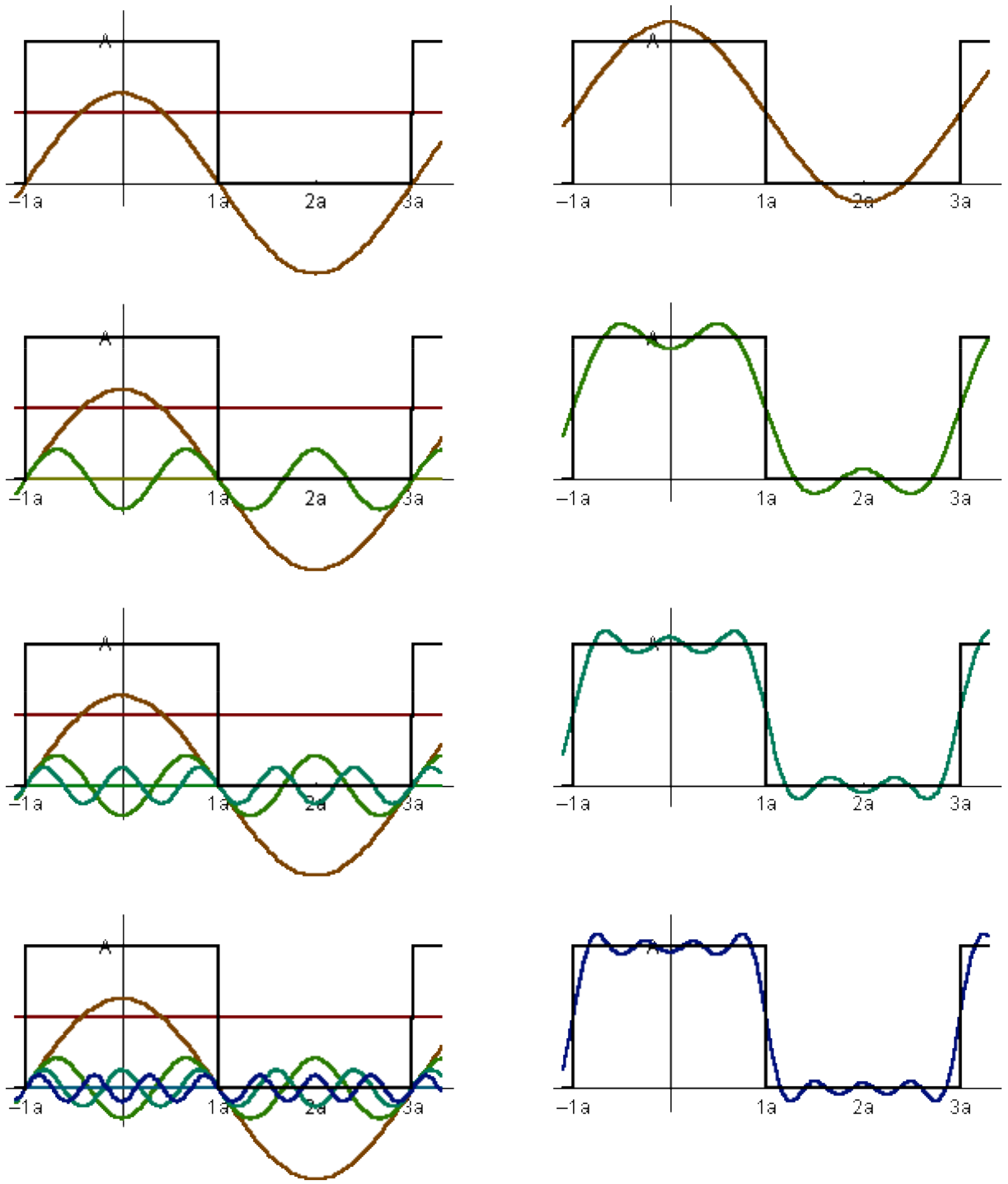
Today

- Finishing up
  - Shadow from environment lighting
- Background knowledge
  - Frequency and filtering
  - Basis functions
- Real-time environment lighting (& global illumination)
  - Spherical Harmonics (SH)
  - Prefiltered env. lighting
  - Precomputed Radiance Transfer (PRT)

Fourier Transform

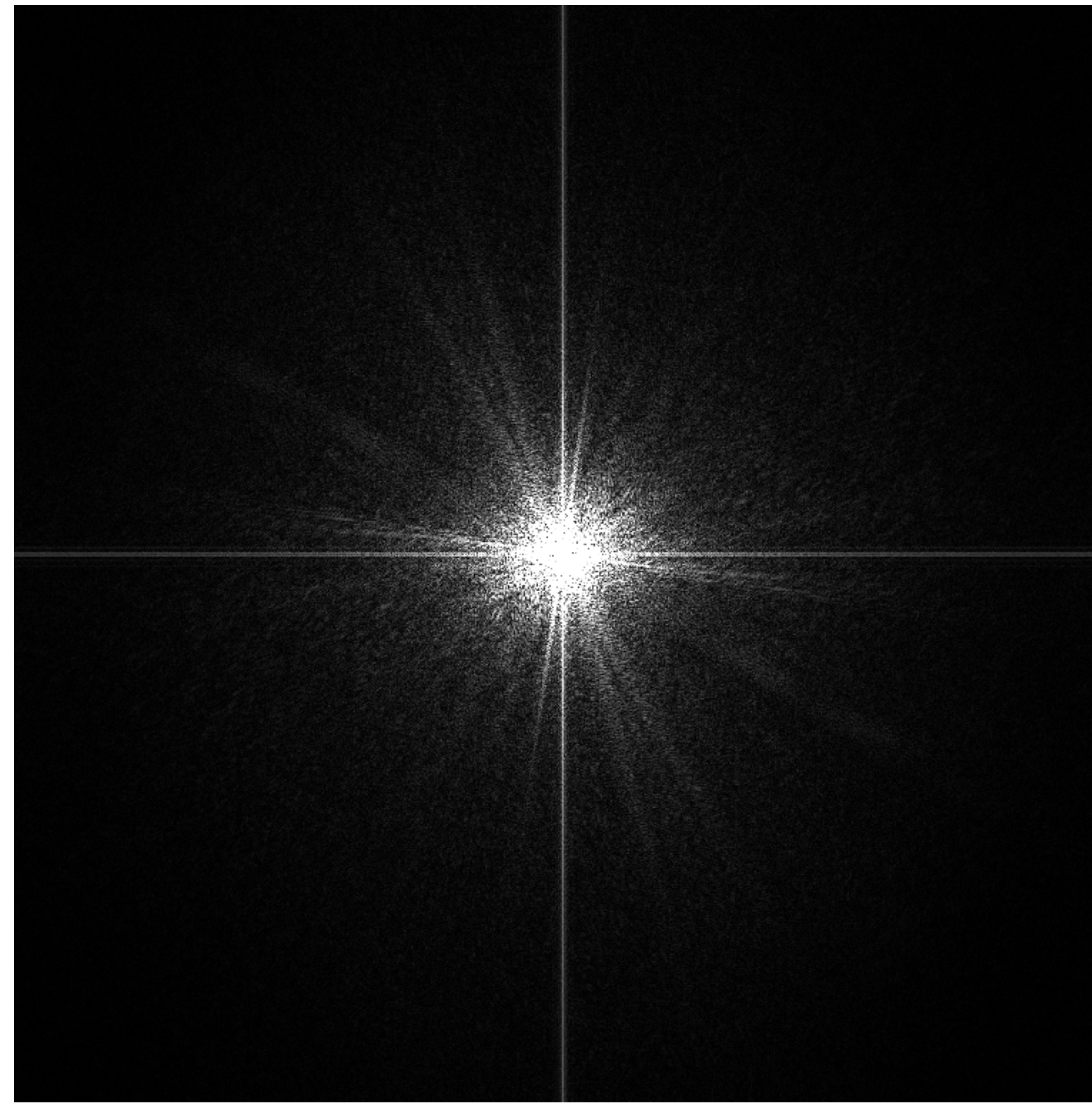
Represent a function as a weighted sum of sines and cosines

Joseph Fourier 1768 - 1830

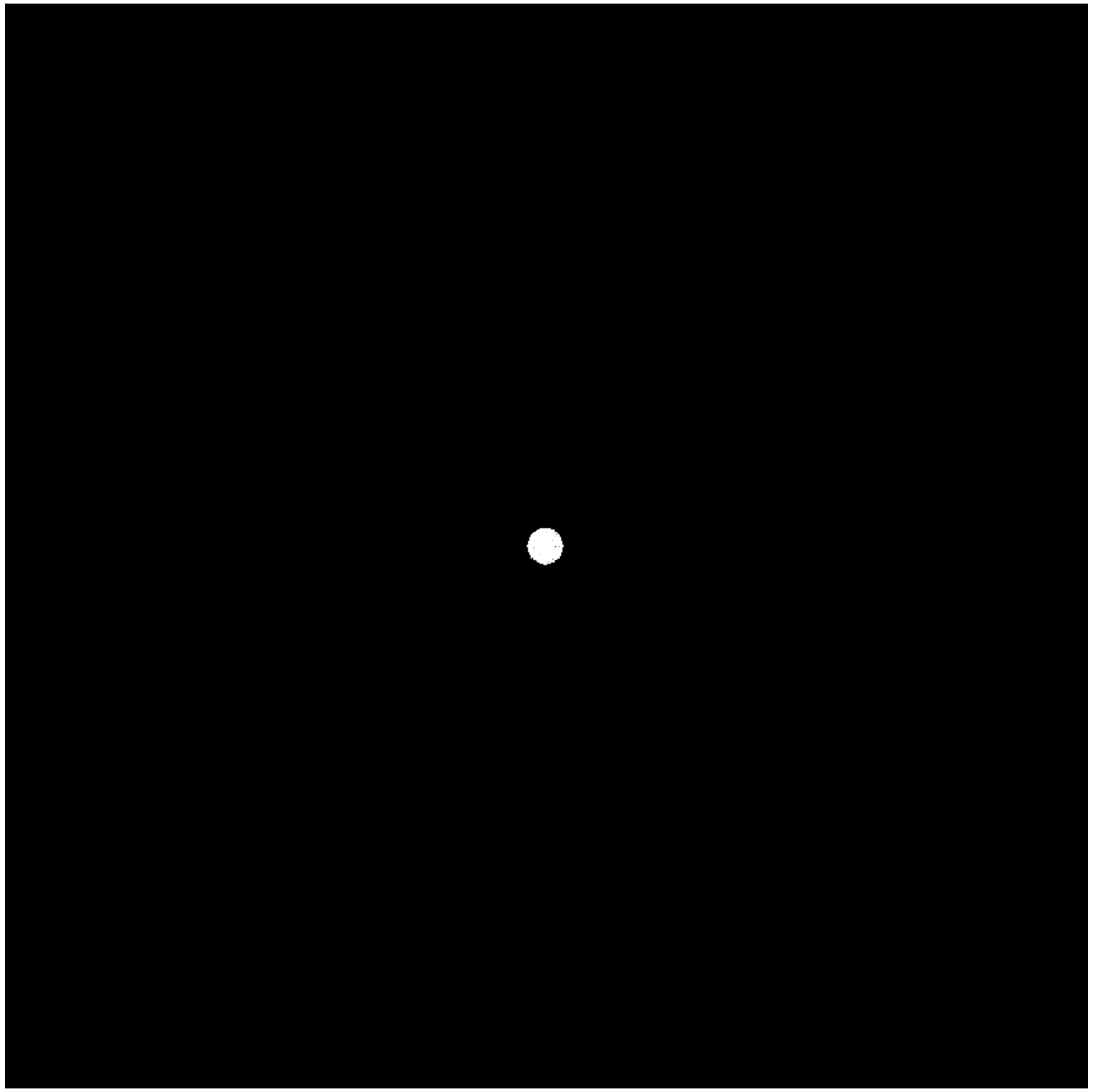


$$f(x) = \frac{A}{2} + \frac{2A\cos(t\omega)}{\pi} - \frac{2A\cos(3t\omega)}{3\pi} + \frac{2A\cos(5t\omega)}{5\pi} - \frac{2A\cos(7t\omega)}{7\pi} + \cdots$$

Visualizing Image Frequency Content

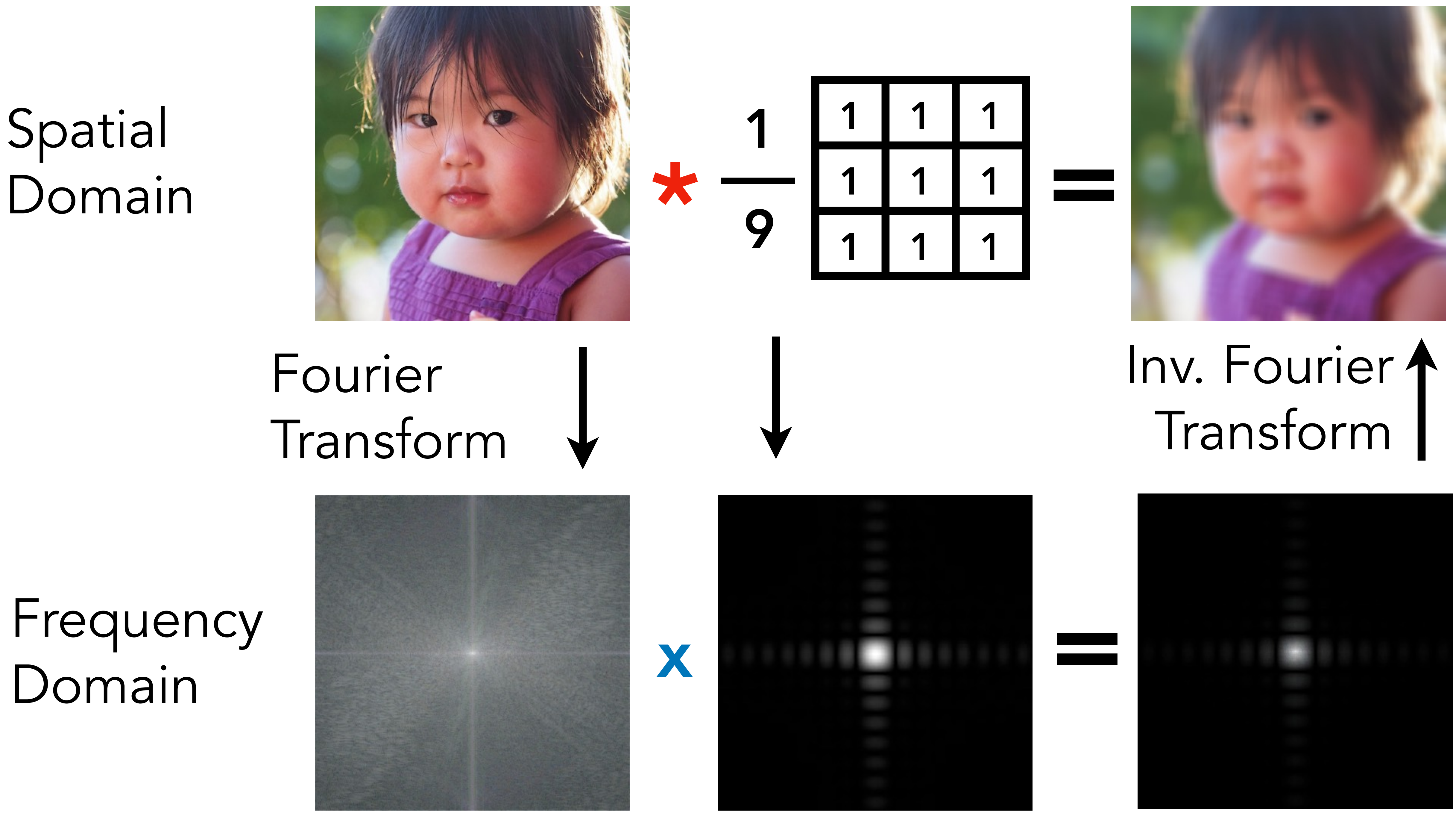


Filtering = Getting rid of certain frequency contents



Low-pass filter

Convolution Theorem



A general understanding

- Any product integral can be considered as filtering

$$\int_{\Omega} f(x)g(x)\,\mathrm{d}x$$

- Low frequency == smooth function / slow changes / etc.

- The frequency of the integral is the lowest of any individual's

Basis Functions

- A set of functions that can be used to represent other functions in general

$$f(x) = \sum_i c_i \cdot B_i(x)$$

- The Fourier series is a set of basis functions

- The polynomial series can also be a set of basis functions

Questions?

Today

- Finishing up
  - Shadow from environment lighting
- Background knowledge
  - Frequency and filtering
  - Basis functions
- Real-time environment lighting (& global illumination)
  - Spherical Harmonics (SH)
  - Prefiltered env. lighting
  - Precomputed Radiance Transfer (PRT)

Spherical Harmonics

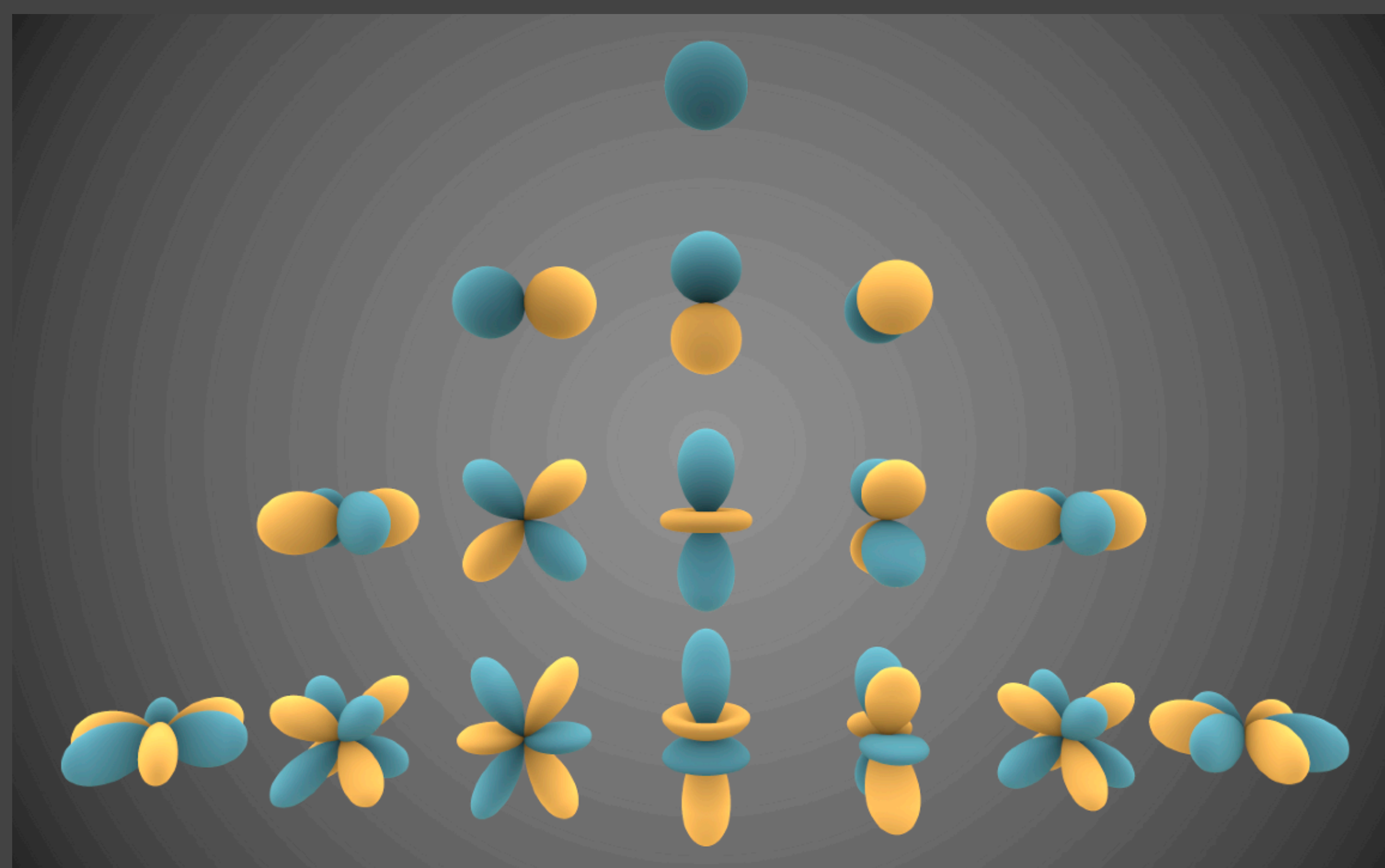
- A set of 2D basis functions $B_i(\omega)$ defined on the sphere
- Analogous to Fourier series in 1D

$l = 0$

$l = 1$

$l = 2$

$l = 3$



$\ldots \quad -2 \quad -1 \quad m = 0 \quad 1 \quad 2 \quad \ldots$

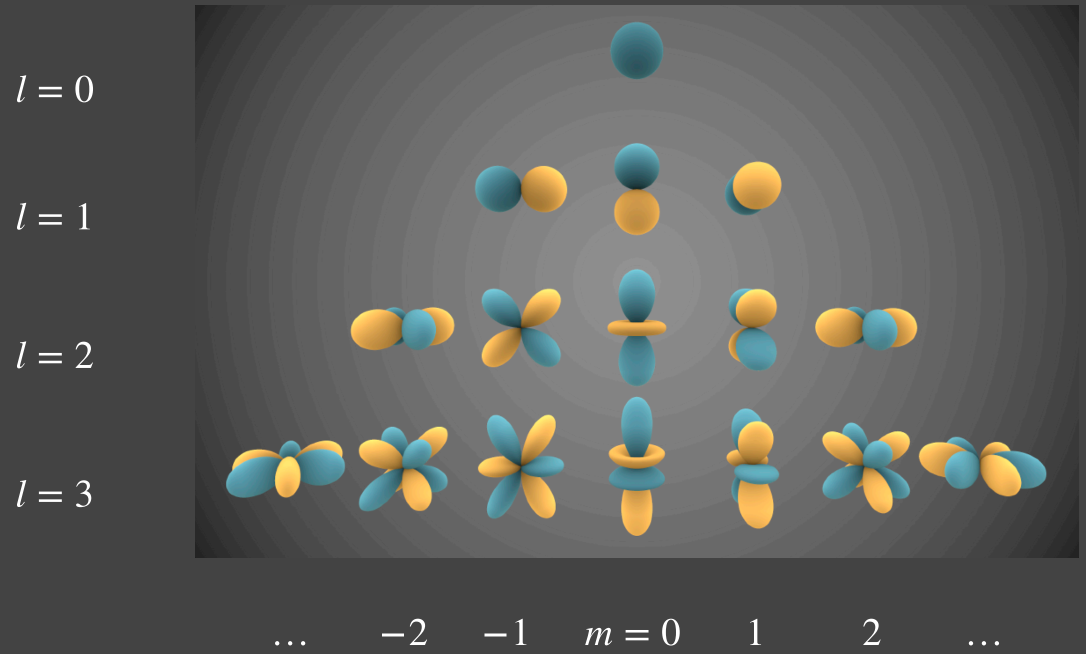
Spherical Harmonics

- Each SH basis function $B_i(\omega)$ is associated with a (Legendre) polynomial

- Projection: obtaining the coefficients of each SH basis function

$$c_i = \int_\Omega f(\omega) B_i(\omega) \, \mathrm{d}\omega$$

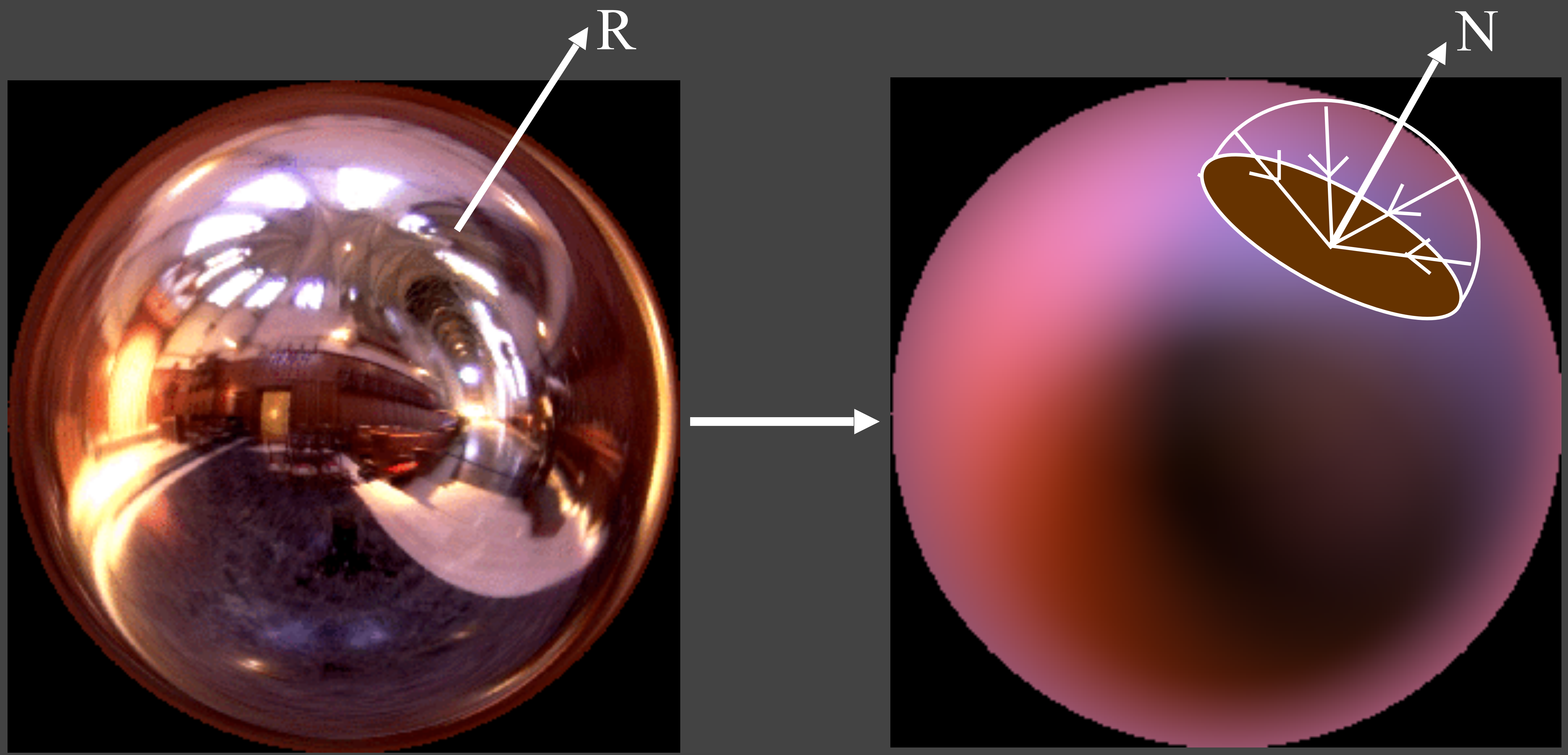
- Reconstruction: restoring the original function using (truncated) coefficients and basis functions



Next Slides Courtesy of Prof. Ravi Ramamoorthi from UC San Diego

Recall: Prefiltering

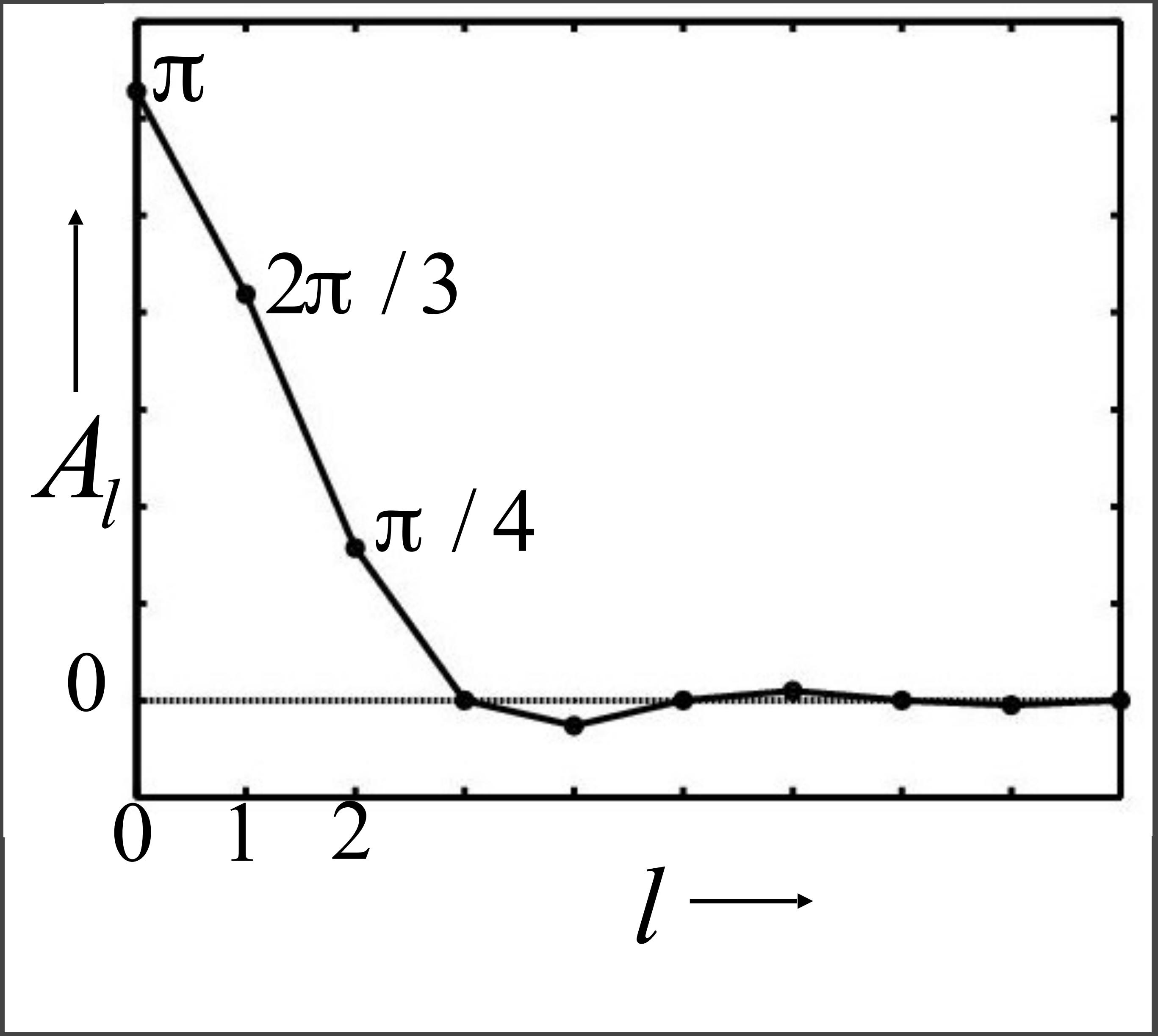
- Prefiltering + single query = no filtering + multiple queries



Analytic Irradiance Formula

- Diffuse BRDF acts like a low-pass filter

$$E_{lm} = A_l L_{lm}$$

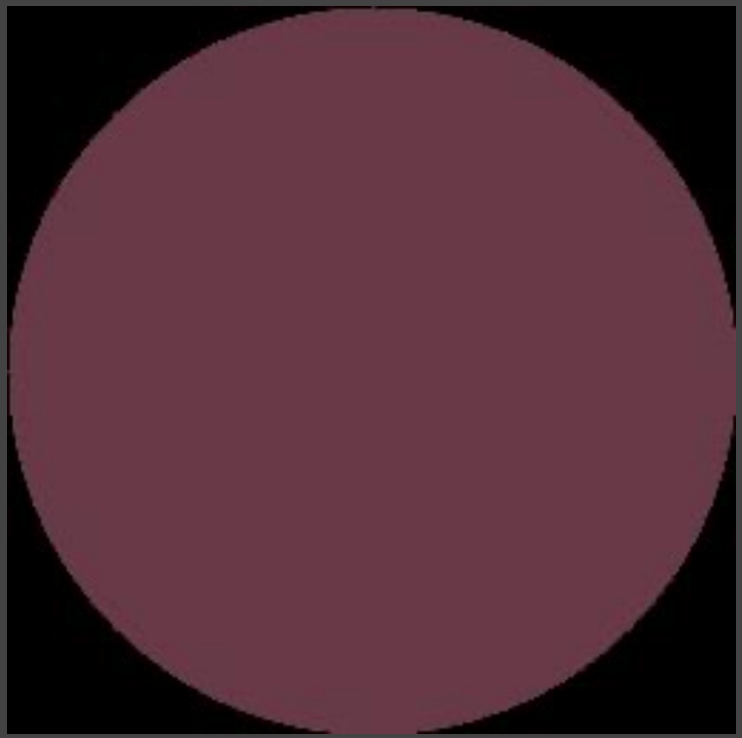


Ramamoorthi and Hanrahan 01
Basri and Jacobs 01

$$A_l = 2\pi \frac{(-1)^{\frac{l}{2}-1}}{(l+2)(l-1)} \left[\frac{l!}{2^l \left(\frac{l}{2}!\right)^2} \right] \quad l \text{ even}$$

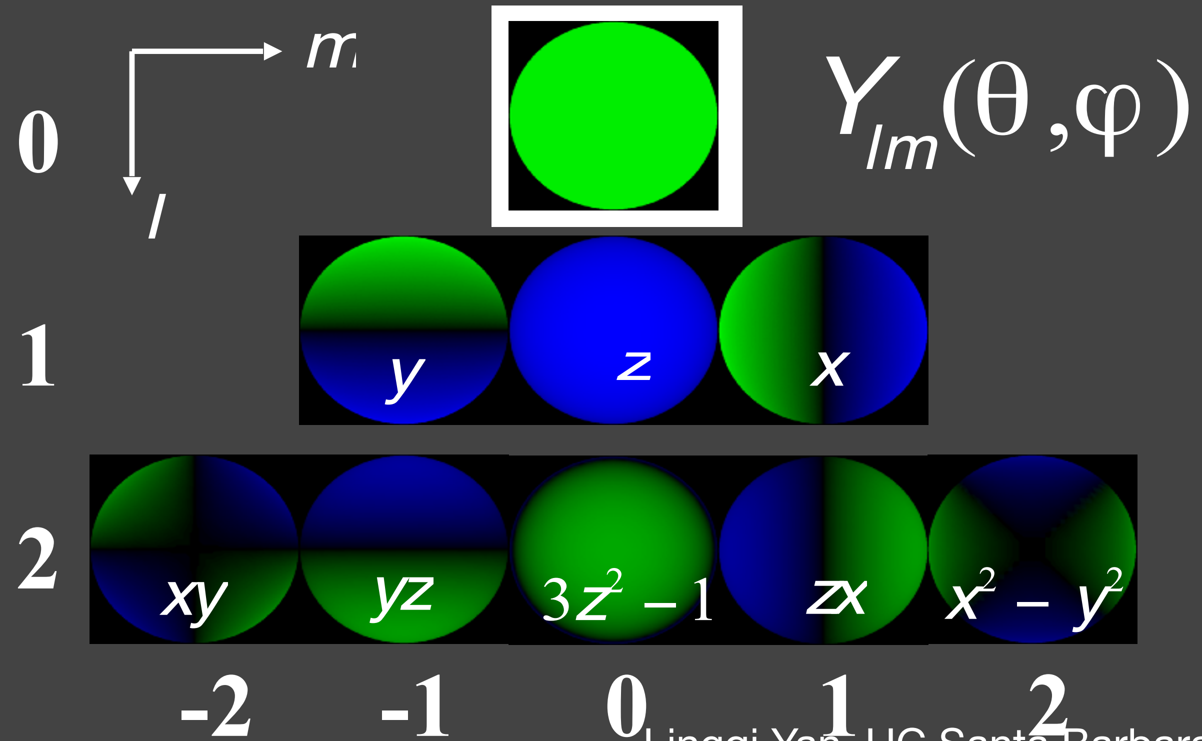
9 Parameter Approximation

Exact image



Order 0
1 term

RMS error = 25 %

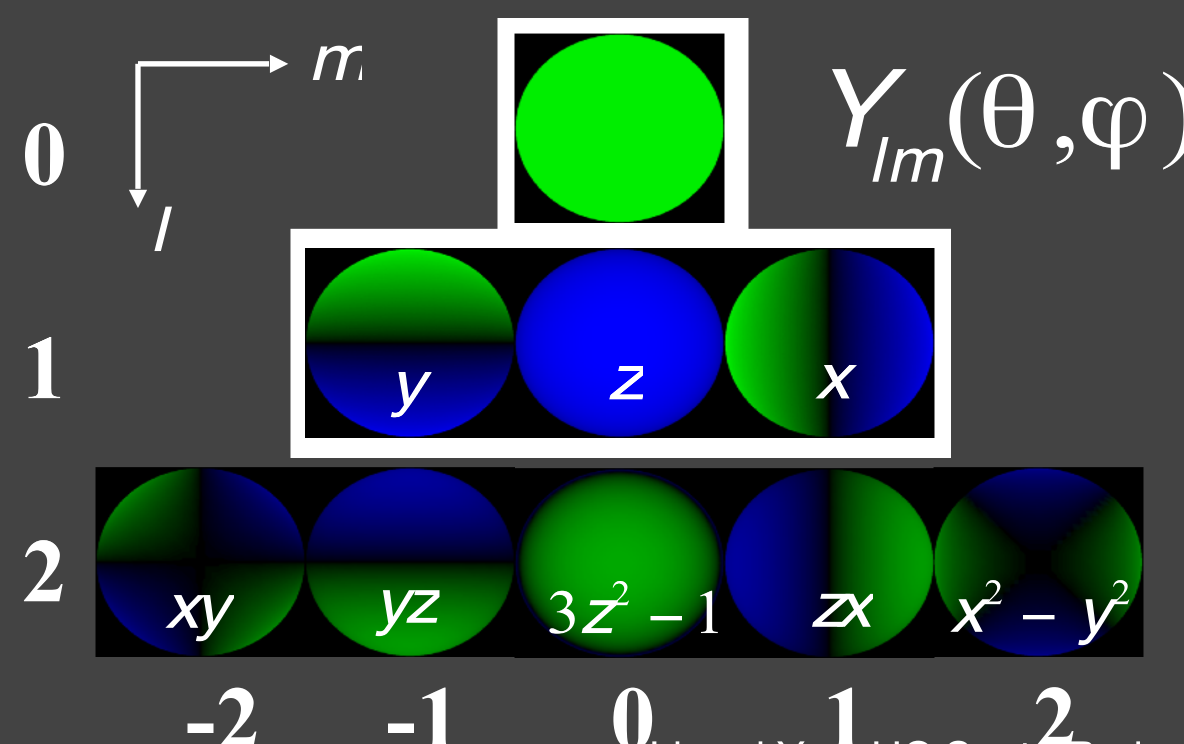


9 Parameter Approximation

Exact image

Order 1
4 terms

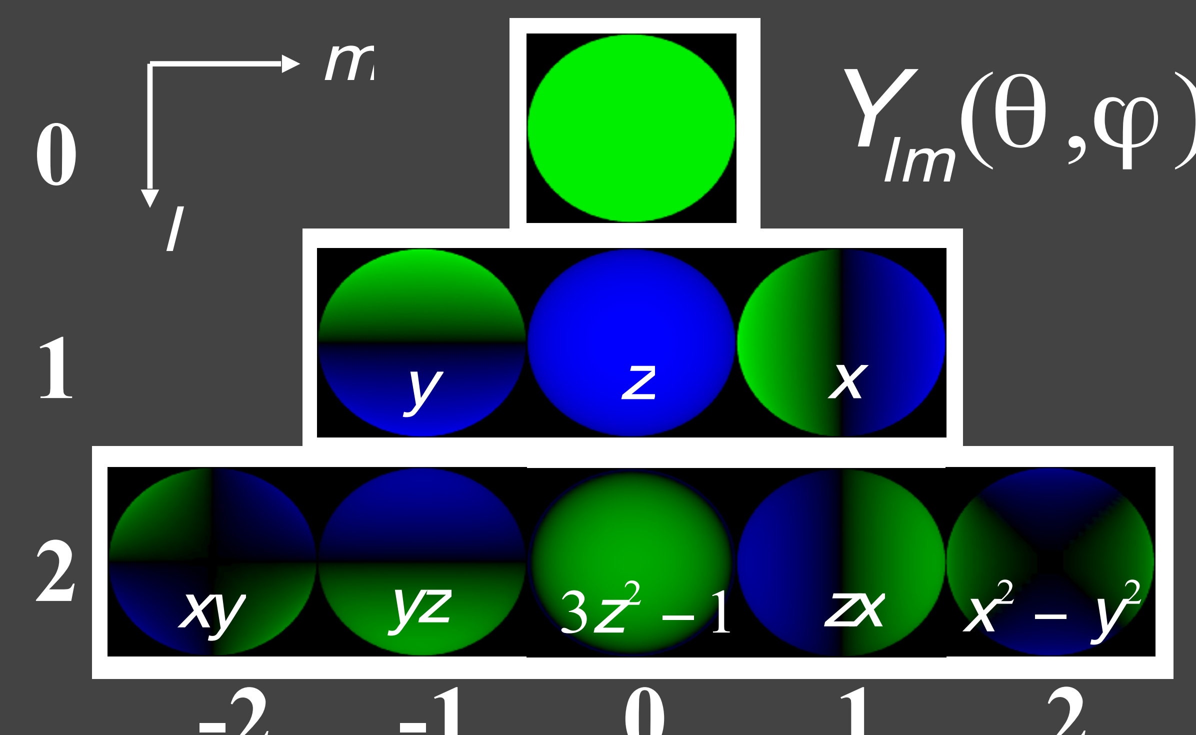
RMS Error = 8%



9 Parameter Approximation

Exact image

Order 2
9 terms



RMS Error = 1%

For any illumination, average error < 3% [Basri Jacobs 01]

In Real-Time Rendering (FYI)

$$E(n) = n^t M n$$

Simple procedural rendering method (no textures)

- Requires only matrix-vector multiply and dot-product
- In software or NVIDIA vertex programming hardware

Widely used in Games (AMPED for Microsoft Xbox), Movies (Pixar, Framestore CFC, …)

```
surface float1 irradmat (matrix4 M, float3 v) {
        float4 n = {v , 1} ;
        return dot(n , M*n) ;
}
```

A Brief Summarization

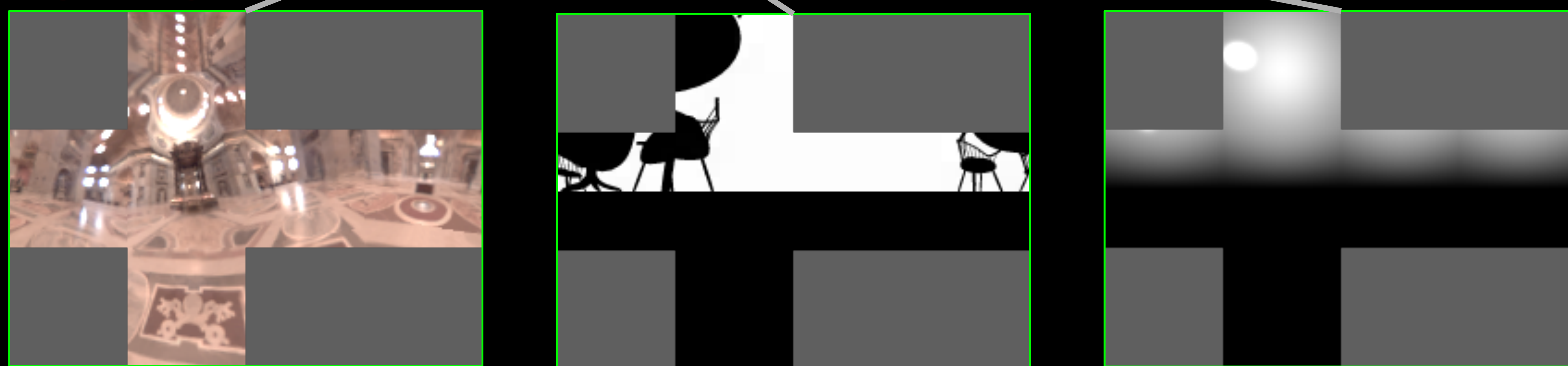
- We have seen the usage of basis functions
  - Representing any function (with enough #basis)
  - Keeping a certain frequency contents (with a low #basis)
  - Reducing integrals to dot products (?)
- But here it's still shading from environment lighting
  - No shadows yet
- Next: Precomputed Radiance Transfer (PRT)
  - Handles shadows and global illumination!
  - What's the cost?

Next Slides Courtesy of Prof. Kun Xu from Tsinghua University

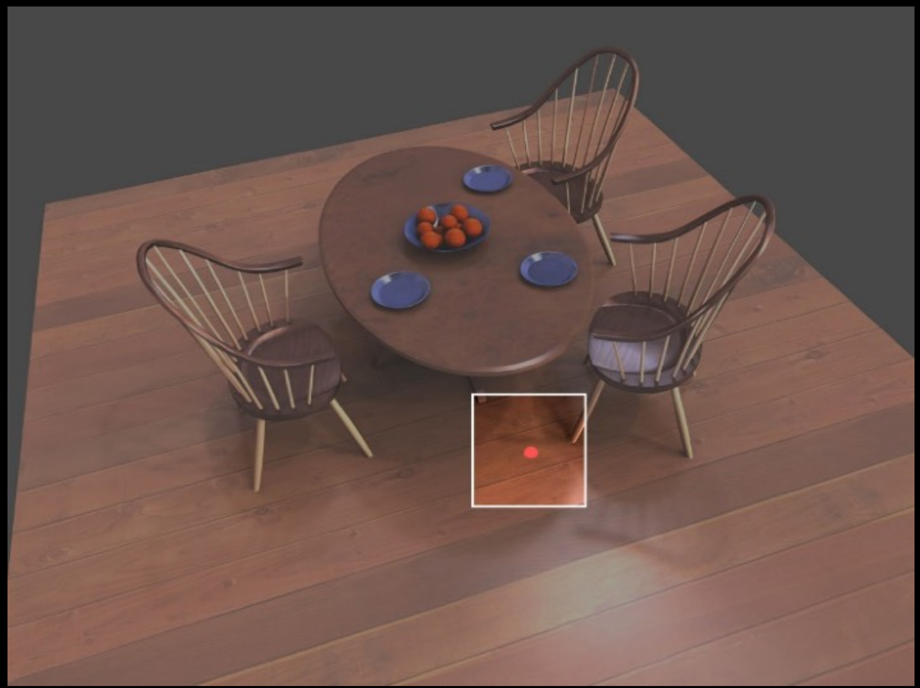
Rendering under environment lighting

$$L(\mathbf{o}) = \int_{\Omega} \underline{L(\mathbf{i})}\,\underline{V(\mathbf{i})}\,\underline{\rho(\mathbf{i},\mathbf{o})\max(0,\boldsymbol{n}\cdot\mathbf{i})}\,\mathrm{d}\mathbf{i}$$

lighting visibility BRDF



- $\mathbf{i}/\mathbf{o}$: incoming/view directions
- Brute-force computation
  - Resolution: 6*64*64
  - Needs 6*64*64 times for each point!



Precomputed Radiance Transfer (PRT)

- Introduced by Sloan in SIGGRAPH 2002
  - *Precomputed Radiance Transfer for Real-Time Rendering in Dynamic, Low-Frequency Lighting Environments* [Sloan 02]



Basic idea of PRT [Sloan 02]

$$L(\mathbf{o}) = \int_\Omega \underbrace{L(\mathbf{i})}_{\text{lighting}} \underbrace{V(\mathbf{i})\,\rho(\mathbf{i},\mathbf{o})\max(0,\boldsymbol{n}\cdot\mathbf{i})}_{\text{light transport}}\mathrm{d}\mathbf{i}$$

- Approximate lighting using basis functions
  - $L(\mathbf{i}) \approx \sum l_i B_i(\mathbf{i})$
- Precomputation stage
  - compute light transport, and project to basis function space
- Runtime stage
  - dot product (diffuse) or matrix-vector multiplication (glossy)

Diffuse Case

$$L(\mathbf{o}) = \rho \int_{\Omega} L(\mathbf{i}) V(\mathbf{i}) \max(0, \boldsymbol{n} \cdot \mathbf{i}) d\mathbf{i}$$

$$L(\mathbf{i}) \approx \sum \underbrace{l_i}_{\text{lighting coefficient}} \underbrace{B_i(\mathbf{i})}_{\text{basis function}}$$

$$L(\mathbf{o}) \approx \rho \sum l_i \int_{\Omega} \mathrm{B}_i(\mathbf{i}) V(\mathbf{i}) \max(0, \boldsymbol{n} \cdot \mathbf{i}) d\mathbf{i}$$

Precompute

$$L(\mathbf{o}) \approx \rho \sum l_i T_i$$

- Reduce rendering computation to dot product

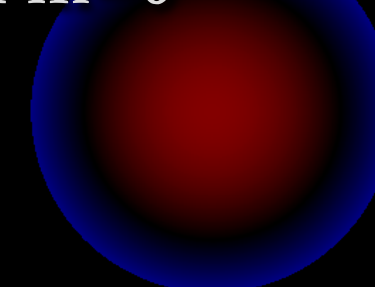
Basis functions $B(\mathbf{i})$

- Spherical Harmonics (SH)
- SH have nice properties:
  - orthonormal
  - simple projection/reconstruction
  - simple rotation
  - simple convolution
  - few basis functions: low freqs

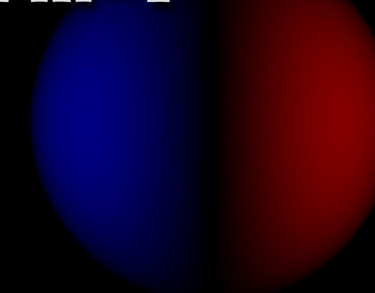
l=0 m=0

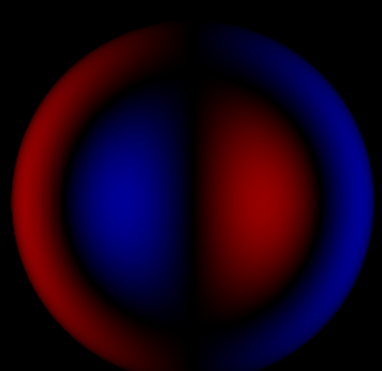
l=1 m=-1

l=1 m=0



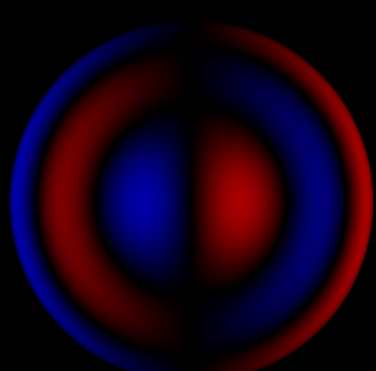
l=1 m=1





l=2 m=1

l=3 m=-1

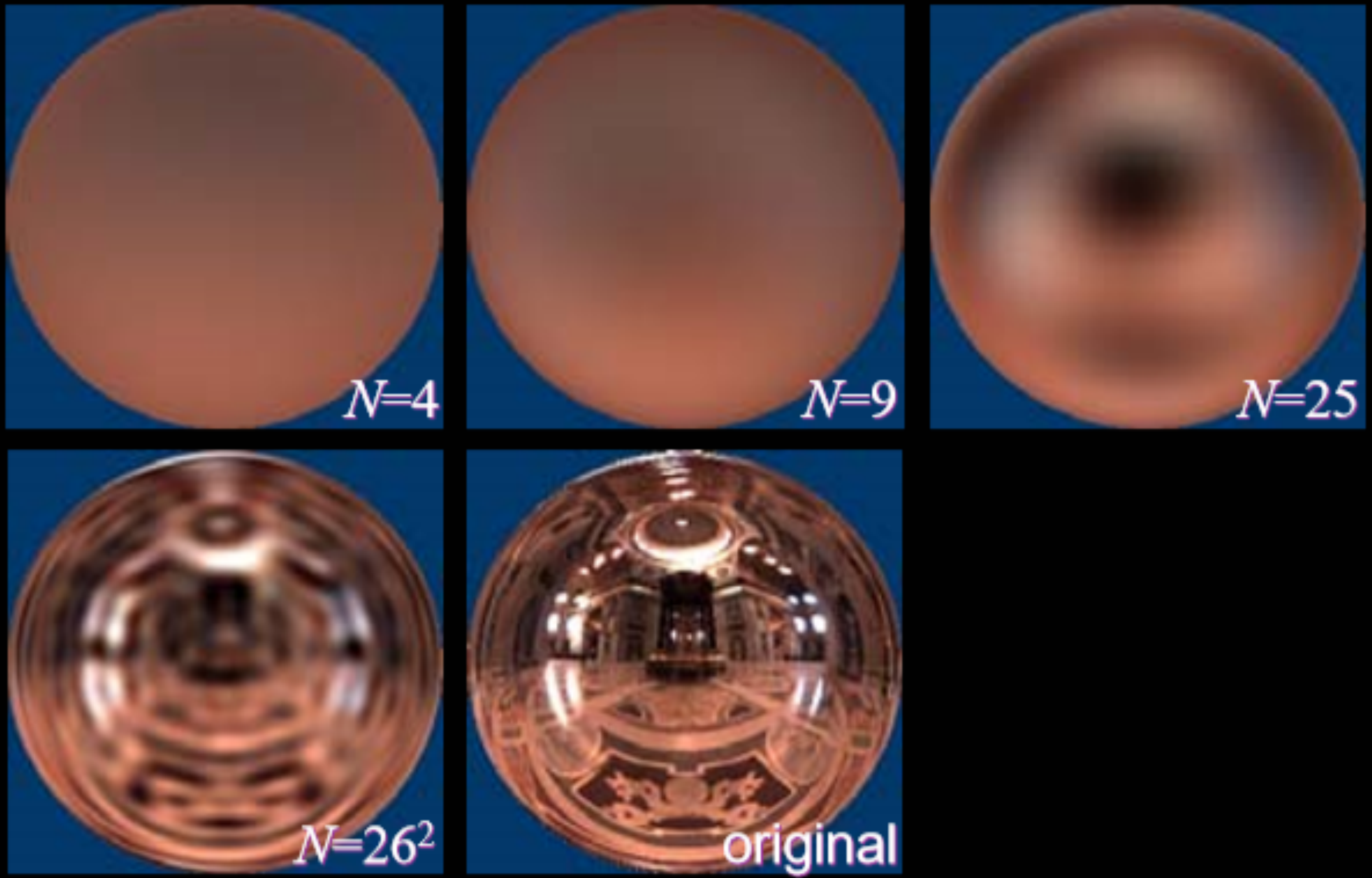


l=3 m=2

l=4 m=-2

Basis functions $B(\mathbf{i})$

- Spherical Harmonics (SH)
- Light Approximation Examples



Low frequency

Basis functions $B(\mathbf{i})$

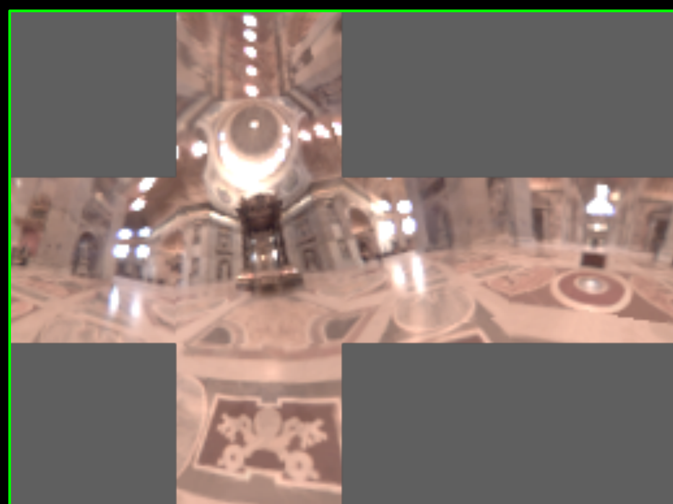
- SH is orthonormal, we have:

$$\int_{\Omega} B_i(\mathbf{i}) \cdot B_j(\mathbf{i}) \mathrm{d}\mathbf{i} = \mathbf{1} \quad (\mathbf{i} = j)$$

$$\int_{\Omega} B_i(\mathbf{i}) \cdot B_j(\mathbf{i}) \mathrm{d}\mathbf{i} = \mathbf{0} \quad (\mathbf{i} \neq j)$$

Basis functions $B(\mathbf{i})$

Original space



lighting

$$L(\mathbf{i}) \approx \sum l_i B_i(\mathbf{i})$$

SH space

lighting coefficients

- Projection to SH space

$$l_i = \int_{\Omega} L(\mathbf{i}) \cdot B_i(\mathbf{i}) \, \mathrm{d}\mathbf{i}$$

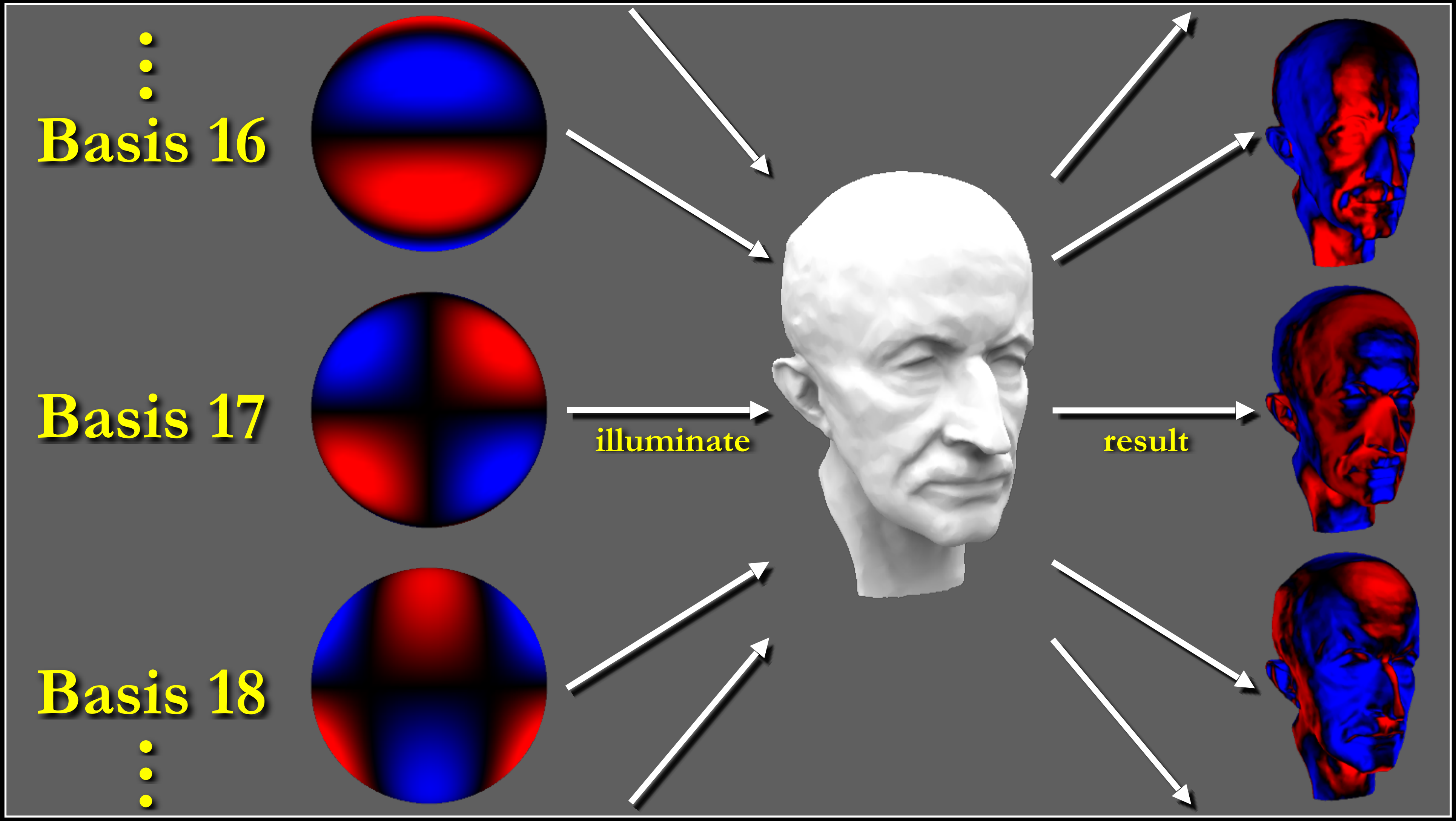
- Reconstruction

$$L(\mathbf{i}) \approx \sum l_i B_i(\mathbf{i})$$

Precomputation

light transport $T_i \approx \int_{\Omega} \mathrm{B}_i(\mathbf{i})V(\mathbf{i})\max(0,\boldsymbol{n}\cdot\mathbf{i})d\mathbf{i}$

- No shadow/ shadow/ inter-reflection



Run-time Rendering

$$L(\mathbf{o}) \approx \rho \sum l_i T_i$$

- Rendering at each point is reduced to a dot product
  - First, project the lighting to the basis to obtain l_i
  - Or, rotate the lighting instead of re-projection
  - Then, compute the dot product
- Real-time: easily implemented in shader

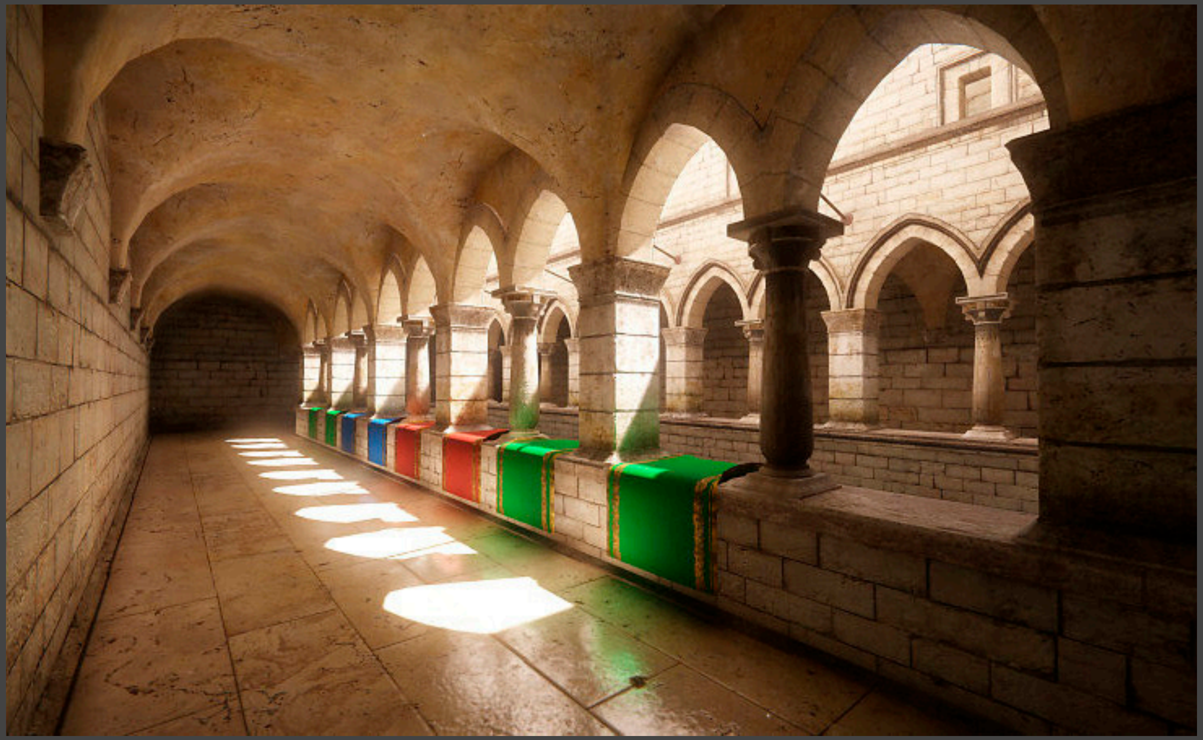
Diffuse Rendering Results

No Shadows | Shadows | Shadows+Inter

Questions?

Next Lecture

- Real-time global illumination cont.
  - ~~By precomputation~~
  - In 3D (LPV, VXGI, RTXGI, etc.)
  - In the image space (SSR, etc.)



[VXGI by NVIDIA]

Thank you!