ECE 595 / CS 491 / CS 591

Real-Time Rendering & Graphics Hardware

Pradeep Sen
Advanced Graphics Lab

Class 17
March 28, 2007

Announcements

- GFX Café this Friday will be given by Christoph Salge on zooming into images arbitrarily

Last time

- Introduction to illumination

Today

- More on real-time illumination
- Irradiance maps
- Spherical Harmonics

Real-time lighting

- Diffuse component:

$$N \cdot L = \cos(\theta)$$

  - View independent, a measure of the irradiance on the surface
- Specular component (Phong)

$$(R \cdot V)^m$$

  - View dependent
- Ambient component
  - Constant term
  - Simulates global illumination (not direct illumination)

Real-time illumination

- Typical illumination equation in real-time rendering

$$I = I_a k_a + \sum_i f_{att} I_i \left[k_d (\vec{N} \bullet \vec{L}_i) + k_s (\vec{R}_i \bullet \vec{V}) \right]$$

Reflection mapping

Blinn and Newell (1976)

Environment mapping

Miller and Hoffman (1984)

Environment maps from light probes

Light probes courtesy Paul Debevec

Bump-mapping

Blinn (1978)

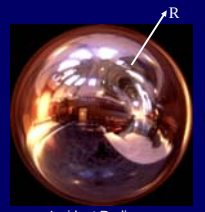
Irradiance maps

Directional Source

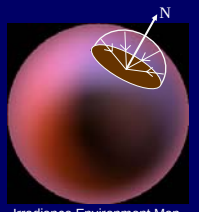
Natural Illumination

Slide courtesy Ravi Ramamoorthi

Precomputed Irradiance Map



Incident Radiance
(Illumination Environment Map)



Irradiance Environment Map

Slide courtesy Ravi Ramamoorthi

Problem with traditional irradiance maps

- Time consuming to compute!
- For an environment map of size S and a target irradiance map resolution of size T, computing the irradiance map would be O(ST)
- What if we were told that we could represent 99% of the information needed from the environment with only 9 values?
- Ramamoorthi and Hanrahan, SIGGRAPH 01

Introduction to spherical harmonics

- Orthonormal basis over the sphere, analogous to the Fourier transform over 1D circle
- Basis functions defined as:

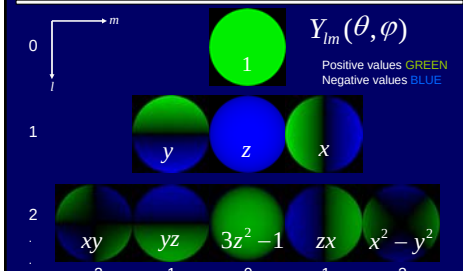
$$Y_{lm}(\theta,\phi) = K_{lm} e^{im\phi} P_l^{|m|}(\cos\theta)$$

- Where

$$K_{lm} = \sqrt{\frac{(2l+1)}{4\pi}\frac{(l-|m|)!}{(l+|m|)!}}$$ (Normalization factor)

- And P_l^m are associated *Legendre polynomial functions*

Spherical Harmonics



Slide courtesy Ravi Ramamoorthi

Representing incident radiance and irradiance

- So both the incident illumination and the irradiance can be written in terms of this basis:

$$L(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} L_{lm} Y_{lm}(\theta,\phi)$$

$$E(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} E_{lm} Y_{lm}(\theta,\phi)$$

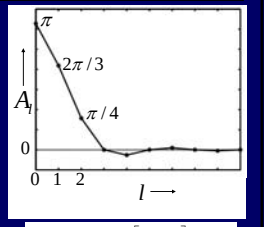
- And the relationship between their coefficients is given by

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

For Lambertian surfaces

- Lambertian surfaces act like low-pass filters

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

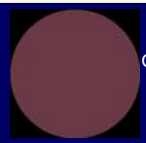


$$A_l = 2\pi \frac{(-1)^{\frac{l}{2}-1}}{(l+2)(l-1)}\left[\frac{l!}{2^l\left(\frac{l}{2}!\right)^2}\right] \quad l \text{ even}$$

Slide courtesy Ravi Ramamoorthi

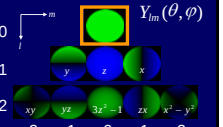
9 parameter approximation

Correct image



Order 0
1 term

RMS error = 25 %



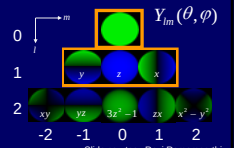
Slide courtesy Ravi Ramamoorthi

9 parameter approximation

Correct image

Order 1
4 terms

RMS error = 8 %



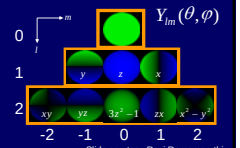
Slide courtesy Ravi Ramamoorthi

9 parameter approximation

Correct image

Order 2
9 terms

RMS error = 1 %



Slide courtesy Ravi Ramamoorthi

Projecting illumination into spherical harmonic basis

- To project a function *f(s)* into spherical harmonic basis we need to perform the calculate integral

$$C_{lm} = \int f(s) Y_{lm}(s) ds$$

- To integrate over the hemisphere for $f(\theta,\phi)$:

$$A = \int_0^{2\pi}\int_0^{\pi} f(\theta,\phi)\sin(\theta)\,d\theta d\phi$$

- Therefore to project the illumination into the spherical harmonic basis:

$$L_{lm} = \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} L(\theta,\phi)\,Y_{lm}(\theta,\phi)\,\sin\theta\,d\theta d\phi$$

Weighted sums of pixels

- We can perform this integral by doing a weighted sum of the pixels in the environment map

$$L_{lm} = \sum_{pixels(\theta,\phi)} envmap[pixel] \times basisfunc_{lm}[pixel]$$

Equations courtesy Ravi Ramamoorthi

During rendering

- Need to evaluate irradiance given a surface normal

$$E(\theta,\phi) = \sum_{l=0}^{\infty}\sum_{m=-l}^{l} E_{lm} Y_{lm}(\theta,\phi)$$

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

- Because we only have 9 terms, we can write this as a polynomial:

$$E(n) = c_4 L_{00} 1 + 2c_2 L_{11} x + 2c_2 L_{1-1} y + 2c_2 L_{10} z + c_5 L_{20}(3z^2-1) + 2c_1 L_{2-2} xy + 2c_1 L_{21} xz + 2c_1 L_{2-1} yz + c_1 L_{22}(x^2-y^2)$$

Equations courtesy Ravi Ramamoorthi

During rendering

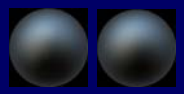
- This polynomial can be executed on the hardware by multiplying by a matrix:

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

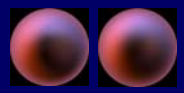
- See implementation details in Ramamoorthi et al.'s paper

Results

Eucalyptus grove



exact approximate



Grace cathedral

Results courtesy Ravi Ramamoorthi

Video

SH lighting in games