CS562 Project 3b

Synopsis

Write an application, using spherical harmonics, to create your own irradiance maps. As an extra challenge, you may implement this via a compute shader on a GPU instead of as a CPU application.

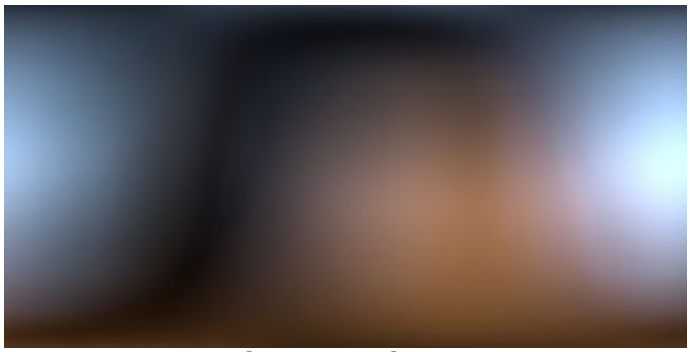
Reference: A*n Efficient Representation for Irradiance Environment Maps* by Ravi Ramamoorthi, and Pat Hanrahan

Instructions

Read in a suitable HDR environment image (HDR Labs has about 50 freely available at http://www.hdrlabs.com/sibl/archive.html). Make one pass through all the pixels, projecting them onto nine (three bands worth of) spherical harmonic irradiance coefficients. Output the nine irradiance coefficients, **OR** evaluate the resulting spherical harmonic over the full sphere and output as a small (suggested 400×200) irradiance map. In either case, the nine irradiance coefficients or the irradiance map created from them is to be input to the shader performing the lighting calculation for the purpose of evaluating the diffuse portion of the Image-Based-Lighting calculation.

Example:

A sample full resolution HDR image



Low resolution irradiance HDR image

Irradiance map equations

The lighting equation (integral) with just the diffuse portion of the BRDF provides the definition of the irradiance map:

$$\int_{\Omega} \frac{K_d}{\pi} L_i(\omega_i) (N \cdot \omega_i)_+ \, d\omega_i = \frac{K_d}{\pi} \int_{\Omega} L_i(\omega_i) (N \cdot \omega_i)_+ \, d\omega_i = \frac{K_d}{\pi} \, irradiance(N)$$

so the irradiance map is

$$irradiance(N) = \int_{\Omega} L_i(\omega_i) (N \cdot \omega_i)_+ \, d\omega_i = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} L(\omega_i) (N \cdot \omega_i)_+ \sin\theta \; d\theta \; d\phi$$

where the integrals are over the full sphere, ω_i is the unit vector defined by θ and ϕ and $L(\omega_i)$ is the input environment image evaluated at ω_i. The single integral form is the easy way to write the integral over a sphere, while the equivalent double integral form provides a practical method of evaluation.

The integrand is projected onto spherical harmonic coefficients in two parts which are then easily combined.

Set 1 of Spherical Harmonic coefficients: Projection of the dot product term:

The $(N \cdot \omega_i)_+$ term has been projected analytically (see the referenced paper) to give three $\hat{A}_l$ (There is no dependence on the subscript.)

$$\hat{A}_0 = \pi; \quad \hat{A}_1 = \frac{2}{3}\pi; \quad \hat{A}_2 = \frac{1}{4}\pi$$

Set 2 of Spherical Harmonic coefficients: Projection of the input image:

The projection of the input image (and extra $\sin\theta$ term) onto nine coefficients L_{lm} of the nine spherical harmonic basis functions Y_{lm} is the inner product of the input image (considered as a function on the sphere) and the individual basis functions:

$$L_{lm} = \int_{\theta=0}^{\pi} \int_{\phi=0}^{2\pi} L(\theta,\phi)\, Y_{lm}(\theta,\phi) \sin\theta \, d\theta \, d\phi \approx \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} L(i,j)\, Y_{lm}(x,y,z) \sin\theta \;\; \Delta\theta \;\; \Delta\phi$$

where $\Delta\theta = \pi/H$, $\Delta\phi = 2\pi/W$ are the step sizes of the two sums.

Notes

- ***I haven't yet resolved whether that double sum should have a*** $1/(4\pi)$ ***factor or not.***
- The $L(...)$ are RGB pixels from the input image.
- The nine L_{lm} are therefore also RGB values.
- These equations have three related ways of specifying points on a sphere
  - i, j refer to a pixel in the input image which is wrapped around a sphere,
  - $\theta = \pi\,(i+1/2)/H$, $\phi = 2\pi\,(j+1/2)/W$ are two angles specifying the corresponding point on the sphere. (The $+1/2$ is to compute those angles at the **center** of pixels.)
  - $(x,y,z) = (\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\theta)$ is the corresponding unit vector on the sphere
- **The double sum is the algorithm**. That is, loop through all the pixels, summing up the product of
  - an RGB pixel value,
  - a basis function evaluation (see below) at that pixel's direction,
  - the sine function,
  - and the two step-size deltas.
- As a test, part of that calculation should approximate the area of the unit sphere:

$$4\pi \approx \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} \sin\theta \;\Delta\theta\;\Delta\phi$$

Final set of Spherical Harmonic coefficients :

The final set of spherical harmonic coefficients are products of the previous two sets

$$E_{lm} = \hat{A}_l \, L_{lm}$$

Evaluation:

We are now ready to evaluate (approximate) the irradiance from the input image into at any point with normal $N = (x,y,z)$ as this spherical harmonic polynomial:

$$irradiance(x,y,z) = \sum_{l,m} E_{lm}\, Y_{lm}(x,y,z)$$

There are two possible scenarios for using the equation for $irradiance(x,y,z)$ in a lighting situation

1. **In the shader:** Send the nine E_{lm} coefficients (each is an RGB value) to the shader. At a point with normal $N = (x,y,z)$, evaluate the $irradiance(x,y,z)$.
2. **Into an irradiance map:** Grid the sphere with a small selection of angles ϕ and θ, (say 400 times 200), convert to a unit direction $(x,y,z) = (\sin\theta\cos\phi,\ \sin\theta\sin\phi,\ \cos\theta)$, evaluate the $irradiance(x,y,z)$, and store the result into the corresponding pixel of an HDR irradiance map image. Write the irradiance map to be used in your lighting calculation.

The spherical harmonic basis functions (first three bands, constant, linear, quadratic):

$$Y_{2,-2}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{15}{\pi}}\right)xy$$

$$Y_{1,-1}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{3}{\pi}}\right)y \qquad Y_{2,-1}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{15}{\pi}}\right)yz$$

$$Y_{0,0}(\theta,\phi) = \frac{1}{2}\sqrt{\frac{1}{\pi}} \qquad Y_{1,0}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{3}{\pi}}\right)z \qquad Y_{2,0}(\theta,\phi) = \left(\frac{1}{4}\sqrt{\frac{5}{\pi}}\right)(3z^2-1)$$

$$Y_{1,1}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{3}{\pi}}\right)x \qquad Y_{2,1}(\theta,\phi) = \left(\frac{1}{2}\sqrt{\frac{15}{\pi}}\right)xz$$

$$Y_{2,2}(\theta,\phi) = \left(\frac{1}{4}\sqrt{\frac{15}{\pi}}\right)(x^2-y^2)$$