Improved Ambient Occlusion

The classical lighting equation to compute the outgoing radiance from a pixel at in direction is given by:

(1)

Where:

* is the incoming radiance at from direction
* is the surface’s BRDF
* is the surface normal
* is the set of all directions covering the upper hemisphere
* is the solid angle covered by the perceived surface along direction

Focusing only on diffuse lambertian reflection, we can rewrite eq. (1) as:

(2)

Where:

* is the *irradiance* arriving at surface location and normal **.**
* represents the diffuse BRDF for a surface with albedo . The division by is here to guarantee energy conservation since .

# Extracting Common Ambient Occlusion

By introducing the visibility term:

We can rewrite eq. (2) as two distinct direct and indirect parts:

(3)

The direct term is (incorrectly) simplified into:

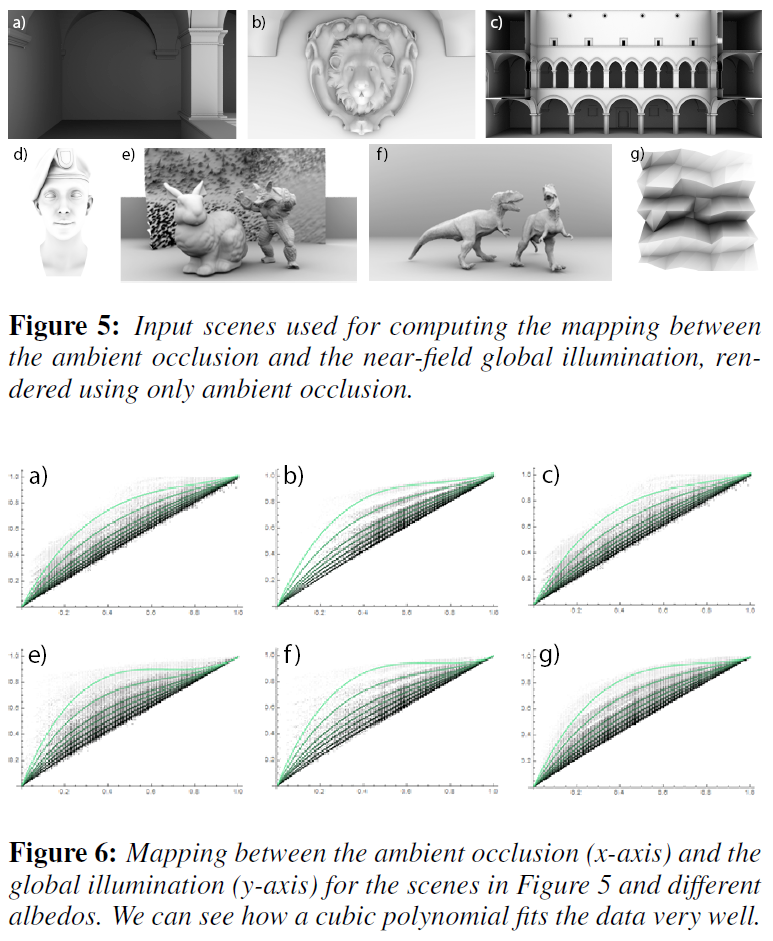
(4)

Where:

* is the irradiance estimate in the direction of the normal which is usually pulled from a distant harmonics (in which case, the location is useless) representation of the scene irradiance like a cube map or spherical
* is the scaled integration of the visibility term over all possible directions that we generally call the “Ambient Occlusion” and is a value we store per-vertex in meshes, or per-texel (i.e. AO map).

# Indirect Term

The second indirect term is often ignored although it provides significant additional energy, especially when the visibility term is largely low (i.e. a very narrow aperture in the surface) and the surface reflectance is high.  
Jimenez et al. attempted to give a simplified value for the added energy due to near-field inter-reflections in their 2016 paper *“Practical Realtime Strategies for Accurate Indirect Occlusion”* in chapter 5 but they provided the energy “in bulk”, from only 3 bounces and averaging empirical data, ignoring the subtle effects of each bounce on the albedo and the color saturation that ensues.



The curves showing the statistical relationship between AO value (x-axis) and the  
energy regained through direct perception and near-field inter-reflections (y-axis).

First, let’s rewrite the indirect irradiance term as:

To simplify the computation, we pose that the surface reflectance at the neighbor site is the same as the current location and since the reflectance is now constant for the entire surface, we can finally write:

Where is the irradiance perceived by the neighbor surface in normal direction , introducing a recurrence relationship between the irradiance terms.

If we start from the initial irradiance term and provide ***unit radiance*** then:

*Warning:* this is NOT the same as AO from eq. (4) because of the dot product within the integral here.

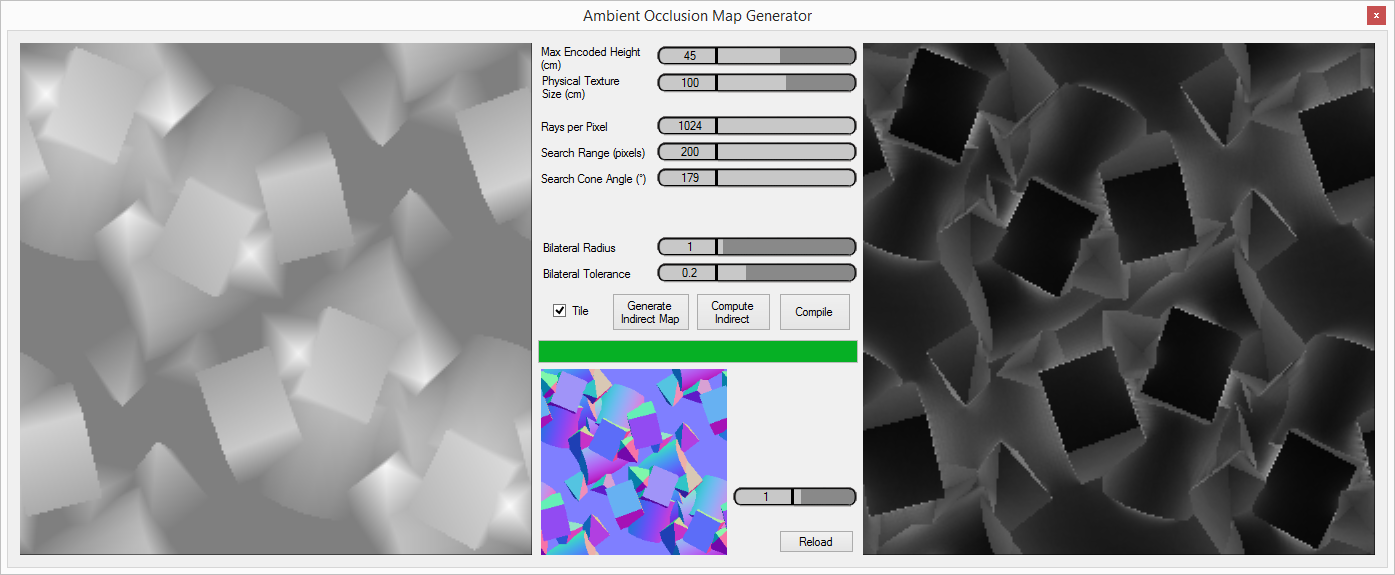
Then the irradiance term after a single bounce is:

Writing the irradiance for yet another bounce gives:

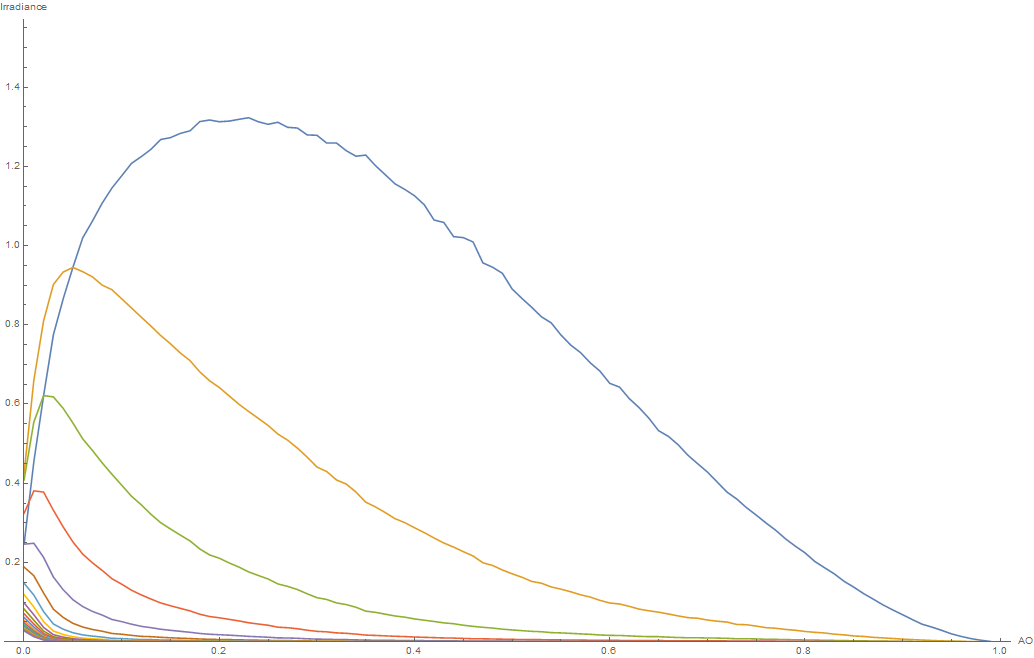
As a general rule, if have a way to compute the integral then we can obtain:

# Collecting Data

I didn’t want to write an entire path tracer like Jimenez did and instead decided to modify my little application that computes AO maps from height maps to also store “form factors” (a huge collection of links from each pixel to each visible neighbor) in order to perform the integration described above.



I collected irradiance data from various height map sources and finally ended up with curves for 20 distinct bounces:



## Effect of Maximum Height over the Curves

Varying the maximum amplitude of the height map didn’t have a lot of influence over the general aspect of the curves except for extrema:

* I noticed that reaching an extremely low amplitude obviously prevented any bounce to occur and the curves had a tendency to flatten.
* Inversely, reaching very high amplitudes prevented light to reach the bottom of the surface, also flattening the curves.
* A good middle ground for these tests is to use height amplitudes that are roughly 1/10 the size of the map (so a 10cm height amplitude for a 1m large texture in my case).

## Effect of the AO Integral Simplification over Actual Direct Illuminance

Let’s show the difference between actual luminance integration:

And the simplified AO model:

Plotting the curve for a *unit* direct radiance , and a constant irradiance value of shows us the approximation is quite good but an additional term is missing:



A good fit is found for the missing irradiance using the new mapping function :

Where:



## Summing the Curves

Interestingly, summing the contribution of all the curves (including direct irradiance) using a surface albedo of 1 yields a constant return of energy equal to π, whatever the AO value:



We could try and fit all the curves individually using some sort of log-normal distribution-looking model:



But at runtime, we will only be given one value of AO and we want to find the irradiance values for each individual bounce.

We need to shift perspective by starting the initial bounce value for a given AO and find the relationship between successive bounces so the irradiance value for the first bounce gives us the 2nd bounce, the 2nd bounce gives us the 3rd and so on.  
Moreover, we are helped by the fact that the infinite sum of bounce values yields the full initial irradiance value π, *for a unit surface albedo only*.



qsd