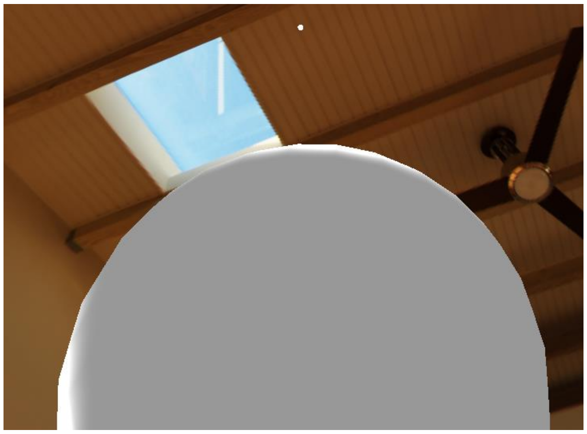
Cody Duncan
CS562
Fall 2014
Instructor: Dr. Gary Herron
Project 3: Physically Based and Image Based Lighting
Report

Contents

Physically Based Lighting using BRDF Lighting Function



Sphere (towards top):

Alpha (roughness): 0.1

White Dot: Directional Spotlight

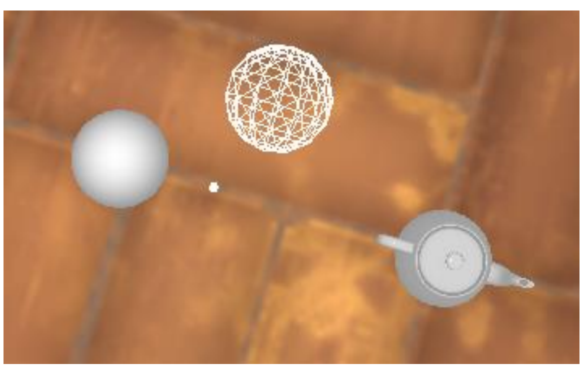
The lighting function is working properly as demonstrated by the specular highlight at the glancing angle to the light.

Bright spot on the left is from a local point light.

Teapot:

Alpha: 20000

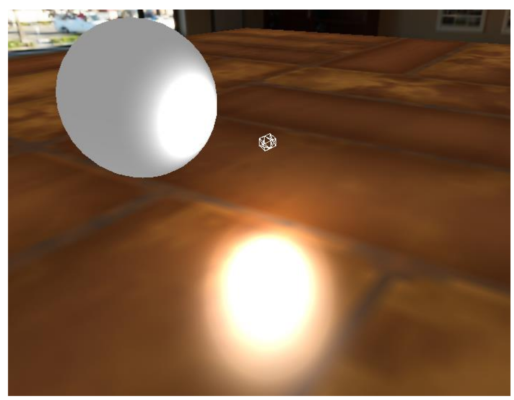
This highly specular object has a small specular highlight warped by the normal of the surface.



Opposing View, From Light's Perspective

White Dot near ground: Point Light (inactive).

From this angle, BRDF diffuses the specular highlight light on the diffuse object.



Point Light on Plane

Plane Alpha: 20.0

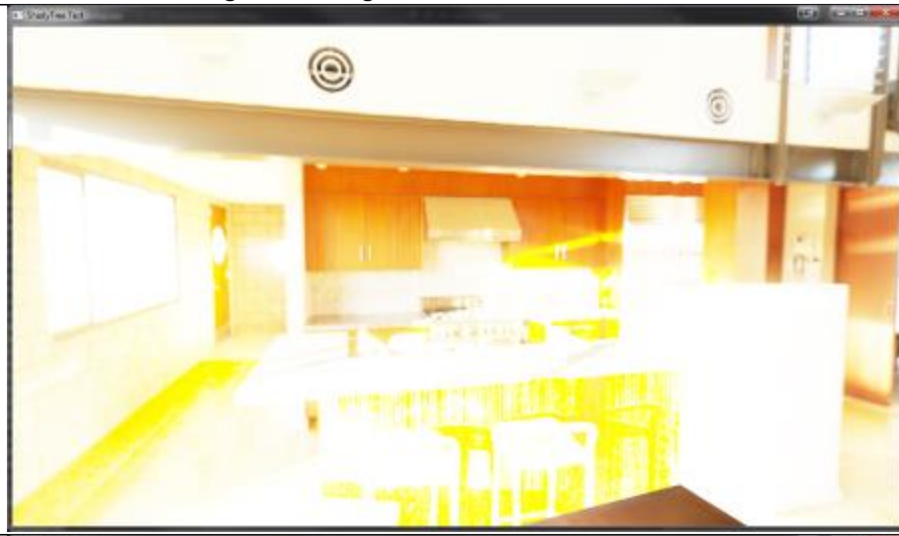
The plane has a "middle ground" alpha value that gives it some specular highlight and some diffusion. This angling shows that the BRDF creating specular highlight at glancing angles, which creates the ovular shape of the light spot.

Image Based Lighting: Diffuse AND Tone Mapping Comparison

Exposure	Non Linear Color Space	Linear Color Space
700		
70		
7		
1		
0.1 (Extra)		

In all cases, performing the calcuations in sRGB color space tends to mute the color saturation a bit due to the exponential factor moving the colors to a brighter, less varied curve.
The linear color space ends up more saturated, as it can express the full range of values without accidentally skipping past parts of the exponential elbow to a lighter or darker color.

Background Image Tone Mapping

Exposure	Environment Background Image.
7000 Perfectly overblown bright colors	
700 More reasonable, like a bright day or bright lights outside	
70 Nice sunny day.	

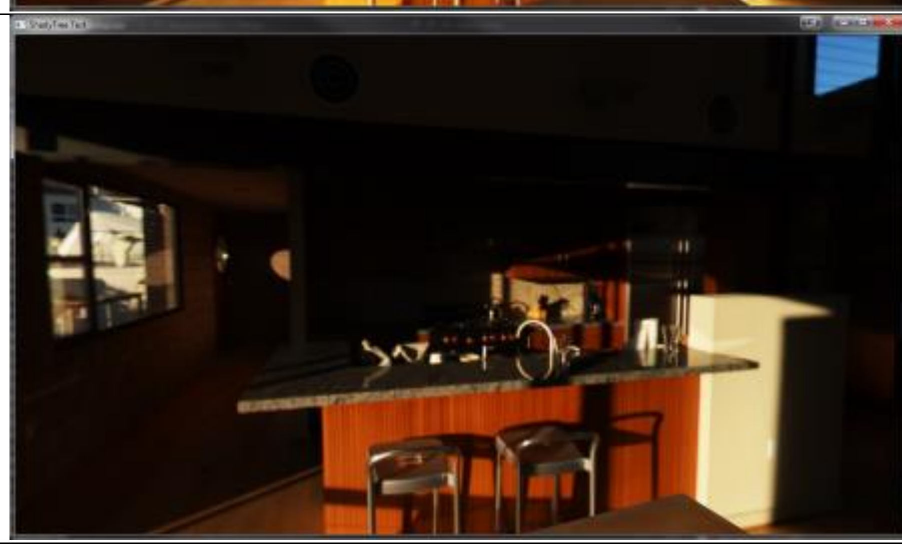
7 More like the environment the image was likely taken in.	
1 Somewhat like twilight, with only bright specular highlights from sun.	
0.1 Overly dark.	

Image Based Lighting: Specular

Alpha: 1 So low, that the sampling only occurs from the lowest mipmap level. All specular applied is just one flat color, hence why this sphere ends up being just a slightly brighter copy of the diffuse IBL sphere.	
Alpha 625 (5^4) By this alpha, mip levels are being sampled that actually have some definition. Note the highlights in the windows and the bright wood kitchen in the lower left.	

Alpha 15625 (5^6) This particular mip map level shows the disadvantage of using mipmaps to blur the ILB specular color sample. The mipmap's low resolution shows through on the window reflecting from the ground plane, while the sphere distorts the effect enough to become unnoticeable.	
Alpha 390625 (5^8) At this alpha, all samples take from the most detailed mipmap, and as a result, the specular color on the objects just looks like a detailed reflection.	

Notes on Implementation

Physically Based Lighting

BRDF function applied to all objects from the point and directional light sources. The distribution term of the BRDF is phong with a roughness parameter. The shadowing term uses the approximation 1/(dot(L,H))^2. The Fresnel term is Schlick's approximation of the Fresnel Equation.

Linear Color Space

Within the shader, values from textures are converted from sRGB to Linear by taking them to the power of 2.2. Computations are done in with linear space values, and returned to sRGB upon output.

Luckily, WIC (windows imaging component) automatically converts texture coordinates to linear space upon load. Only the environment map, which had to be loaded manually, actually needs conversion to linear before usage. All values require conversion back to SRGB on output. Values in the geometry buffer diffuse are in sRGB, so later passes that use it must do the conversion.

Tone Mapping

There are two passes for IBL

- Ambient Pass: Writes out the ambient term.
- Specular Pass: Additively blends the specular value on the ambient.

Both share the same buffer that defines the exposure and contrast values.
For both passes, tone mapping was done by sampling the color from the hdr background image, and then passing it through the equation from the notes:

$$C_{out} = \left(\frac{eC}{(eC + (1,1,1))}\right)^{K\,1/2.2}$$

e = exposure value. (varies)
K = contrast value. (Always 1)

Note that the input color C was converted to linear color space prior to being tone mapped.

Additionally, the skysphere is just given the diffuse value from the geometry buffer in the ambient pass, and ignored in the specular pass. When drawing the skysphere to the geometry buffer, the color is tone-mapped as given above. The skysphere also has its depth set to 0.9999 to keep it within culling distance, but allow it to be elided in the later checks by drawing only diffuse for anything beyond 0.99 depth. For all cases in this scene, it works well.

Image Based Lighting

During the *ambient pass* of IBL, the color from the irrdiance map is sampled and applied as the output color using the equation:

float4 outColor = pow(diffuse, 2.2) / PI * irradiance(normal) * AmbientBrightness;

diffuse: diffuse color of texture/surface from geometry buffer
irradiance: samples the irradiance map by converting normal to UV, performs conversion to linear color space and tone mapping.
AmbientBrightness: 2.0 to perk up the scene brightness a bit, to slightly counteract the division by PI darkening too much.

During the *specular pass*, the shader is passed a collection of pseudo-random pairs of values from a low-discrepancy sequence produced from the Hammersly set code in the project document. These are transformed to match the distribution (step 2), converted from spherical to Cartesian coordinates (step 3 and equation 4 in project doc), and then rotated to point the z axis of the spherical coordinate toward the reflection direction (step 4). All vectors are normalized going in and out of this computation; thanks for the warning.

40 samples of random vectors, as calculated from the above, are then taken from the environment map. As the document suggests, the values are filtered by using the mipmap, using this equation to determine the mip level to use (the one in the project document):

$$level = \frac{1}{2}\log_2\left(\frac{WIDTH * HEIGHT}{N}\right) - \frac{1}{2}\log_2\left(D(H)\right)$$

The sample from the environment mipmap is then used as the light color in the IBL specular equation:

```
float4 specFinalColor = float4(0,0,0,0);
for( i= 0 to N_Dirs)  //N_Dirs = 40 samples
{
    //compute random direction
    //perform fresnel, shadowing, and distribution functions
    //sample environment map using distribution
    float4 specCalc = (_fresnel * _shadowing) / (4) * env_L_Color * dotNL;
    specFinalColor += specCalc;
}
specFinalColor /= N_Dirs;                    //average the values
specFinalColor = pow(specFinalColor, 1 / 2.2);  //back to sRGB
specFinalColor.w = 1;                        //don't blend alpha
return specFinalColor;
```

Note: omega is the random direction computed from the low discrepancy sequence as described above. It's used in place of L, the direction to the light.

_fresnel: F(omega, H)
_shadowing: 1/pow(dot(omega, H), 2)
env_L_Color: mipmap sample from hdr environment map, in random direction
dotNL: dot(Normal, omega)