Diffuse Cube Mapping

Chris Brennan

Introduction

Cube mapping has become a common environment mapping technique for reflective objects. Typically, a reflection vector is calculated either per-vertex or per-pixel and is then used to index into a cube map that contains a picture of the world surrounding the object being drawn. To date, cube maps have been used primarily for specular effects that are layered on top of per-vertex diffuse lighting or static diffuse light maps. In this chapter, we present diffuse cube maps, which happen to come nearly for free when performing specular or environment mapping with a cube map. The main advantage of diffuse cube mapping is that an unlimited number of lights may be included in a single rendering pass. Diffuse cube maps can also be generated from light probes [Debevec98] acquired from a real environment to provide a diffuse or ambient term to really "place" an object in a scene.

Using Diffuse Cube Maps

Diffuse cube maps are different in usage from reflection maps only in that the surface normal is used to index into the map with the intention that it returns the total amount of light scattered from that direction. The reason it is free to use diffuse cube maps when using reflections is that a normal is required for doing the reflection operation given the normal (N) and eye vector (E):

$$R = 2N(N \cdot E) + E$$

or if the normal is not of unit length:

$$R = 2N(N \cdot E) + E(N \cdot N)$$

Even the unnormalized normal may be used in diffuse cube mapping without any additional pixel or vertex operations. The following is an example of per-pixel environment mapped bump mapping with the diffuse component added in bold:

```
ps.1.4
def c0, 1,1,1,1

texld   r0, t0              ; Look up normal map.
texcrd  r1.xyz, t4          ; Eye vector
texcrd  r4.xyz, t1          ; 1st row of environment matrix
texcrd  r2.xyz, t2          ; 2st row of environment matrix
texcrd  r3.xyz, t3          ; 3rd row of environment matrix
```

```
dp3      r4.x, r4, r0_bx2        ; N.x = 1st row of matrix multiply
dp3      r4.y, r2, r0_bx2        ; N.y = 2nd row of matrix multiply
dp3      r4.z, r3, r0_bx2        ; N.z = 3rd row of matrix multiply
dp3_x2   r3.xyz, r4, r1          ; 2(N.Eye)
mul      r3.xyz, r4, r3          ; 2N(N.Eye)
dp3      r2.xyz, r4, r4          ; N.N
mad      r2.xyz, -r1, r2, r3     ; 2N(N.Eye) - Eye(N.N)

phase

texld    r2, r2              ; Sample cubic reflection map
texld    r3, t0              ; Sample base map with gloss in alpha
texld    r4, r4              ; Sample cubic diffuse map

mul      r1.rgb, r3.a, r2    ; Specular = Gloss*Reflection
mad      r0.rgb, r3, r4, r1  ; Base*Diffuse + Specular
+mov     r0.a, c0.a          ; Put 1.0 in alpha
```

The normal is calculated by looking up a normal map. However, the normal is in tangent space, which needs to be transformed into world space through a matrix multiply of the concatenation of the texture tangent matrix and the world matrix and stored in r4, which is used when sampling the diffuse map in the next phase. So with the addition of only one texture fetch and one instruction, an arbitrarily complex diffuse lighting component can be added.

Generating Dynamic Diffuse Cube Maps

The idea is that a cube map is generated dynamically every frame in the same way that a reflection cube map is generated, but instead of drawing actual world geometry from the point of view of the object into the six faces, a diffusely lit hemisphere around the object is accumulated into the cube map for each light. The following figure shows the corresponding diffuse and reflection maps for one light.

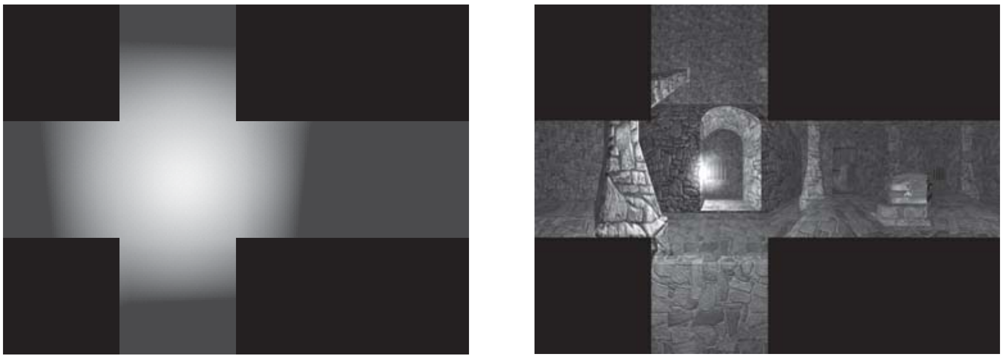

Figure 1: Diffuse cube map (right) and reflection cube map (left) for one light

So for four lights, the process would be to clear the cube map to the ambient color, and then for each light, add in its light component as shown below and in Color Plate 3.

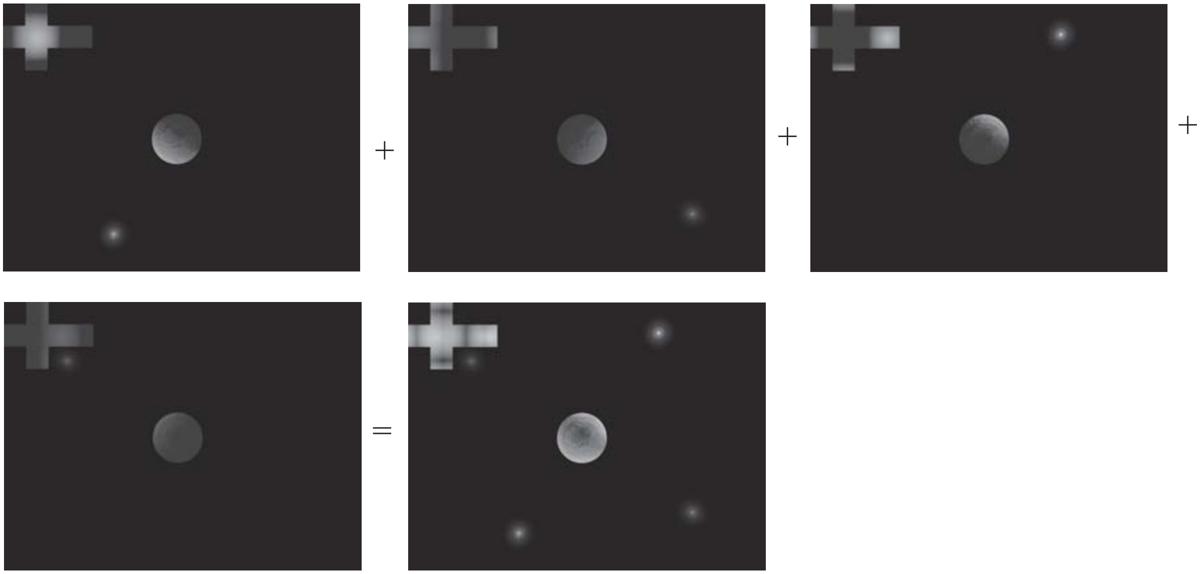

Figure 2: The accumulation of lights when generating a diffuse cube map

The resulting cube map can then be used to account for all light in the scene with one texture lookup. Generally, the cube maps do not need to be very large. 32x32 for each face is sufficient for not having any artifacts. Therefore, the cost of rendering a large number of lights into a cube map is much less than rendering many light passes per object when rendering the final image. This property makes diffuse cube mapping a logical choice when an object can be contained in an environment map and the lighting is complex or a reflection environment map is already being used.

This technique is used in the Treasure Chest demo available on the companion CD included in this book and also at http://www.ati.com/na/pages/resource_center/dev_rel/Treasurechest.html. There are menus to view the diffuse cube map that can be accessed after going to windowed mode by pressing Alt+Enter.

References

[Debevec98] Paul E. Debevec, "Rendering Synthetic Objects into Real Scenes: Bridging Traditional and Image-Based Graphics with Global Illumination and High Dynamic Range Photography," (SIGGRAPH Proceedings, July 1998).