Fully Deferred Rendering Pipeline

Houman Meshkin
houman@houmany.com

1. Introduction

Deferred shading has several advantages over forward shading, allowing scenes to be rendered with numerous dynamic lights in real-time. However, it also has some limitations.

One limitation is that translucent objects aren't handled well by deferred shading, and are typically rendered using forward shading. This adds complexity, as the engine must maintain both deferred shading and forward shading pipelines. Furthermore, forward shaders tend to support a limited number of dynamic lights, potentially causing translucent objects to light differently from deferred lit objects.

Another drawback with deferred shading is that the material shading properties are limited by the number of components stored in the g-buffer. In this talk, we introduce methods to address both of these limitations, making deferred shading a more robust solution.

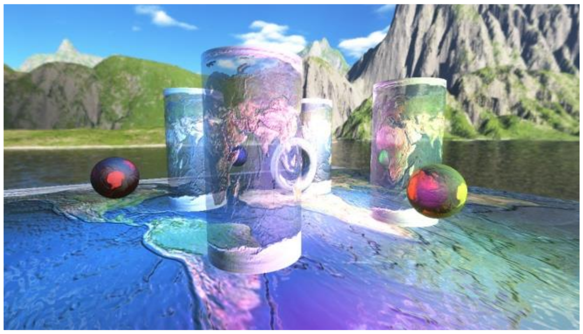

Figure 1. *Deferred Translucency.*

2. Deferred Translucency Rendering

We use deferred shading to render translucent objects, much like opaque objects. After opaque objects have been lit from the g-buffer, translucent objects are rendered into the g-buffer. These translucent objects are rendered back-to-front, fully opaque and with z-buffer writes. Each translucent object that fills the g-buffer is then deferred lit (pixels that it affected are masked in the stencil buffer). The result is then blended into the lit frame buffer of opaque pixels. This process is repeated for each translucent object.

Because translucent objects are rendered opaque in the g-buffer, only the nearest pixel in depth is rendered to the g-buffer. This eliminates any blending between intra-object polygons. In most cases this is actually a desired effect, as concave meshes will often not have their polygons perfectly sorted back-to-front, yielding artifacts otherwise.

3. Arbitrary Materials in the G-Buffer

Materials in a deferred shading pipeline are limited to the g-buffer layout. Deferred lighting (light pre-pass) is an alternative solution to deferred shading, where a separate pass is used to apply the materials. This allows for a variety of materials, but can be costly due to the additional geometry rendering pass.

Like several other engines, we reserve a component of the g-buffer as a material ID. We use it to index into a material texture atlas, stored as a 3D volume texture, where each 2D slice represents a unique material. Each material can be indexed by the N.L and the N.V components, to emulate a subset of BRDF materials. The RGB component of the material texture represents the diffuse component, while alpha is used to scale the specular component, which lets us create Fresnel effects.

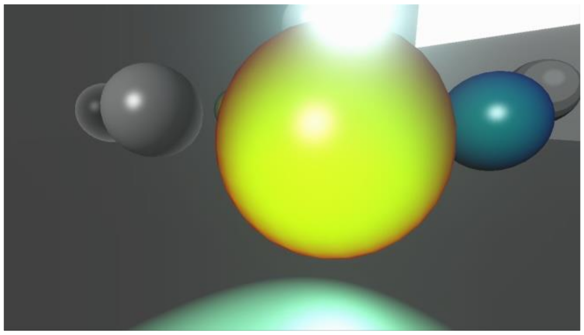

Figure 2. *Iridescent material with back-lighting.*

3.1 BRDF Materials

With our material textures, we can create a variety of materials, including iridescence, subsurface scattering, semi-transparent objects with back-lighting, and even toon shading with outlines. We can also modify our texture atlas to encode BRDF materials using a separable approximation. A BRDF could be approximated by 2 textures, both of which can be represented in a single slice of our volume texture.

4. Conclusion

Our material solution is very fast even on today's video game consoles. Using a 64x32x256 32bpp volume texture costs ~0.5ms on the Xbox 360 GPU. This can be further optimized by creating mipmaps of the volume texture, and using depth to bias the mipmaps.

Deferred translucency rendering is also fast on modern GPUs. Unfortunately, it's not trivial on the Xbox 360, due to the EDRAM limitations. There are many opportunities to optimize both methods, to produce a highly efficient and fully deferred rendering pipeline.

References

SAITO, T., TAKAHASHI, T., 1990. *Comprehensible Rendering of 3-D Shapes*. ACM SIGGRAPH Computer Graphics (ACM Press).

KAUTZ, J., AND MCCOOL, M., 1999. *Interactive Rendering with Arbitrary BRDFs using Separable Approximations*. 10th Eurographics Rendering Workshop.

ENGEL, W., 2009. *ShaderX7: Advanced Rendering Techniques*. Charles River Media.