Forward+: Bringing Deferred Lighting to the Next Level

Takahiro Harada, Jay McKee, and Jason C.Yang

Advanced Micro Devices, Inc.

Abstract

This paper presents Forward+, a method of rendering many lights by culling and storing only lights that contribute to the pixel. Forward+ is an extension to traditional forward rendering. Light culling, implemented using the compute capability of the GPU, is added to the pipeline to create lists of lights; that list is passed to the final rendering shader, which can access all information about the lights. Although Forward+ increases workload to the final shader, it theoretically requires less memory traffic compared to compute-based deferred lighting. Furthermore, it removes the major drawback of deferred techniques, which is a restriction of materials and lighting models. Experiments are performed to compare the performance of Forward+ and deferred lighting.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture

1. Introduction

In recent years, deferred rendering has gained in popularity for rendering in real time, especially in games. The major advantages of deferred techniques are the ability to use many lights, decoupling of lighting from geometry complexity, and manageable shader combinations. However, deferred techniques have disadvantages such as limited material variety, higher memory and bandwidth requirements, handling of transparent objects, and lack of hardware anti-aliasing support [Kap10]. Material variety is critical to achieving realistic shading results, which is not a problem for forward rendering. However, forward rendering normally requires setting a small fixed number of lights to limit the potential explosion of shader permutations and needs CPU management of the lights and objects. Also, with expensive dynamic branching performance on current consoles (e.g., XBox 360) it is understandable why deferred rendering has become appealing.

The latest GPUs have improved performance, more ALU power and flexibility, and the ability to perform general computation – in contrast to current consoles. Thus, rendering with many lights with forward rendering could be a realistic option, however the naïve approach of iterating through every light in a per-pixel shading fashion is impractical.

We present Forward+: a method of rendering with many lights by culling and storing only lights that contribute to the pixel. The lights are evaluated one by one in the final shader.

Figure 1: *A screenshot from the AMD Leo demo using Forward+.*

In this manner, we retain all the positive aspects of forward rendering and gain the ability to render with lots of lights.

This paper first presents the pipeline and gives a high level explanation of the implementation. The theoretical memory traffic of Forward+ is compared to deferred lighting.

2. Related Work

Forward rendering has practical limitations on the number of lights that can be used when shading [AMHH08]. De-