Lighting the Open World of New York Zero for Prototype 2

Keith O'Conor
Radical Entertainment

Josh Blommestein
Radical Entertainment

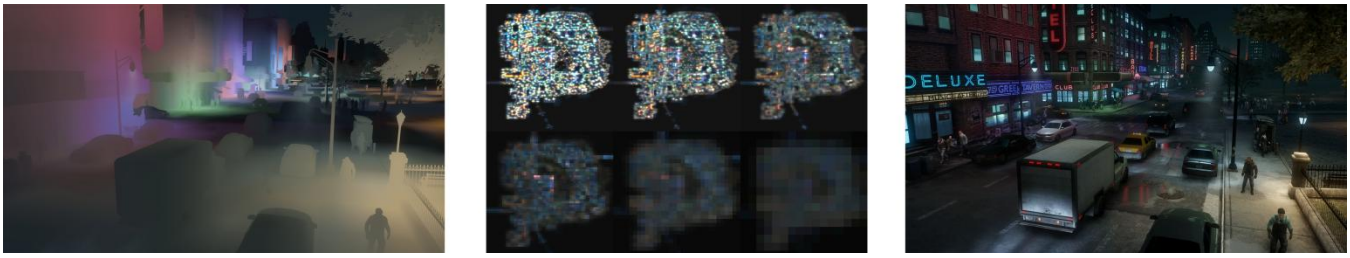

***Figure 1**: (a) Indirect lighting component of a scene, (b) mipmaps of the lightmap representing vertical sections of the world, and (c) the final image*

1. Overview

Prototype 2 is an open-world action game that gives the player superhuman abilities to move anywhere in the world - including the ability to run at speed up the side of a building, leap off the rooftop, and glide across the city. This freedom presented a variety of technical challenges, such as the question of how to get indirect lighting for both static and dynamic parts of a world where verticality and space plays such an important role. We discuss our solution - a novel pseudo-volumetric light baking process which packs vertical slices of the world into different mipmaps of a lightmap - and how it integrates with the art authoring pipeline and the rest of the rendering engine.

2. Lighting workflow

Each of the 3 zones in the game world have multiple times of day (dawn, night etc.). Each time of day is authored by lighting artists who have control over certain aspects of the direct lighting - sunlight direction/colour, local light placement in the world, material specular properties, environmental effects and so on. All lighting is done directly in the game engine (augmented with lighting-specific authoring tools), for immediate and accurate feedback of how the final in-game lighting looks.

2.1 Volumetric lightmap generation

These light locations and properties are then used for generating indirect lighting, which takes place offline. 3Delight is fed these values along with the world geometry, and generates a point cloud of the indirect bounced light of the entire zone. This is sampled at multiple height levels (a total of 9 are generated), and for each level a top-down projection of the zone is rendered that represents the indirect bounced light at that height. The first few levels are close to ground level and only a few meters apart, to give more coverage to the areas with the most variation in bounced light. At upper levels the sample heights are more spaced out, to give more coverage of the higher elevations of the world. All levels are then are used to generate a texture, with each height level being a lower mipmap in the texture - see Figure 1(b)**.** Each texel in this texture represents the diffuse colour of the average directional irradiance at that point.

2.2 Pipeline integration

This entire process is fully automated, to be as transparent as possible for world artists and others who are not directly involved in lighting. Upon any change in world geometry or lighting that is committed to source control, the content pipeline gathers the lighting properties and dispatches a render job to a network of render machines. They generate new light maps for every affected time of day and check in the updated textures.

For the lighting artists there is also the ability to kick off a network render job with their local changes. This gives them the ability to refine and iterate on the lighting of a specific area with a faster turnaround time.

3. Engine integration

To retrieve the lighting value at runtime, all the engine has to do is convert a world-space position into a texture coordinate that will sample from the correct location and mipmap in this texture. Trilinear filtering will produce a continuous indirect lighting component for any position in the world. This value is used as the ambient component of the lighting equation and combined with the our direct lighting, shadowing, SSAO and other terms.

While the main use of this system is for indirect lighting, the same system has also been employed to generate volumetric ambient occlusion maps to great effect, giving more depth to the avenues and alleys of the city. We have even used it to generate a height map of the highest points of the world, which is used to turn off rain effects in underpasses and other sheltered locations.

4. Results

The use of this system for indirect lighting and ambient occlusion has a number of advantages. The memory usage is minimal - a single 512x512 mipmapped texture is all that is needed for an entire zone, much less than an alternative solution based on traditional lightmapping or spherical harmonics. Artists don't need to care about problems like UV unwrapping, as it just works transparently for them. It is also cheap to use - calculating the indirect lighting at any point is as simple as performing a few operations and a single trilinear texture fetch. This means it can be applied to everything in the world, from static geometry to dynamic props and characters (including LODs), and even particle effects and raindrops, with consistent and continuous results.

Most importantly for the *Prototype 2* universe, the result is indirect lighting that can be sampled at any point in the world, regardless of whether it's on the ground, running up a skyscraper, or gliding through the air.

SIGGRAPH 2012, Los Angeles, California, August 5 – 9, 2012.
ISBN 978-1-4503-1435-0/12/0008