# Level-of-Detail and Streaming Optimized Irradiance Normal Mapping

Ralf Habel, Anders Nilsson and Michael Wimmer

# Demo Documentation

## Introduction

This demo is an implementation of the GPU Pro 2 article “Level-of-Detail and Streaming Optimized Irradiance Normal Mapping”. It shows the usage of the modified H-basis in different lighting conditions including extreme conditions where compression and artifacts occur if texture compression is used.

The textures have been pre-computed using Turtle 5.1 (www.illuminatelabs.com) and the Lua script accompanying the demo using image based lighting and final gather.

## Navigation

WASD Move camera
Mouse Rotate camera

## Keys

F1 Toggle help
Displays the help menu

F2 Toggle statistics
Shows scene statistics (frames per second, triangle count, batch count)

Tab Toggle albedo
Toggles the albedo texture on and off. The off state uses white as albedo to show the effects of irradiance normal mapping without albedo multiplication.

T Cycle texture filtering
Switches between anisotropic, trilinear and bilinear filtering of all textures.

R Cycle rendering mode
Switches between full, wire frame and point rendering

U Cycle texture maps
Switches between uncompressed 8-bit, DXT1 compressed and 16bit EXR irradiance normal maps.

H Toggle tonemapping
Toggle Reinhard tone mapper.

1,2,3 Switches accuracy levels using the full quadratic modified H-Basis (1), linear H-Basis (2) and constant H-Basis (3) which is also standard light mapping.

4 A levels-of-detail blend between the linear modified H-Basis and the constant modified H-Basis (light mapping) is performed.

# Scene Description

## Geometry

The demo is implemented in OGRE 1.6.5 and uses the ogremax (www.ogremax.com) scene converter and OGRE importer to import the geometry from Maya into OGRE. Only the geometry and two UV spaces are imported into the graphics engine. The complete scene consists of 18299 triangles.

## Texture Formats

All textures contain the same data, the PNG files contain the standard 8-bit representation, the DDS files contain the same data DXT1 compressed and the EXR files contain a 16-bit representation. The EXR constant terms contain the data in sRGB similar to the 8-bit representation to avoid separate shaders. There is no visual difference between linear and sRGB EXR files.

## Texture Size

All texture resolutions have been chosen to be small to mimic the situation of a real game. The highest resolution is 256x256 for the terrain, hut and cave. All irradiance normal maps are baked separately so smaller resolutions can be experimented with by simply re-scaling the textures.

## Texture Artifacts

The EXR textures provide an artifact free reference solution to which the other options can be compared to. To see the artifacts, it is best to switch off albedo mapping (TAB). Using an 8-bit representation works in nearly all lighting conditions with minor artifacts that are obfuscated by the albedo mapping. DXT compression works well in natural lighting with acceptable artifacts. By switching to DXT compressed textures, the extreme lighting situation that occurs in the cave, DXT compression artifacts become evident. Parts of the cave wall are lit with a light at grazing angles from the artifact and some spilled light from the entrance. Both light contributions are very small so some of the modified H-Basis coefficients become very small as well.

In these cases, a per-texture normalization instead of the global normalization [-3/4..3/4] used in the demo can remedy those artifacts to an acceptable level. Please see the GPUPro 2 article for further information.

# Code Description

The demo is based on OGRE 1.6.5 and ogremax and uses it without modification. The demo can be compiled with the OGRE source code without modification by copying the folder /GPUPro to the /ogre folder.

## Irradiance Normal Maps

The irradiance normal maps in there different representations are located in /GPUProMedia/GPUProScene/IrrMaps.

## Shaders

All shaders are located in /GPUProMedia/Materials/IrrNorm.hlsl with the accompanying OGRE material definition file IrrNorm.material. This file contains different levels-of-detail version of the shader. The shaders and material definitions can be used in any OGRE application.

## Linear Workflow

The demo implements a linear workflow, meaning that all albedo textures are looked up in sRGB (a gamma of 2.2) while the rest of the data (normal maps, irradiance maps) is interpreted as linear data. An exception is the constant term of the H- Basis which is also the light map so it is transported in sRGB to make optimal use of the of the used bytes and also requires a gamma of 2.2. The sRGB lookups should be done using the sampler state (set D3DSAMP_SRGBTEXTURE to true), the demo implementation applies a gamma directly in the shader because OGRE 1.6.5 does not support sRGB texture lookups.

For a correct linear workflow, the resulting image must be written to a sRGB framebuffer. This is achieved by setting the sRGB Gamma Conversion option to “Yes” in the OGRE Rendering Engine Setup at the beginning. This option must be set to “Yes” for the demo to be rendered correctly. In DirectX, a frame buffer can be defined to be in sRGB by using D3DRS_SRGBWRITEENABLE.

A traditional non-gamma corrected implementation needs to use a gamma correction of 1/2.2 after the H-Basis has been evaluated to put the result into sRGB space in which is can be mixed with other calculations. It is recommended to use a linear workflow if it is possible due to the fact that the H-Basis represents a physically correct representation of the directional irradiance it describes. A linear workflow also makes other calculations such as dynamic diffuse and specular shading correct and thus more realistic.

For more information please visit:

http://blog.illuminatelabs.com/2010/03/gamma-correction.html

## Levels of Detail Blending

The demo uses a linear blend defined by the distance of the objects. The blending distances are chosen to be very close to the camera in order to clearly see the effect. The LOD distances can be set in the IrrNorm shader and material files.

## Pre-Computation Using Turtle

The folder /Turtle contains a Lua script that can be used in the renderer/baker Turtle 5.1. The script can also be used with the trial version of Turtle. The script can be loaded using the Baking->Outputs->Lua option. An example scene for Maya 2010 using Turtle fully set up to bake out irradiance normal maps is located in the same folder (please adjust folders in Maya). The example character can be used freely for commercial and non-commercial uses.

Several Tutorials are available on Illuminate Labs web site:

http://www.illuminatelabs.com/support/tutorial-folder/

For further inquiries please contact ralf@cg.tuwien.ac.at