Chapter 1

Validation

1.1 Outline of Expected Results

The expected results of my solution for dense, texture-mapped PRT lighting are images with global lighting style effects, produced in real-time. As components of this, the system must preprocess meshes to produce textures to utilize in the rendering stage. Because the produced textures are very important to the quality of the resulting images, they will be the first artifact considered for each testing setup. After loading those resources into the rendering part of the solution, the system can produce the images which can be used to evaluate it. That component of the system can be evaluated based on the quality of the resulting images, and the speed with which they are produced.

1.2 Hypothesis

With this work, I hope that the rendering speed of my approach, when compared to other rendering methods which produce similar quality images, will be faster. Further, I hope that the data size required to render a mesh using the system will be comparable to that required for rendering using the other methods.

1.3 Independent and Dependent Variables

1.3.1 Independent Variables

Spherical Harmonic Bands The number of Spherical Harmonic bands used in the representation of each texel’s lighting function varies the specificity of the SH model in representing the surrounding space. A low number of SH bands provides poorer resolution for incoming directional light than a higher number of bands does. Increasing this value induces an increase in the number of coefficients needed to represent the SH model.


**Texture Resolution**  The size of the texture used to represent the SH lighting over the object controls how many lighting samples are taken across the object. A higher resolution texture allows more samples to be utilized during the rendering of the object, resulting in a more faithful recreation of smooth lighting across the mesh.

### 1.3.2 Dependent Variables

**Framerate**  Increasing the sampling density (via the texture resolution) and the number of coefficients in the SH representation (via the SH bands) increases the amount of work needed to render the mesh. Concordantly, the number of times that work can be performed per second drops. To make the system practical in a real-time context, it would have to be able to render a scene with many meshes at a framerate greater than thirty frames per second (30 FPS).

**Data Size**  In addition to increasing the workload of rendering the mesh, increasing either independent variable also raises the amount of data which must be stored for the mesh. Because the rendering of the mesh must be done on the GPU, the data must be transferred across a bus to the GPU and stored. While the onboard memory size for GPU’s is expanding quickly, the frugality of the system with respect to memory footprint is important if many different types of meshes (with different textures and vertex positions) wish to be rendered at once.

### 1.4 Measures

**Framerate**  The framerate can be calculated by counting the number of rendering passes the system makes it through in a second.

**Data Size**  The memory footprint of a mesh can be measured by computing the storage required to represent the faces of the mesh, with their texture-coordinates, and the sum of the sizes of the textures.

### 1.5 Experiment Protocol

To conduct the experiments of the system, texture data will be prepared for a few (common) polygonal meshes at varying band sizes and texture resolutions. After the preprocessing has been done for these data sets, the sizes of the textures and polygonal mesh data will be computed. Finally, each data set will be rendered and have its frame-rate measured. During the frame-rate measuring process, the subjective quality of each data-set will be evaluated, with preference to observations about the global-illumination characteristics that it reproduces.
To compare my validity of my approach against other methods, the same meshes will be prepared and rendered using those methods. The data size required to render them will be computed (including both mesh data and any texture data those methods require), and the framerate at which the meshes are rendered will be evaluated.