UNIVERSITY OF CALIFORNIA, SAN DIEGO

Higher Order Light Propagation Volumes

A Thesis submitted in partial satisfaction of the
requirements for the degree
Master of Science

in

Computer Science

by

Timothy Ly Martin

Committee in charge:

Professor Henrik Wann Jensen, Chair
Professor David Kriegman
Professor Jurgen Schulze

2012

The Thesis of Timothy Ly Martin is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____
Chair

University of California, San Diego

2012

DEDICATION

To my father, for his support, advice, and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I would like to acknowledge Professor Henrik Wann Jensen as the chair of my committee. His advice and guidance have been instrumental as I worked on my research and completed my thesis.

I also would like to acknowledge Wolfgang Engel. The discussions with Wolfgang helped improve my understanding of concepts as I was beginning my research.

ABSTRACT OF THE THESIS

Higher Order Light Propagation Volumes

by

Timothy Ly Martin

Master of Science in Computer Science

University of California, San Diego, 2012

Professor Henrik Wann Jensen, Chair

Light propagation volumes is a real-time single bounce indirect lighting approximation that uses spherical harmonics for light representation and a grid-based light transport. The technique is useful for applying indirect light to participating media, glossy specular reflection, and indirect diffuse illumination, but this thesis focuses on the indirect diffuse illumination.

When applying indirect diffuse light, the light propagation volume method has issues with light bleeding, self lighting, and surface lighting artifacts. Also,

when the indirect diffuse lighting is applied to more complex geometry, there is a lack of directional influence in the shading, producing a flat lighting effect, uncharacteristic of diffuse shading.

I modify the light propagation volume approach by extending the spherical harmonics from two bands to three bands and describe a higher order propagation approach that considers 26 directions of propagation versus the original six. The new resulting technique is called higher order light propagation volumes.

Higher order light propagation volumes are able to combat some of the problems present with light propagation volumes. The new technique reduces the appearance of lighting artifacts, and at the same time promotes a light distribution in scenes that resembles reference implementations, such as path tracing. Lastly, indirect diffuse illumination applied with the new technique produces higher quality diffuse shading detail because the directional representation of light is better preserved with the combination of extended spherical harmonics light representation and the higher order propagation method.

Chapter 1

Introduction

1.1 Introduction

One goal of real-time graphics rendering research is to maximize how realistic a rendered scene is while preserving a reasonable interactive rate so the user may have a pleasant experience. Although the effect of indirect light in a scene may be subtle, it is this subtle effect that can vastly improve the realism of a rendered scene. Most real-time graphics applications take into account direct illumination with shadows, while faking global illumination in a couple of different ways; however, with continuing research and the increasing computing power of GPUs, more realistic global illumination techniques have been made possible while trying to provide acceptable performance. These new techniques are making their way into present day video games and have become another focus of real-time

graphics research.

Some of the cheaper ways to produce faked global illumination is the use of ambient terms. The simplest of these techniques is the constant ambient term that is added on top of simple direct lighting, such as in Phong shading [14]. A more elaborate extension of the constant ambient term is the ambient cube, a technique present in Valve's Source Engine [12]. The ambient cube is merely a set of six color values, which correspond to each face of a cube. Surface normals are used to sample the ambient cube and produce an ambient termed which has been influenced by the surface's orientation.

Another set of techniques stem from ambient occlusion [13]. The ambient occlusion technique produces an occlusion term based on the density of geometry nearby a location. These occlusion terms may be precomputed and stored in an ambient occlusion map, where attenuation terms may be sampled from the map and applied to lighting in real-time. There is also the screen space ambient occlusion technique which produces the ambient occlusion term in real-time using the depth buffer to find the density of nearby geometry [2]. A further extension is screen space directional occlusion, which uses the screen space depth buffer with a screen space normal buffer to produce an ambient occlusion term that also considers the orientation of nearby geometry while producing the occlusion term [16]. A screen space color buffer may also be used to simulate a local indirect light bounce. Although these ambient occlusion based techniques run in real-time

and can produce artistically pleasing results, they lack realism because they do not take into account the position of lights.

Another set of techniques uses reflective shadow maps [4]. Reflective shadow maps are the combination of a depth, normal, and color buffers rendered from the position of a light. These three buffers have the necessary data to create virtual point lights with a directional component. The reflective shadow maps may be sampled for a set of virtual points lights to represent indirect light sources, which result from direct lighting. These virtual point lights may be rendered as another set of point lights in order to simulate indirect light in the scene. This technique is limited because while a large amount of virtual point lights is desirable for a higher quality effect, it is proportionally detrimental to performance. Another technique, called Splatting Indirect Illumination, uses reflective shadow maps with a deferred rendering engine [5], because deferred rendering engines are designed to handle a large amount of lights.

A technique that relies on precomputation is precomputed radiance transfer [18]. Precomputed radiance transfer is the precomputation of global illumination effects where the precomputed lighting data is stored using spherical harmonics. Spherical harmonic coefficients can represent a large amount of light information which may be accessed with inexpensive operations. The precomputed light maps are used to add indirect lighting in real-time. Precomputation of lighting allows for more complex and detailed lighting effects, but with a drawback, indirect

lighting is static.

1.2 Light Propagation Volumes

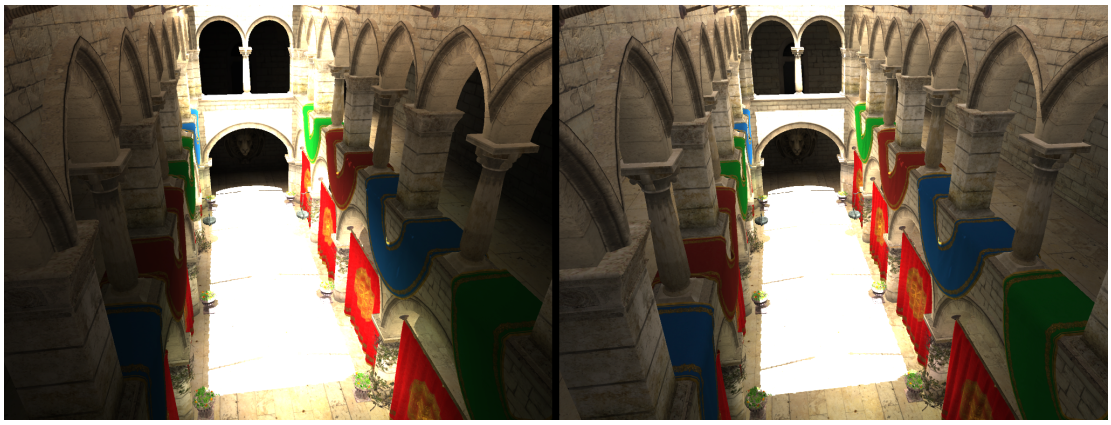

Figure 1.1: A set of example images rendered using the publicly available implementation of light propagation volumes by NVIDIA. The left image is rendered using light propagation volumes. The right image is rendered on a modified version of the NVIDIA implementation, which uses higher order light propagation volumes, the proposed technique of this thesis.

The focus of this thesis will be on the technique known as light propagation volumes(LPVs) [8, 9]. LPVs are used for single bounce indirect lighting with fuzzy occlusion and can be implemented and integrated with today's standard graphics engine setups. The technique first involves collecting light data using reflective shadow maps. This light data is injected into a grid structure called a light propagation volume, where light is represented using spherical harmonics. Geometry data, using normal and depth data obtained from reflective shadow maps and depth peeling, is injected into another grid structure called the geometry volume, where geometry data is also represented using spherical harmonics. Light is prop-

agated iteratively within the light propagation volume while using the geometry volume to account for fuzzy occlusion. After propagation, indirect lighting may be applied by sampling the resulting LPV at the corresponding rendered position for spherical harmonic coefficients which represent the local distribution of indirect light and evaluating the coefficients for an indirect lighting intensity value.

1.2.1 Shortcomings

Some of the shortcomings of light propagation volumes include: low frequency lighting due to spherical harmonics and coarse grid structure used for light transport, grid-like lighting artifacts due to the grid-based propagation approach, light bleeding, and limited occlusion capabilities. I will explain these shortcomings with more detail below.

The use of spherical harmonics for light representation paired with a coarse grid-based light transport suggests that LPVs will only effectively work with low frequency indirect diffuse lighting. Also, the use of only four spherical harmonic coefficients limits the directional representation of light.

Grid-like lighting artifacts are present in the applied indirect lighting as a result of using grid-based light transport. Wherever indirect lighting is applied, lighting artifacts may be present where there isn't a smooth transition of indirect lighting intensity across neighboring LPV grid cells, giving the indirect lighting an unrealistic grid-like appearance. Light bleeding is also a problem because occlusion

may only be applied to light energy that is propagated from one grid cell into another.

Occlusion is limited, and is described to be "fuzzy". The geometry volume's grid cells only hold directional surface information; the positional information of surfaces is only used to locate the grid cell in which they reside. The applied occlusion or attenuation of propagated light is based on the amount of injected geometry with an orientation that opposes the direction of light propagation, with more attenuation being applied when there is a larger amount of these occluders.

1.3 Objective

The objective of this thesis is to first evaluate the LPVs technique by examining the previously mentioned shortcomings. Next, I extend the spherical harmonic light representation from two bands to three bands, which will improve directional light representation and provide a more accurate representation of indirect diffuse light. Because the six direction propagation approach is not able to use all of the band two spherical harmonic coefficients, I also provide a new propagation approach. The new propagation approach not only makes use of the additional spherical harmonic coefficients, but makes lighting artifacts less apparent and produces indirect lighting that more closely matches, relative to the original LPV technique, the result of reference methods, like path tracing. The resulting combination of these modifications produce a technique that can still be

calculated in real-time, and I call the new technique higher order light propagation volumes(HOLPVs).

1.4 Overview

- Chapter 2 explains concepts related to global illumination, that are also relevant to light propagation volumes. The chapter also describes how light propagation volumes serves as a global illumination approximation.
- In Chapter 3, I will explain the light propagation volume technique.
- In Chapter 4, I explain the modifications to the light propagation volume technique that the new technique, higher order light propagation volumes, uses.
- Chapter 5 will summarize the results of the test scenes.
- Chapter 6 summarizes the findings and concludes this thesis.

Chapter 2

Global Illumination

2.1 Relevant Concepts

Here are some relevant concepts that are necessary to understand the global illumination problem and how they relate to light propagation volumes (LPVs). The chapter ends by explaining how LPVs serves as an approximation for the global illumination problem.

2.1.1 Solid Angle

A solid angle Ω can be described as the surface area that is covered on the unit sphere when a surface is projected onto the unit sphere. The unit of measurement of solid angles is the steradian (sr). Mathematically, a solid angle can be written in a couple different ways:

- $\Omega \equiv \iint_S \frac{\vec{n} \cdot da}{r^2}$ where $\vec{n}$ is a unit vector from the origin, r is the distance from the origin to the surface, and the integral is over the surface area of the surface that is getting projected onto the unit sphere.

- $d\Omega = \sin\theta d\theta d\phi$ where θ is the colatitude angle and ϕ is the longitude angle. This form serves as a way to perform integration over the differential solid angles of a hemisphere or sphere.

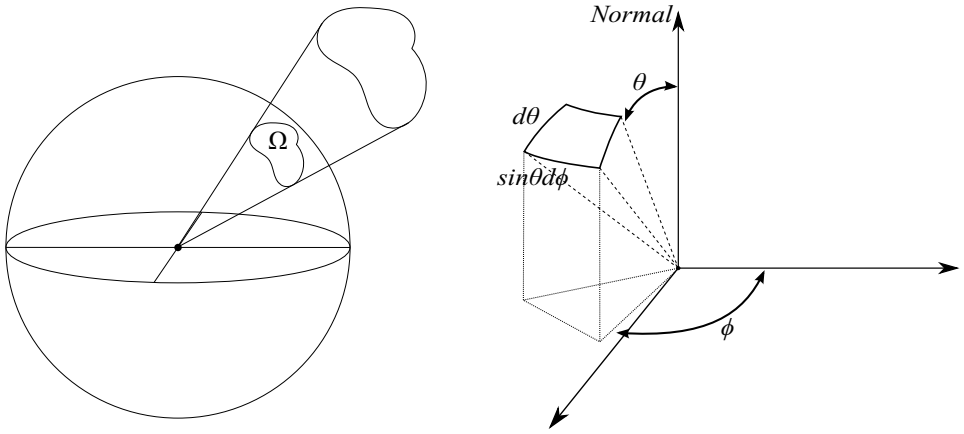

Figure 2.1: On the left is an example of an arbitrary surface with a solid angle that is subtended on the surface of the unit sphere. On the right is a visual representation of integration over a solid angle.

Solid angles are used to define radiometric quantities like radiance. In the case of LPVs, because light propagation is discretized, solid angles must be pre-calculated then used when calculating the propagation of light flux between grid cells. The solid angles represent each discretized propagation quantity from a source cell. The solid angles act as a visibility function for each discretized propagation quantity, propagating only the energy that would reach each destination

grid cell, thus conserving energy. Also, to ensure energy is conserved, the sum total of solid angles used during propagation from a single source cell must be no more than 4π, the solid angle of the entire unit sphere being subtended. More details on the calculation and use of these solid angles are in chapter 3, where the propagation method for LPVs is explained in more detail.

2.1.2 Intensity

Intensity is the measurement of flux density per solid angle and it is measured as watts per unit solid angle (Wsr^{-1}):

$$I = \frac{d\Phi}{d\omega} \tag{2.1}$$

Intensity is an important radiometric quantity because this is how light data is stored in the light propagation volume. The light energy at each LPV grid cell is represented as a point light, and spherical harmonics are used to describe the intensity distribution of the point light at each of these grid cells. Each grid cell will have three sets of spherical harmonic coefficients to represent the distribution of intensity for spectral red, green, and blue channels.

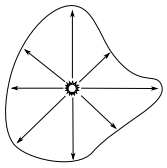

Figure 2.2: A 2D example of a non-uniform intensity distribution from a point light.

2.1.3 Radiance

Radiance is the measurement of flux, or power, per unit solid angle per unit projected area. The unit of measurement of radiance is watt per steradian per square meter ($Wsr^{-1}m^{-2}$). In the following equation, $d\omega$ is the differential solid angle, dA is the projected differential area, and θ is the angle between the surface normal at x and the direction ω.

$$L(x,\omega) = \frac{d^2\Phi(x,\omega)}{dAd\omega cos\theta} \tag{2.2}$$

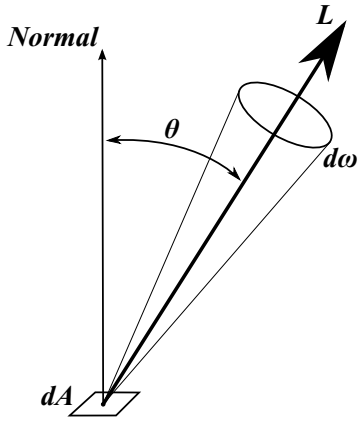

Figure 2.3: A visual representation of Radiance.

Radiance is one of the most important radiometric measurements because when rendering realistic images, the goal is usually to calculate radiance, or to use radiance to calculate other necessary radiometric quantities. Also, when working with radiance, it is important to differentiate between incident and exitant radiance. Incident radiance, denoted $L_i(x,\omega)$, is the incoming radiance to point x, in

the ω direction. Exitant radiance, denoted $L_o(x, \omega)$, is the outgoing radiance from point x in the direction ω.

2.1.4 Bidirectional Reflectance Distribution Function

The bidirectional reflectance distribution function(BRDF) is used for describing the reflectance of a surface. Different surfaces with different reflective properties will have their own BRDFs. BRDFs are used in the rendering equation to find the reflected radiance at surfaces in a scene. Given a point x, incoming direction ω_i, outgoing direction ω_o, and the angle θ between the incoming direction and the surface normal N_x at x, the BRDF f_r may be defined as:

$$f_r(x, \omega_o, \omega_i) = \frac{dL_o(x, \omega_o)}{L_i(x, \omega_i) cos\theta d\omega_i} \tag{2.3}$$

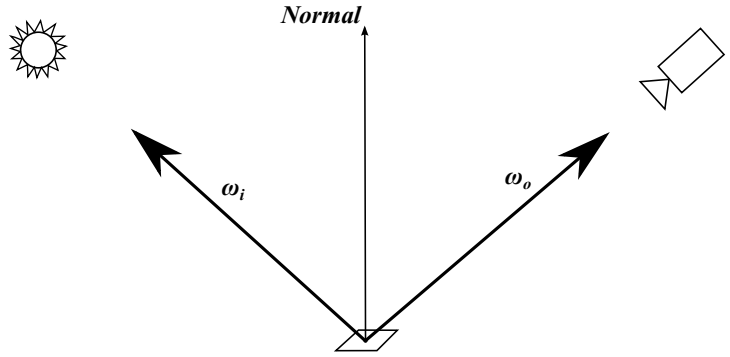

Figure 2.4: The vectors that define the BRDF.

Equation 2.3 shows that the BRDF gives the proportion of reflected radiance in the exitant direction ω_o, for some incident direction ω_i.

For a BRDF to be physically plausible, the following must hold true:

- *Hemholtz Reciprocity:* $f_r(x, \omega_o, \omega_i) = f_r(x, \omega_i, \omega_o)$

  The reflectance of the surface is independent of the travel direction of light, so switching the incoming and outgoing directions generates the same BRDF value.

- *Energy Conservation:* $\int_\Omega f_r(x, \omega_o, \omega_i) \cos\theta d\omega_o \leq 1$

  For any given incoming direction, the total reflected energy over the surface's hemisphere must be less than or equal to the energy received.

2.2 The Rendering Equation

This discussion of global illumination begins by examining the rendering equation that was introduced by Kajiya [7]:

$$L(x, \omega) = L_e(x, \omega) + L_r(x, \omega)$$

$$L(x, \omega) = L_e(x, \omega) + \int_\Omega L(x, \omega') f_r(x, \omega, \omega')(\vec{n} \cdot \omega') d\omega' \tag{2.4}$$

The L_e term is for emissive surfaces, when a light source being visible to the viewer. L_r is for shading a surface. The L_r term found by integrating over the hemisphere Ω at x, gathering all incident radiance, with a cosine term $(\vec{n} \cdot \omega')$ to

take into account the incident angle, and using the BRDF f_r to find the resulting reflected radiance in the direction of the viewer.

The difficulty in calculating this integral is finding the incident radiance, which either comes from a light source or is radiance reflected from another surface. If incident radiance is reflected from another surface in the scene, then $L_r(x', \omega_i)$ must be evaluated, where x' is a surface that is reflecting radiance towards x in the ω_i direction, which involves calculating the rendering equation at x' in the direction of x. To calculate this integral at x', there may yet again be radiance reflecting from another surface, towards x', so the rendering equation appears again. This type of equation is called a *Fredholm equation of the second kind*.

This troublesome pattern repeats infinitely, so in order to compute global illumination without taking an indefinite amount of time, an approximation must be used. There are many approximation approaches, such as path tracing [7], metropolis light transport [19], and photon mapping [6], but for a real-time interactive application, these approaches are too computationally heavy. Most real-time applications only calculate direct lighting, then use techniques that fake global illumination, some of which are described in the introduction chapter. In contrast, LPVs approximates global illumination using direct lighting and a single indirect diffuse bounce, and the next section explains how LPVs accomplishes this approximation.

2.3 Approach by Light Propagation Volumes

According to Kaplanyan [8], it's best to approach the rendering equation by using the Neumann series approximation [7] of the rendering equation. The Neumann series approximation helps explain how the global illumination problem is calculated on a rasterizer-based renderer. In order to understand the approximation, some equations, variables, and linear operators need to be defined [1]:

- $x'(x, \omega)$ maps the point x to a non-local point x' which is found in the ω direction from x.
- G is the field radiance linear operator, defined as $(Gh)(x, \omega) \equiv h(x'(x, \omega), \omega)$, which gives the local incident radiance at x in the direction ω.
- K is the linear local reflection operator, defined as $(Kh)(x, \omega) \equiv \int_\Omega k(x; \omega' \rightarrow \omega) h(x, \omega') d\mu(\omega')$ where $k(x; \omega' \rightarrow \omega)$ is the surface BRDF, $h(x, \omega')$ is incident radiance on x from the ω' direction, and $d\mu(\omega') = \cos\theta' d\omega'$.
- I is the transport intensity.
- e_0 is the emission distribution of primary light sources.

Given these defined variables and linear operators, the rendering equation 2.4 may be re-written as the following:

$$I = Ge_0 + KGI$$

This equation may then be rearranged:

$$(1 - KG)I = Ge_0$$

$$I = (1 - KG)^{-1}Ge_0$$

$$I = \sum_{n=0}^{\infty} G(KG)^n e_0$$

$$I = Ge_0 + GKGe_0 + G(KG)^2 e_0 + ...$$

The first term, Ge_0, of this summation accounts for the case when the light source is visible to the viewer. The second term, $GKGe_0$, is for directly lit surfaces. The third term, $G(KG)^2 e_0$, is for indirect light from a single bounce, and the next term would be for indirect light from two bounces, and so on. For LPVs, global illumination is only approximated for one bounce. After dropping the terms beyond the first bounce of light, terms relevant to LPVs are left, and can be expressed as:

$$I = G_0 e_0 + G_0 K_1 G_1 e_0 + G_0 K_1 G_1 K_2 G_2 e_0$$

$$I = G_0(e_0 + K_1 G_1(e_0 + K_2 G_2 e_0)) \tag{2.5}$$

Equation 2.5 expresses the way the global illumination problem is divided up to be calculated on a rasterizer-based renderer.

The $(e_0 + K_2 G_2 e_0)$ terms are the emission distributions of the direct and indirect light sources, respectively. The e_0 term is taken care of as any simple

direct light source is rendered, and for real-time renderer, these are usually point lights and directional lights. For the next term ($K_2G_2e_0$): G_2 maps the direct light source to the surfaces that are directly lit, and K_2 defines how the direct light interacts with the surface and is reflected. K_2G_2 operate on e_0 to give the emission distribution of indirect light sources. For the light propagation volume approach, indirect light sources are found using reflective shadow maps [9], an efficient way to find the surfaces that are directly lit and gather enough information so virtual point lights may be extracted as used in the next step. A more detailed explanation on the use of reflective shadow maps is in section 3.1.

K_1G_1 operates both on the direct light sources and the indirect light sources. G_1 handles the visibility of direct and indirect light sources to the surfaces that they light, and K_1 is the shading of these surfaces as a result of being directly and/or indirectly lit. With LPVs, direct and indirect lighting is handled separately. Direct lighting is calculated using a simple direct light shading technique, such as diffuse Lambert shading, with shadow mapping for visibility. For indirect lighting, G_1 is handled using the light propagation volume and geometry volume. Indirect light sources sampled from reflective shadow maps are injected into the light propagation volume. After the indirect light sources are initialized in the grid volume, propagation occurs. During propagation, indirect light data is propagated throughout the scene, using the geometry volume to account for occlusion. The resulting volume allows for easy lookup of local incident radiance. K_1 for

indirect lighting is taken care of by using the resulting volume to integrate over the hemisphere, oriented in the direction of the surface normal, to find the total local incident radiance and calculate the reflected radiance as a result of indirect lighting.

The outer G_0 term is for determining what surface locations are visible to the viewer. On a rasterizer-based renderer, this is normally handled with depth buffering [3].

Chapter 3

Light Propagation Volumes

To explain the light propagation volume(LPV) method, I begin by explaining reflective shadow maps and how they are used too generate indirect light source data. Then, I explain how the LPV is initialized with indirect light source data, and the geometry volume(GV) is filled with geometry data. Next, the light propagation process is discussed. Lastly, I elaborate on how the final resulting LPV is used to apply indirect lighting. I end the chapter by going over some of the shortcomings of LPVs.

3.1 Reflective Shadow maps

To understand LPVs, first reflective shadow maps [4] must be understood. Reflective shadow maps are used as a mechanism to gather information on indirect light sources, which will be used to initialize the LPV [9].

Reflective shadow maps extend upon shadow maps [20]. A shadow map is a depth texture, where each depth value represents the distance of the surface at the texel position from the light source position. When applying direct lighting in a scene, a lookup in the shadow map may be performed to check if the current pixel position is farther away from the light than the corresponding texel in the shadow map. If this is the case, then the pixel is in shadow, otherwise the pixel is lit and direct lighting may be calculated.

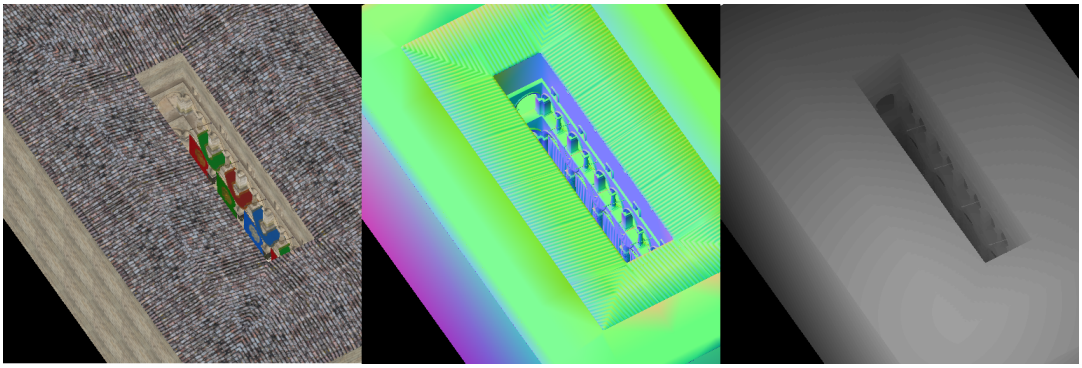

Figure 3.1: An example of a reflective shadow map. Left: surface albedo. Center: surface normals. Right: depth from light position.

Reflective shadow maps, specifically when used in conjunction with LPVs, store depth, reflected flux, and surface normals [9]. The depth value is the same as the depth in a shadow map, and may be used to reconstruct the world position of the texel. The normal is the normal at the surface that each texel corresponds to. The flux value is the reflected flux of the surface. To calculate the flux, the total flux through each texel must be found. For a directional light source, this flux is constant through every texel. For a point light, the flux depends on the solid angle

to the texel from the point light. The stored reflected flux in the reflective shadow maps may be calculated as [8]:

$$Directional\ light:\ \Phi_{rsm} = A_s \Phi_{dirlight}(n_s \cdot l)^+$$

$$Point\ light:\ \Phi_{rsm} = A_s I_s \Omega_s (n_s \cdot l)^+$$

- A_s = surface diffuse albedo
- n_s = surface normal
- l = normalized light vector
- $\Phi_{dirlight}$ = constant flux through texel for directional light
- I_s = point light intensity
- Ω_s = solid angle from point light through texel

The $(...)^+$ operation indicates a quantity that is clamped to positive values only, where negative values are set to zero. The flux (either $\Phi_{dirlight}$ for a directional light source or $I_s\Omega_s$ for a point light source) through the pixel is multiplied by the reflection coefficient of the surface and a cosine term taking into account the incident angle of the light vector to the texel location, and this result is the reflected flux value that is stored in the reflective shadow map.

The combination of the data present for each texel in a reflective shadow map allows each texel to potentially be a virtual point light, an indirect light source. Each virtual point light may have its intensity distribution described as:

$$I(\omega) = \Phi_p(n_p \cdot \omega)^+$$

Φ_p is the reflected flux, n_p is the normal associated with the virtual point light, and ω is the sampling direction that points away from the virtual point light. The cosine term here, which is a positive clamped dot product between the virtual point light normal and sampling direction, gives the virtual point light a hemispherical component, oriented in the direction of the normal, to model the nature of a diffuse reflection.

3.2 Light Propagation Volume and Geometry Volume

LPVs are a uniform grid volume used to represent the distribution of indirect light in the scene [9]. The grid volume is initialized at the beginning of each rendered frame by injecting indirect light data from reflective shadow maps that are rendered for each light. After initialization, light transport is modeled by propagating the collected light to neighboring cells in successive steps. In order to take into account occlusion, the GV is used. Geometry data is injected into the

GV, another three dimensional grid volume with the same resolution as the LPV. During each propagation step, the GV is used to produce an occlusion factor, that is applied to propagated light flux.

Light data and geometry data is represented with spherical harmonics, using the first two bands (a total of four coefficients for each grid cell). To represent red, green, and blue light data, three sets of four coefficients are stored for light, and only one set of four coefficients is stored for geometry.

During the discussion about the LPV and associated spherical harmonic coefficients, only one set of coefficients will be referenced, because the operations are applied to each light channels' coefficients separately.

3.2.1 Injection into LPV

Injection of light data into the LPV involves injecting virtual point lights that are derived from the reflective shadow maps [9]. Each virtual point light is represented as a clamped cosine lobe in its normal direction. Spherical harmonic coefficients can be derived for the clamped cosine lobe along the z-axis using zonal harmonics, then rotated to the appropriate direction [17]. After the rotation, the spherical harmonic coefficients are scaled by the virtual point light's flux value then injected into the LPV at the grid cell that corresponds to the virtual point light location. After the virtual point light is injected into a grid cell, positional information for the light is disregarded.

As mentioned previously, zonal harmonics are used to represent the clamped cosine lobe with spherical harmonics [15]. The reason for using zonal harmonics is because there is a computationally inexpensive way to rotate the cosine lobe to the virtual point light direction for injection [17]. To accomplish this, zonal harmonic coefficients Z_l for each band l must be calculated for the cosine lobe representation. These may be calculated by integrating over a hemisphere, projecting the cosine function onto the $m = 0$ spherical harmonic coefficient for each band [15, 17]:

$$Z_l = \int_0^{2\pi} \int_0^{\pi/2} Y_l^0(\theta) \cos\theta \sin\theta d\theta d\phi \tag{3.1}$$

Using equation 3.1, the cosine lobe zonal harmonics for the first two bands are:

$$Z_0 = \int_0^{2\pi} \int_0^{\pi/2} Y_0^0(\theta) \cos\theta \sin\theta d\theta d\phi = \frac{\sqrt{\pi}}{2}$$
$$Z_1 = \int_0^{2\pi} \int_0^{\pi/2} Y_1^0(\theta) \cos\theta \sin\theta d\theta d\phi = \sqrt{\frac{\pi}{3}}$$

Sloan's paper on spherical harmonics [17] provides the following equation for performing the rotation with zonal harmonics:

$$Y_{rl}^m = \sqrt{\frac{4\pi}{2l+1i}} Z_l Y_l^m(\omega) \tag{3.2}$$

$Y_l^m(\omega)$ is the band l basis m spherical harmonic coefficient in the ω direction. $Y_{rl}^m(\omega)$ is the corresponding rotated spherical harmonic coefficient as a result of the zonal harmonic rotation. Using equation 3.2 and the previously calculated zonal harmonic coefficients for the cosine lobe representation, from equation 3.1,

the equations for each individual spherical harmonic coefficient representing the rotated clamped cosine lobe are as follows:

$$Y_{r0}^{0} = \sqrt{4\pi}\frac{\sqrt{\pi}}{2}Y_{0}^{0}(\omega) = \pi Y_{0}^{0}(\omega)$$
$$Y_{r1}^{-1,0,1} = \sqrt{\frac{4\pi}{3}}\sqrt{\frac{\pi}{3}}Y_{1}^{-1,0,1}(\omega) = \frac{2\pi}{3}Y_{1}^{-1,0,1}(\omega)$$

3.2.2 Injection of Geometry into GV

Depth and normal buffers may be used to inject the GV with data. World space positions may be reconstructed from depth values by applying an inverse projection in order to locate the corresponding GV grid cell. Similarly to light injection, geometry data is injected into the GV using clamped cosine lobes oriented in the direction of the surface normal and represented using spherical harmonics. The spherical harmonic coefficients are also scaled by a blocker potential value, which is calculated based on the world space area the texel of the depth buffer takes up in the scene.

Depth and normal buffer data may come from reflective shadow maps, which get generated for each direct light source to initialize the light propagation volume. Many games use deferred rendering engines which do a geometry pre-pass, generating depth and normal buffers at the beginning of each frame being rendered, which can potentially be another source of geometry data. Depth peeling is another way to gather more depth and normal data, and can be performed from the

light position, to gather geometry data nearby potential indirect light sources, or from the viewer position to gather geometry data near the viewer.

The GV is shifted by half a cell in all directions ($+x$, $+y$, and $+z$) with respect to the LPV. This offset positions the centers of each of the GV's cells on the corners surrounding each LPV grid cell. During light propagation, the corresponding GV grid cells that surround the destination cell are sampled for their spherical harmonic coefficients, and interpolation of these coefficients is used to determine the coefficients representing the blocker geometry at the corresponding cell face which light is propagating through. These interpolated coefficients are evaluated to find an occlusion factor to be applied to the propagated light flux. Figure 3.2 visualizes in 2D the position of the GV with respect to the LPV and the interpolation of a blocker for a LPV grid cell face.

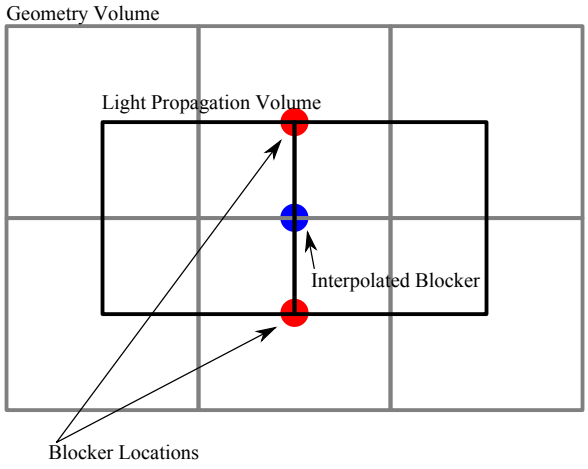

Figure 3.2: The geometry volume is offset by half a cell in each direction resulting in blocker coefficients being located on the corners of each light propagation volume grid cell. In order to account for a blocker when doing a propagation to a neighboring cell, the blocker spherical harmonic coefficients from the corners are bilinearly interpolated to get blocker coefficients for the shared cell face.

3.2.3 Propagation within LPV

Light propagation within the LPV only occurs in six directions, to the neighbors that share a face with the source grid cell. The following explanation will involve propagation to only one neighbor, because propagation to each neighbor will occur in the same way.

The discretization of the light propagation is not only per destination cell, but also per grid cell face in each destination cell. Figure 3.3 gives a 2D visualization of the discretization. For each destination cell, light is propagated to each face that faces away from the source cell.

The solid angle Ω_{vis} from the source cell center to the destination cell face is calculated, then a vector ω_d in the direction of the propagation, from the source cell center to middle of the destination cell face, is used to sample the source cell's intensity $I_{source}(\omega)$. The solid angle is a visibility function that helps ensure only the amount of energy that would reach the destination cell face is propagated. The intensity distribution samples is assumed to be the average intensity over the solid angle of the destination cell face [9]. The calculated propagation flux to a single destination face is:

$$\Phi_{propagate} = \frac{\Omega_{vis}}{4\pi} I_{source}(\omega_d) \tag{3.3}$$

The solid angle for the destination face is normalized by 4π [9], the solid angle subtending the entire unit sphere.

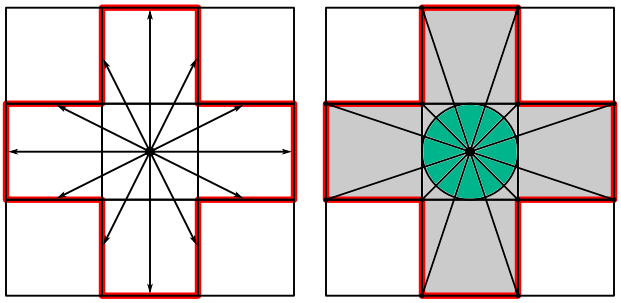

Figure 3.3: Left: A 2D version of the directions of each discretized propagation calculation from the middle source cell to each of the destination cell faces. Right: A 2D version of the corresponding solid angle projections that are used for each of the discretized propagation calculations.

Next, the calculated flux is used to "reproject" [9] a virtual point light in the destination cell. The reprojection process is similar to virtual point light injection during LPV initialization. Spherical harmonics representing a cosine lobe light oriented towards the destination face are generated; these spherical harmonic coefficients are scaled by the propagation flux $\Phi_{propagate}$ calculated using equation 3.3. For each destination cell in the six direction propagation method, this reprojection occurs five times: one propagation to the face furthest away from the source cell, and the other four are to the side faces that are not shared with the source cell.

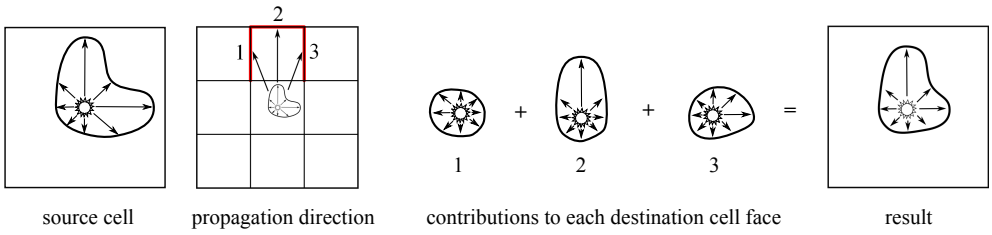

Figure 3.4: A 2D example of the propagation process in one direction. The propagation is calculated by a reprojection of virtual point lights to each visible face of the destination cell. The spherical harmonic coefficients for each reprojected virtual point light is summed together to get the intensity distribution that results from the propagation.

To account for occlusion during propagation, the GV is sampled for the spherical harmonic coefficients that represent geometry at each LPV grid cell corner. When propagating through a grid cell face, the geometry coefficients are bilinearly interpolated for coefficients that represent geometry that resides on that LPV grid cell face. Figure 3.2 visualizes the 2D case of sampling blocker coefficients. These spherical harmonic coefficients for blocker geometry are sampled in the reverse direction of propagation for a blocker term to attenuate the propagation flux. To prevent artifacts from self shadowing, occlusion from the GV is not applied in the first propagation step [9].

3.2.4 Using the resulting LPV for Rendering

Spherical harmonic coefficients C_i are trilinearly interpolated for the location x being rendered. This gives the local incident radiant intensity in the area around x. To find the total incoming flux, an integral over the hemisphere at x must be evaluated. This value of this integral is the dot product of the spherical harmonic coefficients of the local radiance intensity C_i and spherical harmonic coefficients representing a clamped cosine lobe $C_{cos}(\omega)$ in the negative direction of the surface normal n_x at x. This incoming flux is multiplied by the surface albedo A_x and divided by π, to take into account the diffuse BRDF, giving the final reflected intensity I_r at the location x [9, 8]:

$$I_r = (C_i \cdot C_{cos}(n_x))\frac{A_x}{\pi} \tag{3.4}$$

In an attempt to deal with self-illumination and light bleeding, a directional derivative of the spherical harmonic coefficients is determined in the direction of the surface normal, which if large and deviates from the spherical harmonic coefficients that were determined for lighting the surface location, the lighting is dampened [9]. Meeting these conditions means that there is a discontinuity in the propagation of light nearby, and this must be accounted for because the trilinearly interpolated spherical harmonic values do not take into account occlusion, a source of light bleeding.

3.3 Implementation Details

In order to take care of this data representation when programming on a GPU, whether working with OpenGL or DirectX, one full LPV buffer may consist of three four-channel 3D textures for light data and one four-channel 3D texture for the GV.

When the propagation process is programmed to run on the GPU, the propagation steps occur by gathering instead of propagating to the six destination cells [9]. This is easier to implement in shader programming because there will only be one destination render target instead of six. Also, to program the"gathering" version of the propagation process on the GPU, there has to be three separate sets

of LPVs. Two of these LPVs are treated as "ping pong" render targets, where one acts as the read buffer and the other acts as the write buffer and on successive iterations the roles are switched. The third buffer is an accumulation buffer. After each successive light propagation step, the resulting light propagation is accumulated in the accumulation LPV buffer. After all propagation step iterations have finished, the volume used as an accumulation buffer holds the scene's resulting indirect light intensity distribution that will be used for applying indirect light.

3.4 Issues with Light Propagation Volumes

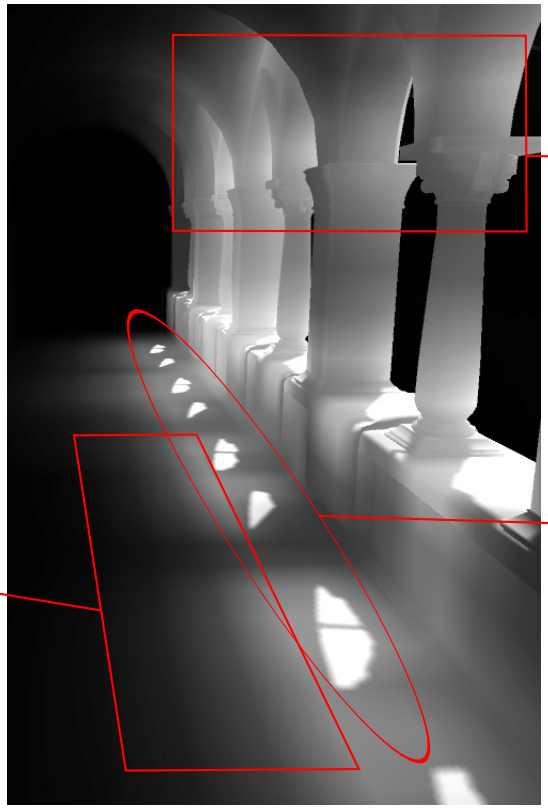

Figure 3.5: An image rendered using a publicly available NVIDIA implementation of light propagation volumes. The image is of a hallway in the Crytek sponza model. The scene is lit with a directional light, and this light reaches the hallway through the windows in the right side of the image. All surfaces in the image are diffuse. Direct lighting is applied with hard shadows, and all other lighting is a result of LPVs. Issues with LPVs are noted in the image.

Grid Lighting Artifacts The grid-like lighting artifacts is due to the use of a grid structure for modeling light transport. Large differences in the lighting distribution in neighboring grid cell causes these noticeable grid-like seams when indirect lighting is applied.

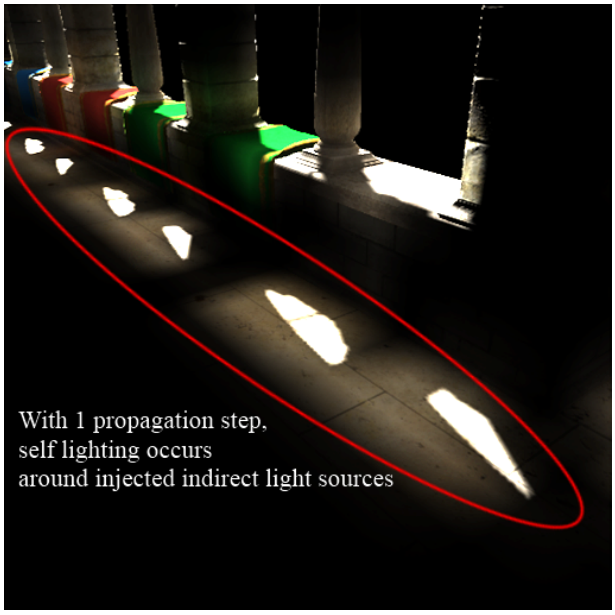

Figure 3.6: An image rendered using a publicly available NVIDIA implementation of light propagation volumes. Only one propagation step is used to show the self lighting problem that occurs near injected indirect light sources.

Self Lighting The first spherical harmonic coefficient is an omni-directional term, so during propagation this band zero coefficient will propagated light flux in all directions. For example, if a grid cell is injected with a single cosine lobe light, during propagation most flux will go in the virtual point light's original direction, but some light flux will be propagated in all directions, causing self lighting artifacts, as seen in figure 3.6. In section 3.2.4, a method using the directional derivative of the spherical harmonic coefficients near the rendered location is used in an attempt to mitigate these self lighting effects, but it is not always reliable.

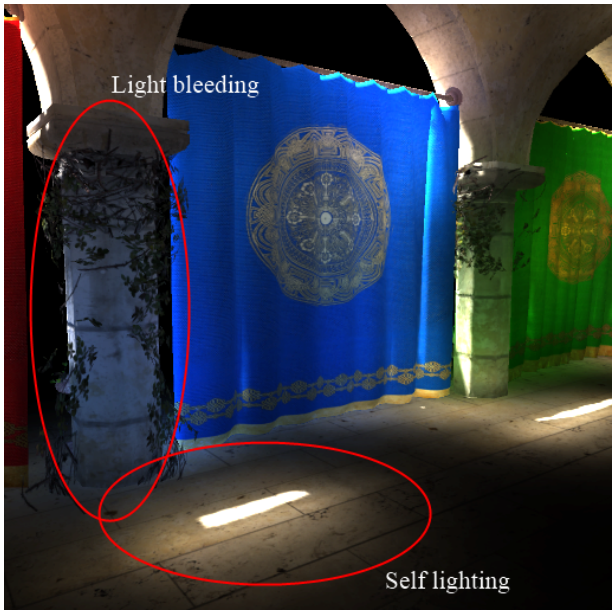

Figure 3.7: An image rendered using a publicly available NVIDIA implementation of light propagation volumes. This image is an example of the light bleeding problem with light propagation volumes.

Light Bleeding and Fuzzy Occlusion Occlusion may occur for light going across grid cells, but occlusion that should occur because of geometry within the same LPV grid cell will fail, resulting in light bleeding. Figure 3.7 shows an example of this light bleeding as the pillar in the left side of the image is lit by indirect light that originates from the opposite side of the pillar.

The "fuzzy" occlusion also allows light bleeding to occur across grid cells. Beyond the injection of geometry data and GV grid cell location, positional data is lost, so in the case of literal physical occlusion in a specific occlusion scenario, there is no guarantee that the generated attenuation term will properly occlude the propagated flux. The light bleeding isn't just a result of occlusion failing within a

grid cell, but also from light propagating across grid cells because the attenuation term did not properly occlude propagated indirect light from the drape. Figure 3.8 shows the fuzzy occlusion that is possible with LPVs; light bleeding artifacts are present in the images of this figure.

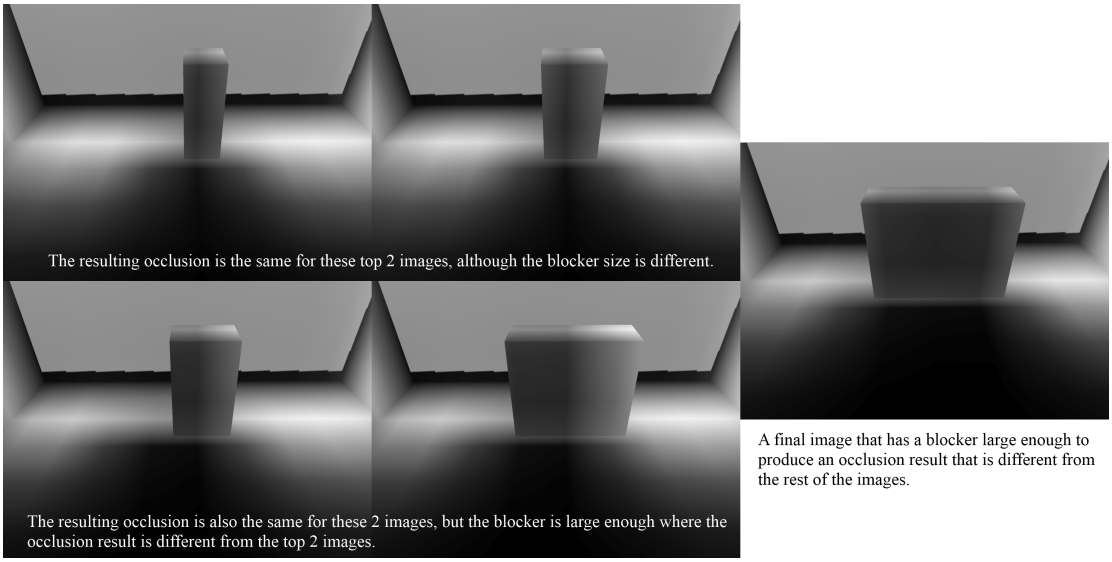

Figure 3.8: These images were rendered using a publicly available NVIDIA implementation of light propagation volumes. These images showcase the fuzzy occlusion that is possible with LPVs. Occlusion is similar with blockers that do not occupy more GV grid cells. When blockers end up occupying different grid cells, then the resulting occlusion will change.

Chapter 4

Higher Order Light Propagation Volumes

I propose modifications in this chapter to the light propagation volume(LPV) technique resulting in higher order light propagation volumes (HOLPVs). First, I extend the light propagation volume(LPV) technique to use up to band $l = 2$ spherical harmonic coefficients, three bands total, with nine coefficients over the previous four. The six direction propagation of the LPV technique is unable to use most of the band two coefficients; all band two coefficients, except the $m = 0$ coefficient, will be zero whenever reprojection occurs during propagation, so the higher order propagation technique is explained in this chapter. Besides being able to use the band two spherical harmonic coefficients, the new propagation method reduces the appearance of grid-based lighting artifacts and produces a more realistic

indirect lighting result.

The reason for extending the light representation to band two spherical harmonics is because the significance of the coefficients when representing clamped cosine lobe lights. Recalling from chapter 3, section 3.2.1, that LPVs use zonal harmonics to represent a clamped cosine lobe light that can be easily rotated, the zonal harmonic coefficients for the clamped cosine lobe are significantly large up to band two [15]. The value of these zonal harmonic coefficients is directly proportional to the value of the final spherical harmonic coefficients that will be used to represent diffuse light in the LPV, so it would be wise to use coefficients that help provide a significant representation. Coefficients beyond band two asymptotically approach zero, providing little to no significance to the clamped cosine lobe representation [15].

One of the reasons for modifying the propagation approach is to make use of the band two spherical harmonic coefficients. With the six direction propagation, the $m = 0$ coefficient is insignificant and the rest of the band two coefficients are zero when calculated during propagation. To sample the source cell light intensity in the direction of propagation, there is a dot product between the spherical harmonic basis vector in the direction of propagation and the spherical harmonic coefficients of the source cell. For the six direction approach, the spherical harmonic basis vector in the direction of propagation has band two coefficients (besides the $m = 0$ coefficient) that are zero. So, the contribution by band two coefficients is

insignificant with the LPV method's six direction propagation. The higher order 26 direction approach alleviates this problem, because when light flux is propagated to destination cells that only share a corner or edge with the source cell, the band two coefficients are non-zero and represent a significant amount of light flux.

The next important consideration is how indirect lighting will be applied; in this case, the modifications to the technique do not affect how the indirect light is applied, except to extend the spherical harmonic integration calculation to handle band two spherical harmonics.

4.1 Extending LPVs to use Band Two Spherical Harmonic Coefficients

This section explains the math and concepts concerning injection of virtual point lights represented with band zero, one, and two spherical harmonics.

4.1.1 Injection of Virtual Point Lights and Geometry Data

As discussed in the light propagation volume chapter, reflective shadow maps are used to generate indirect light sources called virtual point lights, which are injected into the light propagation volume in an initialization step. The injected virtual point lights are represented as clamped cosine lobe lights.

The clamped cosine lobe lights are represented with zonal harmonics, which

may be easily rotated in the direction of the virtual point light. In order to carry this out, the zonal harmonic coefficients Z_l for each spherical harmonic band l must be calculated for the clamped cosine lobe representation. Equation 3.1 is used to calculate these coefficients:

$$Z_l = \int_0^{2\pi} \int_0^{\pi/2} Y_l^0(\theta) \cos\theta \sin\theta d\theta d\phi$$

Using equation 3.1, the Z_l coefficients representing the clamped cosine lobe for band zero, one, and two may be calculated:

$$Z_0 = \frac{\sqrt{\pi}}{2}$$
$$Z_1 = \sqrt{\frac{\pi}{3}}$$
$$Z_2 = \frac{\sqrt{5\pi}}{8}$$

Equation 3.2 is used to rotate the clamped cosine lobe light:

$$Y_{rl}^m = \sqrt{\frac{4\pi}{2l+1}} Z_l Y_l^m(\omega)$$

$Y_l^m(\omega)$ is the band l basis m spherical harmonic coefficient in the ω direction. $Y_{rl}^m(\omega)$ is the corresponding rotated spherical harmonic coefficient as a result of the zonal harmonic rotation. With equation 3.2 and the zonal harmonic coefficients for the cosine lobe representation, the following equations are derived, which calculate each spherical harmonic coefficient for the coefficient vector that represents the clamped cosine lobe light oriented in the direction ω:

$$Y_{r0}^{0} = \sqrt{4\pi}\frac{\sqrt{\pi}}{2}Y_{0}^{0}(\omega) = \pi Y_{0}^{0}(\omega)$$

$$Y_{r1}^{-1,0,1} = \sqrt{\frac{4\pi}{3}}\sqrt{\frac{\pi}{3}}Y_{1}^{-1,0,1}(\omega) = \frac{2\pi}{3}Y_{1}^{-1,0,1}(\omega)$$

$$Y_{r2}^{-2,-1,0,1,2} = \sqrt{\frac{4\pi}{5}}\frac{\sqrt{5\pi}}{8}Y_{2}^{-2,-1,0,1,2}(\omega) = \frac{\pi}{4}Y_{2}^{-2,-1,0,1,2}(\omega)$$

4.2 Higher Order Propagation

Higher order propagation involves propagating light data in the same six orthogonal directions as with the original technique and also to every neighbor that shares a corner or edge with the source cell, for a total of 26 destination cells.

As explained in chapter 3, the propagation is discretized. The propagation calculations are discretized by separate destination grid cells, and among each destination grid cell, the propagation is discretized by grid cell face. Propagation does not occur to every face in each destination grid cell, so it is important to identify the relevant faces for each propagation case. The higher order propagation approach is discretized to different destination grid cell faces to accommodate the additional destination grid cells.

For the six direction approach, figure 4.1 helps to identify the relevant grid cell faces, which are shaded red. In figure 4.1, the center, interior cell is the source cell, and the destination cells are all bordering this center cell. The relevant faces needed for propagation are the faces that do not border the source cell and

face outwards. To ensure that energy is conserved, the total solid angle that is subtended by all of these relevant faces from the center of the source cell is 4π.

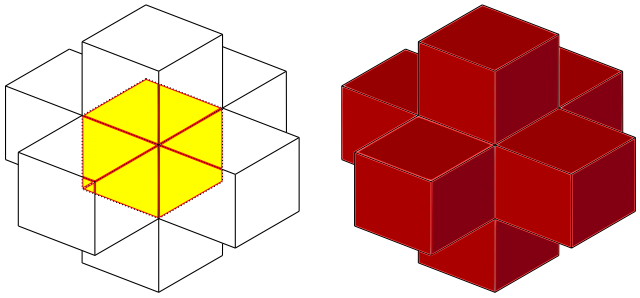

Figure 4.1: A representation of the grid cells involved in six direction propagation. The yellow, interior cell is the source cell. On the right, the red-shaded faces are the relevant faces that light is propagated to during each propagation iteration. These faces are all of the outwards facing grid cell faces.

For the higher order propagation approach, figure 4.2 visualizes the relevant grid cell faces, which are shaded red. The center cell, colored yellow, is the source cell, and the remaining grid cells, all external cells, are the destination cells. The relevant grid cell faces for propagation all face outwards from the arrangement in figure 4.2. Similarly to the set of relevant faces for the six direction propagation, the total solid angle subtended by all of these grid cell faces from the center of the source cell is 4π, ensuring the conservation of energy for the higher order propagation approach. Figure 4.3 gives an overview of the three different destination cell cases and the relevant grid cell faces for each. The figure also provides the calculated solid angles necessary to perform the propagation calculations for the higher order propagation approach. Refer to appendix A for information regarding how the provided solid angles are calculated.

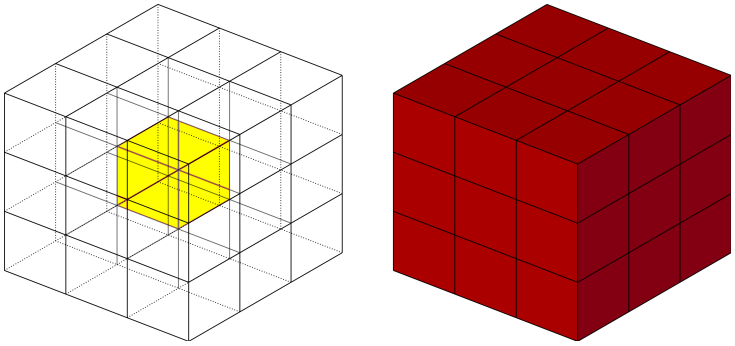

Figure 4.2: A representation of the grid cells involved in 26 direction propagation, a 3x3x3 cell arrangement where the yellow, interior cell is the source cell. On the right, the red-shaded faces are the relevant faces that light is propagated to for each grid cell. The relevant faces for this case are the exterior faces on the 3x3x3 cell arrangement.

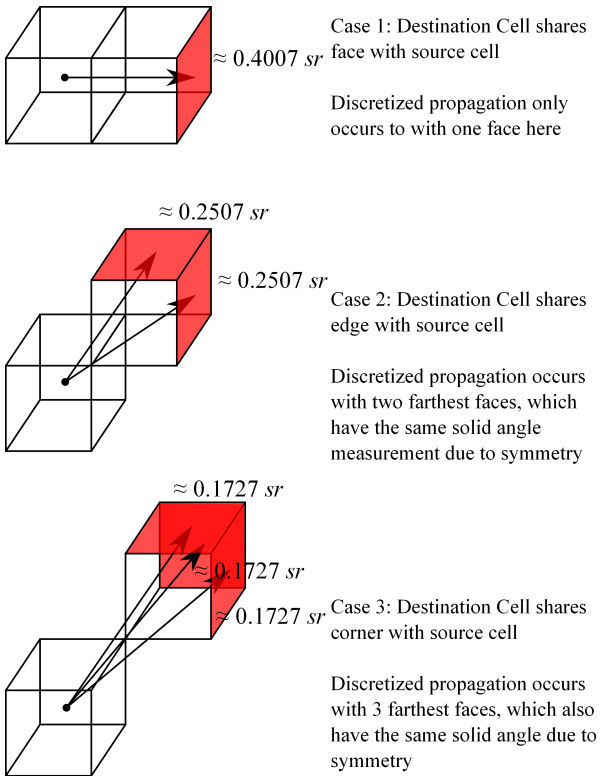

Figure 4.3: This figure visualizes the three destination cell cases for higher order propagation. Each case has the relevant grid cell faces shaded red, and the calculated approximate solid angle necessary for carrying out propagation calculations. Refer to appendix A for information regarding how the provided solid angles are calculated.

The propagation of light flux still occurs using the same method for six direction propagation; propagation to a single relevant cell face occurs with reprojection, where a clamped cosine lobe light oriented in the direction of the destination face is injected into the destination cell. The spherical harmonic coefficients representing this clamped cosine lobe light are scaled by a propagated flux value, which is calculated with equation 3.3:

$$\Phi_{propagate} = \frac{\Omega_{vis}}{4\pi} I_{source}(\omega_d)$$

Ω_{vis} is the calculated solid angle of the destination face, which gets normalized by 4π. $I_{source}(\omega_d)$ is the sampled intensity from the source cell in the direction of propagation ω_d, from the center of the source cell towards the middle of the destination face. To calculate this intensity value, the source cell's spherical harmonic coefficients are sampled in the direction of propagation ω_d. After the spherical harmonic coefficients for each face are calculated, the coefficient vectors are summed up to obtain a vector of spherical harmonic coefficients for the destination cell that represents the total propagated light intensity distribution from one source cell. During a single propagation iteration, there will be a separate spherical harmonic coefficient vector from each source cell (26 total); these spherical harmonic coefficient vectors are summed, resulting in a single coefficient vector representing the total received light flux for that one propagation iteration.

To take into account occlusion, calculations will still be the same as the

original technique, as explained in section 3.2.3. Blocker coefficients are bilinearly interpolated from coefficients sampled from the geometry volume which represent blocker geometry located on the corners of each light propagation volume cell. Although there are more directions now, blocker coefficients may still be bilinearly interpolated for each direction. These blocker coefficients are sampled in the negative direction of propagation to obtain an attenuation factor to apply to the propagated flux.

4.3 Implementation Details

To represent the full light volume buffer for HOLPVs, the data is spread across several volume textures. To account for the first eight coefficients, each color channel has two four-channel volume textures. The last coefficient for each color channel is stored in a three-channel volume texture.

As described for LPVs, the propagation process for HOLPVs is programmed as a gather rather than a propagation, where the flux is gathered into a destination cell from all 26 neighboring source cells. Also similarly to LPVs, three separate full LPV buffers are needed. Two of these buffers act as "ping pong" buffers, where data is read from one and written to the other and for subsequent steps the roles of the buffers are switched. The last buffer is an accumulation buffer where propagated light data is accumulated after each propagation iteration. After all propagation iterations, the accumulation buffer represents the scene's final indirect

lighting distribution and is used for applying indirect lighting.

Chapter 5

Results

In the first section, I discuss the implementations used for producing the results discussed in this chapter. The following sections each discuss a separate test scene and are accompanied by relevant figures. The last section discusses memory usage and peformance. A table detailing the performance statistics for each of the test scenes is located in the last section.

5.1 Implementation

In order to evaluate the light propagation volume(LPV) technique, I used NVIDIA's publicly available implementation of LPVs. This implementation is written in C++, using Microsoft's DirectX 11 computer graphics API. The implementation uses a simple rendering approach, using the LPVs technique to calculate indirect lighting, which is combined with simple diffuse direct lighting in a final

pass. Using NVIDIA's implementation as base code, I extend the implementation to higher order light propagation volumes(HOLPVs).

To compute reference images, I used path tracing for images with direct lighting and a single indirect light bounce, and light sampling for images that only have direct lighting. The implementation is my own, which is written in C++ and supports area light, point light, directional lighting, and texturing.

Geometry in all test scenes were made up of triangle primitives. The LPV and HOLPV implementations use the "sdkmesh" format used by Microsoft in the DirectX 10 and 11 sample programs. The path tracer uses the "wavefront obj" file format for loading geometry information.

5.2 Diagonal Propagation Test Scene

This first test scene is a simple scene to test the propagation of light diagonally across the LPV grid which can be compared to a reference result rendered with path tracing. The scene consists of a flat square with an area light near one of the square's corners, which is perpendicular to the surface and oriented toward the middle of the surface. The LPV and HOLPV images were generated using the same volume dimensions, no occlusion applied, an area light emitting the same amount of energy, and 20 propagation steps. The flat square is five by five units. The grid volume was centered on the surface, covering 10 by 10 by 10 units, oriented in the same direction as the flat plane, which means the injected cosine lobe

lights for the area light will be pointed diagonally across the grid volume in the XZ plane.

The reference image, figure 5.1, shows more light energy distribution over the surface when compared to the LPV image, figure 5.2, and HOLPV image, figure 5.3. The HOLPV image more closely matches the reference image, suggesting that HOLPVs is better at simulating indirect light.

The test scene was set up to test light propagation diagonally across the grid volume to show a situation where HOLPVs has an advantage over LPVs in a lighting case that would be present in many scenes. The higher order propagation approach, within one propagation iteration, propagates light flux to cells that share a face, edge, or corner with the source cell. The older propagation approach only propagates to cells sharing a face, and will take two iteration steps to reach the cells sharing only sharing an edge or a corner with the source cell. The propagation to these extra cells in one iteration helps transport light through the scene with less propagation steps. The graph, figure 5.4, which plots the middle scanline of the reference, HOLPV, and LPV images for this test scene, provides more evidence showing how HOLPVs is an improvement over LPVs for simulating indirect light. The graph shows how the HOLPV approach more closely matches the reference image.

This test scene also shows how the HOLPV technique is less prone to light grid artifacts. The higher order propagation approach promotes a distribution of

light energy that reduces large differences in the light distribution of neighboring grid cells, the reason grid-like lighting artifacts appear.

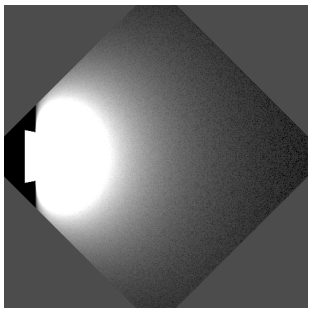

Figure 5.1: A path traced reference image with 100,000 samples per pixel, and took 47 minutes to render.

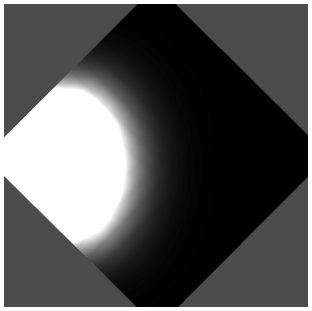

Figure 5.2: The propagation test scene rendered with LPVs.

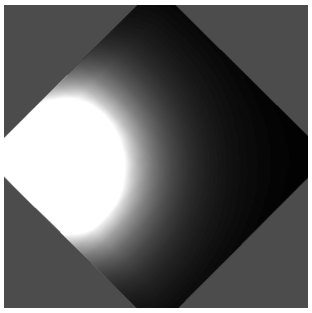

Figure 5.3: The propagation test scene rendered with HOLPVs.

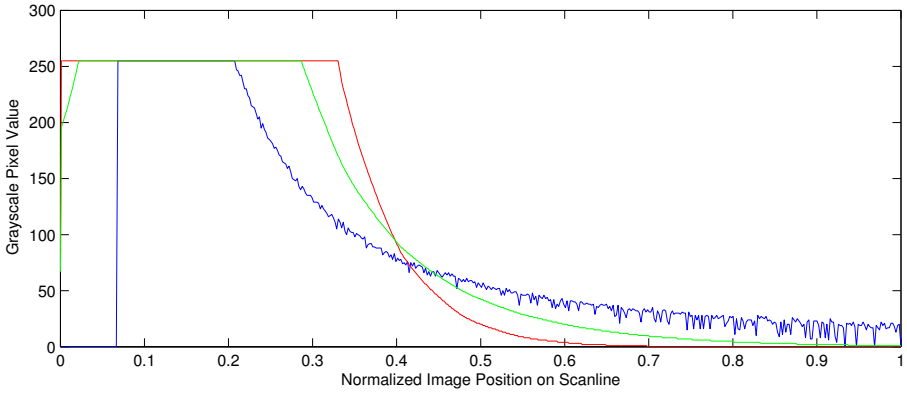

Figure 5.4: The middle scanline from each image is graphed here, with blue for the reference image, green for the HOLPV image, and red for the LPV image.

5.3 Sponza Hallway Test Scene

This sponza hallway scene, in Crytek's version of the sponza model used in NVIDIA's LPV reference implementation by default, is used as a test scene that more closely models what may appear in a video game. The extents of the sponza model are 37.2 units by 15.54 units by 22.88 units, the X, Y, and Z dimensions, respectively. For both the LPV and HOLPV images, the grid volume, had dimensions 32 by 32 by 32, spanned 21 units in each direction. The amount of indirect light flux injected into the scene is the same for both approaches, and both used 16 propagation steps and had occlusion applied for indirect lighting. The geometry is textured and a directional light is used for direct lighting with simple Lambert diffuse shading.

A comparison between the resulting LPV and HOLPV images of this test

scene shows that indirect lighting in the HOLPV image reaches more areas in the scene, with more indirect light reaching the left wall in the hallway, and also the pillars in the background. The LPV image shows none of the pillars in the background are indirectly lit, and the left wall in the hallway receives little indirect light; light in the LPV image is also more concentrated near the sources of indirect light, a problem also in the planter test scene. The indirect lighting in the HOLPV image more closely matches the reference image, figure 5.5, than the LPV image, but is not without errors that the spherical harmonic lighting and grid-based light transport are prone to. Self lighting occurred near the injection of indirect light sources, and light bled to areas that should have been occluded.

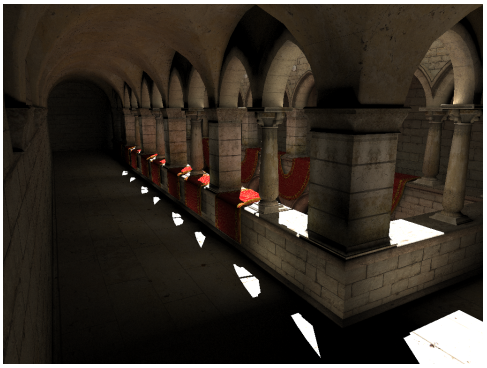

Figure 5.5: The reference image of the test scene in the Crytek sponza hallway. The image was path traced with 4000 samples per pixel, and took 10 hours to render.

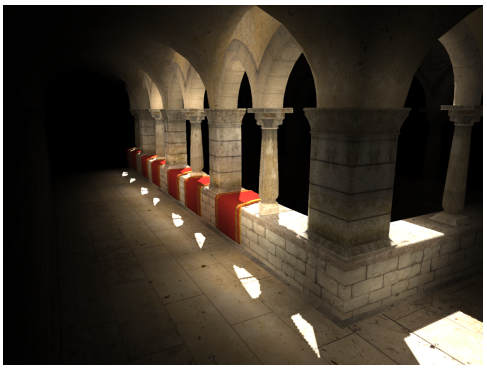

Figure 5.6: A scene in the Crytek sponza rendered using LPVs.

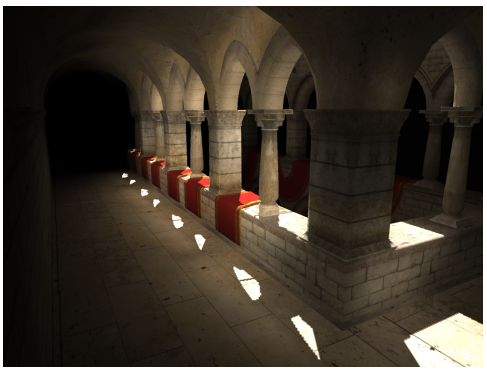

Figure 5.7: A scene in the Crytek sponza rendered using HOLPVs.

5.4 Stanford Bunny Test Scene

This next scene uses the Stanford bunny model to observe indirect lighting on a detailed surface geometry. The test scene used an area light source that lit the front of the bunny model. Figure 5.9 uses LPVs and figure 5.10 uses HOLPVs. The Stanford bunny model here measures about 2.3 by 1.8 by 2.3, in the X, Y, and Z directions, respectively. Both scenes use a grid volume divided into 32 by 32 by 32 uniform grid cells, and the grid volume spanned 20 units. Both scenes also use 10 propagation steps. No occlusion is applied.

The LPV image shows the bunny model is lit with flat and even shading, regardless of surface orientation. The HOLPV image shows that the indirect lighting is affected by the different surface orientations on the details of the bunny. The use of up to band two spherical harmonic coefficients gives the light more directional representation, so the different surface orientations have more influence on the applied indirect lighting.

The higher order propagation approach also has influence on the improved lighting details. The six direction propagation must be causing the first spherical harmonic coefficient, which represents light omni-directionally, to carry more light energy relative to the directional coefficients; when indirect lighting is applied, the contribution from the directional coefficients are sometimes not significant enough to produce lighting that is influenced directionally because of the more significant omni-directional coefficient. This is not the case with HOLPVs. The higher or-

der propagation allows the directional coefficients to carry more significance (also because there are more directional coefficients) relative to the omni-directional coefficient, and thus when indirect lighting is applied, the surface orientation has more influence on the result.

The next test scene is another example of this improvement, but with a planter model in the Crytek sponza.

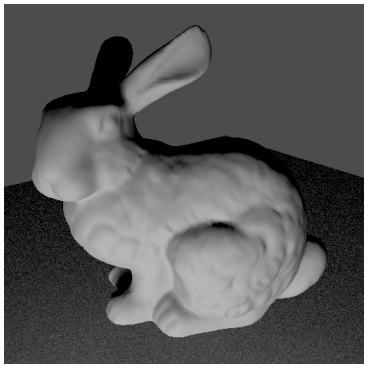

Figure 5.8: This is the reference image for the bunny test scene. It was rendered using light sampling, 1000 samples per pixel, and direct lighting only. The image took four minutes to generate.

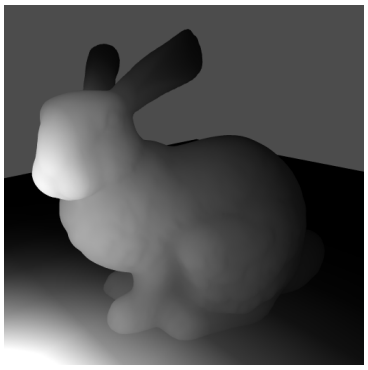

Figure 5.9: An image of the Stanford bunny, lit by an area light, rendered using LPVs.

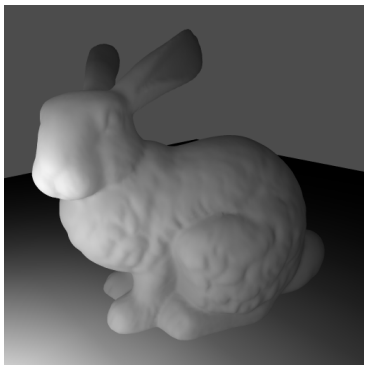

Figure 5.10: An image of the Stanford bunny, lit by an area light, rendered using HOLPVs.

5.5 Sponza Planter Test Scene

This test scene focuses on a planter model in the Crytek sponza. The scene uses the same test parameters for the scene as the sponza hallway test scene. This test scene has additional images this time, where the corresponding images without textures are shown to emphasize lighting details.

This test scene was chosen complement the bunny test scene, to show the improved lighting detail that the HOLPV method is capable of in a setting that more closely resembles a video game scene. Closely observing the planter, which is only indirectly lit, and the drapes, which has some indirect and direct lighting, the HOLPV scene has indirect lighting that is influenced more by the orientation of the surfaces; the lighting details are more apparent. The planter in the LPV image is lit on all sides almost the same, even though the indirect light is only coming from the drapes located to one side of the planter; the HOLPV image has lighting detail that is consistent with the source of the indirect light, as the planter has shadows on the side of features that face away from the indirect light sources.

Comparing the HOLPV image to the reference image, indirect red light from the drapes overtakes other sources of indirect light; this is a problem in both the LPV and HOLPV images because the grid volume doesn't capture all sources of indirect light in the scene, many of which are from surfaces of a whiter color. A grid volume that spans a larger world space volume would help capture these light sources, but at the same time the volume would have a coarser resolution, causing

larger self lighting and light bleeding problems.

The HOLPV images also still show self lighting problems near indirect light sources and light bleeding because of spherical harmonic lighting and the grid-based light transport.

Figure 5.11: The reference image of the test scene for the Crytek sponza planter. The image was path traced with 2000 samples per pixel, and took four hours to render.

Figure 5.12: A rendering of a planter in the Crytek sponza scene, with textures, using LPVs.

Figure 5.13: A rendering of a planter in the Crytek sponza scene, with textures, using HOLPVs.

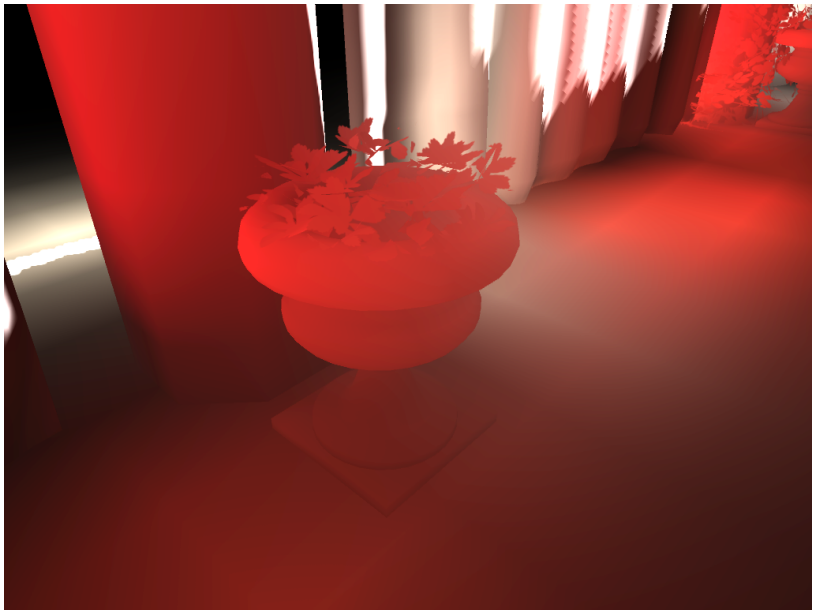

Figure 5.14: A rendering of a planter in the Crytek sponza scene, without textures, using LPVs. Without textures, the flatter lighting effects are more apparent.

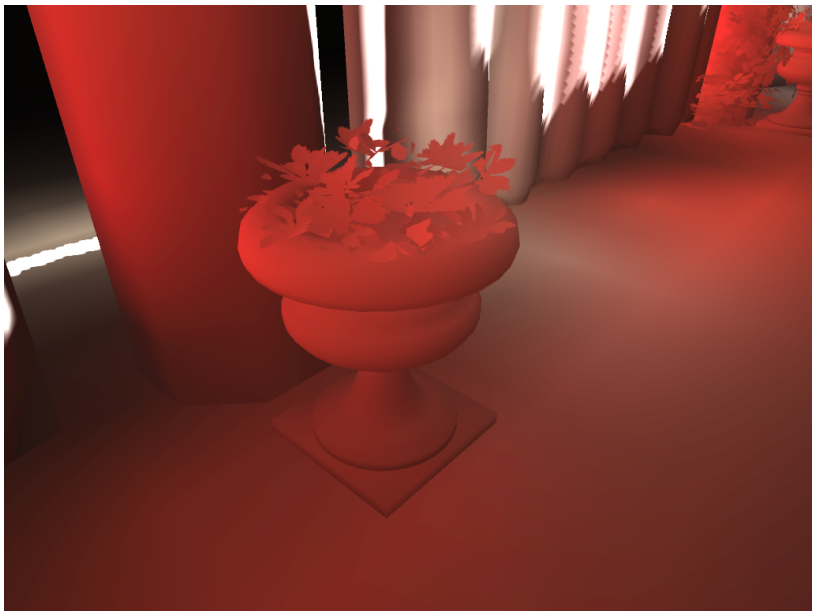

Figure 5.15: A rendering of a planter in the Crytek sponza scene, without textures, using HOLPVs. Without textures, the detailed lighting effects are more apparent.

5.6 Floor Test Scene

For this test scene, I use the same planter test scene, with the same rendering parameters, but use a view of the ground in the planter scene being indirectly lit. The reason for this scene is because this scene helps show how HOLPVs reduces the presence of lighting artifacts. Figure 5.18 shows a graph of the middle scanline of the reference, LPV, and HOLPV result images, with position plotted versus their pixel intensity. For the LPV and HOLPV scanline plots, the locations of each volume grid cell's borders is denoted by an abrupt change in the constant slope between grid cells. The red line (the LPV scanline plot), shows a larger difference between each slope change than the green line (the HOLPV scanline plot). These large differences is what causes the grid-like lighting artifacts for indirect lighting. The HOLPV technique reduces the appearance of these artifacts because the propagation method produces a smaller difference in the intensity distributions of neighboring grid cells, as evidenced by the plot.

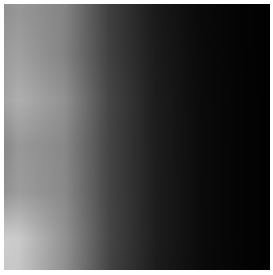

Figure 5.16: A section of floor from the planter test scene rendered with LPVs and no textures.

Figure 5.17: A section of floor from the planter test scene rendered with HOLPVs and no textures.

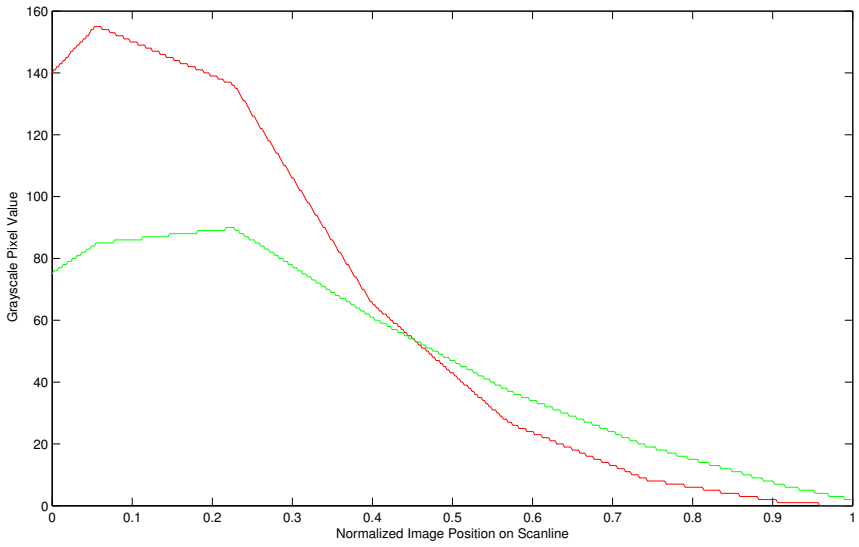

Figure 5.18: The middle scanline from the LPV and HOLPV scene graphed for comparison. The green is for HOLPVs, and red is for LPVs.

5.7 Cornell Box Test Scene

The cornell box scene is a standard scene for testing global illumination algorithms. This test scene is used to compare LPVs and HOLPVs to a reference approach, to evaluate if modifications to LPVs help improve the result for a difficult test scene.

This scene is usually lit with an area light with indirect light bounces, but because LPVs are not equipped to calculate lighting for a direct area light with a single bounce, the test scene is rendered with direct lighting only from an area light. To accomplish this with LPVs and HOLPVs, cosine lobe lights distributed over the surface of the area light are injected into the LPV. Lighting in the scene only results from the light propagation volume.

Figure 5.19 is the reference image, rendered with light sampling and is only directly lit, so the image can be effectively compared to the LPV and HOLPV images. Figure 5.20 was generated using LPVs and figure 5.21 was generated with HOLPVs. Both images generated with LPVs and HOLPVs used a grid cube volume that spanned 12 units and had grid dimensions 32 by 32 by 32. The cornell box is a cube shaped room with a length of 5.5 units. Also, the LPV and HOLPV generated images used 16 propagation steps and had occlusion applied.

It is obvious that the LPV and HOLPV result images exhibit the same light bleeding and self-lighting problems although occlusion is being applied. The light bleeding is still due to the grid-based lighting approach, something that cannot

be addressed with HOLPVs. Self lighting at the top of the scene is due to the omni-directional spherical harmonic coefficient in both scenes building up during propagation and producing a large amount of lighting on the ceiling.

Although HOLPVs still exhibit some of the same problems that LPVs has, there are still improvements. The HOLPV approach was able to transport light throughout the scene with lighting that has a closer resemblance to the reference image. The LPV approach is unable to propagate a significant amount of light energy through the scene, with a large amount of light energy still near the light source, and any light flux that reaches the floor is too insignificant to make lighting noticeable even though the number of propagation steps allows light to reach all parts of the room.

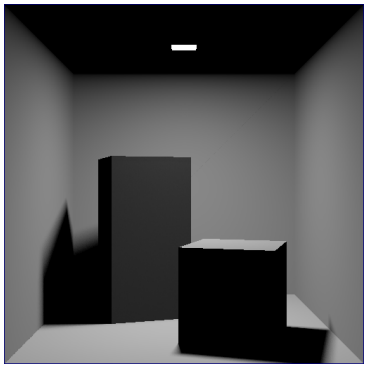

Figure 5.19: A reference image of the cornell box rendered using light sampling, with 1000 samples per pixel and took one minute to render. Only direct light from the area light at the top of the box is considered, in order to make this image comparable to the capabilities of light propagation volumes.

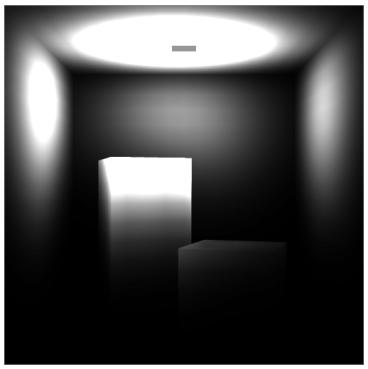

Figure 5.20: The cornell box scene generated using light propagation volumes, with direct lighting from an area light at the top of the box.

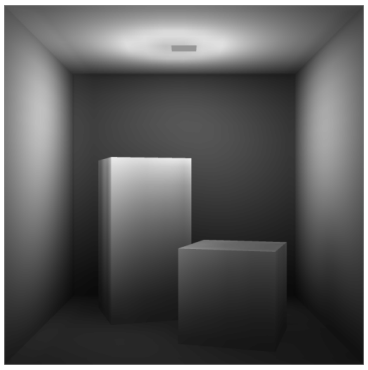

Figure 5.21: The cornell box scene generated using higher order light propagation volumes, with direct lighting from an area light at the top of the box.

5.8 Performance

Table 5.1 provides performance information for HOLPVs and LPVs for the test scenes that are discussed in this chapter. The test scenes were all ran on a computer with the following specs: NVIDIA GTX 460 graphics card, AMD X6 1055T with 2.80 GHz clock speed, and 8.00 GB RAM.

Table 5.1 omits memory information because the memory usage is the same for all images that use the same implementation. All of the grid volumes used in the implementations consist of multiple (or one for the geometry volume) four-channel 16-bit floating point 32 by 32 by 32 volume textures. For LPVs, a single LPV grid volume takes up 0.75 megabytes(MB). Considering that in order to carry out all computations, three sets of LPVs are necessary, the total memory footprint is 2.25 MB. A single LPV grid volume used in HOLPVs is 1.75 MB, and for carrying out propagation calculations, the total memory footprint is 5.25 MB. Both implementations use a single geometry volume, which occupies 0.25 MB of memory.

Another memory consideration is the reflective shadow maps, which is the same for both implementations. The reflective shadow maps are made up of two four-channel, with one unsigned byte per channel, 256 by 256 textures used for light flux and normals, and a single 32 bit texture of the same dimensions is used for depth. The total memory footprint of the reflective shadow map is 0.75 MB.

There is a performance impact with the higher order propagation, espe-

cially when occlusion is considered. The reason for the performance impact is that higher order propagation is discretized into 54 separate clamped cosine lobe light reprojections that occur during each propagation iteration, for each grid cell (in parallel on the GPU), while the six direction propagation only has 30 reprojections. Another performance consideration is the increased number of coefficients means more texture accesses when looking up coefficients. Another source of increased texture accesses is during propagation, because propagation considers more neighboring grid cells, instead of only six grid cells being sampled, there are 26 grid cells that must be sampled during each propagation iteration, for the propagation calculation of each grid cell (but again, this happens in parallel on the GPU for each grid cell).

With more careful implementation considerations for HOLPVs, something I was unable to cover in this thesis, the performance can be vastly improved to run at a reasonable rate.

Table 5.1: Performance of Higher Order Light Propagation Volumes.

Test Scene	Image Resolution	Propagation Steps	Occlusion Used	LPV Frame Time (ms)	HOLPV Frame Time (ms)
Diagonal Propagation	900x900	20	No	6.0	48.0
Sponza Hallway	1024x768	8	Yes	49.0	102.7
Stanford Bunny	900x900	10	No	5.9	26.6
Sponza Planter	1024x768	8	Yes	49.0	99.0
Cornell Box	900x900	16	Yes	9.9	174.0

Chapter 6

Conclusion

6.1 Summary

In this thesis, I have provided modifications to the original light propagation volume (LPV) method, creating the higher order light propagation volume(HOLPV) method. In chapter 4, I explain my modifications to the light propagation volume method. The first modification is the extension of the spherical harmonic light representation to the second band of coefficients. Using the extra coefficients helps improve the quality of the lighting by providing a more precise light representation with more directional lighting data. To make use of the expanded light representation, which LPVs could not, the original six direction LPV propagation approach was extended to the higher order propagation approach, which considers 26 propagation directions. These improvements help achieve the

following: reduced light artifact appearance, improved indirect lighting details, and light transport throughout scenes that result in a lighting distribution that more closely matches reference methods.

6.2 Improved Rendering

The new propagation method transports light through the scene with less propagation steps than the LPV technique, and at the same time the HOLPV technique improves render quality. The distribution of indirect light energy more closely matches more realistic methods, like path tracing, instead of causing a concentration of light energy near the sources of indirect light.

The indirect diffuse lighting detail on surfaces is improved. The band two spherical harmonic coefficients and new propagation method allow surface orientations to have more influence on indirect lighting with HOLPVs. This allows surface details to stand out even when only indirectly lit, a subtle effect that greatly increases render quality.

The presence of lighting artifacts is reduced with HOLPVs. The new propagation method produces a more natural and even distribution of light energy in neighboring grid cells. This even distribution, which helps to create the more realistic indirect lighting effects, also reduces the presence of grid-like lighting artifacts.

6.3 Real-Time Calculations

The HOLPV technique requires more computational power, causing longer frame times relative to LPVs, but the new technique still runs in real-time. The method improves on LPVs, increasing render quality while performing all necessary computation in a fraction of a second. Given the rapidly increasing power of GPUs, the availability of profiling tools for graphics applications, and more careful implementation considerations, the HOLPV method may easily run much faster than the reference implementation used for this thesis, improving the viability of the new technique.

6.4 Future Work

Here I suggest some ideas for future work to extend upon the ideas presented in this thesis:

- A more careful approach to the implementation on the GPU would improve computation time. Also, utilizing the GPU with a different API, such as DirectCompute or CUDA, are worthwhile avenues to explore to optimize computation time.

- Researching a more precise geometry representation that would help improve the occlusion quality with the grid-based light transport.

- Research a way to handle specular light transport in the light propagation grid volume. The grid volume is used only for diffuse lighting in this paper, though it may be possible to extend the grid volume to handle the propagation of specular lighting. Some considerations may be a different light representation, such as wavelets [10], or using more spherical harmonic bands.

- Trilinear interpolation to sample the spherical harmonic coefficients when applying indirect light may be another source of lighting artifacts. The interpolation of spherical harmonic coefficients causes a linear lighting effect for surfaces located between grid cells (see section 5.6). Finding an improved sampling approach may help reduce light artifacts further.

Appendix A

Solid Angle Calculations

The following code is used to calculate the solid angles used in the implementation of the higher order propagation for higher order light propagation volumes. The formulas used to calculate the solid angles come from Mazonka's article [11] detailing solid angle calculations.

The code is a Matlab function. The input is a single matrix that contains the vectors describing each pyramid that will contribute to the solid angle sum output. Each vector should be written with the apex of the pyramid as the origin. The matrix should have three rows and the number of columns should be a multiple of three, where each set of three consecutive columns describes each pyramid.

To find the solid angle of each relevant face for the higher order propagation, each square oblique pyramid, formed using the center of the source cell as the apex and the destination face as the base, is described as two triangular pyramids.

```
function [ omega ] = solidAngle( S )
%SOLIDANGLE Calculate the solid angle of multiple
%           oblique tri-pyramids
% S = sets of 3 vectors defining the pyramids
%     with the origin as the apex
% omega = sum of the solid angles for each of
%         the input pyramids

% get number of vectors
% (assumed to be a multiple of 3)
  numVecs = size(S,2);
  omega = 0;

  for currIndex=1:3:numVecs
    uVec = S(:,currIndex);
    vVec = S(:,currIndex+1);
    wVec = S(:,currIndex+2);
    uLen = norm(uVec);
    vLen = norm(vVec);
    wLen = norm(wVec);
```

```
        stp = det([uVec vVec wVec]);
        denom = uLen*vLen*wLen +...
                uLen*dot(vVec,wVec) +...
                vLen*dot(uVec,wVec) +...
                wLen*dot(uVec,vVec);

        omega = omega + abs(2*atan(stp/denom));
    end
end
```

Bibliography

[1] Arvo, J., Torrance, K., and Smits, B. A framework for the analysis of error in global illumination algorithms. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1994), SIGGRAPH '94, ACM, pp. 75–84.

[2] Bavoil, L., Sainz, M., and Dimitrov, R. Image-space horizon-based ambient occlusion. In *ACM SIGGRAPH 2008 talks* (New York, NY, USA, 2008), SIGGRAPH '08, ACM, pp. 22:1–22:1.

[3] Cook, R. L., Carpenter, L., and Catmull, E. The reyes image rendering architecture. *SIGGRAPH Comput. Graph. 21*, 4 (Aug. 1987), 95–102.

[4] Dachsbacher, C., and Stamminger, M. Reflective shadow maps. In *Proceedings of the 2005 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2005), I3D '05, ACM, pp. 203–231.

[5] Dachsbacher, C., and Stamminger, M. Splatting indirect illumination. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2006), I3D '06, ACM, pp. 93–100.

[6] Jensen, H. W. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.

[7] Kajiya, J. T. The rendering equation. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1986), SIGGRAPH '86, ACM, pp. 143–150.

[8] Kaplanyan, A. Light propagation volumes in cryengine 3, 2009.

[9] Kaplanyan, A., and Dachsbacher, C. Cascaded light propagation volumes for real-time indirect illumination. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games* (New York, NY, USA, 2010), I3D '10, ACM, pp. 99–107.

[10] Lewis, R. R. *Light-driven global illumination with a wavelet representation of light transport*. PhD thesis, Vancouver, BC, Canada, Canada, 1998. UML Order No. GAXNQ-34579.

[11] Mazonka, O. Solid Angle of Conical Surfaces, Polyhedral Cones, and Intersecting Spherical Caps. *ArXiv e-prints* (May 2012).

[12] Mitchell, J., McTaggart, G., and Green, C. Shading in valve's source engine. In *ACM SIGGRAPH 2006 Courses* (New York, NY, USA, 2006), SIGGRAPH '06, ACM, pp. 129–142.

[13] Pharr, M., and Green, S. Ambient occlusion. In *GPU Gems*, R. Fernando, Ed. Addison-Wesley, 2004, pp. 279–292.

[14] Phong, B. T. Illumination for computer generated pictures. *Commun. ACM 18*, 6 (June 1975), 311–317.

[15] Ramamoorthi, R., and Hanrahan, P. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A 18*, 10 (Oct 2001), 2448–2459.

[16] Ritschel, T., Grosch, T., and Seidel, H.-P. Approximating dynamic global illumination in image space. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games* (New York, NY, USA, 2009), I3D '09, ACM, pp. 75–82.

[17] Sloan, P.-P. Stupid spherical harmonics tricks, 2008. This is an electronic document. Date of publication: February 2008. Date retrieved: June 19, 2012. Date last modified: February 2010.

[18] Sloan, P.-P., Kautz, J., and Snyder, J. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. 21*, 3 (July 2002), 527–536.

[19] Veach, E., and Guibas, L. J. Metropolis light transport. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 65–76.

[20] Williams, L. Casting curved shadows on curved surfaces. *SIGGRAPH Comput. Graph. 12*, 3 (Aug. 1978), 270–274.