國立交通大學

資訊工程系

碩 士 論 文

可有效描繪半透明度材質之快取技術

An Efficient Caching Technique for Rendering Translucent Materials

研究生：耿世瓴

指導教授：莊榮宏　博士

中 華 民 國 九 十 三 年 六 月

可有效描繪半透明度材質之快取技術

An Efficient Caching Technique for Rendering Translucent Materials

研究生：耿世瓴　　Student : Shih-Ling Keng

指導教授：莊榮宏　博士　　Advisor : Dr. Jung-Hong Chuang

國 立 交 通 大 學

資 訊 工 程 學 系

碩 士 論 文

A Thesis

Submitted to

Institute of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master

in

Computer Science and Information Engineering

June 2004

Hsinchu, Taiwan, Republic of China

中華民國九十三年六月

可有效描繪半透明度材質之快取技術

研究生：耿世甌　　　　　　　　指導教授：莊榮宏 博士

國立交通大學資訊工程學系

摘　　要

本論文提出一個可有效描繪半透明度材質的快取技術。此快取技術主要靈感來源爲 irradiance caching。我們首先採用 irradiance caching 的基本架構當作我們的基礎。接著提出分割圓(split-disk) 模型決定 cache 的分布並且轉換 dipole diffusion approximation 方程式，推導出 gradient of subsurface illuminance，最後利用已存好的 cache 以及 gradient of subsurface illuminance 來內插圖像。實驗結果顯示，我們只需要很少量的 cache 便可內插出整張圖像，整體加速約爲 5~15 倍，且內插所產生的視覺誤差幾乎可以忽略。

An Efficient Caching Technique for Rendering Translucent Materials

Student : Shih-Ling Keng Advisor : Dr. Jung-Hong Chuang

Department of Computer Science and Information Engineering
National Chiao Tung University

ABSTRACT

This thesis presents an efficient rendering technique for translucent materials using caches. The proposed caching scheme, inspired by the irradiance caching method, is integrated into a hierarchical rendering technique for translucent materials. We propose a split-disk model to determine the cache distribution and derive the subsurface illuminance gradient used for interpolation by reformulating the equation of dipole diffusion approximation as a 3D convolution process. Our experiments show that only a few caches are required to interpolate the entire image, while the visual difference is negligible. The speedup could be achieved up to one order of magnitude.

Acknowledgement

I would like to thank my advisor, Professor Jung-Hong Chuang and Wang-Yeh Lee for their guidance and encouragement throughout my Master's degree. Thanks to my colleagues in CGGM lab., Holmes, imm, dis, malkhut, Rey, mousep, rekan, chihchun, and danki. It's pleasure to be with all of you in the pass two years. Thanks to my girl friend, Jamie. You have made my college life wonderful and colorful. Finally, I would like to thank my family for their love, understanding and continued support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Accurately modeling the behavior of light to produce realistic images is a great challenge in computer graphics. Over the years, many illumination models have been developed for realistic image synthesis, trying to describe the scattering of light from materials. Most of them focus on developing models for the bidirectional reflectance distribution function (BRDF) [Jensen 2001], which assumes that light enters and leaves a material at the same point on the surface. In some cases like metals, this assumption is valid, and results in convincing visual appearances. But when accounting for translucent materials, which exhibit significant light transport below the surface, BRDF is not enough. Light hitting a translucent material does not just bounce from surfaces. Instead, light beams penetrate below the surface, scatter inside the material, and leave the object at a different point on the surface. This phenomenon is known as subsurface scattering.

Subsurface scattering diffuses the incident light, blurs the effects of small geometric details on the surface, and softens the overall looks. In addition, scattered light may pass through an object, which lights up thin geometric details when the objects are illuminated from behind. These effects create a distinct look that cannot be achieved with simple BRDF model. It is therefore necessary to go back to the more general models with bidirectional surface scattering reflectance distribution function (BSSRDF) [Jensen 2001] to simulate subsurface scattering. While BRDF models are just approximations of BSSRDF models, the BSSRDF can describe light transport between any two points on the surface. This requires treating the material as a participating medium with a boundary surface. It therefore needs huge amount of time to simulate all the effects of subsurface scattering.

1.1 Literature Review

Traditionally, subsurface scattering has been approximated as Lambertian diffuse reflection that make final images look hard and distinctly computer-generated. In computer graphics, the first model dealing with subsurface scattering was proposed by Hanrahan and Krueger [Hanrahan and Krueger 1993]. They proposed an analytic expression for single scattering in a homogeneous, uniformly lit slab and also a method for simulating subsurface scattering by tracing photons through the material. But in the end they used a BRDF to represent the final model.

Dorsey et al. [Dorsey et al. 1999] later used photon mapping to simulate full subsurface scattering for the rendering of weathered stones. Pharr and Hanrahan [Pharr and Hanrahan 2000] proposed the idea of non-linear scattering equations and demonstrated how the scattering equations could be used to simulate subsurface scattering more efficiently than traditional Monte Carlo ray tracing. Though these approaches could fully simulate the subsurface scattering, they suffered from very costly computational efferts because they are both based on path sampling techniques. They are particularly inefficient for highly scattering materials, in which light could scatter several hundred times before exiting the materials.

For highly scattering material, Stam [Stam 1995] first introduced the diffusion theory to computer graphics. The diffusion theory could model multiple scattering as a diffusion process which works particularly well in highly scattering materials. Stam also proposed a multi-grid method to solve a diffusion equation approximation, and used this approach to render clouds with multiple scattering. However, while the multi-grid method is suitable for rendering clouds, it is still too costly to render common translucent materials such as milk and skin.

A major breakthrough was recently proposed by Jensen et al. [Jensen et al. 2001]. They applied the dipole diffusion approximation [Farrell et al. 1992] to simulate the subsurface scattering in homogenous semi-infinite plane-parallel media, thus greatly reducing computation time. Jensen et al. [Jensen et al. 2001] achieved more than two orders of magnitude speedup compared to the approach of using full Monte Carlo simulation, whereas Jensen and Buhler [Jensen and Buhler 2002] made a quantum leap further. They decoupled the computation of irradiance at the surface from the evaluation of scattering inside the material and used a hierarchical integration technique to evaluate the dipole diffusion approximation. This dramatically reduces the computation time from several minutes to a few seconds. In addition, they showed that the contribution from single scattering is almost negligible, and considering only multiple scattering could produce well enough visual appearances. They also found that although the dipole diffusion approximation is only valid for planar, infinitely large and thick media, mis-using it for curved surfaces still yields very plausible images. The analytical BSSRDF model proposed by Jensen et al. [Jensen et al. 2001] has so great impact that it is adopted by the following researches on the rendering of translucent materials.

Another acceleration scheme for Jensen's BSSRDF model was proposed by Lensch et al [Lensch et al. 2002]. They used a mixture of radiosity-like finite element computations and texture filtering to evaluate the BSSRDF integral. Although this method can render objects interactively with moving light sources at several frames per second, it requires significant precomputation for a texture atlas, form factors, and filter kernels.

The concept of radiosity-like finite element computation was then further improved by Mertens et al. [Mertens et al. 2003]. They used a hierarchical boundary element method inspired by the hierarchical radiosity with clustering to solve the integral describing the subsurface scattering based on Jensen’s analytical BSSRDF model. Instead of using Jensen’s hierarchical point sampling approach, they derived a semi-analytical integration method that allows for computing necessary point-to-patch form-factor efficiently and accurately. They showed that high-quality renderings of deformable translucent objects consisting of tens of thousands of polygons can be obtained from scratch in fractions of a second.

More recently, Hao et al. [Hao et al. 2003] proposed a simple lighting model to simulate the effects on translucent meshes. Their approximations are based on the observation that subsurface scattering is relatively local due to its exponential falloff. They modified the traditional local illumination model into a run-time two-stage process. The first stage involves the computation of reflection and transmission of light on surface vertices. The second stage bleeds scattering effects from a vertex's neighborhood to generate the final result. They then merged the run-time two-stage process into a run-time single-stage process using pre-computed integrals. By using this approach, they achieved interactive frame rates with about one to two orders of magnitude speedup compared to the previous methods. However, in their method, a large memory storage is required to record the pre-computed integrals. This problem was later alleviated by Hao and Varshney [Hao and Varshney 2004]. They compressed the data by spherical harmonic basis functions and used the "reference points" scheme to reduce the extra storage from 200 bytes per vertex to 20 bytes per vertex.

1.2 Thesis Overview

1.2.1 Motivation

Although recent researches [Lensch et al. 2002; Carr et al. 2003; Dachsbacher and Stamminger 2003; Hao et al. 2003; Mertens et al. 2003; Hao and Varshney 2004] have improved the speed of rendering translucent materials to some extent, none of them could be easily integrated into existing renderers. They require either complex rendering algorithms, or some specific data structures (e.g., the pre-computed integral storage and "reference points" scheme used in [Hao and Varshney 2004]). In other words, most of them are isolated systems for the pure purpose of experiments. The algorithms used in these rendering systems are not suitable for movie industry, where some specific renderers must be used, and objects are often not mesh-based.

In this thesis, we seek a rendering technique suitable for rendering translucent materials in production, i.e., in the film industry, since it is the major application of rendering translucent materials. Currently, the analytic BSSRDF for rendering translucent materials [Jensen et al. 2001] is being used in almost all visual effects for movies and it indeed offers the visual effects industry a mean to circumvent producing computer-generated human faces that look plastic and unconvincing on the silver screen.

To devise an efficient rendering technique for the film industry, we investigate the effect of subsurface scattering and find that it has distinguishing characteristics just as the effect of indirect lighting in the global illumination. They both tend to change slowly and require a lot of sample points to compute. This inspires us to use the classic irradiance caching technique introduced by Ward et al. [Ward et al. 1988] as a basis, and to extend it for calculating the subsurface illuminance, which can be defined as the light flux per unit area arriving at an inner surface point within materials via subsurface scattering from the nearby surfaces.

Irradiance caching was originally designed for accelerating the computation of indirect illumination in a Monte Carlo ray tracer [Ward et al. 1988]. It is a method for caching and re-using irradiance values (via interpolation) on Lambertian surfaces. It uses a split-sphere model to estimate the amount of change in the irradiance. By using this error estimate, it can decide whether to interpolate or to directly compute irradiance at a location, and calculate the weights for interpolation. In 1992, Ward and Heckbert [Ward and Heckbert 1992] further improved the caching technique by deriving a formulae for the irradiance gradients. They found that the gradient can be estimated from the rays used to sample the indirect diffuse illumination. By using the formulae for the irradiance gradient, they achieved a significantly more accurate image than that in [Ward et al. 1988]. The irradiance caching technique was later extended to accelerate the computation of *ambient occlusion* in production [Christensen 2003]. In this thesis, we show that it is feasible to extend the irradiance caching technique to accelerate the computation of the subsurface illuminance as well.

1.2.2 Contribution

The contributions we achieve in this thesis can be summarized as:

- We show that the classic irradiance caching originally used in the ray tracing field for computing indirect illumination could also be extended to render translucent materials.

- We conclude that the dipole diffusion approximation is a 3D convolution process.
- We reformulate the calculation of the gradient of subsurface illuminance using convolution.
- We propose a split-disk model analogous to Ward's split-sphere model [Ward et al. 1988] to determine the spacing of samples.
- We successfully integrate the irradiance caching technique [Ward et al. 1988] into the rapid hierarchical rendering technique for translucent materials [Jensen and Buhler 2002] with up to one order of magnitude speed-up.

1.2.3 Outline

The remainder of the thesis is organized as follows: Chapter 2 gives the fundamentals of realistic rendering and the essential theory behind rendering of translucent materials. Chapter 3 presents the caching technique we propose for accelerating the rendering of translucent materials, which includes some derivations and algorithms. Chapter 4 gives some experimental results from our implementation of the proposed method. Finally, some concluding remarks and future work are presented in Chapter 5.

Chapter 2

Backgrounds

This chapter lays the groundwork for creating plausible images with translucent materials. It is divided into three parts: the first part gives the foundation of physically accurate rendering in computer graphics. The second part deals with the essential theory behind the rendering of translucent materials. Finally, the third part introduces the dipole diffusion approximation – the practical equation, on which recent relevant researches are based to render translucent materials.

2.1 Fundamentals of Realistic Image Synthesis

In this section, we firstly introduce radiometry, the basic terminology used to describe light. Next, we briefly discuss the interaction of light and surface, and then introduce the most important component in realistic image synthesis, rendering equation. Finally, we present how to display the high-dynamic-range images realistically on low-dynamic-range devices, i.e., the so called tone-reproduction operator.

2.1.1 Radiometry

There are several models developed since middle of 1600 attempting to explain the behavior of light, such as ray optics (also known as geometrical optics, which is the most commonly used model in computer graphics), wave optics, electromagnetic optics, and photon optics [Jensen 2001]. The basic terminology used among them is *radiometry*, a measurement of optical radiation, which is electromagnetic radiation within the frequency range between $3*10^{11}$ and $3*10^{16}$ Hz. This range corresponds to wavelength between 0.01 and 1000 micrometers (μm), and includes the regions commonly called the ultraviolet, the visible, and the infrared.

Before we introduce the quantities and units used in radiometry, we explain two basic terms: projected area and solid angle.

Projected area is defined as the orthogonal projection of a surface of any shape onto a plane normal to the unit vector. The differential form is $dA_{proj} = \cos(\beta)dA$, where β is the angle between the local surface normal and the line of sight. We can integrate dA_{proj} over the visible surface area to get

$$A_{proj} = \int_A \cos\beta dA .$$

Here we list some common examples of projected area assuming that there are no obstacles in the line of sight (Table 2.1).

Shape	Area (visible parts)	Projected area
Flat rectangle	$A = Length \cdot Width$	$A_{proj} = Length \cdot Width \cdot \cos\beta$
Circular disc	$A = \pi r^2$	$A_{proj} = \pi r^2 \cos\beta$
Sphere	$A = 2\pi r^2$	$A_{proj} = \pi r^2$

Table 2.1 Some examples of projected area

Solid angle is an extension of plan angle from two dimensions to three dimensions. Recall the definition of a plane angle [Palmer 2003] is "*One radian is the plane angle between two radii of a circle that cuts off on the circumference an arc equal in length to the radius.*" And the definition of a solid angle [Palmer 2003] extends to "*One steradian (sr) is the solid angle that, having its vertex in the center of a sphere, cuts off an area on the surface of the sphere equal to that of a square with sides of length equal to the radius of the sphere.*" The solid angle of an object is thus the area of the projection of the object onto a unit sphere. Note that two objects different in shape can still subtend the same solid angle. We can think of the differential solid angle as representing both a direction and an infinitesimal area on the unit sphere.

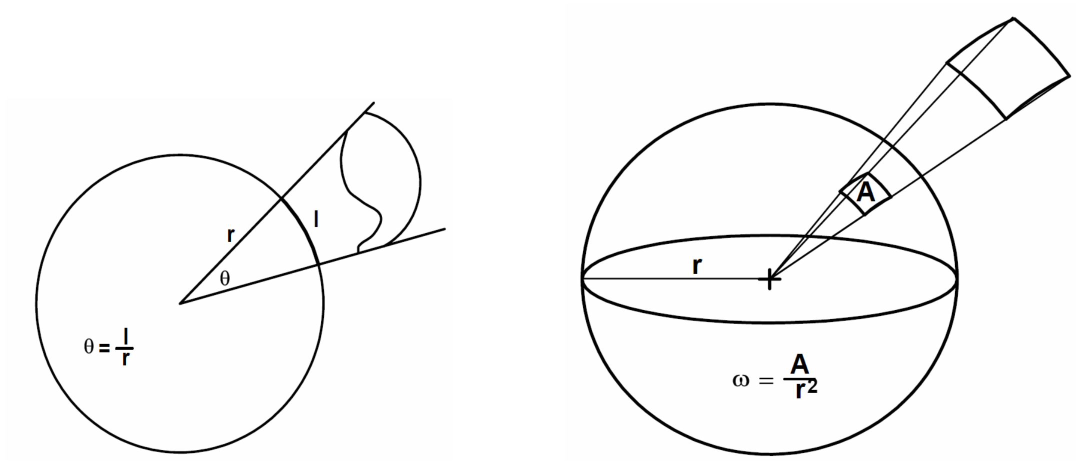


Figure 2.1: Plane angle and solid angle [Palmer 2003]

The most basic quantity in radiometry is the photon. The energy e_λ of a photon with a wavelength λ is

$$e_\lambda = \frac{hc}{\lambda},$$

where $h \approx 6.63 \cdot 10^{-34} J \cdot s$ is *Planck's constant*, and *c* is the speed of light. e_λ is measured in joules (*J*).

Spectral radiant energy Q_λ in n_λ photons with wavelength λ is

$$Q_\lambda = n_\lambda e_\lambda = n_\lambda \frac{hc}{\lambda}.$$

Radiant energy *Q* is the energy computed by integrating the spectral radiant energy over all possible wavelengths:

$$Q = \int_0^\infty Q_\lambda d\lambda .$$

Radiant flux or radiant power *Φ* is the time rate of flow of radiant energy:

$$\Phi = \frac{dQ}{dt},$$

where *t* is measured in second. Radiant flux is often just called the flux.

Radiant flux area density *M or B or E* is defined as the differential flux per differential area at a surface location x:

$$M(x) = B(x) = \frac{d\Phi}{dA}$$

and

$$E(x) = \frac{d\Phi}{dA},$$

where M is referred to as radiant existence, which is the flux leaving a surface, B is referred to radiosity, which is exactly the same as radiant existence; and E is referred to as irradiance, which is the flux arriving at a surface.

Radiant intensity *I* is defined as the differential flux per differential solid angle *dw*:

$$I(w) = \frac{d\Phi}{dw}.$$

Radiance *L* is defined as the differential flux per differential projected area per differential solid angle:

$$L(x, w) = \frac{d^2\Phi}{\cos\theta dA dw}.$$

Radiance is a five-dimensional quantity (three for position and two for direction), which is the most important quantity in radiometry, since it could most closely represent the color of an object. Most light receivers, such as cameras and the human eye, are sensitive to radiance, while the response curve of these sensors may be different.

2.1.2 Light Scattering

The interaction of light and material is complicated in real worlds. In this section, we will introduce two theoretical frameworks used to model the scattering of light by materials.

The Bidirectional Scattering Reflectance Distribution Function or **BSSRDF** *S* is the most general description of light transport, which relates the differential reflected radiance dL_r at x_o in direction w_o to the differential incident flux $d\Phi_i$ at x_i from direction w_i [Jensen 2001]:

$$S(x_i, w_i, x_o, w_o) = \frac{dL_r(x_o, w_o)}{d\Phi_i(x_i, w_i)}.$$

The BSSRDF is eight-dimensional (four for two positions in local two-dimensional coordinates and four for two directions) and costly to evaluate, so there are only a few papers in computer graphics that really accounts for BSSRDF.

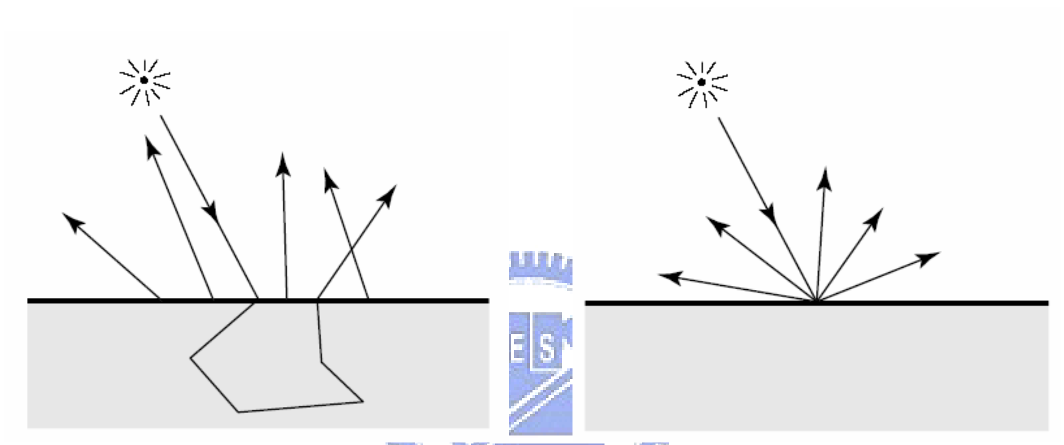


Figure 2.2: BSSRDF and BRDF [Jensen et al. 2001]

The Bidirectional Reflectance Distribution Function or **BRDF** f_r is an approximation of the BSSRDF, which assumes that light enters and leaves the material at the same point (i.e., $x_o = x_i$) [Jensen 2001]. This reduces the BRDF to a six-dimensional function (two for position in local two-dimensional coordinates, four for two directions):

$$f_r(x, w_i, w_o) = \frac{dL_r(x, w_o)}{dE_i(x, w_i)} = \frac{dL_r(x, w_o)}{L_i(x, w_i)(w_i \cdot n)dw_i}$$

where n is the normal at x. BRDF defines the relationship between differential reflected radiance and differential incident irradiance and is widely used in photo-realistic rendering. It has some interesting properties:

- The BRDF can take any positive value, and varies with wavelength.
- The BRDF is independent of the direction, in which light flows (based on Helmholts's law of reciprocity):

$$f_r(x, w_i, w_o) = f_r(x, w_o, w_i)$$

This is a fundamental property that is used by most global illumination algorithms, since that makes it possible to trace light path in both directions.

- The value of the BRDF for some incident direction w_a is independent of the possible presence of irradiance along other incident direction w_b, so the BRDF behaves as a linear function with respect to all incident directions. If we know the incident radiance from the hemisphere of all incoming directions and their respective BRDFs, we can compute the reflected radiance L_r in any direction by integrating the incident radiance L_i:

$$L_r(x, w_o) = \int_\Omega f_r(x, w_i, w_o) dE(x, w_i) = \int_\Omega f_r(x, w_i, w_o) L_i(x, w_i)(w_i \cdot n) dw_i \quad (2.1)$$

where n is the normal at x, and Ω is the hemisphere of incoming directions at x.

- The BRDF must satisfy the following constraint due to energy conservation:

$$\int_\Omega f_r(x, w_i, w_o)(w_o \cdot n) dw_o \leq 1, \forall w_i.$$

2.1.3 The Rendering Equation

The rendering equation is a mathematical formulation of the steady-state equilibrium distribution of energy in a scene without participating media. It forms the mathematical basis for producing realistic images. The rendering equation expresses the outgoing radiance L_o as the sum of the emitted radiance L_e and the reflected radiance L_r:

$$L_o(x, w_o) = L_e(x, w_o) + L_r(x, w_o).$$

By using Equation 2.1 to replace the reflected radiance we get:

$$L_o(x, w_o) = L_e(x, w_o) + \int_\Omega f_r(x, w_i, w_o) L_i(x, w_i)(w_i \cdot n) dw_i. \quad (2.2)$$

This is the basic form of the rendering equation describing all light transport in a scene without participating media. There are some other forms of the rendering equation for specific global illumination algorithms.

2.1.4 Tone-reproduction Operator

Up to now, all we discuss about is computing the correct radiometric values for each pixel in the final images. These values are measured in radiance that can be computed at a specific point in space and in a specific direction. However, these physically based radiance values do not appropriately indicate how the human eye perceives the environment because the human visual system is very complicated and does not simply respond linearly to changing levels of illumination.

Tone-reproduction operator, or tone-mapping operator, is therefore presented to solve this problem by exploiting the limitations of the human visual system to display a high-dynamic-range picture onto a low-dynamic-range display device. There are various tone-mapping operators have been presented in literatures. In general, they function by creating a local scale factor for each pixel in the high-dynamic range image based on the local adaptation luminance of the pixel and the high-dynamic-range value of the pixel. The resulting value produced by tone-mapping operators is typically an RGB value in the range from 0 to 1 that can be displayed on the output device.

2.2 Light Transport Theory

The rendering equation described in section 2.1.3 considers the interaction of light only at the surfaces of objects within an environment without participating media. It assumes the light propagates instantaneously through vacuum without any absorption. However, to render translucent materials, we must take light scattering in participating media (or inside materials) into account. Light transport theory is what could describe the propagation of light in materials. It is a heuristic approximation of electromagnetic scattering theory, but is unable to predict diffraction, interference, or quantum effects [Hanrahan and Krueger 1993].

In light transport theory, photons traveling in a medium will collide with the medium, causing them to be absorbed or change directions (scattering). Absorption means the energy carried by photons is converted into other types of energy, for instance, the energy could be converted from radiation to kinetic energy of particles in the medium.

The probability that a photon gets absorbed in a medium, per unit of distance along its direction of propagation, is called the *absorption coefficient* $\sigma_a(x)$ and the probability that a photon gets scattered in a medium is called the *scattering coefficient* $\sigma_s(x)$. These two coefficients are all measured in 1/m. This means that a photon traveling a distance Δx in a medium has a chance $\sigma_a\Delta x$ and $\sigma_s\Delta x$ of being absorbed and scattered, respectively.

The change in radiance L in the direction w due to out-scattering could be modeled as the following equation:

$$(w\cdot\nabla)L(x,w) = -\sigma_s(x)L(x,w), \qquad (2.3)$$

and the change due to absorption could be modeled as:

$$(w\cdot\nabla)L(x,w) = -\sigma_a(x)L(x,w). \qquad (2.4)$$

We often combine the two coefficients to the *extinction coefficient* $\sigma_t(x)$ as

$$\sigma_t(x) = \sigma_s(x) + \sigma_a(x),$$

and the combined loss in radiance is given by:

$$(w\cdot\nabla)L(x,w) = -\sigma_t(x)L(x,w). \qquad (2.5)$$

On the other hand, when photons move through the media, there will also be a gain in radiance due to in-scattering of light from other directions. The change due to in-scattering is modeled by:

$$(w\cdot\nabla)L(x,w) = \sigma_s\int_{\Omega 4\pi} p(x,w_i,w)L_i(x,w_i)dw_i\,, \qquad (2.6)$$

where the incident radiance, L_i, is integrated over all possible directions. $p(x,w,w_i)$ is the phase function describing the distribution of the scattered light. We assume that the phase function is normalized, $\int_{\Omega 4\pi} p(x,w_i,w)dw_i = 1$, and is a function only of the phase angle, $p(x,w,w') = p(x,w\cdot w')$. The mean cosine g of the scattering angle is defined as

$$g = \int_{\Omega 4\pi}(w\cdot w')p(w\cdot w')dw'.$$

It indicates the type of scattering in the medium. If g is positive, the medium is predominantly forward scattering; if g is negative, the medium is predominantly backward scattering; if g equals zero, the phase function is constant and the medium results in isotropic scattering. Most translucent materials are strongly forward scattering with $g > 0.7$. Such strongly peaked phase functions are costly to simulate in media with multiple scattering since the probability of sampling in the direction of the light source will be low in most situations. In this case we can benefit from a powerful technique known as the *similarity of moments* [Jensen et al. 2002], which allows us to change the scattering properties of the medium without significantly influencing the actual distribution of light [Jensen et al. 2001]. Specifically, we can modify the medium to have isotropic scattering by changing the scattering coefficient to

$$\sigma_s' = \sigma_s(1-g),$$

where σ_s' is the *reduced scattering coefficient.* The absorption coefficient remains unchanged, and we get the *reduced extinction coefficient* $\sigma_t' = \sigma_s' + \sigma_a$.

In addition to the gain in radiance due to in-scattering, there is also a gain in radiance due to volume emission L_e from the medium (i.e., a flame), and it is given by:

$$(w \cdot \nabla)L(x,w) = \sigma_a(x)L_e(x,w)\,. \qquad (2.7)$$

By combining Equation 2.5, 2.6, and 2.7 we get a linear integro-differential equation, which is the so called *light transport equation,* or *radiative transport equation*:

$$(w \cdot \nabla)L(x,w) = \sigma_a(x)L_e(x,w) - \sigma_t(x)L(x,w) + \sigma_s \int_{\Omega 4\pi} p(x,w',w)L_i(x,w_i)dw_i \quad (2.8)$$

2.2.1 The Volume Rendering Equation

In computer graphics, the light transport equation is often represented in another form, which is derived by integrating Equation 2.8 on both sides for a segment of length *s* subject to the appropriate boundary conditions [Jensen 2001] [Dutré et al. 2003].

$$\begin{aligned} L(x,w) = &\int_0^s e^{-\tau(x,x')}\sigma_a(x')L_e(x')dx' + \\ &+\int_0^s e^{-\tau(x,x')}\sigma_s(x')\int_{\Omega 4\pi} p(x',w_i,w)L_i(x',w_i)dw_i dx' \\ &+e^{-\tau(x,x+sw)}L(x+sw,w)\,, \end{aligned} \qquad (2.9)$$

where $\tau(x,x')$ (often called *optical depth*) is given by:

$$\tau(x,x') = \int_x^{x'} \sigma_t(z)dz\,.$$

Equation 2.9 is the so called *volume rendering equation*, which is much more complicated than the rendering equation because the light is influenced by light at every point in space, not just the points on other surface.

2.2.2 The Diffusion Approximation

The light transport equation, either in the form of Equation 2.8 or Equation 2.9, is a five-dimensional equation with integrals, which is very difficult to solve even when the light is scattered isotropically in the medium. Therefore, we need to use the diffusion approximation to make it feasible to solve the light transport equation. The diffusion approximation is based on the observation that as the number of scattering events increases, the angular dependence tends to be smoothed out, i.e., the light distribution in highly scattering media tends to become isotropic.

The diffusion approximation begins by dividing the radiance into two components: the *unscattered radiance* (or *reduced radiance*) L_u and the *scatterd radiance* (or *diffuse radiance*) L_d. The unscattered radiance is the radiance that reaches point *x* directly from a light source, or from the boundary of the participating medium. It decreases exponentially with the distance traveled through the medium [Jensen et al. 2002] [Stam 1995]:

$$L_u(x+\Delta x, w) = e^{-\sigma_t' \Delta x} L_u(x, w).$$

The diffuse radiance is radiance scattered one or more times in the medium. As stated above, after many scattering events, the angular dependence of diffuse radiance tends to be smoothed out, so the diffusion approximation can use the first four terms of the spherical harmonic expansion to represent L_d [Jensen et al. 2002] [Stam 1995]:

$$L_d(x, w) \approx \frac{1}{4\pi}\phi(x) + \frac{3}{4\pi}\vec{E}(x)\cdot w, \qquad (2.10)$$

where $\phi(x) = \int_{\Omega 4\pi} L_d(x, w')dw'$ is the 0th-order spherical harmonic, called the *radiant fluence* and $\vec{E}(x) = \int_{\Omega 4\pi} L_d(x, w')\cdot w' \cdot dw'$ is the 1st-order spherical harmonic, called the *vector irradiance*.

Substituting the diffusion approximation (Equation 2.10) to the light transport equation (Equation 2.8) yields the classic *diffusion equation* [Jensen et al. 2002] [Stam 1995]:

$$D\nabla^2\phi(x) = \sigma_a \varphi(x) - S_0(x) + 3D\nabla S_1(x), \qquad (2.11)$$

where $D = \frac{1}{3\sigma_t'}$; and $S_0(x)$ and $S_1(x)$ represents the 0th-order and the 1st-order spherical harmonic expansions of the source term, respectively.

The diffusion equation can be solved analytically for special cases [Jensen et al. 2001], or numerically by using a multigrid method [Stam 1995]. However, in the case of translucent materials, we are only interested in the outgoing radiance at the material surface as a function of the incoming radiance. Therefore we can further simplify the solution of the diffusion equation using the *dipole diffusion approximation*, which we will describe in next section.

2.3 The Dipole Diffusion Approximation

The dipole diffusion approximation, which approximates the volumetric source distribution using a dipole (i.e. two point sources), was originally developed in medical physics community. The idea was proposed by Eason [Jensen et al. 2001] for modeling the back-scattering of light by blood. Farrell et al. [Farrell et al. 1992] used a single dipole to represent the incident source distribution for the noninvasive determination of tissue optical properties in vivo. Jensen et al. [Jensen et al. 2001] then introduced the dipole diffusion approximation to computer graphics community for modeling the subsurface light transport.

The dipole diffusion approximation consists of positioning two point sources near the surface to approximate an incoming light (see Figure 2.3). One point source, the positive real light source, is locate at the distance z_r beneath the surface, and the other one, the negative virtual light source, is located above the surface at a distance z_v.

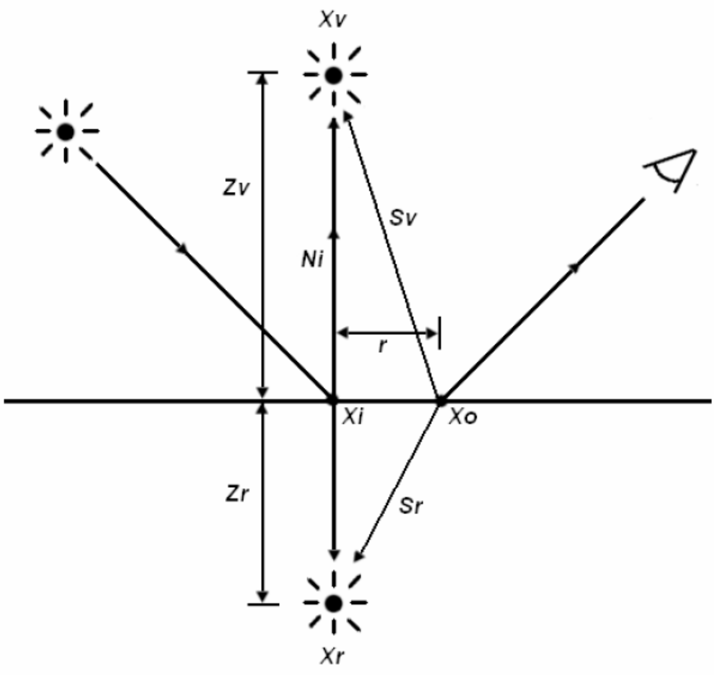


Figure 2.3: An incoming ray is transformed into a dipole source for the diffusion approximation [Jensen et al. 2001] [Poirier 2003].

By using the dipole diffusion approximation to solve diffusion equation, we can get the following expression for the radiant exitance $M_{x_i}(x_0)$ at surface location x_0 due to incident flux $\Phi(x_i)$ at x_i (see [Jensen et al. 2002] for the details of derivation):

$$dM_{x_i}(x_o) = d\Phi(x_i)\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right], \qquad (2.12)$$

where $\alpha' = \frac{\sigma_s'}{\sigma_t'}$ is the *reduced albedo* ($\alpha = \frac{\sigma_s}{\sigma_t}$ is the *albedo*, describing the relative importance of scattering versus absorption); $\sigma = \sqrt{3\sigma_a\sigma_t'}$ is the *effective transport coefficient*; $s_r = \sqrt{r^2 + z_r^2}$ is the distance from x_0 to the positive real light source; $s_v = \sqrt{r^2 + z_v^2}$ is the distance from x_0 to the negative virtual light source; $r = \|x_o - x_i\|$ is the distance from x_o to x_i; and $z_r = l_u$ and $z_v = l_u(1+\frac{4}{3}A)$ are the distance from the dipole source to the surface (shown in Figure 2.3). The mean-free path l_u is the average distance at which the light is scattered: $l_u = \frac{1}{\sigma_t'}$. Finally, the boundary condition for mismatched interfaces is taken into account by the A term which is computed as $A = \frac{1+F_{dr}}{1-F_{dr}}$, where the diffuse Fresnel term F_{dr} is approximated from the relative index of refraction η by [Jensen 2001]:

$$F_{dr} = -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta .$$

By using Equation 2.12, the subsurface illuminance then could be computed as:

$$\begin{aligned}
S(x_o) &= \int_{x_i \in A} dM_{x_i}(x_o) \\
&= \int_{x_i \in A} d\Phi(x_i)\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right] \\
&= \int_{x_i \in A} \hat{E}(x_i)dx_i\frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\cdot\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right] \\
&= \int_{x_i \in A} \hat{E}(x_i)R_d(x_i, x_o)dx_i \qquad (2.14)
\end{aligned}$$

where $\hat{E}(x_i) = F_{dt}(\eta)E(x_i)$ and $E(x_i)$ is the irradiance at point x_i. The diffuse Fresnel transmittance $F_{dt}(\eta)$ is defined as

$$F_{dt}(\eta) = 1 - F_{dr}(\eta),$$

and

$$R_d(x_i, x_o) = -D\frac{(\vec{n}\cdot\nabla\phi)(x_o)}{d\Phi_i(x_i)}$$

$$= \frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3}\right],$$

which is the *diffuse BSSRDF* defined as the ratio of radiant exitance to incident flux [Jensen et al. 2001]. Figure 2.4 shows the graph of R_d (which has exponentially decreasing property that we can employ in Chapter 3).

Finally, since the diffusion approximation already includes a diffuse Fresnel transmittance, the diffuse radiance L is computed as:

$$L(x_o, w) = \frac{F_t\left(\frac{1}{\eta}, w\right)}{F_{dt}(\eta)}\frac{S(x_o)}{\pi},$$

where F_t is the Fresnel transmittance.

Alternatively, we could omit the fresnel terms and assume a diffuse radiance:

$$L(x_o, w) = \frac{S(x_o)}{\pi}.$$

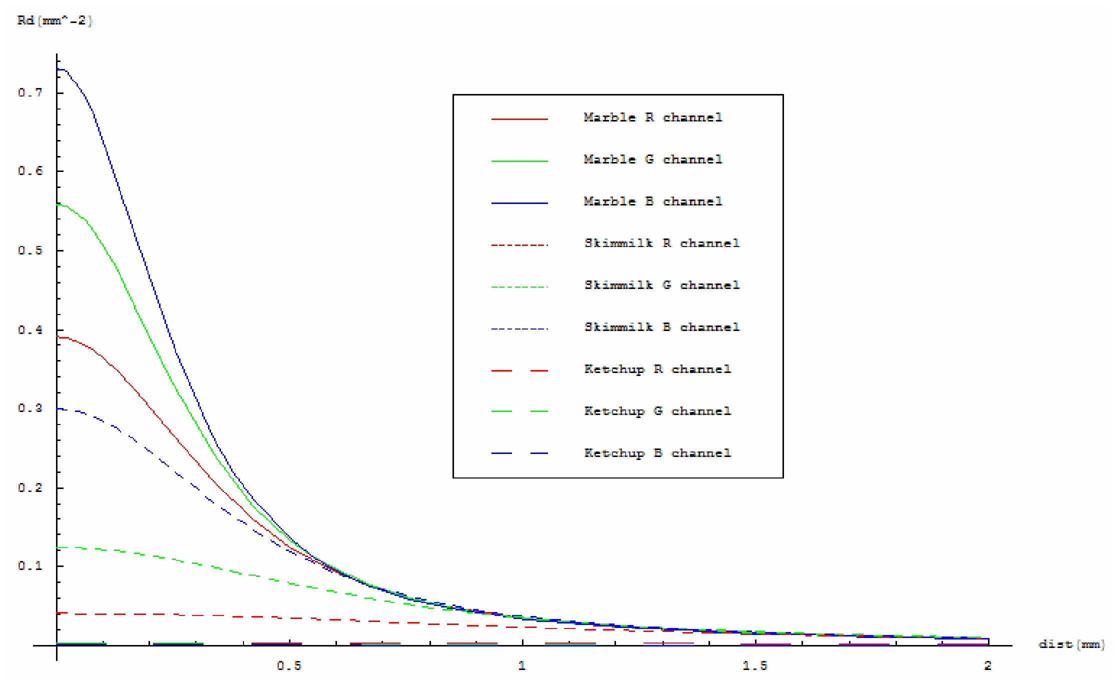


Figure 2.4: The graph of R_d.

The dipole diffusion approximation introduced by Jensen et al. [Jensen et al. 2001] is widely used by the following researchers in the computer graphics community. However, we find that this model could possibly be improved by using the equation proposed by [Kienle and Patterson 1997] from the medical physics community. They expressed the reflectance as the integral of the radiance over the backward hemisphere and considerably reduced the error in deriving the optical coefficients of translucent materials. By integrating their model into [Jensen et al. 2001], we expect that the measured parameters for translucent materials could be more accurate than [Jensen et al. 2001] and the final appearance could be more plausible as well.

Chapter 3

A Caching Technique for Rendering Translucent Materials

Our development of a caching technique for rendering translucent materials is based on the following observations:

- Although the diffusion approximation with a hierarchical integration technique [Jensen et al. 2002] is a very effective way of approximating multiple scattering, it typically requires more than two hundred sample points per pixel to evaluate surbsurface illuminance for each pixel.

- The subsurface illuminance tends to change slowly because the subsurface scattering diffuses the incident light, blurs the effect of small geometric details on the surface, and softens the overall looks. The smooth appearance is distinct especially for high translucent materials and objects with small sizes.

- The resulting subsurface illuminance value is view-independent because the subsurface scattering effect is only dependent on object geometry, object material properties, and how the object is illuminated.

It appears that for the sake of efficiency, we should not recalculate subsurface illuminance at each pixel, but should, instead, calculate it using a small set of previously computed values at nearby surfaces. The size of the small set of computed values should be independent of image size, thus high resolution images could be produced efficiently. Also, since subsurface illuminance does not depend on viewpoints, these computed values could be reused for many images, which is useful for animation production.

In this chapter, we propose a caching technique for rendering translucent materials. The framework of irradiance caching [Ward et al. 1988] is adopted. To do this, we derive an exact solution to calculate the gradient of subsurface illuminance and propose a split-disk model to determine the spacing of samples. Finally, we integrate the caching scheme into Jensen's hierarchical evaluation method [Jensen et al. 2002].

3.1 Dipole Diffusion Approximation as a Convolution Process

Recall that the subsurface illuminance function (Equation 2.14) is

$$S(p_o) = \int_{p_i \in A} \hat{E}(p_i) R_d(p_i, p_o) dp_i, \tag{3.1}$$

where $\hat{E}(p_i)$ is the irradiance scaled by Fresnel term and $R_d(p_i,p_o)$ is the *diffuse BSSRDF* expressed as:

$$R_d(p_i, p_o) = \frac{\alpha'}{4\pi}\left[z_r(1+\sigma s_r)\frac{e^{-\sigma s_r}}{s_r^3} + z_v(1+\sigma s_v)\frac{e^{-\sigma s_v}}{s_v^3} \right], \tag{3.2}$$

where $\alpha' = \frac{\sigma_s'}{\sigma_t'}$ is the *reduced albedo*; $\sigma = \sqrt{3\sigma_a \sigma_t'}$ is the *effective transport coefficient*; $s_r = \sqrt{r^2 + z_r^2}$ is the distance from p_o to the positive real light source; $s_v = \sqrt{r^2 + z_v^2}$ is the distance from p_o to the negative virtual light source; $r = \|p_o - p_i\|$ is the distance from p_i to p_o; and $z_r = l_u$ and $z_v = l_u(1 + \frac{4}{3}K)$ are the distance from the dipole source to the surface (shown in Figure 2.3). The mean-free path $l_u = \dfrac{1}{\sigma'_t}$ is the average distance at which the light is scattered. Finally, the boundary condition for mismatched interfaces is taken into account by the *K* term that is computed as $K = \dfrac{1+F_{dr}}{1-F_{dr}}$, where the diffuse Fresnel term F_{dr} is approximated from the relative index of refraction η by:

$$F_{dr} = -\frac{1.440}{\eta^2} + \frac{0.710}{\eta} + 0.668 + 0.0636\eta .$$

In the assumption of semi-infinite plane-parallel medium, $R_d(p_i, p_o)$ becomes a function of distance between p_i and p_o only. By replacing parameter p_i and p_o with the offset $\|p_i - p_o\|$ and expressing the vector parameter in terms of scalar values, we can rewrite R_d as follows:

$$R_d(p_i, p_o) = R_d(p_{i,x}, p_{i,y}, p_{i,z}, p_{o,x}, p_{o,y}, p_{o,z}) = R_s(p_{i,x} - p_{o,x}, p_{i,y} - p_{o,y}, p_{i,z} - p_{o,z}),$$

where $R_s(x,y,z) = \dfrac{\alpha'}{4\pi}[z_r\left(1+\sigma\sqrt{z_r^2+x^2+y^2+z^2}\right)\dfrac{e^{-\sigma\sqrt{z_r^2+x^2+y^2+z^2}}}{\left(\sqrt{z_r^2+x^2+y^2+z^2}\right)^3} +$

$$z_v\left(1+\sigma\sqrt{z_v^2+x^2+y^2+z^2}\right)\frac{e^{-\sigma\sqrt{z_v^2+x^2+y^2+z^2}}}{\left(\sqrt{z_v^2+x^2+y^2+z^2}\right)^3}]. \quad (3.3)$$

Note that R_s is a three-dimensional radial function with its value decaying exponentially with the distance.

The original equation (Equation 3.1) integrates p_i over the surface *A*. It seems that the integration is a two-dimensional process. But actually the integration is performed in three-dimensional space because p_i is a three-dimensional point. So we change the integral domain and rewrite the Equation 3.1 in a form of three dimensions:

$$S(p_x, p_y, p_z) = \int_X\int_Y\int_Z \hat{E}(x,y,z) R_d(x,y,z,p_x,p_y,p_z)\,dxdydz . \quad (3.4)$$

Substituting the Equation 3.3 into Equation 3.4 yields:

$$S(p_x, p_y, p_z) = \int_X\int_Y\int_Z \hat{E}(x,y,z) R_s(x-p_x, y-p_y, z-p_z)\,dxdydz .$$

Because the R_s is a symmetric function, we can change the sign of the parameter:

$$S(p_x, p_y, p_z) = \iiint\limits_{X\,Y\,Z} \hat{E}(x,y,z) R_s(p_x - x, p_y - y, p_z - z) dxdydz\,. \quad (3.5)$$

Obviously, the resultant equation is in the form of the following three-dimensional convolution of two functions:

$$S = \hat{E} \otimes R_s\,.$$

3.2 Calculating the Subsurface Illuminance Gradient

Given the subsurface illuminance function (Equation 3.1), it is not clear how to calculate the gradient of the subsurface illuminance. With the reformulated convolution form in Equation 3.5, the gradient could be derived straightforwardly.

Recall that the derivative of a convolution function is:

$$\frac{d}{dx}(f \otimes g) = \frac{df}{dx} \otimes g = f \otimes \frac{dg}{dx}\,.$$

The gradient of the subsurface illuminance is then derived as follows:

$$\begin{aligned}\nabla S &= \left(\frac{\partial S}{\partial x}, \frac{\partial S}{\partial y}, \frac{\partial S}{\partial z}\right) \\ &= \left(\frac{\partial}{\partial x}(\hat{E} \otimes R_s), \frac{\partial}{\partial y}(\hat{E} \otimes R_s), \frac{\partial}{\partial z}(\hat{E} \otimes R_s)\right) \\ &= \left(\frac{\partial \hat{E}}{\partial x} \otimes R_s, \frac{\partial \hat{E}}{\partial y} \otimes R_s, \frac{\partial \hat{E}}{\partial z} \otimes R_s\right), \qquad (3.6)\end{aligned}$$

or

$$= \left(\hat{E} \otimes \frac{\partial R_s}{\partial x}, \hat{E} \otimes \frac{\partial R_s}{\partial y}, \hat{E} \otimes \frac{\partial R_s}{\partial z}\right). \qquad (3.7)$$

As shown in the Equation 3.6 and Equation 3.7, once we have either $\nabla \hat{E}$ or ∇R_s, we could calculate the gradient of subsurface illuminace. However, since $\hat{E}$ does not have an analytic form, it is impossible to calculate $\nabla \hat{E}$ analytically. Therefore, in our implementation, we choose Equation 3.7 to calculate the gradient of subsurface illuminance. Note that

$$\frac{\partial R_s(x,y,z)}{\partial x} = \frac{\alpha'}{4\pi} x \left[h_r \frac{-e^{-\sigma s_r}}{s_r^4}\left(\sigma^2 s_r + 3\sigma + \frac{3}{s_r}\right) + h_v \frac{-e^{-\sigma s_v}}{s_v^4}\left(\sigma^2 s_v + 3\sigma + \frac{3}{s_v}\right)\right],$$

$$\frac{\partial R_s(x,y,z)}{\partial y}=\frac{\alpha'}{4\pi}y\left[h_r\frac{-e^{-\sigma s_r}}{s_r^4}\left(\sigma^2 s_r+3\sigma+\frac{3}{s_r}\right)+h_v\frac{-e^{-\sigma s_v}}{s_v^4}\left(\sigma^2 s_v+3\sigma+\frac{3}{s_v}\right)\right],$$

$$\frac{\partial R_s(x,y,z)}{\partial z}=\frac{\alpha'}{4\pi}z\left[h_r\frac{-e^{-\sigma s_r}}{s_r^4}\left(\sigma^2 s_r+3\sigma+\frac{3}{s_r}\right)+h_v\frac{-e^{-\sigma s_v}}{s_v^4}\left(\sigma^2 s_v+3\sigma+\frac{3}{s_v}\right)\right],$$

$$s_r=\sqrt{x^2+y^2+z^2+h_r^2}\,,$$

$$s_v=\sqrt{x^2+y^2+z^2+h_v^2}\,.$$

And the problem is reduced to evaluating the integral of convolution:

$$\frac{\partial S(p_x,p_y,p_z)}{\partial x}=\iiint_{X\,Y\,Z}\hat{E}(x,y,z)\frac{\partial R_s(p_x-x,p_y-y,p_z-z)}{\partial x}dxdydz\,,$$
$$\frac{\partial S(p_x,p_y,p_z)}{\partial y}=\iiint_{X\,Y\,Z}\hat{E}(x,y,z)\frac{\partial R_s(p_x-x,p_y-y,p_z-z)}{\partial y}dxdydz$$
$$\frac{\partial S(p_x,p_y,p_z)}{\partial z}=\iiint_{X\,Y\,Z}\hat{E}(x,y,z)\frac{\partial R_s(p_x-x,p_y-y,p_z-z)}{\partial z}dxdydz \qquad (3.8)$$

Although the integrals still do not have an analytic solution, by exploiting the properties of ∇R_s (see Figure 3.1), we could use some integration techniques such as Monte Carlo and quadrature methods to get a good approximation.

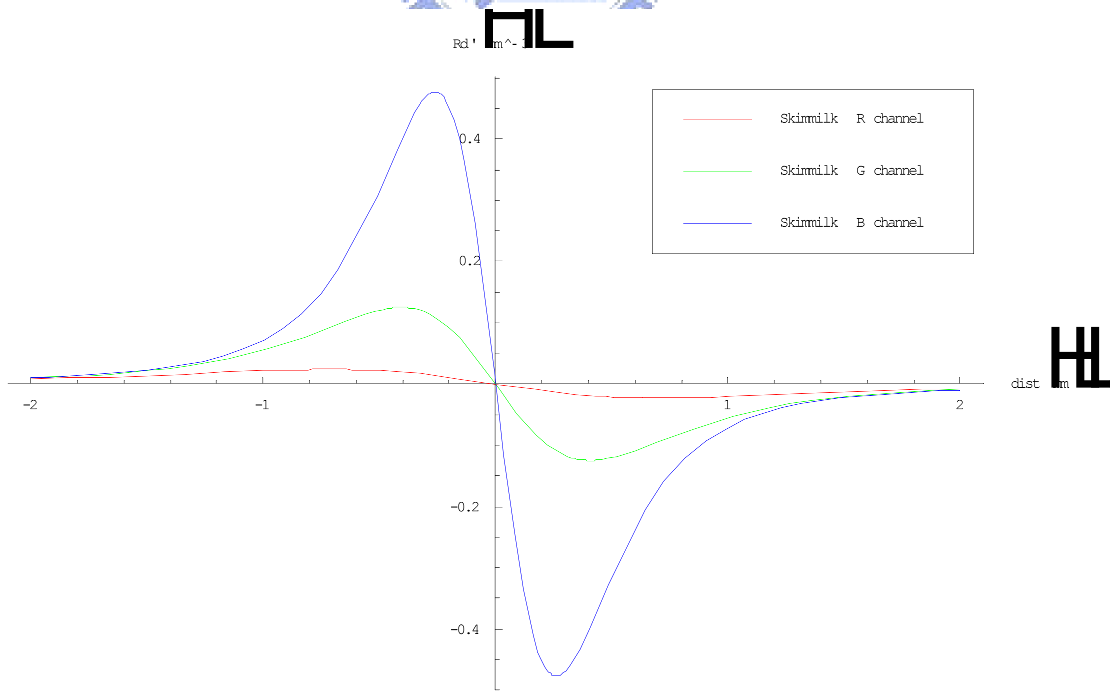


Figure 3.1: The graph of $\dfrac{\partial R_s}{\partial x}$.

Notice that if we choose

$$\frac{\partial S(p_x,p_y,p_z)}{\partial x} = \int_X\int_Y\int_Z \frac{\partial \hat{E}(x,y,z)}{\partial x} R_s(p_x - x, p_y - y, p_z - z)dxdydz\,,$$

$$\frac{\partial S(p_x,p_y,p_z)}{\partial y} = \int_X\int_Y\int_Z \frac{\partial \hat{E}(x,y,z)}{\partial y} R_s(p_x - x, p_y - y, p_z - z)dxdydz$$

$$\frac{\partial S(p_x,p_y,p_z)}{\partial z} = \int_X\int_Y\int_Z \frac{\partial \hat{E}(x,y,z)}{\partial z} R_s(p_x - x, p_y - y, p_z - z)dxdydz$$

to calculate the gradient of subsurface illuminance, it would be infeasible to get a good approximation even using the Monte Carlo method because the $\nabla \hat{E}$ could not be easily sampled.

3.3 Applying the Gradient to Interpolation

Once we calculate the gradient of subsurface illuminance, we could use the gradient to interpolate the subsurface illuminance more accurately. We use the same weighted average as proposed in Ward's caching technique [Ward et al. 1992] to interpolate the subsurface illuminance value:

$$S(p) = \frac{\sum_{k\in C} w_k(p)\left[S_k + (p - P_k)\cdot \nabla S_k\right]}{\sum_{k\in C} w_k(p)}\,, \tag{3.9}$$

where

p is the position of the point to be computed,

P_k is the position of cache k,

$w_k(p)$ is the weight of cache k with respect to p,

C is the set of valid caches, {cache k: $w_k(p) > 1/a$},

S_k is the computed subsurface illuminance of cache k,

∇S_k is the computed gradient of subsurface illuminace of cache k, and

a is a user-specified error bound.

The next problem is how to determine the spacing of samples, i.e., how to determine the weight of each sample, or how to estimate the error of each sample. The simplest and theoretically most accurate solution is directly using the inner product of the offset $(p - p_k)$ and ∇S_k derived in last section as our error estimate ε (assuming that the error due to interpolation is proportional to the estimated directional change of subsurface illuminance), i.e.,

$$\varepsilon \propto \Delta S = \Delta p_x \frac{\partial S}{\partial x} + \Delta p_y \frac{\partial S}{\partial y} + \Delta p_z \frac{\partial S}{\partial z} = \Delta p \cdot \nabla S\,.$$

Unfortunately, this will lead to bias the calculation. Since gradient is a very local property, areas that just happen to have small subsurface illuminance gradient would be sampled at low density, even though there could still be sudden changes in the subsurface illuminance value due to nearby surfaces. A possible solution is to use some approximation models to capture the largest expected gradient in determining the sample density so that we could not miss anything relevant.

To estimate the largest expected gradient, we introduce a *split-disk* model analogous to the *split-sphere* model proposed by Ward et al. [Ward et al. 1988]. The split-disk model, based on the assumption that the geometry is locally flat, relates the subsurface illuminance gradient to the variance V of the irradiance values within nearby surfaces. It assumes that a surface element is located at the center of a disk which approximates nearby surfaces (see Figure 3.2). The radius of the disk, R, is heuristically determined according to the material scattering property. Half of the disk is totally bright with constant irradiance K and the other half is totally dark with constant irradiance of zero. Because the variance of the irradiance values within the disk is V, we can conclude that $K=2V$. The split disk has the largest expected gradient possible for surfaces with variance V.

An approximate bound to the change of subsurface illuminance in the split disk, ε, is given by the first order Taylor expansion of the function S of one variable:

$$\varepsilon(u) \leq \left| (u - u_0) \frac{\partial S}{\partial u} \right|,$$

where u_o is the center of the disk and u is some other point on the disk. Note u_o and u are both one-dimensional value because we only care about the distance between two points.

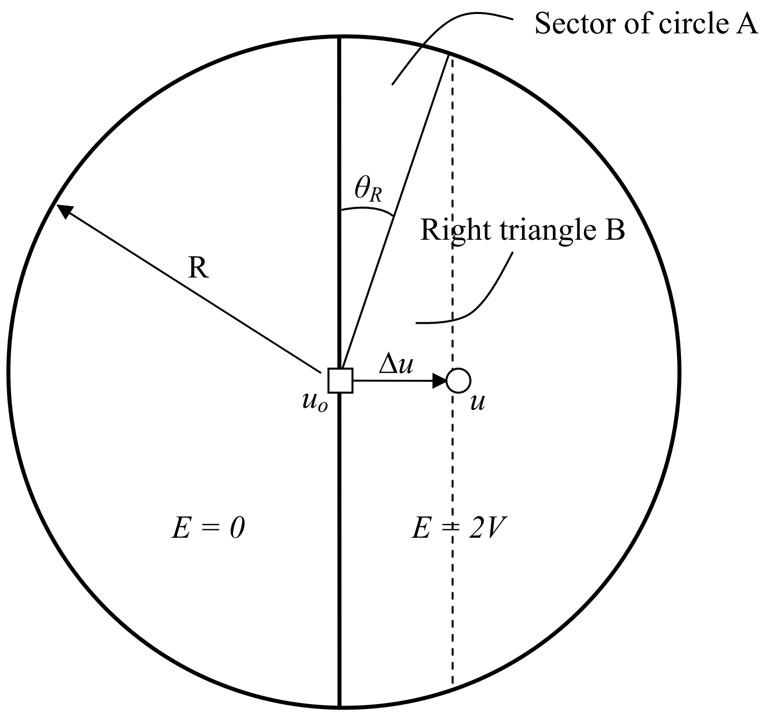


Figure 3.2: The split-disk model. A surface element is located at the center of a half-dark disk.

To derive

$$\frac{\partial S}{\partial u} = \lim_{\Delta u \to 0}\left(\frac{\Delta S}{\Delta u}\right),$$

we firstly consider a surface element moving from u_o to u, the change of S could be computed as twice of an integral over sector of circle A plus an integral over right triangle B (see Figure 3.2):

$$\Delta S = 2(\Delta S_A + \Delta S_B),$$

where

$$\Delta S_A = \int_0^{\theta_R}\int_0^R E(r)R_d(r)rdrd\theta \quad \text{and}$$

$$\Delta S_B = \frac{1}{2}\int_0^{\sqrt{R^2-\Delta u^2}}\int_0^{\Delta u} E(x,y)R_d(\sqrt{x^2+y^2})dxdy\,.$$

Unfortunately, we cannot find an analytic solution of the integral describing subsurface scattering over right triangle ΔS_B. Inspired by [Mertens et al. 2003] where a semi-analytical integration method is derived to solve the integral describing subsurface scattering over an arbitrary triangle, we can approximate ΔS by an integral over four sectors of circle, i.e.,

$$\Delta S \approx 4\Delta S_A .$$

The derivation of ΔS is as follows:

$$\Delta S \approx 4\int_0^{\theta_R}\int_0^{R} E(r)R_d(r)rdrd\theta .$$

Substituting $E(r) = 2V$, we have

$$\Delta S \approx 4\cdot 2V\int_0^{\theta_R}\int_0^{R} R_d(r)rdrd\theta .$$

Then replacing R$_d$(r) (Equation 3.2) yields

$$\Delta S \approx 8V\int_0^{\theta_R}\int_0^{R}\frac{\alpha'}{4\pi}\left[z_r\left(1+\sigma s_r\right)\frac{e^{-\sigma s_r}}{s_r^3}+z_v\left(1+\sigma s_v\right)\frac{e^{-\sigma s_v}}{s_v^3}\right]rdrd\theta$$

By rearranging terms, we can obtain

$$\Delta S \approx V\frac{2\alpha'}{\pi}\int_0^{\theta_R}\left[\int_0^{R} z_r\left(1+\sigma s_r\right)\frac{e^{-\sigma s_r}}{s_r^3}rdr+\int_0^{R} z_v\left(1+\sigma s_v\right)\frac{e^{-\sigma s_v}}{s_v^3}rdr\right]d\theta .$$

Since $\frac{ds_r}{dr}=\frac{1}{2}\frac{1}{\sqrt{r^2+z_r^2}}2r$, which implies $rdr = s_r ds_r$ and $\frac{ds_v}{dr}=\frac{1}{2}\frac{1}{\sqrt{r^2+z_v^2}}2r$, which implies $rdr = s_v ds_v$, we can change the integration variable r into the distance s_r and s_v and get

$$\Delta S \approx V\frac{2\alpha'}{\pi}\int_0^{\theta_R}\left[\int_{z_r}^{R_r} z_r\left(1+\sigma s_r\right)\frac{e^{-\sigma s_r}}{s_r^3}s_r ds_r+\int_{z_v}^{R_v} z_v\left(1+\sigma s_v\right)\frac{e^{-\sigma s_v}}{s_v^3}rds_v\right]d\theta$$

$$=V\frac{2\alpha'}{\pi}\int_0^{\theta_R}\left\{\left[-\frac{z_r}{u}e^{-\sigma u}\right]_{z_r}^{R_r}+\left[-\frac{z_v}{u}e^{-\sigma u}\right]_{z_v}^{R_v}\right\}d\theta$$

$$=V\frac{2\alpha'}{\pi}\int_0^{\theta_R}\left\{e^{-\sigma z_r}-\frac{z_r}{R_r}e^{-\sigma R_r}+e^{-\sigma z_v}-\frac{z_v}{R_v}e^{-\sigma R_v}\right\}d\theta$$

$$=V\frac{2\alpha'}{\pi}\theta_R\left(e^{-\sigma z_r}-\frac{z_r}{R_r}e^{-\sigma R_r}+e^{-\sigma z_v}-\frac{z_v}{R_v}e^{-\sigma R_v}\right)$$

$$=V\frac{2\alpha'}{\pi}ASin\left(\frac{\Delta u}{R}\right)\left(e^{-\sigma z_r}-\frac{z_r}{R_r}e^{-\sigma R_r}+e^{-\sigma z_v}-\frac{z_v}{R_v}e^{-\sigma R_v}\right),$$

where

$$R_r = \sqrt{R^2 + z_r^2}\ ,$$

and

$$R_v = \sqrt{R^2 + z_v^2}\ .$$

Since $\frac{\partial S}{\partial u} = \lim_{\Delta u \to 0}\left(\frac{\Delta S}{\Delta u}\right)$, we get

$$\frac{\partial S}{\partial u} \approx \lim_{\Delta u \to 0}\left(\frac{V\frac{2\alpha'}{\pi}ASin\left(\frac{\Delta u}{R}\right)\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right)}{\Delta u}\right).$$

Therefore,

$$\frac{\partial S}{\partial u} \approx V\frac{2\alpha'}{\pi}\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right)\lim_{\Delta x \to 0}\left(\frac{ASin\left(\frac{\Delta u}{R}\right)}{\Delta u}\right)$$

$$= V\frac{2\alpha'}{\pi}\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right)\frac{1}{R}.$$

Finally, we arrive at

$$\frac{\partial S}{\partial u} \approx V\frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right),$$

and the error estimate can be computed as

$$\varepsilon(u) \le \left|\frac{\partial S}{\partial u}(u - u_0)\right| \le \left|\frac{\partial S}{\partial u}\right||u - u_0|$$

$$= |u - u_0|V\frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r}\frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right).$$

We can extend our approximation to more complicated geometries by replacing *u* with vector-derived values:

$$\varepsilon(p) = \|p - p_0\|V\frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r}\frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right).$$

As in [Ward et al. 1988], the inverse of the error estimate

$$w_i(p) = \frac{1}{\varepsilon_i(p)} = \frac{1}{\|p - p_i\| V_i \frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r} - \frac{z_r}{R_r} e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v} e^{-\sigma R_v}\right)} \qquad (3.10)$$

is then used as our weight.

Substituting Equation 3.10 into Equation 3.9, the subsurface illuminance of some point of interest then can be computed by interpolating nearby caches. Additionally, the interpretation of the subsurface illuminance gradient allows us to perform the calculation separately for each sample wavelength or to combine into a spectral average. The latter was chosen for our implementation to save storage costs and evaluation time, although this may cause some blurring artifacts when the difference of scattering properties within each sample wavelength is too large.

3.4 A Three-Pass Technique for Rendering Translucent Materials

To integrate our model into Jensen's hierarchical evaluation method [Jensen et al. 2002], we use a three-pass approach, in which the first pass consists of computing the irradiance at selected points on the surface, the second pass generates all the necessary cache samples whose values including subsurface illuminance, gradient of subsurface illuminance, and variance of irradiance over nearby surface are computed by using the precomputed irradiance values, and the last pass re-uses the caches to produce the final image via interpolation.

Note that the second pass in our three-pass approach only generates caches. It doesn't use caches to interpolate any value. The interpolation using caches is done in the third pass. The reason why we do not combine generating caches and re-using caches into a single pass is discussed in section 3.5.

Pass 1: Sampling Irradiance

To solve Equation 3.1, firstly, we need to sample the irradiance function $E(x)$. There are a number of methods to generate sampling positions on the surface. In [Jensen et al. 2002], Turk's point repulsion algorithm [Turk 1992] is used to obtain a uniform sampling of points on a polygon mesh. However, a uniform sampling seems irrelevant in their hierarchical approach as each sample point is weighted by the area associated with it. Instead of using the Turk's point repulsion algorithm, which is complicated to implement, we use a very simple method to obtain the sampling positions on the polygon mesh. We directly use the centroid of each face as our sample point and assign the area of the face as the area associated with this sample point. If a model is too coarse and results in low-frequency noise in final image, we subdivide the model until the noise disappears.

For each sample point, we store the position, the area associated with the point, and a computed irradiance estimate. Since we focus on caching technique in this thesis, we do not use any rendering technique that accounts for global illumination (such as photon mapping and distributed ray racing) to compute the irradiance. We simply sum up irradiance contributions from each light source for evaluating direct illumination on each sample point.

As stated in [Jensen et al. 2002], the irradiance samples described in above section should be stored in a hierarchical structure so that,by clustering distant samples, we can exploit the exponential shaped fall-off of R_d, thus facilitating the evaluation of diffusion approximation. There are a number of hierarchical structures that we can use to store irradiance samples. Here we choose an octree as proposed by Jensen et al. [Jensen et al. 2001] in our implementation. Each node in the octree stores some information representing all irradiance samples inside the voxel associated with the node: the total flux Φ, the total area A, and the average position P. Note that the leaf node could have up to 8 irradiance samples for efficiency issue. The Φ, A, and P are computed as follows:

$$\Phi_i = \sum \Phi_j\ ,\quad A_i = \sum A_j\ ,\quad P_i = \frac{\sum A_j P_j}{\sum A_j}\ ,$$

where node *j* is node *i*'s child and if node *i* is a leaf node,

$$\Phi_i = \sum \hat{E}_k A_k\ ,\ \ A_i = \sum A_k\ ,\ \ P_i = \frac{\sum A_k P_k}{\sum A_k}\ ,$$

where irradiance sample *k* is within node *i*.

Furthermore, because our split-disk model needs to compute the variance of irradiance samples over nearby surface and the variance is derived as

$$\begin{aligned} \mathrm{var}[I]^2 &= E[(I - E[I])^2], \\ &= E[I^2] - (E[I])^2, \end{aligned} \tag{3.11}$$

where *I* is the irradiance distribution over nearby surface.

E[I] can be computed as

$$E[I] = \frac{\sum \hat{E}_i A_i}{\sum A_i} = \frac{\sum \Phi_i}{\sum A_i}, \tag{3.12}$$

where $\hat{E}_i$ is the irradiance of nearby sample *i* scaled by Fresnel term, and A_i is associated area of nearby sample *i*.

For the sake of *E[I²]*, which is computed as

$$E[I^2] = \frac{\sum \hat{E}_i^2 A_i}{\sum A_i} = \frac{\sum M_i}{\sum A_i}, \tag{3.13}$$

we store additional information M_i in each node:

$$M_i = \sum M_j\ ,$$

where node *j* is node *i*'s child and if node *i* is a leaf node,

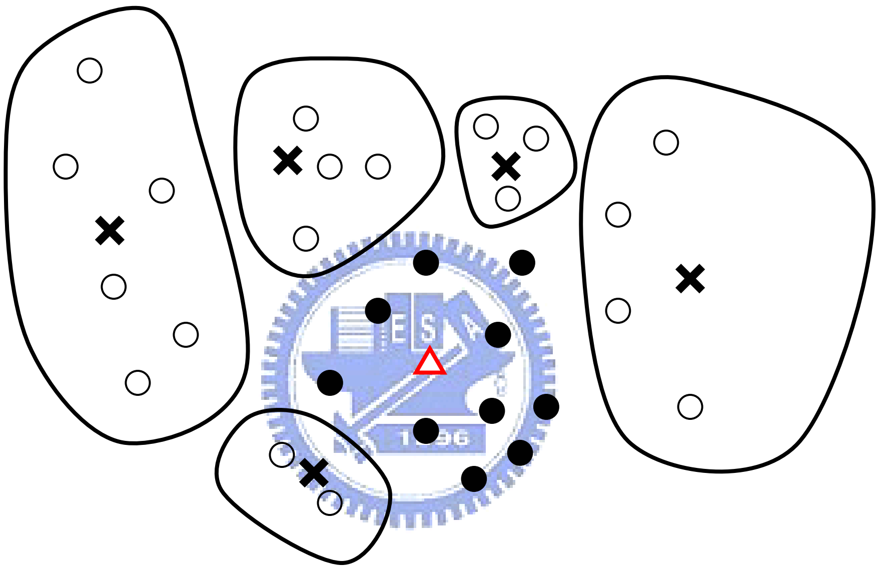
$$M_i = \sum E_k^2 A_k$$

where irradiance sample *k* is within node *i*.

Pass 2: Generating All the Necessary Caches

Before we state how to find where all the necessary caches should be generated, we present how to compute the values stored in each cache. The values we stored in each cache are subsurface illuminance *S*, gradient of subsurface illuminance ∇S, and variance *V* of irradiance over nearby surface. All these values are computed by using the precomputed irradiance values (distributed in Pass 1) stored in a hierarchical structure.

To compute the subsurface illuminance and the gradient of subsurface illuminance, we need to solve the integral in Equation 3.1 and Equation 3.8, respectively. If we directly sum the contribution from all the irradiance samples, i.e., solve the integral in Equation 3.1 and Equation 3.8 using uniform sampling, the computation would be too costly. Thus Jensen et al. proposed to use a hierarchical integration technique [Jensen et al. 2002], i.e., find a set of points *N* (see Figure 3.3) that could effectively represent all the irradiance samples to compute the integral.



● ○ Pre-computed irradiance samples (distributed in Pass 1)

✖ Representative points of a cluster of pre-computed irradiance samples

△ An interested point

● ✖ The set *N*

Figure 3.3: The set *N*.

Finding the set N

To find the set *N* with respect to a point *p*, we follow the algorithm proposed by Jensen et al. [Jensen et al. 2002], and begin by traversing the octree from the root. For each node, we first check whether the node is a leaf. If it is, we include all the irradiance samples within it to set *N*; otherwise, we then check whether *p* is inside the voxel associated with the node. If *p* is inside the voxel, we continue to traverse down to the children; if not, we then compare the solid angle with respect to the voxel with a user-specified value *m,* which controls the error. If the solid angle is larger than *m*, we traverse down to the children; otherwise, we add the representative point of the node to the set *N*. The pseudo code is listed below:

```
Traverse_octree(location $p$, node $i$)
{
    IF node $i$ is a leaf node
        Add all irradiance samples within voxel $i$ to set $N$
    ELSE IF point $p$ is inside voxel i
        FOR each child $j$ of node $i$
            Traverse_octree($p$, $j$)
    ELSE IF the solid angle of voxel $i > m$
        FOR each child $j$ of node $i$
            Traverse_octree($p$, $j$)
    ELSE
        Add representative point of node $i$ to set $N$
}
```

Computing *S*, ∇S, and *V*

After we find the set *N* with respect to a point *p*, the subsurface illuminance at *p* is computed by summing up the contribution from point *i* in *N*:

$$\begin{aligned} S(p) &= \int_{p_i \in A} E(p_i) R_d(p_i, p) dp_i \\ &= \int_{p_i \in A} R_d(p_i, p) E(p_i) dp_i \\ &= \int_{p_i \in A} R_d(p_i, p) d\Phi_{p_i} \\ &\approx \sum_{i \in N} R_d(P_i, p) \Phi_i , \end{aligned}$$

As for evaluating the gradient of subsurface illuminance at *p,* ideally, we should use some integration technique exploiting the properties of ∇R_s (Figure 3.1) to solve Equation 3.8. Also, the solution to the integral should be evaluated very fast, or the cost for computing the gradient will cancel out the gain from the interpolation and extrapolation. Fortunately, we don't need to devise some sophisticated integration technique to solve the integral. We could directly use the set *N* as our sample point to hierarchically solve the integral, just as we use the set *N* to solve Equation 3.1, though the plots of R_s (Figure 2.4) and ∇R_s (Figure 3.1) are not quite the same. The gradient of subsurface illuminance could be computed as follows:

$$\nabla S(p) = \sum_{i \in N} \nabla R_s (p_x - P_{i,x}, p_y - P_{i,y}, p_z - P_{i,z}) \Phi_i$$

Note that intuitively, it seems that we should use vector P_i - p as our parameter to calculate the gradient, but we use vector p - P_i instead. This is due to the sign change during the derivation of the convolution process (Equation 3.5).

Another value we have to compute is the variance *V* of irradiance over nearby surface (see Equation 3.11, 3.12, and 3.13):

$$\begin{aligned} V^2 &= E[(I - E[I])^2], \\ &= E[I^2] - (E[I])^2, \\ &= \frac{\sum_{i \in N} M_i}{\sum_{i \in N} A_i} - \left(\frac{\sum_{i \in N} \Phi_i}{\sum_{i \in N} A_i} \right)^2, \end{aligned} \qquad (3.14)$$

where *I* is the irradiance samples within nearby surface. Note we again use the set *N* to calculate the variance.

Determining where the caches should be generated

To determine where all the necessary caches should be generated, we firstly use ray casting to find a set of visible points *X*. For each point x_i in *X*, we check if there is any previously computed cache at nearby surface that could be used for interpolation, i.e., any cache *k* with $W_k(x_i) > 1/a$. If any, we leave x_i to next pass; otherwise, we generate a new cache at point x_i, evaluate $S(x_i)$ and $\nabla S(x_i)$ associated with the cache. Note that checking each point x_i in *X* in different order could change the resulting distribution of the caches. Here we choose bottom-up scan-line order in our implementation.

As stated is [Ward et al. 1988], each previously computed cache is only valid for interpolation in some finite space. The “valid domain” of each cache is computed as follows:

$$w_k(x_i) = \frac{1}{\varepsilon_k(x_i)} = \frac{1}{\|x_i - p_k\| V_k \frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right)} > \frac{1}{a},$$

which implies

$$\|x_i - p_k\| < \frac{a}{V_k \frac{2\alpha'}{\pi R}\left(e^{-\sigma z_r} - \frac{z_r}{R_r}e^{-\sigma R_r} + e^{-\sigma z_v} - \frac{z_v}{R_v}e^{-\sigma R_v}\right)}.$$

By using this “valid domain”, we could store these computed caches in an efficient data structure so that nearby caches can be located fast. As in [Ward et al. 1988], we choose an octree to store our computed caches. We begin by using the object’s bound box as the root of the octree. When a new cache is generated at some location, the octree is subdivided on demand to contain this value. Each cache value, with valid domain r_k, should be contained in an octree node whose side length is greater than 2*r_k, but not more than 4*r_k. This guarantees that the stored caches could be located in no more than 8 cubes on its own octree level, and a cache with a small valid domain will only be examined in close-range searches (typically only one or two cubes need to be traversed). An analogous two-dimensional extreme case is shown in Figure 3.4.

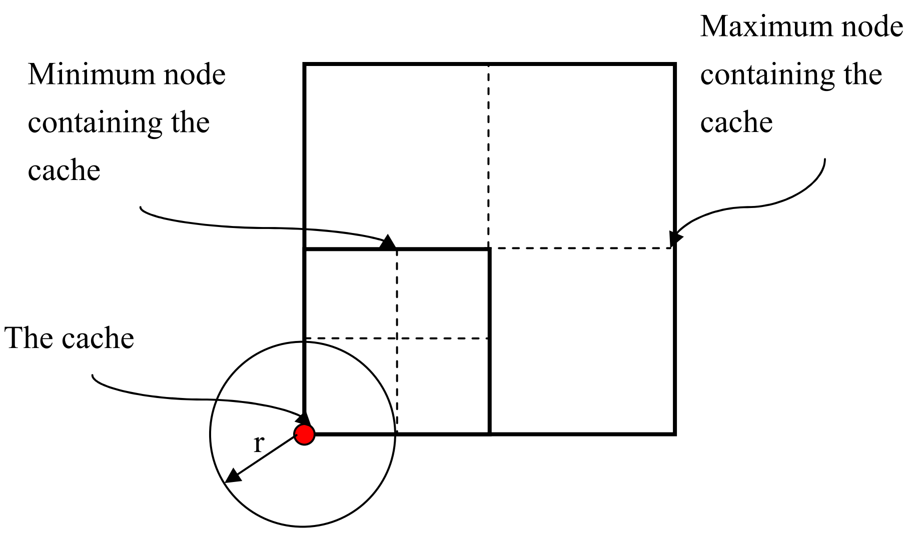

Minimum node containing the cache
Maximum node containing the cache
The cache
r

Figure 3.4: An extreme case of 2D octree node containing a cache with valid domain *r*.

The pseudo code for storing computed caches is listed below:

```
StoredCaches(cache $k$, node $i$)
{
    IF ($r_k$ < (side length of node $i$) / 4)
        IF node $i$ does not have child
            Split node i into node $j$, for $j$ = 1 ... 8
        Determine which child $j$ containing cache $k$
        StoredCaches($k$, $j$)
    ELSE
        Store cache $k$ into node $i$
}
```

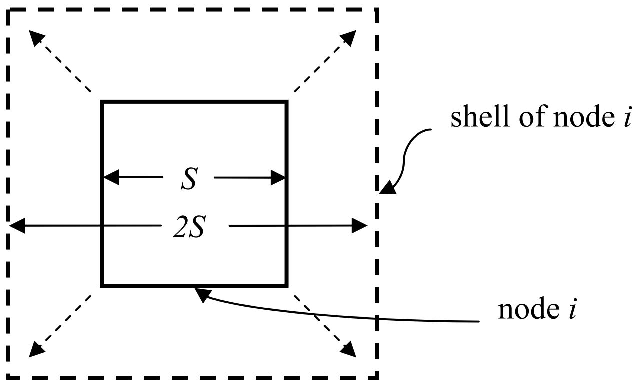


Figure 3.5: The shell of a node is obtained by expanding its side length from *S* to 2*S*.

To search the octree for a cache *k* whose valid domain r_k contains test point *x*, the following recursive procedure is used:

```
FindAnyValidCache(point $x$, node $i$)
{
    FOR each caches $k$ stored in node $i$
        IF $|x-p_k| < r_k$
            Return True;
    FOR each child $j$ of node $i$
        IF $x$ is within the “shell” of the node $j$ (Figure 3.5)
            Return FindAnyValidCache($x$, $j$)
    Return False;
}
```

Pass 3: Reusing Caches to Interpolate Image

After we generate all the necessary caches, we evaluate each point x_i in X (which is a set of visible points computed by ray casting) using Equation 3.9. Again, we use the octree to find the set of valid caches *C*. The pseudo code is listed below:

```
FindAllValidCache(point $x$, node $i$)
{
    FOR each cache $k$ stored in node $i$
        IF $|x-p_k| < r_k$
            Include cache $k$ to $C$
    FOR each child $j$ of node $i$
        IF $x$ is within the “shell” of the node $j$ (Figure 3.5)
            FindAllValidCache($x$, $j$)
}
```

3.5 Discussion

In Ward's original paper [Ward et al. 1988], though the split-sphere model is only a crude estimate of the gradient magnitude, generating cache and reusing cache were proposed to be done in a single pass. This results in some rather disturbing artifacts due to inaccurate interpolation and extrapolation. Figure 3.6 shows an example of inaccurate interpolation.

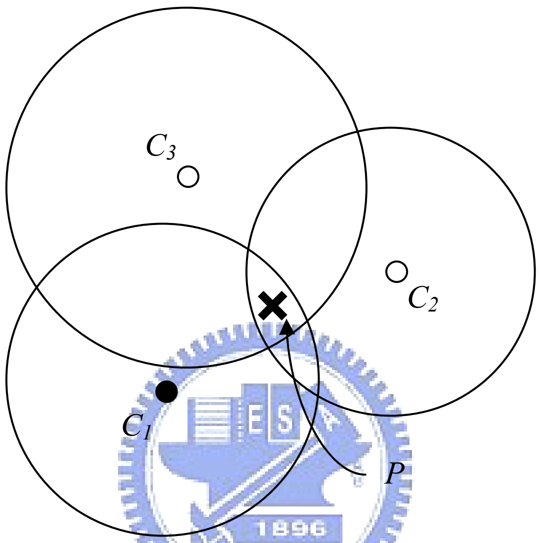


Figure 3.6: Inaccurate interpolation due to a single pass scan-line rendering.

In single-pass scan-line rendering (assuming in bottom-up scan-line order), the four points P, C_1, C_2, and C_3 are examined in the order of C_1, P, C_2, and C_3. Assuming that we generate C_1, C_2, and C_3 as caches; and P is within their valid domains, we can find that when we examine point P, the cache C_2 and C_3 have not yet been generated, thus we will use only one cache C_1 to interpolate point P. If we use two pass, i.e., the first pass generates the cache C_1, C_2, and C_3; then the second pass interpolates P using these three caches. That way will get more accurate interpolation.

To solve this problem, Ward then proposed *irradiance gradient* [Ward and Heckbert 1992], which computes actual irradiance gradient, not just a directionless upper bound, to make the interpolation and extrapolation significantly more accurate, thus avoiding separating generating cache and reusing cache into two passes. However, in our translucent caching technique, although we have successfully computed the gradient of subsurface illuminance, we still need two-pass calculation to get an acceptable image. This is mainly due to the fact that subsurface illuminance gradient is much greater than irradiance gradient. Figure 3.7 compares images rendered in one-pass and two-pass approaches.

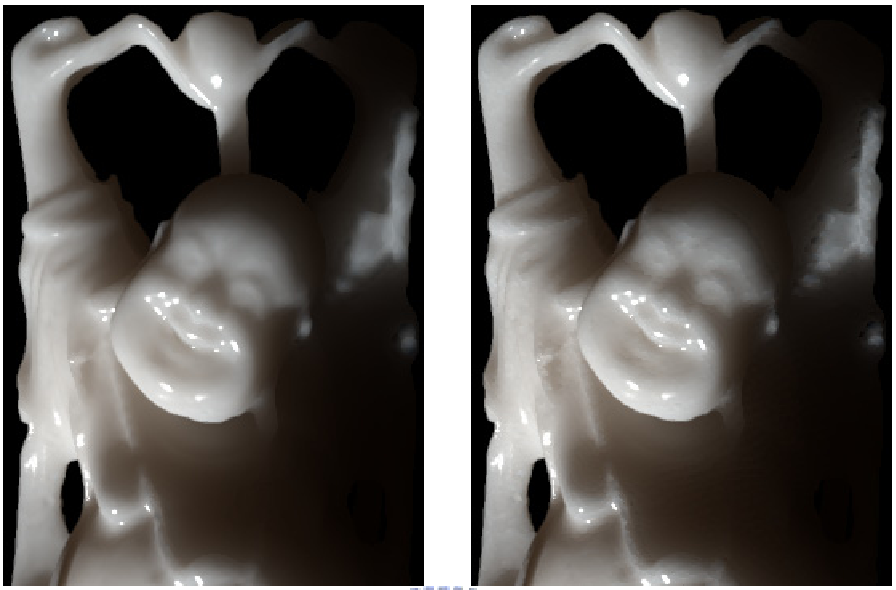


Figure 3.7: Left: two-pass approach. Right: one-pass approach.

As for computing the variance of irradiance for split-disk model, we didn't use the most accurate calculation (Equation 3.14). The most accurate calculation should be evaluated using

$$\begin{aligned} V^2 &= E[(I - I_p)^2] \\ &= \frac{\sum (I_i - I_p)^2 A_i}{\sum A_{i_i}} \\ &= \frac{\sum I_i^2 A_i - 2I_p \sum I_i A_i + I_p^2 \sum A_i}{\sum A_{i_i}} \\ &= \frac{\sum M_i - 2I_p \sum W_i + I_p^2 \sum A_i}{\sum A_{i_i}} \end{aligned}$$

where I_p is the irradiance at the location p. Because using this equation didn't affect the final image quality very much, it is not worthy of re-sampling irradiance at each pixel unless the cost of re-sampling irradiance is negligible.

Instead of the split-disk model, we could use other approaches to determine the cache distribution. One alternative is applying a filter to ∇R_s, i.e., convolving ∇R_s with some filter function G (e.g., Gaussian function):

$$F_x = \frac{\partial R_s(x,y,z)}{\partial x} \otimes G,$$

$$F_y = \frac{\partial R_s(x,y,z)}{\partial y} \otimes G,$$

$$F_z = \frac{\partial R_s(x,y,z)}{\partial z} \otimes G.$$

And then by using the resultant functions, we can derive average subsurface illuminance gradient:

$$\nabla S^* = (\hat{E} \otimes F_x, \hat{E} \otimes F_y, \hat{E} \otimes F_z).$$

This approach has advantages of capturing the average subsurface illuminance gradient more accurately and computing ∇S^* by using the same hierarchical integration technique for S and ∇S. Unfortunately, we didn't find any filter function that could produce analytic solutions of function *F*. Thus we leave this study as our future work.

Chapter 4

Results

In this chapter we present several results from our implementation of the rendering technique. All the images are rendered by a Monte Carlo ray tracer at the resolution of 1024x1024 pixels. Our timings are recorded on a PC with an AMD Athlon XP 1800+ (1.53GHz) processor and 512 MB main memory.

To validate our algorithm, we have implemented Jensen's hierarchical rendering technique [Jensen et al. 2002] and compare the images generated by Jensen's hierarchical rendering technique with ours (Figure 4.1). The Dragons are rendered with material Skimmilk [Jensen et al. 2001]. Our approach gives almost the same visual appearance while achieving about one order of magnitude speedup.
Table 4.1 illustrates the performance and timing statistics of our approach with different models. N is the number of samples for sampling irradiance. C is the number of total caches. H is the number of total hit pixels. R is the ratio of C to H. RMS is the root-mean-squared error with respect to the averaged RGB value of each pixel. T_1 and T_2 are the rendering times used in Jensen's approach and ours, respectively. The time for sampling the irradiance and computing the specular term is not taken into account. Note the ratio of total caches to total hit pixels is about 2%. Other 98% pixels could be calculated via interpolation. While we only use such small amount of caches, we still get very good visual appearance with RMS smaller than 0.01. The speedup ratio is dependent on the average cost for computing subsurface illuminance of each cache. The more the computation of subsurface illuminance costs (e.g. for better image quality), the higher speedup we get. Typically, it varies from 5 to 15.

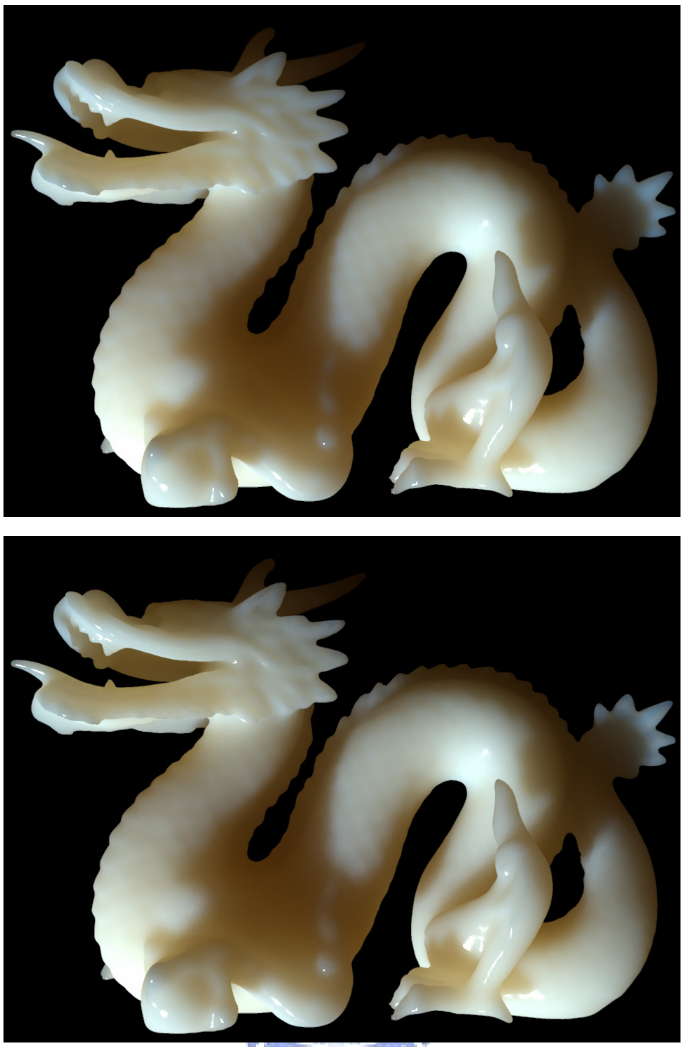


Figure 4.1: Top: the image rendered using Jensen's hierarchical rendering technique.

Bottom: the image rendered using our caching technique.

Model	Buddha	Dragon	Igea	Teapot
N	293,232	202,520	268,686	261,632
C	6,845	8,207	8,240	5,991
H	342,124	446,781	491,436	316,046
R	1.84%	2.00%	1.68%	1.90%
RMS	0.0058	0.0062	0.0040	0.0049
T_1 (sec.)	62.50	76.47	88.09	41.17
T_2 (sec.)	7.53	6.50	7.89	4.84
Speedup	8.30	11.76	11.16	8.51

Table 4.1: Overview of performance with different models.

Figure 4.2 shows distribution of the caches and Figure 4.3 show the direct illumination. Figure 4.4 shows the visualization of the gradient of subsurface illuminance ∇S, where the XYZ channel of the gradient is mapped to the RGB channel in the images. The brightness of a pixel corresponds to the magnitude of the gradient. Figure 4.5 is the variance V of the irradiance. Figure 4.6 then shows these four models rendered with material Marble. Note the models are rendered at scale of 7cm.

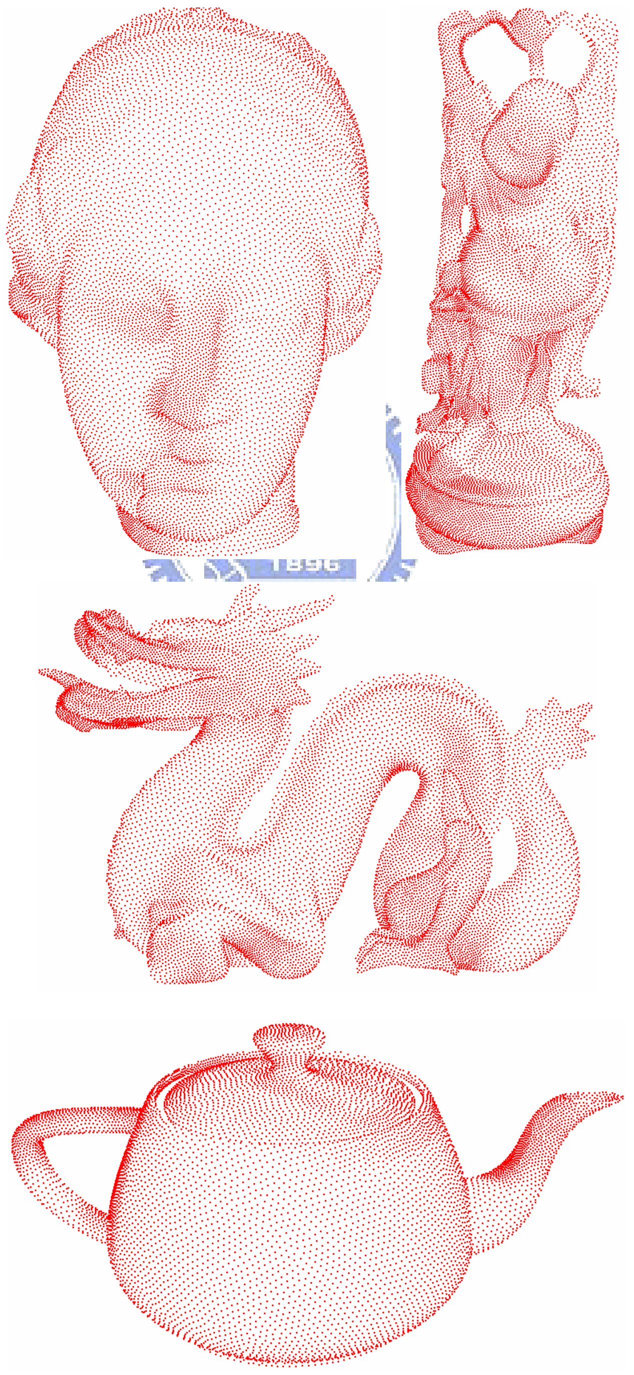


Figure 4.2: The cache distribution.

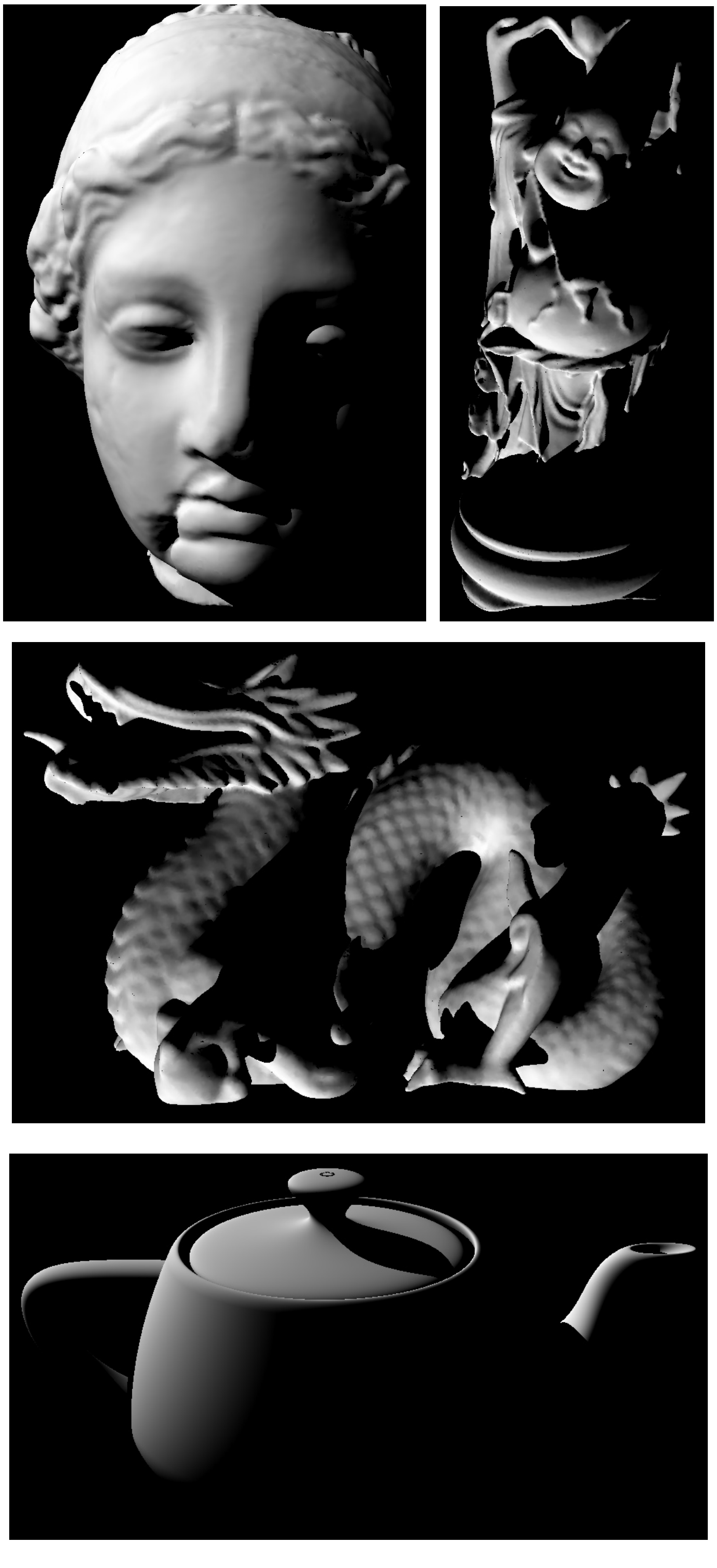


Figure 4.3: The irradiance E (direct illumination only).

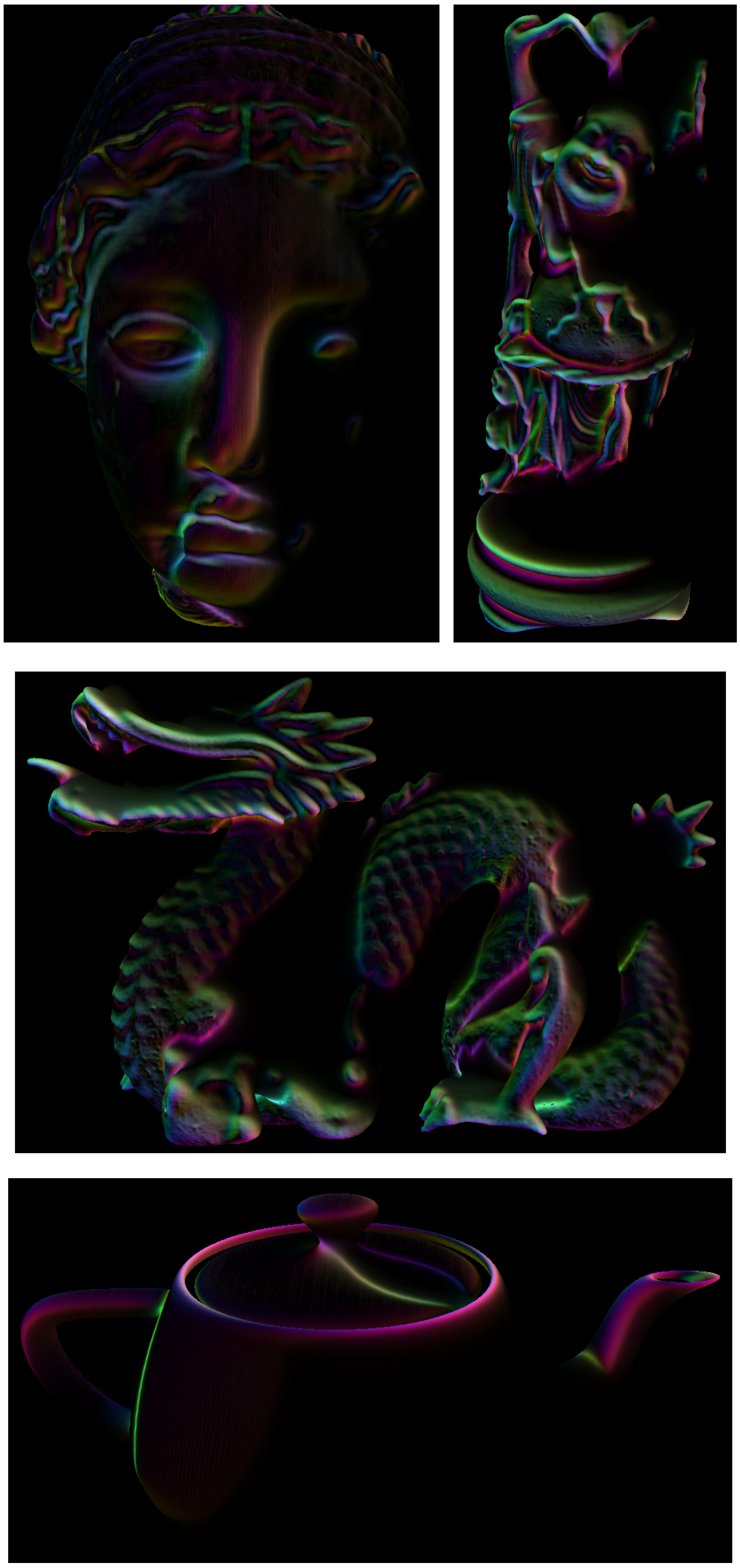


Figure 4.4: The visualization of ∇S .

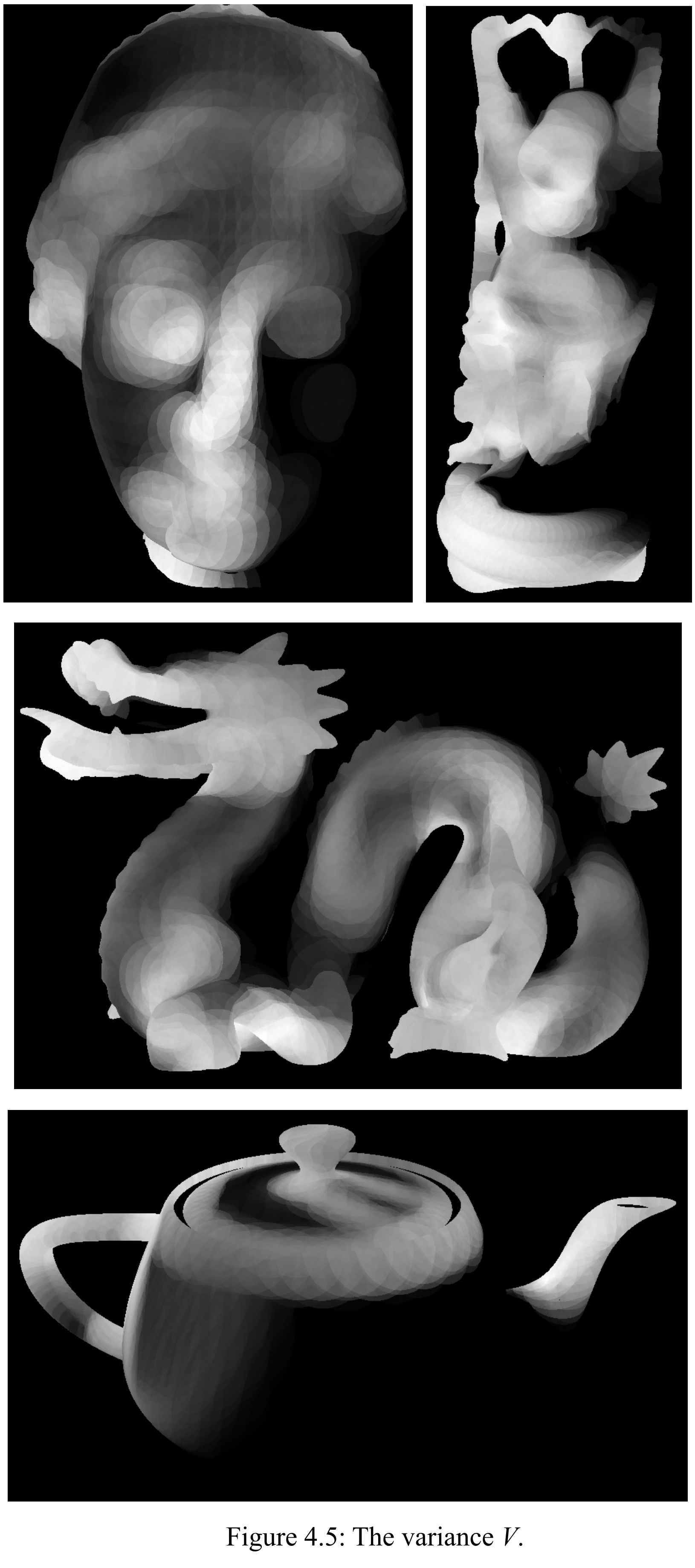


Figure 4.5: The variance V.

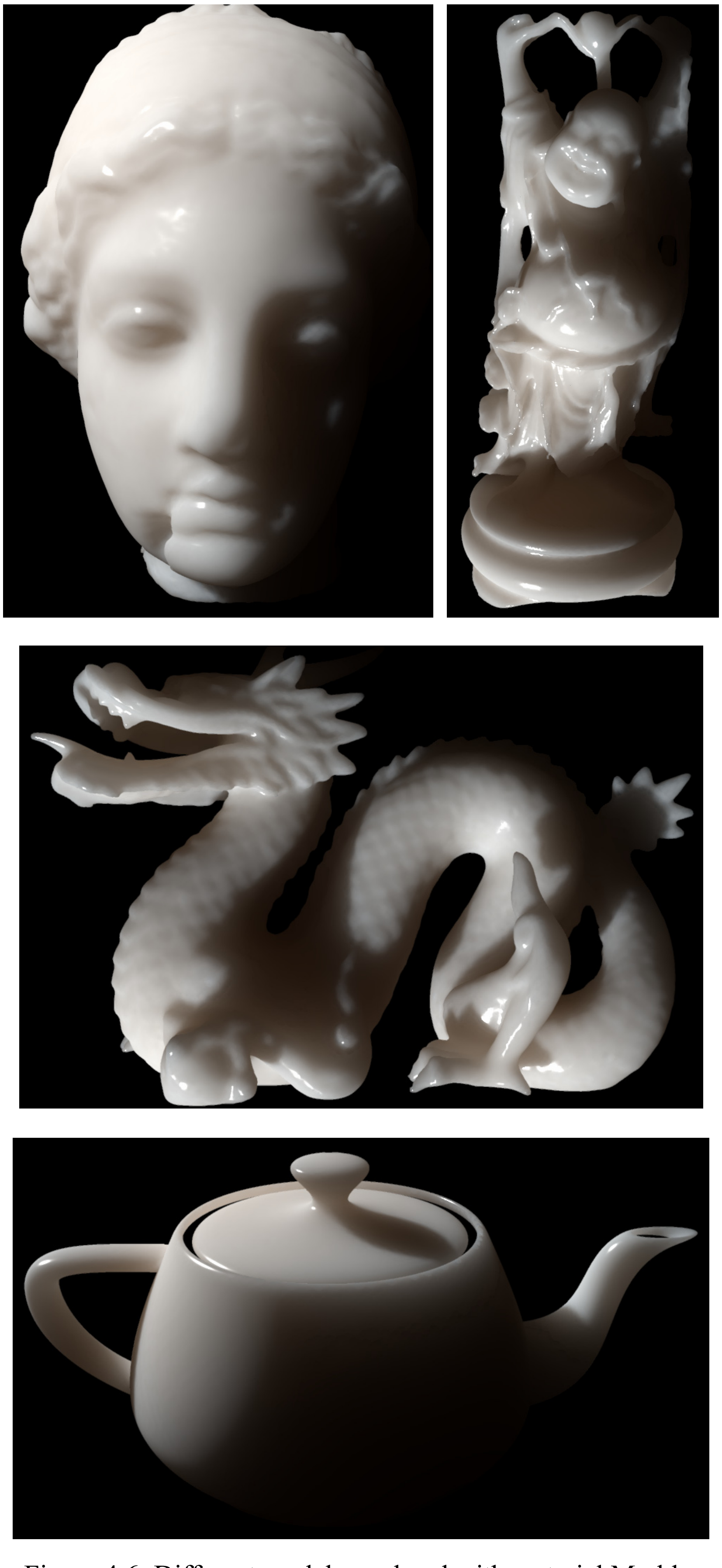


Figure 4.6: Different models rendered with material Marble.

Finally, we show that our caching scheme could be tailored to different sizes of objects and different kinds of materials. In Figure 4.7, the scale of Buddha changes from 7cm, 5cm, 3cm, 1cm, to 5mm. And the dragons in Figure 4.8 are rendered with material Skin1 and Skin2 [Jensen et al. 2001].

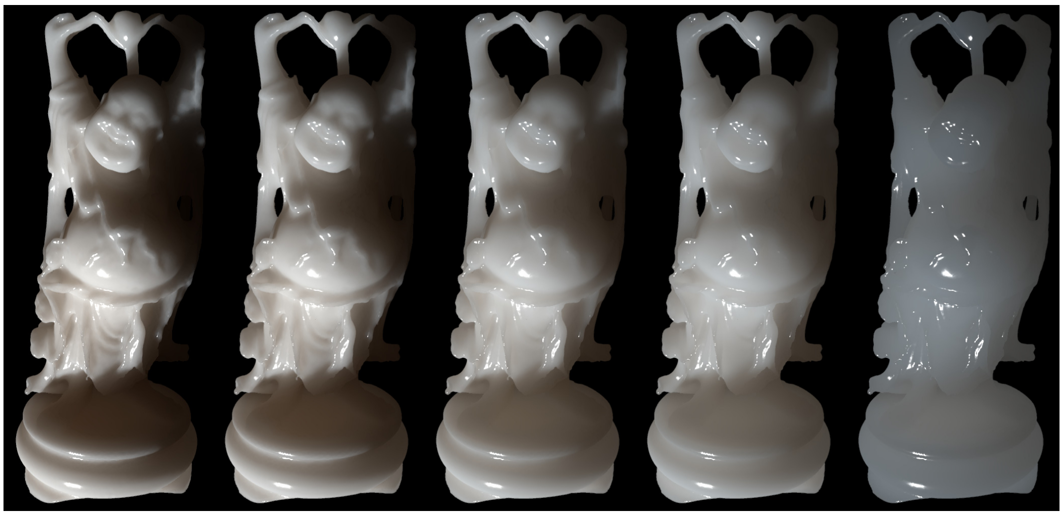


Figure 4.7: Changing the scale of subsurface scattering for Marble.

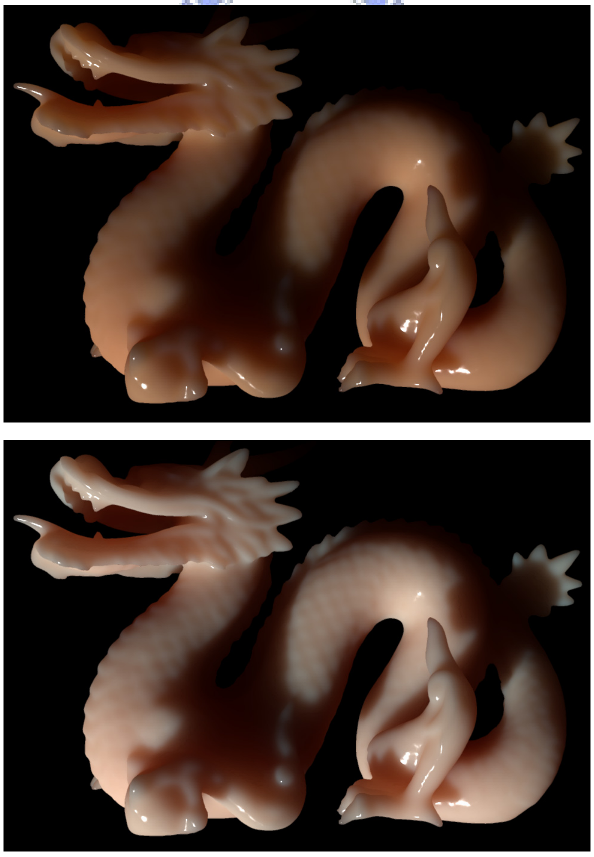


Figure 4.8: Top: dragon with material Skin1.
Bottom: dragon with material Skin2 [Jensen et al. 2001].

Chapter 5

Conclusion and Future Work

In this chapter, we give a brief summary of the thesis, and suggest some directions of future work.

5.1 Summary

We present an efficient caching technique for rendering translucent materials. We successfully integrate irradiance caching technique proposed by Ward et al. [Ward et al. 1988] into a rapid hierarchical rendering technique for translucent materials proposed by Jensen and Buhler [Jensen and Buhler 2002]. Our approach is efficient for producing high-quality images with high resolution and is particularly useful in animations. It also integrates seamlessly with Monte Carlo ray tracing, scanline rendering, and global illumination methods. Our results demonstrate the speedup could be achieved up to one order of magnitude compared to the hierarchical rendering technique proposed by Jensen and Buhler [Jensen and Buhler 2002] with negligible visual difference in the final images. The success of our approach is mainly due to the caching technique using the gradient of subsurface illuminance.

The contributions we achieve in this thesis can be summarized as:

- We show that the classic irradiance caching originally used in the ray tracing field for computing indirect illumination could also be extended to render translucent materials.
- We conclude that the dipole diffusion approximation is a 3D convolution process.
- We reformulate the calculation of the gradient of subsurface illuminance using

convolution.

- We propose a split-disk model analogous to Ward's split-sphere model [Ward et al. 1988] to determine the spacing of samples.
- We successfully integrate the irradiance caching technique [Ward et al. 1988] into the rapid hierarchical rendering technique for translucent materials [Jensen and Buhler 2002] with up to one order of magnitude speed-up.

5.2 Future Directions

Further improvements include exploring more complex models to determine the cache distribution and devising a reasonable model to determine the radius of split-disk and the upper bound of valid domain of each cache, which are set heuristically in our implementation.

Combining our caching technique with programmable graphics hardware is also an interesting topic. The vertex sent to graphics pipeline could be treated as cashes in our caching scheme and the interpolation using subsurface illuminance gradient could possibly be implemented in fragment shader. This approach should enhance the image quality of recent researches [Mertens et al. 2003; Hao and Varshney 2004; Hao et al. 2003; Lensch et al. 2002; Carr et al 2003; Dachsbacher and Stamminger 2003], which are all restricted to mesh-based objects with Gouraud Shading.

Finally, it would also be useful to investigate the accuracy of the dipole diffusion approximation in the presence of complex geometries and to find the solution methods for the heterogeneous materials, which is still a big challenge for current researchers.

Bibliography

CARR, N. A., HALL, J.D., AND HART, J. C. 2003. GPU algorithms for radiosity and subsurface scattering. In *Proceedings of Graphics Hardware 2004*, 51-59.

CHRISTENSEN, P. H. 2003. Global illumination and all that. *Chapter 3, SIGGRAPH Course Note #9*.

DACHSBACHER, C. AND STAMMINGER, M. 2003. Translucent shadow map. In *Proceedings of the 14th Eurographics Symposium on Rendering*, 197-201.

DORSEY, J., EDELMAN, A., LEGAKIS, J., JENSEN, H. W., AND PEDERSEN, H.K. 1999. Modeling and rendering of weathered stone. In *SIGGRAPH 99*, *Computer Graphics Proceeding*, 225-234.

DUTRE, P., BEKAERT P. AND BALA, K. 2003. *Advanced Global Illumination*, AK Peters.

FARELL, T. J., PATTERSON, M. S. AND WILSON, B. 1992. A diffusion theory model of spatially resolved, steady-state diffuse reflectance for the noninvasive determination of tissue optical properties in vivo. In *Med. Phys.,* 19:879-888.

HANRAHAN, P. AND KRUEGER, W. 1993. Reflection from layered surfaces due to subsurface scattering. In *Computer Graphics (SIGGRAPH '93 Proceedings),* Vol. 27. 165-174.

HAO, X., BABY, T., AND VARSHNEY, A. 2003. Interactive subsurface scattering for translucent meshes. In ACM *Symposium on Interactive 3D Graphics.* Monterey, CA, 75-82.

HAO, X. AND VARSHNEY, A. 2004. Real-time rendering of translucent meshes. In *ACM Transaction on Graphics*.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*, AK Peters.

Jensen, H. W. and Buhler, J. 2002. A rapid hierarchical rendering technique for translucent materials. In *SIGGRAPH 2002, Computer Graphics Proceedings,* 576-581.

Jensen, H. W., Marschner, S., Levoy, M., and Hanrahan, P. 2001. A practical model for subsurface light transport. In *SIGGRAPH 2001, Computer Graphics Proceedings,* 511-518.

Lensch, H., Gosele, M., Bekaert, P., Kautz, J., Magnor, M., Lang, J., and Seidel, H. P. 2002. Interactive rendering of translucent objects. In *Proc. IEEE Pacific Graphics 2002.* 214-224.

Kienle, A., Patterson, M. S. 1997. Improved solutions of the steady-state and the time-resolved diffusion equations for reflectance from a semi-infinite turbid medium. In *J. Opt., Soc. Am. A/Vol. 14, No. 1.*

Mertens, T., Kautz, J., Bekaert, P., Seidel, H.-P. and Van Reeth, F. 2003. Interactive rendering of translucent deformable objects. In *Proceedings of the 14th Eurogrpahics symposium on Rendering,* 130-140.

Palmer, J. M. 2003. Radiometry and photometry FAQ.

http://www.optics.arizona.edu/Palmer/rpfaq/rpfaq.htm

Pharr, M. and Hanrahan, P. 2000. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *SIGGRAPH 2000, Computer Graphics Proceedings,* 75-84.

Pharr, M. and Hanrahan, P. 2000. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *SIGGRAPH 2000, Computer Graphics Proceedings,* 75-84.

Poirier, G. 2004. Human skin modeling and rendering. *University of Waterloo Technical Report Number CS-2004-05.*

Stam, J. 1995. Multiple scattering as a diffusion process. In *Eurographics Rendering Workshop 1995,* Eurographics.

Turk, G. 1992. Re-tiling polygonal surfaces. In *Computer Graphics* (*SIGGRAPH '92*

Proceedings), vol. 26, 55-64

WARD, G. J. AND HECKBERT, P. S., 1992. Irradiance Gradient. In *Proceedings of Eurogrpahics Workshop on Rendering*.

WARD, G. J., RUBINSTEIN, F. M. AND CLEAR, R. D. 1988. A Ray Tracing Solution for Diffuse Interreflection. In *Computer Graphics*.