國立交通大學

多媒體工程研究所

碩 士 論 文

可分析的即時容積陰影演算法

Real-Time Analytical Volumetric Shadows Algorithm

研 究 生：翁 戰

指導教授：施仁忠　教授

中 華 民 國 九 十 九 年 六 月

可分析的即時容積陰影演算法
Real-Time Analytical Volumetric Shadows Algorithm

研 究 生：翁　戰　　　　Student：Jan Weng

指導教授：施仁忠　　　　Advisor：Zen-Chun Shih

國 立 交 通 大 學
多 媒 體 工 程 研 究 所
碩 士 論 文

A Thesis

Submitted to Institute of Multimedia Engineering

College of Computer Science

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年六月

可分析的即時容積陰影演算法

Real-Time Analytical Volumetric Shadows Algorithm

研究生：翁　戰　　　　　　　指導教授：施仁忠 教授

國立交通大學多媒體工程研究所

摘　　要

　　本論文透過數學分析的方式計算容積陰影的輻射傳送方程式（Radiative Transport Equation），經由這種輻射傳送方程式產生的結果和一般由數值方法計算的輻射傳送方程式所產生的結果相比, 本論文所採用的方式產生之結果較具有物理意義, 並且本論文在影子的空間中提出新的取樣方法, 和簡化過的可分析輻射傳送方程式, 再將場景分為沒有介質的場景和有介質的場景, 針對有介質的場景作縮減取樣, 計算有介質場景的輻射強度, 最後利用雙邊濾鏡（Bilateral Filter）將兩個場景結合, 和其他方法相比, 本論文的演算法不需要前置處理, 而且在非等向性的介質環境中, 可以達到互動式的效能, 在等向性的介質環境, 可以達到即時的效能
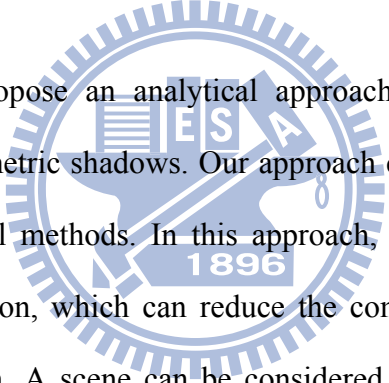
Real-Time Analytical Volumetric Shadows Algorithm

Student: Jan Weng   Advisor: Prof. Zen-Chung Shih

Institute of Multimedia Engineering
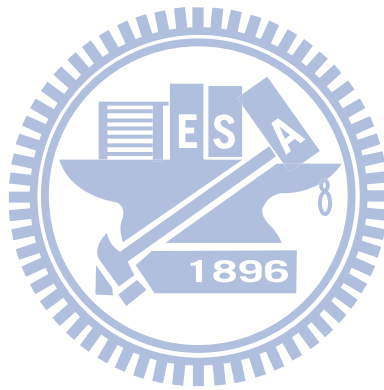National Chiao Tung University

ABSTRACT

In this thesis, we propose an analytical approach to compute the radiative transport equation of volumetric shadows. Our approach can achieve a more realistic result than using numerical methods. In this approach, a new sampling method is applied to the shadow region, which can reduce the computation for the analytical radiative transport equation. A scene can be considered as consisting of two parts, with and without participating media respectively. We only need to downsample the past with participating media, and calculate its radiance by using the simplified analytical radiative transport equation. Finally, we combine these two parts by using a bilateral filter. Compare to other techniques, our algorithm relies neither on pre-computation nor on storage. For a scene with anisotropic participating media, we can achieve interactive frame rate. For a scene with isotropic participating media, its performance can be real-time.

Acknowledgements

First of all, I would like to thank my advisors, Dr. Zen-Chung Shih for their supervision and helps in this work. Then I want to thank all the members in Computer Graphics & Virtual Reality Laboratory for their comments and instructions. I want to dedicate the achievement of this work to my family. Without their support, I couldn't fully on my study.

Contents

List of Figures

Chapter 1
Introduction

1.1 Motivation

Realistic rendering is of concern to the movie and gaming industries. However, efficiently simulating accurate radiance transport in participating media is a challenging task until today. Computer graphics make simplifying assumptions about scenes to maintain interactivity. One common assumption limits light interactions to surfaces, ignoring contributions from light scattered by particles in the air, which means that radiance is constant along rays between surfaces. However, there are many real-world situations where this assumption is inaccurate, like scattering from particles in the air, fog, a shaft of light or volumetric shadows.

Recently introduced interactive methods, Sun et al. [1], more realistically account for single scattering effects. However, these approaches ignore volumetric shadowing where occluders block light from scattering in shadow regions. Volumetric effects have been well studied in the context of offline rendering. While Monte-Carlo techniques or volumetric photon mapping produces volumetric effects, these methods require extensive computation. Wyman et al. [2] propose an algorithm to obtain volumetric shadows, and compute radiance by Sun et al. [1]. Compare to Sun et al. [1], Pegoraro et al. [3] produce a robust result. Pegoraro et al. [3] present a novel analytical approach to single scattering in isotropic participating media. Furthermore, Pegoraro et al. [4] present an analytical approach to single scattering in anisotropic

participating media. However, the computation of the analytical approach is more expensive than other approaches in volumetric shadows.

In our approach, based on Wyman et al. [2] and Pegoraro et al. [3] [4], we propose a new sampling method in the shadow region and reduce analytical radiative transport equation. The FPS of Pegoraro et al. [4] is low in the volumetric shadowing scene which contains anisotropic media. In our approach, the performance can be interactive in the same situation. Compare to other real-time approach, our approach sacrifices few performance to obtain a better result, but it is still real-time in isotropic media, interactive in anisotropic media.

Researchers have proposed a number of methods for interactive volumetric shadows. These methods fall into three categories, based on shadow volumes, ray marching, or volume slicing. The methods based on shadow volumes may produce an artifact as shown in Figure1.1. The methods based on volume slicing and ray marching produce aliasing as shown in Figure1.2.
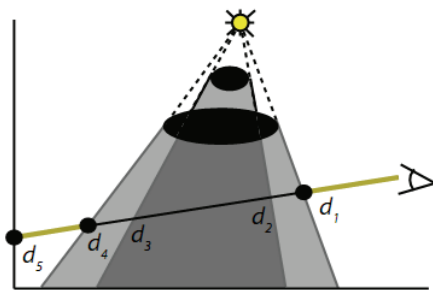
Figure1.1 The region between d_2 and d_3 is too dark
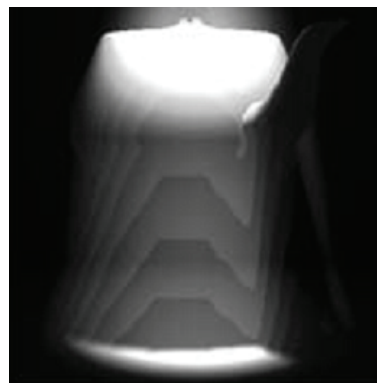
Figure1.2 Ray marching produce aliasing

Our approach uses shadow volume to detect shadow regions. Then we use ray marching to avoid the artifact as shown in Figure1.1, and reduce ray marching costs by using our sampling method and simplified analytical radiative transport equation. The performance is real-time in most scenes. The result is better than other real-time volumetric shadowing techniques.

1.2 System Overview
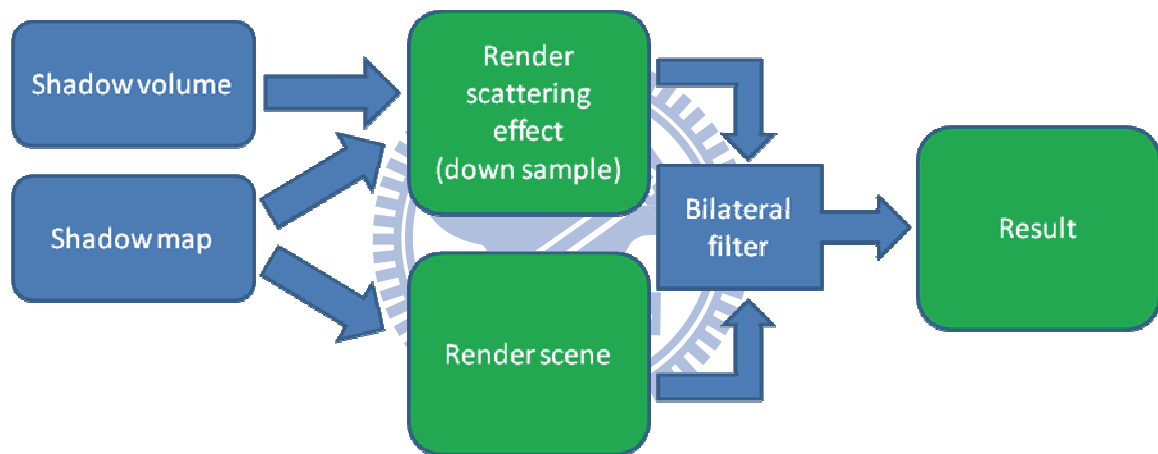
Figure1.3 The system overview

Figure 1.3 gives an overview of our system. First, user creates shadow map from the light view, and shadow volume. The Shadow volume is not tradition; user just creates the frontmost shadow polygon and the final shadow polygon from the eye view. To enter the core of our system, the scene is considered into two parts, "Render Scene" (No participating media, a general scene) and "Render Scattering Effect" (Only render participating media effects) as shown in Figure 1.4.
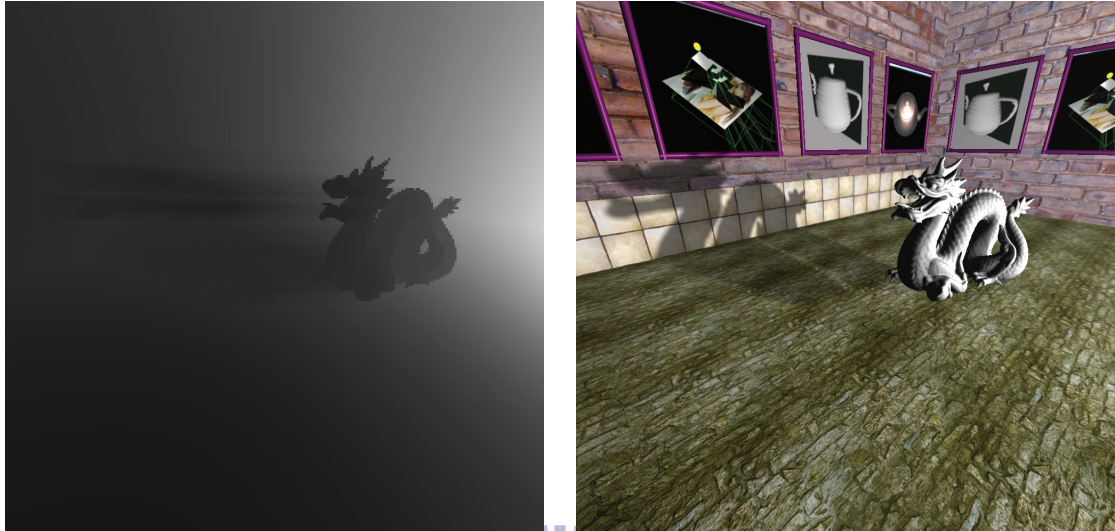
Figure 1.4 (Left) Only render participating media effects, (Right) a general scene

The "Render Scattering Effect" step is the core of our algorithm. First, we downsample the scene, because the way can reduce radiative computation. Second, we shot rays like ray marching. Third, we use the shadow volume data to determine the shadow region as shown in Figure 1.5. Fourth, especially for the sample in the shadow region, we use shadow map to determine whether the sample is in the shadow or not, as shown in Figure 1.6.
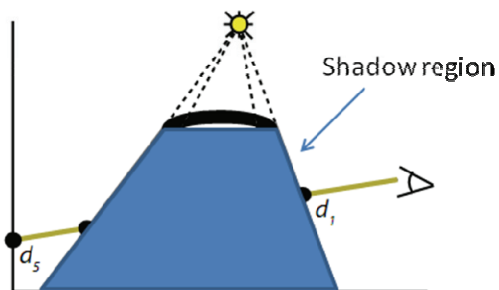
Figure 1.5 Shadow region
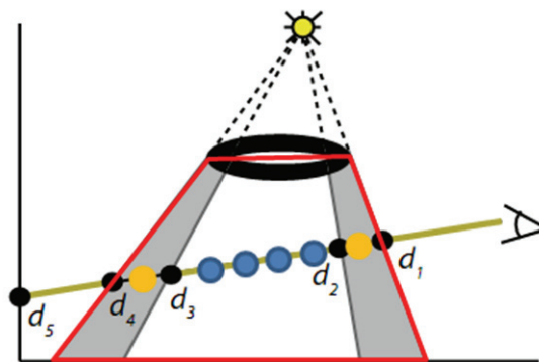
Figure 1.6 $d_1\sim d_2$, $d_3\sim d_4$: in shadows

$d_2\sim d_3$:not in shadows

So, a ray can be separated into two parts, in and out a shadow region respectively. Every ray computes radiance by simplified radiative transport equation, and combines our sampling method in the shadow region.
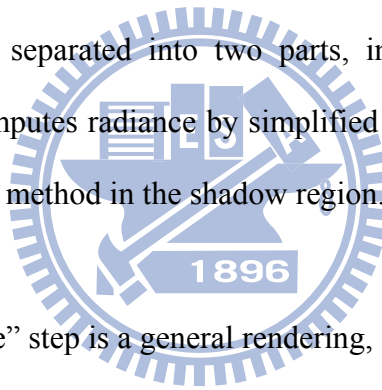
The "Render Scene" step is a general rendering, like phong lighting model. The "Bilateral filter" step combines the image of the "Render Scattering Effect" step and the image of the "Render Scene" step by using a bilateral filter.

The major contributions of this thesis are as follows: First, we apply a new sampling method to volumetric shadows to speed up the radiative computation. Second, we speed up the algorithm to real-time in most scenes. Third, our approach has a better result than other approaches for interactive volumetric shadows.

1.3 Thesis Organization

The rest parts of this thesis are structured in the following manner. Chapter 2 reviews the related work in volumetric shadows and scattering. Chapter 3 introduces background knowledges of the radiative transport equation. Chapter 4 describes the major concept of our real-time analytical volumetric shadows algorithm. Chapter 5 provides the implementation details. Chapter 6 shows the result of our algorithm and comparison with previous works. We also give some discussion in this chapter. Finally, we conclude our work and propose some future works in the Chapter 7.
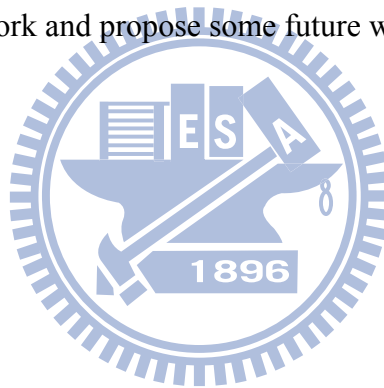
Chapter 2
Related Works

In this chapter, we survey previous works on volumetric shadows and scattering.

2.1 Volumetric Shadow

Shadow and fog play an important role in computer graphics. Volumetric shadow is an effect which combines shadow and fog. Crow [6] proposes a concept about the shadow polygon. Cowley [7] combines shadows and fog to produce the shafts of light through the trees, or the god-rays from the clouds. Dobashi et al. [8] propose a solution that sample the scene, using quads parallel to the image plane, to generate volumetric type effects as shown in Figure 2.1. These quads cut through a 3D representation of the participating media. However, using fewer quads increases aliasing as shown in Figure 1.2. Imagire et al. [9] solve the problem by averaging illumination over regions near each quad.

Figure 2.1 The result of Dobashi et al. [8]

Max [10] presented an approach which used the shadow volume geometry to determine the regions of lit and unlit atmosphere and used depth sorted shadow volumes to accumulate in scatter along each eye-pixel ray. Methods using shadow volumes [11, 12, 13] identify shadowed segments of viewing rays that do not contribute to in-scattering. But the methods do not guarantee correct results when the shadow geometry used overlaps, as shown in Figure 1.1. To solve the problem, it requires repeated depth peeling to sort shadow quads. However, the way adds sorting costs to the rendering time. Wyman et al. [2] propose an algorithm that combine Sun et al. [1] with ray-marching. This method can handle light-shafts and anisotropic light distributions.

2.2 Scattering

Scattering effects arise from participating media, such as smoke, haze, and fog. Several techniques have been proposed in computer graphics for rendering participating media. Cerezo et al. [14] provide a survey of the methods for scattering. Blinn [15] introduces an analytical model for homogeneous media. Max [10] proposes to evaluate the air-light integral by fitting Hermite cubic polynomials.

Nishita et al. [16] propose a method based on ray-marching. Dobashi et al. [8] propose a method based on volume-slicing. These methods are based on Riemann sums prompt to under-sampling artifacts. Willis [17] presents a simple model for homogeneous media considering a constant in-scattering term. Hoffman et al. [18] present a simple model for single atmospheric scattering of directional sun light in the constant and exponentially decreasing media respectively. Considering point light

sources in homogeneous media, Lecocq et al. [19] present an angular formulation of the radiative transport equation (RTE). The resulting RTE is expanded into a Taylor series. Our approach only expands the phase function into a Taylor series. Biri et al. [11] combine the angular formulation of the RTE and volumetric shadows as shown in Figure 2.2.

Figure 2.2 The result of Biri et al. [11]

Sun et al. [1] rederive Lecocq [19]'s RTE using a different notation and perform an additional linear change of variable to simplify it further. The air-light integral value stores a table via precomputation. Pegoraro et al. [20] propose a closed-form solution to the air-light integral in anisotropic media and a combined formulation of the air-light integral which allows both anisotropic phase functions and light distributions. The method can implement on shaders. Compare to Lecocq et al. [19] and Sun et al. [1], the quality of the results is better as shown in Figure 2.3.

| Lecocq et al. [19] | Sun et al. [1] | Pegoraro et al. [20] |

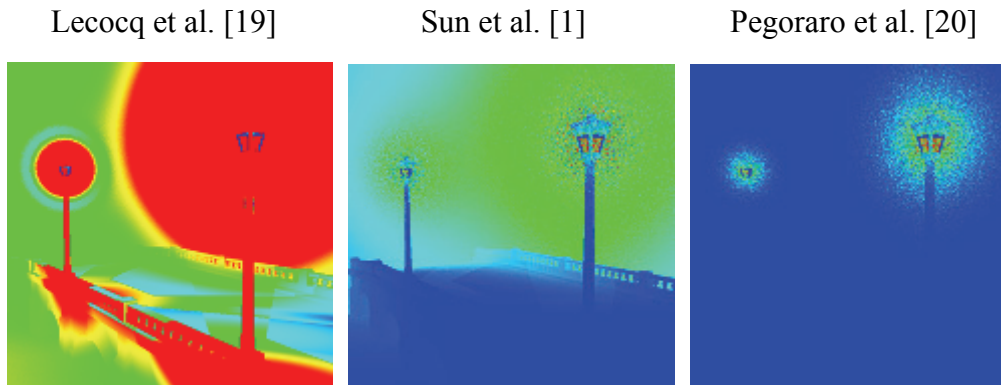Figure 2.3 Visualization of the absolute error mapped to hue (from blue to red)

Dobashi et al. [8] and Pegoraro et al. [20] show methods for achieving scattering effects at interactive rates. These methods are quite powerful but too slow to be used in games. The method which Hoffman et al. [18] proposed is appropriate for use in games. However, the accuracy of the result is lower than Dobashi et al. [8] and Pegoraro et al. [20].
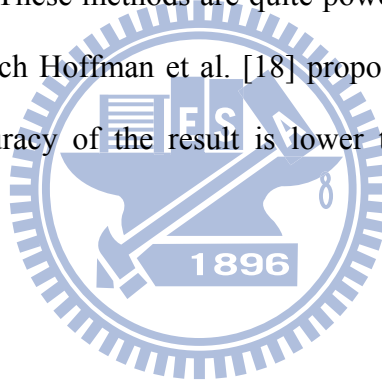
Chapter 3
Background Knowledge

In this chapter, we will introduce some background of the radiative transport equation which is related to our work. Radiative transfer is the physical phenomenon of energy transfer in the form of electromagnetic radiation. The propagation of radiation through a medium is affected by absorption, emission and scattering processes. The radiative transport equation describes these interactions mathematically.

A particle can interact with light hitting it in one of two ways: the particle can scatter the light or it can absorb it. Particles may also emit light on their own – we will ignore this phenomenon. Light interact with particles as shown in Figure 3.1
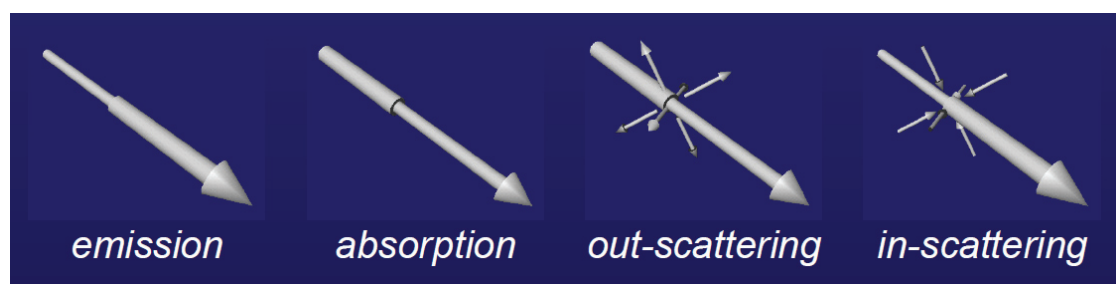
Figure 3.1 Absorption and out-scattering reduce radiance, emission and in-scattering add radiance

Absorption – the reduction in radiance due to the conversion of light to another form of energy, such as heat

Emission – energy that is added to the environment from luminous particles

Scattering – how light heading in one direction is scattered to other directions due to collisions with particles

3.1 Absorption

A particle's absorption of light can be quantified by its absorption cross section $\sigma_{ab}^{(\lambda)}$. We can see in Figure 3.2, the particle effectively behaves like a sphere.
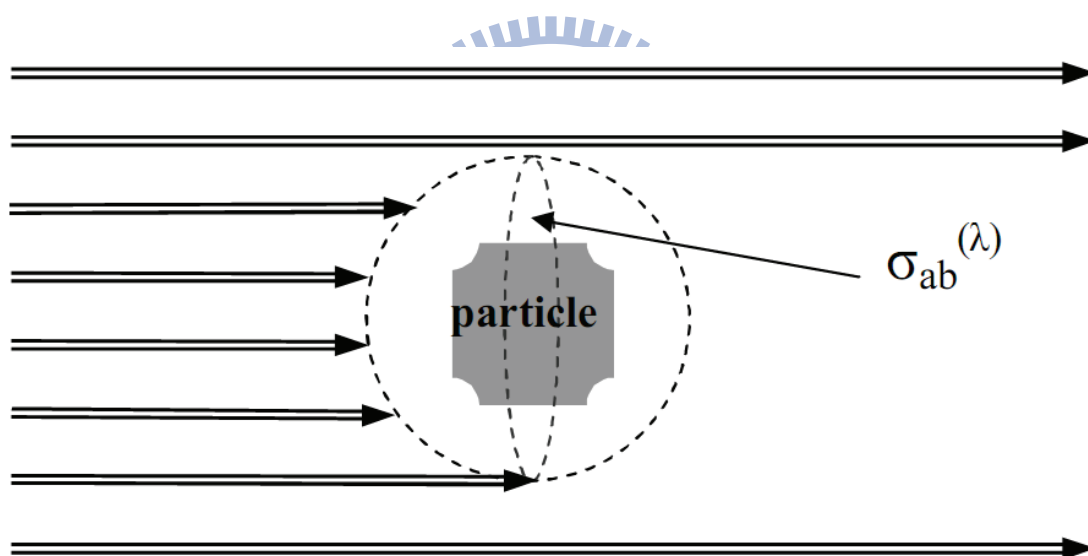
Figure 3.2 Absorption effect

Looking at the cross section of this sphere, we have an area which absorbs the incident irradiance at each point. The total absorbed flux is the incident irradiance integrated over the cross section area, $\sigma_{ab}^{(\lambda)}$. We define the absorption coefficient: $\beta_{ab}^{(\lambda)} = \rho_{ab}\sigma_{ab}^{(\lambda)}$, ρ_{ab} is a certain volume density of particles.

We define a box with an area "*A*" perpendicular to the light ray direction, and a small depth *dx* in the ray direction as shown in Figure 3.3.
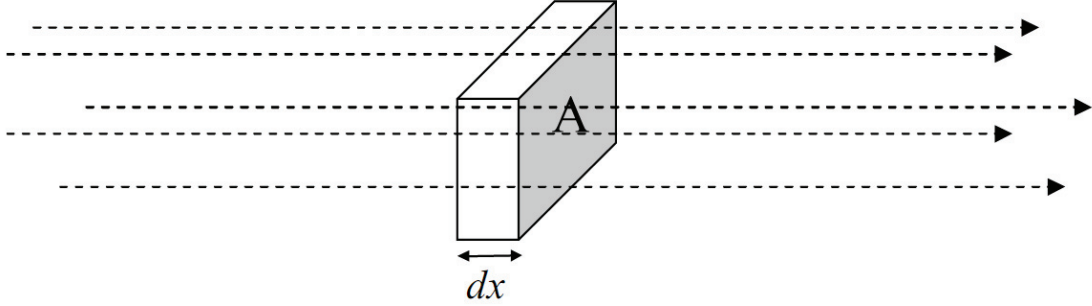
Figure 3.3 microcosmic light effect

The Volume of this box is *Adx,* so the aggregate absorption cross section area in the box is $\beta_{ab}^{(\lambda)}Adx$. We assume that $dL^{(\lambda)}$ is the amount of radiance which is absorbed, $L^{(\lambda)}$ is the radiance traveling a distance *dx* through the media. We can derive a proportional relation.

$$-\frac{dL^{(\lambda)}}{L^{(\lambda)}} = \frac{\beta_{ab}^{(\lambda)}Adx}{A} \tag{1}$$

The radiance $L^{(\lambda)}(x)$ can be found by solving the differential equation in Eq. 1.

$$L^{(\lambda)}(x) = L_0^{(\lambda)}e^{-\beta_{ab}^{(\lambda)}x} \tag{2}$$

$L_0^{(\lambda)}$ is the radiance at point 0.

Eq. 2 assumes that $\beta_{ab}^{(\lambda)}$ is constant. If it is spatially variant, we can derive the following.

$$L^{(\lambda)}(x) = L_0^{(\lambda)}e^{-\int_0^x \beta_{ab}^{(\lambda)}x'dx'} \tag{3}$$

3.2 Out-Scattering

Similar to absorption, a particle's out-scattering of light can be quantified by its scattering cross section $\sigma_{ab}^{(\lambda)}$. We can see in Figure 3.4, the particle scatter light out of a ray and reduce its radiance.
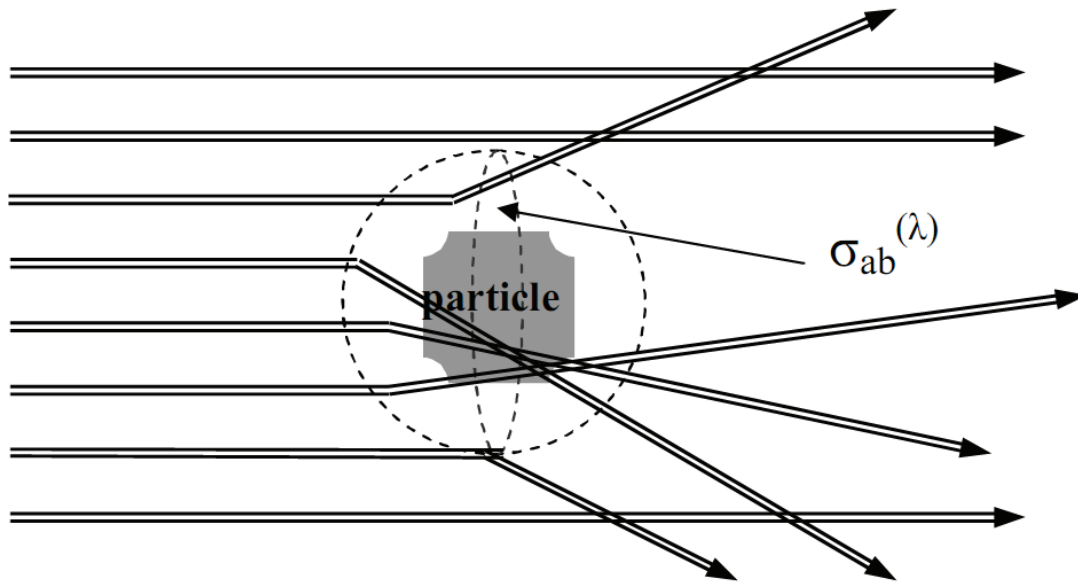
Figure 3.4 The out-scattering effect

As we did for absorption, we define the scattering coefficient: $\beta_{sc}^{(\lambda)} = \rho_{sc}\sigma_{sc}^{(\lambda)}$, ρ_{sc} is a certain volume density of particles. Like a situation for absorption, we can derive a proportional relation.

$$-\frac{dL^{(\lambda)}}{L^{(\lambda)}} = \frac{\beta_{sc}^{(\lambda)} A dx}{A}$$

(4)

If we assume that $\beta_{sc}^{(\lambda)}$ is constant, we can derive the following.

$$L^{(\lambda)}(x) = L_0^{(\lambda)} e^{-\beta_{sc}^{(\lambda)} x}$$

(5)

If $\beta_{sc}^{(\lambda)}$ is spatially variant, we can derive the following.

$$L^{(\lambda)}(x) = L_0^{(\lambda)} e^{-\int_0^x \beta_{sc}^{(\lambda)} x' dx'}$$

(6)

Because absorption and out-scattering decrease radiance, we can add the coefficients for absorption and out-scattering. The result is the extinction coefficient, $\beta_{ex}^{(\lambda)} = \beta_{ab}^{(\lambda)} + \beta_{sc}^{(\lambda)}$.

3.3 In-Scattering

There is one important thing, i.e. a scattered photon has to go somewhere. Some photons that are scattered from other place can add light to a ray. The phenomenon is in-scattering, where light which was originally headed in a different direction is scattered into the path of a light ray and adds to its radiance.

Just as there are a wide variety of BSDF models to describe scattering from surfaces, the scattering phase function $\Phi(\theta)$ gives the probability of scattered light going in the direction (θ) to the viewing direction as shown in Figure 3.5.
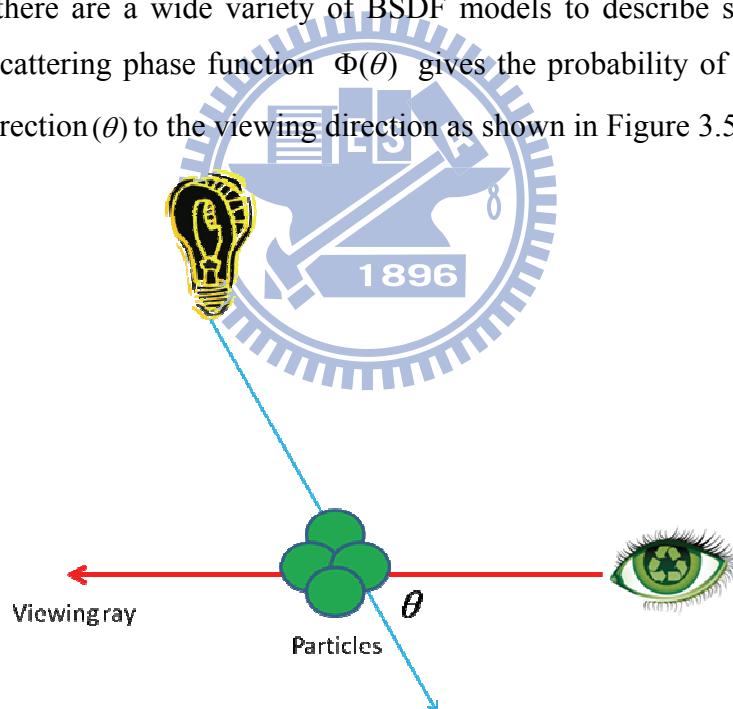

Figure 3.5 The light ray shots to a particle and reflects to eyes

Since this is a probability function, integrated over the entire sphere of directions is 1.

$$\int_{4\pi} \Phi(\theta)d\omega = 1 \tag{7}$$

Looking at a single inscattering event as shown in Figure 3.6.
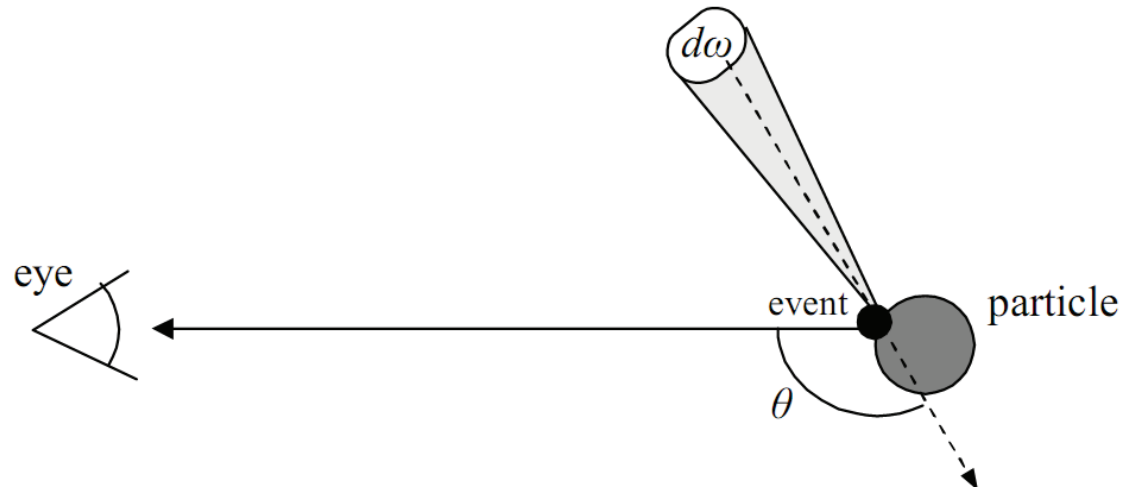
Figure 3.6 A single inscattering event

The solid angle patch of incoming directions $d\omega$ is infinitesimal. $\Phi(\theta)$ is a phase function and $L_i^{(\lambda)}$ is the incoming radiance from the patch. We get $L_i^{(\lambda)}(\theta)\Phi(\theta)d\omega$ which is the inscattered radiance from the patch. We can derive the total radiance added by this inscattering event.

$$\int_{4\pi} L_i^{(\lambda)}(\theta)\Phi(\theta)d\omega \tag{8}$$

Considering the situation of Figure 3.3, the result is that the radiance added due to inscattering over an infinitesimal path dx is:

$$dL_{inscatter}^{(\lambda)} = \beta_{sc}^{(\lambda)}dx\int_{4\pi} L_i^{(\lambda)}(\theta)\Phi(\theta)d\omega \tag{9}$$

$\beta_{sc}^{(\lambda)}dx$ is the volume that catches the inscattering radiance, and $dL_{inscatter}^{(\lambda)}$ is the amount of radiance which is added by inscattering.

We combine the Eq. 1, Eq. 4 and Eq. 9. Considering absorption, out-scattering, and in-scattering, we derive the following.

$$\frac{dL^{(\lambda)}}{dx} = -\beta_{ex}^{(\lambda)}L^{(\lambda)} + \beta_{sc}^{(\lambda)}\int_{4\pi}L_i^{(\lambda)}(\theta)\Phi(\theta)d\omega \tag{10}$$

The solution to which is:

$$L^{(\lambda)}(x) = L_0^{(\lambda)}e^{-\int_0^x \beta_{ex}^{(\lambda)}x'dx'} +$$

$$\int_0^x \left[e^{-\int_{x'}^x \beta_{ex}^{(\lambda)}x''dx''} \beta_{sc}^{(\lambda)}\int_{4\pi}L_i^{(\lambda)}(\theta)\Phi(\theta)d\omega \right]dx' \tag{11}$$

X is the distance along the ray at which we are evaluating the radiance. The formulation has two parts – one multiplicative (extinction) and one additive (in-scattering).

$$L^{(\lambda)}(x) = L_0^{(\lambda)}Part_{extinction} + Part_{inscattering} \tag{12}$$
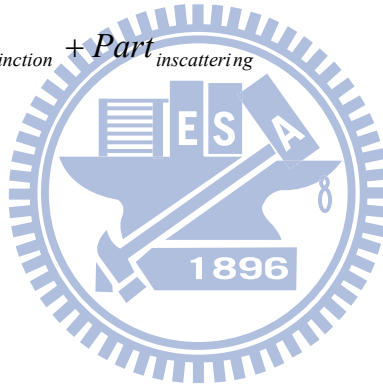
Chapter 4
Algorithm

We develop a novel real-time analytical volumetric shadows algorithm based on Wyman et al. [2] and Pegoraro et al. [3] [4]. A new sampling method is applied to the shadow region. We use a simplified analytical radiative transport equation to speed up the algorithm. We first create the frontmost shadow polygon and the final shadow polygon from the eye view and shadow map from the light view. Next, the scene is separated into two phases: "Render Scene" (No participating media, a general scene) and "Render Scattering Effect" (Only render participating media effects), as shown in Figure 1.3. In this way, we can avoid expensive radiative computation in the original scene.
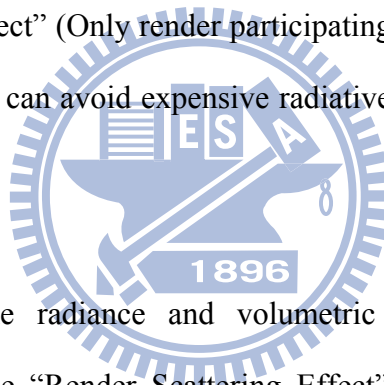
We can compute the radiance and volumetric shadowing effects in the downsampling scene of the "Render Scattering Effect" phase. When we compute volumetric shadows, we use ray-marching in the shadow region. Then we use the shadow map to identify illuminated samples, as shown in Figure 1.6. Finally, we use the simplified analytical radiative transport equation and the proposed sampling method for each light sample (such as the samples between d_2 and d_3 in Figure 1.6). The "Render Scene" phase is a general rendering process, which includes soft shadows.

As a result, we upsample the image in the "Render Scattering Effect" phase and combine it with the resulting image of the "Render Scene" phase by using a bilateral filter.

The following sections are structured as follows. In Section 4.1, we describe types of samples in a scene and the details of our sampling method. In Section 4.2, we derive the radiative transport equation. In Section 4.3, we derive the analytical formulation of radiative transport equation. Finally in Section 4.4 we describe the detail of reformulation of analytical radiative transport equation.

4.1 Shadow Region

A sample has three types, type 1 : not in a shadow region, type 2: in a shadow region and in the dark region of a shadow region, type 3 : in a shadow region and in the illuminated region of a shadow region, as shown in Figure 4.1.
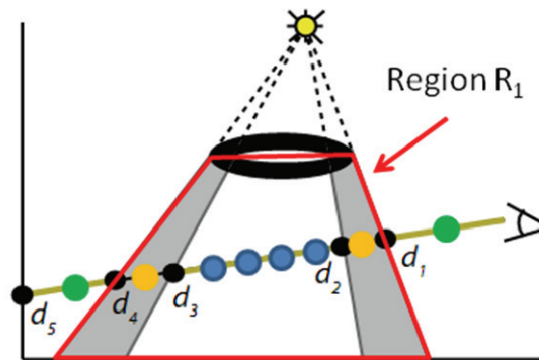

Figure 4.1 Region R_1 : Shadow Region,

samples between d_1 and d_2 : in the dark region of a shadow region,

samples between d_2 and d_3: in the illuminated region of a shadow region,

samples between d_3 and d_4 : in the dark region of a shadow region.

Considering a ray from an eye-pixel, we first use the shadow volume data to determine whether the ray passes through shadow regions or not. If not, we just compute radiance from the surface point to the eye as shown in Figure 4.2.
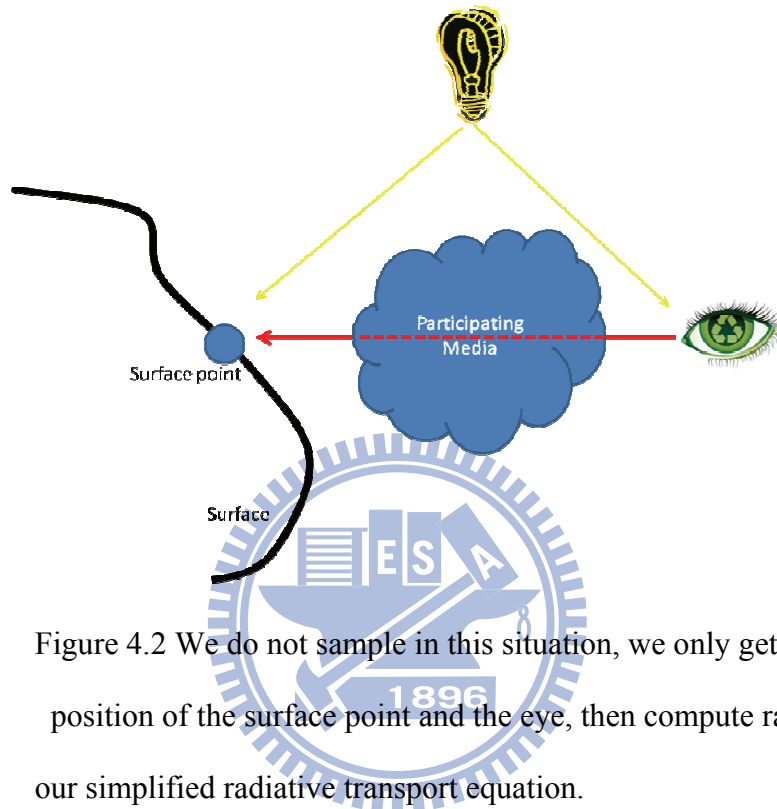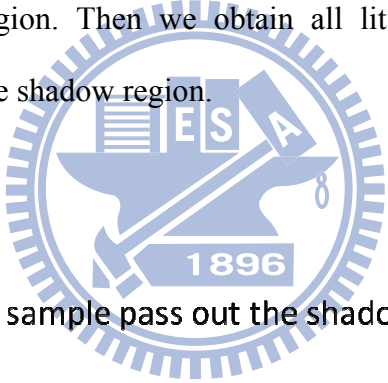
Figure 4.2 We do not sample in this situation, we only get the position of the surface point and the eye, then compute radiance by our simplified radiative transport equation.

If a ray passes through the shadow region, the radiance is the summation of three parts : the radiance from the eye to the frontmost shadow polygon, the radiance in a shadow region, and the radiance from the final shadow polygon to a surface point, as shown in Figure 4.4. In this thesis, we want to compute the radiance in a lit region for a ray. We get two intersection points of a lit region and a ray. Then we compute radiance by our proposed simplified radiative transport equation as shown in Figure 4.5. As a result, the final radiance is the summation of all orange segments of a ray.

In order to determine the lit regions in a shadow region, we use ray marching. Then we apply shadow map to identify lit samples(between d_2 and d_3) as shown in Figure 1.6. Finally, we use our sampling method to determine the lit regions by these samples.

Now we describe our proposed sampling method. There are four sampling situations when a ray passes through the shadow region, as shown in Figure 4.3. We define "startPos" as the first sample of a lit region, "endPos" as the final sample of a lit region, "litType_before" as the type of the previous sample, and "litType_current" as the type of the current sample. We do the following steps repeatedly until the ray passes out the shadow region. Then we obtain all lit regions ([startPos,endPos] represents a lit region) in the shadow region.

Algorithm 4.1

```
While (current sample pass out the shadow region)
{
■ litType_before=litType_current
■ Current sample = next sample
■ litType_current = the type of Current sample
■ If (litType_before is lit and litType_current is dark)
    □ endPos = Current sample
    □ Store [startPos,endPos]
■ If (litType_before is dark and litType_current is lit)
    □ startPos = current sample
}
```
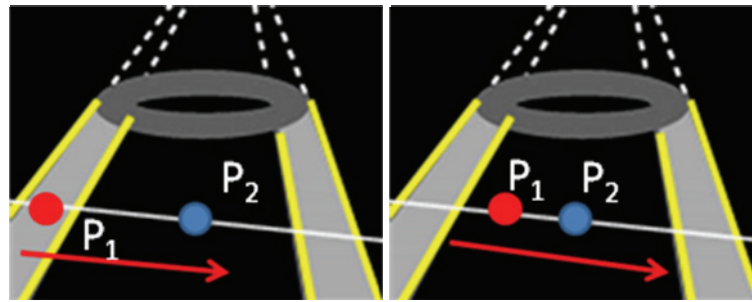
(a) dark to lit   (b) lit to lit
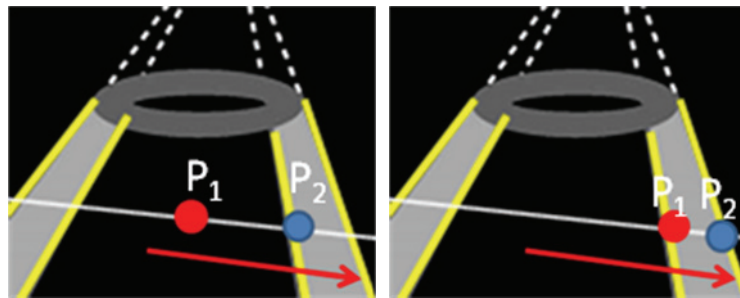
(c) lit to dark   (d) dark to dark

Figure 4.3 P_1 point : previous sample, P_2 point : Current sample
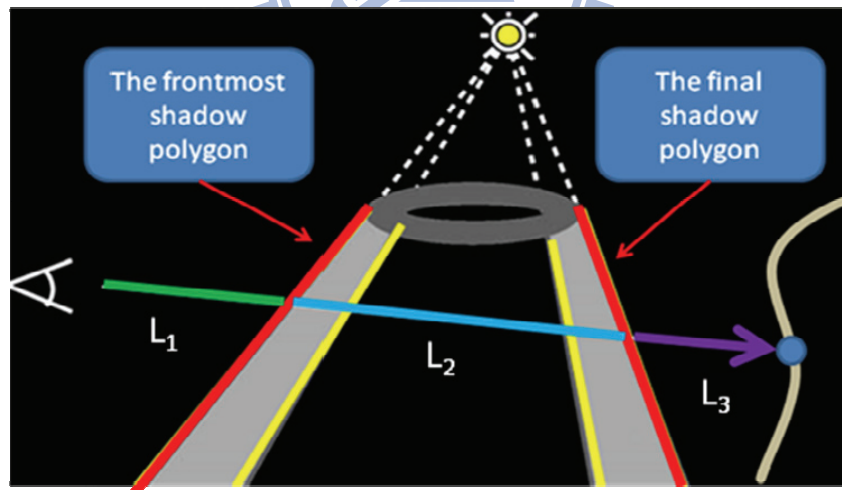
Figure 4.4 L_1 line: the part of the eye to the frontmost shadow polygon,

L_2 line: the part in a shadow region,

L_3 line: the part of the final shadow polygon to a surface point
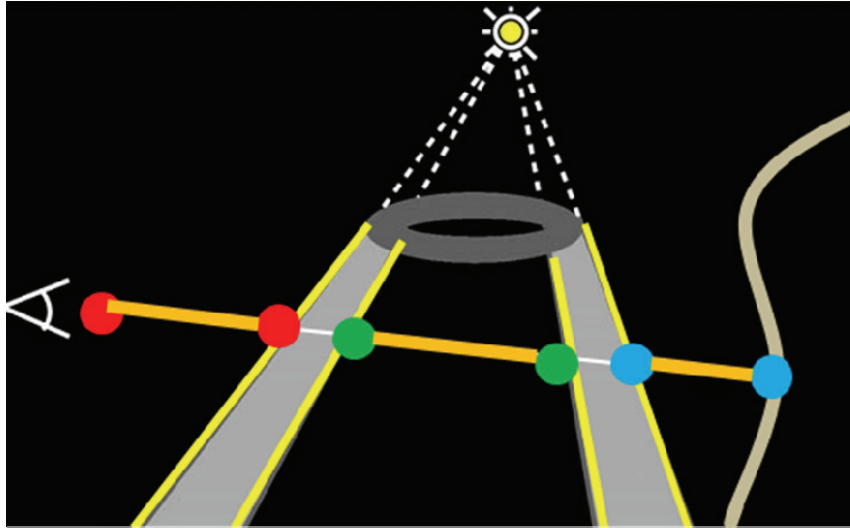
Figure 4.5 The orange segment represents lit region of a ray, these colorful

points are intersection points of lit regions and a ray

Participating Media
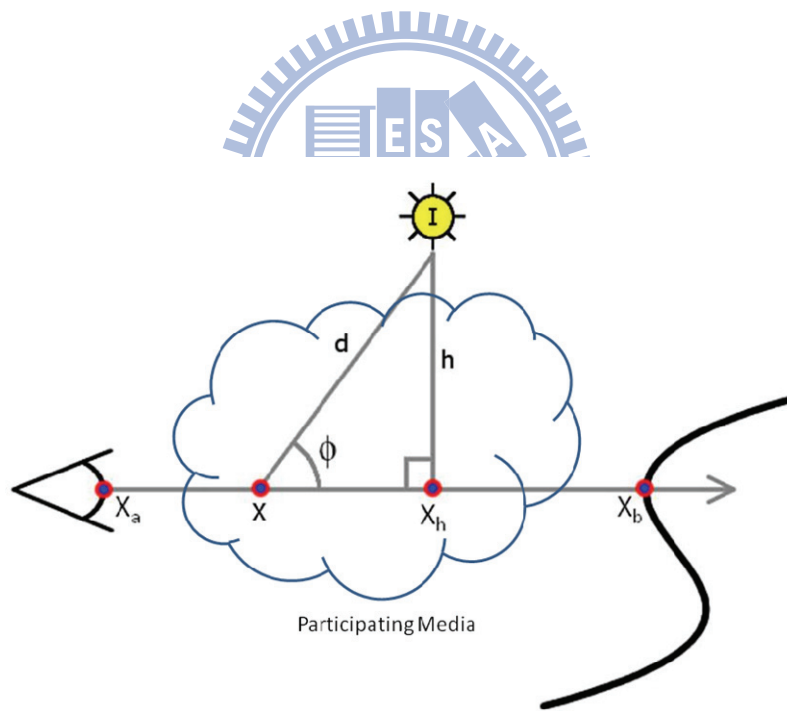
Figure 4.6 We shot a ray from the eye X_a to the surface point X_b, h is the distance

from the light to the given ray and X_h is the coordinate of its projection onto it

We use the shadow volume data to determine whether a ray passes through shadow regions or not. In implementation, we define the distances: "front distance" : the distance between the eye and the P_1 point, as shown in Figure 4.7, distance "final distance" : the distance between the eye and the P_2 point and distance "surface distance" : the distance between the eye and the P_3 point. Then, we use the Algorithm 4.2 to determine the situation of a ray.

Algorithm 4.2

- **If (front distance <surface distance)**
  - The ray pass through the shadow region
- **else**
  - The ray does not pass through

- **Assume a certain sample, "A" on a ray**
- If (front distance <the distance between the eye and "A" <final distance)
  - The sample, "A" is in the shadow region
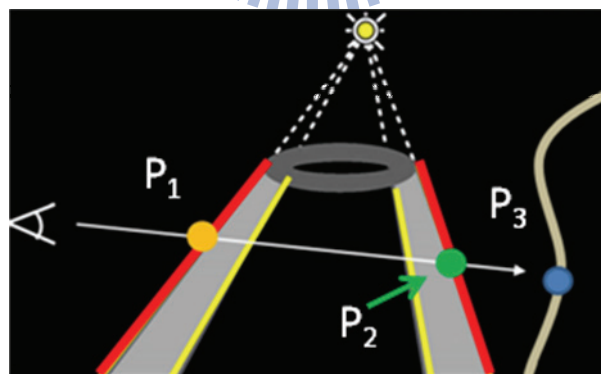
Figure 4.7 P_1: the intersection point of the frontmost shadow polygon

P_2: the intersection point of the final shadow polygon

P_3: the intersection point of the surface

4.2 Radiative Transport Equation

In this section, we derive the formulation of radiative transport equation.

Consider Eq. 11. We change symbols to fit the Figure 4.6.

$$L(x_a, x_b) = L_b e^{-\int_{x_a}^{x_b} \beta_{ex}(x)dx} +$$

$$\int_{x_a}^{x_b} \left[e^{-\int_{x_a}^{x} \beta_{ex}(x')dx'} \beta_{sc}(x) \int_{4\pi} L_i(\phi, x)\Phi(\phi)d\omega \right] dx$$

$$\phi = 180 - \theta \tag{13}$$

where L_b is the original light intensity.

Because we focus on single-scattering, we only consider light radiance of the light

source in a certain angle.

$$L(x_a, x_b) = L_b e^{-\tau(x_a, x_b)} + \int_{x_a}^{x_b} \left[e^{-\tau(x_a, x)} \kappa_s(x)L_i(\phi, x)\Phi(\phi) \right] dx$$

$$\kappa_s(x) = \beta_{sc}(x), \kappa_t(x) = \beta_{ex}(x), \tau(x_a, x_b) = \int_{x_a}^{x_b} \kappa_t(x)dx \tag{14}$$

We derive $L_i(\phi, x)$, the incoming radiance from the light to the point X as shown in

Figure 4.6.

$$L_i(\phi, x) = e^{-\kappa_t DISTANCE(x, light)} \frac{I}{DISTANCE(x, light)^2}$$

$$= e^{-\kappa_t \sqrt{h^2 + (x-x_h)^2}} \frac{I}{h^2 + (x - x_h)^2} \tag{15}$$

where h is the distance from the light to the given ray and X_h is the coordinate of its projection onto it. $e^{-\kappa_t dis\,tan\,ce(x,light)}$ is the ratio of light radiance that is decreased by absorption and out-scattering.

Defining L_m as the medium radiance, Eq. 14 becomes

$$L(x_a, x_b) = L_b e^{-\kappa_t(x_b - x_a)} +$$

$$\boxed{I\kappa_s e^{\kappa_t x_a} \int_{x_a}^{x_b} \frac{e^{-\kappa_t(x+\sqrt{h^2+(x-x_h)^2})}}{h^2+(x-x_h)^2} \Phi(\arctan(\frac{x-x_h}{h})+\frac{\pi}{2})dx} \tag{16}$$

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad}_{L_m(X_a, X_b)}$$

This is the final formulation of a radiative transport equation.

4.3 Analytical Approach to Scattering

We rewrite variables $u = \frac{x-x_h}{h}$ and $H = \kappa_t h$, L_m of Eq. 16 becomes

$$L_m(x_a, x_b) =$$

$$I\frac{\kappa_s}{h}e^{\kappa_t(x_a - x_h)} \int_{\frac{x_a-x_h}{h}}^{\frac{x_b-x_h}{h}} \left[\frac{e^{-H(u+\sqrt{1+u^2})}}{1+u^2} \Phi_c(-\frac{u}{\sqrt{1+u^2}}) \right] du$$

$$assume \quad \Phi(\phi) = \Phi_c(\cos(\phi)) \tag{17}$$

We rewrite variables $v = u + \sqrt{1+u^2}$ and $w = u - \sqrt{1+u^2}$ based on Pegoraro et al. [20]. Then we get two reformulations of Eq. 17 and a combined formulation. According to the relation of X_a, X_b and a light position as shown in Figure 4.6, we choose the fit formulation to compute radiance.

4.3.1 First Analytical Formulation

Let $v = u + \sqrt{1+u^2}$ where $v \in (0,1,\infty)$, then Eq. 17 becomes

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} 2 \int_{v_a}^{v_b} \left[\frac{e^{-Hv}}{1+v^2} \Phi_c(\frac{1-v^2}{1+v^2}) \right] dv$$

$$v_a = v(x_a), v_b = v(x_b)$$

(18)

Assume the phase function is isotropic, $\Phi_c(\frac{1-v^2}{1+v^2}) = \frac{1}{4\pi}$. Then Eq. 18 becomes

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} \frac{2}{4\pi} \int_{v_a}^{v_b} \left[\frac{e^{-Hv}}{1+v^2} \right] dv$$

(19)

Based on Pegoraro et al. [20], we derive the following,

$$\int \frac{e^{av}}{1+v^2} dv = \sin(a)\Re(Ei(av + ta)) - \cos(a)\Im(Ei(av + ta))$$

$$= i_0(a, v)$$

(20)

where $t^2 = -1$, Ei is the exponential integral function, where $\Re$ and $\Im$ is the real and imaginary part respectively.

Using the Eq. 20, Eq. 19 can be rewritten as follows

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} \frac{2}{4\pi} I_0(-H, v_a, v_b)$$

$$where \quad I_0(-H, v_a, v_b) = i_0(-H, v_b) - i_0(-H, v_a)$$

(21)

27

If the phase function is anisotropic, we use an expansion into a Taylor series.

Then Eq. 18 becomes

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} 2 \sum_{n=0}^{N-1} c_n \int_{v_a}^{v_b} \left[\frac{e^{-Hv}}{1+v^2} v^n \right] dv$$

$$\Phi_c \left(\frac{1-v^2}{1+v^2} \right) = \sum_{n=0}^{N-1} c_n v^n$$

(22)

Based on Pegoraro et al. [20] and using Eq. 20, we derive the following,

$$\int \frac{e^{av}}{1+v^2} v \, dv = \cos(a)\Re(Ei(av+ta)) + \sin(a)\Im(Ei(av+ta))$$

$$= i_1(a, v)$$

$$\int \frac{e^{av}}{1+v^2} v^n \, dv = e^{av} \sum_{j=0}^{n-2} v^j c(a, j, n) + (-1)^{\left\lfloor \frac{n\%4}{2} \right\rfloor} i_{(n\%2)}(a, v)$$

(23)

$$where \quad c(a, j, n) = \sum_{\substack{i=(n-2-j)\%2 \\ i+=2}}^{n-2-j} (-1)^{\frac{n-i-j}{2}} \frac{(i+j)!}{j!} \left(-\frac{1}{a} \right)^{i+1}$$

Using the Eq. 23, Eq. 22 can be rewritten as follows

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} 2 \sum_{n=0}^{N-1} c_n$$

$$\left(\sum_{j=0}^{n-2} (e^{-Hv_b} v_b^j - e^{-Hv_a} v_a^j) c(-H, j, n) + \atop (-1)^{\left\lfloor \frac{n\%4}{2} \right\rfloor} I_{(n\%2)}(-H, v_a, v_b) \right)$$

(24)

$$where \quad I_1(a, v_a, v_b) = i_1(a, v_b) - i_1(a, v_a)$$

4.3.2 Second Analytical Formulation

Let $w = u - \sqrt{1+u^2}$ where $w \in (-\infty, -1, 0)$, Eq. 17 becomes

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} 2 \int_{w_a}^{w_b} \left[\frac{e^{\frac{H}{w}}}{1+w^2} \Phi_c(\frac{w^2-1}{w^2+1}) \right] dw$$

(25)

$$w_a = w(x_a), w_b = w(x_b)$$

Similar to Eq. 20 and Eq. 23, we can derive the following,

$$j_0(a, w) = -\sin(a)\Re(Ei(\frac{a}{w} + ta)) + \cos(a)\Im(Ei(\frac{a}{w} + ta))$$

$$j_1(a, w) = \cos(a)\Re(Ei(\frac{a}{w} + ta)) + \sin(a)\Im(Ei(\frac{a}{w} + ta)) - Ei(\frac{a}{w})$$

(26)

We assume that the phase function is isotropic, $\Phi_c(\frac{w^2-1}{w^2+1}) = \frac{1}{4\pi}$. By using Eq. 26,

Eq.25 becomes

$$L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t(x_a - x_h)} \frac{2}{4\pi} J_0(-H, w_a, w_b)$$

$$where \quad J_0(-H, w_a, w_b) = j_0(-H, w_b) - j_0(-H, w_a)$$

(27)

If the phase function is anisotropic, we use an expansion into a Taylor series. Similar

to Eq. 23, we can derive the following,

$$\int \frac{e^{\frac{a}{w}}}{1+w^2} w^n dw = -e^{\frac{a}{w}} \sum_{j=0}^{n-2} w^{j+1} d(a, j, n) + (-1)^{\lfloor \frac{n}{2} \rfloor} j_{(n\%2)}(a, w)$$

$$+ Ei(\frac{a}{w}) a d(a, 0, n)$$

(28)

$$where \quad d(a, j, n) = \sum_{\substack{i=(n-2-j)\%2 \\ i+=2}}^{n-2-j} (-1)^{\frac{n-i-j}{2}} \frac{j!}{(i+j+1)!} (a)^i$$

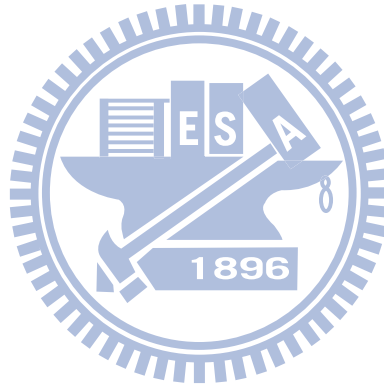Using the Eq. 28, Eq. 25 can be rewritten as follows

$$
L_m(x_a, x_b) = I \frac{\kappa_s}{h} e^{\kappa_t (x_a - x_h)} 2 \sum_{n=0}^{N-1} d_n
$$

$$
\left(
\begin{array}{l}
-\sum_{j=0}^{n-2} (e^{\frac{H}{v_b}} w_b^{j+1} - e^{\frac{H}{v_b}} w_a^{j+1}) d(H, j, n) + \\[2mm]
(-1)^{\left\lfloor \frac{n}{2} \right\rfloor} J_{(n\%2)}(H, w_a, w_b) + J_e(H, w_a, w_b) H d(H, 0, n)
\end{array}
\right)
$$

$$
where \quad J_1(a, w_a, w_b) = j_1(a, w_b) - j_1(a, w_a),
$$

$$
J_e(H, w_a, w_b) = Ei(\frac{H}{w_b}) - Ei(\frac{H}{w_a})
$$

$$
d_n : Taylor \quad coefficient
$$

(29)

4.3.3 Combined Formulation

We split the L_m of Eq.16,

$$
L_m(x_a, x_b) = L_m(x_a, x_h) + L_m(x_h, x_b)
$$
$$
= \underbrace{L_m(x_h, x_b)}_{\text{First Part}} - \underbrace{L_m(x_h, x_a)}_{\text{Second Part}}
$$

(30)

The first part uses the second formulation to compute radiance, the second part uses the first formulation.

According to the coordinate of light's projection onto a ray, we determine which formulation we use, as shown in Figure 4.8.
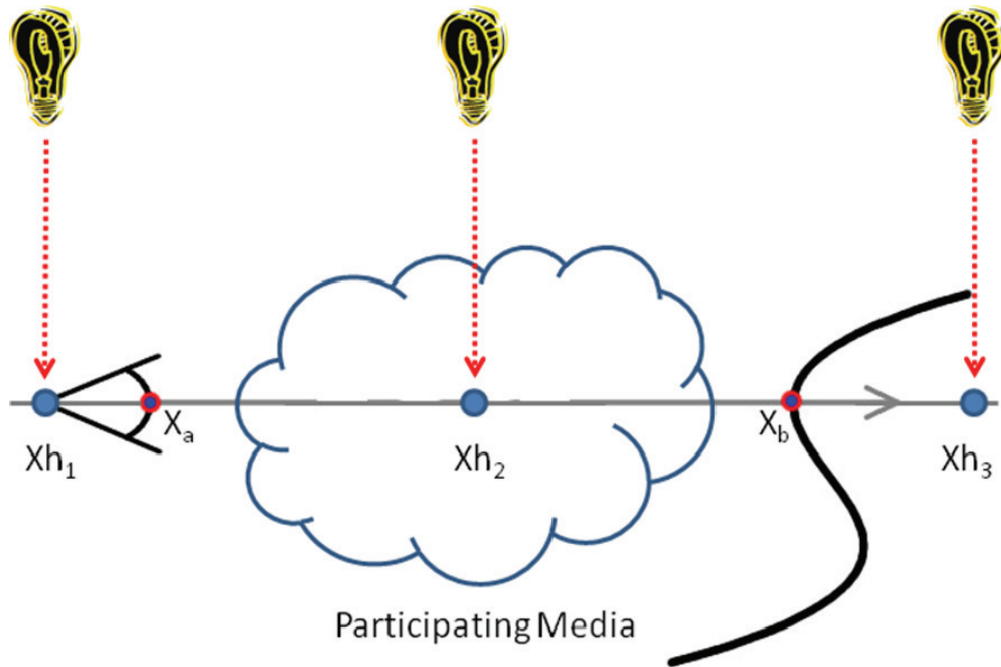

Figure 4.8 Xh_1 : a coordinate is smaller than X_a and X_b

Xh_2 : a coordinate between X_a and X_b,

Xh_3 : a coordinate is bigger than X_a and X_b

If it is Xh_1, we use the second formulation (Eq. 27 or Eq. 29), or it is Xh_3, we use the first formulation (Eq. 21 or Eq. 24), or it is Xh_2, we use the combined formulation

4.4 Reformulation

In this section, we focus on the combined formulation. The combined formulation requires four evaluations of the complex-valued exponential integral Ei with the parameters V_a, V_h, W_b, and W_h as shown in Figure 4.9.

$$v_h = 1, w_h = -1$$

$$I_0(-H, v_h, v_a) = i_0(-H, v_a) - i_0(-H, 1)$$
$$I_1(-H, v_h, v_a) = i_1(-H, v_a) - i_1(-H, 1)$$
$$J_0(H, w_h, w_b) = j_0(H, w_b) - j_0(H, -1)$$
$$J_1(H, w_h, w_b) = j_1(H, w_b) - j_1(H, -1)$$

Figure 4.9 Each square represents an evaluation of the complex-valued exponential integral Ei.

In Figure 4.9, the detail of each square show in Figure 4.10.

$$i_0(a, v) = \sin(a)\Re(Ei(av + ta)) - \cos(a)\Im(Ei(av + ta))$$
$$i_1(a, v) = \cos(a)\Re(Ei(av + ta)) + \sin(a)\Im(Ei(av + ta))$$
$$j_0(a, w) = -\sin(a)\Re(Ei(\frac{a}{w} + ta)) + \cos(a)\Im(Ei(\frac{a}{w} + ta))$$
$$j_1(a, w) = \cos(a)\Re(Ei(\frac{a}{w} + ta)) + \sin(a)\Im(Ei(\frac{a}{w} + ta)) - Ei(\frac{a}{w})$$

Figure 4.10 A blue square represents the same evaluation of i_0 and i_1

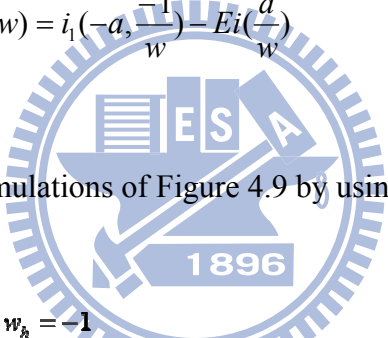An orange square represents the same evaluation of j_0 and j_1

We want to find a relation to reduce the evaluations of the complex-valued exponential integral Ei. Consider Figure 4.10. We derive a relation as shown in Eq. 31.

$$j_0(a,w) = -i_0(a,\frac{1}{w})$$
$$j_1(a,w) = i_1(a,\frac{1}{w}) - Ei(\frac{a}{w})$$

(31)

Using the relation, $Ei(\bar{z}) = \overline{Ei(z)}$, then Eq. 31 becomes,

$$j_0(a,w) = i_0(-a,\frac{-1}{w})$$
$$j_1(a,w) = i_1(-a,\frac{-1}{w}) - Ei(\frac{a}{w})$$

(32)

We can reformulate the formulations of Figure 4.9 by using Eq. 32.

$$v_h = 1, w_h = -1$$
$$I_0(-H,v_h,v_a) = i_0(-H,v_a) - i_0(-H,1)$$
$$I_1(-H,v_h,v_a) = i_1(-H,v_a) - i_1(-H,1)$$
$$J_0(H,w_h,w_b) = j_0(H,w_b) - i_0(-H,1)$$
$$J_1(H,w_h,w_b) = j_1(H,w_b) - i_1(-H,1) - Ei(-H)$$

Figure 4.11 There are only three evaluations of the complex-valued exponential integral Ei

We propose a method to reduce one evaluation of the complex-valued exponential integral Ei, as shown in Figure 4.11. This reduction can speed up our proposed algorithm.

Chapter 5
Implementation and Results

Our algorithm works on the screen space. We shot a ray for each pixel, and then compute the radiance by using our proposed algorithm. Each pixel gets the radiance by using our algorithm. In this thesis, the complex-valued exponential integral *Ei*, as shown in Figure 4.9, is an expensive computation. Based on Pegoraro et al. [20], we can estimate the value of the complex-valued exponential integral *Ei*. We use texture-array to store shadow polygons and texture buffer to store shadow maps, then transfer them to GPU. Our algorithm uses them to check light regions of shadow regions in GPU.
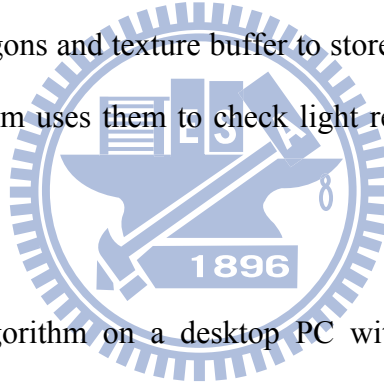
We implement our algorithm on a desktop PC with Intel Core 2 Quad CPU 2.66GHz, 4G RAM, and NVIDIA Geforce 9800GT video card. All results are rendered at 1024 x 1024 pixels. Figures 5.1 and 5.2 show four scenes rendered by our algorithm. Figure 5.1(a) is the result in the "Render Scattering Effect" phase. Figure 5.1(b) is the result in the "Render Scene" phase. Figure 5.1(c) is the final result. Figure 5.1(d)(e)(f) and Figure 5.2 are similar. Figure 5.3 shows the result about the anisotropic phase function, we use Rayleigh phase function, $\Phi(\theta) = \frac{3}{4}(1 + \cos^2 \theta)$, and compare with the isotropic phase function.
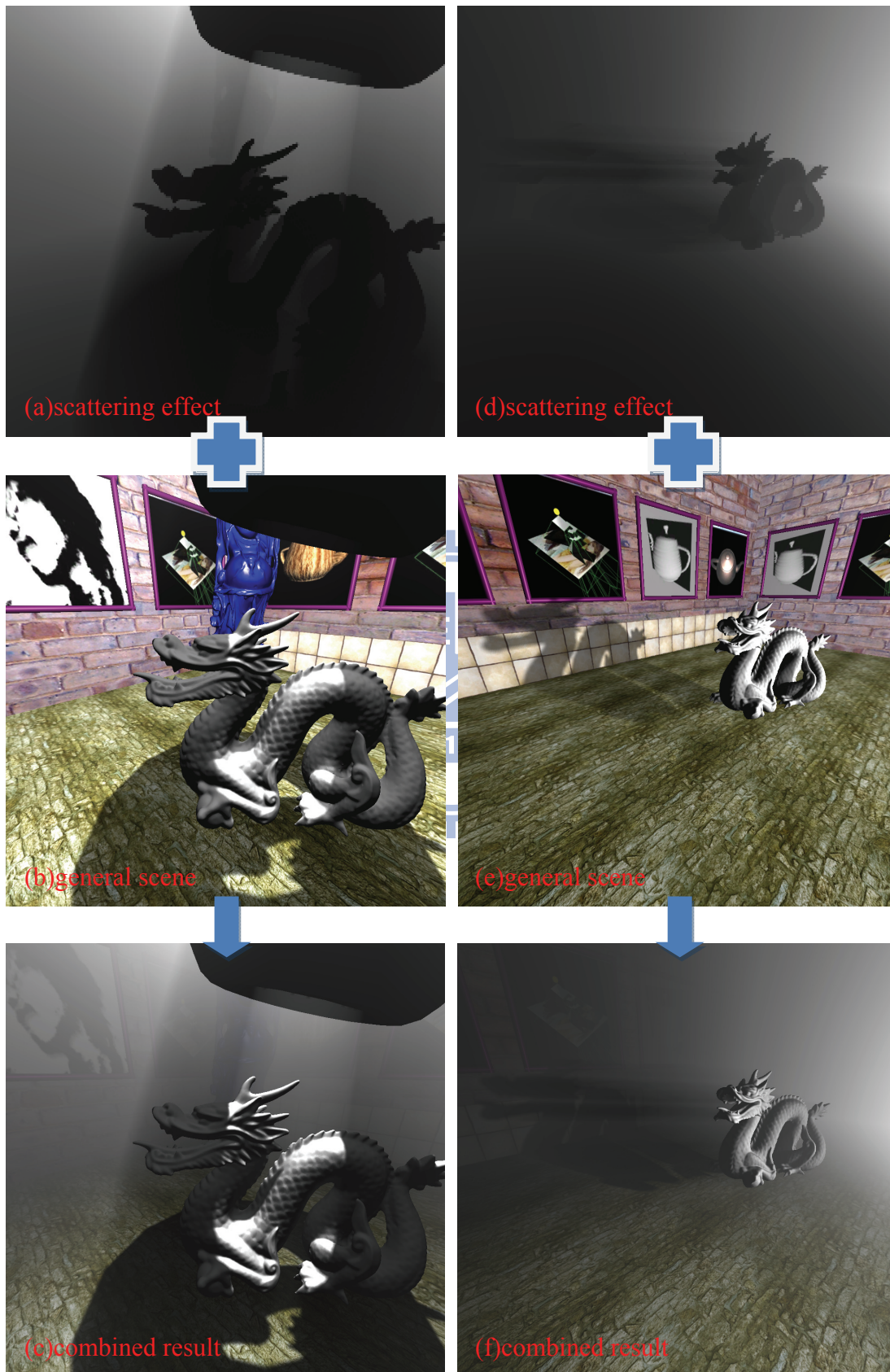
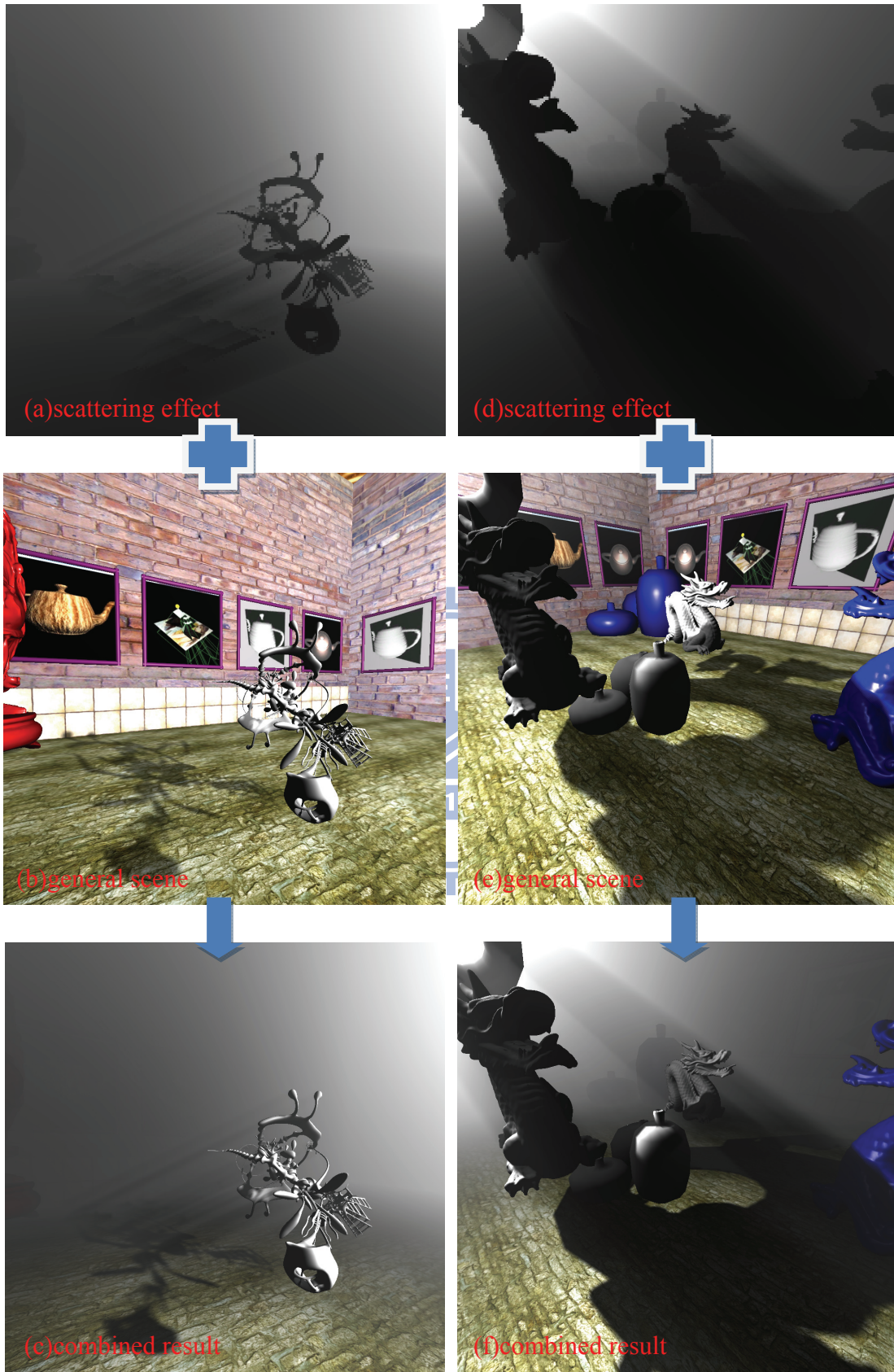Figure 5.1 Final Results (Left) Donut & dragon (Right) Dragon

Figure 5.2 Final results (Left) YeahRight (Right) Many objects
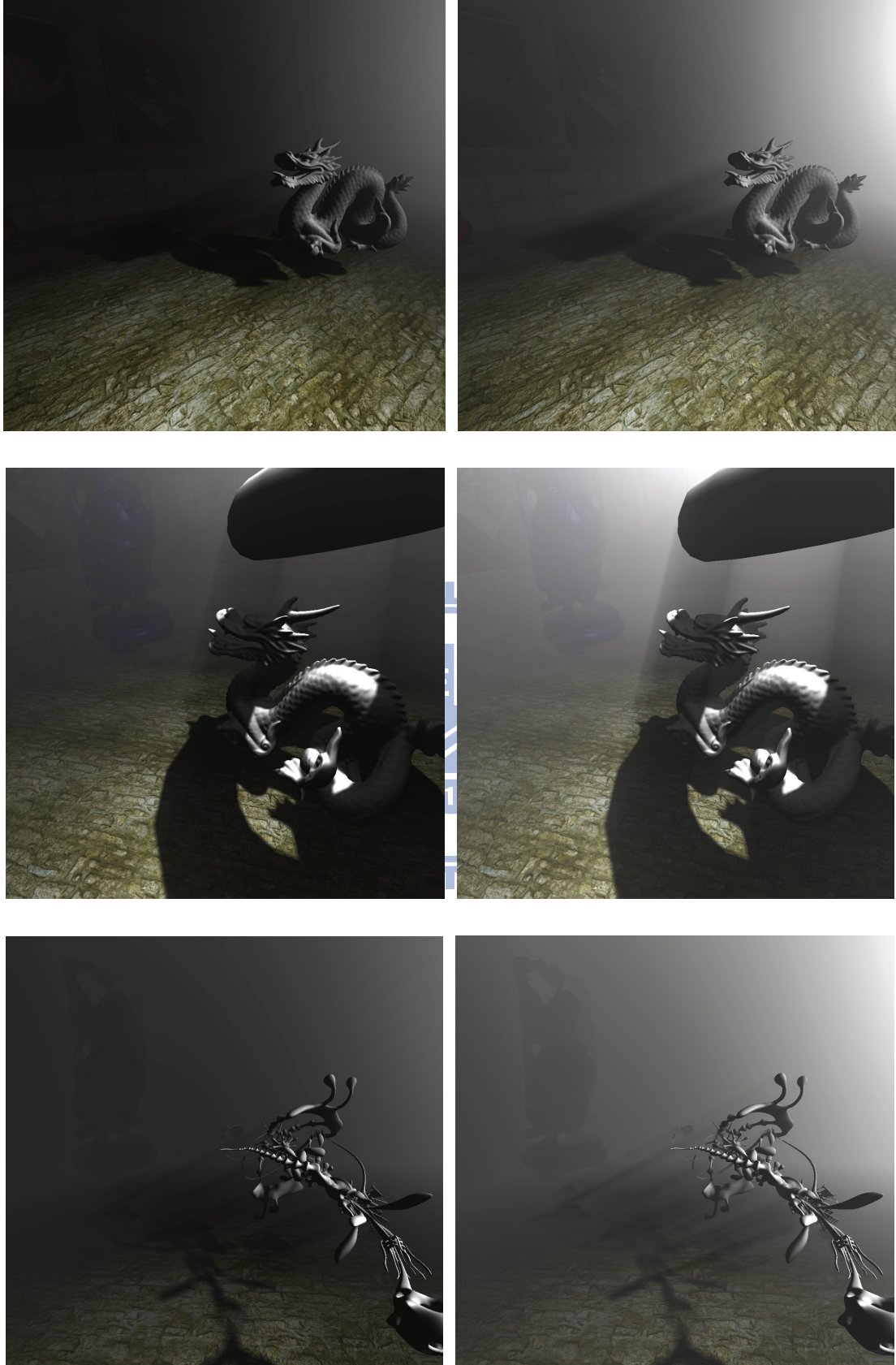
Figure 5.3 (Left) Anisotropic phase function (Right) Isotropic phase function

Table 5.1 shows the performance of our implementation. If the scene contains many light regions in shadow regions, the computation cost is expensive. For example, the wire mesh results in the situation as shown in Figure 5.4.

Table 5.1 frame rates with various scenes

scene	FPS	
	Anisotropic phase function	Isotropic phase function
Donut & dragon	6	23
Dragon	12	24
YeahRight	6	17
Many objects	8	18

Figure 5.4 dark and light regions are interlaced

Chapter 6
Conclusions and Future Work

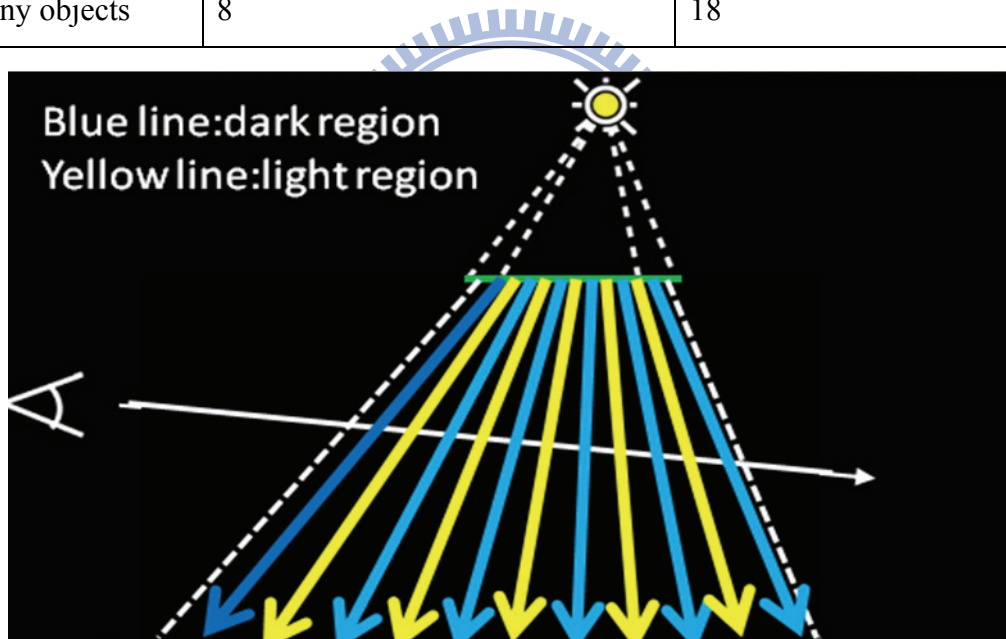In this thesis, we propose a novel method of single scattering for volumetric shadows by using simplified analytical radiative transport equation and our sampling method. Our result can achieve real-time when we use the isotropic phase function, interactive when we use anisotropic phase function. We obtain an accurate evaluation of the radiative transport equation by Pegoraro et al. [20]. And we use the reformulation in Chapter 4.4 and our sampling method to speed up our algorithm.

There are some limitations. If an object is complex like the wire mesh in a scene, the computation cost is expensive. Every light region needs the complex computation of the radiative transport equation, as shown in Figure 5.4. We use the radiative transport equation by Pegoraro et al. [20], restricting us to use point light or spotlight.

In the future we would like to apply our method to other types of light source. Other possible extension would be multiple scattering. One could think about multiple scattering that use point-spread function (PSF) [21] to generate a practical result. Hegeman et al. [22] propose a method that produces high quality imagery at interactive frame rates.

References

[1] Sun, B., Ramamoorthi, R., Narasimhan, S., and Nayer, S. 2005. A practical analytic single scattering model for real time rendering. ACM Transactions on Graphics 24(3), 1040-1049.

[2] Wyman, C., Ramsey, S., 2008. Interactive volumetric shadows in participating media with single-scattering. IEEE Symposium on Interactive Ray Tracing,87-92

[3] Pegoraro, V., Parker, G.,S., 2009. An analytical solution to single scattering in homogeneous participating media. Proceedings of the 30th Eurographics Conference,28(2),329-335

[4] Pegoraro, V., Schott, M., Parker, G., S., 2009. An analytical approach to single scattering for anisotropic media and light distributions. Proceedings of the 35th Graphics Interface Conference,71-77

[5] Hoffman, N., Preetham, A., 2002. Rendering outdoor light scattering in real time. Proceedings of Game Developer Conference

[6] Crow, Frank, 1977. Shadows algorithms for computers graphics. Computer Graphics, Vol 11, no.3, (SIGGRAPH 1977)

[7] Cowley, Les, available online at
http://www.sundog.clara.co.uk/atopics/phenom.htm.

[8] Dobashi, Y, Yamamoto, T, Nishita, T, 2002. Interactive rendering of atomospheric scattering effects using graphics hardware. Graphics Hardware, 99-107

[9] Imagire, T, Johan, H, Tamura, N, and Nishita, T, 2007. Anti-aliased and real-time rendering of scenes with light scattering effects. The Visual Computer, 23(9), 935-944

[10] Max, N, 1986. Atmospheric Illumination and shadows. Computer Graphics (SIGGRAPH 86), Vol 20, no.4, 117-124

[11] Biri, V, Arques, D, Michelin, S, 2006. Real time rendering of atmospheric scattering and volumetric shadows. Journal of WSCG, 14, 65-72

[12] James, R, 2003. Graphics programming methods, chapter True volumetric shadows, pages 353-366. Charles River Media

[13] Mesh, R, 2001. Hardware-accelerated real-time rendering of gaseous phenomena. Journal of graphics tools, 6(3), 1-16

[14] Cerezo, E, Perez-Cazorla, F, Pueyo, X, Seron, F, Sillion, F, 2005. A survey on Participating media rendering techniques. The Visual Computer, 21(5), 303-328

[15] Blinn, F, J, 1982. Light reflection functions for simulation of clouds and dusty surfaces. SIGGRAPH, 16(3), 21-29

[16] Nishita, T, Miyawaki, Y, Nakamae, E, 1987. A shading model for atmospheric scattering considering luminous intensity distribution of light sources. SIGGRAPH, 21(4), 303-310

[17] Willis, J, P, 1987. Visual simulation of atmospheric haze. Computer Graphics Forum, 6(1), 35-42

[18] Hoffman, N, Preetham, J, A, 2002. Rendering outdoor light scattering in real time. ATI White Paper

[19] Lecocq, P, Michelin, S, Arques, D, Kemeny, A, 2000. Mathematical approximation for real-time lighting rendering through participating media. Pacific Graphics, 400-401

[20] Pegoraro, V, Schott, M, Parker, G, S, 2009. An analytical approach to single scattering for anisotropic media and light distributions. Proceedings of Graphics Interface, 324, 71-77

[21] Premoze, S, Ashikhmin, M, Ramamoorthi, R, Nayer, S, 2004. Practical rendering of multiple scattering effects in participating media. Eurographics Symposium on Rendering

[22] Hegeman, K, Ashikhmin, M, Premoze, S, 2005. A Lighting Model for General Participating Media. Symposium on Interactive 3D Graphics