APPROVAL SHEET

Title of Thesis: VISUALIZATION OF SMOKE AND FIRE DATA BASED ON THE FIRE DYNAMICS SIMULATION MODEL

Name of Candidate: TAEKYU SHIN
MASTERS OF SCIENCE,
2013

Thesis and Abstract Approved: ________________________
Marc Olano
Professor
Department of Computer Science and
Electrical Engineering

Date Approved: ______________________________

Curriculum Vitae

Name: TAEKYU SHIN.

Permanent Address: 5001 Westland Blvd, Apt E Halethorpe, Maryland USA 21227.

Degree and date to be conferred: Masters of Science, May 2013.

Place of Birth: Seoul, South Korea.

Secondary Education: Dongjak High School, Seoul, South Korea.

Collegiate institutions attended:

University of Maryland Baltimore County, Masters of Science, 2013.
Sejong University, Bachelor of Science, 2010.

Major: Computer Science.

Minor: Not Applicable.

Professional publications:

Not Applicable.

Professional positions held:

Nemetschek. (June 18th 2012 – Present).

ABSTRACT

Title of Thesis: Visualization of Smoke and Fire Data Based on the Fire Dynamics Simulation Model

Taekyu Shin, Masters of Computer Science, 2013

Thesis directed by: Marc Olano, Professor
Department of Computer Science and
Electrical Engineering

This thesis presents a technique to visualize smoke and fire data generated from the NIST Fire Dynamics Simulation model using Monte-Carlo single and multiple scattering. I use established physics to extract the necessary data to visualize the smoke. The major challenge in smoke rendering with the fire simulation data is that the simulation resolution is not high enough for the visual effects of radiation occlusion. Standard volume lighting methods do not work well for fire since extensive areas can radiate as light sources, making the rendering prohibitively expensive. The proposed concepts of voxel lighting and spherical lighting mimic blackbody radiation in a unit area, approximating area diffuse lights with a sampling of point lights. I address this computational challenge with a selective light sampling scheme based on distance for fire smoke, and additionally with importance sampling of multiple scattering directions. This thesis also provides some analysis of a multiple scattering scheme based on Metropolis-Hasting sampling.

Visualization of Smoke and Fire Data Based on the Fire Dynamics Simulation Model

by

Taekyu Shin

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Masters of Computer Science
2013

ⓒ Copyright Taekyu Shin 2013

I dedicate this thesis to the person who is the highest and most humble that started at the lowest point.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Professor Marc Olano for continuously supporting my MS study and research, with his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me throughout the research and writing of this thesis. I could not have imagined having a better advisor and mentor for my MS study.

I would also like to thank the rest of my thesis committee: Professor Penny Rheingans for her insightful comments, questions and encouragements, and Professor Zhibo Zhang for his tremendous help regarding the physics, as well as his encouragement and questions.

TABLE OF CONTENTS

DEDICATION . . . . **ii**

ACKNOWLEDGMENTS . . . . **iii**

LIST OF FIGURES . . . . **vi**

LIST OF TABLES . . . . **ix**

Chapter 1 INTRODUCTION **1**

Chapter 2 BACKGROUND AND RELATED WORKS **5**

2.1 Fire Model 5
2.2 Global Illumination 6
2.3 Scattering 8
  2.3.1 Single Scattering 9
  2.3.2 Multiple Scattering 10
2.4 Monte-Carlo and Importance Sampling 12
2.5 Material Properties 15
  2.5.1 Blackbody 15
  2.5.2 Effects of Material Properties for Fire Smoke 15

Chapter 3 PHYSICALLY BASED MODEL 17

3.1 Fire Simulator 17
3.2 Light Transport 18
3.3 Fire Lighting 20
    3.3.1 Color 22
    3.3.2 Closest Light Approximation 25
    3.3.3 Light Placement Within Voxel 25
3.4 Multiple Scattering Directions 27

Chapter 4 IMPLEMENTATION 31

4.1 Frameworks 31
    4.1.1 Voxelization 38

Chapter 5 RESULTS 39

Chapter 6 CONCLUSION 52

6.1 Future Work 53
    6.1.1 Closest Light Approximation 53

Appendix A ALTERNATIVE APPROACHES 55

A.1 Overview 55
    A.1.1 Metropolis Light Transport Based Method 56
    A.1.2 Analysis 58

REFERENCES 62

LIST OF FIGURES

2.1 Cloud edges display multiple scattering effects. 9

3.1 Visualization of single and multiple scattering. Green rays represent single scattering while red ones represent multiple scattering. Multiply-scattered rays can reach both the eyes while singly-scattered rays cannot reach the eye 2 very well $(Ashikhmin2004)$. 21

3.2 Visualization of using HRRPUV gradients for picking importance multiple scattering directions. Yellow dots are scattering points. Red lines are multiple scattering directions. 27

4.1 Visualization of the room at the 12th second. The eye position and 'look at' position are shown. Unless indicated, the image is rendered with the view position settings. I flipped the image horizontally because Smokeview is right-handed and mine is left-handed. The fire in the image is based on a visualization of the HRRPUV data. 32

4.2 Visualization of each data. All the data are projected onto one plane (y=0). . 33

5.1 Comparison between single and multiple scattering. Notice that the visual effects from multiple scattering are more obvious at the fire edges. 40

5.2 Performance comparison for single and multiple scattering. 41

5.3 Image comparison based on different light sampling methods and numbers of lights. The RMS values of Figures 5.3(c), 5.3(d), 5.3(e), and 5.3(f) against Figure 5.3(a) are 3.1605, 3.1884, 3.5112, 5.2350. These are computed with the RGB components and averaged. The maximum value for each channel is 255. 44

5.4 The colors are generated by Planck's law and the Planckian Locus respectively. Note that there is no noticeable difference in the colors. The RMS error value of Figure 5.4(b) against Figure 5.4(a) is 2.1829. 45

5.5 Comparison of resulting images between different light placement sampling. Image 5.5(a) looks very regular due to a single point light per voxel while the others don't. Please note that the noise in each image is not the limitation of these techniques, but it comes from a very small number of samples. 46

5.6 Comparison between different light placement sampling. Notice the difference in noise in each of the images. In comparison to the reference image, the RMS error values of Figure 5.6(b), 5.6(c), 5.6(d) and 5.6(e) against Figure 5.6(a) are 2.5290, 1.8379, 2.4212, and 2.5163 respectively. 47

5.7 Image comparison between isotropic and anisotropic multiple scattering directions. Image 5.7(d) uses the variable scattering direction method. The RMS error values of Figures 5.7(b), 5.7(c), and 5.7(d) against Figure 5.7(a) are 3.1090, 2.7658, and 1.8648 respectively. 48

5.8 Comparison between the methods based on the Stefan-Boltzmann law and HRRPUV. 49

5.9 Comparison between constant and variable material coefficients. 51

A.1 The figure is rendered with Premoze et al.'s method (2004). This has homogeneous volume. 56

A.2 Buckyball volume rendered by the metropolis-based method. There are three point-lights at the three densest points in the buckyball volume. The volume for Figure A.2(a) has typical properties of fog. For Figure A.2(b), the volume properties are similar to blackbody. 57

A.3 Illustration of bidirectional tracing for volume (Lafortune & Willems 1993; Rushmeier 1988). 59

A.4 Scattering directions are illustrated around each scattering point over the circle. The scattering angles are anisotropic. Each scattered ray will scatter again after a ray step forward. 59

A.5 Illustration of my modified mutation strategy (Kelemen *et al.* 2002). The length of the blue ray in a denser area is shorter than that of the red one. The yellow ray is shorter than the blue one. Un-mutated directions are in transparent colors while mutated directions are in opaque colors. The yellow ray mutates more in a denser area than the blue one. 60

LIST OF TABLES

5.1 The rendering time comparison between single and multiple scattering. . . 41

5.2 The rendering time comparison between the Monte-Carlo light sampling and closest light sampling. 43

5.3 Rendering time comparison with voxelization and without it. 50

Chapter 1

INTRODUCTION

How do we visualize smoke? In the field of computer graphics, there has been much research done in visualizing volumetric materials such as fog and smoke. There are not as many publications regarding a more interesting situation where light lies inside the volume, such as a fire generating smoke. Many studies address solving light transport for rendering fog; however, few research papers address the rendering of fire smoke. Some of the research covers only part of the problem (Rasmussen *et al.* 2003) and the quality of their results depends greatly on animators rather than being based purely on simulated data (Stam & Fiume 1995).

Rendering participating media such as fog and smoke may be crucial for certain realistic scenes in computer graphics because the unique visual characteristics of rendered media attract attention from viewers. Over the last twenty years, there has been significant research done for realistic visualization of materials. However, the rendering of participating media has been recently studied since it usually requires more computing power. Especially in the last five years or so, there has been an extensive amount of research on volumetric rendering because the physics used for volumetric lighting is complex, resulting in a lengthy rendering time (Ashikhmin 2004).

Smoke generated by fire (fire smoke) is particularly complex. First, there is no fixed light position and a large volume could be emitting light. Typical volumetric rendering methods may not work, or render slowly because of numerous radiating materials. Thus placing virtual lights and picking significantly contributing lights are important.

Second, there are not many ways to simplify multiple scattering because light occlusion can be significant in fire smoke situations. Multiple scattering is often approximated with a blur based on distance and density, but the visual look of fire smoke is very challenging to achieve this way because most of the blurring methods do not consider extinction between scattering points where the extinction contributes significantly to the look.

Third, most light sources have a fixed color, but the chrominance of fire is not fixed. With color determined by temperature, fire colors have to be determined dynamically during the run-time. Previous research has provided only a complex solution. Additionally, the radiant flux may not be fixed at every frame, since fire and smoke animate together based on the physical properties. These are directly related to the visualization of such smoke and thus getting them right is very important. Also, although it is somewhat less significant when it comes to fire smoke volume rendering, the phase function relates the scattering directions of light by the smoke.

In this thesis, I render fire smoke images, using the physically generated data by Fire Dynamics Simulator 5 (FDS5) (McGrattan 2007). FDS5 uses physical laws to generate blackbody smoke density and temperature. For a detailed description for FDS5, please see Section 3.1.

There are five contributions in this thesis. The first is to represent fire blackbody radiation as a set of point lights and physically determine fire color and its power. Previously to determine fire color and power, Planck's law was used (Nguyen, Fedkiw, & Jensen 2002). However, I used a different physics law that approximates the same and the deviation of the approximation is only 9×10^{-5} in the CIE 1960 UCS color space. My method is about 33 times faster and simpler to implement as well. For fire power, Heat Release Rate per

Univ Volume (HRRPUV) should be used. The Stefan-Boltzmann law should not be used because the Stefan-Boltzmann law assumes the emissivity of blackbody.

The second is the combination of selecting a fire light source for faster execution and approximating the light position sampling for the fire light source. For light position sampling, Nguyen et al. (2002) solve a similar problem by relying on the Monte-Carlo sampling as rays hit the flame area. Since their method is based on Monte-Carlo position sampling, it likely introduces bias with a small number of samples. However, because my position sampling method is based on a well-distributed sequence, it may introduce less bias with a small number of samples. Also, the sampling points can be precomputed and repeatedly used for rendering.

The third is the comparison between single and multiple scattering particularly for fire smoke. From the previous research, it is not clear what kind of visual effects multiple scattering contributes to for fire smoke rendering (Stam & Fiume 1995; Nguyen, Fedkiw, & Jensen 2002; Rasmussen *et al.* 2003).

The fourth is the simulation of multiple scattering direction sampling particularly for fire smoke. Utilizing the gradient of the physical data, I achieve a faster convergence for fire smoke rendering.

The fifth is that I show that a constant scattering coefficient may not work well for the rendering of smoke outside a flame area. This is slightly different from Mulholland and Choi's finding (Mulholland & Choi 1998). They state that smoke generated from acetylene fuel has almost universal scattering coefficients, and my results show that the universal scattering coefficients only work well in the flame area.

Chapter 2 discusses the general background and related works. Chapter 3 gives theoretical background related to my methods and discusses my methods. Chapter 4 discusses the overall framework and the details of my implementation. Chapter 5 discusses some important results for my methods. Chapter 6 includes the conclusions of this thesis. Appendix

A discusses my other methods and provides computational analysis of it.

Chapter 2

BACKGROUND AND RELATED WORKS

While considering the simulation of fire and smoke, one must be aware of the combustion process. Combustion is the process in which solid or liquid fuel reaches a particular characteristic temperature causing a phase change within the matter. Typically, it happens when hot gaseous fluid travels from the burning points. Light and energy move through the space, abiding by heat transfer theory (Siegel 2002). These processes involve light transport theory through the participating media, combustion processes that generate heat and radiation, fluid simulation of smoke, radiation of smoke, material physics, and sampling theories of photons. Based on these, this chapter will discuss the basis for global illumination involving fire modelling, non-volume rendering, scattering theories, statistical sampling and material properties.

2.1 Fire Model

Smoke generated by fire creates an interesting and turbulent scene. The visual characteristics of pictures of fire are closely related to how fire and smoke occur. Fire is radiation and heat released through the process of combustion. Combustion involves the state of transition from solid fuel or liquid to gaseous fuel. Through the chemical process of reduction-oxidation, the fuel becomes a gaseous material flowing from the combustion points, meaning, the gas takes convective paths through space. Fire color is known to follow Plancks law of blackbody radiation, describing the relationship between the temperature of materials and its spectral radiance.

For fire rendering, it is very important to simulate both fire and smoke because the material of the fire smoke itself may radiate light. Thus, it is important to know the flow and movement of the smoke. Incompressible gaseous material flows are expressed by the Navier-Stokes equations. Stam and Fiume take user input for a wind field and determine advection from the wind field, so this method allows users to manipulate the smoke movement before final rendering (Stam & Fiume 1995). Fedkiw et al. (2001) use a grid structure where each voxel has a convective velocity field to render animated fire. Nguyen et al. (2002) render hot and non-hot gaseous products with separate sets of incompressible flow equations. Rendering the implicit surface of the gas and the hot gas together, they achieve the rendering of smooth and turbulent fire models. They also describe a transition model from blue core to blackbody radiation. My method, discussed in Section 3.3.1, is inspired by the transition model.

2.2 Global Illumination

Much of the global illumination research considers only the interaction between light and surfaces for realistic lighting of 3D scenes. Local illumination considers only a single energy exchange, from light to surface to eye. Global illumination algorithms consider multiple energy exchanges, and that photons traveling may scatter in multiple directions at each scattering point. Thus global illumination typically requires more computation than the original ray-tracer proposed by Turner Whitted (1980), which considers multiple bounces for each ray, but only one reflection or one refraction direction at each bouncing point. One important characteristic of global illumination is conservation of energy. For example, many bounces of a photon may distribute energy, but the energy cannot disappear, so it must be transferred somewhere or turned into a different form such as heat. The rendering equation by Kajiya (1986) represents transport of light energy, defining incoming radiance and outgoing radiance at a point. There are many different ways of achieving the rendering of 3D scenes: *radiosity, ray tracing, path tracing, Metropolis light transport, and photon mapping*. All of them are a form of solving the rendering equation.

$$L_o(x,\omega_o,t) = L_e(x,\omega_o,t) + \int_{\Omega} f_r(x,\omega_i,\omega_o,t) \cdot L_i(x,\omega_i,t) \cdot (\omega_i,\mathbf{n}) \cdot dw_i \tag{2.1}$$

$L_o(x, w_o, t)$ is light emanating from point x at time t in the direction ω_o. It is the summation of the emitted light, L_e, and reflected light from every incoming direction to the point x. $f_r(x, w_i, w_o, t)$ is the bidirectional reflectance distribution function, describing how much light reflects from direction ω_i to direction ω_o, modeling the attenuation of radiance due to microfacet occlusion and phase attenuation. $L_i(x, w_i, t)$ is the incoming light, and the last term, $(w_i, \mathbf{n})$, the dot product between the incoming direction and surface normal, is the attenuation term, because incoming radiance is projected onto a wider surface based on the incident angle. Since the implication of this equation is that one object in a scene will affect every other object, path tracing solves this integral by tracing paths of photons through the scene. It is computationally very expensive for path tracing to converge to the correct solution to the equation. Photon mapping simulates photon bouncing off of a surface or particle then approximates the rendering equation by accumulating photons from nearby surface locations to achieve global illumination (Jensen 1996). Metropolis light transport is similar to path tracing, but after finding a path that contributes to the final image, it considers other similar paths as they are also likely to contribute to the final image (Veach & Guibas 1997).

For volumetric rendering, much of the previous research attempts to solve the radiative transfer equation (Chandrasekhar 1960).

2.3 Scattering

The scattering model of light transport through a volume probabilistically considers the interaction between photons and particles in the volume. When radiation is scattered by one localized scattering center, it is called single scattering (Gonis & Butler 1999). Scattering events are not deterministic. It is not guaranteed that scattering events will necessarily happen, through a dense area. However, because it is easier to turn single scattering into a form that can be computed, the deterministic approach is often taken to render fog based on the assumption that single scattering takes place at a fixed interval for every eye ray (Ashikhmin 2004; Max 1994; Ren *et al.* 2008; Kajiya & Von Herzen 1984). I have also taken this approach for the single and multiple scattering solutions. However, another method that I implemented is discussed in Appendix A.1.1 that does not take this approach, instead using a variable ray step size for ray-marching. Multiple scattering is analogous to single scattering but for photons that bounce more than once. In light transport, multiple scattering is known to have some physical phenomena: temporal spreading, angular spreading, and spatial spreading (Premože, Ashikhmin, & Shirley 2003; Premože *et al.* 2004). There are certain effects that single scattering cannot achieve, but multiple scattering can. These phenomena occur in any fog, and they may be more visible in fog that has lots of cluttering edges (Tessendorf & Wasson 1994b) and possibly smoke generated by fire. See Figure 2.1 for a picture of the natural phenomena and Figure 5.1 for my rendered example of similar effects.

Another important aspect of multiple scattering, in particular in the field of Computer Graphics, is that it can be approximated as a diffusion process. The probability for multiple scattering in typical volumetric rendering is very high so that simple attenuation of light would work consistently. The transported light through the diffusion process of multiple scattering can provide more of a transparent effect through the gaseous material.

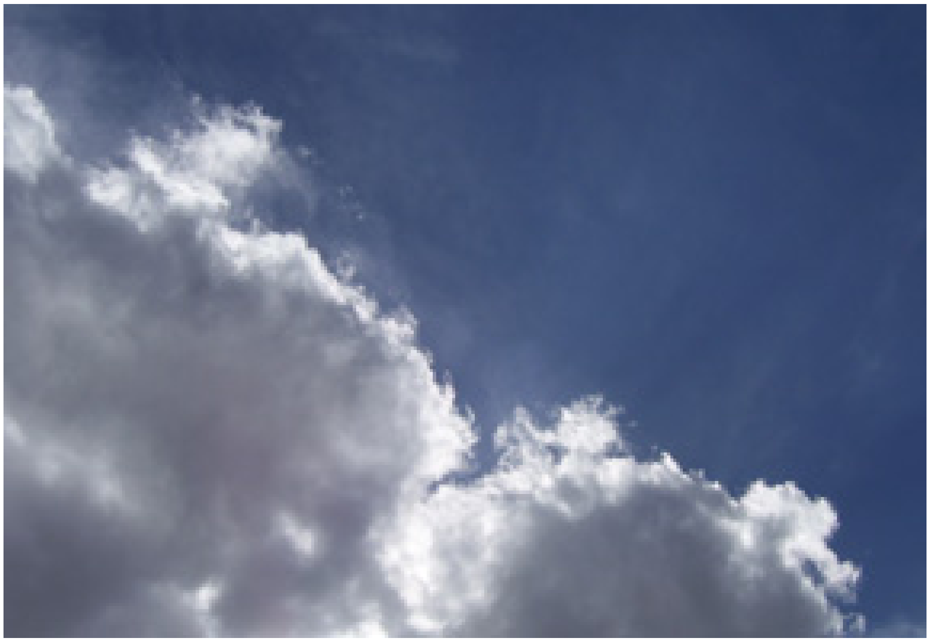


FIG. 2.1. Cloud edges display multiple scattering effects.

2.3.1 Single Scattering

Single scattering has been often used for rendering gaseous volume material. Since Kajiya and Von Herzen (1984) proposed two methods for rendering clouds using single scattering with voxel data structures, many research studies on volumetric rendering have incorporated similar frameworks to visualize inhomogeneous volumes. Rushmeier et al. (1987) explored the rendering-related equations and introduced emission, scattering, and absorption by participating media and extended the radiosity equation to solve such rendering. Many other research methods have used the particular single scattering framework for volumetric rendering (Baran *et al.* 2010; Sun *et al.* 2005;

Ren *et al.* 2008; Dobashi, Yamamoto, & Nishita 2002; Hu *et al.* 2010; Chen *et al.* 2011; Venceslas, Didier, & Sylvain 2006; Inoshita *et al.* 2012; Bernabei *et al.* 2010). Dobashi et al. (2002) proposed a method that turned each ray step along a ray-marching into a texture lookup so it is fit for programmable graphics hardware. Sun et al. (2005) combine the precomputed scattering diffusion and turn the analytical scattering functions into texture lookups in order to express single scattering. Venceslas et al. (2006) show natural shadows from sampling of once-scattered participating media. Baran et al. (2010) do not account for single-scattering, but show through ray-marching how the crespuscular beam effects can be achieved. Chen et al. (2011) store minimum and maximum ray-marching distance in the tree data structure that is used to accelerate the ray-marching. Hu et al. (2010) achieve natural phenomena, such as crespuscular effects, the sky, clouds, etc, by using precomputed single scattering effects. Typically, single-scattering is not very good at achieving translucent effects (Stam & Fiume 1995; Ashikhmin 2004; Premože, Ashikhmin, & Shirley 2003; Premože *et al.* 2004). However, with some special precomputation of adaptive sampling of the inner material, the effects can also be achieved with single-scattering (Bernabei *et al.* 2010) by computing light transport through a set of locations inside the material. Inoshita et al. (2012) give the shape determined by single-scattering.

2.3.2 Multiple Scattering

Radiosity-based multiple scattering was proposed by Max (1994). The discrete ordinate method for approximating multiple scattering was used; however, this method has not been used very much since then. Tessendorf (1994b) reached the theoretical basis for multiple scattering as a diffusion. However, the diffusion process is usually based on the assumption that the volume smoke density is sufficient for scattering events. Another work by Tessendorf (1994a) determines that multiple scattering effects called silver-lining effects. They are obvious on cloud edges and some cluttering visual effects at grazing angles.

Based on the theory of replacing multiple scattering with diffusion of light, practical expansion was done by Stam (1995). His practical solution for diffusion processes is through energy exchange between grids or blobs in the spatial data structure. The two global illumination methods, photon mapping by Jensen et al. (1996) and Metropolis light transport by Veach et al. (1997) were extended to incorporate volumetric rendering by Jarosz et al. (2008) and Pauly et al. (2000). I have implemented a method based on Pauly's work (2000), but my method significantly differs. Because that is not my finally chosen method, I will discuss my implemented method and an analysis on how this would be inefficient for fire smoke simulations in Appendix A.1.1. Jensen et al. use an approximation similar to diffusion to render translucent material (Jensen *et al.* 2001). I explored Premoze et al.'s work and my final method takes a Monte-Carlo raytracing approach stated in the paper (Premože, Ashikhmin, & Shirley 2003). So I take the path integral approach, not the diffusion process, differing from works based on diffusion processes (Donner & Jensen 2005; Stam 1995; Stam & Fiume 1995; Ashikhmin 2004). Premoze et al. (2004) use the point spread function that is a very similar method to diffusion. Ashikhmin et al. (2004) use the Most Probable Path approach but determine angular and spatial space per sample path in order to approximate multiple scattering. Donner et al. (2005) achieve the rendering of translucent materials. Their technique is not technically multiple scattering, but is similar in the sense that they use light transport for diffusion. Szirmay-Kalos et al. (2009) spread rays from a point light and dropped a Gaussian diffusion energy at every scattering point, assuming that the diffusion is multiple scattering and that all volume has the same material properties in the unit spherical area. While Szirmay-Kalos et al.'s work (2009) starts rays from a light point and leaves energy for an area, Guo et al. (2012) start from the camera point and the method is a simple tracing of the radiative transfer equation for the scattering points . The computation of each step along the ray-marching can be done in parallel in NVIDIA's Compute Unified Device Architecture (CUDA) to achieve real-time rendering of the volume data. After the single scattering propagation from Kaplanyan's work (2010), Billeter et al. (2012) turn the energy in each voxel, represented in spherical harmonics, to a light source and do propagation from each light source to adjacent voxels.

Multiple scattering is known to be useful for some other kinds of rendering. Bruneton et al. (2008) show that multiple scattering accounts for the rendering of sun light twilight. Some works have used multiple scattering for the rendering of hair (Moon & Marschner 2006; Zinke *et al.* 2008; Moon, Walter, & Marschner 2008).

2.4 Monte-Carlo and Importance Sampling

Monte-Carlo tracing is commonly used for approximating the numerical simulation for volumetric rendering by naively averaging many photon samples. This proved very costly in terms of rendering times. However, as long as samples are well distributed, the result cannot be biased and thus will converge to the correct solution. For multiple scattering, the computation is more expensive, meaning that the computation takes a much longer time to converge than single scattering. Monte-Carlo tracing has the exponential time complexity of O($kn^{kn^{kn\dots}}$) where k is the number of scattering points and n is the number of incoming light flux directions considered at each scattering point through raymarching. Fortunately, the number of scattering points can be averaged to a constant and thus resulting in complexity O(n^n).

In Monte-Carlo tracing, the integration

$$I = \int_{\Omega} f(x)dx \tag{2.2}$$

is approximated as

$$I = \frac{1}{N}\sum_{i=1}^{N} f(X_i). \tag{2.3}$$

In Monte-Carlo tracing for volumetric rendering, it is typically as below.

$$L_{i(m)}(x, \omega_i, t) = P(L_{o(m-1)}(x, \omega_o, t)) \tag{2.4}$$

$P(\theta)$ is a phase function where $\int_\Omega P(\theta)d\Omega = 1$. $L_{i(m)}(x, \omega_i, t)$ and $L_{o(m-1)}(x, \omega_o, t)$ are from equation 2.1. This equation means that the previous outgoing radiance, $L_{o(m-1)}$, is used as an incoming radiance, $L_{i(m)}$. Because this structure results in significant computation, importance sampling can be used to converge to the final solution more quickly. An importance sampling estimator for function $f(x)$ over domain Ω is

$$I = \int_\Omega \frac{f(x)}{p(x)} \cdot p(x)dx \tag{2.5}$$

and its estimator in implementation is

$$I = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x)}{p(x)} \tag{2.6}$$

where $p(x)$ is the importance density. The choice of $p(x)$ determines if the importance sampling is biased or not. The ideal sampling density is

$$p(x) = \frac{f(x)}{\int_\Omega f(x)dx} \tag{2.7}$$

though achieving this density would require solving the original integral. However, sampling densities that closely estimate $f(x)$ still significantly improve the Monte-Carlo convergence. This implies that, to successfully sample any radiance around the sphere over domain Ω, we need to sample per radiance band. In order to achieve a rendered scene with fine details, there needs to be a large number of samples where the radiance gradient is large. Thus this thesis considers importance sampling when the gradient of the fire smoke data is big.

There has been a lot of similar research to achieve faster or more accurate rendering of volume data. Veach et al. (1997) use Monte-Carlo tracing based on the Metropolis algorithm (Metropolis *et al.* 1953). By accepting samples that are more likely than its previous sample to contribute to the final image, it converges much faster than the naive Monte-Carlo sampling scheme. Veach et al. (1995) provide a way of combining sampling techniques. Lafortune et al. (1996) present bidirectional path tracing for rendering volume data. Kelemen et al. (2002) present another mutation strategy for Metropolis Light Transport based on pseudo random numbers. Ren et al. (2008) present a recursive sampling scheme based on the gradient. It distributes the sample points based on the gradient so that it can visualize detailed volume density. My important sampling method is also based on some gradient data. Zhou et al. (2008) first save the low-frequency approximation with a set of radial basis functions in the precomputation stage and do ray-marching with the data. They use the residual map for extinction effects for the detailed volume data after the first stage. Yue et al. (2010) incorporate the concept of free path sampling, which determines the next step along a ray-marching based on density or optical depth, and extends it. Kulla et al. (2012) place samples along a ray and detach the computation of transmission of the ray from the final one of light transport. For a point light, their PDF function achieves equiangular sampling along a ray. Although it works for many situations, this method is ineffective for fire smoke situations. Specifically, it moves a ray forward based on the CDF, which represents an equiangular step around a single point light. Because fire smoke has numerous light sources, my implementation takes samples from different lights every ray-marching step and thus the CDF will be mixed when sampling different lights.

2.5 Material Properties

2.5.1 Blackbody

Smoke has special properties. It has low albedo, meaning that, the incoming energy tends to be confined in the material at an infinitesimal scattering point near the fire. The gaseous material needs to be at a high temperature to emit light. It loses its radiation and color when the gas generated from fire moves away from the fire. Another visual phenomenon commonly shown in a blackbody is occlusion of radiation. The inner part of the blackbody radiates light and the outer part of the blackbody, whose temperature is not high enough to radiate light, will occlude the radiation from the inner one. This creates the unique visual effects of the blackbody. Because the perfect blackbody does not exist, the term, opaque-body, may be used. An opaque-body is important because some visual characteristics in realistic pictures of smoke and fire are due to light transport through the opaque body rather than the perfect blackbody. The material properties of the blackbody are $transmission = 0\%, absorption = 100\%, and\, reflectivity = 0\%$. In contrast, those of the opaque body are $transmission = 0\%, and\, absorption + reflectivity = 100\%$ (Incropera 2006). For instance, if blackbody smoke lies with some light source in the scene and there is no other volume in the space, it would not be visible because it absorbs all the visible radiance. However, the perfect blackbody does not exist, but the opaque body does in the fire smoke. The black smoke would be still visible due to the reflection of light on the surface of the body. But the FDS5 data show that the emissivity of the opaque body is likely high and close to 1.0.

2.5.2 Effects of Material Properties for Fire Smoke

It is important to use the right properties for fire smoke rendering because the characteristics of the image come from the unique material properties of the smoke itself. Regarding multiple scattering, the scattering angle is important because multiple scattering has a wider scattering angle than single scattering and can make a difference in the visual features of the rendered scene. Mulholland et al. (1990) present a model for the extinction of light from smoke in the RTE (Radiative Transfer Equation) equation. They (1998) also discuss the extinction coefficients of smoke generated by burning fuels made of some materials such as acetylene or ethene. The simulation from the Fire Dynamics Simulator gets input for the properties of burning materials. Those properties include conductivity, specific heat, density, thickness, and burning behavior. Mulholland and Choi (1998) show that a dominant part of the contribution to the extinction coefficient is absorption, not scattering. Absorption makes up a bigger part of the extinction than scattering for the data from FDS5. They also show that the scattering coefficient does not change much in smoke. Tangren et al. (1982) give the explanation of the approximate relationship between the scattering coefficient, mass, and Mie scattering coefficient.

Chapter 3

PHYSICALLY BASED MODEL

This section discusses physical models and my schemes for solving the physical problems. Implementation will be discussed in Chapter 4.

3.1 Fire Simulator

I begin with a simple description of the Fire Dynamics Simulator 5 (FDS5). FDS5 simulates fire dynamics. Its inputs are the grid resolution, burning material properties, ignition source positions, and how long it is simulated. FDS5 runs a form of the Navier-Stokes equations with data dynamically computed and stored in each voxel, including temperature, density, velocity, chemical composition, etc. at every time step. FDS5 can give output data, including soot density, temperature, HRRPUV (Heat Release Rate Per Unit Volume), absorption/extinction coefficient, etc. (McGrattan 2007). At the beginning of Chapter 4, I visit FDS5 again and provide further information.

Although the fire simulation is based on quite rigorous computational fluid physics, the fire rendering software, Smokeview, that comes with FDS5 is not heavily physically based. It is missing some major effects like scattering.

3.2 Light Transport

I use the single scattering equations (Arvo 1993). They are a form of RTE, but simplified for computation. I use the incident volume emittance given below.

$$L_i(x, \omega) = \int_{x_o}^{x} \tau(x', x) \cdot L_e(x', \omega) dx' \tag{3.1}$$

L_i is the in-flux at point x from point x_o. x' is an arbitrary point between point x and x'. L_e is the emittance in the ω direction from point x'. $\tau(x, x')$ function is transmission along the path between point x and x'. Physically, the emittance term is defined as $(1 - \rho) \cdot B(T)$ where $B(T)$ is Planck's law based on temperature T and ρ is the ratio of the scattering coefficient to the extinction coefficient. Since $B(T)$ may be small, I set the emittance to 0 for my simulations for most of the images in this thesis.

$$\tau(x, x') = \int_{0}^{d} \sigma_s(x + t\omega) dt \tag{3.2}$$

where $x' = x + t \cdot d$. x is a point where incident light is coming into and $x' \in \nu$ where $\nu \in R^3$ is the point where the light originates.

σ_e is the extinction coefficient given below as:

$$\sigma_e = \sigma_a + \sigma_s. \tag{3.3}$$

where σ_a is the absorption coefficient and σ_s is the scattering coefficient. Any coefficient is computed as: $\alpha = exp(-ks\Delta x)$ based on the RTE equation (Chandrasekhar 1960). FDS5 gives absorption and extinction coefficients computed based on Mie theory. Please note that the values can't be perfectly accurate because FDS5 computes RTE using a limited number (100 by default) of solid angles and FDS5 only approximately calculates the combustion model (McGrattan 2007). However, I believe that it is accurate enough for visual use.

Based on Equation 3.1, I have the volume transport that incorporates scattering in-flux below (Arvo 1993):

$$T(L_o) = \int_o^x \tau(x', x) \cdot \sigma_s(x') \int_{S^2} f_p(w, x', \omega') \cdot L_i(x', \omega') \cdot d\omega' dx'. \tag{3.4}$$

Now I sum this with the transported incoming radiance by emittance term and refactor it as below (Kajiya 1986; Pauly, Kollig, & Keller 2000).

$$T(L_{total}) = \int_o^x \tau(x', x) \cdot (L_e(x', -\omega) + \sigma_s(x', \omega) \int_{S^2} f_p(w, x', \omega') \cdot L_i(x', \omega') \cdot d\omega')dx'. \tag{3.5}$$

Kajiya (1986) and Pauly (2000) further turn Equation 3.5 into a form of Neumann series by integrating it with the surface integrator operator, which incorporates reflectance from surfaces. I take that approach for my single and multiple scattering. However, an alternative method is described in Appendix A and does not use this approach.

I use the multiple scattering equations from previous works (Nguyen, Fedkiw, & Jensen 2002; Premože, Ashikhmin, & Shirley 2003). It is the same numerical solution. I put the final form without derivation below.

$$\begin{aligned} L_{n+1}(x, \omega) = & \sum_{l}^{allNLights} L_{un}(x, \omega'_l, \omega) \cdot \sigma_s(x) \cdot \Delta x + \\ & \frac{4\pi}{N} \sum_{i=1}^{M} L_{sc}(x, \omega) \cdot P(x, \omega_i, \omega) \cdot \sigma_s(x) \cdot \Delta x + \\ & \tau(x, \Delta x) \cdot L_n(x + \omega \Delta x, \omega) \end{aligned} \tag{3.6}$$

Equation 3.6 defines the relation between the light transport for one step forward while ray-marching along the eye ray. L_{un} is unscattered rays in the direction of ω at the ray step point on the ray. This is computed by directly shooting a ray to the light sources in the direction of ω_l'. $\sigma_s(x, x_l')$ is defined as $\sigma_s(x) = \sigma_s \cdot \rho(x)$. ($\rho(x)$ is density at point x.) L_{sc} is scattered incoming radiance. It is attenuated since it is summed over spherical solid angles. Each incoming radiance is attenuated by the phase function $P(x, \omega_i, \omega)$ where x is the point, ω_i is the incoming ray direction and ω is the eye ray direction. Finally, the last term with τ is a simple attenuation term of the previously computed radiance from the previous ray step.

Regarding multiple scattering effects for my implementation, while Tessendorf (1994a) states that multiple scattering creates some visual effects for smoke, I believe that the essential visual characteristics would not necessarily disappear without multiple scattering because they seem to exist with single scattering (Rasmussen *et al.* 2003). As an example of multiple scattering effects, radiance through the edges of fire smoke should be brighter than that of single scattering. Also, even if fire light is occluded by a blackbody through the single scattering path, light can come out of the smoke by the multiple scattering paths. See Figure 3.1 for a visualization of the situation. This shows that multiple scattering has wider angular and spatial spreading (Ashikhmin 2004).

3.3 Fire Lighting

Fire Dynamics Simulator 5 (FDS5) gives Heat Release Rate Unit Per Volume (HRRPUV) and temperature. However, this information is not enough for rendering. In this section, those problems and their solutions are discussed.

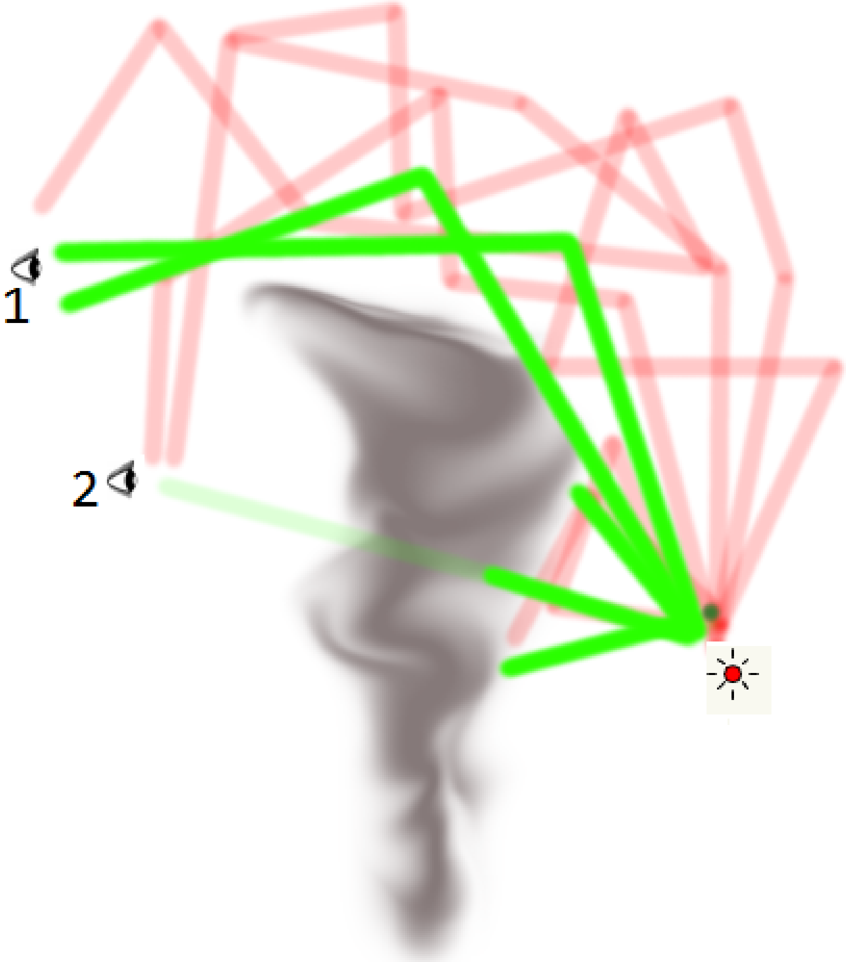


FIG. 3.1. Visualization of single and multiple scattering. Green rays represent single scattering while red ones represent multiple scattering. Multiply-scattered rays can reach both the eyes while singly-scattered rays cannot reach the eye 2 very well $(Ashikhmin2004)$.

3.3.1 Color

It is necessary to first determine what part of the smoke radiates energy because, for fire smoke, not only the fire combustion point, but the blackbody can also radiate energy. However, FDS5 does not give the data of where the fire is or at what point incoming radiance comes from. Also, if some part of the blackbody releases energy, what color or power of the radiance would it be? Nguyen et al. (2002) solve a similar problem. They use Planck's law, though this requires computing power and the spectral radiance for every frequency, summing the power, and using an algorithm such as Bruton's algorithm to convert from frequency data to its corresponding colors in order for the data to be displayed. My solution employs the Planckian locus. It represents the color of fire when blackbody temperature changes. The Planckian locus is solely dependent on the temperature of smoke, which I have. My approach is easier to understand, simpler to implement and also faster.

For simplicity, I assume that the white point for the color space is 1.0. For computational purposes, I use the Planckian locus approximation from Krystek's work (1985) for the temperature from 1000K to 1667k. For higher temperatures than 1667K, I use Kim et al.'s work (Kim Y. 2002). The equations for the approximation are equation below. T is a temperature.

$$\begin{aligned} u &= \frac{0.860117757 + 1.54118254 * 10^{-4} * T + 1.28641212 * 10^{-7} * T^2}{1.0 + 8.42420235 * 10^{-4} * T + 7.08145163 * 10^{-7} * T^2}, \\ &\qquad \text{if } 1000K \leq T < 1667K \\ v &= \frac{0.317398726 + 4.22806245 * 10^{-5} * T + 4.20481691 * 10^{-8} * T * T}{1.0 - 2.89741816 * 10^{-5} * T + 1.61456053 * 10^{-7} * T^2}, \\ &\qquad \text{if } 1000K \leq T < 1667K \end{aligned} \tag{3.7}$$

For the conversion from u and v to RGB, I use the approximate conversion (Macadam

1937). It is

$$\begin{aligned} u &= \frac{4x}{12y - 2x + 3} \\ v &= \frac{6y}{12y - 2x + 3} \end{aligned} \tag{3.8}$$

refactored as:

$$\begin{aligned} x &= \frac{-6uv}{4v(4v - 2 - uv)} \\ y &= \frac{\frac{4x}{u} + 2x - 3}{12} \\ z &= 1 - x - y \end{aligned} \tag{3.9}$$

Once determining x, y and z, the values are converted with sRGB. The sRGB conversion is used for all the resulting images in this thesis.

Below is Krystek's work (1985).

$$
\begin{aligned}
x &= -0.2661239\frac{10^9}{T^3} - 0.2343580\frac{10^6}{T^2} + 0.8776956\frac{10^3}{T} + 0.179910, \\
&\text{if } 1667K \le T < 4000K \\
x &= -3.0258469\frac{10^9}{T^3} - 2.1070379\frac{10^6}{T^2} + 0.2226347\frac{10^3}{T} + 0.240390 \\
&\text{if } 4000K \le T < 25000K \\
y &= -1.1063814x^3 + -1.34811020x^2 + 2.18555832x - 0.20219683 \\
&\text{if } 1667K \le T < 2222K \\
y &= -0.9549476x^3 - 1.37418593x^2 + 2.09137015 * x - 0.16748867 \\
&\text{if } 2222K \le T < 4000K \\
y &= +3.0817580x^3 - 5.87338670x^2 + 3.75112997 * x - 0.37001483 \\
&\text{if } 4000K \le T < 25000K
\end{aligned}
\tag{3.10}
$$

To determine the energy level of the radiation, I use the Stefan-Boltzmann law since it defines that the emitted energy flux of all frequencies per unit area and that the total energy outputted is proportional to the temperature raised to the power of the fourth. All I need to compute the Planckian locus is the temperature. Once I get the power level of each voxel based on the Stefan-Boltzmann law, I multiply the computed color by the power. FDS5 provides the HRRPUV, which is also used as the reference value for power and compare the result later. HRRPUV uses different energy functions depending on whether the space is inside the flame area defined by FDS5. HRRPUV calculated by FDS5 is the same as the Stefan-Boltzmann law except that FDS5 divides it by π, which yields the power per solid angle for the space outside flame (McGrattan 2007). I show later that the Stefan-Boltzmann law does not work well for the space outside flame since the law assumes the blackbody properties.

3.3.2 Closest Light Approximation

The number of point light sources in the volume generated by FDS5 is typically from about 300 to 3000 lights. I used the data from the twelfth second time frame from the fire dynamics simulation with the volume resolution 81x41x21 because it has only 312 virtual light sources. This means that it has to compute light transport for 312 lights every ray-marching step. Since this is prohibitively expensive, I compute light transport for a light at every step but make each step size small enough to compute the transport of most of the light sources. Empirically, the larger the step size is, the more noisy the resulting image is. Due to its heavy computation, I pick the closest voxel and look for lights placed in the voxel. This method yields a good approximation that is also computationally fast.

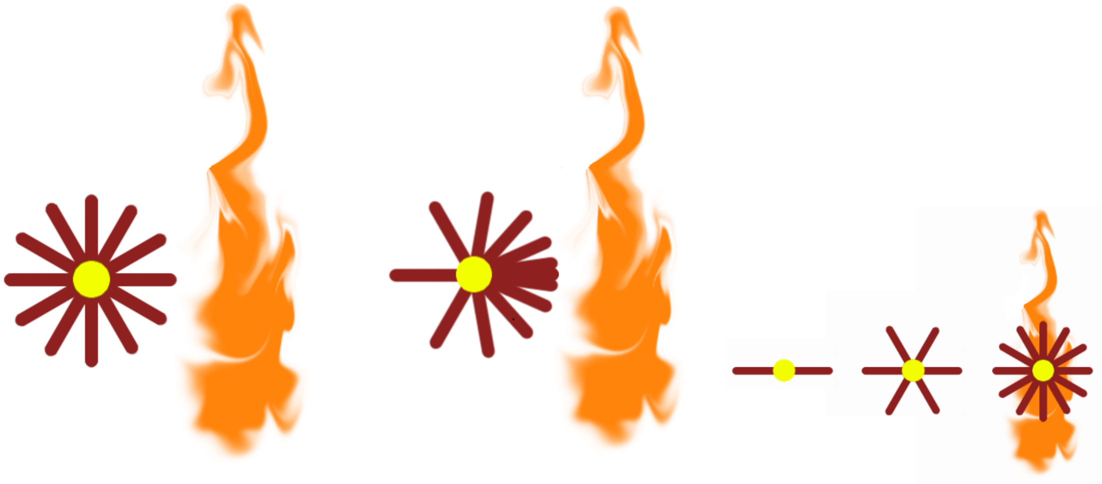
3.3.3 Light Placement Within Voxel

Since FDS5 gives 3 dimensional data for temperature for each voxel of the space, I place lights based on the temperature where $temperature > 1000K$ based on the Planckian Locus. However, the problem is that FDS5 gives data whose voxel dimensions are limited. FDS5 users can control the dimensions, but it is prohibitively slow to generate high resolution data . When it comes to rendering fire smoke, it is imperative to have high resolution data because fire and smoke inherently result in scenes with high frequencies. It is possible that one pixel and its adjacent pixels may have a very different color. Experimentally a simulation, whose voxel dimension was 81 by 41 by 21, took less than a day. A simulation, whose voxel dimension was 121 by 61 by 31, took about three days. A simulation, whose voxel dimension was 201 by 101 by 51 took a couple of weeks. (Notice the exponential growth of the execution time.) 81 by 41 by 21 does not provide high enough detail. Rasmussen et al. (2003) show the data of the resolution of 2000 by 2000 by 2000 displays intricate fire smoke. Fire simulations for movie visual effects often simulate at a lower resolution, and add high resolution detail with procedural noise (Rasmussen *et al.* 2003). A physical visualization of fire smoke data must not add noise because noise functions are not physically based. For this reason, I propose three different light placement schemes for each lit voxel instead.

The first scheme is called *point sampling*. A light is placed at the center of each voxel. The second is called *spherical sampling*. *Spherical sampling* is based on the fact that the shape of fire smoke is often similar to the pattern of the Kolmogorov Spectrum (Stam & Fiume 1995). *Spherical sampling* differs from area lighting because this does not compute any PDF (Probability Distribution Function) or attenuation while area lighting divides the power by the distance of the randomly picked sampling point from the center of the light and also by the solid angle, so typically 4 * PI. For my implementation, the PDF is 1.0 and there is no light attenuation based on the sampling point because its intention is to fake the light position. Thus, its implementation is very similar to disk sampling, but it rejects samples outside the sphere while disk sampling rejects ones outside the disk. The last one is *voxel sampling*. The light position should be well distributed but should not be completely uniform. Some sampling theories that meet the conditions are quasi-random sampling distributions like Sobol sampling, Halton sequences, or random jittered sampling. Thus a modified stratified sampling scheme is used. One voxel is divided by 10x10x10 into 1000 small mini-voxels. Each mini-voxel holds one light point. However, I use a half jitter per sample to avoid completely uniform samples within each voxel. The maximum jitter can move the light position only 1/4 of the mini voxel size to the left or right. Now picking which mini-voxel to light is the issue. From the pool of stratified lights, it is uniformly chosen per every light transport because well-distributed pseudo-random numbers are used as an index in the light array to choose lights. I use van der Corput sequences for that (Kuipers 2005).

3.4 Multiple Scattering Directions

The scattering of light in a blackbody is isotropic. A consequence is that it can take more time to reach the final result because isotropic multiple scattering shoots rays that can end up not contributing to the final image. As common knowledge for importance sampling, it is important to know which direction light changes dramatically, and for fire smoke, it is where the radiating part of the smoke is. Because light power is based on HRRPUV, I use the data to guess where the most dramatic change would be in the 3D space. In the case of the data from FDS5, HRRPUV seems to display the most dramatic changes among all the data. I present two different methods utilizing the HRRPUV gradients for importance sampling. For a visualization of the idea, see Figure 3.2. Figure 3.2(b) shows the first method. It turns the vectors onto fire based on HRRPUV gradients . Thus more samples will go towards fire. Figure 3.2(c) shows the second importance sampling method. Note that more samples are near the flame based on the magnitude of HRRPUV gradients.



(a) Isotropic Directions (b) Important Directions (c) Variable Directions

FIG. 3.2. Visualization of using HRRPUV gradients for picking importance multiple scattering directions. Yellow dots are scattering points. Red lines are multiple scattering directions.

First, I use HRRPUV to determine the directions to compute light transport. I precompute the gradient of HRRPUV between each voxel and its neighboring voxels. I also pre-compute the gradient direction for each voxel. Below is the computation of the gradient and the vector.

$$\vec{v_i} = |NeighborVoxel_i - Voxel| \tag{3.11}$$

$$gradient(i, p) = HRRPUV_i - HRRPUV_p \tag{3.12}$$

$$\vec{v_I} = \frac{1}{n} \sum_{i=1}^{N} \vec{v_i} * gradient(i, p) \tag{3.13}$$

Each voxel has six neighboring voxels. So $N = 6$ in this case. I define six vectors which represent the direction from the centered voxel to its adjacent voxels. $\frac{1}{n}$ is just a normalization term. I sum up all the vectors weighted with each vector's gradient. $HRRPUV_p$ means the HRRPUV value computed at point p. $HRRPUV_i$ is the point p's neighbor voxel's HRRPUV. $\vec{v_I}$ is the gradient vector and it represents the gradient of HRRPUV at point p. A 2D Sobol sequence is pre-computed and turned into 3D vectors as below. Sobol 2D sequences are well distributed position samples on the 2D xy plane. Thus, I can use the Sobol sequences as an input to the inverse Henyey-Greenstein (HG) function, which models the phase function ranging from backward scattering to forward scattering (Henyey & Greenstein 1941). The inverse Henyey-Greenstein function organized based on $\cos\theta$, where θ is the angle between the incoming and outgoing directions (Henyey & Greenstein 1941).

$$\mu = \frac{1}{2g} \left\{ 1 + g^2 - (\frac{1 - g^2}{1 + gs}) \right\}, \text{where } s = 2P - 1 \tag{3.14}$$

μ is $cos(\theta)$. g is a coefficient for the HG function that indicates how forward or backward directed the distribution is. P is a uniformly distributed random variable. Since a 2D Sobol sequence is uniform, I feed it as a probability function to the HG function. Thus I compute s and furthermore μ. The 2D Sobol sequence is called u and v.

$$s = SobolSample.u * 2 - 1 \tag{3.15}$$

Having calculated μ, I can represent well distributed scattering angles based on whatever probability I want. μ is input for simple spherical coordinates to get well distributed vector directions with the starting point $(0, 0, 0)$. The pivotal vector will be $(0, 0, 1)$.

$$x = \sqrt{1 - \mu^2} * cos(2\pi * SobolSample.v) \tag{3.16}$$

$$y = \sqrt{1 - \mu^2} * sin(2\pi * SobolSample.v) \tag{3.17}$$

$$z = \mu \tag{3.18}$$

$\sqrt{1 - \mu^2}$ is $\sin(\theta)$. Because v of the Sobol sequence is inherently normalized to 1.0, it needs to be normalized to 2π to match the maximum azimuth angle.

$$p(\mu) = \frac{1}{2} \frac{1 - g^2}{[1 + g^2 - 2g\mu]^{3/2}} \tag{3.19}$$

Above is the probability function for the HG function about each $cos(\mu)$ based on the pivot vector.

The vectors and PDF can be pre-computed. During the computation stage, a simple transformation turns each pre-computed vector into the vector space of the current eye ray. For instance, when computing light transport, I determine in which direction there is a more important area based on HRRPUV gradients. I need to shoot more samples towards the direction so I do the rotation of every vector based on the direction. It is a simple inverted (transposed) transformation whose basis vectors are $\vec{basis}_1 = \vec{p} \times \vec{a}$, $\vec{basis}_2 = \vec{basis}_1 \times \vec{p}$ and $\vec{basis}_3 = \vec{p}$, where $\vec{p}$ is the pivot vector and $\vec{a}$ is just an arbitrary vector, not aligned with $\vec{p}$. Please note that $\vec{p}$ becomes $\vec{basis}_3$ since the pivot vector was originally (0,0,1). The HG function is forward when g is close to 1. When g is close to 0, it becomes backward scattering. Finally, I sum the computed light intensity based on the basic importance sampling described in Section 2.4.

$$\frac{1}{\sum_1^N pdf(L_i)} \sum_1^N L_i * pdf(L_i) \tag{3.20}$$

Note that $pdf(L_i)$ here is different from $p(x)$ in Equation 2.6. $pdf(L_i)$ here is based on how much of the solid angle the corresponding sample covers. The $pdf(L_i)$ and $p(L_i)$ are inversely proportional.

The second method is to use HRRPUV to determine how many directions light transport should be computed for. Once Equation 3.11 is computed. I use the $\vec{v_I}$ values. The number of multiple scattering directions is the same as $floor(\frac{\vec{v_I}}{200kW})$. This results in a greater number of scattering directions near the fire. See Figure 3.2(c) for a visualization of the idea.

Chapter 4

IMPLEMENTATION

4.1 Frameworks

I use the data from Fire Dynamics Simulator 5 (FDS5) (McGrattan 2007). The data are HRRPUVs, soot density, temperature, absorption, and extinction coefficient per unit 3D space. The units of HRRPUV(Heat Release Rate Per Univ Volume), soot density, absorption/extinction coefficient and temperature are $\frac{kW}{m^3}$, $\frac{mg}{m^3}$, $\frac{1}{m}$ and Celcius respectively. It also provides smokeview, which visualizes the simulated data, but only without considering scattering effects. The Fire Dynamics Simulator is very computationally expensive, so it is difficult to use fine grids. Most of the images in this thesis were generated using the data computed on an 81x41x21 grid. My simulation settings have a room which has a couch, two chairs, a small chair and a wall in the middle of the room. The wall has a door. The fire simulation is simulated for 2000 seconds. For the visualization of the settings, see Figure 4.1. All images shown in this thesis are rendered in the 12th second of simulation. For visualizations of the individual physical quantities, see Figure 4.2.

My implementation has three stages: initialization, precomputation and rendering. The initialization stage is simple initialization of data structures. For the precomputation stage, please see Algorithm 1. My implementation uses Pharr and Humphreys' PBRT as a basis for my implementation (Pharr & Humphreys 2010). It already has single scattering

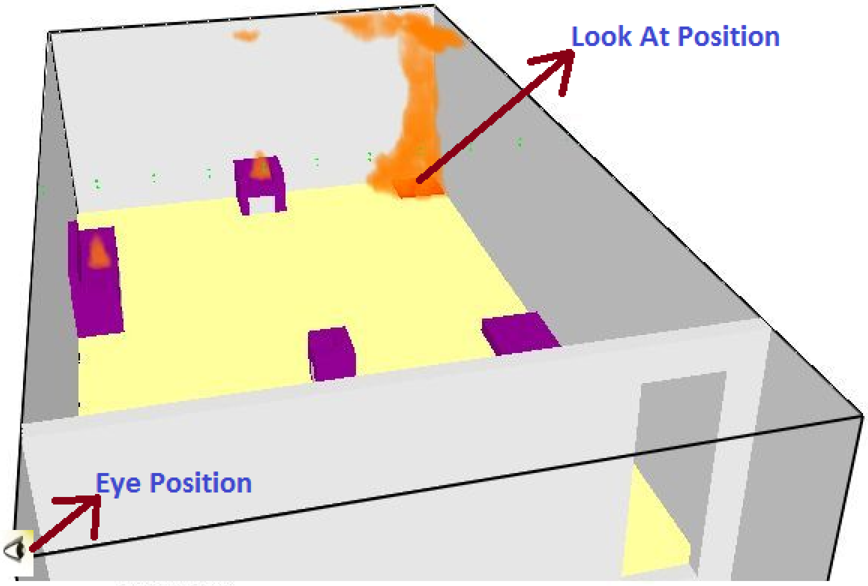
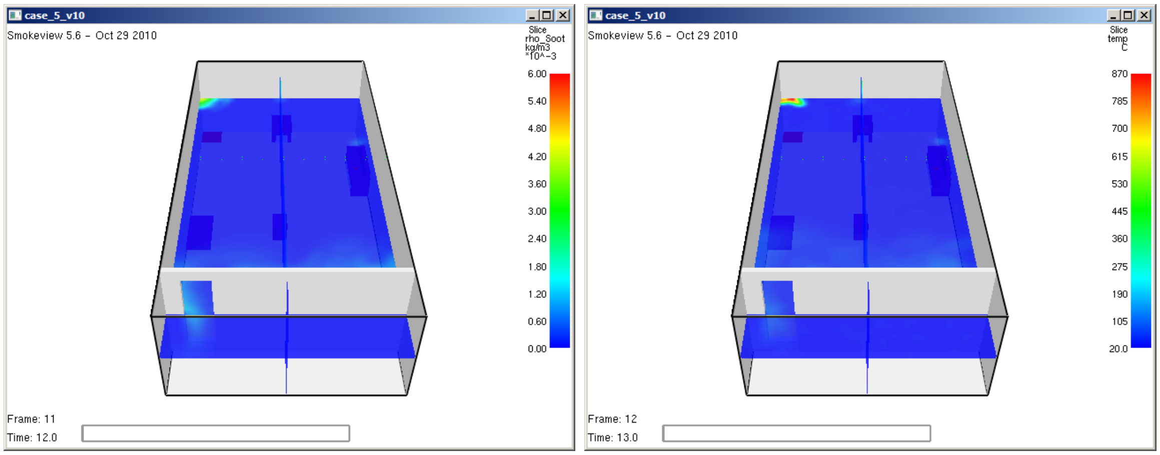
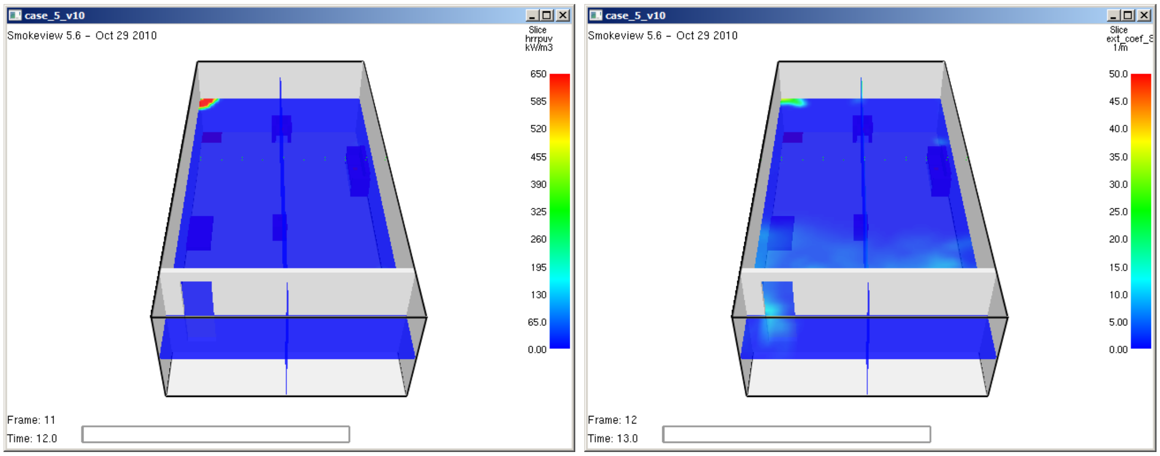


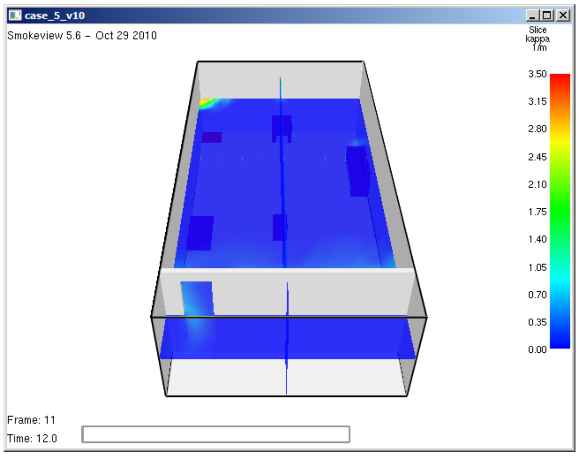
FIG. 4.1. Visualization of the room at the 12th second. The eye position and 'look at' position are shown. Unless indicated, the image is rendered with the view position settings. I flipped the image horizontally because Smokeview is right-handed and mine is left-handed. The fire in the image is based on a visualization of the HRRPUV data.



(a) HRRPUV (b) Soot Density



(c) Temperature(K) (d) Extinction Coefficient



(e) Absorption Coefficient

FIG. 4.2. Visualization of each data. All the data are projected onto one plane (y=0).

built in.

Algorithm 1: Precomputation

```
1 InitializeCameraSamples() ;
2 CreateLightSources() ;
3 ScatteringSamplesGeneration(numbers,g) ;
4 ComputeGradient(voxels) ;
```

Algorithm 1 Line 1 generates samples on the camera lens. This framework shoots rays from the eye, not from light sources because too many light sources make the backward tracing extremely slow. For improved image quality, I use best-candidate sampling with 15 samples per pixel. To pick a new sample, it chooses the farthest sample from all the previously created samples, resulting in a well distributed but irregular set of samples (Pharr & Humphreys 2010). All the samples are precomputed since it is slow to generate the samples.

Algorithm 1 Line 2 places multiple lights based on the data for each voxel. For technical details, see Section 3.3.1. Pseudo-code for this step is shown in Algorithm 2.

Section 3.4 describes Algorithm 1 Line 3 and 4. Their goal is to generate 3D vectors that will be used as multiple scattering directions in the rendering stage. The generated vectors will be rotated towards higher HRRPUVs or energy levels and used for light transport in the rendering stage. See Algorithm 3.

Algorithm 2: CreateLightSources

```
1 for Every Voxel in Volume do
2     color = Compute Planckian Locus ;
3     power = Compute Stefan-Boltzmann or HRRPUV ;
4     if Voxel is lit then
5         Save color and power ;
6     end
7 end
```

As explained in Section 3.3.1, Algorithm 2 Line 2 uses Equation 3.10, computing the color and radiant power based on physical laws as explained in Section 3.3.1. Although a blackbody is known to emit light even below 1,000 K, voxels are lit only when their temperature is above 1,000 K based on the Planckian Locus.

Algorithm 3: ComputeGradient

```
1 for Every Voxel in Volume do
2     for Every adjacent voxel of the voxel do
3         Compute Gradients based on HRRPUV ;
4     end
5     Sum the vectors weighted with their gradient ;
6     Store the vector ;
7 end
```

The details for Algorithm 3 are in Section 3.4. Line 3 uses Equation 3.11, computing and storing a gradient vector of the HRRPUV for each voxel.

Algorithm 4: Rendering

```
1  for Every sample ray for the camera image plane do
2      (start point, end point) ← Intersect eye ray with Volume ;
3      samples ←Create Samples() ;
4      while scattering point ← start point to end point do
5          $L_e$ ← Compute Emittance() ;
6          lights ← Pick Some Lights() ;
7          $L_S$ ← Light Transport(samples, lights) ;
8          if multiple scattering option is on then
9              $samples_M$ ← Generate Multiple Scattering Samples() ;
10             foreach sample $i$ in $samples_M$ do
11                 $L_m$ ← $L_m$ + Rendering($i$) ;
12             end
13         end
14         L ← ($L_S$ + $L_m$) × stepSize ;
15         // If Voxelization option is on, store the $L$.
16     end
17 end
```

Algorithm 4 incorporates the multiple scattering equation 3.6, or the single scattering equation 3.5 when used without Line 8, 9 and 11. The $\sum_{l}^{allNLights} L_{un}(x, \omega_l', \omega)$ part of Equation 3.6 is computed in Line 7. $\sum_{i=1}^{M} L_{sc}(x, \omega)$ is computed in Line 8, 9 and 11. $L_n(x + \omega\Delta x, \omega)$ is handled by the recursive structure of the Algorithm. I limit the depth of reflectance to only 2 for multiple scattering.

Algorithm 4 Line 2 gets the starting point and end point for the ray marching through the volume. Line 3 creates random floating points ranging from 0.0 to 1.0. The number of samples is determined by $(end\,point - start\,point)/(ray\ marching\ step\ size)$. Line

6 has two ways of picking lights. For details, see Section 3.3.2. The first method picks a lit voxel based on the distance between scattering points and the lit voxel center position. Second, it can choose a lit voxel completely randomly in well distributed ways using some well-known pseudo sequences. Based on the picked lights, the eye ray goes through the volume, computing scattering light transport and unscattered light transport at each sampling point until it gets out of the volume.

Please see Algorithm 5 for Line 7. Line 9 determines multiple scattering directions based on gradients of HRRPUV. For the detailed information, see Section 3.4. Line 11 does multiple scattering computation. The function calls itself since it is the same structure but each multiple scattering incoming radiance is weighted based on the PDF by Sobol samples and before Line 15, L_m is divided by the sum of the probability of all the samples as computed by equation 3.20.

Algorithm 5: Light Transport

```
1 Determine Light Position ;
2 L = Light Intensity ;
3 $\tau$ = Determine Extinction ;
4 $L$ = L $\times$ $\tau$ ;
  Result: Return Light Transport L
```

Algorithm 5 Line 1 determines light positions based on light placement schemes described in Section 3.3.3. In Line 2, light intensity is determined based on the Stefan-Boltzmann law or HRRPUV as stated in Section 3.3.1. Line 3 uses Equation 3.2 and Equation 3.3 to compute how much extinction there is per ray-marching. It looks up the absorption(σ_a) and scattering coefficient(σ_s) determined by FDS5 for every scattering position along with the ray until it intersects the light position. Then it computes $T = \sum(\sigma_s + \sigma_a) \times Soot\ Density$. Finally, The extinction is computed as: $\tau = \frac{e^T}{distance}$ where $distance$ is the distance between the light position and the current scattering position.

Please note that it can occur that the extinction coefficient is smaller than the absorption coefficient because FDS5 computes each separately. Thus, it causes the scattering coefficient to be a negative number if Equation 3.3 is applied. If the scattering coefficient is negative, I set it to 0 so it would not cause negative light power.

4.1.1 Voxelization

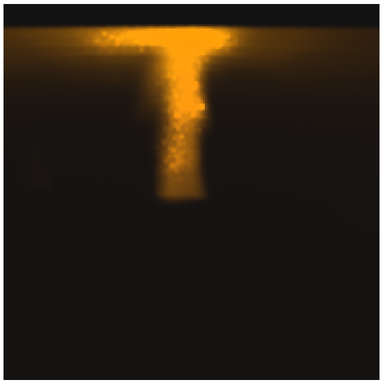
The voxelization scheme stores data when processing light transport so it can be dynamically used for the same computation at a later time. Theoretically, voxelization speeds up the rendering time significantly. However, for this particular fire smoke, voxelization is not effective. I compare the rendering execution time below. Please see Table 5.3. The voxel structure to store the light transport is 81x41x21.

Chapter 5

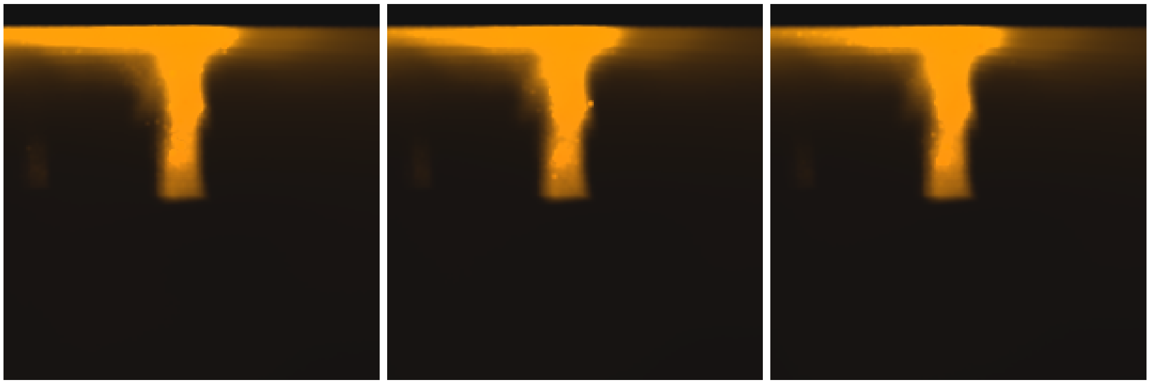
RESULTS

This chapter shows some examples of performance and visual results for the fire smoke rendering.

For **SCATTERING**, Figure 5.1 shows images generated from single and multiple scattering methods described in Section 3.2. See Table 5.1 for the execution time comparison. See Figure 5.2 for the graph of the table. All the performance tables were measured on a computer with i7 CPU Q740 1.73GHz and 4 GB memory. Note that the rendering time increases linearly based on the number of multiple scattering directions because I stopped the scattering after two events. If I had not stopped after a couple of scattering events, it would show an exponential increase. The volume data was in a grid containing 81x41x21 voxels. Each voxel has density, absorption/extinction coefficients and a scattering coefficient, which takes about 2 Mb. For multiple scattering, each voxel also contains gradient information for the multiple scattering direction sampling which also takes about 2 Mb. The precomputation stage takes 0.541 seconds for the volume. However, bigger volumes have longer precomputation stages. For example, it takes 7.975 seconds for a grid of 201x101x51. Every image in this thesis is rendered with a 81x41x21 simulation volume. All the images pick a random light based on a van der Corput sequence at each scattering point along with ray-marching. Multiple scattering directions are isotropic. Figures 5.1(b), 5.1(c) and 5.1(d) scatter to five, ten, and fifteen directions respectively. Tessendorf (1994b) shows multiple scattering has obvious effects on smoke edges, and it may be obvious at grazing angles. This figure shows some visual effects through edges generated by multiple scattering. Although not visible in the figures, there is a wall to the right of the fire as shown in Figure 4.1. This is the reason there is an abrupt fall-off to the right side of the fire in the figures.



(a) Single Scattering



(b) Multiple Scattering- 5 Scattering Directions (c) Multiple Scattering- 10 Scattering Directions (d) Multiple Scattering- 15 Scattering Directions

FIG. 5.1. Comparison between single and multiple scattering. Notice that the visual effects from multiple scattering are more obvious at the fire edges.

For **CLOSEST LIGHT APPROXIMATION**, I compare the results from picking

		Single Scattering	Multiple Scattering		
Number of Scattering Directions		N/A	5	10	15
Rendering Time (seconds)	Image 200x200	471.9	44046.4	75247.7	113934.9
	Image 100x100	107.1	11049.8	18541.5	28112.7
	Image 50x50	28.5	2920.2	4907.4	7437.4

Table 5.1. The rendering time comparison between single and multiple scattering.

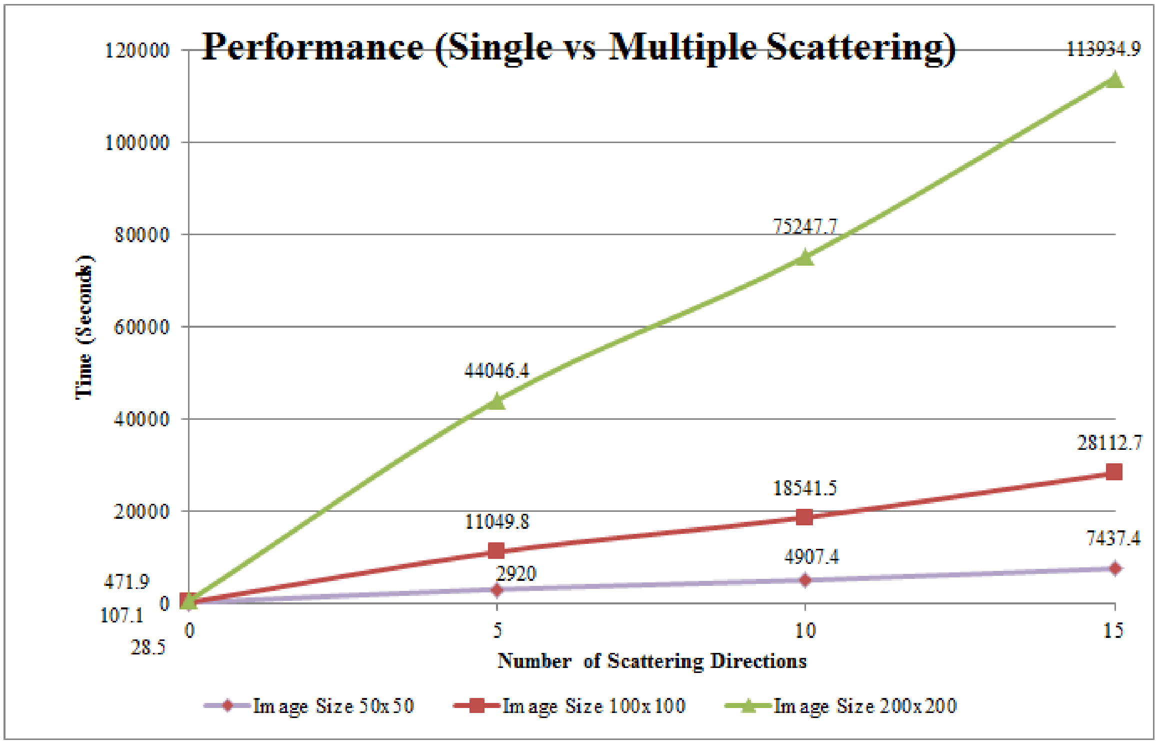


FIG. 5.2. Performance comparison for single and multiple scattering.

the closest light from each scattering point and Monte-Carlo light sampling described in Section 3.3.3. See Table 5.2 for the rendering time comparison. The number of light samples is the average number of lights sampled per ray used for the whole image. See Figure 5.3 for the resulting image comparison. Figure 5.3(f) and Table 5.2 show that the closest light approximation method displays the shape of the fire approximately, but it is faster than the Monte-carlo method. For Table 5.2, I used only one ray per pixel while Figure 5.3 uses 15 rays per pixel and an image resolution of 200x200.

Regarding the Monte-Carlo light sampling method, I used a van der Corput sequence as an index to pick a light. For Figures 5.3(c), 5.3(d), 5.3(e), and 5.3(f), the ray-marching step size varies for picking 531.2156, 318.9267, 16.4373, and 8.4922 lights per ray on average respectively. As shown in Table 5.2, Figure 5.3(d) is slow to render. Figure 5.3(e) is fast to render, but the result is very noisy. Figure 5.3(f) is fast and approximates the computation by picking one closest voxel and looking for lights in it. For Figure 5.3(f), I used 3% of the energy used for Figures 5.3(b) to 5.3(e) because Figures 5.3(b), 5.3(c), 5.3(d), and 5.3(e) samples a random light, while Figure 5.3(f) sample the closest light to the scattering point. Thus incoming radiance for Figure 5.3(f) is stronger, making it necessary to normalize the radiance so that $\sum L_{MC} = c \sum L_{CL}$, where L_{MC} and L_{CL} are incoming radiance from Monte-Carlo light sampling and the closest light sampling respectively, and c is a normalization factor. By experiment, the average incoming radiance per ray for Monte-Carlo light sampling and the closest light sampling are 87.009289 and 2.748891 respectively. Thus $\dfrac{2.748891}{87.009289} = 0.0315$, or a correction factor of 3%. Figure 5.3(b) and 5.3(c) have the same settings but Figure 5.3(b) uses a grey background behind the smoke so that the black smoke is more obvious. It shows that light extinction due to scattering and absorption is present through the top of the smoke. The top part of all of the images intersects with no volume so it just display the background colors. Figure 5.3(a) is a reference image. I used the Monte-carlo light sampling method and it was generated with

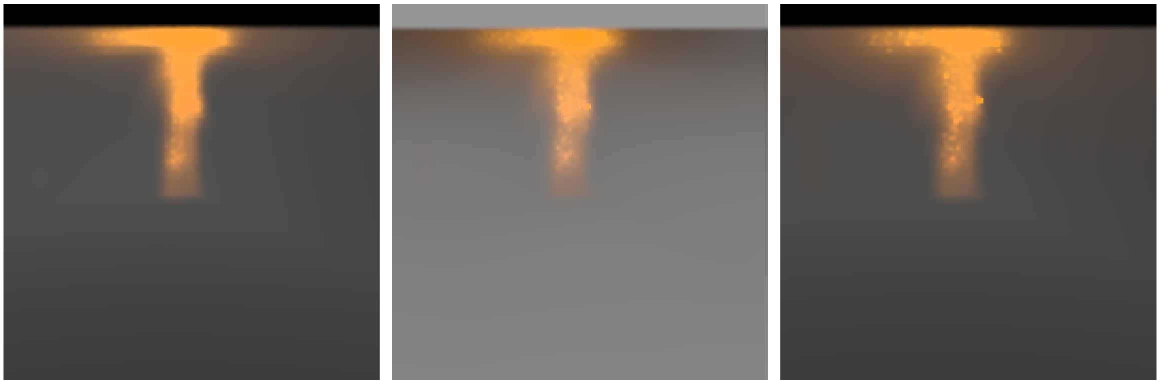
1250 times more light samples per ray, converging without residual noise.

	Monte Carlo light samples per ray		Closest Light
Average Light Samples	318.9267	16.4373	2.222678
Rendering Time(seconds)	5470.4	765.5	380.9

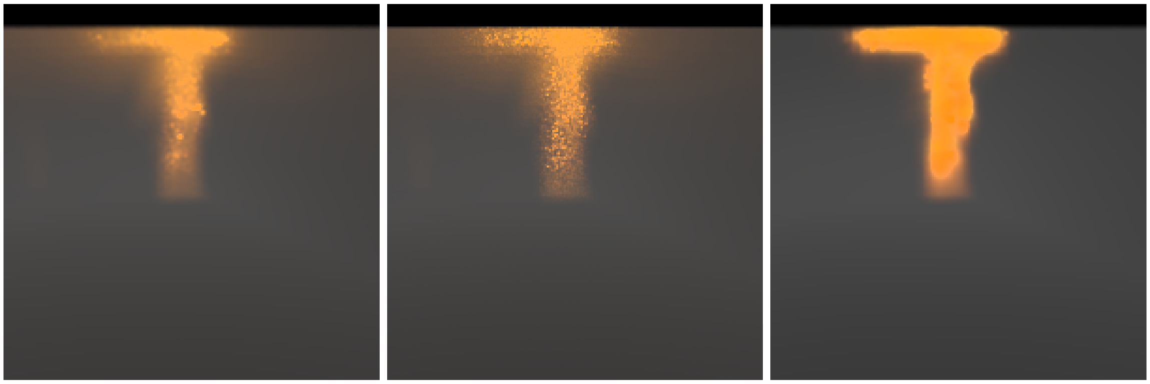
Table 5.2. The rendering time comparison between the Monte-Carlo light sampling and closest light sampling.

For **COLOR DETERMINATION**, Nguyen et al. (2002) used Planck's law. I have tried Planck's law and the Planckian Locus and compared the result in Figure 5.4. For Planck's law, I considered only from wavelengths of 380 nm to 780 nm. The width of each wavelength band is five nm and the values for each bandwidth band are interpolated. Planck's law has been previously used (Nguyen, Fedkiw, & Jensen 2002). However, the figure clearly shows that the Planckian Locus gets a very similar color compared to Planck's law. The Planckian Locus is about 33 times faster to compute than Planck's law. Because the Planckian Locus is an approximation for Planck's law, I use Figure 5.4(a) as the reference image to compute the RMS error value.

For **LIGHT PLACEMENT**, I described three ways of sampling light positions for each voxel: *point*, *spherical*, and *voxel sampling*. Figure 5.5 compares these results. Monte-Carlo integration estimates the integral by averaging a number of samples. Uniformity in the sampling distribution can drastically affect how fast the Monte-Carlo approximation converges to the correct result. To highlight the differences between the sampling strategies, Figure 5.5 uses only one ray and one scattering event per pixel, instead of the many that should be averaged together to converge to a solution. From these images, we see that Figure 5.5(a) has grid artifacts that will slow the Monte-Carlo convergence, while both Figure 5.5(b) and 5.5(c) are better distributed. This shows that the stratefied and spherical sampling methods are better than the point sampling method. However, I used the stratified sampling method because the spherical sampling method can be inefficient due to its

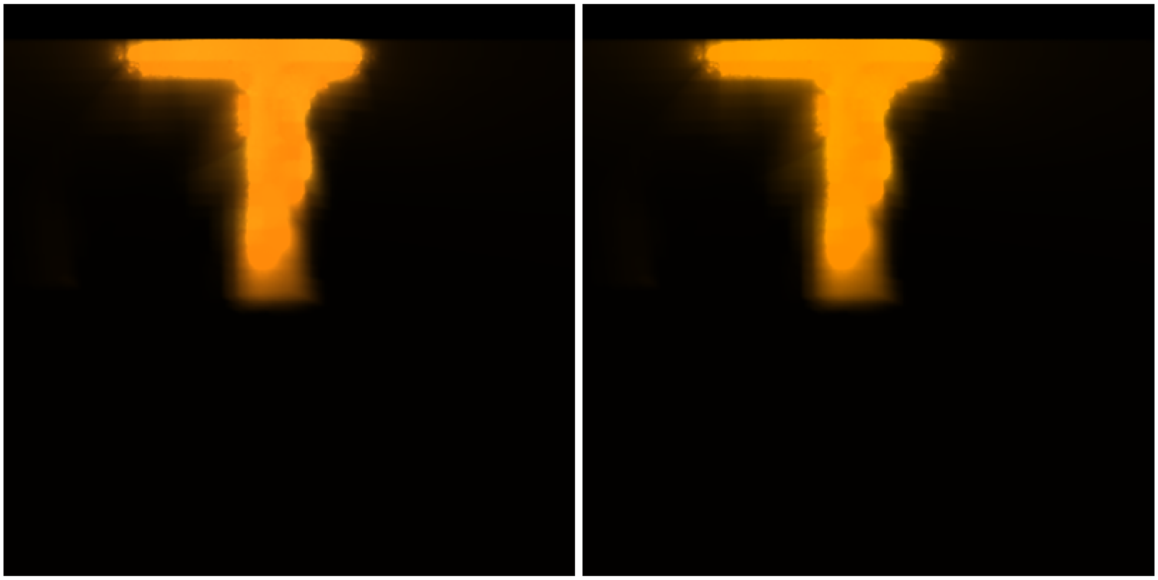


(a) Reference (b) 531.2156 light samples (c) 531.2156 light samples



(d) 318.9267 light samples (e) 16.4373 light samples (f) Closest Light

FIG. 5.3. Image comparison based on different light sampling methods and numbers of lights. The RMS values of Figures 5.3(c), 5.3(d), 5.3(e), and 5.3(f) against Figure 5.3(a) are 3.1605, 3.1884, 3.5112, 5.2350. These are computed with the RGB components and averaged. The maximum value for each channel is 255.

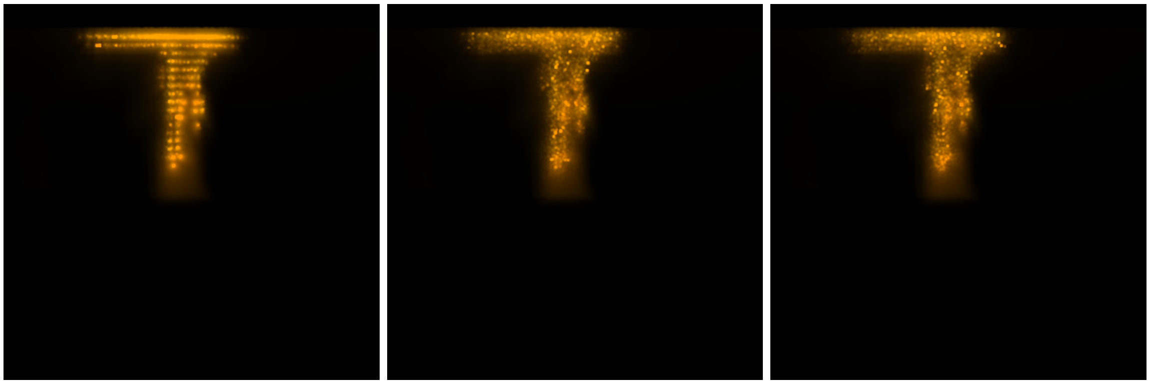


(a) Planck's Law (b) Planckian Locus

FIG. 5.4. The colors are generated by Planck's law and the Planckian Locus respectively. Note that there is no noticeable difference in the colors. The RMS error value of Figure 5.4(b) against Figure 5.4(a) is 2.1829.

rejection scheme.

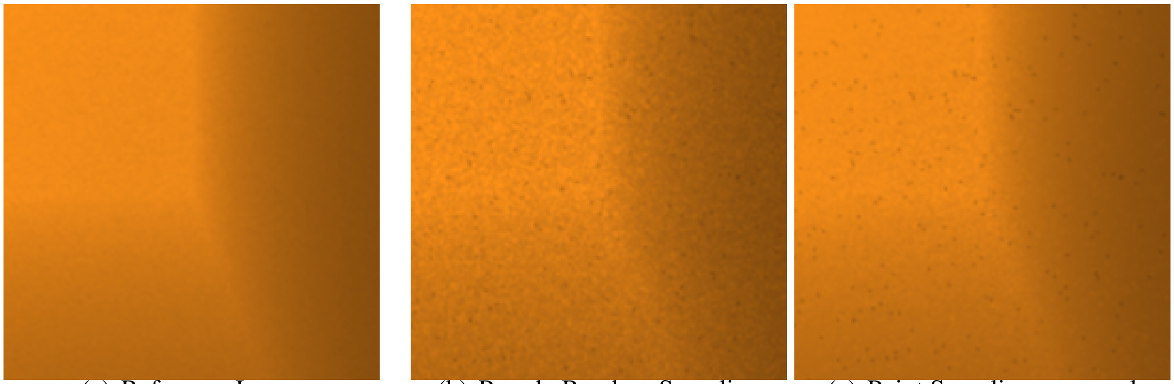
Figure 5.6 compares the Monte-Carlo result for each method to a reference image in 5.6(a). All images are of a close view of the same section of fire. The reference image was generated with 15 times more ray samples per pixel, a step size 20 times denser, and at an image resolution 16 times greater. For quantitative comparisons, the reference image is resized to match the test images with Lanczos resampling. By using 4800 times more samples in the Monte-Carlo integration (15*20*16), the rendering has reached total convergence with no residual noise. Figure 5.6(c) looks smooth because multiple pixels sample light from one position while the other images sample lights from multiple positions. However, the point sampling method causes grid artifacts when seen from a large distance. Based on these results, the best choice is using the closest light at each step, with light positions chosen using voxel sampling.



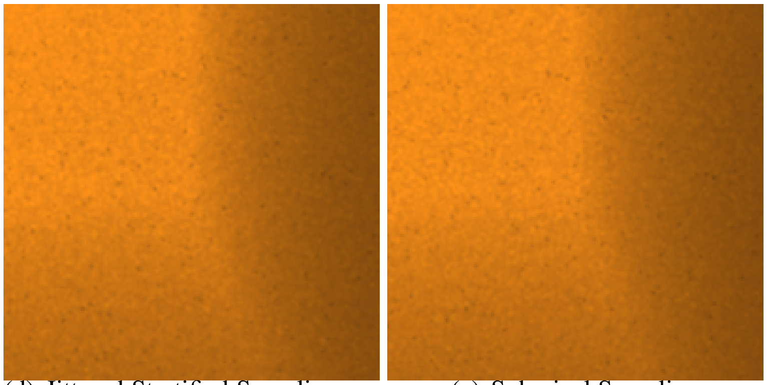
(a) Point Lights (b) Spherical Lights (c) Voxel Lights

FIG. 5.5. Comparison of resulting images between different light placement sampling. Image 5.5(a) looks very regular due to a single point light per voxel while the others don't. Please note that the noise in each image is not the limitation of these techniques, but it comes from a very small number of samples.

For **MULTIPLE SCATTERING DIRECTION SAMPLING**, I compare the results between the different scattering direction sampling based on the Sobol and Henyey-Greenstein functions. For all the images in Figure 5.7, Monte-Carlo light sampling is used without the closest light approximation scheme. Figure 5.7(a) uses 100 times more light samples and 15 isotropic multiple scattering direction samples to have little residual noise as a reference image. Figure 5.7(b) uses only five isotropic directions for multiple scattering and 5.7(c) uses five anisotropic directions based on the HRRPUV gradient and combines in accordance with the importance sampling scheme described in Section 3.4. Figure 5.7(d) uses a variable number of scattering directions proportional to HRRPUV gradient at the scattering point. It increases one scattering direction per HRRPUV increase of 100 $\frac{kW}{m^3}$ up to ten scattering directions. The parameter g of the Henyey-Greenstein phase function determines the anisotropy of scattering directions. If g is close to 1.0, then it is forward scattering and it is isotropic at 0.0. For Figure 5.7(a), 5.7(b), 5.7(c) and 5.7(d), g is 0.01, 0.01, 0.64 and 0.01 respectively. I present two methods for adjusting multiple scattering



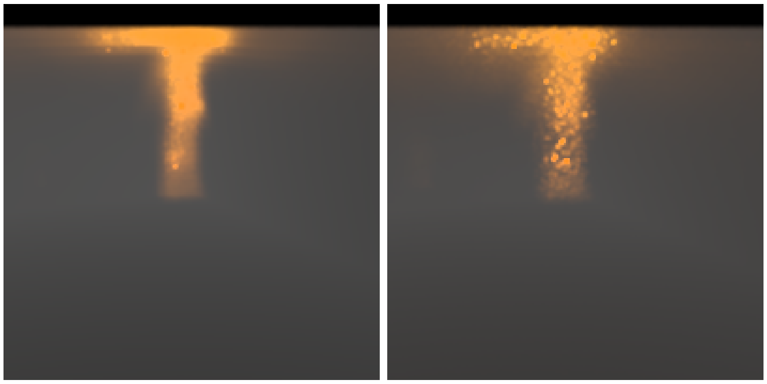
(a) Reference Image (b) Pseudo Random Sampling (c) Point Sampling per voxel



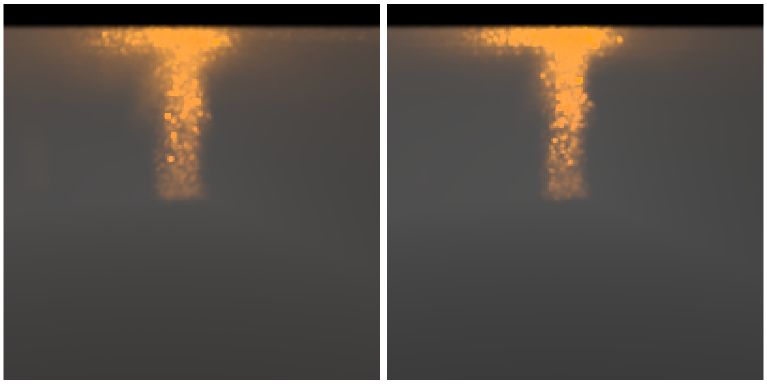
(d) Jittered Stratified Sampling per voxel (e) Spherical Sampling

FIG. 5.6. Comparison between different light placement sampling. Notice the difference in noise in each of the images. In comparison to the reference image, the RMS error values of Figure 5.6(b), 5.6(c), 5.6(d) and 5.6(e) against Figure 5.6(a) are 2.5290, 1.8379, 2.4212, and 2.5163 respectively.

directions. One of the two methods, turning multiple scattering directions towards the fire, does not seem very effective. However, the other method, increasing the number of multiple scattering directions based on the HRRPUV gradient value, can result in a comparable image in less time. It only took 184.2 seconds to render Figure 5.7(d) while Figure 5.7(b) took 919.0 seconds. Note that although Figure 5.7(d) took a shorter time than Figure 5.7(b), Figure 5.7(d) has a lower RMS error value.



(a) Reference(Isotropy) (b) Isotropic Directions

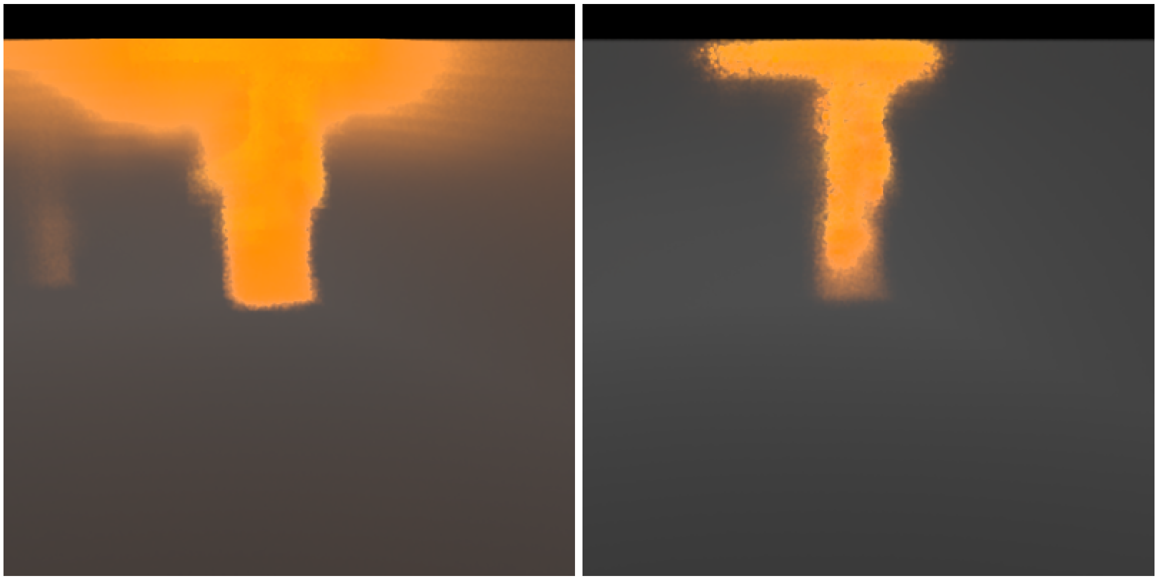


(c) Anisotropic Directions (d) Variable Scattering

FIG. 5.7. Image comparison between isotropic and anisotropic multiple scattering directions. Image 5.7(d) uses the variable scattering direction method. The RMS error values of Figures 5.7(b), 5.7(c), and 5.7(d) against Figure 5.7(a) are 3.1090, 2.7658, and 1.8648 respectively.

For **FIRE POWER GENERATION**, I used both the Stefan-Boltzmann law and

HRRPUV and compared the results. The method is described in Section 3.3.1. Please see Figure 5.8 for the comparison. Figure 5.8(a) uses the Stefan-Boltzmann law as the light intensity. Figure 5.8(b) uses HRRPUV instead. The power of Figure 5.8(b) is stronger because the Stefan-Boltzmann law assumes that the emissivity of the fire smoke is 1.0 while the smoke away from the fire in Figure 5.8(b) has lower emissivity. Another reason is that the Stefan-Boltzmann law expresses the sum of the power at all frequencies. Considering non-visible wavelengths adds extra power to the image. Based on this result, using HRRPUV is the better choice for fire light intensity.



(a) Stefan Boltmann

(b) HRRPUV

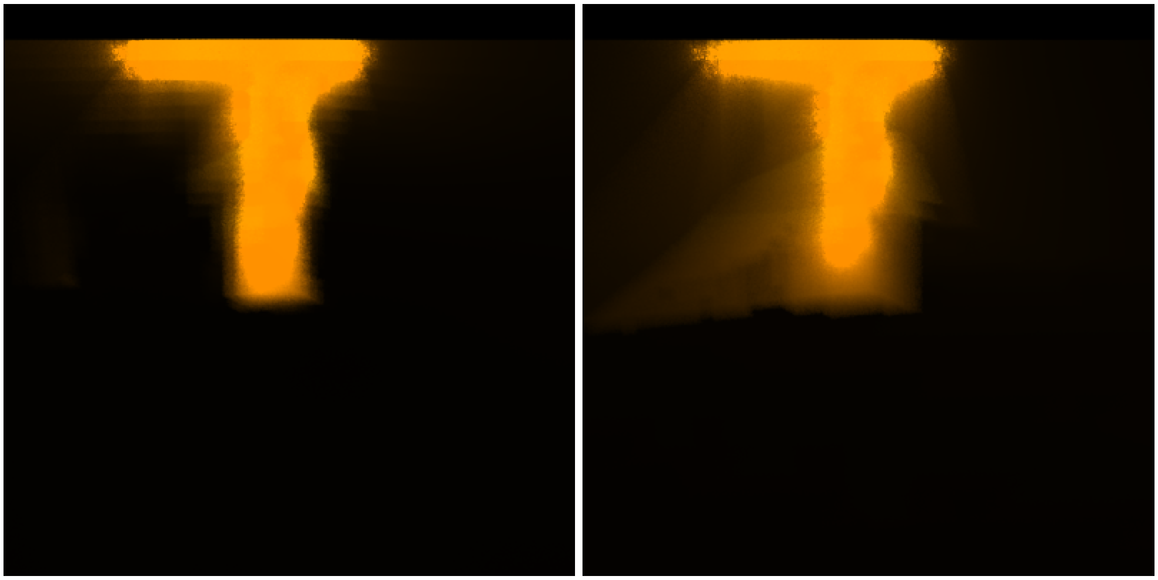
FIG. 5.8. Comparison between the methods based on the Stefan-Boltzmann law and HRRPUV.

For **VOXELIZATION**, I store light transport data in each voxel during the tracing of one ray for use when tracing other rays. The timings shown in Table 5.3 are not very different using voxelization data storage or not because of its low hit rate. I simulated with a data structure for a grid of 81x41x21 and 350 light sources. I simulated accessing 10,000,000 random voxels, which takes 37.98 seconds. For the multiple scattering simulation in Table 5.3, the number of times the simulation accesses the light transport data or stores it is about 26,400,000. Thus $\frac{26,400,000}{10,000,000} \times 37.98 = 100.2672\ seconds$. The access time does not take a significant portion of the rendering time, however, the hit rate is only 8.04%. This means only 8.04% of the times the method utilizes the precomputed data, speeding up the rendering time insignificantly. The low hit rate is because the voxelization method can only utilize the precomputed data if the same light is sampled, and the same light is not likely sampled again due to the many light sources.

	Voxelization Off	Voxelization On	Speed Up
Single Scattering	510.7 s	498.6 s	1.0242679
Multiple Scattering	1792.4 s	1702.4 s	1.0528665

Table 5.3. Rendering time comparison with voxelization and without it.

FDS5 gives data for both **EXTINCTION and ABSORPTION COEFFICIENT**. Because Mulholland and Choi (1998) state that the scattering coefficient for smoke is *nearly universal*, and that the scattering coefficients for the acetylene smoke are $7.80 \pm 0.08\frac{m^2}{g}$, I used a constant as a scattering coefficient to show the difference from the data given by FDS5. Figure 5.9 is the comparison between constant material properties and the material properties by FDS5. For Figure 5.9(a), I converted $7.80\frac{m^2}{g}$ to match the units of FDS5. For Figure 5.9(a), I used the scattering coefficients from FDS5. Figure 5.9 shows that the two images do not look very different near the fire while they are different away from the fire. It is because Figure 5.9(a) has high scattering coefficients far from the fire. For Figure 5.9(b), the scattering coefficients tend to be 0 far from the fire. This shows that the scattering coefficient based on the FDS5 is not *nearly universal* outside the flame area, and that using a constant scattering coefficient may not achieve a physically correct solution.



(a) Constant Coefficients (b) Variable Coefficients

FIG. 5.9. Comparison between constant and variable material coefficients.

Chapter 6

CONCLUSION

I have demonstrated physically based rendering of fire smoke. For the rendering of fire smoke, the most basic, but important, thing is to have the right properties for the smoke to render. There are five contributions in this thesis. First, the colors of blackbody radiation can be physically determined by the Planckian Locus, not only by Planck's law. Previously, Planck's law had been used, but the implementation of the Planckian Locus is simpler and produces comparable results faster. For the blackbody radiation power, HRRPUV can be used, but the Stefan-Boltzmann law should not be used. Second, the combination of selecting a fire light source for faster execution and approximating the light positions for the fire light source can be used. Selecting the closest fire area as a light source can result in a faster execution time. The blackbody radiation for the chosen area can be represented with point, voxel, and spherical sampling. Previously, a simple Monte-Carlo position sampling had been used. Third, multiple scattering directions can be determined based on the HRRPUV gradients to make the computation converge faster. Increasing the number of multiple scattering directions based on the magnitude of the HRRPUV gradients can significantly decrease the amount of the computation. Sampling more towards a higher HRRPUV area can lessen noise very slightly, but it does not affect the amount of the computation. Fourth, multiple scattering can create visual differences for fire smoke rendering, but multiple scattering is significantly slower to converge than single scattering. Fifth, I show that using the same scattering coefficient for fire smoke may not work for the rendering of smoke far from the fire.

6.1 Future Work

6.1.1 Closest Light Approximation

I rendered some images based on the ’closest light approximation’ scheme. Instead, the Wentzel-Kramer-Brillouin (WKB) approximation of the path integral for light transport could be used to determine which light to use to render at every scattering position. The WKB approximation is a multiply-scattered phase function. Premoze (2004) uses this for a blur width function, and a similar function could be used for fire smoke. It would be necessary to compute the blur width function for every voxel for every light in the pre-computation stage. Then each voxel could store the light source that has the highest power based on the function result. This would need to be computed only once, then the light stored in each voxel could continue to be used in the rendering stage. This would most likely lessen the whole simulation time because it would be able to pick light samples that likely contribute to the image at every ray-marching point and thus its solution would converge to the right solution quickly. However, the precomputation for this scheme is heavy because the blur function would need to be computed for $voxel\,dimensions \times number\,of\,lights$. For instance, a voxel grid of 81x41x21 and 860 light sources would need to compute the blur function 59,977,260 times ($860 \times 81 \times 41 \times 21$.). If this turns out to be too computationally expensive, we might just use a simple distance function combined with the light intensity instead of the blur function. Specifically, we can use Algorithm 6. Line 7 traverses the L value for each light for a voxel and picks the strongest light.

Algorithm 6: This computes light attenuation from every light for every voxel only based on distance and light intensity. This picks one light to render based on the attenuated light powers from every light source in the volume.

```
for Every Voxel do
    for Every Light do
        distance ← Distance(lightPosition, voxelPosition);
        L = $\frac{LightIntensity}{distance^2}$ ;
    end
    for Every Voxel do
        lightIndex ← Pick the Strongest Light(L) ;
    end
end
Result: Return one strongest light for each voxel
```

Appendix A

ALTERNATIVE APPROACHES

A.1 Overview

In addition to the method described in this thesis, I implemented two additional multiple scattering methods, which are based on Premoze et al. (2004) and Pauly et al. (2000) respectively.

My implementation of Premoze et al.'s work is the same as theirs except with different scattering angles. It works for homogeneous volume, but the attenuation based on fire smoke properties seems incorrect in my experiment. I suspect that it may be because the WKB approximation they incorporated is designed for homogeneous volume since Premoze et al. (2003) state Tessendorf's rigorous mathematical (1991) derivations assume homogeneous volume, and his method is often incorrect for varying scattering and absorption coefficients. Fire smoke is a very inhomogeneous volume situation with varying coefficients. See Figure A.1 for a rendered image from this method. Figure A.1 has a homogeneous volume. This figure shows that this method works for a homogeneous volume.

The computation of the other method, based on Metropolis light transport by Veach et al. (1997), is far slower than the final method presented in this thesis, making it infeasible for application in fire smoke rendering. It takes a few hours just to compute light transport for an image of size 200x200 with only one sample per pixel. Figure A.2 shows some images generated by this method. Figure A.2(a) and A.2(b) show the volume of a buckyball shape with normal fog properties and the blackbody properties. For the blackbody properties, I used extremely low scattering coefficients and high absorption coefficients. I discuss the method and provide some analysis on why this method is inefficient for fire smoke rendering.

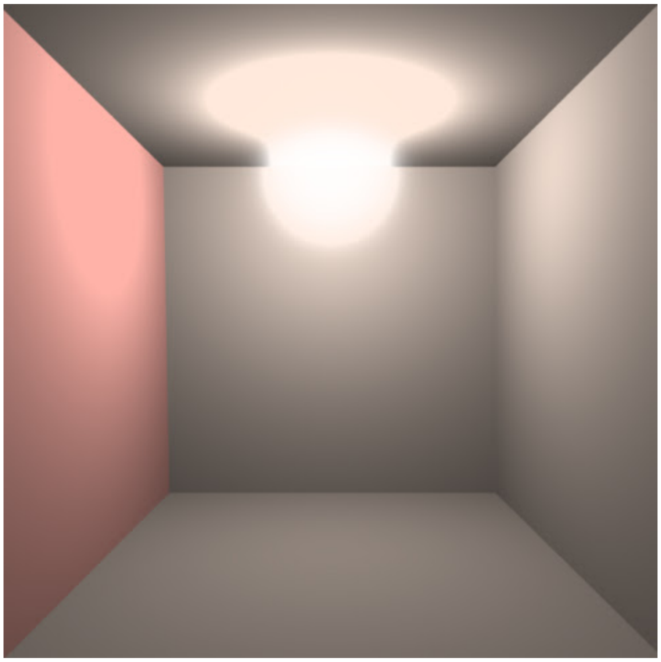
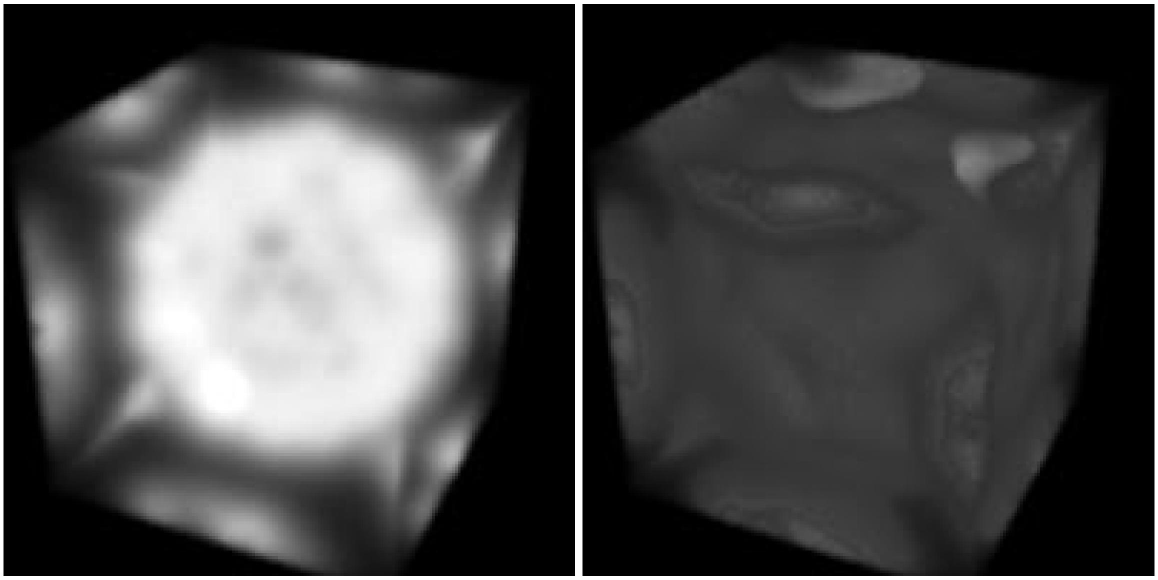


FIG. A.1. The figure is rendered with Premoze et al.'s method (2004). This has homogeneous volume.

A.1.1 Metropolis Light Transport Based Method

My implementation is based on bidirectional tracing, as illustrated in Figure A.3. Rays are moved forward to reach the next scattering point based on the small step method by Kelemen (2002). After multiple steps, it exchanges energy at each scattering point based on the previous works (Lafortune & Willems 1993; Rushmeier 1988). In the figure, yellow lines are exchanges of radiance between scattering points. Rays start from both the eye and



(a) Normal Fog Property (b) Blackbody Property

FIG. A.2. Buckyball volume rendered by the metropolis-based method. There are three point-lights at the three densest points in the buckyball volume. The volume for Figure A.2(a) has typical properties of fog. For Figure A.2(b), the volume properties are similar to blackbody.

the light. The time complexity of the computation for the energy exchange between two rays is $O(N^2)$ where N is the number of steps per ray. In my implementation, N is 20. It starts a ray from the eye and the lights. Rays move forward into 20 different directions at every ray-marching step in volume. Then each scattered ray moves forward and scatters again. The computation time complexity is $O(K^N)$ where N is the number of scattering events per ray and K is the number of scattering directions per event. See Figure A.4 for the illustration. Whenever a ray is mutated in a volume, it scatters in a fixed number of directions. The scattering directions are forward. Each scattered ray will move forward, mutate, and repeat the same process. This continues until every ray gets out of the volume. The length of a step size is determined by $density \times random\, number$ at the scattering point. The higher the density is, the smaller the step size is because it is more likely that scattering occurs in a dense volume. It also mutates its ray direction based on the mutation coefficient, proportional to the density. I modified the mutation strategy by Kelemen et al. (2002) for this. See Figure A.5 for the illustration. This illustration shows that a ray takes a smaller step in a less dense volume. This also means that more scattering events happen in a dense volume, resulting in heavy computation. Since fire smoke has high extinction coefficients by nature, this method is ineffective due to its small step size and thus much computation. Since this method was ultimately ineffective, the mutation strategy details are not given here to focus on the inefficiency of the Metropolis based scheme for rendering of fire smoke.

A.1.2 Analysis

First, the computation is prohibitively expensive due to the considerable number of rays generated and the light transport for each ray. Each ray generated by scattering tends to be very short for fire smoke, because the soot density of the fire smoke tends to be high and the span of a ray is determined by the soot density at the scattering point. Short rays

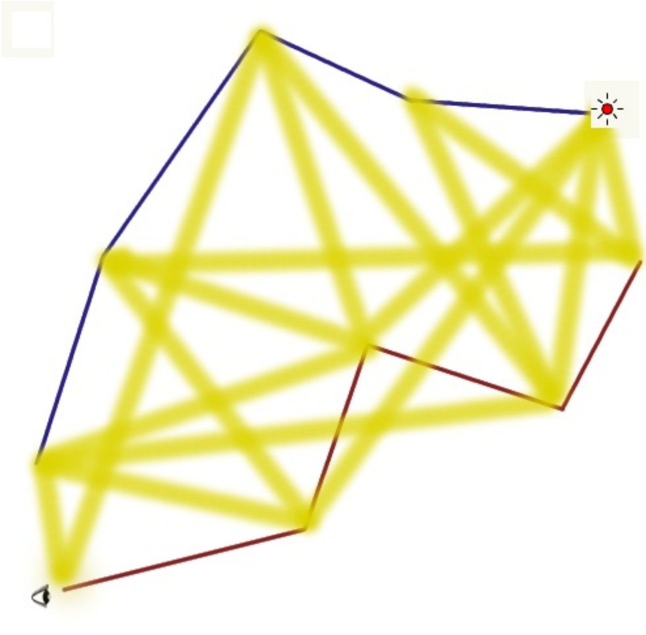


FIG. A.3. Illustration of bidirectional tracing for volume (Lafortune & Willems 1993; Rushmeier 1988).

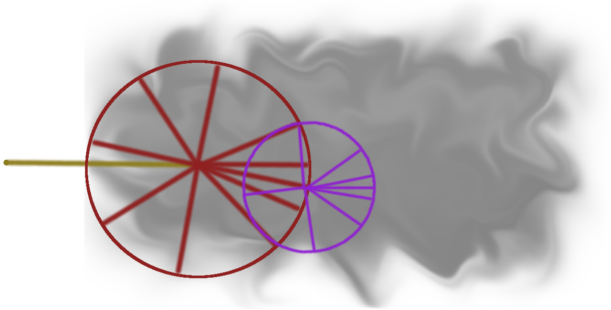


FIG. A.4. Scattering directions are illustrated around each scattering point over the circle. The scattering angles are anisotropic. Each scattered ray will scatter again after a ray step forward.

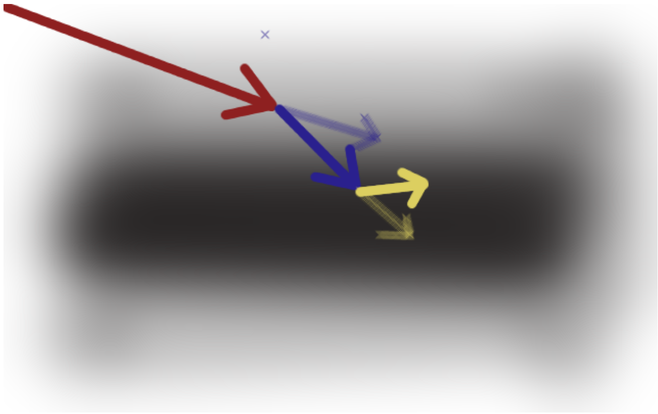


FIG. A.5. Illustration of my modified mutation strategy (Kelemen *et al.* 2002). The length of the blue ray in a denser area is shorter than that of the red one. The yellow ray is shorter than the blue one. Un-mutated directions are in transparent colors while mutated directions are in opaque colors. The yellow ray mutates more in a denser area than the blue one.

cause numerous scattering events since they require many steps before getting out of a volume. Also, radiance from light transport for a short ray is small as well, making its contribution to the whole illumination very small.

Second, this algorithm cannot take advantage of the Neumann series. Equation 3.5 can be turned into the Neumann series (Kajiya 1986).

$$\begin{aligned} I &= (1 - gM)^{g\varepsilon} \\ &= g\varepsilon + gMg\varepsilon + gMgMg\varepsilon + g(Mg)^3\varepsilon + \cdots \end{aligned} \tag{A.1}$$

The first term is direct emittance, the second is scattered once, and the third is scattered twice, etc. This form is useful because direct illumination contributes the most, and once-scattered illumination contributes the second most, etc. That means that it can converge to a correct solution quickly after computing just direct emittance and radiance from once-scattered light photons. Then the image may be accurate enough that the computation can stop. This Metropolis-based multiple scattering method cannot do the same. This can obtain direct emittance after computing all the multiply scattered light transport. In Figure A.4, one of the red rays may head directly towards an eye. Likewise, one of the scattered purple rays may head directly towards an eye. But light transport for the rays is not computed first.

REFERENCES

[1] Arvo, J. 1993. Transfer functions in global illumination. *Siggraph Course Notes* 1–28.

[2] Ashikhmin, Premoze, R. R. N. S. 2004. Blurring of light due to multiple scattering by the medium: a path integral approach. *Technical report CUCS-017-04*.

[3] Baran, I.; Chen, J.; Ragan-Kelley, J.; Durand, F.; and Lehtinen, J. 2010. A hierarchical volumetric shadow algorithm for single scattering. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA 2010, 178:1–178:10. New York, NY, USA: ACM.

[4] Bernabei, D.; Ganovelli, F.; Pietroni, N.; Cignoni, P.; Pattanaik, S.; and Scopigno, R. 2010. Real-time single scattering inside inhomogeneous materials. *The Visual Computer (special issue of CGI 2010)* 26(6-9):583–593. DOI: 10.1007/s00371-010-0449-7.

[5] Billeter, M.; Sintorn, E.; and Assarsson, U. 2012. Real-time multiple scattering using light propagation volumes. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D 2012, 119–126. New York, NY, USA: ACM.

[6] Bruneton, E., and Neyret, F. 2008. Precomputed atmospheric scattering. *Comput. Graph. Forum* 27(4):1079–1086. Special Issue: Proceedings of the 19th Eurographics Symposium on Rendering 2008.

[7] Chandrasekhar, S. 1960. *Radiative Transfer*. Dover Books on Physics Series. Dover Publication.

[8] Chen, J.; Baran, I.; Durand, F.; and Jarosz, W. 2011. Real-time volumetric shadows using 1D min-max Mipmaps. In *Symposium on Interactive 3D Graphics and Games*, I3D 2011, 39–46. New York, NY, USA: ACM.

[9] Dobashi, Y.; Yamamoto, T.; and Nishita, T. 2002. Interactive rendering of atmospheric scattering effects using graphics hardware. In *Proceedings of the ACM Siggraph/Eurographics Conference on Graphics Hardware*, HWWS 2002, 99–107. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

[10] Donner, C., and Jensen, H. W. 2005. Light diffusion in multi-layered translucent materials. 1032–1039. New York, NY, USA: ACM.

[11] Fedkiw, R.; Stam, J.; and Jensen, H. W. 2001. Visual simulation of smoke. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 2001, 15–22. New York, NY, USA: ACM.

[12] Gonis, A., and Butler, W. H. 1999. *Multiple Scattering in Solids*. Springer.

[13] Guo, P., and Wang, Y. 2012. A multiple scattering in participating media for real time rendering. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, 2259 –2264.

[14] Henyey, L. G., and Greenstein, J. L. 1941. Diffuse radiation in the galaxy. *The Astrophysical Journal* 93:70–83.

[15] Hu, W.; Dong, Z.; Ihrke, I.; Grosch, T.; Yuan, G.; and Seidel, H.-P. 2010. Interactive volume caustics in single-scattering media. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, I3D 2010, 109–117. New York, NY, USA: ACM.

[16] Incropera, F. P. 2006. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons.

[17] Inoshita, C.; Mukaigawa, Y.; Matsushita, Y.; and Yagi, Y. 2012. Shape from single scattering for translucent objects. In *Proceedings of the 12th European Conference on*

Computer Vision - Volume Part II, ECCV 2012, 371–384. Berlin, Heidelberg: Springer-Verlag.

[18] Jarosz, W.; Zwicker, M.; and Jensen, H. W. 2008. The beam radiance estimate for volumetric photon mapping. In *ACM Siggraph 2008*, 3:1–3:112. New York, NY, USA: ACM.

[19] Jensen, H. W.; Marschner, S. R.; Levoy, M.; and Hanrahan, P. 2001. A practical model for subsurface light transport. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH 2001, 511–518. New York, NY, USA: ACM.

[20] Jensen, H. W. 1996. Global illumination using photon maps. In *Proceedings of the Eurographics Workshop on Rendering Techniques 1996*, 21–30. London, UK, UK: Springer-Verlag.

[21] Kajiya, J. T., and Von Herzen, B. P. 1984. Ray tracing volume densities. In *Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 1984, 165–174. New York, NY, USA: ACM.

[22] Kajiya, J. T. 1986. The rendering equation. In *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 1986, 143–150. New York, NY, USA: ACM.

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

[24] Kelemen, C.; Szirmay-Kalos, L.; Antal, G.; and Csonka, F. 2002. A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum*, 531–540.

[25] Kim Y., Cho B., B. K. D. H. 2002. Design of advanced color - temperature control system for HDTV applications. In *Journal of the Korean Physical Society*, volume 41, 865–871.

[26] Krystek, M. 1985. An algorithm to calculate correlated colour temperature. *Color Research and Application* 10(1):38–40.

[27] Kuipers, N. 2005. Uniform distribution of sequences. 129,158.

[28] Kulla, C., and Fajardo, M. 2012. Importance sampling techniques for path tracing in participating media. *Comp. Graph. Forum* 31(4):1519–1528.

[29] Lafortune, E. P., and Willems, Y. D. 1993. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques(COMPUGRAPHICS 93)*, 145–153.

[30] Lafortune, E. P., and Willems, Y. D. 1996. Rendering participating media with bidirectional path tracing. In *Proceedings of the Eurographics Workshop on Rendering Techniques 1996*, 91–100. London, UK, UK: Springer-Verlag.

[31] Macadam, D. L. 1937. Projective transformations of I. C. I. color specifications. *Journal of the Optical Society of America* 27(8):294–297.

[32] Max, N. 1994. Efficient light propagation for multiple anisotropic volume scattering. In *In Proceedings of the 5th Eurographics Workshop on Rendering*, 87–104.

[33] McGrattan, K. B. 2007. Fire dynamics simulator (version 5), technical reference guide. *NIST Special Publication 1018-5.*

[34] Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; and Teller, E. 1953. Equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21:1087–1092.

[35] Moon, J. T., and Marschner, S. R. 2006. Simulating multiple scattering in hair using a photon mapping approach. In *ACM Siggraph 2006 Papers*, 1067–1074. New York, NY, USA: ACM.

[36] Moon, J. T.; Walter, B.; and Marschner, S. 2008. Efficient multiple scattering in hair using spherical harmonics. In *ACM Siggraph 2008 papers*, 31:1–31:7. New York, NY, USA: ACM.

[37] Mulholland, G., and Bryner, N. 1990. Laboratory studies of smoke properties at nist past, present and future. In *Smoke/Obscurants Symposium 14*, 671–684.

[38] Mulholland, G. W., and Choi, M. Y. 1998. Measurement of the mass specific extinction coefficient for acetylene and ethene smoke using the large agglomerate optics facility. In *Symposium (International) on Combustion 27th proceedings*, volume 1, 1515–1522.

[39] Nguyen, D. Q.; Fedkiw, R.; and Jensen, H. W. 2002. Physically based modeling and animation of fire. In *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, Siggraph 2002, 721–728. New York, NY, USA: ACM.

[40] Pauly, M.; Kollig, T.; and Keller, A. 2000. Metropolis light transport for participating media. In *Proceedings of the Eurographics Workshop on Rendering Techniques 2000*, 11–22. London, UK, UK: Springer-Verlag.

[41] Pharr, M., and Humphreys, G. 2010. *Physically Based Rendering, Second Edition:*

From Theory To Implementation. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2nd edition.

[42] Premože, S.; Ashikhmin, M.; and Shirley, P. 2003. Path integration for light transport in volumes. In *Proceedings of the 14th Eurographics Workshop on Rendering*, EGRW 2003, 52–63. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

[43] Premože, S.; Ashikhmin, M.; Tessendorf, J.; Ramamoorthi, R.; and Nayar, S. 2004. Practical rendering of multiple scattering effects in participating media. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, EGSR 2004, 363–374. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association.

[44] Rasmussen, N.; Nguyen, D. Q.; Geiger, W.; and Fedkiw, R. 2003. Smoke simulation for large scale phenomena. In *ACM Siggraph 2003 Papers*, 703–707. New York, NY, USA: ACM.

[45] Ren, Z.; Zhou, K.; Lin, S.; and Guo, B. 2008. Gradient-based interpolation and sampling for real-time rendering of inhomogeneous, single-scattering media. In *Computer Graphics Forum*, volume 27, 1945–1953.

[46] Rushmeier, H. E., and Torrance, K. E. 1987. The zonal method for calculating light intensities in the presence of a participating medium. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH 1987, 293–302. New York, NY, USA: ACM.

[47] Rushmeier, H. E. 1988. *Realistic image synthesis for scenes with radiatively participating media*. Ph.D. Dissertation, Ithaca, NY, USA. AAI8821226.

[48] Siegel, J. R. H. 2002. *Thermal Radiation Heat Transfer*. Taylor and Francis.

[49] Stam, J., and Fiume, E. 1995. Depicting fire and other gaseous phenomena using diffusion processes. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques*, Siggraph 1995, 129–136. New York, NY, USA: ACM.

[50] Stam, J. 1995. Multiple scattering as a diffusion process. In *Eurographics Symposium on Rendering/Eurographics Workshop on Rendering Techniques*, 41–50. ACM.

[51] Sun, B.; Ramamoorthi, R.; Narasimhan, S. G.; and Nayar, S. K. 2005. A practical analytic single scattering model for real time rendering. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH 2005, 1040–1049. New York, NY, USA: ACM.

[52] Szirmay-Kalos, Gbor Liktor, T. U. B. T. 2009. Fast approximation of multiple scattering in inhomogeneous participating media. In *Proceedings of the Eurographics Workshop on Rendering*, 53–56. ACM.

[53] Tangren, C. D. 1982. Scattering coefficient and particulate matter concentration in forest fire smoke. *Journal of the Air Pollution Control Association* 7:729–732.

[54] Tessendorf, J., and Wasson, D. 1994a. The WKB approximate solution for radiative transfer in a scattering medium. *Characterization and Propagation of Sources and Backgrounds* 23:477.

[55] Tessendorf, J., and Wasson, D. 1994b. Impact of multiple scattering on simulated infrared cloud sense images. In *Characterization and Propagation of Sources and Backgrounds*, volume 23, 462.

[56] Tessendorf, J. A. 1991. Underwater solar light field: analytical model from a wkb evaluation. 10–20.

[57] Veach, E., and Guibas, L. J. 1995. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of the 22nd Annual Conference on Computer*

Graphics and Interactive Techniques, SIGGRAPH 1995, 419–428. New York, NY, USA: ACM.

[58] Veach, E., and Guibas, L. J. 1997. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, 65–76. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co.

[59] Venceslas, B.; Didier, A.; and Sylvain, M. 2006. Real time rendering of atmospheric scattering and volumetric shadows. *Journal of WSCG* 14(1):65–72.

[60] Whitted, T. 1980. An improved illumination model for shaded display. *Commun. ACM* 23(6):343–349.

[61] Yue, Y.; Iwasaki, K.; Chen, B.-Y.; Dobashi, Y.; and Nishita, T. 2010. Unbiased, adaptive stochastic sampling for rendering inhomogeneous participating media. In *ACM SIGGRAPH Asia 2010 papers*, SIGGRAPH ASIA 2010, 177:1–177:8. New York, NY, USA: ACM.

[62] Zhou, K.; Ren, Z.; Lin, S.; Bao, H.; Guo, B.; and Shum, H.-Y. 2008. Real-time smoke rendering using compensated ray marching. In *ACM Siggraph 2008 papers*, 36:1–36:12. New York, NY, USA: ACM.

[63] Zinke, A.; Yuksel, C.; Weber, A.; and Keyser, J. 2008. Dual scattering approximation for fast multiple scattering in hair. In *ACM Siggraph 2008 papers*, 32:1–32:10. New York, NY, USA: ACM.