Rendering of Teeth and Dental Restorations using Robust Statistical Estimation Techniques

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Jin Woo Jung

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Gary Meyer

February, 2016

Acknowledgements

First of all, I would like to express my sincere appreciation to my advisor, Prof. Gary Meyer, for his advice, patience, motivation, and for sharing his knowledge with me. His advice and counseling about my research helped me achieve my goal in obtaining my Ph.D. I could not have had a better mentor than Prof. Gary Meyer for my Ph.D. advisor.

In addition to my advisor, I would like to express my gratitude to the rest of my thesis committee: Prof. Victoria Interrante, Prof. Vladimir Cherkassky, and Dr. Ralph DeLong. Their insightful comments and encouragement helped me comprehend my research from various perspectives. In particular, I am grateful to Dr. Ralph Delong for his continuous support of my Ph.D study and related research.

My profound thanks also goes to Prof. Maria Pintado, Dr. William Douglas, Prof. Alex Fok, Prof. Conrado Apracio, and Dr. WookJin Seong, who provided a great working environment in MDRCBB and UMBEL. Without their invaluable support I would not have finished my Ph.D study.

Furthermore, I thank 3M and Dr. Brian Holmes for the data used in this thesis, and I also thank my fellow lab mates in Computer Graphics group and MDRCBB for the discussions and for the all the fun we shared.

Last but not least, I would like to thank my family and friends: my parents and my brother for supporting me spiritually throughout my Ph.D study. In particular, I appreciate the support of fellowship of Young Heo, Bonita Van Heel, Dr. Yung Chung Chen, and Dr. David Nedrelow.

Abstract

Robust image estimation and progressive rendering techniques are introduced, and these novel methods are used to simulate the appearance of teeth and dental restorations. The realistic visualization of these translucent objects is essential for computer-aided processes in the field of dentistry, because a successful dental treatment is dependent on the recovery not only of the tooth's function, but also its appearance. However, due to the heterogeneity of the tooth structure and the coupled subsurface scatterings that this causes, simulating the translucency of these objects presents a difficult computational challenge. A Monte-Carlo ray tracing system is employed to model the complex interactions of light within the material and to develop the robust image estimation and progressive rendering techniques.

Because low probability samples are infrequently encountered in an image, for standard Monte-Carlo estimation these samples can become noise. Robust image estimation techniques are suggested as a way to suppress these low probability samples, and it is demonstrated that for a given sample size robust estimation techniques can produce less noisy renderings. In other words, the sample size necessary to satisfy a certain user requirement will decrease, and an improvement in rendering speed can be obtained. The robust estimation techniques are discussed in both pixel and image space, and their statistical analysis is provided. This analysis determines the inclusion rate for sample probabilities and is thus able to specify the sample probability thresholds necessary to discard or include samples. The statistical analysis also makes it possible to determine the performance boundaries in terms of the number of disclosed low probability samples in an image; as a result, a sample size for a given user requirement can be identified.

A progressive approach for rendering translucent objects based on volume photon mapping is also presented. Because conventional volume photon mapping requires long preprocessing to build up a complete volume photon map, it is not able to support progressive rendering. Even worse, due to the limited memory space in a given computer system, the rendering results suffer from a potentially incorrect volume photon map. Progressive volume photon mapping uses a subset of volume photons for rendering, so it provides a high frame rate for preview rendering. In addition, by recycling the volume photons used for previous image estimation, progressive volume photon mapping does not suffer from memory restriction. It is therefore able to use a virtually unlimited number of volume photons and this makes exact rendering plausible.

Although these methods were developed to realistically visualize teeth and dental restorations, they are effective in any rendering situation that suffers from noise, restricted computational performance, and limited memory space; as a consequence, these procedures are expected to be useful for many other types of realistic image synthesis including motion picture special effects and video games. The statistical interpretation developed for robust estimation is based on the pixel radiance sample probability. This allows the image synthesis sampling problem to be studied in a manner similar to how it would be treated in other established fields of science and engineering: in terms of the statistical properties of the signal to be sampled. This approach can provide the groundwork for further stochastic analyses in the context of computer graphic rendering.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Teeth not only serve a function for nutrition, but also play an important role in aesthetics because they influence the overall facial appearance of a person. For instance, the goal of cosmetic dentistry is to improve how the patient looks. Therefore, dentists need to know how the restorations appear with adjacent teeth within the oral cavity after treatment, and the success of the treatment depends on the appropriate selection of shades of dental restorative materials.

For systematic evaluation, design, and manufacturing of dental restorations, recent technological advances introduce computer-aided tools to the field of dentistry, but the appearance evaluation is still dependent on visual estimation or is sometimes even ignored [1]. This is because the appearance of a tooth is a combination of complicated direct and indirect reflectance on the tooth's surface. The appearance is also a result of intricate scatterings of light within the heterogeneously structured and irregularly shaped tooth.

For physically correct rendering of teeth and restorations, two light transport mechanisms need to be addressed: indirect illumination and subsurface scattering. As teeth are enveloped by gums, situated with other teeth within the oral cavity, and exposed to many unexpected environments, their appearance is influenced not only by direct illumination but also indirect lighting introduced by reflected light from other materials around the teeth. The light transport mechanism within a tooth or restoration is called subsurface scattering. In translucent materials, light is not only reflected off the surface, but it also enters the object, scattering throughout and leaving at different locations and directions. As a result, the object appears smooth and soft, blurring surface geometric details.

Because of this complexity, mathematical modeling or computer simulation of the appearance of a tooth or restoration is non-trivial and computationally expensive; many recent dental software solutions, such as design tools for dental CAD/CAM products, do not provide the correct look for the final result. If it is possible to predict the correct appearance of teeth and dental restorations after treatment or fabrication, the productivity of dental color matching and CAD/CAM processes will be significantly improved by the powerful functions of material editing and shade selection. On the other hand, an understanding of light propagation in biological tissues is essential for effective applications of light in medical diagnostics and therapeutics [2, 3, 4]. Thus, the rendering of teeth using physically correct simulations of light transport within dental tissues will be useful in successful research for dental diseases and treatments, such as probing for dental caries [4].

The realistic visualization of these kinds of materials has been a long-standing challenge in the field of computer graphics due to the strong demands from game, movie, and design industries [5]. However, the rendering of translucent objects is still computationally expensive because the light interactions within translucent volumes are too complex to be modeled completely; that expensive cost slows down the rendering process and makes it suffer from limited computing power and storage capacity. Only with unrestricted processing power and memory space, can the correct simulation results through a full Monte-Carlo approach be expected. Otherwise, only simple geometric models or approximate material characteristics can be successfully rendered [6, 7, 8].

Recently, many studies show that better modeling of physical interactions can save costs. In particular, they take advantage of the fact that the light scattering process can be approximated by a diffusion process [9, 10, 11, 12, 13, 14, 15, 16]. This diffusion approximation assumes a homogeneous structure and a specified shape of the objects, which need to be flat, optically thick, or layered. Consequently, when the diffusion approximation is applied to optically thin and non-specified geometric objects, the simulation results are unlikely to be correct even though they can be artistically satisfying. A tooth is small and heterogeneously structured with enamel and dentin, so it is rarely to be expected that the simulations using the diffusion approximation produce the correct results. For this small and heterogeneously structured object, the accurate prediction of the appearance requires more complete and sophisticated models than the diffusion approximation.

On the other hand, the computational efficiency of rendering can be improved by rejecting inadequate samples. Recent sample selection techniques focus on how to reject extreme values, which can introduce dark or bright speckles in a rendition [17, 18]. Another way to improve the computational efficiency can be found in image processing. Particularly, any weighted mean based filtering approach, such as Gaussian, is effective at the expense of image blurriness. The image blurriness can be reduced by choosing appropriate filtering weights [19]. This is because the pixel value in a rendering is the sum of incoming samples that have a high correlation between neighboring samples.

1.1 Robust Estimation and Progressive Rendering

This dissertation first discusses existing approaches for translucent object rendering, sample selection, and filtering in the context of ray-tracing. Then it proposes robust statistical pixel value estimation techniques in pixel and image space for eliminating noise in a rendering, and it introduces a progressive translucent object rendering technique for creating an instant preview and generating the final result.

The standard Monte-Carlo estimator is the average of samples, which is optimal for an underlying Gaussian distribution. However, it may not be optimal for different types of distributions. In particular, different maximum-likelihood estimators have been developed for different distributions of underlying populations. In the field of signal processing, for example, a different estimator is chosen for a different distribution of incoming samples. The distribution is determined by the communication channel of the system and by the characteristics of the noise within that channel. For example, the process of linear signal equalization can be regarded as a weighted averaging process, which is optimal for a Gaussian distribution of weighted incoming samples. For spike noise, which is commonly observed in a power line communication channel, a non-linear signal filter may be useful. A similar approach can be found in the field of image processing. A Gaussian image filter is optimal when the distribution of the weighted pixel values is Gaussian. On the other hand, for salt-and-pepper noise the median image filter is effective.

This thesis discusses the characteristics of sample distributions that occur in ray-tracing, and shows that the distribution potentially suffers from fat-tailedness, which can introduce impulse noise in a rendering. Because the samples from the tails occur with a low probability in a rendering, they can be effectively eliminated by suppressing low probability samples. The robust estimators are designed for eliminating the low probability samples, and through statistical analysis the sample probability threshold for discarding them is determined.

In addition, for high frame rate rendering, this dissertation proposes a progressive translucent object rendering technique which is based on an existing volume photon mapping approach. Because conventional volume photon mapping requires a complete volume photon map before actual rendering can begin, it takes extensive time to generate the first rendering in a progressive rendering process. Even worse, due to limited memory space in a given system, the volume photon map is expected to be incomplete in many cases. In particular, for optically thick objects, the volume photons tend to be localized and fail to cover the whole volume because a single light ray may create a large number of scattering processes. However, the progressive volume photon mapping proposed in this thesis uses a subset of the volume photons to generate a rendering, and it is therefore able to instantly give designers a preview of the approximate appearance. In addition, by recycling the volume photons used for previous image estimation, it does not suffer from memory restriction and it is able to produce a correct rendering. This progressive volume photon mapping can also take advantage of robust statistical estimation to accelerate the rendering speed, because the volume photons are split into multiple subsets in the rendering process.

1.2 Dissertation Overview

The remainder of this dissertation is organized as follows: Chapter 2 discusses the optical characteristics of human teeth and dental restorations. It covers the structure of the human tooth and the optical parameters of the tooth's parts. In addition, this chapter introduces the optical parameters of the restorative materials available in the present market. These optical parameters are the estimates for light scattering, scattering anisotropy, absorption, and refraction.

Chapter 3 presents analytical models for subsurface light transport and summarizes related work on rendering of participatory media. In the existing analytical models for subsurface scattering, the subsurface light transport is approximated by a diffusion process; therefore, these models can be used for estimating the effective scattering and absorption coefficients of participatory media. In addition, this chapter discusses prior work related to the rendering of participatory media in computer graphics.

Chapter 4 introduces statistically optimal techniques for pixel estimation, which allow acceleration of the rendering process. This chapter discusses the outgoing radiance distribution within a pixel, and shows that the standard Monte-Carlo estimate may not be optimal. As a replacement of the standard Monte-Carlo estimate, this thesis proposes robust estimators that can discard or include samples based on their sample probabilities. Because the noise in a rendering is from infrequently disclosed samples, these robust estimators can effectively eliminate the noise.

Chapter 5 presents hybrid approaches for robust pixel estimation and image filtering, which are extensions of robust statistical pixel estimation to image space. Compared to robust pixel estimation, these hybrid approaches require small memory space and quickly converge to the correct results by using information from neighboring pixels. A statistical analysis of these hybrid approaches is also provided in this thesis, and these results can be used to determine the analytical performance and optimal parameters for these estimators.

Chapter 6 discusses a progressive volume rendering approach in combination with robust estimation to render translucent objects. This chapter describes the problem of conventional volume photon mapping, which is inappropriate for the progressive production of a correct rendering. Then, the chapter introduces a progressive rendering approach based on volume photon mapping. This approach uses a subset of volume photons for each rendering frame; therefore, it is able to generate an approximate rendering in a relatively short period of time. In addition, because it does not suffer from memory restriction, it is able to use virtually unlimited volume photons, which allows the rendering to converge to the exact result. Furthermore, it is possible to apply robust estimation to the multiple subsets of volume photons in order to remove the noise introduced by low probability volume photon samples.

Chapter 7 concludes the dissertation. This chapter summarizes the research presented in this dissertation and discusses potential applications. In addition, future work is proposed.

Chapter 2

Teeth and Dental Restorations

2.1 Characteristics of Human Teeth

The primary function of human teeth is mastication for nutrition. In addition, they play an essential role in speaking, and their appearance is aesthetically important [20]. The tooth is composed of two hard tissues, the enamel and the dentin, as seen in Figure 2.1. The enamel is a hard acellular tissue that is formed by epithelial cells supported by the dentin. The dentin is less mineralized but more resilient than the enamel, which is a vital hard connective tissue that is created and supported by formative cells in the dental pulp.

Anatomically the tooth is divided into a crown and a root. The junction between the two parts is called the cervical margin. Only the crown is visible in the oral cavity, while the root is covered by gums. The outer part of the crown is composed of enamel, while the majority of the crown is dentin that surrounds the pulp chamber. Because these hard tissues are translucent and highly scattering media, both of them play an important part in determining the color of the tooth.

The appearance of these translucent tissues is affected by a subsurface process that scatters light. This subsurface scattering process governs an energy distribution in the tissues, which determines the appearance. Therefore, an understanding of light scattering is important to predict the appearance of teeth. However, due to the fact that the dental tissues are inhomogeneous and arbitrary shaped, the prediction of light scattering

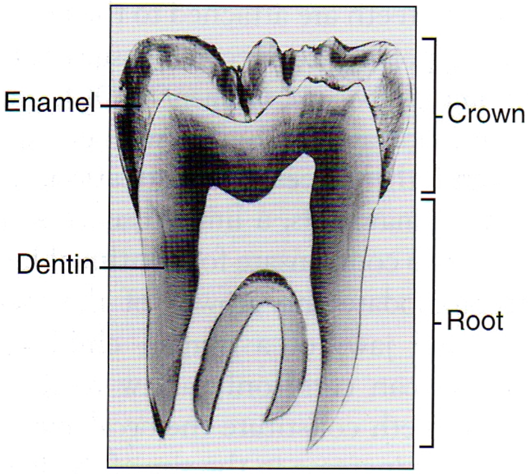


Figure 2.1: Ground section of a tooth

within the tissues is not trivial. Dental enamel is an ordered array of inorganic apatite-like crystals, and dentin is a complex structure, honeycombed with dentinal tubules. The complex structure of enamel and dentin makes the scattering process unforeseeable, which makes the energy distribution in the volume anisotropic and dependent on tissue orientation relative to the irradiating light source.

2.1.1 Enamel

The enamel is the outer part of the anatomic crown. It is the most highly mineralized tissue in the body and consists of about 96% inorganic material, mainly hydroxyapatite crystallites individually wrapped with traces of organic material. Because this inorganic material is vulnerable to demineralization in acid environments created by bacteria, the enamel is susceptible to dental caries. The mineralized tissue in the enamel also shapes much of its microscopic structure. Enamel consists of enamel rod and inter-rod, which are composed of hydroxyapatite crystallites packed in different organized patterns.

The cells that form enamel are ameloblasts. When enamel forms, the ameloblasts cover the whole surface of the enamel, but they are lost as the tooth emerges into the oral cavity. Once these cells are lost, enamel becomes a non-vital and insensitive tissue

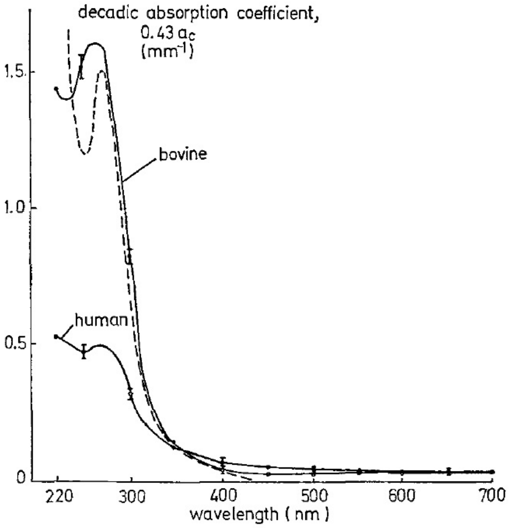


Figure 2.2: The decadic absorption coefficients of one sample of human and one sample of bovine enamel as computed from the total transmittance and reflectance data [21]; the dashed line shows the absorption coefficient caused by some, or all, of the organic material as determined from a urea solution of enamel dissolved in HCL at pH2.

that cannot be recovered or regenerated. In a strict biologic sense, enamel is dead tissue. However, it is capable of being permeated, particularly by the saliva.

Spitzer et al. analyzed the optical characteristics of enamel for light in the range of 200-700 nm using the Kubelka-Munk two-flux model [21]. They measured the reflectance and transmittance of thin slabs of dental enamel and found that the absorption was very weak in the visible range but the scattering was strong. Figure 2.2 and Figure 2.3 show the absorption and scattering coefficients of the enamel slab that are computed from the measurements. Compared to the absorption coefficients in Figure 2.2, the scattering coefficients in Figure 2.3 are much higher because light scattering is dominant over absorption in dental enamel. In addition, Spitzer et al. found no substantial anisotropy in enamel [21]. Fried et al. verified that result by fitting their experimental results to a linear combination of anisotropic and isotropic phase functions [4]. Their phase function is a combination of a HG function with $g = 0.96$ and a 30-60% isotropic fraction. The average cosine of the scattering angle is in the range of $g = 0.384$ to $g = 0.624$.

Figure 2.4 shows the refractive index of enamel as a function of the wavelength.

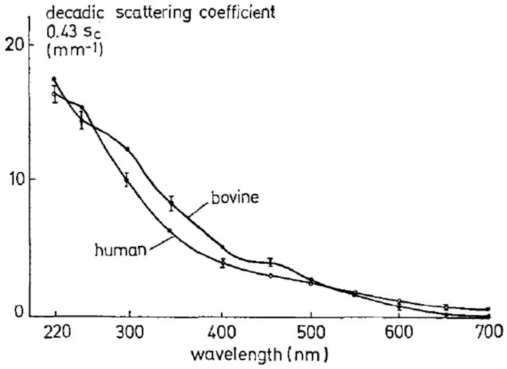


Figure 2.3: The scattering coefficients of the same human and bovine samples as used in Figure 2.2 [21]

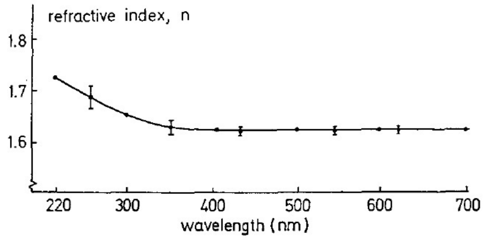


Figure 2.4: The refractive index n of the enamel as a function of the wavelength [21]

These values are consistent with the values of the refractive index of hydroxyapatite [21]. Meng et al. confirmed the value for the refractive index of enamel by obtaining 1.631±0.007 [22].

2.1.2 Dentin

Enamel is brittle because of its high mineral content. Therefore, without dentin, it can be easily broken by the force generated by mastication. Dentin prevents this fracture with the support of its more resilient tissue.

Dentin is a hard, elastic, yellowish white, avascular, sensitive tissue covering the pulp chamber with enamel on top of the dentin. By weight, 70% of dentin consists of hydroxyapatite crystals. The main organic component is the fibrous protein collagen. Closely packed tubules traverse the entire thickness of dentin for permeation and contain

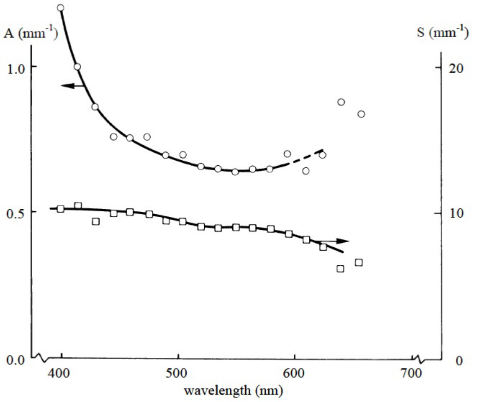


Figure 2.5: Spectral dependency of absorption (○) and scattering (□) for dentin [23]

the cytoplasmic extensions of the cells. These cells are called odontoblasts, which are aligned along the inner edge of the dentin and form the peripheral boundary of the pulp. The biological function of the odontoblasts is the formation of dentin.

Because enamel is optically thin, dentin plays a large role in the color of the complete tooth crown. Bosch et al. did analysis on optical properties of dentin by measuring the diffuse reflectance and transmittance of dentin slices [23]. They showed that the structures of the mineral crystals are the dominant cause of light scattering in dentin. Figure 2.5 and Figure 2.6 show the wavelength and location dependencies of the absorption and scattering coefficients. As seen in Figure 2.5, the absorption spectrum is rather flat, which is consistent with the greyish-yellowish color of dentin. Particularly, Figure 2.6 shows that the absorption and scattering coefficients vary at different locations because of different tubule densities and sizes. Therefore, it is desirable to assume that the volume of dentin is inhomogeneous.

Meng et al. measured the refractive index of dentin and obtained the value of 1.540±0.013 [22]. Kienle et al. showed that light propagation in dentin depends on its microstructure and the phase function strongly depends on the incident angle onto the tubules [2]. In addition, Kienle et al. showed that light propagation in dentin has a strong directional dependence that causes optical magnification due to multiple scattering within the dental microstructure [3]. Fried et al. also found that the complex dentin structure makes light scattering highly anisotropic and they described the scattering phase function using a HG function with $g = 0.93$ and a very weak isotropic scattering

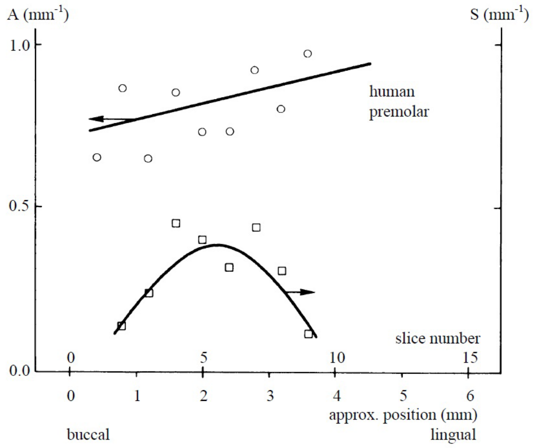


Figure 2.6: Locational dependency of absorption (○) and scattering coefficients (□) for dentin [23]; the plotted values are the averages of values at wavelengths of 535, 550 and 565 nm.

component [4].

2.1.3 Pulp

The central pulp chamber is enclosed by dentin and filled with pulp. While dentin is a hard tissue, the pulp is soft and disappears in dried teeth, leaving an empty chamber. Embryologically, histologically, and functionally, dentin and pulp are the same and should be considered as one entity. Pulp produces the dentin that surrounds it, nourishes the avascular dentin, carries nerves that give dentin its sensitivity, and is capable of producing new dentin when necessary. All these functions of the dental pulp relate also to dentin.

2.2 Characteristics of Dental Restoration Materials

Dental restorative materials are specifically designed bio-materials for dental restorations. Dental restorations need to look like real teeth, and their optical characteristics are expected to be similar to those of teeth. In addition, different teeth can have different appearances due to the lifestyle, age, and biology of patients. Therefore, the

	TT	TR	TA	n	g	Ω	τ
D65 White	61.21%	32.28%	6.51%	1.5399	0.75	0.99424	2.72
450 nm	48.97%	30.16%	20.87%	1.5498	0.75	0.9843	3.63
540 nm	60.16%	32.43%	7.41%	1.5410	0.75	0.9937	2.84
605 nm	65.40%	32.55%	2.05%	1.5369	0.75	0.998	2.44

n: refractive index
g: average cosine of the scattering angle
Ω: scattering albedo
τ: extinction coefficient times the thickness of the sample
Smaple thickness = 0.5 mm

Table 2.1: Optical properties of a dental restorations sample

materials are provided with different shades and translucent levels. To show different appearances, dental composites are mixed with filler, matrix materials, and color pigments [24].

Friebel et al. studied optical characteristics of three composite resins through an inverse Monte Carlo simulation (iMCS) [24]. By minimizing the error between Monte Carlo simulations and experiments, an iMCS process can be used to discover the optical parameters of translucent materials. The iMCS process of Friebel et al. computes the error for the diffuse reflectance and transmittance of those composite resins.

Figure 2.7 shows the calculated optical parameters for shade B2 of three composites using iMCS. The estimated absorption and scattering coefficients show that those composite resins are highly scattering. It is interesting to note that, although those composite resins are all designed for the shade B2, their optical coefficients are different from each other. This means that different combinations of optical parameters can produce a similar appearance. The spectral dependencies of the absorption and (effective) scattering coefficients are in agreement with the results for enamel [21]. In addition, the effective scattering coefficients for the composites are in a similar range of those of enamel (Figure 2.3).

Jung et al. [25] also present optical parameters and renditions of a dental restoration sample, and the parameters are in agreement with the Friebel et al.'s result [24]. Table 2.1 shows the measured data and the calculated optical parameters of the restorative material sample.

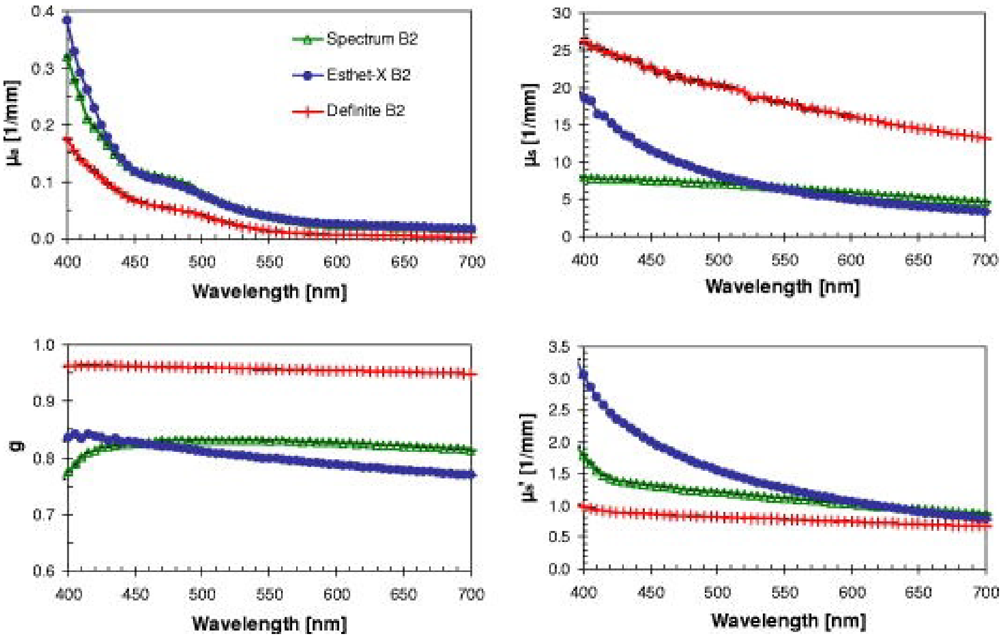


Figure 2.7: Spectral dependency of optical parameters of the three composite resins for shade B2 [24]

Chapter 3

Related Work

3.1 Analytic Model for Subsurface Light Transport

When light rays travel through participating media, the radiance along the rays is affected by physical processes. The main physical processes that affect the distribution of radiance within the media are absorption, emission, and scattering. In the absorption process, the light is converted into another form of energy, such as heat, so it decreases the amount of radiance of the ray. In the emission process, the radiance along a ray is increased by chemical, thermal, or nuclear processes that convert energy into visible light. Sometimes, the rays collide with the particles within the volume and are scattered in other directions [26]. This effect is called out-scattering, which decreases the radiance in the direction of the current ray. Radiance from other rays may be scattered into the path of the current ray and increase the radiance. This effect is called in-scattering.

The light scattering process in participatory media is defined by three physical processes: absorption, emission, and scattering. Each physical process is characterized by a single coefficient that is determined by the stochastic property of the process. The absorption process is characterized by the absorption coefficient, which is the conversion rate of light into another form of energy per unit distance. The volume emission process is defined by the emission coefficient, which is the discharged amount of light per unit distance. The light scattering process is modeled using the scattering coefficient, which is the probability of a light ray changing traveling direction. Another stochastic process to describe the subsurface scattering is the set of the scattering directions, which is determined by a probability density function of scattering angles. The probability density function is defined by the average cosine of the scattering angles for a phase function. These optical parameters may be constant or vary throughout a given volume. If the optical parameters are constant within the volume, the object is considered as homogeneous; otherwise, it is regarded as inhomogeneous or heterogeneous.

The problem of photon propagation in participating media can be treated as a random walk process, which can be simulated by a Monte-Carlo method [27]. Therefore, the optical parameters can be reduced by minimizing the error between simulation results and experimental measurements. However, this process is computationally expensive because comprehensive simulation for every combination of optical parameters may be necessary. On the other hand, for idealized situations, analytic solutions for light transport in turbid media are available.

Ragain et al. show that the Kubelka-Munk theory accurately predicts the reflectance of enamel and dentin without considering light emission [4, 28]. Generally, the telegrapher's equation, diffusion equation, and multiple flux theories such as the Kubelka-Munk and four flux theories are available for analytic models of subsurface scattering. These analytical models describe the transmission and reflection densities of light for homogeneous and isotropic scattering media [27].

Since Kubelka and Munk introduced a theory for the analysis of diffuse reflectance spectra from weakly absorbing material, the photon propagation problem in translucent media has been formulated as a random walk process and simplified to a simple diffusion problem. However, these approaches fail if a significant amount of absorption is present within the volume [27]. Another analytic model for the random-walk process is the telegrapher's equation, which has been shown to be a significant improvement over the diffusion equation. Kong et al. demonstrate that the solution of the telegrapher's equation comes closest to the Monte Carlo simulation data of optical transmission and reflection density for translucent media [27].

In addition, these theories can be extended to the case of anisotropic scattering using an effective scattering coefficient. The effective scattering coefficient is a reduced version of the original scaled down by a factor of $(1 - g)$, where g is the average cosine of the scattering angles. In this chapter we discuss two theories, the telegrapher's equation and Kubelka-Munk, which are used to determine optical parameters from measurements.

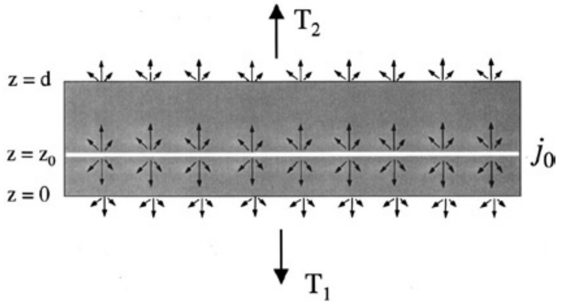


Figure 3.1: Translucent slab with a diffuse light source at $z = z_0$ having a current light source density of j_0

3.1.1 Telegrapher's equation

For a diffuse light source at $z = z_0$ and a turbid slab that is infinite in the x-y plane and has thickness d, a photon density within the slab can be described by the telegrapher's equation of

$$D' \frac{\partial^2 n}{\partial z^2} - acn = -j_0 \delta(z - z_0), \tag{3.1}$$

where

$$D' \equiv \frac{c}{3(s + \beta a)}, \tag{3.2}$$

and

n = number of photons per unit volume
a = absorption coefficient
s = scattering coefficient
c = speed of light
j_0 = current density of the light source
β = constant that decides front propagation speed [29].

Figure 3.1 shows the turbid slab model that the telegrapher's equation describes, where T_1 is the total transmission at $z = 0$ and T_2 is the total transmission at $z = d$.

The general solution to equation 3.1 is given by

$$\begin{gathered} n(z) = n_1(z) = A_1 \sinh[q(z - z_0)] + B_1 \cosh[q(z - z_0)], \quad z < z_0 \\ n(z) = n_2(z) = A_2 \sinh[q(z_0 - z)] + B_1 \cosh[q(z_0 - z)], \quad z > z_0 \\ q = \sqrt{3a(s + \beta a)}. \end{gathered} \tag{3.3}$$

For $z = z_0$, $n_1(z)$ and $n_2(z)$ have the same density, and we therefore have the following boundary conditions:

$$\begin{aligned} n_1(z_0) &= n_2(z_0) \\ j(0) &= -\eta_1 c n_1(0) \\ j(d) &= \eta_2 c n_2(d), \end{aligned} \tag{3.4}$$

where $j(z) = -D' dn/dz$ is the current density, η_1 is a parameter that defined the boundary condition for $z = 0$, and η_2 is a parameter for $z = d$. Because the telegrapher's equation is an approximation of the full radiative transfer equation, there is no rigorously correct way to determine the boundary conditions. Therefore, it is not a trivial task to find η_1 and η_2 [27]. Durian et al. adopted $\eta_1 = \eta_2 = \frac{1}{2}$ assuming that there are no reflections at the boundaries [29]. Kong et al. also used $\eta_1 = \eta_2 = \frac{1}{2}$ for the calculations based on the telegrapher's equation [27].

From these boundary conditions, the diffuse transmittance T_1 and T_2 for the diffuse source location z_0 are obtained:

$$\begin{aligned} T_1(z_0) &= \frac{-j(0)}{j_0} = \frac{\eta_1 q(a\cosh[q(d-z_0)] + \eta_2 q \sinh[q(d-z_0)])}{(\eta_1+\eta_2)qa\cosh[qd] + (\eta_1\eta_2 q^2 + a^2)\sinh[qd]} \\ T_2(z_0) &= \frac{j(d)}{j_0} = \frac{\eta_2 q(a\cosh[qz_0] + \eta_1 q \sinh[qz_0])}{(\eta_1+\eta_2)qa\cosh[qd] + (\eta_1\eta_2 q^2 + a^2)\sinh[qd]}. \end{aligned} \tag{3.5}$$

By setting $z_0 = 0$, the total transmittance can be obtained by doubling $T_2(0)$ because the diffuse emission at z_0 is assumed to occur in both the forward and the backward half-spaces. In a similar way, the total reflectance can be obtained by multiplying $T_1(0)$ by 2. However, it is necessary to subtract 1 for the reflectance, because it is necessary to remove the effect of the diffuse light source. Therefore, the total transmittance (T) and total reflection (R) are given by

$$\begin{aligned} T &= 2T_2(0) \\ R &= 2T_1(0) - 1. \end{aligned} \tag{3.6}$$

From equation (3.5) and (3.6), the following total transmittance (T) and total reflection (R) for transparent media are obtained:

$$\begin{aligned} T &= \frac{2\eta qa}{2\eta qa\cosh(qd) + (\eta^2 + q^2 + a^2)\sinh(qd)} \\ R &= \frac{(n^2 q^2 - a^2)\sinh(qd)}{2\eta qa\cosh(qd) + (\eta^2 + q^2 + a^2)\sinh(qd)}. \end{aligned} \tag{3.7}$$

For normally incident light at $z = 0$, the first scattering event makes a continuum of diffuse light sources within the slab with a current density proportional to $s\exp(-\epsilon z_0)dz_0$, where $\epsilon = a + s$ is the extinction coefficient. In addition, a specularly transmitted component of $\exp(-\epsilon d)$ without scattering contributes to the total transmission. Therefore, by integrating $T_1(z_0)$ and $T_2(z_0)$ over z_0 with the continuum of the diffuse light sources and adding the specularly transmitted term to the total transmission, the following total transmittance and total reflectance are obtained.

$$\begin{aligned} T &= \frac{\eta q s\exp(-\epsilon d)((a\epsilon + \eta q^2)(\exp(\epsilon d) - \cosh(qd)) - q(\eta\epsilon + a)\sinh(qd))}{(\epsilon^2 - q^2)2\eta q a\cosh(qd) + (n^2 q^2 + a^2)\sinh(qd)} + \exp(-\epsilon d) \\ R &= \frac{\eta q s((a\epsilon - \eta q^2)(\exp(\epsilon d) - \cosh(-\epsilon d)) + q(\eta\epsilon - a)\sinh(qd))}{(\epsilon^2 - q^2)2\eta q a\cosh(qd) + (n^2 q^2 + a^2)\sinh(qd)}. \end{aligned} \tag{3.8}$$

3.1.2 Kubelka-Munk and four-flux theories

For diffusely incident light, from the Kubelka-Munk (KM) theory, the total transmittance (T_{KM}) and the total reflectance (R_{KM}) are given by

$$\begin{aligned} T_{KM} &= \frac{Q}{Q\cosh(Qd) + (K+S)\sinh(Qd)} \\ R_{KM} &= \frac{S}{K + S + Q\coth(Qd)} \\ Q &\equiv \sqrt{K(K+2S)}. \end{aligned} \tag{3.9}$$

For normally incident light, from the four-flux theory, the total transmittance (T_{4F}) and the total reflectance (R_{4F}) are given by

$$\begin{aligned} T_{4F} &= \frac{Qs\exp(-\epsilon d)((K\epsilon + Q^2)(\exp(\epsilon d) - \cosh(Qd)) - Q(\epsilon + K)\sinh(Qd))}{(\epsilon^2 - Q^2)(2KQ\cosh(Qd) + (Q^2 + K^2)\sinh(Qd))} + \exp(-\epsilon d) \\ R_{4F} &= \frac{Qs((K\epsilon - Q^2)(\cosh(Qd) - exp(-\epsilon d)) + Q(\epsilon - K)\sinh(Qd))}{(\epsilon^2 - Q^2)(2KQ\cosh(Qd) + (Q^2 + K^2)\sinh(Qd))}. \end{aligned} \tag{3.10}$$

In Equation 3.9 and 3.10, K is the absorption rate and S is the scattering rate for diffuse light. K and S are typically extracted through a experimental process. Unfortunately, the relationship between these rates and optical parameters (absorption and scattering coefficients) is not clear in these theories, and different theories have different definitions for the relationship [27]. For the KM and four-flux theories, S and K are defined by:

$$S = \frac{\alpha s}{2}, \tag{3.11}$$

$$K = \alpha a, \tag{3.12}$$

$$\alpha = \frac{1}{I_0} \int_0^{\pi/2} \int_0^{2\pi} I(\theta, \phi) \tan(\theta) d\theta d\phi, \tag{3.13}$$

$$I_0 = \frac{1}{I_0} \int_0^{\pi/2} \int_0^{2\pi} I(\theta, \phi) \sin(\theta) d\theta d\phi. \tag{3.14}$$

For the revised KM theory, S and K have the following relationship with the optical parameters:

$$S = \frac{\mu \alpha s}{2}, \tag{3.15}$$

$$K = \mu \alpha a, \tag{3.16}$$

$$\mu = \begin{cases} (\frac{s^2}{a^2+as})^{1/4} & \text{for} s^2 \geq a^2 + as \\ 1 & \text{for} s^2 < a^2 + as. \end{cases} \tag{3.17}$$

For the modified KM theory, S and K are defined by:

$$S = \alpha_S s/2, \quad K = \alpha_K a, \tag{3.18}$$

$$\alpha_S = \sqrt{\frac{3}{\beta}}, \quad \alpha_K = \sqrt{3\beta}. \tag{3.19}$$

It is interesting to note that Kong et al. introduce the modified KM theory by setting $\mu = 1$ and allowing different α for S and K. In addition, the solutions obtained by the modified KM theory are equivalent to the solutions obtained by the telegrapher's equation.

3.2 Rendering of Translucent Objects

In computer graphics, rendering of translucent materials has usually been ignored or simplified to surface reflection because of the high computational cost associated with a simulation of subsurface scattering. However, with recent technology advances in high performance computing, more accurate simulation becomes possible for a given time period. Dorsey et al. show that volume photon mapping is effective in rendering of translucent marble and stone [6]. Jensen et al. demonstrate that volume photon mapping with path-tracing is capable of realistic rendering of wet materials [7] . Pharr et al. show that MC evaluation using scattering equations is effective in rendering of low scattering albedo materials [8]. These approaches are more complete than before and synthesize a more realistic translucent appearance on a computer screen, but they are still computationally costly because all the complex physical interactions need to be simulated. These physical interactions can be approximated by a diffusion process; therefore more efficient simulation becomes possible. Since Stam et al. introduced diffusion approximation for rendering to computer graphics, the diffusion model of light scattering has been widely employed [30]. If the materials are assumed to be optically dense, multiple scattering can be approximated by a diffusion process. Jensen et al. formulate a bidirectional surface scattering reflectance distribution function (BSSRDF) for homogeneous materials with a dipole diffusion approximation, and Wang et al. introduce an efficient way to solve the diffusion equation for heterogeneous materials using a finite difference method (FDM) [12, 15]. While the diffusion approximation produces an accurate rendition for highly diffusive and optically dense material, it fails for less diffusive materials or highly anisotropic scattering models [31].

Jensen et al. introduce a BSSRDF of the sum of single scattering and diffusion terms for rendering of homogeneous materials [12]. The single scattering term is an approximation of low-order scattering, and the diffusion term is that of high-order multiple scattering. Jensen et al. then present a two-pass hierarchical integration technique to accelerate the computation of multiple scattering components [13]. In the first pass, the light incident on the surfaces of all translucent objects in a scene is recorded. The second pass uses the recorded irradiance to calculate the subsurface transport at each shading point. The dipole diffusion approximation assumes isotropically scattering semi-infinite homogeneous materials, therefore it can produce an unexpected appearance for optically thin or highly anisotropic scattering objects [26]. In diffusion approximation, anisotropic scattering materials are approximated by isotropic scattering materials with appropriately scaled scattering coefficients, which are called effective or reduced scattering coefficients. By incorporating the BSSRDF with precomputed radiance transfer (PRT), Wang et al. presented a technique for interactive rendering of translucent objects under all-frequency environment maps, where they employ the two-pass hierarchical technique and include an approximated single scattering term in the precomputation stage for homogeneous materials and direct lighting [14]. Xu et al. introduce real-time homogeneous material editing under fixed illumination using PRT with appropriately chosen basis functions [32].

Although the two-pass method can improve performance dramatically by sharing irradiance estimates computed in the first pass, its linear, brute force irradiance evaluation comes with a significant computational cost. Arbree et al. introduce a single-pass algorithm that refines hierarchical and adaptive estimates of both the surface irradiance and the subsurface transport. By using an adaptive and top-down refinement method, the algorithm can dedicate more effort to simulating eye-surface-light paths that make significant contributions to the final image [33].

For correct rendering of thin parts or right angle edges of translucent objects, Donner et al. introduce a diffusion approximation algorithm using multi-poles and quad-poles [9, 10]. While the dipole diffusion approximation assumes that the material is homogeneous and semi-infinitely thick, the multi-pole diffusion approximation can model thin slabs, such as paper, and even multi-layered heterogeneous materials, such as skin. In addition, the quad-pole diffusion approximation can model right angle edges of translucent materials. Using a linear combination of dipole, multi-pole, and quad-pole diffusion reflectance values, Donner et al. showed that more accurate rendering of complex geometries is possible [10].

Although, for homogeneous materials or simple geometries the diffusion approximation can be approximated by dipole, multi-pole, and quad-pole models, for heterogeneously structured translucent materials with complex geometries the diffusion equation needs to be solved with a boundary condition defined by a given lighting condition. Wang et al. introduce a polygrid diffusion algorithm to solve the diffusion equation using an FDM [15]. This algorithm models an object as a polygrid that has regular connectivity and an irregular shape. The regular connectivity makes it possible for this algorithm to be implemented on a GPU for real-time performance. However, the regular connectivity modeled using cubic structures limits the shape of objects to simple geometries (low genus and no sharp features). Wang et al. address this problem by supporting arbitrary geometries with tetrahedra structures [16]. Even though the FDM shows real-time performance for rendering of heterogeneous materials, it can incur approximation errors [34]. Arbree et al. show that a Finite Element Method (FEM) is more accurate than the FDM approach [34].

Although diffusion approximation makes an efficient simulation possible, it fails to capture the appearance caused by low-order scattering that is dominant for optically thin or highly anisotropic scattering materials. For instance, Donner et al. show that the dipole diffusion approximation fails to reproduce the appearance of orange juice [31]. Instead of using the diffusion approximation, Donner et al. suggest the use of empirical data that represent a BSSRDF model [31]. For spatially uniform, homogeneous and semi-infinite thick materials, they reconstructed 2D hemispherical distributions of exitant light through a large-scale simulation of photon tracing. Although the empirical BSSRDF is successful in simulating the effect of low-order scattering, reconstruction of BSSRDFs for various materials with complex geometries and different structures seems to be impossible due to the computational complexities of the simulation.

The interactive rendering of translucent materials has been another challenge in computer graphics, because subsurface scattering is such a complex physical process. One way to provide interactivity is to perform advance complex operations that are required in the rendering phase; this is called precomputation. Approximation of physical interactions or geometries can accelerate the rendering speed by sacrificing visual quality. Alternative mathematical modeling and efficient memory management can be adopted for interactive rendering. In addition, some user interface restrictions can be employed, such as limited camera angles or finite material properties, depending on the purposes of applications.

Sloan et al. first introduce PRT using low-order spherical harmonic basis functions to encode the light transport vectors [35]. PRT allows complex lighting effects to be evaluated at runtime and makes interactive rendering possible because it computes the radiance response at every vertex and stores them in advance as transport vectors. PRT computes the illumination of a point as a linear combination of the incident irradiance at an interactive rate. In particular, a low frequency approximation of the original light transport functions can reduce the size of the transport vectors, thereby decreasing the workload in the rendering phase. However, this approximation allows rendering to lose high spatial frequency information, so that the resulting images may be incorrect. Cheslack-Postava et al. present a method that uses 4D wavelet basis functions for efficient data compression and relighting with high-resolution [36]. For rendering of translucent objects, Wang et al. show that PRT can be combined with the dipole diffusion approximation approach, and Zhang et al. show that PRT can be incorporated with volume photon mapping [14, 37].

Another major limitation of PRT is that it only works on static scenes because the transport vectors are computed for a fixed scene setup; therefore, rendering animations or scenes with dynamic lighting conditions are still computationally expensive. Particularly, the prohibitive cost of sampling the per-vertex visibility on every frame makes it difficult to apply PRT to animated scenes in real-time. In order to address moving geometry efficiently, Nowrouzezahrai et al. introduce a data-driven technique for animated PRT vectors as a function of their joint angles [38].

Representative examples of efficient memory management for photon mapping or volume photon mapping can be found in progressive rendering approaches. Because photon mapping and volume photon mapping require a tremendous number of photons, they need to reserve a large amount of memory space for the photons. The large amount of memory space is not only a restriction on hardware, but also a bottleneck of data flow. Hachisuka et al. reformulate the existing photon mapping algorithm and reorganize the process [39]. Instead of capturing the photons in the preprocess stage, Hachisuka et al. trace camera rays and store all hit points along the paths [39]. In the rendering phase, the photons are captured to calculate the outgoing radiances at the hit points. It is important to note that the photons do not have to be stored because they are discarded right after computing their contribution to the radiances. Collin et al. present a progressive approach for volume photon mapping[40]. To capture the contribution of volume scattering, Collin et al. keep track of the beam information of camera rays, instead of the camera ray-surface intersections.

For more efficient computation, the factorization of a light transport equation can be considered. Each factor can be combined with others in the rendering process to achieve a high sampling rate. Mahajan et al. present a theoretical and empirical analysis of the BRDF in-out factorization [41]. Ma et al. show that the triple product algorithm can work with PRT to accelerate the rendering of scenes with complex materials under dynamic all-frequency illumination and changing views [42]. It is also a promising technique to take advantage of temporal coherence between image frames because consecutive image frames tend to show a high correlation to each other. Nehab et al. investigate a caching scheme based on reverse re-projection to exploit temporal coherence in visible surface regions, lighting conditions, and camera location [43]. Dachsbacher et al. present a method that uses shadow maps to filter out unnecessary computation [44].

For the purpose of material editing, the pixel values need to be parameterized as a function of the material characteristics, such as scattering and absorption coefficients, and the average cosine of the scattering angles. Wang et al. introduce a material editing function by solving the diffusion equation using an FDM for heterogeneously structured objects, but the diffusion equation does not capture the appearance of geometrically complex or optically thin objects [15]. Song et al. [45] show that a factored representation of the dipole diffusion approximation can be used for heterogeneously structured translucent object material editing, but they do not consider internal structures because the dipole diffusion approximation is defined on the surface of objects. Hašan et al. [46] introduce an editing algorithm using path-traced data. From the renderings of a static scene and different material properties, a homogeneous albedo-intensity mapping set and the derivative of each image pixel are obtained. Then, the final appearance is synthesized by combining this information with an arbitrary chosen albedo set, which is the material characteristic of a heterogeneously structured object. However, this approach fails to reproduce various anisotropic scattering and optical thicknesses because it is difficult to obtain comprehensive path-traced data.

Ever since translucent materials were approximated using a simple BRDF model, the realistic and interactive visualization of translucent materials has been a perennial challenge in computer graphics. However, there is no appropriate solution that supports the physically correct simulation of the appearance and real-time rendering speed simultaneously. The full Monte Carlo simulations produce physically correct appearances on a computer screen, but they are computationally expensive. The diffusion approximation offers efficient rendering, but it does not account for the appearance of thin objects or highly anisotropic scattering media. In addition, efficient rendering algorithms, such as PRT, still put restrictions on viewing directions, lighting conditions, and material characteristics.

3.3 Robust Image Reconstruction

The noise in MC ray tracing is traditionally addressed by post processing the image [47]. For example, in a contemporary MC ray tracer, Pharr and Humphreys implemented image filters, such as the box filter and the Gaussian filter, as part of the image reconstruction process [26]. By restricting the bandwidth, the box and Gaussian filters are used not only for anti-aliasing, but also for reducing image noise. Specifically, when the rendering process introduces significant amounts of noise with a small number of samples, these filters can effectively remove high frequency noise. For impulse noise within images, median-type filters are particularly effective [47]. However, these linear and non-linear image filters distort the true pixel value by combining multiple adjacent pixels. In addition, these filters are basically low-pass filters, and thus remove high frequency information without being intended to do so. Therefore, inappropriate use of image filters can make the images too blurry or eliminate important information.

Recent studies propose the removal of impulse noise by discarding outliers in the outgoing radiance sample distribution [17, 18]. These approaches take note of the fact that extremely large values in the distribution cause speckles within the rendering, and the techniques eliminate the extrema using outlier rejection algorithms. Because the sample distribution is multi-modal, these approaches are based on density estimation at sample points.

DeCoro et al. introduce a density-based outlier rejection algorithm to remove distinctive speckling from renditions [17]. They make a space-partitioning tree from the samples and calculate the average distance between samples found by the k-nearest neighbors algorithm (k-NN). If the distance is small enough, the samples are splatted into the final image. Otherwise, they are kept within the tree until they reach a certain threshold. The size of the tree is highly dependent on both k and the total samples, and the computational overhead is almost linear with respect to sample size and k. Specifically, they report that this approach adds a time overhead of around 20% to 40% in their test scenes for values of k in the normal range of 10 to 50 [17]. This approach also takes advantage of inter-pixel correlation by clustering samples over multiple pixels, not within a single pixel. This clustering approach is helpful for reducing the required memory for the space-partitioning tree and improving the image quality. However, the algorithm basically performs like an image filter, and can lose high frequency information in the middle of the rendering process.

On the other hand, Pajot et al. suggest a kernel-density based outlier rejection algorithm to address the bright spot artifacts [18]. Instead of using the average distance between pixels as an estimate of the density, they apply kernel-based density estimation to the outgoing radiance distribution. Calculating the bandwidth of the kernels for the samples generates information about the modes of the sample distribution. Finally, the last mode of the distribution is identified, and any sample which is greater than the last mode is rejected. They define the last mode as a group of overlapped kernels that is located at the end of the distribution. The algorithm updates the upper boundary to identify the outliers by keeping track of the modes. Because the memory size necessary to trace the distribution is not trivial, they introduce an aggressive way to approximate the modes using a constant bandwidth for the kernel. This approach saves the memory required for the mode information by discarding the modes that are smaller than the last extended mode, but it loses robustness by accepting more samples or by failing to create the extended modes. In addition, it still introduces a non-negligible computational overhead from 5% to 10% of the simulation time [18].

These outlier rejection algorithms can be regarded as robust statistical methods that cope with a wide range of probability distributions. Robust statistics is a study of the deviation from ideal parametric models and the dangers that result [48]. Specifically, robust statistical methods provide ways to estimate the distribution's location and scale parameters with good performance for fat-tailed distributions. These methods are well known in other fields such as statistics, image processing and digital communications. In particular, maximum-likelihood type estimators (M-estimators), such as the median, myriad, and meridian, have attracted attention in many fields because they can be generalized to numerous complex situations. The median image filter is a well known application of robust statistics in image processing for the purpose of attenuating impulsive noise [49]. Zurbach et al. study myriad filters to address the impulse noise that median based filters fail to remove properly [50]. Aysal and Barner show that meridian estimators outperformed the myriad and median estimators in power line communications and high-pass filtered multitone signals where impulse noise is present [51].

Chapter 4

Robust Statistical Pixel Estimation

Monte Carlo (MC) ray tracing is an effective but noise prone method for rendering realistic three dimensional scenes. For a surface point seen through a pixel, the ray tracer randomly samples possible incoming light paths and averages the outgoing radiances to form an estimate of the pixel value. Although the choice of an average or mean is best described with a Gaussian distribution, it may have high variance for a given sample distribution [52]. Unfortunately, the optimal estimator cannot be determined until a large number of samples has been collected. However, for the same sample size, using a better statistical estimator can reduce noise.

Salt-and-pepper noise in renderings is a representative example where the standard MC estimator fails. This type of noise is sometimes called spike noise, impulse noise, or impulsive noise. While salt-and-pepper noise in photographic images manifests itself as both dark pixels in bright regions and bright pixels in dark regions, in computer graphics it usually appears only as bright pixels in dark regions (see Figure 4.2a, 4.4a, 4.8a, 4.8b, and 4.8c). This is because the probability distribution of outgoing radiances in a pixel is heavy-tailed to the right [17, 18]. If the distribution was left-skewed the pixel would appear as a dark pixel in a bright region. The distribution of the outgoing radiance is usually highly skewed and multi-modal due to the complex material properties and illuminating conditions.

In addition to noise caused by the unpredictable incident radiance distribution, various rendering approaches have specific weaknesses associated with noise that require collecting high sample sizes. With an accurate sampling distribution, importance sampling will produce an outgoing radiance distribution with low variance even with a small number of samples. However, it is impossible to obtain the correct sampling distribution in advance. Even worse, importance sampling with an inaccurate sampling distribution sometimes produces an outgoing radiance distribution with higher variance and larger range, resulting in greater noise for the final rendition. Ultimately, the importance sampling strategy fails to produce desirable images because it may introduce spiked noise in global illumination simulations by putting excessive weight on very low probability samples.

To address noise from fat-tailed outgoing radiance distributions in an early stage of rendering, this chapter introduces the use of robust statistical estimators to evaluate pixels. This pixel value estimation algorithm identifies the outgoing radiance distribution by employing one more dimension in the final image buffer. The outgoing radiance is measured as a sample mean distribution, rather than a sample distribution. According to the central limit theorem, the sample mean distribution will grow to be normal as more samples are collected. Because sample mean distributions are uni-modal and bell-shaped, it becomes possible to generate final pixel values by applying univariate outlier rejection algorithms or by using robust statistical estimation techniques. The robust estimators not only remove the outliers, but they also produce a smaller variance for the estimate of the fat-tailed distribution. With a large enough sample size, both the MC estimate and the robust estimate converge to the true mean, but the robust statistical estimate is capable of producing a higher quality image even though it uses a smaller sample size.

The remainder of this chapter is divided into three sections. Section 4.1 introduces robust statistical pixel estimation algorithms. Section 4.2 presents simulation results. Section 4.3 discusses the benefits and limits of the proposed approach.

4.1 Robust statistical pixel estimation

This chapter presents a new way to evaluate pixel values based on the pixel's sample mean distribution of the outgoing radiance. Existing approaches are based on a sample distribution that is multi-modal and highly skewed, and thus these methods estimate sample density to identify the outliers [17, 18]. The sample mean distribution becomes uni-modal and symmetric as the sample size increases and eventually changes into normal for a large sample size. Therefore, it is possible to apply existing outlier labeling algorithms and robust estimators. In addition, robust estimates converge to the same value as the mean. It is not only because the mean, median, and mode become identical for the normal distribution, but also due to the fact that the normal distribution becomes narrow as the sample size increases [53].

This section discusses the resampling strategy, univariate outlier detection algorithms, and robust statistical estimators necessary to obtain the best quality renditions for a given sample size.

4.1.1 Resampling

The outgoing radiance distribution in a rendition's pixel is highly skewed and multi-modal. Unfortunately, many statistical estimators, including M-estimators, do not assume this is true, and they are not directly applicable to the distribution. However, the sampling distribution of the sample mean becomes a uni-modal distribution by the central limit theorem, and this distribution can be used to apply robust statistical estimators.

The MC estimator for a known sampling distribution $p(x)$ is given by [26]

$$F_N = \frac{1}{N} \sum_{i=1}^{N} \frac{f(X_i)}{p(X_i)} \tag{4.1}$$

where $f(x)$ is a function of outgoing radiances and N is the number of samples. The MC estimator F_N converges more quickly if the samples are taken from a distribution $p(x)$ that is very similar to $f(x)$ [26]. Equation 4.1 can be expanded to a sum of multiple estimators,

$$F_N = \frac{1}{M} \sum_{j=1}^{M} F_{n,j} \tag{4.2}$$

where $N = M \times n$, and $F_{n,j}$ is a jth sample mean of the resampled data with sample size n. $F_{n,j}$ is given by

$$F_{n,j} = \frac{1}{n} \sum_{i=1}^{n} \frac{f(X_{i+(j-1)\times n})}{p(X_{i+(j-1)\times n})} \tag{4.3}$$

where $f(X_{i+(j-1)\times n})/p(X_{i+(j-1)\times n})$ is the $(i + (j-1) \times n)$th sample. It is interesting to note that sample means are computed in a manner similar to the conventional resampling methods used in statistics, in that they use the subsample sets of input samples [54, 55]. The sample mean is another random variable because it is calculated from random variable X. The probability distribution of F_n is the sampling distribution of the sample means. This sample mean distribution becomes uni-modal and eventually normal for a large number of samples. Although the sample size required for the sample mean distribution to converge to normal is dependent on the underlying population, a sample size of 30 is large enough to make the distribution normal in many cases [53].

The implementation of resampling is reasonably straightforward in any ray tracing system. The camera ray samples in each pixel are random, so they can be interleaved in multiple image buffers whenever they are captured. In each image buffer, the sample mean is updated with the incoming samples. For instance, Equation 4.3 shows that the $(i + (j - 1) \times n)$th sample is kept in the jth image buffer. The order does not matter, because the data values are random samples. Therefore, it is possible to use a dedicated image buffer for each ray tracing process or to construct the image buffers one by one. In addition, the size of the image buffer can be dynamically adjusted during rendering. For instance, the number of sample estimates for each pixel can vary based on a distribution's scale parameter. This dynamic sizing of the sample estimates can make the estimation process more memory efficient and prevent bias due to skewness.

The approach that Equations 4.2 and 4.3 describe is a sampling strategy without replacement. However, sampling with replacement can be another approach to accelerate the speed at which the sample mean distribution converges to normal, by specifying a ratio of overlapped samples within the image buffers. For a fat-tailed distribution, sampling with replacement is effective at obtaining better quality renditions, even with a small number of samples. However, erroneous data can be propagated into multiple image buffers rather than being confined in one buffer, thereby introducing more speckles in the resulting images. Along with error analysis, error propagation is discussed in

Section 4.1.4.

4.1.2 Outlier rejection

Because it is uni-modal and becomes normal for a large enough number of samples, the sample mean distribution allows the use of existing outlier labeling algorithms. There are various methods to label outliers for univariate data sets, such as the standard deviation method, z-score and Tukey's method. Seo summarizes in detail the various outlier labeling algorithms applicable to rendering [56]. A trimmed estimator is probably the simplest robust estimator [57]. The trimmed estimator discards a fraction of the lowest and highest observations and calculates a statistic using the remaining observations [58]. If the discarded fraction is α, the estimator is called the α-trimmed estimator. The median is the most robust trimmed mean, where $\alpha = 0.5$. The computational complexity is dependent on the number of samples and the underlying algorithm employed to sort the samples. Compared to the trimmed mean based on the sample distribution, which DeCoro et al. illustrated, the trimmed mean based on the sample mean distribution will be much more efficient, because the number of sample means is much smaller than the total number of samples [17].

4.1.3 *M*-estimators

Since their development as part of robust statistics, M-estimators have played an important role in applications involving impulsive processes, such as radar clutter, ocean acoustic noise, and multiple access interference in wireless system communications [51]. In addition to applications in digital communications, ray tracing systems also suffer from impulse noise because of unpredictable illumination distributions and importance sampling strategies. Therefore, M-estimators are appropriate for ray tracing pixel estimation. Given a set of input samples $\{x_i|_{i=1}^{N}\}$, an M-estimate of their central tendency β is given by

$$\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \rho(x_i - \beta) \right], \tag{4.4}$$

where $\rho(x)$ is the cost function of the M-estimators. Depending on the cost function, the maximum likelihood (ML) estimates form special cases of M-estimators. For instance, mean, median, myriad and meridian estimators are special cases of M-estimators [51].

The estimators can be ordered from least to most robust as: mean, median, myriad, and meridian [59]. In the following, M-estimators are briefly reviewed from the perspective of pixel value estimation. More details about the estimators can be found in Huber [48], Yin et al. [49], Aysal and Barner [51], and Chen [59].

The ML estimate of location under Gaussian statistics is the mean, which is defined by

$$\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} (x_i - \beta)^2 \right] = \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \text{where } \beta \in [\min\{x_i|_{i=1}^{N}\}, \max\{x_i|_{i=1}^{N}\}]. \tag{4.5}$$

It is important to realize that the mean of the sample means equals the standard MC estimate.

For random samples that follow the Laplacian distribution, the ML estimate of location is defined as the median:

$$\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} |x_i - \beta| \right], \quad \text{where } \beta \in \{x_i|_{i=1}^{N}\}. \tag{4.6}$$

The sample median is more efficient and robust for heavy-tailed distributions than the sample mean. Therefore, the median can produce less noisy pixel values with smaller variance for a small number of samples. On the other hand, for samples drawn from a Gaussian distribution, the sample mean is asymptotically more efficient than the sample median [60].

For Cauchy distributed samples with a scale parameter γ, the ML estimate of location is the myriad, which is given by

$$\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \log\{\gamma^2 + (x_i - \beta)^2\} \right], \quad \text{where } \beta \in [\min\{x_i|_{i=1}^{N}\}, \max\{x_i|_{i=1}^{N}\}]. \tag{4.7}$$

The scale parameter γ can control the robustness of the myriad. As $\gamma \to \infty$, the sample myriad becomes the sample mean [61, 62]. On the other hand, the myriad converges to a mode-type estimator for $\gamma \to 0$ [51]. In other words, a smaller γ makes the estimator more robust, while a larger γ makes it less robust. Unfortunately, it is a non-trivial task to compute the myriad. An analytic solution for the myriad is available if the derivative of Equation 4.7 is a low order polynomial. However, if the derivative is a higher order polynomial, it is necessary to use non-linear optimization to search for zeros of the derivative or extrema of Equation 4.7. Currently, much research is under way to make the calculation more efficient especially in the areas of image and signal processing [63, 64].

For random samples drawn from the Meridian distribution with a scale parameter δ, the ML estimate of location is the meridian, which is defined by

$$\begin{gathered}\hat{\beta} = \arg\min_{\beta} \left[\sum_{i=1}^{N} \log\{\delta + |x_i - \beta|\} \right], \\ \text{where } \beta \in \{x_i|_{i=1}^{N}\}.\end{gathered} \tag{4.8}$$

Similar to the myriad, the tunable parameter δ can adjust the robustness of the meridian estimator. The sample meridian $\hat{\beta}$ converges to the sample median as $\delta \to \infty$, while it becomes a mode-type estimator as $\delta \to 0$. Therefore, like the myriad, a smaller δ also makes the estimator more robust, while a larger δ makes it less robust. The meridian $\hat{\beta}$ is one of the input samples that has the smallest cost function [51, 62]. Therefore, $\hat{\beta}$ can be found by sorting the cost functions, $\rho(x)$, of all input samples.

The computational cost of the M-estimators is largely dependent on the number of samples and each estimator's underlying algorithm. The computation of the median can be slower than that of the mean, because the sorting usually requires more operations (primarily because of comparisons) than the simple addition necessary to determine the mean. Furthermore, the computation of the myriad and meridian is clearly slower than that of the mean and median. The myriad needs to use non-linear optimization with multiple multiplications and logs in Equation 4.7. Although the meridian does not have to search for zeros or extrema, it still needs to compute multiple logs in Equation 4.8.

In many cases, these estimating techniques can be far less expensive than the sampling process in ray tracing. For instance, path tracing examples in PBRT require more than 1,000 samples/pixel, and each sample involves numerous floating-point operations for radiance calculations and intersection tests [26]. On the other hand, the number of sample means involved with the estimation process is usually less than 20 (see Section 4.1.4). Furthermore, the robust estimation process can run periodically or even at the end of the rendering process, because information about the underlying population is kept as the sample mean distribution throughout the rendering procedure. If the estimation is performed with a large number of samples as a post-processing step in the rendering procedure, the computational overhead becomes negligible.

4.1.4 Maximum size of pixel buffers

As the number of pixel buffers increases, the spatial and computational complexities of the proposed method grow as well. Therefore, specifying the maximum pixel buffer size helps to predict the memory requirements and computational overhead, and makes efficient rendering possible. In this section, we discuss an approach, for the median operator, to determine the maximum pixel buffer size based on the probability of a sample to contribute to a pixel.

Low probability samples with high values are often overestimated in an early stage of rendering, because all samples are assumed to have identical probability that is determined by the sample size. Even worse, the noise sometimes remains within the image throughout the rendering process, because the extreme values are too large to be attenuated by the increased sample size.

For a sample that is chosen with probability p_{sample}, the number of samples in a sequence of m independent random samples can be defined by a binomially distributed random variable X_{sample}. The probability with which X_{sample} equals k is given by the probability density function (PDF) [65]:

$$Pr(X_{sample} = k) = {}_mC_k \cdot {p_{sample}}^k (1 - p_{sample})^{m-k}, \tag{4.9}$$

where ${}_mC_k$ is a binomial coefficient for m and k.

The probability with which a sample mean includes the samples can be defined as $p_{estimate}$. For the total sample size N and the number of pixel buffers n, the sample size for a sample estimate is N/n. Therefore, $p_{estimate}$ for a small p_{sample} can be expressed as:

$$\begin{aligned} p_{estimate} &= Pr(X_{sample} > 0) \\ &= 1 - Pr(X_{sample} = 0) \\ &= 1 - (1 - p_{sample})^{N/n} \\ &= \frac{N}{n} p_{sample} + \mathcal{O}(p_{sample}^2) \\ &\approx \frac{N}{n} p_{sample}. \end{aligned} \tag{4.10}$$

For $p_{estimate}$ and n pixel buffers, the number of sample estimates can also be defined by a binomially distributed random variable $X_{estimate}$. Similar to Equation 4.9, the probability with which $X_{estimate}$ equals k is given by the PDF:

$$Pr(X_{estimate} = k) = {}_nC_k \cdot {p_{estimate}}^k (1 - p_{estimate})^{n-k}. \tag{4.11}$$

The median estimator chooses which of the two cases is in the majority. Therefore, only when $X_{estimate}$ is greater than or equal to $n/2$, will the median estimator choose the sample estimate that occurs with probability $p_{estimate}$. In other words, the sample estimate will become the final pixel value, only when it shows up in greater than or equal to $n/2$ pixel buffers. The probability for a sample to make a contribution to the final pixel can be modeled using the cumulative distribution function of the binomial distribution (CDF), and it is given by

$$Pr(X_{estimate} \geq n/2) = \sum_{i=\lceil n/2 \rceil}^{n} {}_nC_i {p_{estimate}}^i (1 - p_{estimate})^{n-i}. \tag{4.12}$$

By plugging Equation 4.10 into Equation 4.12, the probability for a sample with a probability p_{sample} to make a contribution to the final pixel is obtained by the equation

$$\begin{aligned} &Pr(X_{estimate} \geq n/2) = \\ &\quad \sum_{i=\lceil n/2 \rceil}^{n} {}_nC_i \left(\frac{N}{n} p_{sample} \right)^i \left(1 - \frac{N}{n} p_{sample} \right)^{n-i}. \end{aligned} \tag{4.13}$$

If p_{sample} is assumed to be the probability of outliers, Equation 4.13 can be regarded as a pixel error rate.

In Figure 4.1, Equation 4.13 is plotted for various numbers of pixel buffer sizes and the normalized sample probability Np_{sample}. It shows that a larger pixel buffer size is

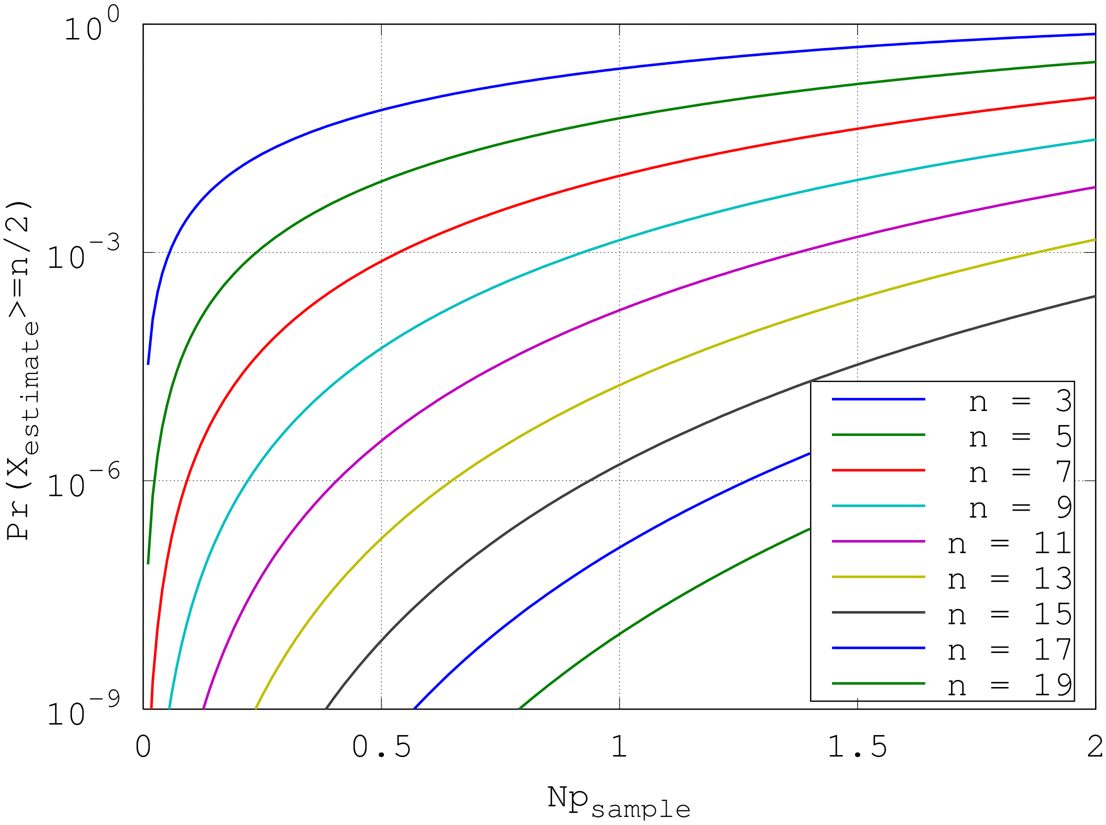


Figure 4.1: Analytic sample contribution probability ($Pr(X_{estimate} \geq n/2)$) curves of normalized sample error rate (Np_{sample}) for various sizes of pixel buffers (n), when using the median estimator.

less vulnerable to overestimation. The normalized sample probability 1.0 corresponds to the sample resolution $1/N$. If a sample with a normalized sample probability less than 1.0 contributes to the final pixel value, it will be overestimated and have high potential to become impulse noise.

From Figure 4.1, it is possible to determine the required number of pixel buffers based on the probability of a sample to make a contribution to a pixel. For example, if a rendering consists of 10^5 pixels, the probability of a sample to make a contribution to a pixel needs to be less than 10^{-5} when the normalized sample rate is less than 1.0. Using more than 15 pixel buffers can produce this result. In the same way, a rendering that consists of 10^6 pixels needs 17 pixel buffers to prevent overestimation of the samples that occur at a probability less than the sample resolution. In fact, using 17 pixel buffers can effectively decrease the overestimation rate to 10^{-7}, which covers image resolutions up to larger than 1080p (2.1 megapixels). Using 19 pixel buffers can decrease the probability of overestimation to 10^{-8}, and therefore we can expect less than a single overestimation for 10^{8} pixels.

Because this analysis is based on the median estimator, more robust estimators, such as myriad and meridian, can be expected to recover the overestimations that the median estimator fails to fix. In addition, from a practical perspective, the area that suffers impulse noise is only a part of the rendering and the error rate is different from pixel to pixel. Therefore, this analysis provides an upper boundary of the error rate of M-estimators for pixel evaluation.

We can also consider a sampling strategy with replacement by allowing the same samples to be used for the calculation of multiple sample means. This sampling strategy increases the probability with which overestimation occurs, because this strategy increases the probability $p_{estimate}$ in Equation 4.12 by using potentially erroneous samples for the calculation of multiple sample means. More robust estimators, such as the myriad and the meridian, can also reduce noise caused by error propagation.

4.2 Simulations

This section discusses the performance of the robust estimators for path-traced scenes. To evaluate the robust estimators as an alternative of the standard MC estimator, we used random sampling to initialize camera rays and excluded post image processing. The simulations were performed using an in-house developed ray tracer based on PBRT [26]. We implemented the median and meridian operations using the quick sort algorithm, and the myriad operation using the golden section search algorithm to find the extremum in Equation 4.7 [66]. In addition, we used a dynamic scale parameter based on the sample range for the myriad and meridian estimators (0.05 times the input range), throughout all of the simulations. The samples are stored as RGB values in the image buffers. Although robust estimates could be determined for each RGB channel, for efficient computation the median and meridian were computed based solely on the luminance. Because the median or meridian is one of the input samples, the RGB values can be recovered from the luminance.

4.2.1 Normalized error for image quality evaluation

The rendering quality can be measured using normalized error instead of the squared error employed for mean squared error (MSE). For quantifying the visual impression of an image, MSE would not be an appropriate metric. In other words, an image with a large MSE value can have better quality than an image with a small MSE value. For instance, relatively small bias in a large area of an image can make a significant contribution to MSE even if it has low variance; therefore, even for a large MSE value the quality of the rendering can be considered acceptable. Furthermore, impulse noise is introduced by a relatively high value in an area. Even when its absolute value is low and hardly contributes to the MSE, this type of noise is clearly noticeable in a rendered image.

The pixel error can be normalized using its reference (ground truth) pixel value, because the variance of high pixel values is usually greater than that of low ones. This is due to the fact that the outgoing radiance distribution in a pixel can be approximated by the illumination distribution scaled by the reflectance. Therefore, the normalized error of the i-th pixel E_i' can be defined by

$$E_i' = \frac{\hat{X}_i - X_i}{X_i}, \tag{4.14}$$

where $\hat{X}_i$ is the estimated pixel value, and X_i is the pixel value in a reference (ground truth) image. The mean of the pixel values can be regarded as the normalized bias, and the variance of the pixel values can be considered to be the normalized noise level.

A similar metric can be found in other papers. Rousselle et al. introduced relative MSE to evaluate their simulation results [67]. This error metric is calculated using normalized error (the error is adjusted using small ϵ to avoid division by zero caused by black reference pixel values). Instead of MSE, many recent papers used this metric to evaluate the renderings [68, 69]. It is interesting to note that relative MSE can be defined by the sum of the variance and squared mean of normalized error. If the mean of normalized error is zero, the image will be considered non-biased and the variance of the normalized error will equal the relative MSE.

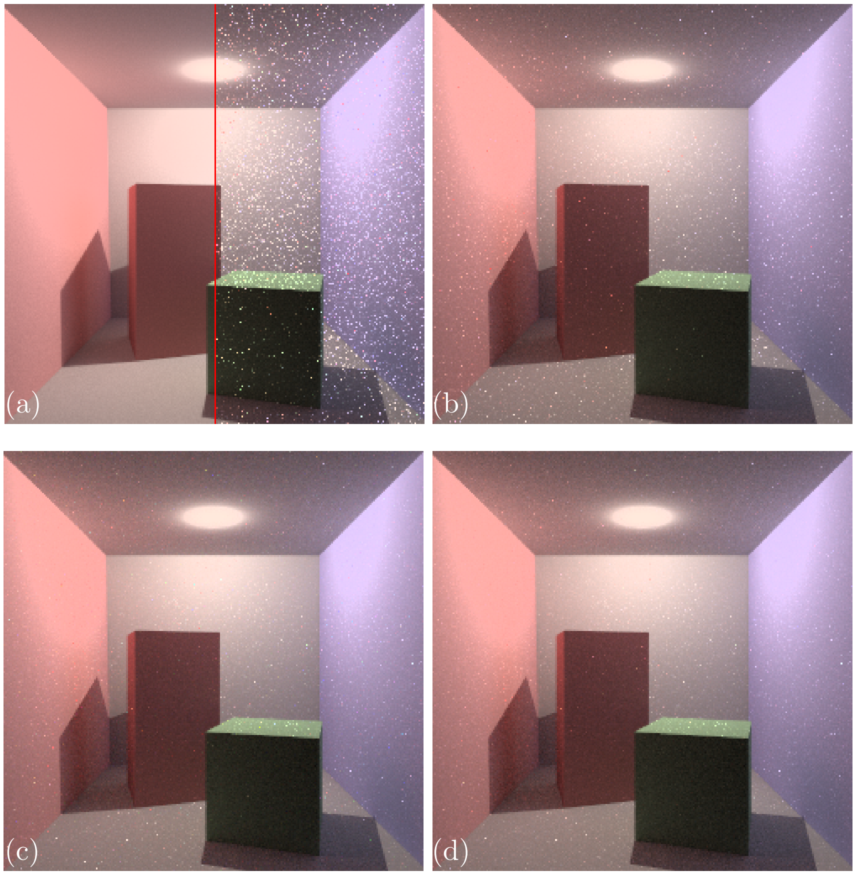
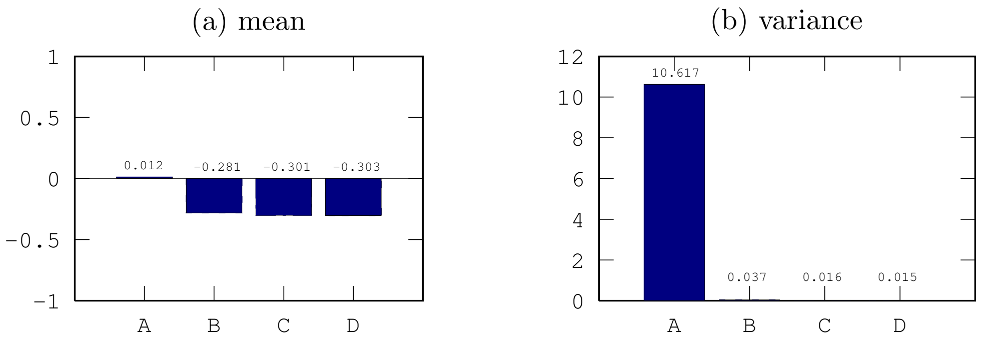


Figure 4.2: Comparison of path traced renditions for 400 samples/pixel: (a) ground truth (left) and the mean estimate (right), (b) the median estimate, (c) the myriad estimate, and (d) the meridian estimate (scene from [26]).

4.2.2 Robust estimation for various estimators

Figure 4.2 shows a set of images rendered using path tracing with 400 samples/pixel. To introduce more impulse noise throughout the entire scene, we moved the point light source to the top of the Cornell box. The small bright light reflection on the ceiling created by the point light source introduces high incident radiance within a small range of angles on the surfaces. Therefore, with importance sampling for light paths, there is a greater chance of impulse noise in the rendering.

The simulations shown in Figure 4.2 were performed using various robust estimators with 4 sample means. Figure 4.2a shows the ground truth and the mean estimate (also



(a) mean

(b) variance

Figure 4.3: Statistics of normalized error for Figure 4.2: The normalized error is A for the mean, B for the median, C for the myriad, and D for the meridian.

the standard MC estimate). The mean estimate suffers severe salt-and-pepper noise. However, as shown in Figure 4.2b, the median with 4 sample means effectively eliminated large amounts of noise. To obtain better quality rendering and remove the remaining noise, a larger sample size can be considered. However, increasing the sample size has the risk of increasing speckles, but it may also attenuate the brightness of the speckles. In this case, the myriad and meridian can be effective in removing the noise, as seen in Figure 4.2c and Figure 4.2d.

Figure 4.3 shows the mean and variance of the normalized error for Figure 4.2. The mean estimate in Figure 4.2a (right) shows the smallest mean of the normalized error but the largest variance. Therefore, it has significant amount of noise even with the smallest bias amongst the renderings. In addition, as seen in Figure 4.3a, the more robust the estimate is the more negative bias it has. However, Figure 4.3b shows that the variance of the normalized error decreases as the robustness increases. Therefore, the more robust estimates suffer from less noise as seen in Figure 4.2.

4.2.3 Robust estimation for various pixel buffer sizes

In Section 4.1.4, it was shown that the median estimate with a smaller pixel buffer size has a greater chance to introduce overestimations in a rendering than a larger pixel buffer size. Because the overestimation is manifested as impulse noise in a rendering, the number of speckles decreases as the pixel buffer size increases.

Figure 4.4 is a set of scenes with 121,500 pixels (450×270). In this scene, the specular

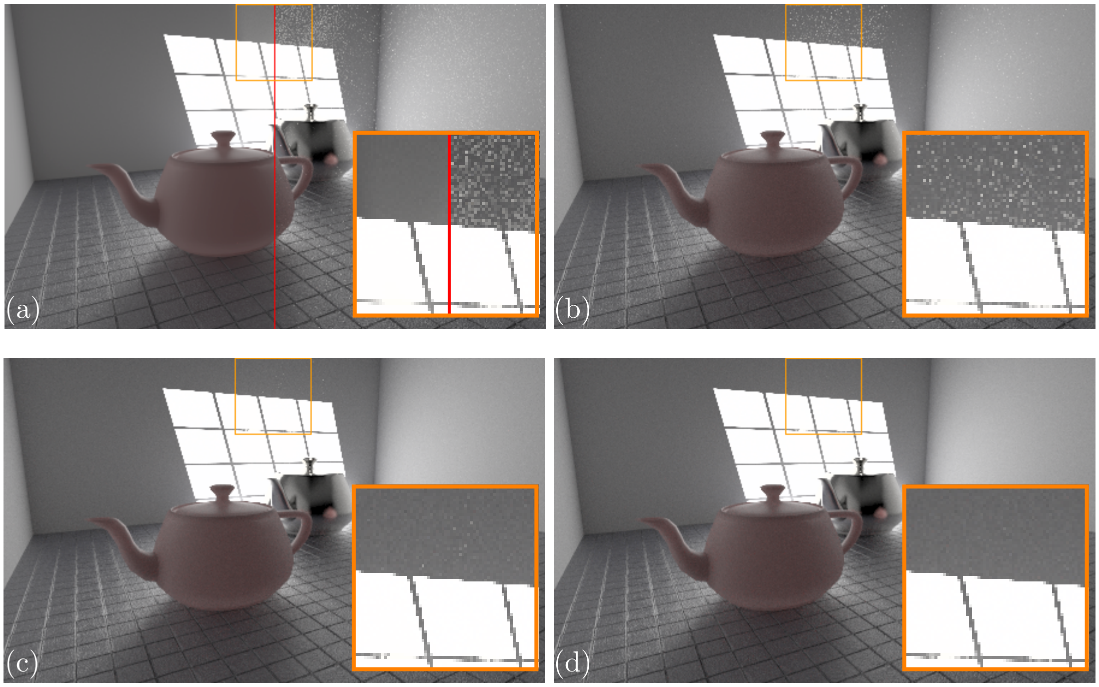


Figure 4.4: Comparison of path traced renditions for 10k samples/pixel and median estimate: (a) the ground truth/mean (left) and the closeup (right), (b) the median with 3 sample pixel buffers (left) and the closeup (right), (c) the median with 9 pixel buffers (left) and the closeup (right), and (d) the median with 15 pixel buffers (left) and the closeup (right) (scene from [26]).

reflection on the top of the middle pot becomes a small light source, and this creates bright speckles on the surfaces the reflection illuminates.

Figure 4.4a shows the ground truth picture on the left side of the red line, and the rendering produced using the mean estimator (standard MC estimator) on the right side of the red line. It also shows a closeup version of the orange rectangular area. For 10^4 samples/pixel, it contains a large quantity of speckles, which will become a slightly brighter area in the scene with a large enough number of samples. These noticeable speckles are the result of overestimation of small probability samples.

Figure 4.4b is the image estimated using the median with 3 sample means. Although the estimator removes large amounts of noise, it still leaves quite a few speckles. This is because, with a pixel buffer size of 3, the overestimation probability is not small enough to eliminate all of the speckles (see Figure 4.1).

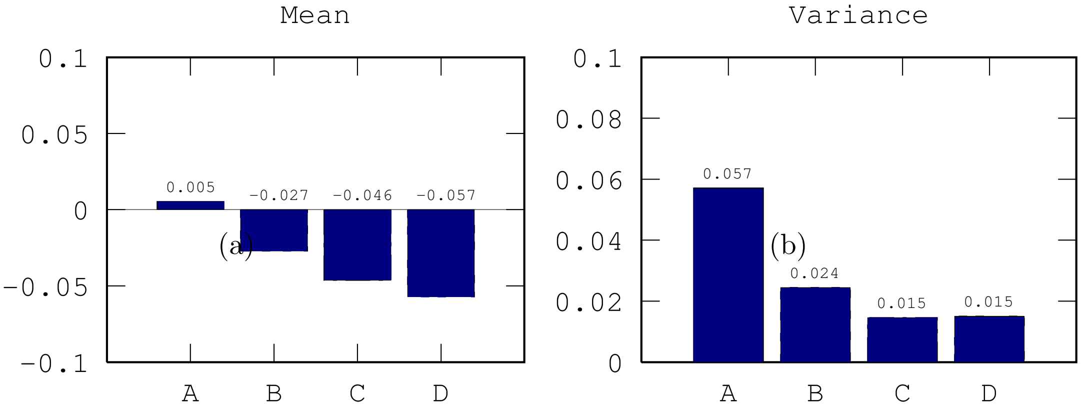


Figure 4.5: Statistics of normalized error for Figure 4.4: The normalized error is A for the mean, B for the median with 3 pixel buffers, C for the median with 9 pixel buffers, and D for the median with 15 pixel buffers.

As seen in Figure 4.4b, Figure 4.4c and Figure 4.4d, the number of speckles decreases as the pixel buffer size increases, and finally all the speckles disappear with the median estimate and 15 pixel buffers. This is because the median with 15 pixel buffers can effectively decrease the probability of overestimation below $1/121,500$ (see Figure 4.1).

Figure 4.5 shows the mean and variance of the normalized error for Figure 4.4. Like the scene discussed in Section 4.2.2, the more robust estimates have more negative bias but less noise. The variance of the median with 15 estimates is slightly higher than the median with 9 estimates. This is because more sample means decrease the sample size for each sample mean and finally the sample mean distribution becomes broader.

4.2.4 Robust estimation for adaptive pixel buffer size

Although the error analysis can specify the maximum size of pixel buffers for the sample mean distribution, in certain memory constrained situations the size may not be manageable. In many practical circumstances, not every pixel suffers impulse noise. Therefore, depending on the statistics, it is possible to apply a different number of pixel buffers to each pixel. For instance, a small number of pixel buffers can be considered for a distribution that has small variance. In addition to the variance, either the skewness, the standard deviation or the range can be used to determine the number of pixel buffers in each pixel, because these measures describe the characteristics of a distribution [70]. In this section, we introduce a heuristic approach, based on the standard deviation, for adaptive pixel buffer size.

Adaptive pixel buffer size

Instead of using the same size of pixel buffers for every pixel, we tailored the size to the variance of the pixel's sample distribution. The adaptive size of the pixel buffers is helpful not only for efficient computation, but also for reducing the bias that makes the scene darker. In our simulations, the ratio of the standard deviation of the outgoing radiance in a pixel, σ_{pixel}, to the standard deviation of the overall scene, σ_{scene}, is used to determine the pixel buffer size, n, and we found that the following decision rule works well in our test scenes.

$$n = \begin{cases} 1 & \text{for } \frac{\sigma_{pixel}}{\sigma_{scene}} < 1 \\ 2 \times (\lfloor \log_2 \frac{\sigma_{pixel}}{\sigma_{scene}} \rfloor + 1) + 1 & \text{otherwise} \end{cases} . \tag{4.15}$$

The standard deviation is a measure of how spread out the samples are, and n increases as the sample distribution is stretched out. σ_{pixel} and σ_{scene} change as the sample size increases, and therefore n varies. However, it is important to note that this is a heuristic approach, and it is also possible to have similar results with other rules based on the variance or the range.

Image quality comparison

Figure 4.6 shows the path traced rendition with 1,000 samples/pixel using various approaches. Except for the sample size, the scene is the same as the setup discussed in Section 4.2.3. For a small sample size, the probability with which speckles show up in a rendition decreases, because the overestimation of only low probability samples becomes speckles. Compared to Figure 4.4a (right side of the red line), Figure 4.6a (left side of the red line) shows fewer bright speckles. Specifying the target error rate in each sample to be 10^{-4}, the simulation used a pixel buffer size of 7 for the maximum pixel buffer size. This target error rate corresponds to the to a normalized sample probability of 0.1 in Figure 4.1. Figure 4.6a shows that the median with 7 pixel buffers (right side of the red line) is free from the noise that shows up with the standard MC estimator

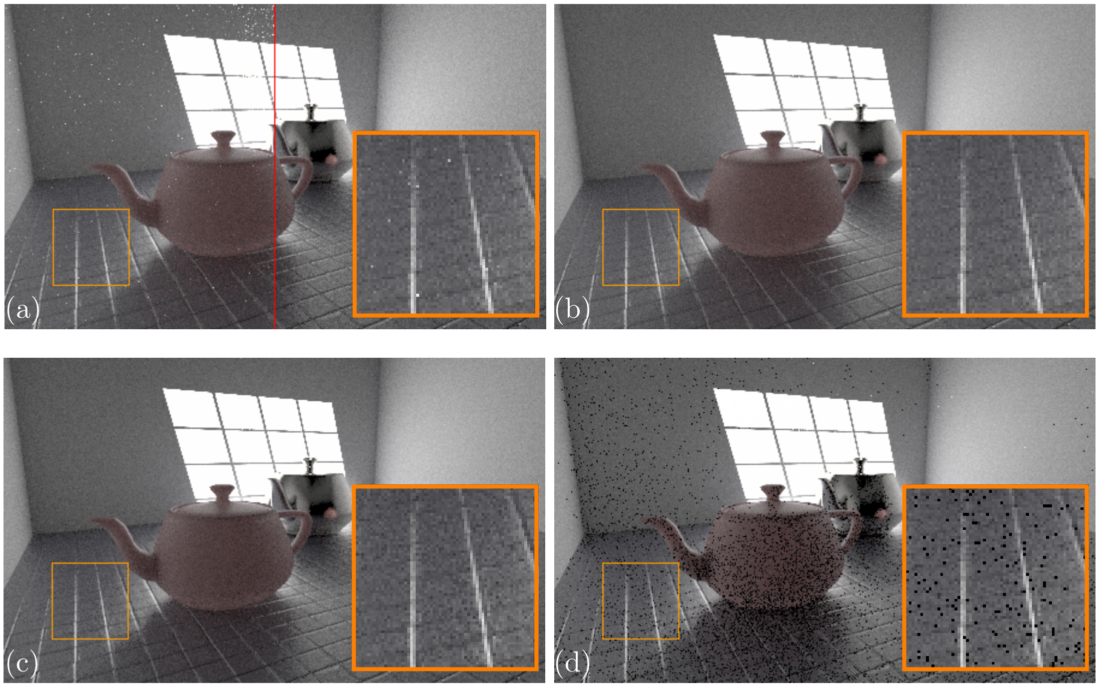


Figure 4.6: Comparison of path traced renditions for 1k samples/pixel rendered using: (a) the mean/median (7 pixel buffers) (left) and the closeup (right), (b) the median with an adaptive pixel buffer size (left) and the closeup (right), (c) ADR of Pajot et al. (left) and the closeup (right), and (d) ROT of Pajot et al. (left) and the closeup (right) (scene from [26]).

(left side of the red line). Therefore, it can be concluded that the speckles are from the overestimation of the samples that occurs at a probability less than 10^{-4}.

Figure 4.6b is the result of the robust estimate with adaptive sizing of the pixel buffers. For some regions in Figure 4.6b, the overall brightness and quality is improved over the median with 7 sample estimates in Figure 4.6a because the smaller number of estimates can produce less biased values. At the same time, the consistency could get worse because variations in the estimate sizes can cause differences in the speed at which distributions for adjacent pixels converge. However, similar estimate sizes between adjacent pixels, which have similar variances, can minimize the inconsistency. In addition, the inconsistency will be negligible as the sample size increases, because every estimate converges to the true mean.

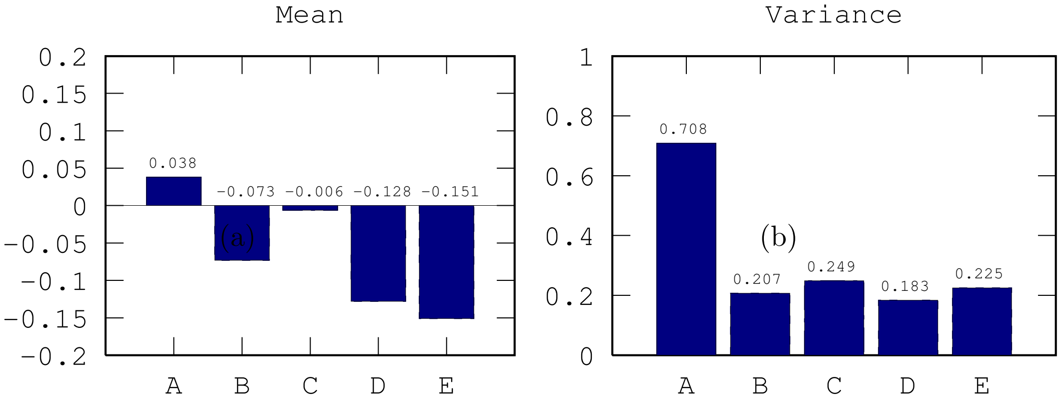


Figure 4.7: Statistics of normalized error for Figure 4.6: The normalized error is A for the mean, B for the median with 7 pixel buffers, C for the median with an adaptive pixel buffer size, D for the ADR of Pajot et al., and E for ROT of Pajot et al..

Figure 4.6c was produced with the density estimation computed using the Approximate Distribution Representation (ADR), and it shows comparable image quality to the robust estimate in Figure 4.6b [18]. However, it exhibits more dark speckles than the robust estimate, as seen in the orange rectangle of Figure 4.6c. The ADR-generated images were poor because the density estimation did not reject outliers on the lower end. In addition, failing to create higher extended modes can also introduce dark speckles. However, our approach removes not only the influence on the upper end, but also from lower end; therefore, it can limit the emergence of these dark speckles. In addition, the reference rendered using Rule-Of-Thumb bandwidth (ROT) shows worse performance than our approach and ADR [18]. Specifically, ROT not only failed in rejecting some outliers on the top of the scene, but also generates many zero pixel values, as seen in Figure 4.6d. This is because there are many zero luminance sample values in this scene, so ROT failed in creating the extended modes for a given sample size and kernel bandwidth in those pixels.

Figure 4.7 shows the statistical analysis of the images in Figure 4.6. Figure 4.7a describes the normalized bias of each image. The standard Monte-Carlo estimate is biased in a positive direction, while the robust estimates are negatively biased. The adaptive robust approach produced relatively small bias compared to the median with 7 estimates, because it uses less than 7 estimates for most pixels. On the other hand,

Median with 7 pixel buffers			
7 sample estimates	=	700.0%	
Total	=	700.0%	
Robust estimate with adaptive size			
Statistics of pixels	≈	100.0%	
1 sample estimate/pixel = 0.937	≈	0.0%	(no cost)
3 sample estimates/pixel = 0.032	≈	9.6%	(0.032×3)
5 sample estimates/pixel = 0.008	≈	4.0%	(0.008×5)
7 sample estimates/pixel = 0.023	≈	16.1%	(0.023×7)
Total	≈	130.0%	
ADR of Pajot et al.			
Single modes/pixel = 24.195	≈	806.5%	(24.195×1/3)
Extended modes/pixel = 14.621	≈	974.7%	(14.621×2/3)
Dubious samples/pixel = 2.216	≈	221.6%	
Total	≈	2000.0%	
ROT of Pajot et al.			
Single modes/pixel = 5.051	≈	168.4%	(5.051×1/3)
Extended modes/pixel = 0.937	≈	62.5%	(0.937×2/3)
Dubious samples/pixel = 5.051	≈	505.1%	
Total	≈	740.0%	

Table 4.1: Required memory (% of original image) for various approaches used in Figure 4.6.

density estimation using ADR and ROT introduced relatively high bias compared to our approach. Figure 4.7b shows the variance of normalized error for each image. Clearly, the robust estimates have less noise compared to the standard Monte-Carlo estimate. The adaptive approach produced slightly higher variance because it uses fewer pixel buffers (less robust) in many pixels than the median with 7 pixel buffers. The density estimation using ADR shows less variance than our approach but it is more biased. The variance for density estimation using ROT is underestimated here, because many values are saturated to zero and finally the normalized error distribution becomes distorted. Specifically, more than 6% of the total pixels are saturated in Figure 4.6d.

Approach	Execution time
Median with 7 pixel buffers	37 sec
Robust estimate with adaptive size	33 sec
ADR of Pajot et al.	3,674 sec
ROT of Pajot et al.	733 sec

Table 4.2: Execution time comparison for various approaches used in Figure 4.6.

Spatial complexity comparison

Table 4.1 shows the memory required for each of the approaches used in Figure 4.6. The median with 7 pixel buffers clearly needs memory space comparable to 7 extra image buffers. This is because the sample estimate has the same data structure as a pixel. The robust estimate with adaptive sizing for pixel buffers required extra memory space that is comparable to 1.3 image buffers. It is interesting to note that 93.7% of the scene did not require robust estimation. ADR required extra memory space comparable to 20.0 image buffers, and ROT required as much extra memory space as 7.4 image buffers. The hybrid approach of ADR and ROT can decrease the memory cost by using ADR in the beginning and ROT later. Still in this case, it would require more than 3 image buffers-worth of extra memory space even if the initial ADR process was successful in consolidating the extended modes. In addition, Pajot et al. confirmed that in their test scenes the approach of DeCoro et al. required more memory space to store samples than the hybrid approach of ADR and ROT [18].

Computational complexity comparison

Table 4.2 shows the execution times of the approaches in Figure 4.6. For this benchmark, we used a single process pixel estimation program for the pre-rendered data. Therefore the execution times consist of data loading time and execution time, but do not include ray-tracing time. The median with 7 pixel buffers is computed every 7 samples. This is because the median for sample means is meaningful only when the sample sizes for the sample means are the same. The robust estimation with the adaptive sample size computes the pixel values at different rates for each pixel. Specifically, the estimate is calculated every 3 samples for the 3 pixel buffers, every 5 samples for 5 pixel buffers, and every 7 samples for 7 pixel buffers. The estimate for the single pixel buffer is the mean, which does not require any extra computation compared to the standard MC estimate. On the other hand, the approaches of Pajot et al. run the estimation process every single pixel. In addition, ADR involves an expensive bandwidth estimation process [18]. ROT is a lighter process than ADR because it does not involve the expensive bandwidth computation, but it still needs to look for the extended modes for every single sample. It is important to note that it is possible for our approaches to increase the duration between each estimation process. Therefore, the execution time of our approaches can decrease further, and ultimately be minimized to a single estimation per pixel by doing the estimation process once at the end of rendering.

Robust estimation for large resolution image

Figure 4.8 shows simulations for a large resolution image and this scene suffers from a fat-tailed distribution of outgoing radiance. To make the original scene brighter, a sunset environment map is used and the intensities of the lamps in the scene are adjusted. The scene consists of 720×600=432,000 pixels, so the maximum number of pixel buffers was set to 17. As described in Section 4.1.4, a pixel buffer size of 17 can effectively decrease the probability of the overestimation down to below than 1/432,000. The same decision rule was used to determine the number of sample estimates. In a pixel, when the measure, $\sigma_{pixel}/\sigma_{scene}$, required more than 17 sample estimates, the meridian estimator was used for simplicity. Compared to the images rendered with the standard MC estimator (in the upper row of Figure 4.8), the renditions that used the robust estimators (in the lower row of Figure 4.8) show how the pixel values converge without impulse noise.

4.3 Discussion

We have proposed a pixel estimation technique as a replacement for the standard MC estimate. The proposed approach consists of two key parts: resampling and robust estimation. Through resampling, the sample mean distribution is reconstructed, and the robust estimators are then applied. Our error analysis offers a mathematically concrete guide to choose the necessary number of image buffers based on the image size. This guidance is not available in the related works of DeCoro et al. and Pajot et al.

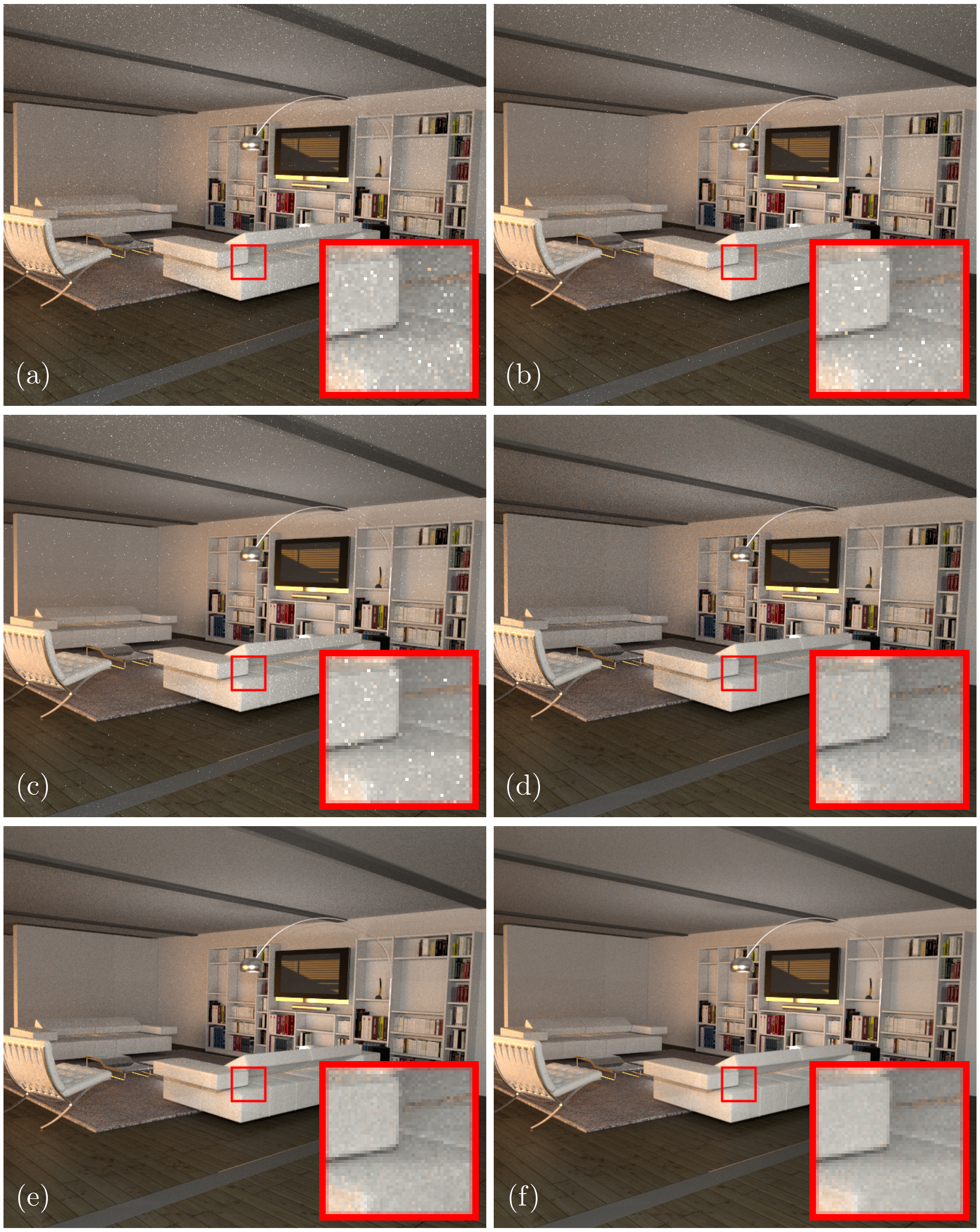


Figure 4.8: Path traced rendition with the standard MC estimator (upper) and robust estimators (lower). The images are rendered using: (a) the mean with 5k samples/pixel, (b) the mean with 10k samples/pixel, (c) the mean with 20k samples/pixel, (d) the robust estimate with 5k samples/pixel, (e) the robust estimate with 10k samples/pixel, and (f) the robust estimate with 20k samples/pixel (scene from [26]).

[17, 18].

The computational efficiency of our approach comes partly from the simplicity of the estimates. It is also the result of the fact that a smaller number of samples is required to compute the estimate, and because the frequency with which this estimate is computed can be varied. In an extreme case, we can do the estimating process just once at the end of a rendering process. By specifying the maximum number of pixel buffers, the memory requirement of our approach is bounded. In addition, by tracking the statistics in each pixel and adjusting the pixel buffer size, it becomes possible to achieve better memory efficiency.

Our approach has two limitations: bias and the potentially high variance for a normal distribution. Like the previous approaches, our method can suffer bias for a small sample size. This is because the sample mean distribution would be skewed for a small sample size. For a large sample size, the sample mean distribution becomes symmetric according to the central limit theorem and the bias will disappear. Although the robust estimates have smaller variance for heavy-tailed distributions than the standard Monte Carlo estimate, they can have higher variance for a Gaussian distribution. This is because the sample mean is asymptotically more efficient than the sample median for a Gaussian distribution. The sample mean distribution becomes Gaussian for a large number of samples. Moreover, when the underlying population is Gaussian, the sample mean distribution quickly becomes Gaussian. However, for a large sample size, the difference between the variances of the robust estimates and the mean converges to zero. In addition, a Gaussian underlying population would not be a common case in complex scenes.

The proposed approach can be used with a variety of strategies. In a memory constrained situation, it may be advantageous to use a small number of pixel buffers with more robust estimators, such as the myriad or the meridian. The median with a relatively large pixel buffer size can deal with limited processing power. While the median is used to evaluate pixels in an interactive process, the more robust estimators can be considered as part of post-processing with the user adjustable scale parameter. Even more, it is also possible to use adaptively sized pixel buffers to make the estimation more memory efficient with the maximum pixel buffer size determined by the error analysis. By using our approach in the context of various image filters and adaptive sampling strategies (for example as preprocessing) better quality rendering can also be expected [19].

Chapter 5

Hybrid Approaches for Robust Pixel Estimation and Image Filtering

Monte Carlo (MC) ray tracing is an effective method for rendering realistic three-dimensional (3D) scenes, but for a small sample size it suffers from noise. Noise-free renditions require large sample sizes because the sample mean radiance distribution in each pixel has high variance. Although the standard MC estimate or mean is best described with a Gaussian distribution, for a small sample size the sample distribution or sample mean distribution of radiance in a pixel is rarely expected to be Gaussian [52, 17, 18, 71]. This is partly because the lighting environment for a surface point is ill-conditioned for a given scene setup. It is also because of the specific weakness of rendering approaches associated with noise, such as importance sampling in path tracing.

One of the characteristics observed from pixel sample mean radiance distributions is fat-tailedness [17, 18]. The fat-tailed distribution of pixel radiance can introduce spike noise in a rendition because of the low probability samples that have excessively large values, as can be seen in Figure 5.10a. Traditional image filters are not effective for eliminating this type of noise. For instance, a median image filter fails for a pixel with a filtering window that includes more than 50% corrupted pixels (Figure 5.10b). Other filters, such as Gaussian and box filters are expected to be worse, because they allow the noise to propagate into neighboring pixels. In addition, a bilateral image filter shows poorer performance than others, because of its edge-preserving characteristic [17].

To address this kind of impulse noise, DeCoro et al. use a density-based outlier rejection algorithm [17], Pajot et al. suggest a kernel-density-based outlier rejection algorithm [18], and Jung et al. introduce robust statistical pixel estimation techniques [71]. Although these approaches show comparable performance, robust statistical pixel estimation is computationally the most efficient algorithm because it is independent of sampling, and it provides a mathematical framework to choose a robustness level [18, 71]. In addition, as an alternative to the standard MC estimator, robust pixel estimation is expected to produce better results when combined with image filtering [71].

This chapter discusses possible combinations of robust pixel estimation and image filtering for noise removal, and proposes two effective yet computationally efficient robust estimation techniques: robust image filtering over robust pixel estimation (Figure 5.10c) and 3D robust filtering over pixel buffers (Figure 5.10d). In addition, it provides a mathematical analysis of the approaches and compares their performance for path-traced images.

Due to the effectiveness of our noise removal approaches, a substantial speedup in rendering can be achieved by reducing the number of samples required for a desired image quality. The analysis presented in this chapter can be used to determine the pixel buffer and filtering window size necessary. It also provides a minimum sample size necessary to meet a certain requirement, which can be an answer to the question of lower bounds on the sample size in ray-tracing [72]. In addition, our experience suggests that there is value in retaining a certain amount of pixel radiance distribution information in an intermediate image format between rendering and post-processing. As is true when working with traditional image filters, exploring variations in the approach can lead to more visually pleasing results.

The remainder of this chapter is divided into six sections. Section 5.1 reviews the existing robust pixel estimation and robust image filtering techniques. Section 5.2 discusses possible hybrid approach scenarios, and introduces robust image filtering over robust pixel estimation and 3D robust filtering over pixel buffers. Section 5.3 presents the performance and spatial/computational cost analysis of the approaches discussed in Section 5.1 and Section 5.2. Section 5.4 presents simulation results. Section 5.5 discusses the potential benefits of the approaches and analysis presented in this chapter.

5.1 Robust Pixel Estimation and Robust Image Filtering

This section discusses existing robust pixel estimation and robust image filtering techniques, and presents their analytical performance and spatial/computational cost using the median estimator.

5.1.1 Robust Pixel Estimation

Robust statistical pixel estimation techniques use the sample mean distribution of radiance in a pixel to determine the pixel value [71]. The sample mean distribution is constructed by interleaving and averaging incoming samples, and by storing them into pixel buffers. As the sample size increases, the sample mean distribution becomes Gaussian by the central limit theorem. Using M-estimates as final pixel values suppresses the emergence of low probability samples, and produces less noisy images. Due to the right-skewness of the sample mean distribution that is commonly observed in renderings, the robust estimates can be negatively biased relative to the standard MC estimates. However, because the M-estimates are the same for Gaussian, robust pixel estimation makes sure that the pixel estimates converge to the correct (unbiased and zero-variance) values. Jung et al. provide the performance boundary of the M-estimator for different pixel buffer sizes using the median estimator.

Although this approach guarantees less than a single pixel error for a given image size, it may not be optimal. This is because different individuals can have different visual sensitivities for noise and bias. Therefore, in a certain situation, a trade off between noise and bias may be necessary. Furthermore, the median (or another M-estimator) may not be optimal for a certain underlying population. For instance, if the underlying population is Gaussian, the mean is asymptotically more efficient than the median [60, 49]. For a Cauchy distribution, the trimmed mean with the breakdown point of 12% has asymptotic minimized sampling variance [57].

The robustness of the estimation can be controlled by varying the pixel buffer size or by using other estimators. In particular, the myriad and meridian can produce more robust renditions than the median [51, 71]. On the other hand, for a fixed pixel buffer size, various combinations of the pixel buffers can produce a smaller pixel buffer size, and other estimators can then be used.

Instead of the M-estimators, the trimmed mean can be used as the final pixel value [57]. For a breakdown point α, a pixel buffer size m and a set of sorted sample radiance means $\mathbf{x} = \{x_i|_{i=1}^{m}\}$, the trimmed mean pixel estimate (TMPE) is defined by [73]

$$\begin{gathered} \mathrm{TMPE}(\alpha, m) = \frac{1}{m - 2\alpha m} \sum_{i=1+\alpha m}^{m-\alpha m} x_i, \\ \text{where } \alpha \in \{i/m|_{i=0}^{\lfloor (m-1)/2 \rfloor}\}. \end{gathered} \tag{5.1}$$

The robustness of the estimation depends on α. As α increases, we can obtain more robust estimates. For maximum α the median pixel estimate (MPE) is defined by

$$\mathrm{MPE}(m) = \mathrm{TMPE}(\lfloor (m-1)/2 \rfloor / m, m). \tag{5.2}$$

The median is the most robust trimmed mean; as m goes to infinity, the maximum α converges to 0.5 [73]. In addition, $\mathrm{TMPE}(0, m)$ is the standard MC estimate (mean). Therefore, the robustness of TMPE is upper-bounded by MPE and lower-bounded by the standard MC estimate. Compared to MPE, TMPE causes negligible extra computational cost due to the additional operations required to add more sample radiance means, but adds no extra cost for memory space.

Error rate: For the performance of MPE and the upper performance boundary of TMPE, the analytical error rate is given by

$$\begin{aligned} p|_{\mathrm{MPE}(m)} &= Pr(X_{estimate} \geq m/2) \\ &= \sum_{i=\lceil m/2 \rceil}^{m} {}_mC_i \left(\frac{N}{m} p_{sample}\right)^i \left(1 - \frac{N}{m} p_{sample}\right)^{m-i}, \end{aligned} \tag{5.3}$$

where $X_{estimate}$ is the number of erroneous sample estimates, N is the sample size, p_{sample} is the sample error rate, and ${}_mC_i$ is a binomial coefficient for the pixel buffer size of m and i [71]. To simplify the subsequent discussion, we assume that p_{sample} is the sample error rate due to impulse noise. Because the sample value is unknown, Jung et al. considers a more general case by assuming that p_{sample} is the sample probability; not all the low probability samples introduce impulse noise [71].

Because Equation 5.3 is also a cumulative distribution function of the inclusion rate for incoming samples, it is possible to find a sample error and estimate error rate for a pixel buffer size. In particular, in order to achieve specific sample error rates to discard and include impulses in the resulting image, a pixel buffer size is determined. Because any impulse occurring with a probability less than $1/N$ can be over-estimated for standard MC estimation, a sample error rate less than $1/N$ can be defined as a threshold to discard the impulse. On the other hand, a sample error rate greater than $1/N$ can be defined as a threshold probability beyond which impulses are included in the rendering. The estimate error rate is then the inclusion rate of the impulse, which is not considered to be noise because the impulse is expected to be included in every pixel. If any of the pixel buffer sizes and inclusion rates of impulses is given, it is also possible to find a satisfying sample size. In other words, we can specify a minimum sample size for a sample probability and inclusion rate to the rendering.

This type of analysis can be found in papers from other fields. For instance, Yin et al. describes the variance analysis for a median filter [49]. In addition, using a binomial distribution, Burkhard et al. analyzes the failure rate of a RAID system [74], and Naghshineh et al. derives the probability of a number of phone calls in a network cell [75].

Spatial complexity: The robust statistical pixel estimation techniques require m extra image buffers to track the sample mean distribution of radiance for pixels. Therefore, the spatial cost I of MPE is defined by the number of extra image buffers

$$I|_{\mathrm{MPE}(m)} = m. \tag{5.4}$$

If an off-line rendering software program reconstructs an image pixel by pixel, it will need only m pixel buffers instead of m image buffers, because the robust pixel estimation does not require any neighboring pixel information. In addition, because different pixels can have different distribution shapes, the necessary pixel buffer sizes for those pixels can be different from each other. Therefore, it is possible to optimize the pixel buffer size for each pixel. Jung et al. suggest a heuristic based on the pixel's standard deviation; other heuristics based on the pixel's variance or range are also possible [71]. Note that for a simplified analysis, throughout the discussion we ignore any temporal memory space required to compute estimates.

Computational complexity: The computational cost of MPE is dependent on the pixel buffer size. Any sorting algorithm can be applied to the m sample means. In this discussion, we consider merge sort, which is one of the common software implementations of median filters [76]. The computational complexity of merge sort for the median pixel estimation is $O(m \log m)$ in the average and worst cases [77]. Therefore, the computational cost per pixel T for MPE is defined by the number of comparisons

$$T|_{\text{MPE}(m)} = m \log m. \tag{5.5}$$

For quantized pixel values, $O(m)$ and $O(1)$ algorithms are also available [78, 79]. However, the direct adaptation of these algorithms is impossible. This is because the renderings discussed in this thesis are high dynamic range and high precision color images that consist of floating point pixel values.

5.1.2 Robust Image Filtering

In computer graphics and image processing, image filters are commonly used to suppress noise or to produce visually appealing images. In particular, robust image filters, such as the median and myriad filters, have been proposed to remove outlying pixels or impulse noise within an image [49, 50]. It is also possible to design a meridian image filter in order to obtain more robust results. These robust non-linear type image filters demonstrate their effectiveness in many circumstances, but they potentially remove important image details by blurring the rendition. The robustness of these image filters can be adjusted by re-sizing the filtering window or by adjusting the weights. As the window size increases, the filtered pixel values become more robust but blurred within an image. As the weights near the central pixel that needs to be estimated increase, the filtered pixel value becomes less robust but it can preserve image details.

For a set of integer weights $\mathbf{w} = \{w_i|_{i=1}^{m}\}$ and a filtering window size of $n = k \times l$, the weighted median image filtered (WMIF) pixel estimate of a set of pixel values

$\mathbf{x} = \{x_i|_{i=1}^n\}$ is defined by [49]

$$\begin{aligned}
\text{WMIF}(\mathbf{w}, n) = \frac{1}{s - 2\alpha s} \sum_{i=1+\alpha s}^{s-\alpha s} y'_i, \\
\text{where } \alpha = \lfloor (s-1)/2 \rfloor, \\
\mathbf{y}' = \text{sort}(\mathbf{y}) \\
\mathbf{y} = \{w_i \diamond x_i|_{i=1}^n\}, \\
s = \sum_{i=1}^n w_i,
\end{aligned} \tag{5.6}$$

and $\diamond$ denotes duplication, i.e.,

$$w_i \diamond x_i = \{\overbrace{x_i, x_i, ..., x_i}^{w_i \text{ times}}\}. \tag{5.7}$$

The median image filter (MIF) is a special case of WMIF, and for the **w** of all ones (or all the same values) it is defined by

$$\text{MIF}(n) = \text{WMIF}(1, n). \tag{5.8}$$

Compared to MIF, WMIF causes relatively small extra computational cost (n/2 multiplications and comparisons on average) to find the weighted median [49]. In addition, WMIF trades off robustness in exchange for image-detail preserving capabilities; therefore, its performance is bounded by MIF. The performance of MIF is also the lower boundary of the performance of image filters that employ M-estimators.

It is interesting to note that weighted trimmed mean image filtering (WTMIF) is also a possible solution. In this case, WTMIF with the breakdown point of 0% is a weighted mean filtering, which can be Gaussian, bilateral, or Non-local mean (NLM) depending on the employed sample weights. Therefore, the existing weighted-mean-based solutions can be regarded as special types of WTMIF (breakdown point of 0%), and their robustness can be improved by increasing the breakdown levels.

Error rate: Image filters utilize the similarity between adjacent pixels. Therefore, if every pixel has an identical sample distribution, the analytical pixel error rate can be computed. Similar to the pixel error rate defined for MPE, for MIF with a window size of $n = k \times l$, the expected pixel error rate in the final image is defined by

$$\begin{aligned} p|_{\mathrm{MIF}(n)} &= Pr(X_{pixel} \geq n/2) \\ &= \sum_{i=\lceil n/2 \rceil}^{n} {}_nC_i {p_{pixel}}^i (1 - p_{pixel})^{n-i}, \end{aligned} \tag{5.9}$$

where X_{pixel} is the number of erroneous pixels, and p_{pixel} is the pixel error rate [49]. For a sample size of N, if it is assumed that the increase of N does not attenuate the impulse noise, p_{pixel} increases as N increases. Thus, p_{pixel} is defined by the normalized pixel error rate of

$$p_{pixel} = Np_{sample}, \tag{5.10}$$

where p_{sample} is the sample error rate.

For n equal to infinity, the expected error rate is

$$p|_{\mathrm{MIF}(\infty)} = \begin{cases} 0 & \text{for } x < 0.5 \\ 1 & \text{for } x \geq 0.5 \end{cases}. \tag{5.11}$$

This is because the binomial distribution is normal for a large n and finally approximates a Dirac delta function as n approaches infinity [80].

Spatial complexity: MIF usually requires an extra single image buffer, because it needs to keep the original image from which the final image is estimated. Therefore, the spatial cost of MIF is the same as that of standard MC estimation or robust estimation with a single pixel buffer, which is ready for post image processing. Therefore, the spatial cost I of MIF is defined by

$$I|_{\mathrm{MIF}(n)} = 1. \tag{5.12}$$

Computational complexity: In general, the computational complexity of a median filter is $O(n \log n)$, where $n = k \times l$ is the filtering window size of MIF [79]. Therefore, the computational cost per pixel T for MIF is defined by

$$T|_{\mathrm{MIF}(n)} = n \log n. \tag{5.13}$$

5.2 Robust Pixel Estimation + Robust Image Filtering

Robust pixel estimation techniques can be used with various image filters in different application scenarios. However, those rendering circumstances have not yet been explored. In this section, we discuss possible combinations of robust pixel estimation and robust image filtering, and introduce two effective approaches.

First, to construct the pixel radiance distribution for robust pixel estimation, we can use pixel samples provided by a robust image filter. Although this approach can eliminate very low probability samples ($p_{sample} \ll 1/N$), it can fail for relatively high probability samples ($0 \ll p_{sample} < 1/N$) that still potentially introduce impulses in a rendering. This is because it can feed a robust pixel estimation process with corrupted pixels caused by the error propagation of image filtering. In addition, the costs of an expensive image filtering process are multiplied by the number of image buffers.

Second, we can use robust estimates as the input for a robust image filter. In this approach, the robust pixel estimation process removes noise within a pixel, and then the robust image filter eliminates the remaining noise in image space. Due to the fact that the expensive image filtering process runs only once at the end of the estimation process, this approach is more efficient than others.

Third, we can use both approaches at the same time. Instead of eliminating noise in each separate stage, we can combine these two stages by building a 3D pixel space, and then apply a robust estimator. Specifically, this method is expected to show better performance than the second approach, because the number of sample radiance means to be evaluated for the single estimation process is a product of the number of sample radiance means for the two separate stages.

Therefore, based on the last two approaches discussed above, we propose two effective robust image reconstruction methods; one is robust image filtering over robust pixel estimation, and the other is 3D robust filtering over image buffers. In the following subsections, we discuss these two approaches, and provide an analysis of their performance and computational/spatial costs using the median estimator.

5.2.1 Robust Image Filtering over Robust Pixel Estimation

The simplest form of the hybrid approach would be to apply a conventional robust image filter to the result of robust pixel estimation. Each estimation stage can control the robustness of the estimation independently. For each part, the approaches that are described in Section 5.1.1 and Section 5.1.2 can be used, and their results and side-effects are as expected.

For a pixel buffer size m, filtering window size $n = k \times l$, and set of TMPE $\{x_i = \text{TMPE}_i(\alpha, m)|_{i=1}^{n}\}$, the weighted median filtered estimate is defined by

$$\begin{aligned} \text{TMPE}(\alpha, m) + \text{WMIF}(\mathbf{w}, n) &= \frac{1}{s - 2\alpha' s} \sum_{i=1+\alpha' s}^{s-\alpha' s} y'_i, \\ \text{where } \alpha' &= \lfloor (s-1)/2 \rfloor, \\ \mathbf{y}' &= \text{sort}(\mathbf{y}) \\ \mathbf{y} &= \{w_i \diamond x_i|_{i=1}^{m}\}, \\ s &= \sum_{i=1}^{m} w_i, \end{aligned} \tag{5.14}$$

For different robustness of results, TMPE provides $\lfloor (m+1)/2 \rfloor$ breakdown levels, and there are multiple combinations of $\mathbf{w}$.

As a special case of TMPE+WMIF, for the maximum α and the $\mathbf{w}$ of all ones, MPE+MIF is defined by

$$\begin{aligned} &\text{MPE}(m) + \text{MIF}(n) = \\ &\quad \text{TMPE}(\lfloor (m-1)/2 \rfloor / m, m) + \text{WMIF}(1, n). \end{aligned} \tag{5.15}$$

The performance of MPE+MIF is the upper boundary of TMPE+WMIF.

Error rate: The analytical error rate can be obtained by replacing p_{pixel} in Equation 5.9 with the median pixel estimator's error rate $p|_{\text{MPE}(m)}$ in Equation 5.3. For the pixel buffer size of m and the filtering window size of $n = k \times l$, the error rate of MPE+MIF is defined by

$$\begin{aligned} p|_{\text{MPE}(m)+\text{MIF}(n)} &= Pr(X_{\text{MPE}} \geq n/2) \\ &= \sum_{i=\lceil n/2 \rceil}^{n} {}_nC_i p|_{\text{MPE}(m)}{}^{i} (1 - p|_{\text{MPE}(m)})^{n-i}, \end{aligned} \tag{5.16}$$

where X_{MPE} is the number of erroneous median pixel estimates.

Spatial complexity: The necessary extra memory for this approach is totally dependent on the image buffer size required by MPE. Therefore, the spatial cost I of MPE+MIF is defined by

$$I|_{\text{MPE}(m)+\text{MIF}(n)} = I|_{\text{MPE}(m)} = m. \tag{5.17}$$

If a rendering system synthesizes an image pixel by pixel instead of image by image, the required extra memory space can be only m pixel buffers.

Computational complexity: The computational complexity of this approach is the sum of the costs of MPE and MIF. Because MPE requires $O(m \log m)$ operations and MIF needs $O(n \log n)$ operations, the total computational cost T is defined by

$$\begin{aligned} T_{\text{MPE}(m)+\text{MIF}(n)} &= T|_{\text{MPE}(m)} + T|_{\text{MIF}(n)} \\ &= n \log n + m \log m. \end{aligned} \tag{5.18}$$

5.2.2 3D Robust Filtering over Pixel Buffers

For better quality image synthesis, it is possible to include the radiance distribution in the adjacent pixels into a single estimation process. Specifically, robust pixel estimation with the pixel buffer size of m or robust image filtering with the filtering window size of $n = k \times l$ can be expanded to 3D robust filtering with the 3D window size of $m \times k \times l$.

The robustness of 3D robust filtering can be controlled by adjusting the 3D filtering window size of $m \times k \times l$ or using a different estimator. As the 3D filtering window size grows, more robust estimates are obtained. However, the window expansion in image space can produce image blurring, and the increase of the pixel buffer size can introduce image bias. As a different estimator, any estimator discussed in Section 5.1.1 and Section 5.1.2 can be used. In addition, the results and side-effects that are discussed in Section 5.1.1 and Section 5.1.2 are expected at the same time. This is because the robustness is adjusted simultaneously in the image space and pixel radiance distribution.

For a pixel buffer size of m, filtering window size of $n = k \times l$, set of integer weights

$\mathbf{w}$, and set of sample radiance means $\mathbf{x} = \{x_i|_{i=1}^{m\times n}\}$ in the 3D space of $m \times n$, the three-dimensional weighted trimmed mean filtered (3DWTMF) pixel estimate is defined by

$$\begin{aligned} \text{3DWTMF}(\alpha, \mathbf{w}, m \times n) = \frac{1}{s - 2\alpha s} \sum_{i=1+\alpha s}^{s-\alpha s} y'_i, & \\ \text{where } \alpha \in \{i/s|_{i=0}^{\lfloor (s-1)/2 \rfloor}\}, & \\ \mathbf{y}' = \text{sort}(\mathbf{y}) & \\ \mathbf{y} = \{w_i \diamond x_i|_{i=1}^{m\times n}\}, & \\ s = \sum_{i=1}^{m\times n} w_i, & \end{aligned} \tag{5.19}$$

Compared to TMPE+WMIF, 3DWTMF offers more options for robustness control. For instance, $\lfloor (m \times n + 1)/2 \rfloor$ levels are available for α. Although there are $m \times n$ adjustable weights, the number of effective weights is $n = k \times l$. This is because it is meaningless to apply different weights to sample radiance means for the same pixel; those sample radiance means are calculated from samples taken from the same probability distribution.

As a special case of 3DWTMF, for the maximum of α, the three-dimensional weighted median filtered (3DWMF) pixel estimate is defined by

$$\begin{aligned} &\text{3DWMF}(\mathbf{w}, m \times n) = \\ &\qquad \text{3DWTMF}(\lfloor (m \times n - 1)/2 \rfloor, \mathbf{w}, m \times n). \end{aligned} \tag{5.20}$$

For $\mathbf{w}$ with all the same values (ones), 3DWMF becomes the three-dimensional median filtered (3DMF) pixel estimate that is defined by

$$\text{3DMF}(m \times n) = \text{3DWMF}(1, m \times n). \tag{5.21}$$

Furthermore, for an infinite weight for the pixel that needs to be estimated, 3DWTMF becomes MPE. Therefore, the upper bound of 3DWTMF is that of 3DMF and the lower one is that of MPE.

Note that 3DWTMF become MPE for n of 1, WMIF for m of 1, and WTMIF for α of 0 and m of 1. In other words, the approaches discussed in this chapter are special cases or combinations of 3DWTMF.

Error rate: In this approach, we look for the number of correct sample estimates out of $m \times n$ sample estimates, instead of m sample estimates as in MPE or n sample estimates as in MIF. Consequently, we can expect better error recovery performance. The analytical error rate for $m \times n$ sample estimates is defined by

$$\begin{aligned} p|_{\mathrm{3DMF}(m\times n)} &= Pr(X_{estimate} \geq m \times n/2) \\ &= \sum_{i=\lceil m\times n/2 \rceil}^{m\times n} {}_{m\times n}C_i \left(\frac{N}{m} p_{sample}\right)^i \left(1 - \frac{N}{m} p_{sample}\right)^{m\times n - i}, \end{aligned} \tag{5.22}$$

where N is the sample size, m is the pixel buffer size, n is the two-dimensional (2D) filtering window size, and p_{sample} is the sample error rate.
Spatial complexity: The extra memory required to perform 3DMF is m extra image buffers. Therefore, the spatial cost I of 3DMF is defined by

$$I|_{\mathrm{3DMF}(m\times n)} = I|_{\mathrm{MPE}(m)} = m. \tag{5.23}$$

Unfortunately, this approach cannot be performed pixel by pixel like MPE or MPE+MIF, because the filtering process requires information from neighboring pixels. However, optimization using area-by-area reconstruction is possible [78, 79].
Computational complexity: The computational complexity per pixel is dependent on the pixel buffer size of m, and the 2D filtering window size of n. The computational complexity to re-order pixel buffers is $O(m \log m)$, and the complexity for the median estimation with the set of $m \times n$ separately sorted pixel buffers for n pixels is $O(m \times n \cdot \log n)$. If the pixel buffers were not sorted, the complexity for the median estimation would be $O(m\times n \cdot \log{(m \times n)})$. Therefore, the computational cost per pixel T is defined by

$$T|_{\mathrm{3DMF}(m\times n)} = m \log m + m \times n \cdot \log n. \tag{5.24}$$

5.3 Analytical Performance Comparison

Using the median estimator, we discuss the analytical error correcting performance, spatial costs, and computational costs of the approaches presented in Section 5.1 and Section 5.2. The median estimator is the most robust trimmed mean estimator, so the following analysis serves as a performance boundary and presents spatial/computational complexities for limiting cases.

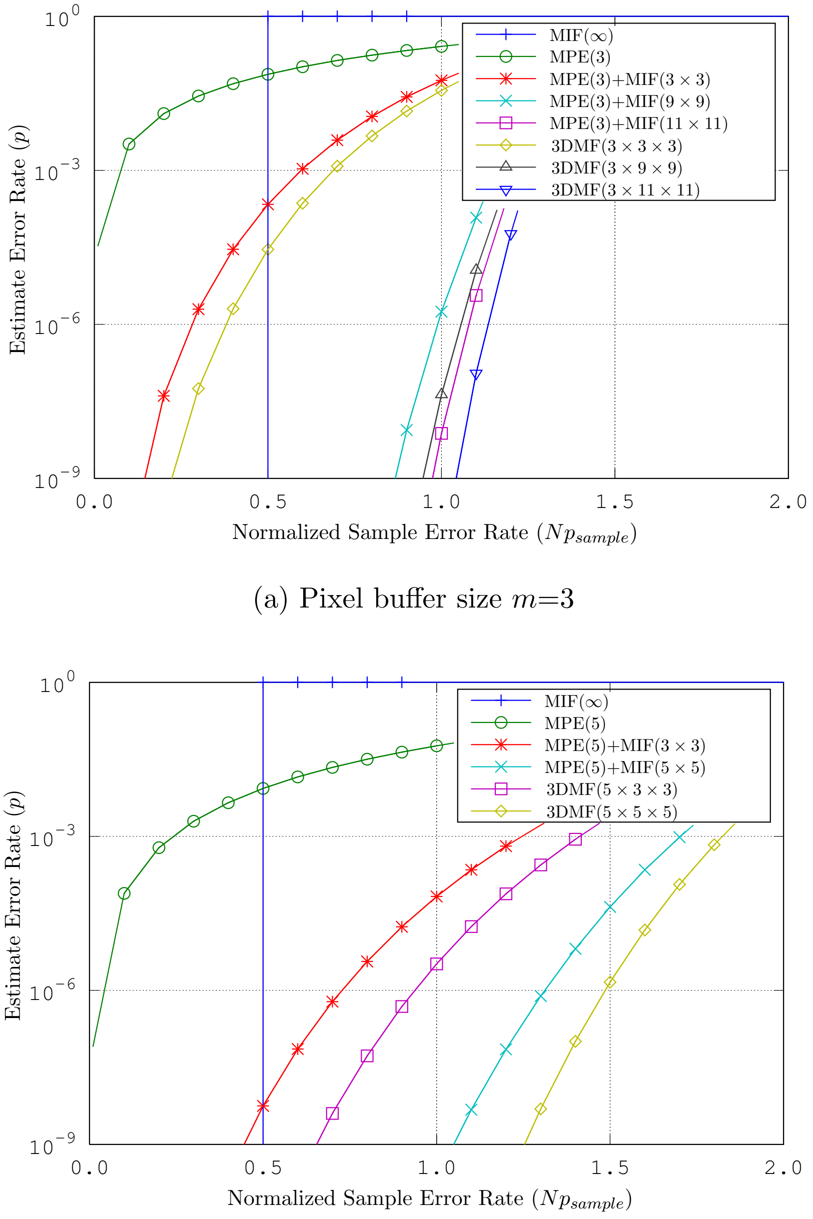
5.3.1 Error Rate

Figure 5.1 shows the analytical error rate of various estimation techniques for the pixel buffer size of 3, 5, and 7. Note that we are interested only in the range where the normalized sample error Np_{sample} is less than 1; if the normalized error rate is greater than 1, the noise will corrupt every pixel, and it will no longer be considered to be impulse noise.

The error correcting ability of MIF increases as the filtering window size expands, and finally becomes bounded by MIF(∞). Although MIF(∞) effectively eliminates the low probability samples in the range between 0 and 0.5, it overestimates relatively high probability samples in the range between 0.5 and 1. Therefore, the error can propagate into neighboring pixels. Furthermore, as the window size grows, the chance of image distortion and blurriness increases. On the other hand, robust pixel estimation can generate renderings without image distortion. However, the resulting images can be relatively biased towards zero, because the low probability samples are concentrated in the range of high values; as the sample size increases, the discarded low probability samples are included in the final pixel values and the resulting images eventually converge to the true mean.

Taking into account the outgoing radiance distribution in neighboring pixels, it is possible to reduce the necessary pixel buffer size of robust pixel estimation and the potential bias of resulting images. For the same pixel buffer size of MPE and filtering window size of MIF, 3DMF shows the best performance, followed by MPE+MIF, and MPE has the worst performance as seen in Figure 5.1.

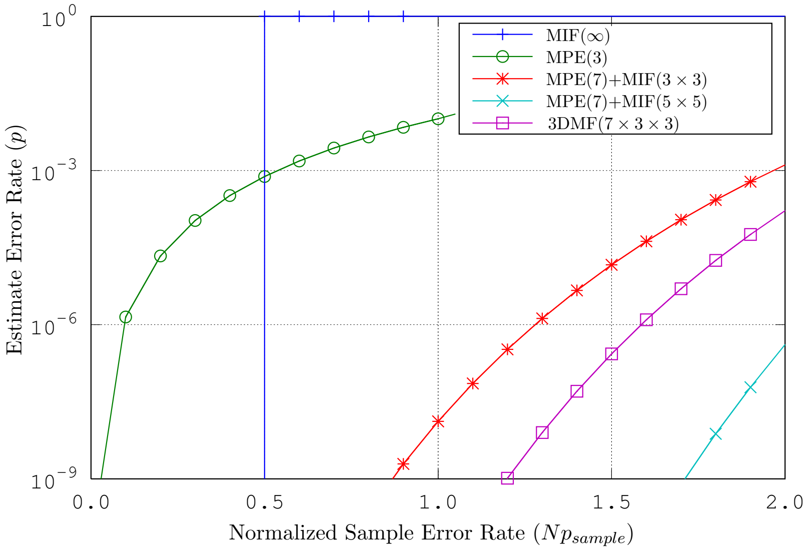
Figure 5.1 also provides information for a set of approaches that satisfies a certain requirement. This requirement can be different depending on the circumstances, and it can be determined based on the target error rates. For instance, if a certain application requires less than a single over-estimated pixel for reconstruction of 100M pixel images, its requirement can be defined by the target error rate of 10^{-8} at the normalized error rate of 1. As seen in Figure 5.1a, to satisfy this requirement with the pixel buffer size of 3, MPE+MIF needs a 11×11 filtering window size and 3DMF requires a 9×9 window



(a) Pixel buffer size m=3

(b) Pixel buffer size m=5

Figure 5.1: Analytical error rate curves of various estimation approaches for different pixel buffer sizes. *(cont.)*



(c) Pixel buffer size m=7

Figure 5.1: Analytical error rate curves of various estimation approaches for different pixel buffer sizes.

size. These approaches can be used for large scale images that can be reconstructed with a large scale filtering window size. If, at the risk of image blurriness, a large filtering window size greater than the specified size in this discussion is allowed, it is possible to consider lower breakdown levels. For a pixel buffer size of 5 (Figure 5.1b), a filtering window size of 5 × 5 satisfies the requirement for MPE+MIF and 3DMF. For a pixel buffer size of 7 (Figure 5.1c), a filtering window size of 5 × 5 for MPE+MIF satisfies the requirement, and a window size of 3 × 3 works for 3DMF. A filtering window size of 3 × 3 can also be effective for MPE+MIF, because not all of the regions in the image suffer from noise.

While a sample from a small normalized sample error rate range is supposed to be discarded to remove impulses in a rendering, it is desirable to include a sample from a large error rate range in the rendering. Therefore, a higher estimate error rate is advisable for a range where the normalized sample error rate is greater than 1. For instance, as seen in Figure 5.1a, using 3DMF(3 × 11 × 11) guarantees with the probability of 100% the inclusion of an impulse that occurs with a probability greater than the normalized sample error rate of 2.0. In other words, samples that occur with a probability greater than $2/N$ are included in the final rendering at the probability of 100%. Therefore, if we plan to use 3DMF($3 \times 11 \times 11$) and want to include samples that occur with a probability greater than 0.1%, the sample size needs to be greater than 2K. This sample size can decrease for a larger filtering window size and smaller breakdown level, but this is at the risk of error propagation and blurriness in the rendering. A similar analysis is possible for the other estimators discussed in this chapter.

In summary, the hybrid approaches of MPE and MIF can effectively eliminate impulse noise, and they can compute a minimum sample size for a sample probability necessary to be included in the rendering. To meet the requirement for less than a single pixel overestimation in a 100M pixel image and for a minimum filtering window size, 3DMF($7 \times 3 \times 3$) is an optimal solution. If a filtering window size larger than 3×3 and smaller than 11×11 is allowed, among the approaches discussed above, MPE(3)+MIF(11×11), 3DMF($3 \times 9 \times 9$), MPE(5)+MIF(5×5), 3DMF($5 \times 5 \times 5$), and MPE(7)+MIF(5×5) are expected to eliminate the impulse noise at the expense of image blurriness. If an even larger filtering window size is allowed, a solution with a smaller pixel buffer size and lower breakdown level is also available. In addition, sub-optimal solutions with smaller pixel buffer sizes, filtering window sizes, or breakdown levels can be effective in many cases because not all over-estimations introduce noise.

5.3.2 Spatial Complexity

Figure 5.1 also shows the number of pixel buffers required for the selected approaches. The spatial cost becomes greater as the number of pixel buffers increases, while it remains constant even if the filtering window size gets larger.

If we want to satisfy the requirement (less than a single over-estimated pixel within a 100M pixel image) without any blurriness, we can use MPE(19) that requires 19 pixel buffers [71]. However, using the filtering window size of 3×3 can decrease the required pixel buffer size down to 7, and decrease it further to 5 with a 5×5 filtering window size while still satisfying the requirement. It is also possible to choose a larger filtering window size and smaller pixel buffer size, such as the filtering window size of 9×9 and pixel buffer size of 3 with 3DMF, which is not included in the subsequent discussion because it can distort an image severely and cost too much computationally. If we want

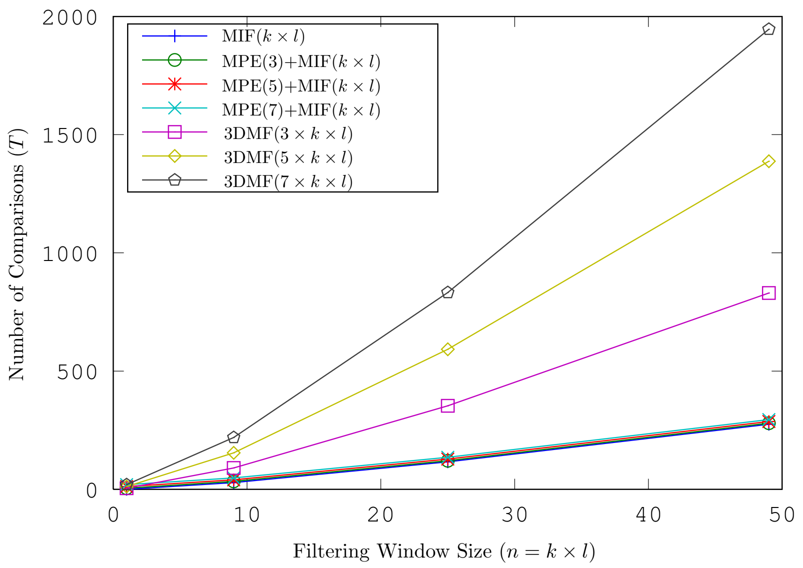


Figure 5.2: Computational costs for filtering window size.

to restrict the image blurring by using the minimum filtering window size, we can choose 3DMF($7 \times 3 \times 3$).

The actual spatial cost depends on the rendering method and it can vary depending on the optimization method [78, 79]. If a process renders pixel-by-pixel, the actual cost can be m extra pixel storages. However, if an interactive rendering process progressively generates an image, the actual cost corresponds to m extra image buffers.

5.3.3 Computational Complexity

Figure 5.2 shows the number of comparisons required for each estimation against the filtering window size. Because the filtering window size increases quadratically, it affects the computational costs more than the pixel buffer size. Therefore, better computational efficiency for a certain requirement can be obtained by increasing the pixel buffer size rather than by expanding the filtering window size.

The cost of MIF(n) is almost the same as that of MPE(m)+MIF(n), because the computational cost of MPE is relatively trivial compared to MIF. Furthermore, the cost of MPE remains constant for a change of the filtering window size. On the other

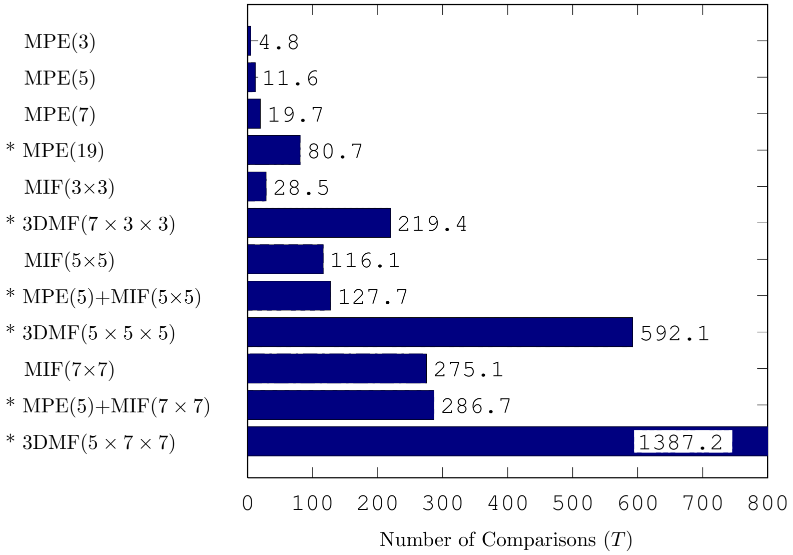


Figure 5.3: Computational costs of various estimation approaches.

hand, the computation of 3DMF($m \times n$) becomes much more expensive than MIF(n) and MPE(m)+MIF(n), because the total number of elements to be compared increases by m times.

Figure 5.3 shows the computational costs for the selected approaches. The approaches that satisfy the requirement (less than a single error for 100M pixels) are marked with an asterisk (*). For our requirement, MPE(19) is expected to be the least expensive solution. Then, the approaches that satisfy the requirement can be ordered from the least to most expensive as: MPE(5)+MIF(5 × 5), 3DMF(7 × 3 × 3), MPE(5)+MIF(7 × 7), 3DMF(5 × 5 × 5) and 3DMF(5 × 7 × 7). Because MPE(19) is expected to suffer the most significant bias among the approaches, MPE(5)+MIF(5×5) or 3DMF(7×3×3) is a good alternative at the expense of image blurring. For limited blurriness in an image, 3DMF(7×3×3) is a better choice rather than MPE(5)+MIF(5×5). Note that these two approaches are less expensive than MIF(7 × 7), so their operation is hardly expected to affect the overall rendering performance.

5.4 Simulation

This section discusses the performance of hybrid approaches of robust estimation and image filtering for a path tracing application. Particularly, the performance of hybrid methods are compared to that of existing MPE and MIF; MPE is the most recent method for impulse noise removal and shows comparable performance to the method of Pajot et al., and MIF is one of the most effective impulse noise removing filters [17, 18, 71]. For the simulation, we used random sampling to initialize camera rays. In addition, we did not apply any post image processing other than the 3×3 median filter (minimum filtering window size) mentioned in each approach.

It is important to note that the sample sizes are chosen to best illustrate the performance comparisons for different approaches. For a large number of impulses in a rendition the difference in performance becomes clear, while for a small number of impulses any approach can successfully eliminate impulse noise. To have a large number of impulses in a rendering, it is necessary to collect enough samples, because low probability samples with excessive values become impulses. In addition, a sample size is relative to an image and image filtering kernel (non-zero weights) size for a rendering quality. For instance, a weighted mean based approach can decrease the sample size by increasing the filtering kernel size. This is because the enlargement of the filtering kernel increases the sample size for the computation of a single pixel. Similarly, a weighted trimmed mean based approach for a large filtering window size does not need a sample size as large as that for for a small filtering window size. In this case, an optimal breakdown level can be computed using the error analysis.

5.4.1 Normalized Error for Image Quality Evaluation

The rendering quality is estimated using normalized error. The normalized error of the i-th pixel E'_i is defined by

$$E'_i = \frac{\hat{X}_i - X_i}{X_i}, \tag{5.25}$$

where $\hat{X}_i$ is the estimated pixel value, and X_i is the pixel value in a reference image [71]. For N pixels in a rendering, the mean of the normalized error can be regarded as

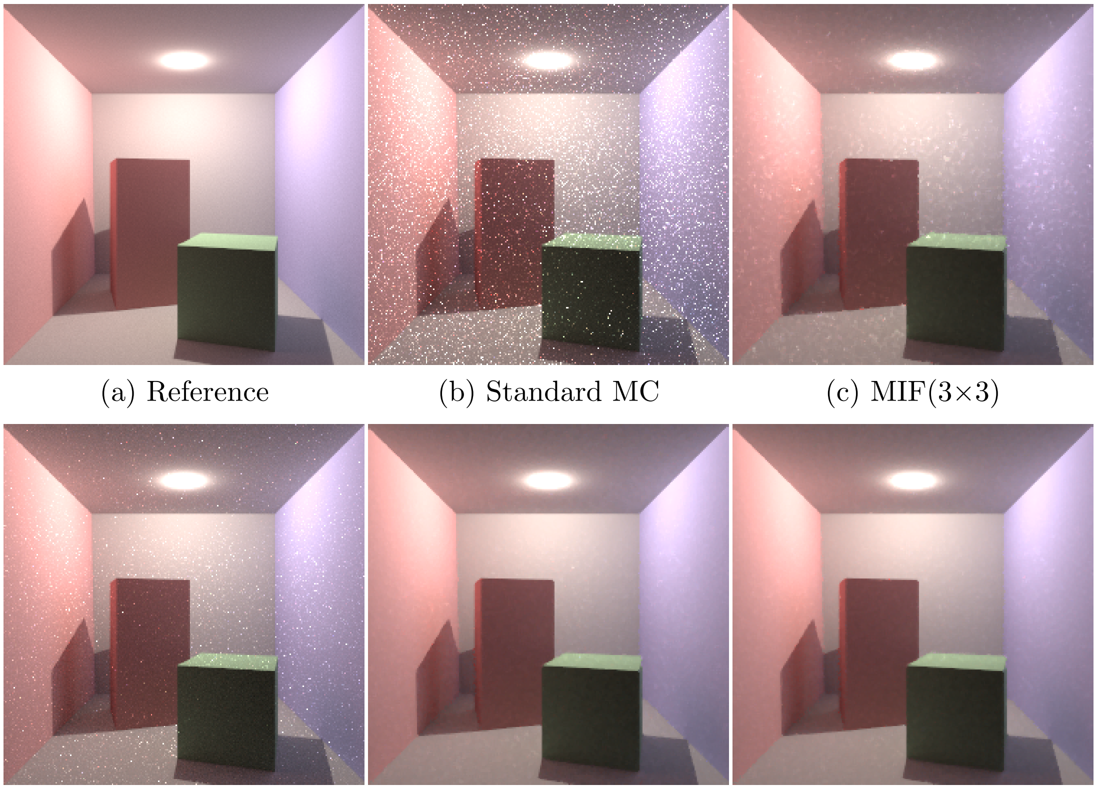


Figure 5.4: Comparison of path traced 250×250 images for 500 samples/pixel (scene from [26]).

the normalized bias that is defined by

$$\mathcal{B}(\hat{X}) = \frac{1}{N}\sum_{i=1}^{N} E'_i, \tag{5.26}$$

and the variance can be considered to be the normalized noise level that is defined by

$$\mathcal{N}(\hat{X}) = \frac{1}{N}\sum_{i=1}^{N} (E'_i - B)^2. \tag{5.27}$$

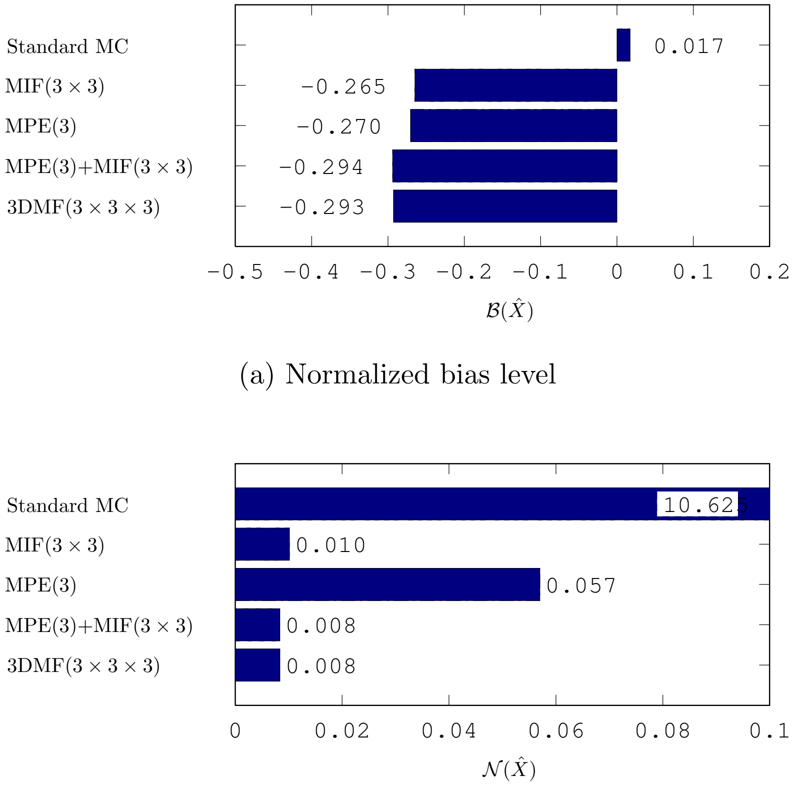
5.4.2 Simulation with a Small Sample Size

Figure 5.4 shows a set of images rendered using path tracing for 500 samples/pixel with various approaches, as well as a reference image. This path traced scene contains heavy impulse noise due to the bright indirect light source created on the ceiling. Therefore, for this scene we can expect a visually obvious change in the artifacts for different approaches.

The simulation was performed for a variety of estimation approaches: standard MC, MIF(3×3), MPE(3), MPE(3)+MIF(3×3), and 3DMF($3 \times 3 \times 3$). According to the analysis, MIF(3×3) is expected to show a comparable performance to MPE(3), but it will blur the image. Particularly, MIF(3×3) shows better performance than MPE(3) for a small normalized sample error rate range, but worse performance for a large normalized sample error rate range. In other words, MIF(3×3) can outperform MPE(3) for low probability noise, but it can also amplify high probability noise. This exaggeration can leave blurred blotches in an output image. MPE(3)+MIF(3×3) is expected to outperform MPE(3) and MIF(3×3), and 3DMF($3 \times 3 \times 3$) is expected to slightly outperform MPE(3)+MIF($3{\times}3$) (see Figure 5.1a).

Figure 5.4b shows that the standard MC estimate suffers severe impulse noise. MIF(3×3) decreases the noise, but leaves the blurred blemishes (Figure 5.4c). On the other hand, MPE(3) also reduces the noise, but the remaining noise still has the characteristic of sharpness (Figure 5.4d). MPE(3)+MIF(3×3) decreases noise further than MPE(3) while blurring the scene (Figure 5.4e). 3DMF($3 \times 3 \times 3$) shows the best quality among the rendition as predicted above (Figure 5.4f). Specifically, 3DMF($3 \times 3 \times 3$) includes the samples occurring with the probability of 0.4% ($2/N$) in the rendering with the probability of 96.41%. If the samples occurring with the probability of 0.1% are required to be included in the rendering, the sample size needs to be greater than or equal to 2K.

Figure 5.5 shows the statistics of the normalized error measured from Figure 5.4. As seen in Figure 5.5a, all approaches that improved the standard MC estimate show the similar negative image bias, while the standard MC estimate is slightly biased to positive. This is because the underlying population of pixel samples is right-skewed [17, 18, 71]. Figure 5.5b shows the normalized error level of the images. As described in Section 5.3, 3DMF($3{\times}3{\times}3$) shows the lowest error level and MPE(3)+MIF($3{\times}3$) shows the second lowest error level. The error level of MPE(3) is worse than MIF($3{\times}3$), which is partly because of the lower performance of MPE(3) than MIF($3{\times}3$) for a smaller normalized sample error rate range. In addition, MIF($3{\times}3$) attenuates extremely large



(a) Normalized bias level

(b) Normalized noise level

Figure 5.5: Statistics for image quality of Figure 5.4.

values by blurring, while MPE(3) just leaves those values when it fails to eliminate them.

The spatial cost of 3DMF(3×3×3) is the image buffer size of 3 and the computational cost of it is 90.3. The cost of 90.3 is smaller than 116.1 for MIF(5×5).

5.4.3 Simulation with Various Robustness Levels

This section presents a simulation with various levels of robustness. Particularly, this chapter discusses the robustness adjusted by various breakdown points and weights of estimators. For the subsequent simulations, the set of weights are defined for the filtering window size of 3 × 3:

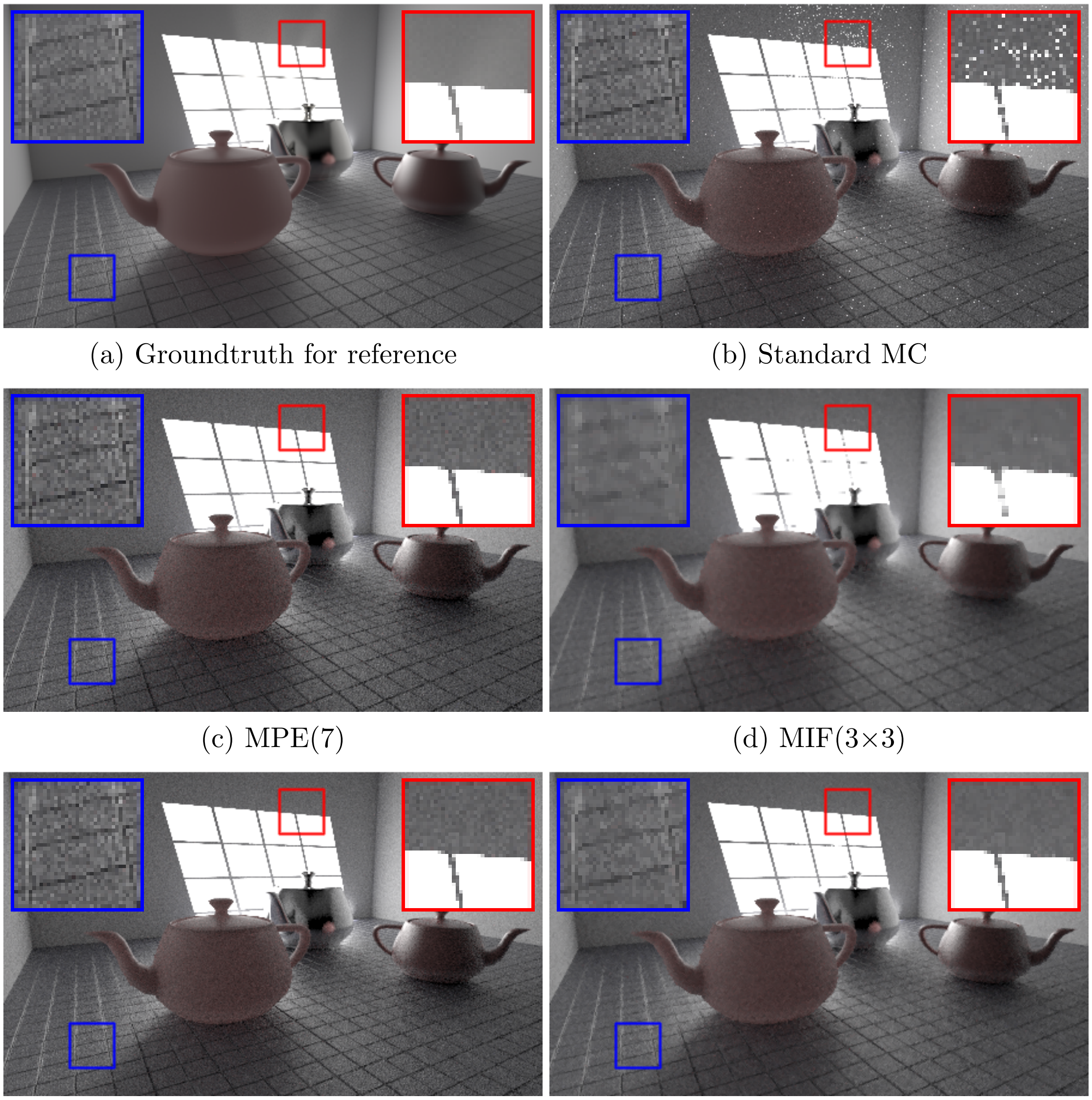
$$\mathbf{w}(w_c) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & w_c & 1 \\ 1 & 1 & 1 \end{bmatrix}, \tag{5.28}$$

Note that, in the case of 3DWMF, the same weight is applied to the sample radiance mean for the same pixel. The robustness control of robust pixel estimation using different estimators and pixel buffer sizes can be found in [71].

Figure 5.6 and Figure 5.8 show a set of renditions for different sample sizes, which are estimated by the various approaches and variations that are discussed in Section 5.1 and Section 5.2. Because this scene is dominated by relatively low probability samples introduced by the indirect reflection on the wall behind the pots, it requires a large sample size to obtain a quality rendering. In addition, the small unpredictable light source created on the metal teapot increases the chance of impulse noise.

The impulse noise is from the small probability samples, therefore it becomes more frequent as the sample size increases even though the overall variance decreases. When comparing Figure 5.6b and Figure 5.8a, it is observed that standard MC with 10k samples/pixel suffers from more impulse noise than the one with 1k samples/pixel. However, as seen in Figure 5.7b and Figure 5.9b, the increased sample size clearly attenuates the normalized noise level from 0.410 to 0.088. This is because the increase in the sample size failed to attenuate the noise down to an acceptable brightness and left them just less bright speckles.

Figure 5.6c shows that MPE(7) effectively eliminates the noise. From this fact, it is expected that the sample probability introducing the noise is less than 0.02% (see Figure 5.1c). For this small amount of noise, MIF(3 × 3) is also effective while blurring the image as shown in Figure 5.6d. Although the erroneous sample probability is less than 0.02%, the exact probability is unknown and it can be much lower than 0.02%. Therefore, less robust estimation approaches can still be effective. As seen in Figure 5.6e, TMPE(2/7, 3 × 3) completely eliminates the speckles. In addition, less robust TMPE can be also effective with robust image filtering as shown in Figure 5.6f. In particular, the potential image blurriness that can occur due to the spatial filtering is effectively reduced by an appropriate set of weights $\mathbf{w}(5)$ (compare insets of Figure 5.6d and Figure 5.6f).



(a) Groundtruth for reference

(b) Standard MC

(c) MPE(7)

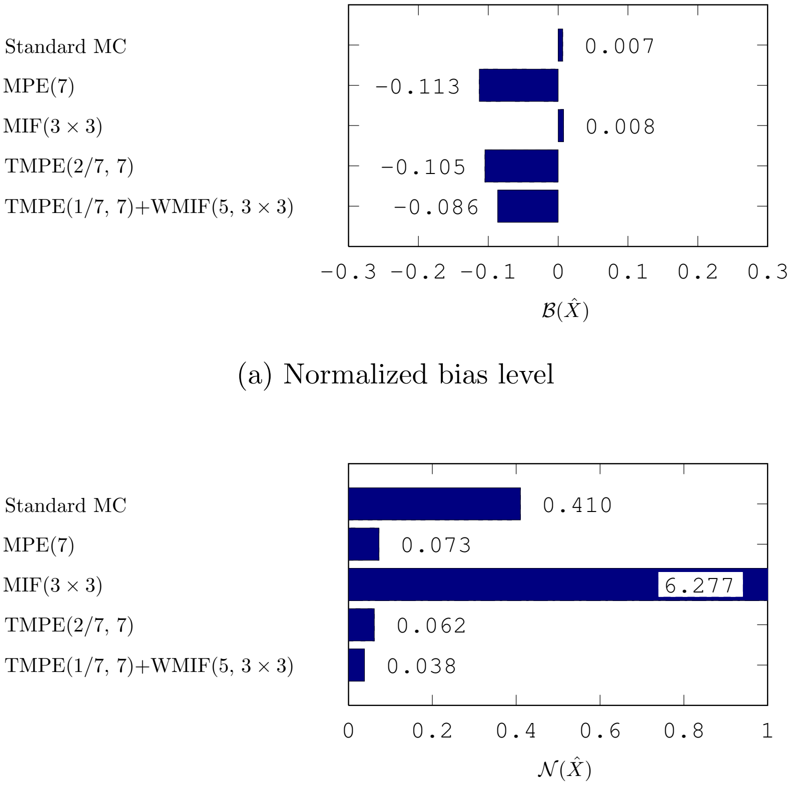
(d) MIF(3×3)

(e) TMPE(2/7, 7)

(f) TMPE(1/7, 7)+WMIF($\mathbf{w}(5)$, 3×3)

Figure 5.6: Comparison of path traced 450×270 images for 1k samples/pixel (scene from [26]).

Figure 5.7 shows the statistics of the normalized error measured from Figure 5.6. As seen in Figure 5.7a, while the standard MC and MIF estimates have similar positive image bias, all robust pixel estimation based methods introduce negative image bias. Because the underlying population is right-skewed, the image bias of standard MC is



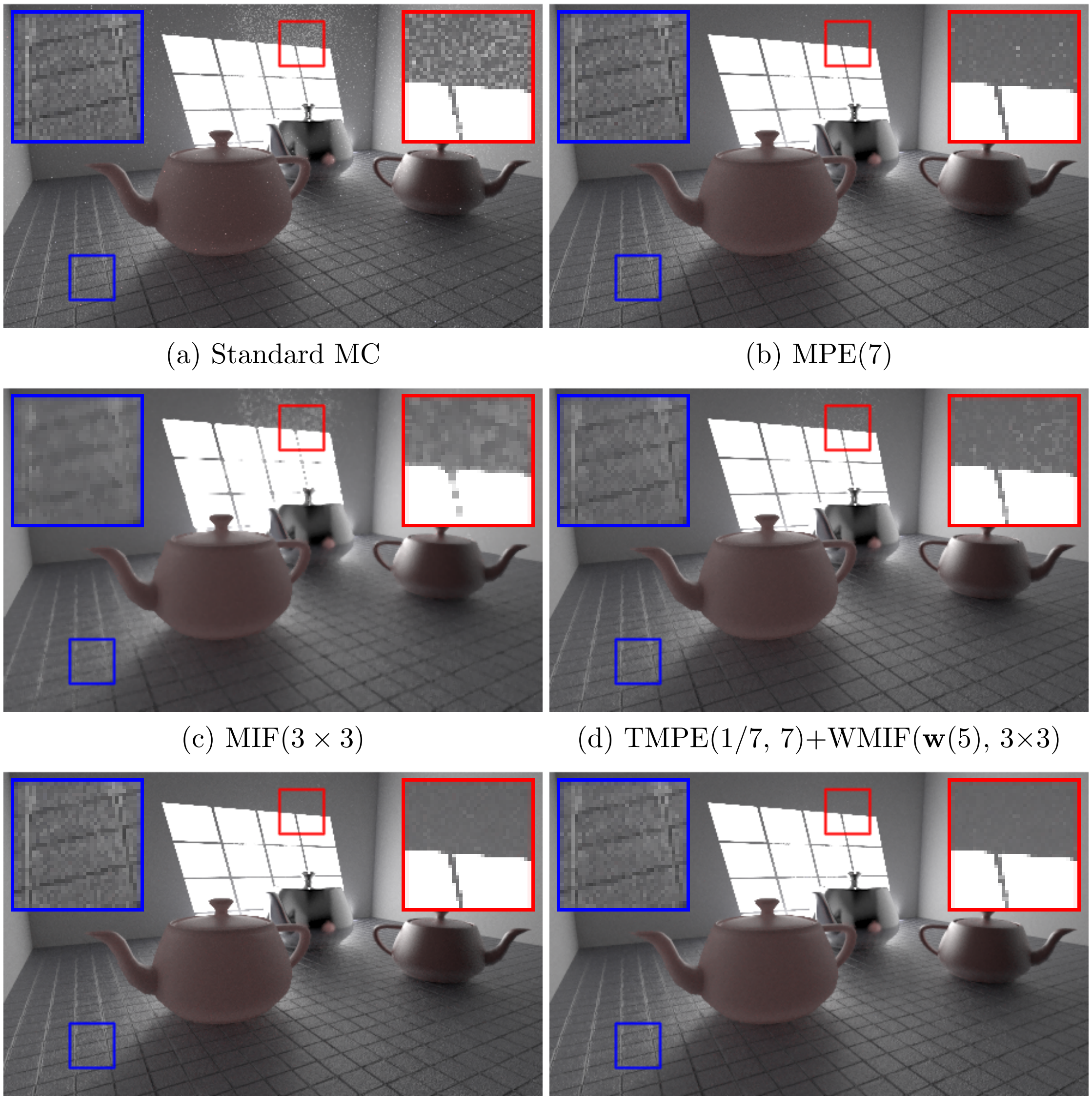
(a) Normalized bias level

(b) Normalized noise level

Figure 5.7: Statistics for image quality of Figure 5.6.

expected to be positive rather than negative for a small sample size. The positive image bias of MIF is due to the fact that the bright pixel values propagate into neighbors (see the window frame shadow in Figure 5.6d). The right-skewed underlying population also introduces negative image bias into the estimates measured using robust pixel estimation based methods. This negative image bias decreases as the sample size increases. For instance, the image bias of MPE(7) decreases from -0.113 for 1k samples/pixel to -0.041 for 10k samples/pixel (in Figure 5.9a).

Figure 5.7b shows that the estimates of the robust pixel estimation based methods have smaller normalized noise levels than that of standard MC. MIF shows the largest noise level due to the bright pixel propagation into neighbors and image distortion.



(a) Standard MC

(b) MPE(7)

(c) MIF(3×3)

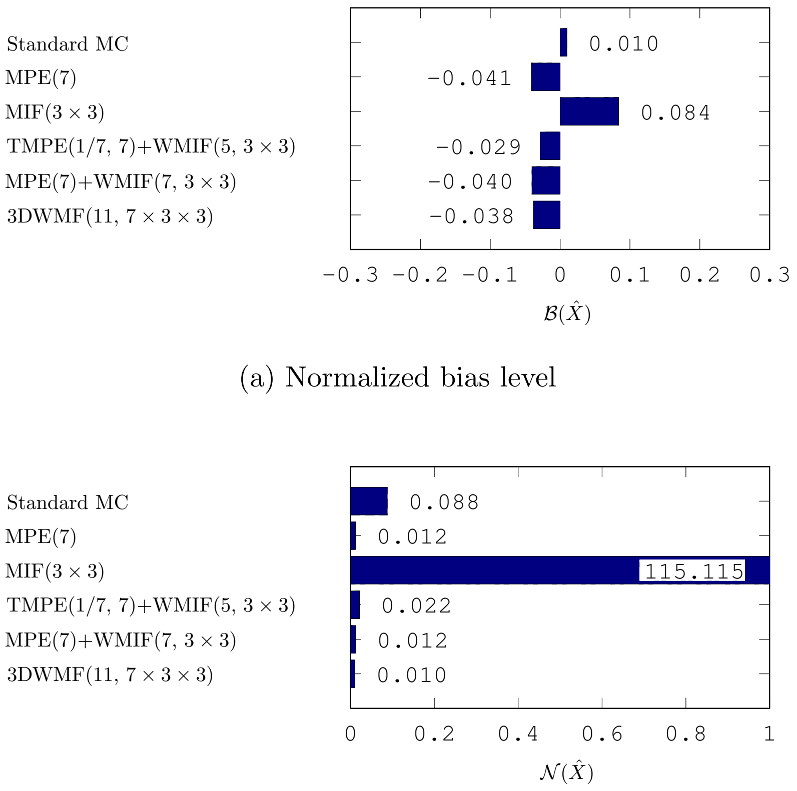
(d) TMPE(1/7, 7)+WMIF($\mathbf{w}$(5), 3×3)

(e) MPE+WMIF($\mathbf{w}$(7), 3×3)

(f) 3DWMF($\mathbf{w}$(11), $7 \times 3 \times 3$)

Figure 5.8: Comparison of path traced 450×270 images for 10k samples/pixel (scene from [26]).

As a result of the increased amounts of noise, even with 10k samples/pixel MPE(7) (Figure 5.8b), MIF(3×3)(Figure 5.8c), and TMPE(5, 7)+($\mathbf{w}$(5), 3×3)(Figure 5.8d) are not effective any more in eliminating impulse noise, as compared to Figure 5.6c, Figure 5.6d, and Figure 5.6f. As seen Figure 5.8e and Figure 5.8f, more robust estimation



(a) Normalized bias level

(b) Normalized noise level

Figure 5.9: Statistics for image quality of Figure 5.8.

approaches with MPE+WMIF(TMPE with bigger α) and 3DWMF are effective.

Figure 5.9 shows that the normalized bias and noise levels of the robust pixel estimation based approaches converge to zero while keeping the noise level smaller than that of standard MC. It is interesting to note that the bias and noise levels of MIF became worse than those for 1k samples/pixel, because the number of bright speckles increased and more error propagation into neighboring pixels occurred.

This simulation result suggests that, even though MPE+MIF and 3DMF with a certain pixel buffer size and filtering window size guarantee a target error rate, the image quality can be improved by adjusting extra parameters, such as the breakdown level and the set of weights for the estimators. This is because the underlying pixel radiance population is unknown. In addition, it is possible to adjust the robustness automatically using a heuristic based on the target error rate or the statistics of the overall scene radiance and pixel radiance.

5.4.4 Simulation with a Large Image and Sample Size

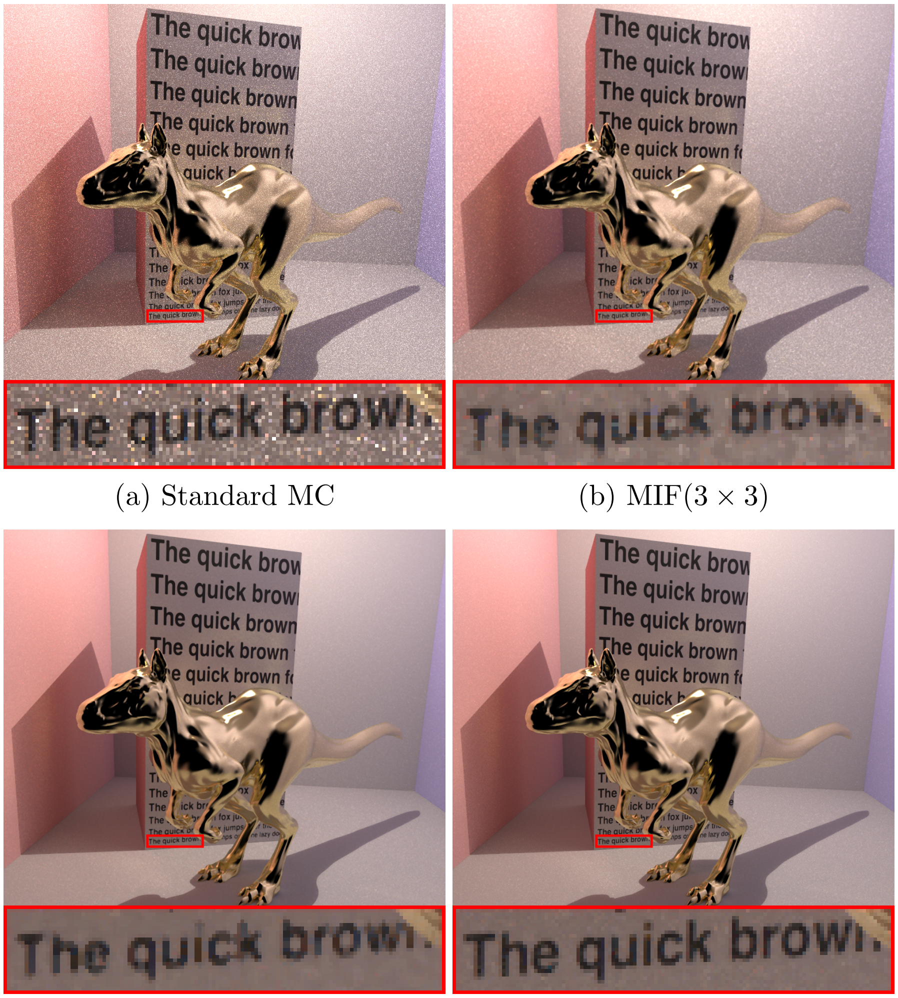
Figure 5.10 shows simulations for a large image size (1k×1k) and sample size (10k samples/pixel). The scene is the same scene as Figure 5.4, but a golden killeroo replaces the small blue box and a textured surface is put on the front wall of the tall box. In this scene, the golden shiny surface of the killeroo creates bright indirect lighting sources on the surface itself, and it increases the chance of impulse noise in the rendition.

Figure 5.10a shows that the scene suffers serious impulse noise. As seen in Figure 5.10b, MIF(3×3) can reduce the noise but leaves blurred blotches. Using MPE$(7)+$ MIF(3×3) can eliminate noise completely as seen in Figure 5.10c, because its expected error rate is 1.3×10^{-8}; this error rate is far smaller than the rate of a single pixel error for the given image size (10^{-6}). The filtering window size of 3×3 is the smallest possible size, so the blurriness introduced by the spatial filtering operation is supposed to be minimal. However, an appropriate set of weights can reduce the blurriness. Figure 5.10d shows the result of 3DWMF with $\mathbf{w}(7)$ and the same pixel buffer and filtering window size used for Figure 5.10c. When comparing the insets, it is clear that a higher weight for the pixel of interest is effective in recovering the image details.

The spatial complexity of MPE(7)+MIF(3×3) and 3DWMF$(\mathbf{w}(3),\ 7 \times 3 \times 3)$ is the image buffer size of 7. The computational cost of MPE(7)+MIF(3×3) is 48.2 and the computational cost of 3DWMF$(\mathbf{w}(3),\ 7 \times 3 \times 3)$ is 219.4+r (r for weighted median search) which are smaller than MIF(7×7)'s cost of 275.1. Note that the computational cost of MPE(7)+MIF(3×3) is more than four times cheaper than that of 3DMF$(7 \times 3 \times 3)$ and 3DWMF$(\mathbf{w}(3), 7 \times 3 \times 3)$; thus, MPE(7)+MIF$(3 \times 3)$ is a computationally efficient alternative with affordable rendering quality.

5.5 Discussion

This chapter evaluated the hybrid approaches for robust pixel estimation and image filtering. These approaches effectively eliminate the noise introduced by low probability



(a) Standard MC

(b) MIF(3×3)

(c) MPE(7)+MIF(3×3)

(d) 3DWMF($\mathbf{w}(7)$, $7 \times 3 \times 3$)

Figure 5.10: Comparison of path traced 1k×1k images for 10k samples/pixel. (a) Because the pixel sample mean radiance distribution for a given sample size is fat-tailed, the standard Monte Carlo estimates suffer from impulse noise. (b) For this type of noise, a median filter (minimum image filtering window size of 3×3) is known to be an effective method, but it fails when its filtering window contains more than 50% corrupted pixel samples. The noise can be effectively eliminated with the same image filtering window size by our proposed methods: (c) robust image filtering over robust pixel estimation and (d) three-dimensional robust filtering. An appropriate set of weights for the estimator can alleviate the potential blurriness from spatial filtering (compare insets of (c) and (d)).

samples. On the other hand, because ray-traced images are more vulnerable to bright speckles than dark ones and because the distribution of pixel values is right-skewed, the robust estimates can be negatively biased. However, as the sample size increases, the underlying population becomes symmetric and the estimates become unbiased.

This chapter examined the weighted trimmed mean as a robust estimator; the median, weighted median, mean, and trimmed mean are special cases of the weighted trimmed mean. In addition, this chapter uses the median estimator for an analysis of error correcting performance boundaries, assumes an image-by-image reconstruction of a rendition for spatial costs, and uses a merge sort algorithm for computational costs.

It is important to note that any variation of the estimation processes discussed in this chapter is possible. For instance, the estimators mentioned in this chapter can be replaced with different robust estimators and target error rates for either more robust results or less robust but more visually appealing results. In addition, a similar analysis can be done for a different pixel buffer size, filtering window size, and breakdown level depending on an application's purpose. Instead of the image-by-image reconstruction method, we can consider a pixel-by-pixel reconstruction approach, which can be a memory efficient approach for an off-line rendering process because it recycles pixel buffers used in different pixels. The estimating algorithm can be replaced with another sorting algorithm for better performance.

For the hybrid approaches of robust pixel estimation and robust image filtering, the analysis and simulation provided in this chapter suggests a way to determine an appropriate pixel buffer and filtering window size, which is based on the target error rate to discard low probability samples. In addition, the analysis shows a sample probability with which a sample is included in the rendering, which can be used to specify a minimum sample size for the sample probability. This analysis can be extended to a different filtering window size and breakdown level.

Furthermore, this analysis suggests the necessity for a new intermediate image format between rendering and post image processing. If the intermediate image format employs 7 pixel buffers for each pixel (7 image buffers for an image) to describe the pixel radiance distribution, it can guarantee an impulse noise rate of less than 10^{-8} within the final result with 3DMF$(7 \times 3 \times 3)$. If a filtering window size is greater than or equal to 11×11, an image format consisting of 3 pixel buffers per pixel will be effective. Then, the final pixel value can be determined by choosing the weighted trimmed mean.

Finally, similar to the application of 2D image filters, other experiments with the image buffers can be expected to generate visually pleasing results. For instance, many experiments have been done for weighted-mean-based filtering with different weights using Gaussian, bilateral, NLM, and different feature based weights. This weighted-mean-based image filtering can be considered as an optimal solution for an underlying Gaussian distribution of weighted pixel samples; therefore, it is not optimal for other types of underlying populations. These image filters can be defined by 3DWTMF with special filtering coefficients and 0% breakdown level. In order to produce more robust or visually appealing results for a given sample size, the pixel buffer size or breakdown level may be explored. In other words, because these weighted-mean-based filters are special cases of 3DWTMF, similar approaches for choosing appropriate sample weights can be explored for 3DWTMF, and various breakdown levels can also be examined.

Chapter 6

Progressive Volume Photon Mapping with Robust Estimation

Volume photon mapping is a full Monte-Carlo simulation that is capable of generating the correct appearance of translucent objects, such as teeth and dental restorative materials [25]. However, it needs a large number of volume photons in order to obtain the correct distribution of the volume photons; otherwise the resulting image becomes distorted or biased. For example, without enough volume photons, the volume photons can create incorrectly populated spaces in the map, which introduces blotches or speckles in the resulting image. Specifically, overpopulated spaces introduce bright speckles and underpopulated spaces introduce dark speckles. Even worse, the number of volume photons necessary for a quality rendition restricts interactivity of the rendering process because of the high computational cost required to collect the photons, which need to be prepared before actual rendering.

The number of necessary volume photons depends on scene setup and the optical properties of materials. For instance, if the optical thickness of translucent objects in a scene is low, a relatively large number of light rays will be consumed to collect the required number of volume photons, and the resulting volume photon map will have a higher chance of being correct. On the other hand, if the optical thickness is high, a relatively small number of light rays will be consumed to collect the same number of volume photons, and the resulting volume photon map will have a higher chance of having undesired populations of volume photons.

This chapter introduces a progressive volume photon mapping technique to eliminate the non-interactivity and bias of the conventional volume photon mapping. This progressive technique divides the heavy computation to build up a volume photon map into multiple small computations that build up multiple volume photon maps. The rendering process provides an image frame using the small volume photon map, and it then responds more actively to user actions while providing a preview. The volume photons collected in each image frame are disposable; therefore, the rendering process is not limited by the memory restrictions of the conventional approach. In addition, this approach does not suffer from the image bias that the conventional volume photon mapping does due to the limited number of volume photons. Finally, because randomly chosen volume photon maps in each image frame make it possible to trace an outgoing radiance distribution in each pixel, the noise introduced by an incorrect photon distribution can be repaired by robust estimation.

The remainder of this chapter is divided into four sections. Section 6.1 reviews the conventional volume photon mapping approach for rendering translucent objects. Section 6.2 discusses the progressive volume photon mapping approach, which is able to provide better interactivity to users while displaying a preview. Section 6.3 presents simulation results. Finally, Section 6.4 summarizes this section and discusses future work.

6.1 Volume Photon Mapping

When a photon travels through a translucent medium, it is either absorbed or scattered. In addition, there can be light emission for self-emitting or fluorescent materials. By combining the light emission, absorption, and scattering, the outgoing radiance, L, at the location, x, in the direction, $\vec{\omega}$, is obtained using the volume rendering equation

$$\begin{aligned}L(x,\vec{\omega}) = &\int_0^s e^{-\tau(x,x')}\sigma_a(x')L_e(x',\vec{\omega})dx' \\ &+ \int_0^s e^{-\tau(x,x')}\sigma_s(x')\int_{\Omega 4\pi} p(x',\vec{\omega}',\vec{\omega})L_i(x',\vec{\omega}')d\vec{\omega}'dx' \\ &+ e^{-\tau(x,x+s\vec{\omega})}L(x-s\vec{\omega},\vec{\omega})\end{aligned} \tag{6.1}$$

where $\tau(x, x')$ is the optical depth, σ_a is the absorption coefficient, L_e is the emitting radiance, σ_s is the scattering coefficient, $p(x', \vec{\omega}', \vec{\omega})$ is the phase function, and L_i is the incident radiance. For non-emissive materials, L_e is zero, and Equation (6.1) is simplified to the equation

$$\begin{aligned} L(x, \vec{\omega}) = &\int_0^s e^{-\tau(x,x')} \sigma_s(x') \int_{\Omega 4\pi} p(x', \vec{\omega}', \vec{\omega}) L_i(x', \vec{\omega}') d\vec{\omega}' dx' \\ &+ e^{-\tau(x, x+s\vec{\omega})} L(x - s\vec{\omega}, \vec{\omega}). \end{aligned} \tag{6.2}$$

The appearance of participating media can be simulated by solving Equation (6.1) or Equation (6.2). Volume photon mapping solves these equations by approximating the incident radiance to the volume radiance. Volume photon mapping consists of two consecutive processes: volume photon capturing and ray marching. The process for volume photon capturing needs to be done before the actual rendering in which ray marching is processed. The long preprocessing time for volume photon capturing and the large number of steps for ray marching are challenges that prevent interactive rendering. Even worse, the preprocessing pass needs to be performed whenever the scene changes, and the rendering pass needs to be performed from the beginning each time the volume photon map is updated.

6.1.1 Construction of Volume Photon Map

The volume photon density information is stored in the form of a map, which is called the volume photon map. Whenever a photon is scattered or absorbed, its power and traveling direction are recorded in the map, and they make a contribution to the outgoing radiance. In order to trace the photons emitted by light sources, the distance the photon travels, the probability of scattering, and the direction of scattering need to be computed based on the optical characteristics of given materials.

The average distance that a photon moves through a medium before the next interaction, d, is determined by the extinction coefficient $\sigma_t = \sigma_s + \sigma_a$ and given by the equation

$$d = 1/\sigma_t. \tag{6.3}$$

If a photon travels a distance d, as a result of absorption and out-scattering, the photon power is scaled by a factor of $e^{-\sigma_t d}$. Because the probability density of the scaled power is uniformly distributed, the distance to the next interaction is determined by the equation

$$d = -\ln \xi / \sigma_t, \tag{6.4}$$

where $\xi \in]0,1]$ is a uniformly distributed random number. When the volume photons are sampled based on the traveling distance, their power does not have to be adjusted, which is otherwise necessary to account for the attenuation by distance.

At the point of interaction, it needs to be determined whether the photon is scattered or absorbed. The probability of scattering is given by the scattering albedo, $\Lambda = \sigma_s / \sigma_t$. For a uniformly distributed random number $\xi \in [0,1]$, Russian roulette can be used to determine the photon's interaction based on the following

$$\xi \in [0,1] \rightarrow \begin{cases} \xi \leq \Lambda & \text{photon is scattered} \\ \xi > \Lambda & \text{photon is absorbed.} \end{cases} \tag{6.5}$$

When the photons are sampled according to Equation (6.5), their power attenuation due to the absorption does not have to be accounted for. However, the power for each wavelength or color still needs to be adjusted. This is because each wavelength or color in a photon can have different optical characteristics. In this case, the average scattering albedo can be used to determine the travel distance and scattering probability, and then to scale the power. The average scattering albedo is given by the equation

$$\Lambda_{avg} = (\Lambda_r + \Lambda_g + \Lambda_b)/3. \tag{6.6}$$

To account for the fact that photons have more chances to survive for a higher scattering albedo, the power of each color needs to be scaled using both its scattering albedo and the average scattering albedo. Equation (6.7) shows the power of the scaled photon:

$$\begin{aligned} \Phi_{s,r} &= \Phi_{i,r} \frac{\Lambda_r}{\Lambda_{avg}} \\ \Phi_{s,g} &= \Phi_{i,r} \frac{\Lambda_g}{\Lambda_{avg}} \\ \Phi_{s,b} &= \Phi_{i,r} \frac{\Lambda_b}{\Lambda_{avg}}, \end{aligned} \tag{6.7}$$

where $(\Phi_{i,r}, \Phi_{i,g}, \Phi_{i,b})$ is the power of the incoming photon, and $(\Phi_{s,r}, \Phi_{s,g}, \Phi_{s,b})$ is the power of the scattered photon. It is important to note that the average scattering albedo in Equation (6.7) keeps the average power of the photon constant throughout the simulation. This approach generally gives a good approximation, and more accurate simulation can be done by tracing the photons for each wavelength or color component.

The probability distribution of the scattering direction is defined by the phase function, $p(x, \vec{\omega}', \vec{\omega})$, which describes the amount of light scattered to the direction $\vec{\omega}'$ at position x for the light incident from the $\vec{\omega}$ direction. As the phase function is the probability distribution function of $\vec{\omega}'$, it must integrate to 1 over the entire parameter space $\Omega_{4\pi}$:

$$\int_{\Omega_{4\pi}} p(x, \vec{\omega}', \vec{\omega}) d\vec{\omega}' = 1. \tag{6.8}$$

If the phase function is circularly symmetric around the axis of the incident ray, it can be written as a function of the scattering angle θ, and it therefore satisfies the equation

$$2\pi \int_0^\pi p(\theta) \sin\theta d\theta = 1, \tag{6.9}$$

where θ is $\cos^{-1}(\vec{\omega}' \cdot \vec{\omega})$.

The anisotropy of scattering can be characterized by the average cosine of scattering, $g(x)$. $g(x)$ is defined as the integral over the sphere multiplied by the cosine of the angle, θ, and given by

$$g(x) = \int_{\Omega_{4\pi}} p(x, \vec{\omega}', \vec{\omega})(\vec{\omega}' \cdot \vec{\omega}) d\vec{\omega}'. \tag{6.10}$$

For $p(\theta)$, $g(x)$ can be represented as the equation

$$g(x) = 2\pi \int_0^\pi p(\theta) \cos\theta \sin\theta d\theta. \tag{6.11}$$

Phase functions for transparent materials are often highly anisotropic so uniform sampling is inefficient. Therefore, importance sampling may need to be considered for the phase function [81].

The phase function for isotropic scattering is a constant, because the probability density is uniformly distributed:

$$p(\theta) = \frac{1}{4\pi}. \tag{6.12}$$

The most commonly used phase function is the empirical Henyey-Greenstein phase function, which is given by the equation

$$p(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{1.5}}, \tag{6.13}$$

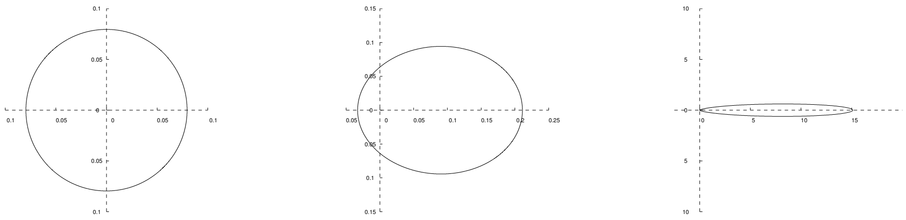


Figure 6.1: The Henyey-Greenstein phase function with different g values. From left to right g = 0.0, g = 0.3, and g = 0.9.

where $g \in]-1, 1[$ is an asymmetry parameter equal to the average cosine of the scattered directions. Positive g gives forward scattering, negative g gives backward scattering, and $g = 0$ is isotropic scattering. Figure 6.1 shows that higher values of g make the scattering more directional.

Given the incoming direction of a photon, the angle, θ, of the new scattered direction is given by the equation

$$\cos\theta = \frac{1}{2g}\left(1 + g^2 - \left(\frac{1 - g^2}{1 - g + 2g\xi}\right)^2\right), \tag{6.14}$$

where ξ is a uniform random number between zero and one. Using Equation (6.14), the importance sampling of the scattered direction can be applied.

As the shape of the function is close to an ellipsoid, the equation can be approximated by a simpler one, the Schlick phase function, which is given by the equation

$$p(\theta) = \frac{1 - k^2}{4\pi(1 + k\cos\theta)^2}, \tag{6.15}$$

where $k \in] - 1, 1[$ is used to control the forwardness of scattering like g in the Henyey-Greenstein function.

The angle, θ, of the new scattered direction is given by the equation

$$\cos\theta = \frac{1\xi + k = 1}{2k\xi + k - 1}, \tag{6.16}$$

where ξ is a uniformly distributed random number between zero and one.

For more complex types of scattering, a linear combination of several phase functions can be considered. Gkioulekas et al.[82] used a linear combination of HG and von Mises-Fisher Lobes to investigate the role of phase function. Fried et al. [4] employed a linear combination of an isotropic scattering phase function and a highly forward-peaked HG phase function to model the light scattering properties of enamel and dentin. A general linear combination of phase functions is defined by the equation

$$p(\theta) = \sum_{i=1}^{N} \omega_i p_i(\theta), \tag{6.17}$$

where $\sum_{i=1}^{N} \omega_i = 1$, $p_i(\theta)$ is the ith phase function and ω_i is the weight.

The linear combination of an HG phase function and an isotropic scattering fraction, which Fried et al. suggests, is defined by

$$p(\theta) = f_d \frac{1}{4\pi} + (1 - f_d) \left(\frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{1.5}} \right), \tag{6.18}$$

where f_d is the fraction of isotropic scattering ($f_d \leq 1$) and g is the average cosine of the HG phase function [4]. When f_d is zero, $p(\theta)$ is a single HG phase function.

The photon coming directly from the light source can be computed using a standard ray-tracing technique as a single scattering radiance. Because volume photon tracing for single scattering creates more variance, the standard ray-tracing technique is a better option to reduce the noise for a given sample size, but it creates a computational overhead in the rendering stage to compute the direction of incoming light rays.

In a volume photon map, the flux carried by the photons is stored at each point where the photons interact within the volume. As the number of photons increases, the simulation becomes more accurate; therefore, the simulation accuracy is proportional to the available memory space for the volume photons. Instead of the random sampling points of the conventional volume photon map, predefined sampling points can be used for the rendering efficiency. For instance, Zhang et al. introduces a Cartesian grid structure to specify the sampling points and uses PRT to model the light scattering for real-time performance [37].

6.1.2 Estimation of Pixel Values

When a light ray enters a translucent material, it is refracted at the surface and travels within the medium. The outgoing radiance at a surface point can be evaluated through tracing the camera ray while evaluating the contribution of light rays. The increased outgoing radiance due to in-scattered radiance is defined by the sum of a direct single scattering term and an indirect multiple scattering term. The radiance estimate for the single scattering is given by the equation

$$(\vec{\omega} \cdot \bigtriangledown) L_s(x, \vec{\omega}) = \sum_{i=1}^{I} R_i'(x, \vec{\omega}_i') p(x, \vec{\omega}_i', \vec{\omega}) \sigma_s(x), \tag{6.19}$$

where I is the number of light sources in the scene, and R_i' is the radiance introduced by the light source R_i [81]. The radiance estimate for the multiple scattering is given by the equation

$$(\vec{\omega} \cdot \bigtriangledown) L_m(x, \vec{\omega}) = \sum_{j=1}^{J} p(x, \vec{\omega}_j', \vec{\omega}) \frac{\Delta\Phi_j(x, \vec{\omega}_j')}{\frac{4}{3}\pi r^3}, \tag{6.20}$$

where J is the number of photons within the volume of the sphere $(4/3\pi r^3)$, and $\Delta\Phi_j$ is the flux carried by the photons that correspond to the indirect illumination in the incoming direction of $-\vec{\omega}_j'$ [81]. Ray marching computes the outgoing radiance along the camera ray within the medium. Using Equation (6.2), Equation (6.19), and Equation (6.20), the ray marching algorithm can be written by the equation

$$\begin{aligned} L_n(x, \vec{\omega}) = & \sum_{i=1}^{I} R_i'(x, \vec{\omega}_i') p(x, \vec{\omega}_i', \vec{\omega}) \sigma_s(x) \Delta x \\ & + \left\{ \sum_{j=1}^{J} p(x, \vec{\omega}_j', \vec{\omega}) \frac{\Delta\Phi_j(x, \vec{\omega}_j')}{\frac{4}{3}\pi r^3} \right\} \Delta x \\ & + e^{-\sigma_t(x)\Delta_x} L_{n+1}(x + \Delta x, \vec{\omega}), \end{aligned} \tag{6.21}$$

where the first term accounts for the contribution due to direct illumination, the second term is the radiance due to multiple scattering, and the last term is the attenuated radiance entering the segment from the back [81].

It is important to note that R_i' in Equation (6.21) is the attenuated radiance of a light source R_i due to extinction inside translucent media; therefore, the intensity of the light decreases by the equation

$$R_i'(x, \vec{\omega}_i') = e^{-\sigma_t d} R(x, \vec{\omega}_i'), \tag{6.22}$$

where d is the distance the light travels through the medium. For non-homogeneous media, the attenuation may need to be integrated using another ray-marching evaluation.

The contribution of multiple scattering defined by Equation (6.20) can account for the contribution of single scattering defined by Equation (6.19) by including the photon's first interaction into the volume photon map. In this case, the estimated outgoing radiance can have a higher variance because the incoming radiance recorded in the volume photon map is an approximation of actual incoming radiance, which can be computed using ray marching. However, by eliminating the extra ray marching processes to compute the incoming radiance, more efficient computation can be obtained.

6.2 Progressive Volume Photon Mapping with Robust Estimation

In the conventional volume photon mapping algorithm, the pixel values are obtained using Equation (6.21). The ray marching process defined in Equation (6.21) can be represented as the sum of the sub-estimates contributed by different single and multiple scatterings. The radiance estimate due to single scattering defined in Equation (6.19) can be replaced by a summation of K sub-estimates:

$$\begin{aligned}(\vec{\omega}\cdot\bigtriangledown)L_s(x,\vec{\omega}) &= \sum_{k}^{K}(\vec{\omega}\cdot\bigtriangledown)L_{s,k}(x,\vec{\omega}),\\ \text{where } (\vec{\omega}\cdot\bigtriangledown)L_{s,k}(x,\vec{\omega}) &= \sum_{i}^{I/K} R_i(x,\vec{\omega}'_i)p(x,\vec{\omega}'_i,\vec{\omega})\sigma_s(x).\end{aligned} \tag{6.23}$$

In a similar fashion, the radiance estimate due to multiple scattering defined in Equation (6.20) can be replaced by a sum of K sub-estimates:

$$\begin{aligned}(\vec{\omega}\cdot\bigtriangledown)L_m(x,\vec{\omega}) &= \sum_{k}^{K}(\vec{\omega}\cdot\bigtriangledown)L_{m,k}(x,\vec{\omega}),\\ \text{where } (\vec{\omega}\cdot\bigtriangledown)L_{m,k}(x,\vec{\omega}) &= \sum_{j}^{J/K} p(x,\vec{\omega}'_j,\vec{\omega})\frac{\Delta\Phi_j(x,\vec{\omega}'_j)}{\frac{4}{3}\pi r^3}.\end{aligned} \tag{6.24}$$

As seen in Equation (6.23) and Equation (6.24), the final pixel value is a linear combination of the direct incoming radiances from light sources and the virtual light sources created by photon interactions within a volume; therefore, the final pixel value can be defined by

$$\begin{aligned} L_n(x,\vec{\omega}) &= \sum_i^I S(R_i)\Delta x + \sum_j^J M(\Delta\Phi_j)\Delta x + e^{-\sigma_t(x)\Delta x} L_{n+1}(x+\Delta x,\vec{\omega}) \\ \text{where } S(I(x,\vec{\omega}')) &= \sum_i^{I/K} R_i(x,\vec{\omega}_i)p(x,\vec{\omega}'_i,\vec{\omega})\sigma_s(x), \\ M(\Delta\Phi(x,\vec{\omega}')) &= \sum_j^{J/K} p(x,\vec{\omega}',\vec{\omega})\frac{\Delta\Phi_j(x,\vec{\omega}'_j)}{\frac{4}{3}\pi r^3}. \end{aligned} \tag{6.25}$$

If the outgoing radiance L_o is defined as a function of a direct incoming radiance set $\mathbf{R}$ and incoming flux set $\Delta\mathbf{\Phi}$, it can also be represented as the sum of K sub-outgoing radiance estimates; therefore, the outgoing radiance L_o can be represented as

$$L_0 = L_0(\mathbf{R},\Delta\mathbf{\Phi}) = \sum_k^K L_{0,k} = \sum_k^K L_0(\mathbf{R}_k,\Delta\mathbf{\Phi}_k). \tag{6.26}$$

It is important to note that $L_{0,k}$ is a partial pixel value and that it can be used for intermediate rendering in a progressive rendering process.

As the number of elements in each $\mathbf{R}$ and $\Delta\mathbf{\Phi}$ increases, L_0 converges to the correct value [81]. However, it is impossible to increase the number of elements up to infinity because of the limited memory available of a system. Consequently, this strategy does not guarantee the correctness of a simulation. On the other hand, by increasing K while keeping the sizes of $\mathbf{R}_k$ and $\Delta\mathbf{\Phi}_k$ constant, L_0 can be computed for an infinite number of volume photons. This is because $\mathbf{I}_k$ and $\Delta\mathbf{\Phi}_k$ can be recycled after $L_{0,k}$ is computed; $L_{0,k}$ can also be discarded after it is added to a screen image buffer. Consequently, the correct simulation results are expected as K increases. In addition, the rendering produced for a given value of K can be a intermediate rendition in the progressive rendering approach.

If $L_{0,k}$ is computed for each frame, it is possible to obtain a higher frame rate in the progressive rendering approach. This is because $L_{0,k}$ requires a smaller number of volume photons than L_0. Thanks to the smaller quantity of volume photons, the processing time for volume photon capturing, volume photon map building and ray marching in each frame decreases.

This progressive rendering approach is also more efficient than the conventional volume photon mapping for the same number of volume photons. The computational complexity of volume photon mapping can be decomposed into volume photon tracing and ray-marching costs, which are highly dependent on the number of volume photons that each process captures and uses. If it is assumed that the volume photons are managed in a k-d tree to store diffuse and caustic photons, suggested by Jensen et al. and used by the PBRT photon mapping implementation, the cost for n volume photon tracing C_{pt} is defined by

$$C_{pt} = nT_{pt} + n\log(n), \tag{6.27}$$

where T_{pt} is a photon tracing cost and $n\log(n)$ is the best cost for building a depth-balanced k-d tree [81, 83, 26, 84, 85]. The cost of ray marching with k-nn for n volume photons is given by

$$C_{rm} = T_{seg}k\log(n), \tag{6.28}$$

where T_{seg} is the number of ray segments and $k\log(n)$ is the cost of the integration of volume photons using k-nn [86].

For K subset of n volume photons, each n/K volume photon tracing is defined by

$$C_{pt}/\text{subset} = n/KT_{pt} + n/K\log(n/K), \tag{6.29}$$

and the total cost C'_{pt} for n volume photon tracing is defined by

$$C'_{pt} = KC_{pt}/\text{subset} = nT_{pt} + n\log(n/K). \tag{6.30}$$

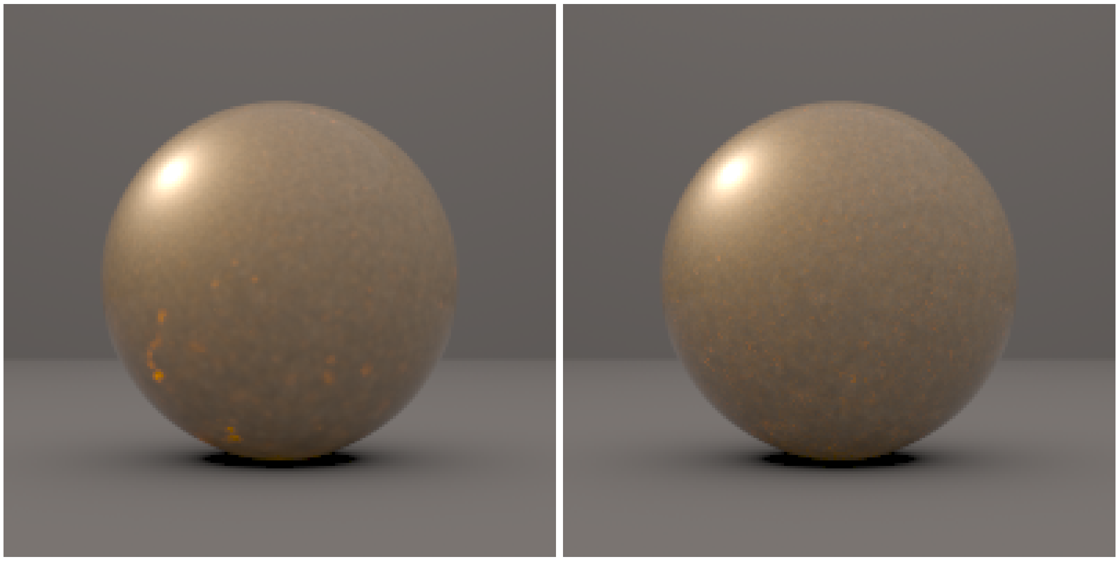
Compared to Equation (6.27), Equation (6.30) shows that the progressive approach has a performance gain of $\log(n)/\log(n/K)$ over the conventional volume photon mapping technique when building a volume photon map.

The ray marching process of progressive volume photon mapping seeks k/K nearest neighbors instead of k neighbors, which the conventional approach searches. This is because the progressive approach searches k/K for each subset, and then searches k neighbors for K subsets. The cost for ray marching with k/K-nn for n/K volume photons is define by

$$C_{rm}/\text{subset} = T_{seg}k/K\log(n/K). \tag{6.31}$$

The total cost for ray marching of progressive volume photon mapping is given by

$$C'_{rm} = KC_{rm}/\text{subset} = T_{seg}k\log(n/K). \tag{6.32}$$



(a) Mean estimate

(b) Median estimate

Figure 6.2: Comparison for 21K samples/pixel

Comparing Equation (6.32) to Equation (6.28) shows that the progressive approach has a performance gain of $\log(n)/\log(n/K)$ over the conventional approach when ray marching. Taking into account the performance gain in photon tracing and ray marching, the overall performance gain is approximately $\log(n)/\log(n/K)$. In addition, as K increases, the performance gain increases.

One more advantage of the approach based on subsets of volume photons is that it is possible to apply robust estimation. This can be done by choosing random light sources for $\mathbf{R}_k$ in a ray marching process and building a random volume photon map $\Delta\mathbf{\Phi}_k$ through Monte-Carlo photon tracing.

6.3 Simulation

6.3.1 Performance of Progressive Volume Photon Mapping

Figure 6.2 shows the images rendered for 21K samples/pixel using the optical parameters presented in Table 2.1. For each sample, 1M volume photons were collected and the density was estimated for 10 volume photons. Therefore, these renderings are results from density estimation of 21M volume photons.

The materials used in the renderings of Figure 6.2 have high anisotropic factors, which prevent the formation of a relatively uniform distribution of photon interactions within the volume. In order to obtain uniform or smooth-changing photon densities within the volume, a large number of volume photons is necessary. Without enough volume photons, unexpected over- or under-populated spaces in the map can be introduced. In particular, the noise introduced by over-populations is more noticeable than that by under-populations because the density of an underlying population is bounded by zero but unlimited to infinity. This type of noise in path tracing is usually impulse, but it may be a pattern in volume photon mapping. This is because the noise is from continuous scattering processes within the volume. In other words, the path on which the noise propagates may be visible because the distance between scattering processes is too short. The bright streaks In Figure 6.2a are the noise introduced by the over-population of volume photons. Because the noise is from the low probability volume photons, it can be eliminated by robust estimation. As seen in Figure 6.2b, the median pixel estimator effectively eliminates these undesired streaks including speckles.

Figure 6.3 shows the execution times of volume photon tracing, k-d tree reconstruction, and multiple scattering evaluation (1 sample/pixel) for different quantities of volume photons from 1M to 1B photons. The scene setup of Figure 6.2 is used for this benchmark. As discussed in Section 6.2, the execution times of k-D tree construction increases by the proportion of $n \log(n)$ and multiple scattering evaluation increases by the proportion of $k \log(n)$. If volume photons are prepared for k-D tree construction and rendering, the single frame rendering using 1M photons takes 2.139 seconds, while that using 1B photons takes 278.006 seconds. When 1B photons are used for rendering with the above simulation setup, the progressive approach using robust estimation accelerates volume photon mapping by approximately 30%.

6.3.2 Rendering of Teeth

Teeth are heterogeneously structured with enamel and dentin, and their material characteristics change for different locations within the volumes. In addition, the materials have the characteristics of highly anisotropic scattering [4]. The simulation of the highly anisotropic scattering media is sometimes unrealistic because it is difficult to obtain a

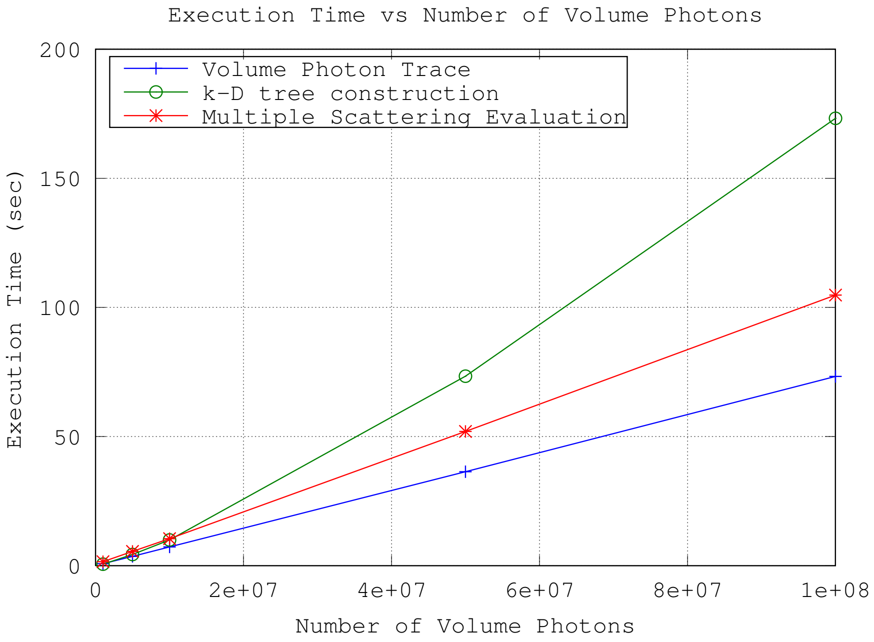


Figure 6.3: Execution time vs. number of volume photons

	Enamel	Dentin
Scattering coefficients	[0.91500 1.24600 1.86850]	[2.2058 2.8090 3.7500]
Absorption coefficients	[0.03850 0.05100 0.06400]	[0.22050 0.15450 0.17650]
Anisotropy parameter	fd = 0.6, g = 0.5	fd = 0.01, g = 0.5

Table 6.1: Optical properties of enamel and dentin for rendering

complete volume photon map; the scattering process in highly anisotropic scattering media is concentrated for an incident light ray, so the complete volume photon map may require the infinite number of photon traces. Without enough photon traces, the resulting volume photon map would suffer unexpected over- and under-population of scattering processes from place to place within the volume.

In the following simulations, each part of a tooth model is assumed to be homogeneous and to have the same material properties, and the appropriate optical parameters were chosen based on the previous studies [21, 23, 4]. In addition, the anisotropy factors were appropriately scaled for realistic simulation time. Table 6.1 shows the parameters used for rendering.

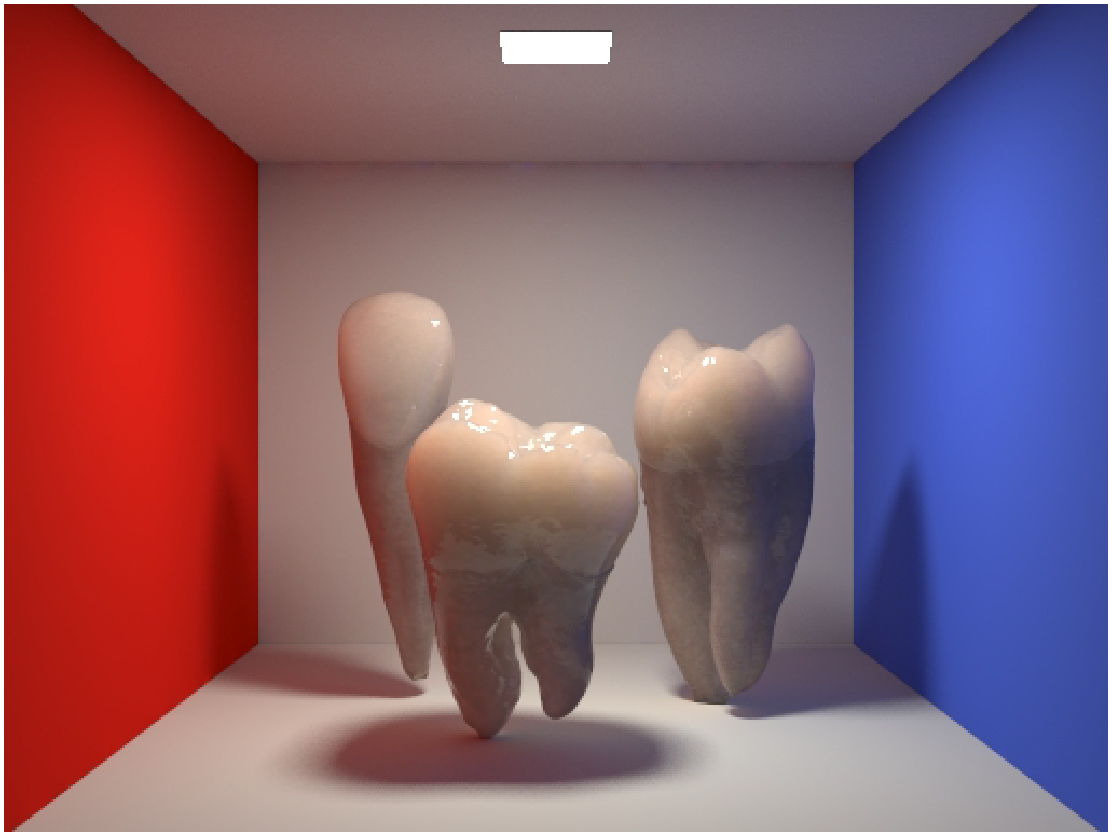


Figure 6.4: Three different teeth in Cornell box. From the left, there are tooth #10, tooth #1, and tooth #32.

Figure 6.4 is the rendition of the three different teeth (from the left, tooth #10, tooth #1, and tooth #32) within the Cornell box. In this simulation, volume photon mapping was used for rendering of translucent objects and photon mapping was employed for the effect of indirect illumination. The number of volume photons collected for this simulation is 100M and the number of photons is 1M. Due to translucency, the geometric details of the teeth appear smooth and soft. In addition, the light scattering due to the reflected light from the walls introduces the diffusion of the wall colors into the teeth. For instance, the red color reflected from the left wall introduces red into the left walls of the teeth, while the blue color reflected from the right wall brings blue into the right walls of the teeth.

Figure 6.5 illustrates how anisotropy factors influence the appearance of enamel. In

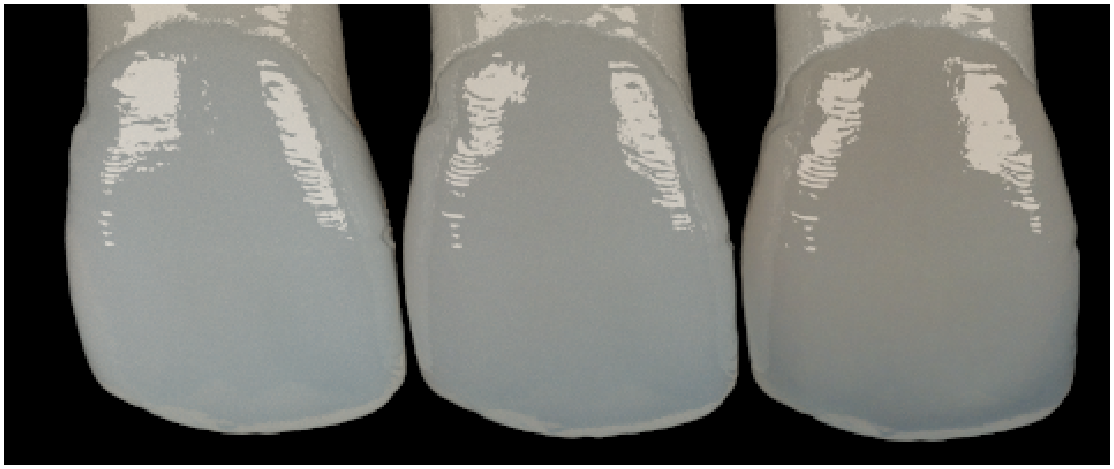


Figure 6.5: Three different translucencies for enamel. From the left, the fraction of isotropic scattering (f_d) of 0.67, 0.33, and 0.0 are used with the anisotropy factor (g) of 0.75.

this simulation, the linear combination of an HG function and an isotropic scattering fraction, which is defined in Equation 6.18, was used for the modeling of scattering direction. For the same anisotropy factor g of 0.5, different fractions of isotropic scattering f_d of 0.67, 0.33, and 0.0 are used. For the same scattering and absorption coefficients, the anisotropy factor of scattering affects the translucency of a medium. Specifically, as f_d increases, the optical thickness (opaqueness) increases because the isotropic scattering process prevents light from traveling in straight lines. On the other hand, as f_d decreases, the optical thickness decreases. Therefore, from the left to the right in Figure 6.5, the optical thickness decreases, and the underlying material and background becomes clear. Similar effects can be achieved by increasing g instead of adjusting f_d.

6.3.3 Rendering of Dental Restorative Materials

Using measured optical parameters, it is possible to simulate the realistic appearances of dental restorative materials. Figure 6.6 shows a rendering of four translucent balls (radius of 10 mm) that are modeled using the optical properties given in Table 2.1. The four different translucent balls are modeled using the same effective scattering coefficients but different anisotropy factors g: from the left, 0.00, 0.25, 0.50, and 0.75. Because the media with the same effective scattering coefficient have a similar transmittance and

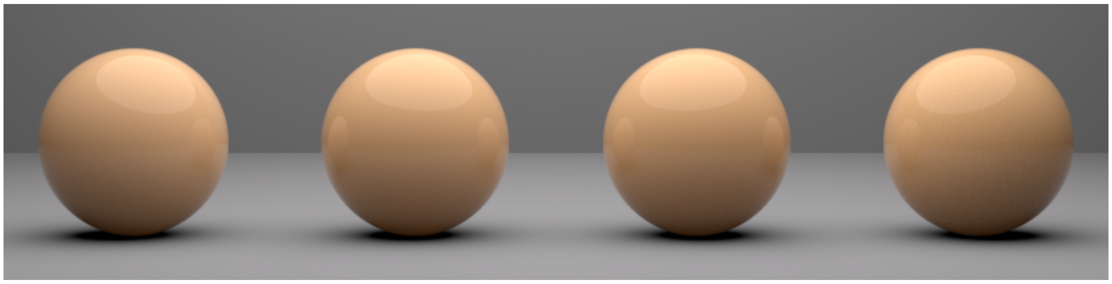


Figure 6.6: Comparison of translucent balls for same effective scattering but different anisotropy factors. From the left, the anisotropy factor g of 0.00, 0.25, 0.50, and 0.75 are used. To make their effective scattering coefficients consistent, the scattering coefficients were adjusted ($s = s'/(1-g)$).

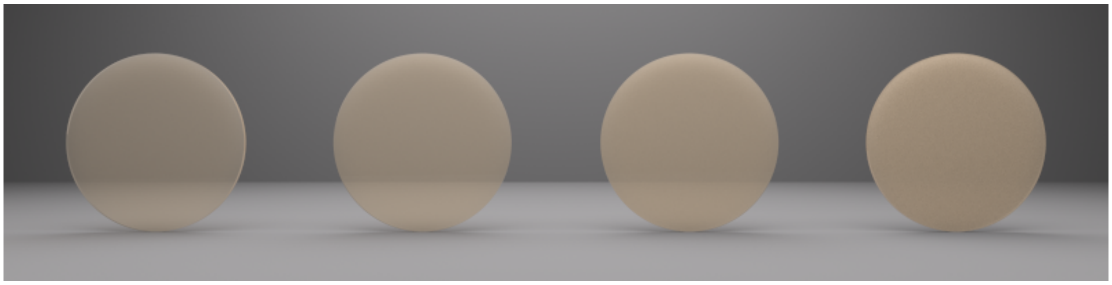


Figure 6.7: Comparison of translucent disks for different anisotropy factors. From the left, the anisotropy factor (g) of 0.00, 0.25, 0.50, and 0.75 are used. To make their effective scattering coefficients consistent, the scattering coefficients were adjusted ($s = s'/(1-g)$).

absorption of light, these four balls have a similar appearance [27].

Figure 6.7 shows the rendition for the same optical characteristics used in Figure 6.6, but for different geometries, which are disks of 0.5 mm thickness and 10 mm radius. The main noticeable difference between Figure 6.7 and Figure 6.6 is the optical thickness of the media. For instance, the disk on the left side looks more transparent than the disk on the right side. This is because the object is too thin to capture the scattering process for a long mean free path of scattered particles. On the other hand, in the disk on the right hand side, the mean free path is short enough for the particles to be captured within the volume. In addition, the higher scattering coefficients weaken the direct transmittance of the medium.

This phenomenon is clearer when the media have a distinguishable background.

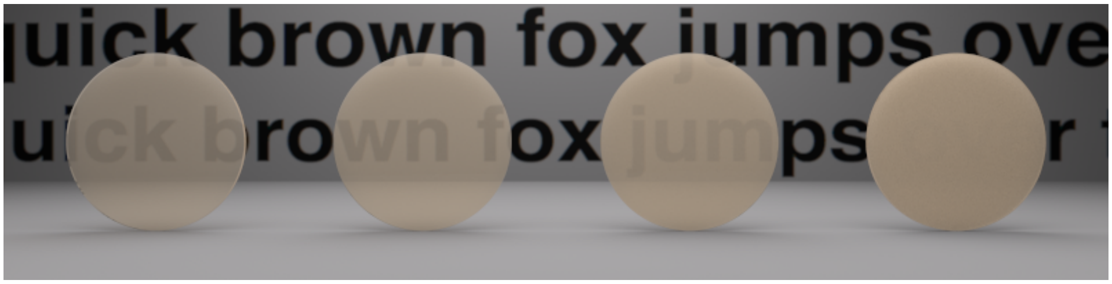
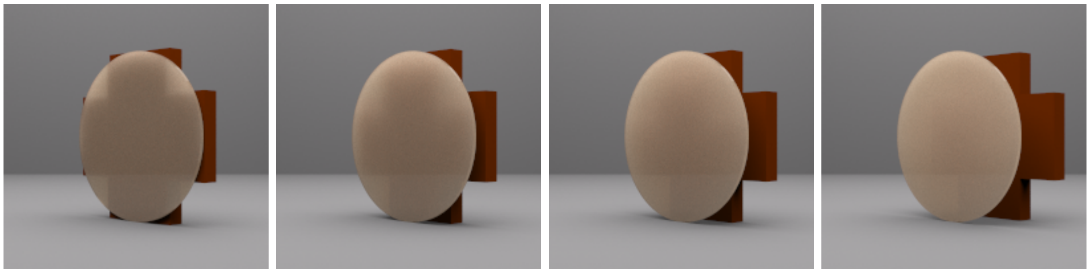


Figure 6.8: Comparison of translucent disks with a texture background for different anisotropy factors. From the left, the anisotropy factor (g) of 0.00, 0.25, 0.50, and 0.75 are used. To make their effective scattering coefficients consistent, the scattering coefficients were adjusted ($s = s'/(1 - g)$).



(a) $d = 0.16$ mm (b) $d = 1.57$ mm (c) $d = 2.99$ mm (d) $d = 5.82$ mm

Figure 6.9: Comparison of translucent disk and red cross pairs for different distances (d) between the two objects.

Figure 6.8 shows the disks with a printed text patten. The disk on the left hand side looks transparent, while one on the right hand side is clearly opaque.

These three illustrations, Figure 6.6, Figure 6.7, and Figure 6.8, show that the anisotropy factor is more influential on the overall appearance of thin media than on the appearance of thick ones. Particularly, dental restorations are thin objects and thus using an accurate model of anisotropic scattering is important for the rendering of the correct appearance.

Figure 6.9 shows the simulation of the translucent disks in front of an opaque object in order to show the influence of different distances between the two objects. As the distance increases from 0.16 mm to 5.82 mm, the object becomes indistinguishable through the disk. This is because the appearance is dominated by the concentration of the scattering process caused by the reflected photons from the red object. For the higher concentration of the scattering process, the pattern on the disk is clear, while for the lower concentration of the process, the pattern becomes blurred. This scene demonstrates that the concentration is dominated by the distance between the two objects. The pattern on the disk is formed by the reflected photons from the surface of the red objects. As the distance between the two objects decreases, the chance of the photons entering into a certain area behind the disk close to the object becomes higher. As the distance between the two objects decreases, the surface points that the photons enter become closer to the reflected points.

6.4 Discussion

This chapter discussed progressive volume photon mapping. Progressive volume photon mapping splits volume photons into multiple volume photon groups, and estimates volume densities for outgoing pixel radiance. Because multiple volume photon groups can be constructed for each sample and it is possible to discard used volume photons, this approach does not suffer from limited memory space. Due to that, it is possible to collect an effectively infinite number of photons. In addition, it is possible to apply robust estimation techniques to the multiple sub-densities in order to remove the noise introduced by the low probability over-population or under-population of volume photon densities. While this chapter discusses simulation results obtained using off-line rendering, it would be interesting to exploit the power available on contemporary graphics cards and apply the algorithms to an on-line rendering process.

Chapter 7

Conclusion and Discussion

This thesis describes the complexities associated with producing computer graphic renderings of teeth and dental restorations, and, as solutions to these problems, it presents robust estimation and progressive volume photon mapping techniques. Teeth and dental restorations are translucent, and the visualization of objects with this appearance property is a perennial challenge in computer graphics due to the high computational costs. In particular, because of the complexity of the physical interactions that occur at a translucent object's surface and within its volume, the simulation of indirect illumination and subsurface scattering requires both a large number of computations and a significant amount of memory space. In order to address these problems, this thesis suggests robust statistical pixel estimation and progressive volume photon mapping with robust estimation.

Robust estimation techniques have shown their effectiveness in the field of signal processing. In particular, for a heavy-tailed distribution of incoming samples, the robust estimator can reconstruct the source information by properly discarding outlying samples at the risk of possible biasing. The robust estimation techniques proposed in this thesis compute a relatively low variance estimate for the final pixel value. Consequently, for a given sample budget, low noise renderings can be obtained. In other words, the sample size required to get a specific rendering quality can decrease; therefore, it is possible to accelerate the rendering by reducing the sample size necessary. The robust estimation techniques are available for both pixel and image space, and even for their merged domains. In particular, robust estimation techniques in combined pixel and image space perform better than those in a single space. Furthermore, this thesis provides the statistical analysis of these techniques, which can be used to predict the performance and, for a specified requirement, to determine the optimal parameters necessary to achieve that result.

The progressive volume photon mapping technique proposed in this thesis can also be used to achieve a high frame rate for preview. Due to the fact that the complete photon map may require an essentially limitless number of photons, the conventional approach needs a long time for pre-processing and extensive memory space for holding the photons throughout the rendering. Progressive volume photon mapping splits the volume photons into multiple groups, simulates the interactions within each group, and combines the results. Because this progressive rendering approach computes a pixel value for each small volume photon group, it is able to overcome limited memory space in a given system, and it also allows the quick generation of intermediate renderings which can be used for preview. The final pixel values are determined by combining the pixels computed for different volume photon groups; therefore, it is possible to combine an effectively unlimited number of volume photon groups, which permits the generation of a correct rendering. In addition, the pixel values computed for the volume photon groups can be fed to a robust estimation process to eliminate the noise introduced by low probability volume photon samples.

To promote a computer-aided dental restoration design process, this thesis presents robust estimation and progressive volume photon mapping techniques for the purpose of interactive visualization of real teeth and dental restorations. In a computer design and fabrication process it is important to predict the appearance of an object that will be manufactured; the correct and fast prediction of the object can save the cost of the fabrication process by reducing the number of actual fabrications for intermediate evaluation. Robust estimation is expected to decrease the sample size necessary by eliminating the image noise introduced by low probability samples. The progressive volume photon mapping approach is capable of producing the correct appearance by using an essentially unlimited number of volume photons, and it is also able to provide a quick approximate preview. Consequently, these approaches are expected to help the development of a computer-aided design process in dental practice. In this thesis, the potential for interactive simulation is demonstrated using the method of off-line rendering, and actual implementation with better hardware support is left for future work.

Finally, this thesis evaluated robust estimation techniques for simple weighting schemes in order to introduce the essential characteristics of these techniques. Similar to other 2D image filters, this robust estimation approach can be extended for different filtering coefficients (eg. Gaussian, bilateral, NLM, etc) for more efficient and visually pleasing results. Even in these cases, the stochastic analysis presented for robust estimation techniques can be used to determine the pixel buffer and filtering window sizes. Furthermore, this thesis showed that rendering can be improved by choosing an appropriate estimate for the pixels; therefore, an in-depth study of the underlying radiance population for various situations (eg. different materials, lighting setups, camera setups, etc) can help in choosing the optimal estimators. Alternatively, it may even be possible to seek a sampling strategy that produces the optimum theoretical distribution suggested for each estimator. Extensions such as these can lead to a full realization of the power of the robust estimation techniques introduced in this thesis.

References

[1] Stephen J. Chu, Richard D. Trushkowsky, and Rade D. Paravina. Dental color matching instruments and systems. Review of clinical and research aspects. *Journal of Dentistry*, 38 Suppl 2:e2–16, Aug 2010.

[2] Alwin Kienle, Florian K. Forster, Rolf Diebolder, and Raimund Hibst. Light propagation in dentin: influence of microstructure on anisotropy. *Physics in Medicine and Biology*, 48(2):N7–14, Jan 2003.

[3] A. Kienle, R. Michels, and R. Hibst. Magnification–a new look at a long-known optical property of dentin. *Journal of Dental Research*, 85(10):955–959, Oct 2006.

[4] D. Fried, R. E. Glena, J. D. Featherstone, and W. Seka. Nature of light scattering in dental enamel and dentin at visible and near-infrared wavelengths. *Applied Optics*, 34(7):1278–1285, Mar 1995.

[5] Patricia Daukantas. Photorealistic rendering: Making the virtual into reality. *Optics & Photonics News*, 20:34–39, 2009.

[6] Julie Dorsey, Alan Edelman, Henrik Wann Jensen, Justin Legakis, and Hans Køhling Pedersen. Modeling and rendering of weathered stone. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 225–234, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[7] Henrik Wann Jensen, Justin Legakis, and Julie Dorsey. Rendering of wet materials. In *Proceedings of the 10th Eurographics Conference on Rendering*, EGWR'99, pages 273–282, Aire-la-Ville, Switzerland, 1999. Eurographics Association.

[8] Matt Pharr and Pat Hanrahan. Monte carlo evaluation of non-linear scattering equations for subsurface reflection. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 75–84, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[9] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Trans. Graph.*, 24(3):1032–1039, July 2005.

[10] Craig Donner and Henrik Wann Jensen. Rendering translucent materials using photon diffusion. In *Proceedings of the Eurographics Symposium on Rendering Techniques, Grenoble, France, 2007*, pages 243–251. Eurographics Association, 2007.

[11] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 311–320, New York, NY, USA, 1998. ACM.

[12] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA, 2001. ACM.

[13] Henrik Wann Jensen and Juan Buhler. A rapid hierarchical rendering technique for translucent materials. *ACM Trans. Graph.*, 21(3):576–581, July 2002.

[14] Rui Wang, John Tran, and David Luebke. All-frequency interactive relighting of translucent objects with single and multiple scattering. *ACM Trans. Graph.*, 24(3):1202–1207, July 2005.

[15] Jiaping Wang, Shuang Zhao, Xin Tong, Stephen Lin, Zhouchen Lin, Yue Dong, Baining Guo, and Heung-Yeung Shum. Modeling and rendering of heterogeneous translucent materials using the diffusion equation. *ACM Trans. Graph.*, 27(1):9:1–9:18, March 2008.

[16] Yajun Wang, Jiaping Wang, Nicolas Holzschuch, Kartic Subr, Jun-Hai Yong, and Baining Guo. Rendering I: Real-time rendering of heterogeneous translucent objects with arbitrary shapes. *Computer Graphics Forum*, 29(2):497–506, May 2010.

[17] Christopher DeCoro, Tim Weyrich, and Szymon Rusinkiewicz. Rendering: Density-based outlier rejection in Monte Carlo rendering. *Computer Graphics Forum*, 29(7):2119–2125, September 2010.

[18] Anthony Pajot, Loïc Barthe, and Mathias Paulin. Sample-space bright spots removal using density estimation. In *Proceedings of Graphics Interface 2011*, pages 159–166. Canadian Human-Computer Communications Society, 2011.

[19] Bochang Moon, Jong Yun Jun, JongHyeob Lee, Kunho Kim, Toshiya Hachisuka, and Sung-Eui Yoon. Robust image denoising using a virtual flash image for Monte Carlo ray tracing. *Computer Graphics Forum*, 32(1):139–151, February 2013.

[20] Antonio Nanci. *Ten Cate's Oral Histology-E-Book: Development, Structure, and Function*. Mosby, 2007.

[21] D Spitzer and JJ Ten Bosch. The absorption and scattering of light in bovine and human dental enamel. *Calcified Tissue International*, 17(2):129–137, 1975.

[22] Zhuo Meng, X Steve Yao, Hui Yao, Yan Liang, Tiegen Liu, Yanni Li, Guanhua Wang, and Shoufeng Lan. Measurement of the refractive index of human teeth by optical coherence tomography. *Journal of Biomedical Optics*, 14(3):034010–034010, 2009.

[23] JJ Ten Bosch and JR Zijp. *Dentine and Dentine Reactions in the Oral Cavity*. IRL Press Ltd., Oxford, 1987.

[24] Moritz Friebel, Kirsten Povel, Hans-Joachim Cappius, Jürgen Helfmann, and Martina Meinke. Optical properties of dental restorative materials in the wavelength range 400 to 700 nm for the simulation of color perception. *Journal of Biomedical Optics*, 14(5):054029, Sep/Oct 2009.

[25] Jin Woo Jung, Gary W. Meyer, Ralph DeLong, and Brian N. Holmes. Rendering of human teeth and restorative biomaterials. In *Color Imaging Conference*, pages 170–176. IS&T - The Society for Imaging Science and Technology, 2012.

[26] Matt Pharr and Greg Humphreys. *Physically Based Rendering: from Theory to Implementation*. Morgan Kaufmann, 2010.

[27] Steven H Kong and Joel D Shore. Evaluation of the telegrapher's equation and multiple-flux theories for calculating the transmittance and reflectance of a diffuse absorbing slab. *JOSA A*, 24(3):702–710, 2007.

[28] JC Ragain and WM Johnston. Accuracy of kubelka-munk reflectance theory applied to human dentin and enamel. *Journal of Dental Research*, 80(2):449–452, 2001.

[29] DJ Durian. The diffusion coefficient depends on absorption. *Optics Letters*, 23(19):1502–1504, 1998.

[30] Jos Stam. Multiple scattering as a diffusion process. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 41–50. Springer-Verlag, 1995.

[31] Craig Donner, Jason Lawrence, Ravi Ramamoorthi, Toshiya Hachisuka, Henrik Wann Jensen, and Shree Nayar. An empirical BSSRDF model. *ACM Trans. Graph.*, 28(3):30:1–30:10, July 2009.

[32] Kun Xu, Yue Gao, Yong Li, Tao Ju, and Shi-Min Hu. Real-time homogenous translucent material editing. *Computer Graphics Forum*, 26(3):545–552, 2007.

[33] Adam Arbree, Bruce Walter, and Kavita Bala. Single-pass scalable subsurface rendering with lightcuts. *Computer Graphics Forum (Proc. Eurographics 2008)*, 27(2):507–516, 2008.

[34] A. Arbree, B. Walter, and K. Bala. Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics*, 17(7):956–969, July 2011.

[35] Peter-Pike Sloan, Jan Kautz, and John Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph.*, 21(3):527–536, July 2002.

[36] Ewen Cheslack-Postava, Nolan Goodnight, Ren Ng, Ravi Ramamoorthi, and Greg Humphreys. 4D compression and relighting with high-resolution light transport

matrices. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, pages 81–88. ACM, 2007.

[37] Yubo Zhang, Zhao Dong, and Kwan-Liu Ma. Realtime volume rendering using precomputed photon mapping. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, I3D '12, pages 217–217, New York, NY, USA, 2012. ACM.

[38] Derek Nowrouzezahrai, Patricio Simari, Evangelos Kalogerakis, Karan Singh, and Eugene Fiume. Compact and efficient generation of radiance transfer for dynamically articulated characters. In *Proceedings of the 5th International Conference on Computer Graphics and Interactive Techniques in Australia and Southeast Asia*, pages 147–154. ACM, 2007.

[39] Toshiya Hachisuka, Shinji Ogaki, and Henrik Wann Jensen. Progressive photon mapping. *ACM Trans. Graph.*, 27(5):130:1–130:8, December 2008.

[40] Charly Collin, Mickaël Ribardiere, Remi Cozot, and Kadi Bouatouch. Progressive volume photon tracing. In *ACM SIGGRAPH 2012 Posters*, page 30. ACM, 2012.

[41] Dhruv Mahajan, Yu-Ting Tseng, and Ravi Ramamoorthi. Local illumination: An analysis of the in-out BRDF factorization for view-dependent relighting. *Computer Graphics Forum*, 27(4):1137–1145, June 2008.

[42] Wan-Chun Ma, Chun-Tse Hsiao, Ken-Yi Lee, Yung-Yu Chuang, and Bing-Yu Chen. Real-time triple product relighting using spherical local-frame parameterization. *The Visual Computer*, 22(9):682–692, 2006.

[43] Diego Nehab, Pedro V Sander, Jason Lawrence, Natalya Tatarchuk, and John R Isidoro. Accelerating real-time shading with reverse reprojection caching. In *SIGGRAPH/EUROGRAPHICS Conference On Graphics Hardware: Proceedings of the 22 nd ACM SIGGRAPH/EUROGRAPHICS symposium on Graphics hardware*, volume 4, pages 25–35, 2007.

[44] Carsten Dachsbacher and Marc Stamminger. Translucent shadow maps. In *Proceedings of the 14th Eurographics workshop on Rendering*, pages 197–201. Eurographics Association, 2003.

[45] Ying Song, Xin Tong, Fabio Pellacini, and Pieter Peers. Subedit: A representation for editing measured heterogeneous subsurface scattering. *ACM Trans. Graph.*, 28(3):31:1–31:10, July 2009.

[46] Milovš Hašan and Ravi Ramamoorthi. Interactive albedo editing in path-traced volumetric materials. *ACM Trans. Graph.*, 32(2):11:1–11:11, April 2013.

[47] Hendrik Wann Jensen, J Arvo, M Fajardo, P Hanrahan, D Mitchel, D Pharr, and P Shirley. State of the art in monte carlo ray tracing for realistic image synthesis. *SIGGRAPH 2001 Course Notes*, 2001.

[48] Peter J Huber. *Robust Statistics*. Springer, 2011.

[49] Lin Yin, Ruikang Yang, Moncef Gabbouj, and Yrjo Neuvo. Weighted median filters: a tutorial. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 43(3):157–192, 1996.

[50] Peter Zurbach, JG Gonzalez, and Gonzalo R Arce. Weighted myriad filters for image processing. In *IEEE International Symposium on Circuits and Systems, 1996. ISCAS'96., Connecting the World., 1996*, volume 2, pages 726–729. IEEE, 1996.

[51] Tuncer C Aysal and Kenneth E Barner. Meridian filtering for robust signal processing. *IEEE Transactions on Signal Processing*, 55(8):3949–3962, 2007.

[52] John Rice. *Mathematical Statistics and Data Analysis*. Cengage Learning, 2006.

[53] Anthony Hayter. *Probability and Statistics for Engineers and Scientists*. Cengage Learning, 2012.

[54] C. Stein B. Efron. The jackknife estimate of variance. *The Annals of Statistics*, 9(3):586–596, 1981.

[55] B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979.

[56] Songwon Seo. *A Review and Comparison of Methods for Detecting Outliers in Univariate Data Sets*. PhD thesis, University of Pittsburgh, 2002.

[57] Thomas J Rothenberg, Franklin M Fisher, and Christian Bernhard Tilanus. A note on estimation from a cauchy sample. *Journal of the American Statistical Association*, 59(306):460–463, 1964.

[58] J Bednar and T Watt. Alpha-trimmed means and their relationship to median filters. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(1):145–153, 1984.

[59] Wai-Kai Chen. *Passive, Active, and Digital Filters.* CRC Press, 2005.

[60] JF Kenney and ES Keeping. *Mathematics of Statistics Part 1. 1962.* van Nostrand, Princeton, NJ, 1962.

[61] Juan G Gonzalez and Gonzalo R Arce. Optimality of the myriad filter in practical impulsive-noise environments. *IEEE Transactions on Signal Processing*, 49(2):438–441, 2001.

[62] M Stork. Adaptive weighted meridian nonlinear filter used for filtering of signal with impulsive noise. In *Proceedings of the International Conference on Circuits, Systems, Signals*, pages 359–364, 2010.

[63] Rafael Camilo Nunez, Juan G Gonzalez, Gonzalo R Arce, and John P Nolan. Fast and accurate computation of the myriad filter via branch-and-bound search. *IEEE Transactions on Signal Processing*, 56(7):3340–3346, 2008.

[64] Sudhakar Kalluri and Gonzalo R Arce. Fast algorithms for weighted myriad computation by fixed-point search. *IEEE Transactions on Signal Processing*, 48(1):159–171, 2000.

[65] George Proctor Wadsworth and Joseph G Bryan. *Introduction to Probability and Random Variables*, volume 7. McGraw-Hill New York:, 1960.

[66] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical Recipes in C: the Art of Scientific Computation.* Cambridge University Press, Cambridge, 1992.

[67] Fabrice Rousselle, Claude Knaus, and Matthias Zwicker. Adaptive sampling and reconstruction using greedy error minimization. *ACM Trans. Graph.*, 30(6):159:1–159:12, December 2011.

[68] Bochang Moon, Nathan Carr, and Sung-Eui Yoon. Adaptive rendering based on weighted local regression. In *ACM SIGGRAPH 2014 Talks*, SIGGRAPH '14, pages 67:1–67:1, New York, NY, USA, 2014. ACM.

[69] Tzu-Mao Li, Yu-Ting Wu, and Yung-Yu Chuang. Sure-based optimization for adaptive sampling and reconstruction. *ACM Trans. Graph.*, 31(6):194:1–194:9, November 2012.

[70] James Pickands III. Statistical inference using extreme order statistics. *The Annals of Statistics*, pages 119–131, 1975.

[71] Jin Woo Jung, Gary Meyer, and Ralph DeLong. Global illumination: Robust statistical pixel estimation. *Computer Graphics Forum*, 34(2):585–596, May 2015.

[72] M. Zwicker, W. Jarosz, J. Lehtinen, B. Moon, R. Ramamoorthi, F. Rousselle, P. Sen, C. Soler, and S.-E. Yoon. State of the art reports: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum*, 34(2):667–681, May 2015.

[73] Stephen M Stigler. The asymptotic distribution of the trimmed mean. *The Annals of Statistics*, pages 472–477, 1973.

[74] Walter A Burkhard and Jai Menon. Disk array storage system reliability. In *Fault-Tolerant Computing, 1993. FTCS-23. Digest of Papers., The Twenty-Third International Symposium on*, pages 432–441. IEEE, 1993.

[75] Mahmoud Naghshineh and Mischa Schwartz. Distributed call admission control in mobile/wireless networks. *IEEE Journal on Selected Areas in Communications*, 14(4):711–717, 1996.

[76] Jesse Scott, Michael Pusateri, and Muhammad Umar Mushtaq. Comparison of 2d median filter hardware implementations for real-time stereo video. In *Applied*

Imagery Pattern Recognition Workshop, 2008. AIPR'08. 37th IEEE, pages 1–6. IEEE, 2008.

[77] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to Algorithms.* MIT press, 2009.

[78] T Huang, G Yang, and G Tang. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 27(1):13–18, 1979.

[79] Simon Perreault and Patrick Hébert. Median filtering in constant time. *IEEE Transactions on Image Processing*, 16(9):2389–2394, 2007.

[80] Athanasios Papoulis and S Unnikrishna Pillai. *Probability, Random Variables, and Stochastic Processes.* Tata McGraw-Hill Education, 2002.

[81] Henrik Wann Jensen. *Realistic Image Synthesis using Photon Mapping.* AK Peters, Ltd., 2001.

[82] Ioannis Gkioulekas, Bei Xiao, Shuang Zhao, Edward H. Adelson, Todd Zickler, and Kavita Bala. Understanding the role of phase function in translucent appearance. *ACM Trans. Graph.*, 32(5):147:1–147:19, October 2013.

[83] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[84] Ingo Wald and Vlastimil Havran. On building fast kd-trees for ray tracing, and on doing that in $O(N \log N)$. In *IEEE Symposium on Interactive Ray Tracing 2006*, pages 61–69. IEEE, 2006.

[85] Russell A. Brown. Building a balanced k-d tree in $O(kn \log n)$ time. *Journal of Computer Graphics Techniques (JCGT)*, 4(1):50–68, March 2015.

[86] Greg Hamerly and Greg Speegle. Efficient model selection for large-scale nearest-neighbor data mining. In *Data Security and Security Data*, pages 37–54. Springer, 2012.