FACULTY OF SCIENCE
UNIVERSITY OF COPENHAGEN

Ph.D. thesis

Lars Schjøth

Anisotropic Density Estimation in Global Illumination

– a journey through time and space –

Academic advisors: Jon Sporring & Kenny Erleben
Submitted: 31/08/2009

Abstract

Density estimation employed in multi-pass global illumination algorithms gives cause to a trade-off problem between bias and noise. The problem is seen most evident as blurring of strong illumination features. This thesis addresses the problem, presenting four methods that reduce both noise and bias in estimates. Good results are obtained by the use of anisotropic filtering. Two methods handles the most common cases; filtering illumination reflected from object surfaces. One methods extends filtering to the temporal domain and one performs filtering on illumination from participating media. The applicability of the algorithms is demonstrated through a series of tests.

Resumé

Et antal af globale belysnings metoder benytter sig af densitets beregninger for at tilnærme sig indirekte belysning. Denne densitets beregning foretages på et stokastisk udpluk. Under rekonstruktion af indirekte belysning, reduceres støjen fra det stokastisk udpluk af densitets beregningen. Desværre betyder denne støj reducering, at der indføres en systematisk fejl. Denne ses som en udglatning af ellers tydelige karakteristika i belysningen. Den nærværende afhandling griber dette problem an gennem fire metoder, der reducere både støjen og den systematisk fejl. Gode resultater opnås gennem rotationsvariant udglatning. To af de præsenterede metoder håndterer problemet for de mest almindelige tilfælde. En metode udglatter i tid såvel som i rum, mens den fjerde metode foretager udglatning for gennemskinnelige medier.

Acknowledgements

First and foremost, thanks to my beautiful daughter, Astrid, whom I love and cherish beyond everything. To my sister and mother.

Thanks to Nanna Glerup for her unfailing support and for mathematical proof-reading.

Also thanks to my supervisor Jon Sporring without who this Ph.D. thesis would not be. To my co-supervisor Kenny Erleben who's open-door policy I have shamelessly exploited. Likewise, thanks to Bent Dalgaard Larsen for inspiration and sparring time.

Special thanks to Jeppe Revall Frisvad for support, proof-reading and good collaboration.

Thanks to Gunhild Hornbech for de-stressing me during the last hectic days of writing.

Finally, thanks to all the Ph.D.-students in the image group on DIKU who made work a lot more lively and entertaining.

Contents

1
Introduction

Since the first photographic film was exposed and developed, reality has been a fragile concept. At the whims of spiteful dictators, ruthless journalists, or inspired designers, the minds and eyes of the people has been cheated, swayed and manipulated. Whether it is Stalin removing an unwanted commissar from official memory, or a multi-million dollar corporation performing a race-change of actors in their adverts, image manipulation has been the weapon of choice. Obviously, image manipulation raises numerous important moral issues and implications, for despite its dubious history, photography is still a source of great credibility. The significants of these moral implications are not trifle and can hardly be underestimated. They greatly impact on society as the legal system is challenged. Interesting and important as these implications are, they will not be discussed further. On the contrary, this thesis will, albeit on a very small scale, attempt to make them even more far reaching.

This thesis is a contribution to the field of photorealistic image synthesis. It presents four contributing methods; *photon differentials*, *photon differentials for participating media*, *temporal photon differentials* and *diffusion based photon mapping*. The first three methods are novel contributions developed during the research period of this Ph.D., while the latter is an improvement of a method developed as part of my master thesis.

In photorealistic image synthesis, the goal is often to graphically render virtual scenes such that they appear indistinguishable from what could be produced with real world photography. This can be archived by simulating the transport of electromagnetic radiation within the virtual scene. With this as purpose, one can attempt to solve the *rendering equation*. The rendering equation was introduced to computer graphics, from the field of radiative heat transfer, by Kajiya [1986]. A group of algorithms that attempts to solve the rendering equation are called *global illumination methods*.

If we were to broadly categorize global illumination methods, we could in one end of a spectrum place those methods that depend on stochastic sampling, eg. path tracing [Kajiya 1986], bidirectional path tracing [Yves and Willems 1993] and metropolis light transport [Veach and Guibas 1997]. These methods use monte carlo integration to solve the rendering equation. The advantage of these methods is that they converge towards the true solution. The disadvantage is that results are noisy, and that convergence is slow even when variance reduction techniques such as importance sampling are used.

Further down our spectrum, we have a large group of global illumination algorithms that employ density estimation in order to eliminate noise in the solutions. This group of algorithms has primary focus in this thesis. Specifically, these algorithms are faced with a prominent trade-off problem between variance and a systematic error (bias). The developed methods presented in the thesis improves on this trade-off. They achieve this by employing anisotropic filtering in the reproduction of indirect illumination. This procedure gives a high illumination accuracy, improving the trade-off between bias and variance considerably as compared to conventional methods. As will become apparent there is a lot to gain tackling this trade-off problem efficiently.

1.1 Thesis overview

At top-level, this thesis is split into four parts; Part I is an evaluation of the problem addressed, Part II describes methods developed to handle the most common cases, Part III extends as to handle animated scenes as well as scenes containing translucent media, and Part IV is the appendix. The order of the methods in Part II and Part III reflects the chronological development of the thesis. Here follows a more exhaustive survey of the four parts:

Part I consists of two introductory chapters. The first of these gives a brief theoretical exposition of statistical density estimation. It clarifies the trade-off problem between bias and variance, and gives a brief survey of density estimators that seek to alleviate this problem. The second chapter explains density estimation in the context of computer graphics. Specifically, it identifies the different issues, global illumination methods are faced with when employing density estimation.

Part II examines diffusion based photon mapping and photon differentials. Both methods seek to improve the trade-off between bias and variance using different forms of anisotropic filtering. Diffusion based photon mapping uses a nonlinear diffusion process to control the filtering, while photon differentials traces beams of light by differentiating their path as they propagate through the virtual scene. Photon differentials was first introduced in [Schjøth et al. 2007] and later extended in [Schjøth et al. 2009] (not yet accepted), whereas diffusion based photon mapping was introduced in [Schjøth et al. 2006] and extended in [Schjøth et al. 2008]. The last chapter in this part examines and compares the two proposed methods and a conventional global illumination method. In this analysis two different objective image quality measures are used to make the comparison as objective as possible.

In Part III two novel methods are presented, namely temporal photon differentials and photon differentials for participating media. The former, employs anisotropic filtering in the temporal as well as the spatial domain. This allows it handle certain forms of temporal aliasing. The latter extends photon differentials such that anisotropic filtering can be used to reconstruct illumination from translucent media.

Finally, Part IV is the appendix. The appendix contains a simple method for reducing a specific form of bias. Furthermore, the appendix contains a technical report by Sporring *et al.* [2009] that extends on a method by Igehy called *ray differentials* [1999]. The method and its extension is essential to many of the methods developed in this thesis. At the time of writing the technical report is still in the process of being registered.

Part I

Density Estimation

2 Statistical density estimation

This chapter aims to give an overview of density estimation. The primary contribution to this sections comes from a well-known book by Silverman [1986] and the more resent work of Wand and Jones [1995].

At top-level density estimation can be classified as either parametric density estimation or non-parametric density estimation. The purpose of non-parametric density estimation is that of finding the unknown function which best fits a set of data samples. In statistics the unknown function is the *Probability Density Function* (*pdf*). The definite integral of a *pdf*, f, defines the probability of a an event, X, within an interval;

$$P(a < X < b) = \int_a^b f(x)dx. \tag{2.1}$$

If the integral of the *pdf* stretches the entire interval it is equal one, that is–the probability that the event should happen within the interval between minus infinity to infinity is a 100%. The *pdf* can provide important information about the data it is modelled to fit and is seen used for many purposes in statistics. For instance geyser eruptions over time, suicide rate after treatment, or–in the field of computer graphics–illumination intensity over a scene.

Parametric density estimation is another approach to finding the *pdf* of a dataset. In parametric density estimation a known function is modeled as to best fit a density of discrete samples. This approach might yield bad results when prior knowledge of the dataset is not adequate. If for example a dataset best resembling a bimodal *pdf* is described as a normal distribution, then the multimodal nature of the dataset would be missed. If on the other hand prior knowledge is adequate, and a simple function fits the data, then the method can be much faster than the non-parametric density estimate.

We will limit ourselves to the non-parametric case as this is the one primarily used in computer graphics, the reason being that the illumination of a scene seldom can be described adequately by a simple function. Henceforth, when referring to density estimation, it should be assumed to be non-parametric density estimation.

Kernel	$K(x)$	
Uniform	$\frac{1}{2}$ 0	if $\lvert x\rvert < 1$, otherwise
Epanechnikov	$\frac{3}{4}(1-x^2)$ 0	if $\lvert x\rvert < 1$, otherwise
Biweight	$\frac{15}{16}(1-x^2)^2$ 0	if $\lvert x\rvert < 1$, otherwise
Gaussian	$\frac{1}{\sqrt{2\pi}}\exp\big(-\frac{1}{2}x^2\big)$	

Table 2.1: Common univariate kernel functions

2.1 The kernel estimator

A general non-parametric density estimator is the kernel estimator. The kernel estimator approximates a density function by weighting the samples of a dataset by their distance to the position, for which the density function is to be approximated. This is done using a kernel function.

The univariate kernel estimator is defined by

$$\widehat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \qquad \boxed{2.2}$$

where n is the number of data, h is the bandwidth, and K is the kernel function. For each position x of $\widehat{f}(x)$, all samples, x_i, are weighted by the kernel function centered over x.

The kernel estimator can be understood in two ways; either as a sum of kernel functions, each centered over a sample, x_i, or as a kernel centered over each position x of $\widehat{f}(x)$, such that it weights each sample, x_i, by its distance.

The kernel function should integrate to one,

$$\int_{-\infty}^{\infty} K(x)dx = 1, \qquad \boxed{2.3}$$

as the kernel density estimate itself otherwise would not integrating to one and therefore would not be a pdf. The kernel function is most often itself a pdf and it is usually symmetric and unimodal. Table 2.1 is a table of different univariate kernel functions common to statistics. With the exception of the Gaussian kernel they all reach zero at a given limit. The kernel functions are illustrated in Figure 2.1 as well.

Because of the additive form of the the kernel estimator, the estimated density function inherits the differentiability and continuity of the kernel function used in the estimate. Another property of the kernel estimator is that it constitutes a smoothing of the density function. In fact the estimated density function converges not to the true density function but to the true density function convolved with the kernel.

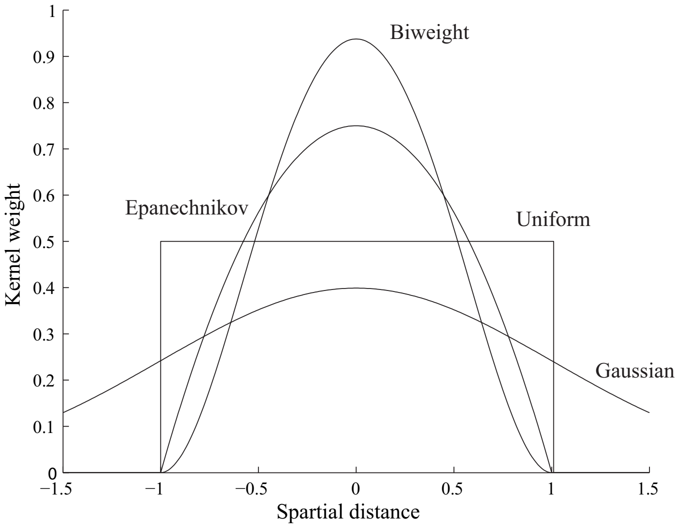


Figure 2.1: The Uniform, the Epanechnikov, the Biweight, and the Gaussian kernel function.

The bandwidth determines the degree of smoothing of the density function, it has a huge impact on its form. In the next section, we will investigate its impact on the kernel estimator using a virtual scene.

2.1.1 A simple analysis

Figure 2.2 is a rendering of a constructed scene used in this and later analyses. In this artificial scene we have a wave, which is illuminated from above by collimated light. Light from the light source (outside the image) is refracted through the wave creating a caustic on the plane beneath. Because the focus point of the light is beneath the plane, envelopes of light creates two narrow bands of high intensity illumination. Between the two bands, the illumination intensity is higher than outside. Along the length of the plane, the caustic in one side slowly fades out. On the opposite side, the caustic is cut off because of the limited extend of the wave. Along the width of the plane, the caustic is on one side cut off by the border of the plane and on the other by the wave. The vertical box can be ignored in this part of the analysis. The red line along the length of the plane is an artificial overlay its function to be explained later.

In the rendering only indirect illumination is visualized. The illumination has been reconstructed from a large distribution of samples that has been gathered by stochastically emitting light rays from the light source.

Reducing the reconstruction problem to one dimension we sample a slice of the distribution along the red line in Figure 2.2. This gives us the distribution illustrated in Figure 2.3.

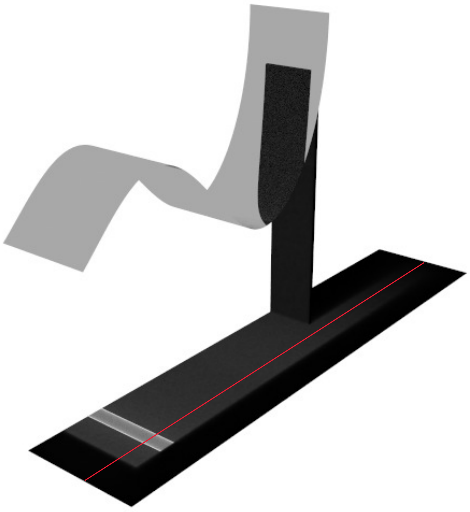


Figure 2.2: The figure is a rendering of a virtual scene, in which a wave is illuminated from above by collimated light. Refraction of light creates a caustic on the plane beneath the wave. In the rendering only indirect illumination is shown.

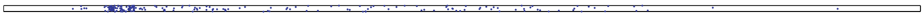
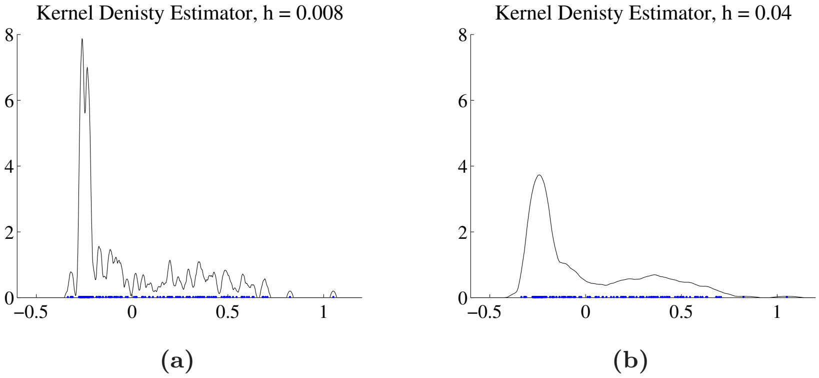


Figure 2.3: Part of the distribution used to reconstruct the caustic in Figure 2.2. The part was sampled from the distribution along the red line.



(a) **(b)**

Figure 2.4: Kernel density estimates of the distribution shown in Figure 2.3. The function plotted in (a) was estimated with a bandwidth of 0.008 while (b) was estimated with a bandwidth of 0.04. Both estimates was done using the Epanechnikov kernel.

The two plots in Figure 2.4 are univariate kernel density estimates of the distribution in Figure 2.3. They illustrate that the bandwidth size has an important impact on form of the estimator, $\widehat{f}$. In fact the plots reveal an important problem inherent to density estimation; namely that of selecting a bandwidth that gives the most suitable result. The rather imprecise term 'suitable' is used, because the suitability of the result depends on its utilization. A human observer might want an estimate with details, while an estimate used for differentiation preferable is noise free.

The first estimate, Figure 2.4(a), has been estimated with a small bandwidth. It contains a lot of details including the two peaks corresponding to the two high intensity bands in the rendering, Figure 2.2. We observe that a lot of noise is visible along the tail of the plot. Must obvious are the last two samples in the right side of the function; these each stand alone as the bandwidth is much smaller than the span in-between these, and between these and the rest of the function.

One way to eliminate noise in the estimate is to increase the bandwidth and thereby the smoothing. This has been done in Figure 2.4(b). Here we see that the tail noise is more or less gone such that the gradual fadeout is more smooth. Unfortunately, two ill-effects of the smoothing has also become apparent: first of all the bimodal form of the function is gone as the two peaks has 'melted' together; secondly, the high intensity part of the curve is much reduced, likewise has contrasts between high and low intensity parts been leveled out.

This analysis suggest that a wish for detail preservation is at odds with the noise removal. In the following a less heuristic analysis is used to identify this common kernel density estimation problem.

2.1.2 Estimation accuracy

In statistics an objective mathematical measure is often used to quantify the accuracy of an estimate. Such a measure can be used to define the discrepancy of an estimator, $\widehat{f}$, compared to the true density, f.

Most often there is a trade-off between variance and bias. Variance is the random error caused by the finite nature of the distribution, while the bias is the systematic error induced by the density estimator. If large values for the bandwidth are used, a smoothing of the density function is incured. The result is a reduction in variance, but an increase in bias. Small values on the other hand means less smoothing, more variance, and a less biased result. This trade-off can be quantified using the *Mean Integrated Square Error* (MISE).

MISE can be defined as

$$\mathrm{MISE}(\widehat{f}) = \int \left(E\widehat{f}(x) - f(x)\right)^2 \mathrm{d}x + \int \operatorname{var} \widehat{f}(x) \mathrm{d}x, \qquad \boxed{2.4}$$

where the first integral is the square bias, the second is the variance, and E is the expectation value. Accordingly, the accuracy of the estimate depends on the sum of the variance and the square bias, thus a dependency exists between variance and bias. This will be elaborated.

It is known [Silverman 1986] that the expectation value of the kernel density estimate is

$$E\widehat{f}(x) = \int \frac{1}{h} K\left(\frac{x-y}{h}\right) f(y)dy. \tag{2.5}$$

Together with Equation 2.4, this tells us that the difference between the expected value of $\widehat{f}$ and f is the convolution of f with the kernel, K, and f itself. This difference depends on the size of the bandwidth, since it scales the kernel. It is possible to express MISE in a form such that the dependency on the bandwidth is more tractable.

Restricting the kernel function such that it satisfies

$$\int K(y)dy = 1, \int yK(y)dy = 0 \text{ and } \int y^2 K(y)dy < \infty, \tag{2.6}$$

it is possible to formulate the expected value of $\widehat{f}$ by Taylor expansion as

$$E\widehat{f}(x) \approx \frac{1}{2} h^2 f''(x) \int y^2 K(y)dy + o(h^2). \tag{2.7}$$

Similarly, the variance of the kernel density estimate can be approximated:

$$\mathrm{Var}(\widehat{f}(x)) \approx \frac{1}{nh} \int K(y)^2 dy f(x) + o\left((nh)^{-1}\right). \tag{2.8}$$

Collectively, Equation 2.7 and 2.8 makes it possible to expand Equation 2.4, this leads to an extensive formulation of the mean square error of the kernel density estimator. Wand and Jones [1995] has reduced this expression, by removing higher-order terms, formulating a useful approximation, which they call the *asymptotic mean integrated square error* (AMISE). By definition the asymptotic mean integrated square error of the kernel density estimator is

$$\mathrm{AMISE}(\widehat{f}) = \frac{1}{4} h^4 \int y^2 K(y)dy \int f''(x)^2 dx + \frac{1}{nh} \int K(y)^2 dy, \tag{2.9}$$

and from this the mean integrated square error is expressed as

$$\mathrm{MISE}(\widehat{f}) = \mathrm{AMISE}(\widehat{f}) + o\left((nh)^{-1} + h^4\right). \tag{2.10}$$

From the definition of AMISE, the influence of the bandwidth is more easily interpreted. In the equation the contribution from the first term is the integrated square bias, while the contribution from second term is integrated variance. As can be seen, the integrated variance is inversely proportional to the nh. That is, we need to decrease the number of samples or the bandwidth in order to decrease the variance. However, increasing the bandwidth has a huge impact on the bias as the integrated square bias is proportional to h^4. In other word increasing the bandwidth will decrease the error caused by bias. Unfortunately, decreasing the bandwidth will increase the error caused by variance. This is the trade-off problem between variance and bias that we considered in our simple analysis, Section 2.1.1.

Another point to consider is the influence of f'' on the bias. The magnitude second derivative of f is large, where f changes rapidly. It is a measure of the curvature of f. This means the more 'spiky' the function is the lower the bandwidth needs to be in order to keep the bias low. This of course conflicts with the error caused by variance, which will increase as the bandwidth decreases. A natural conclusion is that high contrast functions needs more samples to keep the overall error down.

Returning to the trade-off problem, a balance needs to be found between variance and bias. For different measures a bandwidth can be found, which gives the optimal balance for that particular measure.

2.1.3 Optimal bandwidth

From Equation 2.9 it is possible to find, which bandwidth for the kernel density estimator that reduces the asymptotic mean integrated square error best. Wand and Jones [1995] has derived the optimal AMISE bandwidth;

$$h_{opt} = \sqrt[5]{\frac{\int K(y)^2 dy}{n \left(\int y^2 K(y) dy\right)^2 \int f''(x)^2 dx}}. \tag{2.11}$$

The usefulness of this equation is limited by the fact that the optimal bandwidth depends on the unknown function f''. As expected, the optimal bandwidth is decreased if the curvature of f is high. Likewise, the optimal bandwidth decreases, if the number of samples in the estimate increases.

One way to circumvent the unknown term in Equation 2.11 is to assume that the distribution describes some known function such as the a normal function. If, however, the unknown function for example is mulitmodal, then estimating it based on a normal function will over smooth the estimate, perhaps causing the mulitmodal nature of the function to be lost.

As we shall touch on later, AMISE might not be the best measure upon which to balance the trade-off between bias and variance.

2.1.4 Summary

A kernel density estimate $\widehat{f}$, is a kernel smoothed version of the true density function, f, in addition to random error. Increasing the number of samples will reduce the variance making the estimate converge to the true density function convolved with the kernel. If the bandwidth at the same time goes toward zero, then our estimate will converge to f. Adjusting the bandwidth controls the trade-off between bias and variance.

It is possible to improve this trade-off by adapting the bandwidth over an estimate such that it is inversely proportional to the local density. A class of estimator that tries to achieve this is called *adaptive kernel estimators.*

2.2 Adaptive kernel estimators

Adaptive kernel estimators seek to minimize MISE by varying the bandwidth over the estimate. In areas with low density, adaptive kernel estimators reduce noise by using a broad bandwidth, while they at the same time preserve details in high density areas by using a small bandwidth.

Two well-known adaptive kernel estimators are the *local-bandwidth kernel estimator* and the *variable-bandwidth kernel estimator* [Simonoff 1996].

2.2.1 The local-bandwidth kernel estimator

The equation for the local-bandwidth method can be defined as,

$$\widehat{f}(x) = \frac{1}{nh(x)} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h(x)}\right), \tag{2.12}$$

where n is the total number of samples, and $h(x)$ is a bandwidth function.

The most common local-bandwidth kernel estimator is the k'th nearest neighbor kernel estimator. For this estimator, the bandwidth function uses the distance from x to the k'th nearest sample as the bandwidth. This means that the bandwidth varies depending on the local density of the data. In effect the bandwidth changes from being small in high density areas to being large in low density areas. The result is that bias seen in areas with sharp transitions in density is reduced.

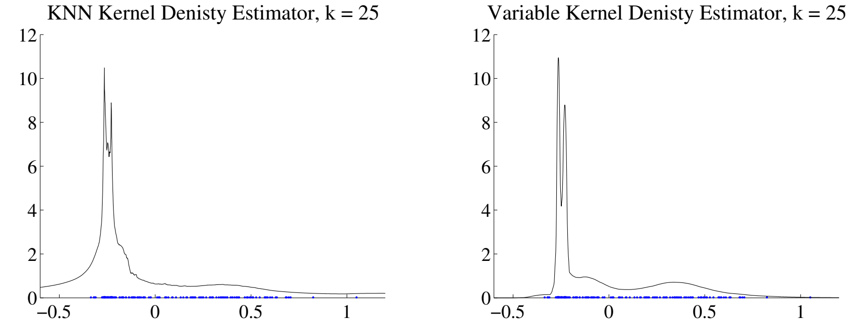
Figure 2.5(a) shows a k'th nearest neighbor kernel estimate of the distribution in Figure 2.3. As can be seen from the curve, the bimodal nature of the function is preserved, while the tail is still smooth. Unfortunately, in both sides the curves continues to the limit of the plot without becoming zero. This is a general problem as the k'th nearest neighbor kernel estimator will never become zero, even outside the domain. Another problem with the k'th nearest neighbor kernel estimator is that, unlike the kernel density estimator, it does not inherit the differentiability and continuity of the kernel function used in the estimate, since the distance function is not continuous.

2.2.2 The variable-bandwidth kernel estimator

The equation for the variable-bandwidth kernel estimator is,

$$\widehat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h(x_i)} K\left(\frac{x - x_i}{h(x_i)}\right). \tag{2.13}$$

Whereas the k'th nearest neighbor estimator adapts the local bandwidth based on a measure of density, the bandwidth varying kernel estimator uses a different bandwidth for each observed sample, x_i, to weight the estimate. Most often a



(a) **(b)**

Figure 2.5: Estimates of the distribution shown in Figure 2.3. The function plotted in (a) was estimated with the k'th nearest neighbor kernel estimator using $k = 25$, while (b) was estimated with variable-bandwidth kernel estimator using the k'th nearest neighbor kernel estimator to estimate the individual kernel bandwidths with $k = 25$. Both estimates was done using the Epanechnikov kernel.

pilot estimate of the unknown density is used to vary the bandwidth [Simonoff 1996]. With $h(x_i) = h_v \tilde{f}(x_i)^{-1/2}$. Where the pilot, $\tilde{f}$, is estimated with a fixed-bandwidth kernel estimator.

In Figure 2.5(b) we have used the k'th nearest neighbor kernel estimator to decide the bandwidth of each individual sample and then used variable-bandwidth kernel estimator to estimate the function from the distribution in Figure 2.3. In this simple case the variable-bandwidth kernel estimator exhibits superior qualities. Both high intensity peaks are preserved, the tail is smooth and curve reached zero in both ends. Like the kernel density estimator, it, furthermore, inherits the differentiability and continuity of the kernel function used in the estimate.

The obvious drawback of the variable-bandwidth kernel estimator is that it can be expensive to estimate bandwidths for each individual sample.

2.2.3 Multivariate kernel density estimation

Expanding the number of dimensions complicates matters as the kernel shape achieves more degrees of freedom. This means that a number of bandwidths are needed to control kernel shape and size. Defining these bandwidths with a matrix, $\boldsymbol{H}$, the multivariate kernel density estimator can be expressed as

$$\widehat{f}(\mathbf{x}) = \frac{1}{n\sqrt{\det \boldsymbol{H}_{d\times d}}} \sum_{i=1}^{n} K_d\Big((\mathbf{x}-\mathbf{x}_i)^T \boldsymbol{H}_{d\times d}^{-1}(\mathbf{x}-\mathbf{x}_i)\Big), \qquad \boxed{2.14}$$

where d is the dimensionality of the estimator and K_d is a multivariate kernel function. The matrix $\boldsymbol{H}$ is called the *bandwidth matrix* [Simonoff 1996; Wand and Jones 1995]. It should be symmetric and positive semidefinite.

Defined as a diagonal matrix, the bandwidth matrix controls the kernel shape and size along the coordinate axes. The full matrix lets the kernel assume any orientation. Restricting the bandwidth matrix to $\boldsymbol{H} = h^2\boldsymbol{I}_d$, Equation 2.14 is reduced to

$$\widehat{f}(\mathbf{x}) = \frac{1}{nh^d}\sum_{i=1}^{n} K_d\left(\frac{(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{h^2}\right) \tag{2.15}$$

where the single parameter, h, is the bandwidth the multivariate kernel K.

The important difference between Equation 2.14 and 2.15 is that the former allows for an anisotropic kernel to be used in the estimate, while the latter uses an isotropic kernel.

2.3 Summary

In this introductory chapter non-parametric density estimation has been examined. From analyses it was shown that a trade-off exists between bias and variance and that this trade-off can be balanced using an objective quality measure. Furthermore, this chapter has described two adaptive density estimation methods that seek to improve the mentioned trade-off. In the following chapter we shall see how kernel density estimation is employed in graphics and what issues its usage effects.

3

Density Estimation in Computer Graphics

As mentioned in the introduction, a group of unbiased global illumination methods exists, which is based primarily on Monte Carlo ray-tracing. The advantages of the these methods is that they can simulate all possible light paths, they have low memory consumption, they can be used on scenes with arbitrary geometry, and they converge towards the true solution. Unfortunately, they converge slowly. As an alternative, a group of biased global illumination methods uses density estimation to achieve a faster convergence rate; the most popular of these being photon mapping [Jensen and Christensen 1995]. By means of density estimation the illumination function is convolved with a smoothing kernel. The advantage is that noise is removed, but as explained in the Chapter 2, the trade-off is the introduction of a systematic error (bias).

The images in Figure 3.1 illustrates how this trade-off between variance and bias manifests itself in computer graphics. Both images were rendered with photon mapping with the same number of samples, but with different bandwidths. The left image was rendered with a small bandwidth, for which reason noise in the indirect illumination stands out. In the right image a large bandwidth was used resulting in a low noise level. The smoothing incurred by the large bandwidth can be quite pleasing especially where the illumination changes slowly. However, when the illumination contains high contrasts as in the vicinity of caustics and shadows, details are lost. In the right image, details and even whole parts of the caustics, otherwise visible in the left image, has disappeared.

This bias defect is one of the most common and most visible. However, as we shall see in the following, bias imposed by kernel density estimation is cause to other types defects often seen to computer graphics.

3.1 Types of bias

When density estimation is used in computer graphics, a number of bias related effects may follow. The effects and the degree to which they are expressed,

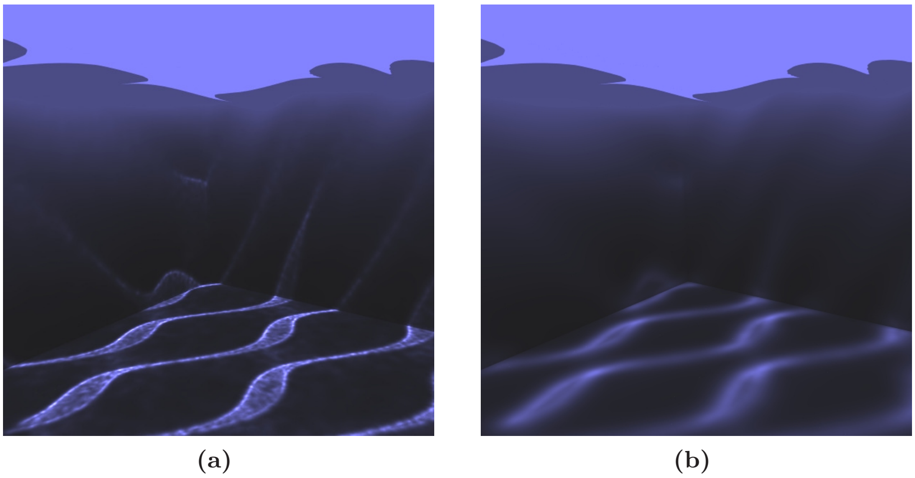


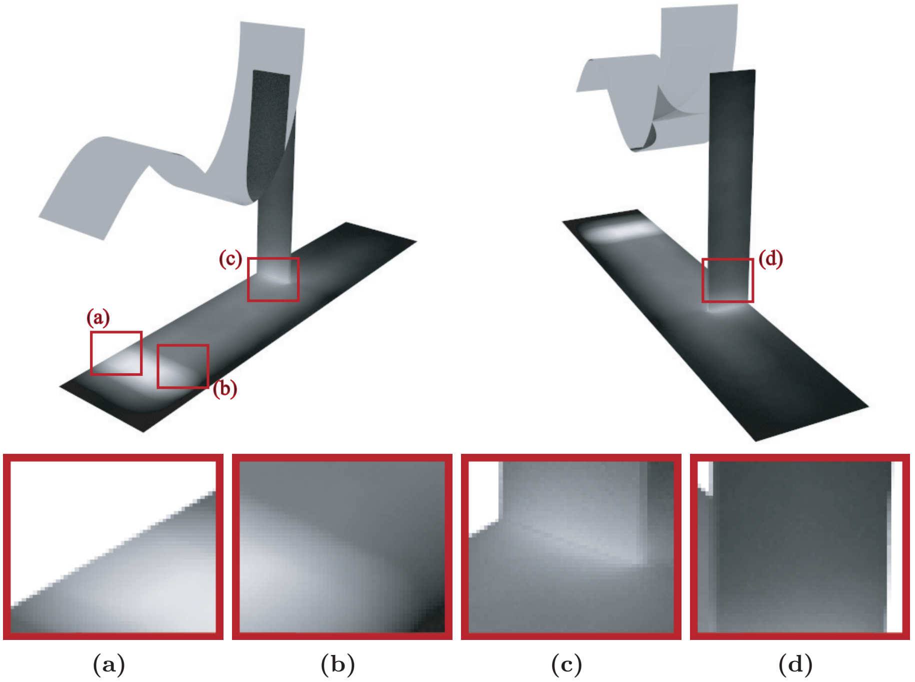
Figure 3.1: Underwater images of a stair. The images demonstrate how biased global illumination methods often need to balance between noise and blurring. Both images was creates with photon mapping; image (a) was rendered using a small bandwidth while (b) was rendered using a large bandwidth.

depend on the global illumination method. It is possible to categorize these effects. We will adopt the terminology used by Schregle in [Schregle 2003] and supplemented by Herzog *et al.* in [Herzog et al. 2007a]. As such, density estimation, when employed in computer graphics, gives cause to *boundary bias*, *proximity bias*, *topological bias*, and *occlusion bias*.

Boundary bias is seen as a darkening of illumination along polygon edges. It exists because the density estimate extends beyond the boundaries of the particle distribution. That is, the estimator assumes the distribution to be unbounded. This problem is similar to the boundary problem seen when filtering images in Image Analysis. In Figure 3.2(a) it is seen as a darkening right before the caustic extends beyond the object upon which it is reflected.

Proximity bias is visible as blurring of edges of caustics and other sharp illumination features. It occurs, when the kernel crosses the boundary of such illumination features. The reason being, that energy from both sides of the boundary contributes to the local estimate, thus displacing energy to areas where it should not be present. This is the basic density estimation bias problem explained in Section 2.1. The effect of proximity bias is illustrated in Figure 3.2(b), where the edge of the caustic slowly fades out.

Topological bias sometimes occurs when density estimation based global illumination algorithms assume that geometric surfaces are locally planar. This assumption often means that a two dimensional kernel density estimate is performed on a three dimensional dataset. The result is that when surface curvature



(a) (b) (c) (d)

Figure 3.2: Rendering of a virtual scene. A wave is illuminated from above by collimated light creating a caustic on the plane beneath. The rendering demonstrates four common bias defects (highlighted in red): (a) boundary bias, (b) proximity bias, (c) topological bias, and (d) occlusion bias. In the rendering only indirect illumination has been shown.

is strong, the density is overestimated, as the support area is underestimated. Typically, this defect is visible in corners. Figure 3.2(c) is an example of topological bias. Along the border between the floor and the wall, the caustic has an artificial increase of intensity; this is topological bias and is due to the fact that when the illumination of the wall is estimated, samples from the floor are also used in the estimate and vice versa. If only samples from either the wall or the floor had been used, then the result would have been a darkening of illumination caused by boundary bias.

Occlusion bias shows itself as light leakage. In Figure 3.2(d) the backside of the thin vertical box is illuminated, even though it should not (only indirect illumination resulting directly from specular refraction is visualized). This false illumination occurs, because samples from the other side of the thin vertical box are included in the density estimate.

This terminology will be used henceforth to distinguish between different types of bias. The next section will explain density estimation, as it is used in photon mapping. It will help build the foundation for our developed methods.

3.2 Density estimation in photon mapping

In photon mapping, indirect illumination is reconstructed through a series of queries to the photon maps. A photon map is a collection of “photons” created during the particle tracing phase–a phase in which photons are reflected around a scene using Monte Carlo ray tracing. Each query is used to estimate the reflected radiance at a surface point as the result of a local photon density estimate. This estimate is called *the radiance estimate*.

The accuracy of the radiance estimate is controlled by two important factors: the resolution of the photon map, and the number of photons used in each radiance estimate. If few photons are used in the radiance estimate, then noise in the illumination becomes visible. If many photons are used, then edges and other sharp illumination features, such as those caused by caustics, are blurred. It is impossible to avoid either of these adverse effects, unless an excessive number of photons are stored in the photon map. This is the mentioned trade-of problem between variance versus bias as it manifests itself in photon mapping.

3.2.1 The radiance estimate

In his book Jensen [2001] derives an equation that approximates the reflected radiance at a point, $\mathbf{x}$, using the photon map. This is done by rewriting the reflected radiance term of the rendering equation such that it involves an integral over radiant power incident per unit area rather than radiance incident across the hemisphere. An approximation of the radiant power incident per unit area is obtained using the k nearest photons around the point, $\mathbf{x}$. In this way the equation for the reflected radiance becomes

$$L_r(\mathbf{x},\omega) \approx \widehat{L}_r(\mathbf{x},\omega) = \frac{1}{\pi h(\mathbf{x})^2}\sum_{i=1}^{k} f_r(\mathbf{x},\omega_i,\omega)\Phi_i, \qquad \boxed{3.1}$$

where Φ_i is the radiant power represented by the i'th photon, f_r is the *bidirectional reflectance distribution function* (abbreviated BRDF), and $h(\mathbf{x})$ is the radius of a sphere encompassing the k nearest photons, such that $\pi h(\mathbf{x})^2$ is the sphere's cross-sectional area through its center. The radius is dependent on $\mathbf{x}$ because its size is decided by the photon density in the proximity of $\mathbf{x}$. In the context of density estimation, the radiance estimate is a bivariate isotropic k'th nearest neighbor kernel estimate with $h(\mathbf{x})$ as the bandwidth.

As mentioned, the bandwidth is important because its size controls the trade-of between variance and bias. A small bandwidth gives a limited support of photons in the estimate; it reduces the bias, but increases the variance of the estimate. Inversely, estimating the radiance using a large bandwidth results in an increase in bias and a decrease in variance.

Using a k'th nearest neighbor search to decide the bandwidth, Jensen helps limit bias and variance in the estimate by smoothing more, where the photon density is sparse, and less where the photon density is dense.

The radiance estimate in Equation 3.1 is simple insofar as it weights each photon in the estimate equally. In the thesis by Jensen [1996] the radiance estimate is refined such that filtering is used to weight each photon according to its distance to the point of estimation.

It is possible to reformulate the radiance estimate to a general form such that it can be used with different filtering techniques. We formulate this general radiance estimate as

$$\widehat{L}_r(\mathbf{x},\omega) = \frac{1}{\pi h(\mathbf{x})^2}\sum_{i=1}^{k} K\left(\frac{(\mathbf{x}-\mathbf{x}_i)^T(\mathbf{x}-\mathbf{x}_i)}{h(\mathbf{x})^2}\right) f_r(\mathbf{x},\omega_i,\omega)\Phi_i, \qquad \boxed{3.2}$$

where $\mathbf{x}_i$ is the position of the i'th photon and $K(\mathbf{y})$ is a bivariate kernel function.

3.3 Bias reduction

As density estimation became relevant to computer graphics, so did the trade-of problem between variance and bias. Numerous papers address this issue; some of these go beyond common kernel density techniques.

3.3.1 Local-bandwidth based methods

Photon mapping usually depends on k'th nearest neighbor kernel estimate to improve the trade-of between bias and variance. However, Jensen [1995] proposed an extended method. The method is called *differential checking*, and it reduces bias by making sure that the kernel does not cross boundaries of distinct lighting features. This is done by expanding the bandwidth; ensuring that the estimate does not increase or decrease drastically, when more photons are included in the estimate.

Myszkowsky [1997] suggested to solve the problem in much the same way as Jensen did with differential checking. However, he made the method easier to control and more robust with respect to noise. Myszkowsky increases the bandwidth iteratively, estimating the radiance in each step. If new estimates differ more from previous estimates than is attributable to variance, the iteration stops as the difference is assumed to be caused by bias. More recently, Schregle [2003] followed up Myszkowskys work using the same strategy, but optimizing speed and usability. Speed is optimized by using a binary search for the optimal bandwidth. This search starts in a range between a maximum and a minimum user-defined bandwidth. The range is split up, and the candidate whose error is most likely to be caused by variance, not bias, is searched.

Redner *et al.* [1995] used b-splines to approximate the illumination function from a particle density distribution. The b-spline function is composed of a number of basis functions, each associated with a control point. The advantage of this form of representation is that the illumination function is easy to evaluate and manipulate and that the storage consumption is negligible. The method is faced with the same dilemma as the kernel density estimator as the number of basis functions used in the representation determines the smoothness of the illumination function.

Shirley *et al.* [1995] introduced an algorithm for estimating global illumination. Like photon mapping this algorithm uses density estimation to approximate the illumination from particles generated during a Monte Carlo-based particle tracing step. However, unlike photon mapping the algorithm is view-independent, and for this reason the illumination is tied to the geometry. Kernel bandwidths was estimate for groups of photons per surface by a simple heuristic; namely, the square root of the surface area divide by the surface hits multiplied by a user defined constant. They called the algorithm *the density estimation framework*, and they refined it in a series of papers.

Bias control was not considered in the first edition of their framework, but in the paper by Walter *et al.* [1997] they extended the framework to handle bias near polygonal boundaries. This was done by converting the density estimation problem into one of regression. In this way they could use common regression techniques to eliminate boundary bias.

Later, in his PhD thesis, Walter [1998] reduced bias by controlling the bandwidth of the estimate using statistics to recognize noise from bias. Benefiting from the field of human perception he used a measure for controlling the bandwidth such that noise in the estimate was imperceptible to the human eye.

Walter recognized that if bias was to be significantly reduced using his method, then perceptual noise had to be accepted in the vicinity of prominent edges and other strong lighting features. This is a common problem that also affects differential checking and both Schregle's and Myszkowsky's method. Hence, in the proximity of strong features such as the edges of a caustic the bandwidth stops increasing, and the foundation on which the estimate is made is supported by few photons. This means that when estimates are made close to edges, the support is limited and noise may occur.

The difference between Jensen's, Myszkowsky's, Schregle's, and Walter's algorithms mainly lie in their method of detecting structure and the degree of change in bandwidth a given proximity of such structure entails. They can be classified as local-bandwidth estimators that use isotropic kernels.

Schjøth *et al.* [2006; 2007; 2008] suggested a bias reducing method inspired by diffusion filtering. Their method use a structure tensor to shape-adapt the kernel of the density estimate in order for it to smooth along edges and structures. The structure tensor is constructed from the first order structure of the photon map, which is estimated in an in-between pass. Similar to Jensen's, Myszkowsky's, Schregle's, and Walter's methods, their method can be classified as a local-bandwidth kernel estimator, the difference being that they use an anisotropic kernel.

3.3.2 Photon splatting

A group of methods uses the concept of splatting. Common for these methods is that they employ some form of the variable bandwidth density estimator; each sample (eg. photon hit) is associated with a bandwidth that is used to estimate its contribution to the illumination. The main difference between these lie in how they selects kernel bandwidths.

In comparison to the other density estimation methods these are also faced with difficulty of finding an optimal bandwidth, however, the problem is aggravated because local density data typically is not available when estimating the radiance.

Much like Shirley *et al.* [1995], Stürzlinger and Bastos [1997] also used splatting. They employed the same heuristic to estimate the photon-bandwidth, but aimed for real time rendering. Also, their method was geometry dependent, and photons was grouped and given the same bandwidth depending on the surface they had hit.

Their method was extended by Lavignotte and Paulin [2003] who made the method adaptive. In a first iteration a pilot estimate is found using a large fixed bandwidth. For subsequent iteration bias and variance in each pixel is estimated. If variance bias was low the pixel was accepted. On the other hand if bias was high and variance low, then a new image was estimated using a lower bandwidth. This proceeds until all pixels are accepted. The method depends on many user defined parameters. Problematically, one of these is the initial bandwidth; which, if chosen too low induces bias, or if chosen too high, raises the estimation time prohibitively (for each iteration the bandwidth is lowered one quarter of the former bandwidth).

In a method, proposed by Collins [1994], the relation between bundles of photons are tracked as they are traced from the light source. At photon hit positions this relation is used to decide the bandwidth of a Gaussian kernel placed over each photon in the bundle. Collins' method is geometry dependent as photon energy is deposited on illumination maps attached to scene objects.

A method with similar trends, has more recently been suggested by Herzog *et al.* [2007b]. In their method an individually kernel bandwidth for each particle is decided by the path traveled by the particle, as the bandwidth size is inversely proportional to probability of the particle path. Their method is an extension of *ray maps* [Bala and Dutre 2005]; a method that compensated for topological and boundary bias by finding the nearest photons based on their path through space.

In [2007] Schjøth in *et al.* also suggested a method based on photon splatting; they called it *photon differentials*. Unlike other splatting methods, which all used isotropic kernels during illumination reconstruction, kernels used by their method was anisotropic.

While Schjøth *et al.* [2006; 2007; 2008] also used an anisotropic kernel, the kernels in that method affect a number of photons based on an average of the local structure. Photon differentials uses an individually shaped kernel for each photon. This means that each local estimate is based on the correlation of a number of fixed kernels; each of which is shaped according to the structure of the illumination. In effect, photon differentials is more accurate and handles corners better.

3.4 Summary

This chapter has explained the role of density estimation in computer graphics. In the first part, it was demonstrated that classical density estimation pass on the trade-off problem between bias and variance described in Chapter 2. This led to a categorization of the bias defects common to many global illumination methods based on density estimation. In the last part of this chapter, a number of methods attempting to reduce the trade-off problem was summarized. Among these, two methods developed by the author was briefly described. Namely, diffusion based photon mapping and photon differentials. The following chapters will describe these method in-depth.

Part II

Anisotropic Density Estimation

4

Diffusion based Photon Mapping

Particle tracing algorithms are frequently used in photo realistic image synthesis. They usually employ two passes - a first pass in which particles representing light are emitted from light sources and reflected around a scene, and a second pass which generates an image of the scene using the light transport information from the first pass. The advantage of particle tracing algorithms is that they effectively simulate all possible light paths. In particular they can simulate lighting phenomena such as color bleeding and caustics.

However, particle tracing algorithms are faced with a severe problem. In the particle tracing pass, particles are stochastically emitted from the light sources and furthermore often stochastically traced through possible light paths. This procedure induces noise. Some particle tracing methods use density estimation to eliminate this noise during illumination reconstruction. Unfortunately, the noise reduction imposes a systematic error (bias) seen as a blurring of sharp illumination features. This is not necessarily a bad effect when concerned with slowly spatially changing illumination, but it becomes an important problem when the illumination intensity changes quickly such as when concerned with caustics and shadows. This problem is describe more thoroughly in Chapter 2.

The method presented in this chapter was first published in [Schjøth et al. 2006]. In later publications [Schjøth et al. 2007; Schjøth et al. 2008] the method has been improved and investigated more thoroughly.

The advantage of the presented algorithm is that it reduces noise and in addition preserve strong illumination features such as those seen in caustics. It has been implemented in photon mapping. Photon mapping is a popular particle tracing algorithm developed by Henrik Wann Jensen [1996].

Our algorithm is based on a filtering method called *nonlinear anisotropic diffusion.* Nonlinear anisotropic diffusion is a popular method commonly used in image processing, it has the property of smoothing along edges in an image instead of across edges [Weickert 1998]. Thus it preserves structures in an images while smoothing out noise.

We use diffusion filtering on densities of photons during the illumination reconstruction. As such our method is a numerical solution to the diffusion equation geared toward photon tracing. The effect is that our method uses a shape adapting anisotropic kernel.

As described in Section 3.3, the methods by Jensen, Myszkowsky, Schregle, and Walter all employ forms of the local-bandwidth kernel density estimator to reconstruct illumination and they all use isotropic kernels in the density estimate. In the context of diffusion filtering these methods are comparable to isotropic diffusion. The difference between the methods mainly lie in their method of detecting structure and the degree of change in bandwidth a given proximity of such structure entails.

In contrast our method employs an anisotropic kernel that allows it to smooth along edges and structures. Therefore, in the proximity of edges a radiance estimate performed by our method will have more support than other methods.

McCool [1999] use anisotropic diffusion filtering to reduce noise and preserve structure in Monte Carlo rendering. He processes the rendering in image space using a coherence map. The coherence map consist partly of world space information about object orientation and depth and partly of image space information about color contrast. His filtering of illumination is based on structure derived solely from color contrast.

Our algorithm differs from McCool's as we use anisotropic filtering only in the illumination reconstruction phase. We work with photon distributions in world space and not with pixel contrasts in image space. To our knowledge anisotropic diffusion has not been employed in connection with the illumination reconstruction of particle tracing algorithms.

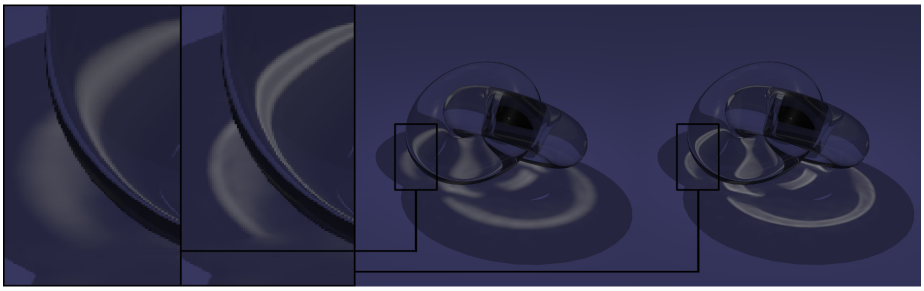


Figure 4.1: Rendering of caustics created by two interlinked toruses. Region zoom of image rendered using (a) regular photon mapping and (b) our method.

Figure 4.1 illustrates two renderings; one using regular photon mapping and the other using our method. The images show how our method reproduces caustics in higher detail than regular photon mapping using the same number of photons.

4.1 Diffusion filtering

In physics diffusion is the flow of matter caused by differences in concentration. Generally, we know that matter moves from areas with high concentration to areas with low concentration with the purpose of reaching equilibrium. This flow is described by Fick's first law

$$\mathbf{J} = -\mathbf{D}\nabla\mathbf{C}, \tag{4.1}$$

which states that the flux, $\mathbf{J}$, is dependent on the diffusivity $\mathbf{D}$, and the concentration gradient $\nabla\mathbf{C}$, where $C : \mathbb{R}^d \times \mathbb{R}_+ \rightarrow \mathbb{R}$, $\mathbf{D} \in \mathbb{R}^{d \times d}$, and d is the dimensionality. Here the concentration gradient is the change in concentration over the volume: $\nabla\mathbf{C} = \mathbf{i}\frac{\partial C}{\partial x} + \mathbf{j}\frac{\partial C}{\partial y} + \mathbf{k}\frac{\partial C}{\partial z}$, where $\mathbf{i}$,$\mathbf{j}$ and $\mathbf{k}$ are orthogonal unit vectors. The diffusivity is the controlling factor, it is a coefficient of the matter which diffuses. The negative sign of the diffusivity indicates that the flux is moving down the concentration gradient as one would assume. Further described by Crank [2004]

Combining Fick's law with the *continuity equation*, yields the diffusion equation

$$\frac{\partial\mathbf{C}}{\partial t} = \mathrm{div}(\mathbf{D}\nabla\mathbf{C}). \tag{4.2}$$

If the continuity equation[1], $\partial\rho/\partial t = -\mathrm{div}(\mathbf{J})$, is used to express the physical principal of mass conservation it states that if matter is moving out of a differential volume then the amount of matter within the volume is decreasing. The negative sign is canceled out by the negative sign in Fick's law when Fick's law is inserted into the continuity equation. In the equation the divergence operator is defined as $\mathrm{div}(\mathbf{V}) = \frac{\partial V_x}{\partial x} + \frac{\partial V_y}{\partial y} + \frac{\partial V_z}{\partial z} + ...$, where $\mathbf{V}$ is an arbitrary vector. Another notation for divergence is $\nabla\cdot$, but the former is preferred as the latter is more common in image processing.

The diffusion equation states that matter is preserved. That given a movement of matter in time caused by a flux, a change in concentration occurs within a differential volume. This flux we know to be dependent on the diffusivity and the concentration gradient.

4.1.1 Diffusion in image processing

In image processing diffusion is used as a means of smoothing images. Intuitively, diffusion can be understood as an erosion of an image. If one think of an image as a landscape where the topology is described by the pixel intensities, then diffusing the image causes peaks and valleys to level out and small cracks and fissures to vanish.

In image processing the diffusion equation becomes

$$\partial_t\mathbf{I} = \mathrm{div}(\mathbf{D}\nabla\mathbf{I}), \tag{4.3}$$

here $\partial_t\mathbf{I}$ is the shorthand notation for $\frac{\partial\mathbf{I}}{\partial t}$ where $\mathbf{I}$ is the image. t should be understood as the degree of smoothing/diffusion of the image. This will be explained in context of scale-space later.

[1] The continuity equation is a fundamental equation in physics. It is used in electromagnetic theory as well as quantum mechanics and fluid dynamics. In electromagnetic theory ρ represents charge, in quantum mechanics it is the probability density and fluid dynamics it is density of fluid.

Nonlinear isotropic diffusion

Nonlinear diffusion filtering was introduced to scale-space by Perona and Malik [1990]. The purpose was to contrive an alternative to Gaussian scale-space. Gaussian scale-space is a convolution of an image with a Gaussian kernel over time, where time is defined as the size of the kernel. What this basically means is that time is synonymous with the degree of image smoothing. Scale-space is a multi-scale representation of an image. In the context of diffusion, convolution with a Gaussian is the same as linear isotropic diffusion, that is the diffusivity is a constant, the image is diffused equally in all directions. Lindeberg gives a good introduction to scale-space in *"Scale-space: A framework for handling image structures at multiple scales"* [Lindeberg 1996].

The problem with Gaussian scale-space is that ridges and edges moves spatially when moving through time, making tracking of corresponding structures cumbersome. Perona and Malik's solution to the problem was what they called anisotropic diffusion. A filtering process which preserved edges while smoothing intra-regions. They introduced the diffusion equation and described the diffusivity by a scalar function. This function controlled the degree of diffusion in the image locally and it was suggested that it should depend on the gradient of the actual image (scale-space smoothed), thus making the process nonlinear. Perona-Malik's function for the diffusivity is given by the equation:

$$g(\nabla\mathbf{I}) = \frac{C}{1+\left(\frac{|\nabla\mathbf{I}|}{K}\right)^{1+\alpha}}, \quad \alpha > 0 \tag{4.4}$$

C is a constant suggested to be 1 by Perona and Malik [1990] and α, also a constant, should be greater than 0. Together with α, the constant K decides the threshold for the diffusion. The function is drawn in Figure 4.2, with $\alpha = 3$, $C = 1$, and $K = 0.3$. Now K decides when the function starts to monotonically decrease and α the steepness of the decline. In practice what it means is, that K is the threshold deciding what is considered an edge and what is considered noise, and α controls the smoothness of transition. It is intuitive that high frequency noise, such as salt and peber noise, can cause trouble if considered an edge by the diffusivity function. Of course adjusting K and α is necessary, but for images where noise is pronounced and edges are weak, the problem remains. To circumvent this, Perona and Malik suggested that the parameters should be adjusted dynamically using local estimates of noise and contrast. However, this method can still cause edge information to be lost.

For this reason and other[2] Catté *et al.* [1992] proposed pre-smoothing the gradient, using this regularized version, as a basis for Perona-Malik's diffusivity function. The advantage of using this regularized gradient is that noise and insignificant details will be ignored when finding the gradient. The solution was found to be stable and theoretically more sound than Perona-Malik's.

Returning to Perona-Malik's diffusitivity function, Equation 4.4, it is important to note that the image gradient, $\nabla\mathbf{I}$, is time dependent, this makes Perona-Malik's diffusion filter nonlinear.

[2] Another problem was that the method was not stable, meaning that images very similar could produce different solutions [Catté et al. 1992; Weickert 1998].

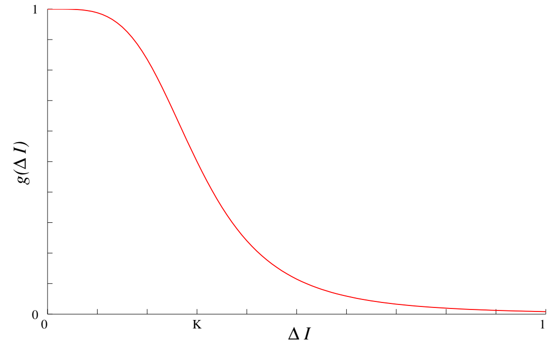


Figure 4.2: Parona-Malik's diffusivity function, $g(\nabla\mathbf{I}) = C/(1 + (|\nabla\mathbf{I}|/K)^{1+\alpha})$, with the parameters: $\alpha = 3$, $C = 1$, and $K = 0.3$

Nonlinear anisotropic diffusion

Another approach to diffusion filtering has been well studied by Weickert [1998]. Instead of using a scalar function to represent the diffusivity, the idea is to use a symmetric positive semidefinite matrix called the *diffusion tensor*, $\mathbf{D} \in \mathbb{R}^{d\times d}$ were d is the dimensionality of the image. The diffusion tensor should adapt locally as to describe the structure of the image. Depending on how the tensor is constructed, this makes the diffusion filtering capable of either edge- or coherence-enhancing smoothing.

The diffusion tensor depends on the image gradient. As with the regularized version of Perona-Malik's diffusion equation the image gradient is pre-smoothed. A notation for this is $\nabla\mathbf{I}_\sigma$ which means $\nabla(\mathbf{G}_\sigma * \mathbf{I})$, that is the image, $\mathbf{I}$, is convolved with a kernel, $\mathbf{G}_\sigma$, where sigma is the standard deviation of the kernel. From this we get the following formulation of Equation 4.3

$$\partial_t\mathbf{I} = \mathrm{div}(\mathbf{D}(\mathbf{I}_\sigma)\nabla\mathbf{I}). \qquad \boxed{4.5}$$

In Weickert's terminology this form of diffusion filtering is called anisotropic, while diffusion with a scalar diffusivity function, as with Perona-Malik's diffusion, is called isotropic. The difference being that the diffusion is parallel to the structure gradient in the isotropic case, as opposed to the anisotropic case where the diffusion can be parallel to the structure, perpendicular to the gradient.

An advantage of anisotropic diffusion is that it, unlike isotropic diffusion, is capable of smoothing along edges and structures thereby avoiding the problem of isotropic diffusion where noise along edges is not removed. Another advantage is that anisotropic diffusion can connect discontinuities and gabs in image structure. This can be used in image restoration.

The diffusion tensor can be constructed in several ways. One of the earlier methods is to use the regularized gradient and its orthogonal as eigenvector for the diffusion tensor. If the eigenvector parallel to the regularized gradient is $v_1 \parallel \nabla \mathbf{I}_\sigma$, and its orthogonal, perpendicular to the regularized gradient is $v_2 \perp \nabla \mathbf{I}_\sigma$, then their corresponding eigenvalues can be given by

$$\begin{aligned} \lambda_1 &= g(\nabla \mathbf{I}_\sigma), \\ \lambda_2 &= 1, \end{aligned} \tag{4.6}$$

here g is the diffusivity function as the one given by Perona-Malik in Equation 4.4. This way of construction will cause the diffusion to enhance edges and structures in the image, in much the same way as the regularized version of Perona-Malik's diffusion filtering, the difference being that smoothing will occur along edges and structures, hence it is anisotropic in the terminology of Weickert.

However, using the regularized gradient as a descriptor for image structure, as seen above, is not optimal. One disadvantage is that it cannot distinguish between corners and edges. Another, that even though it detects edges quite well the overall orientation of structure is not always distinguished. For a more throughly argumentation of why the regularized gradient as a structure descriptor is ill-posed see Weickert [1998; 1999].

The structure tensor is a more advanced structure descriptor introduced to diffusion filtering by Weickert in *"Multiscale Texture Enhancement"* [1995]. The advantage of structure tensor is that even though it does not contain more information than the gradient descriptor, it is, unlike the gradient, possible to smooth it without losing important structure information. Being able to smooth the structure descriptor makes the orientation information less perceptible to noise, while taking the structure descriptor into scale-space.

In this thesis the theoretical well-posedness of the structure tensor as used in diffusion filtering will not be discussed. A throughly account of this has been done by Weickert in his book [1998] another more contemporary examination has been made by Brox *et al.* in [2004]. The structure tensor is the tensor product of the gradient, in this case we will use the regularized gradient giving us

$$\mathbf{S}(\nabla \mathbf{I}_\sigma) = \nabla \mathbf{I}_\sigma \otimes \nabla \mathbf{I}_\sigma. \tag{4.7}$$

The result is a symmetric semidefinite matrix with orthonormal eigenvectors describing the image structure. This matrix can now be smoothed componentwise with a suitable kernel, this gives

$$\mathbf{S}_\rho(\nabla \mathbf{I}_\sigma) = \mathbf{G}_\rho * (\nabla \mathbf{I}_\sigma \otimes \nabla \mathbf{I}_\sigma), \tag{4.8}$$

where $\mathbf{G}_\rho$ is the kernel and ρ is the kernels standard deviation also called the *integration scale*. Let the eigenvectors of the structure tensor be denoted $\mathbf{v}_1$ and $\mathbf{v}_2$ and its corresponding eigenvalues λ_1 and λ_2 where $\lambda_1 \geq \lambda_2$. what this means is that $\mathbf{v}_1$ is the normal to the structure while $\mathbf{v}_2$ is parallel to the structure.

Now given the structure tensor it is possible to construct the diffusion tensor $\mathbf{D}(\mathbf{S}_\rho(\nabla \mathbf{I}_\sigma))$. Depending on whether the diffusion should be coherence- or edge-enhancing, this can be done in different ways. Common to these is that diffusion tensors should be constructed with the same eigenvectors as the structure tensor. The difference is that the eigenvalues of the diffusion tensor should be transformed to either enhance edges or coherence. This is in many ways similar to constructing the diffusion tensor using the regularized gradient as structure descriptor but instead of using the gradient, we here use the eigenvalues of the structure tensor to construct the diffusion tensors eigenvalues.

For edge-enhancement the eigenvalues can be constructed as in Equation 4.6 the only difference being that diffusivity function should depend λ_1 so that the eigenvalues of the diffusion tensor is given by $\mu_1 = g(\lambda_1)$ and $\mu_2 = 0$.

For coherence-enhancement Weickert proposes in [1998] to construct the eigenvalues in the following way

$$\begin{aligned} \mu_1 &= \alpha, \\ \mu_2 &= \begin{cases} \alpha & \text{if } \mu_1 = \mu_2, \\ \alpha + (1-\alpha)\exp\left(\frac{-C}{(\mu_1-\mu_2)^{2m}}\right) & \text{else,} \end{cases} \end{aligned} \tag{4.9}$$

here α should be a small positive constant. It is deciding the amount of diffusion over structures and the minimum amount in isotropic areas. C is the threshold for anisotropy and m is the steepness of the transition from isotropy to anisotropy. The measure for anisotropy is difference of the eigenvalues, $(\lambda_1 - \lambda_2)$, of the structure tensor.

Having transformed the eigenvectors either using the edge-enhancing scheme from Equation 4.6 or the coherence-enhancing scheme in Equation 4.9 it is now possible to construct the diffusion tensor. As mentioned the eigenvectors to be used should be the eigenvectors of the structure tensor.

Constructing the diffusion tensor using the structure tensor we get the following formulation of Equation 4.3

$$\partial_t \mathbf{I} = \operatorname{div}(\mathbf{D}(\mathbf{S}_\rho(\nabla \mathbf{I}_\sigma))\nabla \mathbf{I}). \tag{4.10}$$

This is Weickert's nonlinear anisotropic diffusion equation.

In image processing, nonlinear diffusion is a well-studied concept. Since it was introduced by Perona-Malik in [1990], numerous papers has been written on the subject. It is generally known to been theoretically well-founded and, moreover, results has shown to be visually impressive for both the isotropic and anisotropic method. Both methods has played a huge part in the field of scale-space, which is partly the reason for their success.

4.1.2 Summary

In this section diffusion has been introduced. First diffusion has been introduced as a classical theory in physics and afterwards an account of how this theory has been adapted in image processing has been presented. The intention of this chapter has been to prepare the reader to the following. In particular Weickert's nonlinear anisotropic diffusion equation is important as it will be the base of some of the filtering strategies which will be presented in this thesis.

It is important to note that the diffusion equation, as it is presented here, is continuous. In practice, discretization of the diffusion equation is necessary in order to do numerical computations. For this reason numerous numerical methods has been developed, both explicit, semi-explicit, and combinations of these [Mrázek 2001; Weickert 1998]. When implementing nonlinear diffusion filtering, these methods should be weighted both in respect to stability and efficiency.

In this thesis we will not give an account of these different numerical methods. The reason being that these all have been developed to work on images. In this thesis we will present a discretization of the diffusion equation which works with densities instead of pixels.

4.2 Diffusion filtering in photon mapping

To be able to use anisotropic diffusion filtering in photon mapping, we have to be able to describe the structure of the photon map, to get some guidance as how to adapt the filtering. Furthermore, we need to be able to adapt the kernel according to the structure description, and it is necessary to normalize the adapted kernel estimate to preserve energy, when the kernel changes shape.

4.2.1 Structure description

The gradient of the illumination function denotes the orientation in which the illumination has maximal intensity change, and therefore describes the first order structure of the illumination. This information will be used to steer the filtering.

As the illumination function is estimated in the radiance estimate, the differentiated radiance estimate approximates the gradient of the photon map.

To differentiate the radiance estimate we combine the generalized radiance estimate from Equation 3.2, with a suitable kernel function. Furthermore, it is convenient to simplify the radiance estimate by assuming that all surfaces hit by photons are ideal diffuse reflectors. This means that the BRDF, f_r, is constant regardless of the incoming and outgoing direction of light. In this way the BRDF need not be differentiated, as it does not depend on the position, $\mathbf{x}$, which is the variable in respect to which we differentiate.

This of course is a radical assumption, as photons can be affected much by the type of surfaces they encounter. However, photons are only stored on diffuse or glossy surfaces, so the surfaces involved in the radiance estimate are likely to contain a diffuse element and need therefore not differ much from an ideal diffuse surface. Using a constant BRDF to estimate the structure descriptor only affects the diffusion. In effect we will have equal diffusion regardless of the incoming directions of the photons. However, the radiance will still be estimated using the true BRDF.

If we were to differentiate the BRDF, then our algorithm would not be able to handle arbitrary BRDF's, as we would have to know the BRDF in order to do so. In effect we would not retain the beneficial qualities of photon mapping. Another solution would be to perform reverse engineering, to numerically estimate the BRDF in question. However, this approach is both cumbersome and computationally expensive.

Additionally, we have to make a constraint on the generalized radiance estimate. The estimate should use a fixed bandwidth for $h(\mathbf{x})$, such that the bandwidth is independent of $\mathbf{x}$, effectively reducing the radiance estimate to a common multivariate kernel estimator rather than the k'th nearest neighbor estimator. This is not a severe constraint. The advantage of the k'th nearest neighbor search is its ability to reduce bias. This ability is important in the radiance estimate, however, when estimating the gradient smoothing is an advantage, as the gradient is perceptible to noise.

Combining a simplified version of the generalized radiance estimate with the two-dimensional Epanechnikov kernel we get

$$\widehat{L}_r(\mathbf{x}, \vec{\omega}) = \frac{2f_r}{\pi h^2} \sum_{i=1}^{n} \left(1 - \frac{(\mathbf{x} - \mathbf{x}_i)^T (\mathbf{x} - \mathbf{x}_i)}{h^2} \right) \Phi_i, \qquad \boxed{4.11}$$

This equation can be differentiated giving us the gradient function of the estimated illumination function. Differentiating Equation 4.11 with respect to the j'th component of $\mathbf{x}$ gives the partial derivative

$$\frac{\partial \widehat{L}_r(\mathbf{x}, \vec{\omega})}{\partial x_j} = \frac{4f_r}{\pi h^2} \sum_{i=1}^{n} -\frac{x_j - x_{ij}}{h^2} \Phi_i. \qquad \boxed{4.12}$$

As seen from Figure 4.3, the gradient of the photon map is a plausible structure descriptor. Figure 4.3a is a distribution of photons and Figure 4.3b is a gradient field of the distribution. The gradient vectors are calculated using the photons nearest the center of each quadrant in the grid of the field. The gradient vectors along the edges of the distribution are those with greatest magnitude, and the vectors are as expected perpendicular to edges and structures.

As explained in Section 4.1.1, a more advanced way is to describe the first order structure is with the structure tensor. The advantage of the structure tensor is that even though it does not contain more information than the gradient descriptor, it is, unlike the gradient, possible to smooth it without losing important structure information. Being able to smooth the structure descriptor makes the orientation information less perceptible to noise.

The structure tensor is the tensor product of the gradient. If we denote the gradient of the photon map as $\nabla\mathbf{P}$, where $\mathbf{P}$ is the photon map, then in three dimensions the structure tensor is given by

$$\mathbf{S} = \nabla\mathbf{P} \otimes \nabla\mathbf{P} = \begin{pmatrix} P_x^2 & P_xP_y & P_xP_z \\ P_xP_y & P_y^2 & P_yP_z \\ P_xP_z & P_yP_z & P_z^2 \end{pmatrix}. \qquad \boxed{4.13}$$

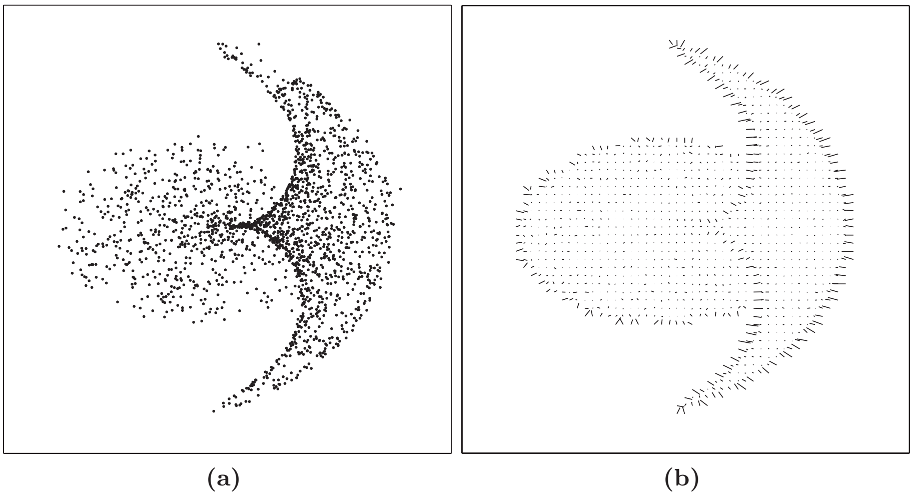


Figure 4.3: (a) Cardioid shaped photon distribution created by light reflection within a metal ring, (b) gradient field of the photon distribution in (a).

In our method we use the structure tensor to describe the structure of the photon map.

To improve performance we can reduce the dimensionality of the problem. This can be done in the density estimate by projecting local photons and gradient vectors onto the tangent plane to the surface at the estimation point. If the local photons and gradient vectors then are transformed into the two dimensional tangent space spanned by the tangent and the bi-normal then all further estimations can be performed in two dimensions. In the Appendix A it is described how to project photons into the tangent plane while reducing the bias associated with high surface curvature.

After projection the structure tensor can be formulated as a 2x2 matrix. Performance wise this is important as the eigenvalues and eigenvectors of the structure tensor are needed to steer the diffusion. The following estimates are performed in two dimensions. Note, however, that they could just as well have been done in three dimensions.

4.2.2 Diffusion tensor

As explained the diffusion tensor can be constructed to either promote edge-enhancing diffusion or coherence-enhancing diffusion. To preserve the finer structures of the illumination during reconstruction we employ edge-enhancing diffusion. This is achieved by constructing the diffusion tensor using information derived from a structure descriptor. Specifically, we can use the eigenvectors and eigenvalues of the structure tensor to construct the diffusion tensor.

The orientation of the local edge structure is contained in the structure tensor. The primary eigenvector, $\mathbf{v}_1$, of the structure tensor is simply the gradient, which is perpendicular to the local edge orientation. The secondary eigenvector, $\mathbf{v}_2$, points in the direction parallel to the structure. The corresponding eigenvalues, λ_1 and λ_2, gives the degree of change in the directions of the eigenvectors.

To achieve edge-enhancing diffusion the diffusion tensor should be constructed such that smoothing occurs parallel to the edges and not across them. The eigenvectors and eigenvalues of the diffusion tensor describe respectively the main directions of diffusion and the amount of diffusion in the corresponding direction. Hence, by constructing the diffusion tensor from the eigenvectors of the structure tensor, diffusion can be steered to enhance the edges.

In our method the diffusion tensor is constructed as

$$\mathbf{D} = \mathbf{M}\ \mathrm{diag}(\mu_1, \mu_2)\ \mathbf{M}^T, \tag{4.14}$$

where $\mathbf{M}$ is $[\mathbf{v}_2\mathbf{v}_1]$ and diag$(\cdot)$ is the diagonal matrix containing the eigenvalues of $\mathbf{D}$ along the diagonal.

It remains to determine the amount of diffusion. That is the eigenvalues, μ of $\mathbf{D}$:

$$\begin{aligned} \mu_1 &= 1, \\ \mu_2 &= \frac{1}{1+\left(\frac{\lambda_1}{q}\right)^{1+\alpha}}, \qquad \alpha > 0, \end{aligned} \tag{4.15}$$

where the secondary eigenvalue, μ_2, is estimated using a function diffusivity function, Equation 4.4. The diffusivity coefficient, q, decides when the function starts to monotonically decrease and α the steepness of the decline. In practice, q is the threshold deciding what value of the primary eigenvalue of the structure tensor, λ_1, is considered an edge and what is considered noise, and α controls the smoothness of transition. The primary eigenvalue of the structure tensor should be normalized such that its range over the photon map is from zero to one. Section 4.3 demonstrates how different values of k affect the diffusion.

We have now constructed a diffusion tensor which favors diffusion parallel to structures while limiting diffusion perpendicular to structures. We will utilize this tensor, such that it controls the filtering of the photon map.

4.2.3 The diffusion based radiance estimate

The next step is to use the diffusion tensor to shape the kernel of the radiance estimate such that it smooths along structures and edges. To do this we have to shape our kernel in some way.

The kernel density estimator is isotropic insofar the single parameter, h, controls the filtering. As such smoothing occurs equally in all directions.

Consider a simple two dimensional normal distribution:

$$f(\mathbf{x}) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left(-\frac{(x_1-\mu_1)^2}{\sqrt{2}\sigma_1} - \frac{(x_2-\mu_2)^2}{\sqrt{2}\sigma_2} \right), \tag{4.16}$$

where σ_1 and σ_2 are the standard deviations with respect to the axes and μ is the center of the distribution. Here we have a Gaussian kernel whose shape is specified by the two parameters for the standard deviation. Unfortunately, this equation only gives control in two directions.

However, generalizing the equation to d dimensions, we can use an inversed $d \times d$ covariance matrix, Σ^{-1}, to shape the normal distribution:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\sqrt{\det \Sigma}} \exp\left(-\frac{(\mathbf{x}-\mu)^T \Sigma^{-1}(\mathbf{x}-\mu)}{2}\right). \qquad \boxed{4.17}$$

Using a matrix we are not limited to control the shape of the Gaussian kernel in only two directions. If we for example had shaped our Gaussian kernel to form an ellipse, we could rotate this kernel by rotating the covariance matrix. The equation will remain normalized as the determinant of a matrix is rotational invariant. So the shape of normal distribution in Equation 4.17 is controlled by the covariance matrix.

We can use Equation 4.17 to extend the generalized radiance estimate from Equation 3.2. To generalize the shape adapting properties we use the Mahalanobis distance from Equation 4.17 to shape the kernel. The Mahalanobis distance is a statistical distance. It is given by:

$$\big(d(\mathbf{x},\mathbf{y})\big)^2 = (\mathbf{x}-\mathbf{y})^T \Sigma^{-1}(\mathbf{x}-\mathbf{y}). \qquad \boxed{4.18}$$

As the shape of the kernel should be controlled by the diffusion tensor, we use the tensor in place of the covariance matrix. We can then reformulate the generalized radiance estimate as:

$$\begin{aligned}\widehat{L}_r(\mathbf{x},\vec{\omega}) = & \frac{1}{h^2\sqrt{\det \mathbf{D}}} \sum_{i=1}^{n} K\left(\frac{(\mathbf{x}-\mathbf{x}_i)^T \mathbf{D}^{-1}(\mathbf{x}-\mathbf{x}_i)}{h^2}\right) \\ & \cdot f_r(\mathbf{x},\vec{\omega}_i,\vec{\omega})\Phi_i.\end{aligned} \qquad \boxed{4.19}$$

We now have a general diffusion based radiance estimate, which filters the photon map adapting the shape of the kernel according to the diffusion tensor. Or to be even more general, we have a radiance estimator, which estimates the illumination function taking into consideration the structure of the photon map, such that edges and structures are preserved.

4.2.4 Implementation

Implementation of our method can differ depending on which structure descriptor is used. However, we propose to use the structure tensor, and for this reason, we need to estimate it, or have it available during the radiance estimate in order to construct the diffusion tensor.

We do this using a preprocessing step that approximates the gradient of the photon map. The preprocessing step occurs between the photon tracing pass and the rendering pass. To approximate the gradient we sample it at all photon positions. The advantage of this procedure is that we can store the local gradient along with the photon, and thus we do not need a separate gradient map. Additionally, we know the sampling positions to be located on a surface, as photons are only stored in connection with a surface. This is useful as the gradient is only relevant at surface positions.

During the radiance estimate we calculate the structure tensors at the photon positions near $\mathbf{x}$. In this way we can estimate the local structure tensor as the weighted average of the surrounding structure tensors. Smoothing the structure tensor reduces noise, and furthermore gives a broader foundation from which to steer the filtering after.

Weickert operates with two important smoothing parameters, namely integration scale and noise scale. These refer to the smoothing parameter of the gradient and the structure tensor respectively. In relation to our method the noise scale is the bandwidth used to estimate the gradient in the preprocessing step and the integration scale is the bandwidth used to smooth the structure tensor. Weickert suggests that the best results are attained if the noise scale is lower than the integration scale. Through experimentation we have found this to be true for diffusion based photon mapping too, although the the amount that they differ seems to have little importance.

Having calculated the local structure tensor, we construct the diffusion tensor as described in the former section. This then is used in the general diffusion based radiance estimate together with a suitable kernel. We use the same bandwidth for smoothing the structure tensor as for the general diffusion based radiance estimate.

4.3 Evaluation and comparison

To evaluate, we compare our proposed method to regular k'th nearest neighbor photon mapping. Furthermore, we demonstrate that our method avoids a specific drawback common to many existing bias reducing algorithms, and finally we evaluate the computational performance of our algorithm using three test scenes of varying complexity. For consistency the Epanechnikov kernel is used in all estimations.

To facilitate the evaluation of our method, we have constructed a synthetic photon distribution. The constructed distribution is seen in Figure 4.4. It is rather simple, yet it contains both edges and ridges and circular and rectangular shapes. The distribution consists of 48 000 photons.

4.3.1 The k'th nearest neighbor photon mapping

The first row of Figure 4.5 is a test suite of the constructed distribution created using regular k'th nearest neighbor photon mapping. The suite is created by

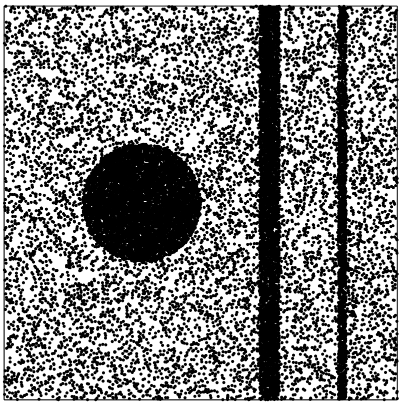


Figure 4.4: A synthetically constructed photon distribution consisting of 48 000 photons (fewer photons are shown in the figure for the purpose of visualization).

combining the Epanechnikov kernel with the general radiance estimate from Equation 3.2.

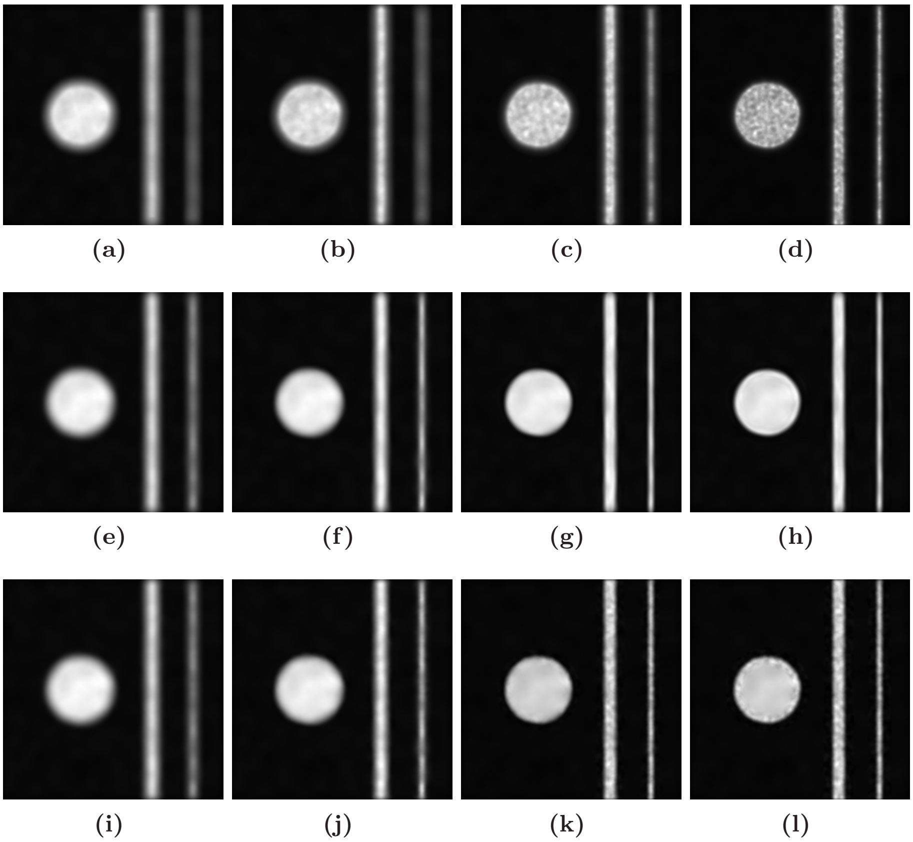
It is seen from the suite that the noise level decreases slowly with respect to the number of photons per estimate. Bias is visible as a clearly identifiable blurring of shape edges. In addition, boundary bias is seen along the boundaries of the images. It should be clear that the bias increases as the noise is reduced. This phenomenon is directly related to the bias versus variance trade-off accounted for earlier. Another thing to notice is how the thin line losses intensity as the number of photons per estimate is increased. This happens because the energy of the line is spread out over a larger area as the smoothing increases.

4.3.2 Anisotropic diffusion based photon mapping

We use the diffusion based radiance estimate together with the Epanechnikov kernel. In contrast to regular photon mapping we do not use the k nearest neighbor method to reduce bias, instead we use a fixed bandwidth letting the shape adaption reduce bias.

We first find a suitable bandwidth. This is done using a large value of the diffusivity coefficient q from Equation 4.15. In this way the kernel will remain rotational invariant, as it will not adapt according to structure. Estimating the radiance with a uniform Epanechnikov kernel using different bandwidths, we then find the bandwidth which reduces noise to an acceptable level.

Using this bandwidth we test the diffusion based radiance estimate by iteratively decreasing the value of the diffusivity coefficient such that the kernel starts to adapt its shape according to the structure described by the structure tensor. The result of this procedure is illustrated in the second row of Figure 4.5.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

Figure 4.5: Visualizations of a constructed distribution estimated using different photon mapping methods. The first row was estimated with **k'th nearest neighbor photon mapping** using the (a) 1200, (b) 600, (c) 300, and (d) 150 nearest photons. The second row was estimated with **anisotropic diffusion based photon mapping** using a diffusivity coefficient, q, of (e) 0.8, (f) 0.4, (g) 0.2, and (h) =0.1. The third row was estimated with **isotropic diffusion based photon mapping** using a diffusivity coefficient, q, of (i) 0.8, (j) 0.4, (k) 0.2, and (l) 0.1.

From the results of the diffusion based radiance estimate we see that edges are enhanced as the diffusivity coefficient is decreasing. Comparing the results to those of k'th nearest neighbor photon mapping visualized in the first row of Figure 4.5, we see that anisotropic diffusion based photon mapping has improved the trade-off between bias and variance significantly. For instance comparing Figure 4.5d with Figure 4.5h we see that even though the bias level is comparable low for the two images, noise is clearly visible in Figure 4.5d. Increasing the bandwidth for k'th nearest neighbor photon mapping as to make the noise level comparable to Figure 4.5h we would at least have to use 600 photons per estimate which in turn would increase bias well beyond the level seen Figure 4.5h.

Another thing to notice is the thinnest line in the constructed distribution. We know that this line has photon distribution as dense as the two other shapes in the distribution. For this reason the thin line should be just as intense as the other shapes. However, as estimates are smoothed using a larger bandwidth and thus more photons per estimate, the energy is spread out. Comparing the results it is seen that the anisotropic diffusion based radiance estimate is most successful in preserving the energy of the thin line, as it has almost the same intensity as the other shapes.

4.3.3 Isotropic diffusion based photon mapping

In order to compare our proposed method to existing bias reducing algorithms, we have developed a method mimicking these. This method we term *isotropic diffusion based photon mapping*. Similar to the existing algorithms, isotropic diffusion based photon mapping reduce the size of the bandwidth in the proximity of edges. However, it uses the structure tensor to detect structure and the diffusivity function to determine to what degree bandwidth should be decreased. As we have shown, the structure tensor is a plausible structure descriptor, we expect isotropic diffusion based photon mapping to exhibit the same bias reducing qualities as existing methods and therefore to be a reasonable proxy for these.

The approach for isotropic diffusion based photon mapping is quite similar to the one for anisotropic photon mapping. First we find the fixed bandwidth, which reduces the visible noise to an acceptable level. Then we iteratively reduce the diffusivity coefficient. However, in contrast to the anisotropic case, we construct the diffusion tensor such that the kernel shape remains rotational invariant. This is done using the result of the diffusivity function as the eigenvalue for both the primary and secondary eigenvector of the diffusion tensor.

The result of this approach is seen in third row of Figure 4.5. Comparing the results with those of regular photon mapping, first row, we see that isotropic diffusion based photon mapping has a superior bias vs. variance trade-off. However, the results also reveal that noise along structure edges becomes apparent as the diffusivity coefficient decreases. The reason for this is that as the diffusivity coefficient is decreasing so is the support area for the estimate. Hence, radiance estimates made near prominent structure is based on a reduces number of photons, causing noise to appear. This is a problem inherent to existing bias reducing techniques used in particle tracing algorithms.

Comparing the results in second row of Figure 4.5 with those in the third row it becomes apparent that anisotropic diffusion based photon mapping does not suffer the same degree adverse effects as its isotropic counterpart. Inspecting the structure edges of Figure 4.5h and Figure 4.5l it is seen that noise is much less pronounced in the anisotropic case. The reason for this is that anisotropic diffusion based photon mapping adapts the filtering according to structure and therefore does not limit the support area near prominent structure in the same degree as its isotropic counterpart.

We find it reasonable to conclude that anisotropic diffusion based photon mapping offers an improved trade-off between bias and variance as compared to existing bias reducing techniques used in particle tracing.

4.3.4 Performance

In order to test the performance of our proposed algorithm we have set up three different test scenes of varying complexity. The image sets in Figure 4.6, 4.7 and 4.8 are renderings of the three scenes in order of increasing complexity. Each set contains one image rendered using diffusion based photon mapping and three images rendered using regular k'th nearest neighbor photon mapping.

The **(d)** images were created with our method using a diffusivity coefficient of q=0.2 and with α equal to one. The **(a)**, **(b)** and **(c)** images were created with regular photon mapping using **(a)** the same number of photon as **(d)**, **(b)** the approximately same rendering time as **(d)**, and **(c)** with a visual quality comparable to **(d)**. For all images created with regular photon mapping the number of photons per estimate was regulated for an optimal trade-off between bias and variance.

Figure 4.6 is a low complexity scene containing a simple caustic created by a translucent cylinder. The contours of the caustics in the images 4.6a and 4.6b are clearly blurred as compared to their counterpart in image 4.6d. The same is evident for the image series in Figure 4.7 and Figure 4.8. This means that even with a substantial increase of photons in the photon map for regular photon mapping, our method still offers a better trade-off between bias and variance.

Table 1 lists the rendering times for the image series in Figure 4.6, 4.7 and 4.8. The rendering times for each scene are relative to a scene rendering using a low resolution photon map. Consulting Table 1 we see that for all three test scenes, our method is more than twice as fast as k'th nearest neighbor photon mapping, when comparing the rendering times of the results with similar visual quality. This is because five to six times as many photons are needed in regular photon mapping to achieve the same visual quality as with the proposed method. At equal rendering time it is evident from the image series that our method produce sharper edges than regular photon mapping.

Scene	Method	Figure	Photons in photon map	Total time
Cylinder	KNN	-	5,000	1.0
		4.6a	20,000	1.1
		4.6b	25,000	1.2
		4.6c	160,000	2.3
	Anisotropic filtering	4.6d	20,000	1.2
Metal ring	KNN	-	20,000	1.0
		4.7a	40,000	1.2
		4.7b	45,000	1.3
		4.7c	160,000	2.7
	Anisotropic filtering	4.7d	40,000	1.3
Water	KNN	-	50,000	1.0
		4.8a	70,000	1.1
		4.8b	120,000	1.7
		4.8c	520,000	3.7
	Anisotropic filtering	4.8d	70,000	1.7

Table 4.1: Performance results of the images in Figure 4.6, 4.7, and 4.8. The rendering times are relative to a low photon resolution rendering. For each scene the three first images have been rendered using k'th nearest neighbor photon mapping (KNN) while the last image has been rendered using diffusion based photon mapping.

Consulting Table 1 we see that for all three test scenes, regular photon mapping needs five to six times as many photons to achieve the same visual quality. This means that rendering times for regular photon mapping are more than twice those for our method for renderings with comparable visual quality. Furthermore, the difference in rendering times increases favorably to our method, concurrent with scene complexity.

4.4 Summary

In this chapter we have presented a method extending particle tracing. Our method is a numerical solution to the diffusion equation adapted to photon mapping. The methods enhance edges and structures of prominent illumination features by shape adapting the filter kernel according to the structure of the photon map. In contrast, existing bias reducing methods only adapts the size of the kernel.

We have evaluated our method using a simple constructed photon distribution and a number of test scenes. In the evaluation we demonstrate that our method achieves a superior trade-off between variance and bias as compared to regular photon mapping based on the k'th nearest neighbor method, with no substantial increase in computer time. Furthermore, we substantiate that our method alleviates an edge problem common to existing popular methods.

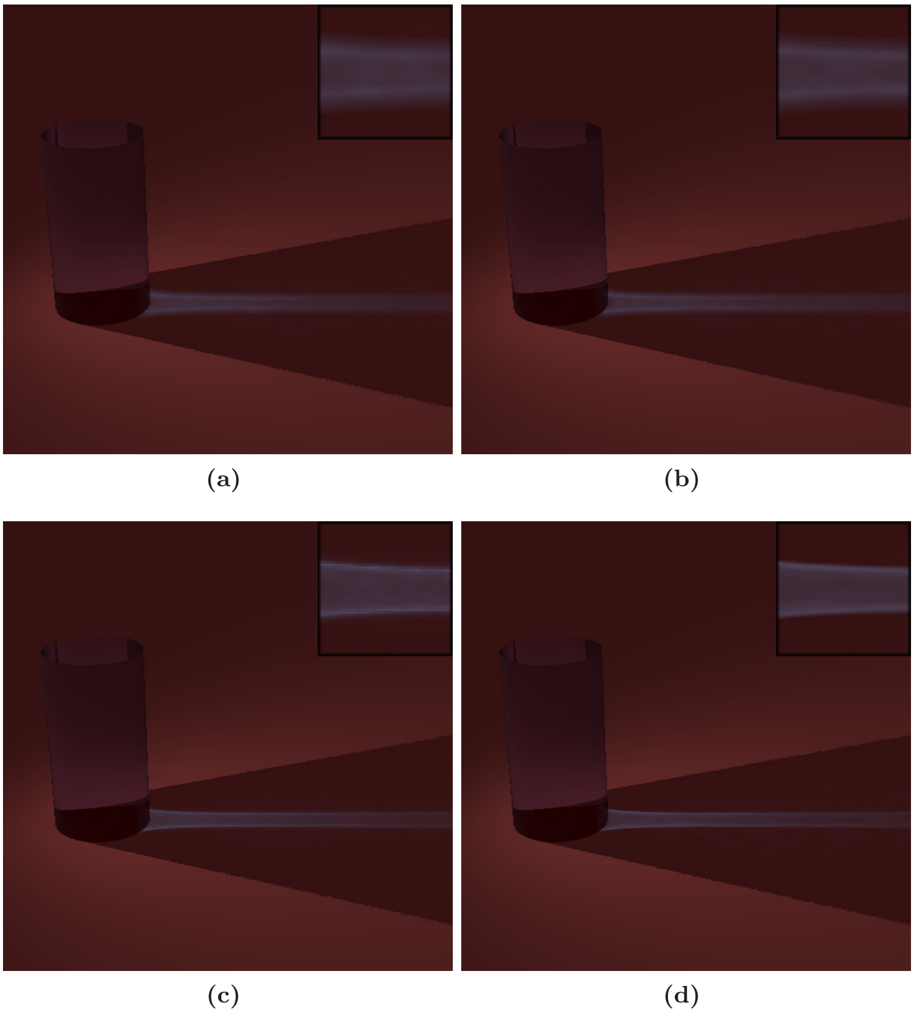
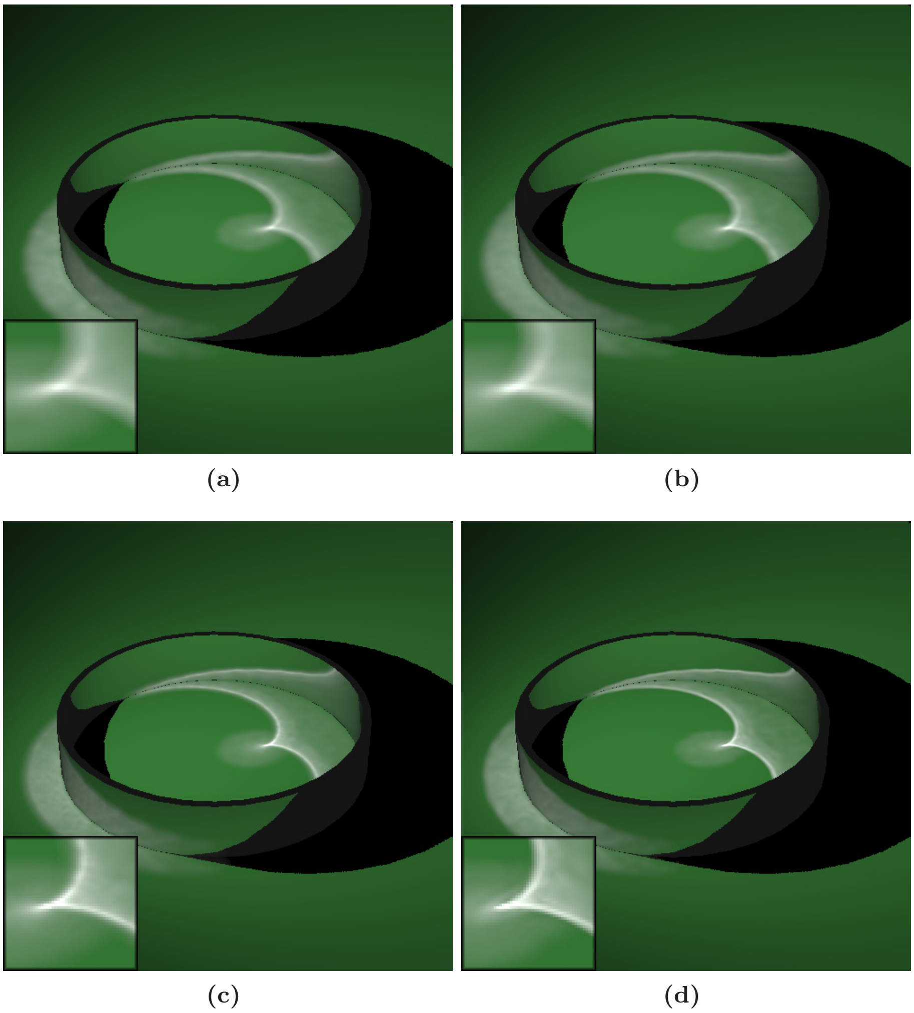


Figure 4.6: Caustic produced by a translucent cylinder posed on a plane. (d) rendered using our method. (a) rendered using regular photon mapping and the same amount of photons as (d), (b) using approximately the same rendering time as (d) and (c) with a visual quality comparable to (d).



(a) (b) (c) (d)

Figure 4.7: A cardioid shaped caustic created by a metal ring. (d) rendered using our method. (a) rendered using regular photon mapping and the same amount of photons as (d), (b) using approximately the same rendering time as (d) and (c) with a visual quality comparable to (d).

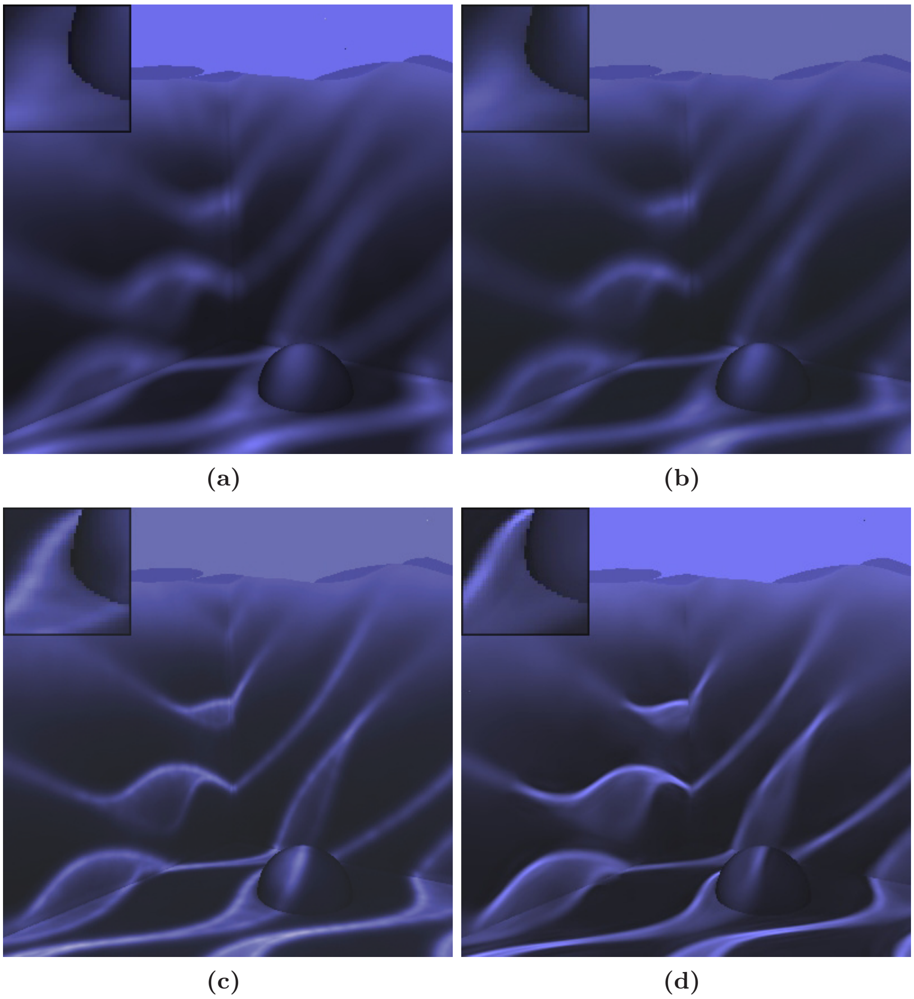


Figure 4.8: Caustics created from water waves. (d) rendered using our method. (a) rendered using regular photon mapping and the same amount of photons as (d), (b) using approximately the same rendering time as (d) and (c) with a visual quality comparable to (d).

5

Photon Differentials

Monte Carlo sampling is used in a number of global illumination algorithms. The stochastic nature of Monte Carlo sampling induces variance that may require a large number of samples to be reduced to an acceptable level. Particle tracing algorithms is a group of Monte Carlo based Global Illumination algorithms that employs density estimation in order to reduce this variance. The density estimate imposes a trade-off between variance and a bias [Silverman 1986]. Bias is noticeable as a blurring of the illumination. This is not necessarily a bad effect when concerned with low frequency illumination, but it becomes an important problem when the illumination intensity changes quickly such as near caustics and shadow-borders.

In this chapter we present a method that enhances edges and structures of prominent illumination features improving the trade-off between variance and bias. The method was published in [Schjøth et al. 2007] and extended in [Schjøth et al. 2009].

The photon mapping algorithm usually employs two steps: a first step in which photons representing light are traced from the light sources and around the scene, and a second step in which the light transport information generated during the first step is used to reconstruct indirect illumination.

Many of the photons traced during the first step have neighbors that tend to follow the same path. We exploit this coherence by tracing imaginary bundles of photons along each trace. Each of our photons represents a beam of light that expands, contracts and reshapes, according to the reflections and refraction it undergoes as it propagates through the scene. This is achieved by using *ray differentials.*

Ray differentials is a technique introduced by Igehy [1999] that traces two virtual rays along with each real ray by differentiating its position and direction as it traverse the scene. This translates into ray footprints which we use to shape the kernels employed in the density estimate such that they adapt to the illumination structure. In effect we improve the trade-off between variance and bias as compared to other particle tracing algorithms. Alternatively, we could have kept track of the spatial relation between closely related photons as they

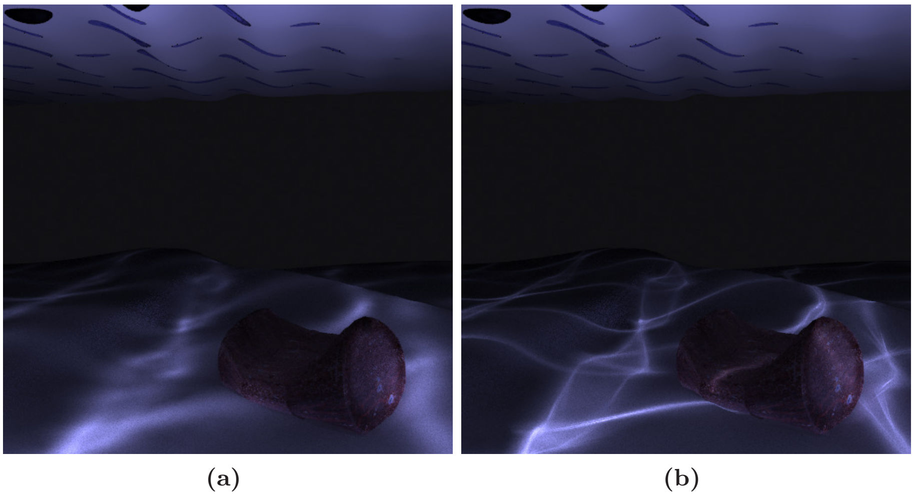


Figure 5.1: Underwater view of seabed. The water is completely clear such that no light scattering occurs. Image (a) rendered using regular photon mapping and (b) using our method. Both images were rendered using a photon map containing 20000 photons.

are traced through the scene. Unfortunately, the relation between photons often becomes meaningless as their paths diverge too much because of the scenes geometric content.

Figure 5.1 illustrates two renderings: one using regular photon mapping and the other using our method. Both are rendered using the same number of photons. As can be seen from the images, our method reproduces caustics with finer details than regular photon mapping.

5.0.1 The kernel function

In photon mapping the kernel function, $K(y)$, is a function that weights a photons flux depending on the photons distance to the estimation point, $\boldsymbol{x}$. It is usually a symmetric unimodal function that decreases monotonically; weighting photons near $\boldsymbol{x}$ higher than those farther away. In photon mapping the kernel weight should be zero for $-1 \geq y \geq 1$. This is necessary in order to keep the computational cost down, as it allows us to work with a limited data set in each radiance estimate. For instance, were we to use an unnormalized Gaussian kernel then we would have to use the entire photon map in each radiance estimate as not to lose energy due to its unbounded form–this would be prohibitively expensive.

Table 5.1 lists a number of kernel functions suited for the equations used in this chapter. The presented kernel functions are not divide with the area they cover as is usual in classic statistics. As such they are not them self probability

Kernel	$K(y)$	
Cone	$(1-\frac{\sqrt{\lvert y\rvert}}{g})/(1-\frac{2}{3g})$	if $\lvert y\rvert < 1$, $g \geq 1$,
	0	otherwise
Epanechnikov	$2(1-\lvert y\rvert)$	if $\lvert y\rvert < 1$,
	0	otherwise
Silverman	$3(1-\lvert y\rvert)^2$	if $\lvert y\rvert < 1$,
	0	otherwise
	$4(1-\lvert y\rvert)^3$	if $\lvert y\rvert < 1$,
	0	otherwise

Table 5.1: Kernel functions

density functions and do not integrate to one over the domain. The reason for this deviation will become apparent later.

The use of a kernel function helps improve the trade-off between bias and variance. Different kernel functions have different efficiency with the Epanechnikov kernel yielding the best trade-off between variance and bias. However, the difference in trade-off improvement is slight and, accordingly, the choice of kernel should be based on other considerations such as computational efficiency and differentiability. [Silverman 1986; Wand and Jones 1995; Simonoff 1996].

Through this chapter we use Silverman's second order kernel function [Silverman 1986], see Table 5.1. It is not as efficient as the Epanechnikov kernel in regard to the variance/bias trade-off, nor is it as computational inexpensive. However, it is differentiable down to the first order, which is an important property for diffusion based photon mapping.

5.1 Ray differentials

In ray differentials, a parameterized ray is differentiated in order to estimate its propagation as it traverse a scene. Igehy [1999] demonstrated the technique on texture filtering. Later, Suykens *et al.* [2001] expanded ray differentials as to include glossy and diffuse reflection, and Per H. Christensen *et al.* [2003] used ray differentials to perform efficient multi-resolution caching of geometry and textures. Recently, Sporring *et al.* [2009] (See Appendix B) have extended ray differentials such that the full differentials for a parameterized ray are evaluated. Here follows a short introduction to ray differentials.

A parameterized ray is defined by its origin, $\boldsymbol{x}$, and its direction, $\boldsymbol{\omega}$. The differential of a ray is the partial derivative of its position and direction with respect to some initial set of variables. The derivatives can be described compactly by the Jacobian matrix, e.g. the differentials of the position, $\boldsymbol{x}$, in respect to some initial variables u and v can be written as

$$D\boldsymbol{x} = \begin{bmatrix} \frac{\partial x}{u} & \frac{\partial x}{v} \\ \frac{\partial y}{u} & \frac{\partial y}{v} \\ \frac{\partial z}{u} & \frac{\partial z}{v} \end{bmatrix} = \begin{bmatrix} D_u\boldsymbol{x} & D_v\boldsymbol{x} \end{bmatrix}, \qquad \boxed{5.1}$$

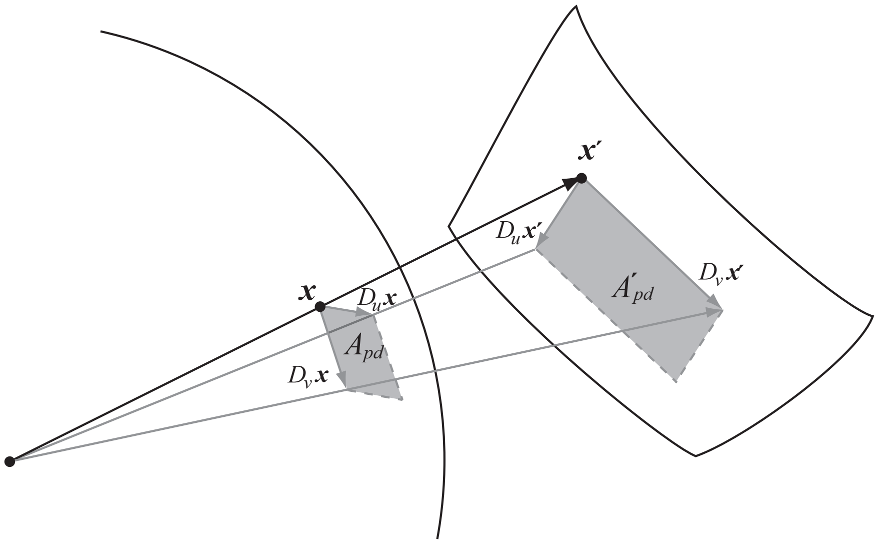


Figure 5.2: Transfer of a ray and its differential vectors from $\boldsymbol{x}$ to $\boldsymbol{x}'$.

where $D\boldsymbol{x}$ is the Jacobian matrix transforming from Cartesian coordinates to a the coordinates u and v. The derivatives describe the spread of the ray beam as it is traced through a scene. The directional derivatives give the rate and direction of change of the ray beam's spread, while the positional derivatives describe its relative size at a given position. In the terminology of Suykens *et al.* [2001], the derivatives multiplied by a finite distance at the offset is the ray's *differential vectors*. We use $D_u\boldsymbol{x}$ to denote a positional differential vector and $D_u\boldsymbol{\omega}$ to denote a directional differential vector; here both with respect to the variable, u.

When a ray intersects an object its positional differential vectors are usually projected down onto the tangential surface of the object at the intersection point. Here they span a parallelogram, see Figure 5.2. This parallelogram is the ray's footprint.

A ray traced through a scene can go through reflections, refractions and transfers. These are simple operations which can be differentiated. The derivatives of a ray going through such interactions can then be computed with respect to the initial offset using the Chain Rule. We refer to Appendix B for the derivatives of these operations.

5.2 Photon differentials

In the following we propose to use ray differentials in connection with photon mapping in order to keep track of the spread of beams of 'photons' as they are traced trough a scene. We call these beams, photon differentials.

Photon differentials are traced through the scene much like ordinary photons; they are stochastically emitted from the light sources, possibly using stratified sampling, and they are traced and stored in a caustics map following the rule that only photon beams having followed a light path from the light source going through one or more specular reflections or refractions before being reflected on a diffuse surface toward the viewer are stored. In the notation of Heckbert [1990] the definition for this light path is LS+DE.

Unlike ordinary photon tracing, the differentials of the photons are accounted for as they are traced through the scene. This is done by keeping track of the positional and directional differential vectors, updating them using Igehys [1999] equations as they are reflected and refracted through the scene.

A photon differential is stored along with information about the positional differential vectors. The exact information stored depends on whether or not filtering is used. Furthermore, it is possible to store in a way that either optimizes for speed or for storage. This is explained later in this section.

In the following we will describe light source sampling and lighting reconstruction. We explain how filtering is performed, and finally we addresses some of the issues concerned with photon differentials.

5.2.1 Emission from a light source

Given a point light source emitting radiant power, Φ_l, uniformly in all directions, the total radiant exitance, M_{total}, leaving the surface of a unit sphere centered around the point light source is

$$M_{total} = \frac{\Phi_l}{4\pi}. \qquad \boxed{5.2}$$

Emitting, n_{pd}, photon differentials, each represents a fraction of the total radiant power of the light sources. If each photon differential spans a fraction of the area of the unit sphere equal to $A_{pd} = 4\pi/n_{pd}$, then a photon differential carries an amount of radiant power equal to

$$\begin{aligned} \Phi_{pd} &= A_{pd} M_{total} && \boxed{5.3} \\ &= \frac{\Phi_l}{n_{pd}}. && \boxed{5.4} \end{aligned}$$

As the photon differential is traced around the scene, the area of the parallelogram spanned by the positional differential vectors changes, $A_{pd} \rightarrow A'_{pd}$. This is illustrated in Figure 5.2.

When the photon differential has been traced around the scene, and has been projected down onto a surface its irradiance can be calculated as

$$E_{pd} = \Phi_{pd}/A'_{pd}. \qquad \boxed{5.5}$$

The irradiance of the photon differential is used to reconstruct the indirect illumination.

In complete analogy to the isotropic point light source, we also consider an area source emitting uniformly in all directions: for a diffuse light source of area A_l, the radiant exitance is

$$M_{total} = \frac{\Phi_l}{A_l} \ . \tag{5.6}$$

Assigning the initial area $A_{pd} = A_l/n_{pd}$ to a photon differential, it will carry the radiant power:

$$\Phi_{pd} = A_{pd} M_{total} = \frac{\Phi_l}{n_{pd}} \ . \tag{5.7}$$

The irradiance due to a photon differential incident on a diffuse surface is then found exactly as in Equation 5.5 (where the area A'_{dp} is a modification of the initial photon differential area A_{dp}. The initial area is modified according to the path which the photon followed and projected onto the surface where the photon is incident).

5.2.2 Lighting reconstruction

Irradiance is radiant power incident per unit area at a point $\boldsymbol{x}$ on a surface. If we consider irradiance due to radiant power incident from one particular solid angle the irradiance will have a directional dependency as well. We have

$$E(\boldsymbol{x}, \boldsymbol{\omega}) = \frac{d\Phi(\boldsymbol{x}, \boldsymbol{\omega})}{dA}. \tag{5.8}$$

By the definition of radiance it follows that

$$L_i(\boldsymbol{x}, \boldsymbol{\omega}) = \frac{d^2\Phi(\boldsymbol{x}, \boldsymbol{\omega})}{(\boldsymbol{n_x} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} dA} = \frac{dE(\boldsymbol{x}, \boldsymbol{\omega})}{(\boldsymbol{n_x} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega}} \tag{5.9}$$

or, in other words,

$$L_i(\boldsymbol{x}, \boldsymbol{\omega})(\boldsymbol{n_x} \cdot \boldsymbol{\omega}) d\boldsymbol{\omega} = dE(\boldsymbol{x}, \boldsymbol{\omega}). \tag{5.10}$$

Then the reflected radiance at $\boldsymbol{x}$ in direction $\boldsymbol{\omega}$ is [Nicodemus et al. 1977]

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}) = \int_{\Omega_x} f_r(\boldsymbol{x}, \boldsymbol{\omega}', \boldsymbol{\omega}) L_i(\boldsymbol{x}, \boldsymbol{\omega})(\boldsymbol{n_x} \cdot \boldsymbol{\omega}') d\boldsymbol{\omega}' \tag{5.11}$$

$$= \int_{\Omega_x} f_r(\boldsymbol{x}, \boldsymbol{\omega}', \boldsymbol{\omega}) dE(\boldsymbol{x}, \boldsymbol{\omega}). \tag{5.12}$$

Using this equation, it is possible to approximate the reflected radiance term of the rendering equation using irradiance due to the radiant power incident from a particular solid angle. This irradiance is exactly what we obtain from the photon differentials, see Equation 5.5. We have

$$L_r(\boldsymbol{x}, \boldsymbol{\omega}) \approx \widehat{L}_r(\boldsymbol{x}, \boldsymbol{\omega}) = \sum_{pd=1}^{n} f_r(\boldsymbol{x}, \boldsymbol{\omega}_{pd}, \boldsymbol{\omega}) \Delta E_{pd}(\boldsymbol{x}, \boldsymbol{\omega}_{pd}), \tag{5.13}$$

where n is the number of photon differentials whose footprints overlap $\boldsymbol{x}$, and $E_{pd}(\boldsymbol{x}, \boldsymbol{\omega}_{pd})$ is the irradiance of the footprint. When finding the overlap, the footprint of each photon differential is centered around the intersection point.

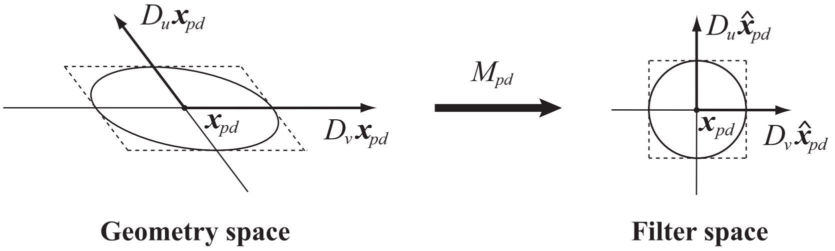


Figure 5.3: 2D illustration of a transformation from geometry space to filter space by the matrix $\boldsymbol{M}_{pd}$. The ellipse inside the parallelogram is the footprint of the photon differential. When transformed into filter space the ellipse becomes a unit circle.

In practice n is found by collecting the photon differentials nearest to $\boldsymbol{x}$ and rejecting the photon differentials not affecting the estimate. Only the photon differentials within a certain fixed radius need to be collected. The radius is limited to half the length of the longest positional differential vector in the photon map. Photon differentials further away do not affect the estimate.

5.2.3 Kernel smoothing

Equation 5.13 provides no smoothing when estimating the illumination. To provide kernel smoothing we reformulate the equation such that

$$\widehat{L}_r(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{pd=1}^{n} f_r(\boldsymbol{x},\boldsymbol{\omega}_{pd},\boldsymbol{\omega}) K\left((\boldsymbol{x}-\boldsymbol{x}_{pd})^T \boldsymbol{M}_{pd}^T \boldsymbol{M}_{pd}(\boldsymbol{x}-\boldsymbol{x}_{pd})\right) \Delta E_{pd}(\boldsymbol{x},\boldsymbol{\omega}_{pd}). \tag{5.14}$$

K is a kernel function from Tabel 5.1, and $\boldsymbol{M}_{pd}$ is the matrix that transforms from world coordinates into a coordinate system in which the surface normal and the positional differential vectors, $D_u\boldsymbol{x}$ and $D_v\boldsymbol{x}$, are basis vectors. This transformation is illustrated in Figure 5.3. We use half the length of the differential vectors, as we center the footprint around the photon differentials intersection point, $\boldsymbol{x}_{pd}$.

Conceptually, each photon differential is associated with an ellipsoid, which in practice works as a three dimensional anisotropic kernel. The ellipsoid is spanned by the positional differential vectors and the surface normal of the object intersected by the photon differential, see Figure 5.4. When filtering, we estimate the irradiance of a photon differential, E_{pd}, using the cross-sectional area, A_{pd}, of the ellipsoid. This cross-section is an ellipse contained within the parallelogram spanned by the positional differential vectors of the photon differential. The area of the ellipse is estimated as

$$A_{pd} = \frac{1}{4}\pi |D_u\boldsymbol{x} \times D_v\boldsymbol{x}|. \tag{5.15}$$

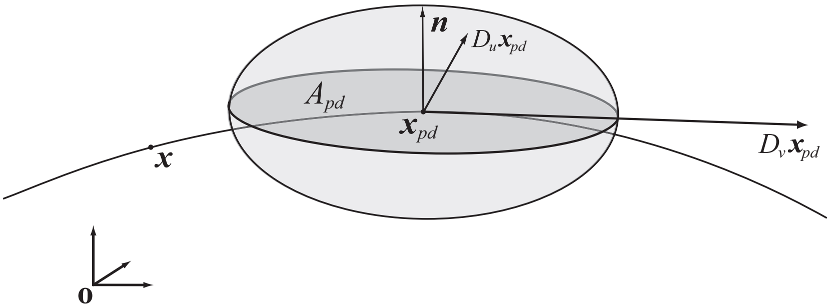


Figure 5.4: Filter kernel of a photon differential illustrated as an ellipsoid. The size and shape of the kernel is defined by the surface normal of the intersected object and the positional differential vectors of the photon differential.

With kernel smoothing we further improve the trade-off between variance and bias. It gives the freedom to choose a suitable kernel depending on the task and purpose.

5.2.4 Implementation

When emitting photon differentials from a light source, the initial size of the photon differential's footprint is a smoothing factor that affects the final image. The size of the footprint corresponds to the bandwidth in the radiance estimate. In effect, a large initial footprint will reduce noise by promoting bias, whereas a small initial footprint will have the opposite effect. The initial footprint size is either set manually or as a function of the number of photons. When changing the initial footprint size, it is important to adjust the radiant power of the photon differentials such that their irradiance remains constant. In other words, we want to balance the equation, $E_{pd} = \Phi_{pd}/A_{pd}$, such that we neither add nor subtract from the total spectral energy of the scene. In this paper we control the footprint size and thus the trade-off between variance and bias using a parameter, s.

The transformation matrix, $\boldsymbol{M}_{pd}$, is a 3 by 3 matrix. It is either stored along with the photon differentials demanding an additional 40[1] bytes per photon differential, or the matrix can be constructed during run-time thereby demanding only 24 extra bytes per photon differential in order to store the positional differential vectors. In our implementation we use the former procedure as the latter imposes an overhead on the estimation time.

A problem is that photon differentials provide no security for the support size of an estimate in a given area. This means that if the footprints of the photon differentials are small or oblong in an area with low photon density, noise may appear. To help this problem, we suggest that *density control* [Suykens and Willems 2000] is employed. Note, however, that we do not use density control in our implementation.

[1] 36 bytes for the matrix and 4 bytes for the area of the footprint.

5.3 Summary

In this chapter we investigated the use of ray differentials for light reconstruction in photon mapping, presenting a method we call photon differentials. First part of the chapter shortly described the photon differentials in the context of density estimation using this context to compare it to existing global illumination methods based on density estimation. From this comparison we saw that the main difference is that photon differentials uses an anisotropic kernel together with the variable kernel density estimator. Following this a short introduction to ray differentials was given. This introduction was continued with a thorough description of photon differentials followed by a few relevant implementation details.

In the next chapter photon differentials will be investigated in a comparison study with conventional photon mapping and diffusion based photon mapping.

6 Analysis

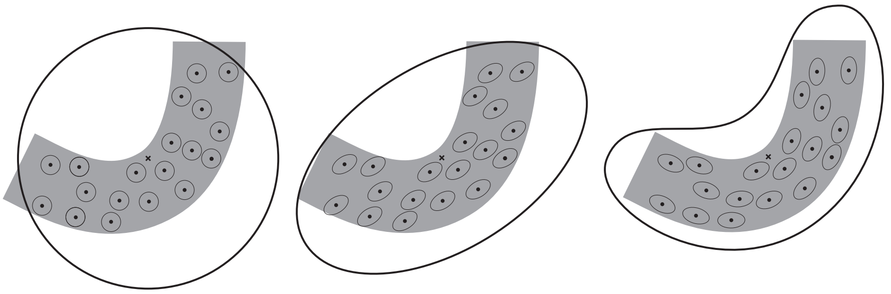
This chapter investigates different approaches that seek to alleviate the trade-of problem described in Chapter 2. The methods that are compared and examined are; traditional photon mapping [Jensen 1996], diffusion based photon mapping [Schjøth et al. 2006; Schjøth et al. 2008], and photon differentials [Schjøth et al. 2007]. Chapter 3 shortly describes photon mapping, Chapter 4 examines diffusion based photon mapping while photon differentials is described in Chapter 5.

The three methods each take advantage of different smoothing schemes in order to improve the trade-of between variance and bias. In terms of density estimation, these schemes can be categorized as adaptive kernel estimators, see Section 2.2. Jensen's photon mapping and the diffusion based approach are local-bandwidth kernel estimators the main difference being that Jensen uses an isotropic kernel that adapts its size in relation to the k'th nearest neighbor distance, while the diffusion based approach uses an anisotropic kernel which is shaped by the first order structure of the photon distribution. Comparably, photon differentials uses an anisotropic variable bandwidth estimator based on ray differentials.

The diagrams in Figure 6.1, illustrates the difference between the density estimates of the compared methods. Figure 6.1(a) illustrates Jensen's k'th nearest neighbor approach. The 'x' marks kernel center–the estimation point–and the large circle the extend and shape of the kernel. The encircled dots are 'photons' in a distribution that approximates illumination forming the broad rounded line illustrated in grey.

A density estimate can either be thought of as placing a kernel over the distribution with the estimation point as center, or as placing a kernel over each sample point. Following that thought, the larger circle in 6.1(a) illustrates the former idea while the smaller circles surrounding the sample points illustrates the latter. The circles surrounding the samples points have been scaled down to accommodate illustration. They should have had the same size as the large circle for the illustration to be in measure.

Figure 6.1(b), illustrates the approach used with diffusion based photon mapping. Here, a single rotational variant kernel is found for each density estimate based on an average of the distribution's first order structure.



(a) Regular photon mapping **(b)** Diffusion based photon mapping **(c)** Photon differentials

Figure 6.1: The diagrams illustrates the conceptual difference between the density estimates used by the methods investigated in this chapter. 'x' marks the estimation point and the center of the kernel, the kernels shape and size is illustrated by the black line surround all the encircled dots. The encircled dots are 'photons' in a distribution that approximates illumination forming the broad rounded line illustrated in grey and the black line surround each photon illustrates the shape of the kernel placed over each.

The approach used by photon differentials is illustrated in Figure 6.1(c). The shape of the kernel used with this approach is better adapted to the structure of the illumination. Each sample point has its own kernel. Together these convolute to a kernel that resembles the shape of the illumination more closely than both traditional and diffusion based photon mapping.

We evaluate the methods in two simple case studies. The cases are simple scenes that both produce a caustic; one scene creates a caustic by refraction and the other by reflection. The purpose of these case studies is to compare the methods ability to reproduce fine structures in indirect lighting.

6.1 Image Quality Measures

In the evaluation of the different renderings, we make use of two different objective image quality measures. Objective image quality measures differ from subjective image quality measures in that they are quantitative, and results are reproducible. Where subjective measures are typically based on qualitative analysis using human observers, objective measures automatically predict perceived image quality based on some mathematical model. Objective image quality measures can be grouped in measures that are generically mathematical and measures that specifically are based on models of the human visual system (HVS). From the former group we use the well known *Mean Integrated Square Error* (MISE) and from the latter group we use the *Structural SIMilarity* (SSIM) index; a method introduced by Wang *et al.* [2004].

The *Mean Square Error* (MSE) is a measure of the difference at a fixed point between an estimator and the function or quantity to be estimated. Similarly, MISE estimates the mean square error of an estimator, but in contrast to MSE the error is estimated over the entire real line. As discussed in Section 2.1.2, MISE is the error of the estimator encompassing both variance and bias. In statistical density estimation, MISE is a standard tool for analysis; it is often used to obtain rate of convergence and optimal bandwidth for kernel density estimators [Silverman 1986; Wand and Jones 1995; Simonoff 1996].

Likewise, MSE is well known and widely used measure. It is easy to estimate and optimize and results have a clear meaning. However, as a quality measure MSE is inconsistent with the human visual system [Girod 1993; Teo and Heeger 1994; Wang and Bovik 2002].

The SSIM index measures the similarity between two images; typically a reference image and its degraded counter part. An index of one means that the two images are identically while a measure of zero means that the images have no similarity. The SSIM method separates image information about contrast, luminance, and structure. These separated components are compared between the two images, and the results are weighted together to a finale similarity index.

SSIM has been noted as a better image quality measure. Kanters, in his Ph.D. thesis [2007], compares different objective image quality measures–including both generic mathematical and the complex HVS based measures–against results using human observers. Kanters asserts that, ”Of all the tested error measures, the Structural Similarity Measure (SSIM) probably resembles the human observer results best.”

Cadík and Slavík [2004] evaluates two different image quality measurement approaches; a structural based approach, represented by SSIM, and a traditional approach, represented by the Visible Differences Predictor. They conclude “that the structural based approach outperforms the traditional approach for involved input stimuli.”

6.2 Refraction

The first case presented is that of a diacaustic created by refraction of light through a simple sinusoidally shaped water wave. Although sinusoidal waves are not the most accurate model for simulating real world water waves, combinations of sinusoidal waves are often used in computer graphics to simulate waves in open water. We find the sinusoidal wave to be well suited for this case study due to its simplicity.

The two case studies are diagrammed in Figure 6.2. Figure 6.2(a) is a diagram of the first case study, it illustrates the interaction between a finite number of collimated light rays–approximating sunlight–and a sinusoidally shaped water wave. The transmission into water causes the light rays to refract, thus creating a caustic where they intersect the bottom line. The form of the caustic depends on the shape of the wave, the distance to the bottom line, and the refractive

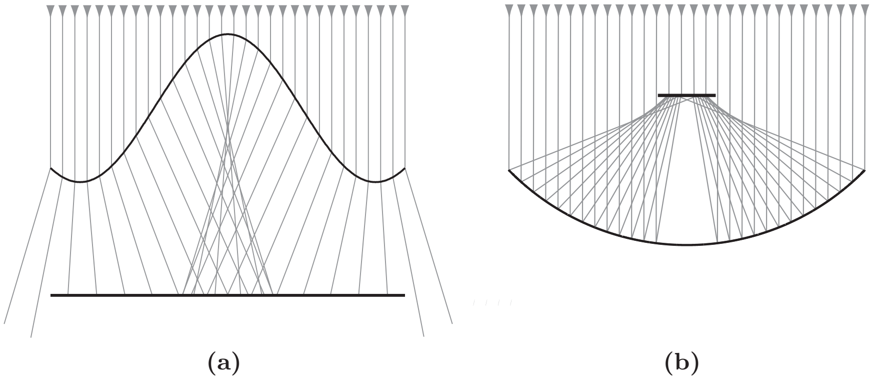


Figure 6.2: Diagram of the case study scenes. (a) Illustrates a sinusoidal shaped water wave illuminated from above by a finite number of collimated light rays. (b) Illustrates a clipped metal ring illuminated by a finite number of collimated light rays.

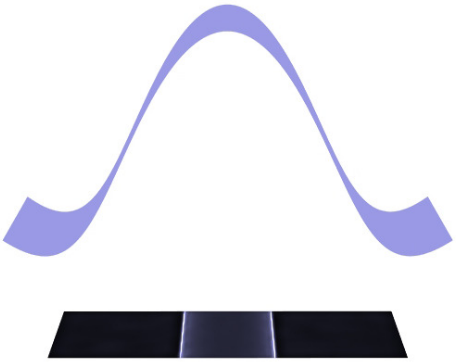


Figure 6.3: Rendering of the refraction case study scene. Rendered with photon differentials using 20 000 photons.

indices of the media in which the rays traverse. In this case light rays create two bright points focused where the ray coherence is high. In-between these focal points is a slightly less bright area. On either side of this region the ray coherence is low, giving the darkest areas.

Figure 6.3 is a rendering of the first case study. The image was rendered with photon differentials using 20 000 photons. The rendering mirror the diagram quite closely; high intensity in the narrow bands of the focal areas, lower intensity in between and lowest intensity on either side of the region bordered by the areas of focus. Now, to approximate the caustics most accurately, how narrow should the focus bands be? How sharp should the edges be? In regular photon mapping the answer can be found in classical density estimation. A kernel density estimate $\hat{f}$, is a kernel smoothed version of the true function, f in addition to random error. Increasing the number of samples will reduce the variance, making the estimate converge to the true function convolved with the

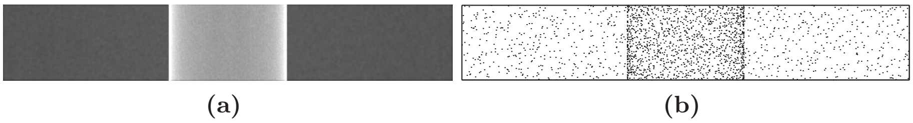


Figure 6.4: Refraction case study - reference images. (a) is a rendering of the caustic seen in the refraction case study scene, Figure 6.3. The image was created with k'th nearest neighbor photon mapping using a photon map containing 4 million photons. (b) visualizes part of the photon distribution used to create (a). Only 2 000 photons are shown as to facilitate visualization.

kernel. If the bandwidth at the same time goes to zero, then our estimate will converge to f. Adjusting the bandwidth controls the trade-of between bias and variance.

In k'th nearest neighbor photon mapping, the bandwidth is variable. It is determined by the number of photons used in each radiance estimate. To increase the accuracy of the lighting function the total number of photons should be increased thus decreasing the bandwidth. Similarly, our method will converge toward the true lighting function as the total number of photons increase and initial differential size decreases.

As a reference image the scene has been rendered with 4 million photons using regular photon mapping. The camera has been positioned as to solely capture the caustic. Figure 6.4(a) shows this rendering, while Figure 6.4(b) reproduces part of the used photon distribution.

6.3 Reflection

The diagram of the reflection case study, Figure 6.2(b), is that of a clipped metal ring illuminated by a finite number collimated light rays. The rays reflection inside the metal ring creates envelopes of light that intersects a plane, stopping them from meeting in the classic cusp known from the bottom of coffee cups. The interaction creates two distinct focal points on the plane. Furthermore, the plane is shadowing itself forming a dark area with shadow borders in the middle of the plane.

Figure 6.5(a) is a reference image of the caustic rendered with regular photon mapping using a photon map containing 4 million photons. In Figure 6.5(b) part of the photon distribution used in the rendering is shown. As with the refraction case study scene the camera has been positioned as to solely capture the caustic.

6.4 Optimal bandwidth

The bandwidth for the k'th nearest neighbor photon mapping is the number of photons per radiance estimate, k. In diffusion based photon mapping the band-

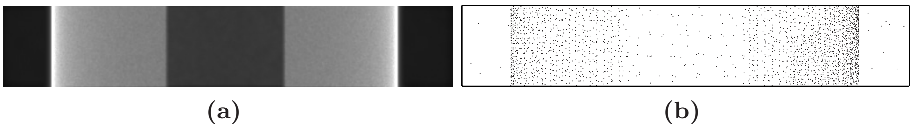


Figure 6.5: Reflection case study - reference images. (a) is a rendering of the caustic seen in the reflection case study scene. The image was created with k'th nearest neighbor photon mapping using a photon map containing 1 million photons. (b) visualizes part of the photon distribution used to create (a). Only 2 000 photons are shown as to facilitate visualization.

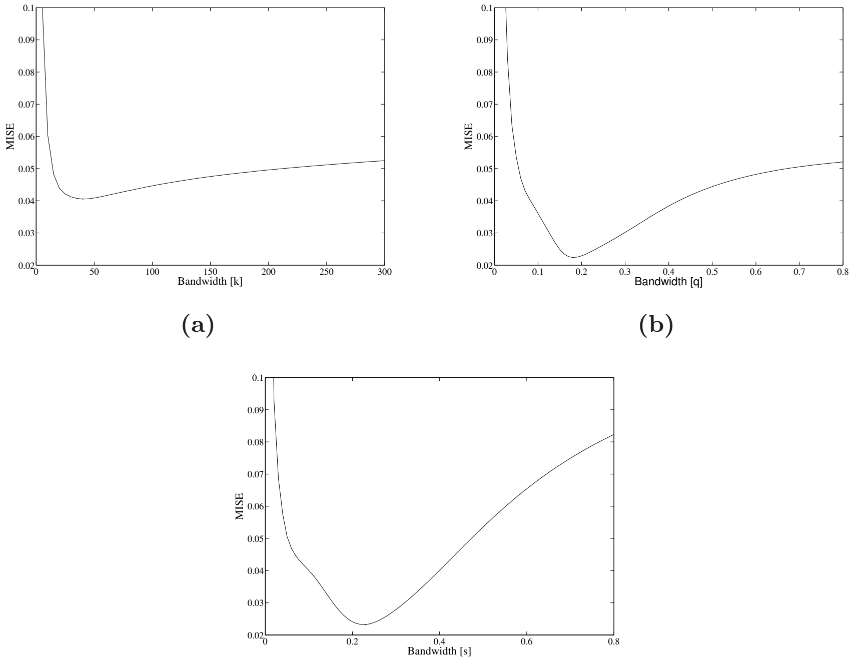
width depends on two parameters; the diffusivity coefficient, q, and the support radius, h. For our anisotropic variable bandwidth estimator it is the smoothing parameter, s, which decides the initial size of the photon differential's footprint. For a photon map with a fixed number of photons the optimal bandwidth is the bandwidth that gives the best trade-of between variance and noise. The optimal bandwidth depends on the measure used.

To find the optimal bandwidth for the three methods we employ MISE and SSIM as image quality measures. With MISE, the optimal bandwidth is found as the one that yields the smallest error as compared to the reference image. For SSIM the optimal bandwidth is the one, for which the similarity with the reference image is highest.

Using a photon map containing 20 000 photons we render the scene with the same camera placement as that used to render the reference image. We do this a number of times increasing the bandwidth for each rendering. This yields a number of images, for which we estimate MISE in respect to the reference image. Likewise, we estimate the similarity between these images and the reference image using the SSIM measure.

Figure 6.6 shows three graphs where MISE is plotted against the bandwidth for the refraction case study . The leftmost graph, (a), is for regular photon mapping, the center, (b), is for diffusion based photon mapping, while the rightmost graph, (c), is for differential photon mapping. Diffusion based photon mapping is a special case as bias/variance trade-of depends on both the diffusivity coefficient and the support radius–to compare the diffusivity coefficient is plotted against MISE using the optimal support radius. The three graphs reflect how noise initially is the predominantly cause of error. As the bandwidth increases the noise is reduced and the error decreases. This changes as bias increases and slowly becomes the main source of error making the error level rise again. Minima of the graphs are the optimal bandwidth for MISE.

The graphs in Figure 6.7 plots SSIM index against the bandwidth for the first case study. These manifest the same trend as the graphs plotting MISE against the bandwidth. In the graphs the similarity is initially low but rapidly increases with the bandwidth. This increase reaches its maximum where noise gives way for bias. The maxima of the graphs are at the optimal bandwidths for SSIM measure.



(a) **(b)**

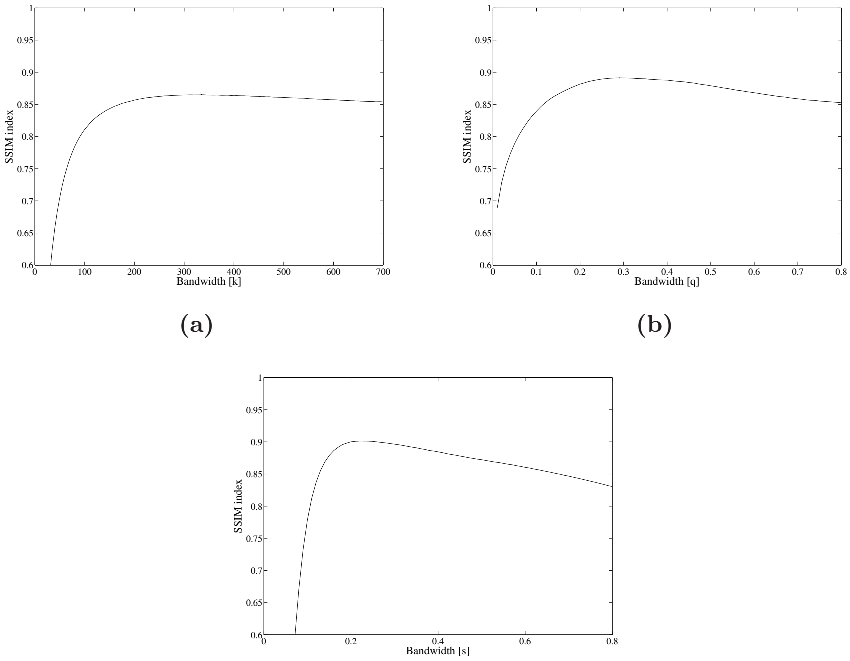
(c)

Figure 6.6: Graphs plotting bandwidth against the mean integrated square error. The measured images are renderings of the refraction case study using a photon map containing 20 000 photons. Graph 6.6(a) is based on images rendered using regular photon mapping, graph 6.6(b) is based on images rendered using diffusion based photon mapping, while graph 6.6(c) is based on images rendered using photon differentials.

The left image-column of Figure 6.8 shows images of the refraction case study scene rendered at MISE-optimal bandwidth. The first image-row is rendered using regular photon mapping, the second is rendered using diffusion based photon mapping and the last image-row is rendered using differential photon mapping. The images contain clearly visible noise indicating that MISE favors noise over bias to a higher degree than a human observer probably would. This substantiates the postulation, discussed above, that MISE might not be a good image quality assessor. The second image-column of Figure 6.8 shows the images of the case study scene rendered at SSIM-optimal bandwidth. These images contain less noise and is possible closer to the results that would be obtained from a trials with human observers.

6.4.1 Comparison

Evaluating our method we compare against standard photon mapping and diffusion based photon mapping at optimal bandwidths. The images in Figure



(a)

(b)

(c)

Figure 6.7: Graphs plotting bandwidth against structural similarity index. The measured images are renderings of the refraction case study using a photon map containing 20 000 photons. Graph 6.7(a) is based on images rendered using regular photon mapping, graph 6.7(b) is based on images rendered using diffusion based photon mapping, while graph 6.7(c) is based on images rendered using photon differentials.

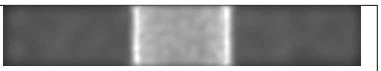
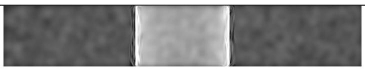
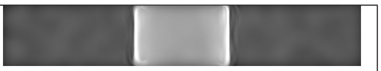
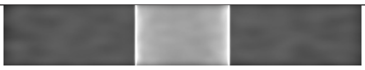
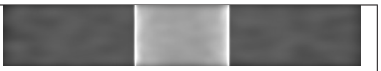
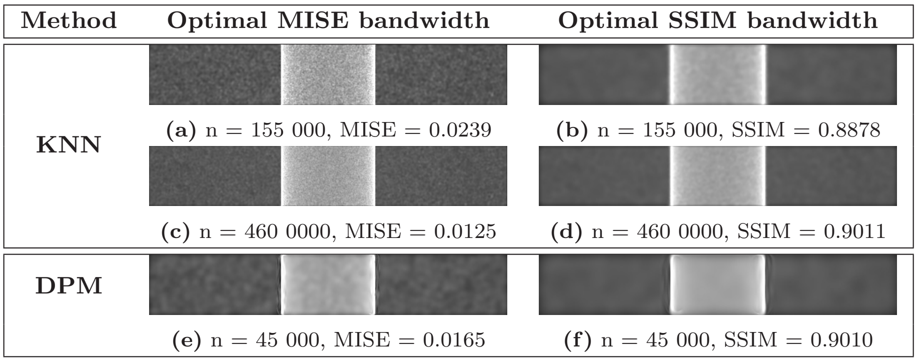
Method	Optimal MISE bandwidth	Optimal SSIM bandwidth
KNN	 **(a)** MISE = 0.0405	 **(b)** SSIM = 0.8650
DPM	 **(c)** MISE = 0.0224	 **(d)** SSIM = 0.8912
PD	 **(e)** MISE = 0.0232	 **(f)** SSIM = 0.9014

Figure 6.8: Renderings of the refraction case study scene using a photon map containing 20 000 photons. First image-row rendered with regular k'th nearest neighbor photon mapping (KNN), the second image-row was rendered with diffusion based photon mapping (DPM), and the last image-row was rendered with photon differentials (PD). First image-column was rendered at the optimal bandwidth for MISE while second image-column was rendered at the optimal bandwidth for SSIM.



(a) n = 155 000, MISE = 0.0239 **(b)** n = 155 000, SSIM = 0.8878

(c) n = 460 0000, MISE = 0.0125 **(d)** n = 460 0000, SSIM = 0.9011

(e) n = 45 000, MISE = 0.0165 **(f)** n = 45 000, SSIM = 0.9010

Figure 6.9: Renderings of the refraction case study scene. The images in the first two image-rows was rendered using regular k'th nearest neighbor photon mapping (KNN), while the images in the last two rows was rendered using diffusion based photon mapping (DPM). Images in the left image-column was rendered at optimal MISE-bandwidth while the images in the right image-column was rendered at optimal SSIM-bandwidth. The number of photons in the photon map used to render the individual images is denoted by n.

6.8 and 6.9 are renderings of the refraction scene. The images in Figure 6.8 were rendered with the three methods all using a photon map containing 20 000 photons.

The images, Figure 6.9(a)(d), were found by increasing the number of photons contained in the photon map until the image quality measure for the optimal bandwidths was approximately the same as that for photon differentials, Figure 6.8(e)(f). From this we see that in order to achieve a MISE comparable to our method using standard photon mapping, the photon map had to be increased from 20 000 photons to 155 000 photons. See Figure 6.9(a) and 6.8(a). From Table 6.1 we see that this increase causes the number of floating point operations needed to render the image using standard photon mapping to exceed that needed to render the image of similar quality–according to MISE–using photon differential. Furthermore, comparing the two images, Figure 6.8(e) and 6.9(a), a human observer would perhaps not judge the images to be of similar quality, probably voting in favor of our method.

Using an image quality measure more compatible to the human visual system, the SSIM index distinguishes photon differentials even more as a superior method. To achieve the same SSIM index with standard photon mapping as with photon differentials the number of photons in the photon map must be increased from 20 000 to 460 000. This increase causes the floating point operations needed to render the image using standard photon mapping to increase to almost seven times that needed to render an image of similar quality–according to SSIM–using photon differential, refer to Table 6.1.

Comparing photon differentials with diffusion based photon mapping the results are more equal. At optimal MISE bandwidth diffusion based photon mapping actually outperforms photon differentials with a margin, see Figure 6.8(c) and (e). The image, 6.8(c), produced by diffusion based photon mapping contains

Case Study	Method	Figure	Photons in map	MISE	Number of FPOs [10^9]	SSIM index	Number of FPOs [10^9]
Refraction	Regular	6.8(a)(b)	20 000	0.0405	0.70	0.8650	1.59
		6.9(a)(b)	155 000	0.0239	5.01	0.8878	7.42
		6.9(c)(d)	460 000	0.0125	12.88	0.9011	15.95
	DPM	6.8(c)(d)	20 000	0.0224	1.03	0.8912	2.81
		6.9(e)(f)	45 000	0.0165	2.22	0.9010	4.91
	Differentials	6.8(e)(f)	20 000	0.0232	2.26	0.9014	2.28
Reflection	Regular	6.10(a)(b)	20 000	0.0451	0.51	0.8419	1.37
		6.10(c)(d)	190 000	0.0238	4.45	0.8816	6.64
		6.10(e)(f)	380 000	0.0179	9.46	0.8931	11.42
	DPM	6.10(g)(h)	20 000	0.0204	1.01	0.8783	2.60
		6.10(i)(j)	22 000	0.0203	1.11	0.8819	3.18
		6.10(k)(l)	33 000	0.0180	1.74	0.8940	4.81
	Differentials	6.10(m)(n)	20 000	0.0178	2.51	0.8821	3.15

Table 6.1: Performance results for regular photon mapping, diffusion based photon mapping (DPM), and photon differentials. The tabel lists the number of floating point operations (FPOs) need to render the scene with a given number of photons in the photon map, at optimal values of the mean integrated square error (MISE) and the structural similarity (SSIM) index.

a lot of noise and additional ringing along the edges and might not be the first choice in a human trial. At optimal SSIM bandwidth, however, diffusion based photon mapping needs more than twice as many photons in order to achieve the same similarity index as photon differentials resulting in an expenditure of twice as many floating point operations, see Figure 6.8(f) and 6.9(f).

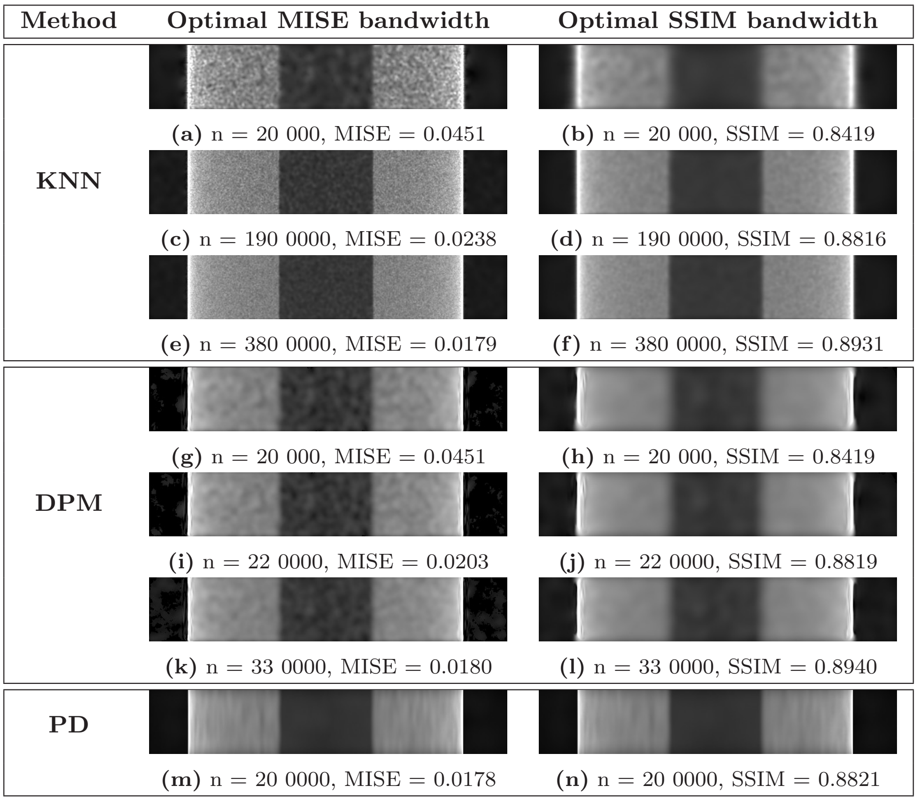
The reflection case study is investigated in the same manner as the first. Here we see a similar trend, but the performance of our method is not quite as good. At similar MISE, k'th nearest neighbor photon mapping uses almost twenty times as many photons as with photon differentials (Figure 6.10(e) and 6.10(m)). At similar SSIM index standard k'th nearest neighbor photon mapping uses around seven times as many photons as our method, see Figure 6.10(d) and 6.10(n). Consulting Tabel 6.1 we see that the gain in computational performance for this scene is not as high as for the first case study. Understandably, the performance gain is highly scene dependent.

Computational performance and image quality measures for optimal bandwidths can be seen in Tabel 6.1.

6.5 Summary

In this chapter we have examined and compared photon differentials, regular k'th nearest neighbor photon mapping and diffusion based photon mapping using two simple case studies. We used two different image quality measures to find the optimal bandwidth for the methods in order to make the comparison as objective as possible. The image quality measures used were the mean integrated square error and the structural similarity index.

From this examination we render probably that with a map of photon differentials one obtains a better caustic quality with far less photons than both regular and diffusion based photon mapping. This potentially saves a large amount of time in renderings that require caustics with high quality edges. We argue that with conventional photon mapping it takes an enormous amount of photons



(a) n = 20 000, MISE = 0.0451

(b) n = 20 000, SSIM = 0.8419

(c) n = 190 0000, MISE = 0.0238

(d) n = 190 0000, SSIM = 0.8816

(e) n = 380 0000, MISE = 0.0179

(f) n = 380 0000, SSIM = 0.8931

(g) n = 20 000, MISE = 0.0451

(h) n = 20 000, SSIM = 0.8419

(i) n = 22 0000, MISE = 0.0203

(j) n = 22 0000, SSIM = 0.8819

(k) n = 33 0000, MISE = 0.0180

(l) n = 33 0000, SSIM = 0.8940

(m) n = 20 0000, MISE = 0.0178

(n) n = 20 0000, SSIM = 0.8821

Figure 6.10: Renderings of the refraction case study scene. The images was rendered at optimal bandwidth using regular k'th nearest neighbor photon mapping (KNN), diffusion based photon mapping (DPM), and photon differentials (PD). Images in the columns from left to right was rendered at optimal MISE-bandwidth and optimal SSIM-bandwidth. The number of photons in the photon map used to render the individual images is denoted by n.

to generate caustics of the quality needed, for example, for animation feature films. In large scenes the additional memory required to store the footprints of the photons might be a problem.

Part III

Expansion in time and space

7
Temporal Photon Differentials

The temporal aspects of rendering animated scenes have less focus in the graphics community than more classical fields such as global illumination, texturing, animation, fluid simulation, and real–time rendering. In spite of this, rendering animated scenes with global illumination methods produces some interesting problems. The perhaps most prominent problem is due to the fact that the temporal domain is discretized at a often very low resolution.

Feature films usually sport a frame rate around 25 *frames per second* (fps). Despite the fact that the human eye is much more perceptible than 25 fps, this frame rate still gives a seemingly fluid motion when a film is produced with a traditional movie camera. However, were we to produce a feature film of an animated virtual scene using an unmodified global illumination method at a frame rate of 25 fps, fluid motion is not guarantied. A typical unmodified global illumination method produces images at instant time in the temporal domain. This procedure can induce temporal aliasing, which is seen as an adverse stroboscopic effect where the illumination changes rapidly over time. A traditional-camera produced feature film will avoid this problem because the camera has non-zero exposure time. This means that camera-perceived illumination is averaged over the exposure time. In effect, high frequency motion is blurred and therefore seems fluid. This effect is often called *motion blur*. A temporal aliasing defect not solved by this blurring is the wagon-wheel effect, which is seen as a backwards turning of spoked wheels due to temporal under sampling. However, as our method does not address this particular problem, it will not be discussed further here.

Brute force methods, such as the accumulation buffer [Haeberli and Akeley 1990], average together in-between frames in order to achieve motion blur. These methods can achieve arbitrary high accuracy, but are often prohibitively expensive as full renderings typically have to be made of a large number of in-between frames.

Different global illumination methods also address temporal aliasing by simulating motion blur. Distributed ray tracing [Cook et al. 1984] achieves motion blur by stochastically sampling the temporal domain as well as the spatial. Myszkowski *et al.* [2001] adaptively controlled the temporal and spatial sampling resolution by examining local variations of indirect illumination over time and space in a pilot estimate.

With a method they called *time dependent photon mapping*, Cammarano and Jensen [2002] extended photon mapping such that indirect illumination was estimated using a four dimensional photon map that expanded into the temporal domain. In this manner photons were filtered not only based on their spatial position, but also their temporal.

A problem common to the discussed methods is that they all need to be able to draw some amount of information from in-between frames in order to achieve motion blur for illumination that is moving quickly. If the frame rate is too low to capture the motion of the illumination, the illumination will need excessive smoothing in order to avoid the stroboscopic effect. Having this information available places certain restrictions on the animated scene; because a scene description is needed at arbitrary time steps, movement of scene elements either needs to be described as a an analytic function, or movement has to be interpolated between frames. The analytical approach is by far the best but demands a certain complexity of the of the animation application, as well as a tight link between this and the rendering software. The interpolative approach is error bound as the animation curve might not be linear. Furthermore, some accelerator for ray-object intersection (such as a bsp-tree) typically needs to be either rebuild, or at least updated an extra number of times equal to the number in-between frames needed.

In this chapter, we propose a method that produces motion blur, and which neither needs in-between frames, nor to over-smooth indirect illumination with high temporal frequency. Our proposed method is an extension of photon differentials [Schjøth et al. 2007]. It takes advantage of ray differentials [Igehy 1999] and their extension into the temporal domain by Sporring *et al.* [2009] (See Appendix B). We call this method *temporal photon differentials*.

7.1 Temporal photon differentials

In our method each photon represents a beam of light that expands, contracts and reshapes in space as well as time, as it propagates through the scene. We keep track of a photon's coherence by deriving the first order structure of its direction and position with respect to both time and space as it traverse the scene.

Representing a photon as a parametrized ray with origin in $\boldsymbol{x}$ and the directing $\boldsymbol{\omega}$, we describe the derivatives of a photon with two Jacobian matrices; one for the positional derivatives and one for the directional derivatives. The positional derivatives are then given by

$$D\boldsymbol{x} = \begin{bmatrix} D_u\boldsymbol{x} & D_v\boldsymbol{x} & D_t\boldsymbol{x} \end{bmatrix}, \tag{7.1}$$

where $D\boldsymbol{x}$ is the Jacobian of the positional derivatives, and $D_u\boldsymbol{x}$, $D_v\boldsymbol{x}$ and $D_t\boldsymbol{x}$ are column vectors that describe the positional derivatives with respect to the scalar variables u, v, and t. Similarly, we write the directional derivatives of the photon as

$$D\boldsymbol{\omega} = \begin{bmatrix} D_u\boldsymbol{\omega} & D_v\boldsymbol{\omega} & D_t\boldsymbol{\omega} \end{bmatrix}. \tag{7.2}$$

Exactly, as with photon differentials (See Chapter 5), $D_u\boldsymbol{\omega}$, $D_v\boldsymbol{\omega}$, $D_u\boldsymbol{x}$ and $D_v\boldsymbol{x}$ are spatially dependent differential vectors. The directional and positional differential vectors with respect to time are new to photon differentials; they are noted as $D_t\boldsymbol{\omega}$ and $D_t\boldsymbol{x}$ in the above equations. For a scene, in which the light sources are static, these time dependent differential vectors will, initially, be zero. If the scene, additionally, is completely static, they will remain zero-vectors through out the photon's traversal of the scene. In this specific case, our method will behave exactly as ordinary photon differentials: the photons will expand and contract depending on the reflections and refraction encountered during tracing, and their spatial dependent positional differential vectors will form a footprint, which is used in the reconstruction of the indirect illumination.

If, on the other hand, we have a dynamic scene, photon differentials, interacting with a non-static scene element, will attain non-zero time dependent differential vectors. In this case, the derivatives of a dynamic scene-element's surface positions or normals with respect to time will be non-zero:

$$D_t\boldsymbol{n} \neq \boldsymbol{0}, \tag{7.3}$$

or

$$D_t\boldsymbol{q} \neq \boldsymbol{0}, \tag{7.4}$$

where $\boldsymbol{n}$ is a surface normal to the element and $\boldsymbol{q}$ is a position on the element's surface. This again will affect the time dependent derivatives of a photon interacting with the scene element.

Sporring *et al.* reiterates ray differentials such that the full differentials for a parameterized ray are evaluated. This allows for an extension of parameters such that the derivatives of a ray can be considered with respect to time. From Sporring *et al.*'s equations for transfer, reflection and refraction, we observe that non-zero time-dependent element differentials (eg. $D_t\boldsymbol{q}$) propagate through these interactions to the differentials of the interacting photon. We exploit this behaviour such that a footprint from a photon differential traveling in a dynamic scene not only describes the spatial coherence of the ray, but also the temporal coherence of the ray.

When a photon differential hits a surface, its positional differential vectors are by transfer projected onto the surface's tangent plane at the intersection point. The spatial footprint of the photon differential is the area on the tangent plane of a parallelogram spanned by the positional differential vectors as illustrated in Chapter 5, Figure 5.2. The spatial footprint can be used to shape an anisotropic filter kernel as illustrated in Figure 7.1.

The time dependent positional differential vector, $D_t\boldsymbol{x}$, tells us either how the photon's footprint is going to behave over consecutive frames, or how the footprint has behaved in former frames. In the former case, the direction of $D_t\boldsymbol{x}$ predicts the direction on the surface that the footprint will move, and the magnitude of the vector predicts how far the footprint is likely to move. Basically, the magnitude and the direction of $D_t\boldsymbol{x}$ depends on the estimation method used to calculate the time derivatives of an element, which again depends on the geometry representation. In the present method, we simply use finite differences

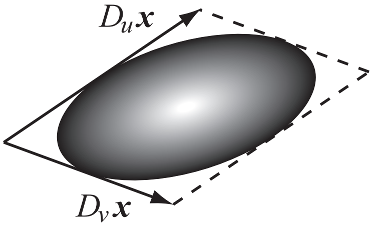


Figure 7.1: Spatial filter kernel shaped by the positional differential vectors, $D_u\boldsymbol{x}$ and $D_v\boldsymbol{x}$.

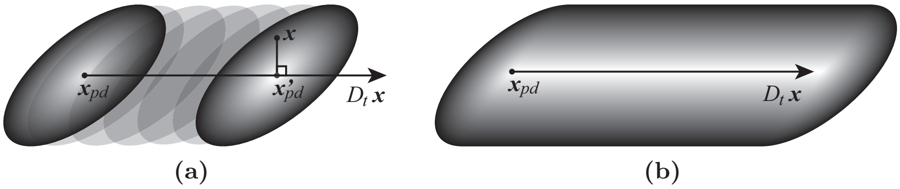


Figure 7.2: Temporal filter kernel shaped by a spatial kernels translation along time dependent differential vector.

and triangle meshes. Except for the last frame, in which we use backward differences, we estimate the time dependent differentials using forward differences. When we want to predict how a footprint is going to behave, having intersected a moving element, we estimate the element's positional time derivatives by

$$D_t\boldsymbol{q}_f = t_s(\boldsymbol{q}_{f+1} - \boldsymbol{q}_f), \tag{7.5}$$

where $D_t\boldsymbol{q}_f$ is derivatives of the vertex q_f with respect to time at frame step f, and t_s is the shutter time. The shutter time is a parameter for how much trust we put in our prediction. Generally, however, it works as a smoothing parameter for the time dependent footprint that decides how much motion blur we induce.

The time dependent footprint constitutes an integration of the spatial footprint over the time dependent differential vector such that the spatial footprint is elongated along the vector. We achieve this by translating the spatial footprint along the time dependent differential vector. As in the spatial case, the time dependent footprint describes a filter kernel. In Figure 7.2(a), $D_t\boldsymbol{x}_{pd}$ is the time dependent differential vector, $\boldsymbol{x}_{pd}$ is the center of the spatial kernel, and $\boldsymbol{x}$ is the estimation point for which the kernel weight is estimated. The kernel is translated along $D_t\boldsymbol{x}$ to the point, $\boldsymbol{x}'_{pd}$, on the line segment, $(\boldsymbol{x}_{pd} \rightarrow \boldsymbol{x}_{pd} + D_t\boldsymbol{x}_{pd})$, where $\boldsymbol{x}'_{pd}$ is the point on the segment having the shortest distance to the estimation point, $\boldsymbol{x}$. Using $\boldsymbol{x}'_{pd}$ as center for the spatial kernel, the resulting time dependent kernel will achieve an elongated shape as illustrated in Figure 7.2(b).

The irradiance of the time dependent photon differential is estimated as

$$E_{pd} = \Phi_{pd}/A_{pd} \tag{7.6}$$

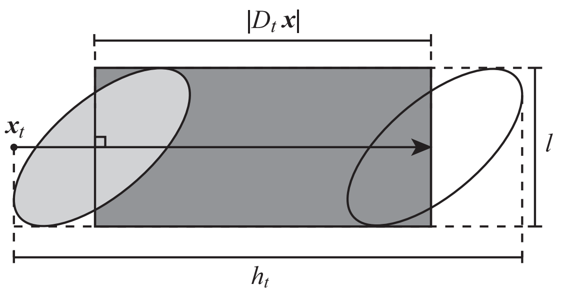


Figure 7.3: The area of the temporal kernel is the sum of spatial kernel area and the area of a triangle with sides lengths l and $|D_t\boldsymbol{x}|$.

where Φ_{pd} is the radiant flux carried by the photon, and A_{pd} is the surface area to which the radiant flux is incident. For the time dependent photon differential, this area is the area of the time dependent kernel. Referring to Figure 7.3 this area is calculated as

$$A_{pd} = \frac{1}{4}\pi|D_u\boldsymbol{x} \times D_v\boldsymbol{x}| + l|D_t\boldsymbol{x}|, \quad \boxed{7.7}$$

where the first term is the area of the spatial kernel and the second term is the area of a rectangle. One side of the rectangle is the length of the time dependent differential vector and the other is the length of the spatial kernel in a direction perpendicular to the time dependent differential vector.

Having defined the time dependent kernel as well as the irradiance of the photon differential, we can now formulate a radiance estimate for temporal photon differentials.

7.1.1 The temporal radiance estimate

Reflected radiance from temporal photon differentials can be estimated by

$$\widehat{L}_r(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{pd=1}^{n} f_r(\boldsymbol{x},\boldsymbol{\omega}_{pd},\boldsymbol{\omega})K\left((\boldsymbol{x}-\boldsymbol{x}'_{pd})^T\boldsymbol{M}_{pd}^T\boldsymbol{M}_{pd}(\boldsymbol{x}-\boldsymbol{x}'_{pd})\right)\Delta E_{pd}(\boldsymbol{x},\boldsymbol{\omega}_{pd}), \quad \boxed{7.8}$$

where $\boldsymbol{x}'_{pd}$ is the translated center of spatial kernel, ΔE_{pd} is the irradiance of the temporal photon differential, and $\boldsymbol{M}_{pd}$ is a matrix that transforms from world coordinates to the filter space of the spatial kernel (See Chapter 5).

The temporal radiance estimate can be extended as to include filtering in time. One intuitive approach is to weight the part of the differential which is closest in time the highest, where the time is estimated form the photon hit point. This can be achieved using a simple univariate kernel as those presented in Table 2.1. To the kernel, we input a distance along the time dependent differential vector,

$D_t\boldsymbol{x}_{pd}$, relative to furthest point of the kernel along negative $D_t\boldsymbol{x}_{pd}$. This is illustrated in Figure 7.3.

With time filtering the temporal radiance estimate is formulated as

$$\begin{aligned}\widehat{L}_r(\boldsymbol{x},\boldsymbol{\omega}) = \sum_{pd=1}^{n} & f_r(\boldsymbol{x},\boldsymbol{\omega}_{pd},\boldsymbol{\omega}) K_s\left((\boldsymbol{x}-\boldsymbol{x}'_{pd})^T \boldsymbol{M}_{pd}^T \boldsymbol{M}_{pd}(\boldsymbol{x}-\boldsymbol{x}'_{pd})\right) \\ & K_t\left(\frac{\boldsymbol{x}_t-\boldsymbol{x}'_{pd}}{h_t}\right)\Delta E_{pd}(\boldsymbol{x},\boldsymbol{\omega}_{pd}),\end{aligned} \qquad \boxed{7.9}$$

where K_s is a bivariate kernel function (See Table 5.1), K_t is a univariate kernel function, h_t is the length of the temporal kernel along $D_t\boldsymbol{x}_{pd}$, and $\boldsymbol{x}_t$ is the furthest point of the kernel in the direction $-D_t\boldsymbol{x}_{pd}$.

With the formulation of the temporal radiance estimate, we now have a method which reconstructs indirect illumination based on a virtual scenes dynamics. This allows for motion blur. In the following we will make a simple analysis of the method.

7.2 Results

We first test our proposed method using a case study. The case study is a simple animated scene in which a sinusoidal wave slowly moves. The wave is illuminated from above by collimated light which it refracts such that the light form caustics on a plane beneath the wave. A virtual camera is placed such that the caustics are clearly visible. The case study is quite similar to the one studied in Chapter 6. The most significant difference being that the scene studied in this chapter is animated.

We have rendered the scene using temporal photon differentials, and Cammarano and Jensen's time dependent photon mapping. The images in Figure 7.4 are renderings of the same frame, but at different shutter times, t_s. They were rendered using temporal photon differentials and a photon map containing only 1000 photons. From the images we see that the temporal photon differentials assume the expected behaviour. As the exposure time increases the caustics are blurred acquiring a comets tail away from the direction of movement. This is the behaviour chosen at implementation time. We could just as well have placed the time dependent kernel centered over the photon intersection point and likewise have centered the time filtering or we could just have centered the filtering. As it is, the time differential is 'trailing' after the photon both in respect to placement and filtering. As we shall see, though it is hardly visible, the same strategy has been implemented for time dependent photon mapping.

The renderings in Figure 7.5 have all except (a) been created with time dependent photon mapping. Additionally, all images was rendered using the same exposure time, namely 9. Figure 7.5(a) has been included for comparison, it was rendered using temporal photon differentials and is a copy of the image in Figure 7.4 with shutter time 9. First of all, what we see from image 7.5(b) is that

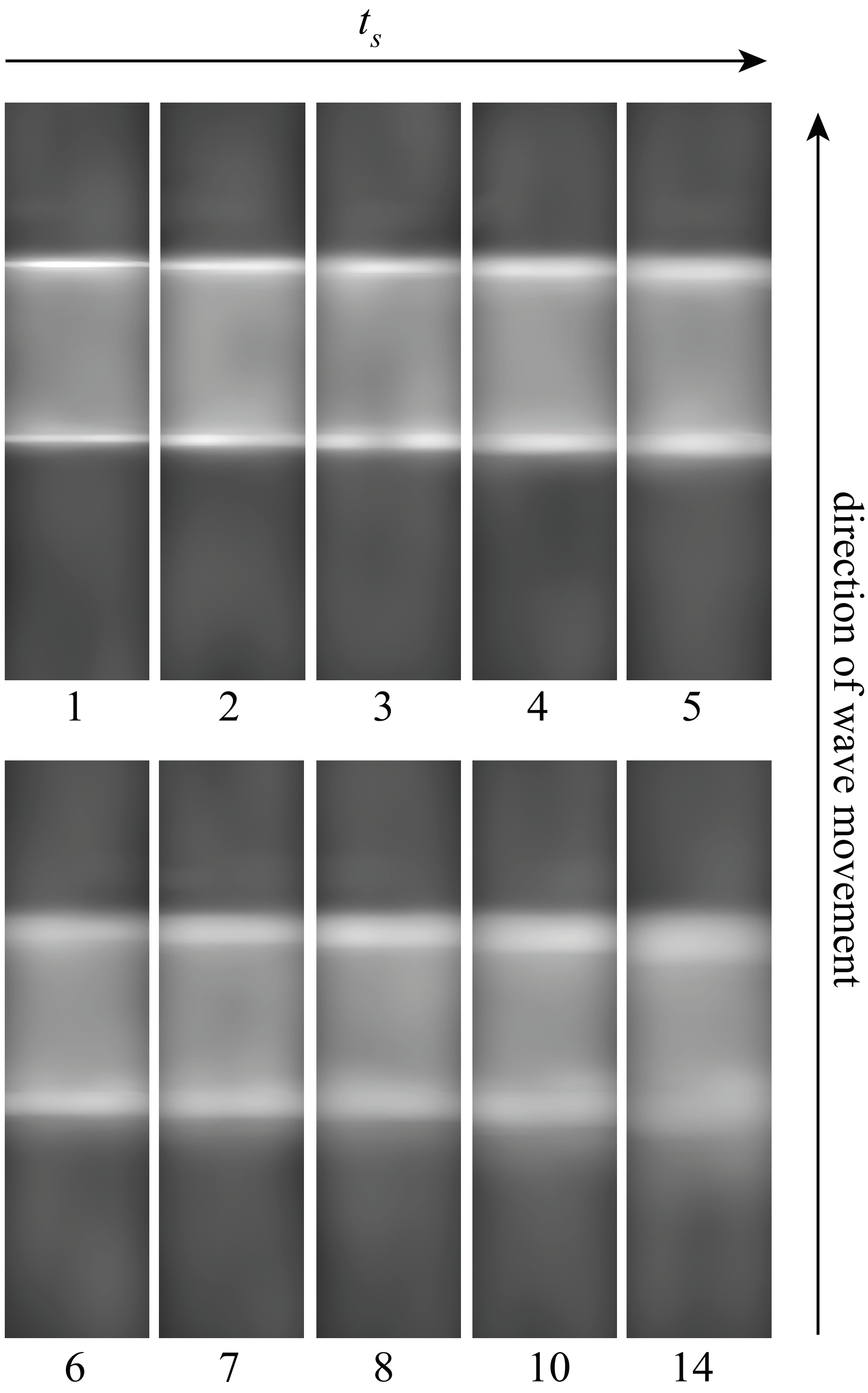


Figure 7.4: Renderings of the case study scene using temporal photon differentials. The number under the renderings indicate shutter time, t_s.

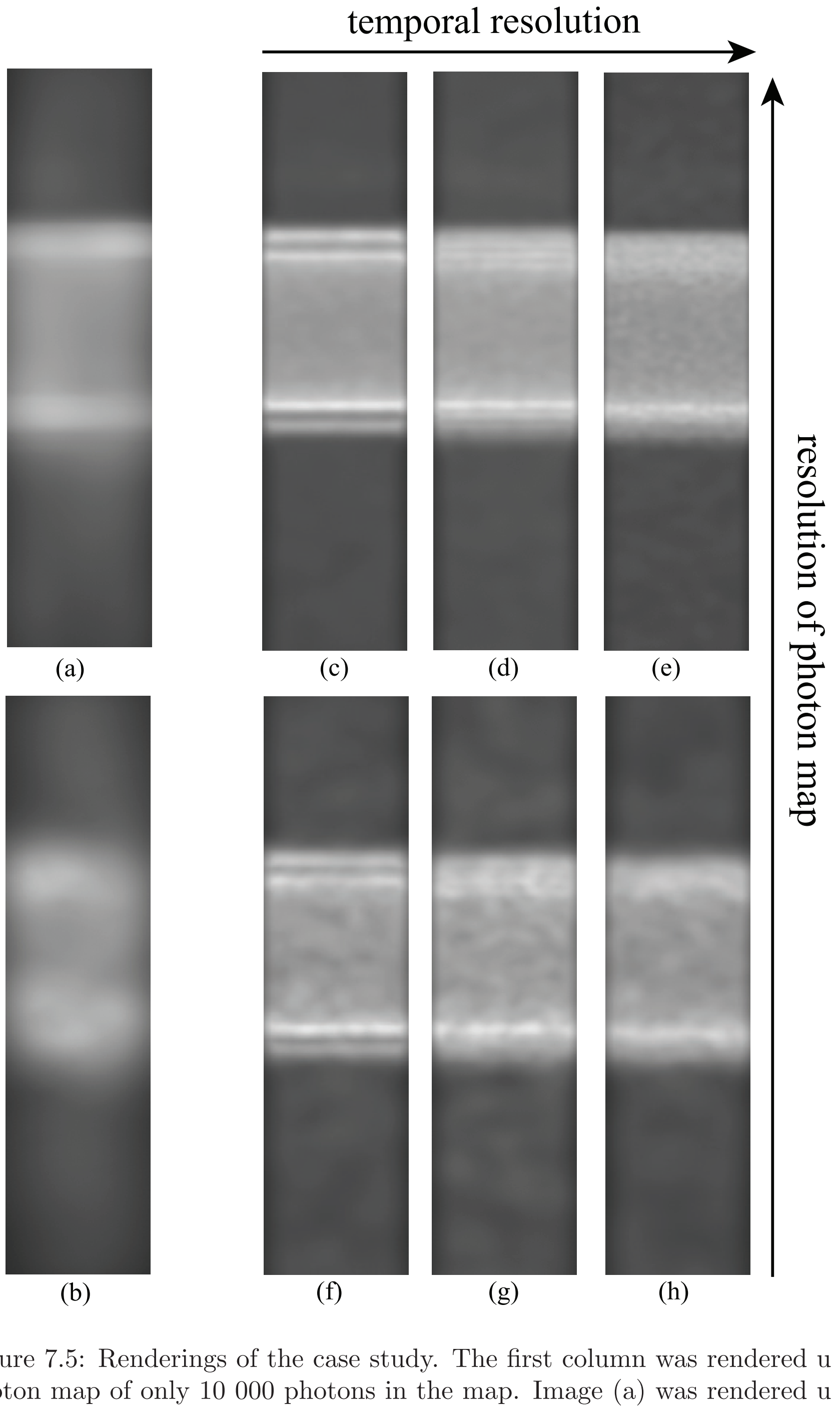


Figure 7.5: Renderings of the case study. The first column was rendered using photon map of only 10 000 photons in the map. Image (a) was rendered using temporal photon differentials the rest using time dependent photon mapping. Of the three rightmost columns the topmost row was rendered using a photon map containing 480 000 photons while the bottom row was rendered using a photon map containing 40 000 photons. From left to right the three columns were rendered using 1, 3, and 7 in-between frames. The shutter time for all images is 9.

the bias versus variance trade-off provided by time dependent photon mapping is too poor to produce palpable caustics. For this reason a much higher number photons have been used to render the images in the three leftmost columns. Of these, the top row is based on a photon map containing as much as 480 000 photons while the bottom row is based on a photon map contain 40 000 photons. From left to right the temporal resolution increases from 1 to 3 to 7 in-between frames. The spatial bandwidth for the renderings was chosen as to decrease noise to an acceptable level. This leads to the perhaps most important observation, namely that a low temporal resolution produces visible bands that can only be removed by filtering beyond what removes normal noise. This complicates matter, as an increase of photons no longer is a guarantee for high quality illumination.

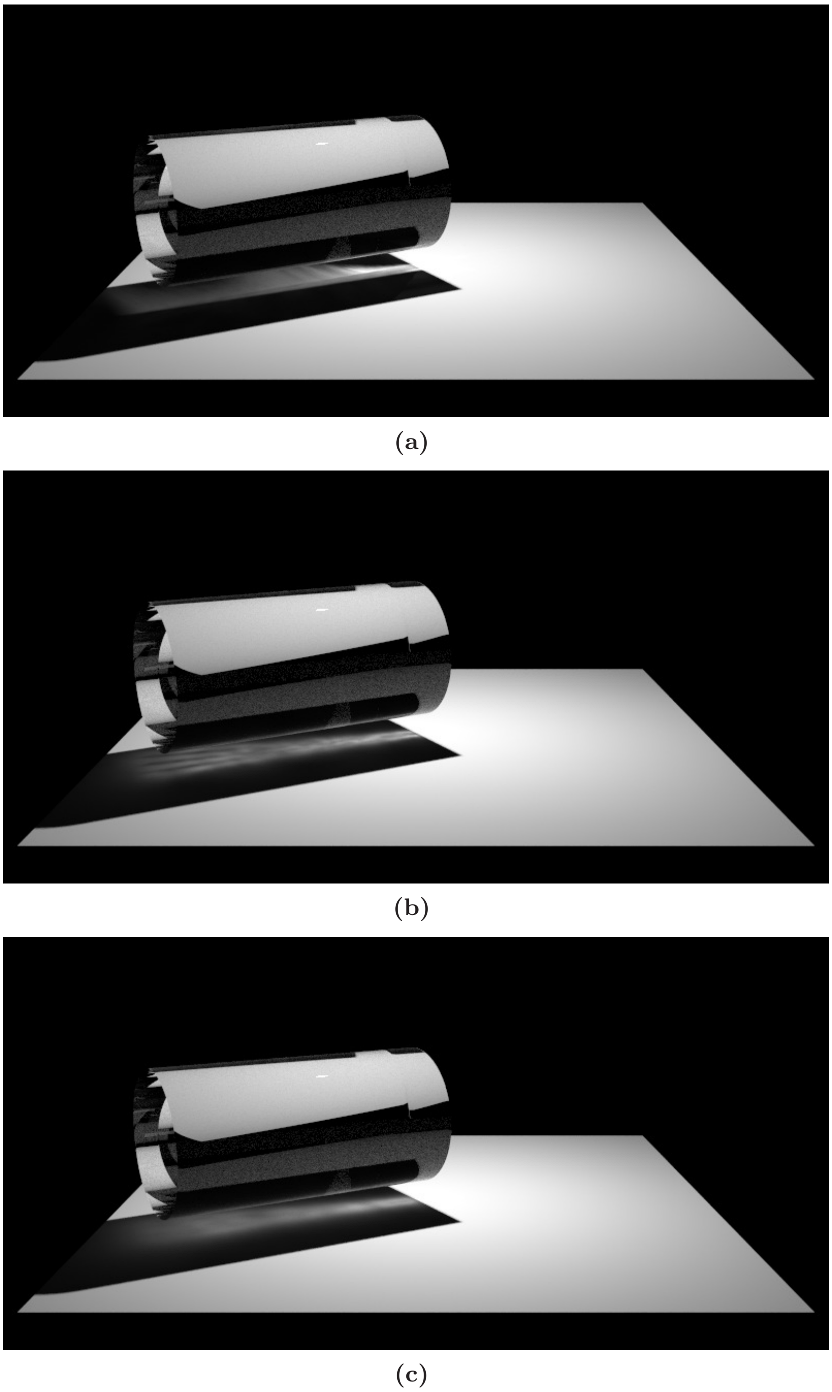
Photon differentials are free of this concern as the blurring is based on the first order derivatives of object movement and not on finite animation steps. In the implementation presented here one additional frame is need in order to estimate the derivatives.

The images in Figure 7.2 present a more complex–all though very artificial–scene in which a cylinder is rotating counterclockwise around its one end. This rotation means that the speed of the cylinder will increase as we move from the turning point down the length of the cylinder. As a result the produced caustic becomes more blurred when refracted from the high speed end of the cylinder. Figure 7.6(a) gives the solution provided by temporal photon differentials while the images in Figure 7.6(b) and 7.6(c) was produced with time dependent photon mapping. All images were rendered with the same number of photons contained in the photon map. However, the two latter images was rendered with different bandwidths. From these two images we see that at this obviously low temporal resolution an increase in bandwidth can help remove the temporal bands that time dependent photon mapping is prone to. The price, however, is an unwanted blurring of the front of the caustic.

Finally, Figure 7.7 solely depicts the photons' time differentials as they are projected down on the plane beneath the cylinder. A high exposure time has been used as to facilitate the illustration. The image confirms that the time differential vectors become longer when refracted from the high speed end of the cylinder, thus elongating the time dependent kernel used in the temporal radiance estimate.

7.3 Summary

In this chapter we have presented a global illumination method for rendering animated scenes. Our method elegantly handles time filtering such that frames can be rendered on a one to one basis. In contrast to similar dynamic scene renderer, our method do not need in-between frames in order to avoid temporal aliasing.



(a)

(b)

(c)

Figure 7.6: Rendering of a rotating cylinder. All images was render with a shutter time of 4. Figure 7.6(a) was rendered using temporal photon differentials. Figure 7.6(b) and 7.6(c) was render using time dependent photon mapping the former using 100 photons per radiance estimate and the latter using 250 photons in the radiance estimate.

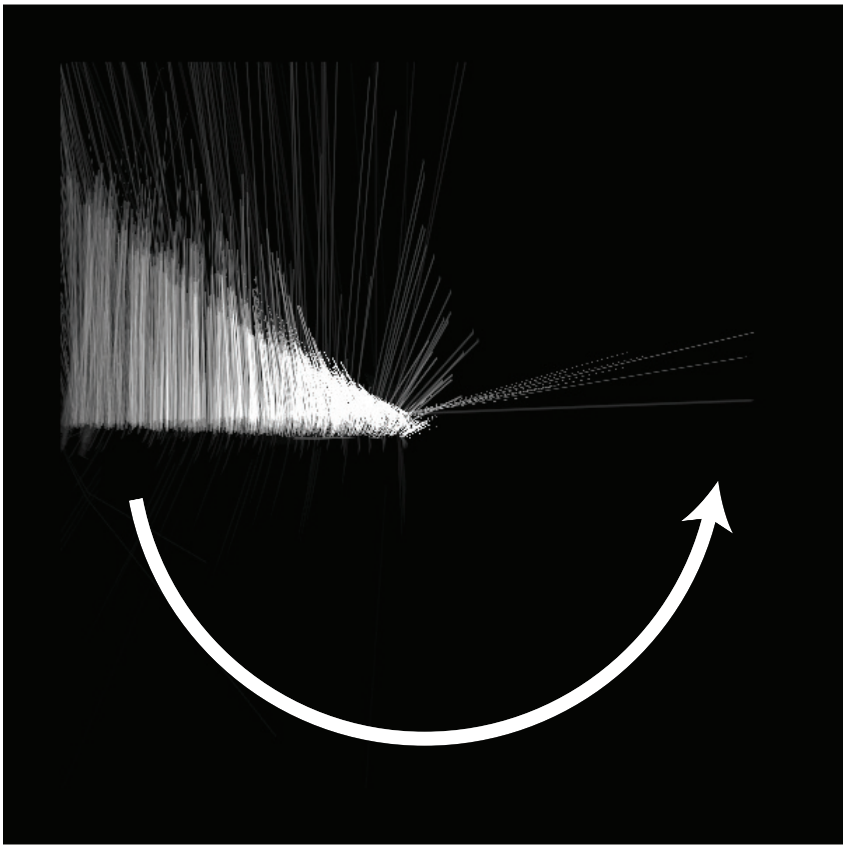


Figure 7.7: Projected time differentials from a rotating glass cylinder.

8 Photon Differentials for participating media

In this chapter a novel approach for rendering translucent media such as smoke, fire, and water is presented. The approach is based on photon differentials discussed in Chapter 5 and published in [Schjøth et al. 2007]. Similar to photon differentials, our approach employs a modified form of ray differentials (introduced by Igehy [1999] and extended by Sporring *et al.* [2009] (See Appendix B) in order to decide the shape and size of anisotropic kernels used in the reconstruction of indirect illumination.

The main additions are that we employ an extra set of heuristics in order to handle photon differentials exposed to diffuse reflection and scattering inside media, and that we employ three dimensional anisotropic kernels in the density estimate.

Furthermore, when reconstructing indirect illumination, we estimate the exact integral of the kernel contribution as opposed to the numerical approximation used by other methods. This gives a higher precision when estimating the illumination.

Media for which the interaction of visible light does not primarily occur on the surface–but also heavily inside–are computationally expensive to simulate; not only is it necessary to account for every possible light interaction on the surface, it is also necessary to account for every possible interaction at every point inside the medium. In order to simulate lights propagation through such media the common approach in graphics is to solve to the *radiative transfer equation* (RTE).

8.1 Radiative transfer equation

The radiative transfer equation describes light transport in a medium which absorbs, emits, and scatters light [Chandrasekhar 1950]. As is customary in graphics, we will refer to such a medium as a *participating medium*. For a medium which does not emit light, the integral form of the RTE is

$$L(\boldsymbol{x}, \boldsymbol{\omega}) = T_r(\boldsymbol{x}, \boldsymbol{x}_s) L(\boldsymbol{x}_s, \boldsymbol{\omega}) \tag{8.1a}$$

$$+ \int_0^s T_r(\boldsymbol{x}, \boldsymbol{x}_t) \sigma_s(\boldsymbol{x}_t) \int_{\Omega_{4\pi}} p(\boldsymbol{x}_t, \boldsymbol{\omega}, \boldsymbol{\omega}_t) L(\boldsymbol{x}_t, \boldsymbol{\omega}_t) \, \mathrm{d}\boldsymbol{\omega}_t \, \mathrm{d}t. \tag{8.1b}$$

In this form, the RTE integrates over a line segment, 0 to s, and over all incoming directions, $\Omega_{4\pi}$. The equation involves the scattering properties of the material: the *extinction coefficient* σ_t, which describes the attenuation of light per unit distance traveled through the material; the *scattering coefficient* σ_s, which describes the amount of light that is scattered per unit distance; and the *phase function* p, which describes the amount of light that is scattered from one direction to another. The first term is called the *direct transmission* term and the second term is called the *in-scattering* term (denoted L_s in the following). In the equation, T_r is the transmittance between two point defined as

$$T_r(\boldsymbol{x}, \boldsymbol{x}') = e^{\int_{\boldsymbol{x}'}^{\boldsymbol{x}} \sigma_t(x) d\boldsymbol{x}}. \tag{8.2}$$

The diagram in Figure 8.1 illustrates how radiance arrives at the eye from different sources.

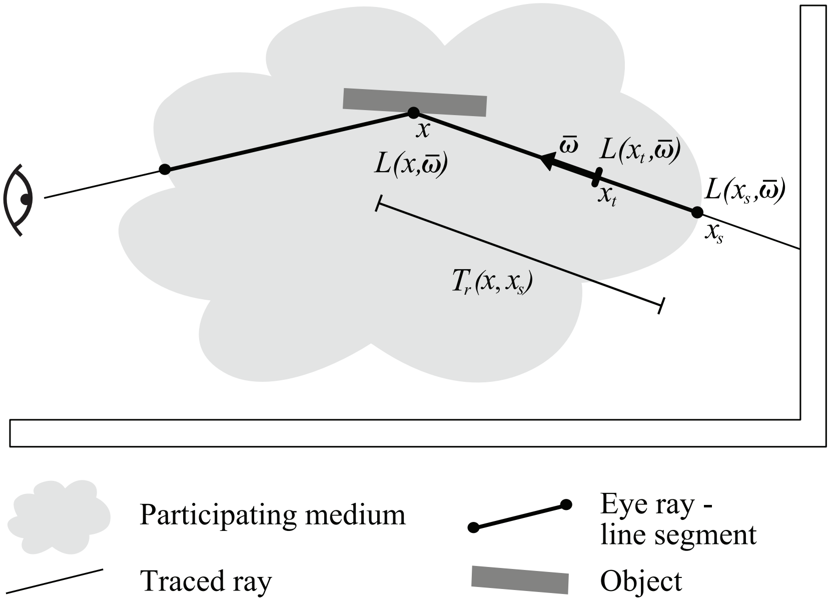


Figure 8.1: Radiance arriving at the point $\boldsymbol{x}$ from the direction $\boldsymbol{\omega}$, consists of radiance coming from direct transmission and in-scattering radiance having suffered single or multiple scattering events inside the participating medium.

Typically, it is only feasible to find a solution to the radiative transfer equation by numerical integration. A common approach, called ray marching, is to take small steps along the ray; at each step estimating the incoming radiance, L_s. This approach assumes that incoming radiance and material properties are constant over segments between the steps. The difficulty in solving the RTE stems from the in-scattering term, which is recursive. The integral in this term accumulates in-scattered radiance that must be found by again solving the RTE.

8.2 Related work

Two common approaches that attempt to solve RTE are *path tracing* [Pattanaik and Mudur 1993] and *volumetric photon mapping* [Jensen and Christensen 1998].

Path tracing is a rendering technique based purely on Monte Carlo integration. For each step taken along a ray, the in-scattered radiance is found by stochastically tracing rays from $\boldsymbol{x}_t$, thus sampling the integral of the in-scattering term. The problem with path tracing is that estimates contain variance. Even with the use of variance reduction techniques, such as importance sampling, a huge number of samples are required to eliminate all visible noise in an image. The advantage is that results are unbiased and converge (albeit slowly) to the exact solution.

Photon mapping is a two pass method, which is much faster than path tracing, but produces biased results. When used to render participating media, the process is as follows: in the first pass photons are stochastically scattered around in the medium storing the position and radiant power of a photon where the photon suffers a scattering event; in the second pass rays are traced from the eye. Where these *eye rays* pass through a participating medium, ray marching is employed gathering up the nearest photons at each step so as to approximate incoming radiance by density estimation.

Gathering photons at intervals along the ray can be problematic; the procedure introduces a user-defined parameter which controls a trade-off between computation time and estimation accuracy. As Jarosz *et al.* [2008] problematizes, taking too large steps along the ray may result in missed photons during the gathering process. On the other hand taking too small steps means that photons might be counted more than once. In both cases estimates are suboptimal.

Jarosz *et al.* avoid this multiple query problem by making a **single** gathering of photons per eye ray in a *beam radiance estimate*. In order to achieve this, each photon is associated with a kernel bandwidth that defines the extent to which a photon's energy is spread. In the context of density estimation Jarosz *et al.* employ a variable kernel density estimator (see Section 2.2) to estimate the radiance arriving along the eye ray. In order to find individual kernel bandwidths Jarosz *et al.* discuss to possible approaches, one is to simple use a fixed bandwidth for each kernel (this in fact reduces estimator to a common kernel density estimator), and the other is to use a pilot estimate. In the pilot estimate, the bandwidth for each photon is found as the distance to the photon's k'th nearest neighbor.

We avoid the multiple query problem using the same approach as Jarosz *et al.*; gathering photons along the eye ray in such a way that duplicates are avoided.

However, unlike their method our method need not perform a potentially expensive pilot estimate in order to find individual photon bandwidths. We derive photon bandwidths by keeping track of the photon's differentials as it is reflected and scattered around the virtual scene and through translucent media. When a photon is scattered or suffers a diffuse reflection, we handle the spread of the photon's differentials using a heuristic similar to Herzog *et al.*'s [2007b] *ray splatting* method. This means that the bandwidth of a scattered photon in our method is proportional to the probabilistic path of the photon.

Furthermore, while Jarosz *et al.* use an isotropic variable kernel estimator in their beam radiance estimate, we use an anisotropic variable kernel density estimate. This approach will improve the trade-off between bias and variance for our method, as we will be able to smooth along prominent structures and not over.

8.3 Photon splatting for participating media

The goal of the following is to evaluate the RTE from the contributions of each individual photon scattering. As with photon mapping, our method stochastically emits photons from light sources; probabilistically tracing them around in the scene and inside participating media. Photons suffering a scattering event inside a media is stored. In the second pass this information is used to reconstruct in-scattered radiance. Radiance from an eye ray that passes through a participating media, is estimated by gathering photons in its proximity. Each collected photon individually contributes to the radiance based on extent of the scattering event, which again is found from the estimated differential of the photon.

Radiance is defined in terms of radiant power Φ that flows through an element of area $\mathrm{d}A$. The energy flows in a directional volume described by an element of solid angle $\mathrm{d}\boldsymbol{\omega}$, and the radiance is the part of the energy flow, which projects to the area $\mathrm{d}A$. Mathematically,

$$L = \frac{\mathrm{d}^2\Phi}{\mathrm{d}A_\perp \, \mathrm{d}\boldsymbol{\omega}} = \frac{\mathrm{d}^2\Phi}{\cos\theta \, \mathrm{d}A \, \mathrm{d}\boldsymbol{\omega}},$$

where $\mathrm{d}A_\perp$ is projected area and θ is the angle between the surface normal of the area $\mathrm{d}A$ and the direction $\boldsymbol{\omega}$ of the solid angle $\mathrm{d}\boldsymbol{\omega}$. This definition works well when light scatters from surface to surface, but when light scatters in a volume there is no surface normal to describe the projected area. Instead, we use the total scattering cross section of the particles that scatter light in an element of volume $\mathrm{d}V$. This cross section is [Siegel and Howell 2002]

$$\mathrm{d}A_s = \sigma_s \, \mathrm{d}V.$$

When inserted in place of the projected area in the definition of radiance ($\mathrm{d}A_\perp = \mathrm{d}A_s$), it provides a way to describe the scattered radiance in a volume. Using this description of scattered radiance in a volume, the in-scattering term of the RTE becomes [Jensen and Christensen 1998]

$$\begin{aligned} L_s(\boldsymbol{x},\boldsymbol{\omega}) &= \int_0^s T_r(\boldsymbol{x},\boldsymbol{x}_t)\sigma_s(\boldsymbol{x}_t)\int_{4\pi} p(\boldsymbol{x}_t,\boldsymbol{\omega},\boldsymbol{\omega}_t)\frac{\mathrm{d}^2\Phi}{\sigma_s(\boldsymbol{x}_t)\,\mathrm{d}V\,\mathrm{d}\boldsymbol{\omega}_t}\,\mathrm{d}\boldsymbol{\omega}_t\,\mathrm{d}t \\ &= \int_0^s T_r(\boldsymbol{x},\boldsymbol{x}_t)\int_{4\pi} p(\boldsymbol{x}_t,\boldsymbol{\omega},\boldsymbol{\omega}_t)\frac{\mathrm{d}^2\Phi}{\mathrm{d}V}\,\mathrm{d}t. \end{aligned}$$

From this formulation Jensen and Christensen [1998] solves the RTE by numerical integration:

$$\begin{aligned} L(\boldsymbol{x},\boldsymbol{\omega}) &\approx T_r(\boldsymbol{x},\boldsymbol{x}_s)L(\boldsymbol{x}_s,\boldsymbol{\omega}) \\ &\quad + \sum_{i=1}^{s} T_r(\boldsymbol{x},\boldsymbol{x}_i)\left(\sum_{p=1}^{k} p(\boldsymbol{x}_i,\boldsymbol{\omega},\boldsymbol{\omega}_p)\frac{\Delta\Phi_p}{V_i}\right)\Delta_i, \end{aligned} \qquad \boxed{8.3}$$

where the outermost Riemann sum is an approximate integration of the in-scattered radiance along the eye ray, such that s is the number of segments, Δ_i, with corresponding sampling points $\boldsymbol{x}_i$; and the innermost Riemann sum is a k nearest neighbors density estimate of the in-scattered radiance at each sampling point. In the local in-scattered radiance estimate V_i is the volume (of the kernel) the sphere encompassing the nearest neighbors, such that $V_i = 4\pi r_i^3/3$, with r_i being the distance to the k'th nearest neighbor of the sampling point, x_i.

In contrast to the solution provided by Jensen and Christensen our solution does not estimate radiance as an average over a segment at determined sample points, instead we estimate the exact contribution from each photon based on its overlap with the eye ray. We let photons contribute to the radiance reaching the eye, by distributing a photon's radiant power to the eye ray. Radiant power, contributing to an eye ray, is attenuated by the medium in-between the eye and the point on the eye ray closest to the scattering event and is then summed up. The contribution from a scattering event to an eye ray depends on the proximity of the scattering event and the extent to which the radiant power of the scattering event is spread. Our numerical solution is

$$L(\boldsymbol{x},\boldsymbol{\omega}) \approx T_r(\boldsymbol{x},\boldsymbol{x}_s)L(\boldsymbol{x}_s,\boldsymbol{\omega}) + \sum_{i=1}^{n} T_r(\boldsymbol{x},\boldsymbol{x}_i)p(\boldsymbol{x}_i,\boldsymbol{\omega}_i,\boldsymbol{\omega})\frac{\Delta\Phi_i}{V_i}w_i, \qquad \boxed{8.4}$$

where n is the number of scattering events contributing to the eye ray, $\Delta\Phi_i$ is the radiant power of scattering event i, V_i is the volume to which the radiant power is spread, and $\boldsymbol{x}_i$ is the point on the eye ray with the shortest distance to center of the scattering event. In the above equation,

$$w_i = 2\int^{\frac{1}{2}\Delta t} K(y(t))dt \qquad \boxed{8.5}$$

is the integrated kernel weight over the part of the line segment, Δt, overlapped by the scattering event (see Figure 8.2), where $K(y)$ is a radially symmetric unimodal kernel function, and

$$y(t) = \sqrt{t^2 + y_s^2} \qquad \boxed{8.6}$$

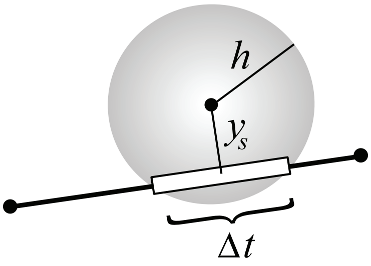


Figure 8.2: The amount of radiant power distributed to an eye ray from a scattering event depends on the extent of the event, h, the shortest distance between the event center and the eye ray, y_s, and energy carried by the scattering photon.

is the distance from the kernel center to the line segment. In the equation, y_s, is the shortest distance between the line segment and the kernel center.

As in Chapter 5, the kernel functions in this chapter, deviate from those common to classical statistics insofar that they do not integrate to unity. The reason is that the kernel is divided by the volume, V, outside the kernel function as seen in Equation 8.4.

As the kernel function is given with respect to the distance, y, from the kernel center and not the position on the line segment t, a change of variable helps reduce the problem. We change variables such that we integrate over y instead of t.

From Pythagoras' theorem we have that

$$t = \sqrt{y^2 - y_s^2} \tag{8.7}$$

and, taking the derivative with respect to y, we get

$$dt = \frac{y}{\sqrt{y^2 - y_s^2}} dy \tag{8.8}$$

Substituting this in to Equation 8.5 we get a change of variables:

$$w_i = 2 \int_{y_s}^{h} K(y) \frac{y}{\sqrt{y^2 - y_s^2}} dy. \tag{8.9}$$

In the most simplistic case we use a simple uniform kernel function such that

$$K(y) = \begin{cases} 1 & \text{if } y < h, \\ 0 & \text{otherwise.} \end{cases} \tag{8.10}$$

Solving Equation 8.9 with the uniform kernel function we simply get that

$$w_i = 2\sqrt{h^2 - y_s^2}. \tag{8.11}$$

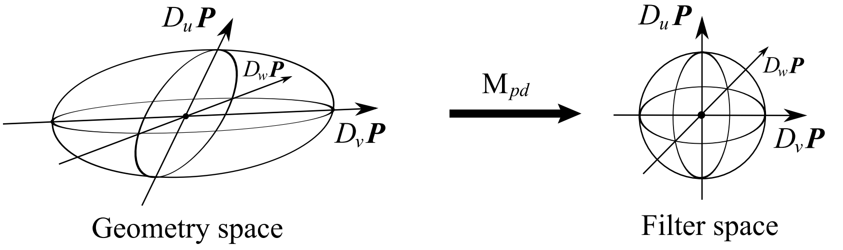


Figure 8.3: Transformation from geometry space to filter space by the matrix M_{pd}. The ellipsoid on the left is the footprint of the photon differential. When transformed into filter space the ellipsoid becomes a unit sphere.

Using a more advanced kernel, the equation becomes somewhat more involved.

A useful and inexpensive kernel is the Epanechnikov kernel [Silverman 1986]. In three dimensions, the Epanechnikov kernel is

$$K(y) = \begin{cases} \frac{5}{2}\left(1 - \left(\frac{y}{h}\right)^2\right) & \text{if } y < h, \\ 0 & \text{otherwise.} \end{cases} \tag{8.12}$$

Using Equation 8.9 together with the Epanechnikov kernel function we integrate along that part of the line segment which is encompassed by the sphere surrounding the scattering event. Thus, we get that

$$w_i = \frac{10}{3h^2}\sqrt{(h^2 - y^2)^3}. \tag{8.13}$$

Extending the radiance estimate to include photon differentials we calculate the integrated kernel weight, w_i, using an anisotropic kernel shaped by the photon differential. As illustrated in Figure 8.3, the photon differentials define a coordinate system in which the differential vectors are basis vectors. In this filter space we estimate the shortest distance between the scattering event and the eye ray as

$$y_{pd}^2 = (\boldsymbol{x} - \boldsymbol{x}_i)^T \boldsymbol{M}_{pd}^T \boldsymbol{M}_{pd}(\boldsymbol{x} - \boldsymbol{x}_i), \tag{8.14}$$

where the matrix $\boldsymbol{M}_{pd}$ transforms to filter space.

This modified distance is then used to calculate the integrated kernel weight. Taking into account that kernel radius in filter space is one, the weight for the anisotropic Epanechnikov kernel is

$$w_i = \frac{10}{3|\boldsymbol{M}_{pd}\boldsymbol{\omega}|}(1 - (\boldsymbol{x} - \boldsymbol{x}_i)^T \boldsymbol{M}_{pd}^T \boldsymbol{M}_{pd}(\boldsymbol{x} - \boldsymbol{x}_i))^{3/2}, \tag{8.15}$$

where $\boldsymbol{\omega}$ is the normalized direction of the eye ray. Additionally, the volume, V_i, has to be estimated differently as the kernel is no longer isotropic. Instead of estimating it as the volume of a sphere it is estimated as an skewed ellipsoid,

$$V_i = \frac{1}{6}\pi|(D_u\boldsymbol{x} \times D_v\boldsymbol{x}) \cdot D_w\boldsymbol{x}|, \tag{8.16}$$

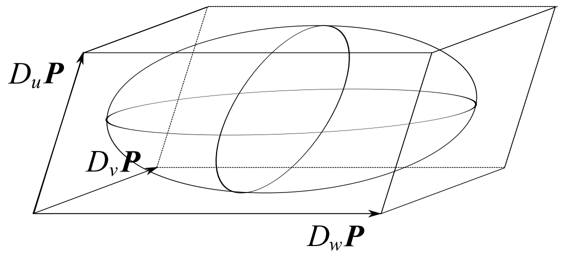


Figure 8.4: The volume of the kernel defined by the differentials vectors.

such that the ellipsoid is fitted into a parallelepiped in which the differential vectors, $D_u\boldsymbol{x}$, $D_v\boldsymbol{x}$ and $D_w\boldsymbol{x}$ define the edges (See Figure 8.3).

With the modified volume estimation from Equation 8.16 and the integrated weight from Equation 8.15, Equation 8.4 now gives a solution to the RTE that is based on photon differentials. It remains to determine the differential vectors.

8.4 Photon differentials in volume rendering

In the context of classical statistics Jensen and Christensen use a k'th nearest neighbor estimator in their solution. In contrast, both Jarosz *et al.*'s and our method employ the variable bandwidth estimator. However, instead of making a pilot estimate to decide the bandwidth of each sampling point as is done in Jarosz *et al.*'s method, we base the bandwidth on the ray path of the photon.

Photon differentials employ ray differentials in order to keep track of the spread of beams of 'photons' as they are traced trough a scene. Chapter 5 described how this concept can be used to estimate radiance reflected from surfaces such that prominent illumination features are preserved.

In the following, we expand the theory so that photon differentials can be used in conjunction with participating media. In order to achieve this, we describe the behaviour of photon differentials inside participating media and, furthermore, how photon differentials can be reflected on diffuse surfaces. However, in a number of cases, our photon differentials behave like 'ordinary' photon differentials. We emit photon differentials in the same manner as described in Chapter 5, and we reflect, refract and transfer photon differentials as derived by Sporring *et al.* [2009] (See Appendix B). Similarly, we use the same notation such that the directional derivatives of a photon with a position $\boldsymbol{x}$ and a direction $\boldsymbol{\omega}$ is described as

$$D\boldsymbol{\omega} = \begin{bmatrix} D_u\boldsymbol{\omega} & D_v\boldsymbol{\omega} & D_w\boldsymbol{\omega} \end{bmatrix}, \qquad \boxed{8.17}$$

where $D\boldsymbol{\omega}$ is the Jacobian of $\boldsymbol{\omega}$ and $D_u\boldsymbol{\omega}$, $D_v\boldsymbol{\omega}$ and $D_w\boldsymbol{\omega}$ are column vectors that describe the directional derivatives with respect to the scalar variables u,

v, and w, these we call the directional differential vectors. Likewise, we describe the positional derivatives of a photon by

$$D\boldsymbol{x} = \begin{bmatrix} D_u\boldsymbol{x} & D_v\boldsymbol{x} & D_w\boldsymbol{x} \end{bmatrix}, \qquad \boxed{8.18}$$

where $D\boldsymbol{x}$ is the Jacobian of $\boldsymbol{x}$ and $D_u\boldsymbol{x}$, $D_v\boldsymbol{x}$ and $D_w\boldsymbol{x}$ are positional differential vectors.

As we trace our photon differentials around the scene, we only keep track of differential vectors with offset in u and v, because reflections and refractions occur after transfer onto tangent surface planes. Differential vectors with offset in w are only valid inside participating media. Therefore, these are only stored in connection with these. When storing a photon in connection with a participating medium the differential vector $D_w\boldsymbol{x}$ is given the same direction as that of the photon at scattering time. Its magnitude is set to be the same as either $D_u\boldsymbol{x}$'s or $D_v\boldsymbol{x}$'s whichever has the greatest. In this way the kernel shape will always be in a state between a sphere and a oblate spheroid.

As with conventional photon mapping, we employ importance sampling to trace photons around the scene and inside participating media.

8.4.1 Scattering

As a photon moves inside a participating medium it can either exit the medium, be scattered, or be absorbed. A photon suffering a scattering event inside a participating media after having moved a distance s attains a position and direction given by

$$\boldsymbol{x}' = \boldsymbol{x} + s\boldsymbol{\omega}, \qquad \boxed{8.19a}$$

$$\boldsymbol{\omega}' = \boldsymbol{\omega}. \qquad \boxed{8.19b}$$

Using Monte Carlo integration, the distance a photon moves before a scattering event occurs can be estimated as [Siegel and Howell 2002]

$$s = -\frac{\ln(\xi)}{\sigma_t}, \qquad \boxed{8.20}$$

where $\xi \in]0,1]$ is a random variable for the interaction. If we assumes that $ds \to 0$, the photon differentials approach

$$d\boldsymbol{x}' = d\boldsymbol{x} + s\,d\boldsymbol{\omega}, \qquad \boxed{8.21}$$

$$d\boldsymbol{\omega}' = d\boldsymbol{\omega}. \qquad \boxed{8.22}$$

When a photon is scattered, it is designated a new direction from importance sampling of the phase function. With photon differentials we handle this with a simple heuristic. First we estimate the new direction $\boldsymbol{\omega}'$, and then we find quaternion that represents the rotation from our original direction $\boldsymbol{\omega}$ to the new direction.

This quaternion is used to rotate the positional and directional differential vectors, $d\boldsymbol{x}$ and $d\boldsymbol{\omega}$, to the new direction $\boldsymbol{\omega}'$.

Furthermore, we scale the area of the differential foot print by the following heuristic

$$k = \frac{1}{4\pi p}, \tag{8.23}$$

where p is the probability that a photon with the direction $\boldsymbol{\omega}$ will assume the direction $\boldsymbol{\omega}'$ when scattered. In the equation, k is constructed such that, given a number of photon differentials, with of equally sized foot prints, whose area in total is equal to the surface area of a unit sphere, the scaling will preserve the total area independent of the phase function used (for large datasets).

The new two dimensional area of the footprint would be estimated as

$$A' = \frac{1}{4}\pi k |D_u\boldsymbol{x} \times D_v\boldsymbol{x}|, \tag{8.24}$$

which equals multiplying the differential vectors by $\sqrt{k}$. The reasoning behind this heuristic is the same as that of Christensen *et al.* [2003]: directions in which many rays are sent have a high ray coherence, in contrast, directions which are sparsely sampled have a low coherence. Translated to photon differentials; differentials sent in directions which are heavily sampled should become more narrow, and, likewise, differentials sent in sparsely sampled directions become more broad. We achieve this by scaling the differentials by $\sqrt{k}$.

8.4.2 Diffuse reflection

Diffuse reflections are handled in much the same way as with scatterings. As is typical in graphics, a diffuse surface is importance sampled proportional to the cosine-weighted solid angle over a unit hemisphere. For a photon differential that has an already defined footprint area we use a heuristic similar to that used for scattering. The new area is found by

$$A' = Ak \tag{8.25}$$

where

$$k = \frac{1}{2\pi p}, \tag{8.26}$$

Again, p is the probability that a photon with the direction $\boldsymbol{\omega}$ will assume the direction $\boldsymbol{\omega}'$ when reflected. As was the case for scattering, we rotate the differential vectors into the new sampled direction, $\boldsymbol{\omega}'$, and multiply them by $\sqrt{k}$.

This simple heuristic is not limited to diffuse surfaces; it can in fact be used with any brdf that can be importance sampled. The disadvantage is that our heuristic does not contributes to the anisotropy of the kernel as it ignores surface curvature and therefore scales the differential vectors uniformly.

8.5 Results

To validate the proposed method we use three test scenes. One scene without volume caustics and two with volume caustics. To compare our method, we have made three renderings of each scene. A reference rendering with a huge number of photons and two renderings with a low number of photons. The two low resolution renderings are rendered with our method and with conventional volumetric photon mapping. For all renderings the optimal bandwidth have been found by a human observer. Observe, that we are not evaluating the full method insofar that the our integration scheme, Section 8.3, is not implemented in the compared method.

The first scene contains heterogeneous smoke illuminated by a large area light source, Figure 8.5. In the scene a faceted sphere is surrounded by warm smoke rising from the floor. This scene contains no prominent volume caustics. It does, however, demonstrate our methods capability in handling heterogeneous media in diffuse environments. The reference image was rendered using 3 000 000 photons while the test images was rendered using a photon map containing 80 000 photons. Comparing the references image, Figure 8.5(a), with the image rendered using conventional volumetric photon mapping, Figure 8.5(b) we see that even though the reference image contain finer details the advantage in quality gained by using huge photon map is slight. However, comparing the reference image with image rendered with our method (Figure 8.5(c)) an interesting problem is exposed. Namely, that our method has loss of illumination intensity in light scattered from the smoke. The explanation for this loss boundary bias. As with ordinary photon differentials, the virtual beam is only considered as a single ray. This means that a differential footprint might very well overlap a polygonal border, resulting in a loss of energy near borders (See Section 3.1). Volumetric photon mapping, which is based on the k'th nearest density estimate, compensates partially for boundary bias in the local density estimate; the kernel size decreases near borders as the local density i diminished. In contrast photon differentials offer no compensation and the method is therefore especially susceptible to heterogeneous smoke that often have a large surface area for the three dimensional footprints to overlap.

The second scene also contains heterogeneous smoke, see Figure 8.5. In this scene a narrow beam of laser light is emitted from the ceiling of a Cornell box. From there it penetrates a volume of heterogeneous smoke before it hits a faceted silver hemisphere on the floor. The hemisphere reflects the incoming light in a number of different directions. As in the former case, the reference image was rendered using 3 000 000 photons while test images was rendered with a map of 80 000 photons. Comparing against the reference image, Figure 8.5(a), it is obviously that volumetric photon mapping, Figure 8.5(b), is not able to reproduce the finer details of the volume caustics. Only two of the three reflected ray are visible and that only partially. The reason for this is that smoothing has all but removed them. In comparison we see from the Figure 8.5(c) that photon differentials exhibit a superior variance-bias trade-off reproducing all of the five reflected volume caustics distinctly.

The last image set, Figure 8.5, is that of a sphere illuminated by a small area light source within a volume of homogeneous smoke. The refraction of light through the sphere creates a volume caustic. Here the test images was rendered using only 10 000 photons while the reference image was rendered with the usual 3 000 000 photons. From the images we see that the volume caustic created by our method is more distinct that the one created by volumetric photon mapping and a photon map containing the same number of photons. The image contain some noise an the method has not reproduced the narrow high intensity beam in the middle of the caustic visible in the reference image.

8.6 Summary

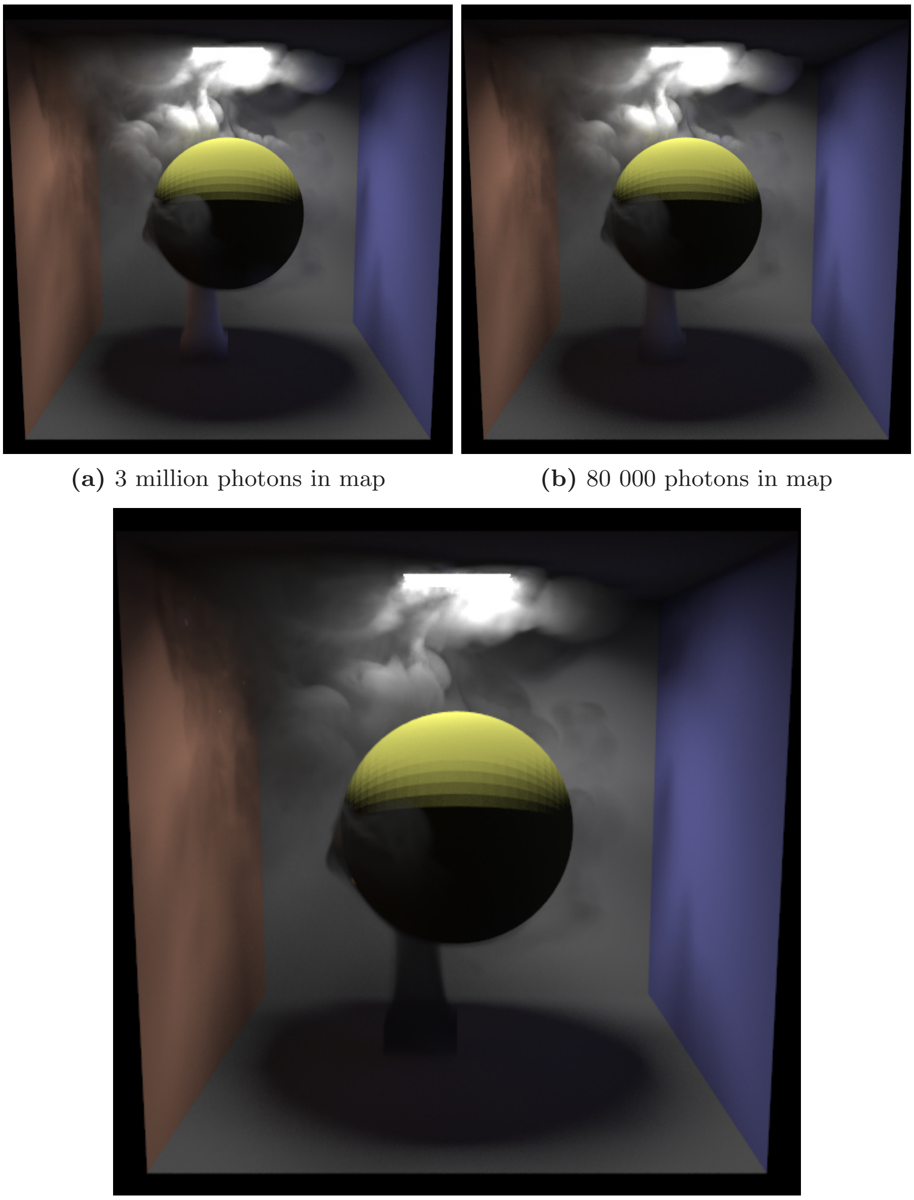
In this chapter we have introduced a method for rendering participating media. The method extends on photon differentials as it keeps track of a photon differentials as they are scattered through translucent materials. This procedure allows us to shape rotational invariant kernels which can be used to estimate reflected radiance as a variable kernel density estimate. We, therefore, achieve a superior trade-off between variance and bias, yielding a more detailed reproduction of volume caustics. However, one disadvantage of the method is that it is susceptible to boundary bias.

8.7 Future work

In the presented work we have integrated our method with the usual two-pass approach: in a first pass photons are emitted, traced, and stored; and in the second pass in photons are collected along eye rays, and the contribution of each photon is estimated.

We suggest to reverse the process. In order to do this we would employ a Voronoi diagram for 3D line segments. The Voronoi diagram would be used to store those eye rays that move through a participating medium. Then, instead of gathering scattering events near rays from the eye, we would distribute energy from scattering events to nearby eye rays. The radius in which light would be distributed from a scattering event, and the radiance contribution to the eye ray from such an event, would be estimated as described in this Chapter.

The advantage of this approach is that not only would we avoid the multiple query problem described in Section 8.2, also we would be able to avoid storing a potential huge amount of photons. The Voronoi diagram is complicated to built, but k nearest neighbor queries can be made in logarithmic time.

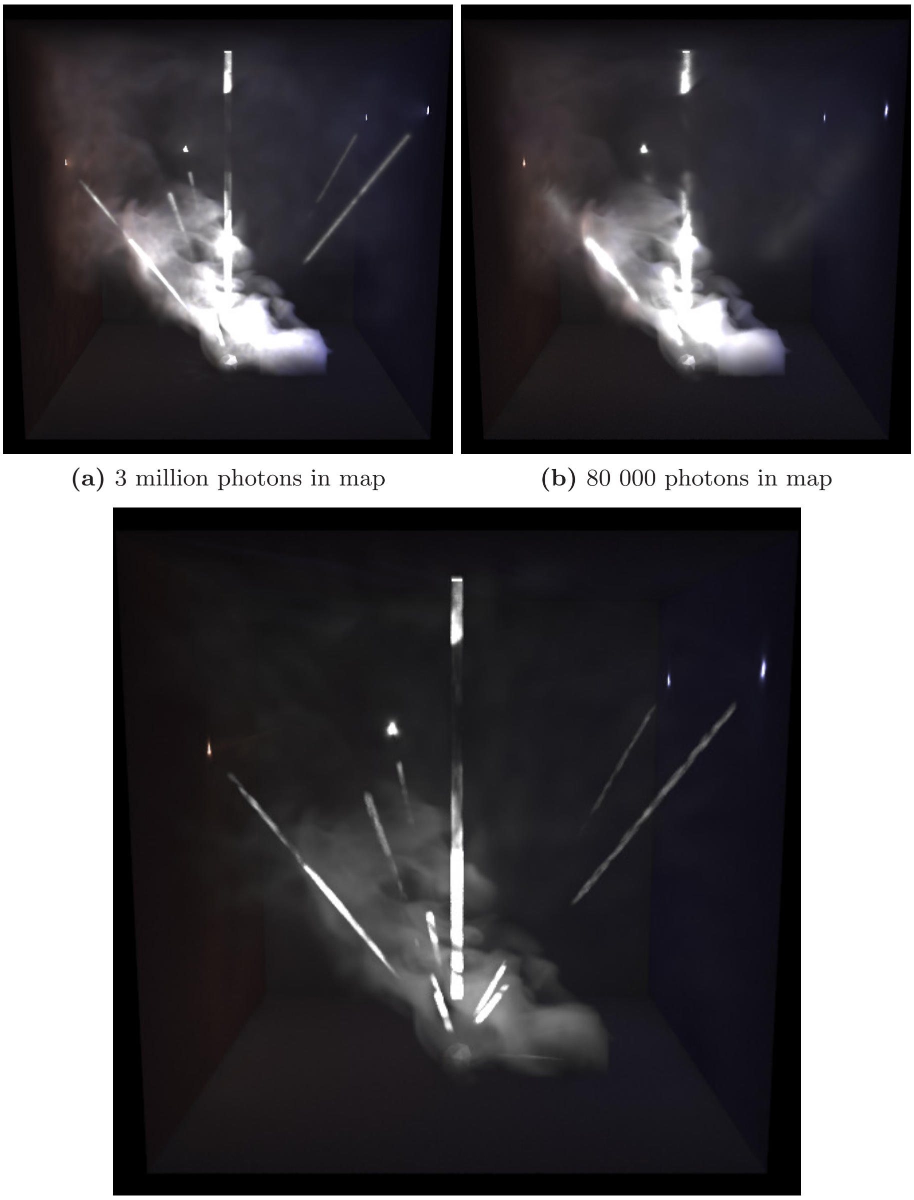


(a) 3 million photons in map

(b) 80 000 photons in map

(c) 80 000 photons in map

Figure 8.5: Renderings of a Cornell box containing heterogeneous smoke. Warm smoke diffuses from the and into the box. It surrounds a faceted sphere levitating in midair. Images (a) and (b) was rendered using conventional photon mapping

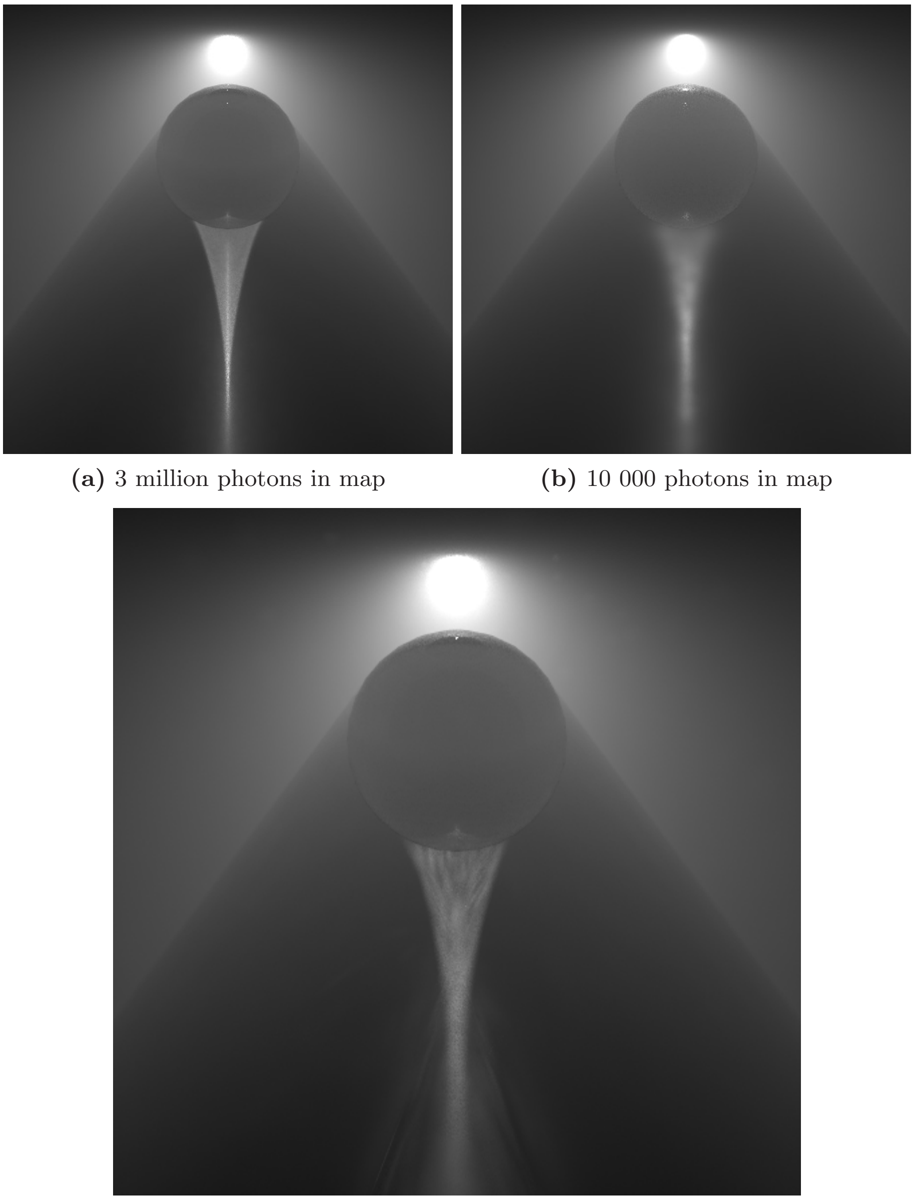


(a) 3 million photons in map

(b) 80 000 photons in map

(c) 80 000 photons in map

Figure 8.6: Renderings of a Cornell box containing heterogeneous smoke. A narrow beam of collimated light shoots from the ceiling traveling through heterogeneous smoke before hitting a faceted silver hemisphere. Images (a) and (b) was rendered using conventional photon mapping



(a) 3 million photons in map

(b) 10 000 photons in map

(c) 10 000 photons in map

Figure 8.7: Renderings of a glass sphere illuminated from a small area light source. Both light source and glass sphere is inside homogeneous smoke. Images (a) and (b) was rendered using conventional photon mapping

9 Conclusion

The theme of this thesis has been anisotropic kernel density estimation. We have introduced four methods that make use of this technique.

We first took it in use developing diffusion based photon mapping. This method employs an anisotropic local-bandwidth kernel density estimator to reconstruct indirect illumination. For each surface estimate point a rotational invariant kernel is shaped based on the first order structure of the illumination. We found that diffusion based photon mapping was superior to conventional photon mapping when reconstruction high contrast illumination.

However, as we saw in Chapter 2 an estimator exists that can be more precise than the local bandwidth estimator. Photon differentials use an anisotropic variable bandwidth estimator to reproduce indirect illumination. This estimator employs an individual kernel for each sampling point. These individual kernels are shaped by differentiating the path traveled by the photon.

Chapter 6 made it apparent that photon differentials can outperform both conventional photon mapping as well as diffusion based photon mapping. Furthermore, this chapter made use of two different objective image quality measures, namely MISE and SSIM. These were used to balance the trade-off between bias and variance. From the images produced using these methods, it was rendered probably that of these two measures, SSIM would choose the balance best suited for illumination reconstruction to a human audience.

In Chapter 7 photon differentials was extended such that anisotropic filtering was perform both in the spatial as well as the temporal domain. This extension meant that precise motion blur could be obtained at a very low temporal resolution. This should be seen on contrast to conventional methods that usually need a number of in-between frames in order avoid temporal aliasing when rendering fast moving object.

Finally, the last chapter introduced a method capable of performing anisotropic filtering on illumination scattered from a translucent media. Results showed that while the method has potential it is still faced with some hampering issues.

It should be clear from reading this thesis that a lot can be gained from anisotropic filtering. Its use, however, is not at all straight forward. The obvious question is whether the difficulty is worth the gain. The answer from this thesis is in most cases yes.

Part IV

Appendices

A

Dimensionality reduction in photon mapping

In photon mapping the surface is presumed to be locally flat. This is reflected by the fact that the radiance estimate performs a two dimensional density estimate on a three dimensional data set, namely the photon map. The consequence of this assumption is that when the surface curvature is high and the photon density is low the error of the radiance estimate can be significant. As an example this error is sometimes evident as an unnatural increase and decrease in illumination intensity near corners. In Figure 4.8a and 4.8b the phenomenon is visible where the caustics are crossing from one wall to the other. Schregle [2003] refers to this error as topological bias.

One way to increase the accuracy of the radiance estimate is to ensure that the photons lie in the same plane. This can be achieved by projecting the photons along their incoming direction onto the tangent plane to the surface at the estimation point. This calculation can be performed in the radiance estimate as:

$$t = \frac{\mathbf{n}\cdot(\mathbf{x}-\mathbf{x}_p)}{\mathbf{n}\cdot\omega_p}, \qquad \boxed{\text{A.1}}$$

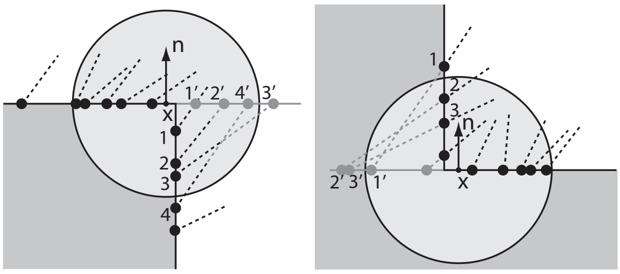
$$\mathbf{x}_{proj} = \mathbf{x}_p + \omega_p t, \qquad \boxed{\text{A.2}}$$

where $\mathbf{x}$ is the estimation point on a surface, and $\mathbf{n}$ is the surface normal at that point. $\mathbf{x}_p$ is the position of the photon and ω_p is the photons incoming direction. Then $\mathbf{x}_{proj}$ is the new projected position of the photon.

Figure A.1a,b illustrates the projection of photons onto the a tangent plane to a surface point. At the surface point a radiance estimate is performed and a circle around the point illustrates the radius in which photons are collected. In A.1a the density estimate is performed close to an outward corner, while the estimate in A.1b is close to an inward corner. In A.1a the photons 1, 2 and 3 are resolved correctly. However, photon 4 is not part of the k nearest photons and does not contribute to the radiance estimate even though it should. Similar for A.1b, photons 2 and 3 are resolved correctly, while photon 1 is incorrectly ignored.

The problem with this method is that it can only guarantee that photons within the search radius are resolved correctly. Photon outside the search radius which should have contributed to radiance estimate are ignored. Havran et al. [2005] has solved this problem by storing photon paths, searching for the k nearest of these instead of searching for photon hit positions. In this way both boundary bias and topological bias is eliminated.

However, our method still has the advantage that it is simple to implement and still resolves certain defects of topological bias such as those seen in Figure 4.8a and 4.8b are reduced. Furthermore, it affects a dimensionality reduction which can be useful in advanced radiance estimation.



(a) **(b)**

Figure A.1: Two diagrams illustrating the projection of photons onto the tangent plane of the surface at the estimation point, x. The photons are projected along their incoming direction.

B

Space and Time Ray Differentials

Jon Sporring, Lars Schjøth, and Kenny Erleben

Abstract

We consider rays bundles emanating from a source such as a camera or light source, and for a particular ray, we estimate the ray density in the neighbourhood using a first order approximation. This work is a generalization of ray differentials [Igehy 1999], and our contribution is to rederive the equations from the literature in a principled fashion, which allows for a generalization both with respect to surface models and time derivatives, and possibly more. Applications are primarily ray tracing and photon splatting.

B.1 Ray differential

In this article we consider reflection and refraction of light rays off and through surfaces as illustrated in Figure B.1, and we will derive the full first order structure of these processes both with respect to nearby light rays and across time.

Consider a point and a viewing direction $\boldsymbol{V}, \boldsymbol{P} \in \mathbb{R}^3$, and a simple 2 dimensional surface $\boldsymbol{x} \in \mathbb{R}^3$, such that

$$\boldsymbol{v} = \boldsymbol{x} - \boldsymbol{P}, \tag{B.1a}$$

$$\boldsymbol{V} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}. \tag{B.1b}$$

We use column vectors, hence $\|\boldsymbol{v}\| = \sqrt{\boldsymbol{v}^T\boldsymbol{v}}$. Following [Igehy 1999; Schjøth et al. 2007] we calculate the partial derivative of $\boldsymbol{P}$ and $\boldsymbol{V}$ with respect to $\boldsymbol{x}$, and we will use the notation of differentials [Magnus and Neudecker 1988]. Differentials are rooted in Taylor series, i.e. consider an analytical function $f : \mathbb{R} \to \mathbb{R}$, and write its Taylor series as,

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + \mathcal{O}(\Delta x^2), \tag{B.2}$$

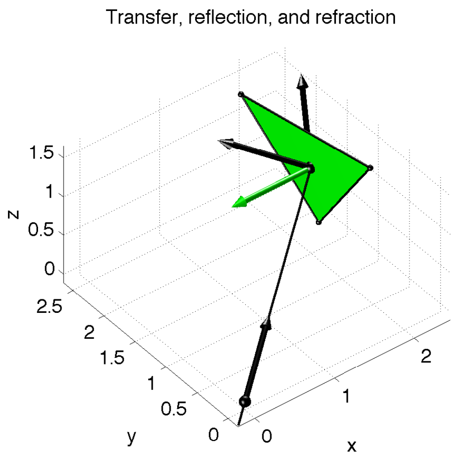


Figure B.1: Transfer, Reflection, and Refraction illustrated for a flat surface. Black arrows are ray directions, green arrow is the normal for the green surface patch.

where $\mathcal{O}$ is the remainder in Landau notation, and f' is the first order derivative of f'. We may reorder (B.2) to become,

$$\Delta f = f(x+\Delta x) - f(x) = f'(x)\Delta x + \mathcal{O}(\Delta x^2), \qquad \boxed{\text{B.3}}$$

and for infinitesimal small Δx's we may ignore the remainder and write,

$$df = f'(x)dx. \qquad \boxed{\text{B.4}}$$

The extension to vector and matrix equations is straight forward, since their Taylor series are element wise Taylor series. We use the same notation except the derivative now is the Jacobian matrix, e.g. for vector equations such as $\boldsymbol{V} \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ and $\boldsymbol{x} \in \mathbb{R}^n$, the Jacobian of $\boldsymbol{V}$ with respect to the variable $\boldsymbol{x}$ is $D_{\boldsymbol{x}}\boldsymbol{V}$ who's ij'th entry is $\frac{\partial \boldsymbol{V}_i}{\partial \boldsymbol{x}_j}$. Hence, the j'th column is the change vector of V when only considering the i'th coordinate direction. The Jacobian with respect to the full space of parameters is often just written as $D\boldsymbol{V}$ for convenience. Matrix equations may be vectorized by simple reordering of their elements, hence avoiding tensor notation. The differential embodies the full first order structure of a function, and a first order estimate of the change is obtained by replacing the infinitesimals with finite values, i.e. dx with Δx.

From (B.1) we may calculate the differential of $d\boldsymbol{V}$ as,

$$\begin{aligned} d\boldsymbol{V} &= \frac{(d\boldsymbol{v})(\boldsymbol{v}^T\boldsymbol{v})^{1/2} - \boldsymbol{v}(\boldsymbol{v}^T\boldsymbol{v})^{-1/2}\boldsymbol{v}^T d\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}} && \boxed{\text{B.5a}} \\ &= \frac{\boldsymbol{v}^T\boldsymbol{v}\boldsymbol{I}_3 - \boldsymbol{v}\boldsymbol{v}^T}{(\boldsymbol{v}^T\boldsymbol{v})^{3/2}}\, d\boldsymbol{v} && \boxed{\text{B.5b}} \\ &= \frac{\boldsymbol{v}^T\boldsymbol{v}\boldsymbol{I}_3 - \boldsymbol{v}\boldsymbol{v}^T}{(\boldsymbol{v}^T\boldsymbol{v})^{3/2}}\,(d\boldsymbol{x} - d\boldsymbol{P}), && \boxed{\text{B.5c}} \end{aligned}$$

where $\boldsymbol{I}_3$ is the 3×3 identity matrix. We are now able to calculate the complete first order structure of $\boldsymbol{V}$ given the first order structure of $\boldsymbol{P}$ and $\boldsymbol{x}$, and with that, we are able to make linear approximation of changes in $\boldsymbol{V}$ given changes in $\boldsymbol{P}$ and $\boldsymbol{x}$. E.g. if $\boldsymbol{P}$ is constant, then $d\boldsymbol{P} = 0$, and

$$d\boldsymbol{V} = \frac{\boldsymbol{v}^T \boldsymbol{v} \boldsymbol{I}_3 - \boldsymbol{v}\boldsymbol{v}^T}{(\boldsymbol{v}^T \boldsymbol{v})^{3/2}}\, d\boldsymbol{x}, \tag{B.6}$$

From this form, we can easily identify the matrix of partial derivatives as

$$\frac{d\boldsymbol{V}}{d\boldsymbol{x}} = \frac{\boldsymbol{v}^T \boldsymbol{v} \boldsymbol{I}_3 - \boldsymbol{v}\boldsymbol{v}^T}{(\boldsymbol{v}^T \boldsymbol{v})^{3/2}}. \tag{B.7}$$

The partial derivative of $\boldsymbol{V}$ with respect to $\boldsymbol{P}$ is similarly found its negative.

If $\boldsymbol{x}$ is a plane, then a natural parametrization will be a set of orthogonal axes spanning the plane, and to continue the example assume that the plane is orthogonal to the third axis, then

$$d\boldsymbol{x} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \end{bmatrix} \tag{B.8}$$

Assuming that we are currently viewing in direction $\boldsymbol{v}^*$ which passes through coordinate $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ and corresponding to $\boldsymbol{V}^*$, then the first order approximation to $\boldsymbol{V}^* + \Delta \boldsymbol{V}$ when $\begin{bmatrix} x_1 + \Delta x_1 & x_2 + \Delta x_2 \end{bmatrix}^T$ is calculated by evaluating $d\boldsymbol{V}$ using $dx_1 = \Delta x_1$ and $dx_2 = \Delta x_2$, i.e.

$$\Delta \boldsymbol{V} = \frac{\boldsymbol{v}^T \boldsymbol{v} \boldsymbol{I}_3 - \boldsymbol{v}\boldsymbol{v}^T}{(\boldsymbol{v}^T \boldsymbol{v})^{3/2}} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \end{bmatrix}. \tag{B.9}$$

If $\boldsymbol{x}$ instead is a sphere of radius 1, we may more naturally use the spherical parametrization,

$$\boldsymbol{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\phi \sin\theta \\ \sin\phi \sin\theta \\ \cos\theta \end{bmatrix}, \tag{B.10}$$

such that

$$\begin{aligned} d\boldsymbol{x} &= \begin{bmatrix} -\sin\phi \sin\theta \, d\phi + \cos\phi \cos\theta \, d\theta \\ \cos\phi \sin\theta \, d\phi + \sin\phi \cos\theta \, d\theta \\ -\sin\theta \, d\theta \end{bmatrix} && \text{B.11a} \\ &= \begin{bmatrix} -\sin\phi \sin\theta & \cos\phi \cos\theta \\ \cos\phi \sin\theta & \sin\phi \cos\theta \\ 0 & -\sin\theta \end{bmatrix} \begin{bmatrix} d\phi \\ d\theta \end{bmatrix} && \text{B.11b} \\ &= \boldsymbol{R}\, d\boldsymbol{\theta}, && \text{B.11c} \end{aligned}$$

where $\boldsymbol{R}$ and $\boldsymbol{\theta}$ are defined as indicated above.

As a note, parallel rays can be implemented by enforcing $d\boldsymbol{V} = 0$ and $\boldsymbol{P}$ non-constant.

The differentials are in know way limited static scenes and cameras. In (B.8) we may add a time derivative as

$$d\boldsymbol{x} = \begin{bmatrix} D_{x_1}\boldsymbol{x} & D_{x_2}\boldsymbol{x} & D_t\boldsymbol{x} \end{bmatrix} \begin{bmatrix} dx_1 \\ dx_2 \\ dt \end{bmatrix} \tag{B.12}$$

where $D_i\boldsymbol{x}$ are vectors of partial derivatives as indicated. I.e. if the plane moves with a unit speed along the third coordinate axis, then $D_t\boldsymbol{x} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$, and if we wish to estimate the change in $\boldsymbol{V}$ as after 1 unit of time, then we evaluate using $dt = 1$.

B.2 Transfer, Reflection, and Refraction

Following [Igehy 1999] we will sketch an iterative process, where a ray originates from a source at location $\boldsymbol{P}$ in direction $\boldsymbol{V}$, is firstly transferred to a surface patch at position $\boldsymbol{Q}$ with direction $\boldsymbol{W}$, and then in parallel reflected and refracted from the patch in directions $\boldsymbol{W}_{\text{reflect}}$ and $\boldsymbol{W}_{\text{refract}}$ respectively. The patch position and reflection and refraction directions respectively are used as source of the ray for next iteration.

B.2.1 Transfer

For a parametrized ray, a transfer onto a surface at distance s the resulting position and view vectors become,

$$\boldsymbol{Q} = \boldsymbol{P} + s\boldsymbol{V}, \tag{B.13a}$$

$$\boldsymbol{W} = \boldsymbol{V}. \tag{B.13b}$$

We will assume that the surface is given implicitly as a scalar function $F : \mathbb{R}^3 \rightarrow \mathbb{R}$, where

$$0 = F(\boldsymbol{Q}), \tag{B.14}$$

and we will assume that there exists a method for solving for the smallest $s^* > 0$, where

$$0 = F(\boldsymbol{P} + s^*\boldsymbol{V}), \tag{B.15}$$

The surface normal, $\boldsymbol{N}$, must exist at $\boldsymbol{Q}$ and will be parallel to DF^T.

The differentials are found to be

$$d\boldsymbol{Q} = d\boldsymbol{P} + \boldsymbol{V}\,ds + s^*\,d\boldsymbol{V}, \tag{B.16a}$$

$$d\boldsymbol{W} = d\boldsymbol{V}. \tag{B.16b}$$

The differential ds is directly related to the curvature of the patch at s^*. For convenience we will in the remainder of this article use the symbol s to denote s^*.

B.2.2 Reflection

Given a ray transferred to a surface, reflection is given by

$$\begin{aligned}
\boldsymbol{Q}_{\text{reflect}} &= \boldsymbol{Q}, \tag*{\boxed{\text{B.17a}}}\\
\boldsymbol{W}_{\text{reflect}} &= \boldsymbol{V} - 2(\boldsymbol{V}^T\boldsymbol{N})\boldsymbol{N}. \tag*{\boxed{\text{B.17b}}}
\end{aligned}$$

Hence, the differentials are, $d\boldsymbol{Q}_{\text{reflect}} = d\boldsymbol{Q}$, and

$$\begin{aligned}
d\boldsymbol{W}_{\text{reflect}} &= d\boldsymbol{V} - 2\left((d\boldsymbol{V}^T\,\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N})\boldsymbol{N} + (\boldsymbol{V}^T\boldsymbol{N})d\boldsymbol{N}\right) \tag*{\boxed{\text{B.18a}}}\\
&= \left(\boldsymbol{I}_3 - 2\boldsymbol{N}\boldsymbol{N}^T\right)d\boldsymbol{V} - 2\left(\boldsymbol{V}^T\boldsymbol{N}\boldsymbol{I}_3 + \boldsymbol{N}\boldsymbol{V}^T\right)d\boldsymbol{N}. \tag*{\boxed{\text{B.18b}}}
\end{aligned}$$

B.2.3 Refraction

Given a ray transferred to a surface, refraction is given by

$$\begin{aligned}
\boldsymbol{Q}_{\text{refract}} &= \boldsymbol{Q}, \tag*{\boxed{\text{B.19a}}}\\
\boldsymbol{W}_{\text{refract}} &= \eta\boldsymbol{V} - \mu\boldsymbol{N}, \tag*{\boxed{\text{B.19b}}}
\end{aligned}$$

where

$$\begin{aligned}
\mu &= \eta\boldsymbol{V}^T\boldsymbol{N} + \sqrt{\xi}, \tag*{\boxed{\text{B.20a}}}\\
\xi &= 1 - \eta^2\left(1 - \left(\boldsymbol{V}^T\boldsymbol{N}\right)^2\right), \tag*{\boxed{\text{B.20b}}}
\end{aligned}$$

and η is the ration of refraction indices of the material bordered by the discussed surface. An often use approximation near $\eta = 1$ is $\xi \simeq \left(\boldsymbol{V}^T\boldsymbol{N}\right)^2$, which we will refrain from, since the refraction ratio between water and air is typically $\eta = 1.33$.

The differentials are found to be, $d\boldsymbol{Q}_{\text{refract}} = d\boldsymbol{Q}$, and assuming that η is constant,

$$\begin{aligned}
d\boldsymbol{W}_{\text{refract}} &= \eta d\boldsymbol{V} - d\mu\,\boldsymbol{N} - \mu d\boldsymbol{N} \tag*{\boxed{\text{B.21a}}}\\
&= \eta d\boldsymbol{V} - \boldsymbol{N}d\mu - \mu d\boldsymbol{N}. \tag*{\boxed{\text{B.21b}}}
\end{aligned}$$

using $d\xi = 2\eta^2\left(\boldsymbol{V}^T\boldsymbol{N}\right)\left(d\boldsymbol{V}^T\,\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N}\right)$, we see that

$$\begin{aligned}
d\mu &= \eta(d\boldsymbol{V}^T\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N}) + \frac{d\xi}{2\sqrt{\xi}} \tag*{\boxed{\text{B.22a}}}\\
&= \eta(d\boldsymbol{V}^T\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N}) + \frac{\eta^2\boldsymbol{V}^T\boldsymbol{N}(d\boldsymbol{V}^T\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N})}{\sqrt{\xi}} \tag*{\boxed{\text{B.22b}}}\\
&= \eta(\boldsymbol{N}^T d\boldsymbol{V} + \boldsymbol{V}^T d\boldsymbol{N}) + \frac{\eta^2\boldsymbol{V}^T\boldsymbol{N}(\boldsymbol{N}^T d\boldsymbol{V} + \boldsymbol{V}^T d\boldsymbol{N})}{\sqrt{\xi}} \tag*{\boxed{\text{B.22c}}}\\
&= \eta\left(1 + \frac{\eta\boldsymbol{V}^T\boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}^T d\boldsymbol{V} + \eta\left(1 + \frac{\eta\boldsymbol{V}^T\boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{V}^T d\boldsymbol{N}. \tag*{\boxed{\text{B.22d}}}
\end{aligned}$$

Gathering terms we find that

$$\begin{aligned} d\boldsymbol{W}_{\text{refract}} = {} & \left(\eta \boldsymbol{I}_3 - \eta\left(1 + \frac{\eta \boldsymbol{V}^T \boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{N}^T\right) d\boldsymbol{V} \\ & - \left(\mu \boldsymbol{I}_3 + \eta\left(1 + \frac{\eta \boldsymbol{V}^T \boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{V}^T\right) d\boldsymbol{N}. \qquad \boxed{\text{B.23}} \end{aligned}$$

B.3 Surface Models

A number of differentials described above depend on the surface of intersection. We will now evaluate the differentials to full depth for a number of popular and practical surface models.

B.3.1 Flat Surface

For a planar surface with normal $\boldsymbol{N}$ and passing through the point $\boldsymbol{Q}_0$ we have that

$$0 = (\boldsymbol{Q}_0 - \boldsymbol{Q})^T \boldsymbol{N}. \qquad \boxed{\text{B.24}}$$

Hence,

$$0 = (\boldsymbol{Q}_0 - \boldsymbol{P} - s\boldsymbol{V})^T \boldsymbol{N}, \qquad \boxed{\text{B.25}}$$

and

$$s = \frac{(\boldsymbol{Q}_0 - \boldsymbol{P})^T \boldsymbol{N}}{\boldsymbol{V}^T \boldsymbol{N}}. \qquad \boxed{\text{B.26}}$$

The full differential of ds is found as follows,

$$\begin{aligned} ds &= \frac{\left(d\left((\boldsymbol{Q}_0 - \boldsymbol{P})^T \boldsymbol{N}\right)\right)\left(\boldsymbol{V}^T\boldsymbol{N}\right) - \left((\boldsymbol{Q}_0 - \boldsymbol{P})^T \boldsymbol{N}\right) d\left(\boldsymbol{V}^T\boldsymbol{N}\right)}{\left(\boldsymbol{V}^T\boldsymbol{N}\right)^2} && \boxed{\text{B.27a}} \\ &= \frac{\left((d\boldsymbol{Q}_0 - d\boldsymbol{P})^T \boldsymbol{N} + (\boldsymbol{Q}_0 - \boldsymbol{P})^T d\boldsymbol{N}\right) - s\left(\left(d\boldsymbol{V}^T\right)\boldsymbol{N} + \boldsymbol{V}^T d\boldsymbol{N}\right)}{\boldsymbol{V}^T\boldsymbol{N}} && \boxed{\text{B.27b}} \\ &= \frac{\left(\boldsymbol{N}^T (d\boldsymbol{Q}_0 - d\boldsymbol{P}) + (\boldsymbol{Q}_0 - \boldsymbol{P})^T d\boldsymbol{N}\right) - s\left(\boldsymbol{N}^T d\boldsymbol{V} + \boldsymbol{V}^T d\boldsymbol{N}\right)}{\boldsymbol{V}^T\boldsymbol{N}} && \boxed{\text{B.27c}} \\ &= \frac{\boldsymbol{N}^T}{\boldsymbol{V}^T\boldsymbol{N}} d\boldsymbol{Q}_0 - \frac{\boldsymbol{N}^T}{\boldsymbol{V}^T\boldsymbol{N}} d\boldsymbol{P} + \frac{(\boldsymbol{Q}_0 - \boldsymbol{P})^T - s\boldsymbol{V}^T}{\boldsymbol{V}^T\boldsymbol{N}} d\boldsymbol{N} - \frac{s\boldsymbol{N}^T}{\boldsymbol{V}^T\boldsymbol{N}} d\boldsymbol{V}. && \boxed{\text{B.27d}} \end{aligned}$$

Combining terms we have:

$$\begin{aligned} d\boldsymbol{Q} &= \boldsymbol{K} d\boldsymbol{P} + s\boldsymbol{K} d\boldsymbol{V} + (\boldsymbol{I}_3 - \boldsymbol{K})\, d\boldsymbol{Q}_0 + \boldsymbol{L} d\boldsymbol{N}, && \boxed{\text{B.28a}} \\ d\boldsymbol{W} &= d\boldsymbol{V}, && \boxed{\text{B.28b}} \\ d\boldsymbol{W}_{\text{reflect}} &= \left(\boldsymbol{I}_3 - 2\boldsymbol{N}\boldsymbol{N}^T\right) d\boldsymbol{V}, && \boxed{\text{B.28c}} \\ d\boldsymbol{W}_{\text{refract}} &= \left(\eta \boldsymbol{I}_3 - \eta\left(1 + \frac{\eta \boldsymbol{V}^T \boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{N}^T\right) d\boldsymbol{V}. && \boxed{\text{B.28d}} \end{aligned}$$

where

$$\boldsymbol{K} = \boldsymbol{I}_3 - \frac{\boldsymbol{V}\boldsymbol{N}^T}{\boldsymbol{V}^T\boldsymbol{N}}, \tag{B.29a}$$

$$\boldsymbol{L} = \frac{\boldsymbol{V}\left(\boldsymbol{Q}_0 - \boldsymbol{P}\right)^T - s\boldsymbol{V}\boldsymbol{V}^T}{\boldsymbol{V}^T\boldsymbol{N}}. \tag{B.29b}$$

Typically, a triangle will be parametrized by its 3 vertices, $\boldsymbol{Q}_0$, $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$, and a more natural parametrization of changes is in terms of the vertices. Such a parametrization allows us to further develop $d\boldsymbol{N}$. Assume that,

$$\boldsymbol{n} = (\boldsymbol{Q}_2 - \boldsymbol{Q}_0) \times (\boldsymbol{Q}_1 - \boldsymbol{Q}_0), \tag{B.30}$$

To be consistent with respect to models for reflection and refraction, we will assume that $\boldsymbol{n}^T\boldsymbol{V} < 0$, otherwise we will interchange $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. For $\boldsymbol{n}^T\boldsymbol{V} < 0$ we find,

$$\boldsymbol{N} = \frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}, \tag{B.31a}$$

$$d\boldsymbol{N} = \frac{\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{I}_3 - \boldsymbol{n}\boldsymbol{n}^T}{(\boldsymbol{n}^T\boldsymbol{n})^{3/2}} d\boldsymbol{n}, \tag{B.31b}$$

$$d\boldsymbol{n} = (d\boldsymbol{Q}_{2\times} - d\boldsymbol{Q}_{0\times})(\boldsymbol{Q}_1 - \boldsymbol{Q}_0) + (\boldsymbol{Q}_{2\times} - \boldsymbol{Q}_{0\times})(d\boldsymbol{Q}_1 - d\boldsymbol{Q}_0) \tag{B.31c}$$

$$= (\boldsymbol{Q}_{2\times} - \boldsymbol{Q}_{0\times})(d\boldsymbol{Q}_1 - d\boldsymbol{Q}_0) - (\boldsymbol{Q}_{1\times} - \boldsymbol{Q}_{0\times})(d\boldsymbol{Q}_2 - d\boldsymbol{Q}_0) \tag{B.31d}$$

$$= (\boldsymbol{Q}_{0\times} - \boldsymbol{Q}_{1\times})d\boldsymbol{Q}_2 + (\boldsymbol{Q}_{2\times} - \boldsymbol{Q}_{0\times})d\boldsymbol{Q}_1 + (\boldsymbol{Q}_{1\times} - \boldsymbol{Q}_{2\times})d\boldsymbol{Q}_0, \tag{B.31e}$$

where for simplicity we convert used the matrix form of cross products, $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{a}_\times \boldsymbol{b} = \boldsymbol{b}_\times^T \boldsymbol{a} = -\boldsymbol{b}_\times \boldsymbol{a}$, where

$$\boldsymbol{c} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \Rightarrow \boldsymbol{c}_\times = \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}, \tag{B.32}$$

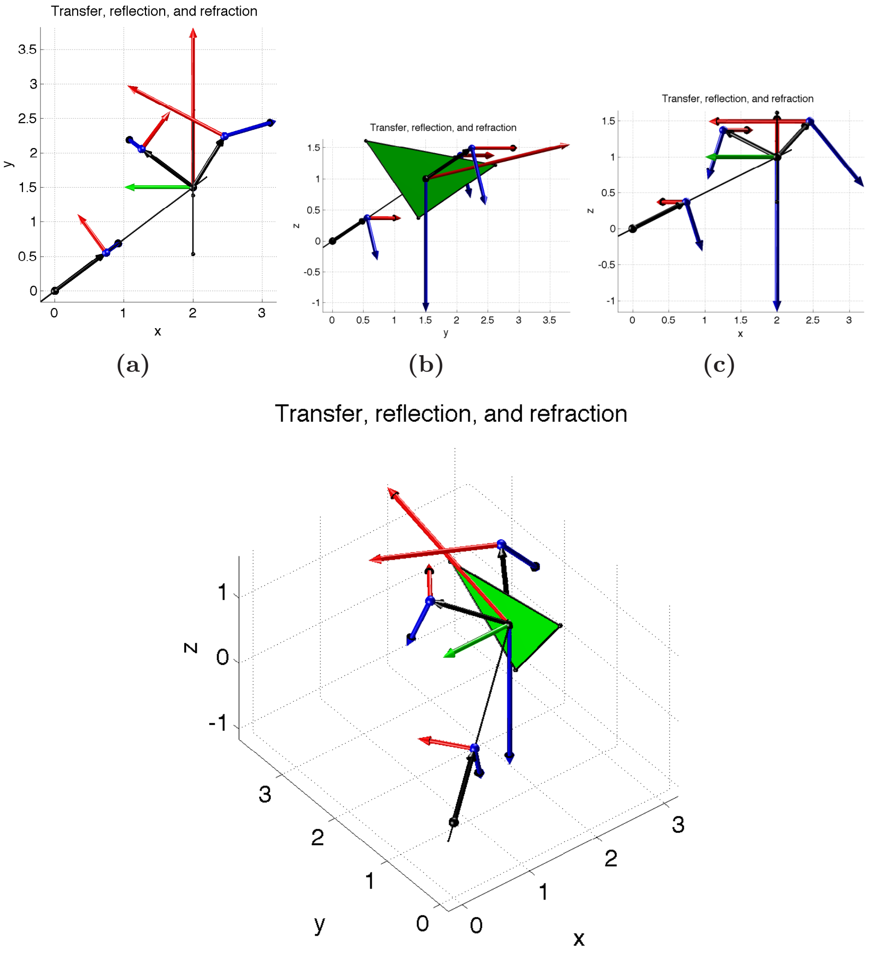
Using

$$\boldsymbol{J} = \frac{\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{I}_3 - \boldsymbol{n}\boldsymbol{n}^T}{(\boldsymbol{n}^T\boldsymbol{n})^{3/2}}, \tag{B.33}$$

we find that

$$\begin{aligned} d\boldsymbol{Q} = {} & \boldsymbol{K}d\boldsymbol{P} + s\boldsymbol{K}d\boldsymbol{V} + \left(\boldsymbol{I}_3 - \boldsymbol{K} + \boldsymbol{L}\boldsymbol{J}(\boldsymbol{Q}_{1\times} - \boldsymbol{Q}_{2\times})\right) d\boldsymbol{Q}_0 \\ & + \boldsymbol{L}\boldsymbol{J}(\boldsymbol{Q}_{2\times} - \boldsymbol{Q}_{0\times})d\boldsymbol{Q}_1 + \boldsymbol{L}\boldsymbol{J}(\boldsymbol{Q}_{0\times} - \boldsymbol{Q}_{1\times})d\boldsymbol{Q}_2, \end{aligned} \tag{B.34}$$

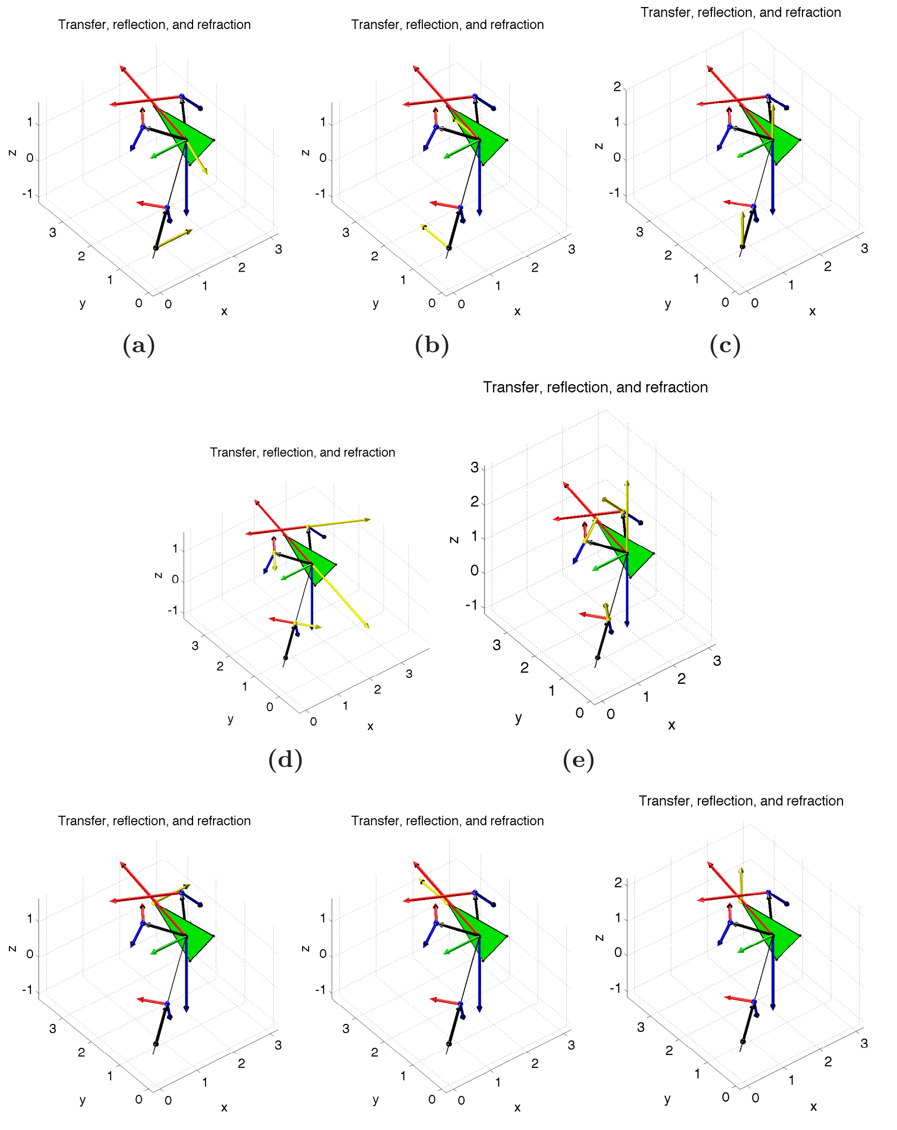
For stationary, flat surfaces $d\boldsymbol{N} = 0$ and $d\boldsymbol{Q}_i = 0, i = 0 \dots 2$, and we may write $d\boldsymbol{Q} = \boldsymbol{K}d\boldsymbol{P} + s\boldsymbol{K}d\boldsymbol{V}$ in agreement with [Igehy 1999]. The rays and spatial differentials are illustrated in Figure B.2. In Figure B.3 are examples of time differentials shown.



(a) (b) (c)

(d)

Figure B.2: Transfer, Reflection, and Refraction for Flat surfaces. Black arrows are ray directions, green is triangle normal, blue and red arrow illustrate the row vectors of $d\boldsymbol{P}/d\boldsymbol{\theta}$, $d\boldsymbol{Q}/d\boldsymbol{\theta}$, $d\boldsymbol{V}/d\boldsymbol{\theta}$, and $d\boldsymbol{W}/d\boldsymbol{\theta}$ as relevant.

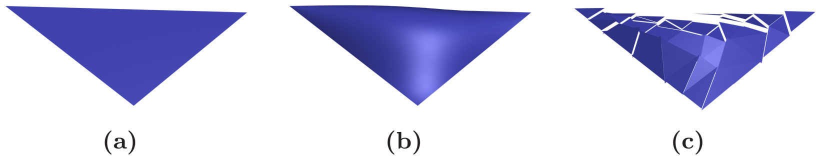


(a) (b) (c)

(d) (e)

(f) (g) (h)

Figure B.3: Time differentials for Flat surfaces. Yellow arrows denote imposed and resulting time derivatives. Subfigures (a)-(c) shows imposed velocities in three orthogonal directions on the origin, $\boldsymbol{P}$, (d)-(e) shows imposed rotational velocities in viewing direction $\boldsymbol{V}$, and (f)-(h) shows imposed velocities in three orthogonal directions on one of the vertices.



(a) (b) (c)

Figure B.4: Phong shading assumes fish scale geometry. A triangle, B.4(a), shaded with Phong's model, B.4(b), expresses a complexity not supported by the real geometry. One way of conceptualizing this model is to think of the triangle as consisting of fish scales, B.4(c); in this mindset every point on the surface of the triangle is associated with an independent local plane or fish scale whose normal is interpolated from the corners of the triangle.

B.3.2 Phong Surface

Phong shading uses a triangle as a base geometry but imposes varying normals across it. Since the flatness of the triangle contradicts the changing normals, we prefer to think of this as a fish scale model as illustrated in Figure B.4.

In the Phong we assume a plane represented by the 3 vertices of a triangle, $\boldsymbol{Q}_0$, $\boldsymbol{Q}_1$, and $\boldsymbol{Q}_2$, and corresponding vertex normals $\boldsymbol{N}_0$, $\boldsymbol{N}_1$, and $\boldsymbol{N}_2$. To calculate the intersection of the view ray with the triangle we use the Flat surface model (B.24) by using the triangle normal,

$$\boldsymbol{n}_{\text{flat}} = (\boldsymbol{Q}_2 - \boldsymbol{Q}_0) \times (\boldsymbol{Q}_1 - \boldsymbol{Q}_0) \tag{B.35a}$$

$$\boldsymbol{N}_{\text{flat}} = \frac{\boldsymbol{n}_{\text{flat}}}{\|\boldsymbol{n}_{\text{flat}}\|} \tag{B.35b}$$

For visualization, reflection and refraction we construct an linearly interpolated normal from the three vertex normals. Note that the flat normal and interpolated vertex normal most often won't coincide, and as a consequence $d\boldsymbol{Q}/d\boldsymbol{N}_{\text{phong}}$ will not span the triangle. Hence, for the Phong surface model we use $\boldsymbol{N}_{\text{flat}}$ for calculating $\boldsymbol{Q}$ and only concern oureselves with $d\boldsymbol{Q}/d\boldsymbol{N}_{\text{flat}}$. As for the Flat model, we find the point of intersection by solving $0 = (\boldsymbol{Q}_0 - \boldsymbol{P} - s\boldsymbol{V})^T\boldsymbol{N}_{\text{flat}}$, as

$$s = \frac{(\boldsymbol{Q}_0 - \boldsymbol{P})^T\boldsymbol{N}_{\text{flat}}}{\boldsymbol{V}^T\boldsymbol{N}_{\text{flat}}}. \tag{B.36}$$

such that

$$d\boldsymbol{Q} = \boldsymbol{K}d\boldsymbol{P} + s\boldsymbol{K}d\boldsymbol{V} + (\boldsymbol{I}_3 - \boldsymbol{K} + \boldsymbol{L}\boldsymbol{J}_{\text{flat}}(\boldsymbol{Q}_{1\times} - \boldsymbol{Q}_{2\times}))\,d\boldsymbol{Q}_0 \tag{B.37}$$

$$+ \boldsymbol{L}\boldsymbol{J}_{\text{flat}}(\boldsymbol{Q}_{2\times} - \boldsymbol{Q}_{0\times})d\boldsymbol{Q}_1 + \boldsymbol{L}\boldsymbol{J}_{\text{flat}}(\boldsymbol{Q}_{0\times} - \boldsymbol{Q}_{1\times})d\boldsymbol{Q}_2, \tag{B.38}$$

$$d\boldsymbol{W} = d\boldsymbol{V}, \tag{B.39}$$

where

$$\boldsymbol{K} = \boldsymbol{I}_3 - \frac{\boldsymbol{V}\boldsymbol{N}_{\text{flat}}^T}{\boldsymbol{V}^T\boldsymbol{N}_{\text{flat}}}, \tag{B.40a}$$

$$\boldsymbol{L} = \frac{\boldsymbol{V}\left(\boldsymbol{Q}_0 - \boldsymbol{P}\right)^T - s\boldsymbol{V}\boldsymbol{V}^T}{\boldsymbol{V}^T\boldsymbol{N}_{\text{flat}}}, \tag{B.40b}$$

$$\boldsymbol{J}_{\text{flat}} = \frac{\boldsymbol{n}_{\text{flat}}^T\boldsymbol{n}_{\text{flat}}\boldsymbol{I}_3 - \boldsymbol{n}_{\text{flat}}\boldsymbol{n}_{\text{flat}}^T}{(\boldsymbol{n}_{\text{flat}}^T\boldsymbol{n}_{\text{flat}})^{3/2}}. \tag{B.40c}$$

To interpolate the vertex normals at the point of intersection, $\boldsymbol{Q}$, we calculate the Barycentric coordinates,

$$\boldsymbol{Q} = \lambda_0\boldsymbol{Q}_0 + \lambda_1\boldsymbol{Q}_1 + \lambda_2\boldsymbol{Q}_2, \tag{B.41}$$

where λ_0, λ_1, and λ_2 are homogeneous Barycentric coordinates such that $\lambda_0 + \lambda_1 + \lambda_2 = 1$. The Barycentric coordinates are then used to interpolate the vertex normals as,

$$\boldsymbol{n} = \lambda_0\boldsymbol{N}_0 + \lambda_1\boldsymbol{N}_1 + \lambda_2\boldsymbol{N}_2, \tag{B.42a}$$

$$\boldsymbol{N} = \frac{\boldsymbol{n}}{\|\boldsymbol{n}\|}. \tag{B.42b}$$

Assuming that a ray passing through $\boldsymbol{P}$ with direction $\boldsymbol{V}$, and that it intersects a triangle within vertices $\boldsymbol{Q}_0$, $\boldsymbol{Q}_1$, and $\boldsymbol{Q}_2$, then $0 \leq \lambda_i \leq 1$, and we may find the Barycentric coordinates using Möller and Trumbore's algorithm [Möller and Trumbore 1997]: Let

$$\boldsymbol{E}_0 = \boldsymbol{Q}_1 - \boldsymbol{Q}_0, \tag{B.43a}$$

$$\boldsymbol{E}_1 = \boldsymbol{Q}_2 - \boldsymbol{Q}_0, \tag{B.43b}$$

$$\boldsymbol{T} = \boldsymbol{P} - \boldsymbol{Q}_0 \tag{B.43c}$$

then

$$\lambda_1 = \frac{(\boldsymbol{V} \times \boldsymbol{T})^T\boldsymbol{E}_1}{(\boldsymbol{V} \times \boldsymbol{E}_0)^T\boldsymbol{E}_1} \tag{B.44a}$$

$$= \frac{\boldsymbol{\tau}^T\boldsymbol{E}_1}{\boldsymbol{\gamma}^T\boldsymbol{E}_1} \tag{B.44b}$$

$$\lambda_2 = \frac{(\boldsymbol{V} \times \boldsymbol{T})^T\boldsymbol{E}_0}{(\boldsymbol{V} \times \boldsymbol{E}_1)^T\boldsymbol{E}_0} \tag{B.44c}$$

$$= \frac{\boldsymbol{\tau}^T\boldsymbol{E}_0}{\boldsymbol{\zeta}^T\boldsymbol{E}_0}, \tag{B.44d}$$

where $\boldsymbol{\tau} = \boldsymbol{V} \times \boldsymbol{T}$, $\boldsymbol{\gamma} = \boldsymbol{V} \times \boldsymbol{E}_0$, $\boldsymbol{\zeta} = \boldsymbol{V} \times \boldsymbol{E}_1$, and $\lambda_0 = 1 - \lambda_1 - \lambda_2$.

The differential, $d\boldsymbol{N}$, is now found to be,

$$d\boldsymbol{N} = \frac{\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{I}_3 - \boldsymbol{n}\boldsymbol{n}^T}{(\boldsymbol{n}^T\boldsymbol{n})^{3/2}}d\boldsymbol{n}, \tag{B.45a}$$

$$d\boldsymbol{n} = \boldsymbol{N}_0 d\lambda_0 + \lambda_0 d\boldsymbol{N}_0 + \boldsymbol{N}_1 d\lambda_1 + \lambda_1 d\boldsymbol{N}_1 + \boldsymbol{N}_2 d\lambda_2 + \lambda_2 d\boldsymbol{N}_2. \tag{B.45b}$$

Since,

$$
\begin{aligned}
d\lambda_0 &= -d\lambda_1 - d\lambda_2, && \boxed{\text{B.46a}}\\
d\lambda_1 &= \frac{(d\boldsymbol{\tau}^T\boldsymbol{E}_1 + \boldsymbol{\tau}^T d\boldsymbol{E}_1)\boldsymbol{\gamma}^T\boldsymbol{E}_1 - \boldsymbol{\tau}^T\boldsymbol{E}_1(d\boldsymbol{\gamma}^T\boldsymbol{E}_1 + \boldsymbol{\gamma}^T d\boldsymbol{E}_1)}{(\boldsymbol{\gamma}^T\boldsymbol{E}_1)^2}\\
&= \frac{\boldsymbol{\gamma}^T\boldsymbol{E}_1(\boldsymbol{E}_1^T d\boldsymbol{\tau} + \boldsymbol{\tau}^T d\boldsymbol{E}_1) - \boldsymbol{\tau}^T\boldsymbol{E}_1(\boldsymbol{E}_1^T d\boldsymbol{\gamma} + \boldsymbol{\gamma}^T d\boldsymbol{E}_1)}{(\boldsymbol{\gamma}^T\boldsymbol{E}_1)^2}\\
&= \frac{\boldsymbol{\gamma}^T\boldsymbol{S}_1 d\boldsymbol{\tau} - \boldsymbol{\tau}^T\boldsymbol{S}_1 d\boldsymbol{\gamma} + \boldsymbol{E}_1^T\left(\boldsymbol{\gamma}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\gamma}^T\right)d\boldsymbol{E}_1}{\boldsymbol{\gamma}^T\boldsymbol{S}_1\boldsymbol{\gamma}}, && \boxed{\text{B.46b}}\\
d\lambda_2 &= \frac{(d\boldsymbol{\tau}^T\boldsymbol{E}_0 + \boldsymbol{\tau}^T d\boldsymbol{E}_0)\boldsymbol{\zeta}^T\boldsymbol{E}_0 - \boldsymbol{\tau}^T\boldsymbol{E}_0(d\boldsymbol{\zeta}^T\boldsymbol{E}_0 + \boldsymbol{\zeta}^T d\boldsymbol{E}_0)}{(\boldsymbol{\zeta}^T\boldsymbol{E}_0)^2}\\
&= \frac{\boldsymbol{\zeta}^T\boldsymbol{E}_0(\boldsymbol{E}_0^T d\boldsymbol{\tau} + \boldsymbol{\tau}^T d\boldsymbol{E}_0) - \boldsymbol{\tau}^T\boldsymbol{E}_0(\boldsymbol{E}_0^T d\boldsymbol{\zeta} + \boldsymbol{\zeta}^T d\boldsymbol{E}_0)}{(\boldsymbol{\zeta}^T\boldsymbol{E}_0)^2}\\
&= \frac{\boldsymbol{\zeta}^T\boldsymbol{S}_0 d\boldsymbol{\tau} - \boldsymbol{\tau}^T\boldsymbol{S}_0 d\boldsymbol{\zeta} + \boldsymbol{E}_0^T\left(\boldsymbol{\zeta}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\zeta}^T\right)d\boldsymbol{E}_0}{\boldsymbol{\zeta}^T\boldsymbol{S}_0\zeta}, && \boxed{\text{B.46c}}
\end{aligned}
$$

where $\boldsymbol{S}_1 = \boldsymbol{E}_1\boldsymbol{E}_1^T$, and $\boldsymbol{S}_0 = \boldsymbol{E}_0\boldsymbol{E}_0^T$. Thus we find that

$$
\begin{aligned}
d\boldsymbol{N} = \boldsymbol{J}\Bigg(&\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2\\
&+ (\boldsymbol{N}_1 - \boldsymbol{N}_0)\left(\frac{\boldsymbol{\gamma}^T\boldsymbol{S}_1 d\boldsymbol{\tau} - \boldsymbol{\tau}^T\boldsymbol{S}_1 d\boldsymbol{\gamma} + \boldsymbol{E}_1^T\left(\boldsymbol{\gamma}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\gamma}^T\right)d\boldsymbol{E}_1}{\boldsymbol{\gamma}^T\boldsymbol{S}_1\boldsymbol{\gamma}}\right)\\
&+ (\boldsymbol{N}_2 - \boldsymbol{N}_0)\left(\frac{\boldsymbol{\zeta}^T\boldsymbol{S}_0 d\boldsymbol{\tau} - \boldsymbol{\tau}^T\boldsymbol{S}_0 d\boldsymbol{\zeta} + \boldsymbol{E}_0^T\left(\boldsymbol{\zeta}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\zeta}^T\right)d\boldsymbol{E}_0}{\boldsymbol{\zeta}^T\boldsymbol{S}_0\zeta}\right)\Bigg)\\
& && \boxed{\text{B.47a}}\\
= \boldsymbol{J}\Bigg(&\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2\\
&+ \left(\boldsymbol{\Delta}_1\boldsymbol{\gamma}^T\boldsymbol{S}_1 + \boldsymbol{\Delta}_2\boldsymbol{\zeta}^T\boldsymbol{S}_0\right)d\boldsymbol{\tau} - \boldsymbol{\Delta}_1\boldsymbol{\tau}^T\boldsymbol{S}_1 d\boldsymbol{\gamma} - \boldsymbol{\Delta}_2\boldsymbol{\tau}^T\boldsymbol{S}_0 d\boldsymbol{\zeta}\\
&+ \boldsymbol{\Delta}_2\boldsymbol{E}_0^T\left(\boldsymbol{\zeta}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\zeta}^T\right)d\boldsymbol{E}_0 + \boldsymbol{\Delta}_1\boldsymbol{E}_1^T\left(\boldsymbol{\gamma}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\gamma}^T\right)d\boldsymbol{E}_1\Bigg) && \boxed{\text{B.47b}}
\end{aligned}
$$

where $\boldsymbol{J} = \frac{\boldsymbol{n}^T\boldsymbol{n}\boldsymbol{I}_3 - \boldsymbol{n}\boldsymbol{n}^T}{(\boldsymbol{n}^T\boldsymbol{n})^{3/2}}$, $\boldsymbol{\Delta}_1 = \frac{(\boldsymbol{N}_1 - \boldsymbol{N}_0)}{\boldsymbol{\gamma}^T\boldsymbol{S}_1\boldsymbol{\gamma}}$, and $\boldsymbol{\Delta}_2 = \frac{(\boldsymbol{N}_2 - \boldsymbol{N}_0)}{\boldsymbol{\zeta}^T\boldsymbol{S}_0\zeta}$. For simplicity we convert cross products into matrix form, $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{a}_\times\boldsymbol{b} = \boldsymbol{b}_\times^T\boldsymbol{a} = -\boldsymbol{b}_\times\boldsymbol{a}$, where

$$
\boldsymbol{c} = \begin{bmatrix} c_1\\ c_2\\ c_3 \end{bmatrix} \Rightarrow \boldsymbol{c}_\times = \begin{bmatrix} 0 & -c_3 & c_2\\ c_3 & 0 & -c_1\\ -c_2 & c_1 & 0 \end{bmatrix}, \qquad \boxed{\text{B.48}}
$$

hence,

$$
\begin{aligned}
d\boldsymbol{\tau} &= d\boldsymbol{V}_\times\boldsymbol{T} + \boldsymbol{V}_\times d\boldsymbol{T} = \boldsymbol{V}_\times d\boldsymbol{T} - \boldsymbol{T}_\times d\boldsymbol{V}, && \boxed{\text{B.49a}}\\
d\boldsymbol{\gamma} &= d\boldsymbol{V}_\times\boldsymbol{E}_0 + \boldsymbol{V}_\times d\boldsymbol{E}_0 = \boldsymbol{V}_\times d\boldsymbol{E}_0 - \boldsymbol{E}_{0\times} d\boldsymbol{V}, && \boxed{\text{B.49b}}\\
d\boldsymbol{\zeta} &= d\boldsymbol{V}_\times\boldsymbol{E}_1 + \boldsymbol{V}_\times d\boldsymbol{E}_1 = \boldsymbol{V}_\times d\boldsymbol{E}_1 - \boldsymbol{E}_{1\times} d\boldsymbol{V}, && \boxed{\text{B.49c}}
\end{aligned}
$$

implying that

$$\begin{aligned} d\boldsymbol{N} = \boldsymbol{J}\Bigg(&\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2 \\ &+ \big(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{E}_{0\times} - \boldsymbol{\gamma}^T \boldsymbol{S}_1 \boldsymbol{T}_\times\right) \\ &\qquad + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T \boldsymbol{S}_0 \boldsymbol{T}_\times\right)\big) d\boldsymbol{V} \\ &+ \left(\boldsymbol{\Delta}_1 \boldsymbol{\gamma}^T \boldsymbol{S}_1 + \boldsymbol{\Delta}_2 \boldsymbol{\zeta}^T \boldsymbol{S}_0\right) \boldsymbol{V}_\times d\boldsymbol{T} \\ &+ \left(\boldsymbol{\Delta}_2 \boldsymbol{E}_0^T \left(\boldsymbol{\zeta}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\zeta}^T\right) - \boldsymbol{\Delta}_1 \boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{V}_\times\right) d\boldsymbol{E}_0 \\ &+ \left(\boldsymbol{\Delta}_1 \boldsymbol{E}_1^T \left(\boldsymbol{\gamma}\boldsymbol{\tau}^T - \boldsymbol{\tau}\boldsymbol{\gamma}^T\right) - \boldsymbol{\Delta}_2 \boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{V}_\times\right) d\boldsymbol{E}_1 \Bigg). \end{aligned} \tag{B.50}$$

Since $d\boldsymbol{E}_0 = d\boldsymbol{Q}_1 - d\boldsymbol{Q}_0$, $d\boldsymbol{E}_1 = d\boldsymbol{Q}_2 - d\boldsymbol{Q}_0$, $d\boldsymbol{T} = d\boldsymbol{P} - d\boldsymbol{Q}_0$, we find that

$$\begin{aligned} d\boldsymbol{N} = \boldsymbol{J}\Bigg(&\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2 \\ &+ \left(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{E}_{0\times} - \boldsymbol{\gamma}^T \boldsymbol{S}_1 \boldsymbol{T}_\times\right) + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T \boldsymbol{S}_0 \boldsymbol{T}_\times\right)\right) d\boldsymbol{V} \\ &+ \left(\boldsymbol{\Delta}_1 \boldsymbol{\gamma}^T \boldsymbol{S}_1 + \boldsymbol{\Delta}_2 \boldsymbol{\zeta}^T \boldsymbol{S}_0\right) \boldsymbol{V}_\times d\boldsymbol{P} \\ &+ \big(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T - \boldsymbol{\gamma}^T\right) \boldsymbol{S}_1 \boldsymbol{V}_\times + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T - \boldsymbol{\zeta}^T\right) \boldsymbol{S}_0 \boldsymbol{V}_\times \\ &\qquad + \boldsymbol{\Delta}_2 \boldsymbol{\Xi}_0 + \boldsymbol{\Delta}_1 \boldsymbol{\Xi}_1\big) d\boldsymbol{Q}_0 \\ &- \left(\boldsymbol{\Delta}_2 \boldsymbol{\Xi}_0 + \boldsymbol{\Delta}_1 \boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_1 \\ &- \left(\boldsymbol{\Delta}_1 \boldsymbol{\Xi}_1 + \boldsymbol{\Delta}_2 \boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_2 \Bigg) \end{aligned} \tag{B.51}$$

where $\boldsymbol{\Xi}_0 = \boldsymbol{E}_0^T \left(\boldsymbol{\tau}\boldsymbol{\zeta}^T - \boldsymbol{\zeta}\boldsymbol{\tau}^T\right)$, and $\boldsymbol{\Xi}_1 = \boldsymbol{E}_1^T \left(\boldsymbol{\tau}\boldsymbol{\gamma}^T - \boldsymbol{\gamma}\boldsymbol{\tau}^T\right)$.

Gathering terms for reflection and refraction we find that

$$\begin{aligned} d\boldsymbol{W}_{\text{reflect}} &= \left(\boldsymbol{I}_3 - 2\boldsymbol{N}\boldsymbol{N}^T\right) d\boldsymbol{V} - 2\left(\boldsymbol{V}^T \boldsymbol{N} \boldsymbol{I}_3 + \boldsymbol{N}\boldsymbol{V}^T\right) d\boldsymbol{N} \\ &= \left(\boldsymbol{I}_3 - 2\boldsymbol{N}\boldsymbol{N}^T\right) d\boldsymbol{V} \\ &\qquad + \boldsymbol{M}\Bigg(\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2 \\ &\qquad\quad + \big(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{E}_{0\times} - \boldsymbol{\gamma}^T \boldsymbol{S}_1 \boldsymbol{T}_\times\right) \\ &\qquad\qquad + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T \boldsymbol{S}_0 \boldsymbol{T}_\times\right)\big) d\boldsymbol{V} \\ &\qquad\quad + (\boldsymbol{\Delta}_1 \boldsymbol{\gamma}^T \boldsymbol{S}_1 + \boldsymbol{\Delta}_2 \boldsymbol{\zeta}^T \boldsymbol{S}_0) \boldsymbol{V}_\times d\boldsymbol{P} \\ &\qquad\quad + \big(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T - \boldsymbol{\gamma}^T\right) \boldsymbol{S}_1 \boldsymbol{V}_\times + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T - \boldsymbol{\zeta}^T\right) \boldsymbol{S}_0 \boldsymbol{V}_\times \\ &\qquad\qquad + \boldsymbol{\Delta}_2 \boldsymbol{\Xi}_0 + \boldsymbol{\Delta}_1 \boldsymbol{\Xi}_1\big) d\boldsymbol{Q}_0 \\ &\qquad\quad - \left(\boldsymbol{\Delta}_2 \boldsymbol{\Xi}_0 + \boldsymbol{\Delta}_1 \boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_1 \\ &\qquad\quad - \left(\boldsymbol{\Delta}_1 \boldsymbol{\Xi}_1 + \boldsymbol{\Delta}_2 \boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_2 \Bigg). \end{aligned} \tag{B.52a}$$

and

$$\begin{aligned}
d\boldsymbol{W}_{\text{reflect}} = {} & \boldsymbol{M}\lambda_0 d\boldsymbol{N}_0 + \boldsymbol{M}\lambda_1 d\boldsymbol{N}_1 + \boldsymbol{M}\lambda_2 d\boldsymbol{N}_2 \\
& + \big(\boldsymbol{I}_3 - 2\boldsymbol{N}\boldsymbol{N}^T + \boldsymbol{M}\big(\boldsymbol{\Delta}_1\left(\boldsymbol{\tau}^T\boldsymbol{S}_1\boldsymbol{E}_{0\times} - \gamma^T\boldsymbol{S}_1\boldsymbol{T}_\times\right) + \\
& \quad \boldsymbol{\Delta}_2\left(\boldsymbol{\tau}^T\boldsymbol{S}_0\boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T\boldsymbol{S}_0\boldsymbol{T}_\times\right)\big)\big)d\boldsymbol{V} \\
& + \boldsymbol{M}\left(\boldsymbol{\Delta}_1\gamma^T\boldsymbol{S}_1 + \boldsymbol{\Delta}_2\boldsymbol{\zeta}^T\boldsymbol{S}_0\right)\boldsymbol{V}_\times d\boldsymbol{P} \\
& + \boldsymbol{M}\big(\boldsymbol{\Delta}_1\left(\boldsymbol{\tau}^T - \gamma^T\right)\boldsymbol{S}_1\boldsymbol{V}_\times \\
& \qquad + \boldsymbol{\Delta}_2\left(\boldsymbol{\tau}^T - \boldsymbol{\zeta}^T\right)\boldsymbol{S}_0\boldsymbol{V}_\times + \boldsymbol{\Delta}_2\Xi_0 + \boldsymbol{\Delta}_1\Xi_1\big)d\boldsymbol{Q}_0 \\
& - \boldsymbol{M}\left(\boldsymbol{\Delta}_2\Xi_0 + \boldsymbol{\Delta}_1\boldsymbol{\tau}^T\boldsymbol{S}_1\boldsymbol{V}_\times\right)d\boldsymbol{Q}_1 \\
& - \boldsymbol{M}\left(\boldsymbol{\Delta}_1\Xi_1 + \boldsymbol{\Delta}_2\boldsymbol{\tau}^T\boldsymbol{S}_0\boldsymbol{V}_\times\right)d\boldsymbol{Q}_2.
\end{aligned} \tag{B.52b}$$

where $\boldsymbol{M} = -2\left(\boldsymbol{V}^T\boldsymbol{N}\boldsymbol{I}_3 + \boldsymbol{N}\boldsymbol{V}^T\right)\boldsymbol{J}$, and

$$\begin{aligned}
d\boldsymbol{W}_{\text{refract}} = {} & \left(\eta\boldsymbol{I}_3 - \eta\left(1 + \frac{\eta\boldsymbol{V}^T\boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{N}^T\right)d\boldsymbol{V} \\
& - \left(\mu\boldsymbol{I}_3 + \eta\left(1 + \frac{\eta\boldsymbol{V}^T\boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{V}^T\right)d\boldsymbol{N}
\end{aligned} \tag{B.53}$$

so that

$$\begin{aligned}
d\boldsymbol{W}_{\text{refract}} = {} & \left(\eta\boldsymbol{I}_3 - \eta\left(1 + \frac{\eta\boldsymbol{V}^T\boldsymbol{N}}{\sqrt{\xi}}\right)\boldsymbol{N}\boldsymbol{N}^T\right)d\boldsymbol{V} \\
& + \boldsymbol{H}\Bigg(\lambda_0 d\boldsymbol{N}_0 + \lambda_1 d\boldsymbol{N}_1 + \lambda_2 d\boldsymbol{N}_2 \\
& \quad + \big(\boldsymbol{\Delta}_1\left(\boldsymbol{\tau}^T\boldsymbol{S}_1\boldsymbol{E}_{0\times} - \gamma^T\boldsymbol{S}_1\boldsymbol{T}_\times\right) \\
& \qquad + \boldsymbol{\Delta}_2\left(\boldsymbol{\tau}^T\boldsymbol{S}_0\boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T\boldsymbol{S}_0\boldsymbol{T}_\times\right)\big)d\boldsymbol{V} \\
& \quad + \left(\boldsymbol{\Delta}_1\gamma^T\boldsymbol{S}_1 + \boldsymbol{\Delta}_2\boldsymbol{\zeta}^T\boldsymbol{S}_0\right)\boldsymbol{V}_\times d\boldsymbol{P} \\
& \quad + \big(\boldsymbol{\Delta}_1\left(\boldsymbol{\tau}^T - \gamma^T\right)\boldsymbol{S}_1\boldsymbol{V}_\times + \boldsymbol{\Delta}_2\left(\boldsymbol{\tau}^T - \boldsymbol{\zeta}^T\right)\boldsymbol{S}_0\boldsymbol{V}_\times \\
& \qquad + \boldsymbol{\Delta}_2\Xi_0 + \boldsymbol{\Delta}_1\Xi_1\big)d\boldsymbol{Q}_0 \\
& \quad - \left(\boldsymbol{\Delta}_2\Xi_0 + \boldsymbol{\Delta}_1\boldsymbol{\tau}^T\boldsymbol{S}_1\boldsymbol{V}_\times\right)d\boldsymbol{Q}_1 \\
& \quad - \left(\boldsymbol{\Delta}_1\Xi_1 + \boldsymbol{\Delta}_2\boldsymbol{\tau}^T\boldsymbol{S}_0\boldsymbol{V}_\times\right)d\boldsymbol{Q}_2\Bigg),
\end{aligned} \tag{B.54}$$

from which it follows that

$$\begin{aligned} d\boldsymbol{W}_{\text{refract}} = \Bigg(& (\eta \boldsymbol{I}_3 - \eta \left(1 + \frac{\eta \boldsymbol{V}^T \boldsymbol{N}}{\sqrt{\xi}}\right) \boldsymbol{N}\boldsymbol{N}^T V + \boldsymbol{H}\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{E}_{0\times} - \boldsymbol{\gamma}^T \boldsymbol{S}_1 \boldsymbol{T}_\times\right) \\ & + \boldsymbol{H}\boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{E}_{1\times} - \boldsymbol{\zeta}^T \boldsymbol{S}_0 \boldsymbol{T}_\times\right) \Bigg) d\boldsymbol{V} \\ & + \boldsymbol{H}\lambda_0 d\boldsymbol{N}_0 + \boldsymbol{H}\lambda_1 d\boldsymbol{N}_1 + \boldsymbol{H}\lambda_2 d\boldsymbol{N}_2 \\ & + \boldsymbol{H} \left(\boldsymbol{\Delta}_1 \gamma^T \boldsymbol{S}_1 + \boldsymbol{\Delta}_2 \zeta^T \boldsymbol{S}_0\right) \boldsymbol{V}_\times d\boldsymbol{P} \\ & + \boldsymbol{H}\big(\boldsymbol{\Delta}_1 \left(\boldsymbol{\tau}^T - \gamma^T\right) \boldsymbol{S}_1 \boldsymbol{V}_\times + \boldsymbol{\Delta}_2 \left(\boldsymbol{\tau}^T - \zeta^T\right) \boldsymbol{S}_0 \boldsymbol{V}_\times \\ & \qquad + \boldsymbol{\Delta}_2 \Xi_0 + \boldsymbol{\Delta}_1 \Xi_1\big) d\boldsymbol{Q}_0 \\ & - \boldsymbol{H} \left(\boldsymbol{\Delta}_2 \Xi_0 + \boldsymbol{\Delta}_1 \boldsymbol{\tau}^T \boldsymbol{S}_1 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_1 \\ & - \boldsymbol{H} \left(\boldsymbol{\Delta}_1 \Xi_1 + \boldsymbol{\Delta}_2 \boldsymbol{\tau}^T \boldsymbol{S}_0 \boldsymbol{V}_\times\right) d\boldsymbol{Q}_2, \end{aligned} \tag{B.55}$$

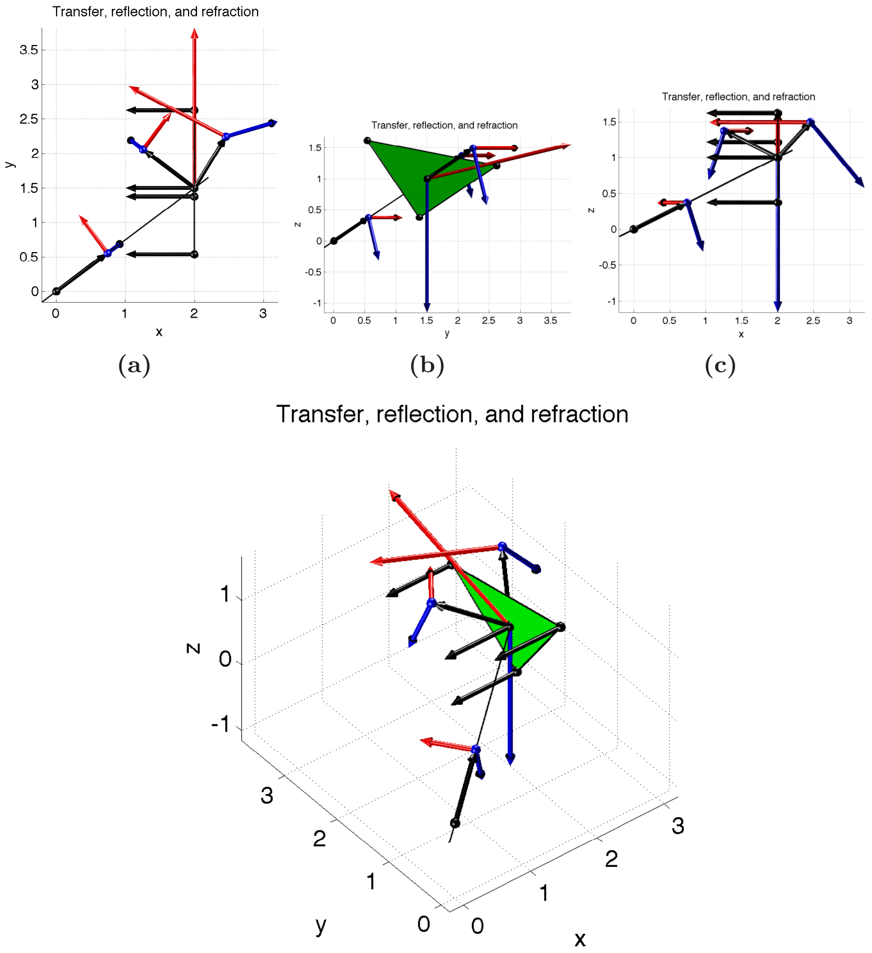
where $\boldsymbol{H} = -\left(\mu \boldsymbol{I}_3 + \eta \left(1 + \frac{\eta \boldsymbol{V}^T \boldsymbol{N}}{\sqrt{\xi}}\right) \boldsymbol{N}\boldsymbol{V}^T\right) \boldsymbol{J}$.

An example of the rotational derivatives using the Phong surface model is shown in Figure B.5, and resulting velocities in Figure B.6.

B.4 Conclusion

In this article we have reiterated [Igehy 1999; Schjøth et al. 2007], and evaluated the full differentials of linear rays emanating from a source and intersecting a surface given on implicit form in a principled manner taking reflection and refraction into account. We have further given closed form solutions for two surface models: flat and Phong. Our differentials are simpler and are more consistent than [Igehy 1999], and they allow for easy extension to other parameters than viewing directions. Specifically, we have considered change in ray origin, allowing for parallel rays, and change in time, allowing for movies.

Conceptually, we model ray bundles instead of rays and obvious applications are ray tracing and photon splatting, but the methodology is naturally and easily extended to all phenomena well approximated by first order Taylor series.



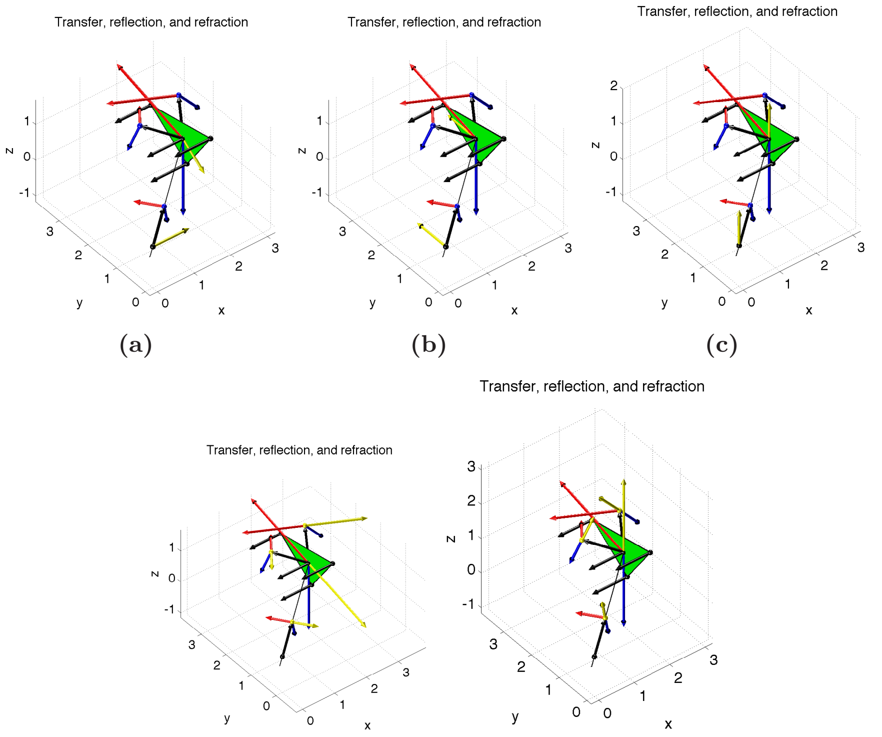
(a)

(b)

(c)

(d)

Figure B.5: Transfer, Reflection, and Refraction for Phong surfaces. Black arrows are ray directions and interpolated normal, green is triangle normal, blue and red arrow illustrate the row vectors of $d\boldsymbol{P}/d\boldsymbol{\theta}$, $d\boldsymbol{Q}/d\boldsymbol{\theta}$, $d\boldsymbol{V}/d\boldsymbol{\theta}$, and $d\boldsymbol{W}/d\boldsymbol{\theta}$ as relevant. Note that $d\boldsymbol{Q}/d\boldsymbol{\theta}$ vectors do not lie in the triangular plane.


Transfer, reflection, and refraction
Transfer, reflection, and refraction
Transfer, reflection, and refraction
Transfer, reflection, and refraction
Transfer, reflection, and refraction
z
y
x
-1
0
1
2
3

(a)

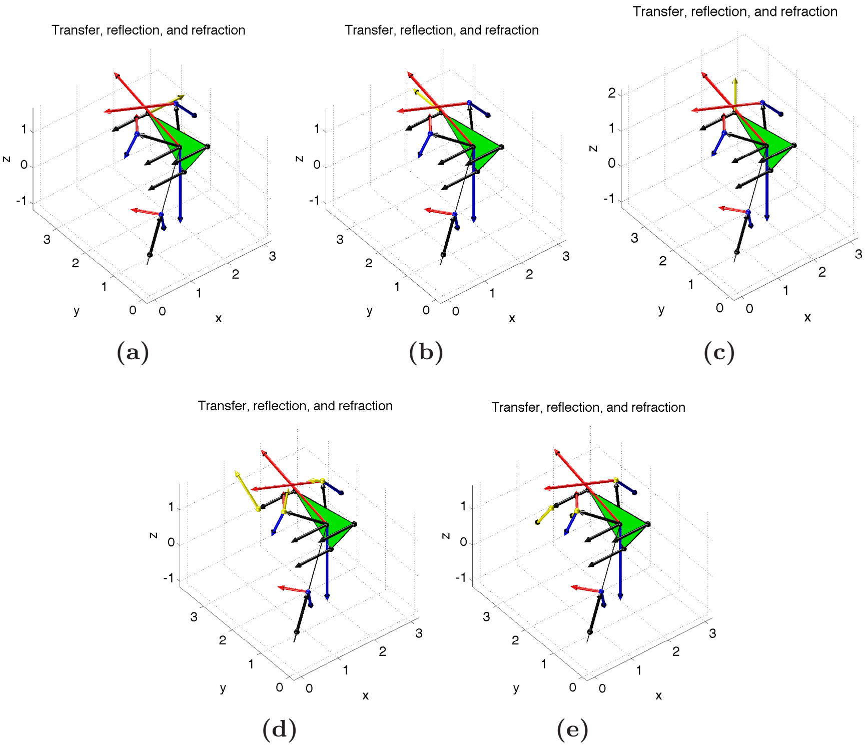
(b)

(c)

(d)

(e)

Figure B.6: Time differentials for Phong surfaces. Yellow arrows denote imposed and resulting time derivatives. Subfigures (a)-(c) shows imposed velocities in three orthogonal directions on the origin, $\boldsymbol{P}$, (d)-(e) shows imposed rotational velocities in viewing direction $\boldsymbol{V}$.



(a) (b) (c) (d) (e)

Figure B.7: Time differentials for Phong surfaces. Yellow arrows denote imposed and resulting time derivatives. Subfigures (a)-(c) shows imposed velocities in three orthogonal directions on one of the vertices, and (d)-(e) shows imposed rotational velocities the same vertex normal.

Bibliography

Bala, K. and Dutre, P., Eds. 2005. *Ray Maps for Global Illumination.* Eurographics Association, Konstanz, Germany.

Brox, T., Weickert, J., Burgeth, B., and Mrázek, P. 2004. Nonlinear structure tensors. Tech. rep., Saarland University, Department of Mathematics.

Cadík, M. and Slavík, P. 2004. Evaluation of two principal approaches to objective image quality assessment. In *In Proceedings of the 8th International Conference on Information Visualisation. IEEE Computer Society.* IEEE Computer Society, 513–518.

Cammarano, M. and Jensen, H. W. 2002. Time dependent photon mapping. In *EGRW '02: Proceedings of the 13th Eurographics workshop on Rendering.* Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 135–144.

Catté, F., Lions, P.-L., Morel, J.-M., and Coll, T. 1992. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM J. Numer. Anal. 29,* 1, 182–193.

Chandrasekhar, S. 1950. *Radiative Transfer.* Oxford, Clarendon Press. Unabridged and slightly revised version published by Dover Publications, Inc., in 1960.

Christensen, P. H., Laur, D. M., Fong, J., Wooten, W. L., and Batali, D. 2003. Ray differentials and multiresolution geometry caching for distribution ray tracing in complex scenes. In *Proceedings of Eurographics 2003.* Computer Graphics Forum. Blackwell Publishing Inc, 543–552.

Collins, S. 1994. Adaptive splatting for specular to diffuse light transport. In *Proceedings of the 5th Eurographics Workshop on Rendering*, B. Peroche and H. Rushmeier, Eds. Springer, Darmstadt, Germany, 119–135.

Cook, R. L., Porter, T., and Carpenter, L. 1984. Distributed ray tracing. *SIGGRAPH Comput. Graphics 18,* 3, 137–145.

Crank, J. 2004. *Mathematics of Diffusion.* Oxford university press, Oxford, Great Britain.

Girod, B. 1993. *What's wrong with mean-squared error?* MIT Press, Cambridge, MA, USA.

Haeberli, P. and Akeley, K. 1990. The accumulation buffer: hardware support for high-quality rendering. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques.* ACM, New York, NY, USA, 309–318.

Heckbert, P. S. 1990. Adaptive radiosity textures for bidirectional ray tracing. In *Computer Graphics*. ACM Siggraph Conference proceedings, 145–154.

Herzog, R., Havran, V., Kinuwaki, S., Myszkowski, K., and Seidel, H.-P. 2007a. Global Illumination using Photon Ray Splatting. Tech. rep., Max-Planck-Institut für Informatik, Saarbrücken, Germany. May.

Herzog, R., Havran, V., Kinuwaki, S., Myszkowski, K., and Seidel, H.-P. 2007b. Global illumination using photon ray splatting. In *The European Association for Computer Graphics 28th Annual Conference: EUROGRAPHICS 2007*, D. Cohen-Or and P. Slavik, Eds. Computer Graphics Forum, vol. 26(3). The European Association for Computer Graphics, Blackwell, Prague, Czech Republic, 503–513.

Igehy, H. 1999. Tracing ray differential. In *Siggraph 1999, Computer Graphics Proceedings*, A. Rockwood, Ed. Addison Wesley Longman, Los Angeles, 179–186.

JAROSZ, W., ZWICKER, M., AND JENSEN, H. W. 2008. The Beam Radiance Estimate for Volumetric Photon Mapping. *Computer Graphics Forum (Proc. Eurographics EG'08) 27,* 2 (4), 557–566.

JENSEN, H. W. 1996. The photon map in global illumination. Ph.D. thesis, Technical University of Denmark, Lyngby.

JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping.* A. K. Peters, Ltd., Natick, MA, USA.

JENSEN, H. W. AND CHRISTENSEN, N. J. 1995. Photon maps in bidirectional monte carlo ray tracing of complex objects. *Computers & Graphics 19,* 2 (Mar), 215–224.

JENSEN, H. W. AND CHRISTENSEN, P. H. 1998. Efficient simulation of light transport in scences with participating media using photon maps. In *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques.* ACM, New York, NY, USA, 311–320.

KAJIYA, J. T. 1986. The rendering equation. In *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques.* ACM, New York, NY, USA, 143–150.

KANTERS, F. M. W. 2007. Towards object-based image editing. Ph.D. thesis, Eindhoven Technical University, Eindhoven, Netherlands.

LAVIGNOTTE, F. AND PAULIN, M. 2003. Scalable photon splatting for global illumination. In *GRAPHITE '03: Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia.* ACM, New York, NY, USA, 203–ff.

LINDEBERG, T. 1996. Scale-space: A framework for handling image structures at multiple scales. In *Proceedings. CERN School of Computing.* Egmond aan Zee, Netherland.

MAGNUS, J. R. AND NEUDECKER, H. 1988. *Matrix Differential Calculus with Applications in Statistics and Econometrics.* John Wiley & Sons.

MCCOOL, M. D. 1999. Anisotropic diffusion for monte carlo noise reduction. *ACM Transactions on Graphics 18,* 2, 171–194.

MÖLLER, T. AND TRUMBORE, B. 1997. Fast, minimum storage ray-triangle intersection. *journal of graphics tools 2,* 1, 21–28.

MRÁZEK, P. 2001. Nonlinear diffusion for image filtering and monotonicity enhancement. Ph.D. thesis, Czech Technical University in Prague.

MYSZKOWSKI, K. 1997. Lighting reconstruction using fast and adaptive density estimation techniques. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97.* Springer-Verlag, London, UK, 251–262.

MYSZKOWSKI, K., TAWARA, T., AKAMINE, H., AND SEIDEL, H.-P. 2001. Perception-guided global illumination solution for animation rendering. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques.* ACM, New York, NY, USA, 221–230.

NICODEMUS, F. E., RICHMOND, J. C., HSIA, J. J., GINSBERG, I. W., AND LIMPERIS, T. 1977. Geometrical considerations and nomenclature for reflectance. Tech. rep., National Bureau of Standards (US). Oct.

PATTANAIK, S. N. AND MUDUR, S. P. 1993. Computation of global illumination in a participating medium by monte carlo simulation. *The Journal of Visualization and Computer Animation 4,* 3 (July/Sept.), 133–152.

PERONA, P. AND MALIK, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-12,* 7 (July), 629–639.

REDNER, R. A., LEE, M. E., AND USELTON, S. P. 1995. Smooth b-spline illumination maps for bidirectional ray tracing. *ACM Transactions on Graphics 14,* 4, 337–362.

Schjøth, L., Frisvad, J. R., Erleben, K., and Sporring, J. 2007. Photon differentials. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*. ACM, New York, NY, USA, 179–186.

Schjøth, L., Frisvad, J. R., Erleben, K., and Sporring, J. 2009. Anisotropic density estimation in global illumination. *ACM Transactions on Graphics ?* Submitted.

Schjøth, L., Olsen, O. F., and Sporring, J. 2006. Diffusion based photon mapping. In *International conference on Computer Graphics – Theory and Applications*. INSTICC Press, Setúbal, Portugal, 168–175.

Schjøth, L., Olsen, O. F., and Sporring, J. 2007. *Advances in Computer Graphics and Computer Vision*. Communications in Computer and Information Science, vol. 4. Springer, Berlin, Germany, 109–122.

Schjøth, L., Olsen, O. F., and Sporring, J. 2008. Diffusion based photon mapping. *Computer Graphics Forum 27,* 8 (December), 2114–2127.

Schregle, R. 2003. Bias compensation for photon maps. *Computer Graphics Forum 22,* 4, 729–742.

Shirley, P., Wade, B., Hubbard, P. M., Zareski, D., Walter, B., and Greenberg, D. P. 1995. Global Illumination via Density Estimation. In *Proceedings of the Sixth Eurographics Workshop on Rendering*. Springer-Verlag, New York, NY, 219–230.

Siegel, R. and Howell, J. R. 2002. *Thermal Radiation Heat Transfer*, fourth ed. Taylor & Francis, New York.

Silverman, B. 1986. *Density Estimation for Statistics and Data Analysis*. Monographs on Statistics and Applied Probability. Chapman and Hall, London-New York.

Simonoff, J. S. 1996. *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer-Verlag, New York.

Sporring, J., Schjøth, L., and Erleben, K. 2009. Space and Time Ray Differentials. Tech. rep., Institute of Computer Science, University of Copenhagen, Copenhagen, Denmark. September. To appear.

Stürzlinger, W. and Bastos, R. 1997. Interactive rendering of globally illuminated glossy scenes. In *Proceedings of the Eurographics Workshop on Rendering Techniques '97*. Springer-Verlag, London, UK, 93–102.

Suykens, F. and Willems, Y. D. 2000. Density control for photon maps. In *Proceedings of the 11th Eurographics Workshop on Rendering*. Springer-Verlag, London, UK, 23–34.

Suykens, F. and Willems, Y. D. 2001. Path differentials and applications. In *Proceedings of the 12th Eurographics Workshop on Rendering*. Springer-Verlag, London, UK, 257–268.

Teo, P. C. and Heeger, D. J. 1994. Perceptual image distortion. In *in Proc. SPIE*. 982–986.

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

Walter, B. 1998. Density estimation techniques for global illumination. Ph.D. thesis, Cornell University.

Walter, B., Hubbard, P. M., Shirley, P., and Greenberg, D. P. 1997. Global illumination using local linear density estimation. *ACM Transactions on Graphics 16,* 3, 217–259.

Wand, M. P. and Jones, M. C. 1995. *Kernel Smoothing (Monographs on Statistics and Applied Probability)*. Chapman & Hall/CRC.

Wang, Z. and Bovik, A. C. 2002. Why is image quality assessment so difficult. In *in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*. 3313–3316.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing 13*, 600–612.

Weickert, J. 1995. Multiscale texture enhancement. *Lecture Notes in Computer Science 970*, 230–237.

Weickert, J. 1998. *Anisotropic Diffusion in Image Processing*. B. G. Teubner, Stuttgart, Germany.

Weickert, J. 1999. Coherence-enhancing diffusion filtering. *International Journal of Computer Vision 31,* 2-3, 111–127.

Yves, E. L. and Willems, Y. D. 1993. Bi-directional path tracing. In *Proceedings of Third International Conference on Computational Graphics and Visualization Techniques (Computer graphics 1993*. 145–153.