

Photon Differentials

Lars Schjøth*
Department of Computer Science
University of Copenhagen

Jeppe Revall Frisvad†
Informatics and Mathematical Modelling
Technical University of Denmark

Kenny Erleben‡, Jon Sporring§
Department of Computer Science
University of Copenhagen

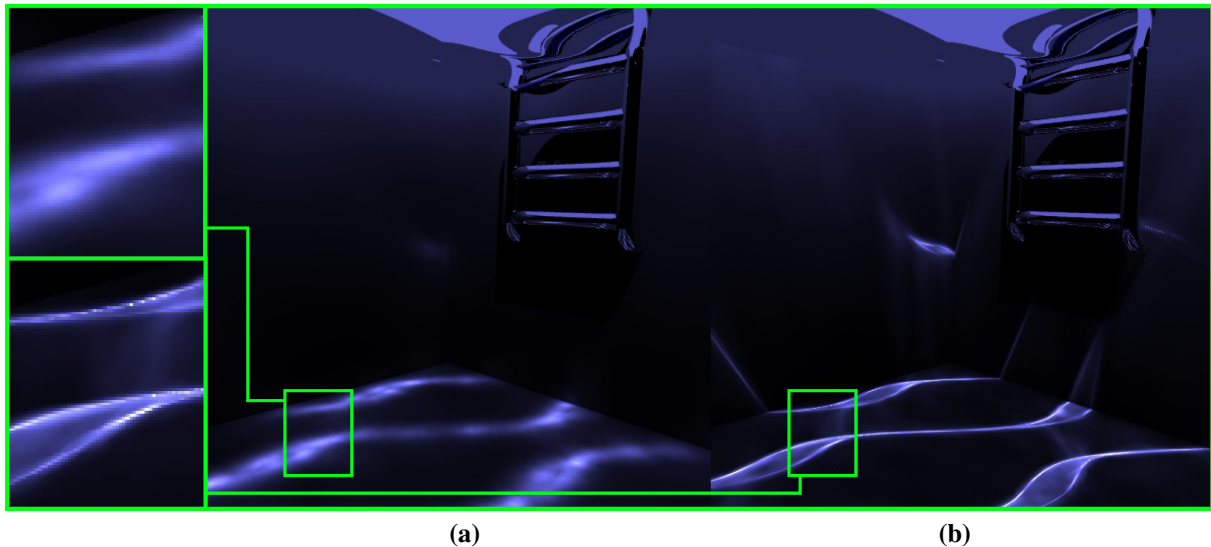


Figure 1: Underwater view of a swimming pool. Image (a) rendered using regular photon mapping and (b) using our method.

Abstract

A number of popular global illumination algorithms uses density estimation to approximate indirect illumination. The density estimate is performed on finite points - particles - generated by a stochastic sampling of the scene. In the course of the sampling, particles, representing light, are stochastically emitted from the light sources and reflected around the scene. The sampling induces noise, which in turn is handled by the density estimate during the illumination reconstruction. Unfortunately, this noise reduction imposes a systematic error (bias), which is seen as a blurring of prominent illumination features. This is often not desirable as these may lose clarity or vanish altogether.

We present an accurate method for reconstruction of indirect illumination with photon mapping. Instead of reconstructing illumination using classic density estimation on finite points, we use the correlation of light footprints, created by using Ray Differentials during the light pass. This procedure gives a high illumination accuracy, improving the trade-off between bias and variance considerable as compared to traditional particle tracing algorithms. In this way we preserve structures in indirect illumination.

*e-mail: schj@diku.dk

†e-mail: jrf@imm.dtu.dk

‡e-mail: kenny@diku.dk

§e-mail: sporring@diku.dk

Copyright © 2007 by the Association for Computing Machinery, Inc. Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions Dept, ACM Inc., fax +1 (212) 869-0481 or e-mail permissions@acm.org.

GRAPHITE 2007, Perth, Western Australia, December 1–4, 2007.

© 2007 ACM 978-1-59593-912-8/07/0012 \$5.00

CR Categories: I.3.7 [Computer Graphics]: Three-dimensional Graphics and Realism—;

Keywords: Ray-Tracing, Global Illumination, Photon Mapping, Caustics, Ray Differentials

1 Introduction

Monte Carlo sampling is used in a number of global illumination algorithms. The stochastic nature of Monte Carlo sampling induces variance, which 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 systematic error (bias). Bias is noticeable as a blurring of the illumination. 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.

In this paper we present a method that enhances edges and structures of prominent illumination features, improving the trade-off between variance and bias. We have implemented this method in photon mapping. Photon mapping is a popular particle tracing algorithm developed by Jensen [1995; 1996].

Particle tracing algorithms usually employ two steps. A first step in which particles 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 particles traced during the first step have neighbors which tend to follow the same path. We exploit this coherence by

tracing imaginary bundles of particles along each trace. Each of our particles represents a beam of light which expands, contracts and reshapes, according to the reflections and refraction it undergoes as it propagates through the scene. This is achieved using Ray Differentials.

Ray differentials is a technique introduced by Igehy [1999], which 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.

Figure 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 more fine details than regular photon mapping.

2 Related works

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

Photon mapping usually depends on k 'th nearest neighbor kernel estimate to improve the trade-off between bias and variance. However, in [1995] Jensen 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 increase the bandwidth iteratively estimating the radiance in each step. If new estimates differ more from previous than what can be contributed variance, then the iteration stops, as the difference is then 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 and 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 function 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. They called the

algorithm *the density estimation framework*, and they refined it in a series of papers.

Bias control was not an issue 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 Walter 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, which 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 these algorithms mainly lie in their method of detecting structure and the degree of change in bandwidth a given proximity of such structure entails. They use rotation invariant kernels.

In contrast to Myszkowsky, Schregle, and Walter's approach our method will smooth along edges and structures, it follows that its support will not be limited to the same degree in the proximity of these.

Recently, Schjøth *et al.* [2006] suggested a bias reducing method inspired by diffusion filtering. Their method uses 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.

While Schjøth *et al.* also use a rotationally variant kernel, their kernels affects a number of photons based on an average of the local structure. The method proposed in this paper uses a 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 our method is more accurate and will be able to handle corners more correctly. Furthermore, our algorithm does not include an extra pass.

3 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 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 effects, unless an excessive number of photons are stored in the photon map. This is the mentioned

trade-off problem between variance versus bias as it manifests itself in photon mapping.

3.1 The radiance estimate

In his book [2001] Jensen derives an equation which 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

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

where Φ_i is the radiant power represented by the i 'th photon, f_r is the *bidirectional reflectance distribution function* (abbreviated BRDF), and $r(\mathbf{x})$ is the radius of a sphere encompassing the k nearest photons, such that $\pi r(\mathbf{x})^2$ 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 $r(\mathbf{x})$ is called the bandwidth.

The bandwidth is important, because its size controls the trade-off 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 (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 weigh 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

$$\widehat{L}_r(\mathbf{x}, \omega) = \frac{1}{r(\mathbf{x})^2} \sum_{i=1}^k K\left(\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{r(\mathbf{x})^2}\right) f_r(\mathbf{x}, \omega_i, \omega) \Phi_i, \quad (2)$$

where \mathbf{x}_i is the position of the i 'th photon and $K(\mathbf{y})$ is a function that weights the photons according to their distance from \mathbf{x} . This function should be symmetric around \mathbf{x} , and it should be normalized such that it integrates to unity within the distance $r(\mathbf{x})$ to \mathbf{x} . In density estimation $K(\mathbf{y})$ is known as the kernel function. Usually, the kernel function decreases monotonically, weighting photons near \mathbf{x} higher than those farther away. In this way the kernel function reduce bias, where the change in density is significant.

In his PhD thesis [1996] Jensen presents the *cone filter*. This filter is used to reduce bias, such that edges and structure in the illumination are less blurred. As a kernel in the general radiance estimate the cone filter has the following form

$$K(\mathbf{y}) = \begin{cases} K(\mathbf{y}) = \frac{1 - \frac{\sqrt{|\mathbf{y}|}}{g}}{(1 - \frac{2}{3g})\pi} & \text{if } \sqrt{|\mathbf{y}|} < 1, \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $g \geq 1$ is a constant which controls the steepness of the filter slope.

Another useful kernel is the Epanechnikov kernel. The Epanechnikov kernel is known from statistics for its ability to reduce the mean integrated square error of kernel density estimation, but it is furthermore popular because it is computationally inexpensive [Silverman 1986]. In computer graphics, Walter [1998] has employed it with good results and Schregle [2003] has in his bias case shown its bias reducing properties to be competitive. In 2D the Epanechnikov kernel is given by

$$K(\mathbf{y}) = \begin{cases} \frac{2}{\pi}(1 - \mathbf{y}) & \text{if } \mathbf{y} < 1, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

In this paper we use the Epanechnikov kernel to examine our proposed method.

4 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 multiresolution caching of geometry and textures.

Here follows a short introduction to ray differentials as presented by Igehy.

A parameterized ray, \mathbf{r} , is defined by its position in space, \mathbf{p} , and its direction, \mathbf{d} . The differential of a ray is then the partial derivative of its position and direction with respect to some initial offset:

$$\frac{\partial \mathbf{r}}{\partial u} = \left(\frac{\partial \mathbf{p}}{\partial u}, \frac{\partial \mathbf{d}}{\partial u} \right) \quad (5)$$

These 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 beams spread, while the positional derivatives describes 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 $\partial_u \mathbf{p}$ to denote a positional differential vector and $\partial_u \mathbf{d}$ 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 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 Igehy [1999] for the derivatives of these operations.

5 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.

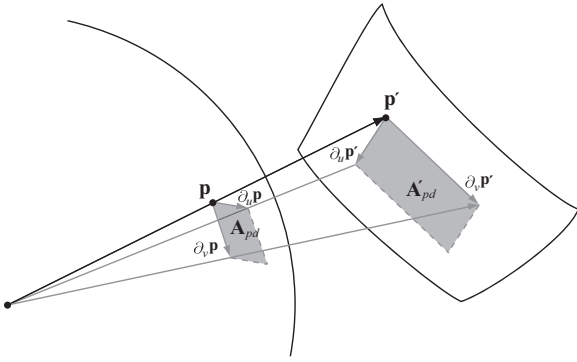


Figure 2: Transfer of a ray and its differential vectors from \mathbf{p} to \mathbf{p}' .

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 caustic 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 address some of the issues concerned with photon differentials.

5.1 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

$$M_{total} = \frac{\Phi_l}{4\pi}. \quad (6)$$

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

$$\Phi_{pd} = A_{pd}M_{total} \quad (7)$$

$$= \frac{\Phi_l}{n_{pd}}. \quad (8)$$

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 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

$$E_{pd} = \Phi_{pd}/A'_{pd}. \quad (9)$$

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

$$M_{total} = \frac{\Phi_l}{A_l}. \quad (10)$$

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

$$\Phi_{pd} = A_{pd}M_{total} = \frac{\Phi_l}{n_{pd}}. \quad (11)$$

The irradiance due to a photon differential incident on a diffuse surface is then found exactly as in (9) (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 Lighting reconstruction

Irradiance is radiant power incident per unit area at a point \mathbf{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

$$E(\mathbf{x}, \omega) = \frac{d\Phi(\mathbf{x}, \omega)}{dA}. \quad (12)$$

By the definition of radiance it follows that

$$L_i(\mathbf{x}, \omega) = \frac{d^2\Phi(\mathbf{x}, \omega)}{(\mathbf{n}_x \cdot \omega)d\omega dA} = \frac{dE(\mathbf{x}, \omega)}{(\mathbf{n}_x \cdot \omega)d\omega} \quad (13)$$

or, in other words,

$$L_i(\mathbf{x}, \omega)(\mathbf{n}_x \cdot \omega)d\omega = dE(\mathbf{x}, \omega). \quad (14)$$

Then the reflected radiance at \mathbf{x} in direction ω is [Nicodemus et al. 1977]

$$L_r(\mathbf{x}, \omega) = \int_{\Omega_x} f_r(\mathbf{x}, \omega', \omega)L_i(\mathbf{x}, \omega')(\mathbf{n}_x \cdot \omega')d\omega' \quad (15)$$

$$= \int_{\Omega_x} f_r(\mathbf{x}, \omega', \omega)dE(\mathbf{x}, \omega'). \quad (16)$$

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, se (9). We have

$$L_r(\mathbf{x}, \omega) \approx \widehat{L}_r(\mathbf{x}, \omega) = \sum_{pd=1}^n f_r(\mathbf{x}, \omega_{pd}, \omega)\Delta E_{pd}(\mathbf{x}, \omega_{pd}), \quad (17)$$

where n is the number of photon differentials whose footprints overlap \mathbf{x} . When finding the overlap, the footprint of each photon differential is centered around the intersection point.

In practice n is found by collecting the photon differentials nearest \mathbf{x} , 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.

5.3 Kernel smoothing

Equation 17 provides no smoothing when estimating the illumination. To provide kernel smoothing we reformulate the equation such that

$$\hat{L}_r(\mathbf{x}, \omega) = \sum_{pd=1}^n f_r(\mathbf{x}, \omega_{pd}, \omega) K(\mathbf{M}_{pd}(\mathbf{x}_{pd} - \mathbf{x})) \Delta E_{pd}(\mathbf{x}, \omega_{pd}), \quad (18)$$

where K , is the kernel function as described in Section 3.1 and \mathbf{M}_{pd} , is the matrix which transforms from world coordinates into a coordinate system with the surface normal and the positional differential vectors $\partial_u \mathbf{p}$ and $\partial_v \mathbf{p}$ as basis vectors. This transformation is illustrated in Figure 3. We use half the length of the differential vectors, as we center the footprint around the photon differentials intersection point, \mathbf{x}_{pd} .

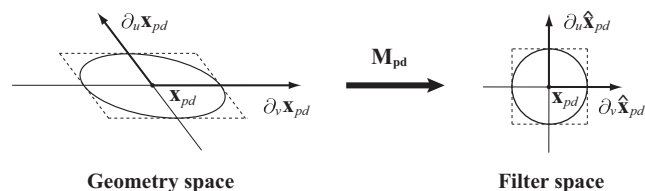


Figure 3: 2D illustration of a transformation from geometry space to filter space by the matrix \mathbf{M}_{pd} . The ellipse inside the parallelogram is the footprint of the photon differential. When transformed into filter space the ellipse becomes a unit circle.

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 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.

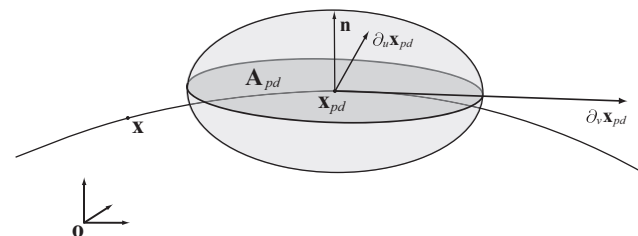


Figure 4: Filter kernel of a photon differential illustrated as an ellipsoid. The size and shape of the kernel is defined by the surface normal of the intersected object and the positional differential vectors of the photon differential.

In the context of density estimation the radiance estimate, (18), is a variable-bandwidth kernel estimator, [Simonoff 1996]. 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.

5.4 Implementation

When emitting photon differentials from a light source, the initial size of the photon differential's footprint is a smoothing factor which 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 while 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 the same. 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.

The transformation matrix, \mathbf{M}_{pd} , is a 3 by 3 matrix. It is either be stored along with the photon differentials demanding an additional 40¹ 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.

6 Results

In this section we present a case study, exploring the diacaustic of a simple water wave. This simple case is used to compare our method with regular photon mapping. Following this we present a more advanced caustic and finally the time results for the renderings are examined.

6.1 Case study

The purpose of this case study is to corroborate the ability of photon differentials to reproduce fine structures in indirect lighting. The case presented is that of a diacaustic created by refraction of light through a simple sinusoidal shaped water wave. Although sinusoidal waves are not the most accurate model for simulating real world 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.

Figure 5 is a diagram of the case study. It illustrates the interaction between a finite number of collimated light rays² and a sinusoidal 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 indices of the media in which the rays traverse. In this case light rays create two bright points focus 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.

¹36 bytes for the matrix and 4 bytes for the area of the footprint.

²Approximating sunlight.

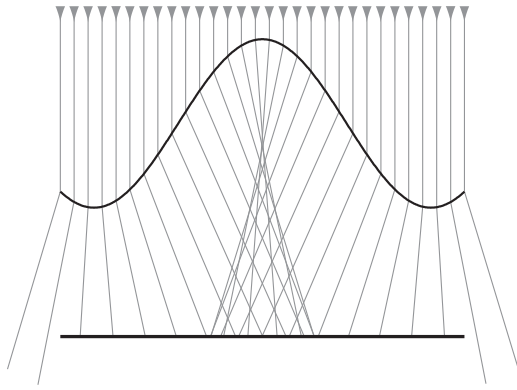


Figure 5: Diagram of the case study scene. Illustrates a ray tracing of a sinusoidal shaped water wave illuminated from above by collimated light.

Figure 6 is a rendering of the case study. The image was rendered with photon differentials using 20 000 photons. The rendering mirrors 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.

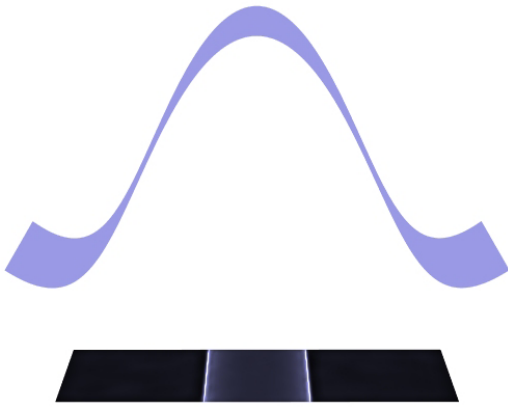


Figure 6: Rendering of the case study scene. Rendered with photon differentials using 20 000 photons.

Now, to approximate the diacaustic 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 kernel. If the bandwidth at the same time goes to zero, then our estimate will converge to f . Adjusting the bandwidth controls the trade-off 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. In the following we will show by example that our method converges much faster than regular photon mapping.

As a reference image the scene has been rendered with 1 million photons using regular photon mapping. The camera has been positioned as to solely capture the caustic. Figure 7a shows this rendering, while Figure 7b reproduces part of the used photon distribution³.

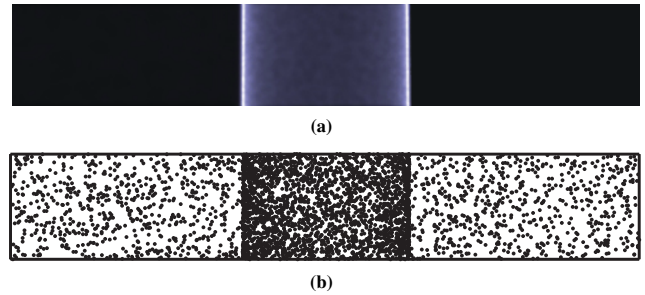


Figure 7: Case study - reference images. (a) is a rendering of the caustic seen in the case study scene, Figure 6. The image was created with k 'th nearest neighbor photon mapping using a photon map containing 1 million photons. (b) visualizes part of the photon distribution used to create (a). Only 2 000 photons are shown as to facilitate visualization.

The image series in Figure 8 are renderings of the case study scene in Figure 6, with a camera placement similar to that in the reference image, Figure 7. The images in the first column have been produced using regular k 'th nearest neighbor photon mapping, while the second column has been produced with our method. Both methods uses the Epanechnikov kernel. The bandwidth for all renderings has been adjusted as to best avoid noise without blurring unnecessarily. Each row is based on different numbers of photons in the photon map. First row has been rendered using 500 photons, second using 2 000 photons and the last row has been rendered using 20 000 photons.

Inspecting the images in Figure 8, we see that the image quality for both methods improve as more photons are included in the photon map. The focal lines become more pronounced and the edges becomes sharper and less blurry. The difference between the two methods is noticeable. Using 500 photons, regular photon mapping reproduces the focal lines as oblong blobs, in contrast our method shows the focus lines as distinct lines. At 20 000, the image produced by our method is comparable to the reference image, Figure 7a.

Furthermore, Figure 8b indicates that even when very few photons are used, the structure of the caustic is still clearly identifiable. This assertion is substantiated by the images in Figure 9.

The images in Figure 9 are rendered using photon differential on a photon map with extremely few photons. In the images the focus lines are partly flawed, however, parts of the lines are still detectable and these are of a quality comparable to Figure 8b.

The quality that the structure of caustics is identifiable even at low photon counts makes photon differentials useful in post-production as comprehensible images can be obtained at low computational cost.

6.2 Squashed torus

Figure 10 illustrates a more advanced caustic. It is created by lights reflection of a mashed-in torus. The caustic, resembling the number

³Only 2 000 photons are shown as to improve visualization



Figure 8: Comparison study. Renderings of the study case scene. First column rendered with regular k 'th nearest neighbor photon mapping, while second column was rendered with photon differentials. First row was rendered using 500 photons, second row using 2 000 photons and third row using 20 000 photons.



Figure 9: Case study - few photons. Both images was rendered using photon differentials. Image (a) was rendered using 50 photons and (b) using 100 photons.

8, is much more detailed in the rendering using photon differentials than the one using regular photon mapping. Notice that the caustics caught on the back wall in Figure 10,b are not seen in Figure 10,a where they are blurred out by the density estimate of regular photon mapping. Both images was rendered with a photon map containing 20 000 photons. Furthermore, the bandwidth of both renderings was adjusted as to best visualize the caustic.

6.3 Timing results

Scene	Method	Figure	Photons in map	Photon tracing [s]	Rendering [s]
Pool	Regular	1a	10,000	0.9	215.02
	Differentials	1b	10,000	1.3	242.58
Case study	Regular	8a	500	0.01	1.86
		8c	2,000	0.02	2.05
	Differentials	8e	20,000	0.14	4.16
		8b	500	0.01	2.63
	Differentials	8d	2,000	0.03	3.43
		8f	20,000	0.21	4.93
Torus	Regular	10a	20,000	0.49	104.02
	Differentials	10b	20,000	0.67	126.91

Table 1: Performance results for the images in Figure 1, 8 and 10. Times are in seconds.

The photon tracing times for our method are larger than for standard photon mapping, see Table 1. If the scene complexity or the number of photons traced goes up this difference will increase. The increase in difference will, however, stay linear as both algorithms have the same computational complexity. Likewise, the computational complexity of the radiance estimates of the two methods are equal. Our method only has a larger constant.

7 Conclusion

In this paper we investigated the use of ray differentials for lighting reconstruction in photon mapping. First we gave an account of the method in which we explored light source sampling and light reconstruction using photon differentials. Then we examined the method using a simple case study of a diacaustic created by a single sinusoidal shaped water wave, then a more advanced caustics was explored and finally we presented the timing results for the renderings presented in this paper.

From this examination we conclude that with a map of photon differentials one obtains better caustic quality with far less photons. 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 to generate caustics of the quality needed, for example, for animation feature films. In a large scene the memory footprint of the needed photons might cause significant problems. With photon differentials high quality caustics are simply much easier to obtain.

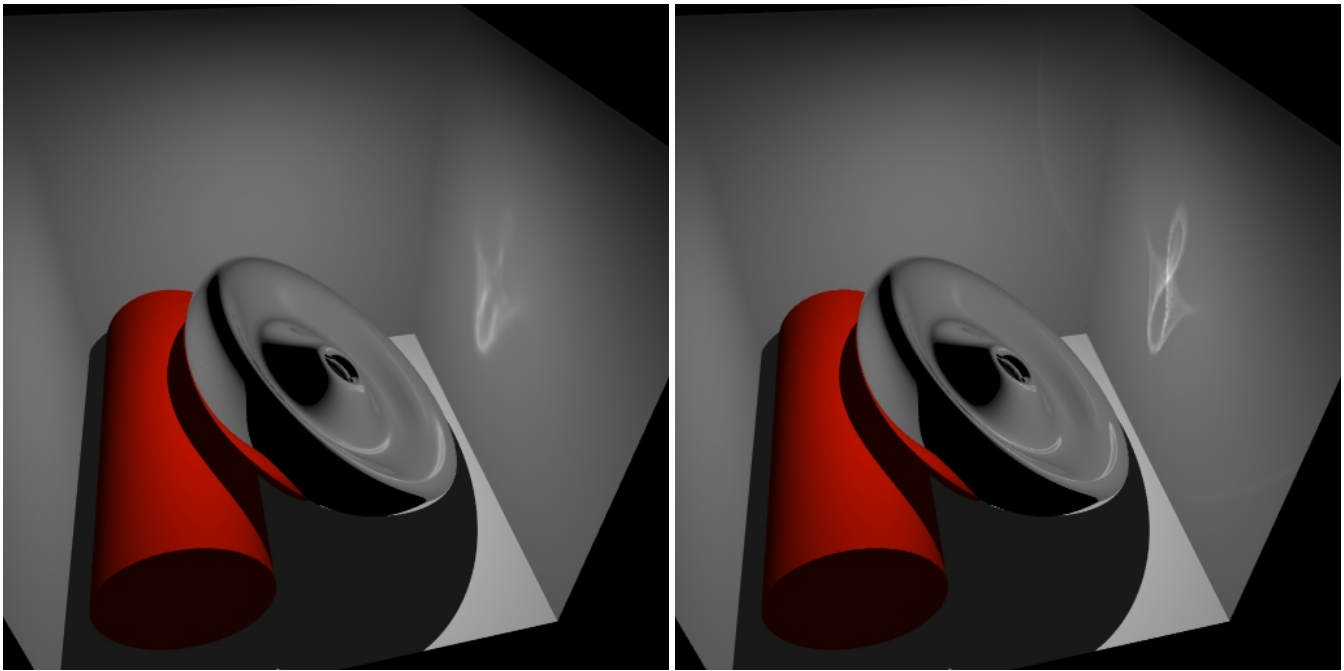
It is furthermore worth to mention that our method is a useful tool in post-production as photon differentials reveals the structure of caustics when using even a very low number of photons.

8 acknowledgment

Thanks to Bent Dalgaard Larsen for inspiration and sparring time.

References

CHRISTENSEN, P. H., LAUR, D. M., FONG, J., WOOTEN, W. L., AND BATALI, D. 2003. Ray differentials and multiresolution



(a)

(b)

Figure 10: Renderings containing a caustic created from lights reflection of a squashed metal torus. Image (a) was rendered with regular k 'th nearest neighbor photon mapping, while image (b) was rendered with photon differentials. Both images was rendered with photon map containing 20,000 photons.

- geometry caching for distribution ray tracing in complex scenes. In *Proceedings of Eurographics 2003*, Blackwell Publishing Inc, Computer Graphics Forum, 543–552.
- HECKBERT, P. S. 1990. Adaptive radiosity textures for bidirectional ray tracing. In *Computer Graphics*, ACM Siggraph Conference proceedings, 145–154.
- IGEY, H. 1999. Tracing ray differential. In *Siggraph 1999, Computer Graphics Proceedings*, Addison Wesley Longman, Los Angeles, A. Rockwood, Ed., 179–186.
- 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. 1996. *The Photon Map in Global Illumination*. PhD thesis, Technical University of Denmark, Lyngby.
- JENSEN, H. W. 2001. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA.
- 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.
- 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.
- REDNER, R. A., LEE, M. E., AND USELTON, S. P. 1995. Smooth b-spline illumination maps for bidirectional ray tracing. *ACM Trans. Graph.* 14, 4, 337–362.
- 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.
- 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.
- 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.
- 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.
- WALTER, B., HUBBARD, P. M., SHIRLEY, P., AND GREENBERG, D. P. 1997. Global illumination using local linear density estimation. *ACM Trans. Graph.* 16, 3, 217–259.
- WALTER, B. 1998. *Density estimation techniques for global illumination*. PhD thesis, Cornell University.