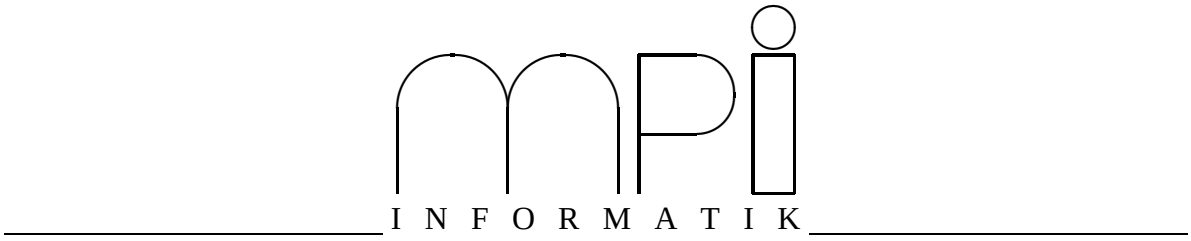

A custom designed density estimator for light transport

Ph. Bekaert[1,4], Ph. Slussalek[2], R. Cools[3], V. Havran[1], H.-P. Seidel[1]

MPI–I–2003–4–004 April 2003

FORSCHUNGSBERICHT RESEARCH REPORT

MAX-PLANCK-INSTITUT
FÜR
INFORMATIK

Stuhlsatzenhausweg 85 66123 Saarbrücken Germany

Author's Address

(1) Max-Planck-Institut für Informatik
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany
(2) Universität des Saarlandes
Fachrichtung 6.2 - Informatik
Im Stadtwald - Geb. 36.1, Raum 018
66123 Saarbrcken
Germany
(3) Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan, 200A
3000 Leuven
Belgium
(4) Current address: Expertise Center for Digital Media
University of Limburg
Wetenschapspark, 2
3590 Diepenbeek
Belgium
Philippe.Bekaert@luc.ac.be
slusallek@cs.uni-sb.de
Ronald.Cools@cs.kuleuven.ac.be
havran@mpi-sb.mpg.de
hpseidel@mpi-sb.mpg.de

Abstract

We present a new Monte Carlo method for solving the global illumination problem in environments with general geometry descriptions and light emission and scattering properties. Current Monte Carlo global illumination algorithms are based on generic density estimation techniques that do not take into account any knowledge about the nature of the data points — light and potential particle hit points — from which a global illumination solution is to be reconstructed. We propose a novel estimator, especially designed for solving linear integral equations such as the rendering equation. The resulting single-pass global illumination algorithm promises to combine the flexibility and robustness of bi-directional path tracing with the efficiency of algorithms such as photon mapping.

Keywords

Three-Dimensional Graphics and Realism; Color, Shading, Shadowing and Texture; Ray Tracing; Monte Carlo methods; Density Estimation

1 Introduction and Previous Work

The global illumination problem is a long-standing problem in computer graphics and a huge amount of research has been spent on it already. The only way known today to reliably solve the problem in a general setting is by means of Monte Carlo methods, inherited from related fields such as neutron transport [16, 9]. One can distinguish two classes of Monte Carlo methods for global illumination, here called *pixel-based* methods and *surface-based* methods.

Pixel-based methods Pixel-based methods compute the light flux through every virtual screen pixel directly. They avoid pre-computing and storing an approximation to the illumination on surfaces in a scene to be rendered. Because of this, they are very flexible and can handle a wide variety of geometry and light emission and scattering descriptions. Stochastic ray tracing [4, 8] and bi-directional path tracing [20, 10] are algorithms belonging to this class.

Bi-directional path tracing is a generalization of stochastic ray tracing and light path tracing. For each image pixel, a pair of random walks are traced, one from the observer position and one from a light source in the scene. The vertices of the light and eye path in the pair are connected by means of shadow rays. In this way, multiple alternative Monte Carlo estimators for direct illumination, first-order indirect illumination, second order indirect illumination etc. . . are obtained. These alternatives are weighed in a way that gives priority to the best available strategy for calculating any light transport path.

Unfortunately, bi-directional path tracing often leads to unacceptable computation times (hours or even days for complex scenes). The main causes are dense occlusion, the presence of difficult lighting effects such as reflections of caustics and the lack of data coherence in the algorithm.

In densely occluded environments, the probability of obtaining unoccluded connections between a pair of a light and eye path vertices is small. This problem is reduced, but not eliminated, by stochastically accepting or rejecting shadow tests based on the distance between and surface orientation at path vertices [19, §10.4] and by metropolis sampling [21].

In the second case, none of the sampling strategies in bi-directional path tracing is well suited for this kind of lighting effects so that there is no good combination. This results in spike noise that disappears only extremely slowly from the images. It can be translated in a smooth, perceptually less objectionable, bias by using adaptive image-space filtering techniques [17].

In addition, visibility tests are performed in an order requiring to access data almost randomly, thus defeating fast but rather small memory caches. Enforcing and exploiting ray coherence can speed up execution by half an order of magnitude and more [13, 22].

Surface-based methods Surface-based global illumination algorithms on the other hand, do pre-compute and store an approximation for the illumination on the surfaces of a scene. In order to generate an image for a particular viewing position, the stored solution is queried and reconstructed at the surface points that are visible in the view. The principle of Monte Carlo surface-based illumination computations is as follows: first, the trajectory of light particles is simulated according to the light emission and scattering properties of the surfaces in a scene. The resulting particle hit points then form the input data for a density estimation method [15].

In the simplest case, the scene to be rendered is discretized into small surface elements and the number of particle hit points on each element is essentially just counted [12]. Numerous variants exist on this scheme [2]. They correspond with the *histogram method* for density estimation.

Similar algorithms for higher-order polynomial approximations of the radiosity on surface elements [3, 5] are known in density estimation literature as *orthogonal series estimation methods*.

Yet other algorithms place normalized *kernel* functions at the particle hit points [14, 23]. The illumination intensity at an arbitrary surface point is reconstructed by adding up the value at the query point of the kernel functions placed at neighboring particle hit points. Such density estimation methods are called *kernel methods*.

Photon mapping [7] corresponds with *nearest neighbor density estimation*: in order to reconstruct the illumination at an arbitrary surface point, an a-priori fixed number of nearest particle hit points is determined. The reconstructed illumination intensity is essentially the number of nearest particle hits, divided by the projected area of the smallest sphere containing the nearest particles. Since the precomputed illumination is stored independently of the surface geometry of the scene, photon mapping trivially handles curved surfaces, object instancing and procedural geometry descriptions. The high cost of the nearest neighbor queries is an important bottleneck.

All above algorithms rely on *generic* density estimation methods. They do not take into account the fact that the input data consists of particle hit points resulting from a random walk simulation as dictated by the rendering integral equation. Their bias manifests itself in the form of blurred illumination detail, for instance at sharp shadow boundaries, and other disturbing image artefacts such as light and shadow leaks (see figure 1). They are also limited to the computation of diffuse or nearly diffuse illumination in a scene.

In order to avoid image artefacts and to add non-diffuse illumination contributions to an image, a final gather pass is required. In such a final gather pass, the pre-computed illumination solution is used only indirectly. Direct illumination, one bounce of indirect diffuse illumination and all indirect non-diffuse illumina-

tion are computed with stochastic ray tracing. Interpolation of diffuse indirect illumination with irradiance caching [24] is required in order to make final gathering efficient. (The visualization of bright caustics in photon mapping is an exception to this rule.)

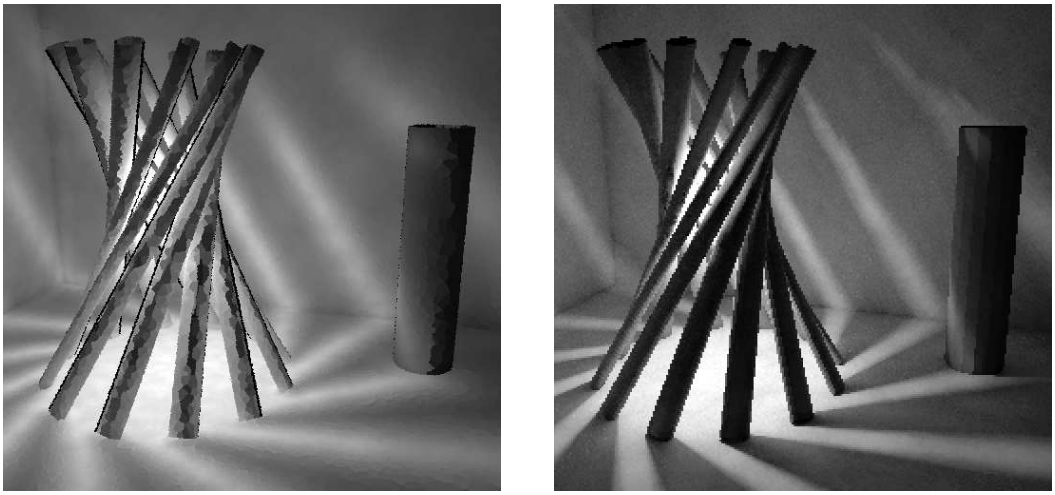

Figure 1: **Bias artefacts.** Both images show only direct illumination. Left: visualization of a photon map without final gather, showing blurred shadows and light leak artefacts; Right: the method proposed in this paper does not result in such disturbing artefacts. It produces convincing images without the need for a separate final gather pass. Computation times are about equal.

Contribution of this paper We present a generalization of the kernel method for density estimation, especially designed for solving integral equations such as the rendering equation [8].

Like bi-directional path tracing (BPT), the algorithm it results in can be viewed as a two-pass algorithm in which both passes are tightly coupled allowing to automatically select the best estimation strategy for a wide range of illumination effects. Unlike BPT, the new algorithm ensures effective connections between light and eye paths, and it automatically generates highly coherent shadow rays. Difficult lighting effects lead to a smooth and perceptually non-objectionable bias, rather than spike noise. Because of this, computation times are in the order of minutes rather than hours or days for highly complex scenes.

Compared with generic density estimation methods, our algorithm faithfully reproduces fine illumination detail like sharp shadows and a wide range of non-diffuse effects without the need for a separate final gather pass (see figure 1). Like photon mapping, it stores illumination information independent of scene geometry, allowing to handle highly complex scenes with object instantiation and

procedural geometry descriptions. Stored illumination queries are however significantly more simple and less costly. In addition, the method proposed in this paper is easy to use as it automatically calibrates its parameters depending on viewpoint and scene. It provides feedback to a user in seconds and converges gracefully until a high quality image is obtained.

This paper proceeds in §2 with a derivation of our generalized kernel method. A practical kernel is proposed next in §3. Some cost reduction techniques follow in §4. Our implementation is described in §5. The paper concludes with some preliminary results in §6.

2 The Generalized Kernel Method

We derive our generalized kernel estimator first in the context of the direct illumination problem. Its extension for indirect illumination is discussed in §2.2

2.1 Direct Illumination

In order to compute the direct illumination $L^d(y_1 \rightarrow y_0)$ at a surface point y_1 in a scene and observed from a position y_0, the following integral needs to be solved:

$$L^d(y_1 \rightarrow y_0) = \int_S L^e(x_0 \rightarrow y_1) G(x_0, y_1) f_r(y_1; x_0 \leftrightarrow y_0) dA_{x_0}. \qquad (1)$$

The meaning of the symbols in this paper is summarized in table 1. We discuss briefly how this integral can be estimated using Monte Carlo integration and density estimation methods, before introducing our generalized kernel method.

Monte Carlo integration Equation (1) can be estimated by randomly picking N points x_0 with probability density $p(x_0)$ and computing the sum:

$$\frac{1}{N} \sum_{s=1}^{N} \frac{L^e(x_0^s \rightarrow y_1) G(x_0^s, y_1) f_r(y_1; x_0^s \leftrightarrow y_0)}{p(x_0^s)} \approx L^d(y_1 \rightarrow y_0). \qquad (2)$$

If $p(x_0)$ is non-zero whenever the numerator — the integrand of (1) — is non-zero, the estimates (2) will converge to the true value of $L^d(y_1 \rightarrow y_0)$ as the number of trials N increases. The number of trials needed for achieving a given accuracy with given confidence is proportional to the square root of the variance of the estimator [16, 9]. The variance can be kept low by chosing $p(x_0)$ so that the terms of (2), the outcome of the trials, are much as possible constant. Common choices for $p(x_0)$ are:

$x_0, x_1, \ldots$	Light path vertices: x_0 is on a light source
$y_0, y_1, \ldots$	Eye path vertices: y_0 is at the virtual camera
$L(y_1 \to y_0)$	Radiance at y_1 emitted towards y_0
$L^e, L^d, L^{(2)}$	Self-emitted, direct and first-order indirect radiance
$f_r(y_1; x_0 \leftrightarrow y_0)$	BSDF at y_1 for scattering to/from x_0 and y_0
$\Theta_{y\|x}$	Direction from x to y (read $y\|x$ as "y after x")
$\cos\theta_{y\|x}$	Cosine of angle between $\Theta_{y\|x}$ and the surface normal at x
r_{xy}	Distance between x and y
$\mathrm{vis}(x, y)$	Visibility predicate
$G(x, y)$	Geometric factor $\cos\theta_{x\|y} \cos\theta_{y\|x} / r_{xy}^2 \cdot \mathrm{vis}(x, y)$
$p(x)$	Probability of sampling x: p^L, p^E for light/eye path tracing
$p(x_k \| x_{k-1}, x_{k-2})$	Probability of sampling a path from x_{k-2}, x_{k-1} to x_k
$K(x_k \| x_{k-1}, \ldots, x_0; y_l)$	Generalized kernel value for combining a light path vertex x_k, successor of $x_{k-1}, \ldots$, with an eye path vertex y_l
$\chi_S(x)$	Characteristic function of a point set S: 1 if $x \in S$, 0 otherwise
$F(x_k \| x_{k-1}, \ldots, x_0)$	Footprint region associated with a light path vertex x_k
$R(x_k \| x_{k-1}, \ldots, x_0)$	Radius of the footprint region

Table 1: The main symbols used in this paper.

- BSDF sampling: a direction $\Theta_{x_0|y_1}$ is sampled according to a pdf which is proportional to the BSDF f_r times $\cos\theta_{x_0|y_1}$ and a ray is shot into the chosen direction. The nearest surface point hit by this ray is x_0. Variations in the self-emitted radiance $L^e(x_0 \to y_1)$ are not accounted for and cause potentially high variance;

- Light source sampling: a point x_0 is chosen randomly on a light source, with a probability proportional to $L^e(x_0 \to y_1)$. Evaluation of (2) requires to compute mutual visibility between x_0 and y_1 by tracing a shadow ray. Variance is due to variations in the BSDF and geometric factors (distance, cosines, visibility), not taken into account during sampling.

With multiple importance sampling [20], the two strategies can be combined into a single strategy that will behave well whenever at least one has low variance.

In graphics, one typically needs to compute not a single integral (1), but hundreds of thousands of those. Using different light source samples for all queries can be highly inefficient. The method in this paper can be viewed as an effective way to use a single set of light source samples for all illumination queries.

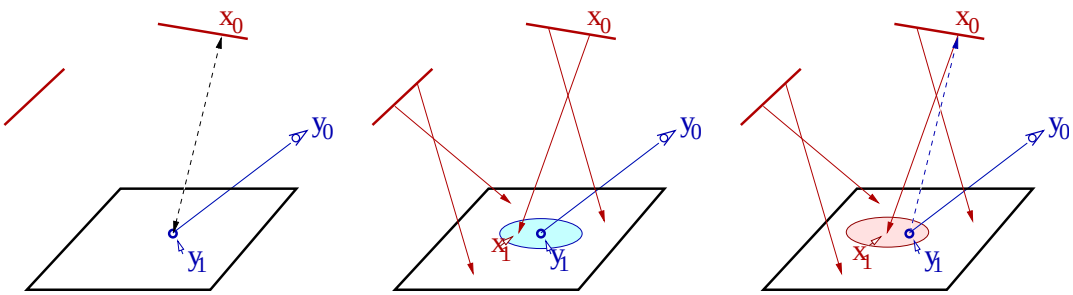

Figure 2: **Direct illumination (DI) solution strategies.** Left: Monte Carlo integration (MCI): a point x_0 is sampled. Using correct factors for visibility, cosines, BSDF, ..., leads to unbiased estimation; Middle: generic density estimation (DE) : DI is computed as a weighed average score of light particles x_1 arriving in the neighborhood of y_1. DE computes a convolution of DI; Right: generalized kernel method: like in DE, DI is computed from light particles x_1 landing in the neighborhood of y_1, but like in MCI, correct factors are used in estimation. This requires to shoot back a ray from y_1 to the origin x_0 of the light particle x_1. Blue lines indicate eye paths, red lines light paths.

Density estimation methods The basic idea of density estimation methods is to estimate $L^d(y_1 \rightarrow y_0)$ as

$$\frac{1}{N}\sum_{s=1}^{N}\frac{L^e(x_0^s \rightarrow x_1^s)G(x_0^s, x_1^s)}{p^L(x_0^s)p^L(x_1^s|x_0^s)} f_r(x_1^s; x_0^s \leftrightarrow y_0)K(x_1^s, y_1) \tag{3}$$

$p^L(x_0)$ denotes the pdf of sampling a light source point x_0. $p^L(x_1|x_0)$ is the pdf of continuing a light particle path from x_0 to x_1, including sampling the directional distribution of self-emitted radiance at x_0 and the pdf corresponding to shooting a ray from x_0 to x_1. $K(x_1, y_1)$ is a density estimation weight function [15]:

- Kernel method: $K(x_1, y_1)$ is the kernel function being used (Gaussian, Epanechnikov, ...);
- Histogram method: $K(x_1, y_1) = \chi_{S_{y_1}}(x_1)/\text{Area}(S_{y_1})$ where S_{y_1} is a surface element containing y_1, $\chi_{S_{y_1}}$ is its characteristic function (see table 1) and $\text{Area}(S_{y_1})$ is the surface area of S_{y_1};
- Nearest neighbor method: $K(x_1, y_1) = \chi_{D_{y_1}}(x_1)/\text{Area}(D_{y_1})$ where D_{y_1} is a a-posteriori determined disc with radius equal to the radius of the smallest sphere centered at y_1 that contains a required number of nearest neighbor particle hit points x_1;
- Orthogonal series estimation: $K(x_1, y_1) = \sum_\alpha \tilde{\phi}_\alpha(x_1)\phi_\alpha(y_1)$, where ϕ_α are

a set of basis functions defined on a surface element containing y_1 and $\tilde{\phi}_\alpha$ are the corresponding set of dual basis functions.

Density estimation methods effectively re-use a single set of light source samples for all required illumination queries, but they are biased: the estimates (3) converge to a convolution of the direct illumination distribution:

$$\lim_{N\to\infty} (3) = \int_S L^d(x_1 \to y_0) K(x_1, y_1) dA_{x_1} \neq L^d(y_1 \to y_0).$$

Intuitively, the bias is due to using incorrect values for the self-emitted radiance at x_0 (to x_1 instead of to y_1), the BSDF and the geometric factors (cosines, distance, visibility).

The generalized kernel method The key feature of our generalized kernels is that they compensate for using the "wrong" values for cosines, distance, visibility and light scattering and emission properties, using a correction factor $C(x_1, x_0; y_1, y_0)$:

$$\begin{aligned} K^\star(x_0, x_1, y_0, y_1) &= C(x_1, x_0; y_1, y_0) K(x_1|x_0; y_1) \\ C(x_1, x_0; y_1, y_0) &= \frac{L^e(x_0 \to y_1) G(x_0, y_1) f_r(y_1; x_0 \leftrightarrow y_0)}{L^e(x_0 \to x_1) G(x_0, x_1) f_r(x_1; x_0 \leftrightarrow y_0)} \end{aligned} \tag{4}$$

In order to obtain unbiased estimation, the weight function K shall satisfy the following conditions:

1. $\int_S K(x_1|x_0; y_1) dA_{x_1} = 1, \forall x_0, y_1 \in S$ (normalization);
2. $K(x_1|x_0; y_1) = 0$ whenever $p^L(x_1|x_0) = 0$ (good sampling).

Proof: Substitute the above kernel in the estimates (3). Multiply with the sampling pdf $p^L(x_0)p^L(x_1|x_0)$ and integrate over x_0 and x_1. Demanding that the resulting double integral equals the single integral (1) yields the normalization condition. The condition for good sampling states that no non-zero contributions shall be overlooked in the sampling process. It is a general requirement for unbiased estimation in Monte Carlo methods.

Within these constraints, the weight function $K(x_1|x_0; y_1)$ can be chosen at will in order to minimize the computation cost. Since the computation of direct illumination turns out to be a special case of the computation of full global illumination, we will work out a practical kernel directly for the general case.

2.2 Full Global illumination

The total radiance $L(y_1 \to y_0)$ emitted at a point y_1 in the direction of y_0 is the solution of the rendering integral equation, first introduced in graphics in [8]. The

common approach to solve this equation is to develop its solution into a Neumann expansion. The Neumann expansion is a sum, the first term being the self-emitted radiance $L^e(y_1 \to y_0)$, the second term being the direct illumination $L^d(y_1 \to y_0)$, next first-order indirect illumination etc.... The terms of the Neumann expansion are sampled by means of random walks: by simulating the trajectory $x_0, x_1, x_2, \ldots$ of light particles originating at the light sources in the scene. Each time such a virtual light particle hits a surface of the scene, a decision is made whether the random walk is terminated (absorption), or continued (survival). In case of survival, a scattered direction is sampled according to the BSDF at the hit surface point, and a ray is shot into the sampled direction to yield a subsequent particle surface hit point at which the game of chance is repeated (see e.g. [19] or [9, §6,7]).

The generalized kernel method is identical as for direct illumination, except that:

- The estimates (3) include also surface hit points x_k^s from particles that do not immediately originate from a light source ($k > 1$);
- For $k > 1$, the correction factor $C(x_k, x_{k-1}; y_1, y_0)$ compensates for using a "wrong" value of the BSDF at the light path parent vertex x_{k-1} rather than for self-emitted radiance;
- The weight function $K(x_k|x_{k-1}, x_{k-2}, \ldots, x_0; y_1)$ can depend on the whole light path history up to x_k.

3 A Practical Generalized Kernel

The most simple generalized kernels contain a weight function of the following form:

$$K(x_k|x_{k-1}, \ldots, x_0; y_1) = \frac{\chi_{F(x_k|x_{k-1},\ldots,x_0)}(y_1)}{N(y_1|x_{k-1}, \ldots, x_0)} p^L(x_k|x_{k-1}, x_{k-2}). \quad (5)$$

$F(x_k|x_{k-1}, \ldots, x_0)$ denotes a footprint region associated with the light path vertex x_k. $\chi_{F(x_k|\ldots)}(y_1)$ is the characteristic function of this footprint region. $N(y_1|x_{k-1}, \ldots, x_0)$ is a normalization factor. Unbiased estimation follows if

$$N(y_1|x_{k-1}, \ldots, x_0) = \int_S \chi_{F(x_k|x_{k-1},\ldots,x_0)}(y_1) p^L(x_k|x_{k-1}, x_{k-2}) dA_{x_k}. \quad (6)$$

Our choice of the footprint regions and the normalization are discussed further below.

Interpretation The generalized kernel method with above weight function can be interpreted as follows: whenever a light particle x_k arrives in the neighborhood of a point y_1 where we want to query the illumination, a shadow ray is shot back from y_1 to x_{k-1}, the point where the light particle came from (see figure 2). The point y_1 is considered to be in the neighborhood of x_k, if it is contained in the footprint region $F(x_k|\ldots)$ associated with x_k. The shadow ray shot back from y_1 to x_{k-1} has a high chance of being unoccluded, because it is known that x_{k-1} casts illumination into the neighborhood of y_1 (to the point x_k). Since the distribution of incoming light particles in the neighborhood of y_1 reflects the incoming light distribution, more visibility tests will be done with regions from which bright illumination is received. In the limit for small footprints, perfect sampling according to incident illumination follows.

The Footprints With proper normalization (see below), any choice of the footprints will result in unbiased estimation. An appropriate footprint definition is however important for the efficiency of the method. Footprint functions associated with rays have been used previously for texture filtering in classical and stochastic ray tracing, as well as in a hierarchical refinement criterion for Monte Carlo radiosity [6, 18]. The footprints are defined according to the following principles:

- The footprint size depends on the light particle density: we desire small footprints where sampling is dense and larger footprints where sampling is more sparse. The density reflects the light path sampling probabilities;
- The footprint filter after a scattering event is, under certain approximations, the convolution of a filter associated with an incident particle and a filter associated with the scattering event.

Since the choice of the footprint filters is not so critical in the generalized kernel method as it is in other applications, we propose to use a very simple constant footprint filter in a spherical region centered at the light particles position. The radius of the spherical regions is chosen as follows:

$$\begin{aligned} & R(x_k|x_{k-1},\ldots,x_0) \qquad (7)\\ &= \quad h/\sqrt{p^L(x_1|x_0)} \quad \text{if } k=1,\\ &= \quad h/\sqrt{p^L(x_k|x_{k-1},x_{k-2})} + R(x_{k-1}|x_{k-2},\ldots,x_0) \quad k>1. \end{aligned}$$

The global bandwidth parameter h is determined automatically in our implementation (§5).

Normalization The normalization condition (6) is unfortunately hard to satisfy exactly. Though introducing bias (discussed below), we obtained good results by approximating the normalization as the product of the two following factors, as well as slight variations on this theme:

$$N(y_1|x_{k-1},\ldots,x_0) \approx \tilde{\Omega}(y_1|x_{k-1},\ldots,x_0)\cdot p^L(\Theta_{y_1|x_{k-1}}|x_{k-1},x_{k-2})$$

with

- $\tilde{\Omega}(y_1|x_{k-1},\ldots,x_0)$ is a disc approximation for the solid angle at x_{k-1}, subtended by surface hit points x_k that can be reached by sampling from x_{k-1} and that have a footprint associated with them that overlaps y_1 (see figure 3):
$$\tilde{\Omega}(y_1|x_{k-1},\ldots,x_0) = \frac{\pi R^2(x_k|x_{k-1},\ldots,x_0)\cos\theta_{x_{k-1}|x_k}}{r^2_{x_{k-1},y_1}};$$
- $p^L(\Theta_{y_1|x_{k-1}}|x_{k-1},x_{k-2})$ is the probability of sampling a scattered (or emitted, if $k=1$) light direction towards y_1 from x_{k-1} for light particles coming from x_{k-2}: $p^L(y_1|x_{k-1},x_{k-2}) = p^L(\Theta_{y_1|x_{k-1}}|x_{k-1},x_{k-2})\cdot \text{vis}(x_{k-1},y_1)\cos\theta_{x_{k-1}|y_1}/r^2_{x_{k-1}y_1}$.

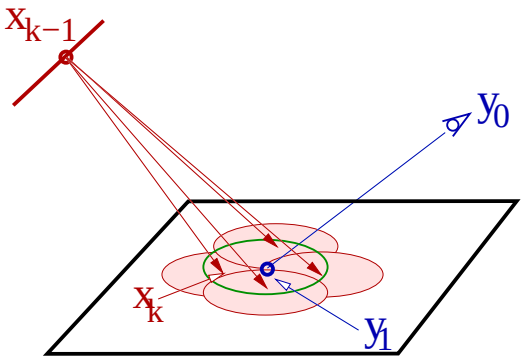

Figure 3: **Footprint normalisation.** The factor $\tilde{\Omega}$ is an approximation for the solid angle Ω at the parent light path vertex x_{k-1}, subtended by the set (green ellipse) of all surface points x_k (red arrows) with footprint (red ellipses) overlapping y_1. We approximate the average probability of expanding the path at x_{k-1} into this solid angle Ω, by the probability towards y_1.

Bias Inexact normalisation introduces bias in the resulting images (see figure 4). In particular, there will be bias near object silhouettes and after strongly directional light scattering or emission. In the former case, the approximation of Ω as

the solid angle $\pi R^2 \cos / r^2$ subtended by a disc will fail. Rather, the projection of the set of possible path vertices x_k with footprint overlapping y_1 will be similar to the projection of a partial disc. In the latter case, $p(\Theta|x_{k-1}, x_{k-2})$ may vary too strongly over Ω.

The amount of bias depends on the footprint radius. Small footprints will yield a lower bias than large footprints. The footprint size can be controlled globally using the bandwidth parameter h. Our choice of the footprint radius (7) results in a low bias for important illumination contributions such as direct lighting. Most of the bias will be in unimportant contributions, sampled with low probability, such as higher order diffuse interreflections.

Even so, the bias by our approximations is smooth and less objectionable to the human eye than the blurred shadow boundaries and light leaks of generic density estimation algorithms. Plenty of ad-hoc measures are possible in order to reduce the bias, by better approximating the exact normalization (6) in special cases.

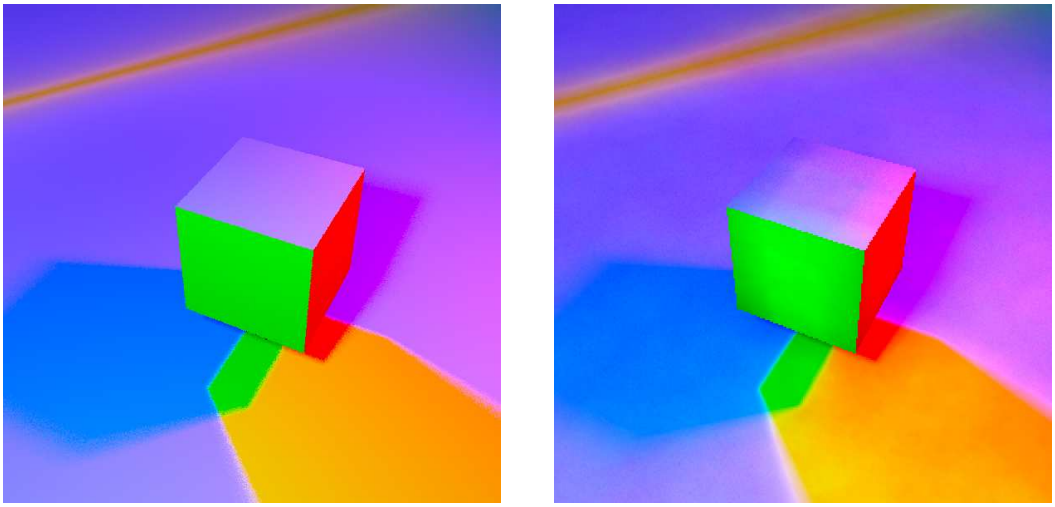

Figure 4: **Bias artefacts in our algorithm.** Left: unbiased solution; Right: inexact normalisation causes bias in our algorithm. Bias artefacts are usually much less noticable than in this exaggerated example however. The line on top is at grazing angles w.r.t. to planar blue light source. The image is also slightly noisy.

Noise In order to study in what cases our generalized kernel will lead to estimation with low variance and to detect cases where it behaves not so good, we need

to have a look at the resulting estimates a la (3):

$$\frac{1}{N}\sum \quad \frac{L^e(x_0 \to x_1)G(x_0,x_1)\cdots f_r(x_{k-1};x_{k-2}\leftrightarrow y_1)\cos\theta_{y_1|x_{k-1}}}{p^L(x_0)\cdots p^L(y_1|x_{k-1},x_{k-2})}$$

$$\times \quad f_r(y_1;y_0\leftrightarrow x_{k-1})\cos\theta_{x_{k-1}|y_1} \tag{8a}$$

$$\times \quad \frac{\chi_{F(x_k|x_{k-1},\ldots,x_0)}(y_1)}{\pi R^2(x_k|x_{k-1},\ldots,x_0)\cos\theta_{x_{k-1}|x_k}}\cdot \mathrm{vis}(y_1,x_{k-1}). \tag{8b}$$

The first factor is called the partial score associated with a light path $x_0,\ldots,x_{k-1},y_1$. Usually, variations in self-emitted radiance, BSDFs, etc... are compensated for by appropriate sampling so that this factor does not introduce significant noise. The main sources of noise in the estimates will be due to:

- variations in the BSDF times cosine at y_1, term (8a). Such variations can be compensated for by using rejection techniques [9, §3.5] or by means of bi-directional sampling, discussed below in §4.1;
- variations in kernel value and visibility, term (8b). The denominator does not introduce significant noise since the cosine factor is compensated for in the R^2 factor. The numerator results in a varying number of non-zero contributions at different y_1 query locations: some locations will receive more contributions than other nearby locations, yielding the same type of noisy artefacts that is also found in other density estimation methods. The rejection sampling technique proposed below in §4.2 tends to replace this "footprint clumping" noise by less noticable noise, uncorrelated between pixels.

4 Cost Reduction Techniques

In this section, we propose two techniques for reducing the computation cost of the generalized kernel method with the kernel proposed in the previous section. They are by no means the only possible cost reduction techniques. Many more techniques remain to be investigated.

4.1 Bi-Directional Estimation

Expression (8a) indicates that naive application of the above generalized kernel method will result in disturbing spike noise if the BSDF at y_1 varies considerably. This will be the case if y_1 is on a surface with strongly directional light scattering or emission. This phenomenon can easily be understood as follows: if the parent vertex x_{k-1} of a light path vertex x_k in the neighborhood of y_1 happens to be near

the ideal reflected direction at y_1, the BSDF at y_1 can assume a very large value. Such events are however sampled with a low probability, so that bright spike noise results. Such noise disappears only very slowly in the resulting images.

In such cases, better estimates for the illumination at y_1 will be obtained by sampling the BSDF at y_1 and querying the illumination using the neighboring light particles at the point y_2 where a ray shot into the sampled direction lands. This leads to multiple alternative sampling strategies for direct illumination, first order indirect illumination, etc. . . in exactly the same manner as in bi-directional path tracing [20, 10]. It turns out that also the same combination weights can be used. We illustrate this by means of first-order indirect illumination (two bounces of light between a light source and the observer, see figure 5).

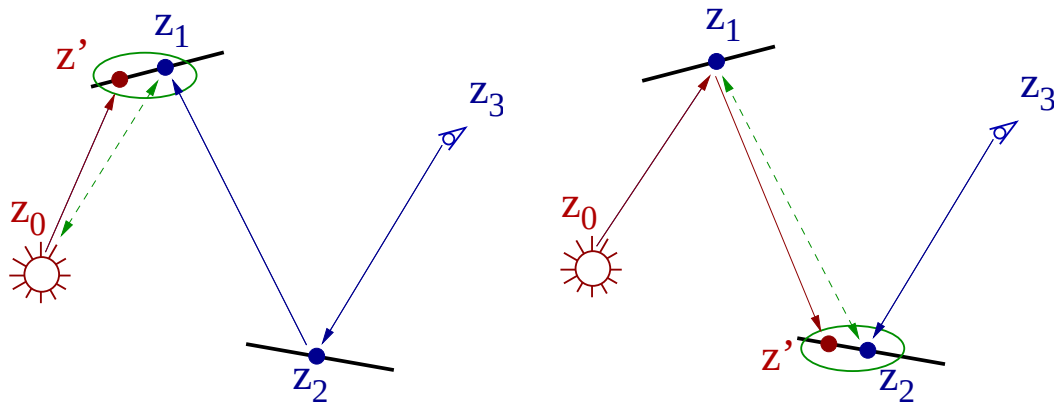

Figure 5: **Bi-directional estimation of first-order indirect illumination.** Left: strategy 1, combining at z_1; Right: strategy 2, combining at z_2. Both strategies can be combined with multiple importance sampling, using the same weights as in bi-directional path tracing.

First-order indirect illumination is described by the following integral:

$$L^{(2)} = \int_S \int_S \int_S \int_S L(z_0, z_1, z_2, z_3) dA_{z_0} dA_{z_1} dA_{z_2} dA_{z_3}$$

$$\begin{aligned} L(z_0, z_1, z_2, z_3) = {} & L^e(z_0 \to z_1) G(z_0, z_1) f_r(z_1; z_0 \leftrightarrow z_2) G(z_1, z_2) \\ & \times f_r(z_2; z_1 \leftrightarrow z_3) G(z_2, z_3) M(z_2 \to z_3). \end{aligned}$$

z_0 denotes a point on a light source and z_3 is a point on the camera film plane. $M(z_2 \to z_3)$ denotes a camera pixel filter function. In the simplest case, this can be a box filter associated with a given image pixel. This integral can be estimated by combining light and eye paths either at z_1 or at z_2:

Strategy 1: merging at z_1 A light particle path (z_0, z') of length 1 is traced as well as an eye particle path (z_3, z_2, z_1) of length 2. A connection is made whenever z_1 is contained in the footprint of the light particle at z'. The resulting estimate turns out to be:

$$\tilde{L}_1^{(2)} = \frac{L(z_0, z_1, z_2, z_3)\chi_{F(z'|z_0)}(z_1)/\pi R^2(z'|z_0)\cos\theta_{z_0|z'}}{p^L(z_0)\cdot p^E(z_3)p^E(z_2|z_3)p^E(z_1|z_2, z_3)\cdot p^L(z_1|z_0)}.$$

Strategy 2: merging at z_2 In this case, a light particle path (z_0, z_1, z') of length 2 and an eye path (z_3, z_2) of length 1 are traced. A non-zero contribution can result if z_2 is within the footprint of z'. The resulting estimate is:

$$\tilde{L}_2^{(2)} = \frac{L(z_0, z_1, z_2, z_3)\chi_{F(z'|z_1,z_0)}(z_2)/\pi R^2(z'|z_1, z_0)\cos\theta_{z_1|z'}}{p^L(z_0)p^L(z_1|z_0)\cdot p^E(z_3)p^E(z_2|z_3)\cdot p^L(z_2|z_1, z_0)}.$$

Combination of both strategies Any weighted sum $w_1(z_0, z_1, z_2, z_3)\tilde{L}_1^{(2)} + w_2(z_0, z_1, z_2, z_3)\tilde{L}_2^{(2)}$ of both estimates, with weights adding up to 1 and chosen so that they do not make any contributions vanish, will also be unbiased. In the balance heuristic [20], the weights are chosen proportional with the probability of sampling according to each strategy. Taking weights proportional with the denominators in the estimates $\tilde{L}_1^{(2)}$ and $\tilde{L}_2^{(2)}$, one obtains the same weights as in bi-directional path tracing with the balance heuristic:

$$\begin{aligned} w_1(z_0, z_1, z_2, z_3) &= \frac{p^E(z_1|z_2, z_3)}{p^E(z_1|z_2, z_3) + p^L(z_2|z_1, z_0)} \\ w_2(z_0, z_1, z_2, z_3) &= \frac{p^L(z_2|z_1, z_0)}{p^E(z_1|z_2, z_3) + p^L(z_2|z_1, z_0)} \end{aligned} \tag{9}$$

Note that the weights do not reflect the sampling probabilities exactly in our case, but this is also not required for unbiased combination.

In our implementation, we attenuated each path score resulting from connecting a light and eye sub-path with the above weights. The attenuation accounts (only) for the fact that a connection could have been made at a light path parent vertex x_{k-1}, by expanding the eye path, rather than at a child vertex x_k.

4.2 Reducing the Number of Visibility Tests

Disregarding the effect of cosines and the BSDF at y_1, expression (8b) indicates that particles with a large footprint, for instance corresponding with higher order indirect illumination, will result in small contributions. On the other hand, particles with a small footprint, for instance corresponding with direct illumination,

will yield large contributions. Straightforward application of the above generalized kernel method will often lead to a large number of small contributions to the illumination at y_1 and a small number of large contributions.

A good way to reduce the number of visibility tests to be performed therefore is to probabilistically accept/reject a candidate contribution with acceptance probability

$$p^{\text{accept}}(x_k) \propto 1/R^2(x_k|x_{k-1},\ldots,x_0)\cos\theta_{x_{k-1}|x_k}. \tag{10}$$

The acceptance probabilities need to be normalized so that the sum of $p^{\text{accept}}(x_k)$ for all x_k with footprint overlapping y_1 equals unity.

5 Implementation

In our implementation, we carry out the computations in steps. Each step results in an image, which exhibits some noise. By averaging these images, noise is smoothly reduced.

Each step basically consists of three sub-tasks: 1) tracing eye paths through each image pixel; 2) tracing a set of light paths and 3) combining the eye and light path vertices that landed in each others neighborhood.

Light and eye path tracing Light and eye path tracing are done in the standard way, except that the particle hit points are stored in main memory. For every light particle hit point x_k, we store a pointer to the BSDF/EDF at the hit point, the location and surface normal at the hit point, a pointer to the parent path vertex x_{k-1}, the accumulated factors $L^e(x_0 \to x_1)G(x_0,x_1)\cdots f_r(x_{k-1}|x_k \leftrightarrow x_{k-1})G(x_{k-1},x_k)$, the accumulated probability density $p^L(x_0)p^L(x_1|x_0)\cdots p^L(x_k|x_{k-1}x_{k-2})$ as well as the footprint radius (7). For eye particle hit points, corresponding values are stored, except that eye particle hit points do not have an associated footprint radius.

Even without compression, it is possible to store millions of particle hit points simultaneously in the main memory of state-of-the-art PC's. In order to compute very large images, or in a distributed implementation, one can apply the algorithm consecutively on separate image tiles. The same light particles can be used for the different image tiles without major overhead cost.

Combining the eye and light ray sets The combination of light and eye ray sets is performed in the following stages:

1. **Candidate link counting:** first, we count for every eye path vertex y in the eye ray set the number of light path vertices x with a footprint that overlaps y. At the same time, the sum of the acceptance probabilities (10)

is accumulated. This step involves spatial querying as explained below. We call every pair (x, y) of a light path vertex x and and eye path vertex y contained in its footprint, a *candidate link*;

2. **Trial linking:** the number of candidate links can be high. The next step therefore consists in reducing the set of candidate links to a more manageable set of *trial links* by probabilistically accepting or rejecting candidate links with probability (10). The number of trial links to accept is automatically determined as discussed below. Trial linking requires spatial querying a second time. We used an optimized version of the algorithm for stratified sampling of a discrete pdf in [11]. In our implementation, the trial link test takes only 2 floating point additions, 1 multiplication, 1 rounding operation and 1 integer comparison per candidate link;

3. **Visibility testing and scoring:** shadow rays are cast between the parent light path vertex and the eye path vertex in each trial link. In case of mutual visibility, the score associated with the trial link is computed and accumulated at the eye path vertices. The score is the product of the partial score associated with the light sub-path (first factor of (8)), the similar partial score associated with the eye sub-path, and the factors (8a) and (8b) attenuated with the bi-directional weights (9) and divided by the acceptance probability (10). The image is obtained by averaging the scores accumulated for each pixel.

Spatial Searching Candidate link counting and trial linking require to find for each eye path vertex y, all the light path vertices x with a footprint overlapping y, thus to find for each point P in a set of points, all the spheres S in a set of spheres that contain P. Such queries can be solved significantly faster the other way around, by querying for each sphere S the set of points P contained in the sphere. The points are pre-sorted in a balanced box decomposition tree [1]. The spatial queries in the algorithm proposed here are considerably more simple and cheap than nearest neighbor queries in photon mapping.

The trial linking stage thus produces trial links in light path vertex order. Also the scoring stage 3 is performed in this order, resulting in highly coherent bundles of shadow rays: all rays in a bundle originate at same parent light path vertex and are aimed towards all eye path vertices in the footprint region of a child light path vertex.

Auto-calibration The parameters of our algorithm are calibrated automatically in such a way that we always obtain about a fixed number $M = 10$ of unoccluded trial links per eye path vertex.

In order to obtain this effect, we need to request a potentially different number $M/P_{\text{vis}}(y)$ of trial links per eye path vertex y in the trial linking stage. $P_{\text{vis}}(y)$ denotes what fraction of the trial links at y is unoccluded. $P_{\text{vis}}(y)$ is estimated on the fly for each eye path vertex during the computations. If there are fewer than $M/P_{\text{vis}}(y)$ candidate links for a given eye path vertex, all candidate links are accepted as trial links in stage 2. Of course, the scores then do not need division by the trial linking acceptance probability in stage 3.

In order to assure that there will be a sufficient number of candidate links, the global bandwidth parameter h in (7) is calibrated in the first iteration and whenever the viewing position changes. It is chosen in such a way that there will be on the average M footprints overlapping each eye path vertex, considering only footprints from particles directly coming from a light source. The number of overlaps can be counted very efficiently with a balanced box decomposition tree data structure. It is to good approximation linear in the bandwidth parameter h, so that optimization is very easy and takes only 1 or 2 iterations virtually independent from any starting guess (we use 0.01 as a starting guess).

6 Results and Discussion

The implementation of the algorithm proposed in §5 amounts to about 1000 lines of C++ code on top of an existing path tracer and available BBDTree library.

Figure 6 shows a view of a building floor model consisting of 315,000 triangles of which more than 3000 are light sources. Most parts of the model are not seen in this views. For this image, we ran 15 steps of the algorithm. In each step, 100,000 light paths were traced. The eye path set, with a single eye path per pixel, was kept fixed. In order to obtain on the average 10 unoccluded connections per eye path vertex in each step, we needed to request about 30 trial links per eye path vertex in the trial linking stage. The image was generated at a resolution of $512 \times 384 \approx 200,000$ pixels. The computation time was about 5 minutes with our prototype implementation running on a 1.5GHz AMD Athlon PC[1]. Candidate link counting and trial linking took about 10% of the computation time. Most of the remaining time was spent in visibility testing and especially BSDF and pdf evaluations.

Bi-directional path tracing the same views resulted in highly noisy images even after one night of computations, because of the high number of light sources. Note that we have not performed any view-importance driven light source sam-

[1]We expect that with an optimized implementation, the computation times can be about 5 times lower. Our current implementation also does not yet allow to use more than 1 eye path through each pixel, which is not suffi cient for non-diffuse lighting effects to the viewer in image 7.

pling in either case. View-importance light source sampling would be highly beneficial in both cases.

Figure 6: This image shows a view of a building floor model consisting of 315k polygons of which over 3000 are light sources. It has been generated at a resolution of 512×384 in about 5 minutes on a PC. It illustrates the capability of our algorithm to deal efficiently with large models containing a high number of light sources, while convincingly reproducing delicate shading effects, such as soft shadows.

7 Conclusion and Future Work

We have proposed a new Monte Carlo algorithm for global illumination, based on a generalization of the kernel method for density estimation. The main strength of the algorithm is its ability to efficiently deal with large models, containing a high number of light sources casting subtle illumination effects without the need for a separate final gather pass. Like previous, generic, density estimation algorithms, also our algorithm is biased. The bias introduced by our algorithm is however

Figure 7: This image demonstrates that our algorithm is also able to reproduce non-diffuse lighting effects. Computation time and resolution are about the same as in the previous image.

much less objectionable than the blurred shadow boundaries and light leaks of the former. Our algorithm is self-calibrating and easy to use. It can provide feedback in seconds, and converges gracefully to a high quality image in minutes.

The bias introduced by the algorithm is however our main concern for future research. Bias reduction is possible by better approximating the kernel normalization condition (6). This accurate prescription how bias can be reduced, is also a unique feature of the generalized kernel method.

The algorithm is able to handle non-diffuse light emission and scattering. The basic method as described in this paper is however not able to reproduce caustic effects due to highly specular reflection or refraction (glossy caustics are reproduced). We expect this will be possible by incorporating additional sampling strategies.

Footprint clumping is the most prominent source of noise in our algorithm. Footprint clumping causes variations in the number of lighting contributions to neighboring locations. Such noise is well known in generic density estimation algorithms. It is straightforward to incorporate view-importance driven light path

sampling and low-discrepancy sampling in order to reduce footprint clumping. State-of-the-art Monte Carlo sampling techniques, promising geometric rather than $1/\sqrt{N}$ error reduction rates, may also be applicable in this context.

Finally, our algorithm stores a lot of information about the illumination in the scene and the current view. This information can be re-used for generating new views of static scenes in a walk-through application, or for rendering animations. Our algorithm is also very well suited for distributed implementation. We believe it might be a major step forward in the direction of interactive rendering of global illumination effects complex environments with general light emission and scattering properties.

Acknowledgments

The authors are grateful to Ingo Wald for plenty of insightful discussions and for allowing access to his software infrastructure. Philippe Bekaert acknowledges financial support by a Marie Curie post-doctoral research fellowship from the European Commission (contract nr. HPMF-CT-2001-01361). The model used in figure 6 is the SODA hall model from the university of Berkeley. The model shown in figure 7 was designed by Mark Mack Architects; 3D model courtesy of Charles Ehrlich and Greg Ward, UC Berkeley, College of Environmental Design. An early implementation of the method was performed in the context of the RenderPark system for global illumination, available from `www.renderpark.be`.

References

[1] S. Arya and D. M. Mount. Approximate range searching. In *11th Annual Symposium on Computational Geometry*, pages 172–181, June 1995.

[2] Ph. Bekaert. *Hierarchical and Stochastic Algorithms for Radiosity*. PhD thesis, K. U. Leuven, Leuven, Belgium, December 1999.

[3] K. Bouatouch, S. N. Pattanaik, and E. Zeghers. Computation of higher order illumination with a non-deterministic approach. *Computer Graphics Forum*, 15(3):327–338, August 1996.

[4] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (SIGGRAPH' 84 Proceedings)*, 18(3):137–145, July 1984.

[5] M. Feda. A Monte Carlo approach for Galerkin radiosity. *The Visual Computer*, 12(8):390–405, 1996.

[6] H. Igehy. Tracing ray differentials. In *Computer Graphics (SIGGRAPH'99 Conference Proceedings)*, volume 33, pages 179–186, 1999.

[7] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, 2001.

[8] J. T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86 Proceedings)*, 20(4):143–150, August 1986.

[9] M. H. Kalos and P. Whitlock. *The Monte Carlo method*. J. Wiley and sons, 1986.

[10] E. P. Lafortune and Y. D. Willems. Bi-directional Path Tracing. In H. P. Santo, editor, *Compugraphics '93 Conference Proceedings*, pages 145–153, Alvor, Portugal, December 1993.

[11] L. Neumann, W. Purgathofer, R. Tobler, A. Neumann, P. Elias, M. Feda, and X. Pueyo. The stochastic ray method for radiosity. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, July 1995.

[12] S. N. Pattanaik and S. P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. *Third Eurographics Workshop on Rendering*, pages 71–83, May 1992.

[13] M. Pharr, C. Kolb, R. Gershbein, and P. Hanrahan. Rendering complex scenes with memory-coherent ray tracing. In *SIGGRAPH 97 Conference Proceedings*, pages 101–108, August 1997.

[14] P. Shirley, B. Wade, Ph. M. Hubbard, D. Zareski, B. Walter, and Donald P. Greenberg. Global Illumination via Density Estimation. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 219–230, 1995.

[15] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1986.

[16] J. Spanier and E. M. Gelbard. *Monte Carlo Principles and Neutron Transport Problems*. Addison-Wesley, 1969.

[17] F. Suykens and Y. D. Willems. Adaptive filtering for progressive monte carlo image rendering. In *WSCG 2000 Conference Proceedings*, February 2000. available at http://wscg.zcu.cz/wscg2000.

[18] F. Suykens and Y. D. Willems. Path differentials and applications. In *Rendering Techniques 2001 (Proceedings of the 12th Eurographics Workshop on Rendering, London, UK)*, pages 257–268. Springer-Verlag, June 2001.

[19] E. Veach. *Robust Monte Carlo methods for light transport simulation*. PhD thesis, Stanford university, Department of Computer Science, December 1997.

[20] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *SIGGRAPH 95 Conference Proceedings*, pages 419–428, August 1995.

[21] E. Veach and L. J. Guibas. Metropolis light transport. In *SIGGRAPH 97 Conference Proceedings*, pages 65–76, August 1997.

[22] I. Wald, C. Benthin, M. Wagner, and Ph. Slusallek. Interactive rendering with coherent ray tracing. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2001*, 20(3), 2001.

[23] B. Walter, Ph. M. Hubbard, P. Shirley, and D. F. Greenberg. Global illumination using local linear density estimation. *ACM Transactions on Graphics*, 16(3):217–259, July 1997.

[24] G. J. Ward and P. Heckbert. Irradiance gradients. In *Third Eurographics Workshop on Rendering*, pages 85–98, May 1992.

Below you find a list of the most recent technical reports of the Max-Planck-Institut für Informatik. They are available by anonymous ftp from `ftp.mpi-sb.mpg.de` under the directory `pub/papers/reports`. Most of the reports are also accessible via WWW using the URL `http://www.mpi-sb.mpg.de`. If you have any questions concerning ftp or WWW access, please contact `reports@mpi-sb.mpg.de`. Paper copies (which are not necessarily free of charge) can be ordered either by regular mail or by e-mail at the address below.

Max-Planck-Institut für Informatik
Library
attn. Anja Becker
Stuhlsatzenhausweg 85
66123 Saarbrücken
GERMANY
e-mail: `library@mpi-sb.mpg.de`

MPI-I-2003-NWG2-002	F. Eisenbrand	Fast integer programming in fixed dimension
MPI-I-2003-NWG2-001	L.S. Chandran, C.R. Subramanian	Girth and Treewidth
MPI-I-2003-4-009	N. Zakaria	FaceSketch: An Interface for Sketching and Coloring Cartoon Faces
MPI-I-2003-4-008	C. Roessl, I. Ivrissimtzis, H. Seidel	Tree-based triangle mesh connectivity encoding
MPI-I-2003-4-007	I. Ivrissimtzis, W. Jeong, H. Seidel	Neural Meshes: Statistical Learning Methods in Surface Reconstruction
MPI-I-2003-4-006	C. Roessl, F. Zeilfelder, G. Nrnberger, H. Seidel	Visualization of Volume Data with Quadratic Super Splines
MPI-I-2003-4-005	T. Hangelbroek, G. Nrnberger, C. Roessl, H.S. Seidel, F. Zeilfelder	The Dimension of C^1 Splines of Arbitrary Degree on a Tetrahedral Partition
MPI-I-2003-4-004	P. Bekaert, P. Slusallek, R. Cools, V. Havran, H. Seidel	A custom designed density estimation method for light transport
MPI-I-2003-4-003	R. Zayer, C. Roessl, H. Seidel	Convex Boundary Angle Based Flattening
MPI-I-2003-4-002	C. Theobalt, M. Li, M. Magnor, H. Seidel	A Flexible and Versatile Studio for Synchronized Multi-view Video Recording
MPI-I-2003-4-001	M. Tarini, H.P.A. Lensch, M. Goesele, H. Seidel	3D Acquisition of Mirroring Objects
MPI-I-2003-2-003	Y. Kazakov, H. Nivelle	Subsumption of concepts in $DL\,\mathcal{FL}_0$ for (cyclic) terminologies with respect to descriptive semantics is PSPACE-complete
MPI-I-2003-2-002	M. Jaeger	A Representation Theorem and Applications to Measure Selection and Noninformative Priors
MPI-I-2003-2-001	P. Maier	Compositional Circular Assume-Guarantee Rules Cannot Be Sound And Complete
MPI-I-2003-1-016	G. Schfer, N. Sivadasan	Smoothed Competitive Analysis of the Work Function Algorithm for Metrical Task Systems
MPI-I-2003-1-015	A. Kovcs	Sum-Multicoloring on Paths
MPI-I-2003-1-014	G. Schfer, L. Becchetti, S. Leonardi, A. Marchetti-Spaccamela, T. Vredeveld	Average Case and Smoothed Competitive Analysis of the Multi-Level Feedback Algorithm
MPI-I-2003-1-013	I. Katriel, S. Thiel	Fast Bound Consistency for the Global Cardinality Constraint
MPI-I-2003-1-012	D. Fotakis, R. Pagh, P. Sanders, P. Spirakis	Space Efficient Hash Tables with Worst Case Constant Access Time
MPI-I-2003-1-011	P. Krysta, A. Czumaj, B. Voecking	Selfish Traffic Allocation for Server Farms
MPI-I-2003-1-010	H. Tamaki	A linear time heuristic for the branch-decomposition of planar graphs

MPI-I-2003-1-009	B. Csaba	On the Bollobás – Eldridge conjecture for bipartite graphs
MPI-I-2003-1-008	P. Sanders	Soon to be published
MPI-I-2003-1-007	H. Tamaki	Alternating cycles contribution: a strategy of tour-merging for the traveling salesman problem
MPI-I-2003-1-006	M. Dietzfelbinger, H. Tamaki	On the probability of rendezvous in graphs
MPI-I-2003-1-005	M. Dietzfelbinger, P. Woelfel	Almost Random Graphs with Simple Hash Functions
MPI-I-2003-1-004	E. Althaus, T. Polzin, S.V. Daneshmand	Improving Linear Programming Approaches for the Steiner Tree Problem
MPI-I-2003-1-003	R. Beier, B. Vcking	Random Knapsack in Expected Polynomial Time
MPI-I-2003-1-002	P. Krysta, P. Sanders, B. Vcking	Scheduling and Traffi c Allocation for Tasks with Bounded Splittability
MPI-I-2003-1-001	P. Sanders, R. Dementiev	Asynchronous Parallel Disk Sorting
MPI-I-2002-4-002	F. Drago, W. Martens, K. Myszkowski, H. Seidel	Perceptual Evaluation of Tone Mapping Operators with Regard to Similarity and Preference
MPI-I-2002-4-001	M. Goesele, J. Kautz, J. Lang, H.P.A. Lensch, H. Seidel	Tutorial Notes ACM SM 02 A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2002-2-008	W. Charatonik, J. Talbot	Atomic Set Constraints with Projection
MPI-I-2002-2-007	W. Charatonik, H. Ganzinger	Symposium on the Effectiveness of Logic in Computer Science in Honour of Moshe Vardi
MPI-I-2002-1-008	P. Sanders, J.L. Trff	The Factor Algorithm for All-to-all Communication on Clusters of SMP Nodes
MPI-I-2002-1-005	M. Hoefer	Performance of heuristic and approximation algorithms for the uncapacitated facility location problem
MPI-I-2002-1-004	S. Hert, T. Polzin, L. Kettner, G. Schfer	Exp Lab A Tool Set for Computational Experiments
MPI-I-2002-1-003	I. Katriel, P. Sanders, J.L. Trff	A Practical Minimum Scanning Tree Algorithm Using the Cycle Property
MPI-I-2002-1-002	F. Grandoni	Incrementally maintaining the number of l-cliques
MPI-I-2002-1-001	T. Polzin, S. Vahdati	Using (sub)graphs of small width for solving the Steiner problem
MPI-I-2001-4-005	H.P.A. Lensch, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing and Interactive Display of High Quality 3D Models
MPI-I-2001-4-004	S.W. Choi, H. Seidel	Linear One-sided Stability of MAT for Weakly Injective Domain
MPI-I-2001-4-003	K. Daubert, W. Heidrich, J. Kautz, J. Dischler, H. Seidel	Effi cient Light Transport Using Precomputed Visibility
MPI-I-2001-4-002	H.P.A. Lensch, J. Kautz, M. Goesele, H. Seidel	A Framework for the Acquisition, Processing, Transmission, and Interactive Display of High Quality 3D Models on the Web
MPI-I-2001-4-001	H.P.A. Lensch, J. Kautz, M. Goesele, W. Heidrich, H. Seidel	Image-Based Reconstruction of Spatially Varying Materials
MPI-I-2001-2-006	H. Nivelle, S. Schulz	Proceeding of the Second International Workshop of the Implementation of Logics
MPI-I-2001-2-005	V. Sofronie-Stokkermans	Resolution-based decision procedures for the universal theory of some classes of distributive lattices with operators
MPI-I-2001-2-004	H. de Nivelle	Translation of Resolution Proofs into Higher Order Natural Deduction using Type Theory
MPI-I-2001-2-003	S. Vorobyov	Experiments with Iterative Improvement Algorithms on Completely Unimodel Hypercubes
MPI-I-2001-2-002	P. Maier	A Set-Theoretic Framework for Assume-Guarantee Reasoning
MPI-I-2001-2-001	U. Waldmann	Superposition and Chaining for Totally Ordered Divisible Abelian Groups
MPI-I-2001-1-007	T. Polzin, S. Vahdati	Extending Reduction Techniques for the Steiner Tree Problem: A Combination of Alternative-and Bound-Based Approaches
MPI-I-2001-1-006	T. Polzin, S. Vahdati	Partitioning Techniques for the Steiner Problem
MPI-I-2001-1-005	T. Polzin, S. Vahdati	On Steiner Trees and Minimum Spanning Trees in Hypergraphs
MPI-I-2001-1-004	S. Hert, M. Hoffmann, L. Kettner, S. Pion, M. Seel	An Adaptable and Extensible Geometry Kernel