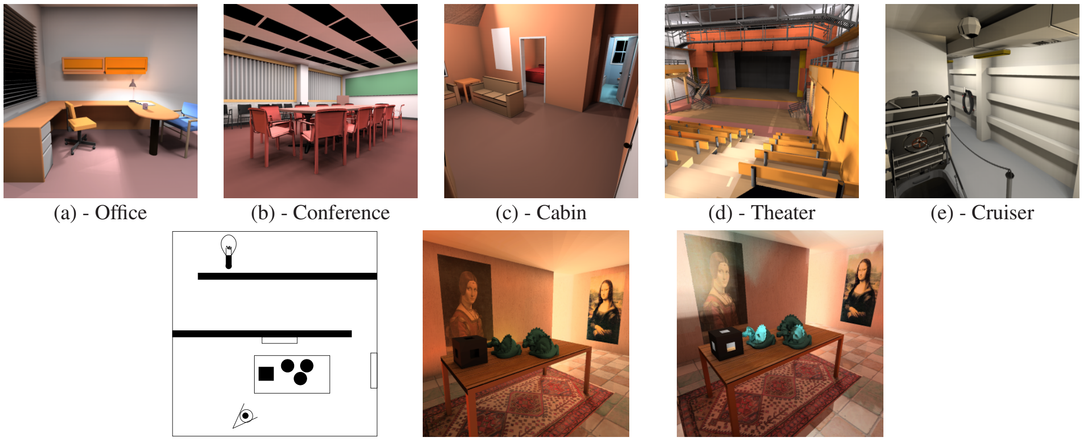
EUROGRAPHICS (2007)
D. Cohen-Or and P. Slavík (Guest Editors)

Metropolis Instant Radiosity

B. Segovia [1,2], J.C. Iehl[2] and B. Péroche [2]

[1]ENTPE: Ecole Nationale des Travaux Publics de l'Etat, Vaulx-en-Velin
[2]LIRIS: CNRS, INSA de Lyon, Université Lyon 1, Université Lyon 2, Ecole Centrale de Lyon



(a) - Office (b) - Conference (c) - Cabin (d) - Theater (e) - Cruiser

(f) (g) - Three Dragon Room (h) - A Jack-o-Lantern in (g)

Figure 1: *Some images rendered with Metropolis Instant Radiosity, a coherent ray tracer, and 1024 VPLs. Our method which consists in describing a Virtual Point Light (VPL) sampler as a Markovian process provides very satisfactory results in many cases: in mostly directly-lit scenes as shown in (a), (b) and, (c) with many light sources or complex scenes as shown in (d) and (e), and with very difficult visibility issues as shown in Figures (f), (g), and (h).*

Abstract

We present Metropolis Instant Radiosity (MIR), an unbiased algorithm to solve the Light Transport problem. MIR is a hybrid technique which consists in representing the incoming radiance field by a set of Virtual Point Lights (VPLs) and in computing the response of all sensors in the scene (i.e. camera captors) by accumulating their contributions. In contrast to other similar approaches, we propose to sample the VPLs with an innovative Multiple-try Metropolis-Hastings ($MTMH$) Algorithm: the goal is to build an efficient, aggressive, and unconditionally robust variance reduction method that works well regardless of the scene layout. Finally, we present a fast ray tracing implementation using MIR and show how our complete rendering pipeline can produce high-quality and high-resolution pictures in few seconds.

1. Introduction

Many applications from film special effects to light industrial design demand realistic rendering of complex scenes and rendering programs that can produce that kind of photo-realistic output. To achieve such a result, an class of methods generally called "global illumination algorithms", proposes to simulate the response of a given sensor to the incoming light flux. In this article, we present a sampling strategy, Metropolis Instant Radiosity, which replaces the incoming radiance field by a effective set of Virtual Point Lights (VPLs). Indeed, the quality of the sample set is primordial: as shown, for example, in Figure 1.f, placing VPLs closed to the light source will provide a bad solution, since the parts of the scene seen by the camera are not illuminated by the regions close to the source. Conversely, Metropolis Instant Radiosity presented here, will compute a VPL set which has an interesting intrinsic property: each VPL will provide the same amount of power to the camera. Figure 1 illustrates the efficiency of our approach with very different situations.

The remainder of the paper is finally organized as follows. Section 2 presents related work and Section 3, an overview of our contribution. Section 4 reminds how the global illumination problem has been formalized as an integration problem suitable for Monte-Carlo integration. We also review, with this formalism, the two techniques upon which we base our method. Sections 5 and 6 expose in details the new sampling strategies we set up to compute global illumination. In Section 7, we present the results we obtained with our new sampler and the comparisons with related approaches. The limitations and future work are given in Section 8. Section 9 concludes.

2. Related Work

The rendering equation [Kaj86] is today the root of all realistic rendering applications. In this article, we focus on Monte-Carlo algorithms to solve the equation since they are certainly the most efficient and versatile techniques. Kajiya was the first to design an unbiased global illumination algorithm of this type: he sampled the light reaching the image plane by tracing light paths backwards from the eye point. Due to intrinsic limitations and variance problems during integration, many other Monte-Carlo numerical schemes were developed: the light tracing algorithm [DLW93] is quite similar but builds paths starting from a light source instead of the camera. Bidirectional path tracing, introduced by Lafortune and Willems [LW93] then Veach and Guibas [VG94], proposes to compute two independent paths, one generated from the camera, the other one from a light source. Unfortunately, despite their robustness, most Monte-Carlo samplers are often inefficient since they do not have enough global context to quickly find all the relevant light transport paths.

To solve these issues, an innovative numerical algorithm, Metropolis Light Transport (MLT) was developed [VG97]. The decisive advantage of a Metropolis sampler over independent Monte-Carlo estimators is its ability to exploit coherence in path space and therefore to preserve the sampling context. Since 1997, the Metropolis-Hastings algorithm has been widely explored. Pauly et al. [PKK00] extended it by adding extra Monte-Carlo Markov Chain (*MCMC*) mutations that handle participating media. Kelemen et al. [KSKAC02] proposed a simplification of the MLT algorithm which increases the acceptance rate and directly works in the space of uniform random numbers used to build up paths. Fan et al. [FCcL05] also used a Metropolis-Hastings algorithm to populate photon maps by exploiting coherence among light paths. More recently, [CTE05] developed an efficient algorithm that uses Metropolis mutation strategies in a standard Monte-Carlo integrator. They first generate a set of path samples from the camera to the light sources, and then use a sequence of *MCMC* mutations to redistribute in an unbiased way the power of each path over the image plane. All these techniques therefore build low variance estimators and try to directly solve the problem in its high-dimensional aspect. Unfortunately, they are slow as they generally efficiently exploit neither the computation coherence nor the current CPU / GPU architectures.

Conversely, another large class of Monte-Carlo rendering techniques focuses on the algorithmic speed rather than on an aggressive variance reduction. The most famous one is certainly Photon Mapping [Jen01] which consists in propagating particles or photons, depositing them on the surfaces of the scene and finally computing the resulting image with a Monte-Carlo non-parametric estimation. Instant Radiosity [Kel97] is another elegant method to compute global illumination for diffuse or not-too-shiny materials. The idea is to propagate particles from the light sources and to store a Virtual Point Light (or VPL) at each bounce. The physical light sources and the indirect radiance field are thus replaced by the VPL set. A gathering pass finally computes the image by evaluating the radiance transfer between the $VPLs$ and the image plane. The main advantage of these methods is their affinity with very efficient rendering systems. One may refer to Wald's PhD [Wal04] for detailed discussions and effective ray tracing implementations of these approaches.

As any Monte-Carlo algorithm, Instant Radiosity and Photon Mapping however suffer from variance problems. Many solutions often called "importance-driven methods" which consist in tracing importance particles from the eye point and in using the resulting distribution to guide the photon tracing step have been proposed to address this issue (see [Chr03] for a detailed discussion). As these techniques do not seem suitable for interactive rendering, several researchers proposed interactive variance reduction techniques for Instant Radiosity. Wald et al. [WBS03] attempt to deal with many light sources by building a cumulative density function (CDF) depending on the light source contributions. Segovia et al. [SIMP06] recently proposed a generic bidirectional solution to sample $VPLs$ for Instant Radiosity. Unfortunately, these two techniques may be slow to converge when some difficult visibility problems have to be taken into account.

3. Overview of our Contribution

All the above problems therefore motivated Metropolis Instant Radiosity (presented in Algorithm 1). As we want to propose a rendering technique which remains numerically robust and fast for all kinds of scenes, combining a Metropolis sampler which can provide very relevant samples and Instant Radiosity which can be very efficiently implemented sounds good. In this paper, we therefore present an innovative VPL sampler using a modified Metropolis-Hastings:

the "Multiple-try Metropolis-Hastings Algorithm" (*MTHM*) [JSL00]. As we will show it, our method provides a faster exploration of the sampled space than any other related technique does, and finally offers very good estimators without important performance penalties.

Algorithm 1 Metropolis Instant Radiosity

```
Set all pixel intensities to 0
Compute the power $P_c$ received by the camera (see Section 5.1)
With a Metropolis-Hastings sampler (either the standard
  version presented in Section 5.2 or the Multiple-try one
  presented in Section 6), compute a set of $n$ $VPLs$ with a
  density proportional to the power they bring to the camera.
  We do not know the outgoing radiance functions of the $VPLs$
  but we know the scene transmits the same amount of the $VPL$
  power to the camera.
for $i = 1$ to $n$ do
    • Suppose that $VPL$ $i$ is on a diffuse surface and that it has
      a constant outgoing radiance function equal to 1. Compute the
      intensity of each pixel in the screen and the total power $P'$
      transmitted to the camera through the scene from $VPL$ $i$.
    • As we know that $VPL$ $i$ transmits a power equal to $P_c/n$
      to the camera and that there is a linear relation between the
      outgoing radiance function of the $VPL$ and the transmitted
      power, rescale the intensities of the pixels by a
      $\frac{P_c}{n\,P'}$ factor (see Section 5.4).
    • Accumulate $VPL$ $i$ contribution.
end for
```

4. Solving the Light Transport Problem with Monte-Carlo

This section gives a short overview of Monte-Carlo integration and reintroduces the appropriate formalism to the general light transport problem. This formalism and this notation will be used further to present our hybrid strategy in Sections 5 and 6.

4.1. Monte-Carlo Integration

The purpose of Monte Carlo integration is to compute an integral of the form:

$$I = \int_{\Omega} f(\omega)\, d\mu(\omega) \qquad (1)$$

where Ω is the integration domain, f is a real valued function and μ is a measure on Ω. I is thus the mean of function f on Ω for the given measure μ.

A Monte-Carlo integrator simply consists in sampling one or several random variable families and evaluating the integrand. Let $Y = (Y_n)_{n \in N}$ denote a sequence of random variables. Under some specific conditions, Monte-Carlo integration can compute the integral by reexpressing it as the expected value of $(Y_n)_n$. In other words, $I = E(Y)$. A common way to generate a suitable random variable family is to consider a sequence of independent random variables $X = (X_n)$ with the same probability density function p defined on (Ω, μ) and to use the weak law of large numbers such that: $I = E(Y) = E\left(\frac{f(X)}{p(X)}\right)$. This strategy is for example, often used in path tracing algorithms.

4.2. Path Integral Formulation

As a Monte-Carlo integrator requires to formalize the problem as an integration one, Veach proposed in his PhD thesis [Vea97] to rewrite the light transport problem.

The Light Transport Equation
Veach first developed the "three point form" formulation which describes the local lighting behavior of materials:

$$\begin{aligned} L(x' \rightarrow x'') &= L_e(x' \rightarrow x'') + \\ &\int_{\mathcal{M}} L(x \rightarrow x') f_s(x \rightarrow x' \rightarrow x'') G(x \leftrightarrow x') dA(x) \end{aligned} \qquad (2)$$

L is the equilibrium outgoing radiance function, $L_e(x' \rightarrow x'')$ is the emitted radiance leaving x' in the direction of x'', and $G(x \leftrightarrow x')$ is the geometric term between x and x'. It represents the differential beam between the two differential surfaces and is given by: $G(x \leftrightarrow x') = V(x \leftrightarrow x') \frac{\cos(\theta_0)\cos(\theta_i')}{||x - x'||^2}$ where $V(x \leftrightarrow x')$ is the visibility term between x and x' which is equal to 1 if x sees x' and zero otherwise. θ_0 (resp. θ_i') is the angle between $x \rightarrow x'$ and the surface normal at x (resp. x'). $f_s(x \rightarrow x' \rightarrow x'')$ is the bidirectional scattering distribution function of the material. $\mathcal{M}$ is finally the union of all scene surfaces and A is the Lebesgue (i.e. uniform) measure on $\mathcal{M}$.

Since formalizing the light transport problem as an integral equation is however not suitable to Monte-Carlo integration, we have to reexpress it.

The Measurement Equation
Any pixel computation can be first defined as the response of a hypothetical sensor to the incident radiance field. If $W_e^{(j)}(x \rightarrow x')$ is the responsivity of sensor j and I_j, the power it receives, we can define the measurement equation by:

$$I_j = \int_{\mathcal{M} \times \mathcal{M}} W_e^{(j)}(x \rightarrow x') L(x \rightarrow x')\; G(x \leftrightarrow x')\; dA(x)\; dA(x') \quad (3)$$

The Path Integral Formulation
Using the light transport equation, the measurement equation can be recursively expanded to be expressed in an

iterative way:

$$I_j = \sum_{k=1}^{\infty} \int_{\mathcal{M}^{k+1}} \Big[L_e(x_k \rightarrow x_{k-1}) G(x_k \leftrightarrow x_{k-1})\, W_e^{(j)}(x_1 \rightarrow x_0) \\ (\prod_{i=1}^{k-1} f_s(x_{i+1} \rightarrow x_i \rightarrow x_{i-1}) G(x_i \leftrightarrow x_{i+1})) dA(x_0) \, ... \, dA(x_k) \Big]$$

The measurement equation can be finally reformulated as:

$$I_j = \int_{\Omega} f^{(j)}(\overline{x}) d\mu(\overline{x}) \tag{4}$$

$f^{(j)}$ is defined for each path length k by extracting the appropriate term from expansion (4), Ω is the set of all finite length paths and μ the natural associated measure given by: $\mu(D) = \sum_{k=1}^{\infty} \mu_k(D \cap \Omega_k)$ where Ω_k is the set of all length k paths and μ_k the associated product measure given by $d\mu_k(x_0 \ldots x_k) = dA(x_0) \ldots dA(x_k)$.

With this formalism, the global illumination problem is now as an integration problem we can solve by a Monte-Carlo algorithm.

4.3. Metropolis Sampling for Light Transport

We give here a short overview of the Metropolis-Hastings algorithm and its application to the global illumination problem as introduced by Veach and Guibas [VG97].

Metropolis-Hastings (MH) Algorithm

We first recall that a sequence of random variables $(X^{(t)})_{t \in N}$ is a Markov Chain if $X^{(t)}$ depends only on $X^{(t-1)}$ through a transition function $g(\cdot|x^{(t-1)})$. The goal of the Metropolis-Hastings algorithm is to construct a Markov Chain that has a equilibrium distribution π_∞ by applying successive mutations on its elements. This algorithm does not solve *a priori* an integration problem but may provide a very elegant variance reduction technique in the case where many correlated integrals have to be computed.

The algorithm starts at $t = 0$ with the selection of $X^{(0)} = x^{(0)}$ randomly drawn from a distribution π_0 with the only requirement that $\pi_0(x^{(0)}) > 0$. Given $X^{(t)} = x^{(t)}$, the algorithms computes $X^{(t+1)}$ as follows:

1. Sample a candidate value X^* from a proposal distribution $g(\cdot|x^{(t)})$
2. Calculate the Metropolis-Hastings ratio $R(x^{(t)}, x^*)$, where:
$$R(u,v) = \frac{\pi_\infty(v) \cdot g(u|v)}{\pi_\infty(u) \cdot g(v|u)}$$
3. Sample a value for $X^{(t+1)}$ according to the following:
$$X^{(t+1)} = \begin{cases} X^* & \text{with probability } \min\{R,\, 1\} \\ x^{(t)} & \text{otherwise} \end{cases}$$

It is possible to show that under general conditions, the sequence $(X^{(t)})_{t \in N}$ is a Markov Chain with equilibrium distribution π_∞.

Ergodicity

With the MH sampler, we can therefore sample almost any distribution π_∞. If we ensure the *ergodic* property of the chain (i.e. that all states are equally probable according to π_∞ after a long time passed in the chain), we are furthermore able to take *all* samples of the Markov Chain as if they exactly follow the stationary distribution. To do this, it is sufficient to ensure that $g(x|y) > 0$ when $\pi_\infty(x) > 0$ and $\pi_\infty(y) > 0$ since all states can be reached with only one mutation step with a non-null probability.

Application to Light Transport

Veach and Guibas proposed to use a MH sampler as a powerful variance reduction technique for the global illumination problem. They first evaluate the total power received by the camera and then use a Metropolis sampler to compute correlated random variables with a density directly proportional to the integrand f as defined in equation 4. During the sampling process, they finally estimate the pixel intensities by counting the number of paths going through each pixel and by proportionally distributing the total power over all of them. For a more detailed introduction to Metropolis sampling and its application to rendering, we refer to [Pha03].

4.4. Instant Radiosity

Instant Radiosity [Kel97] (IR) is a powerful method to compute global illumination for diffuse or not-too-shiny materials. As shown in Figure 2, it actually consists in virtually splitting each path $\overline{x} = \{x_0, x_1, \ldots, x_n\}$ into three parts:

- $\overline{x}_c = \{x_0, x_1\}$ where x_0 is a location on a sensor and x_1 is a point seen by this sensor;
- x_v is a point located just after x_1: in a sense, x_v "illuminates" x_1 and it is the VPL location we are looking for;
- $\overline{x}_s$ is the remainder of the path. Its end is connected to a light source while its first point is connected to x_v. We may notice that $\overline{x}_s$ can be void if x_v is located on a light source.

Using the path integration formulation, we then generate and store N random paths $\{x_v, \overline{x}_s\}$ from the light sources and reuse these sub-paths for all camera pixels (i.e. the camera sub-paths $\{x_0, x_1\}$). Therefore, we have for each pixel j:

$$I_j = E\left(\frac{f^{(j)}(\overline{x})}{p(\overline{x})}\right) = E\left(\frac{f^{(j)}(\{\overline{x}_c, x_v, \overline{x}_s\})}{p(\{\overline{x}_c, x_v, \overline{x}_s\})}\right)$$

As $\overline{x}_c$ and $\{x_v, \overline{x}_s\}$ are independent:

$$p\left(\{\overline{x}_c, x_v, \overline{x}_s\}\right) = p\left(\overline{x}_c\right) p(\{x_v, \overline{x}_s\}))$$

We finally consider all the light sub-paths but only store the ending point of each of them x_v which is often called Virtual Point Light (VPL). For each VPL, we therefore store its location, its density $p(\{x_v, \overline{x}_s\})$ and its outgoing radiance function (which is constant if the surface is diffuse). In the remainder of the article, we will say that x_v is the *geographical* VPL and we will say that $\{x_v, \overline{x}_s\}$ is the *path* VPL. We think that it is important to remind that each VPL is actually a complete light path and not only its ending point. The efficiency of Instant Radiosity furthermore relies on the fact that the $VPLs$ are reused for all the sensors in the scene and that the global illumination problem comes down to visibility requests between the $VPLs$ and a point of the scene.

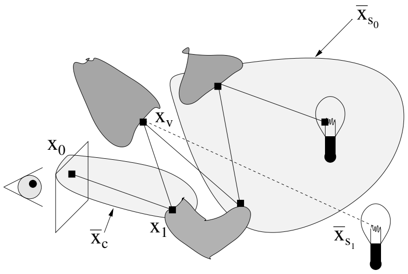


Figure 2: ***The decomposition of a path into three parts****: the camera path, $\overline{x}_c = \{x_0, x_1\}$, the location of the VPL, x_v and, $\overline{x}_{s_0}$, the remainder of the path which "brings some power" to x_v. We have here a geographical VPL x_v which contains two path $VPLs$, $\{x_v, \overline{x}_{s_0}\}$ and $\{x_v, \overline{x}_{s_1}\}$ (see Section 5.3)*

5. Metropolis Instant Radiosity

While the previous section presented the two major contributions upon which our work is based, we introduce here the core of our contribution, Metropolis Instant Radiosity (see Algorithm 1). To be more precise, our goal is to propose an efficient global illumination algorithm by computing a Markov-Chain of $VPLs$ such that each VPL will bring the same amount of power to the camera.

As our algorithm relies heavily on marginals, we first recall their definition. If $X = (X_0, X_1 \cdots X_{n-1}) \in (\Omega_0 \times \Omega_1 \times \cdots \times \Omega_{n-1})$ is a random variable, then X_i is simply called a marginal of X. Furthermore, if f is the density of X, the density f_i of X_i is defined by:

$$\begin{aligned} f_i(y) \; = & \int_{\Omega_0} \cdots \int_{\Omega_{i-1}} \int_{\Omega_{i+1}} \cdots \int_{\Omega_{n-1}} f(x_0, \cdots, x_{i-1}, y, x_{i+1}, \cdots, x_{n-1}) \\ & d\mu(x_0) \cdots d\mu(x_{i-1}) d\mu(x_{i+1}) \cdots d\mu(x_{n-1}) \qquad (5) \end{aligned}$$

We therefore have to integrate f on $\prod_{j \neq i} \Omega_j$ to obtain f_i.

5.1. Compute the power P_c received by the camera (Step 2)

We first compute the power P_c received by the camera by generating a family of bidirectional paths going from the camera to the light sources. To eliminate start-up bias, we also resample the generated paths to provide a "good" initial random variable with a law close to the integrand $f^{(c)}$ described in the next Section. For more details about this technique, one may refer again to [VG97].

5.2. Generating the $VPLs$ with Metropolis (Step 3)

The goal of a Monte-Carlo integrator for the Light Transport problem is to integrate $f^{(j)}$ as shown in Section 4.2 by equation (4). To achieve this goal, the best sampling strategy is to generate samples with a density directly proportional to $f^{(j)}$. What we propose here is thus to sample the entire path space Ω proportionally to the response $f^{(c)}$ of the camera to the power brought by a path, to project these samples on the appropriate sub-space and to finally consider them as $VPLs$.

By directly using the Metropolis sampling technique designed by Veach and Guibas, we are able to sample a distribution of paths proportional to $f^{(c)}$. Indeed, as indicated in Section 5.1, we first simulate a random variable X_0 with a density close to $f^{(c)}$ and then, we use the mutation strategies proposed by Veach and Guibas to compute a Markov-Chain with an invariant law proportional to $f^{(c)}$ (see Section 7.1 for more details about the implementation). It is important to notice that these paths are complete since they go from the camera to a physical light source. In the remainder of the paper, we will finally set the normalization constant a such as: $a = \frac{1}{P_c} = \frac{1}{\int_\Omega f^{(c)}(\overline{x}) d\mu(\overline{x})}$

As the sub-path $\{x_v, \overline{x}_s\}$ is a marginal of $\overline{x} = \{x_0, x_1, x_v, \overline{x}_s\}$ and as we directly sample $\overline{x}$ with density $a \cdot f^{(c)}(\overline{x})$, equation (5) gives us the density $p_{vs}^{(c)}(\{x_v, \overline{x}_s\})$ of $\{x_v, \overline{x}_s\}$:

$$a \cdot f_{vs}^{(c)}(\{x_v, \overline{x}_s\}) = \int_{\mathcal{M}} \int_{\mathcal{M}} a \cdot f^{(c)}(\{x_0, x_1, x_v, \overline{x}_s\}) \, dA(x_0) \, dA(x_1)$$

In other words, by sampling complete paths from the camera to a light source with a density proportional to $f^{(c)}$, we also have an interesting class of sub-paths $\{x_v, \overline{x}_s\}$ with a density proportional to $f_{vs}^{(c)}$. Actually, they all bring *the same amount of power to the camera*. Indeed, if $(\{x_{v_i}, \overline{x}_{s_i}\})_{i \in [1 \dots n]}$ is a set of n $VPLs$, then:

$$\begin{aligned} P_c \quad & = \quad \textstyle\int_\Omega f^{(c)}(\{x_0, x_1, x_v, \overline{x}_s\}) d\mu\{x_0, x_1, x_v, \overline{x}_s\} \\ & \simeq \quad \frac{1}{n} \textstyle\sum_{i=1}^n \frac{\int_{\mathcal{M}} \int_{\mathcal{M}} f^{(c)}(\{x_0, x_1, x_{v_i}, \overline{x}_{s_i}\}) dA(x_0) dA(x_1)}{p(\{x_{v_i}, \overline{x}_{s_i}\})} \\ & \simeq \quad \frac{1}{n} \textstyle\sum_{i=1}^n \underbrace{\frac{f_{vs}^{(c)}(\{x_{v_i}, \overline{x}_{s_i}\})}{p_{vs}^{(c)}(\{x_{v_i}, \overline{x}_{s_i}\})}}_{VPL\ i \text{ contribution}} = \frac{1}{n} \textstyle\sum_{i=1}^n \frac{1}{a \cdot n} = \frac{1}{n} \textstyle\sum_{i=1}^n \frac{P_c}{n} \end{aligned}$$

Additionally, we can remark that even if we know that every VPL brings the same amount of power to the camera (equal to P_c/n if there are n $VPLs$), the outgoing radiance function of each of them is *unknown* (but not needed as we know P_c).

As soon as the sampling step is finished, we finally have a set a sub-paths $(\{x_{v_i}, \overline{x}_{s_i}\})_i$: each of them is a *path* VPL which "emits" light from x_{v_i} and transmits a power equal to P_c/n to the camera.

5.3. Clustering the Physical $VPLs$ (Step 3)

Each VPL is thus a complete sub-path which starts from a light source and goes to the corresponding geographical VPL x_v. While applying mutations on complete paths, it is furthermore possible that the VPL location x_v does not change. This case occurs when:

- The candidate is rejected and the path is duplicated;
- Only the sub-path $\overline{x}_c = \{x_0, x_1\}$ is mutated;
- Only the sub-path $\overline{x}_s$ is mutated.

With this sampler, each geographical VPL can therefore contain several path $VPLs$: if k path $VPLs$ go to the given location x_v, the contribution of the *geographical* VPL x_v will be equal to $k \cdot P_c/n$. Finally, if the user requires m different locations for the $VPLs$ (i.e. m geographical $VPLs$), the number n of path $VPLs$ that must be generated is not known but is automatically determined during the sampling step by precisely monitoring the mutations.

5.4. Accumulating the VPL contributions (Steps 5)

After the sampling step, we have a set of m path $VPLs$ x_{v_i}: each of them brings a fixed amount of power to the camera equal to $P_i = k_i \cdot P_c/n$ where n is the number of paths generated during the sampling step (i.e. the total number of path $VPLs$) and k_i is the number of path $VPLs$ connected to x_{v_i}. To compute the contribution P_i of VPL x_{v_i} for each pixel of the screen, we simply dispatch its contribution P_i among all pixels. To achieve such a result, we first suppose that the surface at x_{v_i} is diffuse and that its outgoing radiance function is constant and equal to 1. Then, we perform the lighting computations and evaluate the intensity of every pixel. Once it is done, we evaluate the total power P'_i received by the camera and scale all pixel intensities by a P_i/P'_i factor such that the total power emitted by x_{v_i} and transmitted to the camera becomes P_i.

5.5. *MIR* with Common Renderers

Metropolis Instant Radiosity is conceptually different from Instant Radiosity since we do not know the outgoing radiance function of each VPL. This may be a practical limitation since most of the implementations of Instant Radiosity assume that this function is known and therefore base their code-design on this assumption. Fortunately, our sampling technique can be easily integrated to any of these renderers by adding an extra pass: the VPL outgoing radiance function estimation. This pass simply consists in randomly casting rays from the camera and then, in scaling their outgoing radiance function in relation to the power they bring to the camera. Thus, we do not have to extend any pre-existing renderer using Instant Radiosity since all the properties of the $VPLs$ (normal, power, and position) are determined.

6. A VPL Multiple-try Metropolis-Hastings ($MTMH$) Sampler

In the previous section, we described a complete rendering pipeline using a standard Metropolis-Hastings (*MH*) sampler. The main problem with such a sampler is the important correlation which may occur between successive samples in the chain: In worst cases, the algorithm may be slow to converge and it may be trapped in a local mode of integrand f (see Figures 3 and 4). To overcome these difficulties, Liu et al. [JSL00] proposed an alternative strategy known as Multiple-try Metropolis-Hastings sampling. What we propose here is to slightly change Step 3 of *MIR* by replacing the *MH* Algorithm by the Multiple-try one.

6.1. The $MTMH$ Algorithm

The approach is to generate a larger number of candidates thereby improving the exploration of π_∞ near x. One of these proposals is then selected in a manner that ensures that the chain has the correct limiting stationary distribution. To achieve such a result, we still use a proposal distribution g, with optional negative weights $\lambda(u,v)$ where the *symmetric* function λ is presented further below. To ensure the correct limiting stationary distribution, it is necessary to require that $g(x^*|x^{(t)}) > 0$ if and only if $g(x^{(t)}|x^*) > 0$, and that $\lambda(x^{(t)}, x^*) > 0$ whenever $g(x^*|x^{(t)}) > 0$. Let $x^{(0)}$ denote the starting value, and define $w(u,v) = \pi_\infty(v)g(u|v)\lambda(u,v)$. Then, for $t \in N$, the algorithm proceeds as follows:

1. Sample p independent proposals $X_1^* \dots X_p^*$ from $g(\cdot|x^{(t)})$.
2. Randomly select a single proposal X_j^* from the set of proposals, with probability proportional to $w(x^{(t)}, X_j^*)$ for $j = 1, \dots, p$.
3. Given $X_j^* = x_j^*$, sample $p-1$ independent random variables $X_1^{**}, \dots X_{p-1}^{**}$ from the proposal density $g(\cdot|x_j^*)$. Set $X_p^{**} = x^{(t)}$.
4. Compute the generalized Metropolis-Hastings ratio:

$$R_g = \frac{\sum_{k=1}^{p} w(x^{(t)}, X_k^*)}{\sum_{k=1}^{p} w(X_j^*, X_k^{**})}$$

5. Set

$$X^{(t+1)} = \begin{cases} X_j^* & \text{with probability } \min\{R_g, 1\} \\ x^{(t)} & \text{otherwise} \end{cases}$$

We can give an intuitive explanation to understand the $MTMH$ algorithm. With a standard Metropolis-Hastings sampler, we test two samples, x and x^*, and keep only one of them with the respective probabilities 1-min$(1,R)$ and min$(1,R)$. With $MTMH$, we conversely test two families of samples, $(x_1^* \dots x_p^*)$ and $(x_1^{**} \dots x_p^{**})$, and keep only one element of each family x_j^* or $x_i = x_p^{**}$ with the respective probabilities 1-min$(1,R_g)$ and min$(1,R_g)$: instead of only testing two points, we finally also deal with their "Metropolis neighborhood".

6.2. Application to *VPL* Sampling

As the limiting properties do not change with a *MTMH* sampler, the generation of complete paths $\overline{x}$ will provide the sub-path class $\{x_v, \overline{x}_s\}$ that has the same properties as those obtained with a standard Metropolis-Hastings sampler. We set $\lambda(u,v) = [g(u|v) \cdot g(v|u)]^{-1}$ to encourage certain types of proposals: by using this specific λ, $w(x_i, x^*)$ corresponds to the importance weight $\pi_\infty(x^*)/g(x^*|x_i)$ and the chosen candidate X_j^* becomes the probably most interesting sample among $X_1^* \dots X_p^*$. The decisive advantage of *MTMH* sampler over a Metropolis-Hastings one is algorithmic. Since the *VPL* generation step is not the most expensive one in a complete rendering pipeline, we have some computation time to finely tune the *VPL* distribution: *MTMH* is therefore quite appropriate to achieve this goal since it tends to decorrelate the successive *VPLs* and to explore the whole integration space faster. For a classical Metropolis Light Transport (*MTL*) implementation, this approach may be however much less interesting since generating $2k-1$ extra paths without using them may be inefficient. Nevertheless, we may notice that if we find a way to use all of them, *MTMH* may also provide an aggressive variance reduction technique in a *MTL* environment.

7. Implementation and Results

We present in this section how we have implemented our complete renderer, the results we obtained with *MIR*, and finally, several comparisons with existing similar approaches.

7.1. The *VPL* Sampling Pass

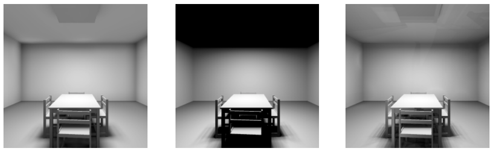
The first thing to do is to virtualize and replace the complete incoming radiance field by a set of Virtual Point Lights computed with our Metropolis Sampler. As described in the previous sections, we generate these *VPLs* by mutating and projecting light paths which go from the camera to a light source. Since our method is limited to diffuse and not-too-shiny environments, we do not deal with caustics and therefore only use bidirectional perturbations: for a complete and detailed explanation of the technique, we refer to [VG97].

7.2. Rendering with Coherent Ray Tracing

The core of our implementation relies on coherent ray tracing algorithms and implements the OpenRT API [DWBS03]. To render a picture, we therefore use the now classical rendering technique: "Instant Global Illumination" presented in [WKB*02]. First, we perform the ray casting requests with a finely tuned coherent ray tracer using the SIMD *SSE** instruction sets today available on almost every commodity PC. This approach simply consists in packing several rays inside one vectorized structure and performing all the operations by using *SSE* operands. Then, we use Interleaved Sampling [KH01] to accumulate distinct *VPL* contributions for every pixel inside a $n \times m$ tile. Once these contributions have been accumulated, we finally filter the resulting picture inside the continuous zones of the screen with a discontinuity buffer, thereby virtually providing $n \times m$ times more samples per pixel. More details about these techniques and the construction of efficient acceleration structures for ray tracing can be found in the literature [Wal04, Ben06, Hav00, WH06].

7.3. MTMH vs MH

We present here some simple configurations to show why it may be attractive to use Multiple-try Metropolis-Hastings rather than Metropolis-Hastings. In this section and the remainder of the article, we will use 10 candidates for *MTMH*, a commonly used value in computational statistics.



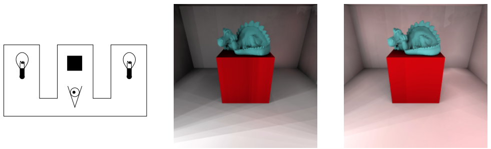
(a) - 12800 *VPLs* (b) - MH (c) - MTMH

	Direct	Indirect	RMS error
Ground Truth	49%	51 %	-
MH (128 VPLs)	100 %	0 %	0.04%
MTMH (128 VPLs)	56 %	44 %	0.005%

Figure 3: *Direct / Indirect modes: (a) is the reference image. (b) shows the results obtained with a MH sampler: the sampler gets stuck in the direct local mode. (c) shows the results obtained with a MTMH sampler: it was able to explore direct and indirect local modes.*

As shown in Figure 3.b, the *MH* sampler gets stuck in the direct local mode. Even if it will finally find the indirect contributions, the large correlation between successive samples will produce a "non-representative" sample set if a small number of *VPLs* are used. Conversely, *MTMH* provides much better results as shown in Figure 3.c: with very few *VPLs*, it proposes a good sample distribution with a direct/indirect repartition close to the reference one.

Figure 4 presents another kind of configuration where we explicitly create two important local modes: the parts of the scene seen by the camera can be illuminated either by the left room or by the right one. As expected, with only 256 *VPLs*, the *MH* sampler does not equally explore the two local modes whereas the *MTMH* Algorithm provides much better results by computing a more representative sample set.



(a) (b) - MH (c) - MTMH

Figure 4: *Exploration of left/right contributions: (a) is the overview of the scene: the two lights have power. (b) shows an image computed with 256 $VPLs$ and a MH sampler. (c) is the same scene with 256 samples, the same computation time, but computed with a $MTMH$ sampler.*

In the remainder of the article, we will therefore always use $MTMH$ instead of MH.

7.4. Results with Easy Configurations

We present here the results we obtained with Metropolis Instant Radiosity in "easy" scenes, i.e. scenes which are mostly directly lit. We compare our approach with two other methods, Bidirectional Instant Radiosity [SIMP06] and {Standard Instant Radiosity + Efficient Light Source Cumulative Distribution Function} (CDF) [WBS03]. The first method consists in generating a larger number of $VPLs$ than desired from the camera or the light sources, in computing the power they transfer to the camera and finally, in keeping the most relevant candidates (i.e. those which bring the larger contributions). The second method consists in computing the power brought by each *physical* light source to the camera through direct and indirect contributions and in associating to each of them the corresponding density. As shown in Figure 5, a standard sampling without any variance reduction technique gives poor results. On the contrary, the three other methods achieve much more satisfactory renderings with very comparable qualities. This can be easily explained by the fact that the three approaches try to generate a VPL distribution with a density close to the power brought to the camera. Since the scene is mostly directly illuminated, Wald's et al. approach approximates it by first computing a suitable CDF for the physical light sources. The resampling method tries to compute it by discarding the less interesting ones. Metropolis Instant Radiosity finally approximates it by directly simulating the desired density through a Markov-Chain.

We may also remark that the three approaches are complementary. Indeed, the $MTMH$ sampler chooses the best candidate with a technique close to a sampling / resampling strategy. Furthermore, associating to each source, a density proportional to the power they bring to the camera, remains interesting with a $MTMH$ sampler since it decreases the rejection rate during the sampling process (see Table 1) and accelerates the $VPLs$ generation. This combination is all the more efficient if we generate a large number of $VPLs$ or if the environment is highly occluded.

	with CDF	without CDF
Office	50 %	54%
Shirley's Scene 10	54 %	91%

Table 1: *Rejection Rate with $MTMH$*

7.5. Results with Difficult Visibility Configurations

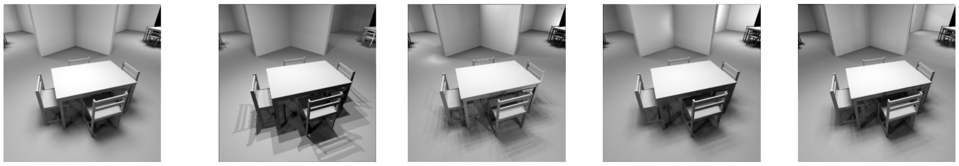
As presented above, *MIR* is efficient for scenes which are mostly directly lit. However, the decisive advantage of our strategy is its ability to handle very hard visibility issues. To prove it, we compared our approach to Standard Instant Radiosity (*SIR*) and Bidirectional Instant Radiosity (*BIR*) by testing the three techniques with two awkward configurations presented in Figures 1.f and 4.a. To be fair, we finally ensure that the VPL generation time with *BIR* is close the VPL generation time *MIR*.

As shown in Figures 6.a, 6.b and 6.c, Standard Instant Radiosity fails to find the relevant $VPLs$ since it does not discard the direct contributions. With *BIR*, the result is much better but an important noise is still noticeable. With *MIR*, we finally achieve an excellent result. The difference between the two approaches can be intuitively explained: *BIR* proposes to build a sample distribution with a density proportional to the power brought to the camera in an approximate way since the sampling / resampling strategy provide the exact density only if we resample an infinity of candidates. Furthermore, to ensure that we do not discard a VPL which brings some power to a small part of the screen, we must set a non-null probability for all $VPLs$ and therefore increase the variance of the estimator. On the contrary, *MIR* proposes to compute the desired density with a Markovian process. Since we have an appropriate initial random variable and an efficient mutation strategy using multiple candidates, the density is obtained very soon in the chain and the overall quality of the resulting estimators is very good.

The scene presented in Figures 6.d, 6.e, 6.f finally shows that sampling $VPLs$ with independent random variables can be extremely inefficient. With this layout, the ratio of the measure of relevant paths and the measure of all paths is indeed so small that both bidirectional and standard VPL sampling strategies are inefficient. On the contrary, *MIR* effectively explores the integration space around the relevant candidates and provides a very good VPL distribution.

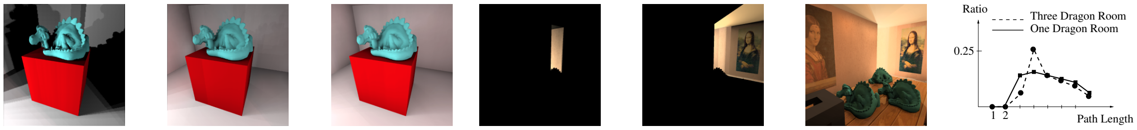
7.6. Overall Results

Table 2 sums up the computation times obtained with our implementation with the scenes and the camera positions pre-



(a) - Reference (standard) (b) - Standard RMS error: 0.02% (c) - Bidirectional RMS error: 0.007% (d) - CDF RMS error: 0.008% (e) - MTMH RMS error: 0.009%

Figure 5: *Tests with Shirley's scene 10. The reference image is computed with 12800 $VPLs$, the other ones, with* 256 *$VPLs$.*



(a) - Standard (b) - Bidirectional (c) - MTMH (d) - Standard (e) - Bidirectional (f) - MTMH (h) - Path Lengths

Figure 6: *Indirect Illumination Stress Tests: all pictures are computed with 1024 $VPLs$. We also give the different numbers of paths per length obtained while performing the mutations.*

sented in the article. For almost all scenes, *MIR* provides more relevant sample sets with a smaller amount of time. For the Three Dragon Room scene, where it is slower, *MIR* spends the most time to the evaluation of P_c (see Section 5.1) because we do not fix the total number of paths but the total number of paths which bring some power to the camera. However, even if we let the same computation time to *BIR*, the resulting distribution is much less relevant. On the other hand, if we want to achieve the quality provided by *MIR* with *BIR*, the needed resampling rate is superior to 1000 and thus inappropriate for interactive rendering. Finally, the current implementations of the samplers is neither optimized nor multi-threaded and we believe that the sampling process can be easily accelerated.

8. Limitations and Future Work

Even if we think that our approach provides significant improvements in difficult cases, we must clearly underline its limitations. First, as our method is view-dependent, flickering issues may occur. To solve this kind of problem, Ghosh et al. [GDH06] recently proposed a sequential Monte-Carlo technique to limit flickering while sampling environment maps. Adapting and applying this strategy to the generation of $VPLs$ may provide satisfactory results. Secondly, Instant Radiosity and *MIR* only handle diffuse or not-too-glossy surfaces. Directly using $VPLs$ to illuminate very specular surfaces will produce very high variance estimators: it would be interesting to generate reflected ray and find a diffuse surface to perform the VPL gathering. We think that this simple approach can provide good results but we must clearly formalize it to make it unbiased (a similar technique has been proposed by Kollig and Keller [KK04]: it consists in handling the numerical exception during the gathering pass by casting camera paths). Thirdly, even if our approach provides good results with directly-lit scenes in comparison with other importance-driven methods, we think that the better sample distribution offered by a low discrepancy sequence will give higher-quality results. Moreover, Owen and Tribble recently proposed a Quasi-Monte Carlo Metropolis Algorithm [OT05] which could provide a good sample space stratification *and* an effective exploration of the integration space thanks to *MCMC* mutations. Finally, our technique cannot handle caustics and this can motivate an alternative but attractive research direction: trying to set up an interactive Metropolis Light Transport system.

9. Conclusion

We presented in this paper Metropolis Instant Radiosity (*MIR*), a new sampling strategy to generate a finely-tuned set of Virtual Point Lights and handle all kinds of visibility or other difficult integration layouts in diffuse and not-too-glossy scenes. In essence, *MIR* uses a Metropolis sampler to compute a VPL family with a density directly proportional to the power they bring to the camera. Our approach is very fast as we use a Markov-Chain and therefore coherently exploit the information from one VPL to the next one. In addition, this strategy is optimal with a one-pixel camera and provides, in practice, very good results with many different layouts. For both directly and indirectly-lit scenes, *MIR* is able to find the most relevant physical light sources and more generally, the most relevant parts of the scene. Finally, our approach handles very difficult visibility layouts with almost no performance penalty compared to simpler configurations. As our method is an extension of Instant Radiosity, the final gathering pass finally remains fast since it can be easily integrated in many efficient rendering pipelines using $GPUs$ [SIMP06] or $CPUs$ [WKB*02].

	Scene 6	Scene 10	Office	Conf	Theater	Cruiser	Three Dragon Room
VPL generation time (MTMH)	0.32s	0.31s	0.31s	0.49s	1.0s	0.92s	2.4s
VPL generation time (Bidirectional)	0.82s	0.82s	0.62s	0.92s	1.0s	1.3s	1.0s
Rendering Time	4.5s	5.0s	3.2s	7.4s	11.1s	7.7s	7.1s

Table 2: *Computation times on a Core Duo T2600: the interleaved sampling pattern size is equal to* 8×8 *and the screen resolution, to* 1024×1024. *The resampling rate for BIR and the number of candidates generated with MTMH are equal to* 10. *As we use the same renderer for Bidirectional Instant Radiosity and Metropolis Instant Radiosity, the rendering times are identical. The given VPL generation time finally includes the construction time of the CDF for the physical light sources.*

References

[Ben06] Benthin C.: *Realtime Ray Tracing on Current CPU Architectures*. PhD thesis, Computer Graphics Group, Saarland University, 2006.

[Chr03] Christensen P. H.: Adjoints and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (2003).

[CTE05] Cline D., Talbot J., Egbert P.: Energy redistribution path tracing. In *SIGGRAPH '05 (Proceedings of the 32th annual conference on Computer graphics and interactive techniques)* (2005), pp. 1186–1195.

[DLW93] Dutre P., Lafortune E. P., Willems Y. D.: Monte carlo light tracing with direct computation of pixel intensities. In *Compugraphics '93 (Proceedings of Third International Conference on Computational Graphics and Visualization Techniques)* (1993).

[DWBS03] Dietrich A., Wald I., Benthin C., Slusallek P.: The OpenRT Application Programming Interface – Towards A Common API for Interactive Ray Tracing. In *Proceedings of the 2003 OpenSG Symposium* (2003), pp. 23–31.

[FCcL05] Fan S., Chenney S., chi Lai Y.: Metropolis photon sampling with optional user guidance. In *Proceedings of the 16th Eurographics Symposium on Rendering* (2005), pp. 127–138.

[GDH06] Ghosh A., Doucet A., Heidrich W.: Sequential sampling for dynamic environment map illumination. In *Proceedings of the 17th Eurographics Symposium on Rendering* (2006), pp. 116–126.

[Hav00] Havran V.: *Heuristic Ray Shooting Algorithms*. Phd thesis, Czech Technical University, 2000.

[Jen01] Jensen H. W.: *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., 2001.

[JSL00] Jun S. Liu Faming Liang W. H. W.: The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association* (2000), 121–134.

[Kaj86] Kajiya J. T.: The rendering equation. In *SIGGRAPH '86 (Proceedings of the 13th annual conference on Computer graphics and interactive techniques)* (1986).

[Kel97] Keller A.: Instant radiosity. In *SIGGRAPH '97 (Proceedings of the 24th annual conference on Computer graphics and interactive techniques)* (1997), pp. 49–56.

[KH01] Keller A., Heidrich W.: Interleaved sampling. In *Proceedings of the 12th Eurographics Workshop on Rendering* (2001).

[KK04] Kollig T., Keller A.: Illumination in the presence of weak singularities. *Monte Carlo and Quasi-Monte Carlo Methods* (2004).

[KSKAC02] Kelemen C., Szirmay-Kalos L., Antal G., Csonka F.: A simple and robust mutation strategy for the metropolis light transport algorithm. In *Computer Graphics Forum (Proceedings of Eurographics 2002)* (2002), pp. 531–540.

[LW93] Lafortune E. P., Willems Y. D.: Bidirectional path tracing. In *Compugraphics '93 (Proceedings of Third International Conference on Computational Graphics and Visualization Techniques)* (1993).

[OT05] Owen A., Tribble S.: A quasi-monte carlo metropolis algorithm. *Proc Natl Acad Sci U S A* (2005).

[Pha03] Pharr M.: Chapter 9: Metropolis sampling. In *Monte-Carlo Ray Tracing, SIGGRAPH'03 Course 44, course notes* (2003).

[PKK00] Pauly M., Kollig T., Keller A.: Metropolis light transport for participating media. In *Rendering Techniques 2000 (Proceedings of the 11th Eurographics Workshop on Rendering)* (2000).

[SIMP06] Segovia B., Iehl J.-C., Mitanchey R., Péroche B.: Bidirectional instant radiosity. In *Proceedings of the 17th Eurographics Symposium on Rendering* (2006), pp. 389–397.

[Vea97] Veach E.: *Robust Monte-Carlo Methods for Light Transport Simulation*. PhD thesis, Standford University, 1997.

[VG94] Veach E., Guibas L.: Bidirectional estimators for light transport. In *Proceedings of Eurographics Rendering Workshop* (1994), pp. 147–162.

[VG97] Veach E., Guibas L.: Metropolis light transport. *SIGGRAPH '97 (Proceedings of the 24th annual conference on Computer graphics and interactive techniques)* (1997), 65–76.

[Wal04] WALD I.: *Realtime Ray Tracing and Interactive Global Illumination*. PhD thesis, Computer Graphics Group, Saarland University, 2004.

[WBS03] WALD I., BENTHIN C., SLUSALLEK P.: Interactive Global Illumination in Complex and Highly Occluded Environments. In *Proceedings of the 14th Eurographics Workshop on Rendering* (2003).

[WH06] WALD I., HAVRAN V.: On building fast kd-trees for ray tracing, and on doing that in O(N log N). In *Proceedings of the 2006 IEEE Symposium on Interactive Ray Tracing* (2006).

[WKB*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray tracing. In *Proceedings of the 13th Eurographics Workshop on Rendering* (2002).