Eurographics Symposium on Rendering (2006)
Tomas Akenine-Möller and Wolfgang Heidrich (Editors)

Bidirectional Instant Radiosity

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

[1] LIRIS: Lyon Research Center for Images and Intelligent Information Systems
[2] ENTPE: Ecole Nationale des Travaux Publics de l'Etat

Abstract

This paper presents a new sampling strategy to achieve interactive global illumination on one commodity computer. The goal is to propose an efficient numerical stochastic scheme which can be well adapted to a fast rendering algorithm. As we want to provide an efficient sampling strategy to handle difficult settings without sacrificing performance in common cases, we developed an extension of Instant Radiosity [Kel97] in the same way bidirectional path tracing is an extension of path or light tracing. Our idea is to build several estimators and to efficiently combine them to find a set of virtual point light sources which are relevant for the areas of the scene seen by the camera. The resulting algorithm is faster than classical solutions to global illumination. Using today graphics hardware, an interactive frame rate and the convergence of the scheme can be easily obtained in scenes with many light sources, glossy materials or difficult visibility problems.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Bitmap and framebuffer operations I.3.7 [Computer Graphics]: Raytracing, Radiosity

1. Introduction

Computer graphics as many other numerical domains is a subtle mixture of mathematical and algorithmic methods. The role of a numerician is to efficiently combine them. In this paper, a new sampling strategy to achieve interactive global illumination is presented. The goal is to find a relevant set of virtual point light sources ($VPLs$) (see [Kel97]) to describe the incoming radiance field and use it to illuminate the areas of the scene seen by the camera. The sampling strategy must be accurate and provide relevant $VPLs$ in a fast and efficient way. Hence, it can be easily combined with a fast rendering method like the simple deferred shading approach we also present in the paper. To achieve this goal, we build bidirectional estimators and generate $VPLs$ from the camera and the light sources. With a sampling / resampling strategy, a VPL distribution with a density proportional to the power they bring to the camera is built.

The remainder of the paper is organized as follows. Section 2 reminds how the global illumination problem can be formalized as an integration problem suitable for Monte-Carlo integration and describes some previous work on the subject. Section 3 suggests in details the new sampling strategies we chose to compute global illumination. Section 4 presents the GPU implementation of our method and all numerical issues which can be encountered. Section 5 presents some results. A conclusion and possible future work are finally given in Section 6.

2. Previous Work

The light transport equation is a simple and short recursive formulation giving the equilibrium of radiance in a given scene [Vea97]:

$$L(x,\omega_e) \;=\; L_p(x,\omega_e) \;+\!\int_{\Omega} L(x,\omega_i)\; f_r(x,\omega_i,\omega_e)\; cos(\theta_i)\; d\omega_i \tag{1}$$

where:

- $L(x,\omega_e)$is the incoming radiance from point x in direction ω_e,
- $L_p(x,\omega_e)$ is the emission from point x in direction ω_e,
- $L(x,\omega_i)$ is the outgoing radiance at point x in direction ω_i,
- $f_r(x,\omega_i,\omega_e)$ is the BRDF of the material at point x.

Therefore, solving the global illumination problem leads to evaluate the light transport equation. In its recursive version, it is unfortunately not suitable for a direct Monte-Carlo application because of its recursive aspect. We can however notice it is appropriate to splitting methods [Whi80] where the law of large numbers is recursively applied. That is why

Veach redefined it in path space with its associated measure [Vea97]. As our work is a pure Monte-Carlo integration method, we briefly present his theoretical framework and the notations used in the whole article.

2.1. A Formalism for Monte-Carlo Integration

Veach first proposed the equivalent "three point form" of equation (1) based only on points [Vea97]. The density function of point x' seen by point x in direction ω_0 and the density $p(\omega_0)$ of ω_0 are related by $p_x(x') = p(\omega_0)\frac{|cos(\theta_i')|}{||x-x'||^2}$. Therefore, if $\mathcal{M}$ is the scene surface and $G(x\leftrightarrow x')$ the geometric term between x and x', the light transport equation can be reformulated as:

$$L(x'\rightarrow x'') = L_p(x'\rightarrow x'') + \int_{\mathcal{M}} L(x\rightarrow x')\, f_s(x\rightarrow x'\rightarrow x'')\; G(x\leftrightarrow x')\; dA(x) \quad (2)$$

Secondly, any pixel computation can be defined as the response of a hypothetical sensor to the incident radiance field. Let $W_e^{(j)}(x\rightarrow x')$ be the sensor responsivity. The power I_j transmitted by such a sensor is then defined by the measurement equation (3).

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

Using the light transport equation, the measurement equation can be recursively expanded to be expressed in an iterative way:

$$I_j = \sum_{k=1}^{\infty} \int_{\mathcal{M}^{k+1}} \Big[L_p(x_0\rightarrow x_1)\; G(x_0\leftrightarrow x_1)\; W_e^{(j)}(x_{k-1}\rightarrow x_k) \prod_{i=1}^{k-1} f_s(x_{i-1}\rightarrow x_i\rightarrow x_{i+1})\; G(x_i\leftrightarrow x_{i+1})\; dA(x_0)\; \ldots\; dA(x_k)\Big] \quad (4)$$

The measurement equation can be finally reformulated as:

$$I_j = \int_{\Omega} f_j(\bar{x})d\mu(\bar{x}) \quad (5)$$

f_j is defined for each path length k by extracting the appropriate term from expansion (4) and Ω is the set of all the finite length paths.

2.2. Path Integration Algorithms

Many Monte-Carlo methods have been proposed to solve the measurement equation over the last twenty years. Many of them share two common properties:

- they generate paths in the scene.
- they *analytically* compute the $pdfs$ of these paths.

Path tracing [Kaj86] is a simple algorithm which computes paths from the camera to a light source. Such a path is a homogeneous Markov chain. Its pdf can thus be analytically expressed as the product of the initial pdf by the transition $pdfs$ used to compute each new bounce. So, $p(\bar{x}) = p(x_0\rightarrow x_1\rightarrow\cdots\rightarrow x_{k-1}) = p(x_0)\prod_{i=0}^{k-2} p(x_i\rightarrow x_{i+1})$. Once the energy flux between the starting point and the light source is established, the light transport equation can be therefore easily evaluated. Light tracing algorithms [DLW93] are quite similar but build paths starting at 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. The first one is generated from the camera while the second one starts at a light source. The density functions can be analytically evaluated for each path. Since the paths are independent, the pdf of a bidirectional path is then the simple product of the pdf of the camera path by the pdf of the light path. It therefore includes and generalizes path and light tracing.

For algorithmic reasons, several estimators are created with the same paths or subpaths. For example, in the recursive version of path tracing, the paths used to compute the light transport equation restricted to energy fluxes after k bounces are recursively reused to compute the next contributions. Unfortunately, all these algorithms remain slow and do not seem suitable for interactive rendering.

2.3. Instant Radiosity

Instant Radiosity [Kel97] (*IR*) is an interesting method to compute global illumination for diffuse or not-too-shiny materials. The main idea is to factorize computations. Using the path integration formulation, N paths $\bar{x}_s$ are first randomly generated from the light sources and stored. Let (x_0) be a set of points on $\mathcal{M}_j$ (the surface of sensor j) and (x_1) a set of points on the surface of the scene $\mathcal{M}$ (See Figure 1.a). Then:

$$I_j = E\left(\frac{f_j(\bar{x}_s,x_1,x_0)}{p(\bar{x}_s,x_1,x_0)}\right)$$

As (x_1,x_0) and $\bar{x}_s$ are independent:

$$p\left[(\bar{x}_s,x_1,x_0)\right] = p(\bar{x}_s)p\left[(x_1,x_0)\right]$$

Therefore, Instant Radiosity is a standard Monte-Carlo path integration. Its originality relies on the fact that the "light parts" of the paths are constantly reused for all the sensors in the scene.

Algorithmically, all light subpaths are also taken into account but only the ending point of each of them is stored. Such a point is generally called a Virtual Point Light source or "VPL". Instant Radiosity is quite appropriate to provide efficient implementations. As Wald et al. showed in [WKB*02], coherent ray tracing can be used with *IR* in a very efficient way.

2.4. Importance-driven methods

Like other Monte-Carlo methods, Instant Radiosity suffers from variance problems due to inappropriate VPL sampling issues. Importance has been used by many researchers to speed up the convergence rates of off-line rendering in radiosity or Monte-Carlo methods. A detailed discussion is available in [Chr03]. More particularly, a subsequent work has been achieved with photon map techniques [Jen01]. Even if photon map algorithms are numerically quite different from Instant Radiosity (the first ones are non-parametric estimation methods while the second one is an analytical MC approach), the VPL generation is quite similar to the photon generation. Peter and Pietrek propose to trace importance particles from the eye point [PP98]. The resulting distribution is then used to guide the photon tracing step. The method was extended by Keller and Wald [KW00] and Suykens and Willems [SW00] to provide more robust numerical results. All these methods can be used to guide the VPL generation. Unfortunately, they do not seem suitable for interactive rendering. Indeed, many importons (i.e., particles generated from the camera) have to be generated and stored in a first pass. Furthermore, costly density estimations must be performed at each rebound. Finally, starting from a light source can be simply inappropriate for specific difficult integration cases since a very high number of particles may have to be generated before finding the relevant $VPLs$. This will be discussed with more details in Section 5.

3. Bidirectional Sampling of the $VPLs$

All the limitations of these previous methods motivated our approach. Inspired by the subsequent work achieved by Veach with path integration methods, we extend Instant Radiosity in the same way bidirectional path tracing is an extension of path or light tracing. Instead of generating $VPLs$ only from the light sources, they are also generated from the camera. The goal is to provide an efficient sampling strategy suitable for fast rendering engines as those proposed by Wald et al. or the one we present in this paper.

3.1. Reverse Instant Radiosity

Only generating $VPLs$ from the light sources is as arbitrary as only generating paths from the camera or the lights. Indeed, one strategy can be better or worse than the other according to the integrand. Finding caustics is for example much easier with light tracing than with camera path tracing. That is why we propose here an adjoint approach to IR. Instead of only sampling $VPLs$ from the light sources, they are also generated from the camera.

Our idea comes from a very simple observation. The only points which can illuminate an area of the scene seen by the camera are the points which can see this area. Therefore, the goal is to sample the VPL locations x_2 along the surface of the scene $\mathcal{M}$ to solve:

$$I'_j = \int_{\mathcal{M}\times\mathcal{M}\times\mathcal{M}} W_e^{(j)}(x_0\to x_1)\, L(x_2\to x_1)\, f_r(x_2\to x_1\to x_0)\, G(x_0\leftrightarrow x_1)\, G(x_1\leftrightarrow x_2)\, dA(x_0)\, dA(x_1)\, dA(x_2) \quad (6)$$

With a simple camera model (just a pinhole without lens), the $VPLs$ can be algorithmically found by randomly generating length 2 paths from the camera (See Figure 1b). The end of such a path gives the location of a VPL which can bring light to the camera after one bounce. Therefore, we have a new and simple sampling strategy which is to find the ending points of length 2 camera paths. This set will be denoted $\mathcal{M}_s$.

Once a VPL has been sampled, the information necessary to perform Monte-Carlo evaluations has to be computed.

Estimation of VPL density of probability $p(x_2)$
Each VPL is the end of a length 2 camera path $\bar{x}$. As $\bar{x}$ is a homogeneous Markov chain, its density p is expressed as $p(\bar{x}) = p(x_1)p(x_1\to x_2)$ where x_1 is a point directly visible by the camera and x_2 is the location of the VPL. $p(x_1\to x_2)$ is therefore the density of x_2 seen by x_1. Hence, computing the density of x_1 consists in evaluating the mean of $p(x_1\to x_2)$.

In other terms, $p(x_2)$ is the probability of reaching this point by a path of length 2. Formally, it is the second marginal law of $\bar{x} = (x_1, x_2)$. Therefore, if $\mathcal{M}_c$ is the set of all the points visible from the camera:

$$p(x_2) = \int_{\mathcal{M}_c} p(x_1)p(x_1\to x_2)dx_1$$

For example, if independent and uniform random variables x_{i_1} are used to sample $\mathcal{M}_c$,

$$p(x_2) = \lim_{N\to\infty} \frac{1}{N}\sum_{i_1=1}^{N} p(x_{i_1}\to x_2) \quad (7)$$

Estimation of VPL exiting radiance field $L(x_2\to x_1)$
Once the density of the $VPLs$ has been estimated, their outgoing radiance field must be computed. We can do a rough but common approximation by considering the surface of the $VPLs$ diffuse and their outgoing radiance field constant for all projected solid angles. Otherwise, representing the outgoing radiance field would be much more difficult since it may vary along the hemisphere according to the incoming radiance. To perform the radiance field estimation, we implemented a bidirectional technique similar to those described in [KK04]. Thus, several paths are first traced from a reverse VPL. Then, several standard $VPLs$ are used instead of light paths. Shadow rays are finally cast between the reverse VPL paths and the standard $VPLs$ to evaluate the energy transfered from the physical light sources to the reverse VPL.

Once the density and the outgoing radiance field of the generated $VPLs$ have been estimated, we can use them to

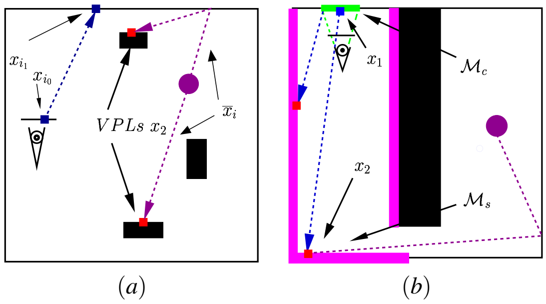


Figure 1: *(a) Standard IR. (b) Reverse IR. Green points x_1 ($\mathcal{M}_c$) are points visible from the camera. Magenta points x_2 ($\mathcal{M}_s$) are points visible from $\mathcal{M}_c$ and the only possible VPL locations for the given point of view. Sampling length 2 paths gives $\mathcal{M}_s$. Connecting $\mathcal{M}_s$ to the light sources gives the outgoing radiance field of the sampled $VPLs$.*

evaluate the measurement equation (6). Hence, we provide a simple sampling strategy to find $VPLs$ proportionally to the importance brought by the camera.

3.2. Bidirectional Instant Radiosity

With path tracing algorithms, the major problem is to find relevant paths. With Metropolis Light Transport [VG97], the ideal random variable with a density proportional to the integrand is directly found by a Metropolis sampling. The integration is finally performed thanks to the useful ergodic properties of the generated Markov chain. Unfortunately, with Instant Radiosity methods, generating an ideal random variable per pixel seems difficult since a part of the paths is already fixed by the $VPLs$. An easy way to do it would be to have a set of $VPLs$ for each pixel. Unfortunately, all the nice algorithmic properties of the method and its efficiency would be lost.

That is why we propose another sampling strategy. Its goal is to generate a VPL family such as the power brought to the camera by each of them is constant. In other words, we want to sample a random variable with a density proportional to the power brought to the camera. In the case where the integrand is the responsivity of the camera to the incoming radiance (i.e., a camera with one pixel), it is the best sampling strategy.

Standard and Reverse *IR* sampling methods can now provide the location of $VPLs$. As shown in Figure 2, Standard *IR* is not always better or worse than Reverse *IR*. According to the radiance field in the scene and the point of view, a method has to be preferred to the other. Therefore, we have to find an efficient way to combine the two sampling strategies.

We propose a Monte-Carlo sampling / resampling approach. In path tracing methods, sampling / resampling is often useless since the most expensive step is the generation of paths. Conversely, *IR* is a two-pass algorithm with an integration pass generally much more expensive than the propagation one. Therefore, a sampling / resampling method can be quite suitable to generate the $VPLs$.

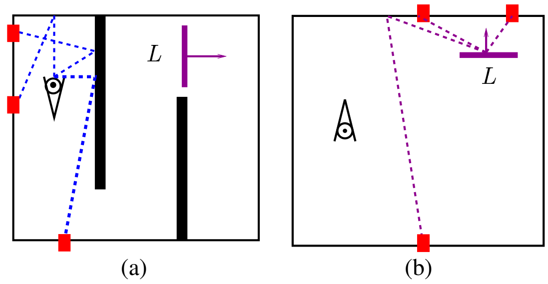


Figure 2: *(a) Here, Reverse IR is more efficient to find relevant $VPLs$. The scene is indirectly lit and the relevant paths are difficult to find from the lights. (b) Standard IR is more efficient. A light source directly illuminates a very small area of the scene. Finding it from the camera is difficult.*

First, a set $\mathcal{V}_N$ of $N/2$ standard $VPLs$ and $N/2$ reverse $VPLs$ is generated. For each of them, we evaluate the power brought to the camera (more exactly the camera response to the incident radiance field created by the $VPLs$). Then, the associated cumulative distribution function (CDF) is built. To estimate the power brought by every VPL, M length 2 paths are cast from the camera to the VPL and the energy transfered along these paths is evaluated and accumulated to build each element of the CDF. Then, we use the CDF to resample $\mathcal{V}_N$ and to determine a relevant subset $\mathcal{V}_{N'}$ ($N' \leq N$). The densities of the resampled $VPLs$ are finally subsequently modified by multiplying their density by their weights in the CDF and their energy by the number of times they have been chosen.

This method does not call for important storage capacity and remains very simple to implement since it does not require any extra sophisticated structures or non-parametric estimations. It can be noticed that a sampling / resampling strategy has been recently used by Talbot et al. [TCE05] to efficiently sample BRDFs and direct lighting.

4. Implementation

The implementation we propose must deal with sampling and performance issues. Two distinct passes are made. The first one generates $VPLs$ while the other one performs the final integration.

4.1. Handling Many Light Sources

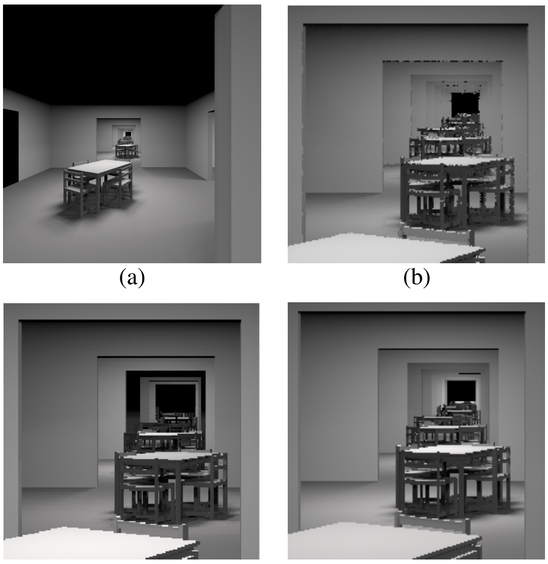
Before generating $VPLs$, the problem of sampling many light sources must be solved. Several papers made attempts to deal with this problem. As Dietrich et al. did [DWBS03], we build a cumulative density function (CDF) depending on the light source contributions. Paths are first traced from the camera and the contribution of every light source is computed along them. Then, the CDF is built and used to choose without bias which light sources must start a light path. This method can simply discard (or likely discard) the sources which barely light the areas seen by the camera.

4.2. VPL Sampling

The VPL sampling pass is done with a raytracer. To achieve efficient raytracing with a $\log(n)$ complexity, a kd-tree is classically built (with a $O(n\ log\ n)$ complexity). However, even if our raytracer is carefully written in C and many of the useful heuristics proposed by Havran in his PhD Thesis [Hav00] were used, we do not use any low-level optimization for our implementation.

We may notice three important numerical issues during the propagation pass described at Section 3.

- The first important issue is the power estimation done for each VPL. M sample paths are generated from the camera. The power brought by each VPL is then estimated using these paths. To prevent a rough estimation from discarding lights with a small contribution, a small uniform quantity is added to each element of the CDF (see Figure 3). We may notice that doing this keeps the estimator perfectly unbiased since no element of the CDF which can bring some power to the camera is nil. However, the resampled VPL density is no more proportional to the power they bring to the camera (but it remains close to it). During all our tests, $M = 10$ provided satisfactory results and did not significantly decrease the performance with our GPU integrator.
- The second numerical issue is to estimate the density of probability of every VPL sampled from the camera (see equation (7)). Once again, M sample camera paths are used to perform the estimation. If M is too small, the final result can be biased. If it is too big, the estimation becomes too expensive. $M = 50$ provided good results without noticeable bias. To speed up the computations, we also use a very simple trick. When the power brought to the camera by a reverse VPL is evaluated, only 5 samples are taken into account. Then, the precise density estimation with 50 camera samples is only done for the $VPLs$ which are resampled. Doing so, the quality of the power estimation is a bit degraded but the resampling rate can be much larger for the same computation time. In our tests, generating more $VPLs$ always provides more satisfactory results than performing a precise power estimation with fewer $VPLs$.
- The last numerical issue is the resampling rate. As the best sampling method is not known, $\frac{N}{2}$ reverse $VPLs$ and $\frac{N}{2}$ standard $VPLs$ are first sampled. Then, N' are resampled and kept. We produced several tests to estimate an efficient value for $r = \frac{N}{N'}$. If r is too large, two many sources are sampled / resampled and the propagation pass becomes too expensive. If it is too small, the integration domain is not sufficiently explored and the convergence will be slow. On most of the scenes, $r = 10$ produces very satisfactory results. With our GPU integrator, the propagation pass remains cheap compared to the integration one. Nevertheless, the value of r can be more accurately tuned depending on the respective performances of the propagation and the integration algorithms. It can be noted that it is not necessary to store all $VPLs$ which will be resampled if the sampling / resampling is done incrementally several times. Therefore, Bidirectional Instant Radiosity does not require important storage.



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

Figure 3: *(a) shows direct lighting for the 10th Shirley's test scene. (b) is a zoom on a small part of the screen computed by Radiance. (c) The same zoom done with a too rough CDF and our GPU integrator. Power brought by remote VPLs is underestimated. (d) A small quantity is added to all the elements of the CDF. The estimator is now unbiased.*

4.3. GPU Final Integration

To perform the final integration pass, we set up a simple GPU integrator. Indeed, major improvements in graphics hardware have been recently achieved. A commodity GPU now offers programming capabilities, very high fillrate and large memory bandwidth. Therefore, new rendering algorithms are now available and competitive. Deferred shading [ST90] suggests to store all geometric information in buffers and to reuse them to perform all lighting computations. As we want to accumulate many VPL contributions, we choose a deferred shading approach. It prevents the geometry from being rasterized many times.

G-buffer Creation: The normal, the position and color of each pixel are stored in a G-buffer. To build it, three float buffers are first created. They respectively contain positions, normals and colors. Material information such as material identifiers is also packed in the remaining components. For bandwidth reasons, precision is limited to 16 bits and therefore, the scene is bounded. It may be noticed that the G-buffer does not have to be recomputed if the point of view remains still. It is therefore well adapted to progressive rendering.

Shadow Map Computation: To perform all visibility requests, a shadow map is computed for each VPL. With the approximation that $VPLs$ lie on diffuse surfaces, they can be considered as uniform hemispherical sources. To compute the visibility requests due to a hemispherical light source, the five shadow maps of the hemicube are unrolled in a standard 2D shadow map and reindexed with a small cube map. The method is described in [KN04]. Due to shadow map aliasing issues, it can be noticed that the GPU implementation is not unbiased. Nevertheless, with high resolution shadow maps and the interior scenes we tested, aliasing is most of the time not noticeable for hemispherical shadow light sources. For exterior scenes and the illumination due to a sky, the parallel projections can be easily reparameterized to avoid aliasing (see for example [SD02] [WSP04]).

Shadowing Operations: Once the shadow map is computed, all the VPL contributions are accumulated using the standard technique of ping-pong buffers. Two buffers are simply created and used. During a computation pass, one buffer is written while the other one is read to fetch and accumulate the previous contributions. Their respective roles are then sequentially switched.

5. Results

We now present the results obtained with the Bidirectional *IR* sampling technique and the GPU implementation presented in Section 4.3. Furthermore, we will compare our method to the closest related work.

5.1. Reverse and Standard *IR*

To illustrate the behavior of the sampling strategies, a "U" office only indirectly lit by another room through a small corridor was designed (see Figure 4). For this kind of scenes, Reverse *IR* is more efficient since finding the $VPLs$ from the lights is quite difficult. Figure 4 presents some results obtained with the two approaches *without* resampling. A second office, indirectly lit by a halogen lamp was also built. It is a scene where Standard *IR* sampling is better than Reverse *IR* (see Figure 6b).

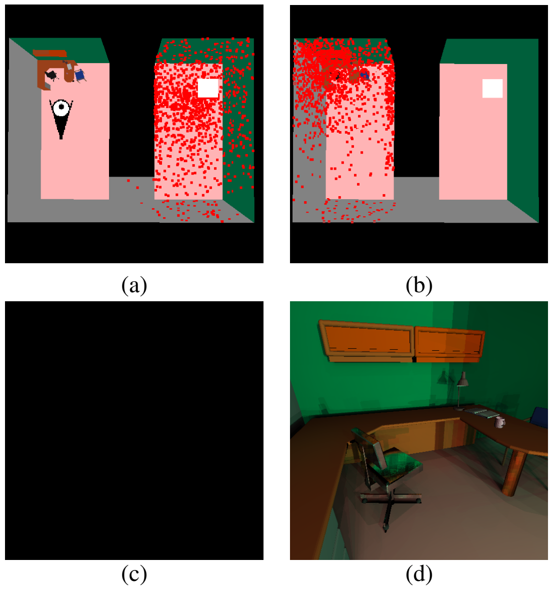


Figure 4: *The "U" office. (a) shows the $VPLs$ sampled with Standard IR sampling. (c) is the resulting picture. No VPL illuminates the areas seen by the camera. The picture is black (b) shows the $VPLs$ sampled with Reverse IR sampling. (d) the resulting raytraced picture with only 200 $VPLs$ and no resampling.*

	standard *IR* $VPLs$	Reverse *IR* $VPLs$
Conference	8.7 %	1.3 %
Cruiser	6.2 %	3.8 %
U-Office	0%	10 %
Q3tourney1	2.1 %	7.9 %

Table 1: ***Percentage of resampled VPLs*** *The resampling rate is equal to 1:10. According to the scene and the point of view, one method or the other provides the more relevant $VPLs$. For the conference room scene, most resampled $VPLs$ are standard $VPLs$. For U-Office, the Reverse IR strategy is more efficient.*

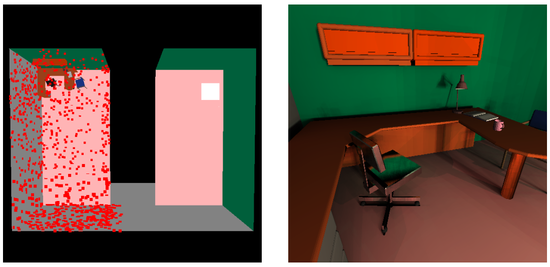
Therefore, depending on the scene, one of the method can be better than the other. To handle all kind of scenes, the two sampling strategies are combined. Table 1 gives the percentage of standard and reverse $VPLs$ kept after resampling. It gives a good idea of the quality of the generated samples for the given point of view. Hence, for U-office and the point of view given at Figure 4, 100 % of the resampled $VPLs$ were reverse $VPLs$. Conversely, on the conference room mainly directly lit by many small light sources, Standard *IR* provides much better light sources than Reverse *IR* does.

	Shadow Map	G-Buffer	Shading	ray/s	fps
Office	2.2 ms	8.3 ms	5.1 ms	1000K	15
Theater	4 ms	9.2 ms	5.1 ms	900K	6
Conf	8 ms	12.2 ms	5.1 ms	800K	4.5
Cruiser	15 ms	31 ms	5.1 ms	500K	2.2

Table 2: ***Performance of our implementation*** *The given times for the GPU passes are obtained with* $512 \times 512 \times 4$ *unrolled hemicube shadow maps and a* 1024×1024 *screen resolution. Frames per second given in the Table are obtained with 30 VPLs / frame. The GPU is an Nvidia GeForce 6800 GT and the CPU an AMD Athlon 2400+.*

5.2. Sampling / Resampling Performances

To handle all visibility difficulties, the *VPLs* are resampled. Figure 5 gives an image of the "U" office obtained by combining and resampling the two estimators. 1250 *VPLs* are created with Standard *IR* sampling strategies and 1250 other ones are generated with Reverse *IR* technique. Then, all of them are resampled with a resampling rate *r* equal to 1:10.



(a) (b)

Figure 5: *Sampling / resampling creates a distribution proportional to the power brought to the camera. (a) shows the VPLs distribution. (b) gives the raytraced image obtained with only* 250 *VPLs.*

Sampling / resampling gives satisfactory results in all the tested cases. Figure 6 presents other images computed with *VPLs* obtained using the sampling strategies described in Section 3. Scene q3tourney1 is particularly awkward. Light must bounce at least three times before reaching the central room through a small corridor.

5.3. GPU and Overall Performance

We now analyze the performance of our GPU integrator. The goal is to present the performance impact of our method compared to Standard Instant Radiosity and other techniques dealing with importance and bidirectional sampling. We just want to show that Bidirectional Instant Radiosity remains suitable for interactive rendering. Table 2 sums up the performances of our raytracer and the different GPU passes. An interactive frame rate is thus obtained with few *VPLs* for scenes with variable complexities (Office contains 35 000 triangles whereas Cruiser contains one million of triangles). Compared to coherent raytracing approaches [WBWS01], our implementation suffers from the linear algorithmic complexity of shadow map computations. Nevertheless, with a commodity GPU and a simple frustum culling on kd-tree leaves, interactive global illumination can be easily performed. Several pictures with the scenes we tested are presented at Figure 7.

We now analyze the impact of Bidirectional Instant Radiosity on the frame rate in comparison with Standard Instant Radiosity. For a 1:10 resampling rate, each *VPL* requires about 10 rays (by using 5 rays for the power estimation, 5 rays for the fast reverse estimation of the *VPL* density of probability and 50 rays for the precise density estimation). With the standard Instant Radiosity method and a mean albedo equal to 0.5, one *VPL* requires about 1 ray. Therefore, our sampling strategy is ten times slower than Instant Radiosity. However, even with 1000 *VPLs* per frame and 100 frame per second, an optimized raytracer can almost provide ten million rays as required by our method. Furthermore, *VPLs* can be reused from one frame to the next one. Therefore, Bidirectional *IR* makes Instant Radiosity suited for a wider variety of cases without sacrificing much of its fine properties (fast sampling strategies and integration).

5.4. Comparison with Importance-Driven Techniques

Another way to extend Instant Radiosity would be to use the importance-driven photon sampling strategy designed by [PP98]. With this technique, light sources and shooting directions are chosen by estimating the importance brought by the camera. Therefore, it would be easy to sample *VPLs* proportionally to the power they brought to the camera. Unfortunately, this method does not seem suitable for interactive or real time rendering. First, large scenes can require a large number of importons (more than 20000). Secondly, choosing a new shooting direction is very expensive since it requires the construction of a hemispherical CDF.

Keller's and Wald's method [KW00] is faster since no CDF is computed every time a new direction is chosen. The importance estimation is only used to determine if a photon is stored or not. Nevertheless, lots of importons have to be stored and the technique does not provide a new direction sampling strategy as Peter's technique does. The method was nevertheless tested with fast estimations with a regular grid. The quality of the resulted images was equivalent to Standard *IR* + sampling / resampling but provides much less satisfactory results than Bidirectional *IR*. This can be explained by the completely new sampling strategy brought by Reverse *IR*. The difference was particularly noticeable for the U-office scene where a large number of particles must be generated from the light sources before finding relevant *VPLs*.

Therefore, even if standard importance-driven techniques provide very good results for off-line rendering, Bidirectional Instant Radiosity outperforms them when relevant *VPLs* have to be found with the constraint of interactivity.

6. Conclusion and Future Work

We presented in this paper new sampling strategies to handle difficult scenes for global illumination. Our method extends Instant Radiosity and proposes a bidirectional sampler to find relevant *VPLs* for a given point of view in a fast and efficient way. Using GPUs and a simple and brute force implementation, we already perform interactive rendering with not-too-shiny materials and all kinds of visibility layouts. Future work will first be focused on a more efficient GPU implementation. Using all recent features of today graphics hardware, we believe that real time global illumination on one computer could be achieved today. For example, it would be very interesting to perform interleaved sampling on today's graphics hardware. Other sampling strategies could also be explored as a Metropolis *VPL* sampling. The idea would be to build an ergodic homogeneous Markov Chain of *VPLs* with an invariant law proportional to the power brought to the camera. Furthermore, Bidirectional Instant Radiosity could certainly be extended to handle coherent caustics generation and reflections in all kinds of visibility layouts.

7. Acknowledgments

We would like to thank the anonymous reviewers for their very valuable comments and pointing out the missing references. All models except q3tourney1 are courtesy of Greg Ward and can be found on the RADIANCE web site. Q3tourney1 is courtesy of ID Software.

References

[Chr03] CHRISTENSEN P. H.: Adjoints and importance in rendering: An overview. *IEEE Transactions on Visualization and Computer Graphics (TVCG) 9*, 3 (2003), 329–340.

[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), pp. 128–137.

[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 OpenSG Symposium* (2003), pp. 23–31.

[Hav00] HAVRAN V.: *Heuristic Ray Shooting Algorithms*. Ph.d. thesis, Czech Technical University (Prague), 2000.

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

[Kaj86] KAJIYA J. T.: The rendering equation. In *SIGGRAPH '86 (Proceedings of the 13th annual conference on Computer graphics and interactive techniques)* (1986), pp. 143–150.

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

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

[KN04] KING G., NEWHALL W.: Efficient omnidirectional shadow maps. In *ShaderX3* (2004), Charles River Media.

[KW00] KELLER A., WALD I.: Efficient importance sampling techniques for the photon map. In *Proceedings of Vision Modelling and Visualization* (2000), pp. 271–279.

[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), pp. 145–153.

[PP98] PETER I., PIETREK G.: Importance driven construction of photon maps. In *Rendering Techniques '98 (Proceedings of the Eurographics Workshop on Rendering)* (1998), pp. 269–280.

[SD02] STAMMINGER M., DRETTAKIS G.: Perspective shadow maps. In *SIGGRAPH '02 (Proceedings of the 29th annual conference on Computer graphics and interactive techniques)* (2002), pp. 557–562.

[ST90] SAITO T., TAKAHASHI T.: Comprehensible rendering of 3-d shapes. In *SIGGRAPH '90 (Proceedings of the 17th annual conference on Computer graphics and interactive techniques)* (1990), pp. 197–206.

[SW00] SUYKENS F., WILLEMS Y. D.: Density control for photon maps. In *Rendering Techniques 2000 (Proceedings of the Eleventh Eurographics Workshop on Rendering)* (2000), pp. 23–34.

[TCE05] TALBOT J. F., CLINE D., EGBERT P. K.: Importance resampling for global illumination. In *Rendering Techniques 2005 (Proceedings of the Eurographics Symposium on Rendering)* (2005), pp. 139–146.

[Vea97] VEACH E.: *Robust Monte Carlo Methods for Light Transport Simulation*. Phd thesis, Standford Univeristy, 1997.

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

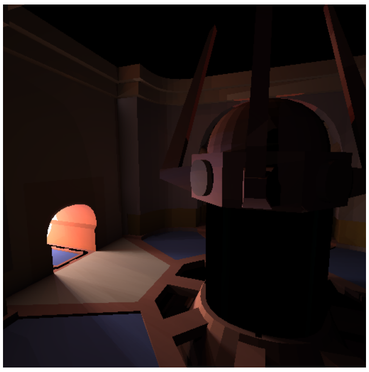
[VG97] VEACH E., GUIBAS L.: Metropolis light transport. *SIGGRAPH '97 (Proceedings of the 24th annual* *conference on Computer graphics and interactive techniques)* (1997), 65–76.

[WBWS01] WALD I., BENTHIN C., WAGNER M., SLUSALLEK P.: Interactive rendering with coherent ray tracing. *Computer Graphics Forum (Proceedings of Eurographics 2001) 20*, 3 (2001).

[Whi80] WHITTED T.: An improved illumination model for shaded display. *Commun. ACM 23*, 6 (1980), 343–349.

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

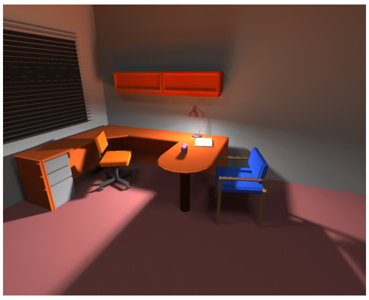
[WSP04] WIMMER M., SCHERZER D., PURGATHOFER W.: Light space perspective shadow maps. In *Rendering Techniques 2004 (Proceedings of the Eurographics Symposium on Rendering)* (2004).

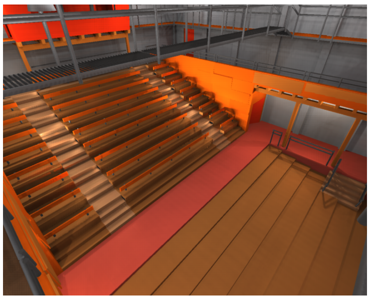

(a)


(b)

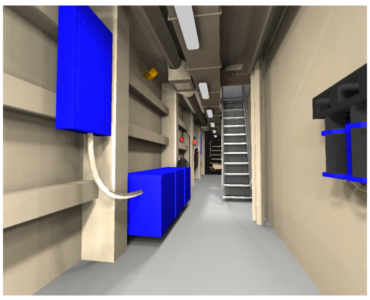

(c)

Figure 6: ***Sampling Performances.*** *Several pictures computed with the sampling techniques described in the article. Bidirectional sampling finds the relevant VPLs no matter the visibility layout. (a) A part of q3tourney1 (courtesy of ID software) indirectly lit trough a small corridor. Lights make at least 3 bounces before coming into the room. Most of the VPL are found with reverse IR strategy. (b) A simple office indirectly lit by a halogen lamp easily handled with Standard Instant Radiosity. (c) The U Office (raytraced). The scene is completely indirectly lit through a small corridor. Bidirectional IR quickly finds the relevant VPLs.*


(a)
2.4 s / 15 f/s


(b)
2.8 s / 6 f/s


(c)
3.8 s / 4.5 f/s


(d.1)
5.9 s / 2.2 f/s

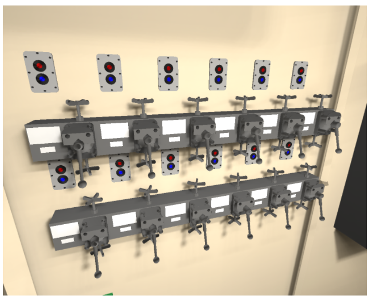

(d.2)
4.6 s / 2.2 f/s

Figure 7: ***Overall performances with a GPU deferred shading engine.*** *With the Bidirectional Instant Radiosity sampling strategy, convergence is obtained in a few seconds for very different scenes. Convergence times are obtained with less than 1% RMS error. Frame rates are obtained with 30 VPLs per second. The screen resolution is equal to* 1280×1024*. The graphics card is a NVidia GeForce 6800 GT. (a) An office with a very bright lamp. 35 000 triangles are rendered. (b) Candlestick Theater with 100 000 triangles. (c) The conference room. 200 000 triangles are rendered. (d) Two different views of Cruiser. One million triangles are displayed.*