A Final Reconstruction Approach for a Unified Global Illumination Algorithm

XAVIER GRANIER
IMAGER/University of British Columbia (Canada)
REVES/INRIA-Sophia Antipolis (France)
and
GEORGE DRETTAKIS
REVES/INRIA-Sophia Antipolis (France)

In the past twenty years, many algorithms have been proposed to compute global illumination in synthetic scenes. Typically, such approaches can deal with specific lighting configurations, but often have difficulties with others. In this article, we present a final reconstruction step for a novel *unified approach* to global illumination, that automatically detects different types of light transfer and uses the appropriate method in a closely-integrated manner. With our approach, we can deal with difficult lighting configurations such as indirect nondiffuse illumination. The first step of this algorithm consists in a view-independent solution based on hierarchical radiosity with clustering, integrated with particle tracing. This first pass results in solutions containing directional effects such as caustics, which can be interactively rendered. The second step consists of a view-dependent final reconstruction that uses all existing information to compute higher quality, ray-traced images.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism— *radiosity*; *ray-tracing*; *texture*

General Terms: Algorithms

Additional Key Words and Phrases: Global Illumination, Hierarchical Radiosity with Clustering, Particle Tracing, Final Gather, Density Estimation

1. INTRODUCTION

Global Illumination techniques have been developed in computer graphics to increase the realism of virtual worlds. These methods allow the simulation of indirect illumination, which is an important light effect that, in many cases, contributes to the impression of realism of a synthetic image. Recent

Diffuse solution

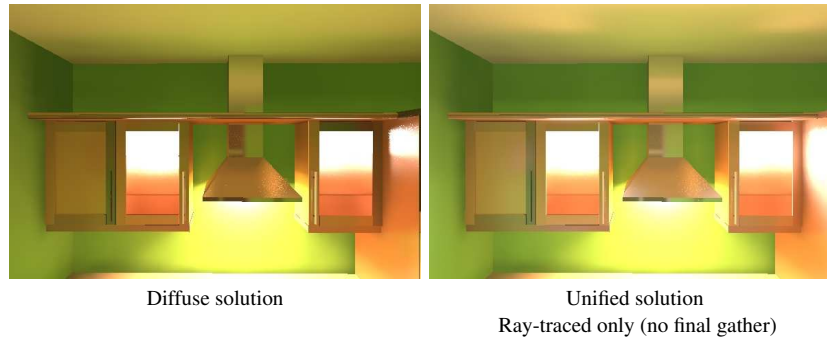Unified solution
Ray-traced only (no final gather)

Fig. 1. Diffuse vs. Full Global Illumination (Simplified model ©LightWork Design Ltd.). A Monte-Carlo pass has been performed to add the view-dependent effects.

radiosity algorithms can deal efficiently with very complex diffuse-only scenes, and combined with a final gathering process, can produce very high quality images. Nonetheless, real world scenes are rarely completely diffuse, and many directional effects such as highlights and caustics can occur, due to single or multiple reflections and refractions. For example, in Figure 1, where two light sources are inside the left wall cupboard, significant differences between a diffuse-only and a complete simulation can be seen in the resulting illumination on the ceiling or on the right. Stochastic methods can deal with such rapidly varying phenomena more easily than deterministic radiosity methods, but they suffer from the inherent problem of noise and often have difficulty computing complex indirect effects.

Recently, the development of a *unified algorithm* Granier et al. [2000; 2001] has allowed the automatic simulation of all light paths for a view-independent solution. This method is based on a Hierarchical Radiosity with Clustering (HRC) to compute diffuse exchanges. During HRC, the algorithm automatically classifies the exchanges for which it uses particles to compute the directional transfer. This method can deal with very complex light paths, in a fully automatic manner, even for hard cases such as nondiffuse indirect illumination. Nonetheless, even though the quality of the view-independent global illumination computation is often sufficient for interactive, walkthrough-type rendering, for some applications, higher quality may be required. For example, in film production very high quality images are required, which is not necessarily the case for design, simulation, or for interactive applications such as virtual reality.

In this article, we introduce a novel final reconstruction step for the unified algorithm which computes a single image, in a view-dependent manner. In the unified algorithm, Granier et al.[2000; 2001] focused on finding a full solution. In this article we focus on computing high quality images based on this approach. We perform a computation for each pixel of the final image, rendering a much more accurate result than the complete view-independent solution on the entire discretized scene. It is thus possible to obtain very high quality images. Most of the information provided by the unified algorithm is used to accelerate the final gather step and to increase the resulting image quality.

The rest of the article is organized as follows: after a presentation of previous work on global illumination related to our approach, we present a general overview of the new method. We then review details of the unified algorithm to gain a better understanding of the information that can be used to guide the final reconstruction step. We next describe the new final reconstruction step in two main parts: first the diffuse component (from links and caustics) and then the view-dependent component (direct and indirect reflections). These are the main contributions of this article. We introduce several techniques that improve on "Thrifty Final Gather" [Scheel et al. 2001] as well as density estimation-type algorithms

(photon map reconstruction). These techniques may be useful in other algorithms, in addition to the one presented. We conclude with results and discussion.

2. PREVIOUS WORK

A large amount of research has been done in global illumination. One major trend is represented by stochastic methods that develop solutions to the rendering equation [Kajiya 1986]. We can divide them into two main categories: unbiased methods (mainly ray-tracing-based, see Veach [1997] for an overview) and biased methods (mainly particle-based). The latter methods produce images with less noise by using a reconstruction process, thus trading noise for bias. The most popular method is the Photon Map, introduced by Wann Jensen [1996]. This approach uses photon (or particle) tracing for all light transfers, including diffuse. The method can quickly compute direct caustics (i.e., caustics caused by a light source directly reflecting or refracting off a nondiffuse object), since special purpose particles are sent to compute these effects. Density estimation [Walter et al. 1997] is another closely related approach. As with Photon Maps, all light transfer is performed with particle tracing. The result, however, can be viewed using a decimated mesh on graphics hardware or using ray tracing to account for the additional paths from the eye to one or more specular surfaces to the light or a diffuse surface. For all these methods however, the automatic treatment of all types of indirect lighting is still difficult (e.g., indirect caustics, mainly indirect lighted scenes, and so forth.) as they are low probability effects.

Deterministic methods have also been proposed, mainly based on the original radiosity algorithm [Goral et al. 1984]. To compute a fast radiosity solution for complex scenes, a hierarchical representation of the radiosity function [Hanrahan et al. 1991] has been introduced. In addition, the possibility of using groups of objects ("clusters" [Smits et al. 1994; Sillion 1995]) or objects more complex than polygons [Stamminger et al. 1997; Willmott et al. 1999; Alonso et al. 2000; Michelin et al. 2000] has been introduced. These methods are well known for their capacity to efficiently deal with purely diffuse environments. However, memory requirements are still a significant limitation [Stamminger et al. 1998; Granier and Drettakis 1999] since very fine subdivision is needed to correctly represent certain high frequency effects such as shadows.

A final gathering step [Christensen et al. 1997; Bekaert et al. 1998] can be done to increase the quality of the images computed on a coarse mesh by these methods. In this step, the information that is contained in the radiosity solution is used to guide a ray tracing pass, resulting in a higher quality image. The most recent method has been introduced by Scheel et al. [2001]. This method tries to find the hierarchical elements that contribute the most to local error for each mesh vertex. Only these elements are sampled to recompute a more precise value. For all others, radiosity is interpolated from the mesh vertices. This allows the correction of the main visible artifacts, such as shadow boundaries. The final gather step becomes more efficient when the precision of the radiosity solution is increased. This approach has two main limitations. First, it is a global method and information has to be gathered on all the mesh vertices, even on those from patches that will not be visible in the final image. Second, the initial radiosity algorithm provides only diffuse information, and it would be difficult to compute some of the nondiffuse components in the resulting image. As an example, the global effect of nondiffuse transfers on overall illumination cannot be integrated easily (see Figure 1), resulting in missing illumination, incorrect global intensity, and incoherent overall illumination. Nevertheless, our final reconstruction step will be derived from this approach for the diffuse part of an image and our unified approach will provide the missing features.

Since the advent of the radiosity methods, several solutions have been proposed to add nondiffuse transfer to the original, diffuse-only solution. One approach is to store directional finite elements on patches. From the earliest such work [Immel et al. 1986] to the most recent [Stamminger et al. 1999]

using clustering and wavelets, the storage required for directional discretization makes these methods unusable for complex scenes which contain highly directional glossy effects such as caustics. Continuous representations (e.g., Sillion et al. [1991]) also require large numbers of base coefficients, leading to the same problem. To avoid the problem of storing directional finite elements, three point methods (e.g., Aupperle and Hanrahan [1993]) have been proposed. However, these still suffer from a severe k^3 link storage requirement where k is the number of input elements. As a consequence, they are restricted to very simple scenes. Dumont et al. [1999] trade memory for time by applying a shooting approach based on the link hierarchy and removing links on the fly. This reduces the memory requirement to quadratic, but increases the computation time. Another approach is three-point clustering [Stamminger et al. 1998]; although its memory requirements are modest compared to diffuse HRC, very fine subdivision of link space is required to get high frequency results such as caustics. Such results are thus very costly both in time and storage.

Combined multipass radiosity/Monte Carlo methods have also been proposed (e.g., Shirley [1990], Chen et al. [1991], Keller [1997], Shirley [1991], Neumann [1995], and László et al. [2001]) but do not fully take advantage of the information provided by one pass to guide, and thus accelerate the next. The approaches of Shirley [1990] and Chen et al. [1991] are precursors to our unified algorithm for integrating radiosity and particle tracing in one algorithm. However, since these methods clearly separate the different types of transfers (e.g., specular reflections are treated as a separate process), they often require complex additions and subtractions of light in the various passes to avoid double counting of light transfers. In addition, these methods are based on progressive refinement radiosity, that cannot efficiently treat complex scenes in which indirect lighting is predominant. Secondary light-source reclassification [Chen et al. 1991] becomes prohibitively expensive when many patches are secondary emitters with about the same power. Also, important information is lost during the iterative process, and thus a final reconstruction step would be difficult and costly.

We have recently introduced a novel global method, the unified algorithm [Granier et al. 2000; 2001], which uses an HRC algorithm to compute diffuse light paths and also to guide and restrict particle tracing for nondiffuse paths. This results in a tight integration that provides rapid and complete simulation of all light paths. The algorithm provides information (such as links and particles) that can be easily used in a final reconstruction process. It also has a built-in quality control mechanism which allows a smooth transition from fast, lower-quality solutions to more expensive high quality lighting simulations by shifting computations higher or lower in the hierarchy. The final reconstruction will not only take advantage of all the information that is provided by such an approach (particles, links, meshes, and so one) but will also incorporate the quality variation mechanism, as in Scheel et al. [2001]. We review details of this approach in Section 4.

In what follows we use Heckbert's [1990] regular expression notation for light paths and light transport. L is used for the light, E the eye, S a directional transfer (S for specular, but also in general, all nondiffuse transfer), D a diffuse transfer, while "*" represents zero or more bounces, "+" at least one bounce, and "|" is the "or" operator. Thus for example LD^*E are light paths leaving the light, bouncing off zero or more diffuse objects, and arriving at the eye. These paths are well treated by radiosity. All possible light paths are thus described by $L(D|S)^*E$.

3. METHOD OVERVIEW

The method we present in this article is divided into two main steps. The first is an iterative process that we call the unified algorithm which unifies hierarchical radiosity with clustering (HRC) and particle tracing. This step can compute a fast global illumination solution which can then be rendered interactively, permitting an initial visualization of the final result. A second, view-dependent final reconstruction
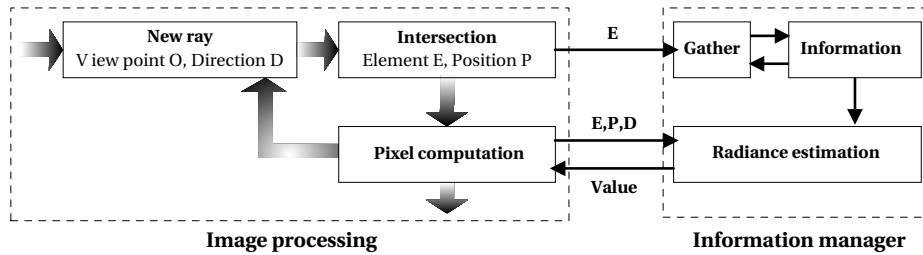
Fig. 2. Overview of our final reconstruction algorithm. "Information" refers to particles, links, whether the intersected element is a leaf of the hierarchy, and so forth.

step follows, which makes extensive use of the information coded in the data structures of the unified algorithm.

3.1 Iterative process

The unified algorithm [Granier et al. 2000] is based on the iterative process of HRC. Initially the cluster hierarchy is built as in HRC. Objects are grouped into clusters, and subdivided into subelements when this is required to accurately represent illumination. We use the generic term *element* both for clusters and for (possibly subdivided) input primitives. Other than the hierarchy itself, the main data structures used are *links*, which are the representation of light exchanges between hierarchical elements. To initialize the iterative process, a link is created from the hierarchy root to itself (self-link).

During an iteration, each link is examined and the refinement process decides whether to *refine* the link based on the chosen criterion. Such criteria are typically based on energy or power, but may also include considerations of visibility. A *gather* step follows, where each link is checked. The algorithm checks for links leaving diffuse objects and arriving on specular objects. In this case, diffuse to specular light transfers are performed by particle tracing. The use of the link structure restricts the number of particles emitted since only a limited part of space is treated (see Section 4.1). This substantially accelerates particle tracing itself by accurate visibility classification. Particles are subsequently propagated into the environment both by nondiffuse reflection and refraction.

Updating the element hierarchy completes a full iteration of this process. Particles are first placed at an appropriate level in the hierarchy (on objects or at subdivided elements, see Section 4.2). The contribution of particles is then added to the diffuse light at the leaves of the hierarchy (see Section 4.2); all transfers to diffuse surfaces are thus recorded. This can also be considered as a reconstruction step for particles. In this manner, all $L(D|S)^*DE$ light transfer is accounted for. Rendering (see Section 4.4) is performed by reconstructing the radiosity values. Three different approaches have been developed: i) reconstruction on a fine grid; ii) using the mesh subdivision to generate triangular elements; iii) creating locally high-resolution textures to increase caustic quality. Specular paths $(L|D)S^+E$ to the eye must then be added by ray tracing.

3.2 Final Reconstruction

For final reconstruction, we are in a view-dependent context; consequently, we base our algorithm on a ray tracing method. Each ray gathers existing information in the scene along its path. This information includes the links, the particles' impacts, as well as the precomputed radiance value. These elements will allow the improvement of the final results and will restrict the computation to the visually important details. We generate anti-aliased images using standard ray tracing techniques [Painter and Sloan 1989].

For each eye ray, we find the intersected object and the leaf element of the unified algorithm hierarchy. On this object, all existing information is gathered, that is, the radiosity values, the particles and the links, including their visibility and form-factor information. Note that this operation is performed only once for each element directly visible in the final image. All this information is used to compute the radiance value along the ray.

The general structure of our algorithm is shown in Figure 2. In a final gather step, only local information is actually needed to compute the pixel value. We store this information and reuse it for pixels shaded later, at the cost of increased memory consumption. With a caching scheme, this can effect a compromise between time and memory.

Since the view-independent component of the global illumination solution has already been computed by the unified algorithm, this value is used directly. If the algorithm detects that this value is not sufficiently precise, its accuracy is increased. This is a radiosity value (LD^*E paths), and the specular effects ($L(D|S)^*S^+DE$ paths) such as caustics. Note that we do not need to propagate the ray further to compute this.

The unified algorithm creates two types of information. The first are the links, which can be used to compute diffuse exchanges $(L|D)D$.[1] The second are the particles. They contain the information on exchanges through a directional reflection ($(L|D)S^+$ paths). These particles help to reconstruct both the view-independent part ($L(D|S)^*DE$ paths), and the view-dependent effects ($L(D|S)^*S^+E$ paths).

4.   UNIFIED METHOD

In this section, we review the main aspects of the unified algorithm Granier et al. [2000; 2001] to give a clear idea of the information provided to the final gather pass. We focus mainly on the issues concerning particle emission and propagation. More details can be found in Granier et al. [2000; 2001]. In the discussion that follows, we assume the reader is familiar with the details of HRC [Sillion 1995]. An appropriate introduction can also be found in Granier et al. [2000].

4.1   Particle Emission

When a link is stored on an element which is not purely diffuse, particles have to be emitted to integrate the nondiffuse transmission into the global illumination solution. These particles will also help us reconstruct the nondiffuse indirect lighting in the final gather process.

One way to proceed with the emission of particles, is to estimate the unoccluded flux between the two elements R (receiver) and S (emitter), by n_{RS} particles with the same user-defined energy φ_{ct}. In a hierarchical radiosity with clustering algorithm (e.g., Sillion [1995]) this flux is:

$$\Phi_{RS} = \varsigma_R\varsigma_S \int_S \int_R L(s,r)\frac{\tau R_r E_s}{\|s-r\|^2} dr\, ds. \tag{1}$$

In this equation, τ is the absorption factor for participating media, R_r is the "receiver factor", E_s is the "emitter factor" E_s, and finally, $\varsigma_S\varsigma_R$ are scale factors to account for the volumetric case. More details on these notations are available in the Table I. Given the standard assumption for radiosity algorithms that $L(s,r)$ is equal to B_S/π (B_S is the radiosity from S), this flux Φ_{RS} is equal to $B_SF_{RS}A_R$ (where F_{RS} is the unoccluded form-factor between R and S).

We can thus determine the number of particles that we have to send to correctly estimate this flux,

---

[1] $(L|D)$ represents either the diffuse self-emission of the surface or the diffusely reflected energy.

Table I. Notations and Values for Hierarchical Radiosity with Clustering

Notation	Name	Surface	Cluster
R_r	receiver factor	$\cos\theta_i$	1
E_s	emitter factor	$\cos\theta_o$	1
ς_S, ς_R	scale factors	Surface	$\frac{1}{4}$

The quantities $\cos\theta_i$, $\cos\theta_o$ are the cosine of the angle of the normal at the surface and the direction of propagation for the emitter and receiver, respectively.

assuming constant flux:

$$n_{RS} = \left\lfloor \frac{B_S A_R F_{RS}}{\varphi_{ct}} \right\rfloor. \tag{2}$$

Since n_{RS} is an integer, we cannot exactly represent the entire flux. We have to consider the residual:

$$res = \frac{B_S A_R F_{RS}}{\varphi_{ct}} - n_{RS}, \tag{3}$$

which can be used to send one more particle of energy φ_{ct}. Given this approach, we can estimate any flux value.

For a given link from S to R, the quantities in Eq. (2) are known: form-factor, area-factor and source radiosity. Based on Eq. (1), the following *probability density function* (pdf) has to be used to randomly select a point r on R and s on S:

$$p(s,r) = \frac{B_S\,\varsigma_R\,\varsigma_S\,\tau\,R_r\,E_s}{\|s-r\|^2}\,\frac{1}{n_{RS}\,\varphi_{ct}} = \frac{\varsigma_R\,\varsigma_S\,\tau\,R_r\,E_s}{\|s-r\|^2\,\pi\,F_{RS}\,A_R}. \tag{4}$$

Using this approach, we need to first choose a point s_0 on S according to the pdf $p'(s)$, and then a point r on R according to the conditional pdf $p'(r/s = s_0)$:

$$p'(s) = \frac{F_{sR}}{A_R F_{RS}} \text{ and } p'(r/s = s_0) = \frac{\varsigma_R\,\varsigma_S\,\tau\,R_r\,E_{s_0}}{\pi F_{s_0 R}\|r - s_0\|^2}. \tag{5}$$

In these equations F_{sR} is the form-factor between the point s and the element R.

Since computing the exact form-factor in the general case is an open problem, using this equation would be impractical. As an approximation, we sample the elements R and S uniformly ($p(r)$ is equal to $1/\mu(R)$ and $p(s)$ to $1/\mu(S)$. The measure $\mu()$ we use is area for surfaces and volume for clusters. Even if this approach is not exact, it is faster than an importance or rejection sampling algorithm, and speed is one of our main concerns. In addition, with a good refinement oracle, $p'(s)$ can be very close to a uniform density function since with most of the refinement oracles F_{sR} tends to be constant over a link.

Using the approach described above, we do not have constant energy per particle. We thus have to determine the respective power φ_{rs} per particle, given the uniform sampling we use on S and R:

$$\varphi_{rs} = \frac{1}{n_{RS}}\mu(R)\mu(S)\frac{B_S\varsigma_R\varsigma_S\tau R_r E_s}{\pi\|s-r\|^2}. \tag{6}$$

Note that, we still need to perform a visibility test when we emit the particles to remove those that will not reach R. In addition, since $(L|D)D$ transfers have already been accounted for by the diffuse transfer across links, we will only propagate particles that are not absorbed and that are reflected nondiffusely on R. As they also represent energy transfers after nondiffuse reflection, we only have to store the impact after the reflection on R.
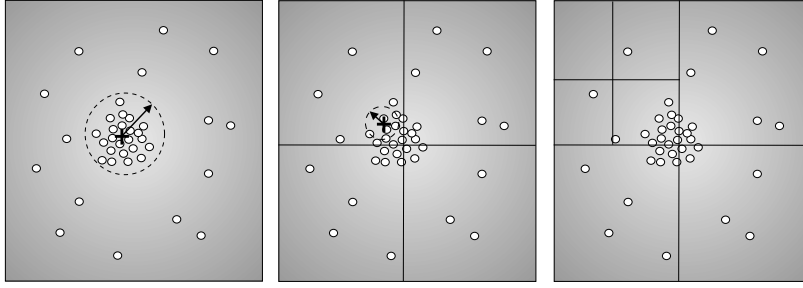
Fig. 3. Hierarchical Placement of particles.

4.2 Hierarchical Particle Placement and Energy Integration

After propagation, particles are stored at the top level of the intersected surfaces, that is, the unsubdivided input primitives. Since directional effects often contain high frequencies, we introduce a criterion to detect if the current level of particle placement results in an accurate representation of these effects. As noted in Tobler et al. [1997], placing the particles at too low a level causes visible noise, while placing them too high blurs out high-frequency directional reflection details. We introduce a criterion with a low computational cost that will determine this level.

During the push-pull step of the traditional HRC algorithm, we traverse the hierarchy. At each element, we have n particles, each carrying φ_i power and located at position x_i. We define a "center" c as the power-weighted average position of the set of particles and "spread factor" SF (see Figure 3):

$$c = \frac{\sum_n \varphi_i x_i}{\sum_n \varphi_i}, \quad SF = \frac{\sum_n \varphi_i \|c - x_i\|^2}{\sum_n \varphi_i}. \tag{7}$$

If A is the area factor [Sillion 1995], we test if

$$\pi SF \leq \zeta A \text{ and } \sum_{i=1..n} \varphi_i \geq \varepsilon, \tag{8}$$

where ε and ζ are user defined parameters[2]. Intuitively, the condition is satisfied when there is a large concentration of particles with high energy compared to the area of the element under consideration. If the condition is false, particles are placed at the current level. Otherwise the particles are pushed further down the hierarchy. This will result in additional subdivision of the hierarchy of elements if necessary, thus automatically adapting the mesh to high-frequency specular reflections.

We can now integrate the particle energy to the diffuse irradiance I of the current level:

$$I = I_d + \frac{\sum_n \varphi_i}{A}. \tag{9}$$

With such an approach, the unified algorithm can compute a view-independent solution containing all the $L(D|S)^*DE$ light paths. The particles represent the $L(D|S)^*S^+$ paths, and the links all types of incoming energy on an element. This method is view-independent and can easily be rendered to give a global overview of the solution. This can be useful to design the scene [Granier and Drettakis 2001] or to choose camera positions for the more expensive, high-quality final rendering.

---

[2]We typically use $\zeta = 0.5$, and $\varepsilon = 30\varphi_{ct}$.
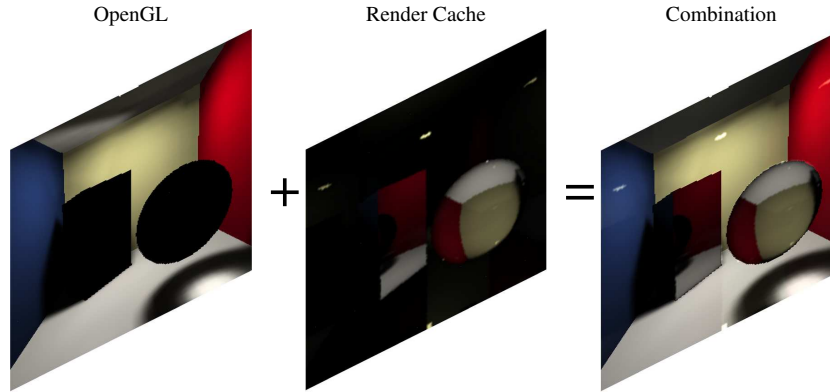
OpenGL   Render Cache   Combination

Fig. 4. Render Cache integration. We use the GPU to render only the view-independent components of the final solution (i.e., the diffuse part and the caustics), and use the CPU to compute the view-dependent part (RenderCache).

4.3 Caustic Textures

Directional effects often contain very fine details such as caustics. To correctly represent these effects a very fine mesh must be used. To avoid this additional subdivision, we can use a texture based representation which permits a local increase in detail for such effects on a coarse mesh, which is sufficient for the diffuse effects [Granier and Drettakis 2001]. This has the additional advantage of permitting better frame-rates for interactive visualization of the current solution.

If such textures are used, they can also increase the visual quality of the ray-traced view-dependent part of the image, by using not only the radiosity values on the mesh, but also by combining the radiosity on the mesh and the texture value. This approach will be used for the final reconstruction step.

4.4 View-Dependent Component

When designing a scene or choosing camera positions, it is important to visualize all light paths, including view-dependent components missing from the radiosity solution. In our system, indirect directional (nondiffuse) illumination is captured in the radiosity solution, but not direct directional reflection, that is, $(L|D)S^+E$ paths. This can be done interactively by ray-tracing, using a method such as the Render-Cache [Walter et al. 1999]. The simplest way to achieve interactive display is to use standard graphics hardware to render the view-independent part of the image (i.e., the mesh representing the solution of the unified algorithm), then to compute a texture containing the view-dependent part with the Render-Cache, and finally to use the graphics hardware to blend (add) the two components (see Figure 4).

5. FINAL RECONSTRUCTION

In this section, we present our new final reconstruction method, which includes all light paths. We first discuss reconstruction of diffuse-only illumination, we then proceed with the reconstruction of indirect nondiffuse phenomena such as caustics, and, lastly, we present the reconstruction of the view-dependent component to the eye. We use the terms *final gather* and *final reconstruction* interchangeably in what follows.

5.1 Diffuse-Only Final Gather

For a given visible point from the eye, we compute pure diffuse illumination using the links attached to the object intersected. Our method for the reconstruction of diffuse illumination is an extension of Scheel et al. [2001]. They presented a global approach to the final gather problem, strictly restricted to diffuse environments. Their method is a global one, gathering information at each mesh point of the HRC mesh. This can be very costly for complex scenes.

We therefore propose a modified method, taking further advantage of the view-dependent nature of the computation. As mentioned before, there are two principal steps. In the first step, we gather information on the intersected element. In a second step, we perform a ray-casting operation to compute the radiance value of each pixel.

Our reconstruction step allows us to compute the illumination component corresponding to the $L(D|S)^*DE$ paths (i.e., the complete view-independent solution). This includes $L(D|S)^*S^+DE$ paths due to the integration of particle energy in the unified algorithm iterations. We next describe how to gather the required information, and then how to use this data to compute the irradiance value at a given pixel.

5.1.1 *Initialization of Element Information.* For each pixel, we select the corresponding visible hierarchical element by tracing a ray. The first time this element is considered, we collect -or push-down-links arriving at the current object and all its parents in the hierarchy.

Initialization on Root Surfaces. We can consider that light exchanges that are stored with links at a high level, contain, in general, only "ambient" information, whereas those at lower levels of the hierarchy contain more rapidly varying information. In Scheel et al. [2001], all links are used. This can become prohibitively expensive for complex scenes, in which no information on visibility variation is available and little information is available on form-factor variation. This is true both for an energy-based refinement criterion (such as the *BFA* used here), and for more involved criteria based on variation or on energy bounds on the elements [Gibson and Hubbold 1996; Stamminger et al. 1997]. Therefore, we do not use high level links to compute variation at each vertex. Instead, we simply accumulate them at the root surface (i.e., input surface).

The remaining links are used to gather information about emitters that interact with this element and to initialize the information needed for final reconstruction, such as the number of samples per vertex, formfactor, visibility, as proposed by Scheel et al. [2001]. This information is not initialized for the entire scene, only for the object considered by the final gather ray-tracing step, thus restricting computation to objects visible in the final image.

Note that the link collection step is performed only once per-input-surface in the scene, when a ray from the viewpoint intersects the root element or one of its children. This approach reduces the amount of information stored compared to the method in Scheel et al. [2001].

Initialization at Vertices. We collect all emitters e, having an interaction with the current element, and store them at each vertex v. For each vertex, we compute a visibility variation $\delta_{vis}(e,v)$, and also a variation of the form-factor $\delta_f(e,v)$ from all the links attached to this vertex. We find these emitters from the links arriving at the current element or its parents, with the exception of those stored at the root element.

As in Scheel et al. [2001], we compute the form-factor $f_{e,v}$ between the emitter e and the vertex v, at which we are currently gathering information. The cost of this computation is very low compared to the cost of recomputing visibility. We obtain a good representation of the form-factor variation by interpolation of the vertex values. The form-factor variation $\delta_f(e,v)$ is given as follows:

$$\delta_f(e,v) \;=\; max_a(|f_{e,v} - f_{e,a}|), \tag{10}$$

where a is a neighboring element of vertex v, and $f_{e,a}$ is the form-factor between the emitter e and element a.

Since it is more expensive to compute visibility, we only store the average value of the neighboring elements of v:

$$vis_{e,v} = \frac{1}{n_a} \sum_a vis_{e,a}, \tag{11}$$

where n_a is the number of neighboring elements for the vertex v, and $vis_{e,a}$ is the visibility between the neighboring element a and the emitter e. The visibility information is then:

$$\delta_{vis}(e,v) = Max_a(|vis_{e,v} - vis_{e,a}|). \tag{12}$$

At this point, we differ from Scheel et al. [2001]. They compute the variation between the neighboring vertices and the current vertex, whereas we compute it between the links that interact with the current vertex. This is needed by our approach since it is restricted to the visible object. The computation is thus strictly limited to the vertices of the current visible element. We prefer to avoid computations on neighboring vertices, maintaining the strictly local nature of our approach. This allows good memory management and coherent access for very complex scenes.

We use these two variation values to compute two kinds of error, one on visibility, $\Delta_{vis}(e,v)$, and one on form-factor, $\Delta_f(e,v)$:

$$\begin{cases} \Delta_{vis}(e,v) = R_{max}\delta_{vis}(e,v)f_{e,v}B_e \\ \Delta_f(e,v) = R_{max}vis_{e,v}\delta_f(e,v)B_e, \end{cases} \tag{13}$$

where R_{max} is the maximum value of the diffuse reflection coefficient on the root element. With this, we obtain the maximum value of the error when the diffuse reflection coefficient is not constant along the surface.

With these two error terms, we determine if we need to recompute the visibility and/or the form-factor, by comparing them to a user-defined error threshold value. We also determine the number of samples $nb_{v,e}$ needed to recompute values as in Scheel et al. [2001] by the distribution of the total number on the emitters according to the irradiance which they emit, without taking visibility into account.

5.1.2 *Irradiance Computation.* The irradiance value computation for a pixel is performed in two main steps. The first is an evaluation based on the links stored at the root element. The second is an evaluation based on the information stored at the vertices. This value is then reflected diffusely to obtain the radiosity value that will be used thereafter in the radiance value computation of the pixel.

Irradiance Computation based on Root Element Links. Once information concerning an element has been gathered, we use it to compute the lighting value of each point visible in the final image. When information is available at the vertices, it is used first, and radiance due to the links stored at the root element, is added to that computed with vertex values. For the unsubdivided elements, where no such information is available, the form-factor and the visibility (when declared partial) are re-evaluated at the intersection point for the most significant links. We distribute the number of samples with respect to the unoccluded irradiance of the links.

$$n_i = \frac{f_i B_i}{\sum_j f_j B_j}. \tag{14}$$

Only those links that will have a significant contribution to illumination are selected. We evaluate the form-factor and the visibility according to this number of samples.

Usage of Vertex Information. For each emitter interacting with the element, a simple interpolation of the form-factor and visibility values stored with each link is performed when a re-evaluation is not

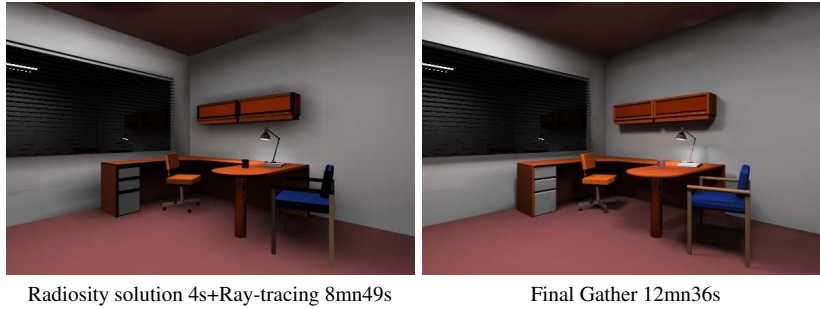Radiosity solution 4s+Ray-tracing 8mn49s   Final Gather 12mn36s

Fig. 5.   Final gather of a diffuse office scene.

needed at each vertex of the element. If a re-evaluation is needed for at least one vertex, two cases need to be considered.

In both cases, a re-evaluation of the visibility and/or the form-factor will be necessary. The number of samples to be used is determined by an interpolation of the number $nb_{v,e}$ decided for each vertex as in Scheel et al. [2001]. On a triangle with vertices v_1, v_2, v_3, the sample number that is needed at a point p, for which the interpolation coefficients are $c_{1,2,3}$, is computed according to the following equation:

$$nb_{p,e} = c_1 \cdot nb_{v_1,e} + c_2 \cdot nb_{v_2,e} + c_3 \cdot nb_{v_3,e}. \qquad (15)$$

The coefficients c_i correspond to the barycentric coordinates of the point p in the triangle v_1, v_2, v_3. When $nb_{p,e}$ is not an integer value ($nb_{p,e} > \lfloor nb_{p,E} \rfloor$), $\lfloor nb_{p,E} \rfloor + 1$ must be computed, by weighting the last sample with $nb_{p,e} - \lfloor nb_{p,E} \rfloor$. The solution thus becomes continuous.

Since we do not use a variation value computed on the neighboring vertices, our criterion may decide that a re-evaluation is needed for one vertex, while for another vertex interpolation is sufficient. When we get closer to this vertex, the number of samples given by the interpolation thus becomes zero.

To avoid this problem for a given value, (e.g., the form-factor or visibility), we use a weighted average of the stored vertex value and the value recomputed by sampling. With their global approach, Scheel et al. [2001] ensure that this case cannot appear. This is a design choice on our behalf which results in both memory and computation savings. The weight for the recomputed value is the corresponding vertex interpolation coefficient, and the weight for the recomputed value is the sum of the interpolation coefficients of the remaining vertices. For example, if the form-factor on triangle vertex v_1 has sufficient quality, but not vertex v_2 and v_3, the value resulting for the form-factor becomes:

$$f = c_1 \cdot f_{v_1,e} + (c_2 + c_3) \cdot f_{p,e}, \qquad (16)$$

where $(c_i, i = 1..3)$ are the interpolation coefficients assigned to each vertex, and $f_{p,e}$, the form-factor recomputed by sampling at the intersection point p. With this approach, we obtain a continuous transition between regions for which re-computation is needed and regions for which interpolation is sufficient.

5.1.3   *Discussion on Diffuse Parameters.* Final reconstruction of diffuse-only illumination (see Figure 5), is controlled by two somewhat inter-dependent parameters, the error threshold and the choice of the total number of samples.

The error threshold defines a set of emitters used to potentially re-evaluate the form-factor and/or the visibility according to the previously elaborated criteria. By reducing this parameter, we increase image quality since additional variations in lighting are captured. When this parameter is set to zero, all emitters belong to the set, and thus maximal image quality is achieved.
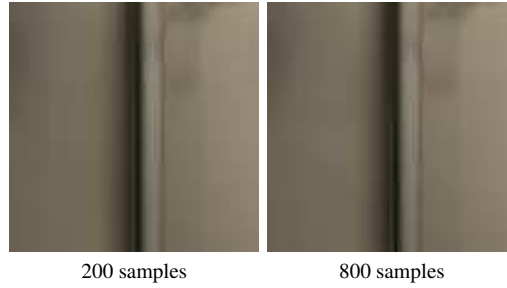
200 samples   800 samples

Fig. 6. Influence of the sample number.Banding effects diminish with a higher sample number and the soft shadows are improved.

As we spread the total number of samples according to the importance of the interaction (i.e., the unoccluded irradiance), this parameter will create a set of the emitters that can have a large influence on the total irradiance. With a small number of samples, only a subset of this set will be used. Increasing the total number of samples will result in a larger set of emitters. However, since more samples are used for each emitter, higher quality soft shadows will result (see Figure 6).

Since the error computed will increase for important emitters due to our criteria, the two sets can be slightly different. Re-evaluation is actually performed using the intersection of these two sets.

5.2   Reconstructing and Integrating Caustics

The unified algorithm we have presented in section Section 4 can deal efficiently with complex light paths, for example, indirect caustics. The use of textures for the representation of these effects (see Section 4.3), results in sufficiently high image quality for interactive display. The results, however, are not always sufficient for the quality expected in a high-quality image created for an animation or for a true photo-realistic representation of a scene. In this section, we present a solution which uses information provided by the particle trace to improve final reconstruction for caustics and for more general effects, notably $(L|D)S^+DE$ light paths.

5.2.1   *Using First-Pass Particles.* The particles placed by the first-pass unified algorithm can be sufficient to reconstruct the low-varying directional (very glossy or diffuse) component for each pixel. In this section we will focus on the diffuse part and the view-dependent part will be introduced in Section 5.3.3. Our approach for reconstruction is in the spirit of the Photon Map [Jensen 2001], but we restrict the computation to the visible elements of an image. Thus the Photon Map is not global and is attached to root elements of the scene. In the same spirit as above, we are able to restrict computation to the visible objects.

When a leaf element is considered for the first time by our final gather method, we collect all particle impacts of the entire hierarchy which fall onto this element, and we sort them in a 3D balanced binary tree [Jensen 2001]. With this structure, the search for the neighboring particles of a point becomes efficient.

Thus for each visible point p considered, the N_p closest particles will be required as in the Photon Map approach. We use an estimator similar to that presented for the Photon Map [Jensen 1996] with a modification for the irradiance calculation:

$$\tilde{I}_c(x) \; = \; \frac{l+2}{l\pi R^2} \sum_{i=1}^{N_p} \left(1 - \left(\frac{r_i(x)}{R} \right)^l \right) \varphi_i, \qquad (17)$$

where $r_i(x)$ is the distance from the particle i to the point x, φ_i the particle energy. R is the longest

distance to x for the N_p particles; It is the radius of the bounding sphere of the N_p particles, centered at x. The value l is the filtering level [3], to reduce the variance for a given number of particles. For $l = \infty$, we have a constant reconstruction kernel, without bias on its support. But visually, a smoother transition is preferred, meaning a small value of l. This value is added to the link irradiance. The irradiance value thus represents all the $L(D|S^+)^*$ paths arriving at the point x.

In the following section we will use the notation $K_{R,l}(x, i)$ to represent the caustic kernel reconstruction, that is,

$$K_{R,l}(x,i) \;=\; \frac{l+2}{l\pi R^2} \left(1 - \left(\frac{r_i(x)}{R} \right)^l \right). \tag{18}$$

The influence of the search radius and of the particles number have been discussed extensively in Christensen [2001]. The parameter l is a generalization of the filtering process of Photon Map. Using this parameter, the influence of each particle can be increased, resulting in a smoother reconstruction result.[4]

5.2.2 *Optimally Balanced kd-Tree.* To construct a compact representation, we need to balance the kd-tree as for a heap construction, instead of balancing on the median particle as in Jensen [1996]. This approach may require an array with a size equal to twice the number of particles.

If we have $N = 2^d + k$ particles, with $k < 2^d$, we balance the kd-tree on the following value:

$$\begin{cases} N - 2^{d-1} + 1 & \text{if } 2^{d-1} - 1 \geq k \\ \quad\; 2^d & \text{otherwise.} \end{cases} \tag{19}$$

With such a balancing strategy, the size for storing the particles is reduced, while keeping the same search cost when finding the nearest particles (as in Jensen [2001]). In addition, no further operation is required on the array for balancing.

5.2.3 *Area Correction.* Photon Map-like reconstruction methods may have difficulty dealing with small surfaces and the borders of surfaces [Jensen and Christensen 1995]. The global kd-tree of the original method may reduce these standard density estimation problems but, does not completely remove them. Walter [1998] has proposed involved local linear (or polynomial) density estimation methods to solve these problems resulting in precise detection of shadow boundaries.

Jensen and Christensen [1995] also note that when adding photons to the estimate near an edge, the changes of the estimate will be monotonic. Based on this observation, they suggested differential checking during the estimate. If the estimate is either constantly increasing or decreasing as more photons are added, they suggest stopping adding on photons. Due to the stochastic nature of this estimation however, it is often difficult to reliably identify a monotonic increase or decrease of the estimate.

We propose a simpler solution by trying to estimate the area of the reconstruction kernel projection on the underlying surface. In the neighborhood of a border, this projection will become smaller. For the standard reconstruction scheme, this area is constant (i.e., $l\pi R^2/(l+2)$). As a result, we tend to underestimate the irradiance. By adjusting the area value of the kernel, we can obtain a more precise value.

As shown in Figure 7, we estimate the projection by uniformly sampling the tangent disk centered on the current point x and with a radius R equal to the maximum distance to the nearest particles. For each sample, we then test the intersection between a ray with a direction parallel to the surface normal, and

---

[3]For the cone filter used in the Photon Map l=1, and for the Epanechnikov kernel, l=2

[4]We typically use l=2, since it removes the square root computation for the distance.
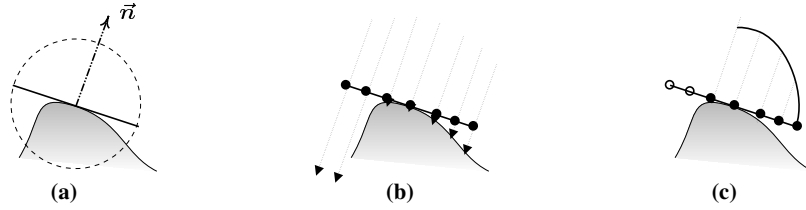
Fig. 7. Area estimation configuration. (a) Search the N nearest particles (b) Sample the tangent disk and test intersection of the surface with rays parallels to the normal (c) Estimate the disk area.

the surface itself. The resulting estimate for the area is:

$$A_{R,l}(x) \; = \; \frac{l\pi R^2}{l+2}\left(1 - \frac{l+2}{l\,N_s}\sum_{j=0}^{N_s}\delta_j\left(1 - \left(\frac{r_j(x)}{R}\right)^l\right)\right),\tag{20}$$

with N_s the number of samples chosen independently of the number of particles N_p to estimate the area, and δ_j is 1 if there is an intersection between the ray and the surface for disk sample j and 0 elsewhere. Note that for a given pattern of samples and for a given radius, $A_{R,l}(x)$ is continuous along the position on the surface as we would expect for the projected area of the kernel.

Note that, for a plane and a position where the tangent disc belongs to the surface, this estimator gives the exact value of the area. On the border, the value can be slightly overestimated (or underestimated), but will be closer to the real value and will converge to the exact value with the increase of the number of samples. By using a constant sampling pattern, and due to the continuous kernel used for the estimation, the transition between the interior of a surface and its border is also continuous.

Thus, we can adjust the reconstruction kernel (see Eq. (18)) as follows:

$$K_{R,l}(x,i) \; = \; \frac{1}{A_{R,l}(x)}\left(1 - \left(\frac{r_i(x)}{R}\right)^l\right).\tag{21}$$

This approach results in a better estimation of illumination at the edge neighborhood since our estimation of the kernel area is closer and converges to the real value. Since the sampling pattern and the weights can be precomputed, the approach introduces only a small computational overhead for significant quality improvement. An example of the improvement can be seen in Figure 8. Compared to the convex hull approach introduced in Christensen [2002], the cost is independent of the particle number and quite low due to the fact that the intersection tests are done on a single object.

5.3 View Dependent Component

With the two previous steps, we can reconstruct the entire view-independent component of the image, that is all $L(D|S)^*DE$ light paths. It is necessary to add the view-dependent lighting effects to the solution, that is, $(L|D)S^+E$ light paths, such as specular reflections, for example.

The straightforward solution is to use the distributed or Monte-Carlo ray-tracing to estimate the integral corresponding to this part of the illumination only, that is, the directional part of a BRDF. We show how this is done in the context of our approach in Section 5.3.1. Recall that we have stored particles as well as their incoming directions. This information can in some cases be sufficient to compute the indirect view-dependent lighting component (i.e., ray paths with length greater than 2, where length is the number of reflections in the path). In Section 5.3.3, we introduce a new reconstruction algorithm which allows the computation of these effects using our method. Nonetheless, in some cases, reconstruction

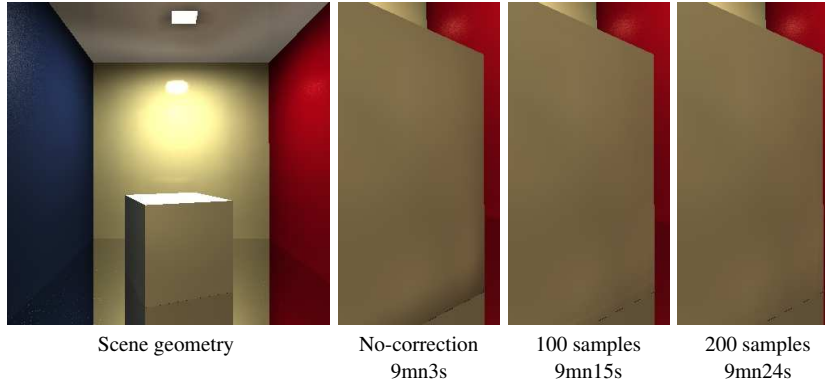| Scene geometry | No-correction 9mn3s | 100 samples 9mn15s | 200 samples 9mn24s |

Fig. 8. Test on area correction. The scene consists of a mirror floor and walls, and a diffuse cube in the middle of the floor. The entire scene is lit by a single area source in the middle on the ceiling. All illumination on the sides of the cube is due exclusively to particle impacts. Note how the illumination estimation at the borders improves due to area correction (especially near the floor).
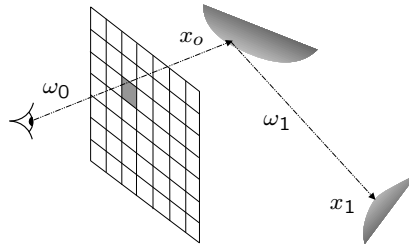

Fig. 9.   Eye ray propagation.

based on particles can result in loss of detail. To deal with this problem, in Section 5.3.4 we introduce a new algorithm to detect such situations and to combine the two previous approaches. In all configurations, the direct view-dependent part of the solution is computed by distributed ray-tracing since no information is provided for such paths by the unified algorithm.

5.3.1   *Distributed or Monte-Carlo Ray Tracing for View-Dependent Directional Lighting.* To compute the view-dependent component, we need to estimate the radiance value along $(D|L)S^+E$ paths. The corresponding equation for this value at a pixel is:

$$L(x_0, \omega_0) = \int_\Omega \rho_s(x_0, \omega_1, \omega_0) < \omega_1 \cdot n_0 > L(x_1, -\omega_1) d\omega_1, \tag{22}$$

where x_0 and n_0 are respectively the intersection point from an eye ray emitted in direction ω_0 (see Figure 9) and the normal at this point, Ω is the set of all directions, x_1 is the first intersecting point in direction ω_1 from x_0, and ρ_s is the directional part of the BSDF [Heckbert 1990]; $< \cdot >$ denotes the dot product. To simplify the equation, we will use the notation $T_s(x, \omega', \omega)$ for $\rho_s(x, \omega', \omega)| < \omega' \cdot n > |$.

The light emitted or reflected by a surface $L(x, \omega)$, can be written as the sum of two components, the radiance from a single directional reflection $L_s(x, \omega)$ and the rest $L_k(x, \omega)$, which is the sum of the diffuse component $B(x)/\pi$ and the self-emitted directional radiance $L_p(x, \omega)$. $L_k(x, \omega)$ is known, since $B(x)$ has already been computed by our unified algorithm, and the quantity $L_p(x, \omega)$ is an input data. We thus

have:

$$L(x, \omega) \;=\; L_k(x, \omega) + L_s(x, \omega), \tag{23}$$

After the unified algorithm computation, only the value of $L_s(x, \omega)$ is unknown. It corresponds to the integral

$$L_s(x, \omega) \;=\; \int_\Omega T_s(x, \omega', \omega) L(x', -\omega') d\omega'. \tag{24}$$

We can now write from Eqs. (22), (23) and (24)

$$L(x_0, \omega_0) \;=\; \int_\Omega T_s(x_0, \omega_1, \omega_0) L_k(x_1, -\omega_1) d\omega_1 + L_{s,2^+}(x_0, \omega_0), \tag{25}$$

where

$$L_{s,2^+}(x_0, \omega_0) \;=\; \int_\Omega \int_\Omega T_s(x_0, \omega_1, \omega_0) T_s(x_1, \omega_2, \omega_1) L(x_2, -\omega_2) d\omega_1 d\omega_2. \tag{26}$$

$L_{s,n^+}(x, \omega)$ represents paths through the directional part of the BRDF with length longer than n, and $L_{s,n}(x, \omega)$ represents paths through the directional part of the BRDF with length equal n. $L_{s,2^+}(x, \omega)$ can be evaluated if the eye ray continues its propagation until it leaves the environment or hits a purely diffuse surface.

5.3.2 *Separation of Direct From Indirect Illumination.* The previous separation can be done also along the path of a ray. The radiance value computed along such a path m of length $n \geq 1$ is:

$$L_{s,n,m}(x_0, \omega_0) \;=\; \sum_{i=0}^{n-1} L_k(x_{i+1}, \omega_{i+1}) \Pi_{j=0}^{i} \frac{T_s(x_j, \omega_{j+1}, \omega_j)}{p(\omega_{j+1})}, \tag{27}$$

where $p(\omega)$ is the probability density for reflecting in direction ω. This function is, in general, proportional to $\rho_s(x_j, \omega, \omega_j)$. The length n is selected such that $n \geq 1$ because we are computing the value along a path with at least one directional reflection. We do not need to compute diffuse values because this is already done in the previous steps.

To simplify the presentation of the combinations in the following sections, the Equation (27) for a path m can be rewritten as:

$$\begin{aligned}
L_{s,n,m}(x_0, \omega_0) \;=\;\; & \frac{T_s(x_0, \omega_1, \omega_0)}{p(\omega_1)} L_k(x_1, \omega_1) + \frac{T_s(x_0, \omega_1, \omega_0)}{p(\omega_1)} \cdot \\
& \sum_{i=1}^{n-1} L_k(x_{i+1}, \omega_{i+1}) \Pi_{j=1}^{i} \frac{T_s(x_j, \omega_{j+1}, \omega_j)}{p(\omega_{j+1})} \\
\;=\;\; & L_{s,1,m}(x_0, \omega_0) + \frac{T_s(x_0, \omega_1, \omega_0)}{p(\omega_1)} \cdot L_{s,n-1,m}(x_1, \omega_1) \\
\;=\;\; & L_{direct,m}(x_0, \omega_0) + L_{indirect,m}(x_0, \omega_0).
\end{aligned} \tag{28}$$

$L_{direct,m}(x_0, \omega_0)$ is the direct directional reflection of the environment (paths with length 1), and $L_{indirect,m}(x_0, \omega_0)$ represent paths in which there are at least 2 directional reflections. Note that if $n = 1$, $L_{indirect,m}(x_0, \omega_0) = 0$.

5.3.3 *Usage of Particle Direction.* When particles exist, they contain information on the $(L|D)S^+$ paths. In Section 5.2, a diffuse reflection enabled us to calculate the caustic value. In the same way, we can compute the value of the paths which are represented by the expression $L_{s,2^+}(x, \omega_0)$. We can
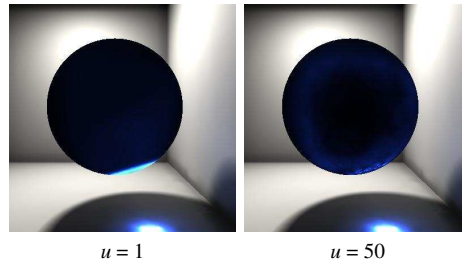
$u = 1$ $u = 50$

Fig. 10.   Particle parameters influence. u is the level of directional smoothing (see Section 5.3.3-Eq. (30))

compute it directly from the particles as is done in the Photon Map [Jensen 1996], and by an estimator similar to Eq. (17):

$$\tilde{L}_{s,2^+}(x,\omega_0) \;=\; \sum_{i=1}^{N_p} T_s(x,d_i,\omega_0) K_{R,f}(x,i)\varphi_i \tag{29}$$

In this equation, d_i is the incident direction of particle i and n the normal on the surface at point p.

For highly directional materials however, this estimator will fail as the probability of $T_s(x,d_i,\omega_0)$ being nonzero is very low (and null for a mirror). This results in high variance. We instead choose to reconstruct the incoming radiance function from the particles. We then use the following irradiance estimator $\tilde{I}$, in the same spirit as that for computing the irradiance volume for nondiffuse surfaces [Greger et al. 1998]:

$$\tilde{I}(x,\omega) \;=\; \frac{u+1}{2\pi} \sum_{i=1}^{N_p} K_{R,f}(x,i) \left(|< d_i \cdot \omega >|^+ \right)^u \varphi_i \tag{30}$$

where $|\;|^+$ represents the positive function (i.e., $|x|^+ = x$ for x positive, elsewhere $|x|^+ = 0$). In this equation, the parameter u controls the interpolation between the particle direction. Decreasing u will result in a smoother solution (see Figure 10). When u increases, noise can appear as in a standard Photon Map reconstruction. u is then experimentally chosen by the user to be the smallest without noise. The reflected radiance is then estimated by sampling the directional part of the material property:

$$\tilde{L}_{s,2^+}(x,\omega_0) \;=\; \frac{1}{N_b} \sum_{j=1}^{N_b} \frac{T_s(x,w_j,\omega_0)}{p(w_j)} \tilde{I}(x,w_j), \tag{31}$$

Here, N_b controls the accuracy of the BRDF sampling. Thus, with this evaluation, the view-dependent component is computed with only one direct reflection instead of a complete propagation in the scene. This approach ensures the locality of our approach, and also reduces noise for such paths.

In the following sections, we will use the notation:

$$H_{R,f,u}(x,i,\omega) \;=\; K_{R,f}(x,i)\frac{u+1}{2\pi} \left(|< d_i \cdot \omega >|^+ \right)^u. \tag{32}$$

5.3.4   *Combining With Rays.*   However, the particle information is not always sufficient to achieve the desired quality. We now introduce a method to increase the final quality by combining the previous estimator with a ray-based estimator.

We will define an error to detect regions where additional computation is required. We develop a

criterion based on variance reduction. When the estimator variance becomes too high, that is,

$$\sqrt{\left| \frac{N_p}{N_b} \sum_{j=1}^{N_b} \sum_{i=1}^{N_p} \left(\frac{T_s(x, w_j, \omega_0)}{p(w_j)} H_{R,f,u}(x, i, w_j) \varphi_i \right)^2 - L_{s,2^+}(x, \omega_0)^2 \right|} > \varepsilon_{max}, \qquad (33)$$

where ε_{max} is a user defined parameter, the previous estimator $L_{s,2^+}(x, \omega_0)$ is combined (as in Suykens and Willems [1999a; 1999b]) with a new estimator $\tilde{L}'_{s,2^+}(x, \omega_0)$, implemented by the propagation of N_r eye rays [5], which are reflected at least twice:

$$\tilde{L}'_{s,2^+}(x, \omega_0) \;=\; \frac{1}{N_r} \sum_{m=1}^{N_r} L_{indirect,m}(x_0, \omega_0). \qquad (34)$$

The combined estimator is:

$$\tilde{L}_s(x, \omega_0) \;\simeq\; \frac{1}{N_r} \sum_{m=1}^{N_r} L_{direct,m}(x_0, \omega_0) +$$
$$\frac{N_p}{N_r + N_p} \tilde{L}_{s,2^+}(x, \omega_0) \qquad (35)$$
$$\frac{N_r}{N_r + N_p} \frac{1}{N_r} \sum_{m=1}^{N_r} L_{indirect,m}(x_0, \omega_0).$$

The weighting scheme ensures a suitable behavior of the correction. In particular, when the number of rays increases, their influence becomes more significant. As a result, the estimation will converge to a correct, unbiased solution. When we emit a small number of rays compared to the numbers of particles to correct the first estimate, the particle-based calculation dominates computation time. In this case, since the number of particles is larger than the number of rays, particles account for more visual detail than the rays. If the number of rays is larger than the number of particles, the influence of rays dominates. The particles can be considered as already propagated light rays. The particles and the rays combined improve final image quality therefore as we shall show in the results.

6. RESULTS

All tests were carried out on an Intel-PC workstation (1GHz Pentium-III processor). We use a simple anti-aliasing method: a user-defined number of rays is sent into the environment for each pixel. If the intersected surfaces are identical, only one radiance value is computed. If not, an average of the values of each ray is used.

Our results show that the new approach can compute high quality images. Compared to typical Monte-Carlo approaches, the main advantage is a diffuse component with fewer noise artifacts, since it is based on a deterministic finite element solution. This solution contains about the same information as in the Ward irradiance caching algorithm [Ward and Heckbert 1992].

Moreover, with this approach, we can efficiently deal with the hard-to-simulate indirect lighting configurations, as will be shown in the tests presented below. Note that our implementation has not been optimized, in particular, we use a bounding box hierarchy for ray-tracing acceleration. With a grid, or a recursive grid, we could potentially obtain better computation times. Note also that we are not using a full-scene anti-aliasing scheme. We are currently using multisamples only for textured objects and when an edge is detected.

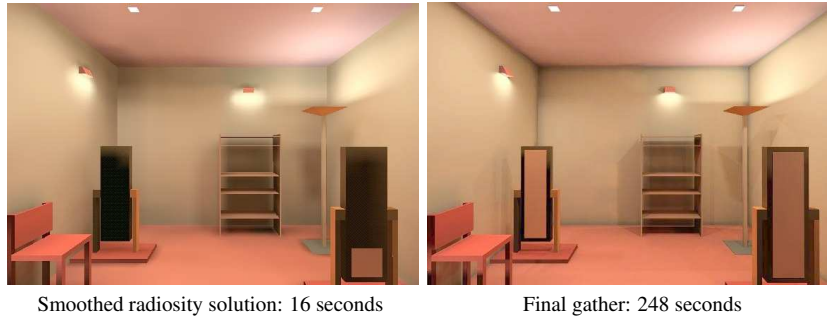[5]Note that this parameter is also experimentally defined by the user.

Smoothed radiosity solution: 16 seconds   Final gather: 248 seconds

Fig. 11.   Diffuse reconstruction on a lounge scene. (195 initial polygons, 7,300 subdivided polygons, image size 600x400)

Unified method (smoothing+texturing) 24s   Final gather 13mn
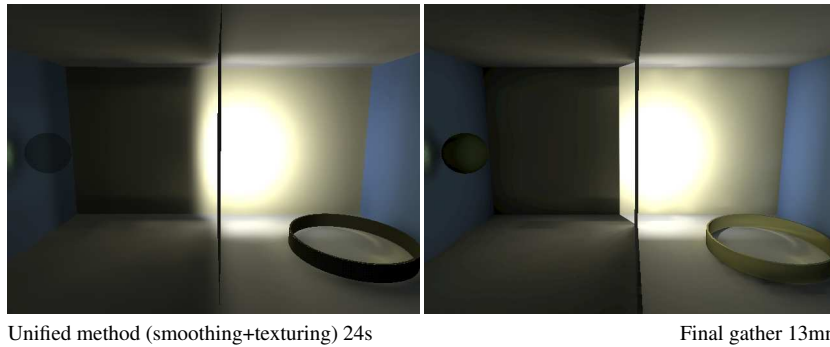
Fig. 12.   Indirect lighting of a two rooms scene. (13 initial objects, 11,105 subdivided objects, image size 640x480)

6.1   Diffuse Only

Our first tests are on diffuse only environments. Some results are shown in Figure 5. The quality improvement is clearly visible, the shadows are particulary well represented. Another example is shown in Figure 11, where we also show computation times. We can see that it is possible to obtain images of high quality with reasonable computation time. Moreover, with our view-dependent approach for the final reconstruction, it is possible to deal with complex scenes with reasonable memory resources (200 Mb for Figure 16). The solution is well adapted to a mainly indirect diffuse lighting (see Figure 15).

6.2   Caustics and View-Dependent Component

We have tested the method presented in Section 5.2.1 and that in Sect. 5.3.1 on a simple scene with mainly indirect lighting (see Figure 12), where traditional ray tracing methods have difficulty finding the light paths. The environment is mainly lit by a source directed toward the back wall which then illuminates the scene. A caustic is visible on the right room floor, due to the ring, and on the left wall, due to the refractive sphere.

The second scene is a dining room (see Figure 13) containing several objects that project caustics such as glasses and metal pans. In this scene, we test the combination of particle and ray information for the final gather. To demonstrate the advantages of the combination of particles and rays for the indirect directional component, we used a corridor scene with a glossy floor and whiteboards on the walls. We can clearly see the reflection of light sources on the floor through the whiteboards. In Figure 14, we show

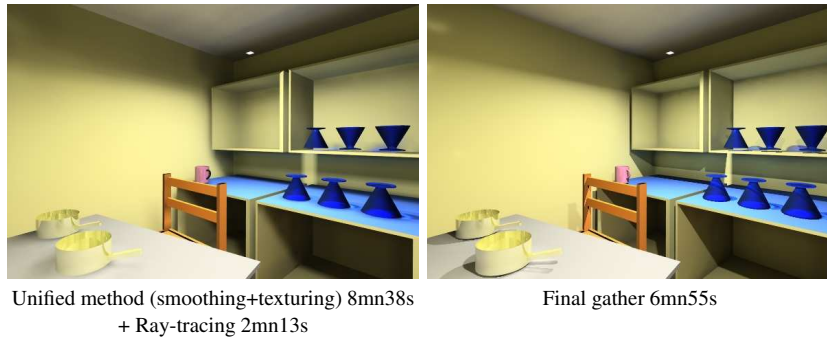| Unified method (smoothing+texturing) 8mn38s + Ray-tracing 2mn13s | Final gather 6mn55s |

Fig. 13.   A dining room. (483 initial objects, 24,000 subdivided objects, image size 600x400)

the three reconstruction methods: the method only based on rays, that based only on particles, and the method combining the two. The first solution is very noisy but more precise than the second solution. The second solution is less noisy but some fine detail is lost. The combined solution is clearly better for both noise and detail.

We have also performed tests to validate our area correction approach for a better illumination estimation (see Figure 8). This is a simple scene with mainly specular objects, with a diffuse-only cube in the middle. The indirect directional illumination reconstruction on the cube show underestimated illumination on the edges. This problem is solved when we increase the number of samples allowing correct estimation of the real area of the density estimation kernel. There is a small increase in computation time, resulting in significant increase in image quality.

6.3   Complex Scenes

We have tested our approach on complex scenes in order to validate our view-dependent approach. These tests show that our approach can deal with quite complex scenes using reasonable amounts of memory. The first scene (see Figure 15) is the "Soda Shoppe" scene, modified with light sources oriented toward the ceiling to force indirect lighting in a complex diffuse scene.

The second scene is a ship interior (see Figure 16). This scene is mainly composed of small objects, in particular the pipes and the valves. This test shows that the algorithm reacts well to such complexity.

6.4   Industrial Scene

We have also tested our approach on a scene used in industrial applications, that includes a wide range of materials, from glossy to transparent, for uniformly diffuse to textured objects (see Figure 17). Note that in this scene, reflection through the glass cupboard panes (background-right part of the image, close to the clock) has been computed using only the particle information , without combining it with ray-tracing. Given that this material is perfectly transparent, a traditional Photon Map , using a reconstruction based only on the particles, will have difficulty on these surfaces (see Section 5.3.3). With our approach, we are able to reconstruct good approximation of the lighting.

7.   CONCLUSION

In this article, we have presented a fully integrated method for final reconstruction of all global illumination paths. This algorithm is a two-pass approach: the first pass is our unified algorithm followed by a view-dependent final reconstruction pass. The unified algorithm combines Hierarchical Radiosity with

Rays
30mn 22s
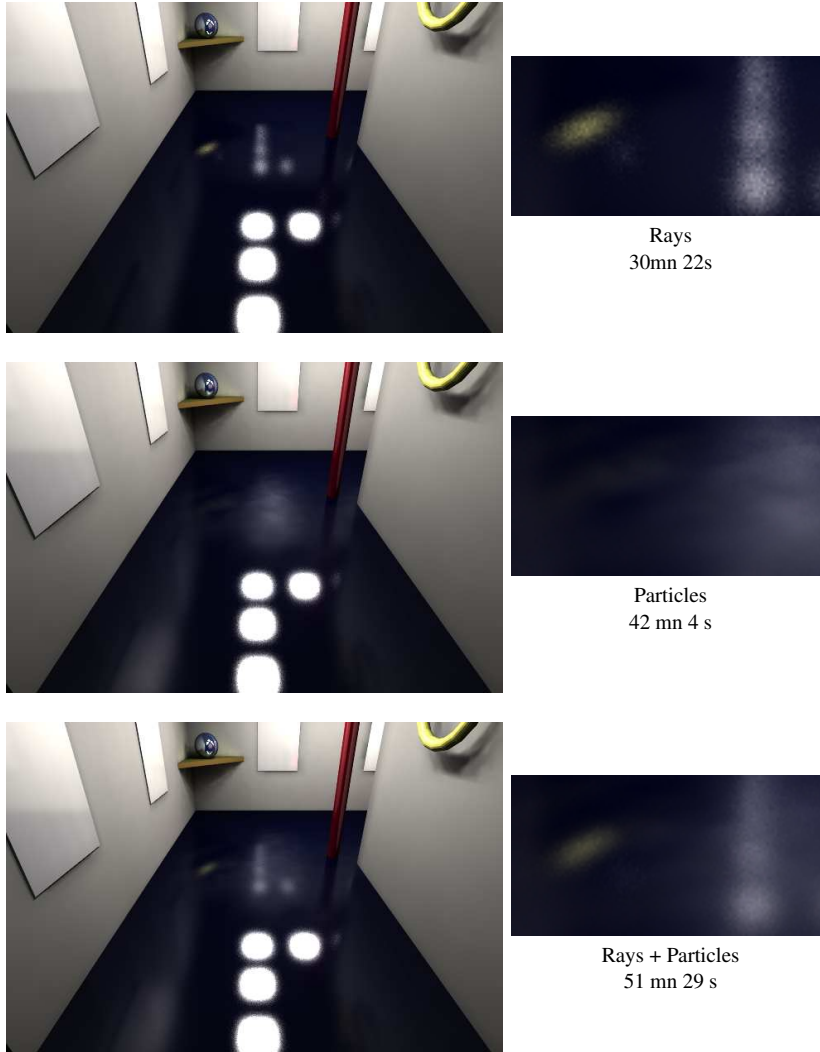

Particles
42 mn 4 s

Rays + Particles
51 mn 29 s

Fig. 14. Corridor scene. (424 initial objects, 6,721 subdivided objects, image 600x400 , 800 particles for reconstruction, 800 rays per pixel, 20 samples for indirect specular, 9 samples for anti-aliasing)

Clustering to compute diffuse exchanges and to guide an integrated particle-tracing for all light interactions other than pure diffuse. The first pass provides a large amount of useful information, which is subsequently used to guide the view-dependent final reconstruction step.

In this step, based on existing methods that have been both improved and adapted in the context of our new approach, all the relevant information from the first pass is collected for each pixel of the final image and then used to improve the radiance estimate, resulting in a high quality image. All of the information is stored efficiently, requiring a modest amount of memory.

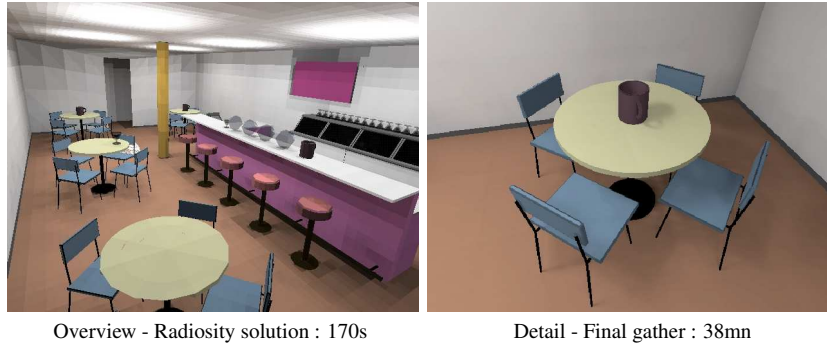Through our implementation and the results presented, we have demonstrated that our approach can

Overview - Radiosity solution : 170s   Detail - Final gather : 38mn

Fig. 15. Diffuse final gather on an indirectly lit "Soda Shoppe". (1,781 initial objects, 44,670 subdivided objects, image size 640x480)

A corridor: 5 hours and 8 minutes   Some valves: 2 hours and 8 minutes
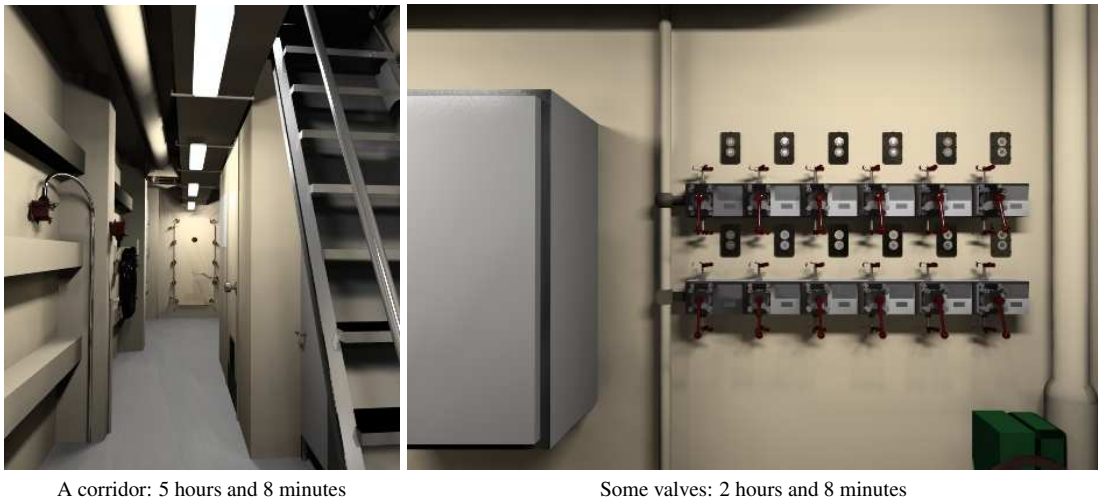
Fig. 16. Two viewpoints of a ship interior (167,708 initial objects, 270,691 subdivided objects, 23 minutes for the unified algorithm, image size 600x400)

compute high quality images, even for complex scenes.

Limitations and Future Work

Despite its high capability in computing a solution for complex light paths, this approach still has a number of limitations. Most of them are inherited from the Hierarchical Radiosity with Clustering algorithm such as problems related to the meshing of general scenes, a relatively large number of parameters that can have a significant effect on solution quality or running time and a complex implementation.

Thus, even with the *unification* of different global illumination techniques, a non-negligible number of parameters still need to be set to achieve a final image. Some of these parameters can also be adapted to a local configuration instead of being defined only globally. This can be done by a user, but we believe that an algorithm that automatically chooses these parameters can be developed, possibly based on perceptual techniques.

Fig. 17.   Kitchen scene (©LightWork Design Ltd).

Another potential avenue of research is the development of a new refinement oracle specifically designed for the combined diffuse and directional case. It will place the links at some better levels of the hierarchy, sufficient for providing more accurate information for reconstruction, and will provide better detection and guidance for particle emission. A good oracle will also provide some more understandable parameters. A variation-based method [Gibson and Hubbold 1996; 1997; Stamminger et al. 1997] could be a suitable basis for such an approach.

It would also be interesting to develop particle emission schemes resulting in closer-to-constant energy values for each particle. In the current scheme, the power carried by different particles can vary significantly. Lower variation would increase the accuracy of the final reconstruction.

We believe that our entirely view-dependent and unified approach will open directions for better user-control of the final result of global illumination algorithms. Such tools could include local control of the solution in image space, the softness or the quality of a shadow, or the local increase of the quality of a caustic. Tools effecting such local parameter controls are indispensable to allow global illumination systems to be truly usable in real world applications.

REFERENCES

ALONSO, L., CUNY, F., AND S. PETITJEAN, J.-C. P. 2000. The virtual mesh: A geometric abstraction for efficiently computing radiosity. Tech. Rep. A00-R-087, INRIA, France.

AUPPERLE, L. AND HANRAHAN, P. 1993. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Computer Graphics (SIGGRAPH '93 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 27. ACM SIGGRAPH, New York, 155–162.

BEKAERT, P., DUTRE, P., AND WILLEMS, Y. D. 1998. Final radiosity gather step using a monte carlo technique with optimal importance sampling. Tech. Rep. CW275, Department of Computer Science, Katholieke Universiteit Leuven, Leuven, Belgium. November.

CHEN, S. E., RUSHMEIER, H. E., MILLER, G., AND TURNER, D. 1991. A progressive multi-pass method for global illumination. In *Computer Graphics (SIGGRAPH '91 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 25. ACM, New York, 165–174.

CHRISTENSEN, P. H. 2001. Making the photon map faster. In *ACM SIGGRAPH '2001 Course Notes - A Practical Guide to Global Illumination using Photon Mapping*. ACM, New York, 91–114. Course 38.

CHRISTENSEN, P. H. 2002. Photon mapping tricks. In *ACM SIGGRAPH '2002 Course Notes - A Practical Guide to Global Illumination using Photon Mapping*. ACM, New York, 93–121. Course 43.

CHRISTENSEN, P. H., LISCHINSKI, D., STOLLNITZ, E. J., AND SALESIN, D. H. 1997. Clustering for glossy global illumination. *ACM Trans. Graph. 16,* 1 (Jan.), 3–33.

DUMONT, R., BOUATOUCH, K., AND GOSSELIN, P. 1999. A progressive algorithm for three point transport. *Comput. Graph. Forum 18,* 1 (Mar.), 41–56.

GIBSON, S. AND HUBBOLD, R. J. 1996. Efficient hierarchical refinement and clustering for radiosity in complex environments. *Comput. Graph. Forum 15,* 5 (Dec.), 297–310.

GIBSON, S. AND HUBBOLD, R. J. 1997. Perceptually driven radiosity. *Comput. Graph. Forum 16,* 2 (June), 119–128.

GORAL, C. M., TORRANCE, K. K., GREENBERG, D. P., AND BATTAILE, B. 1984. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH'84 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 18. ACM SIGGRAPH, New York, 213–222.

GRANIER, X. AND DRETTAKIS, G. 1999. Controlling memory consumption of hierarchical radiosity with clustering. In *Graphics Interface 1999*. Canadian Human-Computer Communications Society, Morgan Kaufmann, San Francisco, Calif., 58–65. http://www.dgp.toronto.edu/gi99/papers/146.

GRANIER, X. AND DRETTAKIS, G. 2001. Incremental updates for rapid glossy global illumination. In *Computer Graphics Forum (Proceedings of Eurographics 2001)*. Eurographics Annual Conference Series, vol. 20. Eurographics, Blackwell Publishers, Oxford, UK.

GRANIER, X., DRETTAKIS, G., AND WALTER, B. 2000. Fast global illumination including specular effects. In *Rendering Techniques 2000 (Proceedings of the 11th Eurographics Workshop on Rendering)*. Eurographics, Springer-Verlag, New York, 47–58.

GREGER, G., SHIRLEY, P., HUBBARD, P. M., AND GREENBERG, D. P. 1998. The irradiance volume. *IEEE Computer Graphics and Applications 18,* 2 (March/April), 32–43.

HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. 1991. A rapid hierarchical radiosity algorithm. In *Computer Graphics (ACM SIGGRAPH'91 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 25. ACM, New York, 197–206.

HECKBERT, P. S. 1990. Adaptive radiosity textures for bidirectional ray tracing. In *Computer Graphics (ACM SIGGRAPH'90 Conference Proceedings)*, F. Baskett, Ed. Computer Graphics Annual Conference Series, vol. 24. ACM, New York, 145–154.

IMMEL, D. S., COHEN, M. F., AND GREENBERG, D. P. 1986. A radiosity method for non-diffuse environments. In *Computer Graphics (ACM SIGGRAPH'86 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 20. ACM, New York, 133–142.

JENSEN, H. W. 1996. Global illumination using photon maps. In *Rendering Techniques'96 (Proceedings of the 7th Eurographics Workshop on Rendering)*. Eurographics, Springer-Verlag, New York, 21–30.

JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. AK Peters, LTD., Natick.

JENSEN, H. W. AND CHRISTENSEN, N. J. 1995. Photon Maps in bidirectional Monte Carlo ray tracing of complex objects. *Comput. & Graph. 19,* 2, 215–224.

KAJIYA, J. T. 1986. The rendering equation. In *Computer Graphics (ACM SIGGRAPH '86 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 20. ACM, New York, 143–150.

KELLER, A. 1997. Instant radiosity. In *Computer Graphics (ACM SIGGRAPH '97 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 31. ACM, New York, 49–56.

LÁSZLÓ, S.-K., FERENC, C., AND GÖRGY, A. 2001. Global illumination as a combination of continuous random walk and finite-element based iteration. In *Computer Graphics Forum (Proceedings of Eurographics 2001)*. Vol. 20. Eurographics, Blackwell Publishers, Oxford, UK, 288–298.

MICHELIN, S., PIRANDA, B., AND ARQUES, D. 2000. Overlapping radiosity: Using a new function base with local disk support. In *Eighth International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media (WSCG 2000)*. University of West Bohemia, Plzen, Czech Republic. Available from http://wscg.zcu.cz/wscg2000.

NEUMANN, L. 1995. Monte carlo radiosity. *Computing 55,* 1, 23–42.

PAINTER, J. AND SLOAN, K. 1989. Antialiased ray tracing by adaptive progressive refinement. In *Computer graphics (ACM SIGGRAPH'89 Conference Proceedings)*. Computer Graphics Annual Conference Series. ACM, New York, 281–288.

SCHEEL, A., STAMMINGER, M., AND SEIDEL, H.-P. 2001. Thrifty final gather for radiosity. In *Rendering Techniques 2001 (Proceedings of the 12th Eurographics Workshop on Rendering)*. Springer-Verlag, New York, 1–12.

SHIRLEY, P. 1990. A ray tracing method for illumination calculation in diffuse-specular scenes. In *Graphics Interface*. Canadian Human-Computer Communications Society, Morgan Kaufmann, San Francisco, Calif., 205–212.

SHIRLEY, P. 1991. Radiosity via ray tracing. In *Graphics Gems II*, J. Arvo, Ed. Academic Press Professional, Boston, MA, 306–310.

SILLION, F. X. 1995. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Trans. Visual. Comput. Graph. 1,* 3, 240–254.

SILLION, F. X., ARVO, J. R., WESTIN, S. H., AND GREENBERG, D. P. 1991. A global illumination solution for general reflectance distributions. In *Computer Graphics (ACM SIGGRAPH'91 Conference Proceedings)*. Computer Graphics Annual Conference Series, vol. 25. ACM, New York.

SMITS, B., ARVO, J., AND GREENBERG, D. 1994. A clustering algorithm for radiosity in complex environments. In *Computer Graphics Proceedings*. Computer Graphics Annual Conference Series. ACM, Ney York, 435–442.

STAMMINGER, M., SCHEEL, A., GRANIER, X., PEREZ-CAZORLA, F., DRETTAKIS, G., AND SILLION, F. X. 1999. Efficient glossy global illumination with interactive viewing. In *Graphics Interface 1999*. Canadian Human-Computer Communications Society, Morgan Kaufmann, San Francisco, Calif., 50–57. http://www.dgp.toronto.edu/gi99/papers/145.

STAMMINGER, M., SCHIRMACHER, H., SLUSALLEK, P., AND SEIDEL, H.-P. 1998. Getting rid of links in hierarchical radiosity. In *Computer Graphics Forum (Proceedings of Eurographics'98)*. Eurographics Annual Conference Series, vol. 17. Eurographics, Blackwell Publishers, Oxford, UK.

STAMMINGER, M., SLUSALLEK, P., AND SEIDEL, H.-P. 1997. Bounded radiosity - illumination on general surfaces and clusters. In *Computer Graphics Forum (Proceedings of Eurographics'97)*, D. Fellner and L. Szirmay-Kalos, Eds. Eurographics Annual Conference Series, vol. 16. Eurographics, Blackwell Publishers, Oxford, UK.

STAMMINGER, M., SLUSALLEK, P., AND SEIDEL, H.-P. 1998. Three point clustering for radiance computations. In *Rendering Techniques'98 (Proceedings of the 9th Eurographics Workshop on Rendering)*. Eurographics, Springer-Verlag, New York, 211–222.

SUYKENS, F. AND WILLEMS, Y. D. 1999a. Combining bidirectional path tracing and multipass rendering. In *WSCG '99 (Seventh International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media)*. University of West Bohemia, Plzen-Borey, Czech Republic, 265–272.

SUYKENS, F. AND WILLEMS, Y. D. 1999b. Weighted multipass methods for global illumination. In *Computer Graphics Forum (Proceedings of Eurographics'99)*. Vol. 18. Eurographics, Blackwell Publishers, Oxford, UK.

TOBLER, R. F., WILKIE, A., FEDA, M., AND PURGATHOFER, W. 1997. A hierarchical subdivision algorithm for stochastic radiosity methods. In *Rendering Techniques '97 (Proceedings of the 8th Eurographics Workshop on Rendering)*. Eurographics, Springer-Verlag, New York, 193–204.

VEACH, E. 1997. Robust monte-carlo methods for light transport simulation. Ph.D. dissertation, Stanford Uni.

WALTER, B., HUBBARD, P. M., SHIRLEY, P., AND GREENBERG, D. P. 1997. Global illumination using local linear density estimation. *ACM Trans. Graph. 16,* 3 (July), 217–259.

WALTER, B. J. 1998. Density estimation techniques for global illumination. Ph.D. dissertation, Cornell University. http://www.graphics.cornell.edu/pubs/1998/Wal98.html.

WALTER, B. J., DRETTAKIS, G., AND PARKER, S. 1999. Interactive rendering using the render cache. In *Rendering Techniques'99 (Proceedings of the 10th Eurographics Workshop on Rendering)*. Eurographics, Springer-Verlag, New York.

WARD, G. J. AND HECKBERT, P. 1992. Irradiance gradients. In *Rendering Techniques'92 (Proceedings of the 3rd Eurographics Workshop on Rendering)*. Eurographics, Consolidation Express Bristol, Bristol, 85–98.

WILLMOTT, A. J., HECKBERT, P., AND GARLAND, M. 1999. Face cluster radiosity. In *Rendering Techniques'99 (Proceedings of the 10th Eurographics Rendering Workshop)*. Eurographics, Springer-Verlag, New York, 293–304.