



HAL
open science

Fast Global Illumination Including Specular Effects

Xavier Granier, George Drettakis, Bruce Walter

► **To cite this version:**

Xavier Granier, George Drettakis, Bruce Walter. Fast Global Illumination Including Specular Effects. Proceedings of the Eleventh Eurographics Workshop on Rendering, Eurographics, 2000, Brno, Czech Republic. pp.47-58, 10.1007/978-3-7091-6303-0_5 . inria-00510057

HAL Id: inria-00510057

<https://inria.hal.science/inria-00510057>

Submitted on 17 Aug 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fast Global Illumination Including Specular Effects

Xavier Granier¹, George Drettakis¹ and Bruce Walter²

¹ iMAGIS-GRAVIR/IMAG-INRIA

iMAGIS is a joint research project of CNRS/INRIA/UJF/INPG

² Cornell Program of Computer Graphics

E-mail: {Xavier.Granier|George.Drettakis}@imag.fr, bjw@graphics.cornell.edu

Abstract.

Rapidly simulating global illumination, including diffuse and glossy light transport is a very difficult problem. Finite element or radiosity approaches can achieve interactive simulations for some classes of diffuse-only scenes, but more general methods are currently too slow and too noisy for interactive use.

We present a new method which seamlessly integrates particle tracing (for non-diffuse transport) into the gather step of hierarchical radiosity (for diffuse transport) to efficiently handle all types of light transport chains. Our integrated approach results in rapid, good visual quality solutions. This is achieved using a radiosity algorithm producing smooth, noise free simulation of diffuse light transfers, and an integrated particle trace for rapid, high quality specular reflections such as caustics.

Using our system, users can interactively visualize and manipulate small environments with global illumination including specular effects. Such general lighting effects can also be simulated for larger environments, albeit at a higher computational cost. Our system can also treat scenes which are lit mainly by indirect lighting, which is very hard using previous methods. With our method, smooth transition from fast, low quality to slower high quality solutions is possible.

1 Motivation

Simulating global illumination is essential for realistic rendering of virtual environments. The incorporation of indirect lighting effects in particular can be very important for many classes of scenes. For diffuse-only scenes, recent radiosity techniques can achieve interactive speeds with moving objects. Real environments however are rarely completely diffuse, and the use of such simulations results in images which are missing many important non-diffuse or glossy effects. The algorithms which treat specular or glossy effects (e.g., caustics) are still too slow and too noisy for interactive applications which require overall high visual quality. Non-diffuse light exchanges contain highly directional components, and are thus unsuitable for discrete, finite-element type methods. As a consequence, they are often treated by stochastic methods or multi-pass approaches, which combine finite element and stochastic solutions. Both are generally too slow to interactively handle dynamic scenes. The former suffer from objectionable noise artifacts while the latter are usually limited in the effects or the environments they can simulate, and often require a series of complex steps to treat all light paths.

The goal of our work is to provide fast, high quality global illumination simulation including both glossy and diffuse effects. We achieve this by integrating hierarchical radiosity with clustering which efficiently treats global diffuse light transfer, with particle tracing for non-diffuse transfers. The integration is simple, and uses the rich light path

information in the form of hierarchical radiosity links, to guide and accelerate particle tracing. The result is rapid and complete simulation of all light paths.

For simple scenes, our algorithm allows interactive viewing (as fast as 2 frames/sec.) of a global illumination solution including both global diffuse illumination and moving caustic effects (see Fig. 1). Our algorithm can also treat more complex scenes, albeit at a higher computational cost. The difficult case of scenes lit primarily by indirect light is also handled well. Finally, it has a built-in quality control mechanism, which allows smooth transition from fast, lower-quality solutions to more expensive high quality lighting simulations by shifting computations higher or lower in our hierarchy.

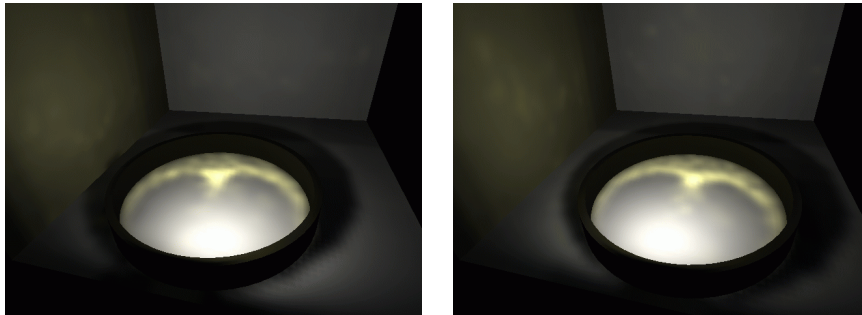


Fig. 1. Two frames from an interactive session using our new algorithm. The ring moves to the right at 2 frames/sec. (SGI R10k workstation). Note the cardioid caustic from the glossy ring.

1.1 Previous Work

In what follows we use Heckbert's [11] regular expression notation for light paths and light transport. L is used for the light, E the eye, S a non-diffuse 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 can be described by $L(D|S)^*E$.

A vast amount of research has been performed on global illumination and combined glossy/diffuse transfer. For example, many different pure Monte Carlo techniques have been proposed (e.g., see [30] for an overview), though these are currently still too slow or too noisy to achieve the level of interactivity that we require.

For diffuse-only transfer, radiosity methods (e.g., [8, 4]) can produce high-quality images. In particular hierarchical radiosity with clustering (HRC) [24, 21] can simulate radiosity for large environments rapidly. One advantage of radiosity (as any diffuse-only $L(DS)^*DE$ solution) is that the result of the simulation can be interactively visualized for walkthroughs on graphics hardware, and in some cases interactive updates can be made [6]. We will draw heavily on HRC. The important steps and structures of HRC which we will be using are reviewed in Fig. 2.

Since the advent of radiosity methods, several solutions have been proposed to add non-diffuse transfer. One approach is to store directional finite elements on patches. From the earliest such work [12] to the most recent [26] using clustering and wavelets, the storage required for directional discretisation makes these methods unusable for complex scenes which contain highly directional glossy effects such as caustics. Continuous representations (e.g., [22]) 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., [1]) have been proposed. However, these still suffer from

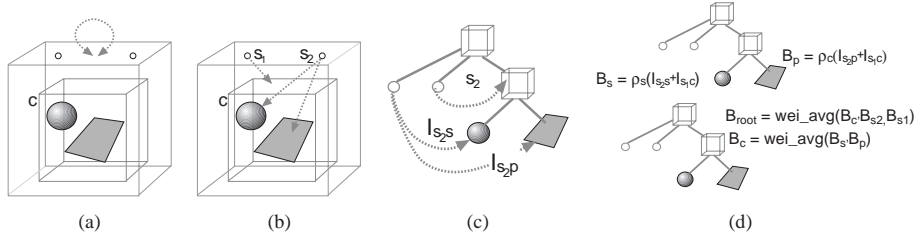


Fig. 2. Basic steps of the hierarchical radiosity algorithm: *Initialization*: the scene, consisting of a sphere and a polygon, contained in a cluster c , and two light sources. Initially there is a *self link* to the root cluster. For each iteration we perform three steps. (a) *Refinement*: a link is created between source s_1 and c , and s_2 is linked to the sphere s and the polygon p . (c) *Gather*: irradiance is gathered over the links, resulting in the values I_{s_2s} (s_2 to s) and I_{s_2p} (s_2 to p) shown in the graph representation of the hierarchy. (d) *Push-pull*: First the irradiance values are pushed to the leaves, where radiosity is set by reflecting the irradiance using reflectance ρ (top). The radiosity values are then pulled up the hierarchy by area weighted averages to maintain consistency (bottom).

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. [7] 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 recent approach is three-point clustering [27]; 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.

Extended form-factors have been introduced to take specular transfers into account, in the context of multi-pass radiosity-style algorithms. Limitations such as planar only mirrors [31], or specular-only non-diffuse transfers [23] are often imposed however.

Wann Jensen has introduced the Photon Map (e.g., [13]), which uses particle or photon tracing for all light transfers including diffuse. His method can quickly compute direct caustics (i.e., caustics caused by a light source directly), but requires a costly gather step to compute other transport paths adequately. Despite acceleration using irradiance gradients [35], the gather is still too slow for interactive use.

Density estimation [33] is another closely related approach. As with Photon Maps, all light transfer is performed with particle tracing. However viewing can be performed 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. We will adopt a similar approach for viewing.

Combined multi-pass radiosity/Monte Carlo methods have been proposed (e.g., [20, 2, 14, 19, 17]) but do not fully take advantage of the information provided by one pass to guide and thus accelerate the next. The approaches of Shirley [20] and Chen et al. [2] are precursors to our work in that they also identified the advantage of using radiosity for diffuse illumination. 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, which cannot efficiently treat complex scenes in which indirect lighting is predominant. Secondary light-source reclassification [2] becomes prohibitively expensive when many patches are secondary emitters with about the same power. In contrast, we will be using hierarchical radiosity with clustering, which deals

with these configurations efficiently. Some attempts have also been made for interactive display of globally illuminated scenes [29, 32].

Finally, it is worth mentioning that specific methods have been developed for the rendering of caustics which are one of the more beautiful lighting phenomena (e.g., [5, 36]). However, they do not treat the entire breadth of all light transfers.

2 Overview

In our new algorithm, we start by building the cluster hierarchy as in HRC. Refinement proceeds by examining each link (see Fig. 2), and deciding whether to refine based on the chosen criterion. During gather, we check each link: for links leaving diffuse objects and arriving on specular objects, diffuse to specular transfers are performed by particle tracing. The use of the link structure restricts the number of particles emitted since we only sample a limited part of space. It also substantially accelerates particle tracing itself by accurate visibility classification. Particles are then propagated into the environment both by reflection and refraction.

The iteration is completed by updating the hierarchy: first particles are placed at an appropriate level in the hierarchy, then the contribution of particles is added to the diffuse light at the leaves of the hierarchy, and all transfers to diffuse surfaces are thus recorded.

This tightly-coupled loop is particularly effective, for all $L(D|S)^*DE$ light transfer. To display, we first perform a reconstruction step generating a smooth representation of radiosity on surfaces. We can then either interactively visualize the result using graphics hardware, or ray-trace. Interactive visualization includes specular reflections such as caustics and direct shadows, as in Fig. 1. Paths or partial paths of the form $(L|D)S^+E$ can be included using ray-tracing.

The main contribution of our approach is to seamlessly integrate hierarchical radiosity with clustering (HRC) and particle tracing for the first time. The integration is clean and simple, and naturally accounts for all light paths. In contrast to previous approaches, HRC results in efficient, noise free simulation of diffuse transfer and can handle complex, indirect diffuse/specular configurations. The result of this approach is very rapid simulations of all light paths, which for simple scenes permit interactive updates, including high-frequency specular reflections such as caustics. We also treat more complex scenes at a higher, but reasonable computational cost, and handle the difficult case of scenes lit mainly by indirect light. Our algorithm also has a built-in quality control mechanism allowing smooth transition from fast, low-quality solutions, to slower but higher-quality simulations.

In the following we first present the new integrated HRC and particle tracing algorithm, we discuss reconstruction approaches followed by the presentation of our results for interactive rendering and simulations for more complex scenes. We then discuss future work and conclude.

3 Integrating HRC and Particle Tracing

As mentioned above, we will be using particle tracing to simulate glossy transfer across links arriving at glossy objects. To do this, we first introduce a modified gather step. During this step particles are emitted to simulate glossy transfer. Gathering is followed by a modified push step in which we decide where to place particles into the hierarchy. Since particles are emitted across the restricted space defined by a link, special care is needed to choose the number of particles as well as the power assigned to each.

We start our algorithm like HRC, by first constructing the cluster hierarchy using the approach presented in [3]. We then initialize the system with the root self-link [21]. To account for non-diffuse transfer, we store BSDF's (bidirectional scattering distribution function) for the non-diffuse objects [30]. We use a mixed diffuse/specular model of [16], which randomly decides whether a reflection will be glossy or diffuse, based on the percentage of diffuse/specular reflection defined by the material property. A cluster containing specular objects is considered non-diffuse.

3.1 Iterative Step

Refine/Gather and Particle Trace. We use an energy refinement criterion, BFA , where B is radiosity, F is the form-factor, A is equivalent area or *area factor* (which is area for surfaces and $4kV$ for volumes or clusters), and k is an equivalent extinction coefficient and V is volume [21]. If the quantity BFA over a link is greater than a threshold, the link is refined, incurring subdivision where necessary. Once the refinement has placed links at the appropriate level of the hierarchy, we traverse the element hierarchy. At each receiving element, we visit each link and gather irradiance from the linked elements.

If the receiving end of a link is a non-diffuse surface, we trace particles across the link to track the light that is scattered non-diffusely. The particle generation is restricted to particles which are emitted from the source end of the link and which will initially strike the receiver. Particles are emitted across the link, which is the region of space defined by the two elements. This is further described in Section 3.2.

Particles are immediately reflected at the receiving element. For clusters, we intersect with the contents of the cluster and perform the reflection with the object or polygon hit. For objects/surfaces we simply reflect via the associated bidirectional scattering distribution function (BSDF). Particles are then propagated for subsequent bounces until they are absorbed, hit a diffuse surface or exit the scene. Particle hits are stored on all objects excluding the first hit (since this transfer is accounted for by the radiosity system). Using the link hierarchy for reflection or transmission other than the first emission could be possible; the naive approach of searching the hierarchy and finding the appropriate link however has currently proven to be too expensive compared to the benefits. We discuss this further in Section 6 on future work. This process, reflection and propagation, is performed iteratively until all particles have been absorbed or have hit a diffuse surface.

Particle Placement and Push Pull. After propagation, particles are stored at the top level of the object intersected (i.e., a polygon, sphere etc.). Specular effects are often high frequency (similar to shadows for example), and thus can require fine meshing to be correctly represented. We will be using the particles to reconstruct irradiance [13, 11] arriving at a surface; our goal is to perform hierarchical reconstruction (see Section 4). As a result it is important to appropriately choose the level at which we place particles. Placing the particles at too low a level causes visible noise, while placing them too high blurs out high-frequency specular reflection details.

Intuitively, we need to identify regions where many particles are concentrated. In such regions, we want to place particles at an appropriate level in the hierarchy, to ensure that the underlying specular effects are well-represented. This is done using a criterion based on a "spread factor"; we push particles down the hierarchy when a concentration of powerful particles occurs over an area which is small compared to the equivalent area of the element being considered.

We traverse the element hierarchy, and at each element, we have n particles, each

carrying ϕ_i power and located at position \mathbf{x}_i . We define a “center” c as the power-weighted average position of the set of particles and “spread factor” SF :

$$c = \frac{\sum_n \phi_i \mathbf{x}_i}{\sum_n \phi_i}, \quad SF = \frac{\sum_n \phi_i |c - \mathbf{x}_i|^2}{\sum_n \phi_i} \quad (1)$$

If A is the area factor (see definition of Section 3.1 above) of the element, we test if: $\pi SF \leq \zeta A$ **and** $\sum_{i=1..n} \phi_i \geq \varepsilon$, where ζ determines the percentage spread we try to achieve. We typically use $\zeta = 0.5$. 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. Note however that links remain at the level defined by refinement, and thus the additional subdivision does not impact the cost of the diffuse part of the radiosity iteration.

Each element has an irradiance I_d from the diffuse gather across its links and a number, n , of particles, each with power ϕ_i . At each level of the hierarchy with equivalent area A , irradiance I becomes: $I = I_d + \frac{\sum_n \phi_i}{A}$. This can be seen as a coarse multi-level smoothing reconstruction of particle irradiance. At the leaves radiosity is then set to $B = \rho I$, in the same manner as for diffuse radiosity. This results in a constant radiosity value per leaf element. Radiosity is then pulled up the hierarchy for light transfer consistency in subsequent iterations.

Note that we can accumulate particle irradiance, by averaging over the irradiance contributed from each iteration. We have found that this typically improves the quality of specular reflection (caustics in particular), since it reduces noise.

3.2 Particle Emission

An important part of the process described above is the transmission and reflection of particles. We first explain how we estimate the non-diffuse flux across a link using particle emission. The fact that we restrict particle emission to the space defined by the link requires special treatment.

In general, the flux leaving element S is given by the following integral:

$$\Phi_S = \int_S \int_{\Omega_s} L(s, \omega) \langle \omega \cdot n_s \rangle d\omega ds, \quad (2)$$

where S is the source (emitter), Φ_S is the exiting flux of S , s is a point on S , Ω_s represents all directions leaving s , ω is a direction, n_s is the normal at s and $L(s, \omega)$ is the radiance leaving s in direction ω , and $\langle \cdot \rangle$ is a dot product.

We stochastically estimate this integral by tracing n_p particles into the environment. These particles are emitted according to a probability density function (*pdf*), $p(s, \omega)$. Following standard Monte-Carlo methodology [30, 34], the power associated with the emitted particles is given by:

$$\phi(s, \omega) = \frac{L(s, \omega) \langle \omega \cdot n_s \rangle}{n_p p(s, \omega)} \quad (3)$$

We change notation to consider the radiance $L(s, r)$ leaving point s on element S arriving at point r on the receiver element R , and perform the appropriate change of variables. Using the notation of Sillion [21], for the case of clustering the flux between two generic hierarchical elements can be written:

$$\Phi_{RS} = S_R S_S \int_S \int_R L(s, r) \frac{\tau R_r E_r}{\|s - r\|^2} dr ds \quad (4)$$

Where “receiver factor” R_r is 1 for clusters and $\cos\theta_i$ for surfaces, the “emitter factor” E_r is 1 for clusters and $\cos\theta_o$ for surfaces, S_S , S_R are scale factors to account for the volumetric case, which are 1 for surfaces and $\frac{1}{4}$ for clusters [21]; finally τ is the absorption factor for participating media. 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.

For the case of radiosity, $L(s, r) = \frac{B_S}{\pi}$, and Eq. (4) becomes $\Phi_{RS} = B_S F_{RS} A_R$ (B_S is the radiosity of S and F_{RS} is the unoccluded form-factor between R and S ; A_R is the equivalent area of R as defined before in Section 3.1 [21]). If we estimate the flux with n_{RS} particles carrying constant power ϕ_{ct} , we have $\Phi_{RS} = n_{RS} \phi_{ct}$ and thus:

$$n_{RS} = \frac{B_S A_R F_{RS}}{\phi_{ct}} \quad (5)$$

For a given link from S to R , are the quantities in Eq. (5) are known: form-factor, area-factor and source radiosity. Using Eq. (3) and (4) we define the following *pdf*:

$$p(s, r) = \frac{L(s, r) S_R S_S \tau R_r E_r}{\|s - r\|^2 \Phi_{RS}} \quad (6)$$

To use this *pdf*, we should first choose a point s_0 with a given *pdf* $p'(s)$ on the sender and then find the conditional probability $p''(r|s = s_0)$ on the receiver. In the general case this proved to be impractical. As an approximation, we sample the elements R and S uniformly $p(r) = \frac{1}{\mu(R)}$ $p(s) = \frac{1}{\mu(S)}$. The measure $\mu()$ we use is area for surfaces and volume for clusters.

We can now determine the power ϕ_{rs} we assign to each particle, given the uniform sampling we use on S and R :

$$\phi_{rs} = \frac{L(s, r) S_R S_S \mu(R) \mu(S) \tau R_r E_r}{\|s - r\|^2 n_{RS}} \quad (7)$$

Some of the particles may be absorbed or reflected diffusely if the receiver has both diffuse and non-diffuse components in its BSDF. Such transfers have already been accounted for by the diffuse (non-particle) transfer across links, and thus we disregard them. We only propagate and store particles which are reflected non-diffusely.

3.3 Accelerated computation

The refinement process provides information about visibility which we use to accelerate particle emission. By using an appropriate object representation for non-planar surfaces, we also achieve significant savings and better quality. Finally, when modifying a scene dynamically, we limit the amount of recomputation. We discuss all of these improvements next.

Visibility classification. We use shaft culling [9] for visibility classification in our approach. For a given link, we determine the list of potential blockers across the link. Bounding boxes are opened at the first level if they overlap the shaft, bounding boxes outside the shaft are removed from the list and bounding boxes entirely contained in the shaft are never opened.

Total visibility, i.e., a completely unblocked transfer between two elements is determined exactly and efficiently if the blocker list is empty. Blocker lists are maintained during recursive refinement for additional acceleration. When transferring particles across a totally visible link, no visibility tests or ray casts need be performed saving much unnecessary computation. If the blocker list is not empty then the visibility factor between elements is computed by ray-tracing as in traditional HRC (e.g., [10]).

Object representation. In order to handle objects such as spheres, cylinders and cones, we use an object representation in our HRC system. This is similar in spirit to [28, 18], in that for the purposes of the radiosity solution, the objects are not explicitly subdivided beforehand, lowering the number of primitives.

All intersection calculations are performed using the untesselated objects which results in significant times savings and improves the quality of the shadows and caustics.

Motion update acceleration. When we move an object (which can also be a source), we maintain the subdivided hierarchy. Once the motion is complete, we traverse the hierarchy, recomputing visibility for all links. We then re-shoot particles across links (we do not accumulate particle irradiances over iterations in this case). The “unoccluded form-factor” estimations are recomputed only for links leaving or reaching the moving object. To achieve better update rates, we limit the number of rays used to compute visibility between elements.

4 Reconstruction and Rendering

As described previously, the push step reconstructs the irradiance due to particles at the different levels of the hierarchy, smoothing the results in the process. These values are thus pushed down the element hierarchy, resulting in a constant radiosity value per leaf patch. To display a smooth solution a supplementary reconstruction step is required. Once this step is complete, we can either render directly using the graphics hardware (but missing the $(L|D)S^+E$ paths) or use ray-tracing which captures all paths.

In traditional radiosity systems, extrapolation is performed to the vertices of the patches, requiring the maintenance of a restricted tree during subdivision. As a first preliminary solution, we reconstruct on a “grid”, which is separate from the element hierarchy. The grid is represented as a collection of *display polygons*. For a quadrilateral for example, we create a regular grid of display polygons, each having the size of the smallest subdivided element. For other objects (e.g., spheres or cylinders), specific parameterizations are used.

Our current solution starts by assigning the constant radiosity value of the corresponding patch to each display polygon. Smoothing is then performed by assigning the average radiosity value of the neighboring display polygons to each vertex. We can iterate to perform higher-order smoothing, by first assigning the average radiosity of all vertices to each polygon, and again assigning the average of the patch neighbor radiosities to each vertex. Evidently, a more involved reconstruction scheme (e.g., [15]) could be adapted to our needs (see also Section 6).

Once the reconstruction has been performed, we can directly render the polygons of the grid, or a texture generated directly from these polygons. The only missing paths are $(L|D)S^+E$, that is paths leaving the eye, hitting at least one specular surface and then hitting either a diffuse surface or a light source.

To render these, we use ray-tracing. We first identify pixels which correspond to specular surfaces, and then perform ray-tracing only for these pixels (Monte-Carlo or distributed ray-tracing could also be used). When a ray hits a diffuse surface, the

smoothed radiosity value is used directly. To optimize this we actually render the diffuse component in hardware, and then add in the ray-traced $(L|D)S^+E$ contributions. To find the corresponding grid element efficiently, we associate them to the hierarchical radiosity elements. The value returned is interpolated appropriately in the interior of the display polygon.

5 Results

All results are on an R10000 (200Mhz) SGI Origin 2000.

Interactive Rendering for Simple Scenes. We have already shown the results of interactive rendering in Fig. 1. Note that only the diffuse component of each object is being shown, using hardware rendering, explaining the dull appearance of the ring. Fig. 5 shows other examples (Fig. 5-8 are in the colour section). Quicktime movies of interactive sessions can be found at:

<http://www-imagis.imag.fr/Publications/xgranier/EGWR00>.

Quality Control. The iterations are governed by the following three parameters: (i) the threshold ϵ for the *BFA* refinement criterion. This controls the level of the links in the hierarchy, and thus the accuracy of the diffuse illumination, as in HRC; (ii) the minimum area, which is the size of the minimal element of the element hierarchy. This influences the subdivision due to diffuse light as well as the additional subdivision due to particle push; (iii) the ϕ_{ct} parameter, which controls the number of particles used as discussed above (Section 3.2). The lower ϕ_{ct} , the larger the number of particles sent over a link (see Eq. (5)).

In Fig. 3, we show the effect of varying the ϕ_{ct} parameter. Note that the quality quickly improves without significant overhead in time. The rightmost image is shown as an example of the quality our system can provide.

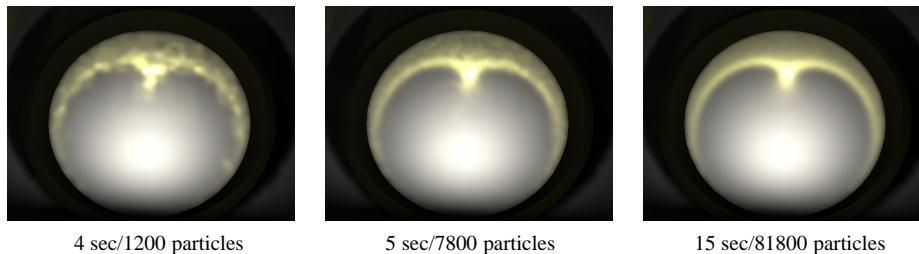


Fig. 3. Effect of varying the ϕ_{ct} parameter. We show the number of particles used to generate each image and the time required for two iterations. The minimum area parameter is fixed. Images rendered in hardware.

Complex and indirectly lit scenes. In Fig. 4 we show a modified version of the conference room scene from Greg Ward-Larson's material geometry format (mgf) test scenes. The model is quite complex (19,000 primitives corresponding to 150,000 polygons if tessellated naively). Due to memory limitations we have computed a coarse solution, but some caustic effects are perceptible.

In Fig. 6, we show a modified version of the Soda Shoppe scene (also an mgf test scene). We have reduced the size of the light sources and added a number of blue glass spheres to clearly demonstrate caustic effects. The scene contains about 1779 primitives (cylinders, spheres, rings and polygons); when initially tessellated to depth five, the input scene contains 80,000 polygons. Note that in a moderate amount of time (17 min 22 sec), we have a number of basic caustic effects. Images of Fig. 6 and Fig. 7

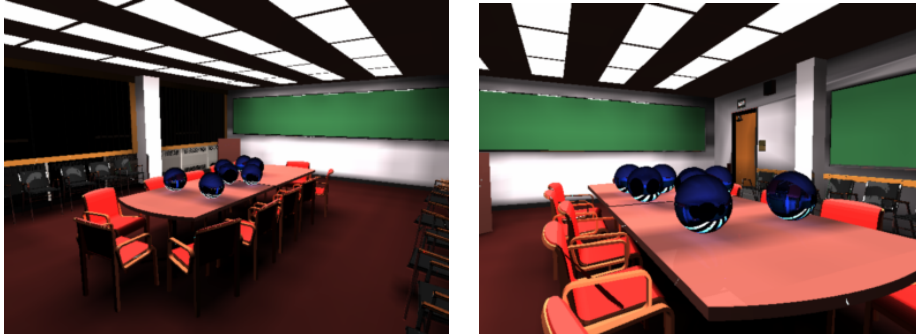


Fig. 4. A coarse solution of a modified version of the conference room scene, light simulation took 4 min 27 sec. (left) General view (right) closer view of the table. Ray-tracing took \sim 4 min.

were ray-traced at 1980 x 980 resolution in about 4 minutes. Our final image ray-tracer uses the bounding box hierarchy of clusters, and is particularly slow. In both the Soda Shoppe and the conference room, many surfaces (floor, tabletops etc.) have materials with specular components, and thus contribute to non-diffuse transfer.

In Fig. 7(left), we show the same scene, but we have replaced the large direct lights with 12 small polygonal lights pointing to the walls. In a progressive refinement context, this would be very hard to treat, even using secondary source reclassification, since a very large number of patches would be characterized as important secondary sources, resulting in the need for a large number of iterations. Note that the simulation time (9 min 2 sec) is in the same order of magnitude as that for direct lighting.

Informal comparison to Particle Tracing. As an informal comparison we have computed the indirect lighting Soda Shoppe scene in Fig. 7(right) using standard particle tracing [34]. We use the same reconstruction as in our method for fairness of comparison. We have placed our solution computed in about the same time side-by-side. As we can see, the main problem with the particle tracing solution is noise.

Interactive Viewing and the Render Cache. We have integrated our method with the Render Cache [32] for interactive display. This approach allows interactive visualization of ray tracing algorithms. Our algorithm is particularly well suited to this approach, since for diffuse surfaces we render the radiosity directly. An example of its use is shown in Fig. 8 and in the Quicktime movies on the web site.

6 Conclusions

We have presented a novel global illumination algorithm which computes both glossy and diffuse lighting in a seamless, integrated manner.

We have achieved this by integrating particle tracing for glossy transport into the hierarchical radiosity gather step. The radiosity system efficiently simulates all diffuse light interactions. In addition, the link hierarchy it builds encodes considerable structural information about the flow of light in the environment. We exploit this information to guide and accelerate the particle emission that simulates the non-diffuse light interactions. This allows us to simulate all types of light path chains including specular reflection (e.g., caustics) without the high cost and noisy results of using particle tracing for all interactions.

Our integrated approach efficiently treats the case of mainly indirect lighting, including specular reflections. It also permits a smooth transition between coarse but

fast solutions and more expensive but higher quality ones, without sacrificing the completeness of our light simulation. A suitable reconstruction step is performed which allows interactive visualisation of diffuse light with specular reflections including complex caustics, and interactive dynamic updates for simple scenes. More complex scenes can also be treated at reasonable cost. For higher quality images, ray tracing can be used to correctly visualize non-diffuse surfaces, and this can even be done interactively for low resolutions using the Render Cache.

Future Work. One limitation of our method is that if high frequencies exist locally in models with large differences of scale, it is hard to avoid explosion in the number of mesh elements. The solution to this will inevitably lead to a separate reconstruction process for low-frequency (mainly diffuse) lighting interactions, and high-frequency (mainly specular) phenomena such as caustics. We are developing a multi-level reconstruction algorithm for the low frequency interactions, while high-frequency effects can be treated by some form of “caustic maps” or textures. For diffuse light transfer, the HRC hierarchy can be used throughout (by averaging the map contents), and thus the caustic maps will be only used for display.

We have performed initial experiments using visual importance [25] to refine the scene locally. However, we need to further develop the concepts, in particular for more complex scenes and the particle tracing part of the algorithm.

We are also investigating better dynamic updates using techniques similar to those of [6]. As with importance, the particle tracing phase needs to be adapted so that work is localized correctly. We are confident that we will be able to achieve interactive object motion within even moderately complex scenes such as those shown here.

Currently, link information is used only for the first segment of a diffuse-specular light path. Use of this information for subsequent bounces is clearly worth investigating.

Acknowledgments

This work was supported in great part by the ESPRIT Open LTR project #35772 SIMULGEN 2. Thanks to P. Poulin for numerous discussions and lots of advice, and to F. Neyret and F. Durand for re-reading the paper.

References

- [1] L. Aupperle and P. Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Proc. SIGGRAPH'93*, Annual Conference Series, pages 155–162, 1993.
- [2] S. E. Chen, Holly E. Rushmeier, G. Miller, and D. Turner. A progressive multi-pass method for global illumination. In *Computer Graphics (SIGGRAPH '91)*, volume 25, pages 165–174, July 1991.
- [3] P. H. Christensen, D. Lischinski, E. J. Stollnitz, and D. H. Salesin. Clustering for glossy global illumination. *ACM Trans. on Graphics*, 16(1):3–33, January 1997.
- [4] M. F. Cohen, S. E. Chen, J. R. Wallace, and D. P. Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics (SIGGRAPH 88)*, volume 22(4), pages 75–84, August 1988.
- [5] S. Collins. Adaptive splatting for specular to diffuse light transport. *5th EG Workshop on Rendering*, pages 119–135, June 1994. Darmstadt, Germany.
- [6] G. Drettakis and F. Sillion. Interactive update of global illumination using a line-space hierarchy. In *Proc. SIGGRAPH'97*, Annual Conference Series, pages 57–64, August 1997.
- [7] R. Dumont, K. Bouatouch, and P. Gosselin. A progressive algorithm for three point transport. *Computer Graphics Forum*, 18(1):41–56, March 1999.
- [8] C. M. Goral, K. K. Torrance, D. P. Greenberg, and B. Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics (SIGGRAPH'84)*, volume 18(3), pages 213–222, July 1984.
- [9] E. A. Haines and J. R. Wallace. Shaft culling for efficient ray-traced radiosity. *2nd EG Workshop on Rendering (Photorealistic Rendering in Computer Graphics)*, 1994.

- [10] P. Hanrahan, D. Salzman, and L. Aupperle. A hierarchical radiosity algorithm. In *Computer Graphics (SIGGRAPH '91)*, volume 25, pages 197–206, July 1991.
- [11] P. S. Heckbert. Adaptive radiosity textures for bidirectional ray tracing. In *Computer Graphics (SIGGRAPH'90)*, volume 24, pages 145–154, August 1990.
- [12] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A radiosity method for non-diffuse environments. *Computer Graphics (SIGGRAPH'86)*, 20(4):133–142, August 1986.
- [13] H. Wann Jensen. Global illumination using photon maps. In *7th EG Rendering Workshop, "Rendering Techniques '96"*, pages 21–30. EG, Springer Wien, June 1996.
- [14] A. Keller. Instant radiosity. In *Proc. SIGGRAPH'97*, Annual Conference Series, pages 49–56, August 1997.
- [15] L. Kobbelt, M. Stamminger, and H-P. Seidel. Using subdivision on hierarchical data to reconstruct radiosity distribution. *Computer Graphics Forum*, 16(3):347–356, August 1997. Proc. of EG '97.
- [16] E. P. Lafortune and Y. D. Willems. Using the modified phong reflectance model for physically based rendering. Technical Report CW 197, Dept. of Computing Science, K.U. Leuven, November 1994.
- [17] L. Neumann. Monte Carlo Radiosity. *Computing*, 55(1):23–42, 1995.
- [18] S. Schäfer. Hierarchical radiosity on curved surfaces. In *EG Workshop on Rendering 1998, "Rendering Techniques '98"*, pages 187–192. EG, Springer Wien, June 1997.
- [19] P. Shirley. Radiosity via Ray Tracing. In *Graphics Gems II*, pages 306–310. Academic Press Professional, Boston, MA, 1991.
- [20] P. Shirley. A ray tracing method for illumination calculation in diffuse-specular scenes. *Graphics Interface '90*, pages 205–212, May 1990.
- [21] F. X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Trans. on Visualization and Computer Graphics*, 1(3):240–254, September 1995.
- [22] F. X. Sillion, J. R. Arvo, S. H. Westin, and D. P. Greenberg. A global illumination solution for general reflectance distributions. In *Computer Graphics (SIGGRAPH'91)*, volume 25(4), pages 187–196, July 1991.
- [23] F. X. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *Computer Graphics (SIGGRAPH'89)*, volume 23, pages 335–344, July 1989.
- [24] B. Smits, J. Arvo, and D. P. Greenberg. A clustering algorithm for radiosity in complex environments. In *Proc. SIGGRAPH'94*, Annual Conference Series, pages 435–442, July 1994.
- [25] B. E. Smits, J. R. Arvo, and D. H. Salesin. An importance-driven radiosity algorithm. In *Computer Graphics (SIGGRAPH'92)*, volume 26, pages 273–282, July 1992.
- [26] M. Stamminger, Annette Scheel, X. Granier, F. Perez-Cazorla, G. Drettakis, and F. X. Sillion. Efficient glossy global illumination with interactive viewing. *Graphics Interface '99*, pages 50–57, June 1999.
- [27] M. Stamminger, P. Slusallek, and H-P. Seidel. Three point clustering for radiance computations. In *9th EG Workshop on Rendering, "Rendering Techniques '98"*, pages 211–222. Springer Wien, 1998.
- [28] M. Stamminger, P. Slusallek, and H-P. Seidel. Bounded radiosity - illumination on general surfaces and clusters. *Computer Graphics Forum*, 16(3):309–318, August 1997.
- [29] W. Stürzlinger and R. Bastos. Interactive rendering of globally illuminated glossy scenes. In *8th EG Workshop on Rendering, "Rendering Techniques '97"*, pages 93–102. Springer Wien, June 1997.
- [30] E. Veach. *Robust Monte-Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. http://graphics.stanford.EDU/papers/veach_thesis/.
- [31] J. R. Wallace, M. F. Cohen, and D. P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. In *Computer Graphics (SIGGRAPH '87)*, volume 21, pages 311–320, July 1987.
- [32] B. Walter, G. Drettakis, and S. Parker. Interactive rendering using the render cache. In *10th EG Workshop on Rendering, "Rendering Techniques '99"*. Springer Wien, June 1999. Granada, Spain.
- [33] B. Walter, P. M. Hubbard, P. Shirley, and D. P. Greenberg. Global illumination using local linear density estimation. *ACM Trans. on Graphics*, 16(3):217–259, July 1997.
- [34] B. J. Walter. *Density estimation techniques for global illumination*. PhD thesis, Cornell University, 1998. <http://www.graphics.cornell.edu/pubs/1998/Wal98.html>.
- [35] G. J. Ward and P. Heckbert. Irradiance gradients. *3rd EG Workshop on Rendering*, pages 85–98, May 1992.
- [36] M. Watt. Light-water interaction using backward beam tracing. In *Computer Graphics (SIGGRAPH 90)*, volume 24(4), pages 377–385, August 1990.

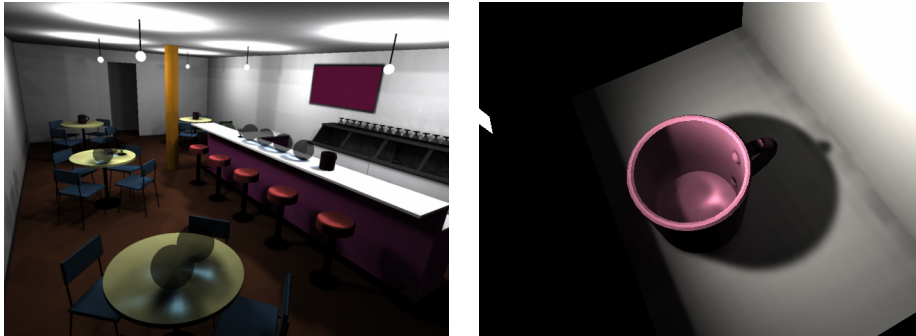


Fig. 5. Interactive walkthroughs using hardware rendering (simple transparency; no refraction). (Left) Soda Shoppe (right) scene with mug; the light source can be moved ~ 2 sec/frame.

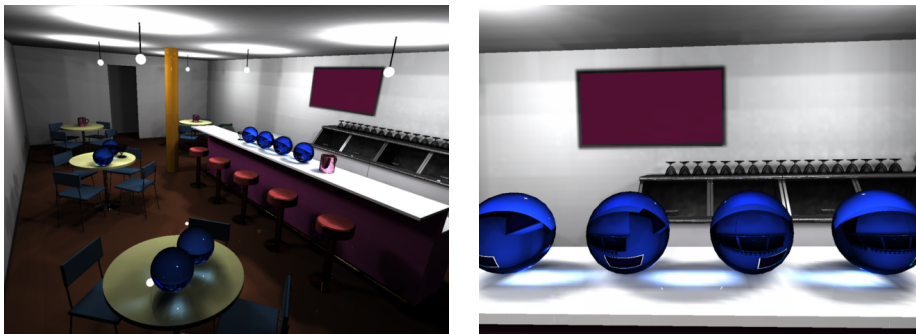


Fig. 6. (left) A general view and (Right) a closeup of the bar. Light simulation took 17 m 22 s.

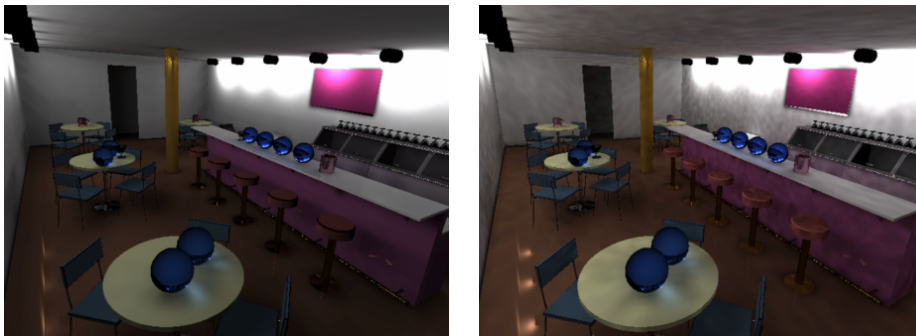


Fig. 7. Soda Shoppe scene with all-indirect light (lights point towards the walls): (Left) computed with our method in 9 min 2 sec; (Right) Particle tracing solution computed in 8 min 2 sec.



Fig. 8. Two frames from a Render Cache interactive session running at 2 Hz, using the output of our system. We are turning left; some artifacts remain where samples are not yet available.