

# Efficient Glossy Global Illumination with Interactive Viewing

Marc Stamminger<sup>1</sup>, Annette Scheel<sup>1</sup>, Xavier Granier<sup>2</sup>,  
Frederic Perez-Cazorla<sup>3</sup>, George Drettakis<sup>2</sup>, François Sillion<sup>2</sup>

<sup>1</sup> Max-Planck-Institute for Computer Science, Saarbrücken, Germany

<sup>2</sup> iMAGIS-GRAVIR/IMAG-INRIA, Grenoble, France

<sup>3</sup> GGG, University of Girona, Spain

## Abstract

The ability to perform interactive walkthroughs of global illumination solutions including glossy effects is a challenging open problem. In this paper we overcome certain limitations of previous approaches. We first introduce a novel, memory- and compute-efficient representation of incoming illumination, in the context of a hierarchical radiance clustering algorithm. We then represent outgoing radiance with an adaptive hierarchical basis, in a manner suitable for interactive display. Using appropriate refinement and display strategies, we achieve walkthroughs of glossy solutions at interactive rates for non-trivial scenes. In addition, our implementation has been developed to be portable and easily adaptable as an extension to existing, diffuse-only, hierarchical radiosity systems. We present results of the implementation of glossy global illumination in two independent global illumination systems.

*Key words:* global illumination, glossy reflection, interactive viewing

## 1 Introduction

Real-world scenes contain materials with different reflective properties, varying from matte (diffuse) to shiny (glossy or specular). Global illumination research has made great advances for the treatment of diffuse environments in the recent years, in particular with the advent of the Hierarchical Radiosity (HR) algorithm [7] and the subsequent introduction of clustering [18, 15]. It is now possible to compute global illumination solutions of complex diffuse environments and perform interactive walkthroughs. Interactivity is achieved using the polygonal model which is appropriately subdivided into sub-polygons to capture shadows and lighting variations. Since the environments are diffuse, no updates are necessary at each frame, and the polygons are drawn as is. In contrast, scenes containing glossy surfaces cannot yet be treated in an interactive context. To generate images with glossy surfaces, ray-tracing based approaches are typically used, such as the RADIANCE system [27] or path-

tracing algorithms (e.g., [11, 23]). Some finite element approaches have been presented, but can only treat trivial scenes (e.g., [13, 1]) or require a second, ray-casting pass to generate an image [3]. Two approaches have been proposed which are capable of interactive viewing [16, 25], but they are limited in their capacity to treat non-trivial environments and reflective behaviours.

We present a novel solution which allows interactive viewing of globally illuminated glossy scenes. To reach this goal, we use a finite element representation of outgoing radiance at surfaces or clusters. This representation is used at each frame to evaluate the radiance leaving a glossy surface and reaching the eye, permitting interactive viewing.

A novel representation of incoming radiance in the form of a structure called *Illumination Samples* is presented, which is efficient both in memory and computation time. This structure replaces an explicit (and costly) finite-element representation of incoming radiance by sets of relevant point samples.

Furthermore, we demonstrate the importance and benefits of using an adaptive hierarchical representation of outgoing radiance improving on both computation time and memory consumption compared to previous approaches such as [16]. Our algorithm produces high quality glossy global illumination solutions which can be directly rendered for interactive walkthroughs, without the need for expensive second-pass final gather as in [3].

## 2 Previous Work

Most previous work in glossy illumination has been centered around ray-tracing. These start with distributed ray-tracing [28, 5] and the rendering equation [9] in the late eighties. A large body of research ensued focusing on Monte-Carlo stochastic algorithms. The goal of this research was to reduce the noise in the solutions, introduced by the stochastic nature of the Monte-Carlo methods (e.g. [22, 11, 14]). Monte-Carlo algorithms that proved to be useful in other research areas were successfully trans-

ferred to the global illumination problem [10, 24].

In parallel, several multi-pass methods have been developed which combine the advantages of ray-tracing and radiosity-style calculations [17]; others have integrated radiosity calculations a stochastic process [2]. The RADIANCE system [27], particle tracing [12] and photon-maps [8] are also interesting since they collect samples of illumination either on surfaces or in a separate structure, and use a ray-cast or trace to render the final image.

“Pure” finite element approaches for glossy illumination have appeared in two main flavours: three-point approaches [1, 13] and finite-element approaches using directional distributions [16, 3]. We concentrate on the last two methods in more detail, since they are closer to our new algorithm.

## 2.1 Wavelets and Final Gather

Christensen et al. [3] extended the wavelet methods which have been used for radiosity [29, 6, 4] to a radiance clustering algorithm. However, it still suffers from some computationally expensive steps which hinder interactive viewing. Patches store a radiance distribution which is represented using a four dimensional wavelet basis accounting for spatial and directional variations. Computing the transport coefficients involves evaluating a six-dimensional integral which is computationally expensive. Importance driven refinement reduces the number of interactions drastically; but the solution becomes view dependent and consequently cannot be used for interactive viewing.

Higher order wavelet bases were also investigated, which provide a sparser transport coefficient matrix and deliver smoother representations. However, the integrations are so complex that the authors resorted to the Haar basis with a smoothing final gather step, which is very time consuming and view dependent.

## 2.2 Radiance Clustering

The Radiance Clustering approach (RC) developed by Sillion et al. [16], used spherical harmonics to store exiting radiant intensity  $I$  on the hierarchical elements of a subdivision of the original scene. An “immediate-push” algorithm is used, which, during the gather operation of light across links, “pushes” the contribution all the way to the leaves. At the leaves, radiant intensity  $I$  is stored as a spherical harmonic function; the new contribution is reflected and added into this function.

The result can be visualised directly by sampling the spherical harmonic representations of  $I$  at each frame. Direct visualisation (i.e. with no acceleration) was performed for simple scenes; since for each frame radiance is evaluated at each vertex or leaf element, frame rates are not optimal. Furthermore, spherical harmonics are a

non-hierarchical representation, and the number of coefficients used is fixed in advance. As a result, there is no control over the level of detail required to represent the directionally dependent glossy illumination.

Our new algorithm provides solutions to the above problems and also reduces memory and time consumption. We start with an improved representation of incoming radiance, which avoids the memory overhead and multiple hierarchy passes of the “immediate-push” solution. In particular we introduce *Illumination Samples* which are an appropriate point sample set representation of incoming light. We then proceed with an adaptive hierarchical representation of outgoing radiance using Haar wavelets, which is well-suited to interactive viewing, and allows smooth control of the memory/quality tradeoff. This avoids the problems of non-adaptive representations which are either not sufficiently accurate or too memory-consuming. Appropriate directional refinement and simple heuristics for accelerated viewing are also introduced.

## 3 The Illumination Samples Algorithm

The goal of the new Illumination Samples algorithm is to extend an existing Hierarchical Clustering algorithm to also handle non-diffuse surfaces. Inter-surface light propagation is the same for diffuse and non-diffuse environments, with the difference that in a non-diffuse setup directional information about incident light must be maintained for a following glossy reflection step.

As in [16], patches and clusters are assumed to have no spatial extent. They store a hierarchical directional distribution for outgoing radiance which will be described separately in Section 4. In contrast to [3], our new algorithm does not differentiate between clusters and patches concerning the representation of exitant light.

### 3.1 Bounded Propagation

Our approach is based on the radiosity clustering method described in [20, 21], which can handle flat and curved surfaces as well as clusters in a uniform manner. Bounding boxes around the objects are used to bound the set of interacting directions. With this information, bounds on the form factor and exitant radiance at the sender are computed, delivering minimum and maximum values for the received radiosity. The difference is used to decide whether to refine a link.

This bounded radiosity approach can be applied to radiance computations easily, since the propagation of light, i.e. the transformation of exitant to incident light, is independent of material properties. The only difference lies in the evaluation of bounds on the radiance of the sender, which is even easier if we have a directional distribution for the sender’s exitant radiance. However, since this exitant radiance representation is only approxi-

mate, the resulting bounds are no longer conservative.

### 3.2 Incident Light

One way to integrate the directional information is to explicitly compute a finite element representation of it. In [3], each incident light contribution computed during propagation is projected separately onto a basis for incoming light (for clusters). This is rather costly (in memory and time) and results in significant blurring of incident light, which can exhibit very strong variations.

An alternative is to reflect incident light contributions immediately after they have been computed [16], while their direction of incidence is still known. The reflection responses are then projected in finite element bases separately. This method circumvents the need to store the incident light, but the storage consumption is not reduced: two representations of exitant radiance are needed for the push/pull phase. In addition, this method is computationally expensive, because of the high number of BRDF evaluations, and the multiple hierarchy traversals involved in the immediate projection.

Our proposed solution is to combine the approaches of incident light representation and immediate reflection. We attach incident light to a receiving patch in the form of *Illumination Samples*. Light propagation is computed similarly to HR by refining links until each link represents what amounts to constant light power. Instead of simply summing the irradiance values at the receiver, an Illumination Sample with the direction to the sender and the transported irradiance is added to the receiver for each link. At the end of the propagation step, the illumination in the scene is represented as a set of point samples, distributed over the scene hierarchy.

### 3.3 Push/Pull and Reflection

A push step as in HR is needed to create a consistent representation of the incident light at the leaves, i.e. all light received by inner nodes is propagated to the children by passing its Illumination Samples downwards. Afterwards, each leaf has a large set of Illumination Samples describing its entire incident light field. Note that the number of hierarchy traversals is much smaller than in Radiance Clustering [16], where each sample is pushed down separately.

After the push step, the incident light has to be reflected according to the object's BRDF. Because Illumination Samples correspond to Dirac impulses, the reflection is an impulse response of the BRDF, i.e. it is the BRDF with a fixed incident light direction multiplied by the irradiance of the sample. The complete reflection is the sum of the impulse responses to each Illumination Sample. Therefore the BRDF must be evaluated once for each Illumination Sample to obtain the reflected radiance

in a particular direction.

Using an adaptive directional distribution described below, reflected light is projected onto an adaptive, hierarchical directional basis to obtain the new exitant light for each patch. These representations are then averaged bottom-up to obtain the distributions for inner nodes.

### 3.4 Shooting

With a gathering iteration scheme, the number of Illumination Samples and thus the time for push/pull increases from iteration to iteration. With a shooting scheme in the spirit of [19] this can be avoided. This requires an additional directional representation of *unshot radiance*, but it also avoids the storage and reflection of all Illumination Samples.

### 3.5 Error Analysis

Note that in our approach propagation and reflection are completely decoupled. Propagation computation does *not* consider the reflection properties of the receiver, which would allow the computation of incident light at a highly glossy patch more accurately than in the diffuse case. The spatial refinement of the patches is done during propagation, while the refinement level of the directional distributions is chosen during reflection. This distinction does not impose a problem on convergence, but it results in memory/computation savings.

Illumination Samples can be interpreted as Dirac-peaks from a particular direction describing incident light and are thus somewhat similar to the photons in the Photon Map approach of Jensen et al. [8]. However, Photon maps are not deterministic and their usage for lighting simulation is very different from ours.

With respect to a standard norm, with Dirac-peaks no convergent representation can be obtained. On the other hand, the Dirac-representation is only used to compute the reflection integral. From another point of view, this representation can be seen as intermediate data in a delayed numerical integration, where each Illumination Sample is a temporarily stored integration sample. So as long as the BRDF is numerically integrable, the computed reflection will converge.

## 4 Adaptive Representation of Outgoing Radiance for Interactive Display

To produce the finite-element solutions suitable for interactive display, we store outgoing light in the form of directional distributions attached to surfaces or clusters. As in Radiance Clustering [16], objects are assumed to have no spatial extent. Instead of the four dimensional radiance only the 2D *radiant intensity* distribution is stored with each object.

## 4.1 Directional Representations

For the representation of directional radiant intensities, we have implemented and examined two options: First, a uniform subdivision of the direction space, where each distribution is represented by a fixed number of coefficients (*non-adaptive basis*)<sup>1</sup>. Second, we implemented an adaptive representation using Haar Wavelets.

The non-adaptive basis is more useful for smooth distributions, because all operations on the fixed subdivision basis are simple and fast. The adaptive Haar basis is better suited for strongly varying functions, because it can use more basis functions in the interesting regions and fewer in smooth regions. However, operations such as the evaluation of the distribution or addition of two distributions are more expensive.

### “Non-adaptive” Representation

For a non-adaptive basis, we use a uniform subdivision of the direction space. To accomplish this task, a tetrahedron is subdivided. We thus obtain  $4^{n+1}$  triangles if the level of subdivision is  $n$ . Since the number of vertices is lower than the number of triangles (this is  $2 * 4^n + 2$ ), we decided to store 3 floats for RGB only at the vertices.

### Haar Representation

For the Haar representation, the domain of directions is parameterized by points on an octahedron. The vertices of the octahedron are selected to lie on the main axes, so each face corresponds to one octant of the directional domain. Simple sign considerations of a direction deliver the corresponding octahedron face.

A hierarchy of basis functions is built by assigning a first level basis function to each of the eight faces of the octahedron. These are then subdivided hierarchically in the usual manner.

In order to quickly compute an adaptive representation, a top-down approach was chosen. Assume that the function to be projected is  $f$ . For each of the first eight basis functions,  $f$  is sampled at the triangle corners and at its center. If  $f$  is almost constant over the triangle, the sample values will only vary slightly. For highly varying  $f$ , one can expect a wide range of function samples. Thus the difference between minimum and maximum sample is considered. If it is too large, the four finer basis functions partitioning the domain are considered recursively.

This top-down approach runs into problems if  $f$  has a sharp peak inbetween the samples. We alleviate this problem by enforcing a minimum subdivision level in the hope that the resulting sampling is dense enough to not miss any peaks.

<sup>1</sup>This allows a comparison with [16]; to be complete we should have tested with spherical harmonics. Since no solution for general surface orientation currently exists, we used the “non-adaptive” basis.

There are several possibilities to decide whether the difference is too large. For the algorithm described in this paper, the difference is compared with the midpoint value, i.e. a maximum *percentage* deviation  $\epsilon$  with respect to the center value is allowed. This turned out to be beneficial for our case (see Section 5.1); for other settings, different criteria can be used.

### Comparison

To see the differences involved in using Haar or non-adaptive representations, consider the following example, which is an empty room (the geometry is taken from the RADIANCE test scene “Soda Shoppe” [26]).

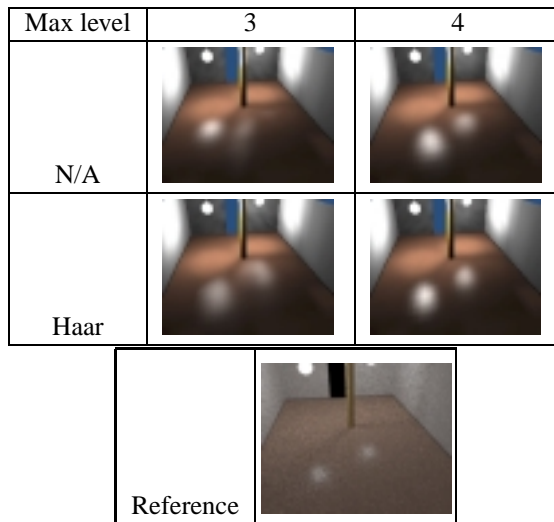


Figure 1: Comparison of the non-adaptive (N/A) vs. Haar.

Max Level	Distr	Triangles		Time	
		N/A	Haar	N/A	Haar
3	2878	782K	640K	510 s	722 s
4	2878	3127K	1829K	731 s	1125 s

Table 1: Comparison of Non-adaptive (N/A) and Haar basis for Radiance Clustering, showing the number of directional functions used and the computation time. Max Level is the maximum level of subdivision.

The reference image was computed using a path-tracer using next-event estimation [11]. The images in Figure 1 were generated using Radiance Clustering, with the non-adaptive and Haar basis (see Section 5 for more details on rendering). The “Max Level” parameter corresponds to the maximum permitted level of subdivision. Clearly, the non-adaptive basis fails to correctly represent the high-lights on the glossy floor for maximum subdivision level 3. For maximum level 4, the result is improved, but at the cost of 4 times more memory (see Table 1). In contrast,

the Haar basis uses less than three times as much memory for an “equivalent” improvement in quality. However, the Haar basis also takes more time. The reason is that arithmetic operations on the regular constant subdivision are of course simpler and faster.

This example demonstrates that for highly glossy scenes, small highlights can only be captured with the adaptive basis or a very fine non-adaptive basis representation, which in turn requires large amounts of memory. More importantly, the user, who has limited memory, can only change the quality in large “quanta”, and often will not be able to get a satisfactory result before running out of memory. Adaptive bases, such as Haar, alleviate this problem. However, the uniformity of the non-adaptive basis results in a smoother, more regular distribution, which becomes especially visible during interactive viewing.

## 5 Interactive Display

After the computation of a global illumination solution using Illumination Samples, we have a representation of outgoing radiant intensity, stored in the directional distribution function. At each frame during interactive display, we need to evaluate radiance for every glossy hierarchical leaf element in the direction of the viewpoint. This implies two requirements: (i) subdivision of the directional distributions appropriately so that a visually pleasing representation of glossy effects is produced and (ii) acceleration of the display process to avoid the cost of the evaluation of radiance at each element at every frame.

### 5.1 Refinement Issues for Display

Recall that we have decoupled directional subdivision, in the form of the Haar-based directional distribution functions, and the spatial subdivision, in the form of the “traditional” hierarchical radiosity element hierarchy. To display the solution, we interpolate radiance in the view direction by evaluating the directional distribution on each element. If subdivision in direction space is performed arbitrarily, the difference in subdivision of the directional function between neighbouring patches may be too abrupt.

This is the case for example if we compare absolute value differences between the center and the vertices of the triangles of the directional subdivision to decide whether to subdivide. The use of relative (percentage) differences avoids this problem since we approximate the form of the function, which varies more slowly across neighbours. The artifacts due to the absolute refinement can be seen in Figure 2.

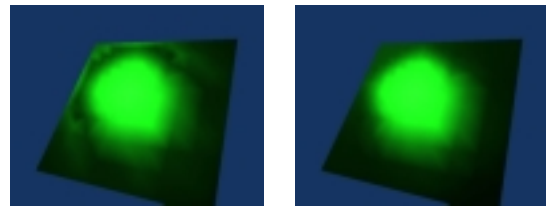


Figure 2: Artifacts when using the “absolute” refiner (left), which are absent when using the “relative” refiner (right).

### 5.2 Interactive Rendering

For efficient display we separate objects into two lists, so that diffuse objects can be rendered once and redisplayed in efficient, display-list mode. The other list, of glossy reflectors, is updated appropriately at each frame and displayed in immediate mode. The acceleration achieved obviously depends on the percentage of diffuse surfaces in the scene. For the scenes tested we achieve update rates varying from a few frames per second to a few seconds per frame for more complex scenes.

To achieve smooth shading for glossy surfaces, we add a field to the data structure associated with vertices in the hierarchy of elements. For planar surfaces, this field is updated during push-pull in a manner slightly different to that of radiosity; i.e. for a vertex belonging to a leaf element or to an edge, the radiant intensity is summed with the radiant intensity stored at the vertex. Since radiant intensity is in Watts/sr (see [16]), at display time we divide by the area of the surrounding elements.

The special case of indexed face-sets is treated separately. Indexed face-sets are common modelling primitives, and often result from the tessellation of curved objects such as spheres or cylinders. The advantage of such a primitive is that vertices are shared between adjacent elements. We can thus avoid the storage of the additional directional distribution at the vertices.

Each vertex stores the list of polygonal elements which share it. Its color is then the average radiant intensity of these polygons (i.e.  $I$  evaluated at the centers of the elements in the viewing direction). For more efficient display, we evaluate this color once per vertex for a given direction. Also, we recompute the color only if the direction changes “sufficiently” i.e. greater than a user-defined  $\epsilon$  threshold. This allows the control of the quality/update rate tradeoff.

## 6 Implementation and Results

One major goal of our approach was the development of a solution which can be considered a simple “add-on” to an existing hierarchical radiosity system. We implemented the algorithm on two very different rendering architectures, namely BRIGHT (iMAGIS) and VISION (Uni-

versity of Erlangen).

We have tested our implementation on several test scenes, shown in Figures 3 and 6. The scenes in Figure 3 were used for the interactive viewing test in BRIGHT. The first scene shows three light sources colored red, green and blue, illuminating a very glossy, small reflector. This reflector in turn indirectly illuminates a diffuse wall. The second scene is a glossy sphere illuminated by a small source and a glossy floor. These in turn produce indirect glossy effects on the lower part of the sphere and the diffuse ceiling. Finally, the ‘‘Simple soda’’ scene is a simplified version of the ‘‘Soda Shoppe’’ scene. In BRIGHT, we require tessellation of all objects initially, which results in a high number of initial objects; in VISION, objects are not initially tessellated. This explains the low number of initial objects in the complete ‘‘Soda Shoppe’’ scene, used for Figure 6.

### 6.1 Radiance Clustering vs. Illumination Samples

In BRIGHT we have implemented both Radiance Clustering (RC) and the Illumination Samples (IS) approach. We have compared running time and memory usage for the RC and IS approaches. The threshold value has the same meaning for both approaches, since we are using a ‘‘relative’’ refiner. Visual inspection also shows that the resulting images are equivalent for the same parameter values. All timings are on a 195Mhz R10k SGI workstation.

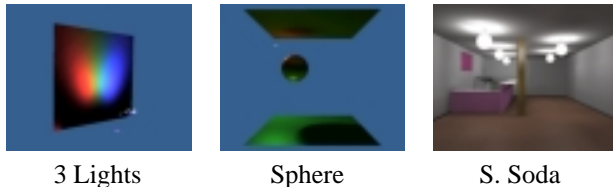


Figure 3: Test scenes using illumination samples

### 6.2 Memory Consumption

In Table 2 we show the memory statistics for the test scenes used. In particular we list the different scenes with the  $\epsilon$  accuracy threshold (see Section 4.1), and the corresponding number of directional distribution basis functions used for the solution by the Radiance Clustering (RC) and Illumination Samples (IS) approach. The rightmost column shows the percent gain of the illumination samples approach.

Memory usage is clearly reduced using Illumination Samples compared to the Radiance Clustering approach for all scenes. The gain varies from 37 % to 41% in the best case. This is mainly due to the fact that Radiance Clustering requires the additional intermediate directional distribution functions to be able to correctly per-

	$\epsilon$	m/M	IS	RC	
3 Lig.	0.5	1/3	8618	13866	38%
3 Lig.	0.1	1/3	8820	14068	37%
3 Lig.	0.5	1/4	27306	43510	37%
3 Lig.	0.5	1/5	79218	125944	37%
Sph.	0.5	1/3	2114324	3598720	41%
Soda	0.5	1/3	2097534	3339794	37%

Table 2: Gain in *memory* usage from the use of the Illumination Samples algorithm.  $\epsilon$  is the accuracy threshold and m/M the min/max levels.

form the push-pull operation (see Section 2.2).

### 6.3 Computation Time

In Table 3 we show the computation time statistics for the test scenes used. In particular we list the different scenes with the  $\epsilon$  threshold, and the corresponding computation time for the solution by the two approaches (RC and IS). The rightmost column shows the percent memory gain of the illumination samples approach.

	$\epsilon$	m/M	IS	RC	
3 Lig.	0.5	1/3	44.6 s	90.1 s	50%
3 Lig.	0.1	1/3	44.2 s	90.0 s	51%
3 Lig.	0.5	1/4	49.3 s	95.5 s	48%
3 Lig.	0.5	1/5	58.6 s	105.3 s	44%
Sph.	0.5	1/3	4167.1 s	6492.9 s	34%
Soda	0.5	1/3	5207.6 s	7117.4 s	27%

Table 3: Gain in *computation time* from the use of the Illumination Samples algorithm.  $\epsilon$  is the accuracy threshold and m/M are the min/max levels.

For all scenes the illumination samples approach provides a speedup of at least 27%. This is mainly due to the reduction in the number of hierarchy traversals, and also the reduction in the number of triangles used to represent the directional distributions as discussed above.

As expected and confirmed by the experimental results, the Illumination Samples approach reduces both memory and computation time with respect to Radiance Clustering. The images produced by both approaches are essentially indistinguishable.

### 6.4 Visual Quality/Comparisons

We qualitatively compare the visual quality of the images of Radiance Clustering and Illumination Samples with those from a path-tracer or the RADIANCE system. The path-tracer tests are rendered using an in-house implementation of the next-event estimation [11]. In Fig. 4 and 5 we show the reference path-tracer or RADIANCE images and the corresponding Illumination Samples images together with the computation times for the test scenes.

There are several interesting observations that we can infer from these examples:

- 1) In the case of the three light scene, RADIANCE runs



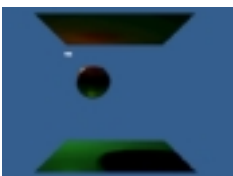
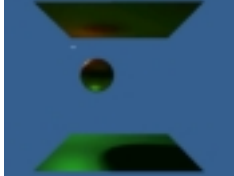
Reference	Illumination Samples
	
Radiance 6537s	59s
	
Radiance 1303s	4167s

Figure 4: Reference solutions (RADIANCE) compared to IS solutions for three lights and sphere scenes.


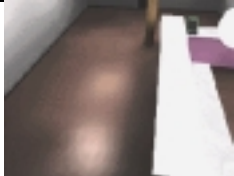
Reference	Illumination Samples
	
Path Tracer 51873s	5208s

Figure 5: Simple Soda reference solution (path-tracer at 455x364 resolution) compared to IS solution.

into sampling problems. Even after more than an hour and a half of computation it does not converge. In contrast, illumination samples achieves a solution in less than a minute, which is in addition viewable from any direction interactively <sup>2</sup>.

2) The computation times of IS are either lower or in the same order of magnitude as those of the reference solutions. The important thing to remember is that the IS solutions can be viewed *interactively*, while the reference (path-tracer or Radiance) require the same amount of time (tens of minutes or even hours) for *every* image.

3) Path-tracing images are very noisy. The “smooth-shaded” solutions produced by IS do not suffer from this problem. Despite being approximate, the smooth-shaded images are therefore much better suited to interactive applications, where noise and flickering are very distracting.

We thus believe that our approach has great promise, since it can be used to generate low to moderate quality solutions for glossy environments, as well as produce solutions suitable for interactive viewing.

<sup>2</sup>A bi-directional path-tracer, or photon-map which would consider the reflection as a caustic may generate better results.

## 6.5 A More Complex Scene

As a last test we applied the Illumination Sample method to a more complex scene, the Soda Shoppe, one of the RADIANCE test scenes. Our version consists of 1,644 initial patches, several of which are non-planar, including the spherical light sources. About one third of the patches are non-diffuse. Since bounded form factor computation is used [20], no initial tessellation of these objects was necessary, which would have increased the initial complexity significantly. The scene is not yet really complex in the sense of an industrial-size model, but sufficiently non-trivial to impose severe problems on previous finite-element radiance methods.

The solution shown in Figure 6 was computed with the implementation of VISION, which incorporates the “shooting” solution described previously (Section 3.5). It was obtained in 8,488 seconds and contains 29,138 final patches. 91% of the computation time was spent on propagation, which in turn is dominated by visibility (97%), only 9% was used for push/pull including the reflection. 743,284 links were computed, resulting in 29,138 patches. Note that the used VISION implementation does not yet incorporate smooth reconstruction capabilities, so that the patch boundaries are clearly visible.



Figure 6: A solution to the glossy soda shoppe, computed in 8,488 seconds.

By far most of the computation is spent on visibility, as with diffuse radiosity computation. This indicates that we were able to reduce the overhead introduced by explicitly storing directional illumination information to reasonable levels. The more costly push/pull step was expected, but it is interesting to notice that it still requires only about 10% of the overall computation for this particular scene. For other scenes with more glossy objects, this percentage is somewhat larger, although always “reasonably small”. Note that this is only true for using shooting instead of gathering! For gathering the push/pull times can increase significantly.

## 7 Conclusions

We have presented a novel solution to global illumination simulation for glossy environments. Our new algorithm

is an important step towards interactive walkthroughs of globally illuminated glossy scenes: (i) We introduced the *Illumination Samples* algorithm which represents incoming light more accurately and efficiently, both in memory and computation time. (ii) We have used an adaptive hierarchical finite-element basis to store outgoing light, in a manner suitable for interactive viewing. This allows fine control of the memory/quality tradeoff, which was not possible in previous solutions. (iii) These algorithms can be implemented with marginal effort over an existing hierarchical radiosity system, by confining the modifications to a small number of phases and data structures. (iv) Interactive viewing of the glossy global illumination solutions is achieved by suitably refining the directional representation of outgoing light and accelerating the display process.

Our solution however is still quite memory-consuming requiring in the order of tens of Mbytes for the smallest scenes and hundreds of Mbytes for the more complex.

To remedy these problems we need to generalise the multi-resolution nature of our solution. Notably we will introduce a multi-resolution representation of outgoing light, which will allow significant savings in memory at the cost of lower visual quality. This approach will require a novel representation as well as a novel refinement algorithm. This representation can be used both for the actual lighting simulation as well as for display.

### Acknowledgements

This research was funded by the ESPRIT Open LTR SIMULGEN (#25772) Thanks to Cyril Soler (iMAGIS) and Ignacio Martin (GGG) for coding help, and to Bruce Walter for proofreading. iMAGIS is a joint project of CNRS/INRIA/UJF/INPG.

### 8 References

- [1] L. Aupperle and P. Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. In *Computer Graphics (SIGGRAPH '93)*, pages 155–162, August 1993.
- [2] S. E. Chen, H. E. Rushmeier, G. Miller, and D. Turner. A progressive multi-pass method for global illumination. *Computer Graphics (SIGGRAPH '91)*, 25(4):165–174, July 1991.
- [3] P. H. Christensen, D. Lischinski, E. Stollnitz, and D. H. Salesin. Clustering for glossy global illumination. *ACM Trans. on Graphics*, 16(1):3–33, January 1997.
- [4] P. H. Christensen, E. J. Stollnitz, and D. H. Salesin. Global illumination of glossy environments using wavelets and importance. *ACM Trans. on Graphics*, 15(1):37–71, January 1996.
- [5] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. *Computer Graphics (SIGGRAPH '84)*, 18(3):137–145, July 1984.
- [6] S. J. Gortler, P. Schröder, M. Cohen, and P. M. Hanrahan. Wavelet radiosity. *Computer Graphics (SIGGRAPH '93)*, 27:221–230, August 1993.
- [7] P. Hanrahan, D. Salzman, and L. Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics (SIGGRAPH '91)*, 25(4):197–206, 1991.
- [8] H. Wann Jensen. Global illumination using photon maps. In Xavier Pueyo and Peter Schröder, editors, *Rendering Techniques '96 (7th EG Workshop on Rendering)*, pages 21–30. Springer, June 1996.
- [9] J. T. Kajiya. The rendering equation. *Computer Graphics (SIGGRAPH '86)*, 20(4):143–150, August 1986.
- [10] A. Keller. Quasi-monte carlo radiosity. In X. Pueyo and P. Schröder, editors, *Rendering Techniques '96 (7th EG Workshop on Rendering)*, pages 101–110, Porto, June 1996. Springer.
- [11] E. Lafortune. *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Katholieke Universiteit Leuven, 1996.
- [12] S. N. Pattanaik and S. P. Mudur. Computation of global illumination by Monte Carlo simulation of the particle model of light. *Third EG Workshop on Rendering*, pages 71–83, May 1992.
- [13] P. Schröder and P. Hanrahan. Wavelet methods for radiance computations. In *Photorealistic Rendering Techniques (5th EG Workshop on Rendering)*, pages 303–311, Darmstadt, June 1994. Springer.
- [14] P. Shirley, C. Wang, and K. Zimmerman. Monte carlo techniques for direct lighting calculations. *ACM Trans. on Graphics*, 15(1):1–36, 1996.
- [15] F. X. Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Trans. on Vis. and Comp. Graphics*, 1(3):240–254, September 1995.
- [16] F. X. Sillion, G. Drettakis, and C. Soler. A clustering algorithm for radiance calculation in general environments. In *Rendering Techniques '95 (6th EG Workshop on Rendering)*, pages 196–205. Springer, August 1995.
- [17] F. X. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. *Computer Graphics (SIGGRAPH '89)*, 23(3):335–344, July 1989.
- [18] B. Smits, J. Arvo, and D. Greenberg. A clustering algorithm for radiosity in complex environments. *Computer Graphics (SIGGRAPH '94)*, pages 435–442, July 1994.
- [19] M. Stamminger, H. Schirmacher, P. Slusallek, and H-P. Seidel. Getting rid of links in hierarchical radiosity. *Computer Graphics Forum (EUROGRAPHICS '98)*, 17(3):165–174, September 1998.
- [20] M. Stamminger, P. Slusallek, and H-P. Seidel. Bounded radiosity – illumination on general surfaces and clusters. *Computer Graphics Forum (EUROGRAPHICS '97)*, 16(3):309–318, September 1997.
- [21] M. Stamminger, P. Slusallek, and H-P. Seidel. Bounded radiosity – finding good bounds on clustered light transport. *Conf. Proc. of Pacific Graphics '98*, 1998. <http://www9.informatik.uni-erlangen.de/eng/research/pub1998/>.
- [22] E. Veach and L. Guibas. Bidirectional estimators for light transport. In *Photorealistic Rendering Techniques (5th EG Workshop on Rendering)*, pages 145–167, Darmstadt, June 1994. Springer.
- [23] E. Veach and L. J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. *Computer Graphics (SIGGRAPH '95)*, pages 419–428, aug 1995.
- [24] E. Veach and L. J. Guibas. Metropolis light transport. *Computer Graphics (SIGGRAPH '97)*, pages 65–76, aug 1997.
- [25] B. Walter, G. Alppay, E. P. F. Lafortune, S. Fernandez, and D. P. Greenberg. Fitting virtual lights for non-diffuse walkthroughs. In T. Whitted, editor, *SIGGRAPH '97*, pages 45–48, August 1997.
- [26] G. Ward. Radiance web site. <http://radsite.lbl.gov/radiance>.
- [27] G. J. Ward. The RADIANCE lighting simulation and rendering system. *Computer Graphics (SIGGRAPH '94)*, pages 459–472, July 1994.
- [28] T. Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, June 1980.
- [29] H. R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. *Computer Graphics (SIGGRAPH '93)*, pages 213–220, August 1993.