

Fast and Detailed Approximate Global Illumination by Irradiance Decomposition

Okan Arikan
okan@cs.berkeley.edu
University of California, Berkeley

David A. Forsyth
daf@cs.uiuc.edu
University of Illinois, Urbana-Champaign

James F. O'Brien
job@cs.berkeley.edu
University of California, Berkeley

Abstract

In this paper we present an approximate method for accelerated computation of the final gathering step in a global illumination algorithm. Our method operates by decomposing the radiance field close to surfaces into separate far- and near-field components that can be approximated individually. By computing surface shading using these approximations, instead of directly querying the global illumination solution, we have been able to obtain rendering time speed ups on the order of $10\times$ compared to previous acceleration methods. Our approximation schemes rely mainly on the assumptions that radiance due to distant objects will exhibit low spatial and angular variation, and that the visibility between a surface and nearby surfaces can be reasonably predicted by simple location- and orientation-based heuristics. Motivated by these assumptions, our far-field scheme uses scattered-data interpolation with spherical harmonics to represent spatial and angular variation, and our near-field scheme employs an aggressively simple visibility heuristic. For our test scenes, the errors introduced when our assumptions fail do not result in visually objectionable artifacts or easily noticeable deviation from a ground-truth solution. We also discuss how our near-field approximation can be used with standard local illumination algorithms to produce significantly improved images at only negligible additional cost.

Keywords: Final gather, approximate global illumination, spherical harmonics.

CR Categories: I.3.3 [COMPUTER GRAPHICS]: Picture / Image Generation, I.3.7 [COMPUTER GRAPHICS]: Three Dimensional Graphics and Realism

1 Introduction

Current rendering techniques can produce highly compelling, beautiful images that realistically capture a variety of interesting effects such as color bleeding between nearby surfaces or indirect shadows due to strong reflected illumination. Capturing these effects requires computing how light propagates and reflects within the scene environment, and the term global illumination refers to the class of algorithms that perform these computations.

Unfortunately, computing a global illumination solution is a fundamentally difficult task, requiring solutions to integral equations that recursively express the radiance leaving a surface as a function

From the ACM SIGGRAPH 2005 conference proceedings.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ACM SIGGRAPH 2005, Los Angeles, CA
© Copyright ACM 2005



Figure 1: An image rendered with our algorithm. Notice the fast change in global illumination near surface relief. This image takes about 5 minutes to render (from end to end) on a desktop PC.

of the radiance leaving other surfaces. The algorithms available for computing these solutions must either discretize the environment or employ some form of stochastic sampling. However, producing solutions free of objectionable discretization artifacts and sampling noise often demands impractically large computation times and/or storage space.

A widely adopted technique, called final gathering, approximates a high quality global illumination solution by combining a relatively coarse global illumination solution with high resolution sampling of select surfaces. First, a standard global illumination method, such as photon mapping, is used to compute a rough solution. Afterward, an image is generated using Monte Carlo techniques to integrate indirect irradiance from the rough solution. Monte Carlo integration is performed only for surfaces that are directly visible, or clearly reflected. The result captures illumination features of the global solution, but because diffuse and glossy surfaces blur reflections, objectionable artifacts present in the rough solution do not appear in the final image. Although final gathering provides a large overall improvement in total rendering time (for an equal quality result), one typically finds that the cost of the final gather dominates the cost of computing the coarse global illumination solution. As a result, improving the speed of the final gather is an effective way of reducing overall rendering time.

One such approach is irradiance caching, which takes advantage of the smoothing property of diffuse surfaces by caching the value of the irradiance integral at points on surfaces. The cached values are then re-used for nearby points using scattered data interpolation. Details, such as precise sampling regime and the interpolation scheme vary from implementation to implementation.

However, irradiance caches are not efficient around geometric detail because they must maintain a denser set of samples to account

for the potentially rapidly changing irradiance. One can arbitrarily limit the sample density, but doing so often obliterates perceptually important lighting variations that reveal surface detail and geometric texture.

In this paper we propose an approximate method for accelerated final gathering that splits the irradiance integral into separate far- and near-field components that can then be approximated individually. As the name suggests, the far-field term accounts for the power coming from distant surfaces, while the near-field term accounts for the reflections and occlusions due to nearby geometry. Our approximation schemes rely mainly on the assumptions that radiance due to distant objects will exhibit low spatial and angular variation, and that the visibility between a surface and nearby surfaces can be reasonably predicted by simple location- and orientation-based heuristics. Motivated by these assumptions, our far-field scheme uses scattered-data interpolation with spherical harmonics to represent spatial and angular variation, and our near-field scheme employs an aggressively simple visibility heuristic. The assumptions underlying our far field scheme are similar to those which motivate irradiance caching, except we cache radiance, rather than irradiance. Since only the radiance due to distant objects is cached our required sampling density is not impacted by local geometry.

Using this decomposition, we are able to approximate final gathering in complex scenes displaying a variety of global illumination effects (see Figure 1). Our tests show that the method uses far fewer visibility queries (ray traces) and achieves an order of magnitude speedup on average compared to irradiance caching. For our test scenes, the errors introduced at locations where the underlying assumptions fail do not result in visually objectionable artifacts or easily noticeable deviation from a ground-truth solution.

We also discuss how our near-field approximation can be used with standard local illumination algorithms to produce significantly improved images at only negligible additional cost. This local enhancement should be particularly useful in production environments where artists often employ unrealistic light placement to achieve a desired look. Like ambient occlusion shading, our local enhancement greatly enhances the realistic appearance of an object but it is significantly cheaper to compute.

2 Related Work

Global illumination has been one of the most heavily researched branches of computer graphics. A thorough review of available methods cannot be presented here, but [Sillion and Puech, 1994] and [Dutr e et al., 2003] provide a good overview of finite element and Monte Carlo based global illumination techniques. We use photon mapping [Jensen, 2001], to generate the global illumination solutions used with our final gathering algorithm.

Originally, final gathering was developed for use with radiosity methods to obtain visually pleasing results from blocky piecewise-constant solutions. [Lischinski et al., 1993] introduced an object space refinement method for increasing the visual quality of the solution of a hierarchical radiosity step. [Rushmeier, 1988], [Rushmeier et al., 1993] use a coarse geometric model for a radiosity solution which is queried by a Monte Carlo algorithm to create the final picture. [Zimmerman and Shirley, 1995] decrease the variance caused by Monte Carlo integration by re-classifying bright reflecting surfaces in a coarse radiosity solution as light sources to ensure that those surfaces are sampled. [Scheel et al., 2001] demonstrated that a final gather could be accelerated by using the link information from a prior radiosity step to identify important senders. [Scheel et al., 2002] also uses the link information to find the senders whose power contribution can be interpolated. A similar idea has also been explored by [Bekaert et al., 1998] where per pixel final gather was obtained using a Monte Carlo integration scheme which uses radiosity solution for importance sampling. All of these methods generate a visually pleasing result from a coarse global illumina-

tion solution. However, they still result in an expensive final gather step that must compute expensive irradiance for each pixel in an image.

In an influential paper, [Ward et al., 1988] introduced the concept of performing very accurate final gather on surfaces only at scattered points and then interpolating those values. Known as irradiance caching, their method adapts the sample locations based on how quickly the sampled final gather result changes. A later paper [Ward and Heckbert, 1992] refined the algorithm to decrease the discontinuities due to the insertion of new samples while rendering.

Instead of caching irradiance, our far-field approximation caches incident radiance. Our near-field correction then adjusts the irradiance integral obtained from the cached representation by explicitly accounting for nearby geometry. This will allow us to cache the incident radiance at a relatively small number of locations and still approximate the high-frequency detail in indirect illumination. A similar idea for extending irradiance caches by caching the incident radiance has been explored by [Krivanek et al., 2005]. However, their method aims at handling glossy surfaces whereas our method attacks oversampling. [Tabellion and Lamorlette, 2004] presents an improvement over irradiance caches, but they must also sample more densely around geometric detail and their limit on the sampling density can lead to smoothed interpolation. Similar to our work, [Greger et al., 1998] stores incident radiance in space, but their method suffers from the same dense sampling problem of irradiance caches.

The behavior of diffuse surfaces in close proximity to other surfaces (such as corners) has been explored in [Rathsfield, 1999] and [Atkinson, 2000]. Their results confirm the existence of high-frequency change, or light reflexes, around corners.

The quality of the results from many of the methods above can be explained by research on the relationship between the appearance of diffuse surfaces and incident illumination. [Epstein et al., 1995] and [Basri and Jacobs, 2000] showed that the color of a diffuse surface was a function of the low dimensional representation of the incident illumination. [Ramamoorthi and Hanrahan, 2001b] used low order spherical harmonics, and demonstrated harmonics to be a good and compact representation. [Gautron et al., 2004] extend the idea onto hemispherical basis functions which provide a more accurate representation for functions defined on hemispheres. Spherical harmonics have also been successfully used in [Ramamoorthi and Hanrahan, 2001a] to represent environment maps for diffuse surfaces. This idea has been expanded in [Sloan et al., 2002] with the introduction of transfer functions to take self occlusions or inter-reflections into account. However such methods require expensive precomputation to obtain the transfer functions.

3 Overview

The final gather in a rendering process computes shading for the surfaces that are either directly visible, or visible by a mirror-like reflection. Presuming that the BRDF of these surfaces can be separated into diffuse and specular components, we observe that the directional dependence of the specular component makes it well suited for fast computation using standard sampling methods while, in contrast, computing the diffuse reflection by sampling can be quite costly. Accordingly, our method for accelerating final gathering focuses on speeding up this bottleneck by replacing the costly irradiance integral used for computing diffuse illumination with a fast approximation.

Let P be a point on some surface we wish to shade, with normal vector N . The incoming radiance at P from a direction ω is $L(P, \omega)$. If we define ρ as the diffuse reflectivity (albedo) of the surface, the diffusely reflected light, or *radiosity*, is defined as follows:

$$B = \frac{\rho}{\pi} \int_{\Omega} (N \cdot \omega) L(P, \omega) d\omega \quad (1)$$

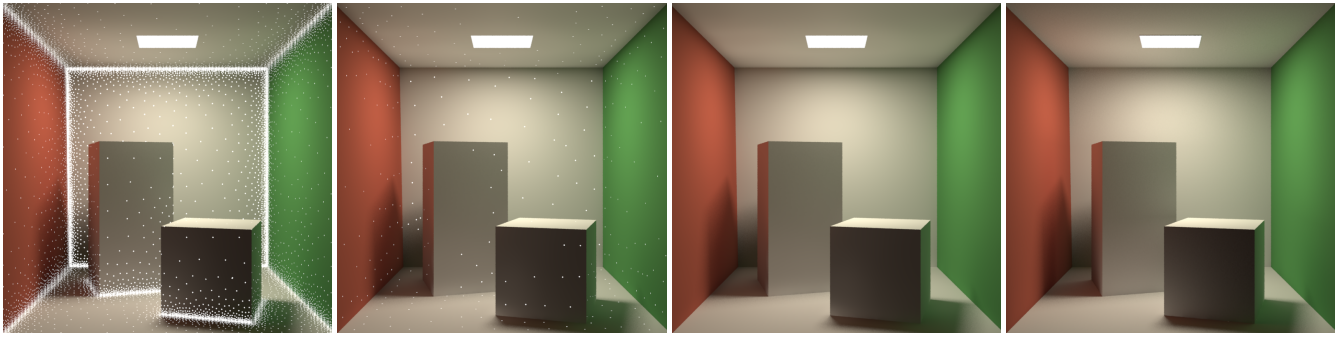


Figure 2: The leftmost image shows the locations of the irradiance cache samples as white dots. The samples are concentrated around the corners to accurately capture the swift change in illumination. The center-left image shows the locations of our spherical harmonic samples. Since the high-frequency changes in diffuse brightness around corners are captured explicitly by our correction term I_N , we require far fewer samples. The image on the center-right shows the sample points removed to show the final image computed using our technique, and the rightmost image shows the ground truth, which is obtained by estimating the irradiance integral (equation 1) using Monte Carlo techniques.

The integral in this equation computes the power that P receives and is called *irradiance*.

The definition of irradiance is implicitly recursive: Finding the incoming radiance $L(P, \omega)$ involves computing the irradiance integral at the source of the radiance. One could perform the computation using path integration (*Monte Carlo* variants) or discretize the scene and convert the problem into a linear system (*radiosity* variants, not to be confused with the term for diffusely reflected light). However, these methods are often too expensive to obtain high quality solutions by themselves.

As we have discussed, the common practice of final gathering uses a Monte Carlo technique to compute the irradiance integral at visible surfaces, but it avoids costly recursion by using a precomputed coarse global illumination solution as a proxy for $L(P, \omega)$. This practice is called Final Gathering, because it “gathers” from the coarse global illumination solution.

Final gathering is expensive because it involves shooting many sample rays for each point to be shaded. Although computing the irradiance integral is expensive, it can often be re-used: if the value of the integral is not changing very rapidly on a surface, it can be computed sparsely, and be interpolated using scattered data interpolation techniques. This technique is often called irradiance caching [Ward et al., 1988; Ward and Heckbert, 1992].

As a point moves across a surface, far away surfaces move less on the incident hemisphere of the point than nearby surfaces (parallax effect). This means the irradiance integral in Equation 1 does not change very rapidly (as P moves on a surface) if the source of the incoming radiance $L(P, \omega)$ is far away. Similarly, if there are other surfaces nearby, then these surfaces can reflect and obstruct light and create rapid changes in the irradiance integral. For this reason, irradiance caching methods are forced to form spatially dense caches in regions of geometric detail (as shown in Figure 2.). Even a phenomenon as familiar as a corner attracts many cache samples to capture potential changes in the irradiance.

Let us designate all the visible points that are farther from P than a distance threshold α as *distant*, and all other points as *nearby*. The incident hemisphere Ω over P can then be partitioned into Ω_D and Ω_N ($\Omega = \Omega_D \cup \Omega_N$) according to whether a distant or near point projects onto a given portion of the hemisphere. Given this distinction, we can split the irradiance integral into two components: Power coming from distant surfaces and power coming from nearby surfaces. Power coming from distant surfaces can be handled easily, because it does not change rapidly on surfaces. Power coming from nearby surfaces can also be handled easily, because it is by definition, a local computation. Formally, we write:

$$B = \frac{\rho}{\pi} \int_{\Omega_D} (N \cdot \omega) L_D(P, \omega) d\omega$$

$$+ \frac{\rho}{\pi} \int_{\Omega_N} (N \cdot \omega) L_N(P, \omega) d\omega \quad (2)$$

Here we denote the incident radiance from distant points with L_D and from nearby points with L_N . In practice, it is difficult to determine Ω_D directly. However finding Ω_N is relatively straightforward by looking at the nearby scene triangles and finding the solid angle they subtend. These properties suggest that we define L_D^* — an extension of L_D from Ω_D to Ω by ignoring any occlusions due to nearby surfaces. Now $L_D^* = L_D$ on Ω_D so we can rewrite the integral above as follows:

$$\begin{aligned} B &= \frac{\rho}{\pi} \int_{\Omega} (N \cdot \omega) L_D^*(P, \omega) d\omega \\ &+ \frac{\rho}{\pi} \int_{\Omega_N} (N \cdot \omega) [L_N(P, \omega) - L_D^*(P, \omega)] d\omega \\ &= \frac{\rho}{\pi} (I_D + I_N) \end{aligned} \quad (3)$$

In equation 3, the I_D can be thought as the irradiance due to distant surfaces (*distant term*) and I_N can be thought as a correction term that accounts for the power being reflected or being occluded due to nearby surfaces (*near term*). Sections 4 and 5 will explain how these terms respectively can be approximated efficiently. Section 7 will discuss the effect of the parameter α which distinguishes distant from nearby surfaces.

4 Distant Term

The distant illumination $I_D = \int_{\Omega} (N \cdot \omega) L_D^*(P, \omega) d\omega$ does not change rapidly over surfaces. We could cache I_D sparsely and use scattered data interpolation techniques, an approach that would be similar to irradiance caching but with the difference of separating irradiance due to far and near surfaces. However, rather than caching the value of I_D (irradiance), we cache a compact representation of L_D^* (radiance). Caching L_D^* instead of I_D will allow us to compute the second term of the correction term which subtracts the portion of I_D that corresponds to distant surfaces obscured by nearby ones. It also facilitates using bump or displacements maps where rapid variation in the surface normal causes rapid changes in I_D but not L_D^* . [Krivanek et al., 2005] demonstrates additional advantages of storing the incident radiance and describes how the information can be used to efficiently handle glossy surfaces.

L_D^* is a function of position and direction: It gives us the incident distant radiance from a given direction arriving at a given position. Spherical harmonics provide a nice representation for such functions defined over a sphere. As demonstrated by [Ramamoorthi and Hanrahan, 2001b], for the diffuse component of a BRDF,

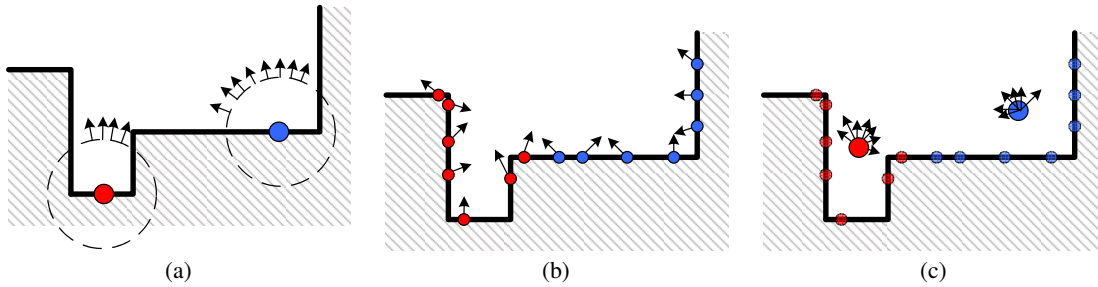


Figure 3: (a) Constructing spherical harmonic samples by shooting rays from a single location can undersample certain directions. Since we only consider sources of incident radiance that are farther away from the sample point than α , a sample computed at the red point would only have samples coming from a very narrow range of directions. The blue sample also has a similar problem because it does not contain any rays shooting towards the right. (b) To avoid this situation, we spread the origin of our rays onto surfaces. We still consider the intersections that are far enough away. But when we compute the harmonic coefficients, we assume all the rays originate from the geometrical average of the origins of the rays. (c) This way when we compute the harmonic sample for the red point we do not suffer from the undersampling problems.

only 9 coefficients provide a sufficient representation of the incoming radiance, and so we use these 9 coefficients for representing L_D^* at a fixed point. Given a query point, the distant incident radiance is obtained by interpolating spherical harmonic coefficients from nearby samples. Note that our query points will be points on surfaces where we wish to perform a shading calculation, but the locations where we cache L_D^* may be arbitrary points in space, though presumably they will be near surfaces we wish to shade. Because spherical harmonics provide an orthonormal basis, I_D can be efficiently evaluated using a dot product of harmonic coefficients for $\max(N \cdot \omega, 0)$ and $L_D^*(P, \omega)$.

We compute the projection of L_D^* onto the harmonic basis using a sampling fit procedure. The incident hemisphere Ω of P is randomly sampled and incoming radiance at these sample points are obtained by ray-cast queries into the coarse global illumination solution. Rays whose intersections are closer to P than α are discarded. The spherical harmonic coefficients approximating L_D^* are computed by a least-squares fit to these incident radiance samples.

4.1 Sampling

Irradiance caching algorithms typically cache irradiance values at specific points on the visible surfaces. The corresponding approach would be for us to select a set of surface points, sample the incoming radiance over a hemisphere centered at each of those points, and then store the resulting harmonic coefficients at each point. However, we have noticed that doing so can create ugly bias artifacts if our implicit assumptions about the smoothness of distant illumination fails. Instead we opt for a procedure that will produce a more benign smoothed solution.

We start with the set of all shading points that will need the global illumination computation¹. We then attempt to fit a spherical harmonic to the incoming radiance for these points by shooting rays from random points in this set in random directions (in the incident hemisphere of the selected points). A single harmonic sample is not good enough to represent the incoming distant radiance if any of these rays intersects a surface within the bounding sphere of the samples. If this is the case, we split (using k-means) the set of shading points into two, and attempt to fit harmonics for each of the subsets recursively. A set is not split if the diameter of the bounding sphere for the shading points in the set is less than α , or if the rays do not intersect other surfaces within the bounding sphere. This method ensures that the harmonics samples will not be closer to each other than α and that the places without geometric detail will not create lots of harmonic samples. Spawning rays from random shading points corresponds to spreading the origins of the rays used to compute the harmonics onto the surfaces. (See Figure 3.)

¹If rendering a 640 by 480 image with one sample per pixel, we have $640 \times 480 = 307200$ shading points.

The number of rays that we use to compute a harmonic sample is roughly equivalent to the number of rays that irradiance caches use to compute a sample. In practice, we use 512 rays per harmonic in all of our examples. Once a harmonic sample is computed using least squares fit to the incident radiances coming along these rays, it is stored at the geometric center of the shading points in the set. For each sample, we also store the radius of the bounding sphere of the shading points for which the sample was collected. The samples are interpolated using scattered data interpolation, similar to irradiance caches: For a query point, we locate all the harmonic samples that are near the query point and perform a normalized distance weighted sum of the harmonic coefficients.

5 Near Term

While distant illumination generally exhibits slow variation over surfaces, the illumination arriving from, and occluded by, near surfaces can vary relatively rapidly. In fact, as illustrated by Figure 7, lighting variation induced by near surfaces often provides many of the cues that reveal the geometric structure of an object. As a result, the smoothness-dependent interpolation method we used for distant terms cannot work well for estimating the irradiance from near surfaces.

Instead we compute an approximation of I_N directly at each shading point. By definition, this computation is local and only involves surfaces within distance α . However, even limited to this small neighborhood, we have found the required visibility tests to be expensive. So rather than actually computing the visibility terms we make use of a fast, aggressive heuristic that approximates visibility based on the relative location and orientation of the surfaces.

First we rewrite the nearby correction term of equation 3 as a sum over all nearby triangles:

$$I_N \approx \sum_{t \in T} FF(t) [B(t) / \pi - L_D^*(P, \omega_t)] \quad (4)$$

In this equation T is the set of triangles whose centroids are closer to the shading point than α . We find such triangles using an octree decomposition of the scene. ω_t is the direction towards the centroid of the triangle, $FF(t)$ and $B(t)$ respectively stand for the form factor of the triangle t to P and its radiosity, which we assume is constant over the triangle.

To avoid problems caused by large triangles that have significant portions within and without a distance α , triangles larger than α are split into smaller ones. Splitting triangles also justifies treating the radiosity as constant over a triangle, as does the fact that we are referring to radiosity from the coarse global illumination solution. A third benefit of splitting triangles is that it prevents shading discontinuities that would be generated if a large triangle's center moved from inside α to outside α as we moved a small distance on

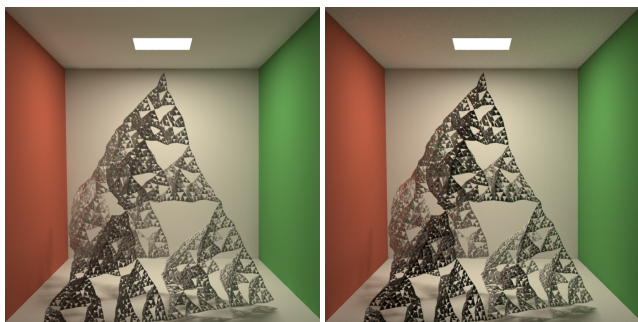


Figure 5: This scene demonstrates a deformed fractal in a Cornell box. The image on the left is the ground truth obtained using Monte Carlo integration. The image on the right is the result of our algorithm. The fractal violates our visibility assumption, because it contains many small triangles that reflect/occlude power from far away surfaces. Even in this contrived scenario, the errors introduced by our algorithm are not too distracting.

the rendered surface. For the scenes that we rendered, this process did not more than triple the number of scene triangles, because this splitting only attacks the big scene triangles.

Our visibility heuristic discards any triangles that are either back-facing to the shading point, P , or that are completely below the plane passing through P with normal N as those triangles cannot provide direct illumination to P . We then assume that all other triangles within distance α are visible from P . Although one can easily come up with example geometries where this aggressive, gross approximation would be wrong, we have found that it works surprisingly well in practice. Later, in section 7, we will discuss this approximation further.

Recall that $L_D^*(P, \omega_t)$ is the approximation of incoming radiance from direction ω_t due to distant sources computed using our spherical harmonic-based interpolation. We also treat this quantity as constant over directions spanned by one of the split triangles, and justify the approximation with L_D 's smoothness and the small size of the split triangles.

The remaining quantity that must be computed is the form factor between P and a triangle, $FF(t)$. It is this computation that would normally involve visibility information. Standard methods would render the triangles onto a hemi-cube or determine visibility by sampling. While we could easily use one of these methods in our form factor computation, doing so would reduce the speed improvement of our algorithm over irradiance caching. Instead, we assume that the triangles selected by our heuristic are all completely visible, and compute form factors using the analytical polygon to point form factor formula given by [Hottel and Saforim, 1967].

6 The Whole Algorithm

Once a coarse global illumination pass has been completed, our algorithm constructs the image in two passes. In pass 1, we compute the spherical harmonics capturing the incident radiance due to distant surfaces. Because the spherical harmonic samples represent only the distant illumination and are used for the diffuse component of the BRDF, they can be scattered sparsely. The near-far threshold α determines the minimum spacing between these samples. In pass 1, we also compute the average radiosity value over all visible triangles and split them if necessary.

In pass 2, for every point we need to shade, we first interpolate the spherical harmonic coefficients to obtain a representation of the incident radiance due to distant surfaces (L_D^*). We then compute the distant term (I_D) and the nearby correction term (I_N) by finding nearby triangles and computing the sum in equation 4. The final diffuse reflected power is obtained by equation 3.

7 Nature of the Approximation

As with any approximation method, the far- and near-term approximations we use can introduce error into the rendered image. For the interpolation method we use for distant illumination, the potential difficulty is that it may smooth over and obscure some perceptible illumination feature. For our simple visibility heuristic, we expect that it may be wrong and treat an occluded triangle as visible.

The potential problems due to our radiance interpolation are generally similar as those that may occur for irradiance caching. However, we gain some advantages by interpolating radiance rather than irradiance. In particular, irradiance changes due to variation in surface orientation will not cause problem for interpolating radiance.

Radiance interpolation can still encounter difficulties for sharp changes in indirect illumination caused by a distant occluder. In practice this happens rarely as most indirect illumination sources tend to appear as area sources and so create soft shadows. Nevertheless, when the situation does arise, our sampling procedure should tend to cluster samples near the illumination boundary to help resolve it.

Our simplistic near-field visibility heuristic will certainly produce errors, and on first consideration one maybe surprised that it works at all. In fact, we originally experimented with fast methods to actually compute local visibility correctly, and if desired one could still use such a method in conjunction with our incident radiance interpolation. However, we have found that for general scenes our much cheaper heuristic results in significant computation saving with little perceptible error.

One possible explanation for why it works well in practice comes from considering which situations break our visibility assumption. The most obvious is a surface that is surrounded by many mutually occluding surfaces. One can easily contrive such an example, but except for some particular examples such as hair or tree leaves, we believe the situation does not often arise in real-world scenes. Additionally, most rendered scenes contain significantly less geometric detail than the real world. Furthermore the commonly occurring examples often require some special treatment that would in any case preclude most generic acceleration methods. (For example, special shadow structures for hair, translucency for leaves.) We also observe that it may be difficult to see surfaces that are surrounded by many occluders so that shading errors on those surfaces may be somewhat excusable.

Finally, there are user-tunable parameters in both the near- and far-field approximations and these parameters can impact the quality of the rendered image. The criteria used for placing interpolation samples and building clusters affects the ability to resolve fast changes in distant lighting. The parameter α changes at what scale occlusions are computed but lighting smoothed, and the scale at which we ignore occlusions but do not smooth illumination. The effect of varying α is illustrated in Figure 4.

In Figure 6 we include a magnified difference image between the our method and the ground truth. The errors we make are concentrated around the corners where the nearby correction term is approximated. The reader is encouraged to evaluate our results perceptually against the ground truth.

In the next section, we discuss the qualitative properties of our approximation and demonstrate our results.

8 Results

To demonstrate our method, we have rendered several scenes and compared them to ground truth results obtained by estimating the irradiance integral in equation 1 using Monte Carlo techniques with a high number of samples. We also compared the images rendered using our algorithm to irradiance caching. Irradiance caching variants represent the current state of art in approximate final gathering and are available in many commercial rendering packages.



Figure 4: This figure demonstrates the effect of the α parameter (used by our approximation) on quality and rendering time. The left image is rendered with $\alpha = 0.25$ in 12 minutes. The overlaid thumbnail shows the locations of the spherical harmonic locations as green dots. The middle image is rendered using $\alpha = 1$ in 5 minutes. Notice that there are fewer samples which explains the speed difference. The rightmost image is rendered with $\alpha = 8$ in 6 minutes. This image uses even fewer spherical harmonic samples but takes longer. This is because more triangles are considered nearby due to the bigger α and the summation in equation 4 contributes to the rendering time. Some of the errors made by our approximation also become visible. For example, the undersides of the arches on the left side are brighter because of the illumination leaking from the brightly illuminated spots. This effort is due to the spherical harmonics being separated too far apart and not being able to represent the incident radiance for the points very accurately.

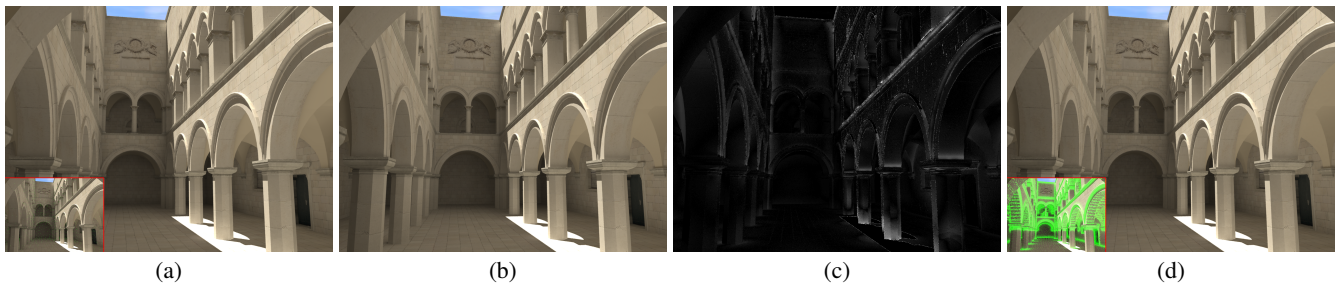


Figure 6: (a) shows the result of our algorithm. The overlaid thumbnail shows the locations of spherical harmonic samples as green dots. The minimum distance between the harmonic samples is our α parameter. (b) is the ground truth obtained using Monte Carlo integration of the irradiance integral. (c) shows a 4 times magnified difference image between (a) and (b). Most of the error is concentrated around high geometric detail. (d) shows the result obtained using irradiance caching. The overlaid thumbnail shows the irradiance cache samples as green dots. The cost of computing a spherical harmonic sample is about the same as obtaining one irradiance cache sample (the cost of the green dots in (a) and (d) is the same). The difference image between (b) and (d) can be found on the Siggraph 2005 DVD.

Figures 1, 4, 6, 8 demonstrate the quality of our approximation in complex environments, displaying complex diffuse interreflections². Notice that high-frequency component of the diffuse interreflection is preserved. For example, in these figures, the niches are correctly shadowed by the nearby geometry. Similarly, the reflections due to nearby geometry are visible in Figure 6 where the pillars on the right touch the ground and around the top right corner of the atrium.

All our figures have been rendered on a dual Athlon 2.2+ system with 2GB of main memory. Only one rendering thread was used. All figures (except the Cornell box) have 1280x960 resolution with 4 samples per pixel. The source code for our method (and our implementation of irradiance caching) is available as a part of an open source renderer *Pixie*.³

Figure 6 was rendered in 0:08 (photon mapping) + 1:24 (1st pass) + 1:18 (2nd pass) = 2:50 (mm:ss) using our method and in 37:04 using irradiance caching. This figure also compares our result against the ground truth. The same atrium model in Figure 8 was rendered in 10:01 (0:46 + 2:17 + 6:58) using our method and in 56:25 using irradiance caching. The time it takes to render this scene using only direct illumination is 8 minutes. The cathedral scene in Figure 4 was rendered in 5:51 (2:02 + 2:22 + 1:31) using our method and in 66:55 using irradiance caching. The night time version of this

scene in Figure 1 was rendered in 7:41 (0:16 + 2:13 + 5:12) using our method and in 70:14 using irradiance caching.

Because our spherical harmonic samples are separated by at least α , we collect fewer samples than irradiance caching. The number of rays shot to compute one spherical harmonic sample is the same as the number of rays that we use to compute an irradiance cache sample. Thus the work of computing a spherical harmonic sample is equivalent to work done for each irradiance cache sample. Fewer samples translate to fewer rays traced and quicker results. On average our method is an order of magnitude faster than irradiance caching while providing the same quality images.

Our method computes the diffusely reflected power. This does not mean it is limited to diffuse surfaces however. Most BRDFs can be split into diffuse and specular lobes to capture reflections and refractions. The usually narrow specular lobes can be importance sampled efficiently. The difficult terms to compute are the diffuse and very glossy (wide) specular lobes. Our method attacks the common diffuse component of BRDF. We believe that by increasing the order of our spherical harmonics, the glossy specular surfaces can also be handled using our method as in [Krivanek et al., 2005].

9 Detail Enhancement

The nearby correction term itself is responsible for capturing the high-frequency reflection / refraction effects due to nearby scene geometry. The local effect can be added without global illumination to enhance the visual detail in a scene. The left portion of Figure 7

²The full resolution versions of all our figures are available on Siggraph 2005 DVD.

³<http://pixie.sourceforge.net>

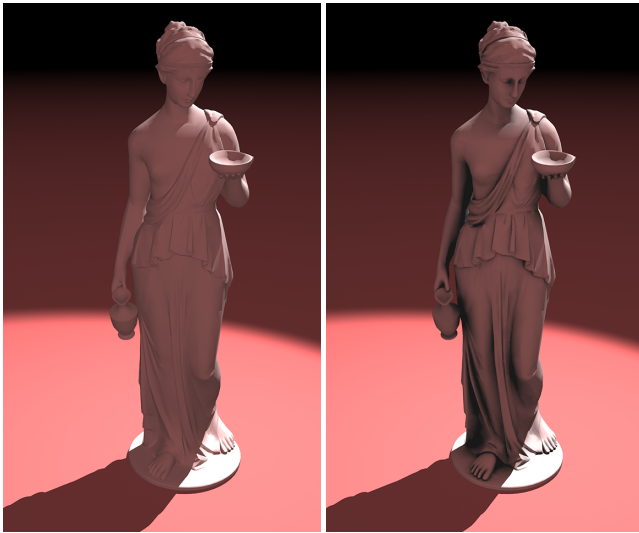


Figure 7: Our method can also be used without global illumination computations. On the left, the statue is illuminated with a spotlight from the top and a bounce light to simulate the light reflecting off the floor. The bounce light is a red point light source without a falloff placed underneath the floor. To compute the right image, we substituted the light contribution from the bounce light as L_D^* in equation 4. The computed I_N is then added on the diffuse color of the object computed using the main light and the bounce light.

shows a rendering of a statue on an infinite red plane. This scene is illuminated by a white spotlight and a reddish bounce light, placed underneath the plane. The right image in Figure 7 has our correction term added. For every shading point, we use the illumination from the bounce light as L_D^* and compute the near-field correction term. We also set the radiosity of the triangles to zero. This essentially is used to approximate the portion of the power coming from the bounce light (which is itself not physically correct) that is occluded by nearby triangles. To compute this image, no rays were traced and the overhead of adding the nearby correction term was 20 seconds (less than 10% of the total rendering time). This technique can easily be used in a traditional pipeline where global illumination is often approximated using many physically incorrect light sources in order to enhance the surface relief and simulate occlusions due to nearby geometry.

Acknowledgements

We thank the other members of the Berkeley Graphics Group and our reviewers for their helpful criticism and comments. We also thank Pixar and Wayne Wooden for providing us with their renderer and helping us with our comparisons. This work was supported in part by ONR N00014-01-1-0890, California MICRO 03-067 and 04-066 and by generous support from Apple Computer, Alias, Pixar Animation Studios, Intel Corporation, Sony Computer Entertainment America, the Hellman Family Fund, and the Alfred P. Sloan Foundation.

References

- ATKINSON, K. E. 2000. The planar radiosity equation and its numerical solution. *IMA Journal of Numerical Analysis* 20, 303–332.
- BASRI, R., AND JACOBS, D. 2000. Lambertian reflectance and linear subspaces. Tech. Rep. MCS00-21, 2000-172R, Waizmann Institute of Science, NEC Research Institute.
- 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.
- DUTRÉ, P., BEKAERT, P., AND BALA, K. 2003. *Advanced Global Illumination*. A. K. Peters Ltd.
- EPSTEIN, R., HALLINAN, P. W., AND YUILLE, A. L. 1995. 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE workshop on physics-based modeling in computer vision*, 108–116.



Figure 8: This figure demonstrates our results in a scene with many light sources. The scene is the same atrium model used in Figure 6. Rather than a single distant light source, this scene is illuminated with 16 yellow point lights placed on the ceilings and a single blue distant light. The overhead of adding the global illumination to this scene using our approximation was less than the time it took to render this scene using direct lighting only.

- GAUTRON, P., KRIVANEK, J., PATTANAİK, S., AND BOUATOUCH, K. 2004. A novel hemispherical basis for accurate and efficient rendering. In *Rendering Techniques 2004: 15th Eurographics Workshop on Rendering*, 321–330.
- GREGER, G., SHIRLEY, P., HUBBARD, P. M., AND GREENBERG, D. P. 1998. The irradiance volume. *IEEE Comput. Graph. Appl.* 18, 2, 32–43.
- HOTTEL, H. C., AND SAFORIM, A. F. 1967. *Radiative Transfer*. McGraw Inc.
- JENSEN, H. W. 2001. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Natick, MA.
- KRIVANEK, J., GAUTRON, P., PATTANAİK, S., AND BOUATOUCH, K. 2005. Radiance caching for efficient global illumination computation. In *IEEE Transactions of Visualization and Computer Graphics*.
- LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. P. 1993. Combining hierarchical radiosity and discontinuity meshing. *Computer Graphics* 27, Annual Conference Series, 199–208.
- RAMAMOORTHY, R., AND HANRAHAN, P. 2001. An efficient representation for irradiance environment maps. In *SIGGRAPH 2001, Computer Graphics Proceedings*, 497–500.
- RAMAMOORTHY, R., AND HANRAHAN, P. 2001. The relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. In *Journal of the Optical Society of America*.
- RATHSFELD, A. 1999. Edge asymptotics for the radiosity equation over polyhedral boundaries. *Mathematical Methods in the Applied Sciences* 22, 3, 217–241.
- RUSHMEIER, H. E., PATTERSON, C., AND VEERASAMY, A. 1993. Geometric simplification for indirect illumination calculations. In *Graphics Interface*.
- RUSHMEIER, H. E. 1988. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Ph.D. thesis.
- SHEEL, A., STAMMINGER, M., AND SEIDEL, H. 2001. Thrifty final gather for radiosity. In *Rendering Techniques 2001 (Proc. of Eurographics Workshop on Rendering 2001)*, Eurographics.
- SHEEL, A., STAMMINGER, M., AND SEIDEL, H. P. 2002. Grid based final gather for radiosity on complex scenes. In *EUROGRAPHICS 2002*, 547–555.
- SILLION, F., AND PUECH, C. 1994. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, CA.
- SLOAN, P. P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH 2002, Computer Graphics Proceedings*, 527–536.
- TABELLION, E., AND LAMORLETTE, A. 2004. An approximate global illumination system for computer generated films. In *SIGGRAPH 2004, Computer Graphics Proceedings*.
- WARD, G. J., AND HECKBERT, P. 1992. Irradiance Gradients. In *Third Eurographics Workshop on Rendering*, 85–98.
- WARD, G., RUBINSTEIN, F., AND CLEAR, R. 1988. A ray tracing solution for diffuse interreflectio. In *SIGGRAPH 1988, Computer Graphics Proceedings*.
- ZIMMERMAN, K., AND SHIRLEY, P. 1995. A Two-Pass Realistic Image Synthesis Method for Complex Scenes. In *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, 284–295.