Radiance Caching and Local Geometry Correction

Okan Arikan, David A. Forsyth, James F. O'Brien
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
{okan,daf,job}@cs.berkeley.edu

Report No. UCB/CSD-4-1318

April 2004

Computer Science Division (EECS)
University of California
Berkeley, California 94720

Radiance Caching and Local Geometry Correction

Okan Arikan, David A. Forsyth, James F. O'Brien
Computer Science Division
University of California, Berkeley
Berkeley, CA 94720
{okan,daf,job}@cs.berkeley.edu

April 2004

Abstract

We present a final gather algorithm which splits the irradiance integral into two components. One component captures the incident radiance due to distant surfaces. This incident radiance is represented as a spatially varying field of spherical harmonic coefficients. Since distant surfaces do not cause rapid changes in incident radiance, this field is smooth and slowly varying and can be computed quickly and represented efficiently.

On the other hand, nearby surfaces may create drastic changes in irradiance, because their position on the visible hemisphere change quickly. We correct the irradiance we obtain from spherical harmonics using an explicit representation of nearby geometry. By assuming nearby geometry is always visible, we can efficiently restore the high frequency detail missing from the irradiance.

Current techniques need to sample the nearby surfaces densely to approximate this rapid change of irradiance. This creates unnecessary visibility tests (or raytraces) that slow down the final gather. We demonstrate that by assuming nearby surfaces are always visible, we obtain very fast final gather results whose quality compares well with standard techniques but is computed much faster. We also demonstrate the feasibility of using nearby surfaces on scenes without global illumination to restore the high frequency shading detail due to geometric detail.

1 Introduction

Global illumination is an essential component of a realistic picture. Many algorithms exist for computing different components of indirect illumination. Unfortunately, most of these algorithms are slow.

Diffuse surfaces are particularly interesting, because they affect the indirect illumination substantially by diffusing the light that falls on them. This means the color of diffuse surfaces is essentially smoothed versions of the radiance they receive. Diffuse surfaces reflect a blurred version of the surfaces they see. If we see a blurred version of an image, we cannot detect the errors in it. Similarly, a global illumination solution

can contain errors if all we see is diffuse surfaces illuminated with this global illumination solution. Whenever we would like to render a point on a diffuse surface, we can compute the irradiance integral at that point by sampling a coarse global illumination solution over the hemisphere of the point. This practice is often called final gather and it allows us to obtain visually accurate pictures from a coarse global illumination solution.

Final gathering is most applicable to diffuse surfaces as specular surfaces admit easy solutions through raytracing. In term, diffuse surfaces are most problematic for Monte Carlo based approaches as they "diffuse" the domain of probability that needs to be integrated.

The computation involved in final gather is non-trivial. We need to estimate the irradiance by sampling the incident radiance which usually involves raytracing. As the geometrical complexity of a scene increases, each raytracing operation takes longer. This leads to the renderer spending most of its time tracing rays to evaluate the irradiance integral.

Visible geometry from a point on a diffuse surface does not change vary rapidly as the point moves on the surface. This is especially true if the diffuse surface is flat and all the visible surfaces are far away. One common practice is to sparsely compute a final gather integral and cache the results. The cached values can then be interpolated for all the points on diffuse surfaces.

Unfortunately, not all diffuse surfaces are completely flat and not all other surfaces are far away. As a point on the floor of a room gets closer to the corner, walls occupy more of its hemisphere, potentially creating a large change in diffuse color. This forces us to compute final gather samples more densely around areas of high geometric detail. For complex environments, this creates a major bottleneck.

An important observation about the human perceptual system is its insensitivity to slow changes in illumination. We are not terribly good at noticing low spatial frequency noise on diffuse surfaces. One cannot usually estimate the exact intensity or color of a point. However, we can easily tell if there are erroneous shadows. This observation leads us to find approximate diffuse global illumination solutions that push the error into low spatial (and temporal) frequencies.

In this paper, we describe an algorithm that splits the diffuse color of an object into two components: one that is a function of the distant surfaces, and a second that is a function of the nearby surfaces. Since the appearance of distant surfaces does not change rapidly, we will approximate the incident radiance from far away surfaces using a low order spherical harmonic representation. Due to the slow change in the incident radiance from such surfaces, a small number of harmonic samples is sufficient. These harmonic samples will be interpolated to obtain a continuous representation of incident radiance from distant surfaces.

The errors made during this interpolation due to the existence of nearby surfaces will then be corrected by accounting for nearby surfaces explicitly. By assuming that all the nearby surfaces are visible, we can do this correction efficiently. This visibility assumption is the main source of computational efficiency of our algorithm.

We have tested our algorithm on scenes displaying a large range of geometric complexity. Figure 1 demonstrates an image computed using our approach. We compare our method to irradiance caches in terms of speed and quality. We also compare our

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. The relevant statistics for this scene can be found in table 1 (figure name Cathedral Night).

method to a commercial renderer which implements an irradiance cache variant. Our results show that our method can generate equal quality images featuring very detailed global illumination significantly faster.

2 Related Work

Global illumination has been one of the most heavily researched branch of computer graphics. Space does not allow a detailed survey of this vast area. [18] and [5] provide a good overview of finite element and Monte Carlo based global illumination techniques.

This paper does not provide a new global illumination algorithm. Our objective is to accelerate the final gather step of rendering, which usually consumes a large portion of the rendering time. In our implementation, we used photon maps [8] to generate a coarse estimate of global illumination.

Traditionally, final gather has been used on radiosity based approaches to obtain visually pleasing results from blocky radiosity solutions. [9] introduced an object space refinement method for increasing the visual quality of the solution of a hierarchical radiosity step. [15, 16] used a coarse geometric model for a radiosity solution which

is queried by a Monte Carlo algorithm to create the final picture. [22] decreased the variance caused by Monte Carlo by re-classifying bright reflecting surfaces in a coarse radiosity solution as light sources. [17] demonstrated that a final gather could be accelerated by using the link information from a prior radiosity step to identify important senders. A similar idea has also been explored by [4] where per pixel final gather was obtained using a Monte Carlo integration which uses radiosity solution for importance sampling. These methods generate a visually pleasing result from a coarse global illumination solution. However, the final gather step is still expensive as it is usually performed for each pixel.

In an influential paper, [20] introduced the concept of performing very accurate final gather on surfaces and then interpolating. Their method adapts the sample locations where the final gather result changes quickly. A later paper [21] refined the algorithm to decrease the discontinuities due to the insertion of new samples while rendering. Although this alleviates the problem, irradiance caches are often used in a multi-pass framework.

In our paper, instead of caching irradiance, we will cache incident radiance. We will then correct the irradiance integral obtained from this cached representation by explicitly accounting for nearby geometry. This will allow us to cache the incident radiance at a small number of locations and still get the high frequency detail in indirect illumination.

The behavior of diffuse surfaces in close proximity to other surfaces (such as corners) has been explored in [14, 1]. Their results confirm the existence of high frequency change, or light reflexes around corners.

The relationship between the appearance of diffuse surfaces and incident illumination has been an important research area for computer vision. [6, 2] showed that the color of a diffuse surface was a function of the low dimensional representation of the incident illumination. [12] used low order spherical harmonic, and demonstrated harmonics to be a good and compact representation. Spherical harmonics have been successfully used in [11] to represent environment maps for diffuse surfaces. This idea has been expanded in [19] by the introduction of transfer functions to take self occlusions or interreflections into account. However such methods require expensive precomputation for obtaining the transfer functions.

3 Overview

For global illumination, we assume that the geometry of our scene is represented with a set of triangles. We do not require any connectivity or manifold properties. As most geometries can be converted into polygons, we do not lose much generality. Note that the scene can still contain non-polygonal surfaces for rendering. However, for global illumination computations, we will tessellate such surfaces.

As a preprocessing step, we create a coarse approximation to the global illumination in the scene using a method like photon mapping [8]. Since all we need is to be able to evaluate the global illumination, any finite element based approach would also work.

We represent the contribution of distant surfaces to the indirect illumination using

a field of spherical harmonic coefficients. We can approximate radiance arriving at each point by evaluating this field. As noticed by other researchers [13], the incident illumination field can be captured with a low order harmonic basis. Since a small number of basis functions are sufficient to compute the radiance integral, this incident illumination field can be low dimensional.

If we assume the indirect illumination is only due to far away surfaces and use an interpolant obtained from these spherical harmonic samples, we will get an overly smooth estimate of indirect illumination. We correct this estimate by explicitly accounting for the nearby triangles and their effects on the indirect illumination for the point of interest.

In order to construct an image, we need to compute the visible radiance of a set of *shading points*. For example, if we are raytracing, the shading points would be the points where rays intersect surfaces (excluding shadow rays). Each shading point has a position, P and a unit normal vector N. The diffuse brightness (or irradiance) of a shading point is defined as:

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

Where the ρ/π is the ideal diffuse Bidirectional Reflectance Distribution Function (BRDF), $N \cdot \omega$ is the cosine term that accounts for the foreshortening on the radiance coming from direction ω and $L(P, \omega)$ is the incident radiance at P from direction ω. Depending on the surface BRDF, a shading point may have specular components as well. Such components will be evaluated by raytracing.

In general, $L(P, \omega)$ contains radiance from light sources (direct illumination) as well as other surfaces (indirect illumination). In our framework, we handle the direct illumination from light sources using conventional techniques such as shadow mapping or raytracing. Direct illumination can create fast changes in diffuse brightness because of shadowing effects and such effects are best dealt with using conventional sampling techniques. Thus in equation 1, we assume $L(P, \omega)$ contains only radiance reflected from other surfaces. For final gathering purposes, this term comes from the coarse global illumination solution (radiosity or photon map).

As we move on a surface, visible points on our hemisphere swing: those points that are near swing more quickly than distant points (figure 2). In other words, the portion of $L(P, \omega)$ from distant surfaces changes more slowly than the portion due to nearby surfaces. This distinction is important because it motivates us to find different approximation methods for different sources of incident radiance.

If the incident radiance changes slowly as we move on a surface, we can compute the diffuse brightness at a coarse set of sample points and interpolate the diffuse brightness for all other query locations. This is the basic idea behind irradiance caches [20].

However, the nearby visible surfaces can move significantly on our hemisphere as we move. This necessitates finer sampling for irradiance caches. If we have substantial geometric detail, most of the points we compute will have nearby surfaces. The surface detail attracts samples and increases the computation time. However, putting lots of samples where the surfaces come close to each other is largely unnecessary: this is

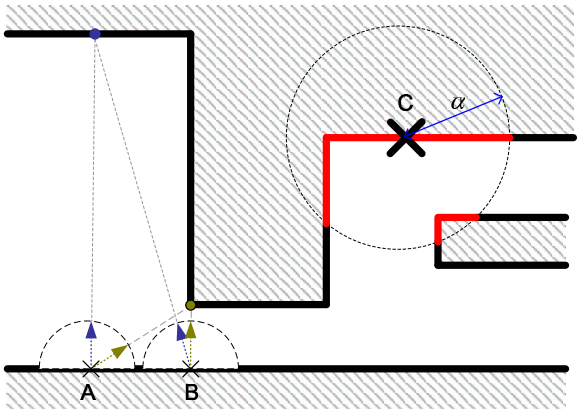


Figure 2: Nearby surfaces move more quickly on the visible hemisphere of a point. In this figure, as we move from B to A, the amount the distant point (blue) swings on the hemisphere is very small whereas the nearby point (green) swings substantially. The circle around the shading point C with radius α denotes the volume we consider nearby. All surfaces inside this volume are considered nearby (indicated as red).

a local phenomenon. The main contributor to the computational complexity is the visibility tests that are required to find the surfaces that a sample sees. Since nearby surfaces are mostly visible, the visibility computations are mostly wasted.

We define nearby surfaces to a shading point P to be those that are in the sphere whose center is P and whose radius is α (figure 2). This is a user chosen parameter whose significance is discussed in section 6.

All points see some portion of the scene on their hemispheres. For a particular shading point P, with its hemisphere Ω, some portion of the surfaces on Ω may be nearby and some portion may be far away. Let the portion of the hemisphere where the visible surfaces are farther away than α be Ω_F and the portion that is closer be Ω_N (see figure 2). Note that $\Omega_F \cup \Omega_N = \Omega$ and $\Omega_F \cap \Omega_N = \emptyset$. We define $L_N(P, \omega)$ and $L_F(P, \omega)$ to be the incident radiance that comes through the respective portions of the

hemisphere.

$$\begin{aligned} B &= \frac{\rho}{\pi}\int_{\Omega} N\cdot\omega L(P,\omega)d\omega \\ &= \frac{\rho}{\pi}\int_{\Omega_N} N\cdot\omega L_N(P,\omega)d\omega \\ &\quad +\frac{\rho}{\pi}\int_{\Omega_F} N\cdot\omega L_F(P,\omega)d\omega \\ &= \frac{\rho}{\pi}\int_{\Omega} N\cdot\omega L_F(P,\omega)d\omega \\ &\quad +\frac{\rho}{\pi}\int_{\Omega_N} N\cdot\omega[L_N(P,\omega)-L_F(P,\omega)]d\omega \\ &= \frac{\rho}{\pi}(B_D+B_C) \end{aligned} \tag{2}$$

In this equation, B_D is the diffuse brightness that is due to distant surfaces and B_C is a correction term which accounts for energy transfer from the nearby surfaces. Notice that the integral over the far field illumination has been expanded onto the entire hemisphere and the expanded amount has been subtracted from the near field illumination. Section 4 will explain explain how B_D can be computed and section 5 will explain how B_C is computed.

4 Far Field Illumination

We need to compute B_D for all the shading points. Since B_D does not change rapidly, we could sample it on surfaces and interpolate this value for the shading points [20]. However, the reason B_D does not change quickly is L_F does not change quickly. This is because far away surfaces move slowly on the hemisphere. Thus, we will sample L_F and interpolate this sampled representation. Since L_F is needed to compute B_C as well as B_D, this is advantageous.

As explained by [12] [13], the integral in equation 1 is determined mostly by the low frequency components of the incident radiance. For diffuse surfaces, this means a smooth representation of L_F is adequate. A spherical harmonic representation is ideal for our purposes. As demonstrated by [11], a small number of spherical harmonic coefficients is good enough for representing the incident radiance on diffuse surfaces. Because $\frac{\rho}{\pi}N\cdot\omega$ and $L_F(P,\omega)$ are functions defined on the unit sphere, we can write the integral as a dot product of the harmonic coefficients:

$$\begin{aligned} B_D &= \int_{\Omega}(N\cdot\omega)L_F(P,\omega)d\omega \\ &\approx \overline{max(N\cdot\omega,0)}\cdot\overline{L_F(P,\omega)} \end{aligned} \tag{3}$$

Where $\overline{f(\omega)}$ is the spherical harmonic representation for a function f defined over the directions on unit sphere.

This nice property allows us to compute and store $\overline{L_F}$ sparsely in space. Given a shading point, we can get a representation for its incident radiance due to far away surfaces by interpolating the spherical harmonic coefficients. $\overline{max(N \cdot \omega, 0)}$ can be computed analytically as a function of N.

This representation is similar to irradiance caching where we cache the result of the irradiance integral and interpolate whereas here we cache spherical harmonic representation of the incident radiance and integrate at each shading point by doing a dot product. As demonstrated in [13], 9 harmonic coefficients provide a sufficient representation of incident radiance for diffuse surfaces. This makes the integral in equation 3 a dot product of 9 dimensional vectors per color channel. The L_F itself then is a field of 9 spherical harmonic coefficients (again per color channel).

The important difference between our method and irradiance caches is that this portion of the integral is only designed to capture the low frequency component of the diffuse brightness. Therefore only few spherical harmonic samples are required (see figure 3).

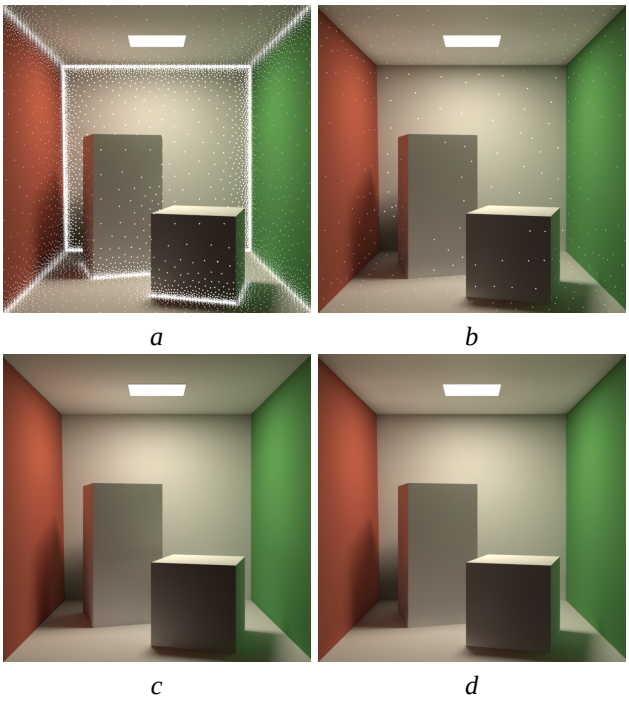
4.1 Computing Spherical Harmonic Field

We want to have a field of spherical harmonic coefficients that represent the incident radiance due to distant surfaces L_F. We can do this by computing the spherical harmonic representation of L_F at particular points and interpolating the coefficients. However, this method is prone to errors. A point at which we compute L_F may be very close to another surface. So the rays that we shoot intersect mostly the nearby geometry. Because the effect of this nearby geometry on the diffuse brightness changes rapidly, the estimate we obtain is only valid for surfaces that are close to the sampled point. Since we would like to compute the spherical harmonics sparsely and expect that these spherical harmonic will be valid for many shading points, these samples should be a spatial average of the incident radiance field.

Let us assume that we know in advance all the shading points that will require diffuse brightness computation. We need a spherical harmonic representation for the incident radiance from distant surfaces for this set of points. We compute the spherical harmonic representation by tracing rays and gathering radiance from the photon map at the locations that they hit. These rays originate from a random subset of the shading points and their direction is randomly chosen by sampling the selected shading points' hemispheres. For n radiance samples, let R_i and D_i where $1 \leq i \leq n$, be the incident radiance of the ray and the direction of the ray respectively. We obtain the spherical harmonic coefficients by solving the following set of linear equations using least squares projection:

$$C(D_i)S = R_i, 1 \leq i \leq n \tag{4}$$

Where $C(D)$ is the vector containing the value of spherical harmonic basis functions evaluated at the unit vector D, and S is the vector of spherical harmonic coefficients. Since we only use the first 9 spherical harmonic basis, S is a vector of 9 numbers for each color channel. Solving this equation is linear in the number of radiance samples and can be done quickly. This construction is output sensitive: since we collect



a *b*

c *d*

Figure 3: *a* 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. *b* shows the locations of our spherical harmonic samples. Since the high frequency change in diffuse brightness around corners are captured explicitly by our correction term B_C, we require far fewer samples. The image *d* shows the sample points removed to show the final image computed using our technique and the image *c* shows the ground truth which is obtained by estimating the diffuse brightness integral (equation 1) using Monte Carlo techniques, which involves evaluating the irradiance integral in equation 1 by gathering from coarse global illumination solution by sampling.

spherical harmonic samples only for the shading points, we do not waste computation on invisible portions of the scene.

If the radius of the bounding sphere of the set of shading points is larger than α and any of these rays hit a piece of surface within this sphere, then we cannot reliably estimate the far field illumination on these points. This is because there is nearby geometry on the hemisphere of the shading points. If this is the case, we split the shading points into 2 groups using k-means clustering and perform the same computations for the children clusters. This method of lazy splitting gives us adaptive sampling on large flat areas.

We stop splitting if the bounding sphere of the shading points has a radius less than α. The choice of this threshold is important as it determines how accurate the spherical

harmonic sample will be. If this threshold is zero, we will collect more samples around nearby surfaces which is what we want to avoid.

This method gives us groups of shading points, each having a spherical harmonic sample representing the incident radiance. We compute the bounding sphere of each group and place the corresponding spherical harmonic sample at the centroid of the sphere with the corresponding radius. Given a new shading point, we compute the spherical harmonic coefficients by radial basis function approximation using these samples.

The method described above requires prior knowledge of what shading points will require diffuse brightness computation. We obtain these shading points by performing a prior pass. Such multi-pass rendering is not new to the irradiance caching community: since insertion of new samples create discontinuities in diffuse brightness, one usually performs an irradiance cache construction pass where all global illumination samples are collected, and a final rendering pass where they are interpolated smoothly.

5 Nearby Geometry Correction

Let T be the set of triangles whose centroids are closer to the query point P than α. Each triangle $t_i \in T$ has an average radiosity which is looked up from the coarse global illumination solution (in our case, from a global photon map). In general, we expect the triangles in T to give us a decent but not very accurate approximation to the nearby geometry and its radiosity. We do not need the nearby geometry to be accurate, because it is difficult to obtain and our experience shows the correction term is not sensitive to the accuracy by which we represent nearby surfaces.

Let N_i and R_i be the normal vector and constant radiosity value across the triangle t_i, then we can re-write the diffuse brightness correction term as a sum of surface integrals over the triangles in T:

$$\begin{aligned} B_C &= \int_{\Omega_N} (N \cdot \omega)[L_N(P,\omega) - L_F(P,\omega)]d\omega \\ &= \sum_{t_i \in T} \int_{t_i} \frac{1}{r^2} cos(\theta_1) cos(\theta_2) H(P_t)[R_i - L_F(P, D_t)]dA \end{aligned} \tag{5}$$

In this equation,P_t is the point on the triangle being integrated, D_t is the unit vector from P to P_t, $H(P_t)$ is the visibility term that evaluates to 1 if P is visible from P_t (and 0 otherwise) and θ_1, θ_2 are the angles between D_t and the normal of the shading point and the normal of the triangle. This is very similar to the form factor integral. Although analytically this equation is easy to write, the presence of H makes it very difficult to evaluate.

In order to compute this correction term, we are going to assume all nearby triangles are visible to the shading point. This is generally a valid assumption: nearby geometry is usually visible. This is also a statement about the intrinsic property of surfaces. If a surface has bounded variance property, given a small enough neighborhood around a point on a surface, all points in this neighborhood that are not backfacing are always visible.

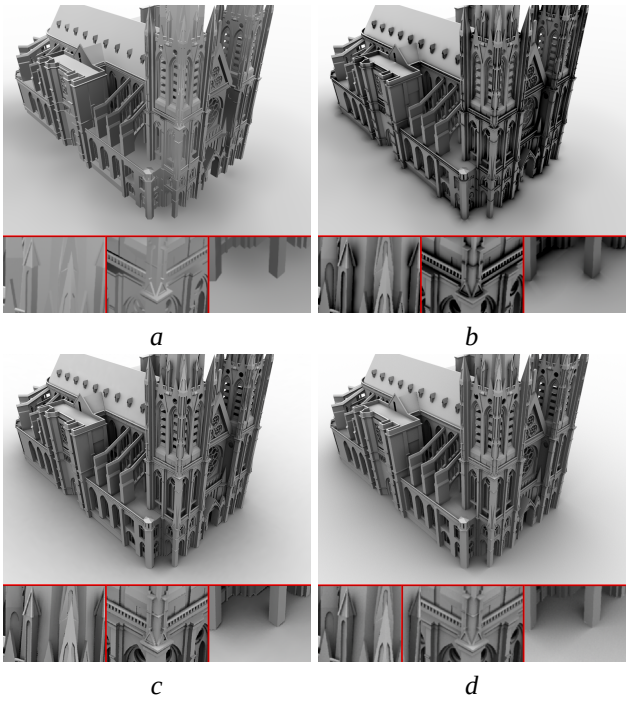


Figure 4: This figure demonstrates the failure cases of our algorithm. This scene does not contain any global illumination. The incident brightness for a point P from direction D ($L(P,\omega)$) for this figure is defined as 0 where the ray from P along ω is occluded by another surface and 1 otherwise. This creates an overcast sky-like illumination. Low order spherical harmonics cannot represent such an irradiance field very accurately. Notice that this definition of incident radiance is not realistic and in practice the incident radiance reflected from other surfaces are much smoother. *a* shows a cathedral model without B_C term applied. The high frequency brightness detail is missing from the image. *b* shows the $B_D + B_C$. *c* is the same image rendered with PrMan. Generating this figure takes about 2 hours while generating our image (*b*) takes less than 5 minutes. *d* is the ground truth obtained by evaluating the irradiance integral by sampling, with the $L(P,\omega)$ defined as above. The thumbnails under the images show a bigger version of the same three locations in all figures. The thumbnails under *a* show the lack of detail in diffuse brightness. The detail is restored in the thumbnails below *b*. Notice that irradiance caches can also have problems representing the sharp diffuse brightness occurring near the base of the pillar in the rightmost thumbnail (under *c*).

This visibility assumption is where we get most of our speedup. Standard irradiance cache based approaches spend a lot of time casting rays to evaluate the visibility of nearby surfaces which are mostly visible. By taking advantage of this fact, such surfaces can be accounted for directly without employing visibility queries.

We can make the integral in equation 5 more manageable, if we also assume L_F

is constant on our hemisphere and is equal to B_D[1]. Since L_F is represented as a low order spherical harmonic, it is mostly smooth and does not feature drastic changes in function values. This is because L_F does not contain any direct illumination which may be very bright. This assumption essentially replaces L_F with its average value. For smooth, non-varying functions, this does not create significant errors.

In order to compute the diffuse brightness correction term, we need to have the far field diffuse brightness term B_D. Thus for a given shading point, we first interpolate the spherical harmonics and obtain B_D by dot product. We then compute the following correction term which we can re-write using the previous two assumptions:

$$B_C \approx \sum_{t_i \in T} [R_i - B_D] \int_{t_i} \frac{1}{r^2} cos(\theta_1) cos(\theta_2) dA \tag{6}$$

Now the integral computes the form factor between a triangle and a point for which analytical solutions exist [7]. In fact, as demonstrated in [3], if the query point is sufficiently far away, relative to the triangle, one point quadrature gives a good estimate:

$$\int_{t_i} \frac{1}{r^2} cos(\theta_1) cos(\theta_2) dA \approx \frac{cos(\theta_1) cos(\theta_2) A_i}{r^2} \tag{7}$$

Since some triangles may be very large, or may have a bad aspect ratio or incur significant radiosity change across them, we split the scene triangles until they are smaller than α. Only those triangles that are near the shading points need to be split. This allows us to split these triangles lazily. Since α is not small, we do not create lots of tiny triangles.

In order to find the set of triangles that are near a shading point, we use a bounding volume hierarchy. Section 7 explains how the nearby triangles can be summed without being overcome by the scene complexity.

6 Near-Far Threshold

If one looks at the tire of a car, there are many surface grooves which may create interesting shadowing and interreflection effects which we may care about. In order to capture such effects, the minimum size threshold (α) should be on the order of the size of a groove. If we're looking at the car as whole, then we might not care about the interreflection effects inside tire grooves, because the tire itself may only occupy a few pixels. But we might be interested in capturing the car looming over the tire's hemisphere. In this case our size parameter should be on the order of the tire. If we're looking at the car from far away, the only global illumination effect we're interested in accurately capturing might be the interreflections between the car and the nearby buildings, which means the size threshold should be on the order of the car.

With this argument, we do not mean small scale phenomena such as grooves on the tire of a car are unimportant if we are rendering a car. We simply mean that such effects are too costly to capture accurately by adaptively sampling the diffuse brightness or incident radiance. The α parameter serves as an accuracy versus speed knob as well. If

[1] It is our experience that the integral in equation 5 can be computed without making this assumption using Monte Carlo techniques. However, this creates very minor visual differences.

α is set to zero, spherical harmonic samples will get denser around high surface detail, just like irradiance caches (see figure 3).

Diffuse brightness that change quicker than this threshold (α) will not be captured accurately using the spherical harmonic samples. This is because average spacing between the spherical harmonic samples is α. Such high frequency effects usually result from nearby geometric detail. Thus we handle such detail with our correction term: B_C.

7 Dealing with Complexity

A scene we would like to render may contain many triangles. Our preprocessing algorithm may split some of these triangles if they are too big as well. It is imperative that we can perform the summation in equation 6 efficiently. We can do this by grouping the triangles hierarchically, and computing the correction factor B_C from important nodes, which represent groups of triangles.

In our implementation, we create a bounding volume hierarchy of all the triangles. Each node of the hierarchy stores an area weighted sum of radiosity and normal vector and the sum of the areas of all the triangles below it. Given a new shading point P, we would like to add over the nearby triangles, their correction factors given in equation 6. That means we are interested in the nodes of our bounding volume hierarchy that intersect the sphere of interest whose center is P and radius is α.

Given a shading point P, there may be many triangles (leaves of our hierarchy) that are closer than α. Finding and summing over these triangles may be a major bottleneck. However, as mentioned in the previous section, one point quadrature can be used to compute the contribution of an individual triangle to the near diffuse illumination term very efficiently. This also works for a collection of triangles.

We can find all such triangles by recursively descending this hierarchy. If a node's bounding sphere does not intersect the sphere of interest, we skip it. Otherwise, if the shading point is sufficiently far away from the node, we account for all the triangles below the node using the average radiosity and normal vector and the total area of all the triangles below it. [3] demonstrate that for a polygon to point form factor, if the shading point is more than 4 times the size of the polygon away, the error due to the quadrature is less than 10% and we obtain a good approximation if the shading point is farther than 4 times the radius of the bounding sphere of the node. Otherwise, we traverse the children of the node.

This method makes the summation is equation 6 adaptive: only the triangles in the immediate vicinity of the shading point will be summed individually. Other triangles that are farther but still inside the sphere of interest will be summed as clusters of triangles.

8 Results

When evaluating a rendering algorithm, it is important to define the phenomena that we would like capture. In this paper, we described an efficient final gather algorithm

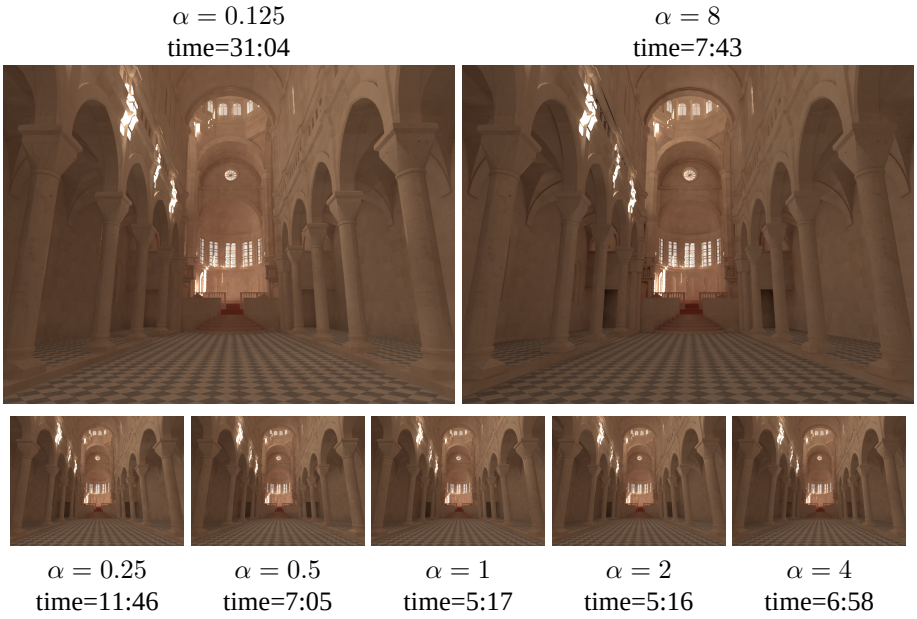


Figure 5: This figure demonstrates the effect of changing α on the rendering time and quality. Notice that as α gets larger, the scene gets slightly darker. This is due to the underestimation of the energy coming from far away surfaces, because the visibility assumption is broken. Bigger α also creates a more uniform spherical harmonic field. This effect can be observed on the dome above the altar area: the illumination distribution becomes more uniform. The α value changes by a factor of 64 in these images without unpredictable errors.

that preserves the high frequency detail that we usually see around high geometric detail. Our algorithm achieves this by splitting the diffuse illumination into two parts and approximating each part separately.

Figure 4 demonstrates our failure cases. For this particular example, we omitted the global illumination and assumed that $L(P, \omega) = 1$ if the ray originating from P going in the direction ω does not hit any geometry and 0 otherwise. This is commonly known as the ambient occlusion illumination. Since this definition of incident radiance on most surfaces contains significant discontinuities (it is a binary function), spherical harmonic representation of $L(P, \omega)$ is not very accurate. This inaccuracy creates a low frequency noise. Our result using radiance caching in the center-left is slightly darker because of this inaccuracy. Another reason for the darkness is the visibility assumption. This particular model contains about 200,000 triangles which creates very high geometric detail. If α is set too large, then B_C term will contain contributions from triangles that in reality occlude each other.

However, this figure also shows how we are able to recover high frequency detail in complex environments. *a* shows the value of B_D. Notice that the diffuse brightness is very uniform and the effect of surface relief is missed. *b* shows that our B_C step

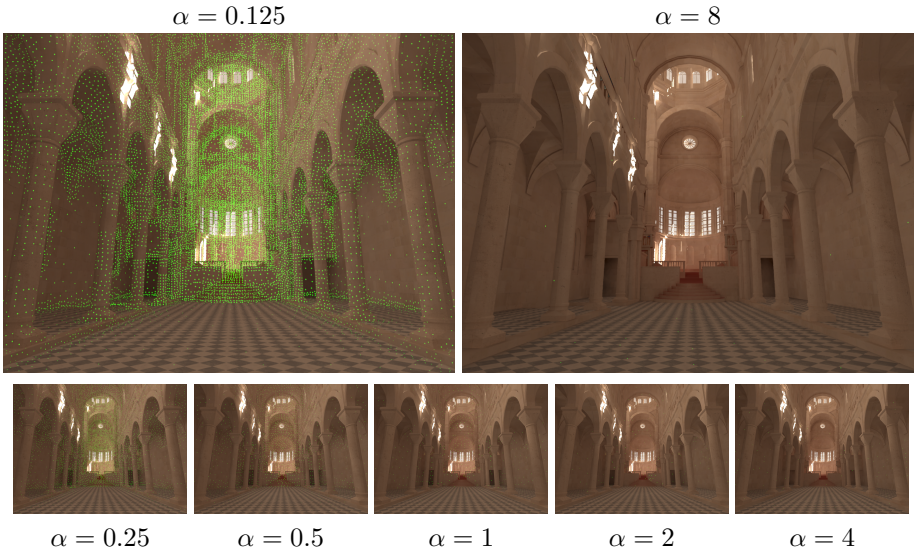


Figure 6: In this figure, we show the spherical harmonic sample locations for the same cathedral pictures in figure 5. The spacing between the harmonic samples is equal to α. As expected, decreasing α results in denser harmonics sampling around high surface detail and increases the running time. Notice that we do not have to split flat clusters in section 4.1. This means the α will actually be bigger around flat pieces of surface. So, the global α value is actually a minimum spacing between the harmonics samples.

restores the detail. *c* shows the same scene rendered through a commercial renderer that implements an irradiance cache variant. *d* is the ground truth obtained by evaluating the irradiance integral using brute-force sampling at each shading point. The bottom portions of the images show closeups of some portions of the corresponding image.

Ambient occlusion is not a physically accurate representation of incident radiance and is chosen to emphasize the failure cases. In reality, the indirect illumination is considerably smoother and can be reliably represented as low order spherical harmonic.

Figure 5 demonstrates the effect of α on the running time and the quality. As α gets larger, the visibility assumption gets violated. This in turn creates under/over estimation of the energy transfer between nearby surfaces, creating a darker appearance for this particular cathedral scene. Notice that as α gets too big, the computation time also gets larger as the form factor computation starts dominating. Figure 6 show the spherical harmonic sample locations for the same cathedral images. Notice that the spacing between the harmonics samples is the α value.

Quantitative error analysis is difficult due to the visibility term in the rendering equation. However, a qualitative analysis is possible. There are two main sources of error in our algorithm:

1. **The visibility assumption** may break if α is too big. This in turn creates an over/under estimation of the energy transfer and hence creates a darkening (or

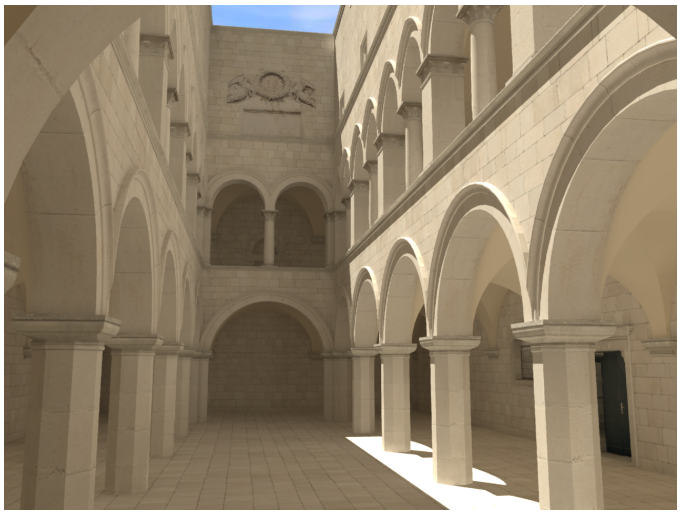


Figure 7: This image demonstrates an atrium illuminated with a single distant light source. Notice how the light propagates under the arches. Our nearby geometry correction term B_C corrects the errors due to the smooth incident radiance approximation, around high geometric detail and create realistic light reflexes. For example, the top of the arch at the far end gets darker around corners, and the pillars receive bright illumination from the floor which is illuminated by the sun. The table 1 contain rendering statistics for this scene (figure name Atrium Sun). Notice that the total rendering time for this figure is about 5 minutes.

lightening) effect on the picture. However, as we demonstrate in figure 5, α needs to change significantly for this effect to become observable (α is increased 64 times between the top-left and the top-right images). This means the visibility error does not change rapidly as a function of α.

2. **The spherical harmonic interpolation** is prone to excessive smoothing if α is too big. If the spatial support of a harmonic sample is too big, we will not be able to capture spatial change in illumination in a particular direction. This can be observed in figure 5 again. The dome above the altar area is brighter on the right hand side for $\alpha = 0.125$. As α gets bigger, the dome area picks up a constant illumination. This is due to the fact that radiance samples over a large surface are being used to create a single harmonics to capture the entire area.

Notice that if a cluster of shading points is flat or does not have nearby occluders, then the cluster is not split further. We can then sample a harmonics for this cluster and use the cluster radius as the α value for that cluster. This allows adaptive adjustment of nearby / far away threshold. This effect can be observed in figure 6 where on flat pieces of surface, the spacing between the harmonics samples (α) is large.

We compare our algorithm to irradiance cache based approaches. Such methods are readily incorporated into many renderers and are widely used. We have implemented

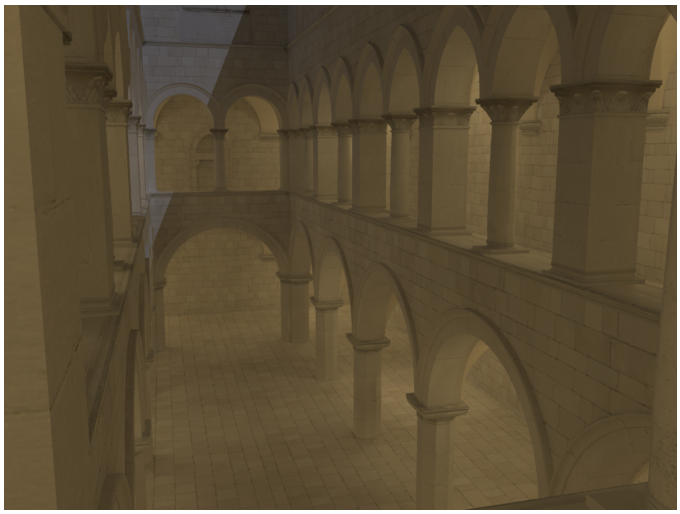


Figure 8: This image shows a nighttime view of the same atrium model used in figure 7. The scene is illuminated by one distant light source (the blue, moonlight) and 16 point light sources long the corridors. The computational cost of adding the global illumination for this scene was less than the time it takes to compute it with direct lighting only (see table 1, figure name Atrium Night).

our algorithm on an open source renderer that already has a very efficient irradiance cache implementation. The source code for our method is also available online along with this renderer. In addition to comparing our algorithm to this particular implementation of irradiance caches, we compare our timings to a highly optimized industrial renderer, Photo Realistic RenderMan [2] (PrMan) v. 11.5.2, which implements an irradiance cache variant.

We identified the following important statistics that indicate the performance:

1. **Absolute Rendering Time (ART)**: This is the total time that is required to compute the image. It includes any global illumination or final gather preprocessing.

2. **Final gather overhead (FGO)**: This number is defined as the ratio of ART over the time it takes to generate the image without any global illumination. It mainly indicates how much overhead the global illumination introduces in terms of time. If this number is less than 2, than the global illumination computations are taking less than the amount it takes to render the image using direct illumination only. Considering most applied photorealistic image synthesis uses direct illumination only, attaining 2 is a good objective.

3. **Number of Rays Traced (NRT)**: This number is the number of rays traced for the global illumination computation. Since each raytracing operation is asymp-

[2]The RenderMan (R) Interface Procedures and Protocol are: Copyright 1988, 1989, Pixar

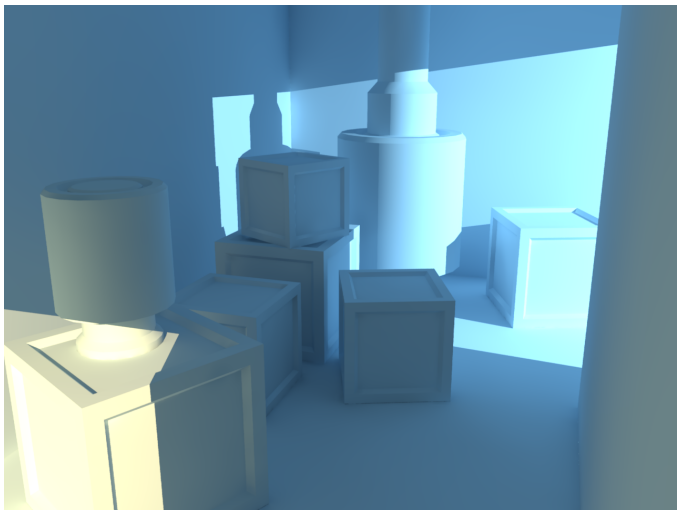


Figure 9: This scene is illuminated with two spotlights. The high frequency indirect illumination can be observed around the far corner near the sharp shadow of the light source and between two boxes stacked on top of each other. The relevant statistic about this scene can be found in table 1 (figure name Toybox).

totically $O(log(n))$ in the number of raytraceable objects (such as triangles), this is an important number to minimize.

4. **Number of Final Gather Samples (NFGS)**: This is the total number of global illumination samples we compute. Each global illumination sample involves sampling the incident radiance by raytracing. Thus, the number of raytraces we perform is influenced by this number.

5. **Cache Size (CS)**: This number measures how much memory the cache consumes.

Our rendering pipeline involves 3 separate passes. In the first pass we compute the coarse global illumination in the scene. For this purpose we use photon mapping as it is easy to implement and gives a coarse answer efficiently. The second pass involves collecting the diffuse illumination samples. For irradiance caching, these are irradiance samples, for our algorithm these are spherical harmonic samples. In this pass we also split the triangles that will effect the global illumination if necessary and create their bounding volume hierarchy. Finally, the third pass involves computing the image with the diffuse illumination computed from the cache.

All our scenes feature only diffuse objects. This is due to the fact that specularities can be easily handled using regular raytracing and tend to draw attention away from global illumination effects that we would like to observe.

Table 1 summarize our rendering times. The timings are collected on a dual Athlon 2200+ with 2 GB of memory. Only one rendering thread is used for all figures.

		ART (mm:ss)	FGO	NRT (M)	NFGS	CS (MB)
Figure 7	RC	5:47	2.22	1.79	1722	73
	IC	51:08	19.67	104.98	102522	5
	PrMan	74:25	13.70	248.12	435542	47
Figure 8	RC	16:09	1.50	0.87	1664	73
	IC	52:40	4.90	87.41	170727	9
	PrMan	68:35	7.00	67.81	392783	46
Figure 9	RC	5:07	1.86	1.81	3326	33
	IC	37:07	13.50	122.22	119360	6
	PrMan	25:08	18.85	96.07	162639	120
Figure 10	RC	4:34	2.53	1.03	1960	55
	IC	59:30	33.05	109.51	106947	5
Figure 1	RC	9:50	1.56	1.48	1444	47
	IC	68:30	10.84	109.51	101630	5

Table 1: The RC row shows the rendering statistics for our radiance caching method. The IC row contains the corresponding statistics for irradiance caches. The PrMan row shows the same statistics for the commercial renderer we used.

Figures 8 , 7, 1 , 10 , 9 and 11 demonstrate the quality of our results. The figure 10 also compares the result of our algorithm to irradiance caches.

In general, our examples demonstrate that we are able to generate high quality images much faster. FGO for our radiance caching algorithm is less than or close to 2 for all examples (see table 1). This means the overhead of computing the global illumination and performing the final gather is about as expensive as rendering the scene with direct illumination. This makes radiance caching and local geometry correction a very practical algorithm especially if we consider most of the professional computer animations use only direct lighting and pseudo light sources for fake global illumination.

The radiance caching needs to collect fewer spherical harmonic samples (NGS). This in turn means we trace fewer rays (NRT), allowing our method increased scalability.

Our method consumes more memory (CS). This is because each spherical harmonic samples requires 9 coefficients per color channel. We need to store 27 numbers with each sample as opposed to 3 for irradiance caches. Another cause of the increased size is the size of the triangles that we use to compute the correction term B_C. The size of our cache does not vary too much as a function of the scene complexity (Figure 10 contains orders of magnitudes more polygons than 9 and yet consumes about twice as much memory. See table 1). Thus we do not expect the cache size to become an important issue.

Unfortunately, the limited dynamic range of the printed medium prohibits us from showing high dynamic range images directly. Any non-linear post processing introduces bias to the errors made by a rendering algorithm. For example, errors in the dark are more emphasized whereas errors in the light regions are suppressed. In order to introduce the least amount of bias while achieving a displayable result, our figures are

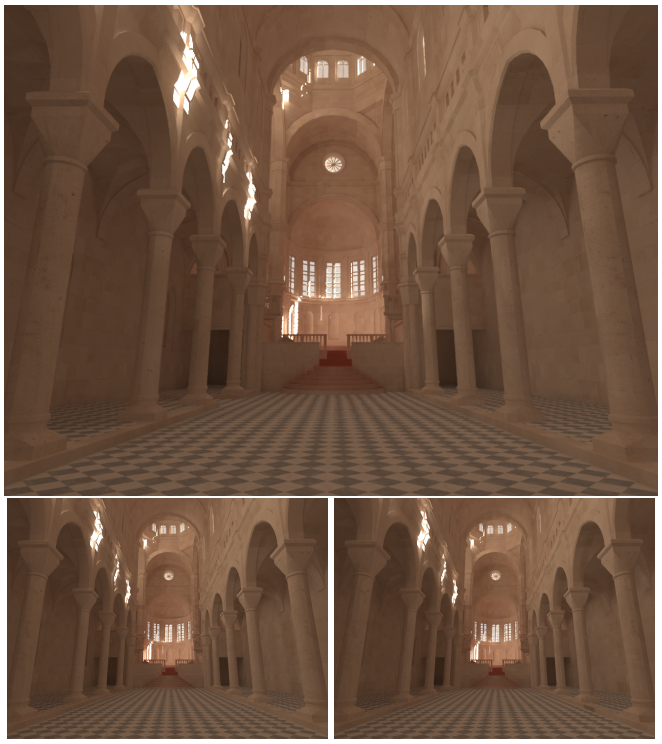


Figure 10: This scene demonstrates the interior of a cathedral illuminated by a single distant light source. The large windows at the far end of the cathedral admit more light and create a brighter altar area. Notice also that the high frequency indirect illumination on the arch supporting the dome. The top figure is computed using our method and the one on the bottom-right demonstrates the irradiance cache output. The bottom-left figure is the ground truth computed using Monte-Carlo techniques. The relevant statistics about this scene can be found in table 1 (figure name Cathedral Sun).

only gamma corrected.

8.1 Fake Global Illumination Applications

Traditionally, graphics artists are used to approximating the global illumination using many different types of conventional light sources such as point lights and ambient lights. The biggest problem of such approaches is the lack of high frequency detail caused by the nearby surfaces such as shadowing or reflecting light onto each other. An interesting feature of our method is it can be used to add illumination detail to such environments. We can set B_D to the diffuse brightness computed from the fake light sources. We will also set the radiosities of all triangles participating in the correction term to 0. The resulting diffuse brightness computed by $B_D + B_C$, will, for each shading point, subtract the portion on the hemisphere that is occupied by nearby trian-

Figure 11: This figure shows the same scene from the same camera as in figure 10 with a different light position. Due to the change in the light direction, the illumination pattern in the scene is drastically different. Most of the lighting in the scene is indirect. The rendering time for this scene is similar to that of 10 and is under 5 minutes.

gles. This creates an appearance similar to accessibility shading ([10]). Notice that the method does not use any global illumination or visibility queries. The images in figure 12 were computed using this method, in 10-15 seconds.

9 Acknowledgements

We would like to thank Pixar and Wayne Wooden for providing us with their renderer and helping us with rendering.

References

[1] K. E. Atkinson. The planar radiosity equation and its numerical solution. *IMA Journal of Numerical Analysis*, 20:303–332, 2000.

[2] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. Technical Report MCS00-21, 2000-172R, Waizmann Instritude of Science, NEC Research Institude.

[3] D. R. Baum, H. E. Rushmeire, and J. M. Winget. Improving radiosity solutions through the use of analytically determined form-factors. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 325–334. ACM Press, 1989.

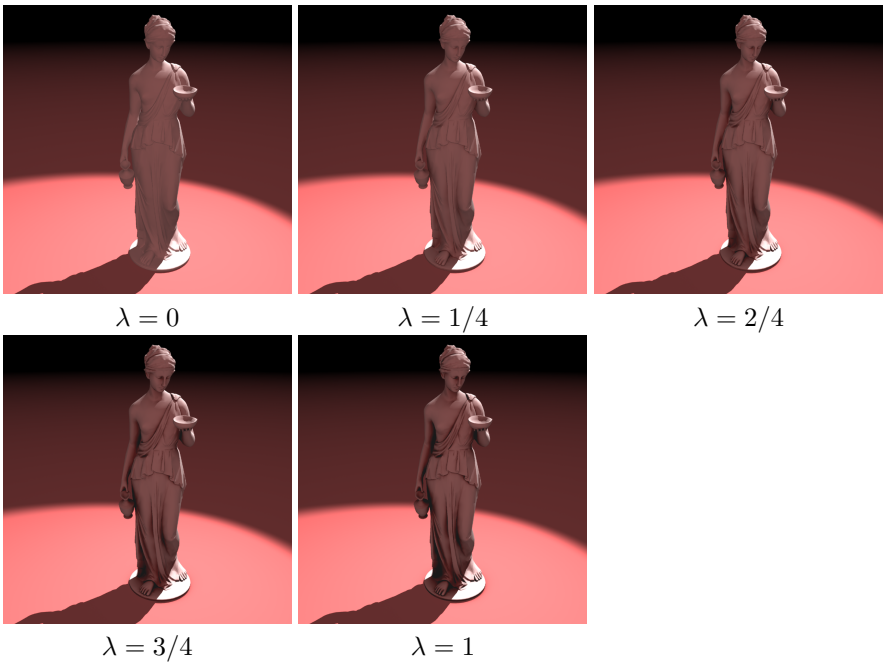


Figure 12: This figure shows that our algorithm can be used without global illumination to add illumination detail due to geometric detail. The scene is illuminated with a spotlight, a bounce light (to fake the global illumination) and an ambient light. The bounce light and the ambient light are used to compute B_D (bounce lights and ambient light vary slowly). The figures from left to right demonstrate the diffuse brightness computed as $B_D + \lambda \times B_C$ for different values of λ. Because B_D is faked, no spherical harmonic samples are collected. The correction factor emphasizes the geometric detail and produces more realistic appearance.

[4] P. Bekaert, P. Dutre, and Y. D. Willems. Final radiosity gather step using a monte carlo technique with optimal importance sampling. Technical Report CW275, 1998.

[5] P. Dutré, P. Bekaert, and K. Bala. *Advanced Global Illumination*. A. K. Peters Ltd., 2003.

[6] R. Epstein, P. W. Hallinan, and A. L. Yuille. 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE workshop on physics-based modeling in computer vision*, pages 108–116, 1995.

[7] H. C. Hottel and A. F. Saforim. *Radiative Transfer*. McGraw Inc., 1967.

[8] H. W. Jensen. *Realistic Image Synthesis Using Photon Mapping*. A. K. Peters, Natick, MA, 2001.

[9] D. Lischinski, F. Tampieri, and D. P. Greenberg. Combining hierarchical radiosity and discontinuity meshing. *Computer Graphics*, 27(Annual Conference Series):199–208, 1993.

[10] G. Miller. Efficient algorithms for local and global accessibility shading. In *International Conference on Computer Graphics and Interactive Techniques*, pages 319–326, 1994.

[11] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 497–500, 2001.

[12] R. Ramamoorthi and P. Hanrahan. The relationship between radiance and irradiance: Determining the illumination from images of a convex lambertian object. In *Journal of the Optical Society of America*, 2001.

[13] R. Ramamoorthi and P. Hanrahan. A signal-processing framework for inverse rendering. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 117–128. ACM Press / ACM SIGGRAPH, 2001.

[14] A. Rathsfeld. Edge asymptotics for the radiosity equation over polyhedral boundaries. *Mathematical Methods in the Applied Sciences*, 22(3):217–241, 1999.

[15] H. E. Rushmeier. *Realistic Image Synthesis for Scenes with Radiatively Participating Media*. Ph.D. thesis, 1988.

[16] H. E. Rushmeier, C. Patterson, and A. Veerasamy. Geometric simplification for indirect illumination calculations. In *Graphics Interface*, May 1993.

[17] A. Scheel, M. Stamminger, and H. Seidel. Thrifty final gather for radiosity. In *Rendering Techniques 2001 (Proc. of Eurographics Workshop on Rendering 2001)*. Eurographics, June 2001.

[18] F. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann, San Francisco, CA, 1994.

[19] P. P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *SIGGRAPH 2002, Computer Graphics Proceedings*, pages 527–536, 2002.

[20] G. Ward, F. Rubinstein, and R. Clear. A ray tracing solution for diffuse interreflectio. In *SIGGRAPH 1988, Computer Graphics Proceedings*, 1988.

[21] G. J. Ward and P. Heckbert. Irradiance Gradients. In *Third Eurographics Workshop on Rendering*, pages 85–98, Bristol, UK, 1992.

[22] K. Zimmerman and P. Shirley. A Two-Pass Realistic Image Synthesis Method for Complex Scenes. In P. M. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95 (Proceedings of the Sixth Eurographics Workshop on Rendering)*, pages 284–295, 1995.