Geometric Simplification for Indirect Illumination Calculations

Holly Rushmeier†, Charles Patterson‡, Aravindan Veerasamy‡

†National Institute of Standards and Technology
Gaithersburg, MD 20899
301-975-3918
e-mail address: holly@cam.nist.gov

‡The Graphics, Visualization and Usability Center
Georgia Institute of Technology, Atlanta, GA 30332
Charles: charliep@cc.gatech.edu
Aravindan: gt8424a@prism.gatech.edu

ABSTRACT

We present a new method for accelerating global illumination calculations in the generation of physically accurate images of geometrically complex environments. In the new method, the environment geometry is simplified by eliminating small isolated surfaces, and replacing clusters of small surfaces with simple, optically equivalent, boxes. A radiosity solution is performed on the simplified geometry. The radiosity solution is then used in a multi-pass method to estimate the radiances responsible for indirect illumination. We present a preliminary implementation of the new method, and some initial images and timing results. The initial results indicate that using simplified geometries for indirect illumination calculations produces images in times significantly less than previous multi-pass methods without a reduction in image quality.

KEYWORDS: Geometric Simplification, Global Illumination, Monte Carlo, Progressive Refinement, Radiosity, Ray Tracing

INTRODUCTION

The outstanding problem in global illumination for computer graphics is the rendering of scenes containing very large numbers of geometric objects. While many useful global illumination methods have been proposed, the generation of physically accurate images of geometrically complex scenes still requires CPU hours on state of the art computer hardware. In this paper we present a method to accelerate global illumination calculations for complex scenes by geometric simplification for indirect illumination (GSII). This method is an extension of the progressive multi-pass method (PMM) described by Chen et al. [2].

The global illumination problem has two major parts — the calculation of direct illumination and indirect illumination. Mathematically, the direct illumination problem requires evaluating an integral with a known integrand over well-defined solid angles. The indirect illumination problem requires evaluating an integral with an unknown integrand over the entire hemisphere. Many successful methods use separate strategies for the two calculations [10][11][14]. The direct illumination problem is difficult when there are many light sources. Recently methods for simplifying the direct calculation have been introduced [12][15]. The indirect calculation becomes difficult when there is a large number of varied objects in an environment. In this paper we focus on the indirect calculation.

A fully correct solution for global illumination would require following all photons through a detailed geometric definition of the environment. Clearly, such a detailed solution is unnecessary to generate a realistic image. One approach to simplifying the calculations is to use many levels of geometric description [8]. These levels include local lighting models (i.e., reflectances and transmittances), mappings (texture and bump maps) and object definitions. For example, in modelling the night sky, the moon can be adequately modelled by a texture map rather than by actually modelling the interaction of photons with the surface micro-structure of the moon's mountains and craters. Such approximations are intuitively acceptable, even though no rules governing their use have ever been developed.

Implicitly, many levels of object definition are used. In static images, objects behind the viewer are modelled with relatively little geometric detail. On one hand this is viewed as "cheating," but is justifiable on the grounds that such details have very little impact on the overall scene illumination. In dynamic radiosity walk-throughs, generally only major objects are modeled for the sake of efficiency. Even without detail the user gets the overall impression of the illumination of the space. The goal of GSII is to begin to formalize the use of simplified and detailed geometries in rendering. Unlike the development of simplifications such as texture and bump mapping, a theoretical basis is developed for determining when GSII is appropriate.

We begin by briefly reviewing previous work for reducing the calculations required for global illumination of geometrically complex environments. Next we describe GSII and the theoretical basis for its application. Finally, we describe an initial implementation of GSII and present preliminary timing results.

PREVIOUS WORK IN GLOBAL ILLUMINATION OF COMPLEX ENVIRONMENTS

A number of approaches for dealing with large numbers of geometric objects have been developed for the radiosity, ray tracing and hybrid approaches to global illumination.

In radiosity methods there are two levels of complexity — the number of geometric objects K, and the number of subsurfaces N into which the K objects must be subdivided to capture illumination detail. In the original radiosity methods [5][10] $\mathrm{O}(N^2)$ interactions had to be calculated to compute a solution. Cohen et al. [3] developed the patch-element hierarchy to avoid the necessity of computing a large number of interactions between small subsurfaces, reducing the complexity of the calculation to $\mathrm{O}(KN)$. Hanrahan et al. [7] built on the hierarchical subdivision idea to further reduce the calculation of surface interactions to a complexity of $\mathrm{O}(K^2)$. While greatly reducing the complexity of radiosity calculations, hierarchical meshing of surfaces does not address the issue of how to deal with large numbers of objects K in the environment.

Xu et al. [17] developed a technique of dividing space into V volume subdivisions, with an average of N/V subsurfaces in each volume. $\mathrm{O}((N/V)^2)$ detailed interactions are computed within each volume. Less detailed interactions are computed between the subsurfaces and the other volumes. The complexity of the second set of calculations is difficult to assess since they depend on maintaining detailed directional geometric factors at volume boundaries that depend on the geometric complexity of the overall environment.

Ray tracing techniques for accurate global illumination, such as Monte Carlo path tracing (MCPT) [9], simplify the problem by only requiring an illumination calculation for objects within the field of view. Furthermore, the solutions for these objects only need to be calculated at screen resolution (e.g., individual radiances of blades of grass need not be calculated if all of the blades project onto the same pixel.) For each of the P pixels, the illumination is found by following paths in D directions. Formally, the complexity of the calculation is just $\mathrm{O}(PD)$. Implicitly however, the number of objects K influences the calculations by influencing the number of directions D required for an accurate solution. More accurately the complexity then is $\mathrm{O}(PD(K))$.

Thompson [13] adds shading attributes to the bounding volumes used by his ray tracer. If a bounding volume subtends a small enough solid angle, the shading attributes are used and the bounding volume's contents are ignored. This reduces the number of object-ray intersection calculations, effectively reducing K to a smaller value K'. Variance is also reduced, requiring a smaller number of directions so that the function $D(K')$ is reduced to the less expensive function $D'(K')$. The complexity is then $\mathrm{O}(PD'(K'))$. Thompson's method gives excellent results for "classic" ray tracing. However, for physically accurate images, Thompson's method is incomplete in that no physically accurate method for computing the "shading attributes" is given.

For both the radiosity and ray tracing approaches, the number of calculations is compounded by the average reflectance of the environment ρ_{ave}. In a progressive refinement radiosity (PRR) solution [4], the number of "shots" required increases with $1/(1-\rho_{ave})$, resulting in an $\mathrm{O}(K^2/(1-\rho_{ave}))$ complexity. In a path tracing approach, the lengths of paths increase with $1/(1-\rho_{ave})$, resulting in an overall complexity of $\mathrm{O}(PD(K)/(1-\rho_{ave}))$.

The goal of hybrid methods, such as the PMM, is to combine the advantages of the radiosity and ray tracing approaches. In a preliminary pass, a radiosity solution is performed on a coarse mesh to allow the user to quickly generate many different views. The coarse mesh gives a low constant for the $\mathrm{O}(K^2)$ radiosity calculation. Since many views are used, the number of pixels P is very high and the constant for simply projecting the surfaces with precomputed radiosities is orders of magnitude smaller than the cost per pixel for a full path tracing solution. Detailed images of selected views are calculated with Monte Carlo path tracing using a radiosity preprocess. The preprocess is used to identify important secondary light sources, and so to reduce the number of directions required to a weaker function of K, $D''(K)$. The preprocess is also used to estimate higher order reflections to eliminate the dependence on ρ_{ave}. The overall complexity of the path tracing pass then is just $\mathrm{O}(PD''(K))$.

The approach used in this paper is to simplify the original K objects in the scene to an orders of magnitude smaller number S. The radiosity pass is then performed on this set of objects for a complexity of $\mathrm{O}(S^2/(1-\rho_{ave}))$. In the Monte Carlo path tracing pass the indirect illumination is calculated using this solution as a preprocess, resulting in a complexity of $\mathrm{O}(PD''(S))$.

THEORY

In this section we present how GSII fits into a solution of the rendering equation, and how rules are developed to perform the geometric simplification.

Solving the Rendering Equation

An image is formed by computing the radiance of each pixel based on the radiance of points in the environment viewed through that pixel. In the rendering equation [9], the radiance $L_o(p,\theta_r,\phi_r)$ leaving a point p in the environment in a direction specified in spherical coordinates as θ_r, ϕ_r is the sum of the emitted radiance at that point $L_{em}(p, \theta_r, \phi_r)$ and the reflected radiance $L_r(p, \theta_r, \phi_r)$. The emitted radiance must be specified to completely define a scene. The reflected radiance is calculated from the following integral:

$$L_r(p, \theta_r, \phi_r) = \int f_r(p, \theta_r, \phi_r, \theta_i, \phi_i) L_i(p, \theta_i, \phi_i) \cos\theta_i d\omega_i \tag{1}$$

where L_i is the radiance incident from direction i, θ_i is the angle between the surface normal and direction i, $d\omega_i$ is a differential solid angle, and the integral is over the incident hemisphere. f_r is the bi-directional reflectance distribution function (BRDF) of the surface.

In the PMM, the BRDF is written as the sum of a highly directional component, $f_{r,h}$ and a weakly directional component $f_{r,l}$. The reflected radiance $L_r(p, \theta_r, \phi_r)$ then can be considered as the sum of the components $L_h(p, \theta_r, \phi_r)$ (defined by Eq.1 with $f_{r,h}$ in place of f_r) and $L_l(p, \theta_r, \phi_r)$ (defined by Eq.1 with $f_{r,l}$ in place of f_r) The radiance of weakly directional reflection, L_l can be further decomposed into four components: $L_{l,s}$ the light reflected from light sources; $L_{l,c}$ the light reflected from light sources via a series of highly directional reflections, $L_{l,h}$ the light reflected from non-light sources via a series of highly directional reflections, and $L_{l,l}$ the light reflected from other weakly directional surfaces.

In the PMM the term $L_{l,l}$ is calculated by:

$$L_{l,l}(p, \theta_r, \phi_r) = \int f_{r,l}(p, \theta_r, \phi_r, \theta_i, \phi_i) L_{rad}(p, \theta_i, \phi_i) \cos\theta_i d\omega_i \tag{2}$$

where L_{rad} is the radiosity solution for the surface visible in direction i. Although this method of calculating $L_{l,l}$ is faster than doing a full path tracing solution, the computation is still extremely time consuming relative to the other illumination components. In GSII the value L_{rad} is replaced by $L_{rad,simp}$, the radiosity calculated for a geometrically simplified version of the environment for light coming from surfaces at a distance of at least r_{thresh} from the point p. This leads to the following set of equations:

$$\text{for } |p - p_i| > r_{thresh}$$
$$L_{l,l}(p, \theta_r, \phi_r) = \int f_{r,l}(p, \theta_r, \phi_r, \theta_i, \phi_i) L_{rad,simp}(p, \theta_i, \phi_i) \cos\theta_i d\omega_i \tag{3}$$

$$\text{for } |p - p_i| \le r_{thresh}$$
$$L_{l,l}(p, \theta_r, \phi_r) = \int f_{r,l}(p, \theta_r, \phi_r, \theta_i, \phi_i) L_{l,l}(p_i, \theta_i, \phi_i) \cos\theta_i d\omega_i$$

where p_i is the point from which incoming light is reflected. We will discuss the parameter r_{thresh} in the following section.

In a similar vein, Hall, [6] retrieved secondary illumination cast upon a surface S_1 from a reflection map of a surface S_2 casting secondary illumination subtended a small enough angle with respect to S_1. Otherwise, secondary rays were continued into the environment. He allowed these rays to intersect simplified objects, e.g. less facets when approximating a sphere, but no simplification of clusters of objects were made.

The justification for using a simplified environment for the radiosity solution is that although the simplified solution is not spatially accurate, the solution is used only to calculate weakly directional, indirect illumination. Weakly directional surfaces reflect a radiance distribution that is much flatter that the incident radiance distribution. Referring to Fig. 1, for a highly detailed spatial input (1a) the weakly directional surface reflects a blurred, or averaged distribution (1b). If the highly spatially detailed distribution is replaced by an averaged distribution (1c), there is little change in the resulting reflected distribution (1d).

The solution using GSII is shown pictorially in Fig. 2. There are two versions of the environment — a detailed version and simplified version. The detailed version is used for all rays calculating high spatial frequency details —

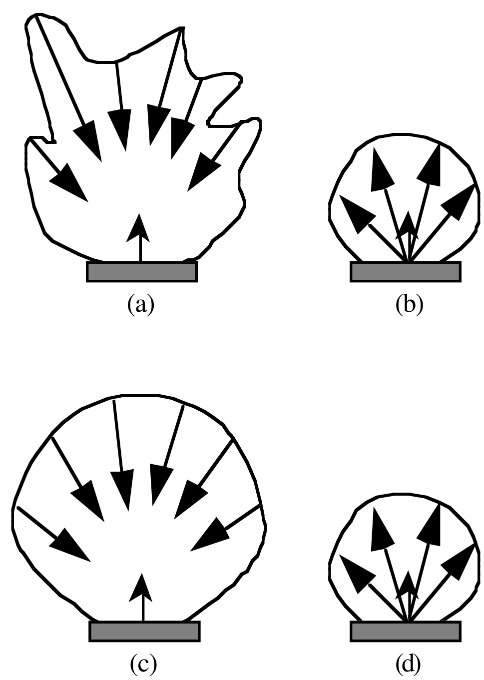
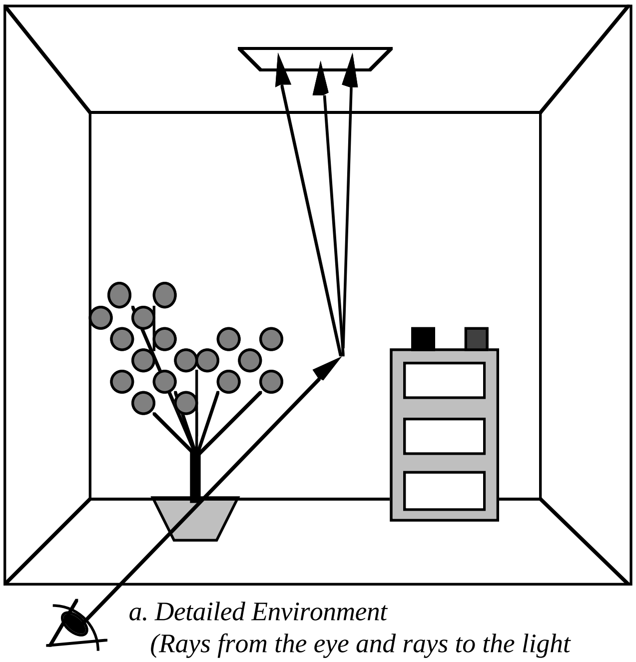
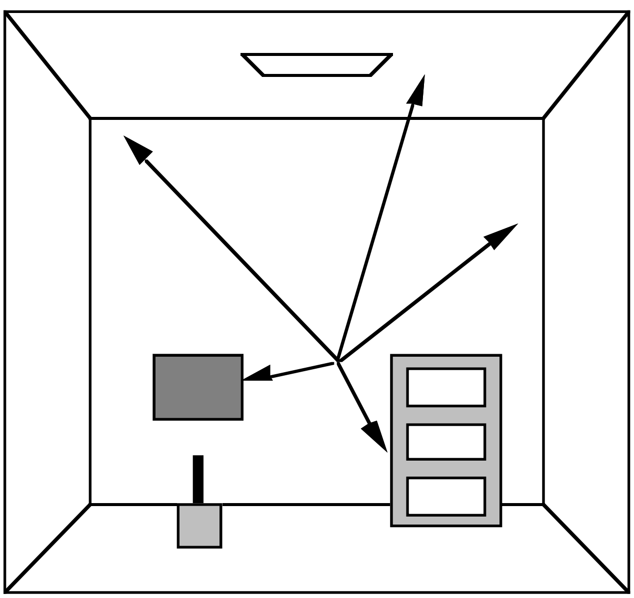


Fig. 1: A weakly directional surface with a spatially detailed incident distribution (a) reflects a blurred or averaged value (b). The same surface with a less detailed distribution with the same average energy (c) reflects approximately the same radiance distribution.



a. Detailed Environment
(Rays from the eye and rays to the light source intersect the detailed environment.)



b. Simplified Environment
(Rays for calculating indirect illumination intersect the simplified environment)

Fig. 2: Diagram showing how detailed and simple geometric representations are used in the calculation of global illumination

view rays, specularly reflected rays, and rays to the light sources. The simplified version is used for indirect illumination rays.

Simplifying Geometries

GSII can be viewed as an extension of the patch/element hierarchy. Surfaces are simplified by methods beyond just using levels of subdivision of individual surfaces. Surfaces of a small enough size are eliminated in the simple form of the environment, and groups of large numbers of small surfaces forming objects or groups of objects are replaced by small numbers of relatively larger surfaces.

Small Isolated Surfaces. The first issue we address is how small must an isolated surface be to be ignored in the simplified solution. To obtain a rule, we assume that the **diffuse non-light source** surfaces in the environment have radiances of approximately the same order of magnitude **relative** to the radiance of the light sources. This condition can be assured by using light source reclassification after the radiosity solution, so that any strong secondary light sources are treated as light sources in the calculation of direct illumination (as in PMM). The importance of a surface in computing $L_{l,l}$ then can be estimated by the solid angle it subtends.

The solid angle subtended by the entire hemisphere above a differential surface dA is equal to 2π steradians. A conservative estimate $\Delta\omega_{\max}$ of the solid angle $\Delta\omega$ subtended by another surface A_j is ,

$$\Delta\omega_{\max} = \frac{A_j}{r_{j,\min}^2} \tag{4}$$

where $r_{j,\min}$ is the minimum distance from dA to A_j. The actual value of $\Delta\omega$ will be less than $\Delta\omega_{\max}$ since $r_{j,\min}$ is the minimum distance, and because of the omission of a cosine term. If $\Delta\omega$ is small compared to 2π, then A_j is small enough to ignore from the point of view of dA. Let β be a user-specified value ranging from zero to one that controls the threshold solid angle, $\Delta\omega_{thresh}$.

$$\Delta\omega_{thresh} = 2\pi\beta \tag{5}$$

If $\Delta\omega_{\max}$ for A_j is less than $\Delta\omega_{thresh}$ with respect to all other surfaces, then A_j can be removed because it will have relatively little effect on the weakly direction component of light for any other surface. The parameter β controls the level of error introduced in using the simplified geometry. Obviously, the error is not exactly equal to β since $\Delta\omega_{\max}$ is not an exact estimate of the importance of the surface.

Calculating $\Delta\omega_{\max}$ for each surface from every other surface is an $O(N^2)$ problem. To avoid this cost, we use the simplified result only for surfaces r_{thresh} away from the point p, as diagrammed in Fig. 3. The value r_{thresh} then can be used as a lower bound on the value of $r_{\min}$. Any surface with an area smaller than A_{thresh} will be removed, where:

$$A_{thresh} = \Delta\omega_{thresh} r_{thresh}^2 = 2\pi\beta r_{thresh}^2 \tag{8}$$

While the parameter β controls accuracy, r_{thresh} is an efficiency parameter. The larger the value of r_{thresh}, the fewer surfaces in the simplified environment, but the results

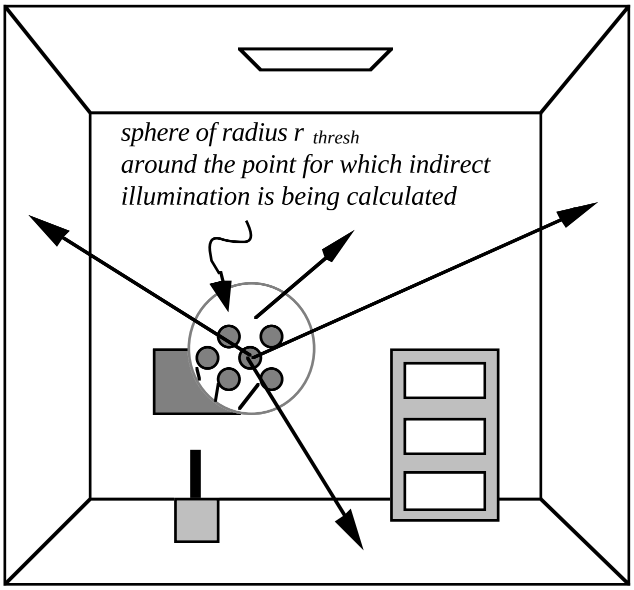


Fig. 3: Use of r_{thresh} in indirect illumination calculation.

of the simplified environment will be used much less frequently in the final rendering. If r_{thresh} is set to zero, all of the original surfaces will be in the simplified environment, and the method will be identical to the PMM. If r_{thresh} is set to infinity there will be no surfaces in the simplified environment, and the values of $L_{rad,simp}$ will never be used — i.e., the method will be identical to MCPT. The tradeoffs in choosing r_{thresh} are similar to the tradeoffs in determining the volume size to use in a uniform space subdivision scheme for accelerated ray casting. Ultimately the value chosen for r_{thresh} should be based on the distribution of sizes of surface in the environment, and on their spatial distribution.

The approach of using r_{thresh} is similar in spirit to the radiosity method proposed by Xu et al. in which detailed form factors are calculated only locally. Unlike their method, however, the local area is not a fixed subdivision of space, but a region which "floats" with the particular location where illumination is being calculated. In its definition of locality, GSII is closer to the method presented by Hanrahan et al.

Clusters of Small Surfaces. An obvious difficulty with the approach just outlined is that small surfaces often occur in clusters. Such clusters cannot simply be deleted from the environment. A simple way to account for the effect of clusters of surfaces is to replace them with a box that is in some sense energy equivalent. The box should provide an energy reflection distribution that is reasonably equivalent to the original surface cluster.

While this replacement of a set of complex surfaces by a simple box appears to be a radical simplification, it is really just an extension of the common replacement of complex microfaceted surfaces with a flat surface and a BRDF. An explicit example of such a replacement is described by Westin et al. [16]. Rays are cast at a microfaceted surface, and the direct of the reflected rays are used to approximate a BRDF using spherical harmonics. In rendering an image of such a surface, the surface is modeled as geometrically smooth, with the BRDF accounting for the true microscopic roughness.

As in Westin et al, we will find the replacement surface by casting a large number of rays at the object. Rather than storing the results as a BRDF using spherical harmonics, we will store the results as a box with size to be determined and with a spatially varying ideal diffuse reflectance. This representation allows us to readily include the simplified object in a radiosity pre-process solution. Obviously, there are many ways in which clusters can be simplified. We have chosen one rather straight forward method to test the idea.

Our method for finding simplified equivalent objects is diagrammed in Fig. 4. The method begins by fitting a minimum bounding box around the cluster, aligned to the coordinate axis used in modelling the object. Each face of the box is discretized into patches. We then cast a series of randomly selected rays perpendicularly through each face of this bounding box. A count is kept of how man rays pass through the box without intersecting the cluster. A list is also kept of the reflectance of cluster surfaces that are hit by the rays. We then represent the transmittance of the box, calculated by the percentage of rays that pass through directly, by shrinking the size of the box. We represent the reflectance of each face by setting the reflectance of each patch on the shrunken box to the average reflectance of surfaces hit. In pseudo-code:

```
for each of the six faces of the box sized X by Y by Z {
    hits_face = 0;
    for each patch on the face 1 to num_patches {
        hits_patch = 0;
        rho_sum_patch = 0.0;
        for each perpendicular ray 1 to num_rays {
            shoot a ray into the box;
            If a surface is hit {
                hits_patch += 1;
                rho_sum_patch += rho(surface hit); } }
        rho(patch) = rho_sum_patch / hits_patch;
        hits_face += hits_patch; }
    frac(face) = hits_face / (num_rays * num_patches); }

new_x_length = sqrt( frac(xy_face) * frac(xz_face) ) * X;
new_y_length = sqrt( frac(xy_face) * frac(zy_face) ) * Y;
new_z_length = sqrt( frac(xz_face) * frac(yz_face) ) * Z;
```

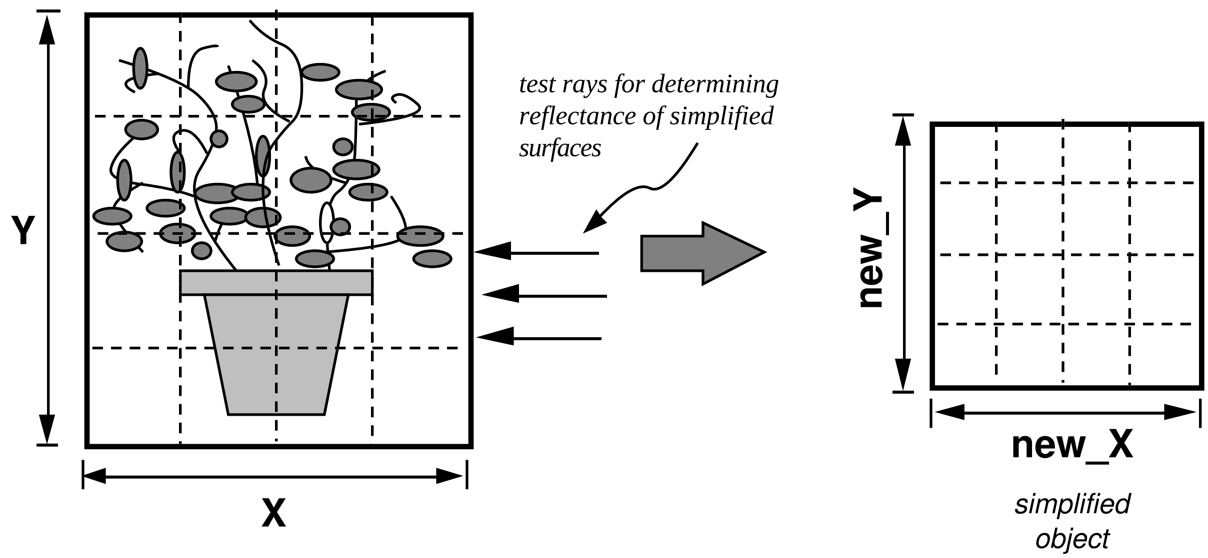


Fig. 4: A method for finding the simplified object definition.

In our solution method then, we define three types of surfaces: normal, complex and simple. Normal surfaces are large enough to be used both in the radiosity solution and in the final rendering. Complex surfaces are small and are not used in the radiosity solution, but are used in the high resolution pass of the final rendering. Simple surfaces are used in place of clusters of complex surfaces in the radiosity solution and in the low resolution pass of the final rendering.

IMPLEMENTATION OF A PRELIMINARY GSII SYSTEM

A preliminary rendering system using GSII is diagrammed in Fig. 5. After creation in the modeller, all surfaces are converted to the format used by the rendering programs. A separate file of objects to be simplified is also converted, and run through the Simplifier module. All of the descriptions are then merged and sent to the Radiosity module that calculates radiosities for the normal and simple surfaces. The Renderer module uses the complete geometric description and the radiosities to compute the final image.

Simplifier

In our preliminary implementation, clusters of small surfaces to be replaced are identified in the modelling process. User supplied values of r_{thresh} and β are used to determine whether the remaining surfaces should be classified as "normal" or "complex".

A final version of the GSII method would need an automated method for identifying such clusters. Our approach is similar to the development of meshing for radiosity. A very simple method was used initially [5]. Once the utility of the radiosity method was established, more sophisticated meshing methods were developed [1]. A possible way of identifying clusters in the future would use a spatial subdivision scheme, such as an octree, to identify surfaces below the minimum size which a located within a minimum radius.

Radiosity

The radiosity solution is calculated by a progressive refinement program that uses the hemi-cube algorithm with a hardware Z-buffer to find form factors. The solver acts only on normal and simple surfaces. Each surface is subdivided into triangular patches. The hemi-cube resolution and percentage original unshot energy stopping criterion are set by the user.

Renderer

The renderer has two passes. In both passes all ray casting is performed using uniform spatial subdivision to reduce ray surface intersections. The first pass is the high spatial frequency (HIRES) pass which accounts for direct illumination, specular reflections and caustics. Rays in this pass, as shown in Fig. 2, must use the normal + complex surfaces to capture the fine detail. At the end of the HIRES pass two values are stored for each pixel for each wavelength band — the radiance L_{HIRES}, and $L_{HIRES,dev}$ the sample standard deviation. For each pixel, trial values are calculated until either the ratio $(L_{HIRES,dev} / L_{HIRES})$ falls below a user supplied value, or a user supplied maximum number of trials is reached.

The second pass is the low spatial frequency (LORES) pass. This pass accounts for the $L_{l,l}$ term in Eq. (2). In the LORES pass the rays from the first visible object out to the environment consider normal + complex surfaces only within a radius r_{thresh}. If a surface is intersected within this radius, the illumination calculation continues as in a Monte Carlo path tracing solution. Beyond this radius, only normal + simple surfaces are intersected, and the radiance of any surface beyond r_{thresh} is taken to be the radiance from the radiosity solution. The values L_{LORES} and $L_{LORES,dev}$ are calculated in this pass. Trial values are calculated until either the ratio $(L_{HIRES,dev} + L_{LORES,dev}) / (L_{HIRES} + L_{LORES})$ falls below a user supplied value, or a user supplied maximum number of trials is reached. In the final image the values of $(L_{HIRES} + L_{LORES})$ are displayed.

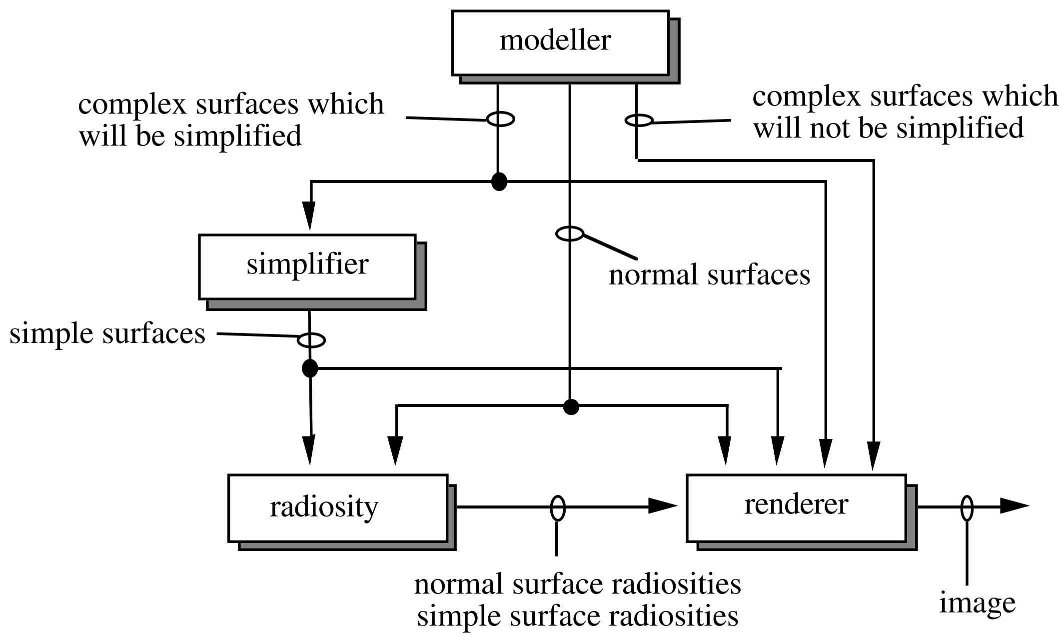


Fig. 5 Structure of preliminary GSII rendering system.

RESULTS

To evaluate the effectiveness of GSII, we built a test environment composed of 10,948 surfaces. These surfaces made up the objects in an office including thumb tacks, pens, a computer keyboard and a plant. The simplified solution, direct illumination, and combined solution using GSII for the environment is shown in Fig. 6. (Note: Figs. 6 to 8 can be found at the end of the paper). The detail in the environment, as rendered by GSII, is shown in the three images in Fig. 7. We considered two cases — one in which surfaces in the environment have relatively low reflectances (ρ_{ave} = 0.41), and indirect illumination is relatively unimportant, and the other in which surfaces have relatively high reflectances (ρ_{ave} = 0.56) and indirect illumination is more important. (The adjustments in average reflectance

were made by adjusting the wall reflectances only.) The images shown in Figs. 6 and 7 are of the high reflectance environment. To make timing comparisons, we generated images for each case using progressive refinement radiosity (PRR) with normal + complex surfaces, PRR with normal + simple surfaces, Monte Carlo path tracing (MCPT), PMM, and GSII. The PRR with normal+complex surfaces, PMM and MCPT solutions for the high reflectance environment are shown in Fig. 8.

In solving for the original and simplified scene with PRR, we used a hemi-cube of 150x150 pixels per full face. The normal + complex environment contained 10,948 surfaces divided into 74,924 patches. We did not use element or sub-element subdivision. The simple environment was generated using a value of $r_{thresh} = 0.053$ times the largest linear room dimension, and $\beta = 1/2\pi$. These parameters were chosen to substantially reduce the number of surfaces in the environment while allowing a relatively modest error level. The normal + simple environment contained 105 surfaces subdivided into 1124 patches. We continued the solution until one percent of the original energy in the room was left unshot. The timings for the PRR solutions are summarized in Table 1.

The simplified geometry contains far fewer surfaces than the original, and the result is orders of magnitude lower timings for the PRR on the simplified geometry. While the absolute values of the timings are machine and implementation specific, the ratios of timings for the simple and complex solutions are significant.

A 12x12x12 uniform spatial subdivision was used for all ray casting in the rendering phase. Solutions for each pass were allowed to run until the sample standard deviation was equal to 5 per cent of the computed value, or a maximum of 30 trials was reached. A minimum of 10 trials were run for each pixel. Table 2 shows the timing results for the HIRES pass for each case.

Table 3 shows the time results for each case for each method for the LORES pass. The advantage of using either the PMM or GSII over the MCPT method is clear when the timings are compared. Furthermore, the increase in time for solution for the MCPT with the increased average reflectance is evident, while the timings for the PMM and GSII are the same for both the low and high cases. The relatively high average value of L_{dev} / L for the MCPT solution indicates that the MCPT would require significantly more time to generate an image with comparable noise level.

Table 1
Timings* for PRR Preprocess

Average Reflectance	Representation	# Patches	Timing (hrs:min)
Low (0.41)	N + S	1124	0:25
Low (0.41)	N + C	74924	50:25
High (0.56)	N + S	1124	0:52
High (0.56)	N + C	74924	**142:14

*Timings for SGI 4D/20 Personal Iris
**Solution did not run to completion
N+S = Normal +Simple
N+C = Normal+Complex

Table 2
Timings* for HIRES Pass

Average Reflectance	Average L_{dev} / L	Timing (hrs:min)
Low (0.41)	0.074	3:17
High (0.56)	0.074	3:16

*Timings for SGI 4D/280 using only 33 MHz processors

Table 3
Timings* for LORES Pass

Average Reflectance	Method	Average L_{dev} / L	Timing (hrs:min)
Low (0.41)	MCPT	0.349	7:22
Low (0.41)	PMM	0.280	3:39
Low (0.41)	GSII	0.276	4:14
High (0.56)	MCPT	0.282	10:29
High (0.56)	PMM	0.189	3:47
High (0.56)	GSII	0.193	4:17

*Timings for SGI 4D/280 using only 33 MHz processors

Table 4
Average Relative Differences
for Indirect Component of Radiance ONLY

Quantity	R	G	B
|MCPT – GSII| / MCPT	0.11	0.10	0.12
|MCPT – PMM| / MCPT	0.18	0.16	0.14
L_{dev} / L for MCPT	0.10	0.07	0.06

The comparison of PMM and GSII timings show that they are approximately the same. In this instance the use of spatial subdivision equalized the cost of casting rays into the simple and complex environments. However, PMM must use the radiosity results from normal + complex objects. Also, it appears that the variance in environment illumination was not significantly different between the complex and simple radiosity solutions. The GSII takes somewhat more time because it uses the radiosity solution only outside of the radius r_{thresh}, instead of for all indirect illumination rays.

The major differences in appearance between the GSII andthe PMM images are on the back wall. These result from taking the radiosity of surfaces that are too close. That is, although the radiosity solution is correct on average, some individual surfaces are too bright and some too dark. In the PMM errors are produced when a surface uses the erroneous radiosity of a nearby surface that subtends a large solid angle. By using the radius r_{thresh}, in the GSII these types of errors are avoided, and the small additional computational time is justified.

The images generated for the timing comparisons can't be used to visually assess the accuracy of the GSII solution relative to the MCPT because of the high level of noise in the MCPT image. Clipping of extreme values in the noisy image results in the uneven clipping of rgb values. The eye cannot average the MCPT results appropriately. To examine the accuracy issue more carefully, we generated images at a lower pixel resolution (100x100) for a larger number of trials per pixel — up to 256 trials per pixel for the PMM and GSII, and up to 1024 trials per pixel for the MCPT.

For accuracy trials, the images are rendered for the most challenging component of the most challenging case used in the timing trials — the indirect component of the high reflectance case. Table 4 summarizes the numerical MCPT solution itself. The absolute value of the difference was used to avoid cancelling errors.

Looking at averaged results, the error in using the GSII was of the same order of magnitude as the error introduced by using the radiosity preprocess, and as the noise in the MCPT result. Error introduced by using the radiosity preprocess was due to relatively poor meshing, and omission of any specular interreflections.

It is difficult to assess how important error levels in the floating point calculation of radiance are in the final image. Ideally, the final results should be judged visually after the floating point values have been mapped to display values using a perceptually based, non-linear, tone reproduction operator.

A few simple tests on one environment cannot prove anything definitively. However, these preliminary results suggest that (1) GSII can be used in place of PMM to generate images without a degradation in the results and (2) the major time savings in using GSII over the PMM is in greatly reducing the time to perform the radiosity preprocess. Many more tests are required to confirm these conclusions.

SUMMARY AND FUTURE WORK

We have described and presented a preliminary implementation of a global illumination method that uses simplified geometric representations for indirect illumination. We have presented initial results that indicate that the method can produce images more quickly than Monte Carlo path tracing or the progressive multi-pass method without degradation of quality.

Potential future work is in three areas — testing, simplification methods, and animation. Clearly more tests are required to examine the relative timings for various types of environments. Detailed analyses of the accuracy of the results of the various methods should also be made. Automated geometric simplification methods are needed if the approach is to become practical. Finally, the use of spatially and temporally less detailed global illumination solutions as a preprocess for animated sequences could be explored.

ACKNOWLEDGMENTS

This work was supported by grants from the National Science Foundation and the Apple Computer Company.

REFERENCES

1. Baum, D.R., Mann, S., Smith, K.P. and Winget, J.M. Making Radiosity Usable: Preprocessing and Meshing Techniques for the Generation of Accurate Radiosity Solutions. Computer Graphics (SIGGRAPH 91), vol. 25, (1991), pp. 51-60.

2. Chen, S. E., Rushmeier, H. E., Miller, G. and Turner, D. A Progressive Multi-Pass Method for Global Illumination. Computer Graphics (SIGGRAPH 91), vol. 25, (1991), pp. 165-174.

3. Cohen, M.F., Greenberg, D.P., Immel, D.S. and Brock, P.J. An Efficient Radiosity Approach for Realistic Image Synthesis. IEEE Computer Graphics and Applications, vol. 6:2, (1986), pp. 26-35.

4. Cohen, M.F., Chen, S.E., Wallace, J.R. and Greenberg, D.P. A Progressive Refinement Approach to Fast Radiosity Image Generation. Computer Graphics (SIGGRAPH 88), vol. 22, (1988), pp. 75-84.

5. Goral, C.M., Torrance, K.E., Greenberg, D.P. and Battaille, B. Modelling Interreflections Between Diffuse Surfaces. Computer Graphics (SIGGRAPH 84), vol. 18, (1984), pp. 213-222.

6. Hall, R. Hybrid Techniques for Rapid Image Synthesis. Course Notes for Image Rendering Tricks (SIGGRAPH 86), (1986).

7. Hanrahan, P., Salzman, D. and Aupperle, L. A Rapid Hierarchical Radiosity Algorithm. Computer Graphics (SIGGRAPH 91), vol. 25, (1991), pp. 197-206.

8. Kajiya, J. Anisotropic Reflectance Models. Computer Graphics (SIGGRAPH 85), vol. 19, (1985), pp. 15-21.

9. Kajiya, J. The Rendering Equation. Computer Graphics (SIGGRAPH 86), vol. 20, (1986), pp. 143-150.

10. Nishita, T. and Nakamae, E. Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection. Computer Graphics (SIGGRAPH 85), vol. 19, (1985), pp. 23-30.

11. Shirley, P. A Ray Tracing Algorithm for Global Illumination. Graphics Interface 90, (1990), pp. 205-212.

12. Shirley, P. and Wang, C. Luminaire Sampling in Distribution Ray Tracing. Indiana University Computer Science Technical Report, #343, (1992).

13. Thompson, K. Ray Tracing with Amalgams. Ph.D. Dissertation, Department of Computer Science, University of Texas at Austin, (1991).

14. Ward, G.J., Rubenstein, F.M. and Clear, R.D. A Ray Tracing Solution for Diffuse Interreflection. Computer Graphics (SIGGRAPH 88), vol. 22, (1988), pp. 85-92.

15. Ward, G.J. Adaptive Shadow Testing for Ray Tracing. Second Eurographics Workshop on Rendering, Barcelona, Spain, (1991).

16. Westin, S.H., Arvo, J.R. and Torrance, K.E. Predicting Reflectance Functions from Complex Surfaces. Computer Graphics (SIGGRAPH 92), vol. 26, (1992), pp. 255-264.

17. Xu, H.P., Peng, Q.-S. and Liang, Y.-D. Accelerated Radiosity Method for Complex Environments. Computers and Graphics, vol. 14:1, (1990), pp. 65-71.