EUROGRAPHICS '97 / D. Fellner and L. Szirmay-Kalos
(Guest Editors)

Volume 16, (*1997*), *Number 3*

A Rendering Equation for Specular Transfers and Its Integration into Global Illumination

Yizhou Yu† and Hong Wu‡

†Computer Science Division, EECS, University of California at Berkeley, CA 94720, USA
‡General Magic, Inc. , Sunnyvale, CA, USA

Abstract

In this paper, we present a rigorous theoretical formulation of the fundamental problem—indirect illumination from area sources via curved ideal specular surfaces. Intensity and area factors are introduced to clarify this problem and to rectify the radiance from these specular surfaces. They take surface geometry, such as Gaussian curvature, into account. Based on this formulation, an algorithm for integrating ideal specular transfers into global illumination is also presented. This algorithm can deal with curved specular reflectors and transmitters. An implementation is described based on wavefront tracing and progressive radiosity. Sample images generated by this method are presented.

Keywords: Rendering equation; Global illumination; Specular transfer; Wavefront tracing; Radiosity; Ray tracing; Meshing

1. Introduction

Computing solutions to the global illumination problem is an essential part of photorealistic image synthesis. Global illumination effects produced by multiple surface reflections are significant in all but the simplest environments. For instance, indirect lighting and color bleeding can be observed in almost all indoor scenes. In a general scene, reflectance types of surfaces can be classified into ideal diffuse, ideal specular and directional diffuse components [10]. Later in this paper, we will use the word *specular* as a short form of *ideal specular*. Kajiya [13] gave an equilibrium equation of radiative transfer. But he did not give the accurate way to get radiance values from ideal specular surfaces. People usually think the reflectance of an ideal specular surface is independent of its geometry. A careful study of the propagation of wavefronts [17] shows that if the specular surfaces are curved, the Gaussian curvatures as well as the whole path from the source emitting the light to the receiving point via a couple of intermediate specular surfaces need to be considered in order to get radiance outgoing from an ideal specualr surface. In this paper, We derive a general equation governing these specular transfers and integrate it into global illumination.

There are many approaches solving this global illumination problem. One major category [2, 12, 11, 20] uses light ray-tracing or photons emitted from the light sources. The primary difficulties with light ray-tracing involve sampling and estimating the distribution of irradiance. Light rays arrive at their final destinations in a nonuniform pattern. We must use variation-reduction sampling techniques to estimate the distribution of irradiance from photon locations and density. The efficiency of these techniques still need to be improved. For large area sources, we must sample the area sources as well as the direction. These are troublesome problems which have not been completely solved.

Another category uses radiosity-like finite-element algorithms[27, 24, 1, 25, 22, 23]. Wallace et al.[27] incorporated planar mirrors into classical radiosity(CR) by reflecting the entire environment into a virtual environment. Sillion and Puech[24] extended radiosity to accommodate curved specular surfaces by introducing extended form factors. Sillion et al.[25] extended progressive radiosity(PR) to include arbitrary reflectance distributions. But it only uses planar surfaces as ideal specular reflectors. Aupperle and Hanrahan [1] extended hierarchical radiosity(HR) to include glossy reflection.

The method proposed by Mitchell and Hanrahan [17] uses wavefront tracing to compute illumination from curved specular reflectors at locations of our choosing in order to avoid those problems arising in light ray-tracing. Elber[7] approxi-

mated the propagation of whole wavefronts. But point and spherical light sources are the only choice. Collins[5] adopted wavefront tracking to improve the performance of light ray-tracing.

We prefer a radiosity-like global illumination solution. One reason is that it does not need sophisticated sampling and estimation techniques. The other reason is that it is relatively efficient. In a radiosity-like algorithm, each patch in the scene is a potential area light source. But as we know, the problem of illumination from area source via curved specular reflectors has not been solved well, both theoretically and practically. Perhaps this is because it is almost impossible to store the radiance distribution accurately over an ideal specular surface. This paper will focus on this fundamental problem and propose a radiosity-like method to combine wavefront tracing with global illumination. In the next section, we discuss a theoretical frame for general specular transfers. The third section is devoted to the presentation of an algorithm combining ideal specular transfers and global illumination, in particular, radiosity. In Section 4, we gave some experimental results. And we conclude this paper in Section 5.

2. A Rendering Equation for Ideal Specular Transfers

In the following, we will use wavefront tracing to give a rendering equation which implies more information about ideal specular transfers. For conciseness, all equations in this paper only take into account the radiosity or radiance contributed by specular transfers. The idea presented here is also useful in ray tracing algorithms.

2.1. Basics on Wavefront Tracing

Wavefront tracing is used to keep track of the Gaussian curvature and intensity of the wavefront along a ray path between a chosen location and a light source via a couple of ideal curved reflectors or transmitters. This involves three operations:

- Transfer
- Reflection
- Refraction

The equations describing the evolution of the wavefront for each of these situations could be found in [17] and [26]. Here we only give the equations of transfer operation:

$$K_1^{'} = \frac{K_1}{1 - dK_1}, K_2^{'} = \frac{K_2}{1 - dK_2} \tag{1}$$

where K_1, K_2 are principal curvatures of the wavefront, and d is the distance the wavefront travels. Note that these equations can also be written in terms of r which is the radius of curvature and equal to $\frac{1}{K}$.

A transfer operation happens between two adjacent reflection(or refraction) operations. During this operation, the intensity at two different points on the ray path obeys the following *intensity law*:

$$\frac{I^{'}}{I} = \frac{r_1 r_2}{r_1^{'} r_2^{'}} = \frac{dA}{dA^{'}} \tag{2}$$

where dA and $dA^{'}$ are two corresponding differential areas sorrounding these two points. This shows the important fact that the intensity along the ray is proportional to the Gaussian curvature of the wavefront $\frac{1}{(r_1 r_2)}$ where r_1 and r_2 are principal radii of curvature.

The Gaussian curvature of the wavefront could change abruptly during a reflection or refraction operation. And Euler's Formulae and their inverse are used to compute the new curvatures. As we know, radiance does not change during a transfer operation. But it may change during a reflection or refraction operation if the specular surface is not planar.

Suppose there are n transfer operations along a ray path RP from some point on a light source L to a chosen point P. Let $I_i^{'}$ and I_i denote the intensity at the beginning and end of the i-th transfer. Then the ratio of the intensity arriving at P to that emitted from L(for point light source, I_1 denotes the intensity at some point on the unit sphere sorrounding the light source) is denoted by *PIF*:

$$PIF(RP) = \frac{I_n^{'}}{I_1} = \prod_{i=1}^{n} \frac{I_i^{'}}{I_i} \bullet \prod_{i=1}^{n-1} \frac{I_{i+1}}{I_i^{'}} = \prod_{i=1}^{n} T_i \bullet \prod_{i=1}^{n-1} R_i \tag{3}$$

where $T_i (= \frac{I_i^{'}}{I_i})$ can be evaluated by the intensity law, and $R_i (= \frac{I_{i+1}}{I_i^{'}})$ is the change of intensity caused by the i-th reflection(refraction) operation. R_i's can be evaluated via Fresnel's law as a function of the angle of incidence. For ideal opaque reflectors, there is little change in intensity during reflection operations. Therefore $R_i \approx 1$. For reflection operations on a transparent reflector, R_i is given by Fresnel's coefficient for reflection which is also the reflectance. For refraction operations, R_i is given by

$$(\frac{n_2}{n_1})^2 (1 - \alpha) \tag{4}$$

if the light passes through the medium M_1 with refractive index n_1 to the medium M_2 with refractive index n_2 and the reflectance of border is α[19].

2.2. Specular Transfer of Area Sources

Originally, wavefront tracing was used for point light sources. For area sources, we have to take into account the ratio of a differential area on the area source over its corresponding differential area on the specular surface, as well as the decay of intensity. In a general environment consisting of planar and curved surfaces, it is necessary to get a mathematical equation for specular transfers of area sources. There are no such governing equations to date. We derived such an equation. To see how it was obtained, first look at the following simple situation(Figure 1) where P is a point on a diffuse

surface, S is a diffuse square light source with radiance I and area A, and M is an ideal planar mirror parallel to S. To get the irradiance at P from M, we first get the image S' of S in M and then compute the irradiance at P from virtual source S', which approximately equals $\frac{IA}{d'^2}cos\theta_1 cos\theta_2$ if A is small compared to the distance d' between S' and P. Let S'' be the intersection of M and the pyramid formed by P and S'. Since S' is parallel to M, S'' is also a square and its area $A'' = \frac{d''^2}{d'^2}A$ where d'' is the distance between P and S''. If we consider S'' as an area source with radiance I, the irradiance at P from S'' is also equal to $\frac{IA''}{d''^2}cos\theta_1 cos\theta_2 = \frac{IA}{d'^2}cos\theta_1 cos\theta_2$, which means the effect of virtual sources S' and S'' on P is the same.

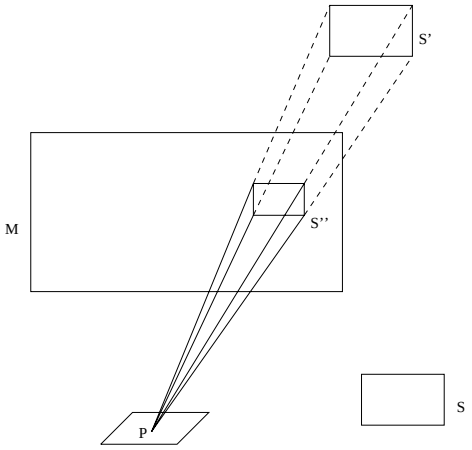


Figure 1: *Planar specular reflection*

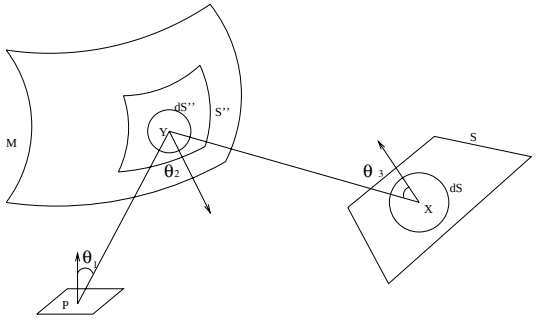


Figure 2: *Curved specular reflection*

Now if M is a curved reflector as in Figure 2, we can not easily get the image of S. But there is one thing similar. As in Figure 1, there is also an area S'' on M, which reflects the light from S to P and which is not necessarily connected. That is, if we look at M from P, S'' is bright and the remaining part of M is completely dark. We want to make the effect of S'' on P the same as that of S on P via M. If this can be done, we can simply integrate over S'' rather than over S, which is much easier.

For a differential area dS'' within S'', there exists a differential area dS within S such that for each point X in dS, there exists (at least)one point Y in dS'' such that $\vec{YP}$ is the reflection ray of $\vec{XY}$. Then for each such ray path $X \to Y \to P$, according to (3), we define the *intensity factor*(IF) at Y to be

$$IF(P,Y) = \|\vec{YP}\|^2 PIF(X \to Y \to P) \tag{5}$$

and the *area factor*(AF) at Y to be the ratio of effective areas

$$AF(P,Y) = \frac{dScos\theta_3}{dS''cos\theta_2} \tag{6}$$

where θ_2 is the angle between $\vec{YP}$ and the normal at Y, and θ_3 is the angle between $\vec{XY}$ and the normal at X.

We can verify that if the radiance at X towards Y is I, and the radiance at Y towards P is artificially assigned to be $I \cdot IF(P,Y) \cdot AF(P,Y)$, the irradiance at P from dS'' is the same as that from dS via M, which is $\frac{cos\theta_1 cos\theta_2}{r^2} I \cdot IF(P,Y) \cdot AF(P,Y)dS''$ where θ_1 is the angle between $\vec{PY}$ and the normal at P and r is the distance between P and Y. So $I \cdot IF(P,Y) \cdot AF(P,Y)$ is called the *virtual radiance* at Y with respect to P. If radiance is assigned in this way to every point within S'', illuminating P by S'' has the same effect as illuminating P by specular transfer of S via M. So we finally arrive at the equation for computing specular transfers of area sources,

$$B(P) = \rho \int_{S''} \frac{cos\theta_1 cos\theta_2}{r^2} I(X) \cdot IF(P,Y) \cdot AF(P,Y) dS'' \tag{7}$$

This equation is for ideal diffuse receivers. $B(P)$ is the radiosity contributed by M at a chosen point P on the receiver and its albedo is ρ.

If there are several other reflectors(transmitters) in the middle of the ray path from X to Y in Figure 2, the ray path $X \to Y \to P$ in (5) should be replaced with a longer path. But (6) is left unchanged. We further define $I'(P,Y) = I(X)$ if S is hit at some point X by ray tracing $\vec{PY}$ for $\forall Y \in M$; otherwise, $I'(P,Y) = IF(P,Y) = AF(P,Y) = 0$. Let

$$VIS(P,Y) = \begin{cases} 1, & \text{if Y is visible from P;} \\ 0, & \text{otherwise.} \end{cases}$$

We can obtain the following equation for a general environment,

$$B(P) = \rho \int_M \frac{cos\theta_1 cos\theta_2}{r^2} I'(P,Y) \cdot IF(P,Y) \cdot AF(P,Y) \cdot VIS(P,Y) dS'' \tag{8}$$

For directional diffuse receivers, let $\rho(\vec{v}, \vec{w})$ be the reflectance distribution of the receiver. Then (8) becomes

$$I(P,\bullet) = \int_M \rho(\vec{YP}, \bullet) \frac{cos\theta_1 cos\theta_2}{r^2} I'(P,Y) \cdot IF(P,Y)$$

$$\cdot AF(P,Y)\cdot VIS(P,Y)dS'' \tag{9}$$

where $I(P,\bullet)$ denotes the radiance distribution at point P.

2.3. Computation of Area Factor

In (8) and (9), we need to compute $I'(P,Y)$, $IF(P,Y)$ and $AF(P,Y)$. $I'(P,Y)$ and $IF(P,Y)$ could be evaluated easily by wavefront tracing. If we trace the ray $\vec{PY}$ (Figure 2) and ultimately hit S at X via a couple of intermediate reflections and/or refractions, $I'(P,Y) = I(X)$ and $IF(P,Y)$ can be evaluated with (5) and (3) by propagating a light wavefront from X to P along the ray path. But the evaluation of $AF(P,Y)$ is not that obvious. We will show it can still be computed by wavefront tracing.

In Figure 2, for any differential area dS'' within S'', there is a corresponding differential area dS within S. From the definition of dS and reversibility of ray path, for any point $Y \in dS''$, we will hit some point X in dS if we trace the ray $\vec{PY}$. Assume P is a point light source and a spherical wavefront W originates from it. When W reaches Y, there must be a differential area dA' on W which sourrounds Y and matches the effective area of dS''. dA' will propagate to X, become another differential area dA and match the effective area of dS. So

$$AF(P,Y) = \frac{dA}{dA'} \tag{10}$$

Along with (2), we can arrive at a formula similar to (3) for $AF(P,Y)$. And we can see that intensity and area factors can be evaluated by propagating two wavefronts from P and X, respectively. Thus we have an efficient way to get these fators. There is no change of differential area on the wavefront during a reflection operation because the incident and outgoing directions are symmetrical to the normal of the surface. During a refraction operation, differential area on the wavefront is changed by a factor, $\frac{\cos t}{\cos i}$, where i is the angle of incidence and t is the angle of refraction.

For the simple case in Figure 1, $IF(P,Y)\cdot AF(P,Y) = 1$ for $\forall Y \in S''$. But in general, it is not. An ideal transmitter is such a counterexample.

From the point of view of light ray-tracing, each photon carries an equal amount of energy. Larger $AF(P,Y)$ means the probability, that a photon reaches P along a path near $\vec{YP}$, is relatively large for the same intensity factor. Small $IF(P,Y)$ means the probability, that a photon emitted from dS reaches P along a path near $\vec{YP}$, is small. From these discussion, we can see that light ray-tracing, which has slow rate of convergence, is a purely stochastic model used to solve (8) or (9).

2.4. Some Comments on Ray Tracing

Conventional ray tracing—only tracing rays from the view point—has produced some of the most impressive realistic images to date. But from the above discussion, we can see it is not physically accurate for curved reflectors or transmitters if it does not consider the effects on radiance caused by the local geometric properties, such as Gaussian curvature, of these reflectors or transmitters. Given the location of a viewpoint V and an image frame, we need to evaluate the incident radiance at each pixel within the frame. In Figure 3, T is a transmitter, D is (directional)diffuse, P'' is the center of a pixel, P and P' are the intersections between D, T and the ray path. We can consider the image frame as a planar transmitter with the same refractive index on both sides. If we do not consider reflection ray path here, the incident radiance at P'' should be $I(V,P'') = I(P)\cdot IF(V,P'')\cdot AF(V,P'') \neq I(P)$. So the computation of rectifying factor $IF(V,P'')\cdot AF(V,P'')$ should be added into conventional ray tracing algorithms. But for thin objects, conventional ray-tracing still gives a good approximation. In order to reduce aliasing, we may wish to compute the incident radiance at several points within each pixel and then take the (weighted)average.

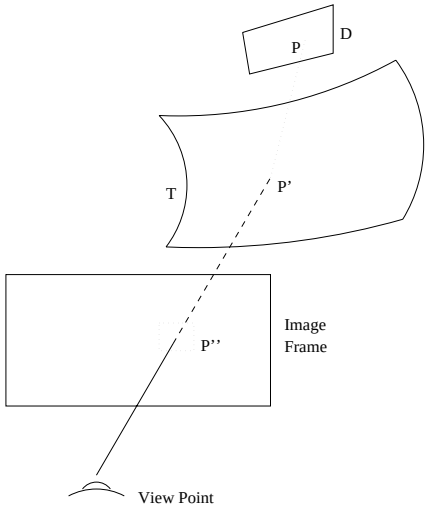


Figure 3: *Rectifying radiance in ray tracing*

Since we are able to wavefront trace any implicit surface[17], the additional computation of intensity and area factors does not put more restrictions on the original ray tracing algorithm.

3. A solution for Parametric Surfaces

Equations (8) and (9) do not put any restriction on the representation of the surfaces. In this section, we propose an efficient solution based on these equations to integrate specular transfers into global illumination for environments consisting of bivariate parametric surfaces. We will discuss the solution for (8), and the solution for (9) can be obtained similarly. Our solution also has two passes. The first pass computes the radiosity of each surface, including the radiosity

contributed by ideal specular surfaces. The second pass uses ray(wavefront) tracing to generate the image. The radiance contributed directly by diffuse surfaces is retrieved from the radiosity stored in the first pass. The radiance contributed by ideal specular transfers is obtained by recursively following reflected and/or refracted rays. Two passes seem necessary to environments with ideal specular surfaces since it is difficult to store radiance distribution on them. The lumigraph [8] and light field [14] can be used for planar specular surfaces. But it is not so straightforward to use them for curved specular surfaces.

3.1. Integral Equations for Parametric Surfaces

Suppose $\{\vec{x_i} | \ i = 1, \cdots n\}$ denotes the set of surfaces in the environment, and $\{\vec{x_i} | \ i \in SS\}$ is the subset of surfaces with ideal specular component. It is convenient to express the radiant distribution and surface geometry in the same parametric domain. The principal curvatures of surface $\vec{x}\ (s,t)$ can be obtained from the *first and second fundamental forms* in *differential geometry*. If written in parametric form, (8) becomes

$$B_i(s,t) = \sum_{j \in SS} \int \int \rho_i(s,t) K_{ij}(s,t,u,v) I_{ij}(s,t,u,v) \cdot dudv \tag{11}$$

where

$$I_{ij}(s,t,u,v) = I'_{ij}(s,t,u,v) IF_{ij}(s,t,u,v) AF_{ij}(s,t,u,v)$$

is the virtual radiance and $I'_{ij}(s,t,u,v)$ has the same meaning as $I'(P,Y)$ in (8), and

$$K_{ij}(s,t,u,v) = F_{ij}(s,t,u,v) VIS_{ij}(s,t,u,v) A_j(u,v)$$

where

$$F_{ij}(s,t,u,v) = \frac{[\vec{l_{ij}} \cdot \vec{n_i}\ (s,t)][-\vec{l_{ij}} \cdot \vec{n_j}\ (u,v)]}{\|\vec{l_{ij}}\|^4} \tag{12}$$

$$A_j(u,v) = \| \frac{\partial \vec{x_j}\ (u,v)}{\partial u} \times \frac{\partial \vec{x_j}\ (u,v)}{\partial v} \| \tag{13}$$

where

$$\vec{l_{ij}} = \vec{x_j}\ (u,v) - \vec{x_i}\ (s,t) \tag{14}$$

We can see this equation is very similar to radiosity integral equation. Equation (9) becomes

$$I_i(s,t,\bullet) = \sum_{j \in SS} \int \int \rho_{ij}(s,t,u,v,\bullet) K_{ij}(s,t,u,v) \cdot I_{ij}(s,t,u,v) dudv \tag{15}$$

In $I_i(s,t,\bullet)$, (s,t) determines the location of the chosen point. In $\rho_{ij}(s,t,u,v,\bullet)$, (s,t) and (u,v) determine $\vec{l_{ij}}$ in (14), therefore the incident direction.

Details about the first pass are given below.

3.2. Global Iterations

We want to combine (11) into progressive radiosity. In order to make further adaptive meshing convenient, we initially triangulate the parametric domain of each surface. Accordingly, each surface is subdivided into a number of triangular curved patches. The final triangular mesh will be interpolated in the second pass to generate the final image. In each iteration, one diffuse shooting patch is chosen. One simple way to combine (11) is to compute the irradiance contributed by specular transfers for each receiving patch after the normal computation for direct diffuse illumination in each iteration. From Figure 2, we can see when we integrate over mirror M, only a small part S'' has nonzero radiance if there is only one patch shooting. In order to increase accuracy, we must detect the boundary of S'', which means subdividing M. And this kind of subdivision is different for different shooting patches. So it is not efficient.

We must change our notion of shooting patch. That is, an ideal specular surface can also be an independent shooting surface. We say a *diffuse iteration* if the shooting surface is diffuse and a *specular iteration* if it is specular. The unshot radiosity of a specular surface is distributed in the whole environment. A new variable, called *unshot_specular_rad*, can be created on each diffuse patch or vertex. In a diffuse iteration, the shooting energy is accumulated into this variable because it is not specularly transferred immediately. When a specular surface M is shooting and a chosen vertex P is receiving, for a sample point Q on M, ray trace $\vec{PQ}$ and suppose some surface with diffuse component is hit. The value of *unshot_specular_rad* at the hit point is retrieved and used to evaluate the virtual radiance at Q. In this way, any point on M has a virtual radiance for each receiving point, and specular surfaces are treated just like diffuse patches. But all specular surfaces must shoot simultaneously or one immediately after another; otherwise, each diffuse patch must hold distinct variables like *unshot_specular_rad* for each specular surface. That is not space efficient. Figure 4 shows the algorithm for global iteration where *rad* and *unshot_rad* are variables to store total and unshot radiosity in conventional progressive radiosity.

This approach decouples specular transfers from diffuse iterations and only has a small number of additional specular iterations in order to compute diffuse-specular-diffuse transport. So it is very efficient.

```
while( Not converged )
 if(the total flux obtained from unshot_specular_rad
     is not large enough)
 begin
  pick up the diffuse patch with
  maximum unshot_rad as the shooting patch;
  for each receiving diffuse patch or vertex
  begin
   compute received radiosity drad;
   rad=rad+drad;
   unshot_rad=unshot_rad+drad;
   unshot_specular_rad+=drad;
  end
  clear unshot_rad of the shooting patch;
 end
 else
 begin
  for each specular surface x_i, i ∈ SS
   for each receiving diffuse patch or vertex
   begin
    compute radiostiy drad received from x_i;
    rad=rad+drad;
    unshot_rad=unshot_rad+drad;
    tmp_specular_rad+=drad;
   end
  for each diffuse patch
  begin
   unshot_specular_rad=tmp_specular_rad;
   tmp_specular_rad=0;
  end
 end
```

Figure 4: *The iterative algorithm*

3.3. Approximation of the Integral

To obtain the value of integral in (11), we use quadtree partition to subdivide the parametric domain of each specular surface into a number of subdomains. (11) becomes

$$B_i(s,t) = \sum_{j\in SS}\sum_{k=1}^{N_j}\int\int_{D_k^j}\rho_i(s,t)K_{ij}(s,t,u,v) \cdot I_{ij}(s,t,u,v)\,du\,dv \qquad (16)$$

Then some quadrature rule is applied over each subdomain D_k^j to evaluate the integrals in (16). Each sample value of $I_{ij}(s,t,u,v)$ is obtained by wavefront tracing. And at each sample point on a subdomain, there is a corresponding ray tree which is set up during recursive wavefront tracing. Each subdomain can also be recursively subdivided into four smaller subdomains under some criteria. The criteria that can be used to guide subdivision are

1)whether the solid angle at the receiving vertex spread by the curved patch corresponding to current subdomain is small enough;

2)whether the normal of the curved patch corresponding to current subdomain does not vary too much;

3)whether the patches hit by the first a few levels of two adjacent ray trees match or are adjacent to each other;

4)whether
the quadrature over current subdomain achieves a specified accuracy;

5)whether wavefront tracing over current subdomain missed some important diffuse patches, such as those bright area light sources.

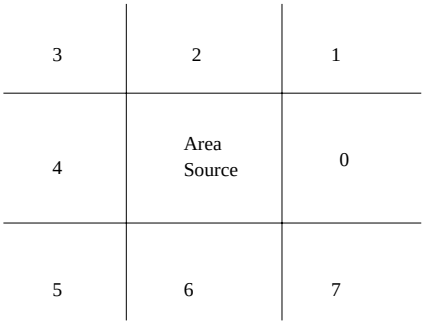


Figure 5: *Regions around a rectangular area light source*

For the last criterion, how can we know something is missing ? We can set up some counters for each such area source. For a rectangular light source (Figure 5), there are eight regions around it and each region is given a corresponding counter. Once a ray hits some region, the counter of that region is incremented. Finally, we get a subset *NZ* of these regions whose counter is nonzero. If the area source is not hit by rays sampled over current subdomain, there are three cases:

i)if $NZ \subseteq \{1,2,3\}$ or $\{3,4,5\}$ or $\{5,6,7\}$ or $\{7,0,1\}$, we know wavefront tracing over current subdomain almost has no chance to hit this area source;

ii)if $NZ \supseteq \{0,4\}$ or $\{2,6\}$, we know it is most likely for some ray to hit this area source if we continue subdividing current subdomain;

iii)otherwise, we are not sure and may wish to subdivide one more time to get more information.

Of course, the regions in Figure 5 can be subdivided and more reliable test can be resulted in. For curved area sources, we can fit a simple polygon or polytope to its boundary and do some similar tests. Some sample Fermat Paths[17] between some sample points on the area source and current receiving vertex are also useful information telling the region where subdivision should be done.

If current parametric subdomain is small enough, but there is still some criterion unsatisfied, we can disturb the current receiving vertex and repeat the evaluation of (16) at some

point very close to the previous one, or we can switch to stochastic sampling over current subdomain. For stochastic sampling, given a distribution function for those sample points, the weight of each sample value is reciprocal to the value of distribution function at that sample point. This kind of sampling is much easier than those in light ray-tracing since it is over a square parametric subdomain.

3.4. Recursive Wavefront Tracing

If all reflectors in the environment are opaque and specular reflectors have no diffuse component, ray trees degenerate into linear trees. The computation of intensity and area factor is straightforward by propagating two wavefronts along the ray path in opposite directions. A recursive function can be written to do this task. The evaluation of area factor is carried out as the function traces the ray path from one object to another, and is done when a diffuse surface is hit. Then the function returns along the previous ray path and the evaluation of intensity factor is carried out.

If some reflectors have both specular and diffuse components, a user stack should be used to save intermediate results of each recursive call. During wavefront tracing, any intermediate reflector with diffuse component has partial contribution to the virtual radiance of the first reflector. Once such a reflector is encountered, the function traverses the user stack from top to bottom to compute its intensity factor and partial contribution to the final virtual radiance.

If surfaces with both specular reflection and transmission are present, ray trees are no longer linear. Each path from root to leaf should be traversed and the use of user stack is still inavoidable.

3.5. Meshing

Discontinuity meshing[15] and backprojection[6] can work very well for polygonal scene. Other meshing techniques can hardly compete with them. But since we have curved surfaces, at least curved specular surfaces, we should come up with something else to catch shadow boundaries and caustics. Two kinds of adaptive meshing used in our algorithm. Quadtree partition is used for specular component because it is convenient to perform quadrature. Triangular mesh is used for diffuse component because it is easy to interpolate and eliminate T-intersection. We use linear interpolation and it will be used as a criterion for mesh refinement. If a surface has both diffuse and specular component, there will be two meshes on its parametric domain. We elaborate on triangular meshing below.

Initially each surface is subdivided into triangles of approximately the same size. This initial mesh will be refined in each iteration. In each iteration, after shooting surface(s)—diffuse or specular— are chosen, the following algorithm is executed for mesh refinement. There are two constants in the algorithm. One is the minimum edge length. If an edge in the mesh has length less than or equal to this constant, no more subdivision and test will be done on this edge. The other is for the shape of triangles in the mesh. If the ratio of the sum of the two smaller edge length to the largest one is less than this constant, the longest edge should be subdivided first before any other subdivision of the triangle. We use 3D object-space Euclidean distance, not the distance in parametric domain, although all subdivision are made there.

Meshing algorithm

1)Compute incremental radiosity for each vertex in the mesh;

2)For each edge in the mesh with length larger than a threshold, search for the break point on this edge by the following Break-point searching algorithm. If there exists such a break point, subdivide the two triangles sharing this edge(Figure 6) into four smaller ones at the break point(if any of these triangles is in bad shape, subdivide its longest edge first), and then recursively check the newly generated edges.

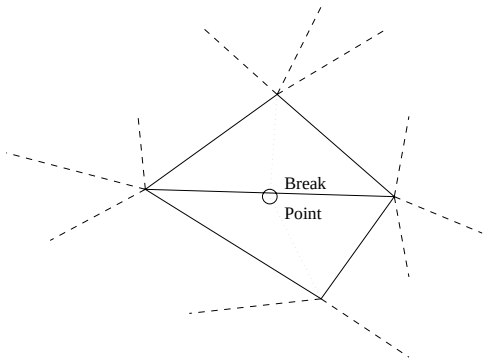


Figure 6: *Subdivision of an edge in the triangular mesh*

Break-point searching algorithm

1)If only one vertex of current edge is in the umbra of current shooting patch(otherwise, go to step 2), use binary search on the edge for a point lying very close to the boundary of the shadow but not in shadow and test linearity at this point as in step 2); if the test fails, return this point as the break point; otherwise

2)choose some sample points on current edge—the number of sample points is determined by its length; compute the radiosity at each sample point by nemerical integration and linear interpolation, respectively; if their difference at any sample point is larger than a threshold, use *Fibonacci* search, a kind of one dimensional search to find the maxima(or minima), for the break point where the above-mentioned difference is maximum and return this point; otherwise, no break point.

To reduce the number of patches, binary search and Fibonacci search are used to find the right position for subdivision in the above algorithm. In step 2) of the meshing algorithm, the edges can be checked in the order of decreasing length. This meshing algorithm can be easily extended to higher order interpolation. We can also incorporate two dimensional search within each triangle for break points. But that will need much more time. A diffuse shooting patch is always picked up from the initial mesh. So it may include many smaller patches. To accelerate the algorithm, we can fit a low order polynomial to the radiosity distribution on the shooting patch. We can also use a smaller minimum edge length in the meshing algorithm if the shooting patch is a part of a bright light source and a larger one for other shooting patches.

So far we have presented the details of the solution for (11). Except for adaptive meshing algorithm which should be further investigated for directonal diffuse component, the solution for (15) is similar because a global illumination algorithm very similar to progresive radiosity was introduced by Sillion et al.[25], who used spherical harmonics to store directional radiance distributions, and we can easily combine our algorithm for specular component into that one.

4. Experimental Results

Our specular transfer equations are correct by derivation. But we still did some experiments to show their effectiveness. In one situation, we tried to get the radiosity distribution when the shooting surface and the receiving surface are in two different media where the boundary between the media is a curved surface defined below

$$\begin{cases} x = x_0 + lx(s-0.5) \\ y = -h \cdot cos(3\pi(s-0.5))cos(3\pi(t-0.5)) \\ z = z_0 + lz(t-0.5) \end{cases} \qquad (17)$$

On this surface, there are five local minima at $(\frac{1}{2},\frac{1}{2})$, $(\frac{1}{6},\frac{1}{6})$,$(\frac{1}{6},\frac{5}{6})$, $(\frac{5}{6},\frac{1}{6})$,$(\frac{5}{6},\frac{5}{6})$ and four local maxima at $(\frac{1}{2},\frac{1}{6})$, $(\frac{1}{2},\frac{5}{6})$,$(\frac{1}{6},\frac{1}{2})$, $(\frac{5}{6},\frac{1}{2})$. Gaussian curvature is positive at these elliptic regions.

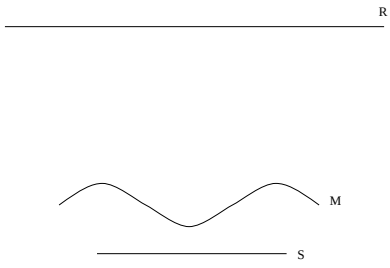


Figure 7: *One testing configuration*

Figure 7 gives a configuration we tested, where M is the surface defined by (17), S is a square light source and R is a square diffuse receiver. M only has ideal transmission. The refractive index of the medium under M is about 1.4 and that of the medium above M is 1.0.

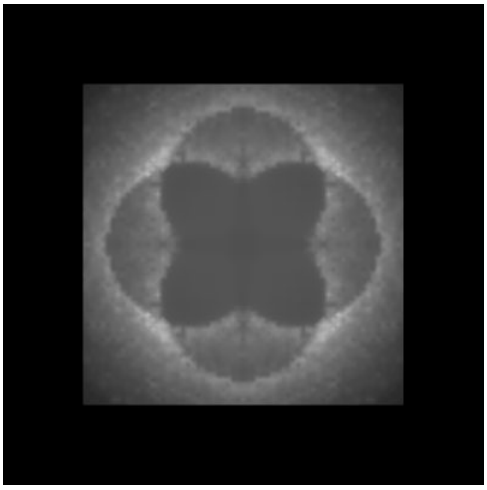


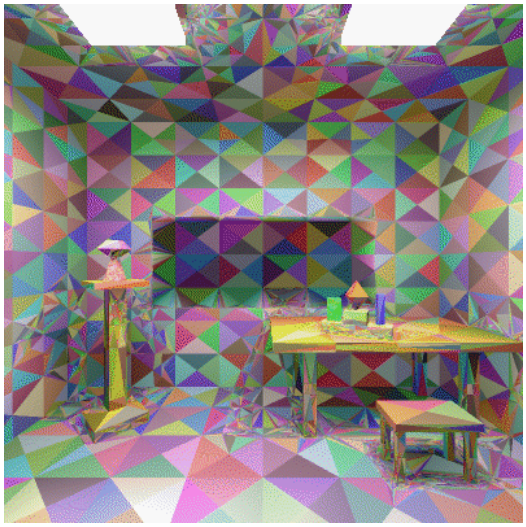
Figure 8: *Radiosity distribution on the receiver for the configuration shown in Figure 7*

Figure 8 shows the picture of the radiosity distribution on the receiver for this configuration and for $lx = lz = 100$ and $h = 5$ which are coefficients in (17). In this experiment, we did not use stochastic sampling. When the quadrature over some parametric subdomain is hard to converge, we simply disturb the receiving vertex to avoid division by a very small number. This is because intensity factor may become very large near some point where the Gaussian curvature of the wavefront tends to infinity. That's why there is some noise in Figure 8. Although this result is for a hypothetical situation, it shows the power of our method which is not only useful in realistic image synthesis, but also useful in visualizing special effects of ideal specular transfers, which might be interesting to mathematical physicists.

We have implemented our algorithm for combining wavefront tracing and radiosity. Among the criteria in section 3.3, we implemented 1),4) and 5). We used octree space-partition to accelerate wavefront tracing, and also adaptive quadrature as in the paper by Yu et al.[30].

Figure 9(a) and 9(b) show two pictures for an indoor scene. There is a concave mirror before the wall facing us in Figure 9(a). Despite some overexposure of the film, we can still see bright regions on the floor and the desktop caused by this mirror. Figure 9(b) is the meshing result for 9(a). There are many tiny triangles around shadow boundaries and there is a dark area on the wall facing us, which is the part of the wall behind the mirror. The color of each triangle was assigned by adding a random component to the radiosity distribution. This figure shows the effectiveness of our techniques, such as integrating specular transfers into radiosity and meshing, introduced

(a)



(b)

Figure 9: *(a) An indoor scene with a concave mirror, (b) meshing result for (a)*

in section 3. There are no obvious effects of using wavefront tracing like those in Figure 8 because the concave mirror is a simple curved reflector.

We also examined the effectiveness of shooting specular and diffuse surfaces separately, which was introduced in Section 3.2, in several test scenes similar to that in Figure 9 on a Sparc-10 machine. We compared the running time of our method with jointly computing specularly transferred energy in each diffuse iteration. The speedup was fairly obvious. If we compute specularly transferred energy in each diffuse iteration, most of the time is spent there. But these specularly transferred energy has relatively small contribution to the whole environment. If we accumulate these energy for later specular shooting, we may save large amount of time.

5. Conclusions and Future Directions

Theory and algorithms were presented for the fundamental problem in graphics: simulation of illumination from area light sources via curved specular surfaces, including ideal reflection and transmission. Three important concepts–intensity factor, area factor and virtual radiance–were introduced. They helped to give general equations for specular transfers. A solution for parametric environment was proposed. Wavefront tracing and numerical integration were used in the solution, along with a meshing technique on parametric domains. We also let a specular surface shoot just as a normal diffuse patch. As a side product, a new version of ray tracing was also proposed.

One problem is whether surface curvature should also be taken into account for directional diffuse transfers. Efforts should also be made to find out situations where the product of intensity and area factor becomes one, in which case the computation for intensity and area factor is unnecessary.

A few other problems should also be investigated, such as the possibility of combining wavefront tracing and hierarchical method[1], the possibility of solving (8) and (9) for surfaces defined by implicit functions, and faster numerical integration techniques suitable for (11) and (15).

References

1. L. Aupperle, and P. Hanrahan, "A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection", *Computer Graphics Proceedings, Annual Conference Series*, pp.155–162 (1993).
2. S.E. Chen, H.E. Rushmeier, G. Miller, and D. Turner, "A Progressive Multi-Pass Method for Global Illumination", *Computer Graphics*, **25(4)**, pp.165–174 (1991).
3. M.F. Cohen, D.P. Greenberg, D.S. Immel, and P.J. Brock, "An Efficient Radiosity Approach for Realistic Image Synthesis", *IEEE CG&A*,**6(2)**, pp.26–35 (1986).
4. M.F. Cohen, S.E. Chen, J.R. Wallace, and D.P. Greenberg, "A Progressive Refinement Approach to Fast Radiosity Image Generation", *Computer Graphics*,**22(4)**, pp.75–84 (1988).
5. S. Collins, "Reconstruction of Illumination form Area Luminaires", *Proc. Eurographics Workshop on Rendering'95*, pp.274–283.
6. G. Drettakis, and E. Fiume, "A Fast Shadow Algorithm for Area Light Sources Using Backprojection", *Computer Graphics Proceedings, Annual Conference Series*, pp.223–230 (1994).

7. G. Elber, "Low Cost Illumination Computation using an Approximation of Light Wavefronts", *Computer Graphics Proceedings, Annual Conference Series*, pp.335–342 (1994).

8. S.J. Gortler, R. Grzeszczuk, R. Szeliski, and M.F. Cohen, " The Lumigraph", *Computer Graphics Proceedings, Annual Conference Series*, pp.43–54 (1996).

9. P. Hanrahan, D. Salzman, and L. Aupperle, "A Rapid Hierarchical Radiosity Algorithm", *Computer Graphics*, **25(4)**, pp.197–206 (1991).

10. X.D. He, K.E. Torrance, F. Sillion, and D.P. Greenberg, "A Comprehensive Physical Model for Light Reflection", *Computer Graphics*, **25(4)**, pp.175–186 (1991).

11. P.S. Heckbert, "Adaptive Radiosity Textures for Bidirectional Ray Tracing", *Computer Graphics*, **24(4)**, pp.145–154 (1990).

12. H.W. Jensen, and N.J. Christensen, "Photon Maps in Bidirectional Monte Carlo Ray Tracing of Complex Objects", *Computer&Graphics*, **19(2)**, pp.215–224 (1995).

13. J.T. Kajiya, "The Rendering Equation", *Computer Graphics*, **20(4)**, pp.143–149 (1986).

14. M. Levoy, and P. Hanrahan, " Light Field Rendering", *Computer Graphics Proceedings, Annual Conference Series*, pp.31–42 (1996).

15. D. Lischinski, F. Tampieri, and D.P. Greenberg, "Discontinuity Meshing for Accurate Radiosity", *IEEE CG&A*, **12(6)**, pp.25–39 (1992).

16. D. Lischinski, F. Tampieri, and D.P. Greenberg, "Combining Hierarchical Radiosity and Discontinuity Meshing", *Computer Graphics Proceedings, Annual Conference Series*, pp.199–208 (1993).

17. D. Mitchell, and P. Hanrahan, "Illumination from Curved Reflectors", *Computer Graphics*, **26(2)**, pp.283–291 (1992).

18. H.P. Moravec, "3D Graphics and the Wave Theory", *Computer Graphics*, **15(3)**, pp.289–296 (1981).

19. E. Nakamae, and K. Tadamura, "Photorealism in Computer Graphics— Past and Present", *Computer&Graphics*, **19(1)**, pp.119–130 (1995).

20. S.N. Pattanaik, and S.P. Mudur, "Adjoint Equations and Random Walks for Illumination Computation", *ACM Trans. on Graphics*, **14(1)**, pp.77–102 (1995).

21. M. Paulin, and J. Jessel, "Adaptive mesh generation for progressive radiosity: A ray-tracing based algorithm," *Proc. Eurographics'94*, pp.C421–C432.

22. H.E. Rushmeier, and K.E. Torrance, "Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials", *ACM Trans. on Graphics*, **9(1)**, pp.1–27 (1990).

23. M. Shao, Q. Peng, and Y. Liang, "A New Radiosity Approach by Procedural Refinements for Realistic Image Synthesis", *Computer Graphics*, **22(4)**, pp.93–101 (1988).

24. F. Sillion, and C. Puech, "A General Two-Pass Method Integrating Specular and Diffuse Reflection", *Computer Graphics*, **23(3)**, pp.335–344 (1989).

25. F.X. Sillion, J.R. Arvo, S.H. Westin, and D.P.Greenberg, " A Global Illumination Solution for General Reflectance Distributions", *Computer Graphics*, **25(4)**, pp.187–196 (1991).

26. O.N. Stavroudis, *The optics of Ray, Wavefronts, and Caustics*, Academic (1972).

27. J.R. Wallace, M.F. Cohen, and D.P. Greenberg," A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods", *Computer Graphics*, **21(4)**, pp.311–320 (1987).

28. J.R. Wallace, K.A. Elmquist, and E.A. Haines, "A Ray Tracing Algorithm for Progressive Radiosity", *Computer Graphics*, **23(3)**, pp.315–324 (1989).

29. G.J. Ward, F.M. Rubinstein, and R.D. Clear," A ray tracing solution for diffuse interreflection", *Computer Graphics*, **22(4)**, pp.85–92 (1988).

30. Y.Yu, and Q.Peng, "Multiresolution B-spline Radiosity", *Proc. Eurographics'95*, pp.C285–C298.