A Global Illumination Solution
for General Reflectance Distributions

François X. Sillion
James R. Arvo
Stephen H. Westin
Donald P. Greenberg

Program of Computer Graphics
Cornell University
Ithaca, NY 14853

Abstract

A general light transfer simulation algorithm for environments composed of materials with arbitrary reflectance functions is presented. This algorithm removes the previous practical restriction to ideal specular and/or ideal diffuse environments, and supports complex physically based reflectance distributions. This is accomplished by extending previous two-pass ray-casting radiosity approaches to handle non-uniform intensity distributions, and resolving all possible energy transfers between sample points. An implementation is described based on a spherical harmonic decomposition for encoding both bidirectional reflectance distribution functions for materials, and directional intensity distributions for illuminated surfaces. The method compares favorably with experimental measurements.

CR Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; I.3.3 [**Computer Graphics**]: Picture/Image Generation.
Additional Keywords and Phrases : global illumination, BRDF, specular reflection, directional-diffuse, progressive radiosity, spherical harmonics.

1 Introduction

The simulation of global illumination is one of the major requirements for realistic 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*, or the transfer of color by reflection, can be observed in almost all indoor scenes. This paper presents a completely general algorithm designed to solve the global illumination problem for arbitrarily complex reflectance models.

Solution techniques for the simulation of complex light transfer mechanisms, where every point in the environment can potentially act as an illuminator for all other points, have thus far been quite limited. Two major paths have been explored. Light can be followed as it leaves the light sources and is propagated and reflected throughout the environment. For example, this approach is used by progressive

refinement radiosity algorithms, and can be characterized as *view-independent* shading [3]. Conversely, standard ray tracing [20] and its derivatives usually start from the eye and follow light paths in the reverse direction. It is therefore strongly *view-dependent*.

These approaches work well for certain types of reflective behaviors, such as ideal diffuse (radiosity), ideal specular (ray tracing), or combinations of these [18, 17, 9]. The actual reflectance distributions of most surfaces are far more complicated, exhibiting some directionality which must be taken into account for accurate simulation.

The approach presented here extends the progressive radiosity method to include arbitrary reflectance distributions. While previous algorithms incorporating general reflectances have relied upon a discrete set of directions [11, 1, 13], no such restriction is introduced here. This is accomplished by using continuous functions to encode the directional dependence of intensity distributions.

In the next section we discuss the applicability of *view-independent* and *view-dependent* approaches to the case of general reflectance distributions and introduce a classification of reflectance types into *ideal diffuse, ideal specular* and *directional diffuse* components. The third section is devoted to the presentation of a complete algorithm to solve the general problem. Two specific issues are then detailed : treatment of ideal specular reflection in Section 4, and storage of directional diffuse contributions in Section 5.

2 Algorithmic choices for a general solution

The goal of this research is to develop a method for the simulation of global illumination that is general enough to provide accurate solutions for scenes incorporating any reflectance distribution. The problems encountered in devising a completely general algorithm are reviewed below, together with some of their design implications.

2.1 General reflectance distributions

The reflective properties of a surface are generally described by means of a *bidirectional reflectance distribution function* (BRDF), which is defined as the ratio of the reflected radiance in a given *outgoing direction* to the *incoming energy flux* (per unit area) in another direction. A similar quantity is defined for transmission. For the sake of clarity, we will refer only to reflection in this paper, although the algorithm is equally applicable to transmission. The two well-understood limiting cases are *ideal diffuse* and *ideal specular* reflection.

- An ideal diffuse reflector has a constant BRDF, that is, the scattered intensity is the same in all directions. Diffuse reflection can thus be fully described by a single scalar value.

- An ideal specular reflector has a Dirac delta function as its BRDF, where the only direction in which there is non-zero scattering is the mirrored direction. The relevant quantity to describe specular reflection is the ratio of the outgoing intensity in the specular direction to the incoming intensity, or *specular reflectance*.

Most materials have BRDFs that are not this simple, exhibiting a more elaborate directionality. Recent work on light reflection models has shown that different physical processes contribute to different parts of a BRDF [8], and the term *directional diffuse* has been introduced to describe the general BRDF excluding its ideal specular component. (Figure 1).
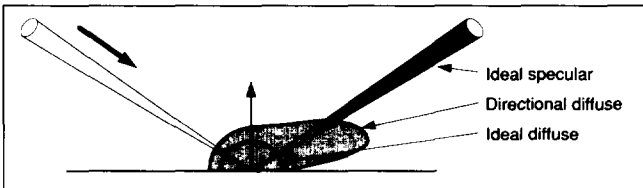

Figure 1: Different components of a general BRDF.

The directional diffuse component of a BRDF is a function of many variables, including surface finish (roughness), wavelength, and the electrical properties of the material. This produces a great variety of behaviors, all of which must be correctly simulated. The algorithm presented below is capable of incorporating both arbitrary directional diffuse and ideal specular reflection into a global solution.

2.2 View-independence vs. view-dependence

View-independent methods in general require the storage of illumination information on the surfaces, both for the purpose of the illumination computation and for use by a final *view-dependent* display algorithm. In the case of diffuse surfaces, storing a single radiosity value per wavelength channel at each sample point is sufficient, resulting in reasonable storage demands. In the same spirit, Immel *et al.* stored the directional information regarding the reflected intensity at each point, using a discrete set of directions [11]. If, however, the distribution of emitted or reflected light is sharply directional, as is the case with specular surfaces, storage becomes unmanageable if accuracy is to be maintained.

On the *view-dependent* side, distribution ray tracing [5] uses brute force, firing many reflection rays to simulate complex BRDFs, while path tracing [12] follows many paths through the scene to obtain a statistically reliable estimate. Here again, the property that made standard ray tracing [20] computationally tractable disappears (namely the restriction to ideal specular reflection that limits the number of rays), as rays must be fired towards all potential illuminators.

The method presented below combines elements of both strategies into a two-pass algorithm. The first pass computes a view-independent solution for the directional diffuse distribution of light, including the effect of intermediate specular reflections, and the second pass supplies the view-dependent ideal specular effects. This partitioning of reflectance behaviors resembles earlier two-pass approaches, but now accounts for all possible transport chains and incorporates arbitrary reflectance distributions, not only the extreme cases of ideal diffuse and ideal specular.

In the following discussion we use the vocabulary of radiosity-style algorithms for two reasons. One reason is the energy consistency of the radiosity method: a physical consideration necessary
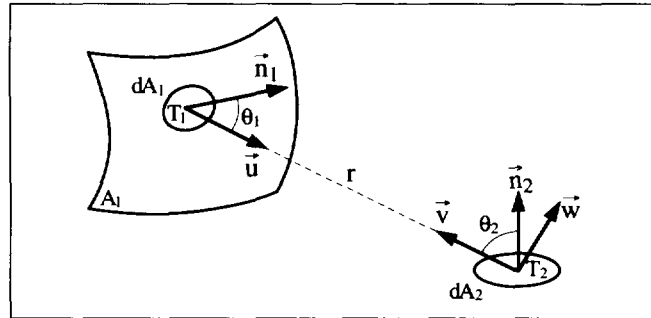

Figure 2: Energy transfer between a patch and a differential area

to obtain accurate simulations. The second reason is the appeal of the progressive refinement paradigm [3], where useful intermediate results can be obtained early in the computation.

The notion of radiosity is extended to include the directional diffuse part of the light reflected at a given point. Because it does not include specularly reflected light, this *intensity distribution* is fairly smooth and thus can be stored at reasonable cost (See section 5). The ideal specular distribution of light is sharply discontinuous and is too costly to store on the surfaces; it is properly computed "on the fly" to resolve *specular to directional diffuse* transfers.

By using ray-casting, which has proven to be an effective sampling method to evaluate light transfers, all sample points in the environment are considered [19]. Furthermore, the use of ray casting imposes no restriction on the geometry of the environment and allows every illuminator to be sampled adaptively.

The algorithm proceeds by successive steps similar to progressive radiosity "shots", but directional intensity distributions are continuously maintained on the surfaces instead of scalar radiosity values. The ideal specular contributions to the energy transfers are propagated immediately so they need never be stored.

2.3 Energy transfers for non-diffuse surfaces

Traditional radiosity methods assume an ideal diffuse behavior on all the surfaces, and express the transfers between surfaces by means of a *form factor* [7]. For the general case the amount of light reflected from a point can be expressed as follows.

Let us denote by $I(T_1, \vec{u})$ the *intensity* (or *radiance*, expressed in Watts per unit solid angle per unit projected area) leaving a surface at point T_1, in the direction of the unit vector $\vec{u}$ (Figure 2). The energy d^2E emitted by a differential surface area dA_1 around T_1 in the direction $\vec{u}$ and falling on a differential surface dA_2 around point T_2 is given by :

$$d^2E = I(T_1, \vec{u}) \underbrace{(dA_1 \cos \theta_1)}_{\text{projected area}} \underbrace{\left(\frac{dA_2 \cos \theta_2}{r^2} \right)}_{\text{solid angle as seen from } T_1} \quad (1)$$

$$= I(T_1, \vec{u}) \frac{\cos \theta_1 \cos \theta_2}{r^2} dA_2 \, dA_1 \quad (2)$$

This energy is scattered by the surface at T_2 in all directions. By definition of the BRDF ρ_2 at T_2, the intensity leaving T_2 in the direction $\vec{w}$, due to the incident light from dA_1, is given by :

$$dI(T_2, \vec{w}) = \rho_2(\vec{v}, \vec{w}) \frac{d^2E}{dA_2} \quad (3)$$

where $\vec{v} = -\vec{u}$ is the unit vector pointing from T_2 to T_1.

To evaluate the total intensity leaving T_2 in direction $\vec{w}$, due to the reflection of light originating from a finite area A_1, Equation (2) must be integrated across A_1 giving :

$$I(T_2, \vec{w}) = \int_{A_1} I(T_1, \vec{u}) \frac{\cos \theta_1 \cos \theta_2}{r^2} \rho_2(\vec{v}, \vec{w}) dA_1 \qquad (4)$$

Equation (4) represents the effect of the light emitted by a particular surface on the light that is scattered around a point on another surface. It describes the elementary *shooting* operation of the radiosity method. The traditional radiosity method simplifies Equation (4) in two ways : first, the BRDF ρ is assumed to be diffuse, which makes it a constant independent of both $\vec{v}$ and $\vec{w}$, and can thus be moved out of the integral. The diffuse assumption also makes $I(T_1, \vec{u})$ independent of $\vec{u}$. Second, the radiosity $\pi I(T_1)$ is assumed to be constant across the surface of the patch. The intensity term can thus be moved out of the integral, which then becomes purely geometric and is called the *differential form factor*.

If more general BRDFs are considered for the surface at T_2, however, the entire integrand must be considered. The next section explains how a form factor computation algorithm is adapted for that purpose.

3 General solution for arbitrary reflectances

A detailed description of the algorithm is presented below. The first pass, or solution pass, is very similar to progressive radiosity, and the implementation is a straightforward modification of an existing radiosity program. The second pass employs a simple ray tracer to retrieve the directional intensity information stored on the surfaces.

A central assumption of the method is that a directional intensity distribution $I(\vec{u})$ can be stored and accessed at each vertex of the environment. We discuss this topic further in section 5, where an efficient storage scheme is presented.

The method is explained here in terms of a meshed environment, composed of patches and elements [4], but it could be applied to radiosity textures [9] as well, if directional distributions are stored in the texture. The second (view-dependent) pass is described first, as it is a straightforward application of ray tracing.

3.1 Second pass

Once the view-independent solution has been computed in the first pass, a simple ray tracing pass is used to supply the view-dependent portion and create the final image. When rays encounter surfaces with a *directional diffuse* component, the intensity leaving a surface is retrieved from the directional distributions computed and stored in the first pass. The intensity contributed by *ideal specular* reflection is obtained by recursively following reflected rays as in conventional ray tracing. Note that a *specular reflectance function* is used to attenuate the reflected rays instead of a simple "specular coefficient". This allows a precise treatment of specular reflection, where roughness effects as well as Fresnel reflection are properly accounted for [8].

3.2 First pass

The first phase of the computation is an extension of progressive radiosity, but directional distributions are used throughout the algorithm in place of diffuse radiosities. The basic *shooting* operation now uses the directional intensity distribution emitted by the shooting patch to update the directional intensity distributions of the receiving vertices according to Equation (4).

This equation can be rewritten as a function of the intensity distributions rather than scalar values. If $\rho_2(\vec{v}, \cdot)$ denotes the BRDF for

an incoming direction $\vec{v}$ as a function of the outgoing direction, and $I(T, \cdot)$ denotes the intensity distribution at point T, then the effect of shooting from T_1 to T_2 is

$$I(T_2, \cdot) = \int_{A_1} I(T_1, \vec{u}) \frac{\cos \theta_1 \cos \theta_2}{r^2} \rho_2(\vec{v}, \cdot) dA_1 \qquad (5)$$

The algorithm presented below follows Equation (5) and decomposes the integral into a discrete sum. This is similar to the form factor computation algorithm of Wallace *et al.*[19], but modified to sum complete directional intensity distributions.

Approximation of the integral

To obtain the reflected intensity distribution given by (5), we follow the computation of the *area-to-differential-area* form factor used in [19].

Patch A_1 is broken into a number of smaller pieces according to any given sampling scheme, and a contribution (a scalar *delta-form-factor* in the diffuse radiosity case, a directional distribution in our case) is computed for each piece. A variety of sampling strategies is available, and this formulation is independent of the particular scheme chosen. For N samples, the total integral is expressed as :

$$I(T_2, \cdot) = \sum_{i=1}^{N} \delta I_i(T_2, \cdot) \qquad (6)$$

Given sample i, with area δA_i centered at T_i, the associated contribution to the integral in (4) could be crudely approximated by assuming the integrand constant, yielding :

$$\delta I_i(T_2, \cdot) = I(T_i, \vec{u}_i) \frac{\cos \theta_{1_i} \cos \theta_{2_i}}{r_i^2} \rho_2(\vec{v}_i, \cdot) \delta A_i \qquad (7)$$

To avoid possible singularities when T_1 and T_2 are close together, we treat piece i as a finite area, and use the approximation of a disk-shaped area as in [19]. This amounts to assuming that the emitted intensity does not vary significantly over the area of the piece, which is a common assumption of the radiosity formulation. The contribution of piece i is then :

$$\delta I_i(T_2, \cdot) = I(T_i, \vec{u}_i) \cos \theta_{1_i} \cos \theta_{2_i} \frac{\pi \delta A_i}{\delta A_i + \pi r_i^2} \rho_2(\vec{v}_i, \cdot) \qquad (8)$$

Introducing the incident energy flux (Watts per unit area) incident on point T_2 from piece i

$$\delta \Phi_i = I(\vec{u}_i) \frac{\cos \theta_{1_i} \cos \theta_{2_i} \pi \delta A_i}{\delta A_i + \pi r_i^2} = I(\vec{u}_i) \delta F_i \qquad (9)$$

Equation (5) can be conveniently rewritten as :

$$I(T_2, \cdot) = \sum_{i=1}^{N} \delta \Phi_i \rho_2(\vec{v}_i, \cdot) \qquad (10)$$

It is apparent from equation (10) that I is simply a weighted sum of the BRDF at T_2 over a set of incident directions. Each energy flux term, $\delta \Phi_i$, is the product of the intensity leaving a sample point on the shooting patch in the direction of the receiving vertex and the *delta-form-factor* for that sample point.
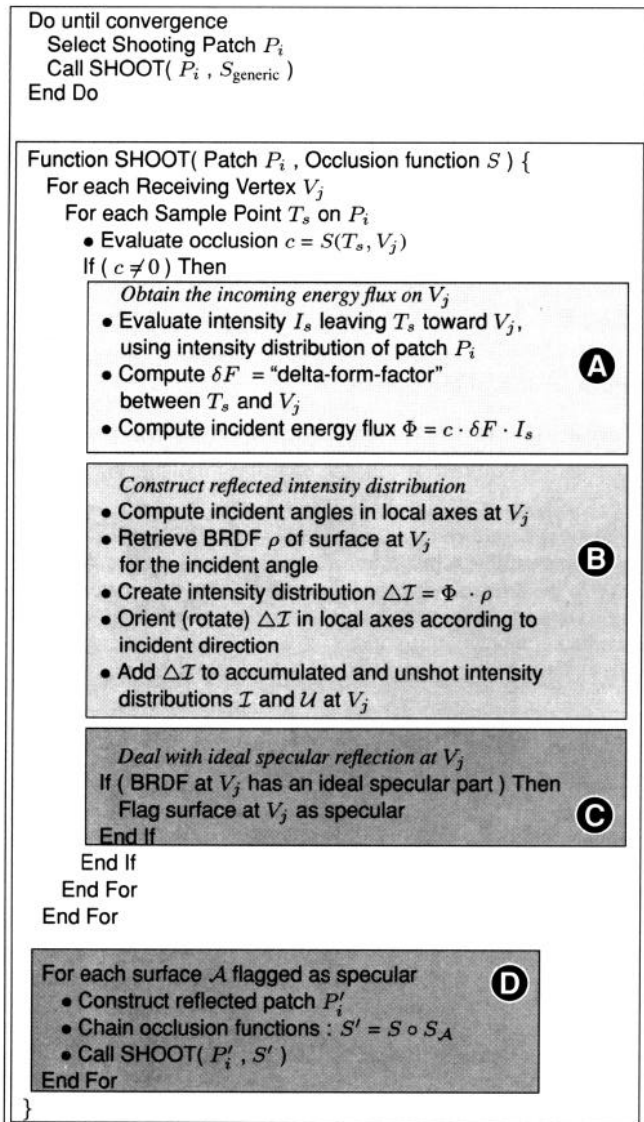
```
Do until convergence
    Select Shooting Patch P_i
    Call SHOOT( P_i , S_generic )
End Do
```

```
Function SHOOT( Patch P_i , Occlusion function S ) {
    For each Receiving Vertex V_j
        For each Sample Point T_s on P_i
            • Evaluate occlusion c = S(T_s, V_j)
            If ( c ≠ 0 ) Then
```

> *Obtain the incoming energy flux on V_j*
> • Evaluate intensity I_s leaving T_s toward V_j,
>   using intensity distribution of patch P_i
> • Compute δF = "delta-form-factor"
>   between T_s and V_j
> • Compute incident energy flux $\Phi = c \cdot \delta F \cdot I_s$ **A**

> *Construct reflected intensity distribution*
> • Compute incident angles in local axes at V_j
> • Retrieve BRDF ρ of surface at V_j
>   for the incident angle
> • Create intensity distribution $\triangle \mathcal{I} = \Phi \cdot \rho$
> • Orient (rotate) $\triangle \mathcal{I}$ in local axes according to
>   incident direction
> • Add $\triangle \mathcal{I}$ to accumulated and unshot intensity
>   distributions $\mathcal{I}$ and $\mathcal{U}$ at V_j **B**

> *Deal with ideal specular reflection at V_j*
> If (BRDF at V_j has an ideal specular part) Then
>   Flag surface at V_j as specular
> End If   **C**

```
            End If
        End For
    End For
```

> For each surface $\mathcal{A}$ flagged as specular   **D**
> • Construct reflected patch P_i'
> • Chain occlusion functions : $S' = S \circ S_{\mathcal{A}}$
> • Call SHOOT(P_i' , S')
> End For

```
}
```

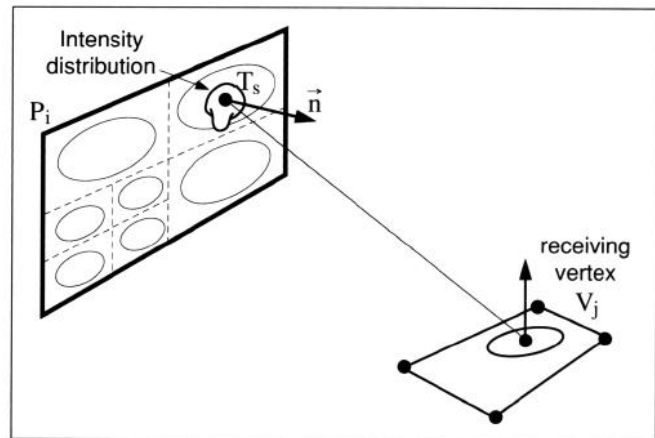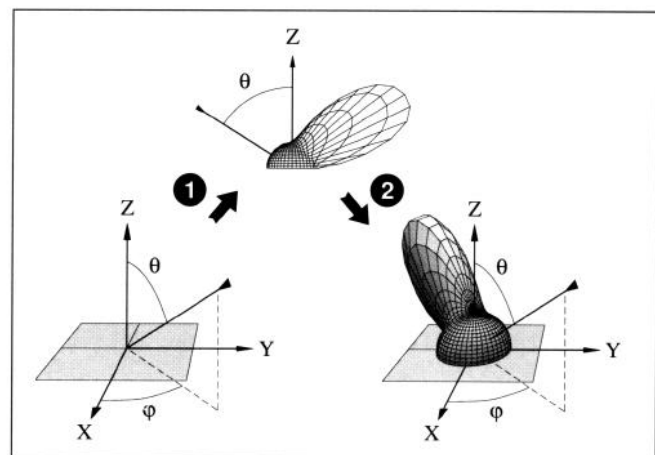Figure 3: Algorithm for first pass. Step D is explained in section 4

Figure 4: Computation of the energy flux.

can "see" each other, the intensity leaving the shooting patch is obtained from the stored directional distribution of the patch, and a *delta-form-factor* δF (geometric attenuation term) is computed (Figure 3-A). The desired energy flux is the product of the intensity, the *delta-form-factor* and the attenuation given by the occlusion function (Figure 4).

Figure 5: Orientation of the BRDF. (1) : obtaining the BRDF according to θ. (2) : rotating the BRDF according to ϕ.

Discussion of the algorithm

The *shooting* operation that propagates the accumulated energy of a patch into the environment is presented as a pseudocode subroutine (called SHOOT) in Figure 3. Given a shooting patch P_i and a receiving sample point (vertex V_j), the following operations are needed to update the directional intensity distribution of the vertex :

Sample points are selected on the shooting patch according to a sampling algorithm. Our implementation uses an adaptive sampling technique where the number and location of the sample points depend on the results obtained from previous samples [19]. For each sample point, a contribution is added to the reflected intensity distribution of the vertex.

The first task of the algorithm is to compute the incident energy flux on the vertex, which is used to weight the BRDF as in equation (10). This involves a visibility determination accomplished by the *occlusion function S*. In the simple cases where no ideal specular reflection is present, this function simply returns 0 or 1 to encode occlusion between the sample point and the vertex The more complex cases are explained in Section 4. If the two points

The second step is to compute the contribution of the current sample point to the intensity distribution (Figure 3-B). We start by obtaining a directional distribution representing $\rho_2(\vec{v_i}, \cdot)$, that is, the BRDF for the given incident direction. This is retrieved by means of the storage method described in section 5. The BRDF is then *scaled* by the energy flux value $\delta\Phi_i$, which results in the distribution of reflected intensity $\triangle\mathcal{I}$ due to the current sample point. If an isotropic BRDF is used, the directional distribution depends only on the incident elevation angle θ, and is obtained in a canonical coordinate system : it must be *rotated* to be properly aligned with the incident azimuth angle ϕ in the local coordinate system of the receiver (Figure 5). Finally this contribution is added to the *unshot* intensity distribution and the *accumulated* intensity distribution, much as in traditional progressive radiosity (Figure 6).

Previous work has shown that a complete treatment of light transfers requires exchanges incorporating different modes of reflection. Our shooting operation also includes a complete treatment of *ideal specular* reflection, so that the effect of specular reflection on the
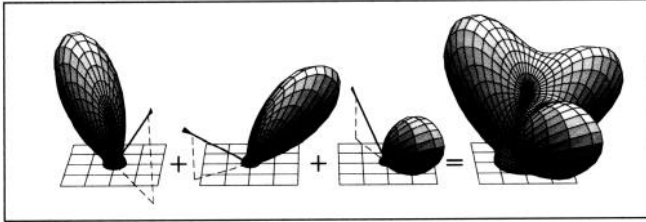
Figure 6: Addition of directional distributions.

intensity distributions is completely evaluated, but no "specular intensity" is stored (see section 2). Section 4 explains this part of the algorithm in more detail.

4 Ideal specular transfers

As explained in Section 2, specularly reflected light is not stored in the directional distributions on the specular reflector. Instead it is immediately propagated to other surfaces where part of it will be stored in a directional distribution, and part may again be specularly reflected to other surfaces.

Our implementation is an adaptation of the "image method" [16] to the ray-traced form factor idea : if the specular surfaces are planar, one can simply reflect the shooting patch across the surface, and shoot light from this "virtual patch" to all receiving vertices (Figure 7). Note that the direction for each shot is chosen deterministically based on the position of a vertex; this is not a Monte Carlo sampling technique.
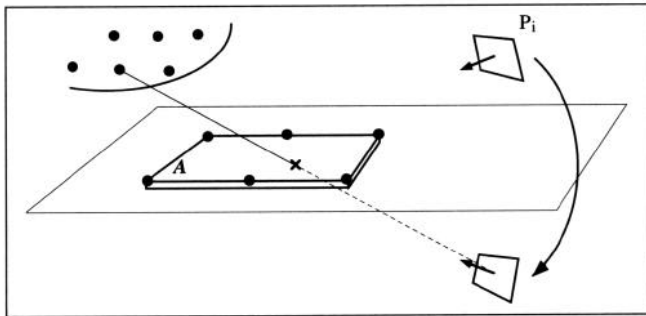

Figure 7: Reflecting the shooting patch on a specular surface. Original vertices from the environment mesh are shown as dots.

This method has several important benefits :

- In contrast to the original "image method", where the entire environment had to be reflected into a virtual environment, only the shooting patch need be reflected.

- The evaluation of one specular reflection on a given specular surface can be implemented as a normal shooting step with a slightly modified occlusion testing routine. Furthermore, since we are shooting directly to vertices, it is possible to restrict the expensive occlusion testing operation to the portions of the environment that can potentially receive reflected light, using a technique similar to a *shadow volume* [14].

This allows us to retain benefits of the normal ray-casting method, such as adaptive meshing based on the results of a shot, and various sampling stategies for the shooting patch.

- Multiple specular reflections can be implemented by recursively creating virtual patches. It should be noted that since the shooting patch is the only one that need be reflected, there is no explosion of the complexity of the scene.

The current implementation is limited to planar specular surfaces. If more complex geometries are needed for the specular surfaces, it is no longer simple to construct a modified patch from which to shoot. Instead, specular rays can be fired from each receiving vertex lying on a specular surface, in a manner similar to [17]. However, the distribution of specular rays then depends on the mesh of vertices on the specular surface, with no guarantee that all vertices in the environment will receive their share of the specularly reflected light. A major benefit of the ray-casting approach to radiosity is then lost. Furthermore, rays must be properly weighted, taking surface curvature into account, to ensure a physically correct energy transfer.

4.1 Algorithm

The pseudo-code algorithm in Figure 3 contains two parts involving specular reflection (note that the treatment of specular reflection within the first pass occurs entirely within the shooting operation).

In the general loop that considers all receiving vertices in turn, specular surfaces are flagged whenever one of their vertices receives some energy from the shooting patch (Figure 3-C). Entire surfaces (planar patches) are flagged regardless of their subdivision into patches or elements, or their number of vertices.

Once all the vertices have had their directional intensity distributions updated with respect to the shooting patch, the specular reflectors are then considered in turn (Figure 3-D). For each specular surface, a new patch is created and the occlusion testing function is modified in preparation for a recursive call to the shooting procedure SHOOT.

The new patch is obtained by reflecting the original shooting patch and its attached coordinate system across the specular surface $\mathcal{A}$. The new, virtual patch possesses the same intensity distributions as the original shooting patch except that they are reflected by virtue of the reflected coordinate system.

A shot from the virtual patch P_i' affects only those vertices in the environments that can "see" the original shooting patch in the specular surface. This is easily accomplished by sampling the virtual patch as a normal shooting patch, but using a modified occlusion testing routine between the receiving vertex and the sample point. Figure 8 depicts the extended occlusion test.

An occlusion function (called $S_{\mathcal{A}}$ in Figure 3, and described as pseudo-code in Figure 9) is first called : this function first looks for an intersection between the specular surface $\mathcal{A}$ and the ray linking the receiving vertex V_j and a sample point on the virtual patch T_S. If no intersection is found, there can be no light reflected in that direction, and the function returns. Next, if an intersection point τ_S was found, a normal occlusion test is performed between V_j and τ_S. If the two points are visible to one another, the *specular reflectance* of the specular surface is computed for the appropriate reflection angle at τ_S.

If $S_{\mathcal{A}}$ returns a non-zero value, the only remaining operation consists in determining the occlusion between τ_S and a sample point on the original shooting patch, that is, the reflected image of T_S (Figure 8). This is accomplished by calling whatever occlusion function was in use at the current level of recursion : if we are dealing with a first specular reflection, the generic occlusion testing routine S_{generic} is used. At deeper levels of recursion, a composite function obtained by previous chaining operations is used.
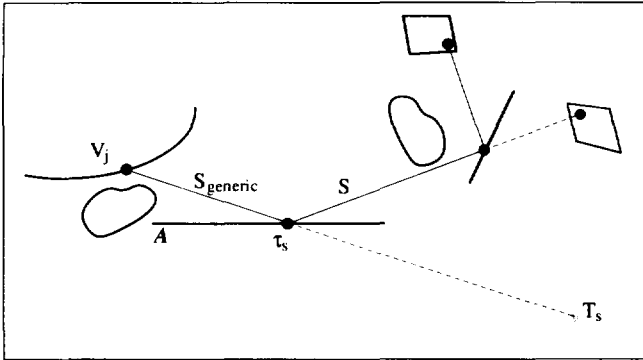
191

Figure 8: Chaining the occlusion functions. Once occlusion between V_j and τ_S is resolved, the composite occlusion test is performed between τ_S and the sample point on the original patch, possibly involving several specular reflections.

The entire operation can be described as chaining together the current occlusion testing function with the occlusion routine for the current specular surface. Note that once point τ_S has been found on the specular surface, the order in which the two occlusion tests are carried out is arbitrary. However, since the occlusion test between τ_S and V_j is generally a simpler test, it is performed first.
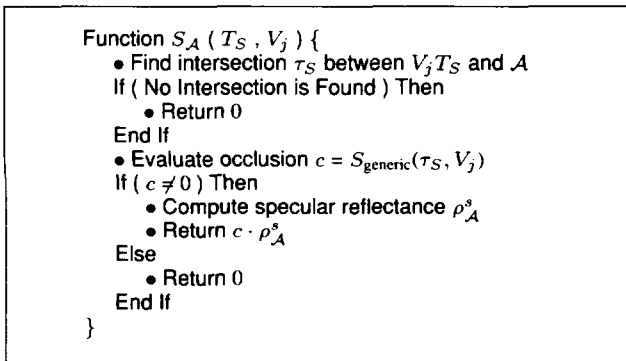
```
Function S_A ( T_S , V_j ) {
  • Find intersection τ_S between V_jT_S and A
  If ( No Intersection is Found ) Then
    • Return 0
  End If
  • Evaluate occlusion c = S_generic(τ_S, V_j)
  If ( c ≠ 0 ) Then
    • Compute specular reflectance ρ^s_A
    • Return c · ρ^s_A
  Else
    • Return 0
  End If
}
```

Figure 9: Occlusion testing for a virtual patch

5   Storing Intensity Distributions

The main departure of the current algorithm from previous progressive radiosity methods is that unshot and accumulated intensities now take the form of distribution functions at each vertex instead of scalar values. Because the number of vertices required for an accurate simulation can be quite large, it is crucial that the representation of these functions be economical in terms of storage. Moreover, the representation must allow for efficient "shooting" steps, which are performed many thousands of times in the course of a single simulation.

To fit within the framework of progressive radiosity, intensity distributions must also be computed incrementally by summing the directional distributions resulting from impinging shots. After accumulating contributions shot from n sample points on other patches, the intensity distribution at vertex k on an isotropic surface is given by the following equation.

$$I_k(\theta, \phi) = \sum_{i=1}^{n} \Phi_i \, \rho_k(\theta_i^{in}, \theta, \phi + \Delta\phi_i) \qquad (11)$$

Here Φ_i, θ_i^{in}, and $\Delta\phi_i$ are the energy flux, angle of incidence, and azimuthal angle of the i'th contribution respectively. Here we have expressed the BRDF parameters as angles with respect to a fixed local coordinate system at vertex k (Figure 5). The vertical axis of this coordinate system corresponds to the surface normal at that vertex though the other axes are arbitrary.

We can interpret Equation (11) as a sequence of four operations applied to the underlying BRDF, ρ_k, for each contribution arriving at vertex k: retrieving the directional distribution for a given angle of incidence, scaling and rotating this distribution, and finally adding it to the accumulated and unshot distributions stored at the vertex. These steps are shown in Figure 3-B.

From these operations it is clear that the shape of each intensity distribution depends solely on the BRDF associated with the vertex and not on the distributions from which the energy was shot. While this constrains the class of distributions that can arise at any given vertex, the distributions resulting from many contributions may nonetheless be quite irregular if the BRDF has a directional component (Figure 6).

We therefore require a representation that is general enough to account for this variation while also accomodating the steps in Figure 3-B. High-order continuity is also a requirement, since a discrete description, such as the *global cube* [11], can result in severe aliasing problems. Furthermore, derivative discontinuities in the intensity distributions can cause artifacts such as Mach-banding on the illuminated surfaces, even if a perfectly accurate transfer of light is computed.

In the following sections we describe an approach based on spherical harmonics which meets these requirements. Using this mechanism we can compactly and accurately represent arbitrary BRDFs and their associated intensity distributions and efficiently perform all of the operations required for shooting and incremental creation. It is therefore a nearly ideal mechanism for storing the intensity distributions for this global illumination algorithm.

5.1   Approximation using Spherical Harmonics

Spherical harmonics form an orthogonal basis for the space of functions defined over the unit sphere [6]. This infinite collection of basis functions is typically denoted by $Y_{l,m}(\theta, \phi)$ where $0 \leq l < \infty$ and $-l \leq m \leq l$. In direct analogy with Fourier series in one dimension, any square-integrable function, $f(\theta, \phi)$, can be represented by an infinite series of the form

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=-l}^{l} C_{l,m} \, Y_{l,m}(\theta, \phi) \qquad (12)$$

where the coefficients are given by

$$C_{l,m} = \int_0^{2\pi} \int_0^{\pi} f(\theta, \phi) \, Y_{l,m}(\theta, \phi) \, \sin(\theta) \, d\theta \, d\phi. \qquad (13)$$

The practical value of this is that a finite number of terms can be used to approximate relatively smooth functions defined on the sphere. This allows us to store intensity distributions as a vector of N coefficients, where N depends upon the characteristics of the underlying BRDF and the desired accuracy of the approximation. A diffuse, smoothly varying BRDF will typically require fewer coefficients than a very directional one.

To construct such a representation for the intensity distributions we begin by approximating the BRDFs in terms of spherical harmonics. In previous work, Cabral *et al.* have used a similar approximation for the purpose of simulating diffuse and glossy reflections of the environment [2]. In the present work the dependence of the BRDF on the angle of incidence is accounted for by representing each spherical harmonic coefficient as a function of θ^{in}. That is, for every BRDF we construct a collection of scalar functions, $B_{l,m}(\cdot)$, such that

$$\rho(\theta^{in}, \theta, \phi) \cos \theta \approx \sum_{l=0}^{N} \sum_{m=-l}^{l} B_{l,m}(\theta^{in}) Y_{l,m}(\theta, \phi). \quad (14)$$

In this way we can model the behavior of a BRDF over the entire range of incident angles. In our implementation the $B_{l,m}$ functions are stored as one-dimensional cubic splines; one for each spherical harmonic in the BRDF approximation. The $\cos \theta$ factor is included at this stage because it reduces evaluation time and tends to reduce ringing in the approximation. Figure 10 shows several of these curves for slightly rough aluminum. Additional details on this approximation can be found in Appendix A.
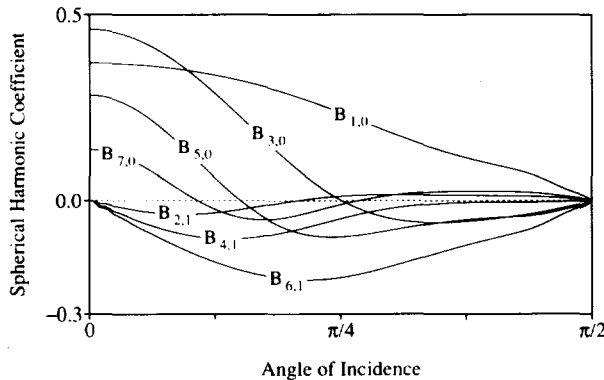

Figure 10: Seven spherical harmonic coefficients for the BRDF of slightly rough aluminum plotted as functions of the incident angle.

5.2 Operations on Spherical Harmonic Coefficients

Given a BRDF approximation of the form in equation (14) we can construct a corresponding intensity distribution using Equation (11). For every intensity contribution we first evaluate the BRDF at the given angle of incidence, θ^{in}, by computing the spherical harmonic coefficients of the resulting directional distribution. This consists of evaluating an interpolating spline, $B_{l,m}(\cdot)$, for each coefficient.

Next, we scale the distribution by multiplying each of these coefficients by the energy flux. The third step, rotating the distribution about the vertical axis, is made simple by the following property of spherical harmonics (shown in real form).

$$\begin{bmatrix} Y_{l,m}(\theta, \phi + \alpha) \\ Y_{l,-m}(\theta, \phi + \alpha) \end{bmatrix} = \begin{bmatrix} \cos(m\alpha) & -\sin(m\alpha) \\ \sin(m\alpha) & \cos(m\alpha) \end{bmatrix} \begin{bmatrix} Y_{l,m}(\theta, \phi) \\ Y_{l,-m}(\theta, \phi) \end{bmatrix}$$

This property follows immediately from the definition of spherical harmonics given in Appendix A. Rotation about this axis is particularly straightforward, and the usual symmetry of the BRDFs with respect to the incident plane simplifies it even further. Because negatively subscripted spherical harmonics are odd functions with respect to ϕ, we are guaranteed that all such coefficients will vanish

in the BRDF approximation. These coefficients reappear in the intensity distributions, however, because the symmetry is destroyed when the BRDFs undergo arbitrary rotations. This can be seen in step 3 of figure 11.

Initialize: $C_{l,m}^{k} \leftarrow 0$

For Each Contribution $(\Phi, \theta^{in}, \Delta\phi)$ arriving at Vertex k

    For Each index pair, (l, m), used in the approximation of ρ_k

        1. **Interpolate:** $A_{l,m} \leftarrow B_{l,m}^{k}(\theta^{in})$

        2. **Scale:** $A'_{l,m} \leftarrow \Phi A_{l,m}$

        3. **Rotate:** $\begin{bmatrix} A''_{l,m} \\ A''_{l,-m} \end{bmatrix} \leftarrow A'_{l,m} \begin{bmatrix} \cos(m\Delta\phi) \\ \sin(m\Delta\phi) \end{bmatrix}$

        4. **Add:** $\begin{bmatrix} C_{l,m}^{k} \\ C_{l,-m}^{k} \end{bmatrix} \leftarrow \begin{bmatrix} C_{l,m}^{k} \\ C_{l,-m}^{k} \end{bmatrix} + \begin{bmatrix} A''_{l,m} \\ A''_{l,-m} \end{bmatrix}$

    End For

End For

Figure 11: Creating an intensity distribution. When $m = 0$, steps 2 through 4 reduce to $C_{l,0}^{k} \leftarrow C_{l,0}^{k} + \Phi A_{l,0}$.

As the fourth and final step we add the resulting distribution to the current total by adding the corresponding coefficients. Thus, we have rephrased each of the steps in Figure 3-B in terms of operations on spherical harmonic coefficients. The actual steps are shown in Figure 11 where the $C_{l,m}^{k}$ denote coefficients of an intensity distribution at vertex k.

It is apparent from these operations that summing scaled and rotated instances of a single representation introduces no additional coefficients once the symmetry has been broken. Therefore, *the storage required for a given intensity distribution does not grow as intensity is accumulated.* Furthermore, the intensity distributions retain the full accuracy of the original BRDF approximations.

To perform the shooting step we must evaluate an intensity distribution in directions toward all vertices to which intensity is to be shot. This requires evaluating the $Y_{l,m}$ functions associated with the coefficients of the intensity distribution in each of these directions. These evaluations can be performed efficiently using the recurrence relations shown in Appendix A.

6 Results

Solutions have been computed for several test environments to demonstrate the feasibility of the simulation for arbitrary reflectance distributions. The resulting pictures exhibit all the expected visual effects produced by directional diffuse as well as ideal specular energy transfers.

Figure 12 shows a side by side comparison of a simulated environment with a scanned physical environment. The scanned picture was obtained by scanning through three colored filters, where each channel is spectrally integrated over a large range of wavelengths. Thus, the comparison with a simulation computed with three well-defined monochromatic channels can only be qualitative (for example the general color tone is noticably different). However, important features such as the structure of the shadow on the left, or the illumination of the ceiling via specular reflection from the top of the tall box, appear to be very similar. A related research project is un-
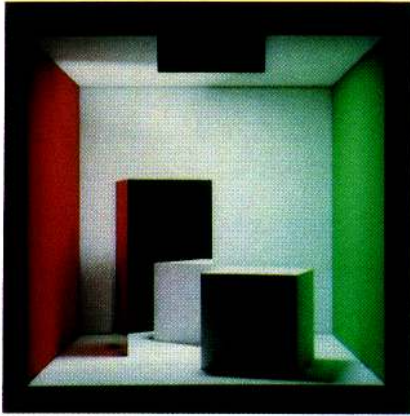
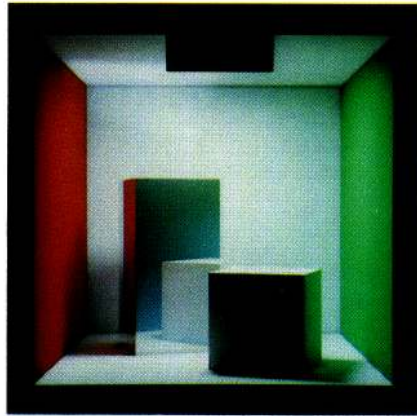Figure 13: First-surface mirror

Figure 14: Smooth Aluminum box #1
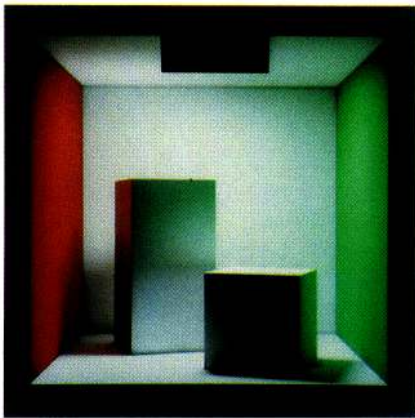

Figure 15: Smooth Aluminum box #2

Figure 16: Rough Aluminum

In figure 13, the tall box is covered with perfectly smooth aluminum, making it a first surface mirror. In figure 14 the surface has been roughened slightly. Note that ideal specular effects are still very strong, both in the reflected image on the box, and in the upper left corner of the environment, where light is reflected by the top of the tall box. The directional nature of the diffused light can also be seen on the front face of the tall box. In figure 15, the box is made somewhat rougher, and more of the light is being reflected in a directional diffuse manner. Note that the specular reflection is stronger on the left face of the box than it is on the front face because of the different incident angles for rays reaching the eye. This is a result of the model used to predict the specular reflectance [8] and is observed on real materials. Figure 16 shows an aluminum box that is rougher still, and the ideal specular component of the reflectance has almost disappeared. Note that there is still a concentration of light in the upper left corner of the scene, because light reflected by the top face of the box has a definite directional character. Figures 17 and 18 show the results of the algorithm run on more complex scenes. In figure 17 the aluminum bowl and cupboard doors are directional diffuse reflectors, and in figure 18 the telephone, the drawer handles and the blackboard frame are made of different varieties of aluminum.

Figure 17: Kitchen with directional diffuse cupboards and mixing bowl. Note the highlights on the cupboards.

Figure 18: Desk with aluminum phone and drawer handles. Note the caustics on the back wall.
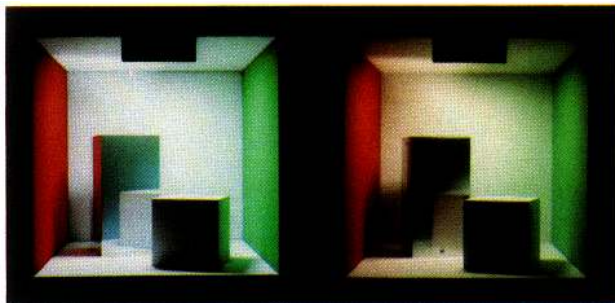
Figure 12: Comparison of a simulation with a scanned image.

der way to develop precise radiometric measurement procedures so that accurate quantitative comparisons can be made [10].

Figures 13 to 16 show simulations of a simple environment with varying surface properties. The tall box is made of aluminum with increasing roughness throughout the sequence. Figures 17 and 18 demonstrate the method applied to complex scenes (consisting of 34400 and 30200 elements, respectively). The last two pictures required an average time of 5 minutes per radiosity shot, on an Apollo DN 10000 computer with 64 Megabytes of main memory.

In all these simulations, between 80 and 133 coefficients were used to approximate the BRDFs, resulting in intensity distributions of 148 to 254 coefficients. The number of coefficients in the latter increases due to loss of symmetry as described in section 5.2. The number of coefficients was determined for each material in order to achieve a given precision in the approximation of the BRDF over the entire range of incidence angles. Ideal diffuse surfaces require only one coefficient, and in general the more directional the BRDF, the more coefficients are required to account for the higher frequencies in the shape.

In all cases the intensity distributions have been stored using fewer than 1 K bytes at each vertex, while the error in the approximation was kept to a few percent. For any fixed number of coefficients, however, the approximation will generally begin to degrade as the BRDF becomes increasingly directional. Fortunately, according to physical optics, this will be somewhat compensated by the fact that the ideal specular component, which is handled separately, begins to dominate in this case [8]. Therefore, even for nearly-perfect mirrors, we can maintain a reasonable accuracy in terms of energy balance without increasing storage.

7 Conclusions and future directions

An algorithm for the simulation of global illumination with arbitrary reflectance behaviors was presented. A general treatment of *ideal specular to directional diffuse* transfers was introduced, allowing all chains of reflection to be effectively simulated. An efficient method using spherical harmonics to store the directional diffuse intensity distributions has been demonstrated, which permits a given accuracy to be maintained with a fixed amount of storage per vertex throughout the simulation.

Other features could be added easily to the system. An extension to anisotropic reflectance models would require an extra level of interpolation to obtain the BRDF, since it would depend on both incident angles (elevation and azimuth). This can be accomplished by using a two-dimensional spline surface instead of the current one-dimensional spline function for the spherical harmonics coefficients.

Polarization effects that are predicted by general reflectance models [8] could also be incorporated into the system very naturally, by

using the complex formulation of spherical harmonics.

This method suffers from several limitations inherited from the progressive radiosity method. For example, the intensity distribution is assumed to be uniform across a patch at shooting time, even though more detailed information might already be available from the subdivision of the patch into elements. Better sampling strategies are also needed. However, we believe that the formulation of the global illumination problem in terms of directional distributions attached to the surfaces, combined with a physically consistent treatment of ideal specular reflection, allows a truly general solution to be computed for arbitrary reflectance distributions.

8 Acknowledgements

References

[1] Buckalew, Chris and Donald Fussell. "Illumination Networks: Fast Realistic Rendering with General Reflectance Functions," *Computer Graphics*, 23(3), July 1989, pages 89–98. (Proceedings SIGGRAPH '89 in Boston.)

[2] Cabral, Brian, Nelson L. Max, and Rebecca Springmayer. "Bidirectional Reflection Functions from Surface Bump Maps," *Computer Graphics*, 21(4), July 1987, pages 273–281. (Proceedings SIGGRAPH '87 in Anaheim.)

[3] Cohen, Michael F., Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. "A Progressive Refinement Approach to Fast Radiosity Image Generation," *Computer Graphics*, 22(4), August 1988, pages 75–84. (Proceedings SIGGRAPH '88 in Atlanta.)

[4] Cohen, Michael F., Donald P. Greenberg, David S. Immel, and Philip J. Brock. "An Efficient Radiosity Approach for Realistic Image Synthesis," *IEEE Computer Graphics and Applications*, 6(3), March 1986, pages 25–35.

[5] Cook, Robert L., Thomas Porter, and Loren Carpenter. "Distributed Ray Tracing," *Computer Graphics*, 18, July 1984, pages 137–147. (Proceedings SIGGRAPH '84 in Minneapolis.)

[6] Courant, R. and D. Hilbert. *Methods of Mathematical Physics*, Interscience Publishers, Inc., New York, 1953.

[7] Goral, Cindy M., Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. "Modeling the Interaction of Light Between Diffuse Surfaces," *Computer Graphics*, 18(3), July 1984, pages 213–222. (Proceedings SIGGRAPH '84 in Minneapolis.)

[8] He, XiaoDong, Kenneth E. Torrance, François Sillion, and Donald P. Greenberg. "A comprehensive Physical Model for Light Reflection," *Computer Graphics*, 25(4), August 1991. (Proceedings SIGGRAPH '91 in Las Vegas.)

[9] Heckbert, Paul S. "Adaptive Radiosity Textures for Bidirectional Ray Tracing," *Computer Graphics*, 24(4), August 1990, pages 145–154. (Proceedings SIGGRAPH '90 in Dallas.)

[10] Himlan, Theodore H., Michael C. Monks, Stephen H. Westin, Donald P. Greenberg, and Kenneth E. Torrance. "Physical Measurement Techniques for Improving and Evaluating Computer Graphic Simulations," January 1991. (Submitted for publication.)

[11] Immel, David S., Michael F. Cohen, and Donald P. Greenberg. "A Radiosity Method for Non-Diffuse Environments," *Computer Graphics*, 20(4), August 1986, pages 133–142. (Proceedings SIGGRAPH '86 in Dallas.)

[12] Kajiya, James T. "The Rendering Equation," *Computer Graphics*, 20(4), August 1986, pages 143–150. (Proceedings SIGGRAPH '86 in Dallas.)

[13] Le Saec, Bertrand and Christophe Schlick. "A Progressive Ray-Tracing based Radiosity with General Reflectance Functions," June 1990. (Proceedings of the Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics (Rennes, France).)

[14] Nishita, T. and E. Nakamae. "Continuous Tone Representation of Three-dimesional Objects Taking Account of Shadows and Interreflection," *Computer Graphics*, 19(3), July 1985, pages 23–30. (Proceedings SIGGRAPH '85 in San Francisco.)

[15] Press, William H., Brian P. Flannery, and Saul A. Teukolsky. *Numerical Recipes*, Cambridge University Press, New York, 1986.

[16] Rushmeier, Holly E. and Kenneth E. Torrance. "Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials," *ACM Transactions on Graphics*, 9(1), January 1990, pages 1–27.

[17] Sillion, François and Claude Puech. "A General Two-Pass Method Integrating Specular and Diffuse Reflection," *Computer Graphics*, 23(4), August 1989. (Proceedings SIGGRAPH '89 in Boston.)

[18] Wallace, John R., Michael F. Cohen, and Donald P. Greenberg. "A Two-Pass Solution to the Rendering Equation : a Synthesis of Ray-Tracing and Radiosity methods," *Computer Graphics*, 21(4), July 1987, pages 311–320. (Proceedings SIGGRAPH '87 in Anaheim.)

[19] Wallace, John R., Kells A. Elmquist, and Eric A. Haines. "A Ray Tracing Algorithm for Progressive Radiosity," *Computer Graphics*, 23(3), July 1989, pages 315–324. (Proceedings SIGGRAPH '89 in Boston.)

[20] Whitted, Turner. "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, 23, 1980, pages 343–349.

Appendix A: More on spherical harmonics

In real form, the normalized spherical harmonics are defined by

$$
Y_{l,m}(\theta,\phi) = \begin{cases} N_{l,m}\, P_{l,m}(\cos\theta)\cos(m\phi) & \text{if } m > 0 \\[2mm] N_{l,0}\, P_{l,0}(\cos\theta)/\sqrt{2} & \text{if } m = 0 \\[2mm] N_{l,m}\, P_{l,|m|}(\cos\theta)\sin(|m|\phi) & \text{if } m < 0 \end{cases} \quad (15)
$$

where the normalizing constants, $N_{l,m}$, are given by

$$
N_{l,m} = \sqrt{\frac{2l+1}{2\pi}\frac{(l-|m|)!}{(l+|m|)!}} \qquad (16)
$$

and the $P_{l,m}(x)$ factors are *associated Legendre polynomials*. The latter can be evaluated with the recurrence relations

$$
\begin{aligned}
P_{m,m}(x) &= (1-2m)\sqrt{1-x^2}\,P_{m-1,m-1}(x) \\[2mm]
P_{m+1,m}(x) &= x(2m+1)P_{m,m}(x) \\[2mm]
P_{l,m}(x) &= x\left(\tfrac{2l-1}{l-m}\right)P_{l-1,m}(x) - \left(\tfrac{l+m-1}{l-m}\right)P_{l-2,m}(x)
\end{aligned}
$$

beginning with $P_{0,0}(x) = 1$ [15]. Applying these in conjunction with recurrence relations for generating $\sin(\phi)$, $\sin(2\phi), \ldots, \sin(m\phi)$ and $\cos(\phi), \cos(2\phi), \ldots, \cos(m\phi)$, it is possible to evaluate spherical harmonic expansions using approximately 10 floating point operations per coefficient and no trigonometric function evaluations whatsoever.

Evaluation of the spherical harmonic functions is required for two purposes: shooting from an intensity distribution and creating the initial BRDF approximations. The former is a straightforward application of Equation 12 while the latter is more complicated and is performed once per distinct BRDF. To approximate an isotropic BRDF for all incident angles, we first compute

$$
b^j_{l,m} = \int_0^\pi \int_0^\pi \rho(\theta^{in}_j,\theta,\phi)\, Y_{l,m}(\theta,\phi)\,\sin(\theta)\,d\theta\,d\phi \qquad (17)
$$

for $j = 0,1,\ldots q$ where $0 = \theta^{in}_0 \le \theta^{in}_1 \le \cdots \le \theta^{in}_q = \pi/2$. Then the functions $B_{l,m}$ can be approximated by cubic interpolating splines through the the points $(\theta^{in}_0, b^0_{l,m}), (\theta^{in}_1, b^1_{l,m}), \ldots, (\theta^{in}_q, b^q_{l,m})$. For each distinct BRDF, ρ, we select q as well as a specific set of spherical harmonic coefficients to achieve the desired accuracy of approximation over all incident angles. The value of q affects the accuracy of the interpolation but does not otherwise influence the intensity distributions. In contrast, the number of coefficients used in the BRDF approximation directly determines the sorage required for the intensity distributions. It is therefore important to keep this number reasonably small.

If the BRDF that we wish to approximate is only defined on the upper hemisphere, as with an opaque material, we extend the function to the lower hemisphere before computing the approximation. We do this in such a way that the complete BRDF satisfies

$$
\rho(\theta^{in},\theta,\phi) = -\rho(\theta^{in},\pi-\theta,\phi). \qquad (18)
$$

This introduces a vertical symmetry which has the advantage of eliminating all spherical harmonics for which $l+m$ is even. It also maintains C^1 continuitiy between the upper and lower hemispheres when the function is zero at the equator, a condition that is guaranteed if the $\cos\theta$ factor is included as described in section 5.1.