A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection

Larry Aupperle Pat Hanrahan

Department of Computer Science
Princeton University

Abstract

We develop a radiance formulation for discrete three point transport, and a new measure and description of reflectance: *area reflectance*. This formulation and associated reflectance allow an estimate of error in the computation of radiance across triples of surface elements, and lead directly to a hierarchical refinement algorithm for global illumination.

We have implemented and analyzed this algorithm over surfaces exhibiting glossy specular and diffuse reflection. Theoretical growth in light transport computation is shown to be $O(n+k^3)$ for sufficient refinement, where n is the number of elements at the finest level of subdivision over an environment consisting of k input polygonal patches — this growth is exhibited in experimental trials. Naive application of three point transport would require computation over $O(n^3)$ element-triple interactions.

CR Categories and Subject Descriptors: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Key Words: adaptive meshing, global illumination, radiosity, ray tracing.

1 Introduction

A major open problem in image synthesis is the efficient solution of the rendering equation. Radiosity methods have been quite successful over environments containing surfaces that exhibit only diffuse reflection. Unfortunately, very few materials are purely Lambertian reflectors, and efficient solution techniques have not yet been developed for more general specular or glossy reflection functions.

The rendering equation is an integral equation, and the solutions to complicated integral equations are generally obtained using either Monte Carlo or finite element techniques. Monte Carlo algorithms sometimes go under the name of distributed or stochastic ray tracing and are the most commonly employed in computer graphics (e.g. see [4, 5, 9, 12, 16]). Monte Carlo techniques have the advantage that they are easy to implement and can be used for complicated geometries and reflection functions. Unfortunately, their disadvantage is that they are notoriously inefficient. The second approach, the finite element method, has been very successfully applied to the rendering equation under the radiosity assumption, but has only begun to be employed in the general case, and with limited success. For example, Immel et al. [8] discretized radiance into a lattice of cubical environment maps, and solved the resulting system. More recently, Sillion et al. [13] used a mesh of spherical harmonic functions to represent radiance, and solved the resulting system using a shooting algorithm.

There are many ways to parameterize the rendering equation, and each leads to a different choice of basis functions. In the transport theory community two techniques are common: directional subdivision (the method of discrete ordinates or S_N), and spherical harmonics (P_N). These two techniques roughly correspond to the methods of Immel et al. and Sillion et al., although many interesting variations are possible. Our approach is somewhat different, and based on Kajiya's original formulation of the rendering equation [9]. Under this formulation, the rendering equation is expressed in terms of three point transport. That is, the kernel of the integral expresses the transport of light from a point on the source to a point on the receiver, via a point on a reflector. Given this formulation, the three point rendering equation can be discretized over pairs of elements to form a linear system of equations. Solving this system yields the radiance transported between elements. Note that this approach is very similar to the radiosity formulation.

The problem with finite element methods is that the matrix of interactions is very large for interesting environments. For a given environment of k input polygonal patches containing n elements at the finest level of refinement, the three point discretization that we are proposing generates an n^3 matrix of interactions. However, in this paper we show that we can accurately approximate the n^3 reflectance matrix with $O(n + k^3)$ blocks, in a way very similar to our recent hierarchical radiosity algorithm [7]. In that paper we showed how the n^2 form factor matrix could be approximated with $O(n + k^2)$ blocks, resulting in a very efficient algorithm in both space and time. Although the results presented in this paper are preliminary, we believe a hierarchical finite element approach along these lines will ultimately lead to a fast, efficient algorithm.

In the following section we describe our application of the finite element element method to the three point rendering equation, yielding a radiance formulation for discrete transport. In Section 3 we present a simple adaptive refinement algorithm for computation over this formulation, and the iterative solution technique employed for the actual calculation of transport. In Section 4 we discuss our implementation of the algorithm over glossy reflection, and in Section 5 we present some experiments and results. An appendix to this paper contains details of our error analysis for discrete transport under the glossy model.

2 Discrete Three Point Transport

The algorithm presented in this paper operates through two functions: refinement of the environment to form a hierarchy of discrete interactions, patches and elements, and the actual computation of illumination over this hierarchy.

In this section we develop the basis for both discretization and transport. We derive a radiance formulation for three point transport, and a new measure and description of reflectance, *area reflectance*. This radiance formulation and associated reflectance provide a natural criterion for discretization under illumination and reflection, and allow both the computation of radiance across triples of individual surface elements, and the expression and computation of all light transport over all surfaces.

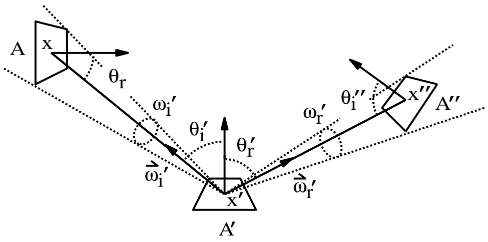

Figure 1: Geometry of Reflection

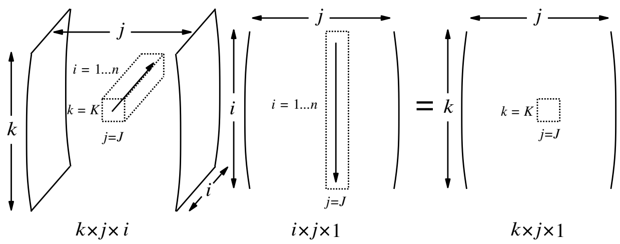

Figure 2: A Reflection Product

2.1 A Radiance Formulation for Three Point Transport

When computing and imaging illumination within an environment, we are interested in the transport of light from surface to surface — it is this interaction of surfaces that characterizes illumination, in the absence of participatory media. Reflection within an environment may thus be naturally expressed over triples of surfaces. Consider surfaces A, A', and A'' (Figure 1) — we will examine the transport of light incident at A' originating at A and reflected toward A''.

Let ω_i' and ω_r' be the solid angles subtended at point x' by A and A'', respectively. Consider differential solid angles at $\vec{\omega}_i'$ and $\vec{\omega}_r'$ — by definition of the bidirectional reflectance-distribution function (BRDF), f_r [11], the radiance $L(\vec{\omega}_r')$ along $\vec{\omega}_r'$ due to illumination through solid angle ω_i' is:

$$L(\vec{\omega}_r') = \int_{\omega_i'} f_r(\vec{\omega}_i', \vec{\omega}_r') L(\vec{\omega}_i') \cos\theta_i' d\omega_i'$$

Integrating this expression over ω_r', and introducing $\cos\theta_r'$, we have:

$$\int_{\omega_r'} L(\vec{\omega}_r') \cos\theta_r' d\omega_r' = \int_{\omega_r'} \int_{\omega_i'} f_r(\vec{\omega}_i', \vec{\omega}_r') L(\vec{\omega}_i') \cos\theta_i' \cos\theta_r' d\omega_i' d\omega_r'$$

We may then reparameterize over A and A'' to yield:

$$\int_{A''} L(x', x'') G(x', x'') dx'' = \int_A \int_{A''} f_r(x, x', x'') L(x, x') G(x, x') G(x', x'') dx'' dx$$

where

$$G(x, x') = \frac{\cos\theta_r \cos\theta_i'}{|x - x'|^2} v(x, x')$$

where $v(x, x')$ is 1 if points x, x', are mutually visible, and 0 otherwise. Note that G is very similar to a differential form factor.

We integrate over A', thus introducing all three areas into the formulation:

$$\int_{A'} \int_{A''} L(x', x'') G(x', x'') dx'' dx' = \int_A \int_{A'} \int_{A''} f_r(x, x', x'') L(x, x') G(x, x') G(x', x'') dx'' dx' dx \qquad (1)$$

We may now rewrite the equation in discrete form. Let A_j and A_k be subareas of A' and A'' such that $L(x', x'')$ is nearly constant over their surfaces. The left side of equation (1) may then be rewritten, bringing radiance out of the integral as L_{jk}:

$$L_{jk} \int_{A_j} \int_{A_k} G(x', x'') dx'' dx' = \pi L_{jk} A_j F_{jk}$$

by definition of the diffuse form factor, F_{jk}.

We may similarly discretize A as A_i, and rewrite the right side of equation (1) as:

$$L_{ij} \int_{A_i} \int_{A_j} \int_{A_k} f_r(x, x', x'') G(x, x') G(x', x'') dx'' dx' dx = \sum_i \pi L_{ij} A_i F_{ij} R_{ijk}$$

where R_{ijk} is defined such that

$$\pi A_i F_{ij} R_{ijk} = \int_{A_i} \int_{A_j} \int_{A_k} f_r(x, x', x'') G(x, x') G(x', x'') dx'' dx' dx$$

Note that, by the symmetry of f_r and G:

$$A_i F_{ij} R_{ijk} = A_k F_{kj} R_{kji}$$

We thus have:

$$\begin{aligned} \pi L_{jk} A_j F_{jk} &= \pi \sum_i L_{ij} A_k F_{kj} R_{kji} \\ &= \pi \sum_i L_{ij} A_j F_{jk} R_{kji} \end{aligned}$$

by the reciprocity of form factors, and thus:

$$L_{jk} = \sum_i L_{ij} R_{kji}$$

The three dimensional character of R_{kji} over indices i, j, k leads naturally to a three dimensional matrix formulation for the above system. Consider a product over an $n \times n \times n$ R_{kji} "matrix" and an $n \times n \times 1$ L_{ij} matrix producing an $n \times n \times 1$ matrix of reflected radiances, as shown in Figure 2. Note that the R_{kji} matrix is of size $O(n^3)$ — the hierarchical method discussed in subsequent sections of this paper addresses more tractable representation of this matrix.

Taking into account emission, we have derived a radiance formulation for three point transport:

$$L_{jk} = E_{jk} + \sum_i L_{ij} R_{kji} \qquad (2)$$

This formulation states that:

> *The radiance at Area j in the direction of Area k is equal to the radiance emitted by j in the direction of k, plus, for every Area i, the radiance at i in the direction of j multiplied by the area reflectance R_{kji}.*

Note that equation (2) is very similar to the radiosity formulation:

$$B_j = E_j + \rho_j \sum_i B_i F_{ji}$$

2.2 Area Reflectance

The quantity R_{kji} has a natural and satisfying physical significance — it is an expression of reflectance over areas A_i, A_j, and A_k.

Consider the fraction of the radiant flux transported from A_i incident to A_j that is reflected in the direction of area A_k:

$$\frac{\int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')L(x,x')G(x,x')G(x',x'')dx''dx'dx}{\int_{A_i}\int_{A_j} L(x,x')G(x,x')dx'dx}$$

If we assume that incident radiance is uniform and isotropic over both ω_i' (as induced by A_i) and A_j, we may divide through by $L(x,x')$, yielding:

$$\rho(A_i,A_j,A_k) \equiv \frac{\int_{A_i}\int_{A_j}\int_{A_k} f_r(x,x',x'')G(x,x')G(x',x'')dx''dx'dx}{\int_{A_i}\int_{A_j} G(x,x')dx'dx}$$

We define $\rho(A_i,A_j,A_k)$ to be *area reflectance*. Note that area reflectance is similar to biconical reflectance [11], save that it is also integrated over the reflecting surface.

By definition of R_{ijk}:

$$R_{ijk} = \rho(A_i,A_j,A_k)$$

Conservation of energy over reflection, and the reciprocity relation derived for R_{ijk} above, constitute fundamental properties of area reflectance:

1. $\sum_k R_{ijk} \leq 1$, for fixed i,j.
2. $A_iF_{ij}R_{ijk} = A_kF_{kj}R_{kji}$.

where equality is achieved in property 1 over complete enclosures and perfect reflectivity.

2.3 Evaluation of R_{kji}

In this section we examine the evaluation of R_{kji} over given patches A_i, A_j, A_k.

Recall:

$$R_{kji} = \frac{\int_{A_k}\int_{A_j}\int_{A_i} f_r(x'',x',x)G(x'',x')G(x',x)dxdx'dx''}{\int_{A_k}\int_{A_j} G(x'',x')dx'dx''}$$

We assume that discrete areas A_i, A_j, A_k are of small enough scale that f_r and G are relatively constant over their surfaces. Then:

$$\begin{aligned} R_{kji} &= \frac{S_{kji}G_{kj}G_{ji}A_kA_jA_i}{G_{kj}A_kA_j} \\ &= S_{kji}G_{ji}A_i \end{aligned}$$

where S is the discretized value of f_r, $S_{ijk} = S_{kji} = S_{x_kx_jx_i}$.

Note that the average value of $G(x',x)$ over A_i and A_j is $\pi F_{ji}/A_i$ — we thus estimate $G_{ji}A_i$ by πF_{ji}, and compute R_{kji} as:

$$R_{kji} = \pi F_{ji}S_{kji}$$

In practice, it will not be possible to compute the exact values of F_{ji} and S_{kji} over A_i, A_j, A_k. We assume that we are able to estimate these values, along with error bounds for each estimation. Let ΔF_{ji} and ΔS_{kji} be error estimates for computed F_{ji} and S_{kji}, respectively. We then have an estimate for area reflectance in the form:

$$\begin{aligned} R_{kji} &= \pi(F_{ji}+\Delta F_{ji})(S_{kji}+\Delta S_{kji}) \\ &= \pi(F_{ji}S_{kji}+\Delta F_{ji}S_{kji}+\Delta S_{kji}F_{ji}+\Delta F_{ji}\Delta S_{kji}) \\ &\approx \pi(F_{ji}S_{kji}+\Delta F_{ji}S_{kji}+\Delta S_{kji}F_{ji}) \end{aligned}$$

Assuming $\Delta F_{ji} < F_{ji}$, $\Delta S_{kji} < S_{kji}$, we have neglected the last term and estimate the error in R_{kji} as $\pi(\Delta F_{ji}S_{kji} + \Delta S_{kji}F_{ji})$.

In general, and as is shown for glossy reflection in Section 4, the accuracy of estimators for F_{ji} and S_{kji} is dependent on the size of the patches over which reflectance is computed, relative to their distance apart. As relative size decreases, so does error in computation, leading directly to the adaptive refinement strategy for illumination presented in Section 3 below.

3 Algorithms for Three Point Transport

3.1 Introduction

Recall equation (2):

$$L_{jk} = E_{jk} + \sum_i L_{ij}R_{kji}$$

This equation suggests both a solution strategy for radiance under three point transport, and a natural representation for illumination within the solution system.

We may interpret equation (2) as a gathering iteration similar to that employed for radiosity under diffuse reflection: the radiance L_{jk} at patch A_j in the direction of patch A_k is found by gathering radiances L_{ij} in the direction of A_j at patches A_i. We may solve for transport by gathering radiance for each L_{jk}, and successively iterating to capture all significant re-reflection.

We are left with the question of what structure we are gathering over and iterating upon. Note that all illumination is expressed as the radiance at a given patch in the direction of another — it is these patch-patch interactions that form the primary structure within the solution system. All operation is over interactions: both the representation and transport of radiance, and the iteration and solution for illumination.

Consider the following structure:

```
typedef struct _interaction {

    Patch *from;
    Patch *to;

    Color  L;
    Color  Lg;

    List   *gather;

    struct _interaction *nw, *sw, *se, *ne;

} Interaction;
```

A given interaction ij is defined by two patches `ij->from` and `ij->to`, and represents the radiance at `from` in the direction of `to`. This radiance is stored within the interaction as attribute `L`. `Lg` is radiance gathered during the current solution iteration from interactions contained in the list `gather`. Subinteractions `nw`, `sw`, `se`, `ne` are the children of ij, induced by subdivision over either `from` or `to`. The structure assumes quadtree refinement, leaving northwest, southwest, southeast, and northeast descendants.

In the following sections we will present an algorithm for the refinement and computation of illumination over a hierarchy of interactions. The algorithm will operate by refining pairs of interactions ij, jk (such that `ij->to == jk->from`), to ensure that computed reflectance across the interaction pairs, and associated patch triples, satisfies user specified error bounds. If a given interaction pair ij, jk is satisfactory, the interactions are linked to record that radiance may be gathered from ij to jk, otherwise one or both interactions are subdivided and refinement applied to their descendants.

After refinement, a gathering iteration may be carried out, each interaction gathering radiance from interactions to which it has been linked. The gathered radiances are then distributed within each receiving interaction hierarchy, and subsequent iterations computed until satisfactory convergence has been achieved.

Note that, within this system, the eye may be regarded as simply another object with which patches may interact. The radiance along interactions to the eye provides the resulting view.

3.2 Adaptive Refinement

Consider the following procedure:

```
Refine(Interaction *ij, Interaction *jk,
       float Feps, float Seps, float Aeps)
{
    float feps, seps;

    feps = GeometryErrorEstimate(ij);
    seps = ReflectionErrorEstimate(ij, jk);

    if (feps < Feps && seps < Seps)
        Link(ij, jk);

    else if (seps >= Seps) {

        switch(SubdivS(ij, jk, Aeps)) {

        case PATCH_I:

            Refine(ij->nw, jk, Seps, Feps, Aeps);
            Refine(ij->sw, jk, Seps, Feps, Aeps);
            Refine(ij->se, jk, Seps, Feps, Aeps);
            Refine(ij->ne, jk, Seps, Feps, Aeps);

            break;

        case PATCH_J:
            /* refine over children of ij and jk */
        case PATCH_K:
            /* refine over children of jk        */
        case NONE:
            Link(ij, jk);
        }
    }

    else {    /* feps >= Feps */

        switch(SubdivG(ij, jk, Aeps)) {

        /* refine over children, or link, as    */
        /* directed by PATCH_I, J, K, or NONE.  */

        }
    }
}
```

This procedure computes over pairs of interactions, and associated patch triples, subdividing and recursively refining if estimated error exceeds user specified bounds, linking the interactions for gathering if the bounds are satisfied, or if no further subdivision is possible. `Feps` and `Seps` are the bounds for geometric and reflection error, respectively; `Aeps` specifies the minimum area a patch may possess and still be subdivided. `GeometryErrorEstimate` and `ReflectionErrorEstimate` provide estimations for $\pi\Delta F_{ji}S_{kji}$ and $\pi\Delta S_{kji}F_{ji}$.

`SubdivS` and `SubdivG` control refinement for reflection and geometry error, respectively. Both routines select a patch for refinement, subdividing the patch and associated interaction(s) if required. An identifier for the selected patch is returned — if no patch may be subdivided, then `NONE` is passed back. Note that a given interaction/patch may be refined against many different interactions within the system, and thus may have already been subdivided when selected by a `Subdiv` routine — in this case, the routine simply returns the proper identifier.

The `Subdiv` routines should select for refinement patches that are of large size relative to their distance from their partner(s) in the transport triple. Form factor estimation is a convenient criterion for the determination of such patches — a large differential to area form factor F_{dpq} indicates that patch q is of large relative size. Care must be taken in subdivision, however, to ensure that each interaction is always subdivided in the same way for all refinements involving that interaction.

The `Subdiv` routines thus choose for refinement the patch of size at least `Aeps` that is of greatest form factor within ij and/or jk that will not induce multiple sets of children over either interaction. If patch p_j is of greatest form factor over both ij and jk, and of area greater than `Aeps`, then it is chosen for refinement (Figure 3 at middle). Otherwise, if p_j is selected over one interaction, but p_i or p_k is selected over the other, then the "outside" patch is chosen for refinement. Given two selected outside patches, `SubdivS` selects the one of greater form factor relative to p_j; `SubdivG` selects p_i over p_k, as p_k has no direct effect on geometric accuracy. Note, however, that even under `SubdivG`, if only p_j and p_k are allowed subdivision, p_k will be selected, although with further subdivision the triple will eventually balance sufficiently to allow refinement over p_j.

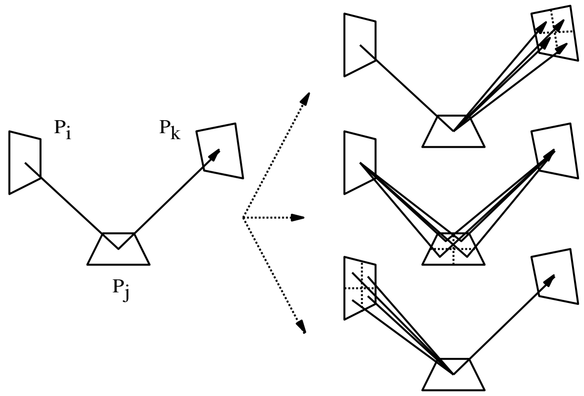

Figure 3: Refinement and Subdivision

3.3 Gathering Radiance

Gathering radiance over interactions may be written as a simple procedure:

```
Gather(Interaction *jk)
{
    Interaction *ij;

    if (jk) {

        jk->Lg = 0;

        ForAllElements(ij, jk->gather)
            jk->Lg += ij->L * Reflectance(ij, jk);

        Gather(jk->nw);
        Gather(jk->sw);
        Gather(jk->se);
        Gather(jk->ne);
    }
}
```

We gather radiance into `jk->Lg` rather than directly into `jk->L` to avoid the necessity of a push/pull with every invocation of the procedure (see Section 3.4). The solution method is thus simple Jacobi iteration, as opposed to Gauss-Seidel, as the hierarchical structure imposes simultaneous rather than successive displacement.

3.4 Radiance within a Hierarchy

A gathering iteration results in received radiance scattered throughout each interaction hierarchy. This gathered radiance must be distributed and accounted for over all ancestors and descendants of each receiving interaction, in order to maintain the consistency and correctness of the hierarchical representation of radiance between patches.

We employ a distribution algorithm similar to that presented in [7] for radiosity over patch/element hierarchies: gathered radiance is "pushed" to the leaf interactions within each hierarchy to ensure propagation to all descendants, and then "pulled" and distributed back up from the leaves through all higher level interactions to their common ancestor at the root. As is shown in [2], radiance may be pushed unchanged within the interaction hierarchy, and area averaged as it is pulled from child to parent.

4 Application over Glossy Reflection

In this section we discuss our implementation of the above algorithms over glossy reflection.

4.1 The Reflection Function

We employ a highly simplified Torrance-Sparrow [15] model for our glossy reflection function:

$$f_g(\vec{\omega}_i, \vec{\omega}_r) = \frac{\kappa+2}{8\pi}\frac{\cos^\kappa\theta_m}{\cos\theta_i\cos\theta_r}sh(\theta_i, \theta_r)$$

This function incorporates the facet distribution function $\cos^\kappa\theta_m$ developed by Blinn [3], normalized for projected facet area under

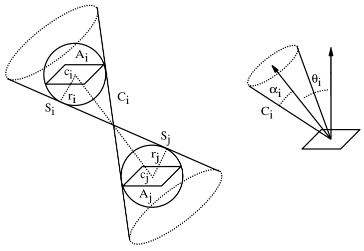

Figure 4: Estimating Cones

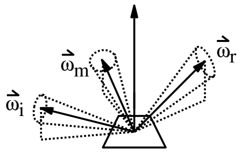

Figure 5: C_i, C_r, and C_m

[10]. Angle θ_m is that made to the mean surface normal by $\vec{\omega}_m$, the microfacet mirror orientation normal lying halfway between $\vec{\omega}_i$ and $\vec{\omega}_r$.

Function $sh(\theta_i, \theta_r)$ expresses self-shadowing over microfacets — for near specular surfaces, such self-shadowing or masking does not become critical until relatively high θ_i or θ_r [6]. The implemented system thus simply clamps sh from 1 to 0 when θ_i or θ_r exceeds a preset θ_{bound} near the horizon. This scheme serves as a crude approximation to the shadowing function; however, a better strategy would be to employ a much fuller tabulation of the function, incorporated into the error analysis presented below. A more complete discussion of shadowing and conservation of energy over f_g is presented in [2].

4.2 Error Estimation

Recall the general expression for error derived in Section 2.3:

$$\pi(\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji})$$

In implementation we have estimated the form factor F_{ji} by F_{dji}, the form factor from a differential area at A_j to a disk of area A_i centered at A_i, as was employed in [7]. As discussed in [7], the relative error in this estimate is proportional to the estimate itself. In our implementation we have thus estimated absolute error ΔF_{ji} as at most proportional to F_{dji}^2.A brief discussion of relative and absolute error over hierarchical methods is presented in [2].

We now consider the error estimate ΔS_{kji}. As discussed in the appendix to this paper, we may compute bounding cones C_i, C_r, and C_m over all possible incident, reflected, and mirror orientation directions induced at A_j by A_i and A_k (Figures 4 and 5 — these figures are discussed more fully in the appendix). We may then compute maximum and minimum $\cos^\kappa \theta_m, \cos\theta_i, \cos\theta_r$ over these cones, and estimate error by interval width. The full expression for estimated error over transport is given in the appendix.

4.3 Clamping and Visibility

Evaluation of glossy reflectance over three surface areas, as required by the gather iteration, may be difficult, particularly if surface subdivision has been limited by `Aeps` rather than satisfaction of error bounds, and if κ, the facet distribution exponent, has high value. In this case we must estimate the integral of a spikey function over a relatively broad area.

Our solution is to band limit the BRDF in a fashion similar to that presented by Amanatides [1]. We employ the cone estimation techniques of the previous section to determine if the BRDF varies significantly over the given patches — if this variance exceeds a set bound, we "roughen" the reflecting surface, lowering κ to broaden the resulting reflection over the estimated cones. We then renormalize the resulting blurred function, as described in [1], to prevent amplification of its low frequency components. We note that the resulting antialiasing is relatively aggressive, significantly dimming or eliminating reflections requiring overmuch blurring.

In implementation, we have computed visibility via jittered ray casting and inheritance similar to that of [7], storing visibility data in interactions as it is computed.

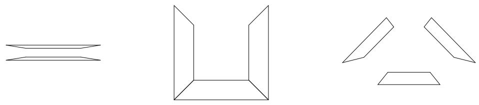

Figure 6: Geometric Configurations

5 Results

5.1 Growth in Transport

We have measured the growth in transport triples (linked interactions) versus n, the maximum number of elements at the finest level of subdivision, over parallel, perpendicular, and "oriented" patches (Figure 7). The corresponding geometries are shown in Figure 6. The graphs show linear or near linear behavior over each range — the graph of triples vs. n for the perpendicular case is slightly concave over the lower data points, but subsides to linear with further refinement.

In previous work [7] on hierarchical refinement for radiosity, it was shown that for error estimate proportional to F_{dji}, and sufficient refinement, each subpatch may only interact with other patches in a limited local neighborhood. As discussed in [7], each patch may thus participate in at most c interactions, for some constant c independent of n and k. Adaptive refinement thus generates at most $O(n)$ transport interactions. We will show a similar bound for discrete three point transport under glossy reflection.

Recall that the estimate for error in computed transport is proportional to $\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji}$. Our argument depends on two assumptions:

1. We may bound both ΔS_{kji} and S_{kji} by some $S_{\max}$.

As discussed below, the lower this $S_{\max}$, the smaller the magnitude of the leading coefficient underling the resulting bound.

Note that our argument thus does not apply to perfect specular reflection, as the corresponding BRDF incorporates the Dirac delta function [11]. Equivalently, the argument does not hold over f_g for $\kappa = \infty$ (inducing mirror reflection), as we can not provide a finite bound for S in this case.

For finite κ, however, the desired bound over glossy reflection is achieved by:

$$\frac{\kappa + 2}{8} \max(\cos^\kappa \theta_m) \max(\sec\theta_i) \max(\sec\theta_r)$$

The maxima over the secant terms are bounded by microfacet self-shadowing.

2. ΔF_{ji} and F_{ji} within our error estimate are at most proportional to F_{dji}.

Recall that we estimate F_{ji} as F_{dji}, and ΔF_{ji} as F_{dji}^2, thus satisfying this assumption.

Given these assumptions, estimated error is at most proportional to $S_{\max} F_{dji}$.

We may now show $O(n)$ growth, for sufficient refinement. Consider refinement over interaction ij under an error estimate at worst proportional to $S_{\max} F_{dji}$. The error estimate is thus proportional to F_{dji}, and therefore, for sufficient refinement, there are at most $O(n)$ such interactions, as discussed in [7].

Consider now an error satisfied link from ij to an interaction jk. For sufficient refinement under our subdivision scheme, we may assume that form factors F_{ij}, F_{ji}, F_{jk}, F_{kj} over p_i, p_j, and p_k are roughly equal. Furthermore, these satisfying form factors depend only on the error estimate, reflection function, and error bounds, not on n or k.

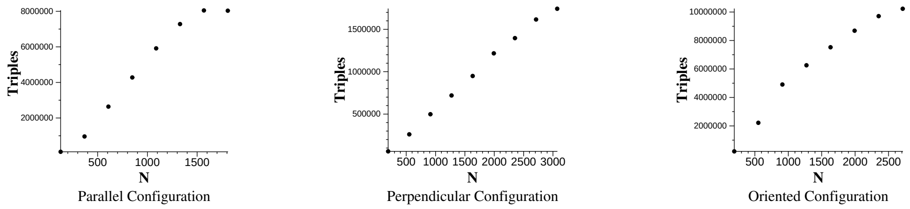

Figure 7: Triples vs. N over Geometry. Error bounds $e = 0.1$. Glossy exponent κ = 25. For oriented case $e = 0.005$.

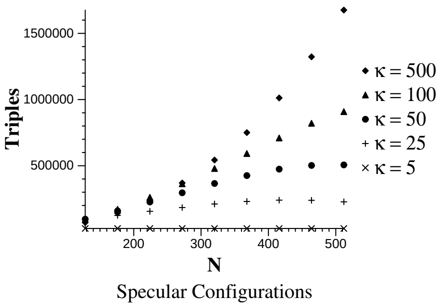

Figure 8: Triples vs. N over κ. The graph is over parallel polygons for which the error bounds and interpolygon distance have been doubled.

At worst the above form factors are such that $F_{**}S_{\max} < \texttt{Eps}$, where Eps is the most restrictive error bound. Note that, as stated above, F_{**} depends only on the error estimate, reflection function (ie. $S_{\max}$), and error bounds. Only some constant number of such form factors may be fitted over the directional hemisphere above p_j, and thus ij may only be linked to some constant number of interactions jk. The total number of linked interactions, and corresponding transport triples, is thus $O(n)$.

Note that the above argument, although it establishes the desired bound, may overstate the potential for links at a given interaction. For a given ij, much of the directional reflection into the hemisphere over p_j may not achieve $S_{\max}$, and may even be of maximum 0. That is, the analysis ignores the modulation between the paired error and value terms within the error estimate.

As κ increases in magnitude, the corresponding bound $S_{\max}$ must increase as well. We may thus expect greater growth in transport computation with higher specular exponent, as shown in Figure 8. Within this graph, growth is superlinear for $\kappa = 500$, though further trials over a higher range of $n = 500 \ldots 2000$ have shown that the rate subsides to linear as n increases, allowing sufficient refinement for the local neighborhood property to obtain.

Finally, we note that under specular reflection each element is reflected across every other element perfectly, and to a first approximation is visible from a constant number of other elements in the environment (at least in the case of a convex enclosed room; the analysis is complicated by occlusion and certain worst case alignments). Thus, the number of interactions is at least $O(n^2)$ — we conjecture that it is no worse than this bound.

5.2 Illumination and Refinement

Figure 9 shows illumination and meshing over surfaces of varying glossiness (specular exponent). Within each image, the reflecting surface is perpendicular to the diamond shaped light source, and we see the resulting reflection in the direction of the eye. Note the conformation of meshing to the highlight over each surface. The "stretched" nature of the highlight along the axis to the eye is characteristic of Torrance-Sparrow reflection over fairly oblique angles, and accounts for the increased sensitivity of meshing along this axis. The rightmost three images in the figure show the meshing from above. The illumination shown in these images is somewhat unusual - it shows the reflection to the eye as though it had been painted on the reflecting surface, and then viewed from a different location, directly above. The images in Figure 11 show similar eye/offset views for the reflection of a garish checkerboard.

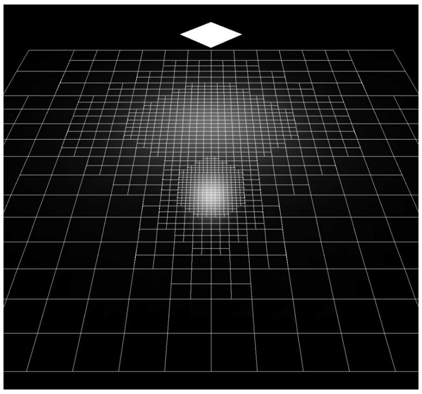

Figure 10: Meshing for glossy and diffuse reflection

Figure 9				
Max Elements		4160		
Max Triples		262144		
Computed	patches	elements	triples	time
κ = 25	790	593	6706 (2.6%)	2.2s
κ = 100	1290	968	24214 (9.2%)	8.0s
κ = 500	874	656	12106 (4.6%)	4.1s
Figure 10				
Max Elements		16448		
Max Triples		1048576		
Computed	patches	elements	triples	time
κ = 500	1578	1184	7834 (0.75%)	5.0s
Figure 12				
Max Elements		15138		
Max Triples		222385209344		
Computed	patches	elements	triples	time
κ = 500	6479	4866	70995 (0.00003%)	3m13s

Table 1: Image Statistics

The image in Figure 10 shows contrasting illumination and meshing induced by diffuse and glossy reflection. Note the distinct meshing for each highlight. Glossy reflection is at a less oblique angle, and thus both the highlight and meshing exhibit less distortion in the direction of the eye.

Note that these scenes are extremely simple — application to more complex environments is still very expensive, despite the employment of hierarchical methods. Motivated by the work of Smits et al. [14] in hierarchical radiosity, we are currently experimenting with importance and radiance weighting over three point transport — preliminary results of this work are shown in Figure 12. The given environment contains four reflectors: the broad face of each of the three "slabs" and the top of the central cube. In addition to the reflections seen in the slabs, note the play of light originating at the lamp at left, reflected off the cube top, and over the upper part of the green wall at right. Total potential transport triples over this environment at the finest level of subdivision is just over 222 billion — our system, under importance and radiance weighting, employs 70,995, a reduction to 3 hundred-thousandths of 1 percent.

Table 1 provides further statistics for the images. Timings are given for a Silicon Graphics indigo workstation with a single 50 MHz R4000 processor. The image shown in Figure 12 was generated after seven complete iterations (gathers to all interactions), and total time just over three minutes.

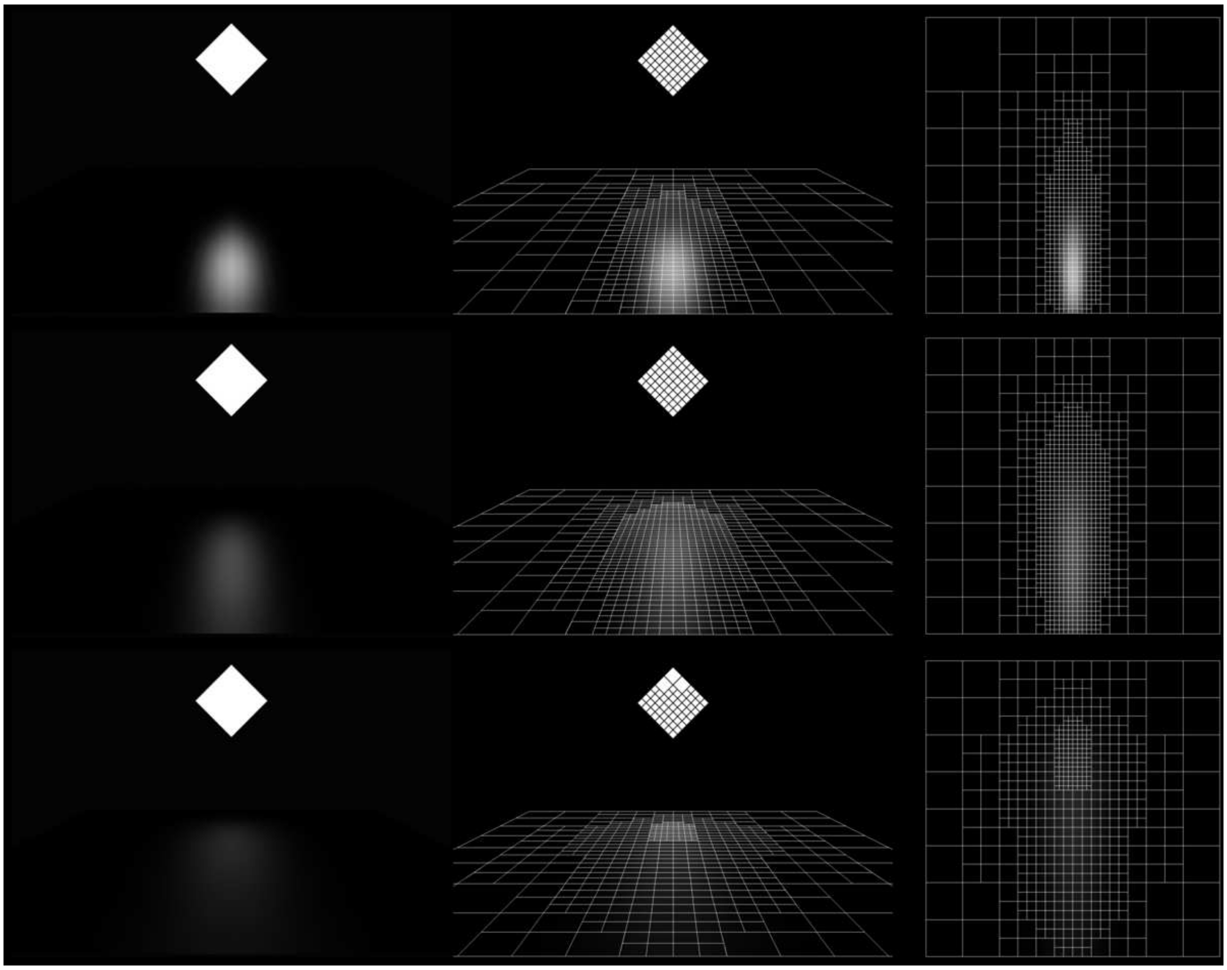

Figure 9: Illumination and Refinement

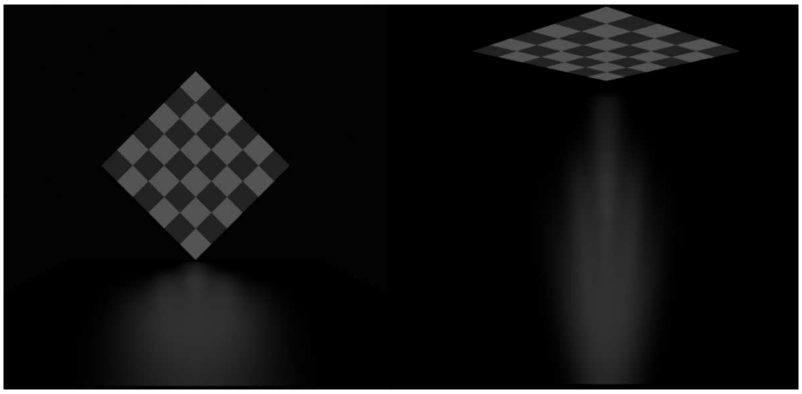

Figure 11: Eye and Offset Views

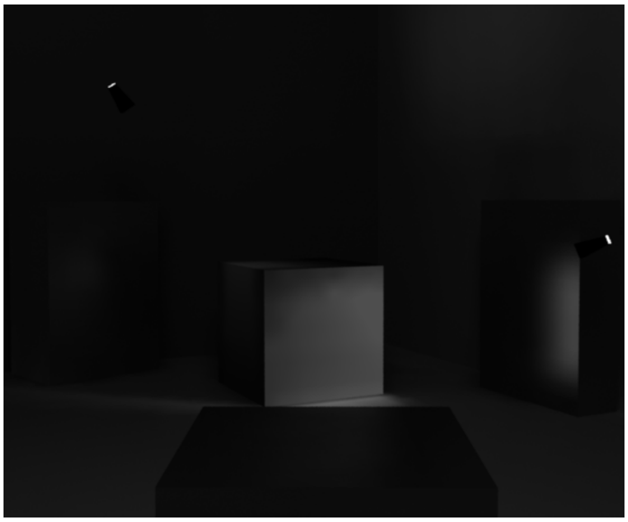

Figure 12: Cube and Slabs

6 Discussion

Recall the matrix formulation shown in Figure 2. For any n of reasonable size, the resulting n^3 matrix will be unmanageable — we have shown, however, that for sufficient refinement the n^3 entries in the matrix may be approximated to within user specified bounds by $O(n)$ subblocks. The gather and push/pull procedures described in preceding sections allow manipulation and solution over this representation. As discussed in [2], the resulting system may be shown to converge.

Growth in transport is more accurately described as $O(n + k^3)$, where k is the number of input polygonal patches within the environment, as opposed to elements. The k^3 term is generated by the initial examination of all polygon triples for reflection, and is subsumed by n as the number of elements increases. As the number of polygons in an environment grows, however, the k^3 term will become prohibitively large. As discussed in [14] with respect to the related problem under hierarchical radiosity, the capability to cluster as well as refine polygons would reduce the difficulty of unnecessary initial interactions. Clustering is arguably the most important open problem in the computation of global illumination.

The hierarchical approach described in this paper was derived by writing the rendering equation in a three point transport formulation. Another option would be to parameterize radiance by position and direction – we believe that a similar hierarchical approach could be employed with the method of discrete ordinates or spherical harmonics.

Finally, we note that, similarly to other algorithms for hierarchical illumination [7, 14], the algorithm described in this paper bounds estimated error over individual transport computations. As discussed in [14], bounding estimated error over individual transport does not easily or necessarily provide a rigorous bound for overall error in the solution. An analysis and means of computing such a bound over hierarchical illumination remains an interesting open problem.

7 Acknowledgements

This research was partially supported by equipment grants from Apple and Silicon Graphics Computer Systems and a research grant from the National Science Foundation (CCR 9207966). The authors would like to thank Dr. P. Prusinkiewicz for access to the graphics research facilities at the University of Calgary during the final stages of this work, and Deborah Fowler for her crucial assistance in shooting test images, paste up and much other support and encouragement. Thanks to Cullen Jennings and David Laur for all of their help recording images. We especially thank the anonymous referees for their many helpful comments and suggestions.

8 References

[1] Amanatides, J. (1992) Algorithms for the detection and elimination of specular aliasing. *Proc. Graphics Interface '92*, 86-93.

[2] Aupperle, L. (1993) Hierarchical algorithms for illumination. Doctoral Dissertation, Princeton University.

[3] Blinn, J.F. (1977) Models of light refection for computer synthesized pictures. *Computer Graphics* 11 (2), 192-198.

[4] Chen, S.E., Rushmeier, H.E., Miller, G., Turner, D. (1991) A progressive multi-pass method for global illumination. *Computer Graphics* 25 (4), 165-174.

[5] Cook, R.L. (1986) Stochastic sampling in computer graphics. *ACM Transactions on Graphics* 5 (1), 51-72.

[6] Hall, R. (1989) Illumination and color in computer generated imagery. Springer-Verlag, New York.

[7] Hanrahan, P., Salzman, D., Aupperle, L. (1991) A rapid hierarchical radiosity algorithm. *Computer Graphics* 25 (4), 197-206.

[8] Immel, D.S., Cohen, M.F., Greenberg, D.P. (1986) A radiosity method for non-diffuse environments. *Computer Graphics* 20 (4), 133-142.

[9] Kajiya, J.T. (1986) The rendering equation. *Computer Graphics* 20 (4), 143-150.

[10] Mitchell, D. (1992) Manuscript.

[11] Nicodemus, F.E., Richmond, J.C., Hsia, J.J., Ginsberg, I.W., Limperis, T. (1977) Geometrical considerations and nomenclature for reflectance. National Bureau of Standards monograph, no. 160.

[12] Shirley, P. (1990) A ray tracing method for illumination calculation in diffuse-specular scenes. *Proc. Graphics Interace '90*, 205-212.

[13] Sillion, F.X., Arvo, J.R., Westin, S.H., Greenberg, D.P. (1991) A global illumination solution for general reflectance distributions. *Computer Graphics* 25 (4), 187-196.

[14] Smits, B.E., Arvo, J.R., Salesin, D.H. (1992) An importance-driven radiosity algorithm. *Computer Graphics* 26 (2), 273-282.

[15] Torrance, K.E., Sparrow, E.M. (1967) Theory for off-specular reflection from roughened surfaces. *J. of the Optical Society of America* 57 (9), 1105-1114.

[16] Ward, G.J., Rubinstein, F.M., Clear, R.D. (1988) A ray tracing solution for diffuse environments. *Computer Graphics* 22 (3), 85-92.

Appendix: Error Analysis

Recall the error expression derived in Section 2.3:

$$\pi(\Delta F_{ji} S_{kji} + \Delta S_{kji} F_{ji})$$

In implementation, we have divided ΔS_{kji} into separate components for each subfactor of f_g. We thus have:

$$\frac{\kappa+2}{8}\Big(\Delta F_{ji}\frac{\cos^\kappa\theta_m}{\cos\theta_i\cos\theta_r} + \Delta\cos^\kappa\theta_m F_{ji}\frac{1}{\cos\theta_i\cos\theta_r} + \Delta\sec\theta_i F_{ji}\frac{\cos^\kappa\theta_m}{\cos\theta_r} + \Delta\sec\theta_r F_{ji}\frac{\cos^\kappa\theta_m}{\cos\theta_i}\Big)$$

In implementation, the refinement procedure of Section 3.2 takes an additional argument, `Ceps`, against which the two estimates of error in reciprocal cosine are tested.

We are left with the computation of $\Delta\sec\theta_i$, $\Delta\sec\theta_r$, and $\Delta\cos^\kappa\theta_m$. The variance (and associated error) in these cosine terms over given patches A_i, A_j, A_k is determined by the set of possible $\vec{\omega}_i$, $\vec{\omega}_r$ lying between the patches (we dispense with $'$ notation in this section).

Consider patches A_i, and A_j (Figure 4): we enclose these patches in spheres S_i, S_j with centers c_i, c_j, and radii r_i, r_j, respectively. For the moment we will assume that the interiors of S_i and S_j do not intersect, and thus there exists a tangent cone lying between the spheres.

Note that this cone is a right circular cone centered on the line joining c_i and c_j. Consider the nappe containing S_i: it may be regarded as a cone of direction vectors centered about the vector $c_i - c_j$. We will call this vector cone C_i. If p_i and p_j are any two points on or in S_i, S_j, then the vector $p_i - p_j$ lies within C_i. C_i thus bounds the set of possible $\vec{\omega}_i$. We may characterize C_i by the angle α_i defined by its axis, $c_i - c_j$, and boundary — cone C_r and angle α_r may be similarly defined over A_j and A_k. If either pair of spheres intersect, we set the corresponding $\alpha = \pi$. We may easily compute maxima and minima for $\sec\theta_i$ and $\sec\theta_r$ given C_i and C_r, and may then compute error in estimation as $(\max - \min)/2$.

The cones C_i and C_r centered about $\vec{\omega}_i$ and $\vec{\omega}_r$ induce a similar cone of variation about $\vec{\omega}_m$ (Figure 5). Application of basic spherical trigonometry yields [2]:

$$\alpha_m \leq \arcsin\min\Big(\frac{\sin(\alpha_i/2) + \sin(\alpha_r/2)}{\vec{\omega}_i \cdot \vec{\omega}_m}, 1.0\Big)$$

Given α_m, determination of $\max(\cos^\kappa\theta_m)$, $\min(\cos^\kappa\theta_m)$, and thus $\Delta\cos^\kappa\theta_m$ immediately follows.

Having computed these estimates and maxima, and incorporating the estimates for form factor computation, we may bound and estimate error in transport as:

$$\begin{aligned}\tfrac{\kappa+2}{8}\big(&F_{dji}^2\max(\cos^\kappa\theta_m)\max(\sec\theta_i)\max(\sec\theta_r) + \\ &\Delta\cos^\kappa\theta_m F_{dji}\max(\sec\theta_i)\max(\sec\theta_r) + \\ &\Delta\sec\theta_i F_{dji}\max(\cos^\kappa\theta_i)\max(\sec\theta_r) + \\ &\Delta\sec\theta_r F_{dji}\max(\cos^\kappa\theta_i)\max(\sec\theta_i)\big)\end{aligned}$$

It is this error measure that we employ in our implementation.