A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection

Larry Aupperle
Pat Hanrahan

Meshing for glossy and diffuse reflection.

Princeton University
Department of Computer Science
Princeton, New Jersey 08544
January 1993

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 interactions is shown to be $O(n \log n)$ for sufficient refinement, where n is the number of elements at the finest level of subdivision within the environment — in trials, this growth has been nearly linear. Naive implementation of three point transport would require $O(n^3)$ 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, which make the assumption that surfaces are Lambertian, have been quite successful. Unfortunately, very few materials are purely diffuse, and up to this point efficient solution techniques have not been developed for more general specular or glossy reflectance functions.

The rendering equation is an integral equation, and the solutions to complicated integral equations are generally solved using either Monte Carlo or finite element techniques. Monte-Carlo algorithms sometime go under the name of distributed or stochastic ray tracing and are the most commonly used in computer graphics (e.g. see Cook 1986; Kajiya 1986; Ward et al. 1988; Heckbert 1990; Shirley 1990; Chen et al. 1991). 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 to the general case, and with limited success. For example, Immel et al. (1986) discretized the radiance into a lattice of cubical environment maps, and solved the resulting system. More recently, Sillion et al. (1991) use a mesh of spherical harmonic functions to represent the radiance, and solve 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 slightly different, and based on Kajiya's original formulation of the rendering equation. In his view, the rendering equation is expressed as a 3-point transport equation. 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. One advantage of this approach is that it 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. In particular, 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 $O(n \log n)$ blocks, in a way very similar to our recent hierarchical radiosity algorithm (Hanrahan et al. 1991). In that paper we showed how the n^2 form factor matrix could be approximated with $O(n)$ blocks, resulting in a very efficient algorithm in both space and time. Although the results presented in this paper are preliminary, we believe a direct finite element approach along these lines will ultimately lead to a fast, efficient algorithm.

This paper is a condensed version of the first author's Ph. D. thesis. The first section applies the finite element method to the three-point rendering equation. In the second section a simple recursive refinement algorithm is presented. This algorithm builds a hierarchical set of piecewise constant, interpatch radiances; this refinement procedure is driven by the desire to approximate the integral operator representing the reflectance accurately. This section also describes the iterative solution technique. The third section discusses some details of the implementation, in particular the details on how errors in the glossy reflection function are estimated. Finally, the fourth section presents some experiments and results.

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 functions — 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.

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''.

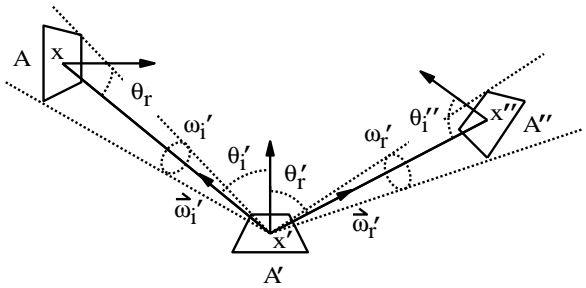

Figure 1 Geometry of Reflection

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, f_r (Nicodemus et al. 1977), 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', we have:

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

We may then multiply both sides of this equation by $\cos\theta_r'$, and 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

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

Note that G is very similar to a differential form factor, and is identical to the geometry term G employed in (Bouville et al. 1990).

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

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

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. Note that this assumption of near constant radiance implies the employment of piecewise-constant basis functions in the resulting finite element method. An approach applying higher

order elements to three point transport presents an interesting open problem. Under our assumption, the left side of equation (1) may 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:

$$\pi L_{jk} A_j F_{jk} = \pi \sum_i L_{ij} A_i F_{ij} R_{ijk} = \pi \sum_i L_{ij} A_k F_{kj} R_{kji} = \pi \sum_i L_{ij} A_j F_{jk} R_{kji}$$

by the reciprocity of form factors, and thus:

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

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

$$L_{jk} = E_{jk} + \sum_i L_{ij} R_{kji} \tag{2}$$

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:

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

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 quantity $\rho(A_i, A_j, A_k)$:

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

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:

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

We define $\rho(A_i, A_j, A_k)$ to be *area reflectance.*

Area reflectance, $\rho(A_i, A_j, A_k)$, is the fraction of the radiant flux transported from A_i incident to A_j that is reflected in the direction of area A_k. We chose the term area reflectance for this quantity as it is defined and expressed entirely over surface areas. Given the surface to surface character of global illumination under computer graphics, the concept of area reflectance appears well suited to the field. Note that area reflectance is similar to biconical reflectance (Nicodemus et al. 1977), 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. $A_i F_{ij} R_{ijk} = A_k F_{kj} R_{kji}$

where equality is achieved in property 1 over complete enclosures.

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, $S_{ijk} = S_{kji}$.

Note that the average value of $G(x', x)$ over A_i and A_j is:

$$\frac{1}{A_iA_j}\int_{A_i}\int_{A_j} G(x, x')dx'dx,$$

whereas:

$$F_{ji} = \frac{1}{\pi A_j}\int_{A_i}\int_{A_j} G(x, x')dx'dx$$

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 f_{ji} and s_{kji} be error estimates for computed F_{ji} and S_{kji}, respectively. We then have an estimate for area reflectance in the form:

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

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

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 Discrete 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 j is defined by two patches `j->from` and `j->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 i contained in the list `gather`. Subinteractions `nw`, `sw`, `se`, `ne` are the children of j, induced by subdivision over either `from` or `to`. Only one defin-

ing patch may be subdivided relative to a given interaction, thus avoiding double sets of children associated with a single parent. 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 i, j (such that `i->to == j->from`), to ensure that computed reflectance across the interaction pairs, and associated patch triples, satisfies user specified error bounds. If a given interaction pair i, j is satisfactory, the interactions are linked to record that radiance may be gathered from i to j, 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 *i, Interaction *j, float Feps, float Seps, float Aeps)
{
    float feps, seps;

    feps = GeometryErrorEstimate(i);
    seps = ThreePointErrorEstimate(i, j);

    if (feps < Feps && seps < Seps)
        link(i, j);

    else if (seps >= Seps) {

        switch(SubdivS(i, j, Aeps)) {

        case PI:

            Refine(i->nw, j, Seps, Feps, Aeps);
            Refine(i->sw, j, Seps, Feps, Aeps);
            Refine(i->se, j, Seps, Feps, Aeps);
```

```
            Refine(i->ne, j, Seps, Feps, Aeps);

            break;

        case PJ: ...    /* refine over children of interactions i and j */
        case PK: ...    /* refine over children of interaction j        */

        case NONE:
            link(i, j);
        }
    }

    else {    /* feps >= Feps */

        switch(SubdivG(i, j, Aeps)) {

        case PI: ...    /* refine over children of interaction i        */
        case PJ: ...    /* refine over children of interactions i and j */
        case PK: ...    /* refine over children of interaction j        */

        case NONE:
            link(i, j);
        }
    }
}
```

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` (F_ϵ) and `Seps` are the bounds for geometric and three point error, respectively; `Aeps` specifies the minimum area a patch may possess and still be subdivided. `GeometryErrorEstimate` and `ThreePointErrorEstimate` provide estimations for $\pi f_{ji} S_{kji}$ and $\pi s_{kji} F_{ji}$.

`SubdivS` and `SubdivG` control refinement for three point 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 criteria for the determination of such patches — a large

form factor F_{dpq} indicates that patch q is of large relative size. Care must be taken in subdivision, however, to ensure that a given interaction has only one set of children.

We say that an interaction *allows* subdivision of one of its component patches if that patch is of area at least `Aeps`, and either it is of greater form factor relative to the other component patch, or the other patch is of area less than `Aeps`. In the event of equal form factors between component patches, we may break the tie in any consistent fashion over the interaction.

Subdivision selection is then as follows: if both interactions allow subdivision of p_j, the patch is chosen and subdivided. Otherwise, at least one interaction "prefers" an outside patch. If there is only one such preferred patch, it is chosen and subdivided. Given two allowed 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 geometry 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.

The given subdivision criteria thus allows for both the selection of patches of large relative size, and the preservation of the structure of the interaction hierarchy.

3.3 Gathering Radiance

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

```
Gather(Interaction *j)
{
    Interaction *i;

    if (j) {

        j->Lg = 0;

        ForAllElements(i, j->gather)
            j->Lg += i->L * Reflectance(i, j);

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

where the linkage of two interactions during refinement causes the receiving inter-

action to add a pointer to the other to its `gather` list.

We gather radiance into `j->Lg` rather than directly into `j->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 (Hanrahan et al. 1991) 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 (Aupperle 1993), radiance may be pushed unchanged within the interaction hierarchy, and area averaged as it is pulled from child to parent.

4 Glossy Reflection Functions

In this section, we describe the glossy reflection function used in our current implementation. In addition, we discuss in detail the methods used to detect whether it is approximately constant over a pair of interactions.

4.1 The Reflection Function

We employ the general model for glossy/rough specular reflection developed by Torrance and Sparrow (Torrance et al. 1967), and introduced and adapted to image synthesis by Blinn (Blinn 1977) and Cooke and Torrance (Cooke et al. 1982):

$$f_{TS}(\vec{\omega}_i, \vec{\omega}_r) = \frac{DGF}{4 \cos\theta_i \cos\theta_r}$$

where D is the microfacet distribution function, G is a function determining self-shadowing over microfacets, and F is Fresnel reflectance.

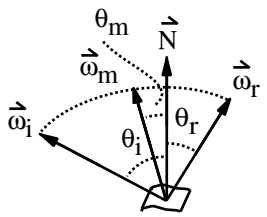

Figure 2 Reflection and Mirror Orientation

In our implementation we have neglected Fresnel reflectance, and employed

$$D(\theta_m, k) = \frac{k+2}{2\pi} \cos^k \theta_m$$

as our facet distribution function, where θ_m the angle made to the surface normal by $\vec{\omega}_m$, the microfacet mirror orientation normal lying halfway between $\vec{\omega}_i$ and $\vec{\omega}_r$ (Blinn 1977) (Figure 2), and $\frac{k+2}{2\pi}$ is a normalization for projected facet area (Mitchell 1992). We will discuss facet self shadowing briefly below.

The resulting glossy reflection function is thus

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

Before we may employ this reflection function within our system, we must ensure that it satisfies conservation of energy (Ward 1992) — otherwise a sequence of multiple reflections may lead to an arbitrarily large increase in the amount of energy within a modelled environment. Conservation of energy may be expressed as the requirement that directional-hemispherical reflectance (Nicodemus et al. 1977):

$$\rho(\vec{\omega}_i, 2\pi) = \int_{2\pi} f_r(\vec{\omega}_i, \vec{\omega}_r) \cos\theta_r d\omega_r$$

be no greater than one over all $\vec{\omega}_i$. Unfortunately, direct evaluation of this integral for arbitrary $\vec{\omega}_i$ appears to be extremely difficult over f_g.

Unable to normalize for conservation of energy directly, we must try to ensure that our function is sufficiently physically consistent to inherently preserve conservation. The Torrance Sparrow model is derived directly from the physics of microfacet reflection, $F \equiv 1$ will not violate conservation of energy, and the facet distribution function has been normalized for physical consistency. The factor that is missing is that of self-shadowing.

For near specular surfaces, facet masking does not become critical until relatively high θ_i or θ_r, as confirmed by the shading function developed by Sancer (Sancer 1969) shown in the polar graph (Figure 3) derived from (Hall 1989).

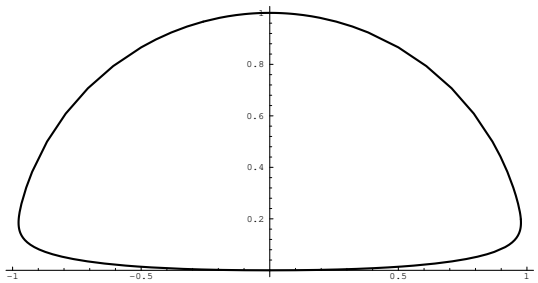

Figure 3 $G_s(\theta_i, \theta_r = 0.5) \quad m = 0.1$

Given the sudden fall off that G exhibits near the horizon for near specular surfaces, the implemented system simply clamps G to 0 when θ_i or θ_r exceeds a preset θ_{bound}. This scheme should preserve conservation of energy, and serves as a crude approximation to the shadowing function; however in full implementation a better strategy, particularly in view of the well behaved nature of G, would be to employ a much fuller tabulation of the function. Given the monotonicity of G over θ, the function could be incorporated into the error analysis presented below.

4.2 Error Analysis

Recall the general expression for error derived in Section 2.3:

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

As shown in (Hanrahan et al., 1991), the use of a differential to area form factor F_{dji} as an estimator to F_{ji} is related to the actual form factor as:

$$F_{ji} \approx F_{dji}(1 - F_{dji}) = F_{dji} - F_{dji}^2$$

We will employ F_{dji} as our estimate to F_{ji}, and calculate f_{ji} as F_{dji}^2.

We now consider the error estimate, s_{kji}, for the *brdf*:

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

In implementation, we have divided s_{kji} into separate components for each subfactor of f_g. Let $\Delta \sec\theta_i$, $\Delta \sec\theta_r$, $\Delta \cos^k \theta_m$ be estimated error in computed $1/\cos\theta_i$,

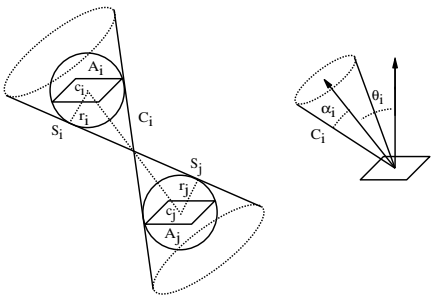

Figure 4 Estimating Cones

$1/\cos\theta_r$, $\cos^k\theta_m$ for patches A_i, A_j, A_k; then:

$$\begin{aligned} R_{kji} &= \frac{k+2}{8}(F_{ji}+f_{ji})(\sec\theta_i+\Delta\sec\theta_i)(\sec\theta_r+\Delta\sec\theta_r)(\cos^k\theta_m+\Delta\cos^k\theta_m) \\ &= \frac{k+2}{8}F_{ji}\frac{\cos^k\theta_m}{\cos\theta_i\cos\theta_r} + \frac{k+2}{8}(f_{ji}\frac{\cos^k\theta_m}{\cos\theta_i\cos\theta_r}+\Delta\sec\theta_i F_{ji}\frac{\cos^k\theta_m}{\cos\theta_r} \\ &\quad +\Delta\sec\theta_r F_{ji}\frac{\cos^k\theta_m}{\cos\theta_i}+\Delta\cos^k\theta_m F_{ji}\frac{1}{\cos\theta_i\cos\theta_r}+\cdots) \end{aligned}$$

We will assume the errors are of less magnitude than the estimates, ignoring the higher order terms, and employing the first four as our estimate for error. 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 $\Delta\cos^k\theta_m$. 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 will dispense with $'$ notation in this section, as all measures are against the reflecting surface, A_j).

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 present 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, and computed by:

$$\alpha_i = \arcsin\frac{r_i + r_j}{D_{ij}}$$

where D_{ij} is the distance between c_i and c_j. Cone C_r and angle α_r may be similarly

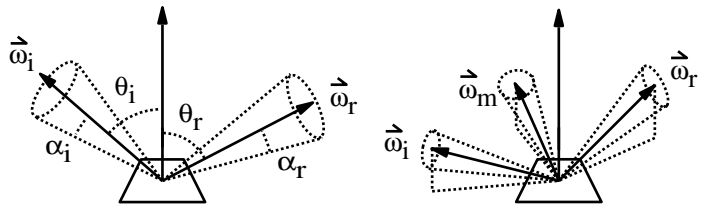

Figure 5 C_i, C_r, and C_m

defined over A_j and A_k. If either pair of spheres intersect, we set the corresponding $\alpha = \pi$.

Given cones C_i and C_r (Figure 5) bounding variation in $\vec{\omega}_i$ and $\vec{\omega}_r$, we may easily compute maximum and minimum bounds for $\sec\theta_i$, $\sec\theta_r$:

$$\max(\sec\theta) = \begin{cases} \sec(\theta+\alpha) & \text{if } (\theta+\alpha) \le \theta_{\text{bound}} \\ \sec\theta_{\text{bound}} & \text{otherwise} \end{cases}$$

$$\min(\sec\theta) = \begin{cases} \sec(\theta-\alpha) & \text{if } \theta > \alpha \\ 1 & \text{otherwise} \end{cases}$$

We 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 (Aupperle 1993):

$$\alpha_m \le \arcsin\min\left(\frac{\sin(\alpha_i/2)+\sin(\alpha_r/2)}{c}, 1.0\right)$$

where $c = \vec{\omega}_i \cdot \vec{\omega}_m = \vec{\omega}_r \cdot \vec{\omega}_m$.

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

Recall the estimate for error in R_{kji}:

$$\frac{k+2}{8}F_{ji}\left(f_{ji}\frac{\cos^k\theta_m}{\theta_i\cos\theta_r} + \Delta\sec\theta_i F_{ji}\frac{\cos^k\theta_m}{\cos\theta_r} + \Delta\sec\theta_r F_{ji}\frac{\cos^k\theta_m}{\cos\theta_i} + \Delta\cos^k\theta_m F_{ji}\frac{1}{\cos\theta_i\cos\theta_r}\right)$$

we replace F_{ji}, f_{ji} by F_{dji} and F^2_{dji}, respectively, and $\frac{\cos^k\theta_m}{\cos\theta_i\cos\theta_r}$ by their respective maxima, to estimate error as:

$$\frac{k+2}{8}\left(F^2_{dji}\max(\cos^k\theta_m)\max(\sec\theta_i)\max(\sec\theta_r) + \Delta\cos^k\theta_m F_{dji}\max(\sec\theta_i)\max(\sec\theta_r)\right.$$
$$\left.+\Delta\sec\theta_i F_{dji}\max(\cos^k\theta_i)\max(\sec\theta_r) + \Delta\sec\theta_r F_{dji}\max(\cos^k\theta_i)\max(\sec\theta_i)\right)$$

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

4.3 Clamping

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 k, 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 in (Amanatides 1992). 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 k to broaden the resulting reflection over the estimated cones. We then renormalize the resulting blurred function, as described in (Amanatides 1992), to prevent amplification of its low frequency components.

4.4 Visibility

The development of Section 2.1 does not take into account visibility between surfaces. Similarly to visibility under radiosity, we may introduce a term v to the geometry expression $G(x, x')$:

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

where $v(x, x')$ is 1 if points x and x' are mutually visible, and 0 otherwise.

In implementation, we compute visibility through jittered ray casting and inheritance similar to that of (Hanrahan et al. 1991), storing visibility data in interactions as it is computed.

5 Results

5.1 Growth in Transport

In this section we will examine how the number of transport triples generated during adaptive refinement grows as greater subdivision is allowed over the environment. That is, we wish to characterize the dependence of triples versus n, the maximum number of elements at the finest level in subdivision in the environment. Three factors may affect this dependency: the geometric configuration of the environment, the given error bounds, and the glossiness exponent, k.

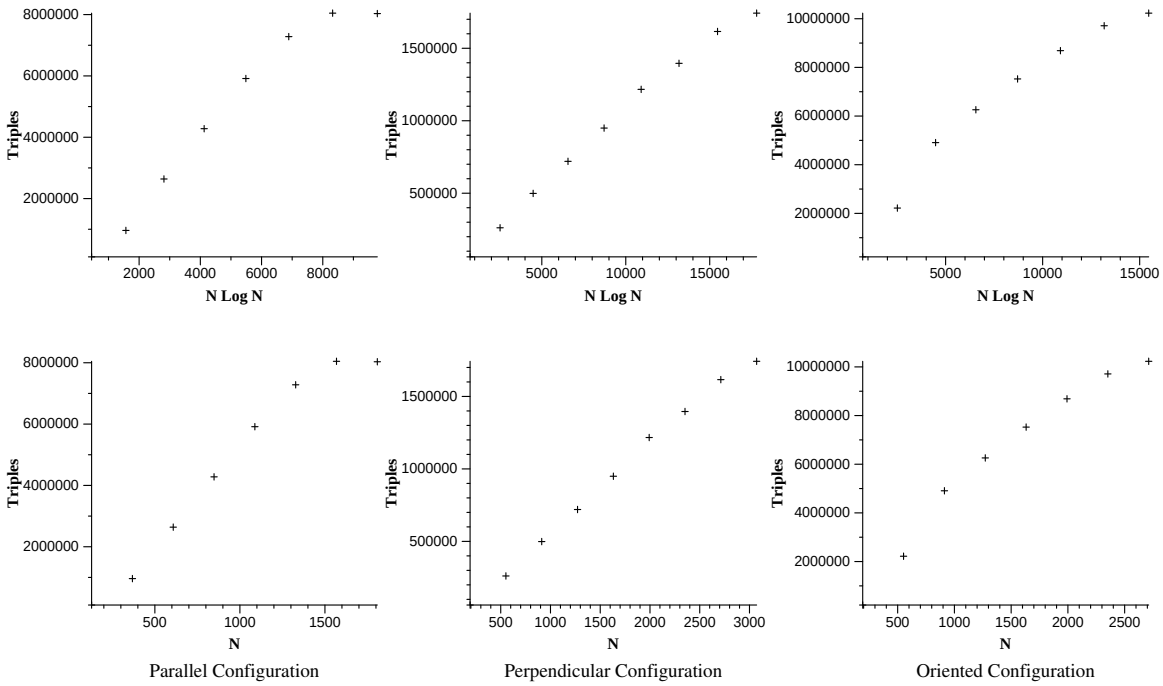

Figure 6 Triples vs. N over Geometry. Error bounds $e = 0.1$. $k = 25$. Over oriented polygons $e = 0.1$ resulted in a flat graph, error bounds were thus reduced to $e = 0.005$.

We have measured the growth in triples subdivision for parallel, perpendicular, and "oriented" patches (Figure 6). The corresponding geometries are shown in Figure 7. We felt that the case of two relatively close parallel patches would be very demanding on algorithm performance as it involved close geometry with great potential for inter-reflection. The perpendicular case was chosen as a sort of opposite boundary — in addition, it requires refinement over patches that approach each other arbitrarily closely. The oriented patches were chosen to lie "between" the other cases, and also as suited to the geometry of reflection.

The graphs in Figure 6 show linear or near linear behavior over each range — the graph of triples vs. n for the perpendicular case is slightly concave up over the lower data points, and thus superlinear although sub $n \log n$. Note, however, that these curves flatten out toward the high end of each range, particularly in the parallel and oriented cases. As greater subdivision is allowed, error is satisfied over nearly all refined triples, reducing potential for further growth. We have tested the parallel (and arguably most demanding) configuration over higher ranges in n, reducing the distance between patches by half to avoid a flat graph, and yielding similar results.

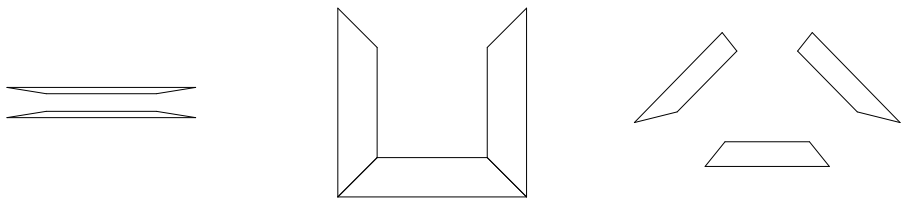

Figure 7 Geometric Configurations

We are left with the question of *why* adaptive refinement induces growth as shown in Figure 6. In previous work (Hanrahan et al. 1991) on hierarchical refinement for radiosity, it was shown that for error estimate $F_{dj\,i}$ and sufficient refinement, each subpatch may only interact with patches in a limited local neighborhood. As a result, each patch participates in at most k interactions, for some constant k, bounding the total interactions to $O(n)$.

Within the error estimate derived in Section 4.2, no term is greater than:

$$\frac{k+2}{8}\max(\cos^k\theta_m)\max(\sec\theta_i)\max(\sec\theta_r)F_{dj\,i}$$

It may appear that this bound is meaningless, as both $\max(\sec\theta)$ terms approach infinity as θ approaches $\frac{\pi}{2}$. Self shadowing, however, bounds this growth, and, under the crude self shadowing we employ in implementation, neither maxima may exceed $\sec\theta_{\text{bound}}$.

As a result, our error estimate is no more than a constant times the error estimate employed in (Hanrahan et al. 1991). Thus, for sufficient subdivision, a local neighborhood argument similar to that for hierarchical radiosity will hold:

Consider a given interaction during refinement under three point transport, as opposed to hierarchical radiosity. There are two ways in which this interaction may be "dragged" lower (and subdivided) within the three point hierarchy than it would for subdivision under radiosity: through the stricter error requirements induced by the constant multiple imposed on its error estimate, in which case the local neighborhood argument will still apply, or by an interaction with which it is associated in a triple.

We will count all triples, and associated dragged down interactions, at the "lower" interaction in the triple. The local neighborhood argument applies to these lower interactions, and thus there are at most $O(n)$. A lower interaction i may drag down (and be linked with) interactions that are "naturally" connected to patches higher in the patch hierarchies to which the component patches of i belong. By the quadtree nature of these hierarchies, there are at most $O(\log n)$ subpatches above each component patch. Furthermore, by the local neighborhood argument, there are

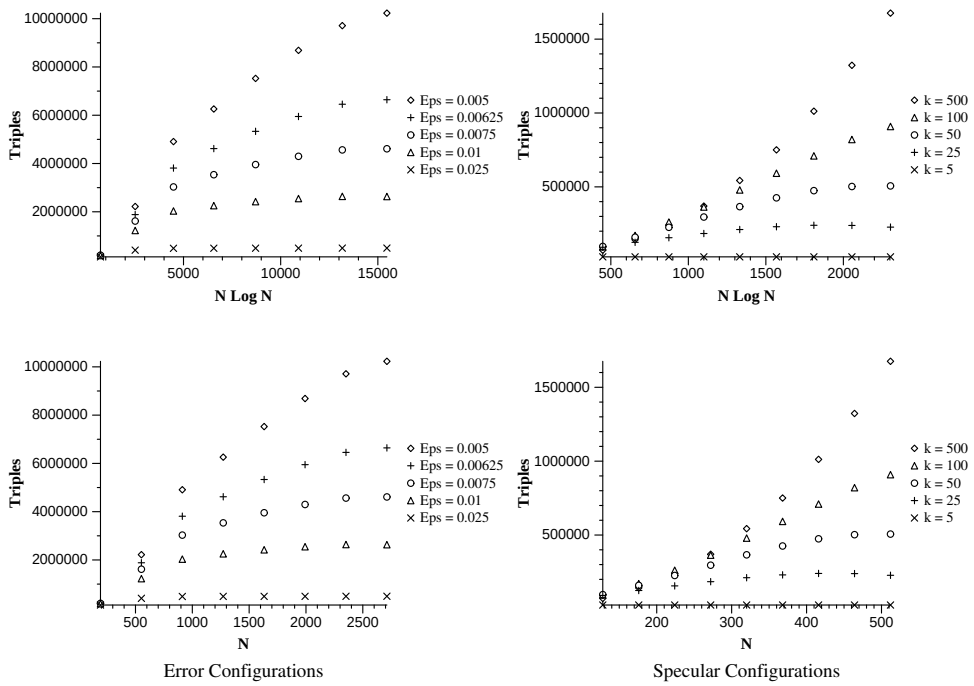

Figure 8 Triples vs. N over E and k. The first graph is over the oriented configuration. The second is over parallel polygons for which the error bounds and interpolygon distance have been doubled.

at most some constant k "naturally" connected interactions at each level. Over $O(n)$ lower interactions, each linked with at most $O(\log n)$ dragged down interactions, we have at most $O(n \log n)$ triples.

Thus, sufficient refinement will induce a local neighborhood restriction over interactions, and we may expect resulting growth to be at most $O(n \log n)$.

We consider now the dependencies upon the error bounds and specular exponent (Figure 8). We expect reduction of the error bound to induce greater refinement and a correspondingly greater rate of growth — as confirmed by the given graph.

The dependence upon the specular exponent is perhaps more interesting: in the above discussion k appears as a coefficient within the multiple applied to F_{djdi}, the error estimate in (Hanrahan et al. 1991). High values of k thus correspond to very restrictive error bounds over F_{djdi}, and we may expect both the rate of growth and the value of n required for the local neighborhood criteria to increase with k. This relation is shown in the graph — note that growth over the given

range is super $n \log n$ for $k = 500$, 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, an interesting degenerate case is when reflecting surfaces are perfect mirrors. In this situation, 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. Figure 10 is a false color image of the same configurations – intensity over the reflecting surface is determined by the number times the adaptive refinement algorithm visits a given subpatch/element, and again shows conformation with the highlight.

The top three images of Figure 11 show the same meshing from above. The "stretched" nature of the highlight along the axis to the eye, visible both in Figure 9 and in Figure 11 is characteristic of Torrance-Sparrow reflection over fairly oblique angles, and accounts for the increased sensitivity of meshing along this axis. 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 bottom four images in Figure 11 show similar eye/offset views for reflection of a garish checkerboard.

The image on the cover of this paper 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.

The final image, Figure 12, shows an environment containing 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 floating lamp at left, reflected off the cube top, and over the upper part of the green wall at right. This image incorporates radiance and importance weighting over error based on the work of (Smits et al. 1992). Total potential transport triples over this

Figure 9 Illumination and Refinement

Figure 10 False Color Image

Figure 11 Eye and Offset Views

Figure 12 Cube and Slabs

environment at the finest level of subdivision is just over 222 billion — our system employs 428,000, a reduction to 2 ten-thousandths of 1 percent.

Table 1 provides further statistics for the images. Note the reduction in computed versus potential transport triples. As the number of elements grows, $O(n^3)$ increasingly outstrips $O(n \log n)$, amplifying the percentage reduction — compare the statistics for Figure 9 to those for the front cover. This tendency is further reinforced by the radiance and importance weighting employed for Figure 12.

Within the table, the abbreviation a. t. i. represents the average time for a complete solution iteration (gather to all interactions), on a silicon graphics workstation with a single 33 MHz IP7 processor (a 40MHz IP7 processor in the case of Figure 12). The solution shown in Figure 12 was achieved after five iterations (14 minutes total time) and satisfactory convergence — the maximum iteration time was 5 minutes, 44 seconds.

Finally, it may puzzle the reader that, for Figure 9 and the front cover, both potential and computed interactions outnumber the corresponding triple counts. In these simple configurations all viable potential triples are composed of pairs of interactions from the light source to the reflector, and from the reflector to the eye. The number of potential triples is thus equal to the number of interpatch light source to reflector interactions. The interaction count includes these interpatch interactions, plus the

Figure 9					
Potential Elements	4160				
Potential Interactions	266304				
Potential Triples	262144				
Computed	patches	elements	interactions	triples	a. t. i.
k = 25	790	593	9660	6706 (2.6%)	5.4s
k = 100	1290	968	33500	24214 (9.2%)	15.0s
k = 500	874	656	17012	12106 (4.6%)	7.8s
Front Cover					
Potential Elements	16448				
Potential Interactions	1065024				
Potential Triples	1048576				
Computed	patches	elements	interactions	triples	a. t. i.
k = 500	1578	1184	18756	7834 (0.75%)	9.97s
Figure 12					
Potential Elements	15114				
Potential Interactions	173489930				
Potential Triples	222026530816				
Computed	patches	elements	interactions	triples	a. t. i.
k = 500	9015	6768	146233	428036 (0.0002%)	2m48s

Table 1 Image Statistics

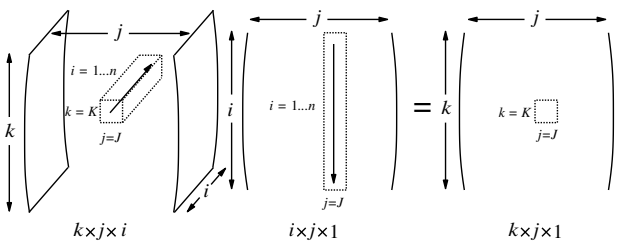

Figure 13 A Reflection Product

potential of an interaction from every element to the eye. For more complex scenes, such as that of Figure 12, potential triples far exceed potential interactions.

In the case of computed interactions versus triples it should be noted that in the course of refinment triples for which maximum transport is found to be zero are eliminated and not counted. The system does not, however, eliminate any interactions generated in the course of this refinement. For the simple configurations of Figure 9 and the front cover, there is very little reuse of interactions by multiple triples, and thus such "orphaned" interactions are relatively plentiful.

6 Discussion

Recall equation (2):

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

If we ignore E_{jk} and write the equation as:

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

the three dimensional character of R_{kji} over indices i, j, k leads (a computer scientist at least) naturally to a three dimensional matrix formulation. Consider a product over an $n \times n \times n$ R_{kji} "matrix" and a $n \times n \times 1$ L_{ij} matrix producing an $n \times n \times 1$ matrix of reflected radiances, as shown in Figure 13.

For any n of reasonable size, the resulting n^3 matrix will be unmanageable — we have, however, shown that for sufficient refinement the n^3 entries in the matrix may be approximated to within user specified bounds by $O(n \log n)$ subblocks. The gather and push/pull procedures described in preceding sections allow manipulation and solution over this representation. As shown in (Aupperle 1993), the three dimensional matrix may be rewritten in two dimensions, incorporating L_{jk}, to represent the operation $E_{jk} = L_{jk} - \sum_i R_{kji} L_{ij}$. The resulting matrix is diagonally dominant, confirming eventual convergence.

Growth in transport may be described as $O(n \log n + k^3)$, where k is the number of polygons 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 $n \log n$ 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 (Smits et al. 1992) 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. Employing nested bounding spheres over groups of polygons, and the resulting cone tests, would represent a partial approach to this problem — such clustering would not be viable for the actual computation of transport, however, only for the elimination of triples for which transport is not possible. Clustering is arguably the most important open problem in the computation of global illumination.

We are experimenting with techniques for enhancing/accelerating our algorithm, in particular radiance and importance driven approaches (Smits et al. 1992). The image in Figure 12 was generated under such weighting. The inherent directional dependency of glossy reflection makes it a promising candidate for an importance driven approach.

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 paramaterize radiance by position and direction – we believe that a similar hierarchical approach could be employed with the method of discrete ordinates or spherical harmonics. A comparison of these approaches is suggested as an open research problem.

The current technique for estimation of error in area reflectance requires complex geometric arguments combined with interval techniques over a relatively unsophisticated kernel. As discussed in (Gortler et al. 1993), wavelet theory provides a very general means for error estimation when approximating any smooth kernel, and has great potential for employment in hierarchical algorithms for more complex reflection functions.

Finally, we note that, similarly to other algorithms for hierarchical illumination (Hanrahan et al. 1991, Smits et al. 1992), the algorithm described in this paper bounds estimated error over individual transport computations. As discussed in (Smits et al. 1992), bounding estimated error over individual transport does not easily or necessarily provide a rigorous bound on over all 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, and for much other support and encouragement. Thanks to David Laur for all of his help recording images.

8 References

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

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

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

Bouville, C., Bouatouch, K., Tellier, P., Pueyo, X. (1990) A theoretical analysis of global illumination models. *Proc. Eurographics Workshop on Photosimulation, Realism, and Physics in Computer Graphics*, 53-66.

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.

Cook, R.L., Torrance, K.E. (1982) A reflectance model for computer graphics. *ACM Transactions on Graphics* 1 (1), 7-24.

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

Gortler, S., Schroeder, P., Cohen, M., Hanrahan, P. (1993) Wavelet Radiosity. Submitted *SIGGRAPH '93*.

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

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

Heckbert, P.S. (1990) Adaptive radiosity textures for bidirectional ray tracing. *Com-*

puter Graphics 24 (4), 145-154.

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

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

Mitchell, D. (1992) Manuscript.

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.

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

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.

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

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.

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

Ward, G.J. (1992) Measuring and modeling anisotropic reflection. *Computer Graphics* 26 (2), 265-272.