Wavelet Methods for Radiance Computations

Peter Schröder **Pat Hanrahan**
Department of Computer Science
Princeton University

Abstract

This paper describes a new algorithm to compute radiance in a synthetic environment. Motivated by the success of wavelet methods for radiosity computations we have applied multi wavelet bases to the computation of radiance in the presence of glossy reflectors. We have implemented this algorithm and report on some experiments performed with it. In particular we show that the convergence properties of basis functions with 1–4 vanishing moments are in accordance with theoretical predictions. As in the case of wavelet radiosity we find higher order bases to have advantages. However, the cost scaling due to the higher dimensionality of the problem is such that the higher order bases only become competitive for very high precision requirements. In practice we rarely go beyond piecewise linear functions.

1 Introduction

One of the main areas of computer graphics research concerns the analysis and synthesis of the propagation of light in a given scene. The goal of this research is the development of efficient algorithms to compute "photo-realistic" images of non-existent scenes. Much attention has been devoted to the diffuse case and a number of radiosity algorithms have been proposed. Two developments in that area, higher order Galerkin schemes [10, 11, 26, 25], and hierarchical approaches [9] have recently been unified using wavelets [8, 19].

Only few results have been reported on attempts to solve the more general rendering equation [13]. There have been some attempts at extending radiosity to handle specular or glossy reflections [12, 20, 21, 14] or to use finite element methods directly to solve the rendering equation [22]. All of these have suffered from high computational complexity and consequently extremely long simulation times.

Significant progress in the reduction of the overall computational order of the radiance problem occurred recently when Aupperle and Hanrahan [2] extended their earlier hierarchical radiosity work to radiance. The success of this approach and the success of the earlier extension of hierarchical radiosity to wavelet radiosity suggests that the tools of wavelet analysis and the sparse realization of operators in wavelet bases should allow for similar improvements when brought to bear on the radiance equation.

In this paper we describe our implementation of a higher order hierarchical Galerkin algorithm using multi wavelet bases [1] to solve for radiance in the presence of glossy reflectors. We examine a number of test configurations for which we have independent means of computing a correct answer (since they are single bounce configurations) and quantify the convergence behavior of piecewise constant, linear, quadratic, and cubic bases.

2 The radiance equation

In this section we describe the particular parameterization of the radiance equation that we are using and the linear system which results from applying a Galerkin method to it. Throughout $\vec{x}$, $\vec{y}$, and $\vec{z}$ will describe points on surfaces, where radiance originates at $\vec{x}$ and is reflected at $\vec{y}$ in the direction of $\vec{z}$ (see Figure 1). Typically there are parameters intrinsic to a surface which parameterize these quantities, but we will abstract from that fact in the following discussion to simplify the notation.

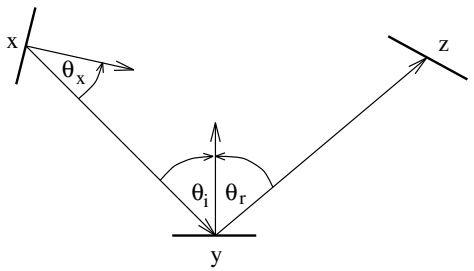

Figure 1: *Geometry of the canonical three point transport with radiance originating at some surface X and being reflected at Y in the direction of Z.*

There are two common ways of parameterizing the radiance equation. One, which we will call the *directional* parameterization makes (outgoing) radiance a function of a point on a given surface and a direction attached at this point

$$L_o(\vec{y},\vec{\omega}) = L_o^e(\vec{y},\vec{\omega}) + \int_{\mathrm{H}^2} f_r(\vec{\omega}',\vec{y},\vec{\omega}) L_o(\vec{x}(\vec{y},\vec{\omega}'),-\vec{\omega}')\,\vec{n}_y \cdot d\vec{\omega}'$$

where we have made the dependence of $\vec{x}$, the point visible from $\vec{y}$ in the direction $\vec{\omega}'$, explicit. $\vec{n}_y$ gives the normal of the surface at $\vec{y}$.

The *spatial* parameterization describes L_o as a function of two points on a given pair of surfaces

$$L_o(\vec{y},\vec{z}) = L_o^e(\vec{y},\vec{z}) + \int_{\mathrm{M}^2} f_r(\vec{x},\vec{y},\vec{z}) L_o(\vec{x},\vec{y}) G(\vec{x},\vec{y})\, dA_x$$

In both cases the symbol L_o^e describes emitted radiance and f_r denotes the BRDF. G is the geometry factor accounting for visibility and the differential element to differential element form factor. We use the symbol H^2 as shorthand for the hemisphere above a surface point and M^2 as the symbol for the union of all surfaces in the scene.

Independent of the parameterization used we can write the radiance transport operator as

$$(1-\mathcal{T})L_o = (1-\mathcal{S}\circ\mathcal{G})L_o = L_o^e$$

The operator symbol $\mathcal{G}$ describes the transformation of outgoing radiance L_o at $\vec{x}$ into incoming radiance L_i at $\vec{y}$, $\mathcal{G}L_o = L_i$. The scattering of incident radiance is denoted with the symbol $\mathcal{S}$ and corresponds to the surface local multiplication of incoming radiance L_i with the BRDF f_r followed by integration over the hemisphere. Consequently we have $\mathcal{S}L_i = L_o$. Note that both $\mathcal{G} = \mathcal{G}^+$ and $\mathcal{S} = \mathcal{S}^+$ are self adjoint with respect to the canonical inner product measure on line space ($d\mu = d\vec{\omega}\cdot d\vec{A}$). For the former this follows from symmetry while for the latter it is a consequence of Helmholtz reciprocity. It follows that $\mathcal{T}^+ = \mathcal{G}\circ\mathcal{S}$.

From now on we will use the symbol L to stand for outgoing radiance since our algorithm will compute outgoing radiances.

A Galerkin approach to the inversion of $1 - \mathcal{T}$ projects the operator into a subspace spanned by some finite basis [6]. Assuming an orthonormal (w.r.t. the *parameter domain*) basis a linear system of the following form results

$$\forall i, j: \; l_{ij} = l^e_{ij} + \sum_{mn} T_{ijmn} l_{mn}$$

for some unknown expansion coefficients l_{ij} of L.

Choosing a spatial parameterization, and hence a basis, of the form $\{N_i(\vec{x})\} \times \{N_j(\vec{y})\}$ we have $L(\vec{x}, \vec{y}) = \sum_{mn} l_{mn} N_m(\vec{x}) N_n(\vec{y})$. The transport coefficients T_{ijmn} are then defined as

$$\begin{aligned} & T_{ijmn} \\ &= \langle \mathcal{T}(N_i N_j), N_m N_n \rangle \\ &= \int_{\vec{z}} \int_{\vec{y}} \int_{\mathrm{M}^2} k^s(\vec{x}, \vec{y}, \vec{z}) N_m(\vec{x}) N_n(\vec{y}) N_i(\vec{y}) N_j(\vec{z}) \, dx \, dy \, dz \end{aligned}$$

where $k^s(\vec{x}, \vec{y}, \vec{z}) = f_r(\vec{x}, \vec{y}, \vec{z}) G(\vec{x}, \vec{y}) \frac{dA_x}{dx}$ is the *spatial kernel*. Note that the coefficients T_{ijmn} appear to be eight dimensional since $\vec{y}$ occurs twice. However, the integral is still only over six dimensions as one would expect. Using this observation we may split off one of the basis functions in $\vec{y}$, say N_n and interpret the T_{ijmn} as expansion coefficients of a modified spatial kernel function k^s_n

$$\begin{aligned} k^s_n(\vec{x}, \vec{y}, \vec{z}) &= k^s(\vec{x}, \vec{y}, \vec{z}) N_n(\vec{y}) \\ &\approx \sum_{ijm} T_{ijmn} N_m(\vec{x}) N_i(\vec{y}) N_j(\vec{z}) \end{aligned} \tag{1}$$

In the case of a directional parameterization the basis set would typically be of the form $\{N_i(\vec{x})\} \times \{N_j(\vec{\omega})\}$ with $L(\vec{x}, \vec{\omega}) = \sum_{mn} l_{mn} N_m(\vec{x}) N_n(\vec{\omega})$. The spatial parameterization only had three free parameters $(\vec{x}, \vec{y}, \vec{z})$ in a natural way. The directional parameterization starts out with four parameters $(\vec{x}, \vec{\omega}', \vec{y}, \vec{\omega})$. Since only points $\vec{x}$ and $\vec{y}$ which "look at each other" will couple across the integral however this parameterization too has only three free parameters as expected. For example, one can chose to express $\vec{x}$ as a function of $(\vec{y}, \vec{\omega}')$ or $\vec{\omega}'$ as a function of $(\vec{x}, \vec{y})$, resulting in two different expressions for T_{ijmn}

$$\begin{aligned} & T_{ijmn} \\ &= \langle \mathcal{T}(N_i N_j), N_m N_n \rangle \\ &= \int_{\vec{\omega}} \int_{\vec{y}} \int_{\mathrm{M}^2} k^d(\vec{\omega}', \vec{y}, \vec{\omega}) N_m(\vec{x}) N_n(\vec{\omega}') N_i(\vec{y}) N_j(\vec{\omega}) \, dx \, dy \, d\omega \\ &= \int_{\vec{\omega}} \int_{\vec{y}} \int_{\mathrm{H}^2} k^d(\vec{\omega}', \vec{y}, \vec{\omega}) N_m(\vec{x}) N_n(\vec{\omega}') N_i(\vec{y}) N_j(\vec{\omega}) \, d\omega' \, dy \, d\omega \end{aligned}$$

with the *directional kernel* $k^d(\vec{\omega}', \vec{y}, \vec{\omega}) = f_r(\vec{\omega}', \vec{y}, \vec{\omega}) \vec{n}_y \cdot \vec{\omega}'$.

Once again the T_{ijmn} admit to an interpretation as coefficients in the expansion of a modified kernel function k^d_n

$$\begin{aligned} k^d_n(\vec{x}, \vec{y}, \vec{\omega}) &= k^d(\vec{\omega}'(\vec{x}, \vec{y}), \vec{y}, \vec{\omega}) N_n(\vec{\omega}'(\vec{x}, \vec{y})) \\ &\approx \sum_{ijm} T_{ijmn} N_m(\vec{x}) N_i(\vec{y}) N_j(\vec{\omega}) \end{aligned} \tag{2}$$

and k^d_m

$$\begin{aligned} k^d_m(\vec{\omega}', \vec{y}, \vec{\omega}) &= k^d(\vec{\omega}', \vec{y}, \vec{\omega}) N_m(\vec{x}(\vec{y}, \vec{\omega}')) \\ &\approx \sum_{ijn} T_{ijmn} N_n(\vec{\omega}') N_i(\vec{y}) N_j(\vec{\omega}) \end{aligned} \tag{3}$$

2.1 Discussion of parameterizations

When trying to decide which parameterization of radiance to use a number of issues arise which we will turn to now. Both spatial and directional parameterizations share the feature that the Galerkin method effectively expands kernel functions which are

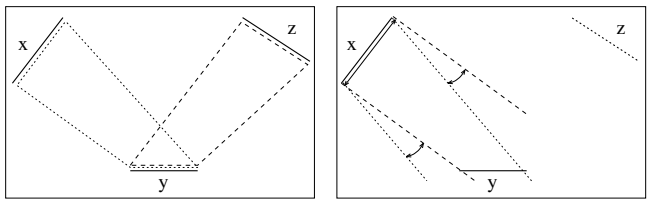

Figure 2: *Illustration of the matchup of support between basis functions. On the left basis functions with spatial parameterization. Their supports (dotted respectively dashed outlines) match at Y. On the right basis functions with directional parameterizations. A basis function with spatial support on X and some directional support (spread of angles) does not match up cleanly with the spatial support of Y.*

modified. In the case of the spatial parameterization we saw above (Equation 1) that instead of expanding k^s we end up expanding k^s_n. In the directional case we gave two choices (Equations 2, 3) k^d_n and k^d_m. In any case the original kernel is multiplied with one of the basis functions. As a consequence the properties of the *modified* kernel function determine the behavior of the algorithm. For example, suppose we use piecewise polynomial functions as bases. In that case the modified kernel will be of higher polynomial order.

A major practical difference between the two parameterizations arises immediately when computing the quadratures inherent in the definition of T_{ijmn}. Suppose our bases have finite support, as is most often the case. The 6D integral of the spatial parameterization can be evaluated in a straightforward way with a product quadrature rule since the domain of the integrand is a 6D hypercube. We say that the supports of the basis functions *match* by definition (see the 2D illustration on the left in Figure 2). This is not the case for the directional parameterization.

Suppose we consider some basis function on surface X with support $(\vec{x}, \vec{\omega}')$ (see the 2D illustration on the right in Figure 2). This rectangle of support in parameter space corresponds to a (convolution) cone of directions in world space. The only time there is any exchange between X and a basis function on Y is when the spatial support of Y intersects this cone. This intersection will in general be awkwardly shaped with respect to the spatial support $\vec{y}$ of a chosen basis function on Y. Similarly, if we choose to fix $\vec{x}$ and $\vec{y}$ to correspond to the spatial support of some basis functions we will only have exchange for that portion of directions $\vec{\omega}'(\vec{x}, \vec{y})$ which overlap the directional support $\vec{\omega}'$ of the basis function on X. Again a region which is generally awkwardly shaped. Consequently product rules are hard to apply and suffer from precision problems [16].

On the other hand in the absence of real clustering algorithms using a spatial parameterization makes the number of basis functions for a given patches' outgoing radiance an a priori function of the input complexity. However, it is not clear how much this difference bears out in practice since the directional approaches still need to identify and process all primitives inside the directional support of a given patches' basis function. Further research and comparison of methods is needed to fully assess the impact of this difference.

2.2 Non-standard realization of the operator

When using wavelets to realize integral operators we need to distinguish between the types of bases used as well as the type of realization of the integral operator itself. These are separate issues and either one admits to so called *standard* and *non-standard* treatments. In particular we are using a non-standard realization of the operator as was done in the original wavelet radiosity

work [19, 8].

Using a non-standard realization of the operator has a number of consequences

- transport due to different sources needs to be consolidated with a push/pull operation
- refinement proceeds by replacing earlier (coarser) couplings
- a simple recursive procedure provably accounts for all important transports (for tree wavelets)
- the number of necessary coefficients is $O(n)$ versus $O(n \log n)$, where n is the number of elements [4].

These issues are further compounded by the independent issues associated with primal and dual bases. For example, if the dual hierarchy does not have vanishing moments then this by itself causes the need for a wavelet up/down transformation independent of the standard or non-standard realization of the operator itself. Since we are using orthonormal bases we will not discuss primal versus dual bases here (for a discussion of these issues in the context of radiosity see [19]).

One of the major issues in any wavelet based algorithm is the enumeration of all important couplings. For tree wavelets (neighboring wavelets have no overlapping support) this is straightforward as wavelets and their supports are uniquely identified with sections of geometry (elements). A simple recursive procedure [9, 8] considers the coupling between surfaces and naturally accounts for all energy transfers while maintaining some error threshold

```
ProjectKernel( Basis ij, Basis mn )
  error = Oracle( ij, mn );
  if( Accept( error ) || RecurLimit( ij, mn ) )
    T_ijmn = Quadrature( ij, mn );
  else
    if( PreferredSubdivision( ij, mn ) == ij )
      ForAllChildren( c, ij )
        ProjectKernel( c, mn );
    else
      ForAllChildren( c, mn )
        ProjectKernel( ij, c );
```

Tree wavelets also make it possible to work with the smooth functions exclusively instead of having to use various combinations of smooth and detail functions, simplifying the implementation.

It is not immediately clear how to use wavelets with overlapping support in this recursive enumeration scheme. The basic difficulty arises from the need to insure that all power is accounted for once and only once. For overlapping wavelets the standard realization of the operator is simpler. In the standard realization there is no overrepresentation of bases at the different levels of resolution. Consequently refinement proceeds by adding a detail function wherever the error criterion is not met. The standard realization does require couplings between all levels of the hierarchy however [4], resulting in the $O(n \log n)$ complexity. Whether the extra factor of $\log n$ is very noticeable for the regimes in which radiosity or radiance computations are performed is an open question of research.

After the initial setup of transport links there is typically refinement in successive iterations of the solver. In a tree wavelet scheme, such as the one we use, links are *replaced* upon refinement by child (finer) links. Using a standard realization of the operator, refinement would proceed by *adding* child links. Replacing coarser links with finer links has clear implications for the error analysis. The earlier links were found to contain too much error and replaced by links with lesser error. In the standard realization finer links are added to decrease the representation error. The difficulty arises from the fact that there is numerical error in the quadrature proportional to some power of the subdivision level. This error remains in the earlier link but is reduced for the newly added, finer links. Further research is needed to understand how these differences in error behavior express themselves in the fidelity of the final solution.

3 Implementation

In this section we discuss some of the implementation details which arose in our approach and how we addressed them.

As mentioned above we use a spatial parameterization with multi wavelet (piecewise polynomial) bases. This approach is a direct extension of the earlier wavelet radiosity work reported in [8, 19]. The main difference is the added number of dimensions. Coefficients of basis functions are not tied to individual surfaces anymore but rather to pairs of surfaces. There are still links which govern the exchange of power. Instead of links only between surface elements there are now also links between pairs of surface elements (with the middle surface matching up). The resulting data structures follow closely the ideas presented by Aupperle and Hanrahan in [2]. A refinement procedure refines links as before based on an oracle while gathering occurs across links (both inter surface and inter surface-pairs). Visibility is determined as described by Teller and Hanrahan [24]. The actual code is a strict superset of our earlier wavelet radiosity code. The quadrature used is straightforward Gaussian quadrature [18]. Since we use a spatial parameterization simple product rules suffice. The Push and Pull code is identical save for the fact that it extends over more dimensions.

In the following sections we address some individual details associated with our particular implementation of the radiance solver.

3.1 Shading model

As a BRDF we use a sum of a pure diffuse component and a microfacet model. We use the approximation to the Beckmann distribution together with an approximation of the Smith shadowing factor and an anisotropy control term all proposed by Schlick [17]

$$\begin{aligned} f_r(t, v, v', w) &= \frac{g(v)g(v')}{4\pi v v'} z(t) a(w) \\ g(v) &= \frac{v}{r + v(1-r)} \\ z(t) &= \frac{r}{(1 - t^2(1-r))^2} \\ a(w) &= \sqrt{\frac{p}{p^2 + w^2(1-p^2)}} \end{aligned}$$

where $t = H \cdot N$ is the cosine of the angle between the local surface normal and the half angle vector between incoming and outgoing directions; v and v' are the cosines of the incoming and outgoing directions respectively with the normal; w gives the cosine of the half angle vector with the preferred direction of the anisotropy model. The parameters $r, p \in [0, 1]$ control roughness and anisotropy respectively. The Fresnel factor was set to one. The images in Figure 10 (see color section) show the use of the anisotropy factor. The approximation as given is not normalized and its hemispherical reflectance is not independent of incoming angle. However, its hemispherical reflectance is bounded and a simple normalization, which is a function of the roughness parameter, suffices to avoid energy gain.

3.2 Importance

Recently a number of papers in the graphics community [23, 3, 5, 15] have shown that the computation of illumination for a given scene can be greatly accelerated if we give up our goal of producing a view independent solution and instead compute a view dependent solution. Because of the great demand placed on

computational resources by radiance computations we included this acceleration technique in our algorithm.

Since importance is a quantity which is propagated under the same operator as radiance it is an easy matter to include importance as a fourth "color" channel in the simulation. This follows directly from the fact that both $\mathcal{G}$ and $\mathcal{S}$ are self adjoint.

Let R describe the response of our sensor (e.g. eye, CCD) to *incoming* radiance at some point $\vec{x}$ from a direction $\vec{\omega}$. Define $\tilde{L}_o$ to be the solution to the formal operator equation $(1-\mathcal{T})\tilde{L}_o = R$. The response of our sensor to the incoming radiance L_i is described as

$$\begin{aligned}\langle R, L_i\rangle &= \langle (1-\mathcal{T})\tilde{L}_o, L_i\rangle \\ &= \langle \tilde{L}_o, (1-\mathcal{T}^+)L_i\rangle \\ &= \langle \tilde{L}_o, (1-\mathcal{G}\circ\mathcal{S})\mathcal{G}L_o\rangle \\ &= \langle \tilde{L}_o, \mathcal{G}(1-\mathcal{S}\circ\mathcal{G})L_o\rangle \\ &= \langle \mathcal{G}\tilde{L}_o, (1-\mathcal{T})L_o\rangle \\ &= \langle \tilde{L}_i, L_o^e\rangle\end{aligned}$$

The last equation shows the well known fact that the response of our sensor to incoming radiance can be computed from the incoming radiance at the light sources due to the formal transport problem which arises when treating the sensor as a radiance source with "emittance" profile R. This also explains why we can use the same algorithm to solve both for importance and radiance: importance is *defined* as the solution to the same operator equation as radiance, only the boundary condition is different.

Since we are using importance refinement each scene consists of a sequence of geometric primitives (quadrilaterals in our current implementation) one of which is our "CCD" camera and the only source of R emittance. The final image results by evaluating the basis functions anchored at the CCD camera.

3.3 Oracle and refinement

Our recursive refinement procedure is driven by an error estimate for a given transport as well as the magnitude of radiance at the source end of a transport coupling and the importance at the receiver end of the transport. The error estimate for the coupling is determined by a a polynomial oracle similar to the one described in [8]. In the case of radiance the polynomial estimator extends over 6 dimensions rather than 4, but the basic logic remains the same. The samples of the kernel function generated as part of the quadrature routine are used to construct an interpolating polynomial. This polynomial is measured against the actual kernel function using a number of new sample points. The resulting error estimate is augmented by a geometric analysis. The geometric analysis was inspired by the observations that Aupperle and Hanrahan [2] made. Very peaked BRDF functions are easily missed if only a small number of samples of the integrand are taken. To decrease the chance of aliasing problems due to missing this peak we have augmented the oracle with a geometric analysis.

For a given triplet of surfaces exchanging power we consider the range of outgoing angles on X, incoming angles on Y, and outgoing angles on Y using straightforward geometric analysis (see [2]). The resulting ranges of angles are used in an interval evaluation of the shading model itself giving max and min values of f_r over the surfaces in question. This spread in the value of f_r is used as a weight of the error estimate derived by the polynomial oracle. Let ϵ be this estimate. We scale it by

$$1 + r(1-r)(f_{r_{\max}} - f_{r_{\min}})$$

The factor or r normalizes the BRDF since its peak value is proportional to r^{-1}. The factor $(1-r)$ insures that the penalty scaling increases for more peaked BRDF functions where the danger of aliasing is much larger.

This augmented oracle has been very successful in finding the peak of a BRDF and insuring the sufficient subdivision occurs around the peak. Figure 8 (see color section) shows the oracle in progress. Notice the detail in the meshing as the error threshold is lowered. In particular for very peaked reflectances ($r = 0.001$) we can see in the images "islands" of fine meshing in the region of the reflection peak. Another example for anisotropic reflectors is shown in the images at the top of Figure 10 (see color section). Notice in particular how the mesh follows the high anisotropy gradients of the reflection (most noticeable in the green part).

3.4 Separation of directional and isotropic radiance

In environments which contain both purely diffuse and glossy reflectors it is desirable to distinguish between diffuse and glossy transport. This approach avoids the cost of the extra directional dimensions in transports which are independent of direction, for example reflection off a diffuse surface.

We take advantage of this by classifying a given link into one of the following categories. A link is called

- *diffuse* if both source and destination are diffuse. In this case only the usual radiosity couplings T_{ijmn}, $j = 0$ and $n = 0$, need to be computed and associated with the given link
- *glossy* if the receiver element exhibits glossy reflection. In this case the source may itself be diffuse or glossy. The diffuse case is simply characterized by coupling coefficients T_{ijmn} for which $n = 0$. This corresponds to writing $L(\vec{x},\vec{y}) = \frac{B(\vec{x})}{\pi}$ and only using the constant basis function in $\vec{y}$. Otherwise these transports are characterized by the full range of subindices on T_{ijmn} and represent the canonical case with all coefficients computed
- *mixed* if a glossy source interacts with a diffuse reflector. In this case the set T_{ijmn} reduces to one for which $j = 0$ and we only compute the remaining T_{ijmn}.

During the solution process radiosity coefficients are kept at elements while radiance coefficients are kept in a separate tree data structure corresponding to the hierarchy of radiance basis functions. Gathering is performed across diffuse links just as in radiosity, i.e. the radiosity of the source is moved across the link, multiplied with the transport coefficients and added to the coefficients of the receiver. Gathering of radiance occurs across glossy and mixed links and is only distinguished by the dimensionality of the quantities being moved and multiplied with coupling coefficients.

4 Results

The above algorithm has been implemented as part of our rendering testbed [8, 19, 7, 24]. In order to verify the algorithm and its implementation we have considered a number of configurations. These configurations were designed to both stress the numerical algorithms and to be verifiable in an independent manner. In this section we describe three of these configurations and give results from simulations which show behavior consistent with a theoretical analysis.

The first test configuration is shown in Figure 3. A diffuse emitter is placed above a glossy reflector. We examine the irradiance on a diffuse receiver wall. Due to the relative orientations any transport to the wall has to occur across a single bounce off the reflector, creating an intersection of the directional lobe of the BRDF with the wall. The irradiance on the wall is equal to

$$E(\vec{z}) = \int_{A_r}\int_{A_s} f_r(\vec{x},\vec{y},\vec{z})G(\vec{x},\vec{y})\frac{B(\vec{x})}{\pi}G(\vec{y},\vec{z})\,dA_x\,dA_y$$

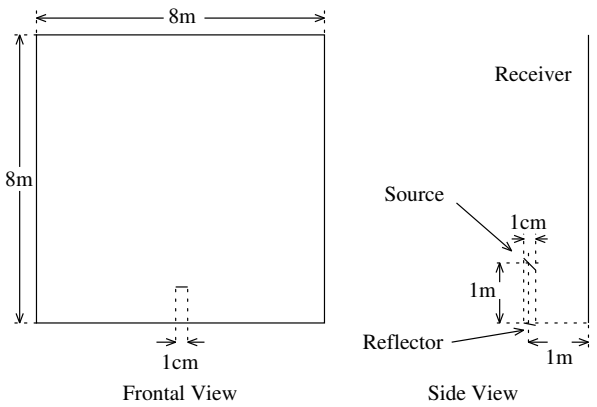

Figure 3*: Geometry of small source/reflector test configuration. The source is inclined* 45° *to avoid any direct illumination of the receiver wall. The reflector is inclined* 15° *to cause a significant intersection between the directional lobe of the BRDF and the receiver wall.*

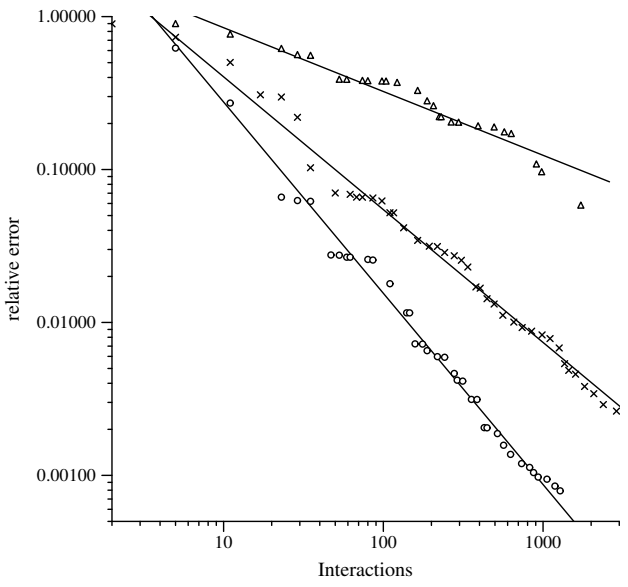

Figure 4*: Relative* L^1 *error for the small source/reflector configuration as a function of number of interactions allocated by the oracle and used for transport. From top to bottom we plotted multi wavelets of 1, 2, and 3 vanishing moments.*

where $\vec{x}$ is located on the source A_s, $\vec{y}$ on the reflector A_r, and $\vec{z}$ on the receiver wall. Because of the relative ratio of size of source/reflector and the distance to the receiver wall a one point quadrature rule will incur an error on the order of 10^{-4}. In other words, the irradiance on the receiver follows the pointwise evaluation of f_r itself to within 10^{-4}. This is used as the reference solution to which to compare our computed solution. Note further that $f_r(\vec{x}, \vec{y}, \vec{z})G(\vec{y}, \vec{z})$ has a very high dynamic range. f_r itself ranges from unity (directions which have only a diffuse contribution) to $\frac{1}{r}$. G ranges over 2 orders of magnitude due to the denominator ranging from $(1m)^2$ to $(9m)^2$. The particular value of r used for the graph in Figure 4 was 0.01.

Figure 4 shows the convergence behavior of multi wavelet bases with $M = 1, 2, 3$ vanishing moments. The error is relative L^1 error defined as

$$\frac{\int_{A_w} \|E(z) - \hat{E}(z)\| \, dA_z}{\int_{A_w} \|E(z)\| \, dA_z}$$

where $\hat{E}(z)$ denotes the computed solution. This integral itself is evaluated with a repeated trapezoid rule 3 subdivision levels finer than the irradiance computation. In this way any error incurred in computing the error integral itself is prevented from contaminating our convergence measurements. The independent variable in the resulting graph gives the number of couplings allocated by our oracle and the associated radiance/importance refinement procedure and consequently used during the computation of transport.

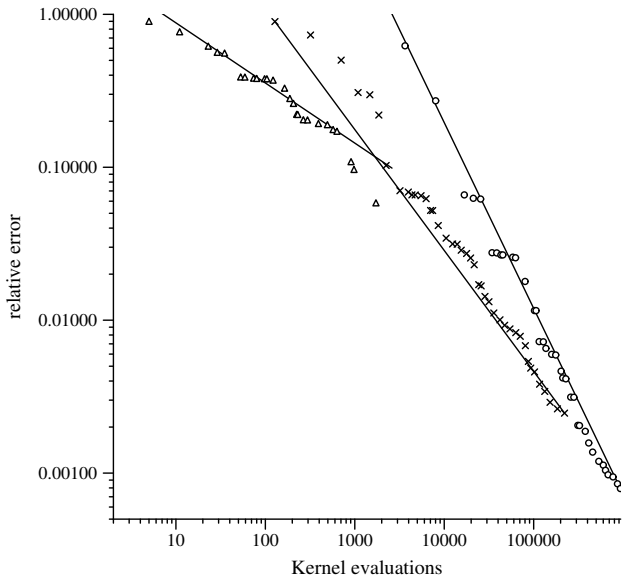

Figure 5*: Relative* L^1 *error for the test configuration as a function of number of kernel evaluations. From top to bottom (left to right) we plotted multi wavelets of 1, 2, and 3 vanishing moments.*

The plot shows clearly the convergence behavior of the three different types of bases. The data points have been fitted with lines. The resulting slopes have ratios very close (within 3%) to $1 : 2 : 3$ which are the theoretically predicted ratios for the given order of basis functions. However, when plotting the data in this way we are ignoring the increasing work necessary for higher order quadratures which are required by higher order basis functions. Figure 5 shows the same data, but this time plotted as a function of the number of kernel evaluations which more closely measures the actual work performed by the algorithm. Due to the sixth order scaling of the samples for a 6 dimensional quadrature rule the higher order basis function data points are moved over to the right. In fact the quadratic basis function data points move over so far that quadratic bases only become competitive at very high accuracy requirements.

The images in Figure 8 (see color section) show the oracle in progress. The top row shows the perfectly diffuse case ($r = 1$), the middle row a more directional case of $r = 0.01$ and the bottom row a very peaked BRDF for $r = 0.001$. In each row the first image has an error on the order of 10%, the second on the order of 1% and both show the associated meshing. The final image shows final accuracy with linear basis functions at a finest allowed subdivision level of $25cm$. We can clearly observe the increased meshing where the kernel function has the highest variance. Note in particular the separating "islands" of fine meshing as r is decreased. One is in an area where the variance is dominated by diffuse reflection the other in the area where the lobe of the BRDF intersects the wall. The pattern of light on the wall

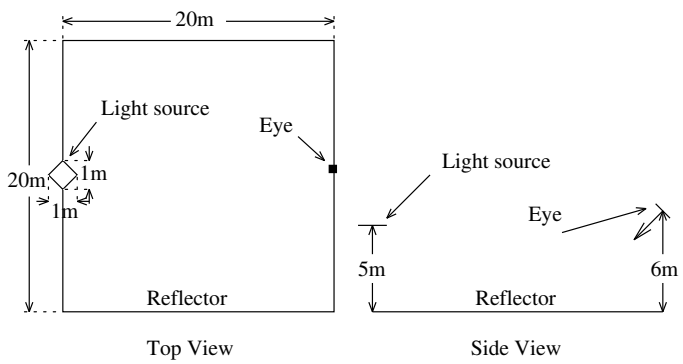

Figure 6*: Geometry of large area source/reflector configuration. The source is parallel to the reflector at the far end, while the eye is inclined by* 45° *towards the reflector.*

shows clearly the pinching off of the peak in the directional lobe of the reflectance function as r is decreased and demonstrates the action of non-diffuse transport.

One aspect which is not stressed by the above configuration is any significant area integration over the light source or the reflector surface. The difficulty with such a configuration is the need to find an independent means of verifying the computed answer. We now turn to such a configuration.

Figure 6 shows a diagram of a diamond shaped light source over a base polygon. The light source is located at the far end, while the eye is at the near end looking down at an angle of 45°. A picture of this configuration can be seen in Figure 9 (see color section). The roughness parameter was set to $r = 0.01$ for a fairly peaked response. There is also a diffuse component of $d = 0.1$ which can be seen at the far end right below the light source. The error in the computed solution was estimated by an independent integration module which evaluated the radiance at a given point on the reflector with respect to the eye by using a 7^{th} order Gauss quadrature over the light source. Evaluating this function pointwise at a grid 3 levels below the subdivision grid on the reflector produced a numerical answer with significantly higher precision than our computed solution. The convergence results are plotted in Figure 7 for multi wavelets of $m = 1, 2, 3, 4$ vanishing moments respectively. Once again we can see the characteristic convergence rates (fitted lines).

Finally we added an anisotropy factor to the shading model [17] and used 3 colored light sources to show the resulting reflections. Figure 10 (see color section) shows a rendered image of this configuration on the top left and with the corresponding meshing on the reflector on the top right. Note the fine level of subdivision in the areas where the radiance changes most rapidly. This simulation used linear basis functions, a roughness parameter of $r = 0.1$, and an anisotropy factor of $p = 0.1$. The preferred direction of the anisotropy model ("scratches") was given by the family of hyperbolas $x * y = \pm c$ (note the singularity in the middle). The sequence of images in Figure 10 (see color section) shows a spatial plot of the resulting radiance field by rotating the eye 90° around the model.

5 Conclusion

In this paper we have described a new algorithm for the computation of radiance in the presence of glossy reflectors using multi wavelet bases. The resulting algorithm has been shown to converge to the correct results in a number of test configurations. The rate of convergence is consistent with the theoretically predicted rate.

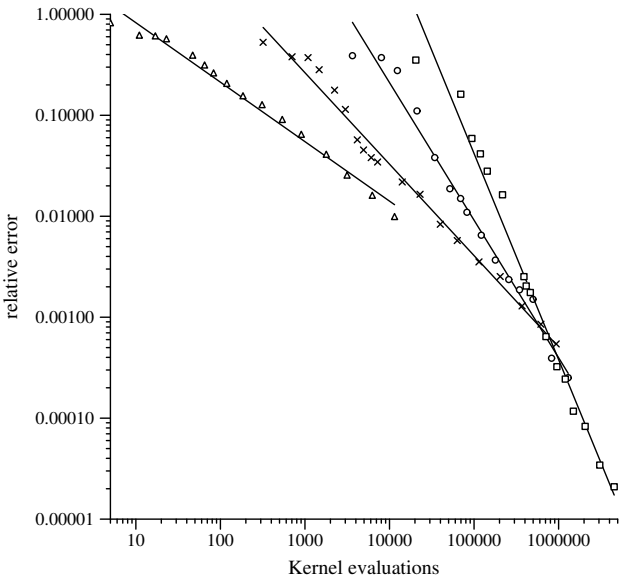

Figure 7*: Relative* L^1 *error for the configuration in Figure 6 as a function of kernel evaluations. From left to right we plotted multi wavelets of* $m = 1, 2, 3, 4$ *vanishing moments.*

As was done in the earlier work concerning wavelet radiosity [8, 19] we have in effect extended an earlier hierarchical technique to higher orders: in the case of radiosity the work reported in [9] and in the case of radiance the work reported in [2] . The benefits are similar. The quality of the computed images increases markedly when going from piecewise constant bases to piecewise linear (or higher). In the case of radiosity the work per entry in the matrix system scales as a fourth power of the number of kernel evaluations, while in the case of radiance it scales as a sixth power. In the case of radiosity this led to a relationship between user experienced work and error which clearly favored higher order basis functions. In the case of radiance this tradeoff is not so clear. Higher order basis functions only become competitive at very high precision requirements (see Figures 5, 7). In practice we have found that the cost increase for linear basis functions is not too high while yielding much better looking pictures. So far we have employed quadratic bases only for the plotting of the graphs. Cubic bases are prohibitively expensive. Unless some major optimizations in the code can be found we do not expect cubic bases to find much application because of their high cost.

The radiance problem remains hard and costly to compute. Considerable research remains to be done to accelerate algorithms which compute the light transport in the presence of glossy reflectors. We do believe that hierarchical techniques will have a place in the repertoire of such algorithms. Interesting areas for further investigations include

- comparisons between the directional and spatial parameterizations
- comparisons between standard and non-standard realizations
- overlapping wavelets

Acknowledgments

We would like to thank Seth Teller for major code contributions to the GUR testbed; Larry Aupperle, Wim Sweldens, Steven Gortler, and Michael Cohen for many fruitful discussions regarding implementation ideas, wavelets, and earlier radiance algorithms; David Laur for production help. Support was provided by Apple, Silicon Graphics, the Scientific Visualization Department at GMD, and the NSF (contract no. CCR 9207966).

References

[1] ALPERT, B., BEYLKIN, G., COIFMAN, R., AND ROKHLIN, V. Wavelet-like Bases for the Fast Solution of Second-kind Integral Equations. *SIAM Journal on Scientific Computing 14*, 1 (January 1993).

[2] AUPPERLE, L., AND HANRAHAN, P. A Hierarchical Illumination Algorithm for Surfaces with Glossy Reflection. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 155–162.

[3] AUPPERLE, L., AND HANRAHAN, P. Importance and Discrete Three Point Transport. In *Fourth Eurographics Workshop on Rendering* (June 1993), Eurographics, pp. 85–94.

[4] BEYLKIN, G., COIFMAN, R., AND ROKHLIN, V. Fast Wavelet Transforms and Numerical Algorithms I. *Communications on Pure and Applied Mathematics 44* (1991), 141–183.

[5] CHRISTENSEN, P. H., SALESIN, D. H., AND DEROSE, T. A Continuous Adjoint Formulation for Radiance Transport. In *Fourth Eurographics Workshop on Rendering* (June 1993), Eurographics, pp. 95–104.

[6] DELVES, L. M., AND MOHAMED, J. L. *Computational Methods for Integral Equations*. Cambridge University Press, 1985.

[7] GERSHBEIN, R. S., SCHRÖDER, P., AND HANRAHAN, P. Textures and Radiosity: Controlling Emission and Reflection with Texture Maps. In *Computer Graphics Annual Conference Series, 1994* (1994).

[8] GORTLER, S., SCHRÖDER, P., COHEN, M., AND HANRAHAN, P. Wavelet Radiosity. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 221–230.

[9] HANRAHAN, P., SALZMAN, D., AND AUPPERLE, L. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics 25*, 4 (July 1991), 197–206.

[10] HECKBERT, P. S. *Simulating Global Illumination Using Adaptive Meshing*. PhD thesis, University of California at Berkeley, January 1991.

[11] HECKBERT, P. S. Radiosity in Flatland. *Computer Graphics Forum 2*, 3 (1992), 181–192.

[12] IMMEL, D. S., COHEN, M. F., AND GREENBERG, D. P. A Radiosity Method for Non-Diffuse Environments. *Compuber Graphics 20*, 4 (August 1986), 133–142.

[13] KAJIYA, J. T. The Rendering Equation. *Computer Graphics 20*, 4 (1986), 143–150.

[14] LESAEC, B., AND SCHLICK, C. A Progressive Ray-tracing-based Radiosity with General Reflectance Functions. In *Photorealism in Computer Graphics (Proceedings Eurographics Workshop on Photosimulation, Realism and Physics in Computer Graphics)* (June 1990), K. Bouatouch and C. Bouville, Eds., Springer Verlag, pp. 101–114.

[15] PATTANAIK, S. *Computational Methods for Global Illumination and Visualization of Complex 3D Environments*. PhD thesis, Birla Institute of Technology and Science, Pilani, India, February 1993.

[16] PER H. CHRISTENSEN, ERIC J. STOLLNITZ, D. H. S., AND DEROSE, T. D. Importance-Driven Wavelet Radiance. Tech. Rep. 94-01-05, University of Washington, Seattle, January 1994.

[17] SCHLICK, C. A customizable reflectance model for everyday rendering. In *Fourth Eurographics Workshop on Rendering* (June 1993), Eurographics, pp. 73–83.

[18] SCHRÖDER, P. Numerical Integration for Radiosity in the Presence of Singularities. In *Fourth Eurographics Workshop on Rendering* (1993).

[19] SCHRÖDER, P., GORTLER, S. J., COHEN, M. F., AND HANRAHAN, P. Wavelet Projections For Radiosity. In *Fourth Eurographics Workshop on Rendering* (June 1993), Eurographics, pp. 105–114.

[20] SHAO, M.-Z., PENG, Q.-S., AND LIANG, Y.-D. A New Radiosity Approach by Procedural Refinements for Ralistic Image Synthesis. *Computer Graphics 22*, 4 (August 1988), 93–101.

[21] SHIRLEY, P. A Ray Tracing Method for Illumination Calculations in Diffuse Specular Scenes. In *Proceedings of Graphics Interface 90* (May 1990), Canadian Information Processing Society, pp. 205–212.

[22] SILLION, F. X., ARVO, J. R., WESTIN, S. H., AND GREENBERG, D. P. A Global Illumination Solution for General Reflectance Distributions. *Computer Graphics 25*, 4 (July 1991), 187–196.

[23] SMITS, B. E., ARVO, J. R., AND SALESIN, D. H. An Importance Driven Radiosity Algorithm. *Computer Graphics 26*, 2 (August 1992), 273–282.

[24] TELLER, S., AND HANRAHAN, P. Global Visibility Algorithms for Illumination Computations. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 239–246.

[25] TROUTMAN, R., AND MAX, N. Radiosity Algorithms Using Higher-order Finite Elements. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 209–212.

[26] ZATZ, H. R. Galerkin Radiosity: A Higher Order Solution Method for Global Illumination. Master's thesis, Cornell University, August 1992.

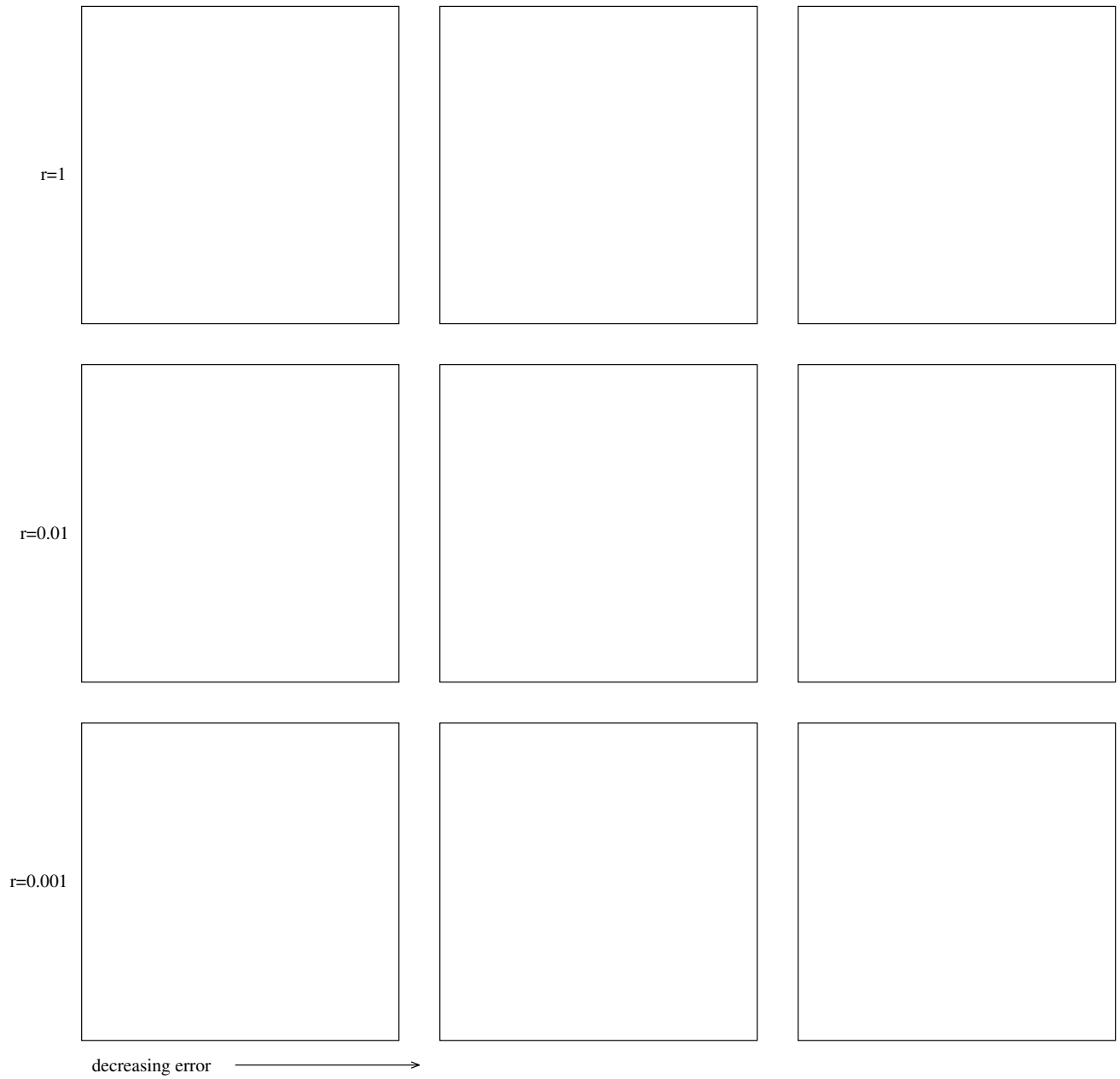

Figure 8: *Images of small source/reflector test configuration for different roughness values ($r = 1, 0.01, 0.001$) and different error criteria (10%, 1%, final accuracy).*

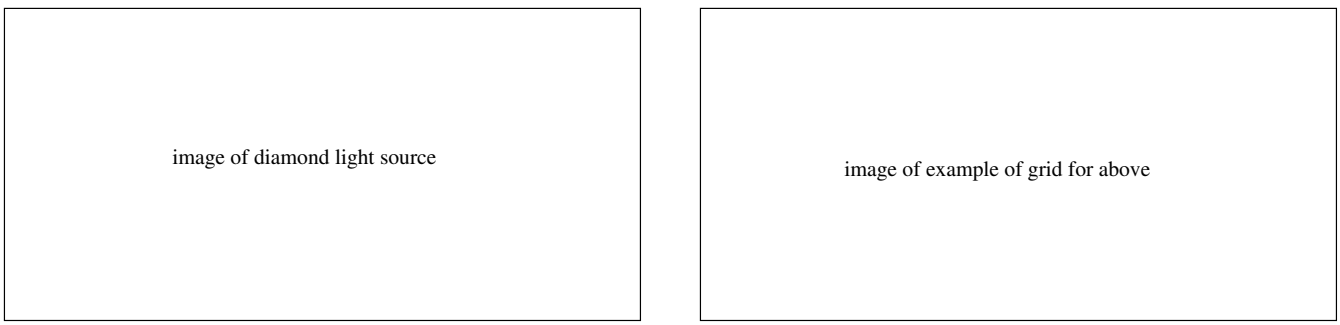

Figure 9: *Images of large area source/reflector test configuration. On the left the radiance as seen from the eye; on the right the induced meshing ($r = 0.01$; linear basis functions).*

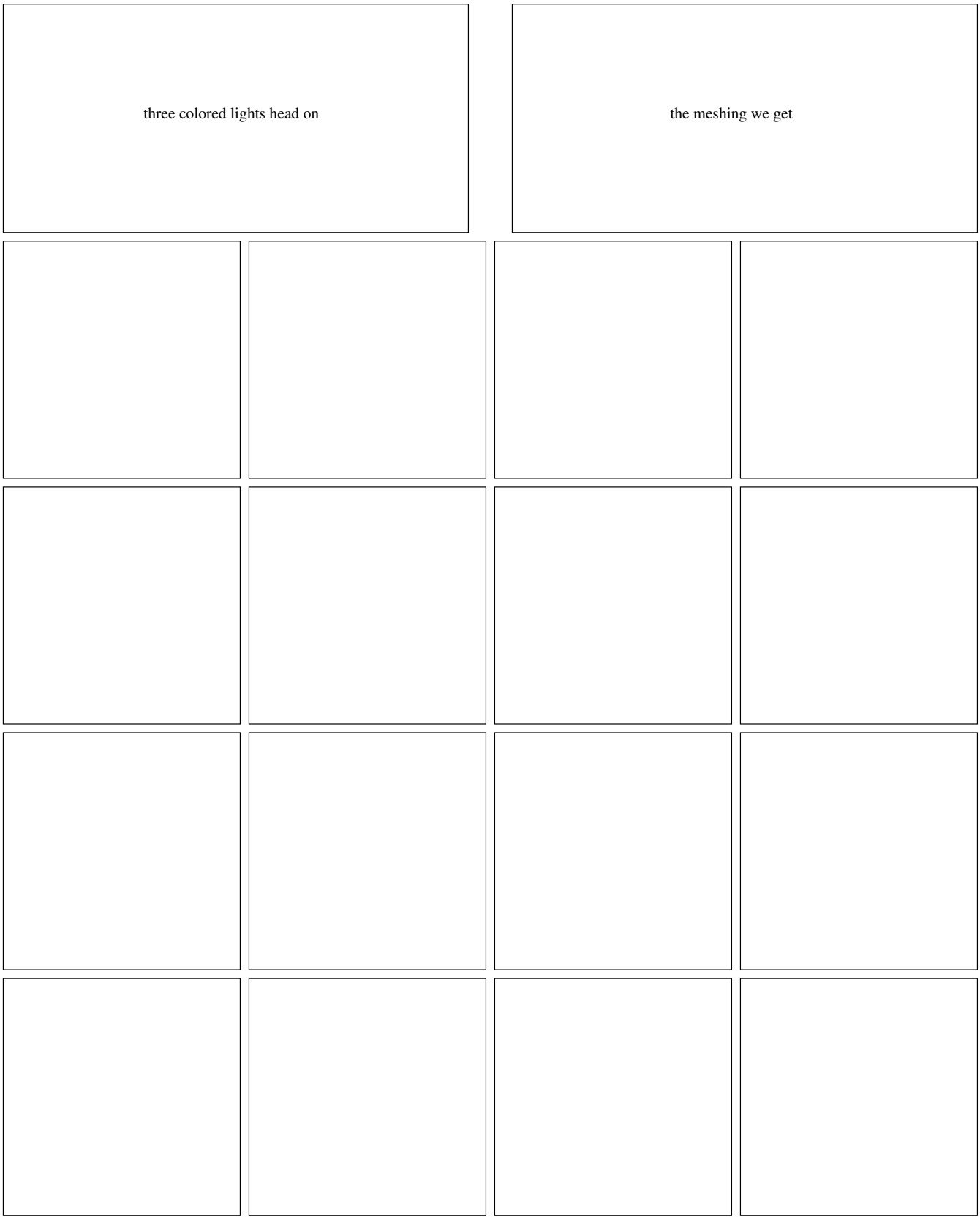

Figure 10: *The top left shows a configuration with 3 light sources and an anisotropic reflector; the top right the induced meshing. Below a sequence illustrating a spatial plot of the radiance field induced by the 3 colored light sources as the view point changes ($r = 0.1$,$p = 0.1$; linear basis functions).*