# Wavelet Algorithms for Illumination Computations

Peter Schröder

Research Report CS-TR-466-94
October 1994

# WAVELET ALGORITHMS FOR ILLUMINATION COMPUTATIONS

Peter Schröder

A DISSERTATION
PRESENTED TO THE FACULTY
OF PRINCETON UNIVERSITY
IN CANDIDACY FOR THE DEGREE
OF DOCTOR OF PHILOSOPHY

RECOMMENDED FOR ACCEPTANCE
BY THE DEPARTMENT OF
COMPUTER SCIENCE

November 1994

# Abstract

One of the core problems of computer graphics is the computation of the equilibrium distribution of light in a scene. This distribution is given as the solution to a Fredholm integral equation of the second kind involving an integral over all surfaces in the scene. In the general case such solutions can only be numerically approximated, and are generally costly to compute, due to the geometric complexity of typical computer graphics scenes. For this computation both Monte Carlo and finite element techniques (or hybrid approaches) are typically used.

A simplified version of the illumination problem is known as *radiosity*, which assumes that all surfaces are diffuse reflectors. For this case hierarchical techniques, first introduced by Hanrahan *et al.*[32], have recently gained prominence. The hierarchical approaches lead to an asymptotic improvement when only finite precision is required. The resulting algorithms have cost proportional to $O(k^2 + n)$ versus the usual $O(n^2)$ ($k$ is the number of input surfaces, $n$ the number of finite elements into which the input surfaces are meshed). Similarly a hierarchical technique has been introduced for the more general radiance problem (which allows glossy reflectors) by Aupperle *et al.*[6].

In this dissertation we show the equivalence of these hierarchical techniques to the use of a Haar wavelet basis in a general Galerkin framework. By so doing, we come to a deeper understanding of the properties of the numerical approximations used and are able to extend the hierarchical techniques to higher orders. In particular, we show the correspondence of the geometric arguments underlying hierarchical methods to the theory of Calderon-Zygmund operators and their sparse realization in wavelet bases. The resulting wavelet algorithms for radiosity and radiance are analyzed and

numerical results achieved with our implementation are reported. We find that the resulting algorithms achieve smaller and smoother errors at equivalent work.
**Advisor:** Prof. Pat Hanrahan

# Acknowledgements

I would like to thank above all my advisor Pat Hanrahan for his guidance in all things academic, and in particular his patience with me.

Thanks also to the readers of this dissertation, Michael Cohen, and Joel Friedman and the members of my committee. Much of the initial work at the basis of this thesis would not have progressed as quickly as it did had it not been for the collaboration with Steven Gortler. Many insights and refinements of the underlying ideas may never have occurred without such a competent collaborator. Joel Friedman always had an open ear whenever my math abilities ran thin and often pointed me to the right nuggets to move forward. Michael Cohen helped clarify many of the underlying ideas through innumerable discussions and the generous sharing of his experience in all things radiosity. Wolfgang Krueger never tired of reminding me how important good science is and continues to amaze me with his knowledge. Last but not least the help, input and overall atmosphere provided by my many and now rather dear colleagues, Craig Kolb, Don Mitchell, Heidi Dangelmeier, Michael Cox, Seth Teller, David Laur, John Danskin, Reid Gershbein, Gordon Stoll, Celeste Fowler, and Larry Aupperle, has made a perhaps harder to quantify, but nonetheless very important contribution to my work and happiness at Princeton.

The research reported in this thesis was made possible in part through support provided by Apple, Silicon Graphics, and the NSF (contract no. CCR 9207966).

# Contents

# List of Figures

# Chapter 1

# Introduction

One of the core problems of computer graphics is the accurate computation of the distribution of light in a given scene. This is often referred to as the problem of photo-realistic rendering: the attempt to compute an image of a scene which is not distinguishable by a human observer from a photograph of the same scene. This is a very demanding challenge and we are far from getting close to achieving this goal. Nonetheless, impressive progress towards this goal has been achieved over the last few years and this dissertation attempts to make a contribution towards this goal.

To model light we must begin with a model of light propagation and behavior. Clearly, a very detailed physical model of light, e.g. one that accounts for quantum effects, would overwhelm our computational abilities in all but the simplest geometric configurations. Luckily most of the visual world as it presents itself to the human observer is dominated by effects adequately modeled by geometrical optics. Notable exceptions include for example interference effects such as those visible on oil films or soap bubbles. In the present work we will only consider geometrical optics and linear effects as they are described by linear transport models. This excludes such non-linear effects as temperature dependent reflection or emission, for example.

The history of computing illumination based on geometrical models is very old and can be traced to Lambert and the publication of his "Photometria" [40]. For example, Lambert already understood how to compute the illumination at a point due to a polygonal lightsource of uniform brightness. He gave a closed form expression

which is still used today in computer graphics.

In the early days of computer graphics, most illumination models considered only local properties to describe the shading at a point on a surface [10, 30, 11]. Light sources were considered as point sources and only the light reaching a surface on a direct path from the light source was accounted for. Since then many algorithms have been proposed to account for so called global effects, such as for example the interreflection between diffuse surfaces. More complete techniques are all based on a linear transport model of radiative transfer which describes the equilibrium transport between a number of surfaces, given a set of emitters and appropriate descriptions of reflection properties. The global nature of these descriptions requires costly computations and many simplifications have been proposed to reduce the computational problem to a manageable size. Some are based on point sampling techniques such as raytracing [70]. Another set of computational procedures is based on finite element techniques. It is these that we will concern ourselves with.

The first such approaches published in the computer graphics literature were based on a technique pioneered in radiative heat transfer: Radiosity [26, 47]. Classical radiosity uses a power balance argument to derive a system of linear equations using the assumption that all surfaces have diffuse reflection and emission properties. That is, all directional effects (e.g., a surface which reflects light preferentially along the mirror direction) are ignored. This leads to a reduction in the dimensionality of the problem and is a reasonable simplification for a large class of interior (architectural) scenes. Although still a very expensive computation for scenes of even moderate complexity, significant progress has occurred with the recent introduction of two approaches for the radiosity problem: Higher order Galerkin methods [71, 68] and hierarchical methods [32]. The former decrease the overall computational cost by a constant factor, while the latter decrease it asymptotically. These two techniques are unified under the framework of wavelets in this thesis.

Progress for the more general radiance problem, which allows for directionally varying properties in both reflection and emission, has been much slower. It is still largely impractical to compute for anything but simple configurations. Without new

and more efficient techniques this more general problem will continue to dwarf computational resources for the foreseeable future. If we hope to achieve rendering quality which deserves the name photo-realism the development of efficient techniques for the radiance problem in particular is crucial.

## 1.1 Outline

In this dissertation we will develop and discuss the use of wavelets for illumination (radiosity and radiance) computations. Because of the historical development of these techniques we begin with the case of radiosity and proceed to the application of these numerical techniques to the radiance problem. As such this dissertation is divided into two parts. In the first we lay the foundations for the use of wavelet techniques to solve the illumination problem. The second part is devoted to the application of these techniques to the case of radiance computations.

Since these techniques unify approaches which were first used in the context of radiosity, and also to make our explanations concrete we use the example of radiosity when describing the basic arguments in the first part. In particular we describe two basic techniques, the use of the Galerkin framework to solve integral equations, and the use of hierarchical arguments to reduce the asymptotic complexity of such computations. We will give much room to describing and analyzing the hierarchical arguments in particular. Using geometric insights, which, it is hoped, are simple and elegant, we explain and analyze hierarchical radiosity. The geometric arguments are chosen so as to directly mirror the underlying mathematical arguments and thus provide for a rigorous foundation while being intuitively obvious. Once these foundations are in place we give a very short introduction to the rather rich field of wavelets. In that section the focus is on the relevant facts for the application of wavelets to the solution of integral equations. Using the theory of linear operators expressed in wavelet bases and, in particular, the fact that operators satisfying certain smoothness conditions will lead to sparse linear systems in wavelet bases, we recognize hierarchical radiosity and its geometric arguments as an instance of wavelet methods applied to illumination computations. Reviewing some of the published results quantifying

the power of these techniques in the case of radiosity we proceed to the more general radiance problem.

The radiance problem has recently seen the application of hierarchical approaches as well [6] and we extend these to higher orders by use of wavelets. The described methods have been implemented and we report on the application of these techniques to a number of geometric configurations. Finally we draw conclusions from our experiments and close with recommendations for future research in this area.

## 1.2 Contribution

When hierarchical approaches were first presented their complexity arguments rested on geometric reasoning, which was intimately tied to the use of constant basis functions. Our contribution lies in the following main points:

- By interpreting the hierarchical algorithms as instances of a family of wavelet algorithms, we embed them in the rich and well studied framework of wavelet theory. This allows us to generalize the underlying techniques to higher orders in a way which would be hard to do by geometric reasoning alone

- Reinterpreting the earlier geometric arguments of Hanrahan et al. [32] in the strict language of Calderon-Zygmund operators we end up giving a new geometric proof for the sparsity of abstract integral operators in wavelet bases [9]. The relevant mathematical quantities have a simple geometric interpretation even if the underlying integral equation has no relationship to any actual geometry. Using this interpretation a proof of the linear asymptotic complexity of the approximation of smooth integral operators in wavelet bases follows trivially

- The wavelet framework allows us to reach a more detailed understanding and characterization of the errors involved in the numerical approximations

- Our reduction to practice of these wavelet approaches leads us to new insights not previously treated in the numerical analysis literature. In particular, we

propose and analyze a design for an *oracle*, which is crucial in achieving the promised linear bound on the asymptotic complexity.

The historical development of the ideas presented in this dissertation proceeded out of a collaboration with Steven Gortler. In it the unification of Galerkin methods and Hierarchical methods under the framework of wavelets for the sparse realization of the radiosity integral operator was achieved. This work was documented in two papers [28, 54]. While the first part of this dissertation deals with these ideas and the radiosity problem in particular we have attempted to not duplicate any of the information already present in the two previously published papers. Instead we attempt to describe the underlying ideas in a purely geometric framework while being truthful to the exact underlying operator theory. Consequently many of the implementation details are not treated but left to the papers. The resulting text is perhaps more tutorial in nature (in fact it is an edited version of a Siggraph tutorial note [53]) and the interested reader is referred to the previously mentioned papers for implementation details. We will however review some of the results reported in these early papers to support our claim that the wavelet techniques yield great benefits for the computation of radiosity problems.

The second part of this thesis is an edited and expanded version of a paper published with Pat Hanrahan in the 5th Eurographics Workshop on Rendering [55]. It covers the application of wavelet techniques to the more general radiance problem and shows the improvements possible with wavelets in this context. Here many implementation details and choices are discussed, while the underlying ideas are assumed from the first part of this thesis.

# Chapter 2

# Radiosity

## 2.1 Introduction

The computation of radiosity $\frac{W}{m^2}$ (i.e., power per unit area) on a given surface is a widely used technique in computer graphics to solve for the illumination in an environment. Radiosity is governed by an integral equation which arises from a more general integral equation known as the rendering equation [37] when one assumes that all reflection occurs isotropically. Solving the underlying integral equation exactly is not possible in general. Thus numerical approximations must be employed leading to algorithms which are generally very expensive.

Much research has been conducted into accelerating algorithms for the numerical approximation of radiosity solutions. In this chapter we will discuss some recent techniques which drastically improve the efficiency of such algorithms. Wavelet radiosity (WR) was first introduced by Gortler *et al.*[28] and Schröder *et al.*[54]. Their algorithm unifies the benefits of Galerkin radiosity (GR) [33, 34, 71, 68] and hierarchical radiosity (HR) [32].

The original papers [28, 54] explain the implementation of such a system giving many of the details including pseudo code. For this reason we will concentrate in this chapter on the arguments behind the method to augment the original papers. Using the historical development and more geometric arguments we attempt to motivate WR. In a later part of this chapter we will also treat the underlying operator theory.

At that point the geometric arguments should make it easier to see the flow of the mathematical arguments.

Before going into the development and discussion of algorithms we first fix our notation by beginning with the formal problem definition.

### 2.1.1 Problem Definition and Notation

Radiosity $B(\vec{y})$ is a function defined over all surfaces $M^2$ which make up a given scene. It is parameterized by some (2D) set of surface intrinsic parameters $\vec{y}$. Here and for the remainder of this dissertation we will assume that our scenes $M^2$ consist of a collection of planar quadrilaterals. The planarity requirement serves to simplify visibility computations which are needed in an actual implementation. Rectangular parameter domains allow for cross product basis constructions, another convenience which is not mathematically necessary but simplifies the implementation[1]. Allowing more complex primitives, such as bi-cubic patches for example, raises implementation issues such as self shadowing and visibility tests, but the underlying mathematical theory continues to be valid. We do however fundamentally assume the existence of a parameterization of our scenes, which excludes surfaces for example for which only implicit forms are known.

Given such a set of primitives and their parameterizations the radiosity $B(\vec{y})$ is governed by a Fredholm integral equation of the second kind which describes the total radiosity as a sum of emitted and reflected radiosities

$$B(\vec{y}) = B^e(\vec{y}) + B^r(\vec{y})$$

This equation holds for all wavelengths. Since we do not allow any coupling between wavelengths (e.g., absorption at one wavelength followed by re-emission at another) it is enough to consider all equations to be written for one wavelength. In practice we will assume three representative wavelengths, red, green, and blue (rgb), in effect solving always three instances of our problem in parallel.

[1] In fact our current implementation allows for triangles as degenerate quadrilaterals with no ill numerical effects.

The reflected component of radiosity arises from the irradiance at the surface point $\vec{y}$ after it is reflected

$$B^r(\vec{y}) = \rho(\vec{y})E(\vec{y})$$

Here $\rho : \mathrm{M}^2 \rightarrow [0,1[$ describes the fraction of power received at a point which is reemitted. For physically realistic scenes $\rho$ will always be bound away from 1, a property we will take advantage of later to argue convergence of our solution algorithms. The reemission is assumed to occur isotropically, thus there is no direction dependence as we will see it later for the case of radiance. The irradiance $E(\vec{y})$ results from the radiosities arising at points on the visible hemisphere above $\vec{y}$ and is defined as an integral over this hemisphere. We parameterize it with respect to other surfaces and write it as an integral over areas

$$E(\vec{y}) = \int_{\mathrm{M}^2} G(\vec{x},\vec{y})\pi^{-1}B(\vec{x})\,dA_x$$

The radiosity arising at all other surfaces $B(\vec{x})$ is scaled by $\pi^{-1}$, which normalizes the integral, and weighted by the function $G$, which is the parameterized version of the measure, including the characteristic function of visibility since we are integrating over all of $\mathrm{M}^2$

$$G(\vec{x},\vec{y}) = \frac{\cos\theta_x \cos\theta_y}{r_{xy}^2}\,\mathrm{v}(\vec{x},\vec{y})$$

The cosines measure the orientation of the surfaces with respect to the line connecting $\vec{x}$ and $\vec{y}$ (see Figure 1). $r_{xy}$ measures the length of this line and its square accounts for the falloff of angle subtended (by a surface element) with distance. $\mathrm{v}(\vec{x},\vec{y})$ is the visibility characteristic function taking on values in $\{0,1\}$ depending whether the line between the two surface points parameterized by $\vec{x}$ and $\vec{y}$ is obscured or unobscured respectively.

Taking all of the above elements together we arrive at the classic radiosity integral equation

$$B(\vec{y}) = B^e(\vec{y}) + \rho(\vec{y}) \int_{\mathrm{M}^2} G(\vec{x},\vec{y})\pi^{-1}B(\vec{x})\,dA_x \qquad (1)$$

The task at hand then is to numerically approximate the following problem:

> Given the reflectances $\rho$, geometry $\mathrm{M}^2$, and the boundary conditions $B^e$ (the light sources), find $B$.

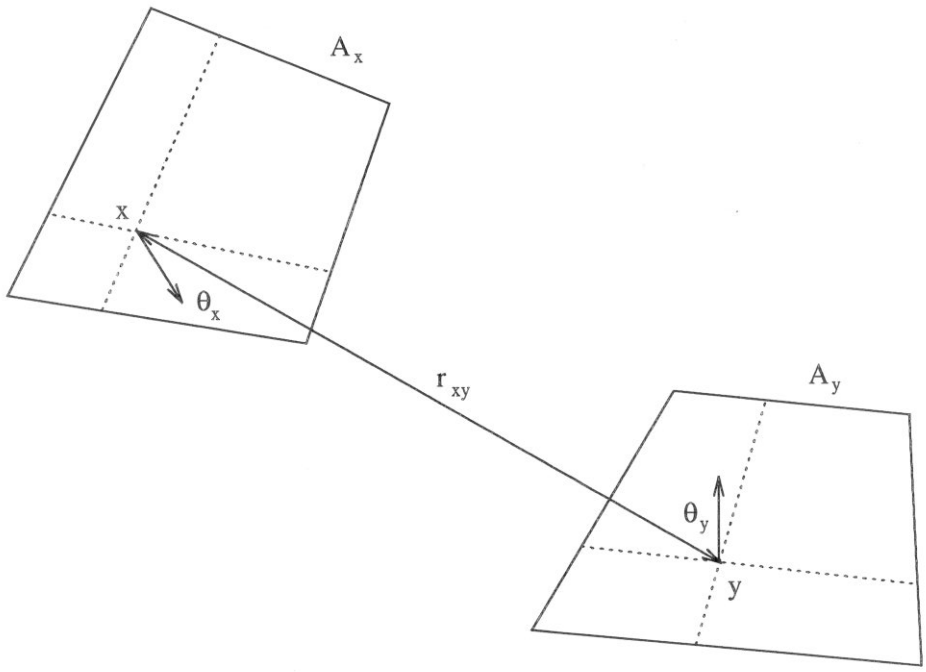



Figure 1: Geometry for the transport between two surfaces. $\theta$ denotes the angles that the vector connecting two points on the surfaces makes with the local surface normals. We assume that points on either surface are given with respect to some intrinsic coordinates $\vec{x}$ and $\vec{y}$ respectively.

Solving Equation (1) exactly is generally only possible for trivial geometric configurations. Most such solutions rely on the exploitation of symmetry (e.g., inside a sphere) or infinite extent assumptions (e.g., two parallel planes). Most of the applications were initially in illuminating engineering [25, 45]. With the advent of computers, radiosity computations were performed extensively in the radiative heat transfer community. They dealt with the difficulty of computation by discretizing the above integral equation and making a constant radiosity assumption for each resulting surface element. Typically the problems, such as the computation of radiative transfer inside simple chambers, or for the skin of satellites [58], were still very simple compared to the complex geometric configurations of computer graphics.

The adoption[2] of radiosity techniques in computer graphics occurred with the publication of an illumination algorithm based on it by Goral *et al.*[26] in 1984 and Nishita *et al.*[47] in 1985. The main problem in computer graphics turned out to be the management of the vastly more complex scenes. Discretization into a few hundred elements for a cuboidal enclosure leads to only moderately expensive linear algebra problems. However, dealing with the arbitrary configurations of possibly thousands of input primitives as one typically finds in computer graphics quickly leads to numerical computation problems whose size (in both space and time) overwhelms even large computational resources.

Thus the focus of the development of radiosity techniques in computer graphics has been on finding more efficient ways to compute the radiosity for a given scene. Most of the early techniques focused on algorithmic developments such as the employment of graphics hardware [17], substructuring [18], Southwell relaxation [16], and raytracing [69]. Only in the last few years have graphics researchers revisited the numerical fundamentals of the solution of integral equations and in the process moved the state of the art forward significantly. In the next section we will review two of those techniques, GR, and HR, which are direct antecedents of wavelet techniques.

[2]One might argue the term "reintroduction" because of the earlier use in illumination computations as far back as Lambert [40].

### A Note on Dimensionality

The radiosity equation for 3D is defined over 2D parameter domains $\vec{y}$. Many times in our exposition it will be easier to deal with radiosity in a plane where line segments interact ("Flatland" radiosity [33]). In this way, all parameterizations will be written as $x$ or $y$ and many techniques will be easier to explain and visualize. The equation will retain its structure although $G$ will only have a single power of $r_{xy}$ in its denominator and the factor $\pi^{-1}$ will become $2^{-1}$. In an abuse of notation we will from now on quietly assume that the factor $\pi^{-1}$ ($2^{-1}$ respectively) is absorbed into the definition of $G$. This frees us from having to distinguish between the two different factors and simplifies our equations. However, any implementation has to be aware of using the right factor when performing actual computations. We will also make liberal use of the symbol $\frac{dA_x}{dx}$ to stand for the size of the area element due to the parameterization in both the flatland and 3D cases.

The chief advantage of this reduction in dimensionality lies in the fact that many quantities, which we have to manipulate, have a number of subscripts or superscripts which is directly proportional to the dimensionality of the problem. It is easy to loose sight of the essential ideas unless we limit ourselves to the 1D domain (flatland) case. Where appropriate we will be explicit about the changes necessary to go to 2D domains (radiosity in 3D). In general the differences are limited to more indices to manipulate, or in the case of a program, more array dimensions to iterate over.

## 2.1.2 Numerical Solution

A canonical solution technique for integral equations such as the radiosity equation (1) is the weighted residual method [21], a special case of finite element methods. The historical development of radiosity algorithms did not proceed from the theory of integral equations and finite element methods. Instead the first approaches were motivated by power balance arguments [26, 47]. Only recently [33] was the traditional mathematical framework brought to bear on these problems. Later on in this chapter we will more formally treat solution methods for integral equations. For now we give just enough of an exposition of the finite element method to understand why we are

trying to solve linear systems. The rest of this chapter discusses techniques which help solve linear systems efficiently.

We begin by noting that the radiosity integral equation has the structure of a linear operator. Introducing the linear operator $\mathcal{G}$

$$\mathcal{G}(B) = \int_{M^2} G(x,.)\frac{dA_x}{dx}B(x)dx$$

which computes irradiance $\mathcal{G}(B) = E$ and the linear operator $\mathcal{S}$ which turns irradiance into radiosity

$$\mathcal{S}(E) = \rho(.)E(.)$$

we can frame the problem of solving the radiosity equation in the context of linear operators and their solution

$$(I - \mathcal{S} \circ \mathcal{G})B = B^e \tag{2}$$

Given some right hand side $B^e$ we effectively need to invert $I - \mathcal{S} \circ \mathcal{G}$ to find the unknown $B$. Since the spectral radius of $\mathcal{S} \circ \mathcal{G}$ is strictly less than one ($\rho \leq 1 - \epsilon$ for some $\epsilon > 0$) we can use a von Neumann series to describe this inverse

$$(I - \mathcal{S} \circ \mathcal{G})^{-1} = \sum_{i=0}^{\infty}(\mathcal{S} \circ \mathcal{G})^i$$

which has a satisfying physical interpretation. $B$ can be written as a sum of successively reflected contributions. First the direct component $B^e$, than the component reflected through the environment once $\mathcal{S} \circ \mathcal{G}(B^e)$, and so forth. In practice the spectral radius of our operator is typically so small that this iterative computation converges within a few iterations.

The main advantage of the observation that the radiosity equation gives rise to a linear operator problem lies in the fact that we can understand the action of the radiosity operator by studying its behavior with respect to a set of basis functions. As such, Equation (2) is infinite dimensional (we take its domain and range to be $L^2$) and we need to work with finite dimensional approximations to it. This is done by limiting ourselves to some finite dimensional subspace. Doing so reduces our original problem of finding a solution to the radiosity equation to the problem of solving a

matrix problem. A particular way of doing so is referred to as the weighted residual method [21].

Any of the weighted residual methods can be derived by expanding all functions in the radiosity equation with respect to some finite set of basis functions $\{N_i\}_{i=1,\ldots,n}$, $B(x) = \sum_{i=1}^{n} b_i N_i(x)$ and a reprojection step. For example, the basis functions might be piecewise constant or polynomials up to some order. The reprojection step induces a linear constraint on the residual $r$ for a given approximation $\hat{B}$

$$r(\hat{B}) = B^e - (I - \mathcal{S} \circ \mathcal{G})(\hat{B})$$

Notice that $r$ will be zero for the exact solution. For example, we can ask that $r$ be orthogonal to the space spanned by the chosen $\{N_i\}$, resulting in the Galerkin methods. Other ways of "weighting the residuals" are possible, but we will stay with the Galerkin method. A solution to the Galerkin form of the weighted residual method insures that the residual is orthogonal to the image of the basis functions under our operator. Other choices such as a least squares solution, which insures that the residual is orthogonal to the basis functions themselves, or a point collocation method which effectively uses Dirac delta distributions in the reprojection step are possible. In either case the result is a system of linear equations in the unknown coefficients $b_i$

$$\forall i: \; b_i = b_i^e + \rho_i \sum_{j=1}^{n} G_{ij} b_j \tag{3}$$

For the Galerkin method the coefficients of this system are integrals of the form

$$G_{ij} = \langle\langle G_A, N_j\rangle, N_i\rangle = \int_{\text{supp } N_i} \int_{\text{supp } N_j} G(x,y) \frac{dA_x}{dx} N_j(x) N_i(y)\, dx\, dy \tag{4}$$

where we defined $G_A(x,y) = G(x,y)\frac{dA_x}{dx}$ for notational convenience. Note that the well known form factors arise as $F_{ij} = G_{ij}$ when $N_i = 1$ and $N_j = 1$ (over the support of elements $i$ and $j$ respectively).

Note that we separated $\rho$ and moved it in front of the sum in Equation (3). In so doing we have made another simplifying assumption which is often made, namely that the reflectance is a constant over the support of each basis function. This assumption is not necessary as is shown by Gershbein *et al.*[24]. We make it here since by far

the largest part of the computational challenge in approximating radiosity solutions arises from the approximation of $\mathcal{G}$, not $\mathcal{S}$. This is easily seen by observing that the computation of irradiance is a global operation involving quadratures and visibility computations over possibly large numbers of primitives. In contrast the application of $\mathcal{S}$ is purely local and only a simple multiplication.

In order to write the coefficients in as simple a form as Equation (4) we have implicitly assumed that we are dealing with orthonormal basis functions as is customary for the Galerkin method. If in fact we want to use a bi-orthogonal basis system[3], a freedom which can be useful for certain wavelet bases, the reprojection step has to be performed against the dual basis functions. The basic principles are unperturbed, however we need to carefully keep track of which basis (primal or dual) is appropriate at which point in the computation. The presentation in this thesis will not distinguish between primal and dual bases, and ignore any scaling factors due to non-normalized basis functions. All bases will be orthonormal. This will simplify our notation considerably and help bring out the essential properties of the equations.

For purposes of the following discussions we only need to remember that the coefficients of the linear system we wish to solve arise from (inner product) integrals between the kernel function $G_A$ and our chosen basis (Equation 4). For this reason the coefficients are often called couplings or interactions to remind us that they have the physical interpretation of measuring how much one basis function physically interacts or couples with another. The coefficients also "inherit" some of the properties of $G_A$, a fact that we will take advantage of in wavelet methods.

## 2.2 Galerkin and Hierarchical Solvers

Two important techniques have been discussed in the recent radiosity literature, GR and HR. In this section we give a short overview of these two techniques and explain how they found their culmination in the development of the wavelet radiosity algorithm.

[3]Two bases $\{N_i\}$ and $\{\tilde{N}_j\}$ are bi-orthogonal iff $\langle N_i, \tilde{N}_j \rangle = \delta_{ij}$.

### 2.2.1 Galerkin Radiosity

Galerkin radiosity, first introduced by Heckbert [33, 34] aims to increase the order of basis functions used in radiosity algorithms. In this context, classical radiosity (CR) [26, 47] is seen to be a Galerkin method using piecewise constant functions. The original goal of applying Galerkin methods to radiosity was to improve the quality of the answers computed, as well as the efficiency of the computations. In particular, using higher order basis functions allows the use of much coarser meshes than CR required while still meeting a requested error bound. In his original work, Heckbert applied these ideas in a flatland environment using piecewise linear basis functions. More recently Troutman and Max [68] and Zatz [71] have applied higher order basis functions to the computation of 3D radiosity. Zatz in particular has pushed the ideas to their extreme by leaving many surfaces unmeshed. Instead he increased the polynomial order of the basis functions so that the radiosity even over large surfaces, such as entire walls, could be computed with high accuracy without any subdivision.

### 2.2.2 Hierarchical Radiosity

The first use of hierarchies was made by Cohen *et al.*[18]. They introduced a technique called substructuring based on the observation that a fine subdivision was only necessary on the receiver of a transport of light, while a coarser subdivision was sufficient on the source. Since the roles of receivers and sources are reversible a two level hierarchy over each geometric primitive resulted. These ideas were developed further in a paper by Hanrahan *et al.*[32]. They introduced HR, which applied some arguments from the n-body literature [3, 31, 8] to CR. In their approach a possibly very deeply nested subdivision hierarchy was imposed on every primitive. Light transport was allowed to occur throughout these hierarchies. They showed that to within some user selectable error tolerance a linear number of interactions amongst all possible interactions was sufficient to compute an accurate answer. Because the algorithms up to that point always used a quadratic number of interactions HR improved the performance of radiosity computations considerably.

### 2.2.3 Performance Analysis

To put these two techniques and their respective advantages into perspective we need to look at their costs. Given $k$ input surfaces, say quadrilaterals, any one of the above algorithms will use some number of basis functions $n$ defined over the totality of input surfaces. For example in the case of CR the surfaces are typically subdivided into many elements with each element carrying an associated constant basis function (whose support is exactly the element itself). In this case, $n$ elements correspond to $n$ basis functions. Similarly, for higher order Galerkin methods we will probably do some meshing into elements as well, albeit not as fine a mesh. Each resulting element will then typically carry some number of basis functions. For example, if we are using piecewise linear basis functions each surface (2D) element will typically have four basis functions associated with it. For each parameter axis we need two basis functions (constant and linear) and we have two parameter axes for a total of four combinations. In general an $M-1$ order piecewise polynomial basis will have $M^2$ basis functions defined over each (2D) element. Counting in this manner it makes sense to talk about $n$ basis functions in total for $\frac{n}{M^2}$ elements.

Once we have a set of $n$ basis functions the Galerkin method will give rise to a linear system relating all of these basis functions with each other resulting in a system of size $O(n^2)$ (see Equation 3). This linear system needs to be solved to find the coefficients of all the basis functions. Using some iterative solver the solution cost is proportional to $O(n^2)$. The constant (i.e., the number of iterations of the iterative solver) is directly related to the spectral radius of our operator [64]. As mentioned earlier for practical applications (realistic reflectances) this radius is usually significantly smaller than 1 leading to fast convergence. While it is hard to make this statement quantitatively precise it support the experiential observation that radiosity linear systems converge after only few iterations.

GR, by going to higher order bases, manages to decrease $n$ and thus get efficiency gains. Even though the number of bases per element increases ($M^2$) the number of elements necessary for a given overall accuracy falls faster for a net gain. To see why this is, we use the fact that a Galerkin method using a piecewise polynomial basis of order $M-1$ will have an accuracy of $O(h^M)$, where $h$ gives the sidelength of the

elements in the mesh [21, 39]. To make this concrete, suppose we are willing to allow an error proportional to $\frac{1}{256}$. Using piecewise constant basis functions, $h$ would have to be on the order of $\frac{1}{256}$ to meet our goal. Now consider piecewise linear functions. In this case $h$ would only need to be on the order of $\sqrt{\frac{1}{256}} = \frac{1}{16}$. So even though the number of basis functions per element goes up, we still come out ahead. In the case of flatland there are two linear basis functions per element and we go from $n = 256$ to $n = 2 * 16$ bases total. In 3D radiosity were we have $2 * 2$ linear basis functions per element $n$ goes from $256^2$ down to $(4 * 16)^2$ basis functions overall.

We have seen that for $n$ basis functions we have $O(n^2)$ interactions in general. It is also immediately clear on an intuitive level that not all interactions are equally important. HR makes this statement precise and takes advantage of it to reduce the number of interactions which need to be computed, to $O(n)$. For example, "far" interactions do not need as much subdivision as "near" interactions. The exact argument as to why $O(n)$ elements are enough will be given below. However, even if we can make statements about the number of elements generated during meshing, and how they will interact, we still need to consider at least one interaction between each pair of the original set of incoming surfaces. Consequently the work of an HR algorithm will be $O(k^2 + n)$. Even though there still is a $k^2$ dependence we will often have $n >> k$ resulting in significant savings. Note that in a case in which the original set of $k$ surfaces is presented premeshed as $n$ elements HR will be reduced to CR. Thus it will perform no worse, but in practice often dramatically better, than CR.

### 2.2.4 Wavelet Radiosity

Wavelet radiosity, introduced in [28, 54], aims to unify the advantages of both GR and HR. Following the lead of HR it uses a multi-level structure of basis functions over all surfaces to gain the asymptotic improvement from $n^2$ to $n$. It also gains the advantages of GR by using higher order basis functions to get the improvement due to coarser meshing and a decreased number of basis functions (smaller $n$) overall.

In the following section we will first take a closer look at the basic structure of the HR algorithm and then show how it can be extended to higher order bases. The

geometric argument which we use to show that HR uses only $O(k^2 + n)$ interactions will be given in a way that it immediately applies to higher order methods as well. At that point we will make the connection with wavelets and show how wavelets and the theory of linear operators realized in wavelet bases can be used to abstract the geometric ideas of the HR algorithm. Making this connection with wavelets allows us to apply the basic ideas of HR in a much larger set of circumstances. We can also generalize HR to higher order basis functions and make explicit claims about the errors involved.

This development of the ideas is somewhat unusual in that we do not start with the theory of linear operators and their sparse realization in wavelet bases to then proceed to show how these ideas can be exploited for radiosity. Instead we will pursue a more intuitive approach (which also happens to follow the historical development more closely). While being somewhat "messier" from the pure mathematical vantage point it will be more appealing to the geometric intuition and easier to relate to actual implementations. The readers can rest assured that the mathematical underpinnings are solid and those interested will find the details in a later part of this chapter.

## 2.3 Algorithms

All radiosity algorithms have roughly two components for purposes of this discussion. These can be described as setting up the equations (i.e., *computing* the entries of the linear system) and *solving* the linear system. The latter typically invokes some iterative solution scheme, for example Jacobi or Gauss Seidel iteration [64], or Southwell relaxation [29]. In actual implementations, these two phases are often intermixed, for example when refining a subdivision mesh (adding basis functions) during successive iterations. Nonetheless we can distinguish these two fundamental operations in our algorithms. Since the actual act of iterating (i.e., performing row updates) or matrix/vector multiplies is much simpler we will first focus on the aspect of setting up the equations.

### 2.3.1 Hierarchical Radiosity

Hierarchical radiosity considers the possible set of interactions in a recursive enumeration scheme. We have to insure that every transport (i.e., every surface interacting with other surfaces) is accounted for once and only once. Energy must neither be missed, nor introduced into the simulation multiple times. To do this we call the following procedure for every input surface with every other input surface as a second argument (once again we consider the problem over 1D domains)

```
ProjectKernel( Element i, Element j )
    error = Oracle( i, j );
    if( Acceptable( error ) || RecursionLimit( i, j ) )
        G_ij = Quadrature( i, j );
    else
        if( PreferredSubdivision( i, j ) == i )
            ProjectKernel( LeftChild( i ), j );
            ProjectKernel( RightChild( i ), j );
        else
            ProjectKernel( i, LeftChild( j ) );
            ProjectKernel( i, RightChild( j ) );
```

This procedure consists of several parts which we discuss in turn.

First we call a function `Oracle`, which is capable of estimating the error across a proposed interaction between elements `i` and `j`. If this estimated error satisfies the predicate `Acceptable`, the required coefficient is created by calling a quadrature routine which evaluates the integral of Equation 4. We have, in effect, created an entry in the matrix system, as well as implicitly decided on a particular basis function. However, resource limitations may require us to terminate the recursion even if the error is not acceptable yet. This predicate is evaluated by `RecursionLimit`. We will show in the error analysis below that we can always force the error associated with a particular transport below any arbitrary positive constant. From this point of view we are always assured that `Acceptable` will become true at some point and `RecursionLimit` will not need to be invoked. However, we may not have enough

storage to subdivide elements enough for a very low error threshold request. In this case `RecursionLimit` will be needed. Nonetheless, we hope that it will be invoked only for a small number of cases so that at least over large parts of the solution we can enforce the requested error bound.

If the error is too high we recurse by subdividing. Typically we will find that the benefit in terms of error reduction is not equal for the two elements in question. For example, one element might be much larger than the other and it will be more helpful to subdivide the larger one in order to reduce the overall error (see the error analysis below). This determination is made by `PreferredSubdivision` and a recursive call is initiated on the child interactions which arise from splitting one of the parent elements. For 2D elements there would typically be four recursive calls each, not two. The preferred element would be split into four children (quadrants).

As mentioned earlier, the process of iterating and subdividing is not typically separated in a real implementation. For example, we could imagine that the predicate `Acceptable` takes into account

- the brightness of the sender (brightness refinement [32]) by multiplying the error estimate with the amount of power transported over the given link before comparing to the error threshold;
- the importance of the receiver (importance refinement [63]) again by multiplying the error estimate with the importance magnitude before comparison with the error threshold;

The error threshold may itself become smaller upon successive iterations (multi-gridding [32]), creating a fast but inaccurate solution first and using it as the starting point for successive solutions with lesser error. Any of these techniques we might refer to as refinement. Thus we will typically reexamine interactions created in an earlier iteration when iterating again.

In an implementation, this is easily done by keeping a list of all $G_{ij}$ created and calling a modified version of `ProjectKernel` on these before the next iteration. If none of the parameters which influence `Acceptable` had changed, `ProjectKernel` would simply return; otherwise it would delete the interaction $G_{ij}$ because it has too

much error and replace it with a set of finer interactions. This would correspond to replacing some set of basis functions (and their interactions) with a new and finer set of basis functions (and their interactions).

From the structure of the recursion, it is clear that every transport will be accounted for once and only once. The remaining task is to show that for a strictly positive amount of allowable error[4] we will create only a linear number of interactions amongst all the (implicit in the subdivision) basis functions created. Furthermore we need to show that the function `Oracle` can be implemented in an efficient way.

### 2.3.2 Bounding the Error

Recall that the ultimate goal is the approximation of the original integral equation (Equation 1), as best as possible, with finite resources. Because of the particular structure of the radiosity integral equation the set of possible solutions comes from an infinite dimensional space of functions. In theory we would need an infinite sized matrix system to compute an exact answer. Using a finite matrix system therefore has to entail some amount of error before we even start to compute. There are other sources of error, such as imprecision in the specification of the boundary conditions or the geometry (for a careful analysis of these see Arvo *et al.*[4]). Here we will only concern ourselves with errors due to the particular basis functions chosen. Expressed differently, we are examining the errors which arise from replacing an otherwise infinite sized matrix with a finite matrix.

We proceed by analyzing the function `ProjectKernel` more closely to understand how many recursive calls it will generate. Again in order to streamline the presentation we first analyze the case of radiosity defined over 1D domains (flatland radiosity). When we used the name `ProjectKernel` we already anticipated one meaning of the $G_{ij}$ coefficients which we will now use to analyze the relationship between allowed error and number of interactions necessary.

In order to proceed we need an inner product. The inner product between two

[4]Imagine `Acceptable` always returns `False`. In this case the recursion would always bottom out and in fact all $n$ bases at the finest level of meshing, as determined by `RecursionLimit` would interact, resulting in $n^2$ interactions.

arbitrary functions $f$ and $g$ is defined as $\langle f, g \rangle = \int f(x)g(x)\,dx$. Recall that in finite dimensional vector spaces (e.g., Euclidean three space) inner products of some vector $\mathbf{y}$ against some set of orthonormal basis vectors $\mathbf{e}_i$ give the expansion coefficients $y_i$ of the vector $\mathbf{y}$ with regard to the basis: $\mathbf{y} = \sum_i y_i \mathbf{e}_i$. The case for function spaces is exactly analogous.

Recall the definition of $G_{ij}$ (Equation 4). Supposing that our basis functions are orthonormal[5] we may interpret the $G_{ij}$ as expansion coefficients of $G_A$ as follows

$$\begin{aligned} G_{ij} &= \int_{\text{supp } N_i} \int_{\text{supp } N_j} G_A(x,y)\, N_j(x) N_i(y)\, dx\, dy \\ &= \langle \langle G_A, N_j \rangle, N_i \rangle \\ G_A(x,y) &\approx \hat{G}_A(x,y) = \sum_{i,j=1}^{n} G_{ij} N_j(x) N_i(y) \end{aligned} \tag{5}$$

In other words, computing some set of $G_{ij}$ is equivalent to approximating the function $G_A(x,y)$ with a *projected* version $\hat{G}_A(x,y)$. We say *projected*, since our finite basis only spans some subspace of the otherwise infinite dimensional space of functions we are operating in.

Before proceeding, we define the symbol $T = (I - \mathcal{S} \circ \mathcal{G})$ for notational convenience. Similarly, we will use the symbol $\hat{T}$ to denote our approximation of the actual transport operator. Using the fact that the radiosity integral equation is a bounded linear operator for physically realistic reflectances [4], the error in our computed solution $\hat{B}$ against the actual solution $B$ can be bound as follows

$$\begin{aligned} \|\hat{B} - B\| &= \|\hat{T}^{-1}(B^e) - T^{-1}(B^e)\| \\ &\leq \|\hat{T}^{-1} - T^{-1}\|\, \|B^e\| \\ &= \|\hat{T}^{-1}(T - \hat{T})T^{-1}\|\, \|B^e\| \\ &\leq \|\hat{T}^{-1}\|\, \|T - \hat{T}\|\, \|T^{-1}\|\, \|B^e\| \\ &= C\|T - \hat{T}\| \\ &= C\|G_A - \hat{G}_A\| \end{aligned}$$

where we assumed that $B^e$ can be realized exactly in the chosen basis, an assumption

[5] In the bi-orthogonal case is similar with inner products against the primal basis functions determining the coefficients of an expansion with respect to the dual basis and vice versa.

which is generally true in practice. $C$ is some constant associated with the input (geometry, emission, and reflectances), but independent of the basis functions used. Clearly given a user selectable $\epsilon > 0$ the error in the computed solution can be forced below $\epsilon$ by making $\hat{G}_A$ sufficiently close to $G_A$.

This approach to bounding the error is very conservative. By enforcing a good approximation of the kernel function we enforce a good approximation on our solution. Note that it is possible to have a good approximation to the correct solution even though the kernel approximation may not be very good. To see this consider that an application of the operator always implies an integration. Even though the kernel may vary rapidly, by the time we have integrated over those variations they may make little contribution to the result (i.e., the integration acts as a strong smoothing operator). Other researchers have used similar approaches to bounding the error in their approximation. Smits *at al.*[63] use an upper and lower bound on the differential form factors across an interaction. Lischinski *et al.*[43] use individual estimates of upper and lower bounds on differential form factors, visibility, and brightness, which are combined to estimate a worst upper and lower bound for the variation of radiosity on the receiver of a transport. Both of these approaches are estimating the potential impact of the kernel approximation on the solution. A different approach was suggested in [42] by Lischinski *et al.* Instead of bounding the error a priori they actually compute an actual upper and lower bound solution to give an a posteriori error estimate. This latter approach can avoid the overly conservative nature of a priori derived error bounds.

Imagine that we are willing to allow some amount of error over the entire solution. If we divide this error by the area of the entire scene, we have a bound describing how much local error we can allow. Staying below this local error everywhere the total global error will be no larger than our allowed error. Given that the total area in a given scene is bounded we can make the following statement: Given some $\epsilon$, there exists some $\delta(\epsilon)$ such that if the error in each of the interactions we enumerate stays below $\delta$, the total error will stay below $\epsilon$.

This is a very important property, since it means that we can control the overall error by controlling local error. Now we already know that the simplest version (no

brightness, importance, or multi-gridding refinements) of the function `Acceptable` is a test of `error` against $\delta$.

### 2.3.3 Bounding the Number of Interactions

Before going into the mathematical details we point out that the ideas behind the geometric argument that we are about to make were first given by Hanrahan *et al.*[32]. However, we believe that the mathematical basis of their argument was not as rigorous as the one we are about to give. Furthermore, we give the argument so that it in fact applies to a larger class of operators, which will be discussed later in this chapter. We first give the argument for the case of piecewise constant basis functions, but our mathematical argument involving the mean value theorem remains valid for higher order basis functions. In fact later on we will see that the abstract theory of Calderon-Zygmund operators is the exact analog of our geometric arguments.

Suppose now that we stay in the framework of classical radiosity in so far that we only allow constant basis functions (as HR does [32]) and that we simply set $\hat{G}_A = G_A(x_0, y_0)$ where $x_0$ and $y_0$ are the midpoints of the respective intervals we are considering. The norms being used in the following argument are all understood to be the $L^2$ norm. We observe that

$$\begin{aligned} \|G_A(x,y)\| &\leq \frac{1}{r} \\ \|\partial_x G_A\| + \|\partial_y G_A\| &\leq \frac{C}{r^2} \end{aligned}$$

with $r$ the distance between the two points on the respective lines (surfaces) as measured in the surrounding space (2D for flatland; 3D for 3D radiosity). The bounds for 3D radiosity are $\frac{1}{r^2}$ and $\frac{C}{r^3}$ respectively. We also need the general version of the mean value theorem [22]

$$\begin{aligned} \|G_A(x_0,y_0) - G_A(x_0+t_x, y_0+t_y)\| &\leq \|(t_x,t_y)\| \sup_{\xi\in[0,1]} \|\frac{d}{d\xi} G_A(x_0+\xi\, t_x, y_0+\xi\, t_y)\| \\ &\leq C\frac{\|(t_x,t_y)\|}{\min r^2} \end{aligned}$$

where $(t_x, t_y)$ denotes the offset of some point $(x, y)$ from the center $(x_0, y_0)$. For what follows we make the observation that the magnitude of $(t_x, t_y)$, when integrated

over some parameter domain can be bounded above by the diameter of the parameter domain. This diameter in turn can be bounded, with the use of the triangle inequality, by the sum of all side lengths of the parameter domain.

Consider now the intervals of support $I_x$ and $I_y$ for some pair of basis functions respectively

$$\begin{aligned}
\|\hat{G}_A - G_A\| &= \left(\int_{I_y}\int_{I_x} \|G_A(x_0,y_0) - G_A(x,y)\|^2\, dx\, dy\right)^{\frac{1}{2}} \\
&\leq \frac{C}{\text{dist}^2(I_x,I_y)}\left(\int_{I_y}\int_{I_x} \|(t_x,t_y)\|^2\, dx\, dy\right)^{\frac{1}{2}} \\
&\leq C\frac{\|I_x\| + \|I_y\|}{\text{dist}^2(I_x,I_y)}\left(\int_{I_y}\int_{I_x} 1\, dx\, dy\right)^{\frac{1}{2}} \\
&\leq C\frac{\sqrt{\|I_x\|\,\|I_y\|}(\|I_x\| + \|I_y\|)}{\text{dist}^2(I_x,I_y)} \\
&\leq C\left(\frac{\|I\|}{r}\right)^2 \qquad (6)
\end{aligned}$$

where $I$ denotes the maximum of $\|I_x\|$ and $\|I_y\|$ and $r = \text{dist}(I_x, I_y)$ measures the closest approach between $I_x$ and $I_y$ as sets. For 3D radiosity between surfaces $A_{\vec{x}}$ and $A_{\vec{y}}$ we have the bound

$$\|\hat{G}_A - G_A\| \leq C\left(\frac{\|I\|}{r}\right)^3$$

In this case $I$ denotes the maximal side length of $A_{\vec{x}}$ and $A_{\vec{y}}$, the two surfaces in question, while $r = \text{dist}(A_{\vec{x}}, A_{\vec{y}})$ still gives the closest approach between the two surfaces. Note that this is a stronger bound than the one given in the original HR work, $\left(\frac{\|I\|}{r}\right)^2$.

The bound given above is small whenever the ratio of size to distance is small. Clearly $I$ must be smaller than $r$, which gives rise to the well known notion of "well separatedness." Note that it is only useful when dist $> 0$. When the two intervals (areas) overlap, a more careful analysis must be applied. The difficulty arises because of the $\frac{1}{r}$ ($\frac{1}{r^2}$ respectively) nature of the radiosity kernel. In other words, the bound given above holds everywhere so long as we exclude an arbitrarily small region around the intersections of any of the line segments (surfaces). To deal with these remaining

regions we need the boundedness of our original operator. For this small region around the intersection set $\hat{G}_A = 0$ to get

$$\|\hat{G}_A - G_A\| = \|G_A\| = \|I_y\| F_{I_y, I_x}$$

(in 3D $\|A_{\vec{y}}\| F_{A_{\vec{y}}, A_{\vec{x}}}$). In practice we will likely not use $\hat{G}_A = 0$ at the singularity, but doing do is sufficient to establish our bound. The fundamental reason is the boundedness of the operator. Intuitively speaking, we can say that ignoring the region directly bordering on the singularity will introduce an error which is no larger than the "width" of this region times the "height" (i.e., a bound on the operator). Given that the height is finite we can make the product arbitrarily small by making the width small. More precisely we have the fact that the form factor $F \leq 1$ allowing us to force $\|\hat{G}_A - G_A\|$ below any desired threshold by making $\|I_y\|$ ($\|A_{\vec{y}}\|$ respectively) small enough. Recall that we said earlier that the recursion will eventually bottom out for any strictly positive error, although we may not have the resources to wait for this natural recursion termination. The above bounds are at the base of this claim.

Taking both bounds together we can make the distance between $G_A$ and its approximation arbitrarily small by making the ratio of size to distance small or, when we are at the singularity, by simply making the size itself small. The region over which we have to employ the second bound can be made arbitrarily small, and with it the bound itself. For sake of our argument we allocate $\frac{\epsilon}{2}$ of our total allowed error to the regions touching the singularity and continue to consider only the case of elements which are separated. Their error must now be kept below $\frac{\epsilon}{2}$, for a total of the given $\epsilon$. In practice there is a difficulty in evenly dividing the error budget. Since we do not know the exact form of all the constants involved in the error expressions, but only their functional form, it is hard to force an even distribution of the error. However, as the allowable error is reduced both the error at the singularity and everywhere else is forced towards zero. We have seen this phenomenon in practice and will discuss it in the context of an actual example in the section on the performance of WR (2.5.1).

Given that we have a remaining error budget of $\frac{\epsilon}{2}$ we need to show that for this allowable error any recursive call will create at most a constant number of calls to the function `Quadrature`. From the above error bound we see that an interaction will be

created whenever the size to distance ratio is smaller than some threshold. How many elements can there be for which this is true? To answer this question we interpret the size to distance ratio geometrically as a measure of angle subtended. In other words, this ratio is proportional to the angle that one interval (surface) subtends from the point of view of the other interval (surface)[6].

On the initial call to `ProjectKernel` for element $i$ there can at most be $k$ elements (the original input surfaces) less than this threshold (hence the $k^2$ in the overall performance analysis). Suppose that some of those initial input surfaces are too large relative to their distance from element $i$ (i.e., their angle subtended is above the threshold). These surfaces will result in recursive calls. How many can there be?

We first observe that the total angle subtended above a given point on an element is bounded ($2\pi$). This implies that there can be at most a constant number of elements larger than our threshold at any stage of the recursion (only a constant number of elements of a given size fit into the hemisphere above a point). Elements whose angle subtended is smaller than the threshold have already been allowed to interact, elements whose angle subtended is still too large will be forwarded recursively. Abstractly we can say that only elements within a local neighborhood will be allowed to interact. Those that are further away have already been allowed to interact at a higher level in the recursion, those that are closer are forwarded. Since we are considering a *ratio* of size to distance the distance can shrink as the size shrinks (due to subdivision), maintaining a constant sized neighborhood over which interactions are generated, at every level of the recursion. Taken together with the fact that eventually the elements have gotten so small that the error must be below our threshold, and the recursion will bottom out, we have established our claim:

> Each element—below the top level call to `ProjectKernel`—interacts with at most a constant number of other elements. This means that the total number of interactions created due to recursive calls is proportional to the total number of elements, leading to $O(k^2 + n)$ interactions overall. The constant of proportionality is a function of the problem definition and

[6]The angle subtended by element $j$ can be different from different points of view on element $i$. We always assume the maximum over all points on $i$.

error requested, but not of the discretization itself.

### 2.3.4 Oracle

From the above arguments, we have seen that the function `Oracle` can be implemented by estimating the ratio of size to distance, or in the vicinity of the singularity, simply the size itself. In the case of radiosity with constant basis functions, measuring the ratio is particularly simple since it is given by the point to finite area form factor, a quantity for whose computation many formulas are known (see for example [58] or [45]). This was the oracle used in the original HR algorithm [32]. For higher order methods this simple form factor estimate is sufficient to argue the asymptotic bound on the number of couplings created, but it does not take full advantage of the information present. We will see later that in the case of higher order basis functions within a hierarchical framework we will find the bound on our error to be given by a higher power of the ratio of sizes to distance, implying that the falloff is much more dramatic.

In either case there are other, more direct methods, to estimate the quantity $\|\hat{G}_A - G_A\|$ discussed in the next section.

### 2.3.5 Higher Orders

Consider again the arguments used above to show that HR constructs only a linear number of interactions. There was nothing particular in the argument which ties it to constant basis functions. Suppose we wish to employ a Galerkin scheme with higher order basis functions. In this case each interaction between two elements entails a number of quadratures. For constant basis functions there was simply one coefficient $G_{ij}$ for elements `i` and `j`. We will continue to use the indexing $G_{ij}$, but think of the quantity $G_{ij}$ as consisting of an array of numbers describing all the possible coupling terms over the given elements due to higher order basis functions. For example, in the case of piecewise linear basis functions we have two basis functions along each dimension. In flatland $G_{ij}$ now consists of $2 \times 2$ couplings and in 3D $G_{ij}$ has $2^2 \times 2^2$ numbers associated with it. If $M - 1$ is the order of basis functions used we will

abstract $M \times M$ (flatland) and $M^2 \times M^2$ (3D) couplings respectively into $G_{ij}$.

The basic reasoning of the recursion count argument still holds. $\|\hat{G}_A - G_A\|$ is still the quantity which needs to be kept below some $\delta(\epsilon)$, however $\hat{G}_A$ is not constant anymore. The form factor argument to measure angle subtended does not take full advantage of the power of higher order basis functions. However, it is still sufficient to argue the asymptotic bound. In practice we will of course want to take advantage of the higher order nature of the basis functions. One way to do this is to have the function `Oracle` use an estimate of the $G_{ij}$ to construct a polynomial and measure how well this polynomial interpolates the real kernel $G_A$. This type of oracle was employed in the case of WR [28, 54] and estimates the quantity $\|\hat{G}_A - G_A\|$ directly by computing the implied integral

$$\|\hat{G}_A - G_A\| = \int_{I_y} \int_{I_x} \|G_A(x_0, y_0) - G_A(x, y)\| \, dx \, dy$$

For the implementation in [28, 54] the above integral was computed with a Gaussian quadrature rule. See Appendix B for the implementation details.

### 2.3.6 Iterative Solvers

As pointed out at earlier there are two parts to a complete algorithm, setting up the equations, and solving them. Above, we described how to set up the equations and argued why there are $O(k^2 + n)$ interactions total for any given finite accuracy requirement. To complete the algorithm we need the iteration function. This function corresponds to the matrix/vector multiply in an iterative solver. In HR this was referred to as `Gather`, a function which moves radiosity from element `j` across $G_{ij}$ to element `i`, multiplying it with the factor $G_{ij}$ (the form factor for constant basis functions). Once this has occurred we still need a function referred to as `PushPull` in [32].

For each input surface (element) `i`, `ProjectKernel` is called with all other input surfaces (elements) `j`. As pointed out above, the choice of interactions $G_{ij}$ actually created corresponds to an implicit choice of basis functions. Consequently when `ProjectKernel` was called on, say `i` and $\mathtt{j}_0$, versus `i` and $\mathtt{j}_1$, different basis functions

may have been constructed on i for those two calls. Put differently, irradiance at a surface will be computed at different levels of the hierarchy, due to different sources. These incoming irradiances need to be consolidated.

Consider the function PushPull as proposed in Hanrahan *et al.*[32]. Irradiance of a parent in the subdivision hierarchy is added to the children on a downward pass, while on an upward pass the radiosity at a parent is the area average of the radiosity at the children

```
PushPull( Element i )
    if( !Leaf( i ) )
        ForAllChildren( i.c )
            i.c.E = Sum( i.c.E, i.E ); //Push
            PushPull( i.c );
        i.B = Average( i.children.B ); //Pull
    else
        i.B = i.Be + i.rho * i.E;
```

where we used the symbols B to denote radiosity, E to denote irradiance, Be for the emitted part of radiosity, and rho for the reflectivity of the given element.

The summing of irradiance on the way down follows immediately from the physical meaning of irradiance. The irradiance at a given element is the sum of all the irradiances received at the element itself and all its ancestor elements. The area averaging on the way up follows directly from the definition of constant radiosity, which is a density quantity per area.

How to extend this PushPull reasoning to the higher order hierarchical algorithm briefly outlined above is not immediately clear. In fact this is where wavelets come in since they not only generalize the notion of higher order hierarchical basis sets, but also the attendant notions of pushing and pulling throughout such a hierarchy. However, before describing this connection with wavelets and giving an exact prescription of the transfer of power from parents to children and vice versa, we give an intuitive description here.

### 2.3.7 `PushPull` for Higher Orders

Suppose we use some higher order polynomial basis set and a given subdivision hierarchy has received irradiance at different levels of the hierarchy. Consider an element and one of its children. The parent has some set of coefficients $E^{parent}$ describing the irradiance with respect to the basis set over that element. Similarly the child has some set of coefficients $E^{child}$. We cannot simply add the coefficients because they are with respect to different functions, wider ones on the parent, and smaller ones on the child. For constant functions this did not matter, since constant functions "look the same" at any level of the hierarchy. However, if we have some prescription which tells us how to transform the basis functions on the parent level into some set of basis functions on the child level we could perform the corresponding transformation on the coefficients of $E^{parent}$. The resulting coefficients would be expressed with respect to the same functions as the $E^{child}$, and could now be added coefficientwise. This is the generalized function `Push`. Similarly on the way up we must apply *some* averaging operation, since the children have a finer resolution than the parent. Suppose we can compute a linear combination of basis functions on the parent which is in some sense the closest match to a child basis function. Once again we could apply the corresponding transformation on the $B^{child}$ to get a set of coefficients at the parent level. These can now be assigned to $B^{parent}$.

Wavelets with their two-scale relation fulfill exactly these requirements in a simple and efficient way.

## 2.4 Wavelets and Operators

Earlier we used geometrical reasoning to justify the claims of the original HR algorithm. The arguments were also used to justify higher order basis functions in an extended HR algorithm. These arguments find a much more general framework in the context of wavelets. In this section we first review some simple facts about wavelets which are needed to apply wavelets to our problem, we then show how wavelets can be used to justify a much more general approach, opening up many new possibilities

for efficient radiosity algorithms.

The basic roadmap is as follows:

1. Wavelets can be used to construct hierarchical basis function systems which have fast ($O(n)$) basis change algorithms associated with them

2. Using these bases, a large class of integral equations (operators) can be realized with sparse matrices if only finite precision answers are desired.

### 2.4.1 Wavelets

Wavelets form a rapidly evolving field of inquiry originating from several different scientific and engineering disciplines. Most notably perhaps are signal processing [44] and pure mathematics [20]. Consequently there are many possible approaches to motivate the subject. We will take the view of operator theory here since it most directly applies to the question of solving integral equations. Before going into the mathematical details we go through the simplest example of a wavelet basis. This is also the wavelet basis which corresponds to the HR algorithm.

#### An Example: The Haar basis

To provide the following definitions with some intuition we consider the simplest example of a wavelet construction, the so called Haar basis. Figure 2 shows a piecewise constant basis on the unit interval in the upper left hand corner. This basis consists of a single function and its translates, a hallmark of wavelet constructions. The *resolution* of this basis is $2^3 = 8$. There are $\log_2 8 = 3$ *levels*. Throughout, the first subscript on the $\phi$ and $\psi$ functions denotes the resolution level, while the second subscript describes the translation parameter. Now take these basis functions pairwise and add them or subtract them to get the new functions shown in the top middle. The adding, or more generally *averaging* process is denoted by $h$, while the *differencing* process is denoted by $g$. After reordering we arrive at the basis set on the top right. This is still a basis for the same space, though it uses different functions. The main feature to note is that we have another set of 4 functions $\phi_{2,k}$ which are the same

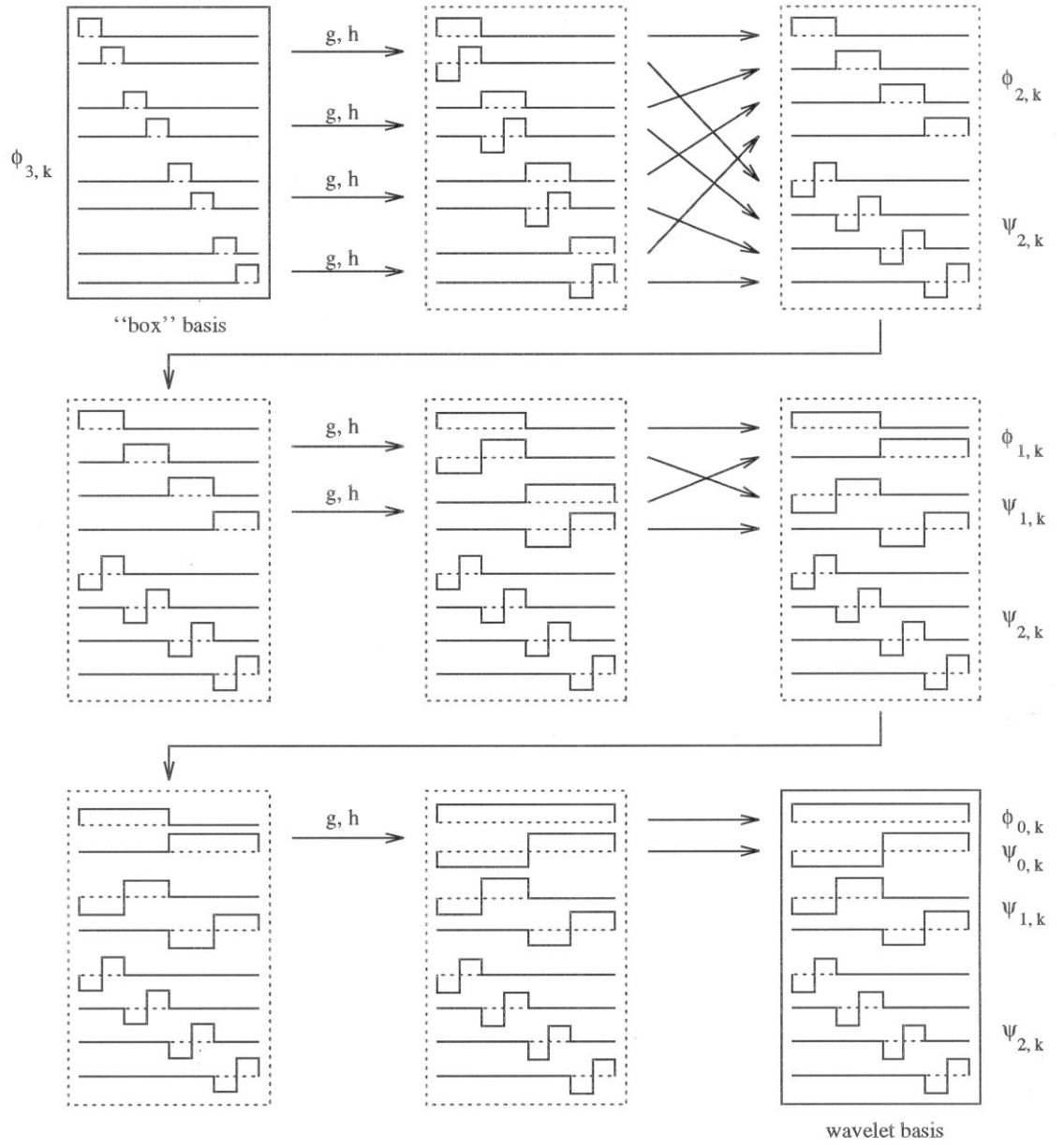



Figure 2: The Haar basis construction, taking a piecewise constant basis (8 box functions over the unit interval) into the Haar wavelet basis consisting of the overall average and detail functions at all lower levels. Note, the heights of the functions are not drawn to scale. (Adapted from [28].)

as our original set except wider (i.e., at lesser resolution). Consider only the 8 box functions we transformed from level 3 to the 4 box functions at level 2. The difference between these two resolution levels is spanned by the 4 new functions $\psi_{2,k}$. These latter are the wavelets proper. We say they span the *difference subspace* between the functions at resolution level 3 and 2. These functions too consist of only one shape and its translates.

Clearly we can now repeat the construction with the 4 box functions rewriting them as 2 wider box functions and 2 wider wavelets. After repeating one more time we arrive at the final basis set (bottom right). This basis set consists of one function which encodes an overall average (the box function $\phi_0$) and a sequence of finer and finer wavelet functions which encode for the successive levels of detail needed to describe all functions in the original space.

What we just described in this simple setting is the essence of any wavelet construction. In general, the functions as well as the operators $g$ and $h$ will be more complicated, but the intuition is exactly the same. There is always a sequence of resolution levels, and there are spaces which describe the differences between successive resolution levels. The operators $g$ and $h$ will generalize to high pass filters ($g$) and low pass filters ($h$). The individual spaces are always described by some basic shape (sometime a small set of basic shapes) and their translates and these shapes will remain the same throughout the hierarchy, except they become coarser and coarser.

After this example we proceed to describe the general construction.

### 2.4.2 Multi-Resolution Analysis

We begin by defining a multi-resolution analysis (MRA). A MRA is a ladder of spaces defined with respect to some underlying function space. In our case (radiosity) it suffices to consider the space of all finite energy functions $L^2(D)$ where $D$ is some domain of interest. Generally the domain is the whole real line R, but it can be higher dimensional or consist of only a finite subset, such as the interval $[0, 1]$. For radiosity we will need $D = [0, 1]$ for flatland and $D = [0, 1]^2$ for 3D. We begin with $D = \mathrm{R}$ which is the classical case.

A MRA of $L^2(\mathrm{R})$ is defined as a sequence of subspaces $V_i \subset L^2(\mathrm{R})$, $i \in \mathrm{Z}$ with the following properties

- The spaces are nested: $V_i \subset V_{i+1}$
- As they become finer they eventually cover the entire space: $\overline{\bigcup_{i\in\mathrm{Z}} V_i} = L^2(\mathrm{R})$
- As linear subspaces their only element in common is the zero element: $\bigcap_{i\in\mathrm{Z}} V_i = \{\mathbf{0}\}$
- The relationship between a resolution level and the next finer level is one of doubling frequency: $f(x) \in V_i \iff f(2x) \in V_{i+1}$
- $V_0$ is invariant under (unit) translation: $f(x) \in V_0 \iff f(x+1) \in V_0$
- $V_0$ has a structure such that there is a special function whose translates span $V_0$: $\exists \phi \in V_0 : \{\phi(x-k) | k \in \mathrm{Z}\}$ is a basis for $V_0$

Given these definitions, bases for all spaces can be built from the basis for $V_0$: $\{\phi_{i,k}(x) = \sqrt{2^i}\phi(2^i x - k) | k \in \mathrm{Z}\}$ is a basis of $V_i$ for all $i \in \mathrm{Z}$. The powers of $\sqrt{2}$ in front of the bases insure that all basis functions are normalized.

In a certain sense all the $V_i$ "look" the same except for their different resolution (see the Haar example in Figure 2). $\phi$ is often referred to as the *scaling* or *smooth* function. Since the $V_i$ have a containment relation and their union is dense in $L^2(\mathrm{R})$ we can approximate any function $f \in L^2(\mathrm{R})$ at various resolution levels $f_i$ and in the limit $f = \lim_{i\to\infty} f_i$. The $f_i$ can be found by projecting into the associated subspace.

Since $\phi \in V_0 \subset V_1$ there exists a sequence $h_k$ such that

$$\phi(x) = \sqrt{2}\sum_k h_k \phi(2x - k)$$

Or in words, $\phi$ can be built up from smaller versions of itself (see the Haar construction in Figure 2). This will be a crucial ingredient in navigating from one subspace to another.

Since $V_i \subset V_{i+1}$ we can define $W_i$ as the extra detail added when going from $V_i$ to the finer scale space $V_{i+1}$

$$V_{i+1} = V_i \oplus W_i$$

where $\oplus$ is used to denote the direct sum. In the Haar example (Figure 2 top right) $V_3$ for example could be written as $V_2$ plus the space spanned by the $\psi_{2,k}$.

Notice that this definition does not make $W_i$ unique. Under some very general conditions we will get a function $\psi \in W_0$, called the *wavelet* or *detail* function, whose translates span $W_0$. In this case we find that $\{\psi_{i,k}(x) = \sqrt{2^i}\psi(2^i x - k) | k \in \mathrm{Z}\}$ is a basis for the $W_i$ (see the $\psi$ functions in Figure 2). Since $\psi \in W_i \subset V_{i+1}$ there exists a sequence $g_k$ such that

$$\psi(x) = \sqrt{2}\sum_k g_k \phi(2x - k)$$

Or in words, $\psi$ (just as $\phi$ above) can be built up from smaller versions of $\phi$.

The sequences $(h_k, g_k)$ together are referred to as the *two scale relation*. Their usefulness arises from the fact that they describe the basis functions of $V_i$ and $W_i$ as linear combinations of the basis functions of $V_{i+1}$

$$\begin{aligned} \phi_{i,l} &= \sum_k h_{k-2l}\phi_{i+1,k} \\ \psi_{i,l} &= \sum_k g_{k-2l}\phi_{i+1,k} \end{aligned}$$

Because of this property we can use the sequences $(h_k, g_k)$ as a way of "navigating" between these spaces. If a function is expressed with respect to the basis in $V_{i+1}$ we have a straightforward way of finding its expression with respect to the bases of $V_i$ and $W_i$. $h_k$ can be thought of as a low pass filter, since it allows us to take a finer level approximation $f_{i+1}$ into a coarser (smoother) approximation $f_i$. $g_k$ is a corresponding high pass filter encoding the differences between the approximation $f_{i+1}$ and $f_i$. Note that after the convolution there is a subsampling step (expressed as the $2l$ in the subscript of $h$ and $g$).

So far the ladder of spaces was bi-infinite. If the construction is limited to a finite interval such as $[0, 1]$ there is a natural choice for the coarsest space $V_0$ (see Figure 2). Throughout we will always use constructions on the interval $[0, 1]$ (flatland) or $[0, 1]^2$ (3D) and the coarsest space will consist of some set of polynomials over the entire interval (square).

In the Haar example the sequences $g$ and $h$ are: $g_k = (-2^{-1/2}, 2^{-1/2})$ and $h_k =$

$(2^{-1/2}, 2^{-1/2})$ and the original space $V_3$ was decomposed as

$$V_3 = W_2 \oplus V_2 = W_2 \oplus W_1 \oplus V_1 = W_2 \oplus W_1 \oplus W_0 \oplus V_0$$

in the bottom right hand corner.

Suppose that the filter sequences are finite (either by construction of because of truncation for practical reasons), then the total work in going from the finest level $\phi$ basis to the wavelet basis is linear. We begin with $n$ smooth functions and transform them into $n/2$ wider smooth functions and $n/2$ detail functions. Continuing with the $n/2$ smooth functions we turn these into $n/4$ etc., for a total of $O(n)$ (again consider the example in Figure 2). This is one of the remarkable properties of wavelets since in general a basis change has an operations count of $O(n^2)$. In particular for the purposes of an asymptotically efficient algorithm for radiosity this $O(n)$ property will be crucial.

So far we only went from a canonical basis (the box basis in our example) into a wavelet basis but this process is also reversible with a similar pyramid structure. Since $\phi(2x-k) \in V_1 = V_0 \oplus W_0$ there exist sequences of coefficients such that $\phi(2x-k)$ is a linear combination of the $\phi(x-l)$ and $\psi(x-l)$. If the wavelets are orthonormal, as we assume for simplicity throughout, the sequences $g_k$ and $h_k$ serve this purpose as well

$$\sqrt{2}\phi(2x-k) = \sum_l h_{k-2l}\phi(x-l) + \sum_l g_{k-2l}\psi(x-l)$$

In the more general bi-orthogonal case there are related sequences $\tilde{h}$ and $\tilde{g}$ arising from the two scale relation of the dual bases. Details can be found in [54]. When the filter sequences are finite this transformation too will have cost linear in the total number of basis functions.

So far we have only talked about transforming the basis functions. All these transformations have their corresponding transformations which will be applied to the coefficients, which are the quantities we have to deal with in a program.

### 2.4.3 Transforming Coefficients

Given that we can transform one basis into another and back with these filter sequences we can transform coefficients of some function with respect to one basis into coefficients of that function with respect to the other basis (and back). Let $f_i$ be an approximation of some function $f$ at resolution level $i$ with coefficients $s_{i,k}$, $f_i = \sum_{k=0}^{2^i-1} s_{i,k}\phi_{i,k}$ then

$$\begin{aligned} s_{i-1,l} &= \sum_k h_{k-2l} s_{i,k} \\ d_{i-1,l} &= \sum_k g_{k-2l} s_{i,k} \end{aligned}$$

will give the coefficients of this function with respect to the space $W_{i-1}$ and $V_{i-1}$. Conversely given the coefficients of some function with respect to the space $W_i$ and $V_i$ we can find its coefficients with respect to $V_{i+1}$

$$s_{i+1,k} = \sum_l h_{k-2l} s_{i,l} + \sum_l g_{k-2l} d_{i,l}$$

These identities follow directly from substituting the definitions.

These two convolution equations are used extensively when manipulating functions at different resolutions. In particular they will be the building blocks of the generalized `PushPull` function in WR.

### 2.4.4 Locality and Smoothness

An important property of wavelets is the fast transform between a canonical basis and the wavelet basis (and back). Another important property that makes wavelets so useful, is their capability to localize phenomena well in both space and time. Compare this with the extremes of Fourier series, which localizes completely in time, but looses all localization in space, and Dirac delta distributions which localize completely in space but loose all localization in time. From a practical point of view the localization in both time and space implies, for example, that we only need fine detail descriptions in regions which exhibit high frequencies while in other regions, where the underlying function is very smooth, a coarse approximation will be sufficient.

Consider again the example of the Haar basis. Figure 2 shows the Haar basis on the bottom right. This set of functions can be interpreted intuitively as describing functions in the original space $V_3$ as realizable by writing them as a combination of 1 function which describes the overall average ($\phi_0$); 1 function which encodes for the difference on either half of the interval ($\psi_0$); 2 functions which encode the differences on quarters ($\psi_{1,k}$); and 4 functions which encode the differences on eights ($\psi_{2,k}$). The power of wavelets rests in the fact that for smooth functions some of the coefficients corresponding to these basis functions will be very small. Consider a function which is almost constant on the first and second eighth. In this case the coefficient associated with $\psi_{2,0}$ would be very small.

This last observation is the basis for lossy wavelet compression [44]. In such a compression scheme we transform an image into a wavelet basis. If the image has regions with little high frequency information then the associated wavelet coefficients will be small. If two neighboring pixels have almost equal value, then their difference—the coefficient of the associated Haar function—will be near zero. Using thresholding and quantization a significant compression can be achieved without a noticeable loss in image quality. Similarly we can apply a lossy compression scheme to our matrix system and achieve an answer which is very close to our desired answer, yet use only a few coefficients to do so. This idea is the essence of wavelet methods for the solution of integral equations and in particular WR.

In the Haar case we can only take advantage of neighboring pixels (or multi-pixel blocks at higher levels) having almost equal value. How about considering larger sequences of pixels and asking whether their relationship is almost linear or quadratic, etc.? As pointed out above, the Haar basis is only the simplest of all wavelet bases and the basic ideas generalize to higher order bases. The bases are capable of detecting higher order coherence amongst neighboring pixels (or entries in a matrix system) and exploit it. This ability arises from a property called *vanishing moments*. Different wavelets have more or less vanishing moments. In the next section we will show why this property is helpful in realizing sparse solution methods for the radiosity equation.

### 2.4.5 Vanishing Moments

We begin with the definition of vanishing moments. A function $\psi$ is said to have $M$ vanishing moments if its projection against the first $M$ monomials vanishes

$$\langle \psi, x^i \rangle = 0 = \int dx\, \psi(x) x^i \;\; i = 0, \ldots, M-1$$

The Haar wavelet for example has 1 vanishing moment. Other wavelets can be constructed to have more vanishing moments.

To see why this leads to small coefficients in general consider some function $f \in L^2$. Suppose we want to write it with respect to a wavelet basis. The coefficients of such an expansion can be found by taking inner products against the basis functions

$$f(x) = \sum_{i,j \in \mathbb{Z}} \langle \psi_{i,j}, f \rangle \psi_{i,j}$$

We want to show that for smooth $f$ many of the coefficients $f_{i,j} = \langle \psi_{i,j}, f \rangle$ are small. If $f$ is smooth we can apply Taylor's theorem to expand it about some point $x_0$ (for simplicity, let $x_0 = 0$) to get

$$f(x) = \sum_{i=0}^{M-1} \frac{f^{(i)}(0)}{i!} x^i + \frac{f^{(M)}(\xi)}{M!} x^M$$

for some $\xi \in [0, x]$. Now consider computing $f_{i,j}$. To simplify the argument we consider the inner product necessary to compute $f_{0,0}$ (i.e., the inner product with $\psi$) all others being related by translations and scalings. Using the vanishing moment property we can bound the resulting coefficient as follows

$$\begin{aligned} \left| \int dx\, f(x)\psi(x) \right| &= \left| \int dx\, \frac{f^{(M)}(\xi)}{M!} x^M \psi(x) \right| \\ &\leq \sup_{\xi \in I_x} \left| \frac{f^{(M)}(\xi)}{M!} \right| \|I_x\|^M \int dx\, |\psi(x)| \end{aligned} \tag{7}$$

where $I_x$ is the interval of support of $\psi$. From this bound we can see that the associated coefficient will be small whenever either $\|I_x\|$ is small or the $M^{th}$ derivative of $f$ is small. Similar arguments can be made for functions of more than one variable. Notice the similarity with our earlier arguments bounding the error in HR. There we

forced the error—which will become the detail coefficient in wavelet radiosity—below a ratio of size to distance. Here we have size again, but this time multiplied with the derivative. Below we will argue that for integral equations whose kernel function has derivatives which fall off as an inverse power of distance many detail coefficients will be small. In that way the correspondence can be made with the size to distance argument.

This bound allows us to argue that many of the entries in a matrix system arising from an integral operator will be very small and can be ignored, leading to a sparse matrix system. Recall that integral operators led to linear systems whose coefficients are integrals of the kernel function against the chosen basis functions. In the case of radiosity this lead to the $G_{ij}$ (Equation 4). Suppose that the basis functions for the integral operator are chosen to be wavelets and that these wavelets have vanishing moments. If $G$ is smooth then many of the $G_{ij}$ will be quite small because of the vanishing moment property, and can be ignored without incurring too much error. Below we will make this argument mathematically precise.

For now we note that these arguments are in some sense the converse of the our earlier arguments, which showed that only a limited number of interactions need to be considered in a hierarchical radiosity algorithm. There we argued about the coupling coefficients which *cannot* be ignored. The wavelet arguments will describe the coupling coefficients which *can* be ignored. In particular the projections onto wavelet (detail) functions will measure the error incurred by higher order hierarchical algorithms and embed the arguments we made earlier—using angle subtended—into a much more general context.

With these preparations we now go into the details of applying wavelets to the numerical solution of integral equations.

### 2.4.6 Wavelets and Integral Equations

Since wavelets can be used as bases for function spaces it makes sense to consider them in the context of a Galerkin method to solve an integral equation. For all the arguments that follow, we will use the example of flatland radiosity again. We will

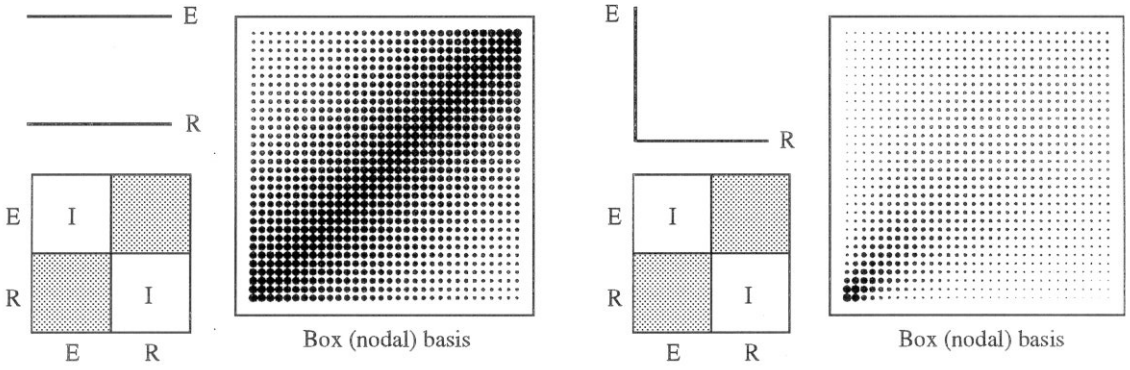



Figure 3: Two simple environments in flatland, two parallel line segments (left), and two perpendicular line segments (right), and the resulting matrix of form factors assuming a discretization of the line segments into 32 elements each and a constant basis function over each element. (Adapted from [54].)

also simplify the arguments by considering only orthonormal wavelet bases (the case of general bi-orthogonal bases is explained in [54]).

Recall that a Galerkin method applies projection operators to the integral equation at hand: Choose a subspace, express all functions in this subspace (i.e., project them into this subspace) and then solve the resulting linear system relating all the unknown coefficients to each other. We will now study the projection operators involved in this computation more carefully. In particular we will use the decomposition of subspaces implied by a MRA to the subspace chosen by a Galerkin method. This will result in a decomposition of the operator into actions on various constituent subspaces. The power of applying such an analysis lies in its ability to discover that many of these constituent subspaces are very sparse when using wavelets for smooth integral operators. We begin with the so called standard decomposition or realization and proceed to the more powerful non-standard realization of the operator.

We begin by defining projection operators for the resolution spaces $V_i$ of a MRA

$$P_i = \sum_{k=0}^{2^i-1} \langle \, . \, , \phi_{i,k} \rangle \, \phi_{i,k}$$

and for the associated detail spaces $W_i$

$$Q_i = P_{i+1} - P_i = \sum_{k=0}^{2^i-1} \langle \, . \, , \psi_{i,k} \rangle \, \psi_{i,k}$$

Given the radiosity integral equation in operator form

$$(I - \mathcal{S} \circ \mathcal{G})B = B^e$$

and some finest resolution space $V_L$, the Galerkin method gives

$$(I - P_L(\mathcal{S} \circ \mathcal{G})P_L)B = P_L B^e$$

Recall that we assumed throughout that the reflectance $\rho$ would be constant over each basis function support. Consequently we will now focus exclusively on the realization of $P_L \mathcal{G} P_L$ in wavelet bases.

CR for example uses as $V_L$ the space of all piecewise constant functions over the elements at some finest resolution level $L$. Two examples of this for flatland radiosity are given in Figure 3. On the left is the flatland environment of two parallel line segments. The resulting matrix of form factors (piecewise constant basis functions) has a block diagonal form. The diagonal blocks are identity matrices while one of the off diagonal blocks is shown enlarged. The size of dots is proportional to the magnitude of the form factor ($G_{ij}$ in our notation). Similarly on the right we see the resulting matrix for an environment with two line segments meeting in a corner, for which the domain contains the singularity. The most notable property is the extreme smoothness or coherence of the resulting matrices.

As we saw earlier there are alternative ways of writing $V_L$ using wavelets. Writing $V_L = V_0 + \sum_{i=0}^{L-1} W_i$ corresponds to writing the projection operator as $P_L = P_0 + \sum_{i=0}^{L-1} Q_i$. Using this identity we have

$$\begin{aligned} P_L \mathcal{G} P_L &= (P_0 + \sum_{i=0}^{L-1} Q_i)\mathcal{G}(P_0 + \sum_{i=0}^{L-1} Q_i) \\ &= P_0 \mathcal{G} P_0 + \sum_{i=0}^{L-1} P_0 \mathcal{G} Q_i + \sum_{i=0}^{L-1} Q_i \mathcal{G} P_0 + \sum_{i,l=0}^{L-1} Q_i \mathcal{G} Q_l \end{aligned} \tag{8}$$

This decomposition corresponds to a particular two dimensional basis construction. Given a one dimensional wavelet basis $\{\phi_0, \psi_{i,k}\}$, $i = 0, \ldots, L-1$, $k = 0, \ldots, 2^i - 1$ we can build a two dimensional basis via a tensor product construction $\{\phi_0, \psi_{i,k}\} \times \{\phi_0, \psi_{l,m}\}$, $i, l = 0, \ldots, L-1$, $k = 0, \ldots, 2^i - 1$, and $m = 0, \ldots, 2^l - 1$. This is often referred to as the standard realization of the integral operator [9].

The pyramid algorithms that were mentioned above for transforming a function of a single variable between a basis of $V_L$ and the bases in $V_0 + \sum_{i=0}^{L-1} W_i$ can be applied to matrices (functions of two variables). In particular the standard decomposition corresponds to applying such a pyramid transform to all rows (transforming the right hand side $P_L$) followed by a transform of all row transformed columns. This is nothing more than a vector space basis change as is often done with matrices for various reasons. The remarkable property of the change to the wavelet basis is that it can be performed in time proportional to the number of basis functions, $O(n^2)$. In general expressing a matrix of size $O(n^2)$ with respect to another basis entails a transform of cost $O(n^3)$.

Figure 4 shows the effects of transforming form factor matrices expressed originally in the piecewise constant nodal basis (see Figure 3) into different wavelet bases (the standard form in this case). On the left the Haar basis was used, while on the right the Flatlet basis with two vanishing moments [28] was used. The top row gives matrices for the example of two parallel line segments, while the bottom row shows the case of two perpendicular line segments. Notice how many of the coefficients are small in magnitude (small disks). As the number of vanishing moments increases from one to two (left to right) we can observe many more entries becoming small. This demonstrates for two particular cases how more vanishing moments lead to more (approximate) sparsity in the matrices.

At this point we have seen how the linear system associated with a linear operator, such as an integral equation, can be transformed into a wavelet basis. This can be done with the wavelet pyramid transform in time proportional to the number of basis functions. The resulting matrices will have many entries of negligible magnitude due to the vanishing moment properties of wavelets, if the kernel function of the original integral operator satisfies some smoothness conditions. In the next section we will see how many of these entries are small enough to be ignored leading to sparse solution algorithms for a large class of operators.

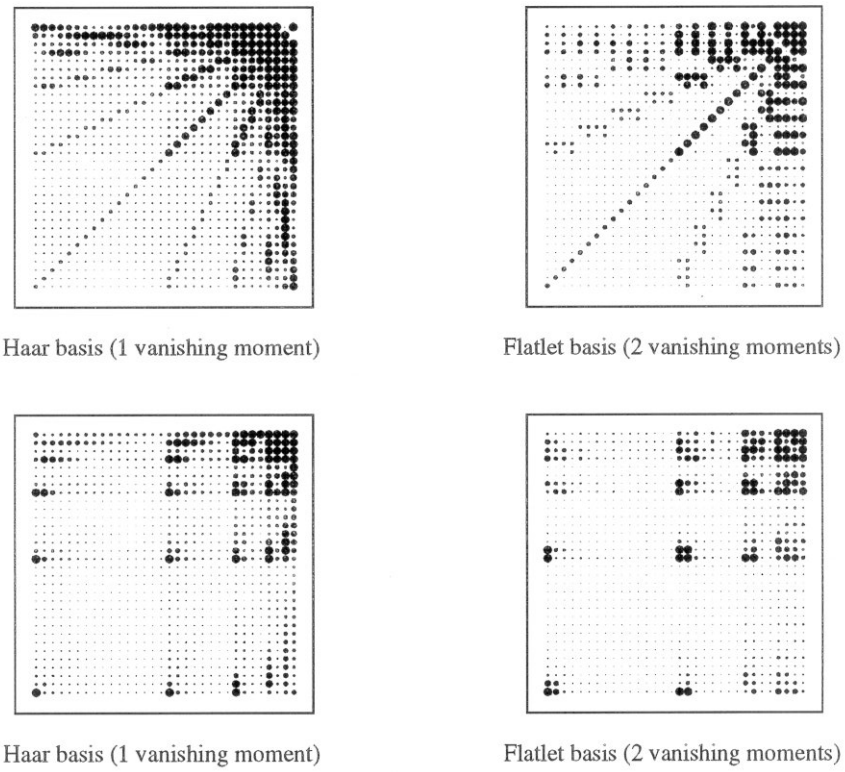



Figure 4: Form factor matrices for two flatland environments (see Figure 3) expressed in wavelet bases (standard form). The top row shows the form factor matrix for two parallel line segments expressed in the Haar basis (top left) and in the $F_2$ basis [28] (top right), which has 2 vanishing moments but remains piecewise constant at the finest level. The bottom row shows the same bases applied to the form factor matrix for two perpendicular line segments. (Adapted from [54].)

### 2.4.7 Calderon-Zygmund Operators

In a seminal paper published in 1991 Beylkin *et al.*[9] showed that for a large class of linear operators the resulting linear system, when using wavelet bases, is approximately sparse. More specifically they showed that for a class of integral operators known as Calderon-Zygmund and any $\epsilon > 0$ a $\delta(\epsilon)$ exists such that all but $O(n \log n)$ of the matrix entries will be below $\delta$ and can be ignored without incurring more than an error of $\epsilon$ in the computed answer.

An integral operator $\mathcal{K}(f) = \int K(.,y)f(y)\,dy$ is called a Calderon-Zygmund operator it its kernel function $K : \mathrm{R}^n \times \mathrm{R}^n \rightarrow \mathrm{R}$ satisfies the following smoothness criteria

$$\begin{aligned} \|K(x,y)\| &\leq \frac{1}{\|x-y\|^n} \\ \|\partial_x^M K\| + \|\partial_y^M K\| &\leq \frac{C_M}{\|x-y\|^{n+M}} \end{aligned}$$

for some $M > 0$. In this classical definition the arguments $x$ and $y$ are variables of the parameter domain. In the case of radiosity we have a kernel of the form

$$\begin{aligned} \|G_A(x,y)\| &\leq \frac{1}{\|p(x)-q(y)\|^n} \\ \|\partial_x^M G_A\| + \|\partial_y^M G_A\| &\leq \frac{C_M}{\|p(x)-q(y)\|^{n+M}} \end{aligned}$$

where $p, q : [0,1]^n \rightarrow \mathrm{R}^{n+1}$ are the parameterizations of the surfaces in question (here $n = 1$ for flatland and $n = 2$ for 3D radiosity), and $C$ and $C_M$ are only finite for parameterization for which the magnitude of $\frac{dI_x}{dx}$ ($\frac{dA_x}{dx}$) is finite. The latter condition holds for parameterization which are non-singular. Typically $p$ and $q$ are polynomials. For example, for quadrilaterals they are bi-linear. Similarly for bi-cubic patches they are bi-cubic. The argument for the classical Calderon-Zygmund case as presented by Beylkin *et al.*[9] carries over with little modification to what we might consider the generalized Calderon-Zygmund operator of radiosity.

Recall the definition of the matrix entries $G_{ij}$ in a Galerkin method

$$G_{ij} = \int_{\mathrm{supp}\, N_i} \int_{\mathrm{supp}\, N_j} G_A(x,y)\, N_j(x) N_i(y)\, dx\, dy$$

Suppose now that we use wavelets with $M$ vanishing moments for our $\{N_i\}$. If we use Taylor's theorem ([22] Theorem 8.14.3) of order $M$ for $G_A$, abbreviating for convenience

$$G' = \sup_{\xi \in [0,1]} \frac{d^{(M)}}{d\xi} G_A(x_0 + \xi t_x, y + \xi t_y)$$

we find that the first $M$ terms in the expansion will not contribute to $G_{ij}$ leaving

$$\begin{aligned}
\|G_{ij}\| &= \left( \int_{\text{supp } N_i} \int_{\text{supp } N_j} |G_A(x,y)\, N_j(x) N_i(y)|^2 \, dx\, dy \right)^{\frac{1}{2}} \\
&\leq \left( \int_{\text{supp } N_i} \int_{\text{supp } N_j} \left| \frac{G'}{M!} \|(t_x, t_y)\|^M \, N_j(x) N_i(y) \right|^2 \, dx\, dy \right)^{\frac{1}{2}} \\
&\leq \frac{C_M}{\|p(x) - q(y)\|^{n+M}} \left( \int_{\text{supp } N_i} \int_{\text{supp } N_j} \|(t_x, t_y)\|^{2M} \, |N_j(x) N_i(y)|^2 \, dx\, dy \right)^{\frac{1}{2}} \\
&\leq \frac{C_M}{\|p(x) - q(y)\|^{n+M}} \|I\|^M \|I\|^n \left( \int_{\text{supp } N_i} \int_{\text{supp } N_j} |N_j(x) N_i(y)|^2 \, dx\, dy \right)^{\frac{1}{2}} \\
&\leq C_M \left( \frac{\|I\|}{r} \right)^{n+M} \qquad (9)
\end{aligned}$$

where: $r = \|p(x) - q(y)\|$; $n$ denotes the dimensionality of the domain; $\|I\|$ is defined to be the maximum sidelength of any of the involved supports; $\|(t_x, t_y)\|$ is bounded by $C\|I\|$ with the help of the triangle inequality (i.e., the diameter of a cube is no larger than some constant times the longest of its sides); to achieve our bound. The last expression is identical to the classical Calderon-Zygmund claim of Beylkin *et al.*[9] save for replacing $r$ by $\|x - y\|$. This proves that

> Radiosity satisfies a Calderon-Zygmund type inequality making the claims of Beylkin *et al.*[9] applicable to the radiosity integral equation.

Using this result we can assert that the radiosity matrix system arising from a Galerkin method using a wavelet basis leads to a sparse system with only $O(n \log n)$ important entries. Examining the matrices in Figure 4 we can immediately see the $O(n \log n)$ structure. There are approximately $\log n$ bands visible, each of length approximately equal to $n$. This is particularly noticeable in the case of two parallel lines and the Haar basis (upper left in Figure 4).

The fact that there is an extra factor of $\log n$ may seem surprising since we already know from our geometric arguments that in the case of radiosity only $O(n)$ couplings (entries in the matrix) are important. Considering the meaning of a row in the wavelet transformed matrix (see Figure 4) though we can see the $\log n$ dependence. For every basis function on one primitive there are important entries across the entire row (i.e., at all levels of resolution of the other primitive). Since there are $\log n$ levels it is not surprising that we have the extra $\log n$ factor. This statement is entirely compatible with the bound given above for (generalize) Calderon-Zygmund operators. The bound involves the *largest* of the dimensions of support for a pair of basis functions. Across a row (or column) of the matrix one of the dimensions is kept constant while the other gets successively halved. In the case of HR this would correspond to a basis function system which required every function on one primitive to couple with basis functions *at all levels* on the other primitive. Our insight that $O(n)$ entries are enough was also achieved by Beylkin *at al.*[9]. They proceeded to analyze the $\log n$ dependence in the number of non-negligible entries in the matrix and showed that by decoupling all the scales (i.e., reducing all dimensions in the subdivision equally) it is in fact possible to reduce the number of needed entries to $O(n)$. We will describe this so called non-standard construction next. This is the final missing piece to connect wavelets to HR and the resulting WR algorithm.

### 2.4.8 Non-Standard Operator Realization

We saw earlier how the decomposition $P_L = P_0 + \sum_{i=0}^{L-1} Q_i$ applied to $P_L \mathcal{G} P_L$ on both sides resulted in a realization of $\mathcal{G}$ in the wavelet basis. The resulting sum consisted of terms involving all possible combinations of subspaces $\{P_0, Q_i\}_{i=0,\ldots,L-1}$ on either side of $\mathcal{G}$. Said differently, the operator was expressed as a sum of contributions between subspaces at *all* resolutions. To remove this coupling across *all* levels[7] we use a telescoping sum argument to write

$$P_L \mathcal{G} P_L = P_0 \mathcal{G} P_0 + \sum_{i=0}^{L-1} (P_{i+1} \mathcal{G} P_{i+1} - P_i \mathcal{G} P_i)$$

[7] As distinct from couplings across a *constant* number of levels.

$$= P_0\mathcal{G}P_0 + \sum_{i=0}^{L-1} Q_i\mathcal{G}P_i + \sum_{i=0}^{L-1} P_i\mathcal{G}Q_i + \sum_{i=0}^{L-1} Q_i\mathcal{G}Q_i \quad (10)$$

using the fact that $P_{i+1} = P_i + Q_i$ and rewriting each summand in turn as

$$\begin{aligned} P_{i+1}\mathcal{G}P_{i+1} - P_i\mathcal{G}P_i &= (P_i + Q_i)\mathcal{G}(P_i + Q_i) - P_i\mathcal{G}P_i \\ &= P_i\mathcal{G}Q_i + Q_i\mathcal{G}P_i + Q_i\mathcal{G}Q_i \end{aligned}$$

With this algebraic "trick" we have managed to write the $P_L\mathcal{G}P_L$ as a sum involving summands, each of which carries the same index $i$ on the left *and* right. This implies that each one of the subspaces interacts with only one subspace (the one with the same index). Compare this to the standard decomposition (Equation 8) whose summands carry different indices on the left and right sides, resulting in each space interacting with a logarithmic number of other spaces. The decoupling of spaces results in a representation of the operator which is referred to as the non-standard realization. The standard realization uses a cross product basis, expressing the fact that the operator *goes from one space to another.* Therefore a standard realization always looks like a cross product basis. The non-standard realization involves terms which cannot be written as a cross product basis. As a set of functions they are a basis for a two parameter function space, but that is a different object from a basis which realizes an operator. If in fact we want to write it as a matrix product we have to use an *over representation* in both the domain and the range of the operator. To see why, consider the summands that occur. We find that for each $i$ both $P_i$ *and* $Q_i$ occur on both the left and right side for some summand. This is the over representation since $P_L = P_0 + \sum_{i=0}^{L-1} Q_i$ and *not* $P_L = P_0 + \sum_{i=0}^{L-1} Q_i + \sum_{i=0}^{L-1} P_i$. However, the total number of *couplings* that occur is still only $n^2$, reflecting the fact they they *are* a basis for a two parameter function space. However, they cannot be written as a cross product of one dimensional bases. For this reason the set of functions, $\{\phi_0(x)\phi_0(y), \phi_{i,m}(x)\psi_{i,j}(y), \psi_{i,m}(x)\phi_{i,j}(y), \psi_{i,m}(x)\psi_{i,j}(y)\}$, $i = 0, \ldots, L-1$, and $j, m = 0, \ldots, 2^i - 1$, is also referred to as the non-standard basis.

Figure 5 shows the non-standard realizations of the operators for the two flatland environments considered earlier (Figure 3). Each level consists of three blocks. The sets of triples consist of the $Q_i\mathcal{G}Q_i$ block in the lower left, the $P_i\mathcal{G}Q_i$ block in the

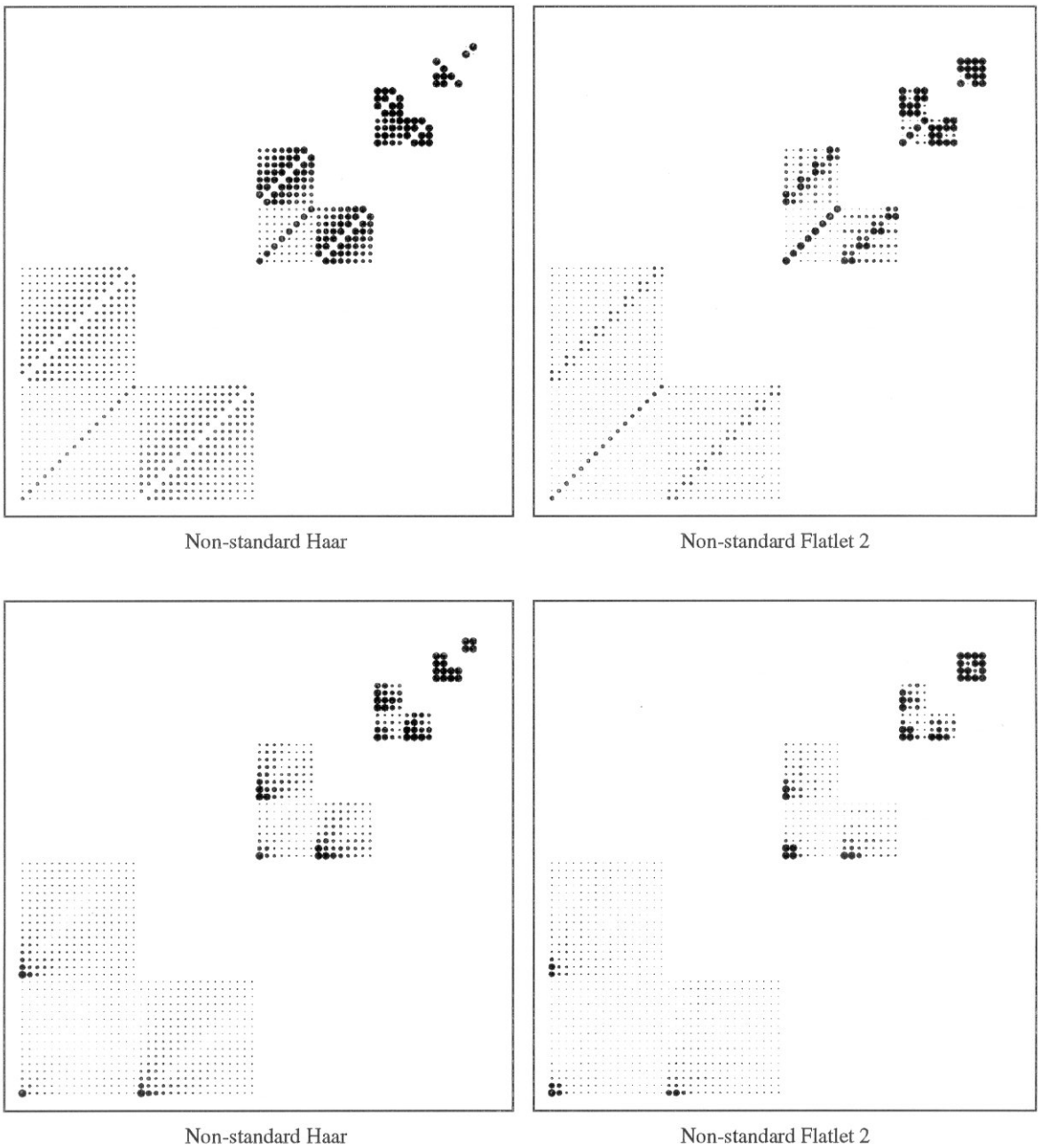



Figure 5: Form factor matrices for two flatland environments (see Figure 3) expressed in wavelet bases using the non-standard operator realization. The top row shows the form factor matrix for two parallel line segments expressed in the Haar basis (top left) and in the $\mathcal{F}_\in$ basis [28] (top right). The bottom row shows the same bases applied to the form factor matrix for two perpendicular line segments. (Adapted from [54].)

upper left and the $Q_i\mathcal{G}P_i$ block in the lower right. The empty quadrant would have corresponded to $P_i\mathcal{G}P_i$, however this is the block that the recursion occurs on. This last observation also suggests how to transform a matrix from the nodal basis into the non-standard realization. Instead of performing complete pyramid transforms on each row, followed by complete transforms on each column, the non-standard realization can be achieved by interleaving the individual transforms. First all rows are split into high pass and low pass bands (a single level application of the two scale relation), then all columns are subjected to a single level transform. Now recurse on the low pass/low pass quadrant ($P_i\mathcal{G}P_i$). When writing this out as a matrix suitable for matrix/vector multiplies the matrices in Figure 5 result.

### 2.4.9 Properties of the Non-Standard Realization

#### $O(n)$ Sparsity

The proof that only $O(n)$ entries in the resulting system are above some threshold $\delta(\epsilon)$ is exactly analogous to our earlier arguments counting the number of interactions in HR. The operator realization in HR is the non-standard type. This follows from the fact that a given element $i$ may interact directly with some other element $j_0$ while requiring subdivision when interacting with some element $j_1$. In those two cases element $i$ is represented with both a single function and with a set of finer functions, creating the over representation which is characteristic for the non-standard operator realization.

The general proof of the $O(n)$ sparsity property for general Calderon-Zygmund operators is *identical* to our earlier geometric arguments. We argued that for a given allowable error of $\epsilon$ we can permit some amount of error ($\delta$) across each link and that there would only be a linear number of such links. Smooth functions (piecewise polynomial) were used as basis functions. Saying that there is an error of $\delta$ for one such approximation is equivalent to saying that the associated detail coefficient is less than $\delta$. Hence we can ignore it. Recall that the detail coefficient measures the difference between one level of approximation and the next finer level (recursive splitting) of approximation. To say that a detail coefficient is small (and can be ignored) then

is equivalent to saying that no further subdivision is necessary, our assertion in the recursive enumeration scheme. In contrast, the function `ProjectKernel` induced further subdivision if the detail coefficient (the quantity effectively estimated by the function `Oracle`) was found to be too large. However, instead of adding the detail function in that region, it subdivided and chose to represent the region with finer smooth functions, creating the over representation of the non-standard realization.

While we used an "angle subtended" argument to bound the number of coefficients created in this fashion the (generalized) Calderon-Zygmund property is the abstract analog of this geometric statement. Recall the bound we gave earlier for the case of flatland radiosity and 3D radiosity

$$\|G_A - \hat{G}_A\| \leq C \left( \frac{\|I\|}{r} \right)^{n+1}$$

where $n$ is the dimensionality of the domain (1 for flatland and 2 for 3D radiosity). Examining the argument we gave above for a bound on the $G_{ij}$ for wavelets with $M$ vanishing moments, we get the bound

$$\|G_A - \hat{G}_A\| \leq C \left( \frac{\|I\|}{r} \right)^{n+M}$$

(recalling the definition of $\hat{G}_A$ from Equation 5). This bound on $\|G_A - \hat{G}_A\|$ is just a restatement of the fact that the next coefficient $G_{ij}$ in the expansion of $\hat{G}_A$ is small and has the bound given in Equation 9.

Using the Calderon-Zygmund bound in this way reduces the argument as to why there are only a linear number of important entries in the non-standard realization of the operator to the geometric argument given for HR. This time our statement is more general since it extends to wavelets with more than 1 vanishing moment and to a general class of operators. Even if we have an integral operator of Calderon-Zygmund type which does not arise from any kind of geometric problem, we can still impose a geometric interpretation on the various quantities. In effect we are supplying our abstract problem with a geometric structure. Having done so it is immediately obvious why there should only be a linear number of coefficients which are less than some size to distance ratio: angle subtended is a finite quantity even if the underlying

abstract space has no obvious relation to "geometry" as we know it in 3D. In this way the Calderon-Zygmund argument generalizes to other integral operators an idea that is perhaps more obvious in the geometrical context of graphics.

Creating this correspondence we have made one assumption, that the support of our basis functions is finite, but this is no different from the claims of Beylkin *et al.*[9]. In the case that the supports are finite, but neighboring basis functions overlap, the argument continues to hold. With a finite overlap the total angle subtended "coverage" grows at most by a constant factor (the overlap), but continues to stay finite, the crucial assumption.

### Elimination of Detail Functions

In the case of HR we only used smooth functions and never any detail functions, although the action of `Oracle` can properly be described as evaluating a detail coefficient. The abstract theory of integral operators has us use the smooth *and* detail functions to construct the sparse linear system. WR [28], or higher order HR methods, continue to use only the smooth functions. Where are the detail functions?

To understand the answer, consider again the Haar example. Suppose we are using the Haar basis for a non-standard realization of our operator (see Figure 5 left column). If we ignore all entries in the matrix less than some threshold we will be left with some set of entries corresponding to couplings between a mixture of smooth and detail functions. For example one such entry might be

$$G_{ij} = \langle\langle G_A, \phi_{k,j}\rangle, \psi_{k,i}\rangle$$

for some level $k$ and translation $i$ and $j$ respectively. For simplicity we will stay with the flatland case, the 3D case being only different by virtue of more basis functions and more indices being involved. Using the two scale relationship for $\psi$ we can rewrite the above as

$$\begin{aligned} G_{ij} &= \langle\langle G_A, \phi_{k,j}\rangle, \sqrt{2}^{-1}(\phi_{k+1,2i+1} - \phi_{k+1,2i})\rangle \\ &= \langle\langle G_A, \phi_{k,j}\rangle, \sqrt{2}^{-1}\phi_{k+1,2i+1}\rangle - \langle\langle G_A, \phi_{k,j}\rangle, \sqrt{2}^{-1}\phi_{k+1,2i}\rangle \end{aligned}$$

The last two summands are again coupling coefficients but this time they only involve smooth functions. In this way we can systematically replace all instances of couplings involving detail functions with ones involving smooth functions only. In the actual algorithm we directly use the smooth functions instead of first introducing detail functions only to replace them with finer level smooth functions.

Figure 6 gives a simple 1D example for the difference between the standard and non-standard representation as it replaces all detail functions with their smooth function equivalents (via the two scale relation). On the top we see an example of some function which we wish to approximate to within some fidelity with the Haar basis. This is done in the left column. We begin with the overall average ($V_0$) and invoke an oracle to determine which detail functions we should add. This process is symbolized by the sequence of finer and finer details (from spaces $W_0$ through $W_3$) being added until the sum of all the functions adds up to a close approximation of our original function (see bottom left). Instead we could proceed differently (as HR does). Again we begin with the overall smooth function on the top right ($V_0$ or level 0). This time the oracle determines where we should introduce a split. Note that a split is equivalent to adding a detail function with its center at the split. But instead of adding the detail we simply create a new level of our hierarchy. We now have a best approximation of our function at two levels of resolution, level 0 and level 1. From our previous study of the Haar system we know that the space spanned by level 1 is exactly the space spanned by $V_0 + W_0$ as we did on the left. Continuing with further splits, each corresponding to the addition of a detail function at the split site we arrive at the same final representation.

The HR algorithms actually perform the refinement as shown on the right. We can also see how this is equivalent to adding detail functions (left column) of the proper resolution at the appropriate places. Nonetheless there is a fundamental difference. Imagine that another function in our system wants to interact across the integral operator with the subsection of our approximation numbered 5, which we will call $f_5$ for the sake of this argument. Call the other function $g$. In order to have $f_5$ interact with $g$ we need to link up $g$ with all the basis functions which contribute to the final height of section 5. In the left column this includes all functions at all levels which

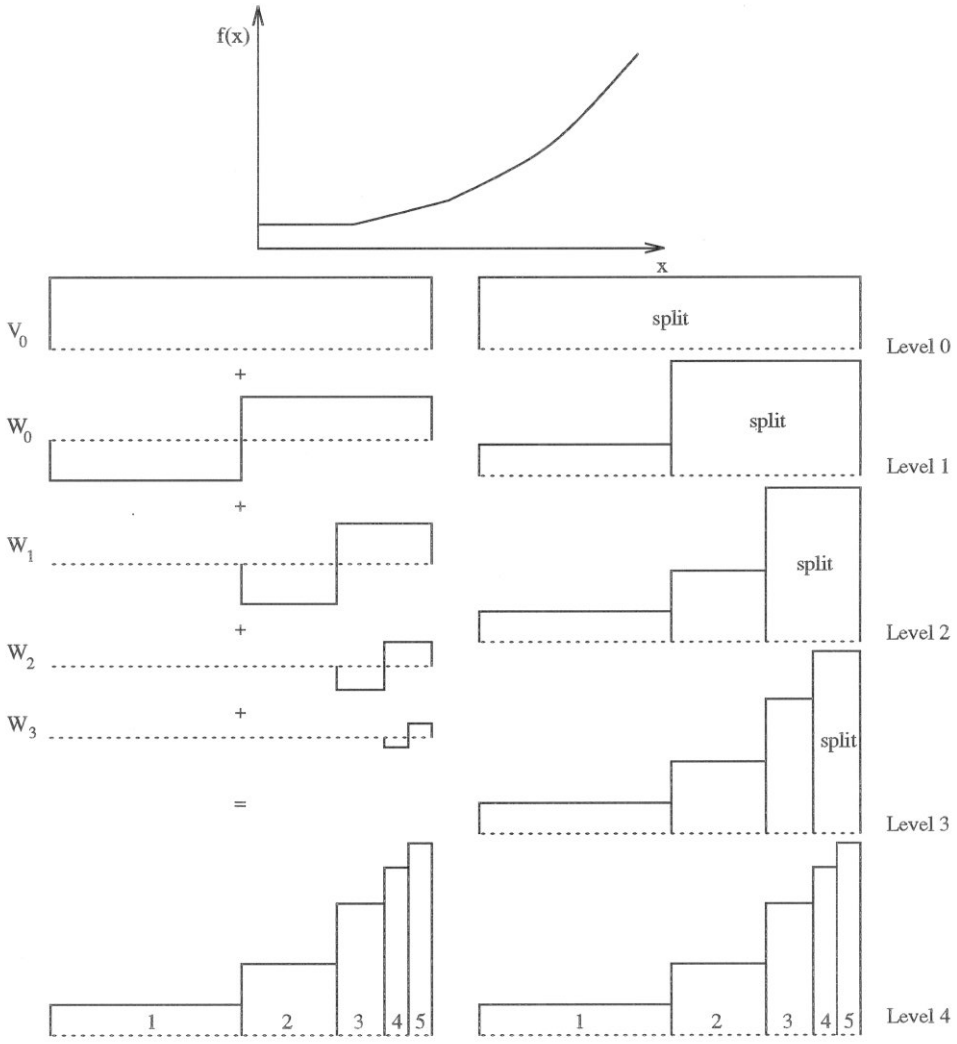



Figure 6: Given the approximation of a function at a certain level and an oracle which decides where to add details we refine the representation of the function. On the left this occurs by adding detail functions of ever finer levels where they are needed. On the right we instead cause a splitting of (smooth) basis functions to occur. The two processes are equivalent, but the one on the left corresponds to the standard decomposition, while the one on the right is akin to the non-standard decomposition.

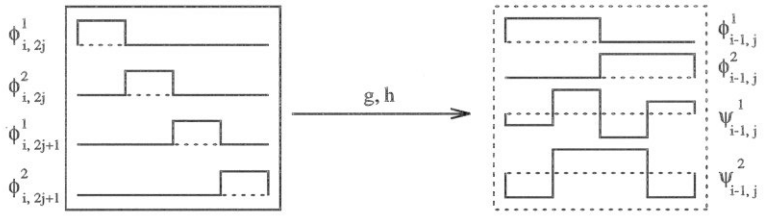



Figure 7: The $\mathcal{F}_2$ wavelet construction. $\mathcal{F}_2$ bases have two different detail shapes. Both of the detail shapes have two vanishing moments (from [28]).

overlap section 5. This is the reason why we find the $\log n$ dependence in the standard realization of the operator. However, if we use the realization of the function as shown on the right, the only function that needs to link up with $g$ is the smooth function of section 5 (which lives on level 4). None of the functions at levels above participate!

Note though that the realization on the left uses a basis proper, while the realization on the right uses an overrepresentation. The easiest way to see this is to note that all the functions in the left column are orthogonal to each other, thus they are clearly independent. On the right this is not true. For example, smooth functions 4 and 5 at level 4 can be combined to build their parent at level 3. Similarly 3, 4, and 5 can be combined to build their parent at level 2, and so on. Another way to put the same observation is to say that the smooth functions at each level can be used to realize the next coarser level, they are in some sense "proxies" for all the levels above.

**Tree Wavelets** The reason the Haar basis allowed us to do this simplification lies in the fact that the smooth functions in the Haar system do not overlap. For more general wavelets there is overlap between neighboring functions. Consequently the above substitution, while still possible [27], is not as straightforward. The problem arises with overlapping basis functions because some regions may be accounted for multiple times, in effect introducing the same energy more than once into the system. Using only detail functions like we did in the left column of Figure 6 avoids this problem, since by definition detail functions at one level are linearly independent to the detail functions at another level. The wavelets that were used in the original WR work [28, 54] did not suffer from this difficulty because they were tree wavelets. In a tree wavelet the filter sequences do not overlap.

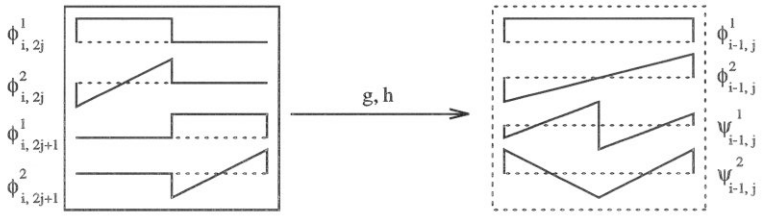



Figure 8: The $\mathcal{M}_2$ wavelet construction whose smooth shapes are the first two Legendre polynomials. Both of the detail shapes (lower right) have two vanishing moments (from [28]).

The Haar basis is a tree wavelet basis. When trying to extend these ideas to more vanishing moments we have to allow more than one wavelet function over a given interval to keep the filter sequences from overlapping. In essence, neighboring intervals are decoupled. WR used so called Flatlets, which are still piecewise constant, but combine more than two smooth functions to increase the number of vanishing moments (Figure 7 shows the shape of Flatlets with two vanishing moments [28]). Another set of wavelets explored for WR was introduced by Alpert [1] under the name multi-wavelets. Over each interval a set of Legendre polynomials up to some order $M - 1$ is used and a wavelet hierarchy is imposed. Here too, neighboring intervals decouple giving multi-wavelets the tree property as well (see Figure 8 for a multi-wavelet with two vanishing moments). Details regarding these functions in the context of WR can be found in [28, 54].

Using tree wavelets and the substitution of all detail functions by sequences of smooth functions leads to an obvious simplification of the code and follows naturally from the historical development. However, if we had started from the abstract theory, this development might have appeared odd.

`PushPull`

In the case of HR (piecewise constant basis functions) we pointed out the need for a `PushPull` function. Its definition was derived immediately from the physical meaning of irradiance and radiosity respectively and followed from the fact that a given surface had different refinements with respect to different sources. These different "views" of the same surface, or said differently, the fact that different sources would deposit

power at different levels of the hierarchy, needed to be consolidated. We also indicated that such an operation needs a non-trivial extension to higher orders with the help of the two scale relationship.

Using the point of view of operator realizations we can now see that `PushPull` was a consequence of the non-standard operator realization of HR. `PushPull` is the function needed to consolidate these over representations after every iteration. In the standard realization this consolidation is not needed since all the functions involved are linearly independent (they form a proper basis).

However, there are other reasons to use a `PushPull` function, namely to change between two sets of bases. This is accomplished by using push to go from one wavelet basis to a canonical basis at the leaves, followed by a pull which goes to another wavelet basis. An example, in which this property can be used, arises from the desire to employ bi-orthogonal wavelets. Recall that the sparsity is a direct function of the number of the number of vanishing moments. This constraint, taken together with other constraints, may lead to a basis which cannot be orthonormal, giving rise to the desire to use a bi-orthogonal basis. This was the case for the Flatlets used in [28, 54]. The desire for a maximal number of vanishing moments together with the constraint that the smooth functions should continue to be box functions, led to a construction in which the dual basis functions did not have vanishing moments. Consequently the projection operators were arranged such that only primal basis functions were under the integral operator. Formally this leads to a projection of the form $P_L \mathcal{G} P_{\tilde{L}}$ (where the "~" denotes the dual hierarchy). Applying a step of our iterative solver results in expansion coefficients with respect to the wrong basis ($P_L$ instead of $P_{\tilde{L}}$). Intuitively we change from the primal basis to the dual basis when "going through" the integral operator. In this case `PushPull` can be used to change back to the primal basis after an iteration as was done in [28, 54].

### Subdivision

We asserted that HR is the non-standard realization of the radiosity operator in the Haar basis with the detail functions replaced by their constituent smooth functions. Above we showed that subdivision is equivalent to the addition of detail functions

under our replacement policy resulting in the over representation. There still remains a difference to the abstract version of the non-standard representation.

The abstract derivation has us consider subdivision on *both* ends of a transport when more detail is needed, while the function `ProjectKernel` only divides *one* of the surfaces involved. Equation 10 has summands which involve level indices which are identical on the left and right. If we want to allow for subdividing only one of two elements[8] under consideration, we appear to be violating this rule. In fact HR allows couplings across the hierarchies irrespective of the level. The crucial observation is that for each element, there are only couplings with a *single* level. Suppose we subdivided both elements in a given interaction evenly as the abstract derivation has us do. Doing so we are likely to discover that some of the rows of the resulting operator have more coherence than some of the columns, or vice versa. This implies that one of the two surfaces did not need as much subdivision. In this case we may decide to compress the rows further while keeping the columns untouched. Equivalently, during the recursive subdivision scheme, we can decide to only subdivide one end of an interaction instead of both ends. The resulting set of interactions still has the non-standard structure, even though the indices on the left and right may be different.

### Refinement

Another important difference between the standard and non-standard realization concerns quadrature error. In the non-standard realization we keep descending down the subdivision call sequence until a link is created. In this sense only the leaves of a given subdivision hierarchy interact with the leaves of an appropriate subdivision hierarchy on another primitive. The fact that this set of leaves is different for different interaction partners lead to the over representation and the need for consolidation with `PushPull`. In contrast, the standard realization has couplings across all levels of a pair of subdivision hierarchies, involving basis functions corresponding to the root all the way down to some appropriately chosen leaf level.

Consider the role of quadrature error in these two schemes. In the standard realization refinement occurs because a given interaction is found to fail to capture

[8]The term "element" encompasses both leaf and internal nodes in our subdivision hierarchy.

adequate detail in the answer and a finer detail function is added (see Figure 6). The coupling associated with the parent function continues to be used. Note however, that it was computed to within an accuracy which is a function of its level. Asserting that that level has too much error and inducing the creation of a finer level implies that the coupling coefficient at the parent level has too much quadrature error as well. Consequently any standard realization needs to improve the accuracy in a parent interaction coupling (the quadrature) whenever it adds a detail function at a finer level (Christensen *et al.*[14] made this observation in the context of radiance). This contrasts with the non-standard realization which, by its definition, replaces a link which is found to have to much error with a set of finer links satisfying the error criterion. In practice it actually never generates the link with too much error, but instead recurses. Only if one of the successive refinement strategies is employed can it occur that previously generated links are replaced. In any case, the quadrature error associated with a given link is a function of the size of the elements interacting, and in this way is of the same order as the oracle error estimate, which is also a function of the sizes of the interacting elements. This insures that the links do not have too much quadrature error or too little. The notion of "too little error" refers to the fact that there is no benefit gained from computing a quadrature more accurately than the given basis function support can realize anyway. To see this, consider the extreme case of a quadrature being computed analytically. Even though the quadrature (coupling coefficient) is exact the resulting accuracy in the answer can be no better than afforded by the order of basis functions and the subdivision level.

## 2.5 Wavelet Radiosity

Early on in this chapter we discussed how the reasoning of HR extends easily to higher orders simply by using piecewise polynomial bases over each element and otherwise following the basic structure of HR. The counting argument worked just the same. In practice, higher order bases imply that the bounds fall below our threshold even faster than for the case of piecewise constant functions. The bounds given had an $M^{th}$ power on the size to distance ratio for $M-1$ order piecewise polynomial bases or more

generally wavelets with $M$ vanishing moments. We discussed the need for an oracle and gave a simple prescription of how to implement it. Seeing the connection with wavelets we also know now how to perform push/pull in such a higher order hierarchy by using the two scale relation. All the pieces for WR as described in [28, 54] are now in place.

We intentionally ignored many of the actual implementation details in this chapter. These are best found in the original papers. Instead we concentrated on the overall structure behind these ideas. The geometric reasoning of relative size to distance ratios leads to asymptotically faster algorithms for radiosity. This geometric argument finds its analog in the theory of Calderon-Zygmund operators and their realization in wavelet bases.

We conclude this chapter with a review of some of the experiments using Flatlets and multi-wavelets for radiosity computations to show in a more quantitative way how these bases perform. These experiments were first reported in [28, 54] and executed in collaboration with Steven Gortler.

### 2.5.1 Performance of Wavelet Radiosity

The first experiment was designed to see the relationship between sparsity and the achieved error. For this purpose the configuration in the upper right hand corner of Figure 9 was uses. A $1 \times 1$m emitter of uniform emitted radiosity and 0 reflectivity was placed above a $2 \times 2$m receiver with no emission and a reflectivity of 1. The distance between the two was set to 0.1m. The short distance was chosen to create a strong gradient in the resultant radiosity on the receiver. The emitter was rotated $45^o$ to avoid a serendipitous lineup of the subdivision lines with the strongest features in the resultant radiosity on the receiver. Both this rotation and the closeness between emitter and receiver make the approximation of the radiosity function on the receiver numerically much more challenging. We used piecewise constant basis functions. As our reference solution for the error determination we took advantage of the fact that an analytic solution can be given for the radiosity on the receiver at any point.

Four experiments were conducted. In each one the finest level of subdivision was

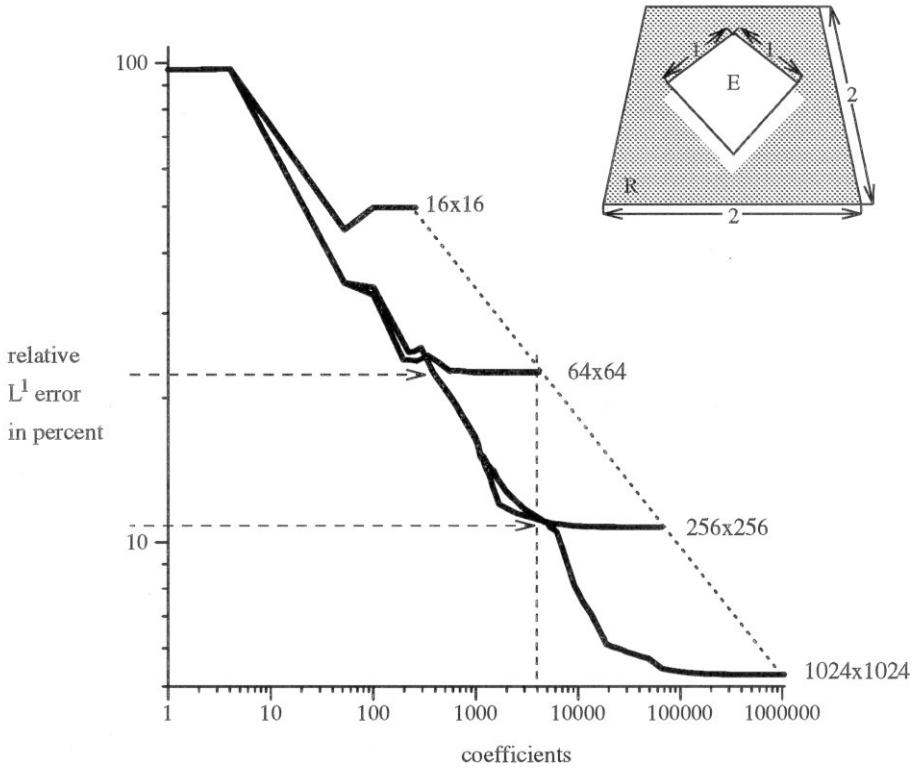



Figure 9: Experimental data for the configuration shown in the upper right. The finest level subdivision for the 4 curves was to to $h = 1/4, 1/8, 1/16, 1/32$ respectively. Each curve represents the action of the oracle and plots the relative $L^1$ error as a function of the number of $G_{ij}$ coefficients allocated by the oracle. (Adapted from [28])

limited to $h = 1/4, 1/8, 1/16, 1/32$ (via the function `RecursionLimit`) respectively, resulting in interaction matrices of size $16 \times 16$, $64 \times 64$, $256 \times 256$, and $1024 \times 1024$ at the far right of each of the curves. The curves themselves result from running the tests with different error bounds, in each case letting the oracle decide which couplings to generate. The results were plotted as relative $L^1$ error versus the number of interactions ($G_{ij}$) created. For a given subdivision limit turning the error criterion down would eventually force the oracle to create all possible $n^2$ interactions (without necessarily realizing the requested error bound due to the artificial termination of the recursion). These points are indicated at the far end of the curves and connected by the diagonal dotted line. The slope of this line as a function of the associated $h = \text{interactions}^{1/4}$ parameter is 1, as one would expect.

The first conclusion that can be drawn from this data is that the final accuracy for a given subdivision limit is reached long before all coefficients are being used. These "knee" points are reached at approximately 10 coefficients per element for $h = 1/8, 1/16, 1/32$ (i.e., when each element is connected to 10 other elements on average final accuracy has been achieved). This is also the point where the oracle would ordinarily demand more subdivision but we artificially limited it. The significance of the factor 10 lies in the fact that it characterizes the sparsity of the associated matrix. Instead of $n^2$ interactions we have approximately $10 * n$ interactions, which supports our claim that the resultant algorithm is linear in the number of elements. An informal survey amongst users of the system [65, 23] has shown that these factors vary from about 9 to 10 linkups on average per element. These observations were made across different scenes (including scenes with significant occlusion), different order basis functions (constant to cubic), and different $\epsilon$ parameters.

In practice we will not limit the finest subdivision level artificially, but instead allow the oracle to stay on the downward line on the left rather than force it onto one of the horizontal curve paths. To illustrate this point consider the following observation. Using approximately 4000 interactions (vertical dashed line) we can achieve an accuracy of only about 22% when considering full matrices only. If we use the oracle to allocate the 4000 interactions without limiting the subdivision level, we can reach an accuracy of about 11% (see the horizontal dashed lines).

When letting the oracle decide where to put subdivisions it gets its power from the ability to leave some subdivisions coarse while making others very fine. It is hard to relate this to the classical notion of $h$ since interaction occur across many levels of $h$. However, we can make the following observation. Define an effective $\tilde{h}$ by taking the 4th root of the number of coefficients. Doing so has consistently (for the data reported here and below) resulted in an exponent of $p = 1.2$ for $O(\tilde{h}^p)$. We can think of this as one way to to characterize the sparsity we achieve. At this point we can not give a theoretical justification for this number, but it occurs consistently.

The next experiment shows a different configuration for which once again we have an analytic solution. Figure 10 shows heightfield plots of the analytic solution, an error surface for the computed solution using piecewise constant basis functions with 1 vanishing moment (Haar; top right), 2 vanishing moments (Flatlet 2; bottom left), and 3 vanishing moments (Flatlet 3; bottom right). For each of the computations a total of 8000 individual couplings were computed ($h$ was limited to 1/32). This experiment shows how increasing the number of vanishing moments, while keeping everything else constant, decreases the error and also creates a smoother error. From it we can see that for the same amount of work we can generate better solutions by increasing the number of vanishing moments.

Finally we show the results for increasing both the number of vanishing moments and the smoothness of the basis functions. In this case we show the results for the same configuration as used in Figure 9. The basis functions are the multi-wavelets with 1, 2, 3, and 4 vanishing moments (piecewise Legendre bases of order 0, 1, 2, and 3 respectively). Figure 11 shows the convergence behavior (fitted lines) for a sequence of experiments. Once again we measure relative $L^1$ error as a function of the number of $G_{ij}$ allocated by the oracle. We find that the ratio of the slopes is almost exactly $1:2:3:4$ as one would expect from the $O(h^p)$, $p = 1, 2, 3, 4$, approximation property of the underlying space. As observed in the context of Figure 9 we have effective $p$ values, with respect to $\tilde{h}$, of 1.2, 2.4, 3.6, and 4.8.

When plotting the data as we did in Figure 11 on a scale counting the number of coefficients computed we are not accounting for the increased amount of work per coupling when using higher order basis functions. To take advantage of the higher

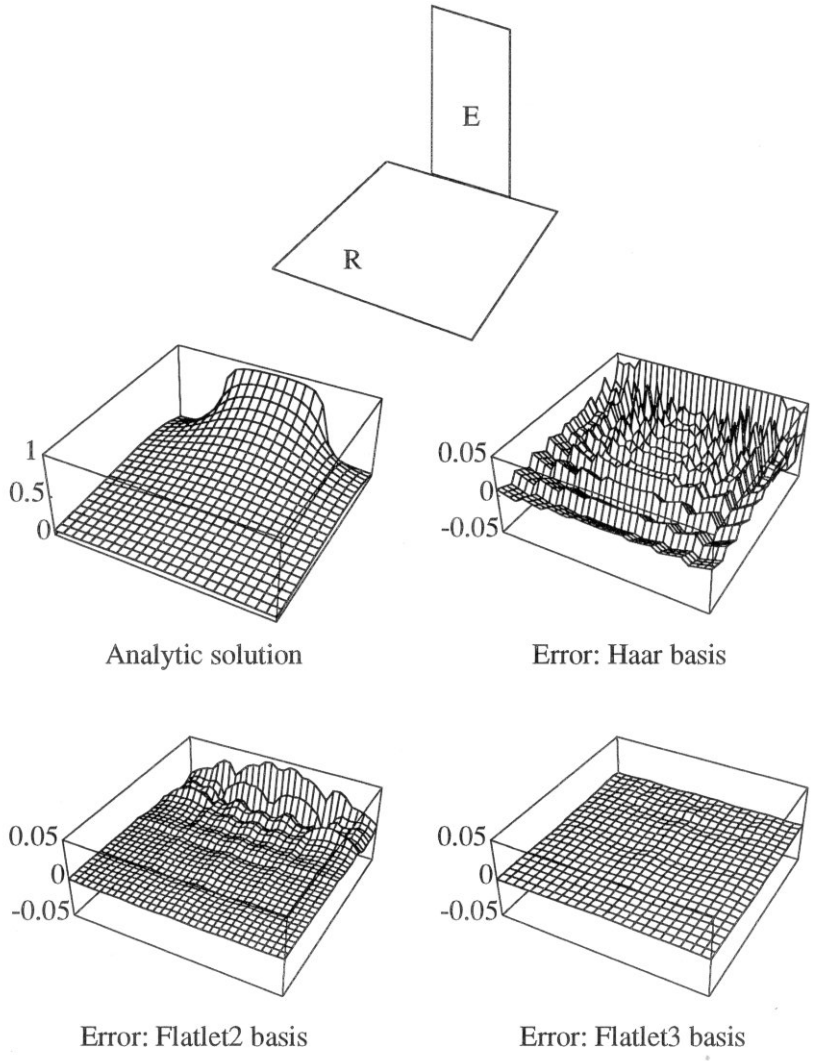



Figure 10: In the upper left the analytic solution plotted as a height field for the configuration at the top, a receiver of $2 \times 2$m at right angles to an emitter of $1 \times 2$m. On the top right the difference of the computed solution and the analytic solution for the Haar basis. The bottom left shows the difference of computed versus analytic solution for a piecewise constant basis with 2 vanishing moments, and on the bottom right using a piecewise constant basis with 3 vanishing moments. (Adapted from [28])

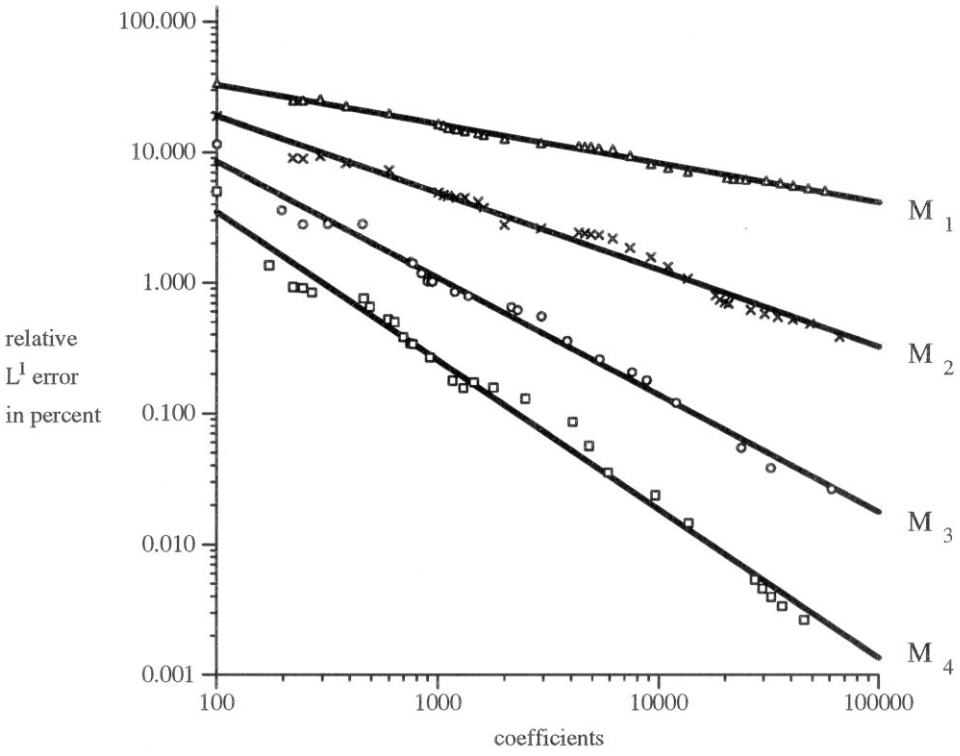



Figure 11: Using the test configuration of Figure 9 again this graph shows relative $L^1$ error as a function of the number of couplings generated for the multi-wavelet family with 1-4 vanishing moments. (Adapted from [28])

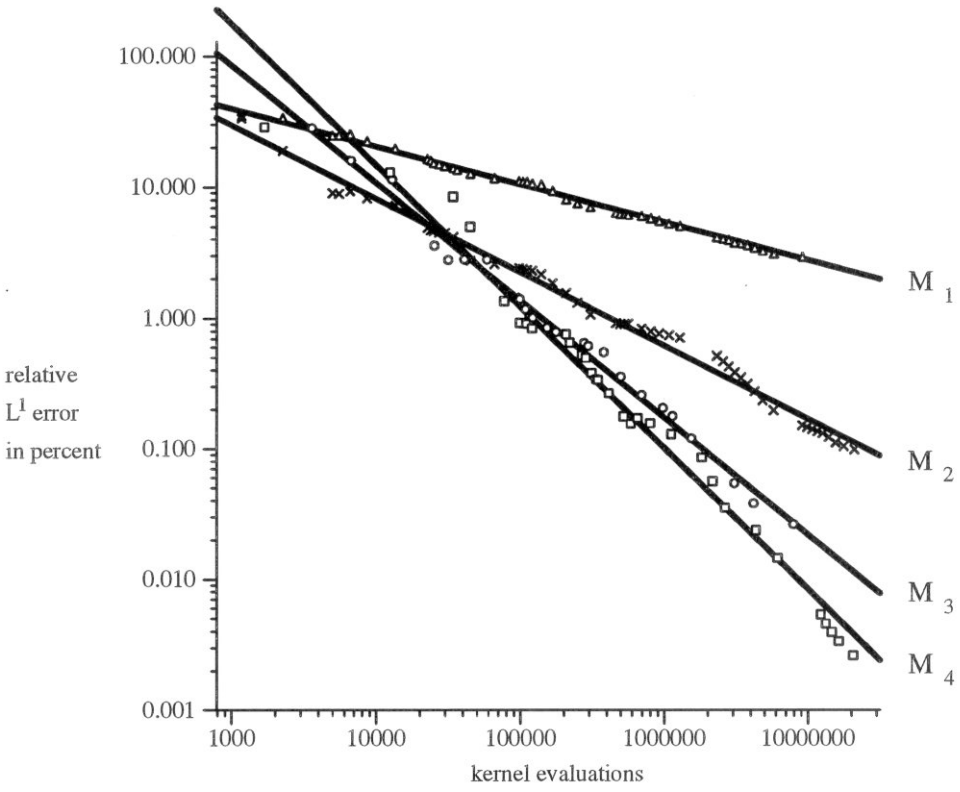



Figure 12: This plot shows the same data as Figure 11 but this time with relative $L^1$ error as a function of the number of kernel evaluations, which is higher for higher order basis functions per coupling. Kernel evaluations are roughly proportional user experienced work. (Adapted from [28])

order basis functions, quadrature rules of higher order, which are more expensive, need to be used. Since we are doing 4 dimensional product integration (two surfaces, each with 2 parameter directions), the cost of the quadratures grows as a 4th power of the number of sample points required. Using Gaussian quadrature the number of samples in each dimension can be set equal to the number of vanishing moments. We can account for the work increase by plotting relative $L^1$ error as a function of the number of evaluations of $G_A$, which more closely relates to user experienced compute time. Doing so results in the plot in Figure 12. Here the higher order basis function curves are translated over to the right by $\log(M^4)$[9].

In this graph we can see that for a given amount of error, say 1% or 0.1% the cubic basis functions ($M_4$) will give us the desired accuracy for the least amount of work. Only for errors above approximately 4% do the linear basis functions provide the most efficient solution. This justifies the conclusion that in the case of radiosity (we will see that it is different for radiance) higher order basis functions generally outperform lower order bases (i.e., linear is better than constant, quadratic better than linear, and cubic better than quadratic).

Figures 13 and 14 show a sampling of images which have been computed with the WR system by different users over the past 18 months. At the top of Figure 13 is an interior scene from [28]. It uses multi-wavelets with 2 vanishing moments. The input consisted of 161 polygons, which were meshed into 40000 elements, giving rise to 340000 couplings. The computation time was 100 minutes on a 50MHz R4000 SGI computer.

Below it are four images taken from Teller *et al.*[66]. They used the WR solver as a component in an out-of-core radiosity system capable of dealing with very large architectural databases. It uses extensive visibility analysis to cluster computations into subtasks which can be solved in core. On the middle left is an overview of an entire furnished floor of Soda Hall on the UC Berkeley campus. It contains 54,448 quadrilaterals which were meshed into 734,665 elements and generated 8,366,229 couplings. Linear basis functions were used and the computation time on a 50MHz R4000 SGI

[9] $M_1$ and $M_2$ maintain their relative position since $M_1$ uses as many sample points as $M_2$, which is not numerically necessary, but was done to avoid aliasing problems.

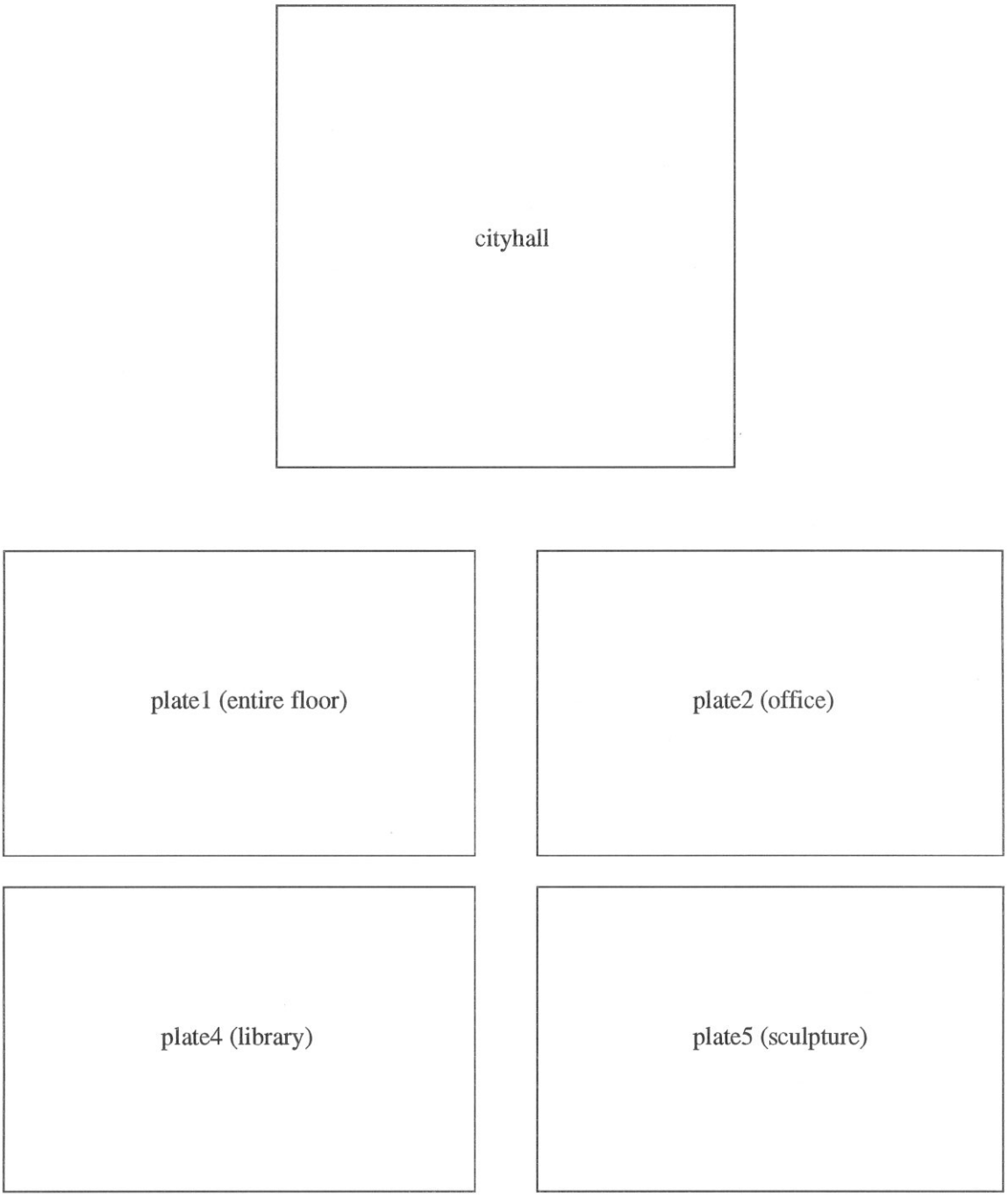
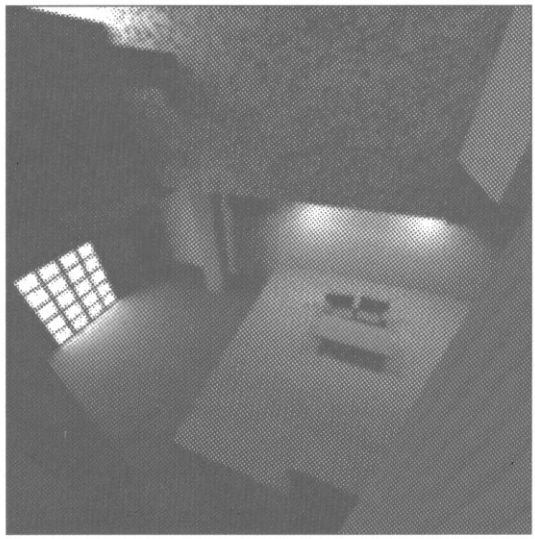
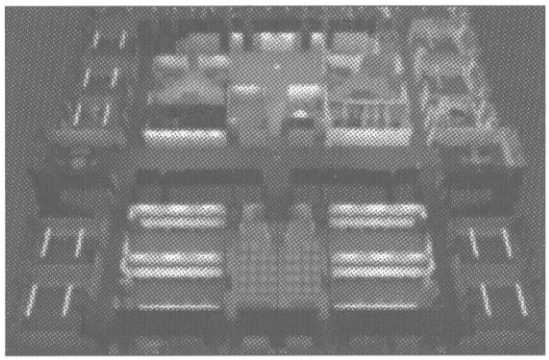
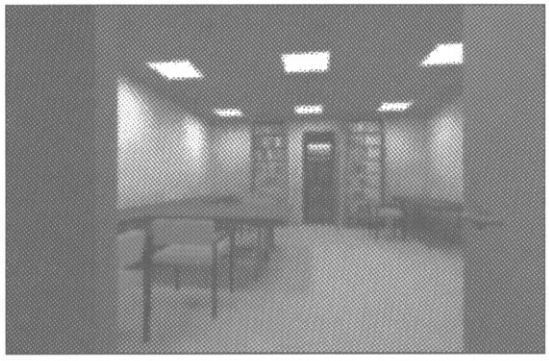



Figure 13: Example images from users of the WR system.







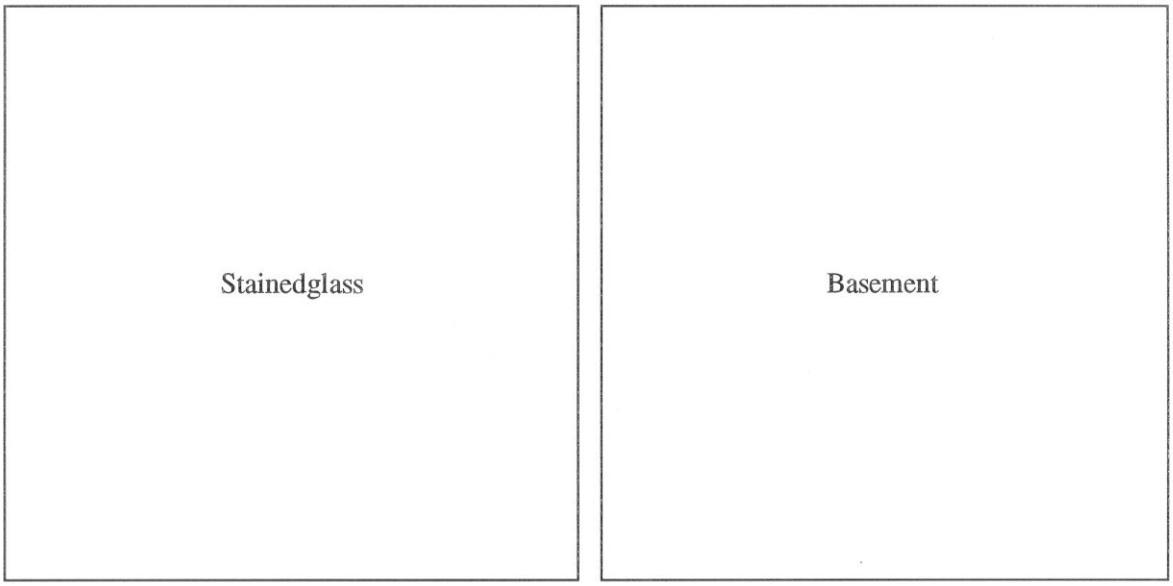
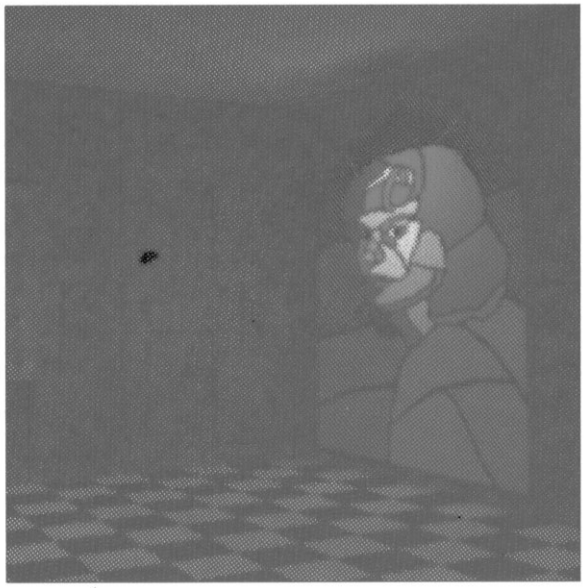



Figure 14: Example images from users of the WR system.

computer was 48.5 hours. The middle right and bottom show close ups of individual rooms and hallways.

Figure 14 shows two images from Gershbein *et al.*[24]. They added a facility to include texture maps in the WR system and take their effects into account during the solution process. Texture maps are used to modulate both emission and reflection properties, allowing the user of the system to increase visual complexity without increasing geometric complexity. On the left is an image of a stained glass window. The entire wall containing the window is a modeled by a single quadrilateral and an emissive texture (the window) and a reflective texture (the surrounding brick). The scene contained only 23 quadrilaterals, 10 texture maps and was meshed into 1800 elements connected by 13000 interactions. Solution time using cubic basis functions (multi-wavelets with 4 vanishing moments) was 36 minutes on a 50MHz R4000 SGI computer. On the right is another scene using emissive and reflective textures. It was modeled with 99 quadrilaterals and 62 textures. These resulted in 9500 elements and 90000 interactions using linear multi-wavelets. The solution time was 10 minutes on a 50MHz R4000 SGI computer.



## 2.6 Summary and Discussion

We have discussed the radiosity equation and in particular two approaches to solving it efficiently, Galerkin radiosity and hierarchical radiosity. Unifying them under the framework of wavelets has allowed us to reap the benefits of both approaches, higher order basis functions and asymptotic improvements from $O(n^2)$ to $O(k^2 + n)$, where $k$ is the number of input surfaces and $n$ the number of elements the environment is meshed into. We showed that the kernel function of the radiosity operator satisfies a generalized Calderon-Zygmund property justifying the application of results on the sparse realization of Calderon-Zygmund operators in wavelet bases. The resulting wavelet radiosity algorithm performs as predicted by theory both in terms of the benefits of higher order basis functions and the benefits of sparsity with increasing numbers of vanishing moments. Implementing the algorithm led us to consider an oracle function which identifies the important entries in the sparse linear system, which approximates the solution of the original integral equation.

Because of the historical development out of HR the bases used up to this point were limited to tree type wavelets, which decouple neighboring intervals, lead to simple realizations of the non-standard form, and require pushpull operators to consolidate the representations at the different levels of resolution. This lead to great code simplifications in the form of tree structured subdivision. It also means that there are no continuity constraints between neighboring elements per se. The connection with wavelets and their much more general context suggests that many interesting bases, some of which might work even better for the radiosity problem, are awaiting their use.

So far we have not directly addressed the discontinuities in the kernel function itself, which arise from the visibility term. The polynomial oracle we have employed reacts to these by subdividing further along the resulting shadow boundaries. Although asymptotically a valid strategy, it appears advantageous to consider discontinuity meshing, relaxing the currently used uniform subdivision. This approach has already been demonstrated for constant basis functions by Lischinski *et al.*[43] and awaits its generalization to higher orders. Another potential avenue to address this

issue would be the use of adapted wavelets as proposed by Andersson *et al.*[2].

In the algorithm as implemented we have only considered refinement strategies, but not yet the clustering problem (i.e., grouping sets of small input surfaces into groups). This led to the $k^2$ dependence in the asymptotic bound argument. Recently a number of researchers have published algorithms to address this issue [59, 62, 38, 50] and we look forward to further developments in this area in particular.

As evidenced by the many papers published, which aim to extend and improve the basic HR algorithm, we believe that the use of multi-resolution hierarchical representations is firmly established at this point and will continue to grow.

# Chapter 3

# Radiance

## 3.1 Introduction

In the first part of this dissertation we discussed the radiosity equation and how wavelets can be used to solve it efficiently. The second part is devoted to the radiance equation. Recall that radiosity described light transport under the assumption of diffuse (uniform in all directions) reflection and emission. This implies an idealization which is often violated in practice. In particular we would like to be able to describe reflective and emissive properties which exhibit directionality. The extreme example in this category is mirror reflection in which light coming from a particular incoming direction will exit in only one outgoing direction. Most materials exhibit properties which are best described by reflectivities which are in the regime between diffuse and mirror reflection. These are typically referred to as "glossy" materials.

Our treatment of radiance attempts to capture the regime of glossy materials. Purely mirror reflectivity is best treated with methods other than finite elements due to the numerical problems of modeling the Dirac delta distribution like response of a mirror material.

The radiance equation in the form used today in computer graphics (see the next section) was first formulated by Kajiya as the "Rendering Equation" [37][1]. Few results have been reported on attempts to solve the general rendering equation because of

[1]Kajiya actually used a slightly different physical quantity in describing transport than we do.

the enormous cost involved in doing so.

There have been some attempts at extending radiosity to handle specular or glossy reflections [35, 56, 57, 41]. Immel *et al.*[35] used a discretization of the sphere of directions to describe transport of light as a function of surface parameters and directions. They pointed out that their approach was wholly impractical due to the large size of the resulting linear system, but it did yield view independent solutions for scenes which contained glossy reflectors. Shao *et al.*[56] used a form of procedural refinement to maintain the basic framework of radiosity. They introduced weightings into the usual form factors between surfaces to account for directional effects. These weightings were refined on successive iterations of the solution algorithm to converge to the desired answer. Shirley [57] and Saec and Schlick [41] used Monte Carlo ray tracing to estimate the transport between surfaces in the presence of glossy reflectors. Sillion and Puech [60] used a method integrating mirror reflection and diffuse reflection through a hybrid raytracing and radiosity approach.

The most immediate antecedents of the method described in this thesis are those given by Sillion *et al.*[61] and Aupperle and Hanrahan [6]. Sillion *et al.*[61] were the first to apply a finite element framework to the general glossy case. They used constant basis functions over surfaces and parameterized the directional properties of light with the use of spherical harmonics basis functions. Using this technique they achieved a unified framework using finite elements as distinct from the earlier hybrid algorithms. Aupperle and Hanrahan [6] extended their earlier HR work [32] to glossy reflectors, demonstrating the first instance of an algorithm which exhibits linear growth in the number of couplings.

Our extension of WR to wavelet radiance in some sense mirrors the extension of HR to hierarchical radiance performed by Aupperle and Hanrahan. We will show below how wavelets provide a framework to analyze hierarchical techniques for the radiance problem and extend them to higher order. As a result improvements in performance similar to those achieved in the case of radiosity will be found.

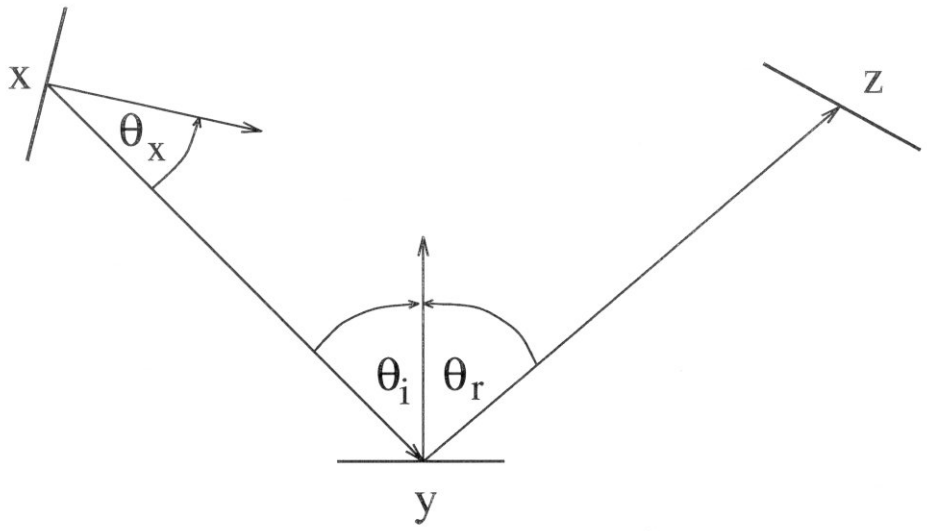



Figure 15: Geometry of the canonical three point transport with radiance originating at some surface X and being reflected at Y in the direction of Z.

## 3.2 The Radiance Equation

In this section we review the basic derivation of the radiance equation. This presentation borrows heavily from [19, Chapter 2].

The fundamental radiometric quantity we are interested in is *Radiance*. Radiance is a function of spatial and directional coordinates. Its units are $\frac{W}{m^2\, sr}$, or power per unit area per unit solid angle. Note that the area is perpendicular to the direction of the solid angle. Why is this a sensible quantity to use? Using the image of photons as particles it makes sense to consider the number of photons going by a point in space ($\vec{x}$), in a particular direction ($\vec{\omega}$), in an instant. This quantity is of course zero since space, direction, as well as time are zero. Defining this quantity infinitesimally (i.e., considering photons going through a small area in a small set of directions in a small amount of time and letting all of these quantities go to zero) we naturally arrive at the density of particles $p(\mu)\cos\theta$ at a point, in a single direction, at an instance. Here and in the following $\mu = (\vec{x}, \vec{\omega})$ is used as a coordinate free way of talking about a point and an attached directed line. The factor $\cos\theta$ arises from the orientation of the small area which we considered. Since we do not want to tie the density to a particular surface it makes sense to separate out the term $\cos\theta$. We thus

have radiance, $L(\mu)$ giving us the density of flux at $\mu$.

Based on the above motivation and in order to not tie ourselves to any particular parameterization we can consider $L$ to be a function defined on $\mathrm{M}^2 \times \mathrm{H}^2$, i.e. the set of all points on some two dimensional manifold (not necessarily connected), and a set of directed line segments attached to each point $\vec{x} \in \mathrm{M}^2$, which is isomorphic to a hemisphere. $\mathrm{M}^2$ will be the set of all surfaces in our scene. In all our considerations below we will assume opaque surfaces and hence $\mathrm{H}^2$ will always be the hemisphere above a given surface point and characterize all line segments going through that point. The fact that $\mathrm{H}^2$ is equivalent to $[0,1]^2$ simplifies our approach, since we are already dealing with a $[0,1]^2$ parameterization of surface patches. Thus radiances are members of the space of finite energy functions which map $\mathrm{M}^2 \times \mathrm{H}^2$ into the reals, $L^2(\mathrm{M}^2 \times \mathrm{H}^2; \mathrm{R})$. We equip this space with the standard inner product $\langle f, g \rangle = \int_{\mathrm{M}^2 \times \mathrm{H}^2} f(\mu) g(\mu) d\mu$, where $d\mu = d\vec{\omega} \cdot d\vec{A}$ describes the appropriate measure. The fact that this is the right measure follows immediately from the observation that the measure of rays is a function of the diameter of the beam ($d\vec{\omega}$) and its orientation (dot product) with respect to the receiving surface element ($d\vec{A}$).

To describe surfaces and how they reflect light we use the bidirectional reflection distribution function (BRDF) $f_r$. This function, $f_r(\vec{\omega}', \vec{y}, \vec{\omega}) = f_r(\mu_y', \mu_y)$, describes the probability of a given photon arriving from $\vec{\omega}'$ at $\vec{y}$ transitioning to $\vec{\omega}$ and thus gives us the right quantity to use with radiances. We use the symbol $\mu_y = (\vec{y}, \vec{\omega})$ to denote the 2 parameter family of line segments anchored at $\vec{y}$ going in some direction $\vec{\omega}$. Note that due to Helmholtz reciprocity we have $f_r(\vec{\omega}', \vec{y}, \vec{\omega}) = f_r(\vec{\omega}, \vec{y}, \vec{\omega}')$.

Distinguishing between incoming and outgoing quantities (subscript $i$ and $o$ respectively) for now, we can write the outgoing radiance (due to reflection) for all $\mu_y = (\vec{y}, \vec{\omega})$ as

$$L_o(\mu_y) = \int_{\mathrm{H}^2} f_r(\mu_y', \mu_y) L_i(\mu_y')\, d\mu_y'$$

where $d\mu_y' = \vec{n}_y \cdot d\vec{\omega}'$ is the appropriate measure on the set of lines at a given point $\vec{y}$ with normal vector $\vec{n}_y$. More intuitively speaking, we compute the number of outgoing photons at $\vec{y}$ (on some surface) for any of the lines $\mu_y$ by summing up all the photons coming in from all directions $L_i(\mu_y')$ weighted by the probability that they will be reflected to $\mu_y$

This formulation has some drawbacks. It involves two functions defined over all surfaces, the outgoing radiance $L_o$, and the incoming radiance $L_i$. Furthermore when writing functions with respect to some set of smooth basis functions we will generally have difficulty representing $L_i$ since even for environments with no occlusion, in which $L_o$ will have no discontinuities, we find that $L_i$ is discontinuous as a function of the angular parameter. These discontinuities arise from the fact that as the line $\mu'_y$ at $\vec{y}$ varies, the point visible from $\vec{y}$ in the direction $-\vec{\omega}'$ will move across different surfaces in the environment [12].

In order to express the entire equation in terms of outgoing radiances only we begin by defining a unary minus operator on line segments. Given a line segment $\mu = (\vec{y}, \vec{\omega})$ we let $-\mu$ describe the line segment anchored at the point $\vec{x}$ visible from $\vec{y}$ in the direction $\vec{\omega}$, $-\mu = (\vec{x}, -\vec{\omega})$ (see Figure 15). Given two points $\vec{x}$ and $\vec{y}$ in a scene which can "see" one another we can write the line connecting them as $\mu = (\vec{y}, \vec{\omega})$ or $-\mu = (\vec{x}, -\vec{\omega})$ depending on which way we traverse it. Because radiances are conserved along a ray in the absence of a participating medium we observe that $L_i(\mu) = L_o(-\mu)$. In words, the incoming radiance at $\vec{y}$ from direction $\vec{\omega}$ is the same as the outgoing radiance at $\vec{x}$ in the direction $-\vec{\omega}$ if $\vec{x}$ is the point visible form $\vec{y}$ in the direction $\vec{\omega}$. Using this fact we may now write

$$L_o(\mu_y) = \int_{\mathrm{H}^2} f_r(\mu'_y, \mu_y) L_o(-\mu'_y)\, d\mu'_y \tag{11}$$

Note that we have hidden a geometrical search task in the expression $-\mu'_y$ (i.e., given $y$ and $\vec{\omega}'$ we need to determine the surface point $x$ in the direction $\vec{\omega}'$ to look up its outgoing radiance in the direction $-\vec{\omega}'$).

Accounting for emitted radiance (source terms) as well we get the radiance equation

$$\begin{aligned} L(\mu_y) &= L^e(\mu_y) + L^r(\mu_y) \\ &= L^e(\mu_y) + \int_{\mathrm{H}^2} f_r(\mu'_y, y, \mu_y) L(-\mu'_y)\, d\mu'_y \\ &= L^e(\mu_y) + \mathcal{T}(L)(\mu_y) \end{aligned}$$

where we used superscripts $r$ and $e$ to denote the reflected and emitted components of radiance respectively. The symbol $\mathcal{T}$ is used to denote the transport operator.

Note the similarity to the radiosity equation (1). Instead of $B(\vec{y})$ we have $L(\mu_y)$ (i.e., our quantities are defined over a 4 dimensional parameter domain instead of a 2 dimensional one). While the reflectance $\rho$ in the case of radiosity was outside the integration, it is now inside the integral in the form of $f_r$. This follows from the fact that in the case of radiosity no directional dependence was included in the reflectance properties. The kernel function consisted only of $G$ in the case of radiosity. Below we will see that this is in fact a parameterized version of the measure $d\mu'_y$ itself. The basic structure of the radiance equation falls once again into the class of Fredholm equations of the second kind, and once again we are dealing with a smooth kernel function, justifying our pursuit of wavelet methods for the approximate solution of this equation.

We now examine some of the properties of $\mathcal{T}$ which will be helpful in our later analysis.

### 3.2.1 Properties of $\mathcal{T}$

To understand some of the properties of $\mathcal{T}$ better we recall our way of arriving at it. Initially we distinguished between incoming and outgoing radiances in terms of which we can write

$$L_o = L_o^e + \mathcal{S}L_i$$

where $\mathcal{S}$ denotes the scattering operator. Following a suggestion by Arvo *et al.*[4] we introduce the symbol $\mathcal{H}$ to denote the operation of taking an outgoing radiance into incoming radiance, $\mathcal{H}L_o = L_i$. Substituting we arrive at

$$L_o = L_o^e + \mathcal{S} \circ \mathcal{H}L_o$$

from which we find $\mathcal{T} = \mathcal{S} \circ \mathcal{H}$. Applying $\mathcal{H}$ to both sides of the latter equation and reordering we arrive at

$$L_i = L_i^e + \mathcal{H} \circ \mathcal{S}L_i$$

where we defined a somewhat unusual quantity $L_i^e = \mathcal{H}L_o^e$.

We now show that $\mathcal{H} = \mathcal{H}^+$ since for arbitrary $f$ and $g$ we have

$$\langle \mathcal{H}f, g \rangle = \int_{\mathrm{M}^2 \times \mathrm{H}^2} (\mathcal{H}f)(\mu) g(\mu) d\mu$$

$$\begin{aligned} &= \int_{\mathrm{M}^2\times\mathrm{H}^2} f(-\mu)g(\mu)d\mu \\ &= \int_{\mathrm{M}^2\times\mathrm{H}^2} f(\nu)g(-\nu)d\nu \\ &= \int_{\mathrm{M}^2\times\mathrm{H}^2} f(\nu)(\mathcal{H}g)(\nu)d\nu \\ &= \langle f, \mathcal{H}g\rangle \end{aligned}$$

where we used the fact that the measure of $\mu$ is the same as the measure of $-\mu$. Similarly we have $\mathcal{S} = \mathcal{S}^+$ which follows trivially from Helmholtz reciprocity. Consequently we have $\mathcal{T}^+ = \mathcal{H} \circ \mathcal{S}$ and the observation that incoming quantities observe the adjoint transport operator.

We emphasize that all these observations are intricately linked to the inner product used (e.i., a different inner product would yield different notions of the adjoint of all the involved operators). For example it might be possible to define an inner product such that $\mathcal{T} = \mathcal{T}^+$. While this is an interesting open problem we will not pursue it here, since we prefer to use the natural inner product on line space, and in any case there is no need to have $\mathcal{T} = \mathcal{T}^+$.

### 3.2.2 The Question of Importance

Recently a number of papers in the graphics community [63, 7, 12, 48] 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.

The derivation is motivated by the fact that we only aim to minimize the error of the computed solution at our sensor (e.g., a virtual CCD camera). Let $R(\mu)$ describe the probability of our sensor producing a response to radiance arriving along $\mu$. Now 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

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

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

The last equation shows the well known fact that the response of our sensor to incoming radiance can be computed from the incoming radiance, $\tilde{L}_i$, 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 quantity is referred to as importance. The above algebraic relationship 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.

The fact importance is propagated using the same integral operator and its solution algorithm, allows us to use data structures which treat importance as if it were another set of wavelengths of light. Recall that visible light is typically modeled as a linear combination of some small set of basis colors such as red, green, and blue (r,g,b). Since we assume no coupling between different wavelengths, each one of these corresponds to an instance of the solution algorithm. In practice this is realized by making all color quantities be vector quantities. The instance of our operator and its solution due to importance fit into this framework as well. For example, if we choose to consider the receptivity of our sensor as a function of 3 basic wavelengths as well we could treat all power in the system as a 6 vector and transport this 6 vector just as we would the (r,g,b) 3 vector of a system which does not use importance. In our implementation we have chosen to consider importance as a gray quantity resulting in a 4 vector of wavelengths. The question that remains is that of how much "power" our sole source of importance should emit. Since the operator is linear the magnitude does not matter, it enters as an overall scaling into the error criterion. In practice we have found that the numerical behavior of our algorithm is more balanced if the total power of importance coming from the virtual observer is approximately of the same order as the total power of all visible light emitters.

That we should be able to use this uniform treatment of colors and importance(s)

may seem obvious from a mathematical point of view, but has not always been taken advantage of in previously published algorithms [63, 7].

## 3.3 Galerkin Methods for Radiance

Independent of the parameterization used we can write the operator equation we wish to solve as

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

Since $\mathcal{T}$ is a linear operator we can study the inversion of $(1-\mathcal{T})$ using linear operator theory.

A Galerkin approach to the inversion of $1-\mathcal{T}$ projects the operator into a subspace spanned by some finite basis [21]. Assuming an orthonormal (w.r.t. the *parameter domain*) basis a linear system of the following form results

$$\forall i, j: \quad l_{ij} = l_{ij}^e + \sum_{mn} T_{ijmn} l_{mn}$$

for some unknown expansion coefficients $l_{ij}$ of $L$ with respect to a basis $\{N_{ij}(\mu)\}$. The fidelity of such a projection and its generated solution is intimately linked to the basis functions used and their properties. For example, we will generally choose bases which span approximation spaces of some fidelity $O(h^p)$ where $h$ is the sidewidth of some domain subdivision and $p$ the order of approximation.

So far we have only discussed the radiance operator as an abstract entity. In order to apply a Galerkin method to it and derive an algorithmic prescription for an approximate numerical algorithm we need to choose some parameterization of the radiance operator. There are several choices which have different implications. We now turn to examining these

### 3.3.1 Parameterizing the Radiance Operator

In this section we describe the particular parameterization of the radiance equation that we will be using in our algorithm together with some alternatives. 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 15). Typically there are parameters intrinsic to a surface which parameterize these quantities (e.g., $\vec{x}(u,v)$). We will abstract from that fact in the following discussion to simplify the notation.

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 (i.e., writing $\mu = (\vec{x}, \vec{\omega})$)

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

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

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

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

$$G(\vec{x}, \vec{y}) = \frac{(\vec{n}_x \cdot (\vec{y} - \vec{x}))(\vec{n}_y \cdot (\vec{x} - \vec{y}))}{\|\vec{y} - \vec{x}\|^4} \, \mathrm{v}(\vec{x}, \vec{y})$$

which is in fact a parameterized version of the measure $d\mu_y = G \frac{dA_x}{dx} dx$.

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 $k_n^s$

$$\begin{aligned} k_n^s(\vec{x},\vec{y},\vec{z}) &= k^s(\vec{x},\vec{y},\vec{z})N_n(\vec{y}) \\ &\approx \sum_{ijm} T_{ijmn}N_m(\vec{x})N_i(\vec{y})N_j(\vec{z}) \end{aligned} \tag{12}$$

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 had only 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 $T_{ijmn}$

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

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 $k_n^d$

$$\begin{aligned} k_n^d(\vec{x},\vec{y},\vec{\omega}) &= k^d(\vec{\omega}'(\vec{x},\vec{y}),\vec{y},\vec{\omega})N_n(\vec{\omega}'(\vec{x},\vec{y})) \\ &\approx \sum_{ijm} T_{ijmn}N_m(\vec{x})N_i(\vec{y})N_j(\vec{\omega}) \end{aligned} \tag{13}$$

and $k_m^d$

$$\begin{aligned} k_m^d(\vec{\omega}',\vec{y},\vec{\omega}) &= k^d(\vec{\omega}',\vec{y},\vec{\omega})N_m(\vec{x}(\vec{y},\vec{\omega}')) \\ &\approx \sum_{ijn} T_{ijmn}N_n(\vec{\omega}')N_i(\vec{y})N_j(\vec{\omega}) \end{aligned} \tag{14}$$

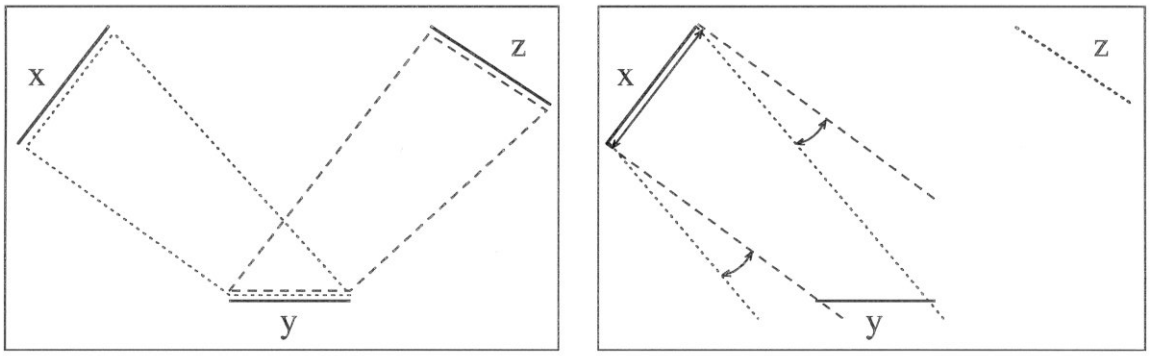



Figure 16: Illustration of the matchup of support between basis functions. On the left basis functions with spatial parameterization. Their supports (dotted respectively dashed outlines) match at Y. On the right basis functions with directional parameterizations. A basis function with spatial support on X and some directional support (spread of angles) does not match up cleanly with the spatial support of Y.

### 3.3.2 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 *modified*. In the case of the spatial parameterization we saw above (Equation 12) that instead of expanding $k^s$ we end up expanding $k^s_n$. In the directional case we gave two choices (Equations 13, 14) $k^d_n$ and $k^d_m$. In any case the original kernel is multiplied with one of the basis functions and the parameterization of the measure itself. 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 16). 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 16). 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 [13].

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.

From now on we will limit ourselves to the spatial parameterization since this is the parameterization we have chosen for an implementation.

### 3.3.3 Bounding the Number of Interactions

Earlier we argued a bound of $O(k^2+n)$ on the total number of coefficients enumerated in the case of radiosity. The key to the argument, after translating the abstract Calderon-Zygmund property into geometric language, was the observation that only a small constant number of surfaces can have an angle subtended which is too large for hookup at a given subdivision level. This argument extends in a straightforward fashion to the case of radiance as pointed out by Aupperle [5]. In the case of radiance we have not one hemisphere over which this argument is applied, but two hemispheres, one incoming, one outgoing. As in the case of radiosity we can bound the error in the

solution above by giving a bound on the difference between the approximate kernel as computed by `ProjectKernel` and the actual kernel.[2] Unfortunately the radiance kernel is not of the proper form to directly apply the Calderon-Zygmund properties, but we can reduce it to a concatenation of two such operators.

Suppose we are using constant basis functions only (a sufficient condition to argue the asymptotic bound even for higher order basis functions). We begin by letting $F_r = \sup |f_r|$ over the supports in question. Using this bound[3] we can now ignore $f_r$ entirely in the involved integrals. Assuming that the basis functions are bounded (as they are in our case) we can also replace the presence of $N_n$ in $k_n^s$ with its upper bound. Since we are using a spatial parameterization and constant basis functions only for the sake of this argument we are left with showing

$$\|G(\vec{x},\vec{y})G(\vec{y},\vec{z}) - G(\vec{x}_0,\vec{y}_0)G(\vec{y}_0,\vec{z}_0)\|$$

observes an angle subtended bound. Once again $\vec{x}_0$, $\vec{y}_0$, $\vec{z}_0$ are chosen to be the midpoints of the supports of areas involved.

Note that the integral operator corresponding to the kernel function $G(x,y)G(y,z)$ is simply the second power of the ordinary radiosity operator, $\mathcal{G}^2$. With this observation we can immediately reuse our arguments from the case of radiosity. If a given element $i$ in the radiosity case interacts with an element $j$, and $j$ only interacts with a constant number of other $k$, then the total number of basis functions with $j$ in the middle can at most be this "constant number squared." This is still only a constant number of basis functions, the property we need.

The bound on such a concatenation is simply the product of the individual bounds leading to

$$\|k_n^s(\vec{x},\vec{y},\vec{z}) - k_n^s(\vec{x}_0,\vec{y}_0,\vec{z}_0)\| \leq C\left(\frac{\|I\|}{r_{xy}}\right)^3\left(\frac{\|I\|}{r_{yz}}\right)^3$$

for a triple of surfaces away from the singularity (the case touching the singularity is handled just as it was in the case of radiosity and is not repeated here). $\|I\|$ gives the

[2] Again taking advantage of the boundedness of the radiance operator when mirror reflection is excluded (see Arvo [4]).

[3] The bound is infinite for mirror reflection, thus we must exclude it.

length of the longest dimension involved and $r_{xy}$ stands for the radius vector between $x$ and $y$, and similarly for $r_{yz}$.

Once again we have the crucial bound involving this time a *product* of angles subtended. If the product of these two angles subtended is small enough our error will be small. From this follows immediately the claim that subdivision will eventually bring the error below any strictly positive bound. The total number of triples involving the middle surface is again bounded by a small constant based on the observation that the two hemispheres above $y$ (one incoming, one outgoing) can only have a small constant number of surfaces with angle subtended larger than some threshold. At worst all these pairs of surfaces cause recursion. In practice we can generally do much better since the distribution of outgoing power as a function of some incoming direction is not uniform anymore. In some sense the uniform case is the worst case for this argument. In any case, we still have only a small constant number of such "pairs which are too large" at each level of the hierarchy, leading to a bound of $O(k^3 + n)$. This time the dependence on the number of input surfaces $k$ is cubic since the initial refinement is over all triples of (mutually visible) surfaces.

While in the case of radiosity the features of the kernel function $G$ were responsible for the (local) number of interactions created we now have a dependence on both $G$ and $f_r$, the BRDF. The asymptotic argument relies on our ability to bound $f_r$, which leads to the exclusion of mirror reflection from our implementation. Mirror reflection corresponds to a Dirac delta distribution for $f_r$. Given some finite upper bound the algorithm is most sensitive to the actual variation in the kernel function. Very peaked BRDFs require fine subdivision in the peak area to model it in a piecewise polynomial fashion, while away from the peak only few piecewise polynomial basis functions are required.

## 3.4 Implementation

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

As mentioned above we use a spatial parameterization with multi-wavelet (piecewise polynomial) bases. This approach is a direct extension of the earlier WR work reported in [28, 54]. The main difference is the added number of dimensions. Using a spatial parameterization, the directional dependence is parameterized with respect to other surfaces, resulting in basis functions whose domain is $[0,1]^4$. Consequently the quadrature rules are still product rules. Instead of being four dimensional as in the case of radiosity where two basis functions over $[0,1]^2$ interacted across the radiosity kernel, they are now six dimensional with two basis functions over $[0,1]^4$ interacting across the radiance kernel with the middle two dimensions matching up.

Coefficients of basis functions are not tied to individual surfaces anymore but rather to pairs of surfaces

```
typedef struct Basis{
  Element i;
  Element j;
  float coeffs[M][M][M][M];
  Basis children[2][2];
  List<Link> gather;
};
```

where $M$ is the number of vanishing moments. Each basis is coupled to some set of other bases in the system, those with which it interacts due to the action of the oracle. These interactions are the links which govern the exchange (gathering) of power.

Instead of having only links between surface elements there are now links between *pairs* of surface elements, with the middle surface matching up. These are the $T_{ijmn}$ of our expansion

```
typedef struct Link{
  Basis ij;
  Basis mn;
  float couplings[M][M][M][M][M][M][M][M];
};
```

The resulting data structures follow closely the ideas presented by Aupperle and Hanrahan in [6]. Visibility is determined as described by Teller and Hanrahan [67]. The actual code of our radiance implementation is a strict superset of our earlier WR code. The quadrature used is straightforward Gaussian quadrature [52]. Since we use a spatial parameterization simple product rules suffice. In the following sections we address some individual details associated with the particular implementation of the radiance solver.

### 3.4.1 ProjectKernel

Once again a simple recursive procedure [32, 28] considers the coupling between surface pairs (bases) and naturally accounts for all power 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 );
```

Note that we only subdivide one of the bases. Since its support is a four dimensional domain it would be natural to divide all dimensions in half, resulting in 16 children. Instead we follow the suggestion of Aupperle and Hanrahan [6] and divide only one of surfaces associated with a basis function. The result is that we have only four children, leading to a considerably smaller branching factor. When doing so it is important that the decision as to which element associated with a basis (i or j) is subdivided, be made consistently. Suppose when refining $ij$ with respect to $mn_0$ we

decide to subdivide $i$, and when refining $ij$ with respect to $mn_1$ we decide to subdivide $j$. In this case we will have an inconsistent notion as to what the children of $ij$ are. This must be prevented. In order to make the decision of `PreferredSubdivision` be unique it is based on properties of $ij$ and $mn$ in isolation. In particular we subdivide the larger of the two elements (as measured in angle subtended from the point of view of the other one).

### 3.4.2 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 [51]

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

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 24 show the use of the anisotropy factor. The approximation as given is not normalized and its directional-hemispherical reflectance is *not* independent of incoming angle. However, its directional-hemispherical reflectance is bounded by 1 which suffices to avoid energy gain.

We chose this model since it is based on a well established theoretical model (Beckmann/Smith) but easy to implement and evaluate (due to the Padé approximation). Physically however this model is still unsatisfying since its parameters are phenomenological and its directional-hemispherical reflectance is dependent on the

incoming ray direction. Clearly better models would be desirable for use in computer graphics. We point out that nothing in the algorithm as we implemented it actually depends on this particular BRDF and it could easily be exchanged for another.

### 3.4.3 Oracle

The oracle is implemented as a polynomial estimator which tests whether the kernel over the support of the basis functions in question is almost a polynomial of order $M-1$. Given the samples used in the quadrature for the coupling coefficients we construct an interpolating polynomial with Neville's method [64] (see Appendix B for implementation details). Taking a set of samples of both the actual kernel and the interpolating polynomial we estimate the distance between the two. Using this kind of estimator is straightforward and independent of the BRDF actually used. However, since we are point sampling the kernel function we may miss important features in it. In the case of radiosity this was not an issue since the reflectance was uniform in all directions. For radiance the reflectance properties can become very peaked and it is easy to miss the all important peak of such a distribution. Aupperle [6] first made this observation and used a geometric analysis over a given transport triple to bound individual terms in his shading model. We adopt a similar approach. As in Aupperle we use geometric analysis to put interval bounds on incoming and outgoing angles. If a spatially varying anisotropy factor ($a(w)$) is included we must also bound the range of angles of the preferred direction ("the scratches") over the reflector. Instead of "hardwiring" the analysis for a given shading model as Aupperle did we proceed by directly applying an interval evaluation of our BRDF to bound the total range of values taken on [46]. Let $\Delta t$, $\Delta v$, $\Delta v'$, and $\Delta w$ be the individual interval bounds. Using interval arithmetic we get

$$\Delta f_r = f_r(\Delta t, \Delta v, \Delta v', \Delta w)$$

$\Delta f_r = (f_{r_{\min}}, f_{r_{\max}})$ gives the min and max 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 $\gamma$ be this error 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 successful in finding the peak of a BRDF and insuring the sufficient subdivision occurs around the peak. Figure 20 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 24. Notice in particular how the mesh follows the high anisotropy gradients of the reflection (most noticeable in the green part).

### 3.4.4 `Accept`

The function `Accept` is similar to its counterpart in the case of radiosity. It uses a product of kernel error and power (brightness refinement). As before we can employ multi-gridding by reducing the error criterion on successive iterations. Radiosity computed a view independent solution. In the case of radiance this is much too expensive in terms of both storage and time. We have therefore used importance driven refinement as well.

As pointed out earlier importance can be treated as just another color channel (our importance is "gray") and we have implemented it in this way. Each scene contains one special quadrilateral which models our virtual CCD camera. It is $1cm^2$ and the only source of importance. The final image is generated by evaluating all outgoing radiance basis functions connected to this quadrilateral. It is also possible to assign importance emission values to any other primitive in the scene, but we have not done so at this point.

### 3.4.5 `PreferredSubdivision`

If the error over a given interaction is found to be too large the constituent pair with the larger angle subtended is subdivided. There are four possibilities. $A_x$ or $A_z$ may

be too large from the point of view of $A_y$, or $A_y$ may be too large from the point of view of either $A_x$ or $A_z$. In each case a unique decision is reached (see the discussion above considering the creation of only four children) just as was done in Aupperle and Hanrahan [6].

### 3.4.6 `Gather` and `PushPull`

The iterative solver is implemented as before. Given a set of links (iterations) we move radiance from one end of the interaction (basis $ij$) to the other (basis $mn$), multiplying it with the couplings stored in the link. Each such iteration, which starts at the roots of all the basis hierarchies, is followed by a `PushPull` up and down the basis function hierarchies. The only difference to the function `PushPull` in the radiosity system is the fact that the radiance `PushPull` extends over four dimensions, not two.

### 3.4.7 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.

## 3.5 Results

The above algorithm has been implemented as part of our rendering testbed [28, 54, 24, 67]. 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 17. 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

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

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

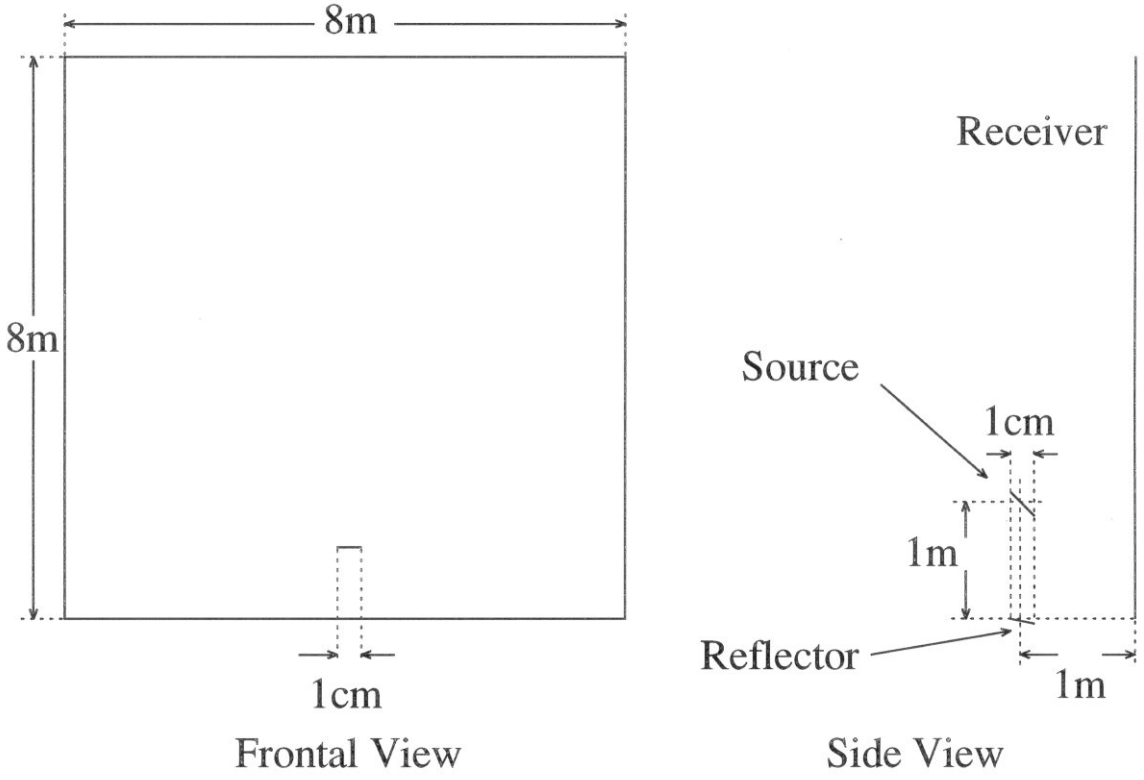



Figure 17: Geometry of small source/reflector test configuration. The source is inclined 45° to avoid any direct illumination of the receiver wall. The reflector is inclined 15° to cause a significant intersection between the directional lobe of the BRDF and the receiver wall.

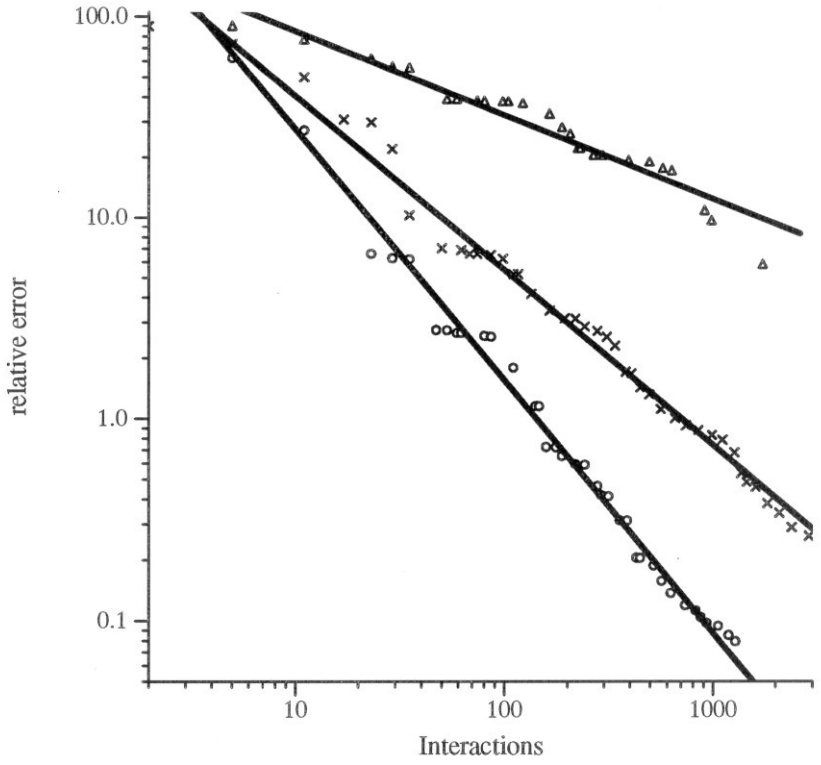



Figure 18: Relative $L^1$ error for the small source/reflector configuration as a function of number of interactions allocated by the oracle and used for transport. From top to bottom we plotted multi-wavelets of 1, 2, and 3 vanishing moments.

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 18 was 0.01.

Figure 18 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

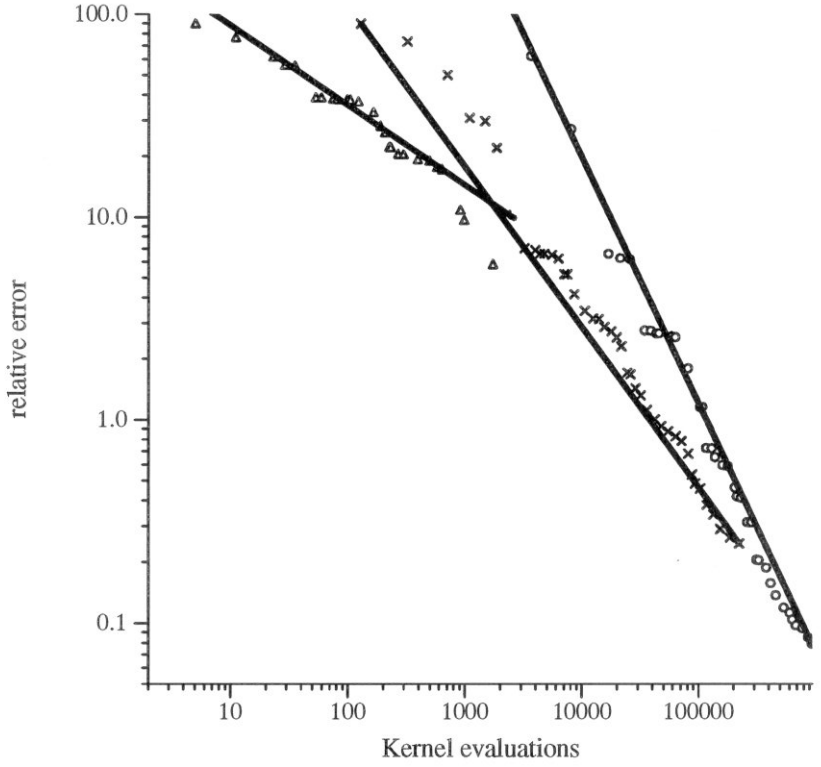



Figure 19: Relative $L^1$ error for the test configuration as a function of number of kernel evaluations. From top to bottom (left to right) we plotted multi-wavelets of 1, 2, and 3 vanishing moments.

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.

The plot shows clearly the convergence behavior of the three different types of bases. The data points have been least squares 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 19 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 20 (frontal views of the receiver wall) 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 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 21 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 22. 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

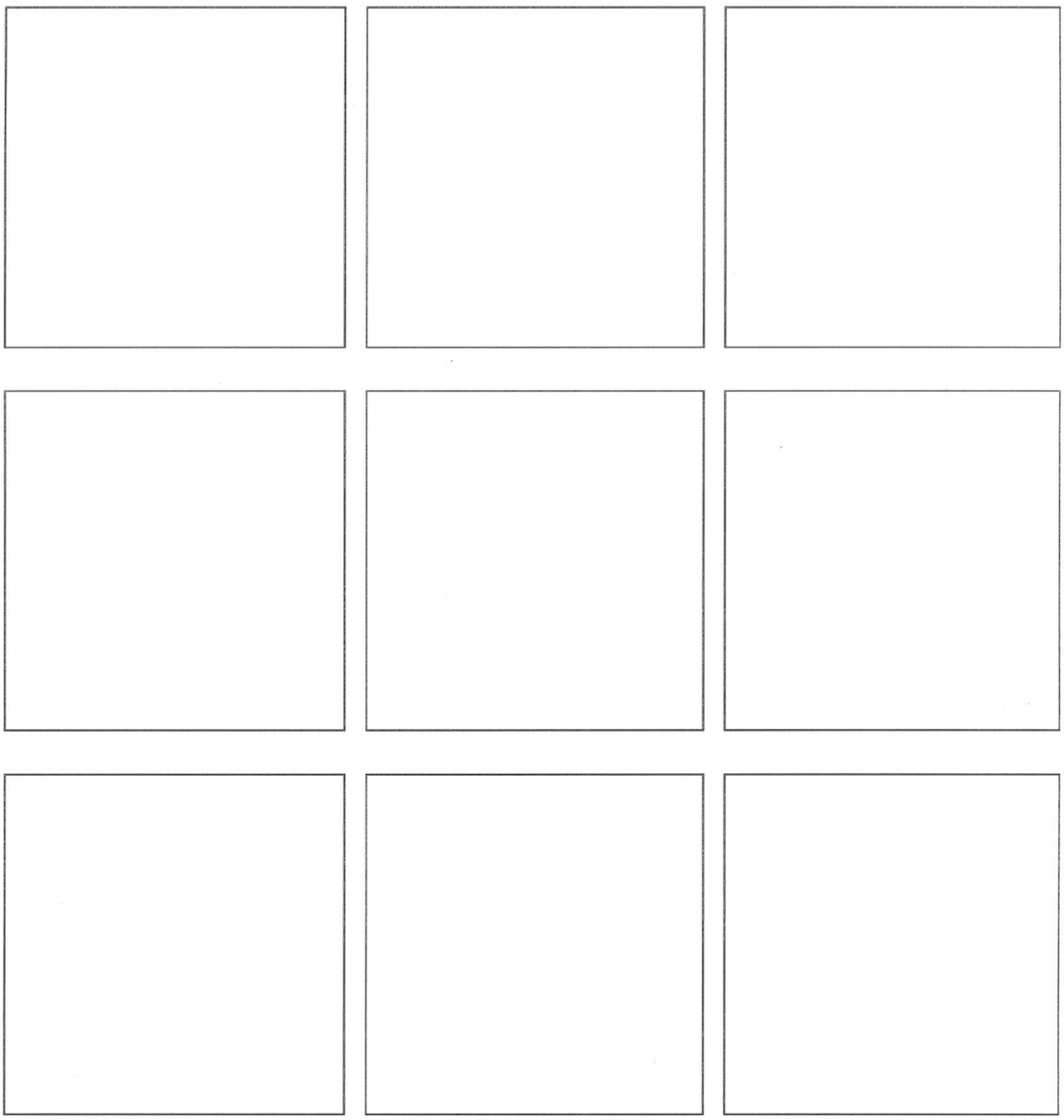
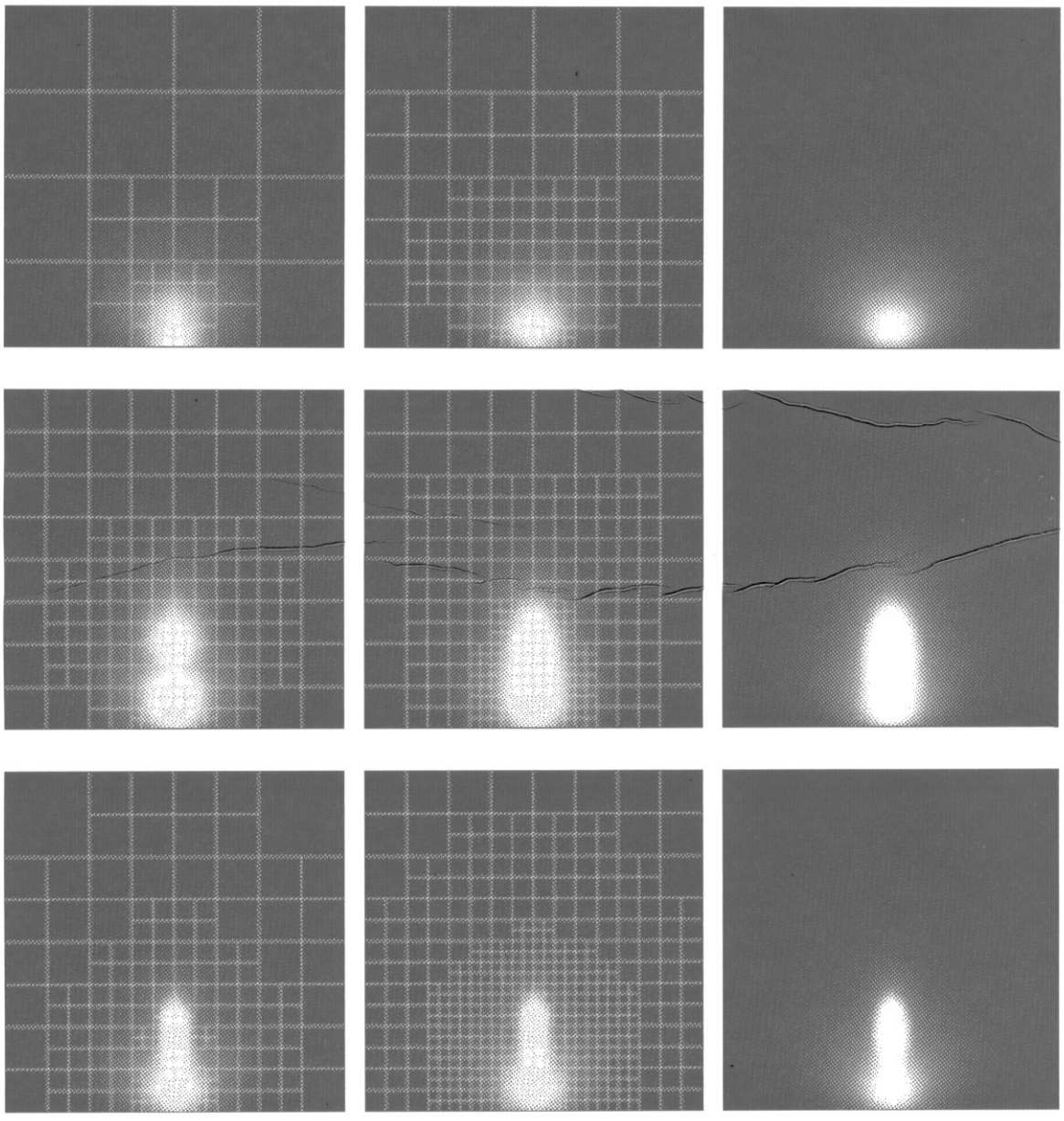



Figure 20: Images of small source/reflector test configuration for different roughness values ($r = 1, 0.01, 0.001$; top to bottom) and different error criteria (10%, 1%, final accuracy; left to right).


decreasing error

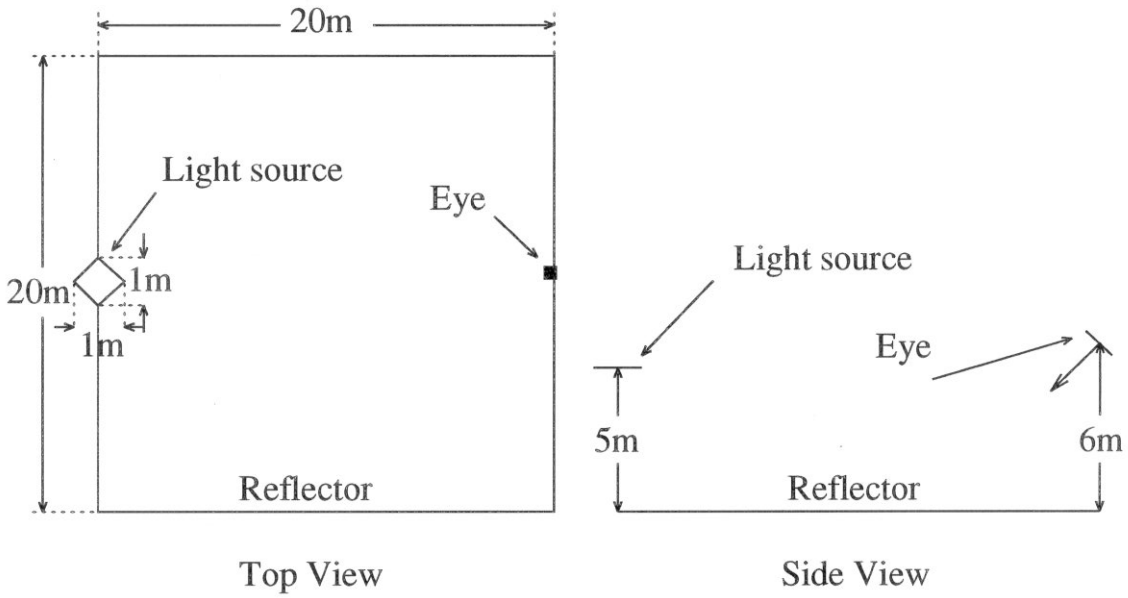



Figure 21: Geometry of large area source/reflector configuration. The source is parallel to the reflector at the far end, while the eye is inclined by 45° towards the reflector.

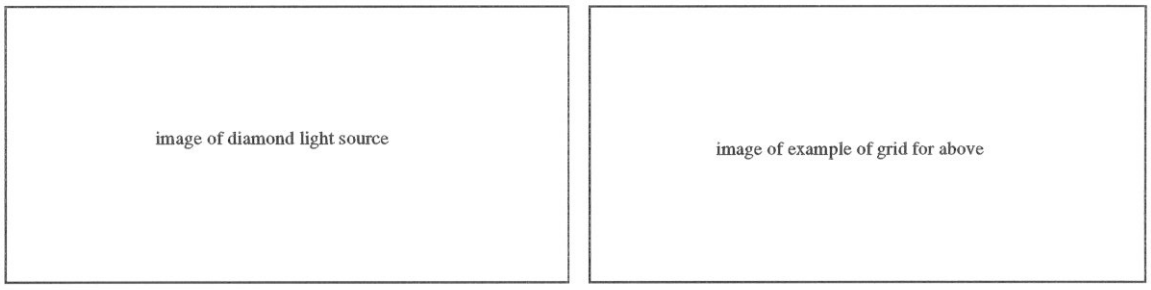
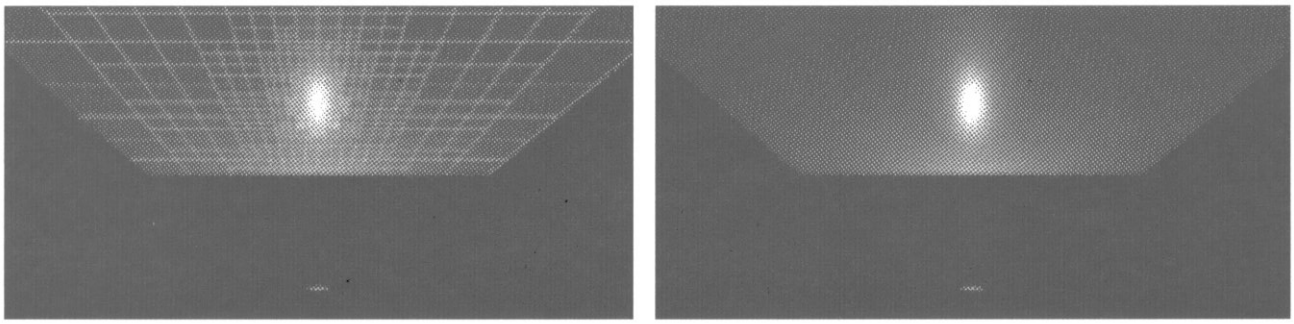



Figure 22: Images of large area source/reflector test configuration. On the left the radiance as seen from the eye; on the right the induced meshing ($r = 0.01$; linear basis functions).



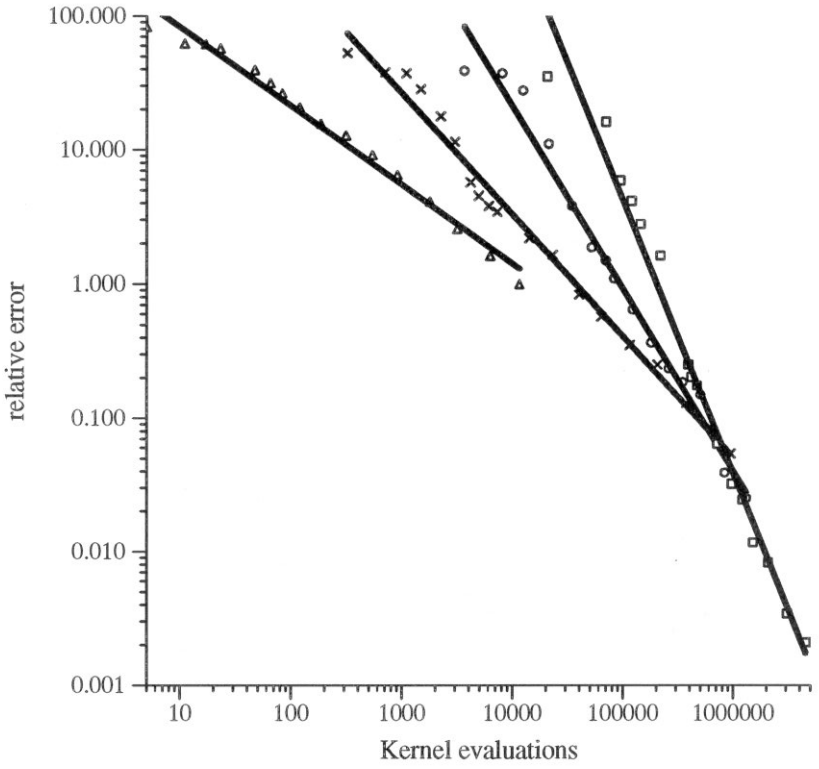



Figure 23: Relative $L^1$ error for the configuration in Figure 21 as a function of kernel evaluations. From left to right we plotted multi-wavelets of $m = 1, 2, 3, 4$ vanishing moments.

our computed solution. The convergence results are plotted in Figure 23 for multi-wavelets of $m = 1, 2, 3, 4$ vanishing moments respectively. Once again we can see the characteristic convergence rates (fitted lines).

Schlick's BRDF [51] also allows for an anisotropy factor, which we took advantage of in Figure 24. It uses 3 colored light sources to show the resulting reflections. Figure 24 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). Using this anisotropy factor which was spatially varying further shows that the refinement

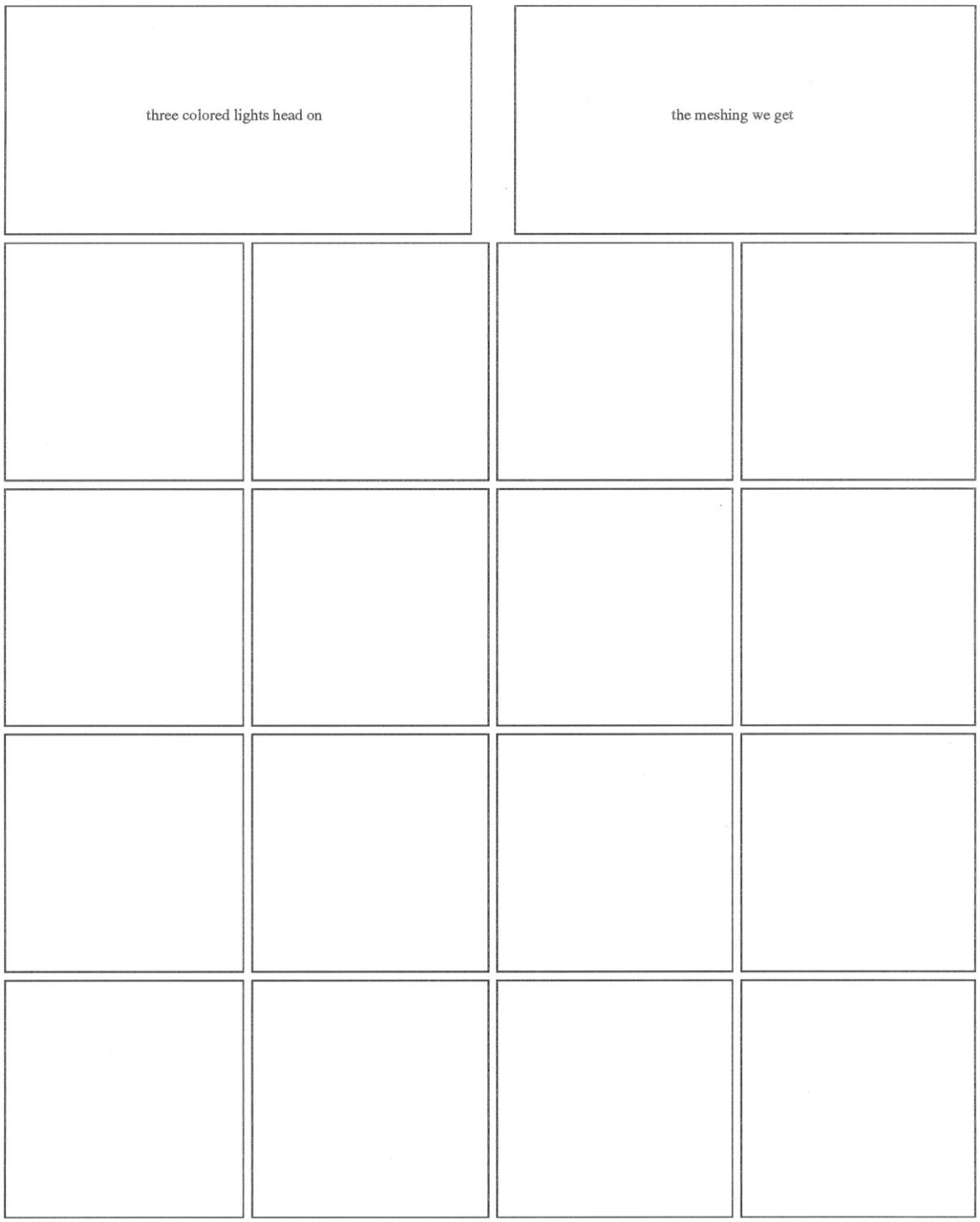
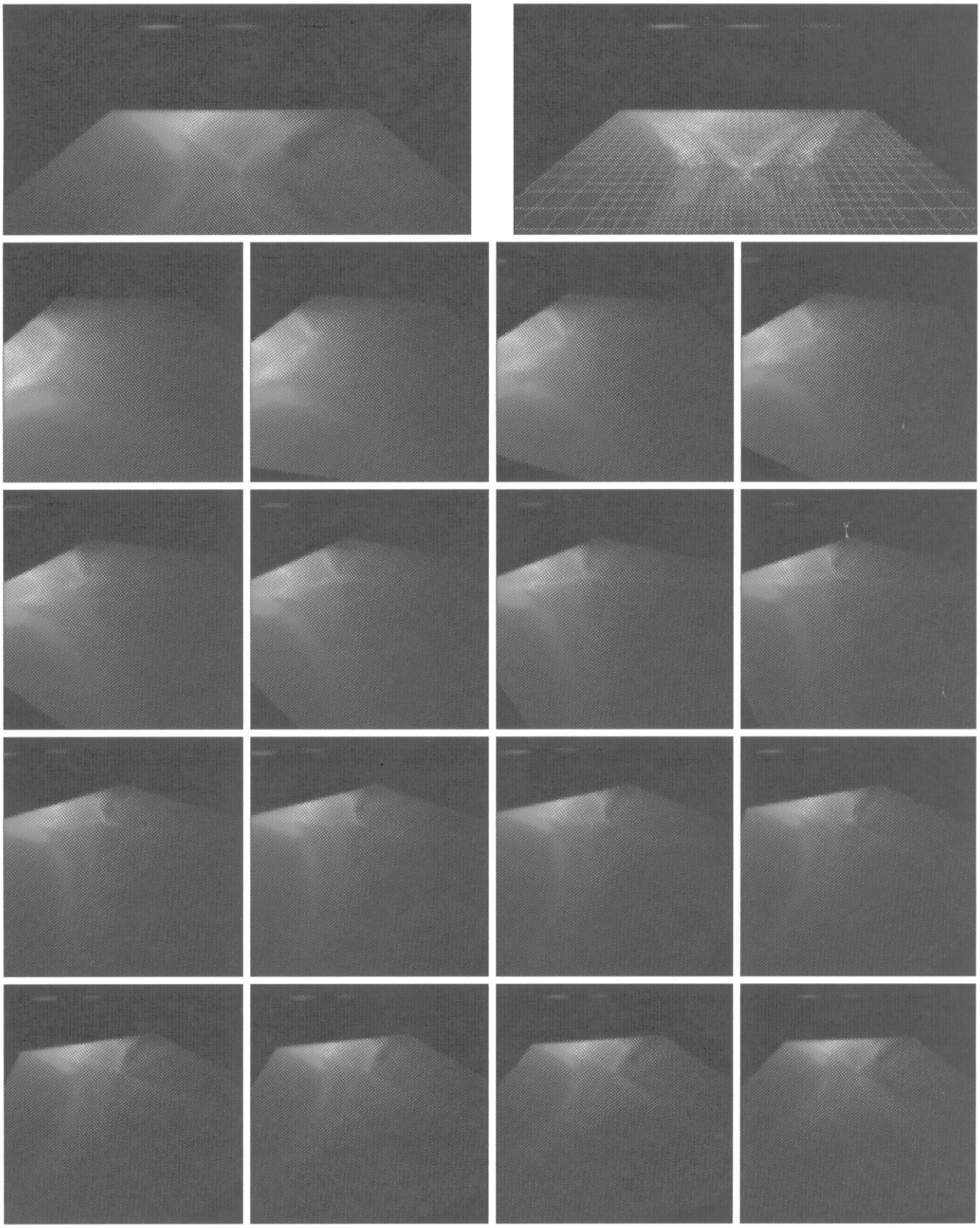



Figure 24: Top left: configuration with 3 light sources and an anisotropic reflector; top right: the induced meshing. Below is a sequence of views rotating about the center. Note the moving highlights ($r = 0.1$,$p = 0.1$; linear basis functions).



oracle captures the important details reliably. The sequence of images in Figure 24 shows a spatial plot of the resulting radiance field by rotating the eye 90° around the model. Notice the motion of highlights above the floor as is characteristic for glossy reflectors.

Figure 25 shows one of the typical effects when simulating glossy reflectors. The floor is glossy while the far wall is modeled as a diffuse, textured reflector. Depending on the glossiness of the floor the reflection is more or less "smeared" out. In the upper left is the diffuse case ($r = 1$) while in the lower right a highly specular case is seen with ($r = 1/2048$). To characterize the "peakiness" of the latter value we can relate it to the angle about the mirror direction at which the value of the BRDF has fallen off to 1/2 peak: 0.82°. For such peaked reflection responses a finite element technique as we employed it is probably not competitive with a raytracing approach. However, it served as a stress test for our code and was meant to probe the limits of the numerics.

We next considered a scene in which multiple glossy bounces occur. Figure 26 shows nine multi-gridding refinement steps (i.e., each image from top left to bottom right represents a decrease of the error criterion by $\sqrt{2}$ followed by refinement and an iteration of the operator) of a simulation involving multiple glossy reflection. It contains a single area light source (diameter 25cm) above and to the right of the viewer's eye position. The walls are modeled as pure glossy (i.e., metallic) reflectors with $r = 0.03125$, corresponding to a halfangle spread on the peak of the BRDF of 6.6°. All other surface elements are pure diffuse reflectors.

The highlights on the left and right are paths involving one glossy reflection between the light source and the eye. The blurred images of the floor in the metallic walls correspond to a diffuse followed by a glossy reflection. Finally there is a fainter highlight on the left wall close to the corner. This corresponds to a double glossy reflection (light to right wall to left wall towards the eye). Table 27 gives some of the relevant statistics for this scene. The basis functions used were linear (2 vanishing moments). All timings were performed on a 150MHz R4400 SGI computer. Note that the times are dominated by glossy link refinement as it is the most expensive part of the computation. There is also a rhythm noticeable in the refinement steps. Every

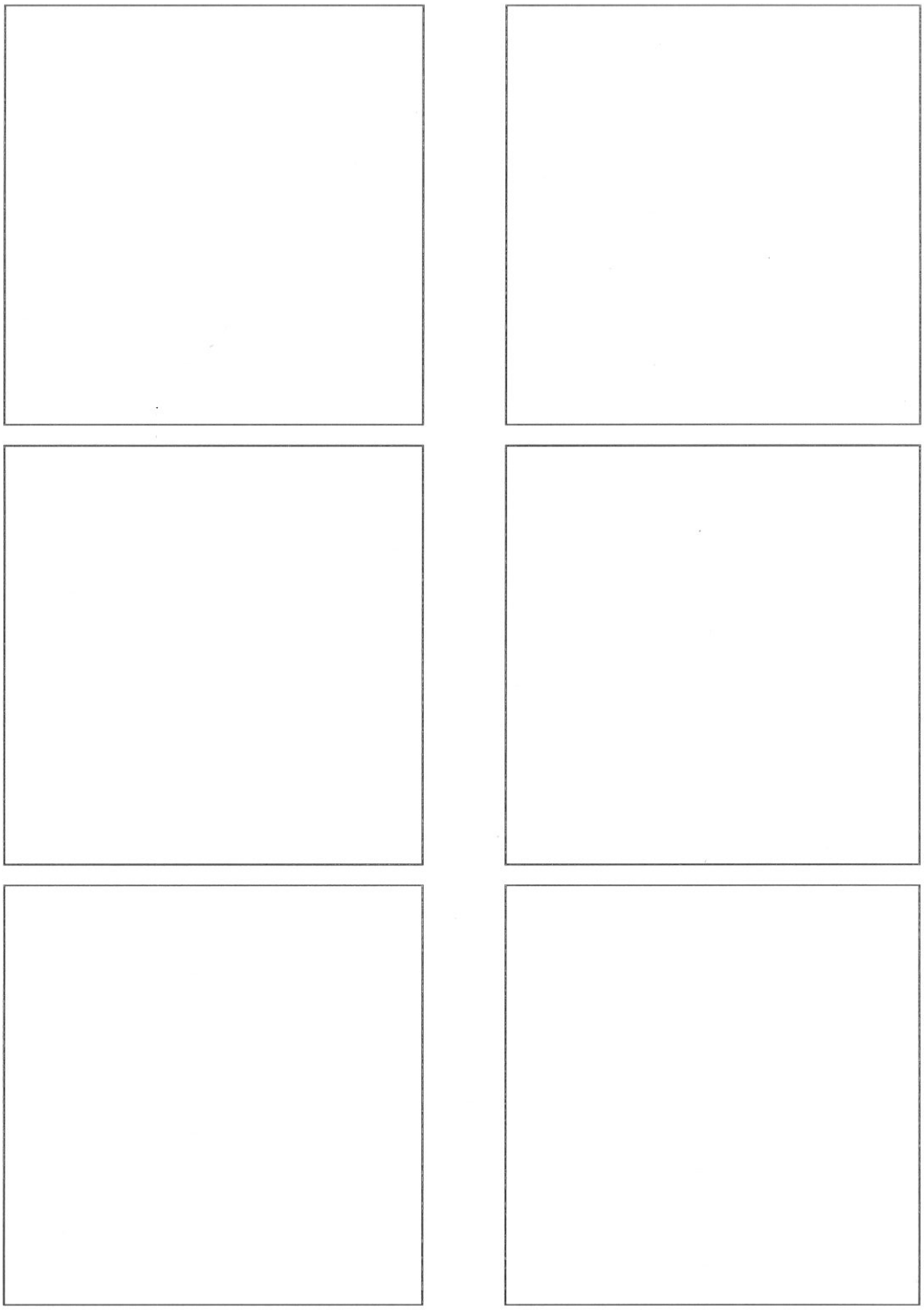
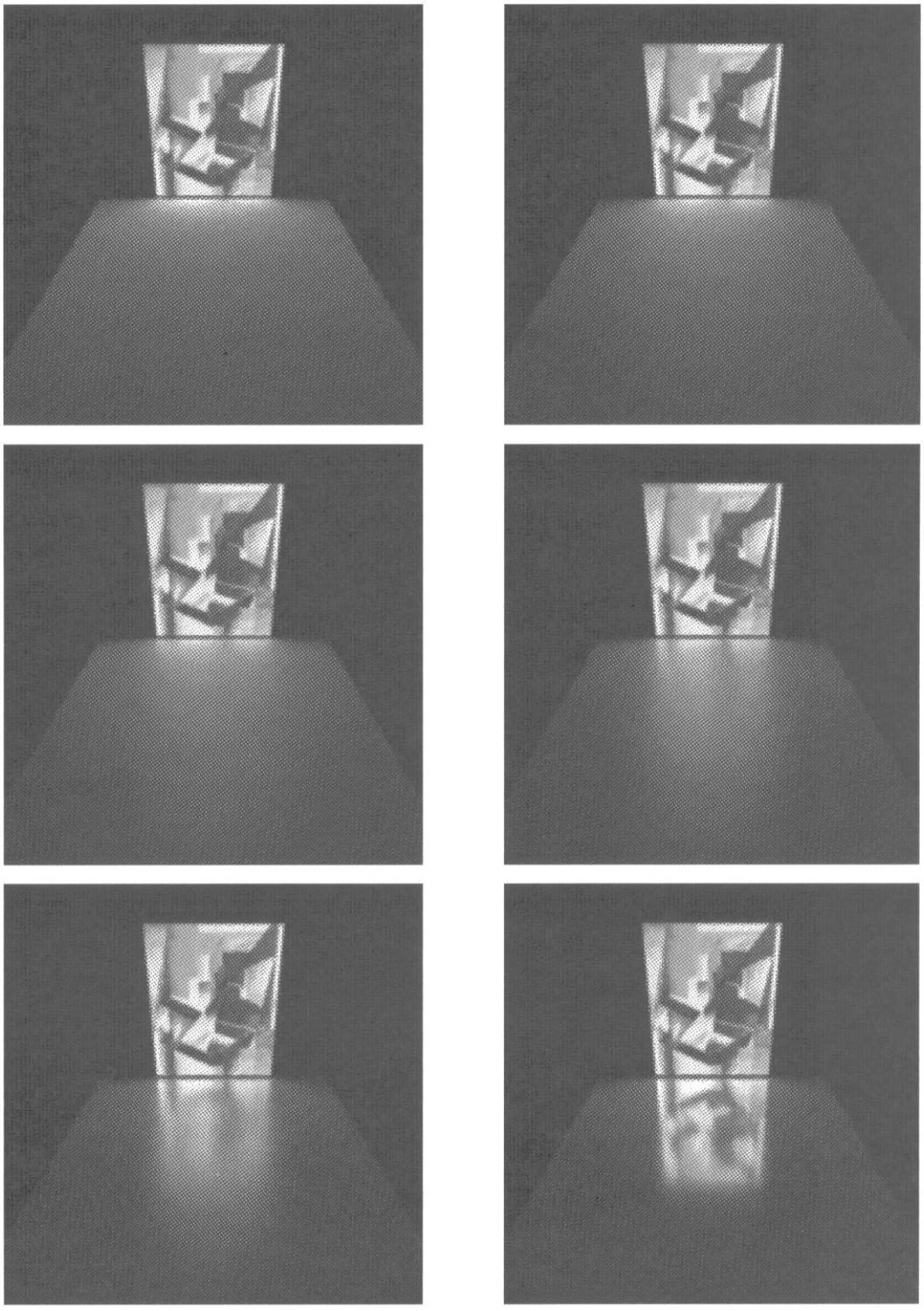


Figure 25: "Smearing out" due to a glossy reflector. From left to right, top to bottom the floor has an $r$ value of 1, 1/4, 1/8, 1/32, 1/128, and 1/2048.



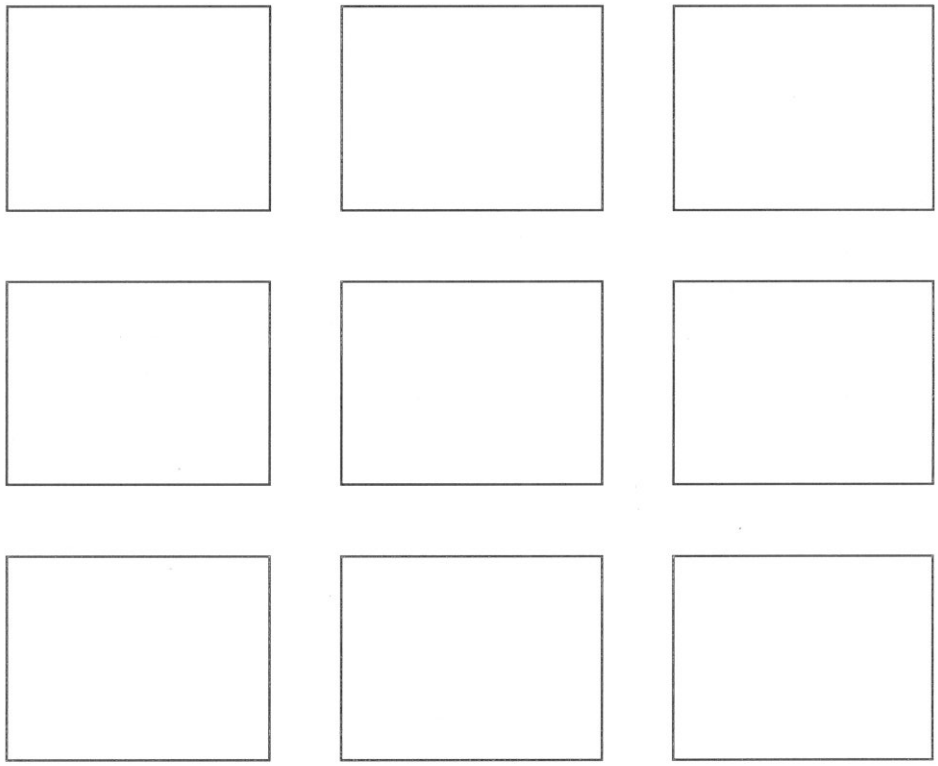
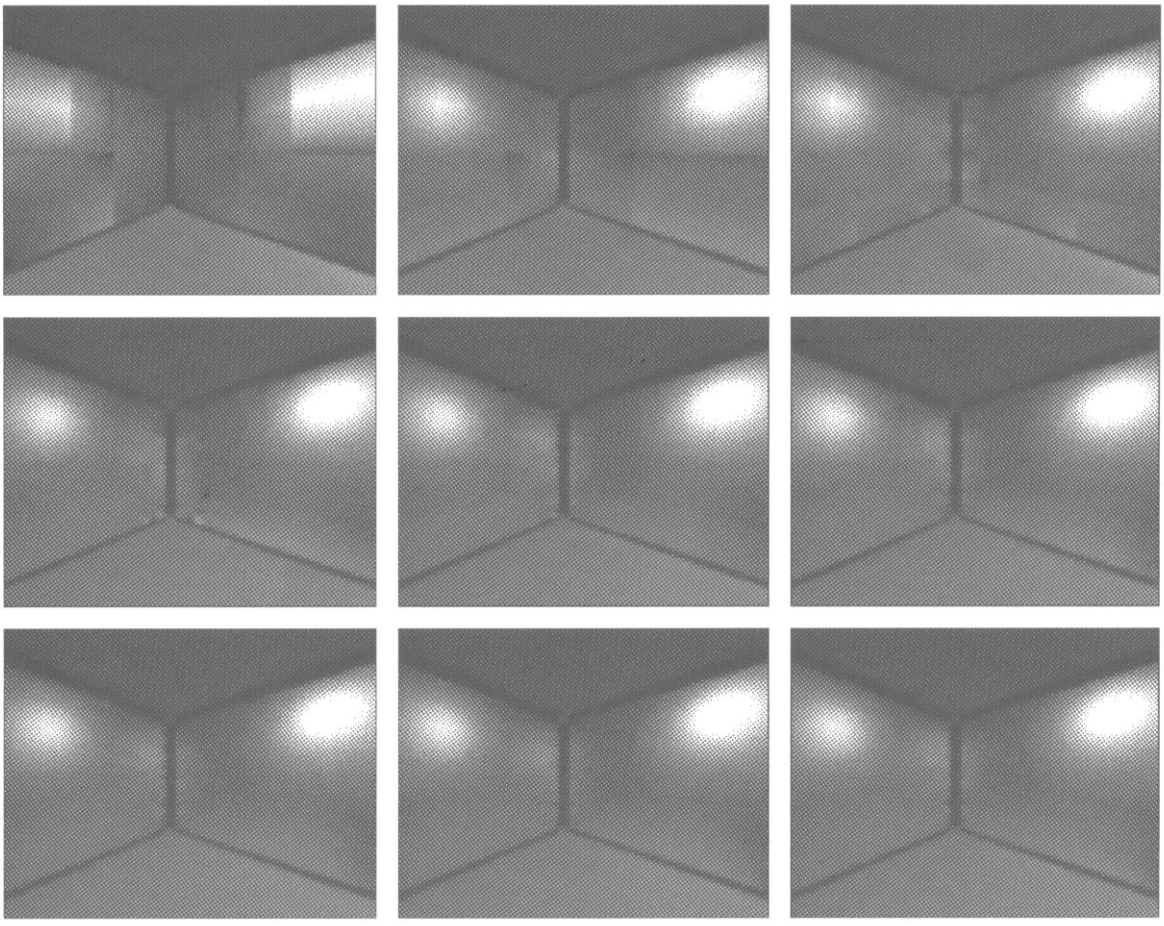


Figure 26: These images show a multi-gridding refinement of a simple scene with multiple glossy reflections.

| iteration | radiosity coefficients | radiance coefficients | interaction triples | $\Delta$ time |
|---|---|---|---|---|
| 1 | 56 | 282 | 462 | 2m |
| 2 | 96 | 549 | 1320 | 7m |
| 3 | 160 | 1092 | 2811 | 12m |
| 4 | 260 | 2307 | 6573 | 29m |
| 5 | 360 | 3570 | 11040 | 35m |
| 6 | 476 | 6873 | 22206 | 85m |
| 7 | 612 | 9489 | 32040 | 74m |
| 8 | 1024 | 16902 | 58524 | 188m |
| 9 | 1276 | 20034 | 71997 | 104m |

Figure 27: Statistics for Figure 26. For linear basis functions each radiosity coefficient consists of 4 floats, while each radiance coefficient holds 16 floats.



second refinement step has a marked increase in cost followed by a refinement step of similar cost. This is due to the length of the transport paths over which refinement occurs (e.g., the double glossy bounce across the two glossy walls).

Finally Figure 28 shows the same environment with a colored cube added to create more complicated transport paths and force some partial visibility refinement. The front and right side of the cube are diffuse, while the left and back side (neither one directly visible) are pure glossy reflectors of blue and green color respectively. The top of the cube is a white pure glossy reflector. The far wall now shows some green. This is due to a path involving a triple glossy transport originating at the light, bouncing off the far wall, the back of the cube, off the far wall again, and finally into the observer's eye. Similarly the blue reflection on the left wall a triple glossy bounce from the light source, across the left wall, onto the left side of the cube, back onto the left wall and finally into the eye (see the diagram at the bottom of Figure 28). Note that the light source placement was chosen such that the backsides of the cube do not receive any direct light. The image shown in Figure 28 used linear basis functions, and had 2725 radiosity coefficients (each 4 floats), 63959 radiance coefficients (each 16 floats), and 272433 transport triples. Total compute time for 11 multi-gridding refinement steps (similar to the refinement shown in Figure 26) on a 50MHz R4000 SGI computer was 50 hours.

### 3.5.1 Performance Issues

We have profiled the current wavelet radiance implementation to understand better where it spends its time. All percentages we give in the following discussion are for radiance with linear basis functions and an environment containing significant occlusion.

About 20% of all cycles are spent in the polynomial interpolator oracle. This is the single most expensive function which has room for improvement. In the case of constant basis functions the original HR code used a simple point to disk form factor approximation, whose evaluation cost was very low. In the case of radiosity going to a polynomial estimator oracle for higher order basis functions was only a modest cost

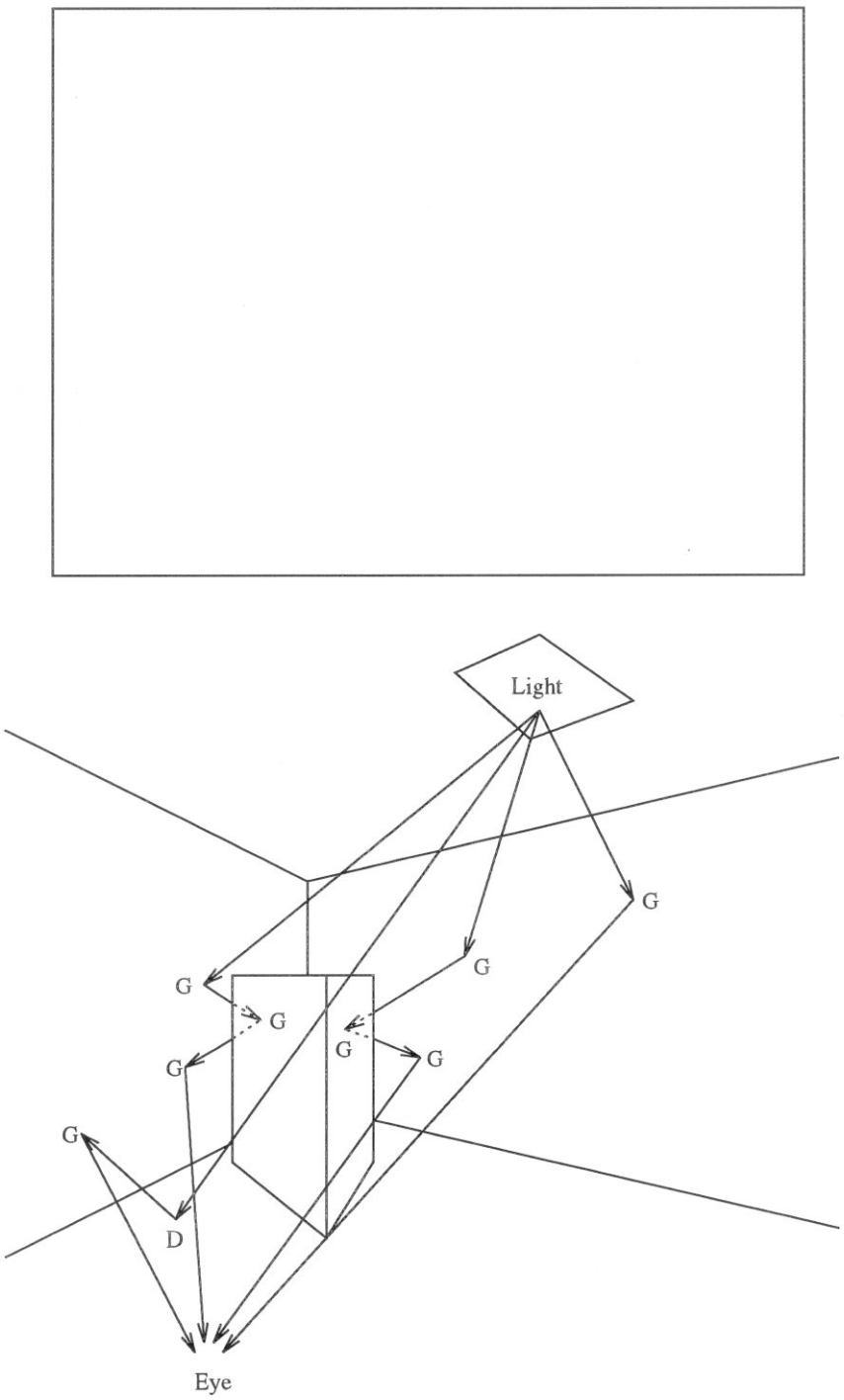
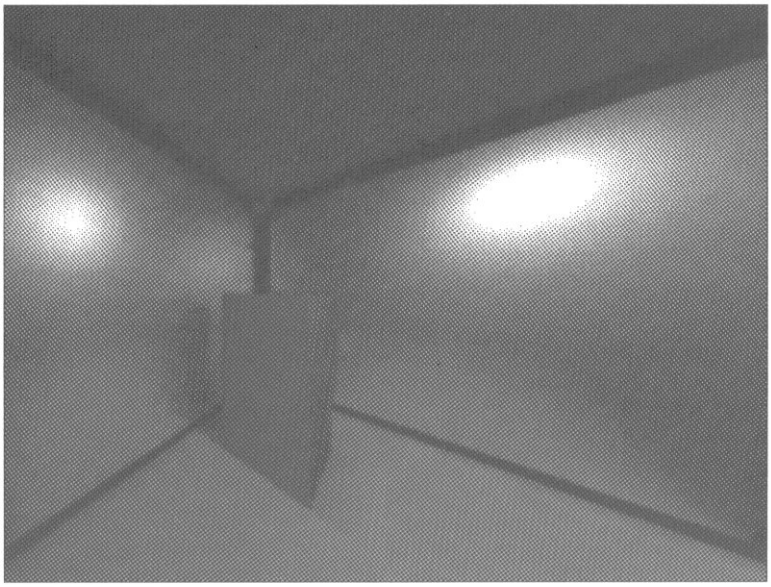
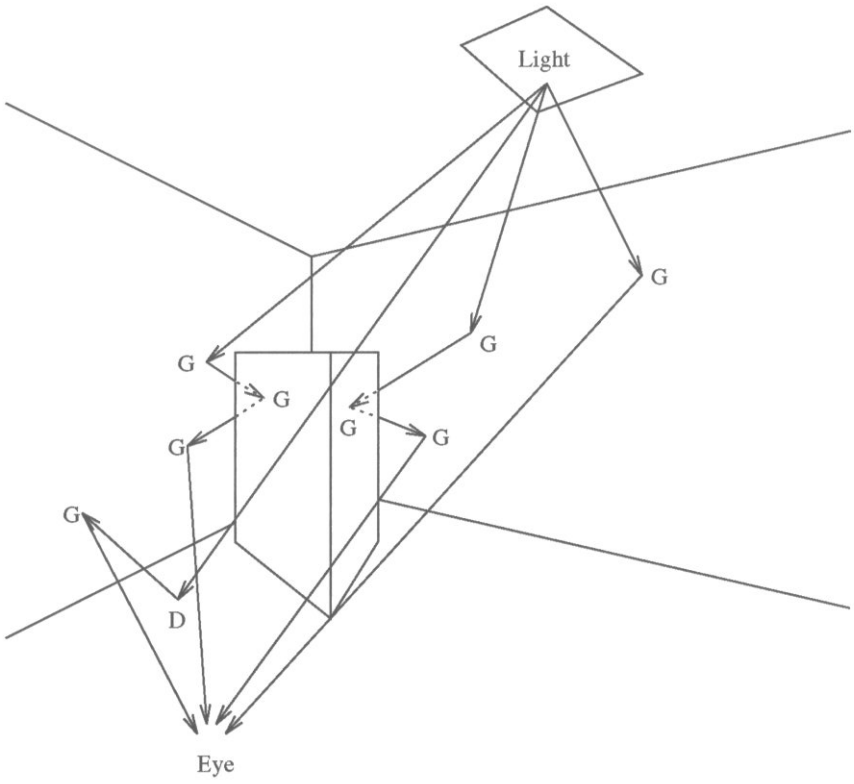



Figure 28: A more complicated environment with glossy reflectors and significant interreflection as shown in in the diagram. The letter D indicates a diffuse bounce, while G indicates a glossy bounce. Note the paths involving multiple glossy bounces.




Light
G
G
G
G
G
G
G
G
G
D
Eye

increase. This is not so for radiance. Because the kernel is 6 dimensional the cost of a polynomial estimator increase by a 6th power of the number of vanishing moments as well. For example, going from 2 samples to 3 samples in each dimension increases the cost per polynomial estimator evaluation by a factor of $\approx 11.4$. In contrast, we hypothesize that an oracle based on powers of angle subtended could be competitive with the original hierarchical radiosity/radiance oracle.

Another costly part of the overall computation is given by visibility tests. These accounted for about 25% of all cycles in our profiled runs. This cost is more difficult to decrease since visibility tests will continue to be important and the current implementation already uses a fairly optimized visibility oracle due to Teller (based on work reported in [67]).

Some other observations we made are more qualitative, in that they are based on our (and our colleagues) subjective experience in using the system. For example, the use of memory in the current system is problematic. Our current implementation was designed to facilitate experimentation with bases of different orders (constant through cubic). Consequently all coefficient and interaction (i.e., the $T_{ijmn}$) data structures are built dynamically. In the case of linear basis functions and radiance this yields a total of 256 floating point numbers per link with a pointer overhead of 254 (to allow C indexing into dynamically sized arrays). If instead we built a system which "hardwired" linear basis functions, pointer overhead would disappear, resulting in roughly half the memory requirements for a link. For cubic basis functions a single link contains 65536 $T_{ijmn}$ coefficients and a similar amount of pointer overhead! In the case of radiosity this cost scaling was not nearly as dramatic affording us the luxury of a fairly general implementation.

The use of tree wavelets is also more problematic in the case of radiance. Tree wavelets, especially when no discontinuity meshing is employed, require fairly fine subdivision to capture details of the resulting radiance and to avoid objectionable artifacts. In the case of radiosity the extra cost of higher order basis functions and finer meshing was not as dramatic, allowing us to generate images with little perceptible error at reasonable cost. Once again, the more dramatic cost scaling of radiance makes this approach less feasible and we feel that the use of non-tree wavelets and/or

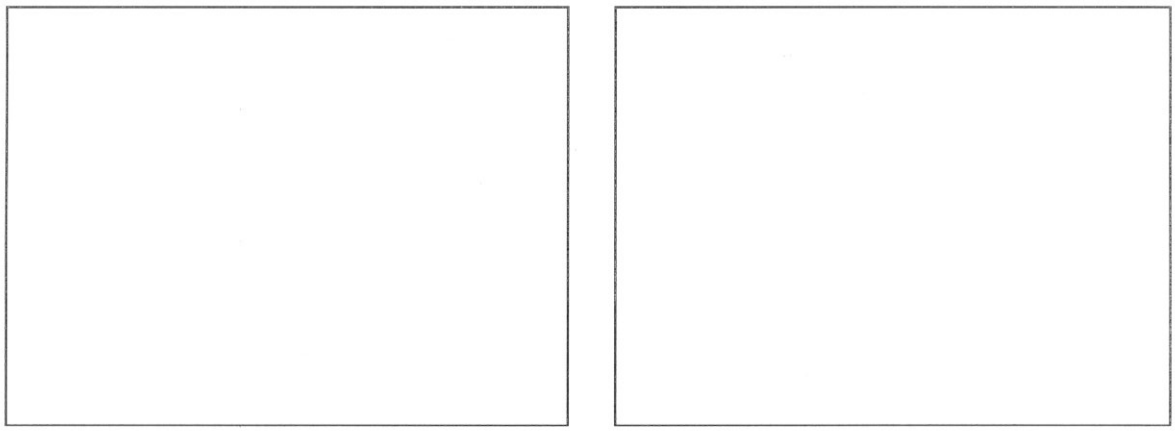


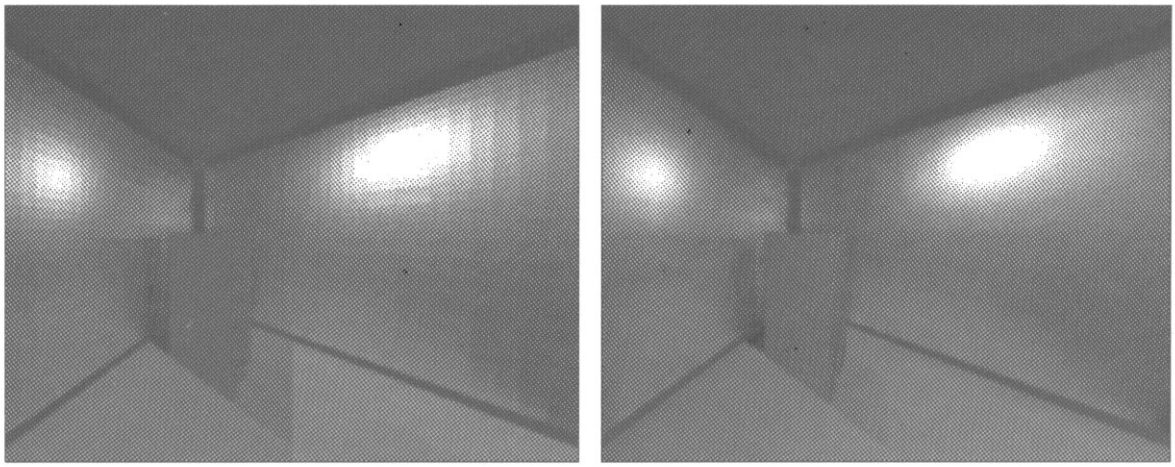
Figure 29: The two images compare the solution computed with constant (left) and linear basis functions (right). Both have been refined so as to have equal cost (as counted by number of kernel evaluations).

discontinuity meshing will lead to large efficiency gains.

## 3.6 Summary and Discussion

In this chapter 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 for which we have independent means of verifying the answers. The rate of convergence is consistent with the theoretically predicted rate and the benefits of higher bases (i.e., smoother answers and smaller error as a function of work) have been realized.

As was done in the earlier work concerning WR [28, 54] we have in effect extended the hierarchical technique to higher orders: in the case of radiosity the work reported in [32] and in the case of radiance the work reported in [6]. The benefits are similar. The quality of the computed images increases markedly when going from piecewise constant bases to piecewise linear (see Figure 29 for two solutions computed at the same cost). 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 (see [28, 54]). In the case of radiance this tradeoff is not as clear. Higher order basis functions only become competitive at very high precision requirements (see Figures 19, 23). In practice we have found that the cost increase for linear basis functions is not too high while yielding numerically more accurate as well as better looking pictures. So far we have employed quadratic and cubic 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.

More complicated environments remain challenging for our current algorithm implementation. For example, there are significant aliasing problems due to the regular point sampling nature of the quadrature rules. The images in Figures 26 and 28 used only linear basis functions, nonetheless we employed three point Gaussian quadrature. A two point rule is theoretically sufficient, but in practice it led to too much aliasing for these configurations. Some traces of this aliasing problem are still visible near the far corner of the room (Figure 26).

# Chapter 4

# Conclusion

## 4.1 Summary

In this dissertation we have described and analyzed a class of algorithms based on HR and its extension to radiance. The first part of the thesis was devoted to radiosity while the second treated radiance. We proceeded by analyzing HR and GR and showing that HR is in fact an instance of a Galerkin technique by interpreting HR as equivalent to the use of the Haar basis in a non-standard wavelet based operator realization. Doing so allowed us to apply the analytical tools of wavelets. In particular we showed that radiosity belongs to a class of operators characterized by smooth kernel functions which satisfy a (generalized) Calderon-Zygmund property. Having established this property we were able to use a very general theorem introduced by Beylkin *et al.*[9] to argue that wavelets with vanishing moments can be used to create sparse realizations of the radiosity operator. Analyzing the geometric arguments at the foundation of HR we recognized that they provided a geometric interpretation of the more abstract Calderon-Zygmund operator bounds. This allowed us to apply the geometric insights into the linear bound on the number of necessary coefficients back to the more general Calderon-Zygmund claim.

Implementing this algorithm we found that it in fact realizes more sparsity with increasing number of vanishing moments, leading to solutions which required less work for a given allowable error. In particular the highest order (and most vanishing

moments) basis functions gave us the best error per work. In order to realize the asymptotically linear bound on the number of interactions created we had to design an oracle function for higher order bases, which identifies the important entries in the resulting matrix system. Our polynomial interpolator oracle took the prescription of vanishing moments literally by testing whether the kernel function is a low order polynomial over the support of a given set of basis functions interacting.

We proceeded in the second part to apply the insights into the use of wavelets for the radiosity problem to the more general radiance equation. Radiance allows us to introduce glossy reflectors. The underlying machinery of WR was extended in a straightforward way by using the spatial parameterization of the radiance equation. The shading model of Schlick [51] was employed including the anisotropy factor. The oracle continued to be of the polynomial estimator type, albeit of 6th order now instead of 4th order as in the case of radiosity. A number of test configurations, for which independent means of checking our computed answers existed, were simulated and the results quantitatively analyzed. We found that just as in the case of radiosity higher order basis functions realized smaller and smoother errors. In contrast to radiosity we found however that the cost of quadratures is so much higher for 6D integrals than it is for 4D integrals that higher order basis functions beyond the linear case are hardly competitive on a scale of error per work. Linear basis functions through present a major improvement over the constant case.

In both chapters we discussed a number of implementation issues, some of which may lead us in the future to pursue alternate strategies from the ones explored in this thesis. In the next section we consider some of the issues in particular the contrast between radiosity and radiance as pertains these choices.

## 4.2 Implementation Choices

So far we have only used tree wavelets (i.e., multi-wavelets whose filters $(h, g)$ for neighboring basis functions do not overlap). As a consequence we were able to use simple data structures and could eliminate the use of detail functions $(\psi)$ entirely by effectively replacing them with the equivalent set of smooth functions $(\phi)$. The

resulting basis function systems have the property that neighboring intervals (areas) completely decouple and surface elements have a one-to-one correspondence to the basis functions. This allowed us to easily argue our recursive enumeration scheme for the generation of all important transports. It also means that there are no continuity constraints between neighboring surface elements. At first sight this certainly appears as a disadvantage since the human eye is very sensitive to discontinuities in the computed answers. In practice it only becomes an issue if the error criterion is set so high that these discontinuities are above the display's intensity resolution. For high precision answers the results are satisfactory. However, in many applications the computed answer is judged solely on its "looks", even though it may still contain large amounts of error from a physical point of view. For these applications continuity enforcing wavelets would be preferable.

The usual construction of wavelets in fact produces functions which overlap and thus ensure (variable) degrees of continuity across neighboring basis functions. We have not used such functions yet. In order to use these successfully a number of issues need to be addressed. The recursive enumeration scheme we used would need to be modified. In particular with overlapping wavelets there is no unique identification of surface elements and basis functions anymore. Consequently when using a non-standard realization of the operator an enumeration scheme has to make extra efforts to avoid the multiple accounting of energy and cannot use simple geometric arguments anymore. Further, we took advantage of the fact that for tree wavelets a given detail function can be immediately replaced by its constituent smooth functions (one level finer). This is not as straightforward for overlapping wavelets, since the constituent smooth functions are shared by neighboring wavelets. One avenue to address this would be to use the standard realization of the operator. Since successively finer detail functions are added and no over representation occurs (detail functions at finer levels are linearly independent of those at coarser levels) there is no danger of multiple accounting of energy. However, the enumeration scheme has to ensure that for a given region of detail, hookups throughout the hierarchy with all relevant functions occur (see Figure 6).

Recall that the standard realization requires $O(n \log n)$ interactions. Even though

the asymptotics are worse from the theoretical analysis point of view, it is not clear that the extra factor of $\log n$ makes a noticeable difference. Extensive experiments for flatland radiosity were reported in [54] comparing standard and non-standard realizations. In these experiments no appreciable difference was found between the two approaches. This matches observations by other researchers who compared standard and non-standard realizations [36].

As pointed out earlier one difficulty with the standard realization is the control of error in the quadratures. Since couplings are never replaced but only added, the errors in any couplings at coarser scales have to be reduced at the same rate that more detail functions are added. This requires recomputation of earlier couplings, possibly offsetting the other advantages of the standard realization.

A way to retain the advantages of the non-standard realization with tree wavelets, while addressing the disadvantages of visual discontinuities due to missing overlap, might be to add a final reconstruction step. For example, Lischinski *et al.*[43] used such an approach in the context of discontinuity meshing for HR. After some number of iterations using piecewise constant basis functions they perform a final iteration of the integral operator evaluating it at specifically chosen points to construct interpolating polynomials of higher orders for smooth display. This procedure yields very smooth results, but is somewhat adhoc since the operator is projected into a piecewise constant subspace for purposes of inversion, while the display occurs supposing that the solution comes from a higher order space.

It may be possible to address this issue with the use of bi-orthogonal systems of wavelets. One could choose the primal basis to consist of piecewise constant functions while using a dual hierarchy which has much higher regularity [15]. By arranging the primal basis functions to be the ones under the integral operator we would have a simple solution method using only piecewise constant functions. The `PushPull` function would be used to switch from dual back to primal hierarchies after each iteration (independent of the operator realization we choose). The final display would occur using the dual functions which can be constructed to have arbitrary regularity.

In the case of radiosity the cost of quadratures was not too high and the fact that only diffuse reflection occurs led to fairly stable quadratures. For radiance we found

aliasing due to the regular nature of samples of the integrand taken by the quadrature routine, to be a problem. At the same time increasing the number of samples in our Gaussian product quadrature leads to explosive growth in the cost due to the 6th order scaling. Gaussian quadrature rules are also very sensitive to discontinuities as they arise due to visibility changes. The power of a Gaussian quadrature rule comes from its high polynomial order. If the integrand however is not a polynomial the error in the integral estimate can become rather large. Taken together, we can see that although Gaussian quadratures are asymptotically optimal they can suffer from high sensitivity to discontinuities, and systematic (aliasing) errors due to their regular sampling nature. The fact that we use product rules leads to a 4th order (radiosity) resp. 6th order (radiance) scaling of the cost. There may be other quadrature rules which do not suffer from these problems and additionally exhibit a less dramatic cost scaling. Since quadratures are one of the main expenses in our algorithms this is an important issue that needs to be explored further.

Throughout we have not addressed questions of discontinuity. Recall that the kernel function contains the visibility function v(.,.) which is discontinuous. Employing the wavelet arguments for Calderon-Zygmund operators we had to argue the smoothness of the kernel function to argue the asymptotic complexity of our algorithm. In our case the kernel function (both for radiosity as well as radiance) is only piecewise smooth. Our current implementation relies on the subdivision oracle to capture these discontinuities. Whenever two elements interact across a discontinuity the polynomial estimator will find that the kernel is not well approximated by a polynomial and cause subdivision. In this way any discontinuity is handled the same way as the pole in the kernel function. Lischinski *et al.*[43] have demonstrated in the case of radiosity that significant economies can be gained from the use of discontinuity meshing (i.e., aligning the subdivision boundaries with discontinuity features). They did so for the case of constant basis functions. To extend their approach to higher orders we need to construct multi-wavelets on triangular basis elements with the added degree of freedom to choose the subdivision point freely, rather than be forced to use the midpoint. This is a trivial if tedious exercise and we give the basis functions and two scale relation for the case of linear triangular elements in Appendix A.

## 4.3 Future Directions

The radiance problem remains hard and costly to compute. Hierarchical methods have clear advantages over naive finite element approaches and we have seen that increasing the order of the elements in a hierarchical scheme leads to further improvements. Without these efficiencies the problem would hardly be tractable. With these efficiencies it remains costly and further improvements are needed. We already suggested some implementation choices above which would lead to interesting investigations to understand the various tradeoffs.

The most pressing problem at this point is the cubic dependence of the overall algorithm on the number of input surfaces. This problem can be alleviated with a clustering algorithm, which groups surfaces into larger ensembles. First steps in this direction have recently been taken [59, 38, 62]. We are confident that it will be possible to address this problem adequately in the future.

As other researchers move into the field of wavelet algorithms for the radiance problem opportunities for comparison between different approaches appear. For example, our implementation uses the spatial parameterization. In contrast the algorithm published by Christensen *et al.*[13] uses the directional parameterization. While we use the non-standard realization, they use the standard realization. At this point comparisons are still difficult, but we expect to learn more about the various tradeoff between these different approaches.

In our implementation we have limited ourselves to planar quadrilaterals rather than allowing more general surfaces such as bi-cubic patches. Clearly extending the set of primitives would be desirable. The mathematical formulation as given in this dissertation continues to be valid and we hypothesize that visibility determination is the main problem in moving to more general primitives. As pointed out above, going to triangular domains is not difficult from the point of view of basis construction. This too would further increase the set of primitives allowable.

The BRDF used in this dissertation is a very efficient (Pade) approximation to a model derived from physical principles. Nonetheless it is still not completely physically realistic and it would be desirable to remedy this by employing physically

realistic BRDF models. More fundamentally, we would like to permit very general reflection functions. This would give users of such a system more freedom in matching real materials not just qualitatively but quantitatively, clearly a step necessary if we hope to reach our goal of photo realistic rendering.

We suspect that to reach the ultimate goal of an efficient, photo-realistic renderer, a range of techniques will need to be combined. For example, it seems clear that there are regimes, such as the diffuse case, in which finite element techniques are very efficient, while others, such as pure mirror reflection, are clearly more efficiently modeled with raytracing techniques. Efficient algorithms will need to be able to unify these techniques and understand how to switch between them. We do believe that wavelet approaches will have a solid place amongst the techniques used to perform illumination computations.

# Appendix A

# Linear Multi-wavelets on Triangular Domains

Heckbert [33], in his examination of Galerkin methods for Flatland radiosity, showed that considerably more accurate solutions can be computed when using discontinuity meshing. Recently, Lischinski *et al.*[43] demonstrated a similar result for 3D radiosity using a HR algorithm in which the subdivisions of surfaces are chosen to align with discontinuities in the radiosity function. Such discontinuities occur for example along shadow boundaries. In order to implement discontinuity meshing for higher orders we must satisfy two requirements. First, the elements must not be constrained to be rectangular since even for rectangular input surfaces discontinuity boundaries can occur in any direction. Second, given triangular elements we must allow the subdivisions to occur anywhere along the edges.

Such constructions are possible using the ideas of multi-wavelets generalized to triangular domains and making the two scale relation a function of the subdivision points. As an example of such a construction we give here the case of linear elements. Higher orders can also be achieved by the same procedure.

We begin by defining a canonical support domain of $s \in [0,1]$, $t \in [0, 1-s]$. With this domain and the ordinary inner product over it three orthonormal basis functions are given by

$$l_{00}(s,t) = \sqrt{6}(1-2s-2t)$$

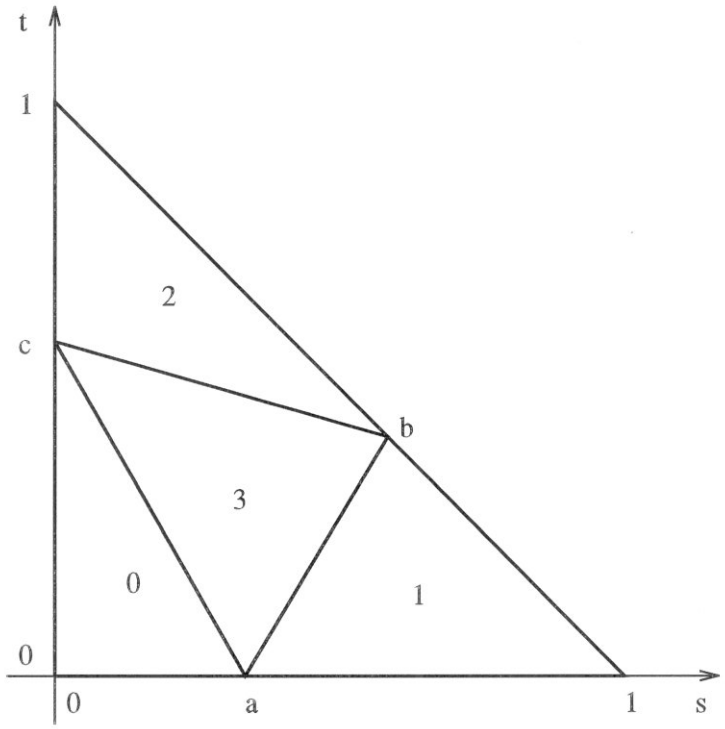



Figure 30: This figure shows the geometry for triangular linear elements. The canonical domain is $s \in [0,1]$ and $t \in [0, 1-s]$. The support domains of the four children are numbered 0–3, while the subdivision points along each axis are named $(a, b, c)$, each of which is in $[0,1]$ as a parameter of the respective edge.

$$
\begin{aligned}
l_{10}(s,t) &= \sqrt{6}(2s-1) \\
l_{01}(s,t) &= \sqrt{6}(2t-1)
\end{aligned}
$$

Letting $a$, $b$, and $c$ denote the subdivision points along the respective sides of the triangles (see Figure 30) we get the two scale relation for the smooth function coefficients as a function of the parameters $(a,b,c)$ as follows (subscripts denote one of three basis functions, while superscripts denote one of four child support areas as indicated in Figure 30)

$$
\sqrt{ac}\begin{pmatrix} -1+a+c & 1-a & 1-c \\ -1+c & 1 & 1-c \\ -1+a & 1-a & 1 \end{pmatrix}\begin{pmatrix} s_{00} \\ s_{10} \\ s_{01} \end{pmatrix} = \begin{pmatrix} s^0_{00} \\ s^0_{10} \\ s^0_{01} \end{pmatrix}
$$

$$
\sqrt{1-a-b+ab}\begin{pmatrix} 1 & -b & b \\ a & 1-a-b & b \\ a & -a & 1 \end{pmatrix}\begin{pmatrix} s_{00} \\ s_{10} \\ s_{01} \end{pmatrix} = \begin{pmatrix} s^1_{00} \\ s^1_{10} \\ s^1_{01} \end{pmatrix}
$$

$$
\sqrt{b-bc}\begin{pmatrix} 1 & 1-b & -1+b \\ c & 1 & -c \\ c & 1-b & b-c \end{pmatrix}\begin{pmatrix} s_{00} \\ s_{10} \\ s_{01} \end{pmatrix} = \begin{pmatrix} s^2_{00} \\ s^2_{10} \\ s^2_{01} \end{pmatrix}
$$

$$
\sqrt{a-ab-ac+bc}\begin{pmatrix} -1+a+c & 1-a & 1-c \\ a & 1-a-b & b \\ c & 1-b & b-c \end{pmatrix}\begin{pmatrix} s_{00} \\ s_{10} \\ s_{01} \end{pmatrix} = \begin{pmatrix} s^3_{00} \\ s^3_{10} \\ s^3_{01} \end{pmatrix}
$$

where we used the letter $s$ to denote the coefficient with respect to a basis function as given by the super/subscripts. Since the basis functions are orthonormal the two scale relation is identical for Push as well as Pull.

The derivation proceeds by orthonormalizing the Lagrange polynomials which span all linear functions on a triangular domain. This is followed by a contraction and reparameterization of each of the orthonormal basis functions on each of the four children. The coefficients in the two scale relation then follow from taking inner products of the child functions against the parents (in effect expanding the parent function with respect to the child functions). This construction generalizes to higher order elements as well, although the expressions become more involved.

# Appendix B

# Polynomial Estimator Oracle

When the refinement function is called we employ an oracle to decide whether a given transport needs to be subdivided. We know that a given transport is sufficiently subdivided if the kernel is locally (over the support of the basis functions) approximately a polynomial of order no higher than the basis functions employed. Let $M-1$ be the order of basis functions. To generate our polynomial estimate we take advantage of the sample points needed by the quadrature. In order to perform a quadrature we need to generate $M$ samples at the Gauss points along each dimension (4D for radiosity, 6D for radiance). These samples define an interpolating polynomial. We do not actually construct this polynomial, but instead use Neville's algorithm [64] to evaluate the polynomial at selected points directly. Since all our rules are product rules we can evaluate all quantities along each dimension in turn and in this way reduce everything to a 1D Neville function.

Below we give pseudocode for the 4D case, the 6D case being exactly analogous save for two more dimensions over which to iterate. We being with a single array of the Gauss points in 1D for a particular order of quadrature $M$, which is used across all four dimensions. The first step is to evaluate the kernel at all the cross product points, resulting in a 4D array of kernel values. Given different set of test locations, which are also used for every dimension, we evaluate the polynomial interpolator at all such test locations. The polynomial interpolator uses the kernel samples generated for the Gauss rule to uniquely define a multivariate polynomial which interpolates all

the values. Comparing this estimate with the actual kernel value and summing over all such errors we arrive at our `PolynomialEstimator` value:

```
float gauss_points[M];
float value_at_gauss_points[M][M][M][M];

EvaluateKernelAtPoints( value_at_gauss_points, gauss_points );

estimate = 0;
for( i = 0; i < no_of_new_samples; i++ ){
  for( j = 0; j < no_of_new_samples; j++ ){
    for( l = 0; l < no_of_new_samples; l++ ){
      for( m = 0; m < no_of_new_samples; m++ ){
        float po = Polint4( gauss_points,
                            value_at_gauss_points,
                            test_locations[i],
                            test_locations[j],
                            test_locations[l],
                            test_locations[m] );

        float ke = EvaluateKernel( test_locations, i, j, l, m );

        estimate += abs( po - ke );
      }
    }
  }
}

return estimate;
```

It remains to show how to realize `Polint4`. The function `Polint4` takes a set of sample locations and the array of corresponding values and successively reduces

the interpolation problem to lower dimensions. Consider a bi-linear interpolation problem. Given four values at the corners, compute any value in between. We can first perform two interpolations in the first dimension (along each of two parallel edges) and then interpolate between these to get the final value. This basic idea extends to higher dimensions

```
float
Polint4( float x[M],
         float f[M][M][M][M],
         float u, float v, float s, float t )
{
  float y2tmp[M];
  for( int i = 0; i < M; i++ ){
    float y3tmp[M];
    for( int j = 0; j < M; j++ ){
      float y4tmp[M];
      for( int k = 0; k < M; k++ ){
        y4tmp[k] = Polint( x, f[i][j][k], t );
      }
      y3tmp[j] = Polint( x, y4tmp, s );
    }
    y2tmp[i] = Polint( x, y3tmp, v );
  }
  return Polint( x, y2tmp, u );
}
```

Finally the function `Polint` is the one dimensional Neville algorithm. We refer the interested reader to Press *et al.*[49] for an efficient implementation.

# Bibliography


[1] ALPERT, B. A Class of Bases in $L^2$ for the Sparse Representation of Integral Operators. *SIAM Journal on Mathematical Analysis 24*, 1 (January 1993).

[2] ANDERSSON, L., HALL, N., JAWERTH, B., AND PETERS, G. Wavelets on Closed Subsets of the Real Line. Tech. Rep. 2, University of South Carolina, Industrial Mathematics Initiative, 1993. To appear in: Topics in the Theory and Application of Wavelets, Larry L. Schumaker and Glenn Webb eds., Academic Press.

[3] APPEL, A. An Efficient Program for Many Body Simulation. *SIAM Journal of Sci. Stat. Computing 6*, 1 (1985), 85–103.

[4] ARVO, J., TORRANCE, K., AND SMITS, B. A Framework for the Analysis of Error in Global Illumination Algorithms. In *Computer Graphics Annual Conference Series, 1994* (1994).

[5] AUPPERLE, L. *Hierarchical Algorithms for Illumination.* PhD thesis, Princeton University, 1993.

[6] 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.

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

[8] BARNES, J., AND HUT, P. A hierarchical $O(n \log n)$ Force Calculation Algorithm. *Nature 324* (1986), 446–449.

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

[10] BOUKNIGHT, W. J. A Procedure for Generation of Three-Dimensional Half-Toned Computer Graphics Presentations. *CACM 13*, 9 (September 1970).

[11] BUI-TUONG, P. Illumination for Computer Generated Pictures. *CACM 18*, 6 (June 1975), 311–317.

[12] 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.

[13] CHRISTENSEN, P. H., STOLLNITZ, E. J., SALESIN, D. H., AND DEROSE, T. D. Importance-Driven Wavelet Radiance. Tech. Rep. 94-01-05, University of Washington, Seattle, January 1994.

[14] CHRISTENSEN, P. H., STOLLNITZ, E. J., SALESIN, D. H., AND DEROSE, T. D. Wavelet Radiance. In *Proceedings of the 5th Eurographics Workshop on Rendering* (Darmstadt, June 1994), pp. 287–302.

[15] COHEN, A., DAUBECHIES, I., AND FEAUVEAU, J.-C. Biorthogonal Bases of Compactly Supported Wavelets. *Communications on Pure and Applied Mathematics XLV* (1992), 485–560.

[16] COHEN, M., CHEN, S. E., WALLACE, J. R., AND GREENBERG, D. P. A Progressive Refinement Approach to Fast Radiosity Image Generation. *Computer Graphics 22*, 4 (August 1988), 75–84.

[17] COHEN, M. F., AND GREENBERG, D. P. The Hemi-Cube: A Radiosity Solution for Complex Environments. *Computer Graphics 19*, 3 (July 1985), 31–40.

[18] COHEN, M. F., GREENBERG, D. P., IMMEL, D. S., AND BROCK, P. J. An efficient radiosity approach for realistic image synthesis. *IEEE Computer Graphics and Applications 6*, 3 (March 1986), 26–35.

[19] COHEN, M. F., AND WALLACE, J. R. *Radiosity and Realistic Image Synthesis.* Academic Press, 1993.

[20] DAUBECHIES, I. *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1992.

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

[22] DIEUDONNÉ, J. *Foundations of Modern Analysis*, vol. 7th. Academic Press, 1968.

[23] GERSHBEIN, R. S. personal communication, 1994.

[24] 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).

[25] GERSHUN, A. *The Light Field.* Moscow, 1936. Translated in Journal of Mathematics and Physics, Vol. 18, No. 2, 1939.

[26] GORAL, C. M., TORRANCE, K. E., GREENBERG, D. P., AND BATTAILE, B. Modelling the Interaction of Light between Diffuse Surfaces. *Computer Graphics 18*, 3 (July 1984), 212–222.

[27] GORTLER, S. personal communication, 1993.

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

[29] GORTLER, S. J., COHEN, M. F., AND SLUSALLEK, P. Radiosity and Relaxation Methods; Progressive Refinement is Southwell Relaxation. Tech. Rep. CS-TR-408-93, Department of Computer Science, Princeton University, February 1993. To appear in IEEE Computer Graphics and Applications.

[30] GOURAUD, H. Continuous Shading of Curved Surfaces. *IEEE Transactions on Computers C-20*, 6 (June 1971), 623–629.

[31] GREENGARD, L. *The Rapid Evaluation of Potential Fields in Particle Systems.* MIT Press, 1988.

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

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

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

[35] 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.

[36] JAFFARD, S., AND LAURENÇOT, P. Orthonormal Wavelets, Analysis of Operators, and Applications to Numerical Analysis. In *Wavelets: A Tutorial in Theory and Applications*, C. K. Chui, Ed. Academic Press, 1992, pp. 543–602.

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

[38] KOK, A. J. Grouping of Patches in Progressive Radiosity. In *Proceedings of the 4th Eurographics Workshop on Rendering* (June 1993), pp. 221–231.

[39] KONDO, J. *Integral Equations.* Oxford Applied Mathematics and Computing Science Series. Kodansha, 1991.

[40] LAMBERT, J.-H. *Photometria sive de mensura et gradibus luminis, colorum et umbrae.* 1760. German translation by E. Anding in *Ostwald's Klassiker der Exakten Wissenschaften*, Vol. 31-33, Leipzig, 1892.

[41] 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.

[42] LISCHINSKI, D., SMITS, B., AND GREENBERG, D. P. Bounds and Error Estimates for Radiosity. In *Computer Graphics Annual Conference Series 1994* (July 1994), Siggraph, pp. 67–74.

[43] LISCHINSKI, D., TAMPIERI, F., AND GREENBERG, D. P. Combining Hierarchical Radiosity and Discontinuity Meshing. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 199–208.

[44] MALLAT, S. G. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 11* (July 1989), 674–693.

[45] MOON, P. *The Scientific Basis for Illuminating Engineering*, dover, 1961 ed. McGraw-Hill, 1936.

[46] MOORE, R. E. *Interval Analysis.* Prentice-Hall, Englewood Cliffs, NJ, 1966.

[47] NISHITA, T., AND NAKAMAE, E. Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection. *Computer Graphics 19*, 3 (July 1985), 23–30.

[48] 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.

[49] PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A., AND VETTERLING, W. T. *Numerical Recipies.* Cambridge University Press, 1986.

[50] RUSHMEIER, H. E., PATTERSON, C., AND VEERASAMY, A. Geometric Simplification for Indirect Illumination Calculations. *Proceedings Graphics Interface* (May 1993), 227–236.

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

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

[53] SCHRÖDER, P. Wavelet Methods for Radiosity. In *Course Notes: Advanced Topics in Radiosity.* ACM Siggraph, July 1994, ch. 3, pp. 1–21.

[54] 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.

[55] SCHRÖDER, P., AND HANRAHAN, P. Wavelet Methods for Radiance Computations. In *Proceedings 5th Eurographics Workshop on Rendering* (June 1994), Eurographics.

[56] 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.

[57] 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.

[58] SIEGEL, R., AND HOWELL, J. R. *Thermal Radiation Heat Transfer.* Hemisphere Publishing Corp., 1978.

[59] SILLION, F. Clustering and Volume Scattering for Hierarchical Radiosity Calculations. In *Proceedings of the 5th Eurographics Workshop on Rendering* (Darmstadt, June 1994), pp. 105–117.

[60] SILLION, F., AND PUECH, C. A General Two-Pass Method Integrating Specular and Diffuse Reflection. *Computer Graphics 23*, 3 (July 1989), 335–344.

[61] 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.

[62] SMITS, B., ARVO, J., AND GREENBERG, D. A Clustering Algorithm for Radiosity in Complex Environments. *Computer Graphics Annual Conference Series* (July 1994), 435–442.

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

[64] STOER, J., AND BULIRSCH, R. *Introduction to Numerical Analysis.* Springer Verlag, New York, 1980.

[65] TELLER, S. personal communication, 1994.

[66] TELLER, S., FOWLER, C., FUNKHOUSER, T., AND HANRAHAN, P. Partitioning and Ordering Large Radiosity Computations. In *Computer Graphics Annual Conference Series 1994* (July 1994).

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

[68] 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.

[69] WALLACE, J. R., ELMQUIST, K. A., AND HAINES, E. A. A Ray Tracing Algorithm for Progressive Radiosity. *Computer Graphics 23*, 3 (July 1989), 315–324.

[70] WHITTED, T. Am Improved Illumination Model for Shaded Display. *Communications of the ACM 23* (1980), 343–349.

[71] ZATZ, H. R. Galerkin Radiosity: A Higher-order Solution Method for Global Illumination. In *Computer Graphics Annual Conference Series 1993* (August 1993), Siggraph, pp. 213–220.