Wavelet Projections for Radiosity

Peter Schröder Steven J. Gortler Michael F. Cohen Pat Hanrahan

Department of Computer Science
Princeton University

Abstract

One important goal of image synthesis research is to accelerate the process of obtaining realistic images using the radiosity method. Two important concepts recently introduced are the general framework of projection methods and the hierarchical radiosity method.

Wavelet theory, which explores the space of hierarchical basis functions, offers an elegant framework that unites these two concepts and allows us to more formally understand the hierarchical radiosity method.

Wavelet expansions of the radiosity kernel have negligible entries in regions where high frequency/fine detail information is not needed. A sparse system remains if these entries are ignored. This is similar to applying a lossy compression scheme to the form factor matrix. The sparseness of the system allows for asymptotically faster radiosity algorithms by limiting the number of matrix terms that need to be computed. The application of these methods to 3D environments is described in [9]. Due to space limitations in that paper many of the subtleties of the construction could not be explored there. In this paper we discuss some of the mathematical details of wavelet projections and investigate the application of these methods to the radiosity kernel of a flatland environment, where many aspect are easier to visualize.

1 Introduction

Heckbert [12] introduced the general concept of using projection methods for solving the radiosity rendering equation. With this formulation of radiosity, he was able to explore more general forms of the associated linear system which use a variety of *basis functions*. Further research along these lines was reported in [16]. In [11] Hanrahan et al. introduced the hierarchical radiosity (HR) method which uses constant basis functions like the more traditional radiosity formulation, but accelerates this process by using a hierarchically subdivided representation of the environment.

More recently Gortler et al. [9] introduced the wavelet radiosity (WR) method which is based on projecting the radiosity rendering equation into a set of *wavelet basis functions*. This method uses recent numeric techniques described in [3, 1, 2]. In this framework HR can be understood as the result of a projection into a particular wavelet basis. Wavelet theory provides a rigorous mathematical foundation for hierarchical radiosity algorithms and extends them to higher order basis functions.

The general idea behind wavelet based radiosity methods is as follows. We project the kernel of the radiosity integral rendering equation into a hierarchical wavelet basis. If the kernel is smooth over some interval of its arguments (perhaps we are dealing with two distant patches) then the fine scale detail information will not be important. If this is the case, the corresponding wavelet coefficient will be near zero. In WR all coefficients below some threshold ϵ are set to zero. This leaves us with a *sparse* matrix for which fast solution methods exist.

Going from a normal finite element basis to a wavelet basis (formally a vector space basis change) can be viewed as a pyramid filter transformation applied to the form factor matrix. If the matrix entries vary smoothly we will find coefficients representing high frequencies to be negligible. Just as in lossy compression schemes, we ignore coefficients with sufficiently small magnitude. In an efficient radiosity implementation we do not actually compute the entire matrix and then perform the filtering, instead we work top down through the hierarchy and directly compute the significant wavelet coefficients through a quadrature method [9]. An implementation using the simplest wavelet basis, the Haar basis, gives us the HR method of [11]. There are other wavelets we can choose that will give us greater compression resulting in faster radiosity algorithms.

The purpose of this paper is to introduce the underlying principles of wavelet projections to the computer graphics community in more detail than was possible in [9]. This presentation follows that in [3] with an emphasis on the relevant issues for radiosity. To allow a more detailed analysis and a clear presentation of some of the involved subtleties, we illustrate the principles using "flatland" radiosity [12] (radiosity in the plane). The application of these principles to full 3D radiosity and experimental results are described in [9].

2 Projection Methods

After a short review of function projections, we discuss how projections can be used to find approximate solutions to integral equations such as the radiosity equation. We begin with the complete, infinite dimensional case and proceed to the finite dimensional, approximate case. The ideas presented here can be found in greater detail in [12, 16, 7, 4].

Since we are working with linear function spaces and operators defined on these spaces we must first consider bases. In general we will be dealing with functions of finite energy, i.e. functions in the linear vector space L^2. There are many possible bases for this space. Consider some primal basis $\{N_i\}_{i\in\mathbf{Z}}$. This set must contain infinitely many functions since L^2 is infinite dimensional. By definition, every function $B \in L^2$ can be written as a linear combination of the basis functions $B(s) = \sum_i b_i N_i(s)$ for some coefficients b_i. In order to find the coefficients of a given function we need an inner product on L^2. The inner product of two functions F and G is defined as $\langle F, G\rangle = \int ds\, F(s)G(s)$. Throughout we will denote functions by capital letters. Their expansion coefficients with respect to a basis will be small letters with indices as subscripts. When denoting functions we will often drop the argument s (or t).

Given a function B and a set of basis functions N_i, one way of finding the coefficients of B with respect to this basis is by performing inner products against the dual basis functions $\tilde{N}_i$. The dual functions form a basis for the same space, and are defined through the property

$$\langle N_i, \tilde{N}_j\rangle = \delta_{ij}$$

where δ_{ij} is the Kronecker Delta. Using the duals we can write the expansion of a function with respect to a basis as

$$B(s) = \sum_i b_i N_i(s) = \sum_i \langle B, \tilde{N}_i\rangle N_i(s)$$

Since the duals form an alternate basis, we can also expand the function with respect to the dual basis

$$B(s) = \sum_i \tilde{b}_i \tilde{N}_i(s) = \sum_i \langle B, N_i\rangle \tilde{N}_i(s)$$

Now consider a linear operator defined on L^2. One example is integration against a kernel function, which is a linear operator since integration is linear. In particular we consider the radiosity integral equation, which can be written as a linear operator

$$B = E + \mathcal{K}B \tag{1}$$

where $\mathcal{K}B(s) = \int ds\, K(s,t)B(t)$. Since $\mathcal{K}$ is a linear operator it is fully described by its action on our chosen basis. Intuitively we can write this as an infinite sized matrix. The columns of the matrix $\mathcal{K}$ are given by $\mathcal{K}N_j$. The coefficients of these columns with respect to our basis can be found by taking inner products with the dual functions. Thus the entries in the matrix representation of $\mathcal{K}$ are given by

$$k_{i,j} = \langle \mathcal{K}N_j, \tilde{N}_i \rangle = \int ds \int dt\, K(s,t)N_j(t)\tilde{N}_i(s) \tag{2}$$

With this matrix representation we can formally write the radiosity integral equation as an infinite sized matrix equation

$$(I - \mathcal{K})B = E$$

Given E we can solve this equation, at least formally, $B = (I - \mathcal{K})^{-1}E$. In practice we have to deal with finite sized matrices. This corresponds to ignoring all but a finite subsquare of the infinite matrix, or said differently, the use of a finite basis. In this case we fix V to be be a subspace of L^2 spanned by some chosen finite basis. There are many choices for V (and its associated basis). For example one choice is the space of functions piecewise constant over fixed intervals, and one basis for that space is the set of "box" functions. Other examples are spaces spanned by "hat" functions or B-splines of higher orders. It is important to remember the difference between a choice of subspace and a choice of basis for this subspace. Once we make a choice of subspace, e.g. all functions which are piecewise linear, we still have considerable freedom in choosing a basis for this space. In particular we will consider wavelet bases as alternative bases.

Let V be our chosen subspace with $\dim(V) = n$ and $\{N_i\}_{i=1,\ldots,n}$ a basis for this space. Given a function in L^2, we define its orthogonal projection into V as

$$\hat{B} = P_V B = \sum_{i=1}^{n} \langle B, \tilde{N}_i \rangle N_i$$

where the $\tilde{N}_i$ span the same finite dimensional space. The orthogonal projection has the property

$$\forall i:\ \langle B - \hat{B}, N_i \rangle = 0$$

Restricting our integral operator to this space, we arrive at the *related* integral equation

$$\hat{B} = \hat{E} + P_V \mathcal{K} P_V \hat{B} \tag{3}$$

In this equation $\hat{E}$ and $P_V\mathcal{K}$ are the orthogonal projections into the chosen finite dimensional space. The P_V to the right of $\mathcal{K}$ notates that we are only interested in the action of $\mathcal{K}$ on our subspace. $\hat{B}$ is the finite dimenstional function we solve for (note that the solution $\hat{B}$ of this equation is not necessarily the orthogonal projection of the B which solves the original integral equation (1)). Writing Equation 3 out as a matrix equation we get

$$b_i = e_i + \sum_{j=1}^{n} k_{i,j} b_j$$

whose solution b_i describes our answer with respect to the basis N_i. The integrals, $k_{i,j}$, must be computed. This is usually done with some form of numerical quadrature or closed form solution [15]. The system of linear equations must then be solved, which is usually done with some form of iterative algorithm [5, 10].

This method of approximately solving an integral equation is known as the Galerkin method. The Galerkin method can also be derived using weighted residuals. This and the point collocation method are discussed in [12].

We are left with a choice of finite dimensional function space, and a basis for this space. In classical radiosity the space of piecewise constant functions is chosen and the box basis is used. Heckbert [12] and Zatz [16] have examined function spaces that were made up of piecewise higher order polynomials. Wavelets provide another choice of basis. Some wavelets produce entirely new subspaces, while others provide new bases for well known subspaces.

3 Wavelets

Wavelet theory is a rapidly developing field that has its roots in pure mathematics [6] and signal processing [14]. In this section we will review the construction of a family of hierarchical bases called wavelets focusing on the relevant issues for radiosity. We will describe fast algorithms that obtain the coefficients of a function with respect to a wavelet basis and explain why smooth functions can be approximated accurately with only a few of these wavelet coefficients.

The simplest wavelet construction is the Haar construction shown in Figure 1. In the upper left is a set of basis functions which span all piecewise constant functions at resolution 8 on the interval. Using the operators g (pairwise differencing) and h (pairwise averaging) we can construct another basis for the same space (upper right). Four of these functions are just like the original basis, only wider, thus we can repeat the construction (middle right). Repeating once more we finally have a basis for the original space of functions consisting of the overall average ϕ_0 and the difference functions $\psi_{i,j}$ from all the lower levels. The last set of functions is known as the Haar *wavelet basis*. This construction is very similar to an image pyramid that one might use for texture mapping. In such a pyramid the image (function in our case) is represented at different levels of resolution by successive averaging steps. In the Haar pyramid we only remember the overall average and all the *differences* between successive levels of the pyramid. The Haar basis is only the simplest example of an infinite family of such constructions, however the basic principles are the same for all wavelet bases.

One approach to wavelets considers the decomposition of a given function $B \in L^2$ into components representing the function at various levels of detail. More precisely we start with a reference space V_0 which contains all functions at a particular level of detail. For example, this space might consist of all functions defined on the real line, which are constant over subintervals $[k, k+1)$ for all $k \in \mathbf{Z}$. A basis for this space would be given by the integer translates of $\phi(s) = 1$, $s \in [0,1]$, 0 otherwise (i.e. a "box" basis). Clearly this space *contains* all functions which are constant over intervals of twice the length ($[2k, 2k+2)$, $k \in \mathbf{Z}$). Similarly V_0 is *contained in* the space of all functions constant over intervals of half the length ($[k/2, k/2 + 1/2]$, $k \in \mathbf{Z}$). Formally we define a multi resolution analysis (MRA) [14] on L^2, by decomposing L^2 into a sequence of nested subspaces

$$\ldots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \ldots$$

with some function ϕ whose translates $\{\phi(s-j) | j \in \mathbf{Z}\}$ form a basis for V_0 and $\{\phi_{i,j}(s) = 2^{i/2}\phi(2^i s - j) | j \in \mathbf{Z}\}$ a basis for V_i by construction. Thus all the V_i are spanned by *translates* and *scales* of a single function which is often referred to as the *smooth shape*. With this indexing, the function $\phi_{i-1,j}$ is just like the function $\phi_{i,j}$ except that $\phi_{i-1,j}$ is twice as wide, and $\frac{1}{\sqrt{2}}$ times as tall. This scaling insures that $\langle \phi_{i,j}, \phi_{i,k} \rangle$ remains constant independent of i. We have given one example for a V_0 and ϕ above, but there are many others. The space spanned by translates of a B-spline of some order with the integers as its knot sequence is another such example.

Given the above nested sequence of subspaces we can define the complement W_i of V_i in V_{i+1}

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

In this way every function in V_{i+1} can be written uniquely as a sum of two parts, one from V_i and one from W_i. Because of the nesting of the V_i we can write $L^2 = \bigoplus_{i \in \mathbf{Z}} W_i$, which

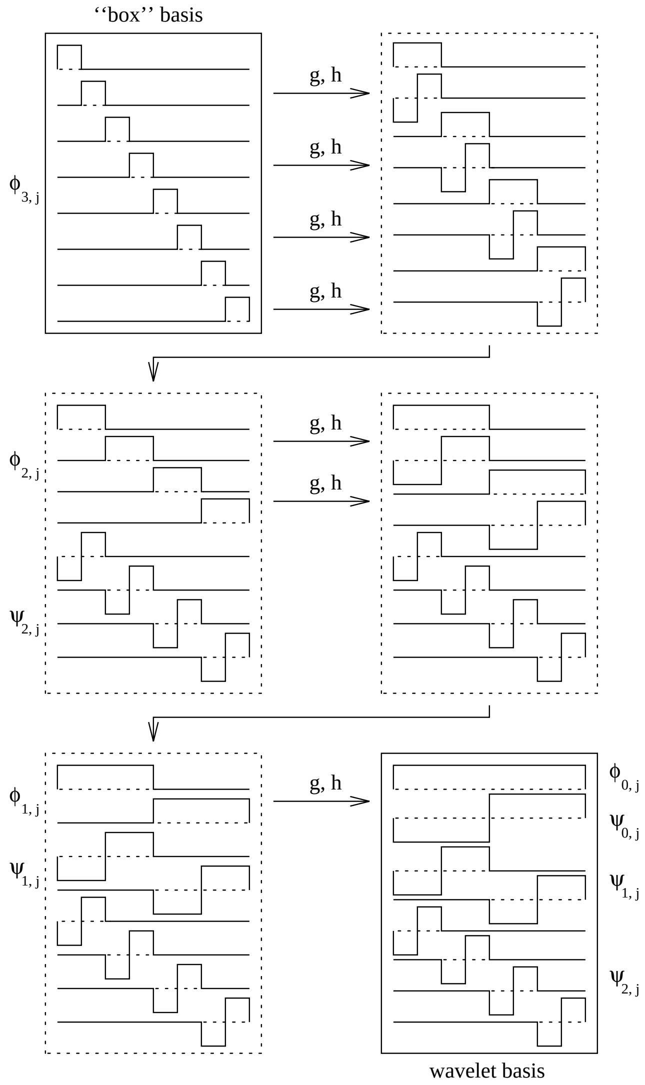

Figure 1: Transformation of a piecewise constant basis into the Haar wavelet basis.

means that all of L^2 can be written as the sum of the spaces W_i. Every function $B \in L^2$ can therefore be written as a sum of functions, each of which comes from a specific level of detail W_i. Similarly to ϕ, whose translates span V_0 we consider the *wavelet* ψ whose translates span W_0. Due to our construction we find that $\{\psi_{i,j}(s) = 2^{i/2}\psi(2^i s - j) | j \in \mathbf{Z}\}$ will be a basis for W_i. Since the W_i encode differences between successive levels of smoothness, the basis function ψ is often referred to as the *detail shape*. Above we gave the box function as an example of a basis for a particular sequence of V_i. In this case we find the corresponding ψ to be given by the Haar wavelet (see Figure 1).

In order to project functions into the V_i and W_i subspaces we define projection operators P_i for V_i and $Q_i = P_{i+1} - P_i$ for W_i. P_iB describes B at the resolution level i, while Q_iB describes the detail of B at level i which is lost when going from level $i+1$ to level i.

If we limit ourselves to some finite range of levels of detail $(0, \ldots, L)$, we can arbitrarily fix V_0 as the coarsest level and consider an L level hierarchy of functions which are scales and translates of ϕ and ψ. A basis for the entire space is created by taking the union of the bases for V_0 and the W_i, $i = 0, \ldots, L-1$. This basis is called the *wavelet* basis (Figure 1, lower right). The basis functions are then $\psi_{i,j}$ where $i = 0, \ldots, L-1$ and $j = 0, \ldots, 2^i - 1$ and the single smooth shape on the top level ϕ_0.

Since $\phi_{i,j} \in V_i \subset V_{i+1}$ we can express $\phi_{i,j}$ as a linear combination of basis functions for V_{i+1}

$$\phi_{i,j} = \sum_k h_{k-2j}\phi_{i+1,k}$$

and similarly for $\psi_{i,j} \in W_i \subset V_{i+1}$

$$\psi_{i,j} = \sum_k g_{k-2j}\phi_{i+1,k}$$

These equations are known as the *two-scale relationship*. This linear combination can be expressed as convolutions with sequences h and g which are then subsampled by 2 (the subsampling is expressed by the 2 in the index "$k-2j$" of h and g). h can be thought of as a low pass filter and g as a high pass filter.

Given the wavelet coefficients $(b_{\phi_0}, b_{\psi_{i,j}})$, the two-scale relationship can be used to go down the hierarchy and obtain coefficients $b_{\phi_{L,j}}$ which describe the function expanded with respect to the finest level smooth functions

```
XformDown( vector $b_\phi$, vector $b_\psi$, int i )
  for( $j = 0$; $j < 2 * 2^i$; $j++$ )
    $b_\phi^{down}[j]$ =$\sum_k h_{j-2k} b_\phi[k] + \sum_k g_{j-2k} b_\psi[k]$;
  return $b_\phi^{down}$ ;

PyramidDown( $b_{\phi_0}$, $\{b_{\psi_{i,j}} | i = 0, \ldots, L-1\}$ )
  for( $i = 0$; $i < L$; $i++$ )
    $b_{\phi_{i+1,j}}$ = XformDown( $b_{\phi_{i,j}}$, $B_{\psi_{i,j}}$ , i );
  return $b_{\phi_{L,j}}$ ;
```

This process can be reversed, obtaining the wavelet coefficients using a similar looking algorithm **PyramidUp** that uses dual convolutions $\tilde{h}$ and $\tilde{g}$ [1]

```
XformUp( vector $b_\phi$, int i )
  for( $j = 0$; $j < 2^i/2$; $j++$)
    $b_\phi^{up}[j]$ = $\sum_k \tilde{h}_{k-2j} b_\phi[k]$;
    $b_\psi^{up}[j]$ = $\sum_k \tilde{g}_{k-2j} b_\phi[k]$;
  return ($b_\phi^{up}$, $b_\psi^{up}$);

PyramidUp( vector $b_{\phi_{L,j}}$ )
  for( $i = L$; $i > 0$; $i--$ )
    ($b_{\phi_{i-1,j}}$, $b_{\psi_{i-1,j}}$) = XformUp( $b_{\phi_{i,j}}$ , i );
  return ($b_{\phi_0}$, $\{b_{\psi_{i,j}} | i = 0, \ldots, L-1\}$);
```

If the $\tilde{h}$ and $\tilde{g}$ convolutions have constant width (with respect to i) then a single call to **XformUp** will take time linear in the size of the vector b_{ϕ_i}. Each successive call in **PyramidUp** cuts the length of the vectors by 1/2. Since the geometric series $n + \frac{n}{2} + \frac{n}{4} + \ldots + 1$ converges to $O(n)$, the entire pyramid can be built in $O(n)$ (see Figure 1). This is important since a general basis change can take $O(n^2)$ time.

Going up the hierarchy we apply a low pass filter to the current level to get the smooth part one level up in the hierarchy (after subsampling by a factor of 2). Applying the high pass filter to the current level gives us the detail at this level. Since this corresponds to space W_i it encodes the difference between the two levels of smoothness. Considering the example of the Haar wavelet again we see that $h = \tilde{h}$ averages two neighboring values (a crude low pass filter) while $g = \tilde{g}$ differences two neighboring values (a crude high pass filter). This process can be understood as a recursive filter bank [14]. Indeed one choice for the convolution sequences, which define the smooth and detail function, is a set of Quadrature Mirror filters.

The dual convolutions define a dual wavelet hierarchy with the two-scale relationship

$$\tilde{\phi}_{i,j} \;\; = \;\; \sum_k \tilde{h}_{k-2j}\tilde{\phi}_{i+1,k}$$

[1] In an orthonormal wavelet basis, the dual convolutions are the same sequences h and g.

$$\tilde{\psi}_{i,j} = \sum_k \tilde{g}_{k-2j}\tilde{\phi}_{i+1,k}$$

This dual wavelet basis then gives rise to a dual **PyramidUp** procedure that uses h and g and a dual **PyramidDown** procedure that uses $\tilde{h}$ and $\tilde{g}$.

If **Xb** is the matrix representation of **PyramidUp** being applied to **b**, then $\mathbf{X}^{-1}\mathbf{b}$ computes **PyramidDown**, $\mathbf{X}^{-T}\mathbf{b}$ computes dual **PyramidUp** and $\mathbf{X}^{T}\mathbf{b}$ computes dual **PyramidDown**

Given a wavelet basis we can also express the expansion of an arbitrary function $B(s)$ with respect to our wavelet basis by taking inner products

$$\begin{aligned}\hat{B}(s) &= P_0 B(s) + \sum_{i=0}^{L-1} Q_i B(s) \\ &= b_{\phi_0}\phi_0(s) + \sum_{ij} b_{\psi_{i,j}}\psi_{i,j}(s) \\ &= \langle B, \tilde{\phi}_0\rangle\phi_0(s) + \sum_{ij}\langle B, \tilde{\psi}_{i,j}\rangle\psi_{i,j}(s)\end{aligned}$$

If the dual wavelets form a basis for the same space, then the same projection may be written as

$$\begin{aligned}\hat{B}(s) &= \tilde{P}_0 B(s) + \sum_{i=0}^{L-1} \tilde{Q}_i B(s) \\ &= b_{\tilde{\phi}_0}\tilde{\phi}_0(s) + \sum_{ij} b_{\tilde{\psi}_{i,j}}\tilde{\psi}_{i,j}(s) \\ &= \langle B, \phi_0\rangle\tilde{\phi}_0(s) + \sum_{ij}\langle B, \psi_{i,j}\rangle\tilde{\psi}_{i,j}(s)\end{aligned}$$

The equivalence between the inner products and the **PyramidUp** procedure is seen from the following equation

$$\begin{aligned}b_{\psi_{i,j}} &= \langle B, \tilde{\psi}_{i,j}\rangle \\ &= \sum_k \tilde{g}_{k-2j}\langle B, \tilde{\phi}_{i+1,k}\rangle \\ &= \sum_k \tilde{g}_{k-2j} b_{\phi_{i+1,k}}\end{aligned}$$

From the filtering point of view, it is not surprising that a sufficiently smooth function $B(s)$ will have negligible coefficients since they are the result of high pass filtering. By ignoring these small coefficients (i.e. setting them to zero) we are left with a *sparse*, approximate representation.

From the inner products point of view, we find that the small coefficients occur because wavelet functions have *vanishing moments*. We say that a function ψ has M vanishing moments if

$$\int ds\, s^i\,\psi(s) = 0 \quad i = 0, \ldots, M-1$$

The Haar wavelet has one vanishing moment. If a function is nearly constant over some (possibly small) region we will find the corresponding wavelet coefficient at that level to be near 0. Similarly, if a wavelet basis function has two vanishing moments, the coefficients corresponding to regions over which the function is nearly linear will vanish.

More formally, let $B \in L^2$ be an M-times differentiable function at s_0. Such functions can be written in a Taylor expansion [8] up to order $M-1$ about s_0

$$B(s_0 + h) = \sum_{i=0}^{M-1} \frac{B^{(i)}(s_0)}{i!}h^i + \frac{B^{(M)}(\xi)}{M!}h^M$$

for some $\xi \in [s_0 - h, s_0 + h]$. This formula is also sometimes referred to as the generalized meanvalue theorem. Note that ξ is a function of h. In words we say that such a B is approximated by a polynomial in h near s_0 with an error of $O(h^M)$. Substituting this expansion into the inner product with the detail shape and using the vanishing moment property, we get

$$\begin{aligned}\left|\int ds\, B(s)\psi(s)\right| &= \left|\int ds\, \frac{B^{(M)}(\xi)}{M!}h^M\psi(s)\right| \\ &\leq \left|\frac{B^{(M)}(\xi)}{M!}\right| W^M \int ds\, |\psi(s)| \qquad (4)\end{aligned}$$

where W is the width of the interval over which we are considering our expansion (the support of the particular wavelet function). From the last equation we can see that the coefficient b_ψ will be small whenever W and/or the M^{th} derivative of B is small. For example if we are at level i, W will be 2^{-i}. Going to level $i+1$ will decrease the bound by 2^{-M} until eventually the coefficient is below our desired threshold and we have represented B sufficiently well.

3.1 Tradeoffs

The Haar wavelet is just one basis from an infinite family of wavelet bases. When choosing or constructing a particular wavelet, we must balance various properties with one another

- Vanishing moments lead to sparse projections of smooth functions
- Orthogonality simplifies the algorithms and leads to numerical stability
- Continuity is desirable for representing continuous functions
- Finite (small) support decreases the number of operations involved in the pyramid transform

Increasing the number of vanishing moments requires that convolution sequences have broader support, since extra degrees of freedom are needed to satisfy the larger number of constraints. Orthogonality simplifies the algorithm since $g = \tilde{g}$ and $h = \tilde{h}$. It also assures numerical stability of the transformation since the condition number of the basis change that we perform is 1 in this case. While it is possible to derive convolution sequences by purely algebraic means without regard to the underlying functions ϕ and ψ this often leads to highly irregular smooth and detail functions. As a result any quantization noise will show itself as highly irregular, and noticeable error in the computed function. In general, the length of the convolution sequences h and g can be large or even infinite, however, for obvious efficiency reasons, sequences with small support are desirable.

3.2 Flatlets

To illustrate these principles we give a hierarchical basis with two vanishing moments ($\mathcal{F}_2$) and show how to construct bases with even more vanishing moments. Note that these hierarchies are technically not true wavelet bases since they have more then one different detail shape.

We begin with ϕ^1 and ϕ^2 which are adjacent box functions and define the hierarchy of basis functions using the two-scale relationship (Figure 2) [2]

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ -1 & 3 & -3 & 1 \\ -1 & 1 & 1 & -1 \end{bmatrix}\begin{bmatrix} \phi^1_{i,2j} \\ \phi^2_{i,2j} \\ \phi^1_{i,2j+1} \\ \phi^2_{i,2j+1} \end{bmatrix} = \begin{bmatrix} \phi^1_{i-1,j} \\ \phi^2_{i-1,j+1} \\ \psi^1_{i-1,j} \\ \psi^2_{i-1,j+1} \end{bmatrix} \qquad (5)$$

The top two rows of this matrix are chosen to give us box functions twice as wide while the bottom two rows are orthogonal to constant and linear variation (the vectors $[1,1,1,1]$, $[0,1,2,3]$). From this relationship we see that **XformDown** would use the

[2] In this section we drop all normalization constants from the matrix equations.

identity

$$\begin{bmatrix} 1 & 0 & -1 & -1 \\ 1 & 0 & 3 & 1 \\ 0 & 1 & -3 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} b_{\phi^1_{i-1,j}} \\ b_{\phi^2_{i-1,j+1}} \\ b_{\psi^1_{i-1,j}} \\ b_{\psi^2_{i-1,j+1}} \end{bmatrix} = \begin{bmatrix} b_{\phi^1_{i,2j}} \\ b_{\phi^2_{i,2j}} \\ b_{\phi^1_{i,2j+1}} \\ b_{\phi^2_{i,2j+1}} \end{bmatrix}$$

`XformUp` is the inverse operation and hence uses the inverse matrix.

$$\frac{1}{8}\begin{bmatrix} 5 & 3 & 1 & -1 \\ -1 & 1 & 3 & 5 \\ -1 & 1 & -1 & 1 \\ -2 & 2 & 2 & -2 \end{bmatrix} \begin{bmatrix} b_{\phi^1_{i,2j}} \\ b_{\phi^2_{i,2j}} \\ b_{\phi^1_{i,2j+1}} \\ b_{\phi^2_{i,2j+1}} \end{bmatrix} = \begin{bmatrix} b_{\phi^1_{i-1,j}} \\ b_{\phi^2_{i-1,j+1}} \\ b_{\psi^1_{i-1,j}} \\ b_{\psi^2_{i-1,j+1}} \end{bmatrix}$$

This relationship then defines the dual two-scale relationship

$$\frac{1}{8}\begin{bmatrix} 5 & 3 & 1 & -1 \\ -1 & 1 & 3 & 5 \\ -1 & 1 & -1 & 1 \\ -2 & 2 & 2 & -2 \end{bmatrix} \begin{bmatrix} \tilde{\phi}^1_{i,2j} \\ \tilde{\phi}^2_{i,2j} \\ \tilde{\phi}^1_{i,2j+1} \\ \tilde{\phi}^2_{i,2j+1} \end{bmatrix} = \begin{bmatrix} \tilde{\phi}^1_{i-1,j} \\ \tilde{\phi}^2_{i-1,j+1} \\ \tilde{\psi}^1_{i-1,j} \\ \tilde{\psi}^2_{i-1,j+1} \end{bmatrix}$$

where we define the $\tilde{\phi}^1_{L,j}$ and $\tilde{\phi}^2_{L,j}$ to also be adjacent box functions, i.e. the primal and dual hierarchies agree on the lowest level.

If a linear function is projected into the dual basis, many coefficients will be near zero due to the two vanishing moments of the wavelet functions. The representation of linear variation is possible since the dual functions are made up of ramp like functions that are piecewise constant on the finest level.

This construction can be generalized for any number of vanishing moments. To construct the two-scale relationship for $\mathcal{F}_3$ we would use a matrix, like that of Equation 5, where the first 3 rows are chosen to give us box functions of width 2 and the bottom 3 rows are chosen to be orthogonal to constant, linear, and quadratic variation. This implies that the last three rows will be a basis for the null space of

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 4 & 9 & 16 & 25 \end{bmatrix}$$

Clearly there are many bases for this space each giving us a set of wavelets and associated two-scale relationship. To fix these degrees of freedom we give the wavelets even more vanishing moments. We require the first wavelet to have 5 vanishing moments, giving us the row $(-1, 5, -10, 10, -5, 1)$. The next wavelet is required to have 4 vanishing moments and to be orthogonal to the first wavelet, giving us $(1, -3, 2, 2, -3, 1)$. Finally, the third wavelet is required to have 3 vanishing moments and be orthogonal to the first 2 wavelets, yielding the final row $(-5, 7, 4, -4, -7, 5)$ and the matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ -1 & 5 & -10 & 10 & -5 & 1 \\ 1 & -3 & 2 & 2 & -3 & 1 \\ -5 & 7 & 4 & -4 & -7 & 5 \end{bmatrix}$$

After normalizing each row we have the two-scale relationship for $\mathcal{F}_3$. This procedure is very similar to the one used in [2].

4 Linear operators in wavelet bases

In Section 2 we used a projected version of the radiosity function to develop a finite version of the integral operator. We can also interpret this projection as the expansion of the kernel function itself with respect to the given basis. Since the kernel is a function of two arguments we need to consider basis functions of two arguments. Just as we were able to project functions of one argument into a wavelet basis and get a sparse representation we now consider projecting the kernel into a wavelet basis of two arguments. The hope is that this will result in a sparse representation for the kernel function as well.

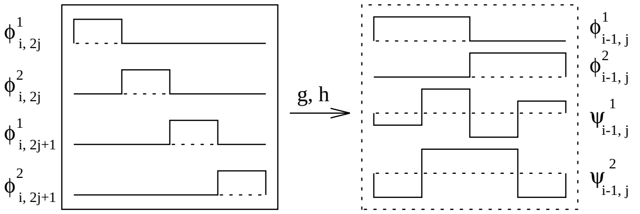

Figure 2: The $\mathcal{F}_2$ wavelet construction. $\mathcal{F}_2$ bases have two different detail shapes. Both of the detail shapes have two vanishing moments, although the smooth shapes are still box functions.

Using a wavelet basis of two arguments, we can represent the projection $\hat{K}(s,t)$ of the kernel function in that basis with a matrix of coefficients. These coefficients form the matrix of the radiosity system with respect to our basis. Using this matrix representation we can solve the linear system to compute the radiosity in the environment. If we keep all n^2 wavelet coefficients representing the kernel, we will arrive at the same solution that we would have obtained if we had not used the wavelet basis. The advantage of the wavelet basis is that most of the coefficients will be negligible due to the vanishing moments property, and can be ignored while still obtaining a highly accurate solution. Beylkin et al. [3] have shown that for kernel functions which are sufficiently smooth we can indeed ignore all entries below a certain threshold without incurring an error larger then some desired ϵ. They show that the number of terms which cannot be ignored for a given error tolerance is $O(n \log n)$ and $O(n)$ for the so called standard and non-standard basis respectively. The conditions to invoke this theorem are satisfied by the radiosity kernel and we will give an explicit proof of this for two particular configurations in the appendix.

4.1 Standard Basis

Assume a kernel function $K(s,t)$ of some integral operator which acts on our finite dimensional space V_L. We can approximate it by some function $\hat{K}(s,t)$ that lies in the tensor product space $V_L \times V_L$ using the $k_{i,j}$ matrix terms of Equation 2

$$\begin{aligned} \hat{K}(s,t) &= \sum_{i,j} k_{i,j} \tilde{N}_j(t) N_i(s) \\ &= \sum_{i,j} \langle \langle K, N_j \rangle, \tilde{N}_i \rangle \;\; \tilde{N}_j(t) N_i(s) \qquad (6) \end{aligned}$$

where the inner-most inner product is with respect to the variable t. This operator can be expressed in the so called standard wavelet basis using the identity $P_L = P_0 + \sum_{i=0}^{L-1} Q_i$

$$\begin{aligned} & P_L \mathcal{K} P_L \\ &= (P_0 + \sum_{i=0}^{L-1} Q_i)\mathcal{K}(P_0 + \sum_{i=0}^{L-1} Q_i) \\ &= P_0 \mathcal{K} P_0 + \sum_{i=0}^{L-1} P_0 \mathcal{K} Q_i + \sum_{i=0}^{L-1} Q_i \mathcal{K} P_0 + \sum_{i,l=0}^{L-1} Q_i \mathcal{K} Q_l \end{aligned} \qquad (7)$$

Each one of the terms in the above expansion describes the action of the integral operator between the two spaces on its left and right side. Alternatively these matrix terms can be considered coefficients of the kernel function with respect to the basis functions

$$\begin{array}{cc} \tilde{\phi}_0(t)\phi_0(s) & \tilde{\phi}_0(t)\psi_{i,j}(s) \\ \tilde{\psi}_{l,m}(t)\phi_0(s) & \tilde{\psi}_{l,m}(t)\psi_{i,j}(s) \end{array}$$

where $i, l = 0, \ldots, L-1$, $j = 0, \ldots, 2^i - 1$, and $m = 0, \ldots, 2^l - 1$. This is just the tensor product basis for a two parameter function space generated from the one parameter bases $\{\tilde{\phi}_0, \tilde{\psi}_{l,m}\} \times \{\phi_0, \psi_{i,j}\}$.

Just as in the case of functions in V_L we can use a **SPyramidUp** algorithm to compute all the expansion coefficients if we start with the projection into the smooth functions on the finest level $\langle\langle K, \phi_{L,l}\rangle, \tilde{\phi}_{L,j}\rangle$. Picturing these coefficients as a matrix we first pyramid transform all the columns (using the **PyramidUp** procedure) to compute $P_L \mathcal{K} P_L = (P_0 + \sum_{i=0}^{L-1} Q_i)\mathcal{K} P_L$. The resulting set of coefficients is now pyramid transformed along all rows (using the dual **PyramidUp** procedure), replacing the right hand side P_L with $P_0 + \sum_{i=0}^{L-1} Q_i$.

In matrix notation this is equivalent to rewriting the application of the kernel matrix **K** to the vector **b** as

$$\mathbf{Kb} = \mathbf{X}^{-1}(\mathbf{XKX}^{-1})\mathbf{Xb} \tag{8}$$

XK is an application of **PyramidUp** to the columns of **K** while $((\mathbf{XK})\mathbf{X}^{-1})$ is an application of dual **PyramidUp** to the rows of the resulting matrix.

As with one dimensional wavelets, if ψ has vanishing moments, and the kernel function $K(s,t)$ is locally well represented by a low order polynomial in one of its coordinate directions over the support of the basis function, many of its coefficients will be set to near zero during the horizontal and/or vertical **PyramidUp** passes.

The indexing of the individual terms in Equation 7 implies that we have "coupling" between all levels of the hierarchy. For some kernels this results in not as many coefficients being near zero as would be if all scales were decoupled. As an intuitive argument why this is so, consider the two **PyramidUp** passes. In the first pass we transform all columns and thus exploit smoothness along the columns. In the second pass we transform all rows *of the column transformed* matrix. Thus, much of the smoothness that was present in the original rows may not be present anymore in the column transformed version. We now describe a realization of the operator that decouples all scales.

4.2 Non-Standard Basis

The non-standard basis [3] attempts to take advantage of smoothness in both coordinate directions simultaneously. Instead of first transforming all rows followed by a transformation of all resulting columns, the row and column transforms are interleaved. Formally this corresponds to the following operator decomposition

$$\begin{aligned} & P_L \mathcal{K} P_L \\ &= P_0 \mathcal{K} P_0 + \sum_{i=0}^{L-1} (P_{i+1} \mathcal{K} P_{i+1} - P_i \mathcal{K} P_i) \\ &= P_0 \mathcal{K} P_0 + \sum_{i=0}^{L-1} Q_i \mathcal{K} P_i + \sum_{i=0}^{L-1} P_i \mathcal{K} Q_i + \sum_{i=0}^{L-1} Q_i \mathcal{K} Q_i \end{aligned} \tag{9}$$

where we use the fact that $P_{i+1} = P_i + Q_i$ and rewrite each summand in turn as

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

The matrix coefficients of the above decomposition can be considered coefficients of the kernel function expanded with respect to the following basis

$$\begin{array}{cc} \tilde{\phi}_0(t)\phi_0(s) & \tilde{\phi}_{i,m}(t)\psi_{i,j}(s) \\ \tilde{\psi}_{i,m}(t)\phi_{i,j}(s) & \tilde{\psi}_{i,m}(t)\psi_{i,j}(s) \end{array}$$

where $i = 0, \ldots, L-1$, $j = 0, \ldots, 2^i - 1$, and $m = 0, \ldots, 2^l - 1$, i.e. we only couple basis functions from the same level i. This set of functions is known as the non-standard basis.

To obtain the above coefficients, we use the procedure **NSPyramidUp** which begins with the kernel coefficients on the finest level. We apply one level of the pyramid transform to all columns (using the **XformUp** procedure) followed by a single level pyramid transform of the rows (using the dual **XformUp** procedure). The first step decomposes each column into a *detail* and *smooth* part $(P_i + Q_i)\mathcal{K} P_{i+1}$. Running a single level transform on each row we obtain their *smooth* and *detail* coefficients $(P_i + Q_i)\mathcal{K}(P_i + Q_i)$. This leaves us with four types of coefficients, $Q_i \mathcal{K} Q_i$, $Q_i \mathcal{K} P_i$, $P_i \mathcal{K} Q_i$, and $P_i \mathcal{K} P_i$. We now recurse on the $P_i \mathcal{K} P_i$ quadrant of coefficients until we reach the top level $P_0 \mathcal{K} P_0$.

We noted above that for the case of changing to the wavelet basis in V_L only $O(n)$ operations are required in contrast to the general basis change which requires $O(n^2)$ operations. Similarly we find that the wavelet basis change applied to a function defined on $V_L \times V_L$ requires $O(n^2)$ work while in the general case such a basis transformation would require $O(n^3)$ work.

5 Using Wavelets for Radiosity Solutions

When using wavelets for radiosity in a practical algorithm we want the $k_{i,j}$ to be symmetric for two reasons. Symmetry implies that we only need one quadrature to compute both $k_{i,j}$ and $k_{j,i}$ and half the storage for the hierarchical matrix representation. Furthermore it may be true (as is the case for flatlets) that the primal wavelets ψ have vanishing moments, while the dual wavelets $\tilde{\psi}$ do not. In this case we can only hope to reap the benefits of sparsity by having only primal basis functions under the integrals of the associated inner product (see Equation 6). To achieve this, we need to expand the columns of all matrices in Equation 7 and 9 with respect to the dual basis. This corresponds to expanding the finite operator $\tilde{P}_L \mathcal{K} P_L$ where all left hand projection operators will be of the " ˜ " type. The procedure **NSPyramidUp** would use the dual **XformUp** on both rows and columns. We call this the symmetric version.

Using this symmetric approach, we are expanding the kernel function as

$$\hat{K}(s,t) = \sum_{i,j} \langle\langle K, N_j\rangle, N_i\rangle \;\; \tilde{N}_j(t)\tilde{N}_i(s)$$

and rewriting the application of the kernel matrix to a vector as

$$\mathbf{Kb} = \mathbf{X}^T(\mathbf{X}^{-T}\mathbf{KX}^{-1})\mathbf{Xb}$$

This implies that after an application of the integral operator our result is expressed in the dual basis and needs to be transformed back to the primal basis. For flatlets this program is simple. Given $\hat{B}$ in the dual wavelet basis we pyramid transform it in linear time to the canonical basis on the finest level, which is shared between both primal and dual flatlet hierarchies, and immediately pyramid transform back to the primal basis.

The linear system for the standard basis is obtained from the operator expansion of Equation 7 and matrix expansion of Equation 8. This follows directly Equation 3 using the wavelet basis for B.

Considering the non-standard form of the operator (Equation 9) we see that we need *both* the detail and smooth coefficients of $\hat{B}$ on all scales. This is not a representation of $\hat{B}$ in any basis [3] but rather an over representation of length $2n$. By arranging all coefficients $k_{i,j}$ as shown in Figure 4 this too can be written as a matrix multiply. The entries in this matrix are given by Equation 9.

[3] In order to express this expansion as in Equation 8 we would need to use non square matrices.

5.1 Using the Non-Standard Basis

In order to compute the action of the operator $\mathcal{K}$ expanded with respect to the non-standard basis on a function B, we interpret Equation 9 (symmetric version) as the following three phase algorithm.

Step 1 *Pull*: Obtain the n (n = number of bases) coefficients $b_{\phi_{i,j}}$ and the n coefficients $b_{\psi_{i,j}}$ of the radiosity function. If we are initially given the coefficients $b_{\phi_{L,j}}$, the $2n$ needed coefficients can be obtained by calling a procedure **Pull** which is just like **PyramidUp** except it returns *both* the ϕ and ψ coefficients. This step transforms n coefficients into $2n$ coefficients. **Pull** is then

```
Pull( vector $b_{\phi_{L,j}}$ )
  for( $i = L$; $i > 0$; $i--$ )
    $(b_{\phi_{i-1,j}}$, $b_{\psi_{i-1,j}})$ = XformUp( $b_{\phi_{i,j}}$ , i );
  return $(\{b_{\phi_{i,j}}$, $b_{\psi_{i,j}} | i = 0, \ldots, L-1\})$ ;
```

Step 2 *Gather*: Let the projected kernel operate on the projected radiosity function. This is the application of Equation 9 to $(b_{\phi_{i,j}}, b_{\psi_{i,j}})$ **Gather** collects the coefficient vectors resulting from the $\tilde{P}_i$ and $\tilde{Q}_i$ that appear on the left of each summand. This step is essentially a matrix multiply and is where we take advantage of the sparseness of the wavelet representation. **Gather** results in $2n$ coefficients $\beta_{\tilde{\phi}_{i,j}}$ and $\beta_{\tilde{\psi}_{i,j}}$ that represent the resultant radiosity function as a combination of $\tilde{\phi}_{i,j}$ and $\tilde{\psi}_{i,j}$.

Step 3 *Push*: Reconstruction of the radiosity function using the $2n$ functions $\tilde{\phi}_{i,j}$ and $\tilde{\psi}_{i,j}$ is done with the procedure **Push** which is similar to **PyramidDown** but takes as arguments *both* the ϕ and ψ coefficients. **Push** is then

```
Push( $\{b_{\phi_{i,j}}$, $b_{\psi_{i,j}} | i = 0, \ldots, L-1\}$ )
  for( $i = 0$; $i < L$; $i++$ )
    $b_{\phi_{i+1,j}}$ += XformDown( $b_{\phi_{i,j}}$, $b_{\psi_{i,j}}$ , i );
  return $b_{\phi_{L,j}}$ ;
```

Wrapping this projected operator within a Jacobi iteration loop results in the following algorithm

```
$b_{\phi_{L,j}}$ = $e_{\phi_{L,j}}$;
$\hat{k}$ = ProjectKernel();
while( !converged )
  $(b_{\phi_{i,j}}$, $b_{\psi_{i,j}})$ = Pull( $b_{\phi_{L,j}}$ );
  $(\beta_{\tilde{\phi}_{i,j}}$, $\beta_{\tilde{\psi}_{i,j}}$) = Gather( $b_{\phi_{i,j}}$, $b_{\psi_{i,j}}$, $\hat{k}$ );
  $\beta_{\phi_{L,j}}$ = $\beta_{\tilde{\phi}_{L,j}}$ = DualPush( $\beta_{\tilde{\phi}_{i,j}}$, $\beta_{\tilde{\psi}_{i,j}}$ );
  $b_{\phi_{L,j}}$ = $\beta_{\phi_{L,j}} + e_{\phi_{L,j}}$;
Display();
```

where $\hat{k}$ is the matrix representing the non-standard wavelet expansion of the kernel K. The push and pull can be done in $O(n)$ (linear in the number of elements) steps. The gather step (this is a complete gather sweep which updates all of the entries) can be done in $O(m)$ time where m is the number of terms in the kernel expansion (matrix) that are significant. We want m to be as small as possible. Wavelet bases will lead to $m = O(n)$ where the constant factor in $O(n)$ decreases with the number of vanishing moments.

What remains is to project the kernel into the wavelet basis, which may be done as follows

```
ProjectKernel()
  $k_{\tilde{\phi}_{L,j},\tilde{\phi}_{L,j}}$= Quadrature( $K$, $\tilde{\phi}_{L,j}$, $\tilde{\phi}_{L,m}$ );
  return( NSPyramidUp( $k_{\tilde{\phi}_{L,j},\tilde{\phi}_{L,j}}$ ) );
```

Note that in practice this is not the preferred way of computing the projection of K into the wavelet basis since it requires the computation of n^2 quantities. Thus even if all other steps in the algorithm are $O(n)$ we will already have spent $O(n^2)$ time. Instead of projecting into the finest level and then pyramid transforming up we could start with a projection into the coarsest level and then selectively expand downward, directly computing the non-negligible entries as inner products. Hierarchical radiosity [11] can be considered an application of this method using the Haar basis. This top down approach is described in detail in [9].

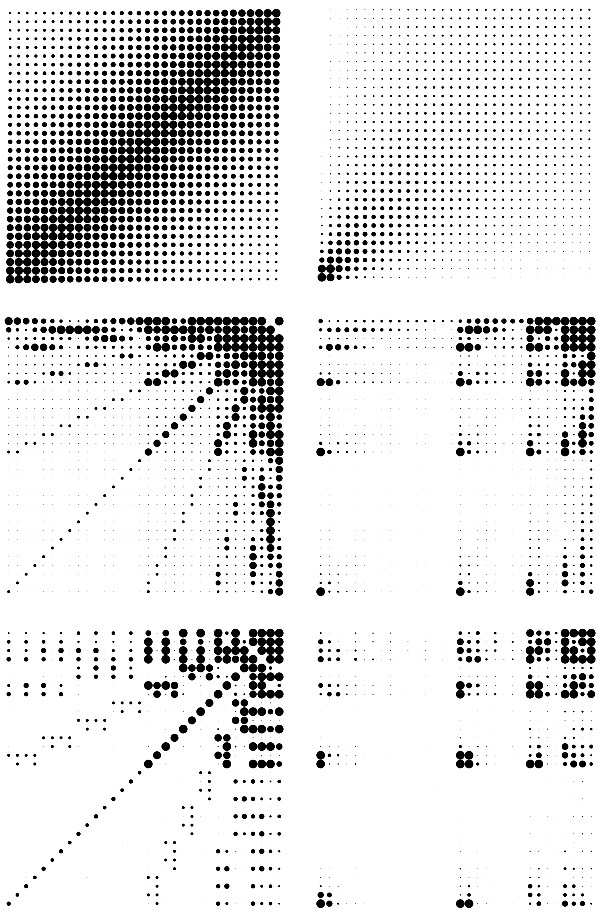

Figure 3: The left column shows the matrix of the flatland kernel for two parallel line segments in the finest level basis, the standard Haar basis and the standard $\mathcal{F}_2$ basis (top to bottom). The right column shows the matrix of the flatland kernel for two line segments meeting at right angles in the same three bases.

6 The Flatland Radiosity Kernel and Wavelets

In this section we illustrate what happens when the kernel from flatland radiosity is projected into wavelet bases. We have investigated the kernel arising from two flatland environments. The environment consisting of two perpendicular unit length line segments meeting in a corner, and the environment of two unit length parallel line segments that are separated by unit length. In the appendix we give proofs that bound the size of the wavelet coefficients of these flatland radiosity kernels. In this section we discuss our experimental results.

In our experiments we started by projecting the kernel functions into the two variable box basis $\phi_{L,l}(t)\phi_{L,j}(s)$ (top row in Figure 3). This is the same basis used in classical radiosity (applied to flatland) [4]. In this basis, all n^2 entries are non-negligible, but the matrix is very coherent. We then performed the **SPyramidUp** algorithms on the matrix which gives us the coefficients with respect to the two argument standard basis. The middle row in Figure 3 shows the result for the Haar basis while the bottom row shows the resulting matrix in the $\mathcal{F}_2$ basis. In the resulting matrices,

[4] In the figures we show only $1/4$ of the full symmetric radiosity matrix.

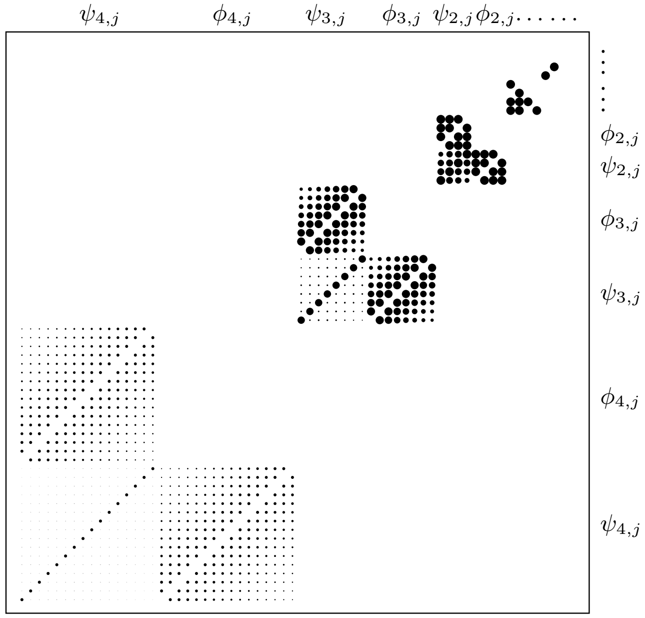

Figure 4: The kernel for two parallel lines realized in the non-standard Haar basis.

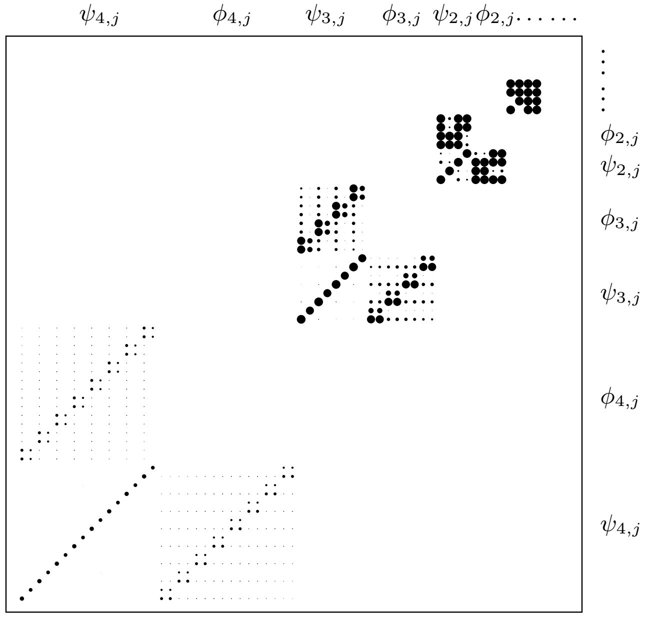

Figure 5: The kernel for two parallel lines realized in the non-standard $\mathcal{F}_2$ basis.

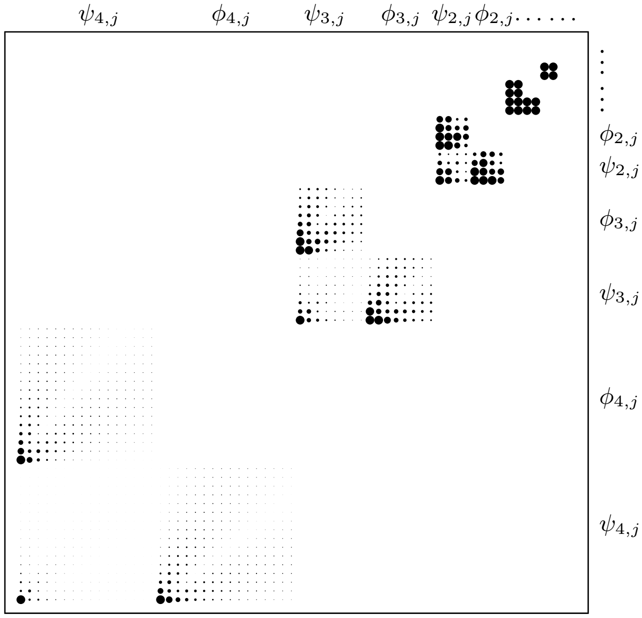

Figure 6: The kernel for two perpendicular lines realized in the non-standard Haar basis.

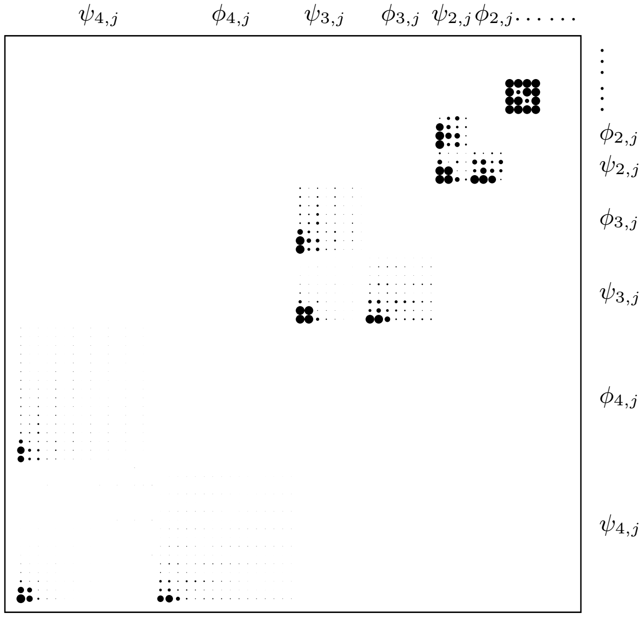

Figure 7: The kernel for two perpendicular lines realized in the non-standard $\mathcal{F}_2$ basis.

some of the coefficients are significant, but most of them are negligible. Note in particular the larger number of negligible entries in the $\mathcal{F}_2$ basis compared to the Haar basis. The large number of negligible coefficients occurs because in most regions of the kernel, the variation in magnitude is well approximated by a linear function in two variables. We also performed the **`NSPyramidUp`** on these matrices. Figures 4, 5, 6, and 7 show the matrix realized in the non-standard Haar and $\mathcal{F}_2$ basis for two parallel and perpendicular lines respectively. Again we find many more entries to be of small magnitude in the $\mathcal{F}_2$ basis as compared to the Haar basis.

Beginning with an original 256 by 256 matrix for each of the configurations, we measured the error incurred when only the m largest matrix entries in each basis were kept and the rest were assumed to be zero. Keeping these m coefficients, we performed the appropriate **`PyramidDown`** on the matrix which effectively gave us an approximate kernel $\hat{K}$. The error using this sparse representation was computed as

$$\frac{\int dt \int ds \mid K(s,t) - \hat{K}(s,t) \mid}{\int dt \int ds \mid K(s,t) \mid}$$

We chose this measure since it describes how well the kernel has been represented *in general*, instead of just measuring the specific error incurred when one has a fixed emission function E. We also measured the error for a series of full matrices with finer and finer resolution $n = 2^0, 2^1, \ldots, 2^8$ for comparision. No change to a wavelet basis was performed on these matrices. The results are plotted in Figure 8 (parallel case) and 9 (perpendicular case).

In both configurations we found that the series of full matrix solutions had the most error per amount of work, although in the parallel configuration the non-standard Haar basis was only better by a constant factor. The $\mathcal{F}_2$ basis performed asymptotically better as seen by the steeper slopes in the graphs. Notice the steeper

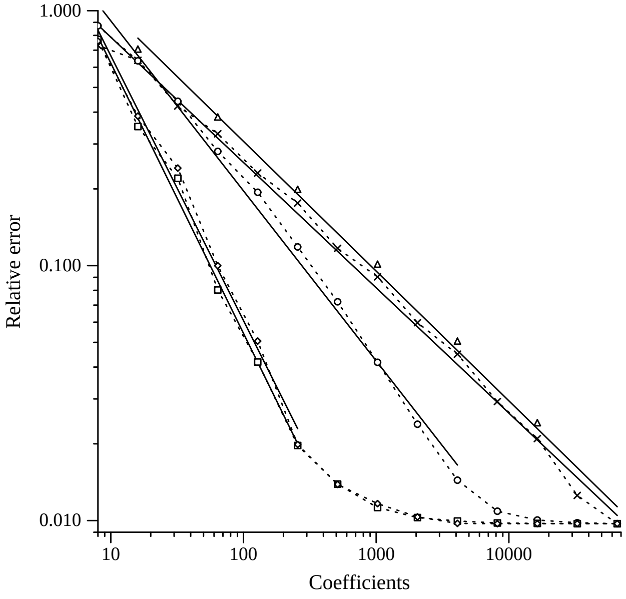

Figure 8: Parallel segments: Relative L^1 error plotted against number of coefficients used with full matrices, non-standard haar, standard haar, non-standard $\mathcal{F}_2$, and standard $\mathcal{F}_2$ bases (top to bottom)

wavelet curves flatten out towards the end. This occurs because as we increase the number of coefficients kept, we eventually approach the 256 by 256 full matrix solution. We have in effect artificially limited the finest level of our representation so that we have a common standard against which to measure our error. In practice we would allow the scales to become finer in a dynamic fashion as higher accuracy is requested, preventing the "bottoming out" of the graphs.

Although the theoretical bounds are stronger for the non-standard basis than for the standard basis [3], we found in our experiments that the standard basis performed slightly better. This is consistent with the experimental results obtained in [13].

The advantage of using wavelets with a higher number of vanishing moments can be clearly seen from the graphs. One might conclude that one should use wavelets with as many vanishing moments as possible. But there are of course tradeoffs involved with going to wavelets with more vanishing moments. Wavelets with more vanishing moments are either of higher polynomial order or are wider. This means that the quadrature required for computing the kernel terms becomes more expensive. Another concern is that as the support of the wavelets increases non-smooth features such as singularities and visibility discontinuities will fall under the support of more basis functions and will result in more non-negligible coefficients. This points to an interesting direction of research. As pointed out in Equation 4 we can force coefficients to be small by either increasing the number of vanishing moments or by decreasing the size of the interval. Ideally then we would like an algorithm which dynamically decides whether to increase M or decrease the interval size to stay within a prescribed error bound.

7 Conclusion

In this paper we have discussed the mathematics of how to obtain radiosity solutions using wavelet projections. As discussed in [9], this allows us to obtain a better theoretical understanding of HR, by casting it as an application of wavelet projections. On the practical side, this framework generates a wide variety of choices of wavelet bases, and allows us to improve on the efficiency of HR.

Future work includes exploring the large space of wavelet bases with an eye towards algorithms that are efficient, accurate and are free of visual artifacts. Clearly the effect on the expansion of the kernel due to complex visibility relationships must be further examined. In addition, to obtain efficient general radiosity algorithms, accurate methods of clustering small polygons together are needed.

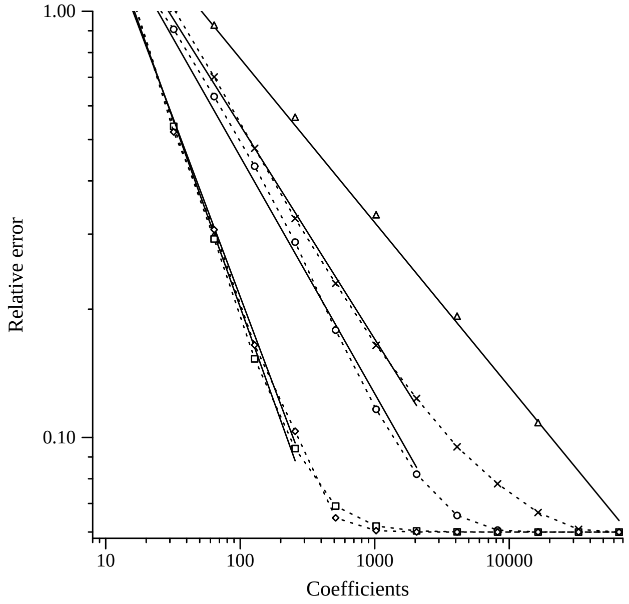

Figure 9: Perpendicular segments: Relative L^1 error plotted against number of coefficients used with full matrices, non-standard haar, standard haar, non-standard $\mathcal{F}_2$, and standard $\mathcal{F}_2$ bases (top to bottom)

Acknowledgements

The authors would like to thank Joel Friedman for many helpful discussions regarding functional analysis, the Scientific Visualization department at HLRZ, Germany where part of this work was performed, and the Chazelle Library for its generous lending policy. The research reported here was partially supported by Apple, Silicon Graphics Computer Systems, and the National Science Foundation (CCR 9207966).

References

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

[2] Alpert, B., Beylkin, G., Coifman, R., and Rokhlin, V. Wavelet-like Bases for the Fast Solution of Second-kind Integral Equations. *SIAM Journal on Scientific Computing 14*, 1 (Jan 1993).

[3] Beylkin, G., Coifman, R., and Rokhlin, V. Fast Wavelet Transforms and Numerical Algorithms I. *Communications on Pure and Applied Mathematics 44* (1991), 141–183.

[4] Chui, C. K. *An Introduction to Wavelets*, vol. 1 of *Wavelet Analysis and its Applications*. Academic Press Inc., 1992.

[5] 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.

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

[7] Delves, L. M., and Mohamed, J. L. *Computational Methods for Integral Equations*. Cambridge University Press, 1985.

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

[9] Gortler, S., Schröder, P., Cohen, M., and Hanrahan, P. Wavelet Radiosity. In *Computer Graphics, Annual Conference Series, 1003* (August 1993), Siggraph.

[10] 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 CG&A.

[11] Hanrahan, P., Salzman, D., and Aupperle, L. A Rapid Hierarchical Radiosity Algorithm. *Computer Graphics 25*, 4 (July 1991), 197–206.

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

[13] 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.

[14] 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.

[15] Schröder, P., and Hanrahan, P. On The Form Factor Between Two Polygons. In *Computer Graphics, Annual Conference Series, 1003* (August 1993), Siggraph.

[16] Zatz, H. R. Galerkin Radiosity: A Higher-order Solution Method for Global Illumination. In *Computer Graphics, Annual Conference Series, 1003* (August 1993), Siggraph.

A Bounding the Wavelet Coefficients of the Radiosity Kernel

At the end of Section 3 we gave a bound for the wavelet coefficients of smooth functions with a single argument (Equation 4). We now give bounds for the wavelet coefficients of two particular kernels of flatland radiosity. The first is given by two perpendicular line segments

$$K(s,t) = \frac{st}{2(s^2+t^2)^{\frac{3}{2}}}$$

where the lines lie on the s and t axes respectively and meet at a right angle at the origin. The kernel for two parallel line segments at unit distance parallel to one of the coordinate axes is given by

$$K(s,t) = \frac{1}{2((s-t)^2+1)^{\frac{3}{2}}}$$

The kernel for perpendicular segments is singular in the corner of the environment, where it has a pole of order 1. For parallel segments we do not have a singularity for real valued arguments. However, when considering Taylor expansions we need to take into account any singularity in the complex plane, which this kernel exhibits. We must do this because the radius of convergence of the Taylor expansion is limited to the largest disk in the complex plane which does not contain the singularity.

We now bound the coefficients $k_{i,j}$ in the wavelet basis just as we did in the case of a function of one argument (Equation 4) using a Taylor expansion. Let (s_0, t_0) be the center of the square over which we are considering the computation of $k_{i,j}$. Since our kernels are symmetric in s and t it is sufficient to show a bound for a powerseries of K with respect to s. The bound with respect to t is given by an identical proof with the roles of s and t reversed. Define $F(s) = K(s_0+s, t_0) = \sum_{n=0}^{\infty} a_n s^n,\ s \in \mathbf{C}$. Let d be the distance from (s_0, t_0) to the nearest (possibly complex) singularity of K.

We can now apply the Cauchy integral formula [8], which gives the coefficients a_n in the powerseries of a complex function F as

$$a_n = \frac{F^{(M)}(0)}{M!} = \frac{1}{2\pi i}\oint_\gamma \frac{F(s)}{s^{M+1}} ds$$

where γ is some path in the complex plane not containing a pole and the integral is a path integral over this curve. We choose this curve to be a circle with radius $R = \alpha d$. Figure 10 shows an example of this with a pole at distance d from the origin. The circle is just inside this disk. Using this we can use the Cauchy integral formula to bound a_n in magnitude

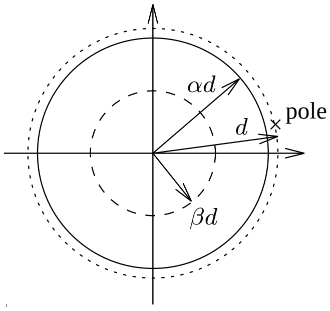

Figure 10: Geometry for the disks of convergence in the complex plane. The pole limits the convergence of the Taylor expansion about the center to a maximum disk with radius d. A circle of radius $R = \alpha d$ is chosen to lie right inside this disk. The interval of integration is assumed to fit into a disk of radius βd.

$$|a_n| = \left|\frac{F^{(M)}(0)}{M!}\right| \leq \left|\frac{\sup_{|s|=R} F(s)}{R^M}\right| = \frac{C_{s_0,t_0}}{R^M}$$

for some finite C_{s_0,t_0}. Now assume that the maximum radius of the interval over which we are integrating is βd, where $\beta < \alpha < 1$. The radius is given by the support of the respective wavelet basis function. This limits the argument s of the power series to $|s| \leq \beta d$. Using both the bound on $|a_n|$ and $|s|$ we may write

$$\begin{aligned}
\left|\int ds\, K(s,t_0)\psi(s)\right| &= \left|\int ds \sum_{n=M}^{\infty} a_n s^n \psi(s)\right| \\
&\leq \int ds \sum_{n=M}^{\infty} |a_n||s|^n|\psi(s)| \\
&\leq C_{s_0,t_0}\int ds \sum_{n=M}^{\infty} \frac{(\beta d)^n}{(\alpha d)^n}|\psi(s)| \\
&= C_{s_0,t_0}\int ds \left(\frac{\beta}{\alpha}\right)^M \sum_{n=0}^{\infty}\left(\frac{\beta}{\alpha}\right)^n |\psi(s)| \\
&= C_{s_0,t_0}\frac{\alpha}{\alpha-\beta}\left(\frac{\beta}{\alpha}\right)^M \int ds\, |\psi(s)| \\
&= \tilde{C}_{s_0,t_0}\frac{\alpha}{\alpha-\beta}\left(\frac{\beta}{\alpha}\right)^M
\end{aligned}$$

Repeating the argument for the parameter t completes the calculation and gives us a bound such that we can use M and/or $\frac{\beta}{\alpha}$ to force $k_{i,j}$ to be small. For example consider rectangles which are "well separated" from the singularity, i.e. their width does not surpass their distance from the singularity, then $\frac{\beta}{\alpha} = \frac{1}{3}$. Similarly we can see that for a fixed width of the interval, $k_{i,j}$ falls off as d^{-M}. For our kernels we can set as (loose) upper bounds

$$d = \sqrt{s_0^2 + t_0^2}$$

for the perpendicular case, and

$$d = |s_0 - t_0|$$

for the parallel case. More general versions of these theorems and their proofs can be found in [3].