

# **Wavelet Radiosity in Computer Graphics**

by

Philipp Ziegler

Dipl.Inform., Universität Kaiserslautern, Germany, 1996

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

**Master of Science**

in

THE FACULTY OF GRADUATE STUDIES

(Department of Computer Science)

we accept this thesis as conforming  
to the required standard

**The University of British Columbia**

September

1998

© Philipp Ziegler, 1998

*In presenting this thesis/essay in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying for this thesis for scholarly purposes may be granted by the Head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.*

Sept 29, 1998

Date

Computer Science  
The University of British Columbia  
2366 Main mall  
Vancouver, BC  
Canada V6T 1Z4

# Abstract

The thesis presents an overview of the recent development of radiosity methods in computer graphics in a uniform mathematical framework. The focus is on hierarchical methods using wavelets.

The thesis experimentally analyzes the behavior of higher-order wavelet bases in hierarchical methods. The functions applied are multiwavelets and a family of wavelets proposed by Cohen, Daubechies and Vial. The latter wavelets have overlapping support. Generally, we find that higher-order wavelet bases save memory compared to Haar wavelets; while they require more time for the computation.

Furthermore, we investigate how Krylov subspace methods can be employed to solve the discrete system of equations arising in hierarchical methods. We show that the Generalized Minimal Residual method (GMRES) is advantageous compared to the usually employed Picard iteration.

# Contents

<b>Abstract</b>	ii
<b>Contents</b>	iii
<b>Acknowledgements</b>	vi
<b>Dedication</b>	vii
<b>1 Introduction</b>	1
<b>2 The Radiosity Problem and Its Numerical Solution</b>	4
2.1 The Radiosity Model . . . . .	4
2.2 Discretizing the Radiosity Equation . . . . .	8
2.2.1 Fundamental Problems . . . . .	8
2.2.2 Basic Methods . . . . .	9
2.2.3 Hierarchical Methods . . . . .	13
2.2.4 Discontinuity Meshing . . . . .	23
2.2.5 Hybrid Methods . . . . .	24
2.3 Solution Methods . . . . .	25
2.3.1 Properties . . . . .	25

2.3.2	Gauss-Seidel-like Methods . . . . .	26
2.3.3	Other Methods . . . . .	33
2.3.4	Comparison . . . . .	34
2.4	Computing the Discrete System . . . . .	35
2.4.1	Form Factor Approximation . . . . .	36
2.4.2	Form Factor Estimation . . . . .	39
2.4.3	Comparison . . . . .	39
2.5	Summary . . . . .	40
<b>3</b>	<b>Exploring Wavelet Radiosity</b>	<b>42</b>
3.1	Wavelet Bases for Radiosity . . . . .	42
3.1.1	Haar Wavelets . . . . .	43
3.1.2	Multiwavelets . . . . .	44
3.1.3	CDV Wavelets . . . . .	46
3.1.4	Comparison . . . . .	48
3.2	Solution Methods for Wavelet Radiosity . . . . .	52
3.3	Implementation Aspects . . . . .	52
3.3.1	Galerkin Discretization . . . . .	53
3.3.2	Form Factor Estimate . . . . .	53
3.3.3	Form Factor Computation . . . . .	54
3.3.4	Memory Representation of the Sparse Matrix . . . . .	55
3.3.5	Solution Method . . . . .	56
<b>4</b>	<b>Numerical Experiments</b>	<b>57</b>
4.1	Comparison of Wavelet Bases . . . . .	57
4.1.1	Test Scenes . . . . .	59

4.1.2	Form Factor Computation . . . . .	59
4.1.3	Measurements . . . . .	60
4.1.4	Interpretation . . . . .	61
4.2	Comparison of Solution Methods . . . . .	67
4.2.1	Test Scenes . . . . .	67
4.2.2	Experiments . . . . .	69
4.2.3	Interpretation . . . . .	72
<b>5</b>	<b>Conclusions</b>	<b>73</b>
5.1	Results . . . . .	73
5.1.1	Wavelet Bases . . . . .	73
5.1.2	Solution Methods . . . . .	74
5.2	For Further Investigation . . . . .	75
<b>Bibliography</b>		<b>76</b>
<b>Appendix A</b>		<b>82</b>
A.1	Two-Dimensional Wavelet Transform . . . . .	82
A.2	Non-Standard Representation . . . . .	83

# Acknowledgements

I would like to thank Professor Uri Ascher for his work as my research supervisor, in particular for his fairness and for providing a very friendly research environment. Professor Alain Fournier gave constructive criticism and pointed out valuable resources. One of these being Ian Ashdown's global illumination bibliography without which writing this thesis would have taken much longer. Last but not least I would like to mention Lorine Chang, who endured all my complaining, Dave Graves, who patiently answered all my questions regarding English style and grammar, as well as Mike Horsch and Ashley Wijeyeratnam, who convinced me that a thesis has to be complete rather than perfect.

PHILIPP ZIEGLER

*The University of British Columbia  
September  
1998*

to

MOUSY

# Chapter 1

## Introduction

The problem of *global illumination* is of major importance in computer graphics.

Global illumination is concerned with computing radiance (i.e. brightness values) in a scene from its geometrical and physical description. This description contains objects (such as walls, furniture etc.) as well as light-sources. Global illumination problems can widely vary in size, ranging from illumination of single rooms with box-like furniture to highly complex scenes of entire buildings with the environment modeled into detail (e.g. [tell94]).

A well known method for the solution of a global illumination problem is ray-tracing, which is concerned with the computation of radiance of the area of the surface of the scene seen from one given viewpoint ([whit80]). In contrast, methods based on Galerkin discretizations (e.g. radiosity methods) compute radiance for each point of the surface. This allows for rapid change between views without expensive re-computations. Solutions of this kind are useful in architectural design and animations. A hybrid method which computes a solution for a restricted set of views is importance-driven radiosity presented in [smit92].

In this thesis we consider the radiosity problem. Surface properties are restricted to Lambertian reflection and light-sources are assumed to be Lambertian emitters."Lambertian"

means incoming light is equally likely to be scattered in any direction independent of the incident direction. So the amount of light reflected is independent of the outgoing direction of the light.

The radiosity problem is modeled by the *radiosity equation*, a Fredholm integral equation of the 2nd kind (e.g. [kaji86]). The kernel of the integral operator of the radiosity problem is positive, very smooth in a wide range of the domain on which it is defined; but has a certain number of discontinuities.

The accurate solution of the radiosity equation is a hard problem mainly because the evaluation of the kernel is very expensive. Currently known techniques are slow, generally inaccurate or may require large amounts of memory.

The solution approach this thesis is concerned with is called *wavelet radiosity* [gort93b], [schr94], [chri94], [gort95], [schr96]. The idea of wavelet radiosity is essentially to employ a Galerkin approach where basis functions are chosen to be wavelets with vanishing moments. The discrete system generated by the Galerkin method is generally a blockwise sparse system of linear equations,  $Mx = b$ . Using wavelet basis functions with the vanishing moment property results in numerous elements of the matrix  $M$  being very small. Wavelet radiosity follows the general method for the solution of integral equations presented by [beyl91] and exploits this property in two ways:

- Small kernel coefficients are set to zero, with the remaining system being sparse. This allows for a fast solution method.
- A hierarchical approach allows not having to compute the small elements of  $M$  at all. This is essential since the computation of each element of  $M$  is costly.

This thesis investigates two components of the wavelet radiosity method:

1. The choice of the wavelet basis. A wavelet radiosity algorithm is implemented in

a very general framework. We use the implementation to analyze the behavior of higher-order wavelet bases compared to the commonly used Haar basis, which consists of piecewise constant functions. The higher-order wavelets we use are multiwavelets [alpe93] and the wavelets presented in [cohe93a]. The latter wavelets have a higher degree of smoothness than multiwavelets.

2. The applicability of Krylov methods as solvers for the sparse system resulting from the wavelet radiosity method and their convergence properties as compared to the commonly used Picard iteration.

We find that higher-order wavelet basis functions tend to have a perceptual advantage over the Haar basis and reduce the required amount of memory. On the other hand the same error in the  $L_1$  norm can be reached faster using a Haar basis.

Our experiments show that the Krylov method GMRES (see e.g. [saad95]) is an easy-to-realize alternative to Picard iteration which in our experiments always performed better than the Picard iteration. Roughly speaking we can reach the same accuracy as the Picard iteration with at most half the number of iterations.

Following this introduction the thesis presents a general formal framework and an overview of existing radiosity methods in Chapter 2. In Chapter 3 we present different types of wavelet bases and solution methods. In Chapter 4 we describe and justify details of our implementation of wavelet radiosity in order to properly specify the numerical experiments conducted. Chapter 5 concludes with a summary and evaluation of the obtained results.

## Chapter 2

# The Radiosity Problem and Its Numerical Solution

### 2.1 The Radiosity Model

The problem of global illumination considered here is how to obtain the *radiance*  $L(\underline{p}, \omega)$  (i.e. angle-dependent emitted power of light per unit area for a fixed wavelength) leaving a point  $\underline{p}$  of the surface  $S$  of a scene at a solid angle  $\omega \in \Omega$  ( $\Omega$  is the set of solid angles). The scene is described by geometric and physical properties. This formulation is view-independent. By computing radiance for a fixed wavelength we assume that computing color values corresponds to the computation of radiance values for a sufficient number of wavelength samples. We assume that emitted radiance is independent of  $\omega$ . This type of reflection is called *Lambertian reflection*. Light-sources with this property are called *Lambertian emitters*. This assumption allows characterizing the illumination of the scene in terms of *radiosity* (i.e. the emitted power of light per unit area). The radiosity  $u : S \rightarrow \mathbb{R}$  is defined as  $u(\underline{p}) = \int_{\Omega} L(\underline{p}, \omega) d\omega$ . We restrict our considerations to interaction between

surface points and exclude effects taking place in the media in which the light is transferred. Furthermore, we will not include translucent objects into our model. For a discussion of these effects, see [lang95] and [rush90].

The mathematical description of this physical model is given by the *radiosity equation*, an integral equation of the second kind:

$$u(\underline{p}) = e(\underline{p}) + \iint_S r(\underline{p}, \underline{q}) u(\underline{q}) dS(\underline{q}) \quad (2.1)$$

We will also write the radiosity equation in the short-hand notation

$$u = e + T_r u \quad (2.2)$$

where  $T_r$  is the corresponding integral operator.

The quantities  $S$ ,  $e$  and  $r$  are defined as follows:

The set  $S \subset \mathbb{R}^3$  is the surface of the scene. The surface is composed of mutually disjoint surfaces  $S_i$ :  $S = \bigcup S_i$ .

The function  $e : S \rightarrow \mathbb{R}$  describes the light-sources of the scene in terms of emitted radiosity.

The support of this function is usually small compared to  $S$  since only few areas of the surface of a scene are light-sources.

The function  $r : S \times S \rightarrow \mathbb{R}$  is the non-negative radiosity kernel given by  $r(\underline{p}, \underline{q}) = \rho(\underline{p})r_1(\underline{p}, \underline{q})$ , where

$$\begin{aligned} r_1(\underline{p}, \underline{q}) &= v_S(\underline{p}, \underline{q})r_2(\underline{p}, \underline{q}) \\ r_2(\underline{p}, \underline{q}) &= \frac{\cos \angle(n_x, \underline{q} - \underline{p}) \cos \angle(n_y, \underline{p} - \underline{q})}{\pi \| \underline{p} - \underline{q} \|_2^2} \end{aligned}$$

where  $v_S$  is the so called *visibility function*. The function  $v_S$  describes the geometry of the scene. It satisfies  $v_S(\underline{p}, \underline{q}) = 0$  if the line segment from  $\underline{p}$  to  $\underline{q}$  intersects with  $S$  (which means that there are obstructions between  $\underline{p}$  and  $\underline{q}$  see Figure 2.1),  $v_S(\underline{p}, \underline{q}) = 1$  otherwise.

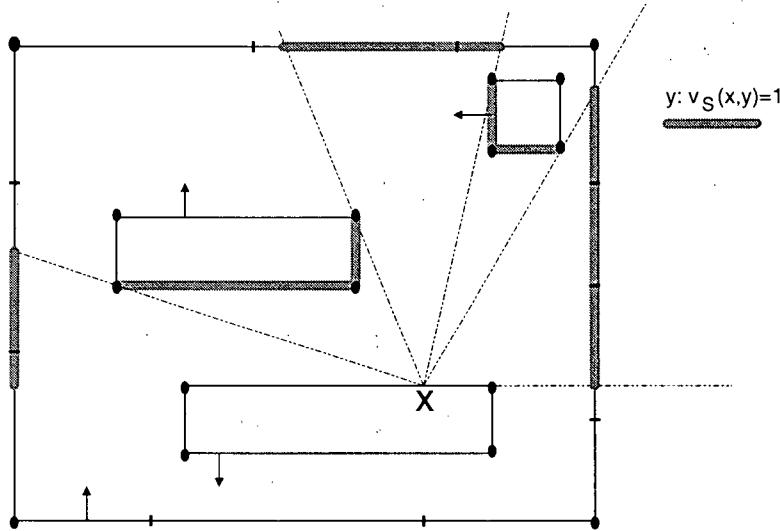


Figure 2.1: Visibility function. The gray line shows points  $\underline{y}$  such that  $v_S(\underline{x}, \underline{y}) = 1$ .

The function  $\rho : S \rightarrow [0, 1]$  is the *reflectivity*. It describes the physical properties of the scene, specifically, it describes the fraction of radiosity reflected in a point of the surface.

The vector  $n_x \in \mathbb{R}^3$  is the normal in  $\underline{p}$  pointing to the thereby defined interior of  $S$ .

Sillion and Puech [sill94] argue based on physical grounds that  $\| T_r \|_{L_2} < 1$ , which guarantees a unique solution for the radiosity equation. A more rigorous analysis is given in [gwin87]. For sufficiently smooth surfaces  $S$  the singularity in  $r$  for  $\underline{p} = \underline{q}$  can be removed by setting  $r(\underline{p}, \underline{p}) = 0$ . The kernel is generally discontinuous due to the visibility function. Since  $e$  is typically discontinuous the solution  $u$  can be discontinuous as well. The computation of the kernel  $r$  is expensive: Naively the cost of computing one value of  $v_S$  is linear in the number of surfaces. This will turn out to be the major challenge in solving the equation.

The solution of equation (2.1) can be expressed in terms of the Neumann series:

$$u = (I - T_r)^{-1}e = \sum_{i=0}^{\infty} (T_r)^i e \quad (2.3)$$

The components  $T_r^i e$  of the sum have an intuitive physical interpretation: they represent the contribution of the light-sources after  $i$  reflections on the surface.

Important effects that cannot be modeled with the radiosity equation are specular highlights and ideal mirrors (due to the assumption of diffuse reflection and Lambertian light-sources [cohe93b]).

The radiosity equation has become a standard model in Computer Graphics. Exhaustive introductions can be found in [cohe93b] and [sill94].

Numerous methods have been suggested to solve the radiosity equation. Those methods typically have four objectives: being fast, accurate, requiring little memory and being robust (i.e., require little user interaction). However, speed and accuracy can have different interpretations in computer graphics than in the common understanding of numerical analysis. A method which obtains a final result more slowly but produces useful intermediate results may be preferable to an overall faster method. Such methods are called *progressive*. A less accurate result which reproduces certain visual effects, like small shadows, well might be more useful than a method with an overall smaller error in a simple error norm.

In the following sections we will describe principles of important deterministic methods for the solution of the radiosity equation. For an overview of stochastic methods we refer to [veac97]. Section 2.2 is concerned with how methods discretize the radiosity function and kernel. Section 2.3 presents methods for the solution of the discretized system. In section 2.4 we discuss how the discretized system itself, the form factors, can be computed. We chose to present the three components of discretization, solution method and form factor computation independently, since it allows us to point out more clearly various advantages and disadvantages of single components of a radiosity method.

## 2.2 Discretizing the Radiosity Equation

Before publications of [gwin87] and especially [heck91], which introduced Galerkin methods for the solution of the radiosity equation, radiosity methods were based purely on physical considerations of transfer of energy between surface patches governed by the principle of energy balance. The presentation here gives a uniform overview in the framework of the notation introduced in subsection 2.2.1. In the following the domain  $S$  will be composed of planar surfaces  $S_i$  (see [scha97] for an overview of methods for curved surfaces).

### 2.2.1 Fundamental Problems

The underlying principle of all the procedures presented here is the Galerkin method. This method approximates the exact solution  $u$  of equation (2.1) by a function  $\tilde{u}$  in a finite dimensional space  $V = \text{span}\{\phi_0 \dots \phi_{n-1}\}$ , where  $\phi_i : S \rightarrow \mathbb{R}$  are linearly independent functions. The support  $A_i$  of each  $\phi_i$ , also called a *patch*, is entirely contained in a surface  $S_i$ . The reflectivity  $\rho$  is assumed to have constant value  $\rho_i$  on patch  $A_i$  (for the efficient treatment of non-constant reflectivities see [gers94]).

We define the projection  $P$  onto  $V$  for  $v \in L_2(S)$ :

$$Pv := \sum_{i=0}^{n-1} \frac{\langle v, \phi_i \rangle}{\langle \phi_i, \phi_i \rangle} \phi_i \quad (2.4)$$

where  $\langle v, w \rangle = \iint_S v(p)w(p)dS(p)$  is the scalar-product in  $L_2(S)$ . It can be shown that the function  $\tilde{v} = Pv$  minimizes  $\| v - \tilde{v} \|_{L_2}$  in  $V$ .

The Bubnov-Galerkin approximation approximates  $u$  by  $\tilde{u} \in V$  such that the residual of the integral equation projected into  $V$  is minimized. This results in the following approximation:

**Definition 2.2.1** *The equation*

$$P\tilde{u} = Pe + PT_rP\tilde{u}$$

is called Bubnov-Galerkin approximation of the integral equation  $u = e + T_r u$ .

With  $\tilde{u} = \sum_{i=0}^{n-1} x_i \phi_i$ ,  $Pe = \sum_{i=0}^{n-1} b_i \phi_i$ , and expansion of the projections, the Bubnov-Galerkin approximation results in a discrete system of equations:

$$x_i = b_i + \sum_{j=0}^{n-1} f_{ij} x_j \quad i = 0, 1, \dots, n-1 \quad (2.5)$$

The coefficients  $f_{ij} = \rho_i g_{ij}$  are called *form factors* where  $G = (g_{ij})_{n \times n}$  is defined as follows:

$$g_{ij} = \frac{1}{\langle \phi_i, \phi_i \rangle} \langle T_r \phi_j, \phi_i \rangle = \frac{1}{\langle \phi_i, \phi_i \rangle} \iint_{A_i} \iint_{A_j} r_1(\underline{p}, \underline{q}) \phi_i(\underline{p}) \phi_j(\underline{q}) dA_j(\underline{q}) dA_i(\underline{p}) \quad (2.6)$$

We will write the system in matrix notation as  $M\underline{x} = \underline{b}$  with  $M = (m_{ij})_{n \times n} = I - F$ ,  $F = (f_{ij})_{n \times n}$ ,  $\underline{b} = (b_0, \dots, b_{n-1})^T$ . We call  $F$  the matrix representation of  $PT_rP$ .

The form factors physically relate to the transfer of energy from patch  $A_i$  to patch  $A_j$ . Their computation is what makes the radiosity problem so time-consuming. The approximation of  $g_{ij}$  involves the computation of samples of the visibility function  $v_S(\underline{p}, \underline{q})$  which is costly (as will be discussed in Section 2.4). Numbers of patches range from about 1000 for simple scenes (e.g. [cohe86]) to  $10^6$  for complex scenes (e.g. [tell94]). A major objective of the methods presented is to compute an approximation  $\bar{M}$  of  $M$  which requires as little of the coefficients  $g_{ij}$  as possible.

## 2.2.2 Basic Methods

### Classical Radiosity

Classical radiosity, also called *full matrix radiosity*, starts with a subdivision of the surface  $S$  into disjoint surface patches  $A_k$ . The operator  $T_r$  is approximated by  $PT_rP$ , hereby directly

realizing the Galerkin approach described in subsection 2.2.1. The mesh has to be adapted to the solution such that shadow boundaries and high radiosity gradients can be adequately represented. The method uses basis functions which are constant on the patches:

$$\phi_k(\underline{p}) = \begin{cases} 1 & : \underline{p} \in A_k \\ 0 & : \text{otherwise} \end{cases} \quad (2.7)$$

With these basis functions we obtain for the form factors:

$$g_{kl} = \frac{1}{|A_k|} \iint_{A_k} \iint_{A_l} r_1(\underline{p}, \underline{q}) dA_l(\underline{q}) dA_k(\underline{p}) \quad (2.8)$$

In case of piecewise constant basis functions the form factors represent the fraction of energy transferred from patch  $k$  to patch  $l$ .

Classical radiosity requires user interaction for the generation of the mesh. The number of form factors that have to be computed is quadratic in the number of patches. The computed solution has discontinuities due to the type of basis functions that we use. These discontinuities are smoothed out in a post-processing step that linearly interpolates between the centers of the computed patches. The post-processing step is not intended to decrease the numerical error but rather aims at making the obtained result visually more pleasing.

Patches can have arbitrary shape and so they can be flexibly adjusted to the problem.

This method was first presented for convex scenes (scenes without obstruction) by [gora84]. It was then generalized to non-convex scenes by [nish85] and [cohe85].

## Substructuring

*Substructuring* aims at reducing the number of form factors to compute. Form factors are initially computed on a coarse mesh. An approximate solution is found on this coarse mesh. Then the solution is transferred to an initial fine mesh, which is subsequently iteratively

refined based on the gradient (e.g. computed by a centered difference) of the solution found so far.

The method uses piecewise constant basis functions  $\phi_{0,k} : 0 \leq k < n_0$  on the coarse level:

$$\phi_{0,k}(\underline{p}) = \begin{cases} 1 & : \underline{p} \in A_{0,k} \\ 0 & : \text{otherwise} \end{cases}$$

On the finer level there are  $n_1 = \sum_{k=0}^{n_0-1} q_k$  piecewise constant basis functions  $\phi_{1,(k,q)} : 0 \leq k < n_0, 0 \leq q < q_k$  such that:

$$\phi_{0,k} = \sum_{q=0}^{q_k-1} \phi_{1,(k,q)}$$

The refinement process is local and the initial mesh is sufficiently fine that  $q_k = 1$  for most  $k$ . The projections  $P_0$  and  $P_1$  are defined as in equation (2.4) based on the basis functions  $\phi_{0,.}$  and  $\phi_{1,.}$  The operator  $P_1 T_r P_0$  is described by the form factors:

$$k_{1,(k,q),l} = 1/|A_{1,(k,q)}| \iint_{A_{1,(k,q)}} \iint_{A_{0,l}} r(\underline{p}, \underline{q}) dA_{0,l}(\underline{q}) dA_{1,(k,q)}(\underline{p})$$

The form factors of  $P_0 T_r P_0$  can be computed based on  $k_{1,(i,q),j}$ :

$$k_{0,i,j} = 1/|A_{0,i}| \sum_{q=0}^{q_i-1} k_{1,(i,q),j} |A_{1,(i,q)}|$$

The approximate solution  $\tilde{u}_1$  is then obtained by first solving the discrete system of equations  $\tilde{u}_0 = P_0 f + P_0 T_r P_0 \tilde{u}_0$  and then transferring this solution onto the fine mesh  $\tilde{u}_1 = P_1 f + P_1 T_r P_0 \tilde{u}_0$ . This latter step is repeated while refining  $P_1$  until the radiosity gradient decreases below a threshold value  $\epsilon$ . In total the second step is just one step of Picard iteration for  $P_1 T_r P_0$  and as such improves the solution  $\tilde{u}_1$  in comparison to  $\tilde{u}_0$ .

The physical intuition of this algorithm is that  $\tilde{u}_0$  is a sufficiently good approximation of the average radiosity on the coarse mesh. This coarse mesh however is not able to reproduce sharp shadow edges. To increase the structure of the radiosity solution within a coarse

patch  $A_{0,i}$  we transfer radiosity from the other coarse patches to the fine patches  $A_{1,(i,q)}$ . Some fine patches might not receive radiosity since they are obstructed by an object. The goal of this operation is to improve the shadow representation.

Substructuring was introduced in [cohe86]. The number of form factors computed is still quadratic,  $\mathbf{O}(n_0^2 + n_0 n_1)$ , however,  $n_0$  is much smaller than in full-matrix radiosity. The process still requires user interaction to determine the coarse mesh and the initial fine mesh. Those meshes have to be fine enough and adapted to the solution. If the the initial fine mesh is not properly adapted to the solution then refinement takes place in areas of small radiosity gradient. If the coarse mesh is not fine enough visual artifacts might appear like the reflection of radiosity from shadows. The final step which transfers the solution onto a fine grid can only redistribute radiosity within a coarse patch but not among coarse patches.

This method is an attempt to approximate the discretized kernel  $P_1 T_r P_1$  (represented by an  $n_1 \times n_1$  matrix) by a sparser kernel requiring fewer form factors (represented by an  $n_0 \times n_0$  and an  $n_1 \times n_0$  matrix). The computation of this approximation is driven by the approximate solution.

## Higher-Order Bases

Galerkin methods which use higher-order basis functions have the potential of decreasing the number of necessary form factors. They provide a better approximation of the radiosity solution in areas where it is smooth. A small number of higher-order basis functions can approximate such a function well, whereas for piecewise constant basis function strong refinement would be necessary. Furthermore, with higher-order basis functions it is possible to use approximation spaces with certain smoothness properties. Such an approximation makes the post-processing step redundant.

While higher-order methods with basis functions with wide support are useful in areas

of smooth variation patches have to be refined carefully around discontinuities.

Higher-order methods have been applied to the radiosity problem by [heck91], [trou93] and [zatz93]. The method presented by [zatz93] uses Legendre polynomials as basis functions and is called *Galerkin radiosity*.

### 2.2.3 Hierarchical Methods

This section gives a brief motivation for hierarchical methods. The general idea of the methods is to find a hierarchical representation of the approximate operator  $\tilde{PT}_rP$ . The hierarchical representation contains form factors between hierarchies of patches. While there will generally be significant light transfer, and consequently big form factors, between big patches which are close to each other, the magnitude of transfer decays when patches are smaller and are further apart. Furthermore we want to find a basis such that form factors between patches with small variation of the kernel are small themselves. We can then set all form factors below a certain threshold  $\epsilon$  to zero, thereby obtaining a sparse representation  $\tilde{M}_\epsilon$ . A particular challenge is to obtain the sparse representation without having to compute all the form factors beforehand, i.e. to decide which form factors would be below the threshold without computing them first.

For the sake of clarity we temporarily switch to the abstract case of a one-dimensional integral equation with kernel  $s : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$  such that the following smoothness requirements are fulfilled:

$$\begin{aligned} |s(x, y)| &\leq \frac{1}{|x-y|} \\ |(\partial_1^v s)(x, y)| + |(\partial_2^v s)(x, y)| &\leq \frac{C_v}{|x-y|^{1+v}} \end{aligned} \tag{2.9}$$

These requirements allow for one discontinuity along the diagonal  $x = y$ . The following discussion follows [beyl91].

## Hierarchical basis functions

We define hierarchical basis functions with certain properties required for the procedure described below.

First we define a hierarchy of finite dimensional spaces  $V_0 \subset V_1 \subset \dots \subset V_m$ . Furthermore we choose spaces  $W_j$  such that ( $\oplus$  is the direct sum):

$$W_j \oplus V_j = V_{j+1}$$

The space  $V_j$  is defined by basis functions  $\phi_{j,k}$ :  $V_j = \text{span}\{\phi_{j,k} : 0 \leq k < n_j\}$ .

Equivalently, we choose basis functions  $\psi_{j,k}$ :  $0 \leq k < n_j$  for the spaces  $W_j$ .

The hierarchy of spaces implies a hierarchy of basis functions. This hierarchy is expressed by the *dilation equation*:

$$\begin{aligned} \phi_{j,k} &= \sum_{l=0}^{n_{j+1}-1} h_{k,l} \phi_{j+1,l} \\ \psi_{j,k} &= \sum_{l=0}^{n_{j+1}-1} g_{k,l} \phi_{j+1,l} \end{aligned} \quad (2.10)$$

We represent functions  $v_j \in V_j$ ;  $v_j = \sum_{k=0}^{n_j-1} c_{j,k} \phi_{j,k}$ :  $0 \leq j \leq m$ , by the vector of coefficients  $\underline{c}_j = (c_{j,k})_{0 \leq k < n_j}$  and functions  $w_j \in W_j$ ;  $w_j = \sum_{k=0}^{n_j-1} d_{j,k} \phi_{j,k}$ :  $0 \leq j < m$ , by the vector of coefficients  $\underline{d}_j = (d_{j,k})_{0 \leq k < n_j}$ .

If we start up with a function  $v_m$  represented by  $\underline{c}_m$  in the finest space  $V_m$  then we can successively decompose it in the form  $v = w_{m-1} + w_{m-2} + \dots + w_0 + v_0$ . The computation of  $\underline{c}_j$ :  $0 \leq j < m$  and  $\underline{d}_j$ :  $0 \leq j < m$  from  $\underline{c}_m$  is called a *wavelet decomposition*. If the functions  $\phi_{j,k}$  are orthogonal for a fixed  $j$  then the computation of  $\underline{c}_m$  from given  $\underline{c}_j$ :  $0 \leq j < m$  and  $\underline{d}_j$ :  $0 \leq j < m$  is possible. This reverse operation is called a *wavelet reconstruction*. If we require that for fixed  $k$  only a constant number of the coefficients  $h_{k,l}$  and  $g_{k,l}$  be non-zero then both computations can be conducted in a straightforward fashion in  $\mathcal{O}(n_m)$  operations via the fast wavelet transform.

We can represent the wavelet decomposition by a non-singular square matrix  $D_{S,m}$ .

This allows expressing the decomposition as:

$$\begin{pmatrix} \underline{c}_0 \\ \underline{d}_0 \\ \underline{d}_1 \\ \vdots \\ \underline{d}_{m-1} \end{pmatrix} = D_{S,m} \underline{c}_m \quad (2.11)$$

Analogously, a decomposition which includes  $\underline{c}_j$  on all levels is denoted by  $D_{N,m}$ . The matrix  $D_{N,m}$  is non-square. A reconstruction which adds up the functions represented by  $\underline{c}_j$  and  $\underline{d}_j$  and returns a representation  $\underline{c}_m$  is denoted by  $D_{N,m}^-$ :

$$\begin{pmatrix} \underline{c}_0 \\ \underline{d}_0 \\ \vdots \\ \underline{c}_{m-1} \\ \underline{d}_{m-1} \end{pmatrix} = D_{N,m} \underline{c}_m \quad \underline{c}_m = D_{N,m}^- \begin{pmatrix} \underline{c}_0 \\ \underline{d}_0 \\ \vdots \\ \underline{c}_{m-1} \\ \underline{d}_{m-1} \end{pmatrix} \quad (2.12)$$

The matrices  $D_{S,m}, D_{S,m}^{-1}, D_{N,m}$  and  $D_{N,m}^-$  are never formed explicitly. They merely serve as a convenient notation for wavelet decomposition and reconstruction. Note that  $D_{S,m}$  transfers the representation  $\underline{c}_m$  of  $v \in V$  into a different basis while the representation obtained by  $D_{N,m}$  is not a basis representation at all. For this reason  $D_{S,m}$  is called a *standard decomposition*.  $D_{N,m}$  is called a *non-standard decomposition*.

Analogously to equation (2.4) we define the projections  $P_j$  onto  $V_j$  and  $Q_j$  onto  $W_j$ . If we assume orthonormality of the functions  $\phi_{j,k}$  for fixed  $j$ , i.e.  $\langle \phi_{j,k}, \phi_{j,k} \rangle = 1, \langle \phi_{j,k}, \phi_{j,l} \rangle = 0$  for  $k \neq l$ , then  $P_{j+1} = P_j + Q_j$ .

It is possible to define sets of functions  $\phi_{j,k}$  and  $\psi_{j,k}$  such that:

$$\int \psi_{j,k}(x) x^\mu dx = 0 \quad \forall \mu : 0 \leq \mu < \nu$$

The function  $\psi_{j,k}$  is then said to have  $\nu$  *vanishing moments*. We will give examples of such functions in Section 3.1.

The functions  $\phi_{j,k}$  and  $\psi_{j,k}$ , which we have defined here, fit into the framework of orthogonal wavelets discussed, e.g., in [daub88] and [jawe94]. In this framework the functions  $\psi_{j,k}$  are called *wavelets*, and the functions  $\phi_{j,k}$  are called *scaling functions*. The pair of wavelets and scaling functions is called *multiresolution analysis* (MRA).

We ignore the more general framework of biorthogonal wavelets (e.g. [jawe94]) since we do not apply it in this thesis.

## Wavelet Compression of Integral Operators

With the above properties we can express  $P_m$  as:

$$P_m = P_0 + \sum_{j=0}^{m-1} Q_j$$

As a consequence we can rewrite  $P_m T P_m$ :

$$P_m T P_m = P_0 T P_0 + P_0 T \left( \sum_{j=0}^{m-1} Q_j \right) + \left( \sum_{j=0}^{m-1} Q_j \right) T P_0 + \left( \sum_{j=0}^{m-1} Q_j \right) T \left( \sum_{j=0}^{m-1} Q_j \right) \quad (2.13)$$

According to equation (2.13) another way to write  $P_m T P_m v$  in matrix notation is

$$\underline{c}'_m = D_{S,m}^{-1} F_S D_{S,m} \underline{c}_m$$

where the matrix  $F_S$  is defined as

$$F_S = \begin{pmatrix} F^{(P_0, P_0)} & F^{(P_0, Q)} \\ F^{(Q, P_0)} & F^{(Q, Q)} \end{pmatrix}$$

and  $F^{(P_0, P_0)}$  with entries of the form  $\langle T\phi_{0,l}, \phi_{0,k} \rangle$ ,  $F^{(P_0, Q)}$  with entries  $\langle T\psi_{j,l}, \phi_{0,k} \rangle$ ,  $F^{(Q, P_0)}$  with entries  $\langle \phi_{0,k}, T\psi_{j,l} \rangle$ , and  $F^{(Q, Q)}$  with entries  $\langle T\psi_{j,l}, \psi_{i,k} \rangle$ . The matrix  $F_S$  is called the *standard representation* of  $P_m T P_m$ .

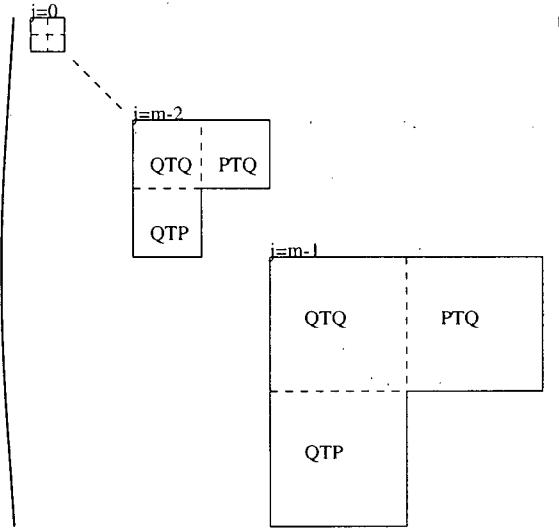


Figure 2.2: Non-standard representation  $F_N$  of  $P_mTP_m$ .

Another way to rewrite  $P_mTP_m$  (see Appendix A.2) is:

$$P_mTP_m = P_0TP_0 + \sum_{j=0}^{m-1} Q_jTQ_j + \sum_{j=0}^{m-1} Q_jTP_j + \sum_{j=0}^{m-1} P_jTQ_j \quad (2.14)$$

With equation (2.14) we get the *non-standard representation* of  $P_mTP_m$ :

$$\underline{c}'_m = D_{N,m}^- F_N D_{N,m} \underline{c}_m$$

The matrix  $F_N$  is composed of blocks  $F^{(P_0, P_0)} = (\langle T\phi_{0,i}, \phi_{0,k} \rangle)_{0 \leq k < n_0, 0 \leq i < n_0}$ ,  $F^{(P_j, Q_j)} = (\langle T\phi_{j,l}, \psi_{j,k} \rangle)_{0 \leq k < n_0, 0 \leq l < n_0}$ , and  $F^{(Q_j, Q_j)} = (\langle T\psi_{j,l}, \psi_{j,k} \rangle)_{0 \leq k < n_0, 0 \leq l < n_0}$ .

The structure of  $F_N$  is shown in Figure 2.2.

Analyzing the sparsity of the matrices  $F$ ,  $F_S$  and  $F_N$  we obtain the following numbers of non-zero entries for the typical case of a *dyadic hierarchy* ( $n_{j+1} = 2n_j$ ):

representation	matrix	non-zero entries
full matrix	$F$	$n_m^2$
standard	$F_S$	$n_m^2 - n_0^2 = n_m^2(1 - 2^{-2m})$
non-standard	$F_N$	$n_m^2$

In terms of sparsity the wavelet representation has not brought any advantage in comparison to the full matrix representation. However, if we use wavelets with vanishing moments we see that entries  $\langle T\psi_{j,k}, \varphi \rangle$  and  $\langle T\varphi, \psi_{j,k} \rangle$  will be small if  $\psi_{j,k}$  has its support in an area where the kernel is sufficiently smooth. The following result is presented in [beyl91]:

**Theorem 2.2.2** *Assume an integral operator  $T$  with kernel  $s$  is given which satisfies equation (2.9). Consider  $\epsilon > 0$  and wavelets with local support. Then:*

- only  $\mathbf{O}(n_m \log(n_m))$  of the entries of  $F_S$  are larger than  $\epsilon$
- only  $\mathbf{O}(n_m)$  of the entries of  $F_N$  are larger than  $\epsilon$

The intuition behind this result is that only those coefficients which are based on wavelets that are close to or overlap the discontinuity along the diagonal  $x = y$  will have significant value. The enumeration of such wavelets for the case of the Haar MRA is shown in Figure 2.3 and Figure 2.4.

For the standard representation we can compute a matrix  $F_{S_\epsilon}$  by setting elements of  $F_S$  which are smaller than  $\epsilon$  to zero. This procedure is called kernel compression. We denote the operator represented by  $D_{S,m}^{-1} F_{S_\epsilon} D_{S,m}$  with  $T_{S_\epsilon}$ . For  $v \in V_m$  this procedure allows us to compute an approximation of  $P_m T P_m v$  with only  $\mathbf{O}(n_m \log(n_m))$  entries in comparison to  $\mathbf{O}(n_m^2)$  entries of the full matrix approach. If  $\tilde{u}$  is the solution of the integral equation  $P_m \tilde{u} = P_m e + P_m T P_m \tilde{u}$  and  $\tilde{u}_\epsilon$  is solution of  $P_m \tilde{u}_\epsilon = P_m e + T_{S_\epsilon} \tilde{u}_\epsilon$  then it can be easily shown that  $\| \tilde{u} - \tilde{u}_\epsilon \|_2 = c_{n_m} \epsilon$ , for sufficiently small  $\epsilon$  (e.g. [atki76]). The same holds for the non-standard representation, however, the operator here has only  $\mathbf{O}(n_m)$  entries.

To summarize we have found approximations to the matrix  $F$ ,  $D_{S,m}^{-1} F_{S_\epsilon} D_{S,m}$  (and  $D_{N,m}^{-1} F_{N_\epsilon} D_{N,m}$ ), which allow computing a matrix vector product in  $\mathbf{O}(n_m \log(n_m))$  (and

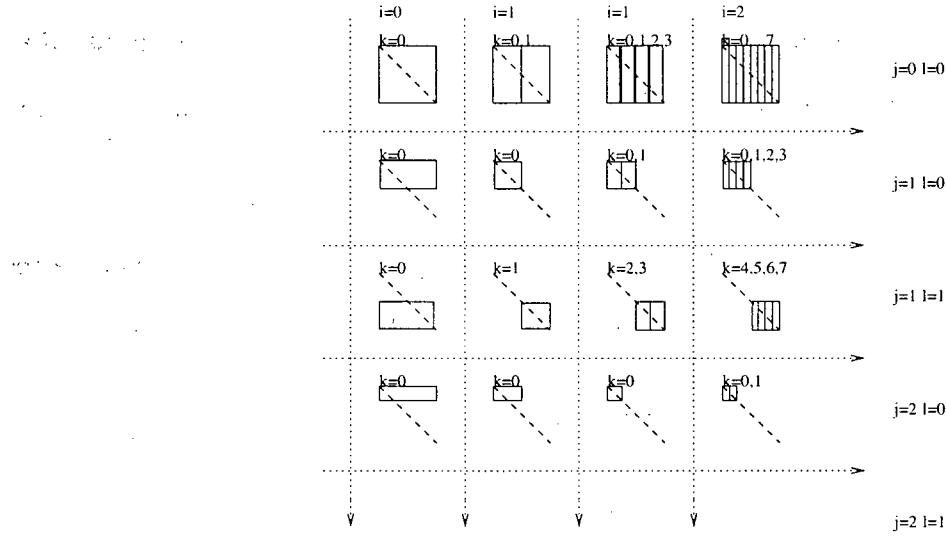


Figure 2.3: Standard representation, Haar MRA: supports of  $h(x, y) = \psi_{i,k}(x)\psi_{j,l}(y)$  for  $i, j, k, l$  such that the support of  $h$  overlaps the discontinuity.

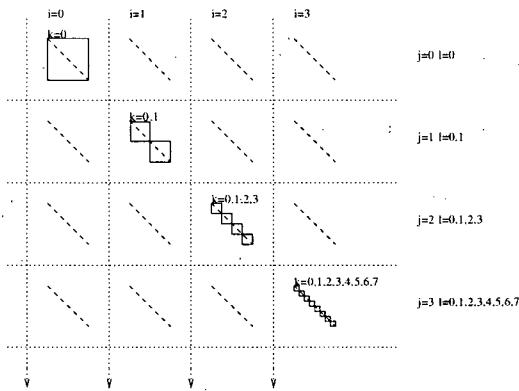


Figure 2.4: Non-standard representation, Haar MRA: supports of  $h(x, y) = \psi_{i,k}(x)\psi_{j,l}(y), i = j$  for  $i, j, k, l$  such that the support of  $h$  overlaps the discontinuity.

$\mathbf{O}(n_m)$ ) operations. For this method to work we need hierarchical basis functions with the following properties:

- **Wavelets with vanishing moments.** This property is the main requirement necessary for compression. It guarantees that coefficients in areas where the kernel is smooth are small.
- **Local support.** Local support is necessary to guarantee a large number of wavelets with support in areas where the kernel is smooth, i.e. away from the discontinuity.
- **Dilation equation.** The basis functions must fulfill equation (2.10) with a finite number of non-zero coefficients  $h_{k,l}$  and  $g_{k,l}$ . This property is necessary to be able to compute the decomposition  $D_{S,m}$  and  $D_{N,m}$  in  $\mathbf{O}(n_m)$  operations.

Technically the application of the method presented here to the radiosity equation is straightforward with two-dimensional scaling functions and wavelets (see Appendix A.1). Schröder [schr94] shows that the same complexity bounds hold by extending Thm. 2.2.2 to the integral operator that appears in the radiosity equation.

### Computation of the Compressed Operator

In the framework of hierarchical basis functions the coefficients  $\langle T\psi_{j,l}, \psi_{j,k} \rangle$ ,  $\langle T\psi_{j,l}, \phi_{j,k} \rangle$ ,  $\langle T\phi_{j,l}, \psi_{j,k} \rangle$  and  $\langle T\phi_{j,l}, \phi_{j,k} \rangle$  for  $k, l : 0 \leq k, l < n_j$  can be computed recursively using the

dilation equation:

$$\begin{aligned}
\langle T\psi_{j,l}, \psi_{j,k} \rangle &= \sum_{\bar{l}=0}^{n_{j+1}-1} \sum_{\bar{k}=0}^{n_{j+1}-1} g_{l,\bar{l}} g_{k,\bar{k}} \langle T\phi_{j+1,\bar{l}}, \phi_{j+1,\bar{k}} \rangle \\
\langle T\psi_{j,l}, \phi_{j,k} \rangle &= \sum_{\bar{l}=0}^{n_{j+1}-1} \sum_{\bar{k}=0}^{n_{j+1}-1} g_{l,\bar{l}} h_{k,\bar{k}} \langle T\phi_{j+1,\bar{l}}, \phi_{j+1,\bar{k}} \rangle \\
\langle T\phi_{j,l}, \psi_{j,k} \rangle &= \sum_{\bar{l}=0}^{n_{j+1}-1} \sum_{\bar{k}=0}^{n_{j+1}-1} h_{l,\bar{l}} g_{k,\bar{k}} \langle T\phi_{j+1,\bar{l}}, \phi_{j+1,\bar{k}} \rangle \\
\langle T\phi_{j,l}, \phi_{j,k} \rangle &= \sum_{\bar{l}=0}^{n_{j+1}-1} \sum_{\bar{k}=0}^{n_{j+1}-1} h_{l,\bar{l}} h_{k,\bar{k}} \langle T\phi_{j+1,\bar{l}}, \phi_{j+1,\bar{k}} \rangle
\end{aligned} \tag{2.15}$$

The algorithm outlined above in the presented form is not useful for radiosity problems. It reduces the cost for solving the discrete system to  $\mathbf{O}(n_m)$ , but the cost of computing  $F_{S_\epsilon}$  or  $F_{N_\epsilon}$  still requires  $\mathbf{O}(n_m^2)$  operations. Furthermore, in radiosity problems the time spent on the solution of the system is usually negligible compared to the time it takes to compute the form factors. Therefore we have to find a method which allows us to compute all form factors above the threshold  $\epsilon$  without having to compute those which are smaller.

The ideas proposed in [hanr91] (for piecewise constant basis functions) and [gort93b] (for higher-order basis functions) are based on a method known from n-body problems presented in [appe85]. The method computes form factors starting at the coarsest level. If a form factor between patches  $p$  and  $q$  is above the threshold  $\epsilon$  then we compute the form factors between all children of  $p$  and all children of  $q$ . This process is repeated recursively for the children. If a form factor is below the threshold  $\epsilon$  then we stop refining. This procedure assumes that if a form factor between patches  $p$  and  $q$  is small then this also will hold for form factors between subpatches of  $p$  and  $q$ .

In addition it is too expensive to compute the form factors before comparing them. On coarse levels this would require a high number of kernel samples for the quadrature used for their approximation. Computing all these samples is exactly what the hierarchical

method tries to prevent. To avoid the full computation of form factors a procedure called an *oracle* is applied which estimates form factors based on a small number of samples or geometrical properties. These estimates are discussed in Section 2.4.2. Refinement in the recursive procedure is then based on the estimate rather than a computation of the form factor.

## Clustering

Hierarchical methods require computation of the form factors of the operator  $P_0 T_r P_0$ , i.e. the form factors between basis functions on the coarsest level. Supports of basis functions on the coarsest level are located within the surfaces that the scene is composed of. If  $s$  is the number of surfaces in a scene the complexity for the computation of the form factors on the coarsest level is  $\mathbf{O}(s^2)$ . Smits et. al. [smits94] present a *clustering algorithm* which reduces the cost to  $\mathbf{O}(s)$ . They use an octree to hierarchically cluster surfaces. The energy transfer between these clusters is estimated. Pairs of clusters with energy transfer above a certain threshold are hierarchically refined until the level of a single surface is reached. On that level the algorithm continues with the hierarchical method outlined in 2.2.3. The clustering technique can be applied in a framework with piecewise constant scaling functions since here form factors relate to the energy transfer between patches.

## Summary

Hierarchical methods with hierarchical basis functions as required in subsection 2.2.3 find an approximation to the operator  $P_{n_m} T P_{n_m}$  of arbitrary accuracy with  $\mathbf{O}(n_m)$  form factors. The accuracy is proportional to the threshold  $\epsilon$ . In an implementation these rigorous bounds do no longer hold due to the estimates used for a top-down computation of the form factors and the termination of recursion at a certain level. The meshing process is fully automatic up to

the selection of the threshold parameter  $\epsilon$  and the number of levels. These two parameters provide the user with an easy way to trade speed for accuracy.

The realization with standard representation has a theoretical disadvantage because it requires  $\mathbf{O}(n_m \log(n_m))$  form factors. However, it allows computations without the normalization and denormalization steps ( $D_{N,m}^{-1}$ ,  $D_{N,m}$ ) since  $D_{S,m}$  is applied to a basis representation of  $\tilde{u}$  and returns such a representation. The transformation into this representation is applied once in the beginning of the algorithm, the back-transformation applied once at the end.

Due to the hierarchical structure of the procedure, form factors cannot be computed independently. This implies that all form factors have to be stored until the final radiosity solution has been found. This typically results in high memory requirements.

A hierarchical method with piecewise constant scaling functions was first presented in [hanr91] under the name *hierarchical radiosity*. The general framework of wavelet methods for integral equations was presented by [beyl91]. Gortler et. al. [gort93b] applied that method to the radiosity problem and called it *wavelet radiosity*. While [gort93b] use the non-standard representation [chri94] and [stol96] describe a wavelet method using the standard representation. This method needs only one wavelet decomposition step in the beginning and one final wavelet reconstruction step. The entire solution procedure takes place with a representation of the unknown in the wavelet basis, not in a basis of scaling functions as with the non-standard decomposition.

## 2.2.4 Discontinuity Meshing

*Discontinuity meshing* constructs an appropriate basis based on the geometry of the scene. One computes potential locations of shadow boundaries by intersecting planes which touch edges of surfaces and corner vertices of light-sources with other surfaces in the scene. The lines generated by the intersection are potential locations of discontinuities. This process is

iterated using the newly generated lines in place of the light-sources.

The areas generated by these intersections are used as supports for piecewise constant basis functions in the Galerkin discretization.

Heckbert [heck92] and Lischinski et. al. [lisc92] presented the first algorithms for discontinuity meshing. Discontinuity meshing can accurately resolve shadow boundaries. Following [lisc93] the number of basis functions computed in discontinuity meshing is  $\mathbf{O}(l^2s^2)$  if  $l$  is the number of light-sources and  $s$  is the number of surfaces in the scene. In [lisc93] discontinuity meshing is regarded as an expensive process.

### 2.2.5 Hybrid Methods

Lischinski et. al. [lisc93] combine hierarchical methods with piecewise constant basis functions and discontinuity meshing. They build the hierarchy of patches by subdividing along potential discontinuities. This process helps to terminate refinement on a coarse level since the number of basis functions overlapping discontinuities is efficiently reduced.

Bouatouch et. al. [boua95] experiment with a basis derived from multiwavelets which allows them to incorporate discontinuity information. Assume a discontinuity dividing the support  $A_{j,k}$  of a basis function  $\phi_{j,k}$  into sets  $A_{j,k}^1$  and  $A_{j,k}^2$ . Then  $\phi_{j,k}$  is replaced by two functions  $\phi_{j,k}^1$  and  $\phi_{j,k}^2$  which are obtained by restricting the support of  $\phi_{j,k}$  to  $A_{j,k}^1$  and  $A_{j,k}^2$ . This newly established basis is not orthogonal anymore resulting in more complicated expressions for wavelet decomposition and reconstruction.

Bekaert [beka96] presents higher-order hierarchical basis functions with non-rectangular support which appear to be useful for combining discontinuity meshing and hierarchical methods.

## 2.3 Solution Methods

We do not consider direct methods in this section since in the context of the radiosity equation they are clearly inferior to iterative methods. The advantage of iterative methods here stems less from the desire of exploiting sparsity than from the strong diagonal dominance of the discretized system, which implies fast convergence of simple iterative methods. Moreover, the elements of the matrix relate to a discretization of smooth quantities, raising the prospect of efficient multilevel treatment.

### 2.3.1 Properties

Consider the linear system of equations (2.5) written as  $M\underline{x} = \underline{b}$ , where  $M = I - RG$  with  $R = \text{diag}(\rho_0, \dots, \rho_{n-1})$  and  $G$  as introduced in equation (2.6). As mentioned in 2.2.1 the size of  $M$  can become too large to fit into a computer's primary memory.

$M$  is blockwise sparse if the scene has obstructions such that entire patches are not visible from each other. Sparsity is significant if the scene is complex, i.e. if it is composed of different isolated rooms.  $M$  is generally not symmetric. However, it can be easily made symmetric by multiplication with  $D_A R^{-1}$ , where  $D_A = \text{diag}(\langle \phi_0, \phi_0 \rangle, \dots, \langle \phi_{n-1}, \phi_{n-1} \rangle)$ . Also, by definition two points in a plane are not visible from each other, so  $g_{kk} = 0$  and hence  $m_{kk} = 1, k := 0 \leq k < n$ .

Now we restrict our considerations to the case of piecewise constant basis functions.

Here the identity  $d\omega(\underline{x}) = \frac{\cos \angle(n_y, \underline{x} - \underline{y})}{\| \underline{x} - \underline{y} \|_2^2} dA(\underline{y})$  ( $\omega$  is the solid angle) yields  $\sum_l g_{kl} = 1$  (e.g. [cohe93b]). Since  $0 \leq \rho_k < 1$ ,  $M$  is strictly row diagonally dominant. Furthermore [neum95] and [niev97] show that  $D_A R^{-1} M$  is not only symmetric but also positive definite (SPD). This property can be used to solve the SPD system  $D_A R^{-1} M \underline{x} = D_A R^{-1} \underline{b}$  instead. However, this method does not allow for very small or zero  $\rho_i$ , which is a significant drawback since such

situations can easily arise in computer graphics when objects are dark or black.

## Summary

The system  $M$  arising from a Galerkin discretization of the radiosity equation has the following properties:

1. moderate sparsity;
2. it can be made symmetric by multiplication with the diagonal matrix  $D_A R^{-1}$  if  $\rho_k > 0, k : 0 \leq k < n$ ;
3.  $m_{kk} = 1, k : 0 \leq k < n$ .

Furthermore, for a discretization with piecewise constant basis functions the following holds:

1. row diagonal dominance:  $\forall i : |m_{jj}| > \sum_{i \neq j} |m_{ij}|$
2.  $m_{ij} \leq 0, i \neq j$
3.  $D_A R^{-1} M$  is SPD (if  $\rho_k > 0, k : 0 \leq k < n$ )

### 2.3.2 Gauss-Seidel-like Methods

In the following discussion we will use the terms *update* and *sweep*. The term update refers to the change of a single variable  $x_i$  while a sweep means an update of all components of  $\underline{x}$ .

#### Jacobi Iteration

One sweep of Jacobi iteration can be written as:

$$\underline{x}^{(k+1)} = \text{diag}(M)^{-1} \underline{b} + (I - \text{diag}(M)^{-1} M) \underline{x}^{(k)}$$

In case of the radiosity equation  $\text{diag}(M) = I$ . Recalling  $F = RG$  we obtain the simpler form:

$$\underline{x}^{(k+1)} = \underline{b} + F\underline{x}^{(k)} \quad (2.16)$$

This form is also known as one step of *Picard iteration* (e.g. [hack95]). Picard iteration simply approximates the Neumann series from equation (2.3). So each sweep corresponds to one additional level of reflection. If we choose  $\underline{x}^{(0)} = \underline{b}$  then  $F^k \underline{b}$  is the contribution of the light-source after  $k$  reflections. Also note from equation (2.16) that one step of the Picard iteration is equivalent to the addition of the current residual  $\underline{r}^{(k)} = \underline{b} - M\underline{x}^{(k)}$  resulting in  $\underline{x}^{(k+1)} = \underline{x}^{(k)} + \underline{r}^{(k)}$ .

### Gauss-Seidel Iteration

Let  $l$  count single updates rather than sweeps. Then Gauss-Seidel iteration updates the unknowns  $x_i$  in the following way:

$$x_i^{(l+1)} := \begin{cases} (b_i - \sum_{j \neq i} m_{ij} x_j^{(l)}) & i = \hat{i}(l) \\ x_i^{(l)} & i \neq \hat{i}(l) \end{cases} \quad (2.17)$$

The index  $\hat{i}(l)$  specifies which variable is updated in the  $l$ -th step. The usual order is:  $(\hat{i})_{l \in \mathbb{N}} = (0, 1, 2, \dots, n-1, 0, 1, 2, \dots, n-1, \dots)$ . To incorporate the residual  $\underline{r}^{(l)}$  explicitly equation (2.17) can be easily rewritten yielding the following algorithm:

#### Algorithm 2.3.1 Gauss-Seidel Update( $M, \underline{b}, i, \underline{x}, \underline{r}$ )

$i := \text{next } i \text{ in turn}$

$x_i := x_i + r_i$

$\forall j : j \neq i \text{ do:}$

$r_j := r_j - m_{ji} r_i$

```

 $r_i := 0$ 
return ( $i, \underline{x}, \underline{r}$ )

```

If we choose  $\underline{x}^{(0)} = \underline{0}$  as the initial guess we can compute the initial residual:  $\underline{r}^{(0)} = \underline{b}$ .

This gives us all the initial parameters we need for the algorithm. For one update of the approximate solution and for one update of the residual only one column of  $M$  is necessary.

In the radiosity context equation (2.17) has a simple physical interpretation: in one Gauss-Seidel update light from all patches  $j$  is gathered and accumulates in patch  $i$ . The residual  $\underline{r}^{(k)}$  can be written as  $\underline{r}^{(k)} = M(\underline{x} - \underline{x}^{(k)})$  where  $\underline{x}^{(k)}$  is the current approximate solution at any give stage. The quantity  $\underline{x} - \underline{x}^{(k)}$  is the energy surplus in the scene (we show later that this quantity is always positive for piecewise constant basis functions). For that reason the residual is also imprecisely called *unshot energy* in the graphics literature. Gauss-Seidel in the graphics literature is also often referred to as *gathering*.

### **Southwell Iteration**

The Southwell iteration ([sout46]) is a modification of Alg. 2.3.1. Here the index  $i$  of the next variable to update is not just chosen “in turn” but it is chosen such that  $|r_i|$  is maximized. Convergence of this method for the radiosity problem is shown in [gort93a].

### **Progressive Refinement Iteration**

Progressive refinement (PR) is a modification of the Southwell iteration. Here a certain number of Southwell updates is performed. After this the residual is added to the solution  $\underline{x}^{(k)}$  obtained so far and the sum is returned as the approximate result  $\hat{\underline{x}} = \underline{x}^{(k)} + \underline{r}^{(k)}$ . This final step corresponds to exactly one complete sweep of the Jacobi iteration based on  $\underline{x}^{(k)}$ . For this sweep no new entries of  $M$  have to be computed.

We rewrite the algorithm maintaining the invariant  $\hat{\underline{x}}^{(k)} = \underline{x}^{(k)} + \underline{r}^{(k)}$  ( $\hat{\underline{x}}^{(k)}$  is the iteration variable in PR). This form of the PR algorithm gives it an immediate physical interpretation. The approximation  $\hat{\underline{x}}^{(k)}$  is more accurate than  $\underline{x}^{(k)}$ . We can use  $\hat{\underline{x}}^{(k)}$  for display purposes after each update.

**Algorithm 2.3.2 Progressive Refinement Update( $M, \underline{b}, \hat{\underline{x}}, \underline{r}$ )**

choose  $i$  such that  $r_i$  is maximized

$\forall j \neq i$  do:

$$r_j := r_j - m_{ji}r_i$$

$$\hat{x}_j := \hat{x}_j - m_{ji}r_i$$

$$r_i := 0$$

return  $(i, \hat{\underline{x}}, \underline{r})$

Note that  $\underline{r}^{(k)}$  is not the residual regarding  $\hat{\underline{x}}^{(k)}$  but rather the residual of  $\underline{x}^{(k)}$  from the Southwell iteration. So as initial guess  $\hat{\underline{x}}^{(0)}$  we have to choose  $\hat{\underline{x}}^{(0)} := \underline{x}^{(0)} + \underline{r}^{(0)} = \underline{b}$ . So in contrast to the Gauss-Seidel method the residual of the approximate solution is not available in this method.

The physical interpretation is in some sense exactly opposed to the one of the Gauss-Seidel method. In PR unshot radiosity from one patch  $i$  is distributed to all other patches  $j$  (*shooting*). Choosing the maximum residual (*sorting*), which is initially concentrated in the light-sources, guarantees that significant light transports are conducted early in the iteration.

PR was introduced based on physical arguments in [cohe88]<sup>1</sup>. [gort94] showed the above relation to Gauss-Seidel iteration.

---

<sup>1</sup>the original version of progressive refinement chooses an  $i$  such that  $|A_i||r_i|$  is maximized

method	reference	selection of $i$	$\bar{M}$	
PR	[cohe88]	$D_{A\mathcal{L}}$	$I$	$= I$
Shao	[shao93]	$D_{A\mathcal{L}}$	$I + F$	$= I + F$
Gortler	[gort94]	$D_{A\mathcal{L}}$	$\sum_j \tilde{F}_i^j$	$\approx I + F$
Feda	[feda92] [xu94]	$(I - \hat{F})D_{A\mathcal{L}}$	$\sum_i \hat{F}^i$	$= I + \frac{1}{1-\rho_{avg}} \hat{F}$
Xu	[xu94]	$(I - \hat{F})D_{A\mathcal{L}}$	$I + F + \sum_{i \geq 2} \hat{F}^i$	$= I + F + \frac{\rho_{avg}}{1-\rho_{avg}} \hat{F}$

Table 2.1: Overrelaxation methods. The index  $i$  is selected such that the absolute value of the  $i$ -th component of the given matrix is maximized.

### Overrelaxation Methods

A more general form of iterative methods is given by Alg. 2.3.3.

**Algorithm 2.3.3** `Update( $M, \underline{b}, i, \underline{x}, \underline{r}$ )`

*Choose next  $i$*

*Choose  $\Delta$*

$x_i := x_i + \Delta$

$\forall j \text{ do :}$

$r_j := r_j - m_{ji}\Delta$

**return**  $(i, \underline{x}, \underline{r})$

Special cases of this algorithm are the Gauss-Seidel method and progressive refinement, with  $\Delta = r_i$ . Choosing  $\Delta = \omega r_i$ ,  $0 < \omega < 2$  yields the SOR method (i.e. [stoe80]). The SOR method can be shown to converge for diagonally dominant systems.

Ideally we would like to choose<sup>2</sup>  $\Delta = e_i M^{-1} \underline{r}$ . Since  $M^{-1}$  is not known it has to be approximated. Several such approximations  $\bar{M}$  have been suggested (see Table 2.1). They are based on approximation of  $M^{-1}$  by the leading components of the Neumann series.

---

<sup>2</sup> $e_i$  is the  $i$ -th vector of unity

The quantities in Table 2.1 are defined as follows:  $\rho_{avg} = \sum_i \rho_i/n$ ,  $\hat{F} = \frac{1}{|S|} R \mathbf{1} D_A (\mathbf{1}$  is a matrix with all entries equal to 1) and  $\tilde{F}_i = (\tilde{f}_{ikl})_{n \times n}$ :

$$\tilde{f}_{ikl} = \begin{cases} \rho_k f_{kl} & : k \neq l \text{ and } (k = i \text{ or } l = i) \\ 0 & : \text{otherwise} \end{cases}$$

Note that with all the above methods  $\Delta$  can be computed in  $\mathbf{O}(n)$  operations. For one update one row and one column of the matrix  $F$  are necessary.

Physically the computation of  $\Delta$  using more components of the Neumann series corresponds to not only accounting for unshot radiosity on a patch  $A_i$  but also for the unshot radiosity reflected back to patch  $A_i$  from other patches of the environment.

[xu94] developed the mathematical framework for the methods presented above, which were originally derived based on physical arguments. For systems with piecewise constant basis functions they give convergence proofs for a modified version of the Feda method and the Xu method.

## Comparison

Let  $n$  be the number of unknown variables. Then all algorithms presented need  $\mathbf{O}(n)$  time to compute one update. Southwell iteration, progressive refinement and the presented overrelaxation methods are more expensive per update since we have to find the maximum among all residual values. However, in the context of the radiosity equation the cost of computing the relaxation updates is in fact negligible compared to the cost of computing the entries of  $M$ . In the methods presented above the entries of  $M$  can be computed and stored while they are needed for an update. One hopes that such a procedure allows computation of rough approximate solutions without having to compute all entries of the matrix  $M$ . Furthermore, we will never have to store more than individual columns and rows of  $M$  reducing the memory

complexity to  $\mathbf{O}(n)$ . This memory saving procedure, however, comes at the cost of having to recompute certain columns and rows.

We assume the case of piecewise constant basis functions with properties outlined in subsection 2.3.1. For the light-sources we have  $r_i^{(0)} = b_i \geq 0, \forall i$ . Since  $m_{ij} \leq 0$  for  $i \neq j$  we see that  $r_i^{(k)} \geq 0$  in Gauss-Seidel and progressive refinement updates (the same was shown for the Gortler and the Shao overrelaxation methods). Thus, the iterates approach the solution monotonically, i.e. just by positive steps. Therefore choosing the  $i$  with the greatest  $r_i$ , like in progressive refinement, yields the fastest error reduction during the initial updates, as long as no variable is updated a second time.

Based on physical considerations [cohe88] compute an estimate for the total amount of light that still has to be distributed. This amount, called *ambient term*, is distributed over all patches weighted by area, resulting in a contribution  $\underline{x}_{\text{ambient}}$ . The ambient term is not part of the iteration. It is added to the current approximate solution before it is output. This results in a progressive development of the output solution as required in Section 2.1. The ambient term was motivation for the development of the Feda overrelaxation method.

The figures in [gort94] show that Southwell iteration initially has an advantage over Gauss-Seidel. The error obtained after  $n$  updates of the Southwell iteration is less than a third of the error obtained after  $n$  updates of Gauss-Seidel iteration. In the experiments presented by [gort94] a visual convergence is obtained after less than  $6n$  Gauss-Seidel updates. Cohen et. al. [cohe88] obtain visual convergence with progressive refinement and an ambient term after  $\frac{1}{5}n$  updates.

Xu et. al. [xu94] compare progressive refinement and overrelaxation methods. They obtain a visually good result with the Xu method after  $\frac{1}{4}n$  updates. Convergence of high accuracy of the Xu method is obtained after about  $2n$  updates, which is  $\frac{1}{3}$  of the number of updates needed by the PR method to obtain the same accuracy. The other overrelaxation

methods perform inbetween these two extremes.

### 2.3.3 Other Methods

#### Krylov Subspace Methods

Krylov subspace methods often converge faster than the Gauss-Seidel method. A popular member of that family is the conjugate gradient (CG) method ([hest52], [saad95], [shew94]).

It converges if the system  $M$  is SPD. A simple modification, CGNR, applies the CG method to the SPD system  $MM^T$  and solves for  $\underline{y}$ . The solution of the original system can then be computed as  $\underline{x} = M^T \underline{y}$ . CGNR converges slower than the CG method but it does not require the matrix  $M$  to be SPD. Another method which does not require an SPD matrix is GMRES ([saad95]). Generally, the convergence of any Krylov method cannot be better than the convergence of CG if the system is SPD. It requires to store the entire sequence of computed iterates. However, since iterative methods for radiosity problems converge usually in less than 10 iterations this is no significant drawback here. While the methods based on Gauss-Seidel iteration require information of individual entries of the matrix the only information required for CG and GMRES is how to compute a matrix-vector multiplication. CGNR additionally requires the ability to compute a multiplication with the transposed matrix.

[bara95], [niev97] apply the CG method to a discretization of the radiosity problem with piecewise constant basis functions. They solve the SPD system:

$$D_A R^{-1} M \underline{x} = D_A R^{-1} b$$

Nievergelt [niev97] reports convergence of the conjugate gradient method in 10 times fewer iterations than Gauss-Seidel. [bara95] report slower convergence for scenes with low average reflectivity ( $\rho_{avg} = 0.24$ ). [will97a] apply the CG method to the non-symmetric system  $M$ .

They observe that the CG method still converges. They compare CPU-time and find that the CG method works as fast as Gauss-Seidel for scenes with  $\rho_{avg} > 0.5$ , however slower for scenes with smaller reflectivity.

### Multilevel-like Methods

A method which works with a sequence of approximations  $F^{(\epsilon)}(\underline{x})$  to  $F$  is introduced in [hanr91] and further developed by [stol96]:

#### **Algorithm 2.3.4**

*while not converged*

*compute  $F^{\epsilon(k)}(\underline{x}^{(k)})$  by refinement of  $F^{\epsilon(k-1)}(\underline{x}^{(k-1)})$*

*$\underline{x}^{(k+1)}$  := solution of  $(I - F^{\epsilon(k)}(\underline{x}^{(k)}))\underline{x}^{(k+1)} = b$*

*$k := k + 1$*

We require  $F^{(\epsilon)}(\underline{x})\underline{x}$  to converge to  $F\underline{x}$  with decreasing  $\epsilon > 0$ . The computation of a matrix  $F^{(\epsilon)}(\underline{x})$  which satisfies this requirement can be less expensive than the computation of an approximation to  $F$  itself. This method might need more iterations to converge, however, these additional costs are negligible compared to the potential savings of having to compute a smaller number of coefficients in total.

### 2.3.4 Comparison

In the following paragraph we will compare the solution methods in the context of the discretization methods from section 2.2.

Gauss-Seidel-like methods are suitable for standard Galerkin discretizations. To be efficient they require access to single rows and columns of the system to discretize. Krylov methods on the other hand are not well suited for this type of discretization since each

iteration involves a full matrix vector multiplication. This would require to compute and store all  $\mathbf{O}(n^2)$  entries of the matrix before starting the solution process. Such a procedure is only feasible for small systems and contradicts the requirement of being able to produce intermediate results during computation of the solution. Additionally reports on the behavior of the CG method do not indicate an advantage over Gauss-Seidel-like methods. We do not know about results obtained with other Krylov methods.

Looking at the discretization obtained by the hierarchical methods we come to exactly the opposite conclusion. Here we do not have access to single elements of the matrix. The only Gauss-Seidel-like method applicable seems to be Picard iteration. As well all the Krylov methods are applicable. Storing the complete system with  $\mathbf{O}(n)$  entries is inevitable in hierarchical methods, so this is no particular drawback of Krylov methods.

The multilevel-like method is applicable to the hierarchical discretization since the computation of  $F^{(\epsilon)}$  is a very natural process here, which can be realized by successive refinement of the hierarchy computed so far.

## 2.4 Computing the Discrete System

The discussion so far has dealt with ways to discretize the radiosity equation and solving the discretized system. As shown in section 2.2 computing the system itself is generally the most costly part of the entire solution process. The entries of the linear system of equation are based on the quadruple integrals of the form factors. Due to speed considerations numerical integration will always be restricted to a small number of samples per dimension, typically  $n_{samp} \leq 4$ . Considerable effort has been spent in accelerating the computation of form factors. We give a brief overview of methods used in computer graphics. A more in depth discussion can be found in [cohe93b]. We leave the discussion of the handling of singularities

in the integrand to [zatz93] and [schr93].

### 2.4.1 Form Factor Approximation

#### Piecewise Constant Basis Functions

We discuss the most commonly used techniques here. Other suggested methods are a transformation of the surface integral in a contour integral (*contour integral method*), analytic computation for unoccluded scenes by approximation with standard geometries, and Monte-Carlo integration.

For piecewise constant basis functions we have to compute the integral:

$$\begin{aligned} g_{kl} &= \frac{1}{|A_k|} \iint_{A_k} \iint_{A_l} v_S(\underline{p}, \underline{q}) \frac{\cos \angle(n_p, \underline{q} - \underline{p}) \cos \angle(n_q, \underline{p} - \underline{q})}{\pi \| \underline{p} - \underline{q} \|_2^2} dA_l(\underline{q}) dA_k(\underline{p}) \\ &= \frac{1}{|A_k|} \iint_{A_k} \iint_{\Omega} v_S(\underline{p}, \underline{q}) \frac{\cos \angle(n_x, \underline{q} - \underline{p})}{\pi} d\omega(\underline{q}) dA_k(\underline{p}) \end{aligned}$$

If we choose  $\underline{p}$  to be the center of patch  $A_k$  then a simple approximation for  $g_{kl}$  is:

$$\tilde{g}_{kl} = \iint_{\Omega} v_S(\underline{p}, \underline{q}) \frac{\cos \angle(n_x, \underline{q} - \underline{p})}{\pi} d\omega(\underline{q}) \quad (2.18)$$

If patches  $A_k$  and  $A_l$  are small, far apart and there are no occluders close to  $A_k$  then this approximation is justified. Still, one double integral involving the visibility function  $v_S$  has to be computed. This can be done by simple numerical integration or by one of the methods described below.

#### Hemicube

This technique subdivides  $\Omega$  into a fixed number  $n_{hemi}$  of subsets  $\Omega_i$   $0 \leq i < n_{hemi}$ . Assuming that the part of patch  $A_l$  subtended by  $\Omega_i$  is completely visible or invisible from  $\underline{p}$  then the integral of equation (2.18) is independent of the location of  $A_l$  itself. If the  $\Omega_i$  are sufficiently

small then this assumption approximately holds. The method computes a complete row of form factors  $f_{kl}$ ,  $0 \leq l < n$ . With each  $\Omega_i$  we store a coefficient  $\tilde{g}_{hemi,i}$ , an index  $k_i$  of a patch and a distance  $d_i$ . Initially  $d_i$  is set to  $\infty$ . The algorithm is shown below:

#### **Algorithm 2.4.1**

```

for all patches  $A_l$ :
  for all  $\Omega_i$  which intersect the solid angle generated by  $A_l$ :
    if  $distance(\underline{p}, A_l) < d_i$ 
       $d_i := distance(\underline{p}, A_l)$ 
       $k_i := l$ 

```

$$\tilde{g}_{hemi,i} := \iint_{\Omega_i} \frac{\cos \angle(n_x, \underline{q} - \underline{p})}{\pi} d\omega(\underline{q})$$

$\tilde{g}_{hemi,i}$  is computed based on just one sample of the kernel.

```
for all  $l$ :
```

$$g_{kl} := 0$$

```
for all  $0 \leq i < n_{hemi}$ :
```

$$g_{kk_i} := g_{kk_i} + \tilde{g}_{hemi,i}$$

This method has obvious disadvantages. Far away or small patches might not be properly represented by the discretization of  $\Omega$  and result in aliasing in the obtained solution. On the other hand the method can use existing Z-buffer hardware. The method was introduced by [cohe85]. The name *hemicube* is based on the fact that  $\Omega_i$  are chosen such that they correspond to rectangles on the surface of a cube with center in  $\underline{p}$ .

## Higher-Order basis functions and Hierarchical Methods

There are two major differences between the quadrature for piecewise constant basis functions and higher-order basis functions and hierarchical methods. First, the patches found in higher-order methods and on coarse levels of hierarchical methods are potentially bigger than those found with piecewise constant basis functions. Secondly, simple approximations assuming constant radiosity over a patch would clearly defeat the purpose of higher-order methods.

Willmott and Heckbert [will97b] use *visibility-in-quadrature* and *fractional visibility* to compute the form factors. Visibility-in-quadrature approximates the form factor integral by numerical quadrature. Various quadrature rules are discussed in [gers95]. These methods are expensive since each sample requires evaluation of the visibility function  $v_S$ . Fractional visibility is the method introduced in [hanr91]. Both patches  $A_k$  and  $A_l$  involved in the computation of the form factor are subdivided into a  $4 \times 4$  grid. Next, for each grid point of patch  $A_k$  the visibility function with only one grid point of  $A_l$  is computed. [hanr91] use a fixed scheme to allocate grid points on  $A_k$  to a grid point on  $A_l$ . From these visibility samples we obtain an estimate  $\alpha$  of the fraction of  $A_k$  that is visible from  $A_l$ . The form factor is then computed without visibility function (*unoccluded formfactor*) and afterwards multiplied by  $\alpha$ . Fractional visibility reduces the number of samples required from  $n_{\text{samp}}^4$  to  $n_{\text{samp}}^2$ .

In hierarchical methods we can compute form factors on coarse levels accurately by using the dilation equation and base the computation on the form factors computed for the next finer level as given by equation (2.15).

### 2.4.2 Form Factor Estimation

To be able to realize the top-down approach taken by hierarchical methods we need a function that can estimate the magnitude or an upper bound of a form factor involving wavelets without having to expensively compute it. Their computation on coarse levels is prohibitively expensive since they would require a large number of kernel samples to be able to represent the kernel accurately.

For piecewise constant basis functions [hanr91] give an upper bound for the unoccluded form factor (this corresponds to the form factor involving only scaling functions). Visibility is incorporated using fractional visibility yielding an estimate  $\hat{f}_{kl}$  for  $\langle T\phi_{j,l}, \phi_{j,k} \rangle$ . To estimate the value of form factors involving the Haar wavelets,  $\hat{f}_{kl}$  is multiplied by a parameter  $\xi > 1$  if  $A_{j,k}$  is only partially visible from  $A_{j,l}$ .

In the general case of higher-order basis functions there is no method known to bound the unoccluded form factor. Therefore we have to take a different approach. Let the number of vanishing moments of the considered wavelets be  $\nu$ . Then we know that a form factor is small if the kernel behaves similar to a polynomial of degree  $\nu - 1$ . [gort93b] suggest to estimate the form factor by estimating the deviation of the kernel from a polynomial of degree  $\nu - 1$ . The kernel is sampled and an interpolating polynomial is constructed based on these samples. The deviation is computed by comparing the polynomial to samples computed inbetween the previously computed samples. This gives a qualitative measure for the magnitude of the form factor.

### 2.4.3 Comparison

Form factor computation and estimation for higher-order basis functions are more expensive than for piecewise constant basis functions. The cheaper method of fractional visibility

produces inaccurate results in this case. Quadrature formulae need higher accuracy, consequently a higher number of visibility samples. Furthermore, hardware-supported methods like the hemicube method are not available here.

## 2.5 Summary

The method most commonly used in computer graphics is *progressive radiosity*. This method combines a discretization with piecewise constant basis functions, substructuring and progressive refinement as the solution method. Form factors are typically computed using the hemicube approach. The hemicube method can compute complete rows or columns of form factors as needed by progressive refinement. Numerous ways of parallel implementations on different parallel architectures have been presented ([reck90], [chen89], [bric95], [baum90], [guit91], [garc97]). These methods exploit that form factors can be computed independently. Progressive radiosity requires only  $\mathbf{O}(n)$  memory ( $n$  is the number of patches), but requires  $\mathbf{O}(n^2)$  execution time. The hierarchical method with piecewise constant basis functions is discussed frequently in the literature. It requires  $\mathbf{O}(n)$  memory and  $\mathbf{O}(n)$  execution time. However, the constant factors in these time and memory estimates are much bigger than those for progressive radiosity.

All considered methods try to reduce the system of equations (2.5) such that it can be computed and solved efficiently. On the one hand this is done by exploiting smoothness properties of the solution. Discontinuity meshing and higher-order Galerkin methods choose a basis which approximates the solution well. On the other hand hierarchical methods and substructuring use smoothness properties of the kernel. The ultimate goal is to find a sparse representation fully automatically without user interaction by an adaptive process. This process can be based purely on the kernel itself, or in an iterative process it can be

based on the approximate solution found so far. Hierarchical methods reduce the number of parameters to choose to the compression parameter  $\epsilon$  and the number of levels.

Efforts have been made to integrate concepts from discontinuity meshing with higher-order hierarchical methods. However, no complete analysis of such a system has been presented.

Clustering has been applied to hierarchical methods with piecewise constant basis functions. We do not know of reports on the application to hierarchical methods with higher-order basis functions.

For piecewise constant basis functions numerous provably convergent solution methods exist. These solution methods are also applied to systems based on higher-order basis functions and the compressed systems arising in hierarchical methods. However, we do not know of any careful mathematical analysis of their convergence.

# Chapter 3

## Exploring Wavelet Radiosity

In this chapter we give background and motivation for the numerical experiments presented in Chapter 4. We will be concerned with the behavior of different wavelet bases in wavelet radiosity and the analysis of iterative methods for solution of the system of equations arising in that method. Section 3.1 presents and compares different wavelet bases, Section 3.2 motivates the application of GMRES to the solution of our system. Finally Section 3.3 describes and justifies the algorithms used in our experiments.

### 3.1 Wavelet Bases for Radiosity

We introduced the notion of multiresolution analysis (MRA) with scaling functions and wavelets in Section 2.2.3.

MRA with various properties exist. We list some criteria that typically distinguish them:

- Bounded or infinite support of scaling functions and wavelets.
- Number of vanishing moments of the wavelets.

- Properties of the space spanned by the scaling functions. Normally MRA are defined on the real line. With *wavelets on the interval* we can span spaces defined on a bounded interval (e.g. [cohe93a], [alpe93]). A special type of wavelets on the interval are wavelets which satisfy boundary conditions on the boundaries of the interval ([chia97]).
- Smoothness of the spaces spanned by the scaling functions.

We will be concerned with three types of MRA: the Haar MRA, the multiwavelets, and an MRA introduced in [cohe93a] which we will denote as CDV MRA. All three types of MRA are orthonormal ( $\forall k \neq l : \langle \phi_{j,k}, \phi_{j,l} \rangle = 0, \langle \phi_{j,k}, \phi_{j,k} \rangle = 1, \forall i, j, k \neq l : \langle \psi_{i,k}, \psi_{j,l} \rangle = 0, \langle \psi_{i,k}, \psi_{i,k} \rangle = 1$ ). We do not consider *flatlets* here (flatlets [gort93b] are based on the Haar wavelet and have a higher number of vanishing moments).

In Figure 3.1 we show the hierarchical structure of the three types of MRA. This allows us to use a simplified notation for indexing in the following exposition. Furthermore here we stay within a one-dimensional MRA. We show how to extend this framework to two-dimensional MRA in Appendix A.1.

### 3.1.1 Haar Wavelets

The Haar scaling function is defined as the solution  $\phi$  of the following equation with  $h_0 = h_1 = 1/\sqrt{2}$ :

$$\phi(x) = \sqrt{2} \sum_{l=0,1} h_l \phi(2x - l) \quad (3.1)$$

The Haar wavelet is defined based on orthogonality considerations with  $g_0 = 1/\sqrt{2}, g_1 = -g_0$ :

$$\psi(x) = \sqrt{2} \sum_{l=0,1} g_l \phi(2x - l) \quad (3.2)$$

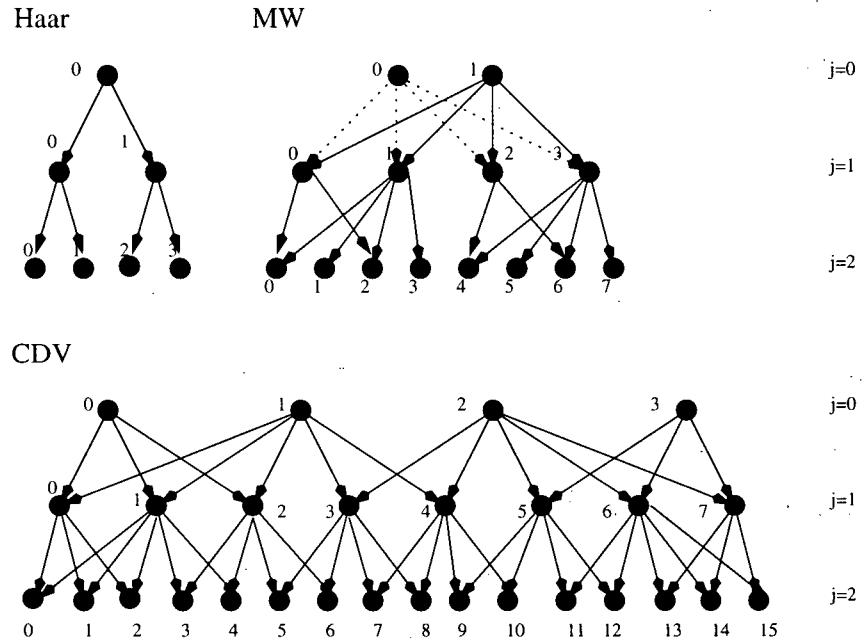


Figure 3.1: Dependencies between different levels of the MRA. The nodes indicate scaling functions and wavelets.

We define the scaling function and wavelet on the coarsest level  $\phi_{0,0}(x) = \phi(x)$ ,  $\psi_{0,0} = \psi(x)$ . The hierarchy of basis functions is then obtained by the dilation equation, equation (2.10), where we define  $h_{k,2k} := h_0$ ,  $h_{k,2k+1} := h_1$ ,  $g_{k,2k} := g_0$  and  $g_{k,2k+1} := g_1$  (corresponding to Figure 3.1).

Functions constant on the entire domain can be represented exactly in the Haar basis. The Haar wavelets have one vanishing moment. We show the graph of the Haar scaling function and wavelet in Figure 3.2.

### 3.1.2 Multiwavelets

The Multiwavelet MRA was presented by [alpe93]. It is based on Legendre polynomials. Here we are concerned with multiwavelets based on linear functions. We can give explicit expressions for these functions (Figure 3.3). The hierarchy is constructed analogously to

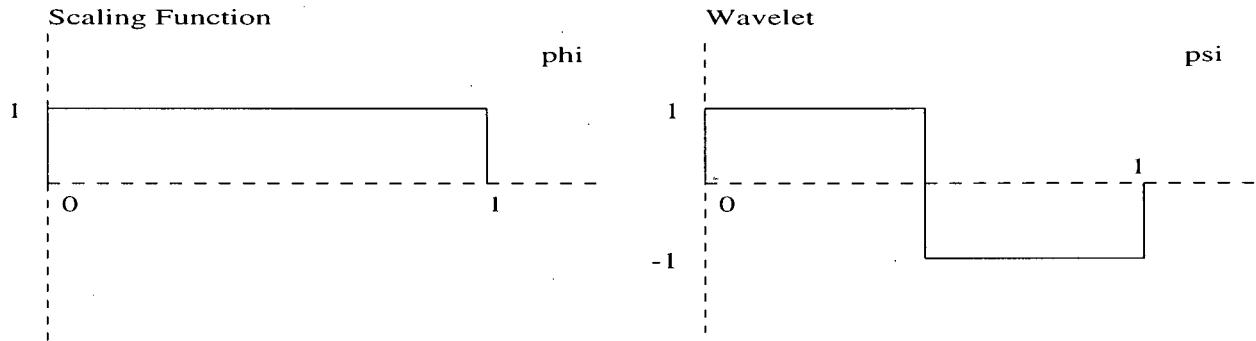


Figure 3.2: Haar scaling function and wavelet.

the case of the Haar wavelets using the dilation equation. The wavelets have two vanishing moments. They are able to reproduce linear functions exactly. The space spanned by multiwavelet scaling functions contains discontinuous functions.

The coefficients for the generation of the hierarchy are listed in the following table:

MW Scaling Functions			MW Wavelets		
$k$	$l$	$h_{k,l}$	$k$	$l$	$g_{k,l}$
0			0		
	0	$1/\sqrt{2}$		1	$-1/\sqrt{2}$
	2	$1/\sqrt{2}$		3	$1/\sqrt{2}$
1			1		
	0	$-\sqrt{3}/\sqrt{8}$		0	$1/\sqrt{8}$
	1	$1/\sqrt{8}$		1	$\sqrt{3}/\sqrt{8}$
	2	$3/\sqrt{8}$		2	$-1/\sqrt{8}$
	3	$1/\sqrt{8}$		3	$\sqrt{3}/\sqrt{8}$

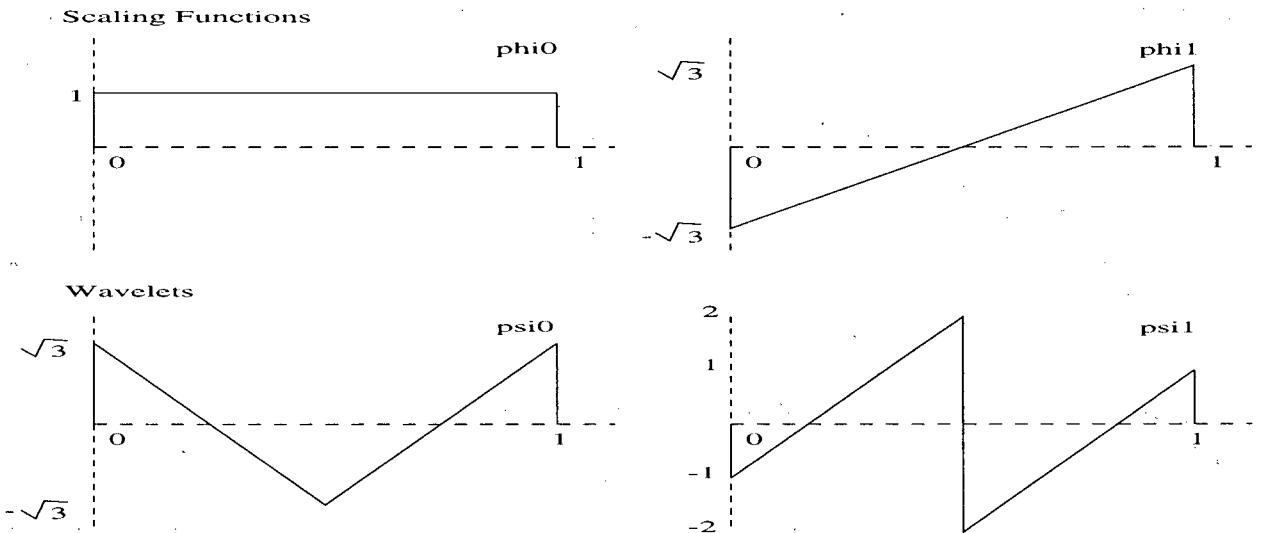


Figure 3.3: Multiwavelets. Scaling functions and wavelets.

### 3.1.3 CDV Wavelets

The CDV MRA is presented in [cohe93a]. This MRA is derived from the Daubechies MRA [daub88] in order to make them span spaces with functions defined on a finite interval (the Daubechies wavelets are defined on  $\mathbb{R}$ ). Analogous to the Daubechies wavelets, CDV wavelets are available with different degrees of smoothness and different widths of support. Here we are solely concerned with the MRA derived from the Daubechies  $D_4$  wavelet.

The CDV scaling functions and wavelets in the interior of the interval, i.e. the functions with indices  $k : 2 \leq k < n_m - 2$ , are identical to the  $D_4$  scaling function and wavelet. On the boundaries,  $D_4$  wavelets and scaling functions are modified, resulting in special edge scaling functions  $\phi_0, \phi_1, \phi_{n_m-2}, \phi_{n_m-1}$  and wavelets  $\psi_0, \psi_1, \psi_{n_m-2}, \psi_{n_m-1}$ .

As in the case of Haar wavelets and multiwavelets, these coefficients also serve to generate the hierarchy using the dilation equation. The wavelets presented here have two vanishing moments. They can represent linear functions exactly. Functions in the spaces spanned by the wavelets are continuous but not differentiable. We show graphs of the CDV

scaling functions and wavelets in Figure 3.4.

The  $D_4$  scaling function and wavelet are defined by a recurrence relation analogous to equation (3.1) and equation (3.2) with the following coefficients:  $h_0 = (1 + \sqrt{3})/(4\sqrt{2})$ ,  $h_1 = (3 + \sqrt{3})/(4\sqrt{2})$ ,  $h_2 = (3 - \sqrt{3})/(4\sqrt{2})$ ,  $h_3 = (1 - \sqrt{3})/(4\sqrt{2})$ ,  $g_0 = -h_3$ ,  $g_1 = h_2$ ,  $g_2 = -h_1$ ,  $g_3 = h_0$ .

For the functions at the boundaries we have the coefficients:

CDV Scaling Functions

$k$	$l$	$h_{k,l}$	$k - n_m - 1$	$l - n_m - 1$	$h_{k,l}$
0			-0		
0	0	$6.033325e - 01$		-0	$8.705088e - 01$
1	1	$6.908955e - 01$		-1	$4.348970e - 01$
2	2	$-3.983130e - 01$		-2	$2.303890e - 01$
1			-1		
0	0	$3.751746e - 02$		-0	$-1.942334e - 01$
1	1	$4.573277e - 01$		-1	$1.901514e - 01$
2	2	$8.500881e - 01$		-2	$3.749553e - 01$
3	3	$2.238204e - 01$		-3	$7.675567e - 01$
4	4	$-1.292227e - 01$		-4	$4.431490e - 01$

### CDV Wavelets

$k$	$l$	$g_{k,l}$	$k - n_s - 1$	$l - n_m - 1$	$g_{k,l}$
0			-1		
0	0	$-7.965435169e - 01$		-0	$-2.575129195e - 01$
1	1	$5.463927140e - 01$		-1	$8.014229620e - 01$
2	2	$-2.587922483e - 01$		-2	$-5.398225007e - 01$
1			-0		
0	0	$1.003722456e - 02$		-0	$3.717189665e - 01$
1	1	$1.223510431e - 01$		-1	$-3.639069596e - 01$
2	2	$2.274281117e - 01$		-2	$-7.175799994d - 01$
3	3	$-8.366029212e - 01$		-3	$4.010695194e - 01$
4	4	$4.830129218e - 01$		-4	$2.315575950d - 01$

#### 3.1.4 Comparison

Multiwavelets and CDV wavelets have higher order and more vanishing moments than Haar wavelets. They therefore potentially lead to a better compression and better approximation than Haar wavelets. Multiwavelets can result in discontinuous approximations. CDV wavelets have the advantage of a higher degree of smoothness, which might result in better visual results, since the eye is particularly sensitive to discontinuities.

For a qualitative comparison we show in Figure 3.5 how a function  $f : [0, 1] \rightarrow \mathbb{R}$  is approximated in the spaces spanned by the scaling functions. The function  $f$  is partially smooth but also has a discontinuity (we compare *Haar*, *MW*, and *CDV* bases):

$$f(x) = \begin{cases} \tan(\pi x) & 3x < 2 \\ 0 & \text{otherwise} \end{cases} \quad (3.3)$$

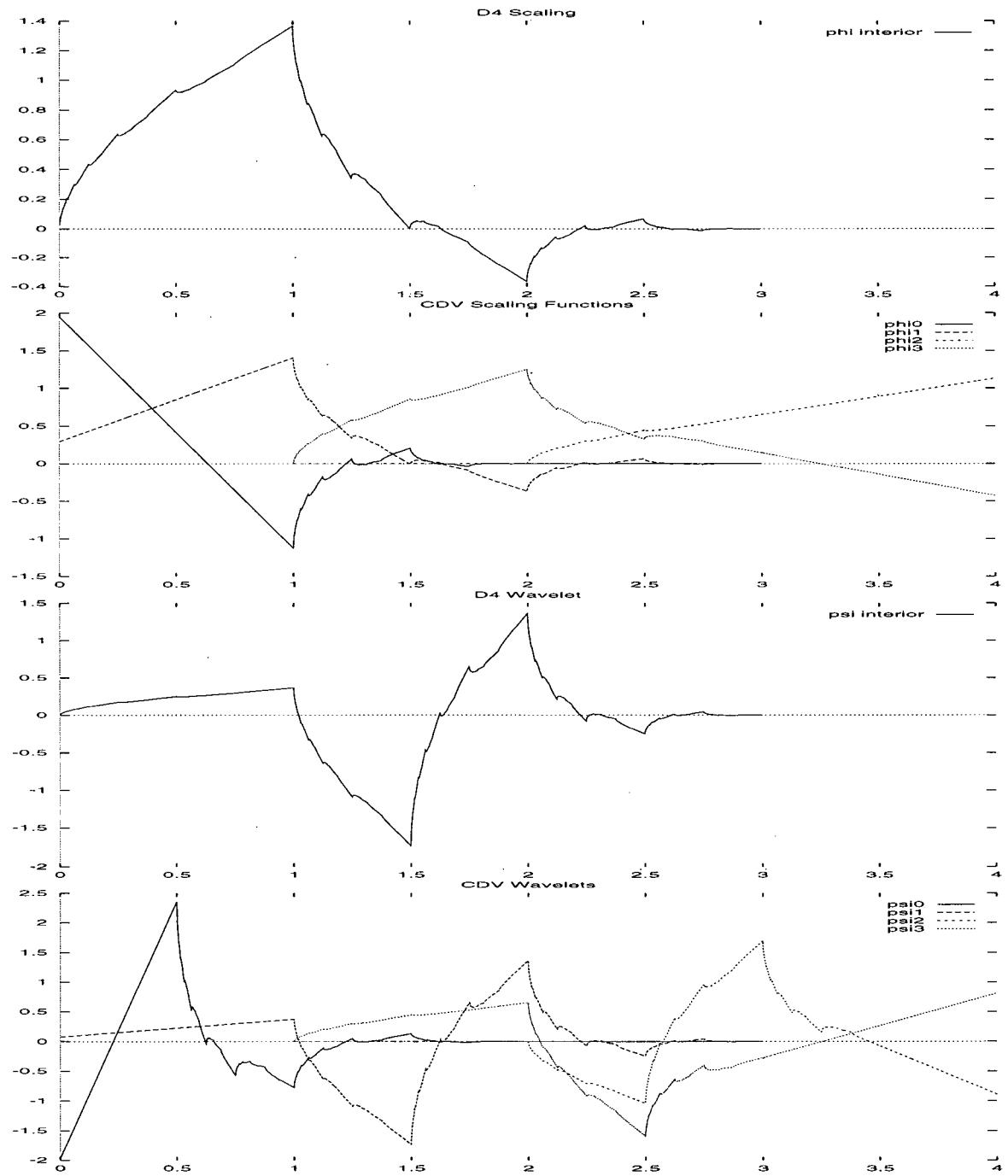


Figure 3.4: CDV scaling functions and wavelets. The  $D_4$  scaling function and wavelet form the CDV scaling functions and wavelets for the interior of the interval.

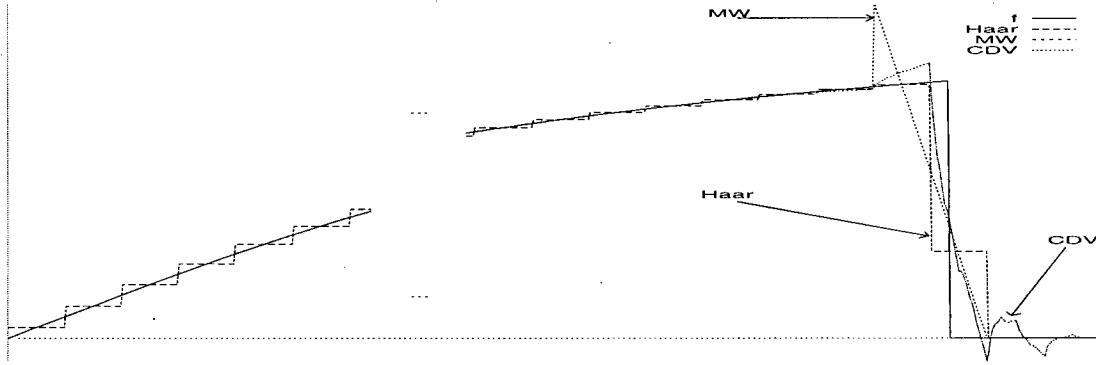


Figure 3.5: Projection of the function  $f$  of (3.3) into different spaces. All bases are with  $n = 32$  functions. Haar basis on level  $j = 5$ , multiwavelet basis on level  $j = 4$  and CDV basis on level  $j = 3$ . In the smooth part of  $f$  the approximation by higher-order bases is indistinguishable from the graph of  $f$  itself.

In Figure 3.6 we show results for an equivalent experiment with a function:

$$f(x, y) = \begin{cases} \sqrt{xy} & 3(x + y) \leq 4 \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

Here is the table for the compression rate of the two-dimensional case. We show the proportion of coefficients of wavelets that were below a given threshold  $\epsilon$ :

$\epsilon$	Haar	$MW_1$	$CDV_4$
$10^{-1}$	100%	100%	99%
$10^{-2}$	96%	96%	88%
$10^{-3}$	73%	78%	71%
$10^{-4}$	38%	68%	52%
$10^{-5}$	24%	56%	33%
$10^{-6}$	16%	42%	25%
$\vdots$			$\vdots$
$10^{-12}$	16%	13%	9%

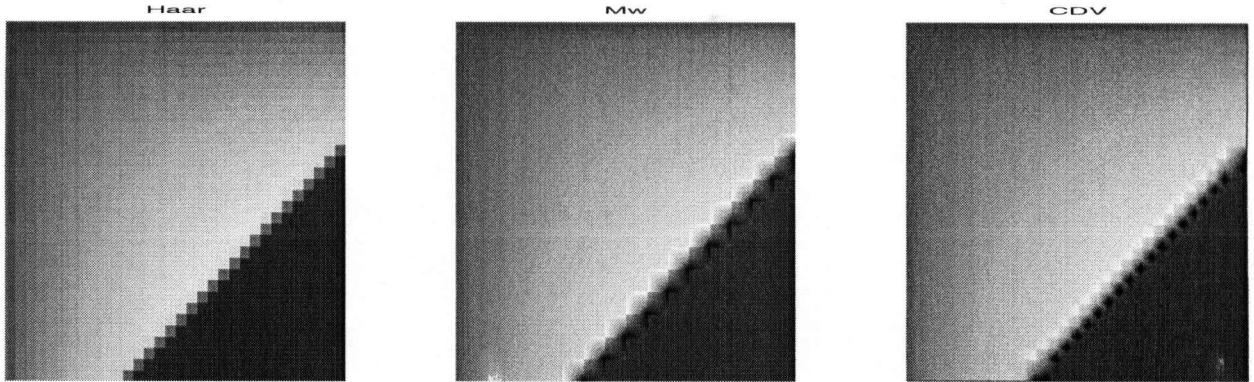


Figure 3.6: Projection of the function  $f$  of (3.4) into different spaces. All are bases with  $n = 1024$  functions. Haar basis on level  $j = 5$ , multiwavelet basis on level  $j = 4$ , and CDV basis on level  $j = 3$ .

We also compare the relative  $L_1$  error obtained for the function  $f$  defined by (3.4) to the relative  $L_1$  error obtained for the function  $g(x, y) = \sqrt{xy}$ :

	Haar	MW	CDV
$f$ (discontinuous)	$4.42e - 2$	$3.04e - 2$	$3.00e - 2$
$g$ (smooth)	$1.72e - 2$	$1.87e - 3$	$1.31e - 3$

For the discontinuous function the error in the approximation is very similar for all bases. The higher order bases have a small advantage since they approximate the smooth part of the function better. Both higher order bases produce wiggles along the discontinuity. The wiggles produced by *CDV* are less severe than those produced by *MW*. For the smooth function both higher-order bases show a clear advantage compared to the Haar basis.

The results for the compression rate are also not unexpected. Due to the additional vanishing moment multiwavelets compress the smooth part of the function better. CDV wavelets, on the other hand suffer from the fact that more basis functions overlap the discontinuity. This results in larger coefficients of such basis functions.

In Chapter 4 we will experimentally analyze how these three types of wavelet bases

perform in hierarchical methods for the solution of the radiosity equation, where the process involves solving an integral equation, not just function approximation.

## 3.2 Solution Methods for Wavelet Radiosity

As discussed in Section 2.3.3 Krylov subspace methods technically present an alternative to the Picard iteration for Wavelet Radiosity. In many applications they perform better than Picard's. The matrix system in hierarchical methods will generally not be SPD. In particular, for higher-order basis functions we do not have a theorem which would allow us to make the system SPD. Therefore we choose the methods CGNR and GMRES for our analysis. We will analyze the performance of these methods for the case of wavelet radiosity. In our implementation of the methods we follow [saad95], where we replace the matrix vector multiplication  $M\underline{x}$  by  $D_{N,m}^- M_{N_\epsilon} D_{N,m} \underline{x}$  and the matrix vector multiplication  $M^T \underline{x}$  by  $D_{N,m}^- M_{N_\epsilon}^T D_{N,m} \underline{x}$ , as described in Section 2.2.3.

The investigation of fast solution methods for the linear algebra system in case of wavelet radiosity is of interest. On the one hand, wavelet radiosity requires keeping the entire sparse system  $M_{N_\epsilon}$  in memory unlike progressive radiosity where solution process and computation of form factors are intermixed. So the solution process is an additional computational step, the cost of which seems to be worthwhile to minimize. On the other hand, in connection with the multilevel-like methods from Section 2.3.3 a reduction of the number of iterations could also reduce the number of costly refinement steps needed.

## 3.3 Implementation Aspects

Hierarchical methods for radiosity introduce the following types of approximation error into the computation:

- discretization error caused by the Galerkin discretization.
- discretization error caused by compressing the Galerkin discretization.
- error caused by the solution method.
- error caused by the form factor approximation.
- error caused by a faulty estimation of form factors.

In our analysis we want to focus on the first three types of error. We therefore designed our algorithm and test environments to minimize the last two types of error. We do not use any of the multilevel-like methods since this would make the independent analysis of solution methods and discretization difficult. In the following we describe how we realized the various components of the algorithm for wavelet radiosity in the non-standard representation.

### 3.3.1 Galerkin Discretization

In our experiments we use Haar wavelets, multiwavelets and *CDV* wavelets. We use a separate MRA for each surface  $S_i$ . Haar wavelets have one vanishing moment and the ability to approximate functions constant on the surface exactly. Multiwavelets and CDV wavelets have both two vanishing moments and can approximate all functions linear on the surface exactly. All the wavelets and scaling functions were made orthonormal in  $L_2(S)$  by multiplication with  $\sqrt{\frac{s}{|S_i|}}$  ( $s$  is the support of  $\sum_{k=0}^{n_0-1} \phi_{0,k}$ ,  $s = 1$  for Haar and multiwavelets,  $s = 4$  for CDV wavelets).

### 3.3.2 Form Factor Estimate

The form factor is estimated as described in Section 2.4.2 for higher-order basis functions. For each of the four dimensions we compute the average of the deviation over the three

remaining dimensions. The estimate is the maximum of these four averages. To compute deviation from a constant function (for a basis with one vanishing moment) we compute the difference of the kernel values at the boundaries of the interval. For the deviation from a linear function (for bases with two vanishing moments) we interpolate by a linear function based on the kernel values at the boundaries of the interval and compute the difference of the value at the center of the interval and the kernel value at the center of the interval.

### 3.3.3 Form Factor Computation

We do not deal with the problem of singularities. We design our experiments to exclude singular form factor integrals.

Unlike [gort93b] and [will97b] in most of their experiments, we use visibility-in-quadrature for the computation of form factors. We think that in the context of higher order basis functions it is particularly important to have a good approximation to the form factors along discontinuities since we have larger patches than with piecewise constant basis function. Fractional visibility does not allow to exploit the better approximation of higher order basis functions along discontinuities since it essentially assumes constant visibility within a patch. We will support this claim experimentally in chapter 4.

We illustrate the procedure of how we approximate the form factors for the one-dimensional case  $\int f(x)w(x)dx$ , where  $f$  corresponds to the kernel,  $w$  to a basis function. Due to the irregularity of the CDV basis functions a simple rule of sampling the integrand  $f(x)g(x)$  is not applicable in our case. We therefore approximate  $f$  by a piecewise linear function, and compute the integrals involving  $w$  separately. If we approximate  $f$  by  $\tilde{f}(x) = a_i + b_i x$  if  $x : x_i \leq x < x_{i+1}$  then the integral to compute is:

$$\int \tilde{f}w(x) = \sum_{i=0}^{n_{\text{samp}}-2} a_i \int_{x_i}^{x_{i+1}} w(x)dx + b_i \int_{x_i}^{x_{i+1}} xw(x)dx$$

The following table gives values for the integrals on the right-hand-side for the Haar, multiwavelet and CDV scaling functions. We computed with equidistant samples over the support of  $\phi$  and  $n_{\text{samp}} = 3$ . The Haar and multiwavelet coefficients can be computed analytically. The CDV coefficients were computed numerically with a quadrature of high granularity based on an approximation to the CDV scaling functions computed via the *cascade algorithm* ([daub92]).

		$\int_{x_0}^{x_1} \phi(x)dx$	$\int_{x_1}^{x_2} \phi(x)dx$	$\int_{x_0}^{x_1} x\phi(x)dx$	$\int_{x_1}^{x_2} x\phi(x)dx$
Haar	$\phi$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{3}{8}$
MW	$\phi_0$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{8}$	$\frac{3}{8}$
	$\phi_1$	$-\frac{1}{4}\sqrt{3}$	$\frac{1}{4}\sqrt{3}$	$-\frac{1}{24}\sqrt{3}$	$\frac{5}{24}\sqrt{3}$
CDV	$\phi_0$	$3.574485e - 01$	$4.603835e - 03$	$-1.020265e - 01$	$6.665019e - 03$
	$\phi_1$	$1.078854e + 00$	$-7.740917e - 02$	$7.821774e - 01$	$-1.472866e - 01$
	$\phi_2$	$1.090248e + 00$	$-4.041994e - 04$	$8.495431e - 01$	$-1.586078e - 01$
	$\phi_3$	$1.384212e - 01$	$1.157058e + 00$	$1.806193e - 01$	$2.736973e + 00$
$D_4$	$\phi$	$1.077350e + 00$	$-7.734999e - 02$	$7.812263e - 01$	$-1.472518e - 01$

This table gives the integrals if  $w$  is a scaling function. If  $w$  is a wavelet then the integrals are computed via the dilation equation. In the two-dimensional case we have one scaling function and three wavelets. This results in 15 form factors to compute on levels  $j > 0$  and 16 form factors on level  $j = 0$  for each pair of patches. Following a computer graphics convention we call such a structure of 15 or 16 form factors a *link*.

### 3.3.4 Memory Representation of the Sparse Matrix

We associate a hierarchical graph corresponding to the graphs shown in Figure 3.1 with each surface. Each vertex of the graph is associated with a scaling function and three wavelets. Vertices of different hierarchical graphs can be connected by a link if the corresponding entry

in the matrix was determined to be non-zero. Note that the hierarchy of the CDV MRA is not a tree structure. Therefore we designed the structure such that it is possible to check whether a link had already been computed to prevent redundant computations during the top-down form factor computation.

### **3.3.5 Solution Method**

Unless otherwise noted we use Picard iteration with 80 iterations for the solution of the system of equations. Picard iteration is the standard method used for hierarchical methods and it converged stably in all our experiments.

# Chapter 4

## Numerical Experiments

In contrast to the approach of deriving theoretical error bounds we try to analyze the behavior of the algorithms in the framework of characteristic scenes. Such an analysis can never be exhaustive due to the heterogeneous nature of possible input in computer graphics and the associated effects. Therefore we try to choose scenes with characteristic features encountered in practice which stress potential weaknesses and strengths of the analyzed algorithms. A particular behavior could then be a motivation for a more detailed analysis.

### 4.1 Comparison of Wavelet Bases

We computed radiosity solutions for two simple test scenes. The scene “Unoccluded” is specified in Figure 4.1, the scene “Occluded” in Figure 4.2. We will measure error in the  $L_1$  norm and additionally present images of the computed solution to allow for a visual comparison.

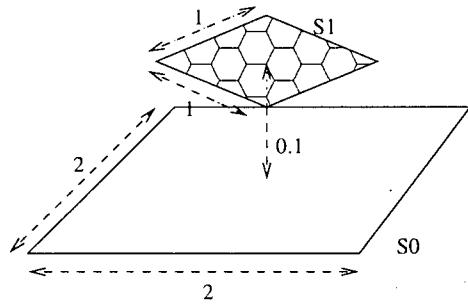


Figure 4.1: “Unoccluded”. Surface  $S_0$  has constant reflectivity  $\rho_0 = 0.4$ , no emission. Surface  $S_1$  has zero reflectivity, i.e.  $\rho_1 = 0$ , and constant emission  $e_1 = 100$ . Note the proximity of the light-source to the surface  $S_0$ , which causes a steep radiosity gradient.

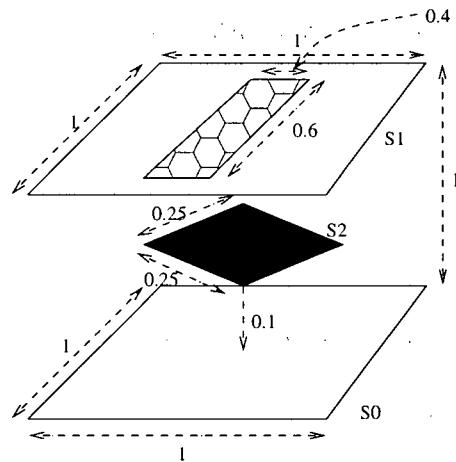


Figure 4.2: “Occluded”. Surface  $S_0$  has constant reflectivity  $\rho_0 = 0.4$ , no emission. Surface  $S_1$  has constant reflectivity  $\rho_1 = 0.4$  and constant emission  $e_1 = 100$  in the indicated area. Surface  $S_2$  has zero reflectivity and has no emission.

### 4.1.1 Test Scenes

The scene “Unoccluded” is the same scene that was used by [gort93b] for their measurements. It allows us to compute an analytic solution. The proximity of emitter and receiver results in a high but smooth radiosity gradient. The light-source is rotated to prevent alignment with the basis-functions. The radiosity on face  $S_1$  is equal to the emission  $e$ , therefore, we get the following analytic expression for the radiosity:

$$u(\underline{p}) = \rho(\underline{p}) \int_{S_1} r_2(\underline{p}, \underline{q}) e(\underline{q}) dS_1(\underline{q})$$

For our reference solution we approximated this integral by a midpoint rule with  $128 \times 128$  samples.

The scene “Occluded” has a blocker that causes a shadow on  $S_0$ . Like in the unoccluded scene we rotated the blocker to prevent alignment of the shadow boundaries with the supports of the basis-functions. We do not have an analytic solution for this scene so for quantitative results we compare with a master solution obtained with a discretization of high granularity. The master solution was computed using multiwavelets with  $\epsilon = 10^{-4}$ , with  $m = 5$  levels (resulting in  $n_5 = 4096$  basis-functions) and  $n_{samp} = 3$ .

In both scenes all the interesting events take place on the surface  $S_0$ . Therefore, in the subsequent discussion we will only display the computed images for this surface. Errors are also only computed for  $S_0$ .

### 4.1.2 Form Factor Computation

To support the claim made in subsection 3.3.3 that the method of fractional visibility is not suitable for higher order basis-functions we computed a solution with multiwavelets using both techniques. We show the results in Figure 4.3. They show that fractional visibility fails to adequately represent shadow boundaries. A similar observation was made by [will97b].

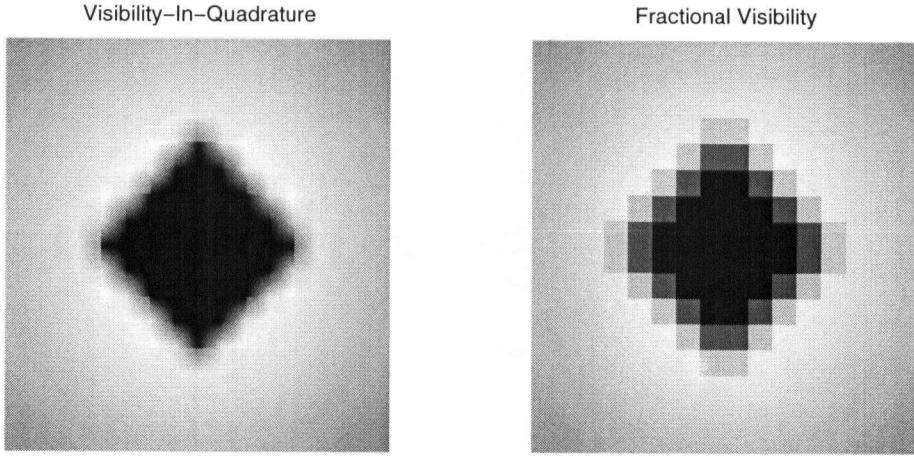


Figure 4.3: Result for the scene “Occluded” obtained using a basis of multiwavelets with  $n_4 = 1024$  functions. The left image was obtained using visibility-in-quadrature, the right image using fractional visibility.

#### 4.1.3 Measurements

We measure the cost of the methods in terms of the number of form factors computed for the solution. The number of form factors reflects memory requirements as well as computational time. The cost of computing a form factor is determined by the number of kernel samples we need. For Haar wavelets we need at least two samples in each dimension to be able to estimate if the solution is constant. For first order basis functions we need at least three samples in each dimension to estimate linearity. Over four dimensions this means that the computation of a form factor for MW and CDV is more than 5 times more expensive than the computation of a form factor for the Haar basis.

The solutions with a given number of form factors were obtained by first computing a sparse representation with a high number of form factors. Subsequently, the smallest coefficients were discarded until the desired number of form factors was reached. To be able to compare the quality of approximation we chose the number of levels such that we have the same number  $n$  of basis-functions. Furthermore we chose the following parameters for

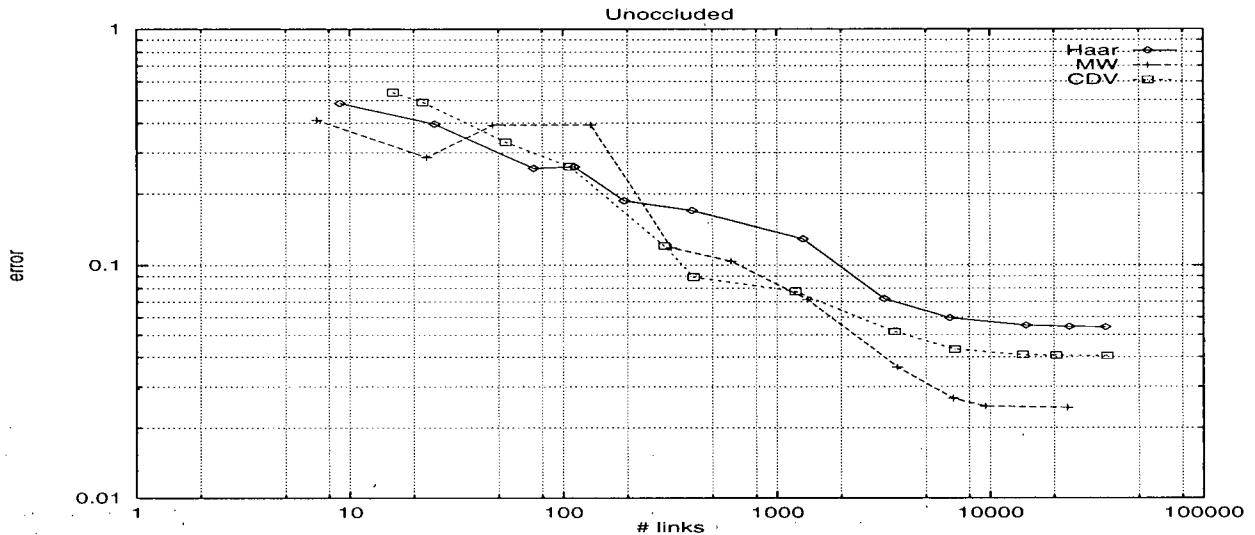


Figure 4.4: “Unoccluded”. Relative  $L_1$  error. Obtained by comparison to an analytical solution.

our computations:

	$\epsilon$	$n_{samp}$	$m$	$n$
Haar	$10^{-5}$	3	5	$n_5 = 1024$
MW <sub>1</sub>	$10^{-4}$	3	4	$n_4 = 1024$
CDV <sub>4</sub>	$10^{-4}$	3	3	$n_3 = 1024$

Following [gort93b] we calculate the relative error in the  $L_1$  norm. These errors are shown for our two scenes in Figure 4.4 and Figure 4.5. We show images of the radiosity solution computed for surface  $S_0$  in Figure 4.8 and Figure 4.9. In Figure 4.6 and Figure 4.7 we show the distribution of the error for a computation with 20000 links.

#### 4.1.4 Interpretation

We observe that in case of the unoccluded scene acceptable solutions could be computed based on 2000 links ( $< 32000$  form factors). That is far less than the number of form factors required for a standard Galerkin approximation with the same number of basis-

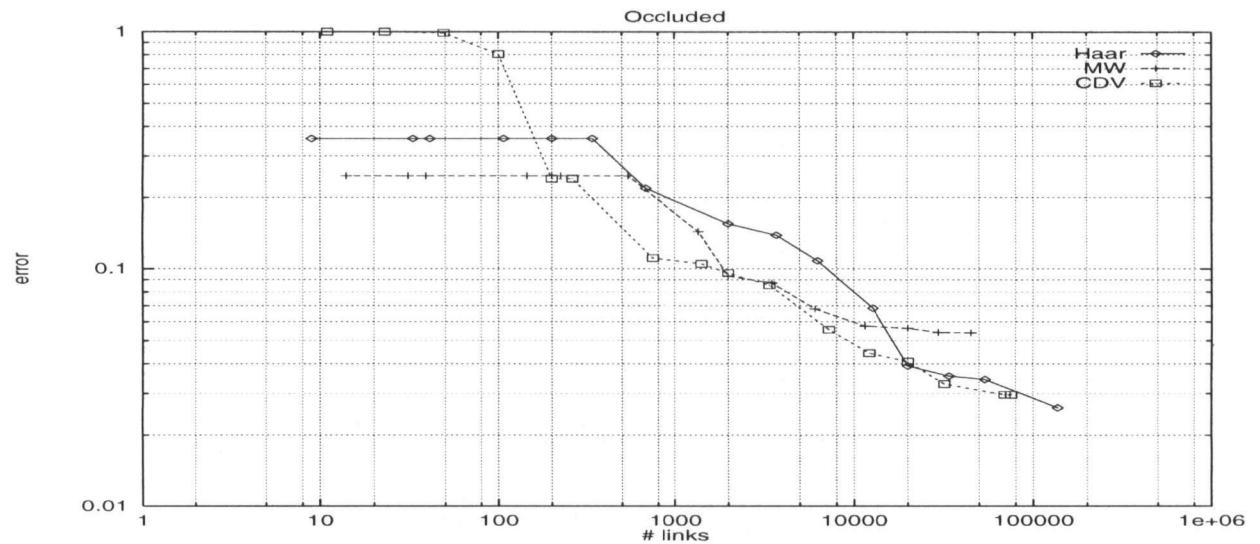


Figure 4.5: “Occluded”. Relative  $L_1$  error. Obtained by comparison to a master solution.

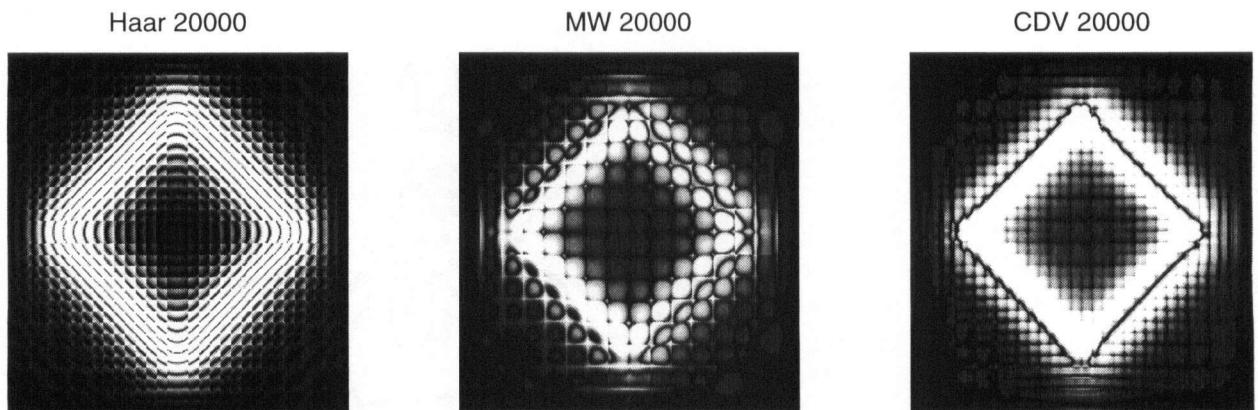


Figure 4.6: “Unoccluded”. Pointwise error for a computation with 20000 links. Obtained by comparison to an analytical solution. Black indicates small error, white indicates significant error.

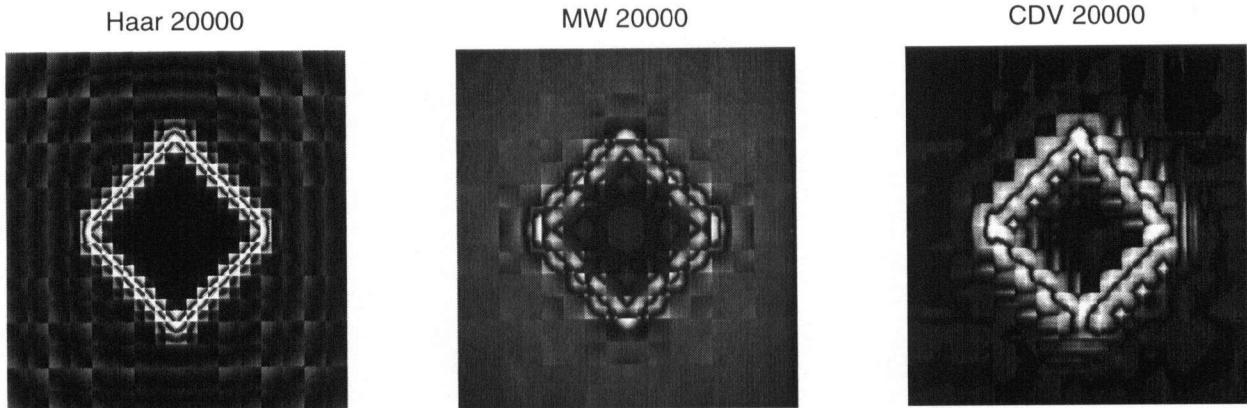


Figure 4.7: “Occluded”. Pointwise error for a computation with 20000 links. Obtained by comparison to a master solution. Black indicates small error, white indicates significant error.

functions. A full matrix method in our case would require  $1024^2 = 1\,048\,576$  form factors. This corresponds to a reduction to 3%. The compression looks worse in the occluded case. This is due to the fact that we chose the light-source for this scene not to cover the entire surface. Consequently numerous form factors were computed which do not contribute to the solution since the value of the solution is zero.

Let us first focus on the error in the  $L_1$  norm. Regarding the  $L_1$  error for the unoccluded scene multiwavelets perform best. They are followed by CDV wavelets. As expected the Haar basis shows the most significant error because the piecewise constant functions cannot approximate the smooth solution well. Multiwavelets approximate the solution well in areas with low gradient. In the area with high gradient the error is significantly higher. The same holds for CDV wavelets, however, due to the wider support the area with high error is larger. The behavior is different in the occluded scene. Here generally multiwavelets perform worst. CDV wavelets perform best, however for a higher number of links Haar is almost as good. The error for all three bases is high along the discontinuity. For Haar there is also error in the smooth areas simply because Haar is only a piecewise constant approx-

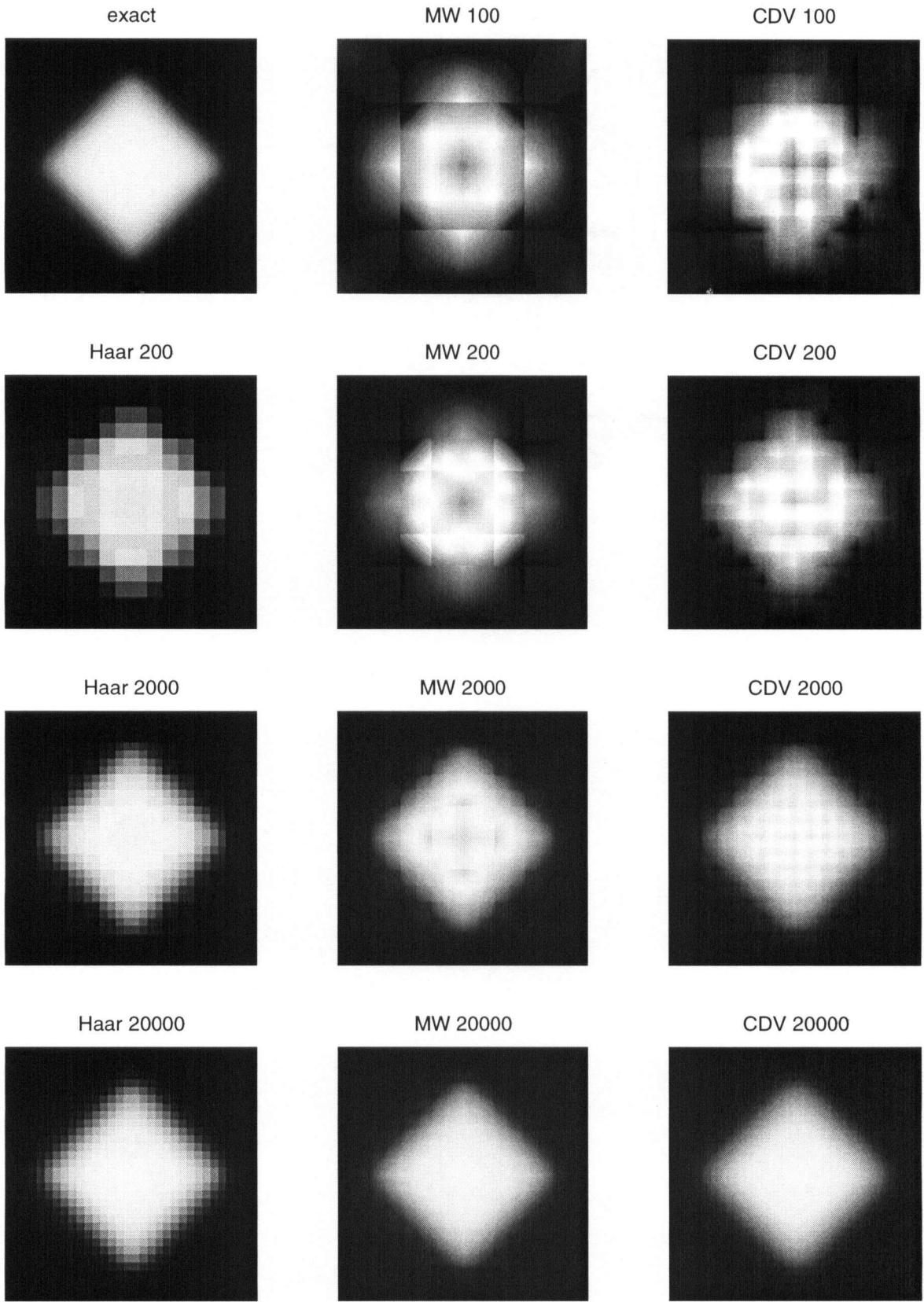


Figure 4.8: “Unoccluded”. Images computed with the basis and number of form factors as indicated. The analytic solution is shown in the upper left corner.

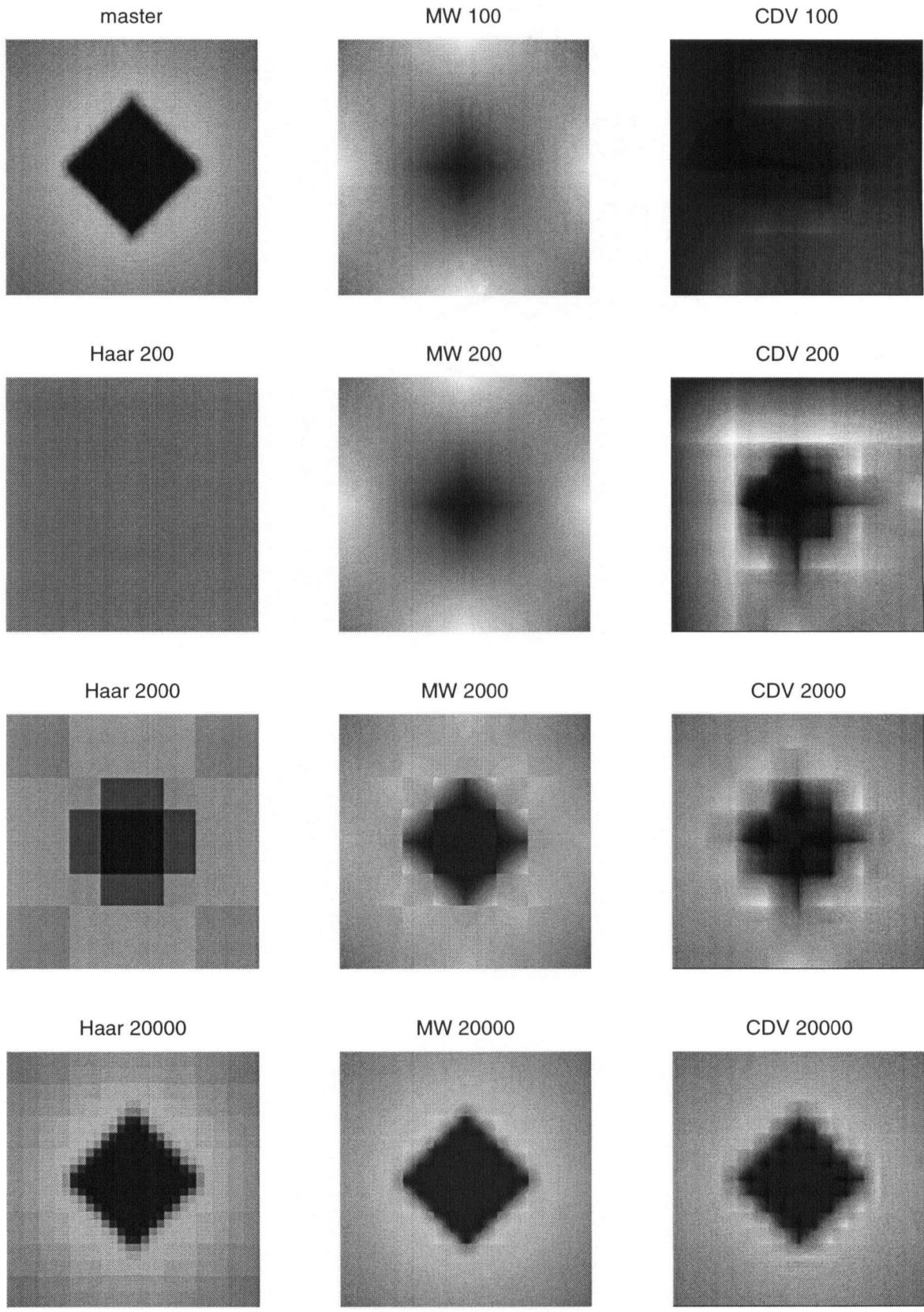


Figure 4.9: “Occluded”. Images computed with the basis and number of form factors as indicated. The master solution is shown in the upper left corner.

imation. The error for multiwavelets is also high in the smooth areas. At first glance this seems surprising since multiwavelets should be able to approximate the smooth function adequately. However, it is a known phenomenon in Galerkin discretizations when the solution cannot be approximated well. In our experiment CDV wavelets are able to cope with that problem better.

Perceptually multiwavelets and CDV wavelets perform very similarly. We leave the judgement to the reader. The CDV wavelets introduce an unexpected grid pattern for small numbers of form factors (see Figure 4.8 *CDV2000*). This grid pattern is due to the Galerkin discretization itself. It cannot be seen in the simple projection shown in Figure 3.6. We also tried a higher number of form factor samples  $n_{samp}$ , however, this did not result in any change. Both higher order bases produce clearly better results than the Haar basis without post-smoothing. Considering the wide support of the CDV wavelets their performance in our experiments for the occluded scene is surprisingly good.

For the case of multiwavelets in the occluded case we note the difference in perceptual error and measured error. Perceptually multiwavelets appear to perform better than Haar wavelets, not so however in terms of  $L_1$  error.

Generally we found it hard in our experiments to choose the appropriate parameter  $\epsilon$  for the scene with occlusion. We noticed a pretty abrupt change between an inaccurate solution and a solution which requires a large number of form factors, and consequently overly long computations. The selection of the parameter was more difficult for multiwavelets and CDV wavelets than for Haar. We also tried the multilevel like refinement method from subsection 2.3.3. Here we also found it hard to devise a good scheme for reducing  $\epsilon$ .

## 4.2 Comparison of Solution Methods

We apply different solution methods to solve the system of equations resulting from a compressed kernel and evaluate their convergence. Our main interest is in the behavior of certain Krylov subspace methods (CGNR, GMRES and CG) in comparison to the commonly applied Picard iteration.

### 4.2.1 Test Scenes

As seen in Section 2.3 the Picard iteration generally converges rather rapidly for radiosity problems. One additional step of iteration corresponds to one additional reflection of light on a surface. So, scenes with significant degree of reflection potentially need a higher number of iterations. These scenes are where we anticipate other iterative methods to improve compared to the Picard iteration. We experiment with two geometries in two different physical configurations. The geometry of Figure 4.10 has no occlusion, the geometry of Figure 4.11 has. We compute the error for each geometry for surfaces with high average reflectivity  $\rho_{avg} = 0.52$  and with low average reflectivity  $\rho_{avg} = 0.35$ :

surface	$S_0$	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$	$S_8$	$S_9$	$S_{10}$	$S_{11}$
high $\rho_i =$	0.8	0.8	0.8	0.8	0.4	0.4	0.4	0.4	0.4	0.4(a)/0.8(b)	0.4	0
low $\rho_i =$	0.4	0.4	0.4	0.4	0.2	0.2	0.4	0.4	0.4	0.4(a)/0.8(b)	0.4	0

Convergence of the Picard iteration in the high reflectivity setting is slower than in the low reflectivity setting due to the increased spectral radius of the system. We anticipate slower convergence of Picard for the scene with occlusion since there are surfaces which receive only indirect illumination and consequently need an additional step of reflection to receive light.

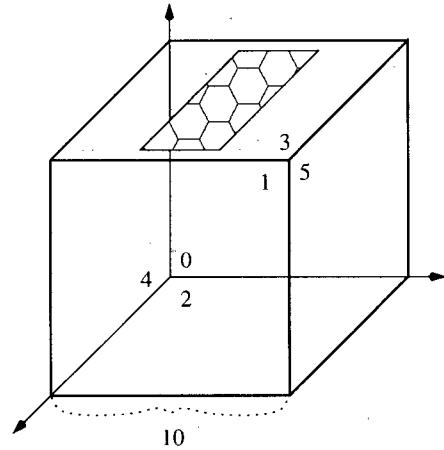


Figure 4.10: Box scene without occlusion. A number indicates the index  $i$  of a surface  $S_i$ . All surfaces have constant reflectivity  $\rho_i$  as specified in the text. The position of the light-source is as specified in Figure 4.2.

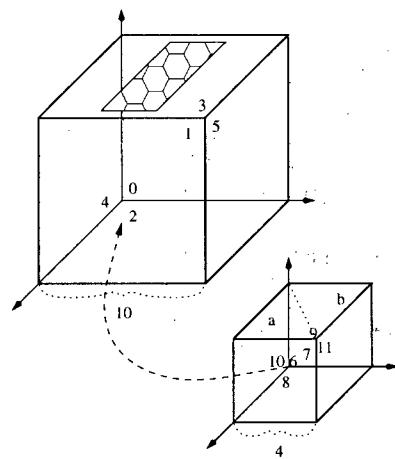


Figure 4.11: Box scene with occlusion. All specifications are as for Figure 4.10. The small cube is located at position  $(1, 1, 1)$  inside the big cube.

## 4.2.2 Experiments

In order to get a clear distinction between error caused by the discretization and error caused by the solution method we need to compare the solution  $\underline{x}^{(k)}$  obtained in the  $k$ -th step of the iteration to the exact solution  $\underline{x}$  of the system. Since such a solution is not available we compare with a solution  $\hat{\underline{x}}$  obtained after 80 steps of the Picard iteration. The change in the solution after 80 iterations was in the order of  $10^{-11}$ , which is way below the analyzed error: the latter was in the order of  $10^{-3}$  or larger.

We compute the error without weighting by area since here we are only interested in the convergence of the solver, independent of the radiosity equation itself. We use the following formula:

$$\text{error} = \sqrt{\frac{\sum_k (\underline{x}_k - \hat{\underline{x}}_k)^2 / n}{\sum_k \hat{\underline{x}}_k^2 / n}}$$

All computations were conducted with a Haar basis. As initial guess for the iteration we chose the discrete representation  $\underline{b}$  of the light-sources. The results presented were computed for  $\epsilon = 10^{-4}$ , however we also made measurements for  $\epsilon = 0.1$  and  $\epsilon = 10^{-3}$  and obtained similar results. The maximum number of levels is  $m = 5$ . Form factors were computed with  $n_{samp} = 3$ . The total number of links computed in these computations is given in the following table:

Scene	Unoccluded		Occluded	
# of potential links	2 516 580		10 066 320	
reflectivity	High	Low	High	Low
# of computed links	484 452	283 272	898 016	606 398

We present graphs of the computed error in Figure 4.12, Figure 4.13, Figure 4.14 and Figure 4.15.

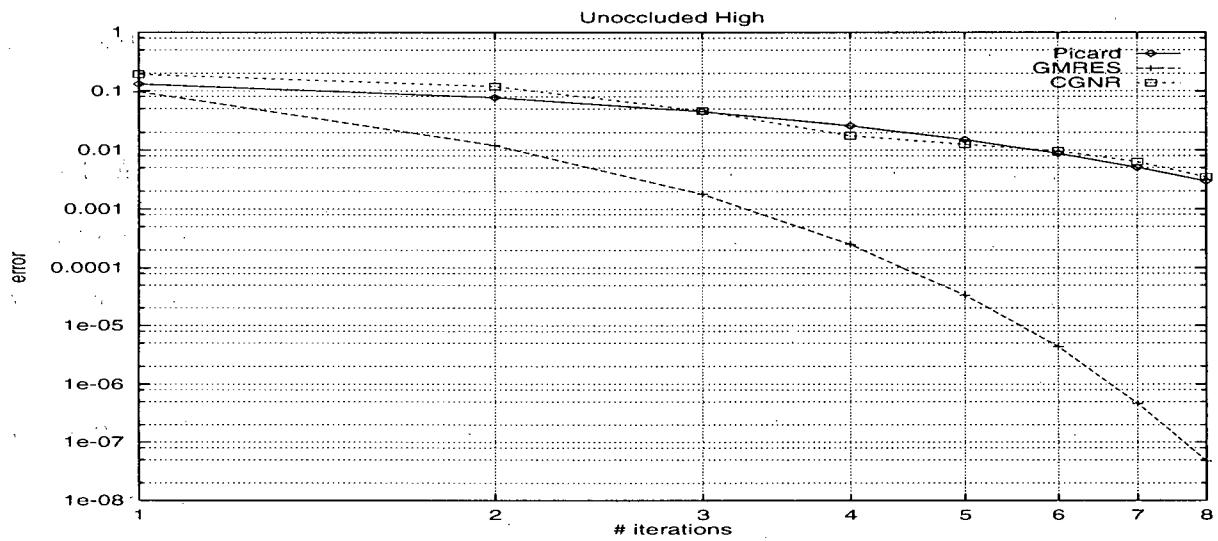


Figure 4.12: Convergence of iterative methods for a test scene without occlusion and high average reflectivity.

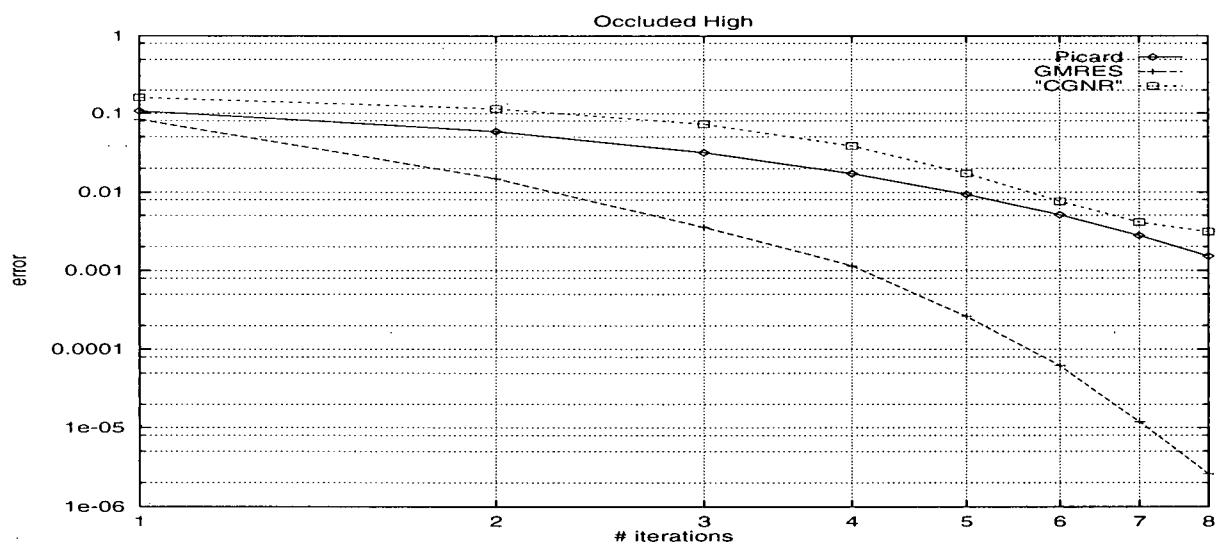


Figure 4.13: Convergence of iterative methods for a test scene with occlusion and high average reflectivity.

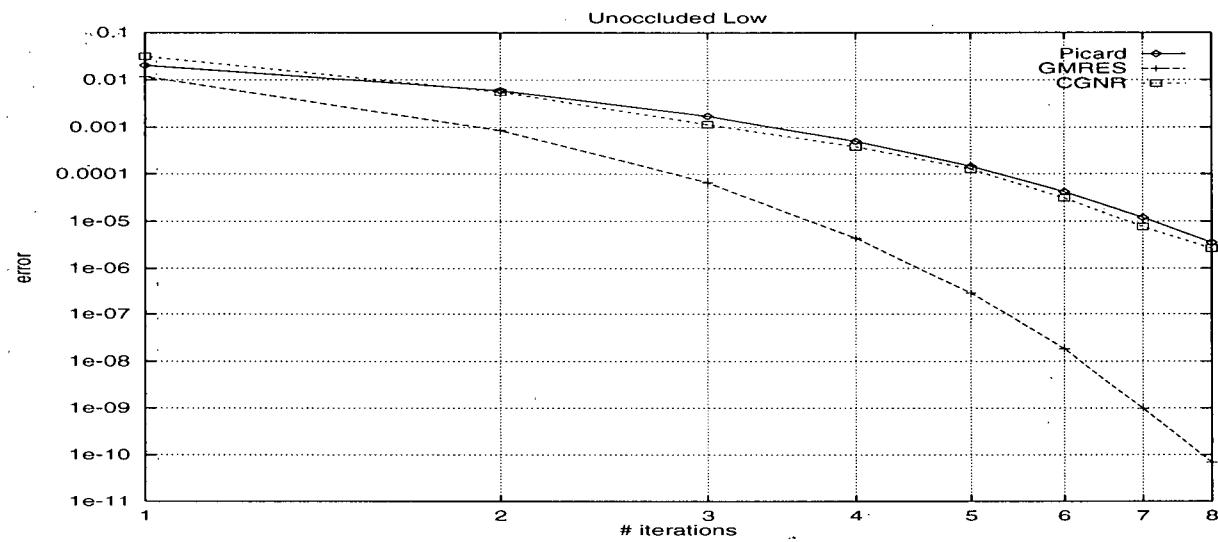


Figure 4.14: Convergence of iterative methods for a test scene without occlusion and low average reflectivity.

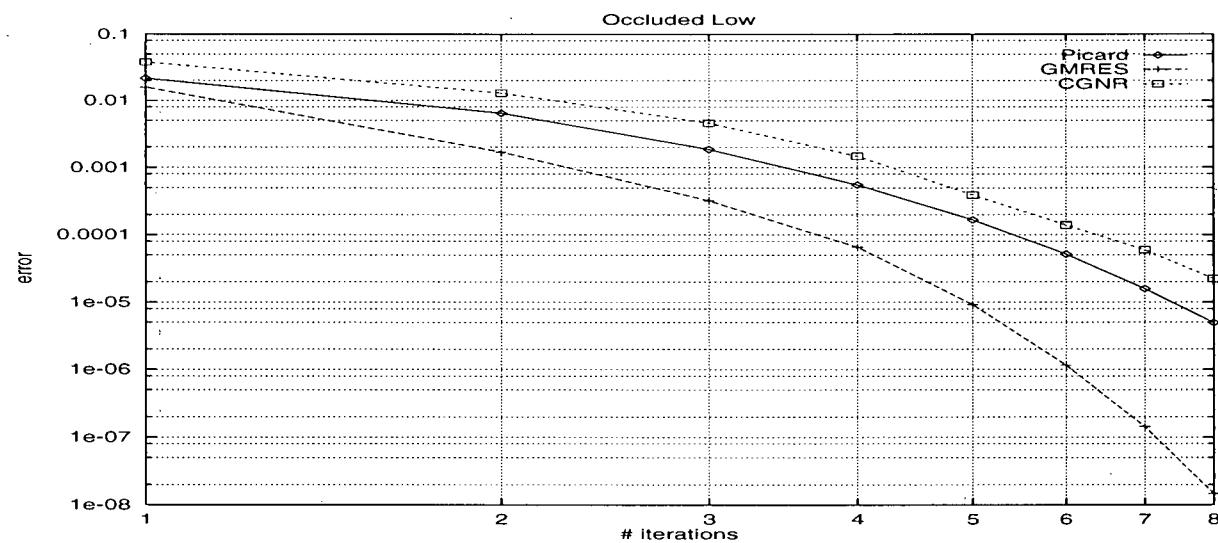


Figure 4.15: Convergence of iterative methods for a test scene with occlusion and low average reflectivity.

### 4.2.3 Interpretation

The graphs clearly show that CGNR has no advantage over the Picard iteration. In particular, for the occluded scene CGNR has a significantly larger error. On the other hand, GMRES has a clear advantage over Picard iteration. After three iterations the error is only  $\approx \frac{1}{25}$  of the error of Picard in the unoccluded cases,  $\approx \frac{1}{9}$  and  $\approx \frac{1}{5}$  for the occluded scenes. In all scenes three iterations of GMRES are sufficient for a perceptual convergence.

The major cost in each step of Picard iteration and GMRES is one sparse matrix-vector multiplication. In wavelet radiosity this step includes wavelet decomposition and wavelet reconstruction. GMRES additionally has a number of vector-vector multiplications the cost of which however is negligible compared to the operations mentioned before. So in terms of cost GMRES has a clear advantage over Picard. CGNR on the other hand performs clearly worse than Picard since it is about twice more expensive per iteration than Picard.

# **Chapter 5**

## **Conclusions**

We have presented a framework of radiosity methods which we hope will help researchers in the areas of numerical analysis and computer graphics to gain better access to the problems arising in the solution of the radiosity equation. We compared solution methods for wavelet radiosity as well as the behavior of different basis-functions. We believe that this analysis will contribute to a better understanding of wavelet radiosity.

We will first give a brief summary of results obtained in our experiments. Subsequently we list a number of questions which arose in our research, which we could not find answered in the literature and which remain to be pursued.

### **5.1 Results**

#### **5.1.1 Wavelet Bases**

In our comparison both examined higher order bases give perceptually better results than the Haar basis. CDV wavelets tend to produce perceptually more accurate results than multiwavelets.

We observed that CDV wavelets allow computation of a more accurate solutions without necessitating more links than needed for Haar wavelets. The number of links is reduced to up to half of the number of links required for Haar wavelets. In order to reduce memory consumption CDV wavelets appear to be a safe alternative to Haar wavelets. This advantage is important for hierarchical methods since we have to store the entire discretized system at a time.

In terms of speed Haar wavelets outperform both higher-order wavelets even if we do not consider the faster methods available for the computation of Haar form factors. The computation of a higher-order link is intrinsically at least 5 times slower than the computation of a Haar link. This results in a clear disadvantage of higher-order wavelets in all observed cases.

We confirm the observation made in [will97a] that fractional visibility produces inaccurate results with higher-order basis-functions.

### 5.1.2 Solution Methods

We showed that GMRES is an easy-to-realize alternative to the Picard iteration. For settings with high average reflectivity GMRES converged in only 3 iterations to a solution with an error for which Picard needed more than 8 iterations. In low reflectivity settings GMRES obtained a solution after 3 iterations that was reached by Picard after 5 – 6 iterations. In the context of wavelet radiosity one step of GMRES and one step of Picard iteration have about the same cost. CGNR performed always slower than Picard iteration.

## 5.2 For Further Investigation

There remain a number of unanswered questions and straightforward extensions to the work presented in the foregoing chapters. In the following we present a list of those which we consider to be most important.

- We did not analyze GMRES in the framework of the multilevel-like methods from subsection 2.3.3. Employing GMRES in this framework might lead to a significant speed-up since it can potentially result in a computation of less form factors, however, with possibly more iterations. A natural question then would also be the analysis of preconditioners for GMRES. A simple preconditioner would be the matrix  $R$  mentioned in subsection 2.3.1.
- Modeling the solution with a mix of different bases appears to be an interesting approach. Areas with small gradient in the solution can be well approximated with a small number of higher order basis-functions while according to our experiments big gradients in the solution are best approximated with the Haar basis. Different bases appear to be easy to integrate as long as each surface has only one type of basis. Also, multiwavelets and CDV wavelets of higher than first-order can be integrated without much effort.
- An important element missing in the analysis is how well the oracle actually estimates the form factor, especially in the case of the scene having occlusions.

# Bibliography

- [alpe93] B. K. Alpert. “A class of bases in  $L^2$  for the sparse representation of integral operators”. *SIAM J. Math. Anal.*, Vol. 24, No. 1, pp. 246–262, January 1993.
- [appe85] A. A. Appel. “An efficient program for many-body simulation”. *SIAM J. Sci. Stat. Computing*, Vol. 6, No. 1, pp. 85–103, 1985.
- [atki76] K. E. Atkinson. *A Survey of Numerical Methods for the Solution of Fredholm Integral Equations of the Second Kind*. SIAM, Philadelphia, 1976.
- [bara95] G. V. Baranowski, R. Bramley, and P. Shirley. “Iterative Methods for Fast Radiosity Solutions”. Technical report, Indiana University, Bloomington IN, <http://www.cs.indiana.edu/ftp/techreports/index.html>, 1995.
- [baum90] D. R. Baum and J. M. Winget. “Real Time Radiosity Through Parallel Processing and Hardware Acceleration”. *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, Vol. 24, pp. 67–75, March 1990.
- [beka96] P. Bekaert and Y. D. Willems. “Hirad: A Hierarchical Higher Order Radiosity Implementation”. *Proceedings of the Twelfth Spring Conference on Computer Graphics (SCCG '96)*, June 1996.
- [beyl91] G. Beylkin, R. Coifman, and V. Rohklin. “Fast Wavelet Transforms and Numerical Algorithms 1”. *Comm. Pure Appl. Math.*, Vol. 64, pp. 141–183, 1991.
- [boua95] K. Bouatouch and S. N. Pattanaik. “Discontinuity Meshing and Hierarchical Multiwavelet Radiosity”. *Proceedings of Graphics Interface '95*, pp. 109–115, May 1995.
- [bric95] F. Bricout and E. Lepretre. “Distributed Progressive Radiosity on a Workstation Network”. *Proceedings of Parallel and Distributed Processing Techniques and Applications (PDPTA '95)*, November 1995.

- [chen89] Sh. E. Chen. “A Progressive Radiosity Method and its Implementation in a Distributed Processing Environment”. M.Sc. thesis, Program of Computer Graphics, Cornell University, Ithaca, NY, January 1989.
- [chia97] G. Chiavassa and P. Liandrat. “On the effective Construction of Compactly Supported Wavelets Satisfying Homogeneous Boundary Conditions on the Interval”. *Applied and Computational Harmonical Analysis*, Vol. 4, pp. 62–73, 1997.
- [chri94] P. H. Christensen, E. J. Stollnitz, D. H. Salesin, and T. D. DeRose. “Wavelet Radiance”. *Fifth Eurographics Workshop on Rendering*, pp. 287–302, June 1994.
- [cohe85] M. F. Cohen and D. P. Greenberg. “The Hemi-Cube. A Radiosity Solution for Complex Environments”. *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, Vol. 19, No. 3, pp. 31–40, July 1985.
- [cohe86] M. F. Cohen, D. Greenberg, D. Immel, and P. Brock. “An Efficient Radiosity Approach for Realistic Image Synthesis”. *IEEE Computer Graphics and Applications*, Vol. 6, No. 3, pp. 26–35, March 1986.
- [cohe88] M. F. Cohen, S. Chen, J. Wallace, and D. Greenberg. “A Progressive Refinement Approach to Fast Radiosity Image Generation”. *Computer Graphics (ACM SIGGRAPH '88 Proceedings)*, Vol. 22, No. 4, pp. 75–84, August 1988.
- [cohe93a] A. Cohen, I. Daubechies, and P. Vial. “Wavelets on the Interval and Fast Wavelet Transforms”. *Applied and Computational Harmonic Analysis*, Vol. 1, pp. 54–81, 1993.
- [cohe93b] M. F. Cohen and J. R. Wallace. *Radiosity and Realistic Image Synthesis*. Academic Press, Cambridge, 1993.
- [daub88] I. Daubechies. “Orthonormal Basis of Compactly Supported Wavelets”. *Comm. Pure. Appl. Math.*, Vol. 41, 1988.
- [daub92] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.
- [feda92] M. Feda and W. Purgathofer. “Accelerating Radiosity by Overshooting”. *Third Eurographics Workshop on Rendering*, pp. 21–32, May 1992.
- [garc97] B. Garcia and X. Pueyo. “Progressive Radiosity Solutions on SIMD Architecture”. *Proc. First Eurographics Workshop on Parallel Graphics and Visualisation*, September 1997.

- [gers94] R. Gershbein, P. Schroder, and P. Hanrahan. “Textures and Radiosity: Controlling Emission and Reflection with Texture Maps”. *Computer Graphics (ACM SIGGRAPH '94 Proceedings)*, pp. 51–58, 1994.
- [gers95] R. Gershbein. “Integration Methods for Galerkin Radiosity Couplings”. *Rendering Techniques '95*, pp. 264–273, June 1995.
- [gora84] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. “Modelling the Interaction of Light between Diffuse Surfaces”. *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, Vol. 18, No. 3, pp. 212–222, July 1984.
- [gort93a] S. J. Gortler and M. F. Cohen. “Radiosity and Relaxation Methods”. Technical report, Princeton University, 1993.
- [gort93b] S. J. Gortler, P. Schroeder, M. F. Cohen, and P. Hanrahan. “Wavelet Radiosity”. *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, August 1993.
- [gort94] S. J. Gortler, M. F. Cohen, and Philipp Slusallek. “Radiosity and Relaxation Methods”. *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, November 1994.
- [gort95] S. J. Gortler. *Wavelet Methods for Computer Graphics*. Ph.D. thesis, Technical Report, Department of Computer Science, Princeton University, Princeton, NJ, January 1995.
- [guit91] P. Guittot, J. Roman, and C. Schlick. “Two parallel approaches for a progressive radiosity”. *Second Eurographics Workshop on Rendering*, May 1991.
- [gwin87] J. Gwinner. “On the Galerkin approximation of nonsmooth boundary integral equations arising in radiative heat transfer”. *Boundary elements IX, Vol. 3 (Stuttgart, 1987)*, pp. 257–264. Comput. Mech., Ashurst Lodge, Ashurst, Southampton SO40 7AA, UK, 1987.
- [hack95] W. Hackbusch. *Integral Equations*. Birkhäuser Verlag, Basel Boston Berlin, 1995.
- [hanr91] P. Hanrahan, D. Salzman, and L. Aupperle. “A Rapid Hierarchical Radiosity Algorithm”. *Computer Graphics (ACM SIGGRAPH '91 Proceedings)*, Vol. 25, No. 4, pp. 197–206, July 1991.
- [heck91] P. S. Heckbert and J. M. Winget. “Finite Element Methods for Global Illumination”. Technical report, Computer Science Division; University of California, Berkeley, July 1991.

- [heck92] P. S. Heckbert. "Discontinuity Meshing for Radiosity". *Third Eurographics Workshop on Rendering*, pp. 203–226, May 1992.
- [hest52] M. R. Hestenes and E. Stiefel. "Methods of conjugate gradients for solving linear systems". *J. Res. National Bureau of Standards*, Vol. 49, pp. 409–436, 1952.
- [jawe94] B. Jawerth and W. Sweldens. "An Overview of Wavelet Based Multiresolution". *SIAM Review*, Vol. 36, No. 3, pp. 377–412, September 1994.
- [kaji86] J. T. Kajiya. "The Rendering Equation". *Computer Graphics (ACM SIGGRAPH '86 Proceedings)*, Vol. 20, No. 4, pp. 143–149, 1986.
- [lang95] E. Langenou, K. Bouatouch, and M. Chelle. "Global Illumination in Presence of Participating Media with General Properties". *Photorealistic Rendering Techniques (Proceedings of the Fifth Eurographics Workshop on Rendering)*, pp. 71–86, 1995.
- [lisc92] D. Lischinski, F. Tampieri, and D. P. Greenberg. "Discontinuity Meshing for Accurate Radiosity". *IEEE Computer Graphics and Applications*, Vol. 12, No. 6, pp. 25–39, November 1992.
- [lisc93] D. Lischinski, F. Tampieri, and D. P. Greenberg. "Combining Hierarchical Radiosity and Discontinuity Meshing". *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pp. 199–208, 1993.
- [neum95] L. Neumann and R. F. Tobler. "New Efficient Algorithms with Positive Definite Radiosity Matrix". *Photorealistic Rendering Techniques (Fifth Eurographics Workshop on Rendering)*, pp. 227–243, 1995.
- [niev97] Y. Nievergelt. "Making Any Radiosity Matrix Symmetric Positive Definite". *Journal of the Illuminating Engineering Society*, Vol. 26, No. 1, pp. 165–172, 1997.
- [nish85] T. Nishita and E. Nakamae. "Continuous Tone Representation of Three-Dimensional Objects Taking Account of Shadows and Interreflection". *Computer Graphics (ACM SIGGRAPH '85 Proceedings)*, Vol. 19, No. 3, pp. 23–30, July 1985.
- [reck90] R. J. Recker, D. W. George, and D. P. Greenberg. "Acceleration Techniques for Progressive Refinement Radiosity". *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, Vol. 24, pp. 59–66, March 1990.

- [rush90] H. E. Rushmeier and K. E. Torrance. "Extending the Radiosity Method to Include Specularly Reflecting and Translucent Materials". *ACM Transactions on Graphics*, Vol. 9, No. 1, pp. 1–27, January 1990.
- [saad95] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 20 Park Plaza, Boston, MA, 1995.
- [scha97] S. Schaefer. "Hierarchical Radiosity on Curved Surfaces". *Rendering Techniques '97 (Proceedings of the Eighth Eurographics Workshop on Rendering)*, pp. 187–192, 1997.
- [schr93] P. Schröder. "Numerical Integration for Radiosity in the Presence of Singularities". *Fourth Eurographics Workshop on Rendering*, Series EG 93 RW, pp. 177–184, June 1993.
- [schr94] P. Schröder. *Wavelet Algorithms for Illumination Computations*. Ph.D. thesis, Technical Report, Department of Computer Science, Princeton University, Princeton, NJ, November 1994.
- [schr96] P. Schröder. "Wavelet Radiosity: Wavelet Methods for Integral Equations". *ACM SIGGRAPH '96 Course Notes - Wavelets in Computer Graphics*, pp. 143–165. SIGGRAPH, 1996.
- [shao93] M. Shao and N. I. Badler. "Analysis and Acceleration of Progressive Refinement Radiosity Method". *Fourth Eurographics Workshop on Rendering*, EG 93 RW, pp. 247–258, June 1993.
- [shew94] J. R. Shewchuk. "An introduction to the conjugate gradient method without the agonizing pain". Technical report, Carnegie Mellon University, Pittsburgh, PA <http://www.cs.cmu.edu/~quake/papers.html>, 1994.
- [sill94] F. X. Sillion and C. Puech. *Radiosity and Global Illumination*. Morgan Kaufmann Publishers, San Francisco, 1994.
- [smit92] B. E. Smits, J. R. Arvo, and D. H. Salesin. "An Importance-Driven Radiosity Algorithm". *Computer Graphics (ACM SIGGRAPH '92 Proceedings)*, Vol. 26, pp. 273–282, July 1992.
- [smit94] B. Smits, J. Arvo, and D. P. Greenberg. "A Clustering Algorithm for Radiosity in Complex Environments". *Computer Graphics Proceedings, Annual Conference Series, 1994 (ACM SIGGRAPH '94 Proceedings)*, pp. 435–442, 1994.

- [sout46] R. Southwell. *Relaxation Methods in Theoretical Physics*. Clarendon Press, Oxford, 1946.
- [stoe80] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer Verlag, New York Berlin Heidelberg, 1980.
- [stol96] E. Stollnitz, T. DeRose, and D. Salesin. *Wavelets for Computer Graphics*. Morgan Kaufmann Publishers, San Francisco, California, 1996.
- [tell94] S. Teller, C. Fowler, T. Funkhouser, and P. Hanrahan. “Partitioning and Ordering Large Radiosity Computations”. *Computer Graphics (ACM SIGGRAPH '94 Proceedings)*, pp. 443–450, 1994.
- [trou93] R. Troutman and N. L. Max. “Radiosity Algorithms Using Higher Order Finite Element Methods”. *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pp. 209–212, 1993.
- [veac97] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. Ph.D. thesis, Stanford University, December 1997.
- [whit80] T. Whitted. “An improved illumination model for shaded display”. *Communications of the ACM*, Vol. 23, No. 6, pp. 343–349, 1980.
- [will97a] A. Willmott and P. S. Heckbert. “An Empirical Comparison of Radiosity Algorithms”. Technical Report CMU-CS-97, Carnegie Mellon University School of Computer Science, 1997.
- [will97b] A. Willmott and P. S. Heckbert. “An Empirical Comparison of Radiosity Algorithms”. *Rendering Techniques 97 (Proceedings of the Eurographics Workshop on Rendering in St. Etienne)*, pp. 175–186, June 1997.
- [xu94] W. Xu and D. S. Fussell. “Constructing Solvers for Radiosity Equation Systems”. *Photorealistic Rendering Techniques (Proceedings of the Fifth Eurographics Workshop on Rendering)*, pp. 207–217, June 1994.
- [zatz93] H. R. Zatz. “Galerkin Radiosity: A Higher-Order Solution Method for Global Illumination”. *Computer Graphics (ACM SIGGRAPH '93 Proceedings)*, pp. 213–220, August 1993.

# Appendix A

## A.1 Two-Dimensional Wavelet Transform

We generalize the one-dimensional MRA presented in section 3.1 by forming products of scaling functions and wavelets. This results in a new two-dimensional scaling function and three new two-dimensional wavelets:

$$\begin{aligned}\phi_{k_0, k_1}(x, y) &= \phi_{k_0}(x)\phi_{k_1}(y) \\ \psi_{k_0, k_1}^{\alpha}(x, y) &= \psi_{k_0}(x)\phi_{k_1}(y) \\ \psi_{k_0, k_1}^{\beta}(x, y) &= \phi_{k_0}(x)\psi_{k_1}(y) \\ \psi_{k_0, k_1}^{\gamma}(x, y) &= \psi_{k_0}(x)\psi_{k_1}(y)\end{aligned}$$

With these definitions we get new two-dimensional dilation equations for scaling functions and wavelets:

$$\begin{aligned}\phi_{j, k_0, k_1} &= \sum_{l_0=0}^{n_{j+1}-1} \sum_{l_1=0}^{n_{j+1}-1} h_{k_0, l_0} h_{k_1, l_1} \phi_{j+1, l_0, l_1} \\ \psi_{j, k_0, k_1}^{\alpha} &= \sum_{l_0=0}^{n_{j+1}-1} \sum_{l_1=0}^{n_{j+1}-1} g_{k_0, l_0} h_{k_1, l_1} \phi_{j+1, l_0, l_1} \\ \psi_{j, k_0, k_1}^{\beta} &= \sum_{l_0=0}^{n_{j+1}-1} \sum_{l_1=0}^{n_{j+1}-1} h_{k_0, l_0} g_{k_1, l_1} \phi_{j+1, l_0, l_1}\end{aligned}$$

$$\psi_{j,k_0,k_1}^{\gamma} = \sum_{l_0=0}^{n_{j+1}-1} \sum_{l_1=0}^{n_{j+1}-1} g_{k_0,l_0} g_{k_1,l_1} \phi_{j+1,l_0,l_1}$$

Bekaert et. al. [beka96] present a way to define two-dimensional MRA on non-rectangular grids.

## A.2 Non-Standard Representation

For a given MRA the interaction of the projector  $P_m$  with  $T$  can be written as

$$P_m T P_m = P_0 T P_0 + \sum_{j=0}^{m-1} Q_j T Q_j + \sum_{j=0}^{m-1} Q_j T P_j + \sum_{j=0}^{m-1} P_j T Q_j. \quad (\text{A.1})$$

This equality can be easily obtained using the identity  $Q_j = P_{j+1} - P_j$  and the identities

$$P_m T P_m = P_0 T P_0 + \sum_{j=0}^{m-1} P_{j+1} T P_{j+1} - P_j T P_j$$

and

$$\begin{aligned} P_{j+1} T P_{j+1} - P_j T P_j &= (P_{j+1} - P_j) T (P_{j+1} - P_j) + P_j T P_{j+1} + P_{j+1} T P_j - 2P_j T P_j \\ &= Q_j T Q_j + P_j T Q_j + Q_j T P_j. \end{aligned}$$