

# The Global Cube A Hardware-accelerated Hierarchical Volume Radiosity Technique

*Andreas Hubeli<sup>†</sup>, Lars Lippert<sup>\*</sup>, Markus Gross<sup>†</sup>*

<sup>†</sup>Computer Science Department  
ETH Zurich, Switzerland  
e-mail: {hubeli, grossm}@inf.ethz.ch  
<http://graphics.ethz.ch>

<sup>\*</sup>Compaq, München, Germany



# The Global Cube

## A Hardware-accelerated Hierarchical Volume Radiosity Technique

A. Hubeli<sup>‡</sup>, L. Lippert<sup>†</sup>, and M. Gross<sup>‡</sup>

<sup>‡</sup> Computer Science Department, Swiss Federal Institute of Technology,  
Zurich, Switzerland

<sup>†</sup> Compaq, München, Germany

### ABSTRACT

In this paper we introduce the notion of the *global cube algorithm*. Its concept and its implementation allow the computation of the energy transfer of light in a volumetric medium by exploiting graphics hardware. Until now, volumetric radiosity techniques are limited to compute the equilibrium of energy for a maximum of a few tens of thousand volume elements due to the complexity of the underlying computations and the memory requirements. The method we present is accurate, efficient and can be applied to evaluate several million volume elements.

We use wavelets to construct a multigrid method which approximates the underlying physical model. Based on a hierarchical model, our method first computes the radiosity distribution for a coarse representation of the volume of interest, then the representation is refined adaptively.

### KEYWORDS

volume radiosity, global illumination equation, hierarchical methods, wavelets, transport theory, texture mapping hardware, multigrid methods

## 1 INTRODUCTION

Within computer graphics, the problem of determining the appearance of an environment by simulating the transport of light is known as global illumination. For global illumination, the governing volume radiosity equation can be derived directly from the linear transport theory [1, 11, 20]. In order to compute the light distribution of an environment, the emissions of the light sources are scattered in the volume; the light scattered in this process also creates indirect illumination effects that have to be integrated into the light transport model. The resulting global illumination equation is an integral equation, which is often referred to as the *volume rendering equation*. The solution to this equation is very expensive to compute since in order to evaluate the radiosity at a single point we require the knowledge over the radiosity distribution in the scene, which corresponds to the solution we are looking for. Although a great deal of research has been directed at Monte Carlo [2, 3, 19, 28] and finite element methods [34, 13] for solving the rendering equation and the physical process of particle interactions, the size of the scenes is still limited to several thousand volume elements or surface patches.

Solving the rendering equation either by Monte Carlo or finite element methods adds subtle, nevertheless visually important effects to the computed images such as color bleeding and penumbras along shadow boundaries. These effects, along with other global illumination effects, are essential to convey the illusion of looking at a real scene.

A common solution technique for the computation of the energy distribution within a volume is the zonal method [31]. This technique handles both volumetric and surface objects by strictly dividing the scene into discrete segments that contain either participating media or surface patches. The energy distribution is finally determined by a numerical approximation of the distribution between the surface and volume segments. The hierarchical formulation of the zonal method [32] simulates the energy exchanges between hierarchically organized diffuse surfaces, isotropic participating media, and object clusters but does not incorporate the wavelet theory.

The classical radiosity method also numerically approximates the distribution of energy of a given scene. The underlying radiosity equation [9, 27] is derived from the equation of transfer based on the assumption that all surfaces are ideally diffuse and there is no existence of a participating medium. For a more complete global illumination model, however, all phenomena must be accounted for.

For both the radiosity and the zonal method, numerical approximation leads to discrete versions of the Fredholm integral equation that uses a large dense matrix of form factors to describe the energy interactions between the discrete surface patches and/or volumes. To obtain a detailed solution, one must set up the matrix and solve the linear system that results. For practical issues this can become very expensive in terms of memory requirements and computational costs.

The method proposed in this paper computes the energy exchange within the volume without any explicit form factor computation. The global cube concept [21] is further extended to hierarchical, wavelet-based structures, which enables the computation of the energy equilibrium to any specified precision.

Following this introduction, in the second section we will briefly resume the mathematical and physical concepts underlying the our new algorithm. The third section discusses the global cube algorithm in detail. We then present the hierarchical extension of the global cube concept in section four. The results and performance analyses are presented in section five. The final section concludes and points out future work.

## 2 VOLUME RADIOSITY

The purpose of this chapter is to define a new pure volumetric illumination model upon which our new global illumination rendering approach will be based. We further motivate the finite element solution method and establish a link between wavelet theory and the illumination model.

### 2.1 Pure Volumetric Model

The stationary gray equation of transfer in integral form [20] describes the losses and gains of the particle intensity in a volume  $V$ . The intensity  $\chi(\mathbf{x}, \omega)$  denotes the number of particles at a point  $\mathbf{x} \in V$  which move into direction  $\omega$ . It is given as

$$\chi(\mathbf{x}, \omega) = \chi'(\mathbf{x}, \omega) + \int_{\partial V} \varphi(\mathbf{x}, \omega', \omega) L(\mathbf{x}_0', \omega') e^{-A(\mathbf{x}_0', \mathbf{x})} d\mathbf{x}_0' + \int_V \varphi(\mathbf{x}, \omega', \omega) \frac{\chi(\mathbf{x}', \omega')}{|\mathbf{x} - \mathbf{x}'|^2} e^{-A(\mathbf{x}', \mathbf{x})} d\mathbf{x}' \quad (1)$$

Equation (1) reveals that the value of the intensity is computed from three components

- the source term  $\chi'(\mathbf{x}, \omega)$  describes the number of particles emitted by  $\mathbf{x}$  in direction  $\omega$ .

- the energy received from the boundary elements  $\mathbf{x}_0' \in \partial V$ . This term results from the introduction of boundary conditions to the equation of transfer and accounts for the radiance  $L$  of the boundary elements. The localization of  $\mathbf{x}_0'$  where the boundary surface is encountered depends both on  $\mathbf{x}$  and  $\omega'$ .
- the volumetric contribution, which counts for intensities  $\chi(\mathbf{x}')$  received from all  $\mathbf{x}' \in V$ .

The *optical depth*  $A(\mathbf{x}_1, \mathbf{x}_2)$  between two points  $\mathbf{x}_1 = \mathbf{p} + s_1 \mathbf{n}$  and  $\mathbf{x}_2 = \mathbf{p} + s_2 \mathbf{n}$  is given as

$$A(\mathbf{x}_1, \mathbf{x}_2) = \int_{s_1}^{s_2} \alpha(\mathbf{p} + s\mathbf{n}) ds \quad (2)$$

where  $\alpha(x)$  denotes the total extinction coefficient, which characterizes the general loss of energy in the volume. Figure 1 depicts the relevant parameters of the transport theory model.

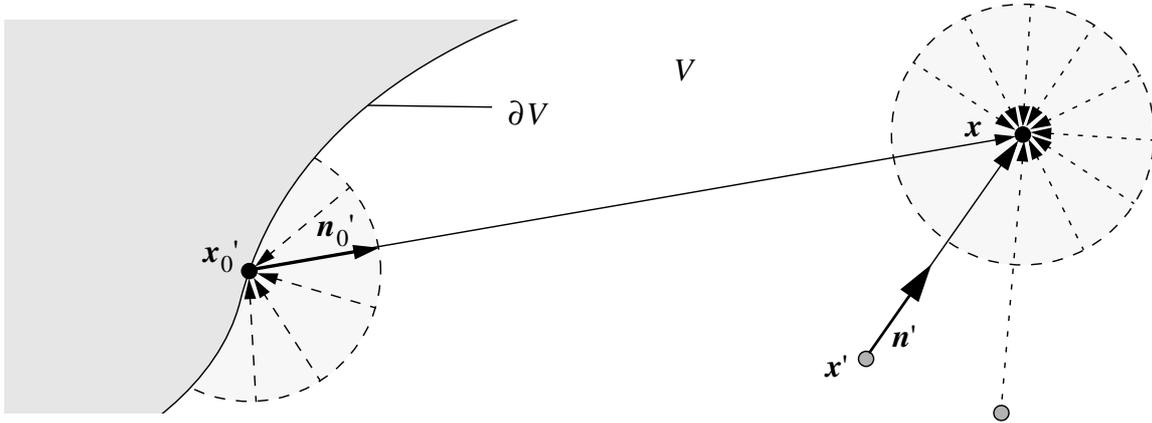


Figure 1: Volume radiosity setup.

From equation (1) we observe that the energy received from the boundary elements and the volumetric contribution are weighted by the *bidirectional reflection distribution function*  $\varphi(\mathbf{x}, \omega', \omega)$  which describes the ratio of energy reflected from the point  $\mathbf{x}$  in direction  $\omega$  to energy received from the point  $\mathbf{x}'$  from direction  $\omega'$ .

By assuming non-emitting and non-reflecting boundary elements and only diffuse reflection such that

we can define an isotropic scattering function  $\alpha^s(\mathbf{x})$  as  $\varphi(\mathbf{x}, \omega, \omega') = \frac{\alpha^s(\mathbf{x})}{4\pi}$  we obtain

$$\chi(\mathbf{x}, \omega) = \chi'(\mathbf{x}, \omega) + \frac{1}{4\pi} \alpha^s(\mathbf{x}) \int_V \frac{\chi(\mathbf{x}', \omega')}{|\mathbf{x} - \mathbf{x}'|^2} e^{-A(\mathbf{x}', \mathbf{x})} d\mathbf{x}' \quad (3)$$

It is important to note that the upper assumptions are not general restrictions of our method, since surfaces can be represented as thin volumetric objects in this model. In the following sections we will denote this model as the *pure volumetric model*.

By introducing a kernel function

$$\kappa(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi} \alpha^s(\mathbf{x}) \frac{1}{|\mathbf{x} - \mathbf{x}'|^2} e^{-A(\mathbf{x}', \mathbf{x})} \quad (4)$$

we can rewrite (3) as

$$\chi(\mathbf{x}, \omega) = \chi^t(\mathbf{x}) + \int_V \kappa(\mathbf{x}, \mathbf{x}') \cdot \chi(\mathbf{x}', \omega') d\mathbf{x}' \quad (5)$$

To keep the notation simpler, we have from (5) in operator notation

$$\chi = \chi^t + K\chi \quad (6)$$

where  $K$  denotes the *kernel integral operator*. This type of equation corresponds to a Fredholm integral equation of the second kind. An inherent feature of this equation is that the unknown volume emission appears outside and inside the integral. It can be shown that since  $\|\kappa(\mathbf{x}, \mathbf{x}')\| < 1$ , this equation can be formally expressed as a Neumann series:

$$\begin{aligned} \chi &= (I - K)^{-1} \chi^t \\ &= \sum_{i=0}^{\infty} K^i \chi^t \\ &= \chi^t + K\chi^t + K^2\chi^t + \dots \end{aligned} \quad (7)$$

From a physical point of view, each application of the kernel integral operator can be seen as an additional volume interaction. Thus, the term  $K^i$  accounts for paths in which light bounces through the scene  $i$  times.

## 2.2 Finite Element Formulation of the Pure Volumetric Model

Goral et al. [13] introduced the finite-element solution or projection methods to global illumination. The method of finite-elements has been used before to simulate thermal radiation, a problem similar to global illumination. Both problems lead to integral equations that describe the energy equilibrium. Generally spoken, the finite-element procedure allows one to obtain an approximate solution of continuous integral equations.

To obtain a finite-element solution, we have to introduce a finite set of shape functions or basis functions  $N_p(\mathbf{x})$  ( $p=1, \dots, n$ ) and associated nodal values  $\chi_{V_p}$ . As in section 2.1, boundary surfaces of the volume will be neglected here. In contrast to the infinite dimensional solution space of the correct function  $\chi$  we project the solution onto the finite-dimensional functional subspace  $\mathcal{V}$  that is spanned by the shape functions. Therefore, the approximation  $\tilde{\chi}$  of the pure volume emission term can be defined in terms of a linear combination of the shape functions, such that

$$\tilde{\chi}(\mathbf{x}) = \sum_{p=1}^n \chi_{V_p} N_p(\mathbf{x}), \quad V_p \in \mathcal{V} \quad (8)$$

It is useful to consider interpolating shape functions with

$$N_p(\mathbf{x}_q) = \delta_{pq} \quad (9)$$

$$\delta_{pq} : \text{Kronecker delta} = \begin{cases} 1 & \text{if } p = q \\ 0 & \text{if } p \neq q \end{cases} \quad (10)$$

This property ensures an exact solution at the sample points  $\mathbf{x}_q$  such that

$$\tilde{\chi}(\mathbf{x}_q) = \chi(\mathbf{x}_q) \quad (11)$$

To measure the quality of approximation we introduce a residual function

$$R_\chi = \tilde{\chi} - \chi^t - K\tilde{\chi} \quad (12)$$

Ideally this residual should be zero or at least minimized in some sense. Two factors will influence the computed results for the finite element formulation:

- the norm that is used for computing the magnitude of the residual  $R_\chi$ .
- the functional space in which we compute the solution.

A very useful approach for the finite element formulation of the pure volumetric model is the *Galerkin method*. Since our global illumination algorithm takes advantage of this method, its concepts will be explained in the following section.

### Galerkin Radiosity

For the case of the classical Galerkin radiosity we require:

$$\langle R_\chi, N_p \rangle = 0 \quad \text{for all } p \in \{1, \dots, n\} \quad (13)$$

This expression is equivalent to:

- the residual function is orthogonal to the subspace  $\mathcal{V}$
- the approximate solution  $\tilde{\chi}(\mathbf{x})$  differs from  $\chi(\mathbf{x})$  in ways that cannot be represented in  $\mathcal{V}$
- the residual projected on  $\mathcal{V}$  equals zero.

Inserting (12), (8) and (5) into (13) results in a linear system of equations

$$\sum_{q=1}^n \chi_{V_q} \left( \langle N_q, N_p \rangle - \left\langle \int_V \kappa(\cdot, \mathbf{x}') \cdot N_q(\mathbf{x}') d\mathbf{x}', N_p \right\rangle \right) = \langle \chi^t, N_p \rangle \quad (14)$$

The solution of this general Galerkin representation is depending on the choice of the basis functions. In case of constant basis functions, such as the 3D tensor-product extensions of the Haar scaling functions described by (23), which correspond to the first order B-Spline, [8] we have

$$\begin{aligned}
 \langle N_q, N_p \rangle &= \delta_{pq} \\
 \int_V f(\mathbf{x}) N_p(\mathbf{x}) d\mathbf{x} &= \int_{V_p} f(\mathbf{x}) d\mathbf{x} \\
 |V_p| &= 1
 \end{aligned} \tag{15}$$

Equation (15) can now be used to simplify (14) which reduces to

$$\sum_{q=1}^n \chi_{V_q} \left( \delta_{pq} - \int_{V_p} \int_{V_q} \kappa(\mathbf{x}, \mathbf{x}') d\mathbf{x}' d\mathbf{x} \right) = \int_{V_p} \chi^t(\mathbf{x}) d\mathbf{x} \tag{16}$$

And we finally obtain

$$\sum_{q=1}^n \chi_{V_q} (\delta_{pq} - \alpha_{V_p}^s F_{V_q V_p}) = \chi_{V_p}^t \tag{17}$$

with

$$F_{V_q V_p} = \frac{1}{4\pi} \int_{V_q} \int_{V_p} \frac{1}{|\mathbf{x} - \mathbf{x}'|^2} e^{-A(\mathbf{x}', \mathbf{x})} d\mathbf{x}' d\mathbf{x} \tag{18}$$

This relationship, which corresponds to a 3D version of the well known *form factor equation* in the classical radiosity, can be expressed as a matrix equation:

$$\mathbf{M} \mathbf{c} = \mathbf{c}^t \tag{19}$$

where  $\mathbf{c}$  denotes the emission vector and  $\mathbf{c}^t$  describes the given source vector. In the context of finite elements, the matrix  $\mathbf{M}$  is denoted as stiffness matrix or mass matrix [17]. In (19) the matrix elements  $m_{pq}$  and vector elements  $c_p$  and  $c_p^t$  are given as

$$\begin{aligned}
 m_{pq} &= \delta_{pq} - \alpha_{V_p}^s F_{V_q V_p} \\
 c_q &= \chi_{V_q} \\
 c_q^t &= \chi_{V_q}^t
 \end{aligned} \tag{20}$$

In order to compute the emission values  $\chi_{V_p}$  from the given intensities  $\chi_{V_p}^t$ , a naive approach could be to compute the inverse of  $\mathbf{M}$  by using the Gaussian elimination technique. This technique, however, exhibits a computational complexity of  $O(n^3)$ . Thus, it becomes rather impractical even for small volume data sets. Therefore, instead of an explicit inversion of the matrix we have to apply relaxation methods that iteratively compute the solution [12].

The *Jacobi iteration* starts with an initial guess  $\mathbf{c}^0$  of the solution  $\mathbf{c}$  and updates the emission coeffi-

icients in iteration  $i$  using the coefficients computed in iteration  $i-1$ . Consequently, we have from (19) for the  $i$ -th generation

$$m_{pp}c_p^{(i)} = c_p^t - \sum_{\substack{q=1 \\ q \neq p}}^n m_{pq}c_q^{(i-1)} \quad (21)$$

Note that the coefficient  $m_{pp}$  measures for all interactions that occur inside the volume element  $V_p$ .

## 2.3 Hierarchical Representation

### Multiresolution Analysis

Volume radiosity techniques must handle very large volumetric data sets. Multiresolution analysis (MRA) allows to construct hierarchical representations of the information, which can then be used to efficiently compute the radiosity distribution in the scene. In this subsection we will briefly describe the components that are used in the *global cube algorithm*; detailed information on MRA and wavelets can be found in [8], [6], [24], [26] and [33].

The hierarchical set of basis functions that is introduced in a MRA provides us with a hierarchical representation of a function  $f(x)$ . Any function  $f(x)$  can be written as a linear combination of a coarse overall shape defined by  $(c_{Mp})_{p \in Z}$  at a level  $M$  and additional detail information  $(d_{mp})_{(m,p) \in Z^2}$  at higher resolutions  $m$ ,  $M \geq m > m_0$ . The resulting representation is shown in (22):

$$f(x) = \sum_p c_{Mp} \phi_{Mp}(x) + \sum_{m=m_0+1}^M \sum_p d_{mp} \psi_{mp}(x) \quad (22)$$

where  $(\phi_{Mp}(x))_{p \in Z}$  is a set of *scalar basis functions*, and  $(\psi_{mp}(x))_{m_0 < m < M, p \in Z}$  is a set of so-called *wavelet basis functions*. In general, this hierarchical representation allows an economical representation of functions. If a function has little or no detail in some region, then the coefficients that code the detail information will be zero or negligibly small in this region. If the corresponding basis function or the associated coefficient is removed from the basis, only a small error will be introduced.

Historically, the first orthonormal multiresolution basis function is the Haar basis in one dimension [16]. The Haar scalar basis functions  $(\phi_{Mp}(x))_{p \in Z}$  are piecewise constant and defined by the scalar mother function:

$$\phi(x) = \begin{cases} 1, & \text{for } 0 \leq x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (23)$$

The Haar wavelet basis functions  $(\psi_{mp}(x))_{m_0 < m < M, p \in Z}$  are also piecewise constant and defined by the wavelet mother function:

$$\psi(x) = \begin{cases} 1, & \text{for } 0 \leq x < 0.5 \\ -1, & \text{for } 0.5 \leq x < 1 \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

The *global cube algorithm* that we present in this paper is based on a wavelet representation that uses the Haar scalar and wavelet basis functions. Further research may shed light on the advantages that higher-order bases such as B-splines on a bounded interval [5, 10] or orthonormalized basis functions on the bounded interval [25] can bring.

### Hierarchical Representation of Volume Data Sets

In the following we describe the relationship between global illumination and wavelet theory by describing hierarchical representations of relevant parameters, and by generalizing the wavelet-based radiosity concept introduced in [14]. Other applications of wavelets in volume rendering can be found in [22], [15], and [23].

Starting from a set of source coefficients  $\chi^t(\mathbf{x})$  and opacity coefficients  $\sigma(\mathbf{x}) = 1 - e^{-A(\mathbf{x})}$  that are given on an equidistant discrete grid, we define  $c_{pqr}^{\chi^t} = \chi^t(x_p, y_q, z_r)$  and  $c_{pqr}^{\sigma} = \sigma(x_p, y_q, z_r)$ . To formalize the representation for emission and absorption, we can derive two functions for the source and opacity terms using the Haar scalar basis functions  $\phi_{m_0 pqr}^3(x, y, z)$ :

$$\begin{aligned} \chi^t(x, y, z) &= \sum_{pqr} c_{m_0 pqr}^{\chi^t} \phi_{m_0 pqr}^3(x, y, z) \\ \sigma(x, y, z) &= \sum_{pqr} c_{m_0 pqr}^{\sigma} \phi_{m_0 pqr}^3(x, y, z) \end{aligned} \quad (25)$$

At this point we can apply the 3D wavelet transform to these two functions to obtain a multiresolution representation. Since we are interested in direct resolution control, we apply the non-standard decomposition [8] to obtain a representation at different levels of resolution. The resulting pyramidal decomposition scheme is illustrated in figure 2 for the source term; a similar representation can be built for the opacity term. Starting from the initial level  $m_0$ , we decompose the source and opacity coefficients up to level  $M$ .

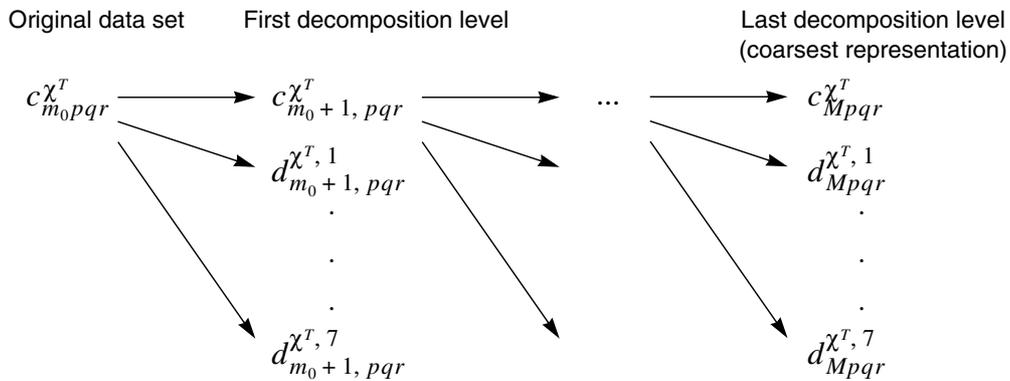


Figure 2: 3D wavelet transform to decompose the source samples.

By expressing these two terms as linear combinations of weighted basis functions, we obtain

$$\begin{aligned}\chi^t(x, y, z) &= \sum_{m=m_0+1}^M \sum_{type=1}^7 \sum_{p,q,r} d_{mpqr}^{\chi^t, type} \psi_{mpqr}^{3, type}(x, y, z) + \sum_{p,q,r} c_{Mpq}^{\chi^t} \phi_{Mpq}^3(x, y, z) \\ \sigma(x, y, z) &= \sum_{m=m_0+1}^M \sum_{type=1}^7 \sum_{p,q,r} d_{mpqr}^{\sigma, type} \psi_{mpqr}^{3, type}(x, y, z) + \sum_{p,q,r} c_{Mpq}^{\sigma} \phi_{Mpq}^3(x, y, z)\end{aligned}\quad (26)$$

Note that since source and opacity functions use the same basis functions, the coefficients that result from the wavelet transform are spatially aligned to each other. Furthermore, we can consider the low-pass approximations

$$\begin{aligned}\tilde{\chi}^t(x, y, z) &= \sum_{pqr} c_{Mpq}^{\chi^t} \phi_{Mpq}^3(x, y, z) \\ \tilde{\sigma}(x, y, z) &= \sum_{pqr} c_{Mpq}^{\sigma} \phi_{Mpq}^3(x, y, z)\end{aligned}\quad (27)$$

as the average source and opacity functions of clustered objects. Thus, these approximations can act as representatives of a group of neighboring objects in lighting calculations. The wavelet transform implies that the total number of clusters of the coarsest approximation level  $M$  equals  $(1/8)^{m_0-M}$  of the initial number of volume elements. Hence, we rapidly obtain small data sizes if we choose  $M$  to be large.

### 3 THE GLOBAL CUBE METHOD

This section describes the hardware-accelerated volume radiosity computation scheme. The concept of our global illumination computation scheme is founded on the traditional Galerkin radiosity method [34, 13] since it describes the energy distribution in terms of a functional system defined by the underlying wavelet functions.

#### 3.1 Concept

As mentioned in section 2.1 and section 2.2, the Neumann series and the Jacobi iteration essentially compute the emission of each volume element as the sum of its source energy and the amount of energy that is received from its surrounding. We can apply this theory and compute the interaction between any volume element  $i$  and the rest of the volume of interest. We keep computing these interactions until we converge to the solution of the volume radiosity problem.

We adopted this general concept to develop our hardware-assisted volume radiosity computation. Our algorithm works with a pure volumetric description of a scene, which is defined by a set of basic volume elements  $(V_p)_{p \in Scene}$  called *voxels*; no surface elements are used to describe opaque objects.

A voxel  $V_p$  is a small cube with associated constant source term  $\chi_{V_p}^t$  and constant opacity term  $\sigma_{V_p}$ . This information describes the initial status of the scene, before the interactions between volume elements are considered.

Next, we have to consider the interactions between voxels. Recall from equation (3) that the emission  $\chi_{V_p}$  of the volume element  $V_p$  equals the sum of the source term  $\chi_{V_p}^t$  and the emission gathered from all other volume elements in the scene, weighted by the scattering coefficient. The gathering process can be expressed as a projection operation: the emissions of all the voxels  $(V_q)_{q \in \text{Scene}, q \neq p}$  are projected onto the sides of the voxel  $V_p$ . This approach allows to rewrite equation (21) using the projection operator  $P_p(\chi_{V_q})$  as

$$\chi_{V_p}^{(i)} = \chi_{V_p}^t + \frac{\alpha^s(p, \mathbf{n})}{4\pi} \sum_{\substack{q=1 \\ q \neq p}}^n P_p(\chi_{V_q}^{(i-1)}) \quad (28)$$

The term  $P_p(\chi_{V_q}^{(i-1)})$  expresses the projection of the emission of the volume element  $V_q$  onto the volume element  $V_p$ . Note that since we are dealing with the voxel faces we have  $m_{pp} = 1$ , since we do not allow the faces of the voxel to interact with one another.

The core idea of the *global cube algorithm* is how to construct the projection operator  $P_p(\chi_{V_q})$ : each face of every voxel  $V_p$  in the scene is subdivided into cells (pixels) with a fixed orientation in space. Each of the voxel's faces is used as a projection plane. Therefore we can define a viewing frustum for each of the six faces, as shown in figure 3, by setting the center of projection in the center of the voxel  $V_p$ . Note that by definition, the viewing frustum has a viewing angle of  $90^\circ$ : everything outside of this angle is not projected onto the face of the voxel  $V_p$ .

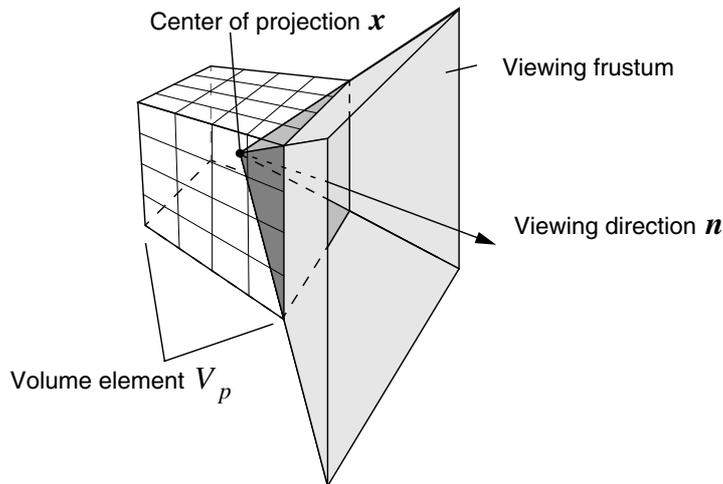


Figure 3: The global cube.

The projection itself is computed via the 3D texture memory low-albedo volume rendering scheme [4], which has the advantage of significantly increasing the computation speed. This is accomplished by slicing the volume at discrete positions and using alpha blending to accumulate the slices; slices are accumulated back-to-front: the first slice is the farthest from the voxel  $V_p$ , and additional slices are closer and closer to the projection plane. The slices are parallel to the projection plane, and the projec-

tion plane is perpendicular to one of the main axes  $x$ ,  $y$ , or  $z$ . As a result the slicing process can be computed very efficiently. Figure 4 illustrates a simple two-dimensional example of how the projection is computed.

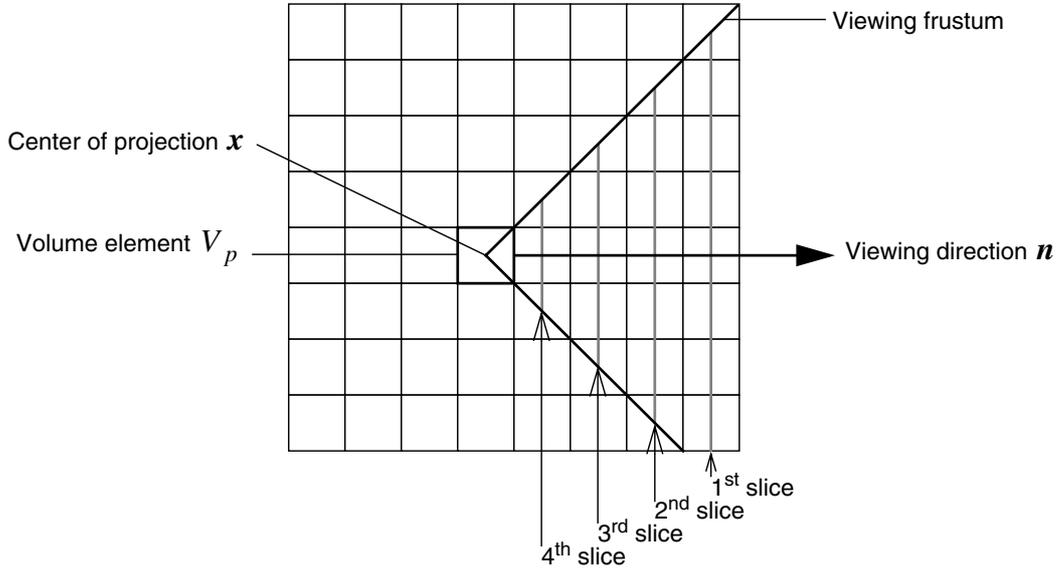


Figure 4: Projection computed with the 3D texture hardware

The result of this operation is a 2D field of intensities projected onto every side of voxel  $V_p$ . The correct energy transfer can then be computed from these 2D fields using the correction terms described in the next subsection. Furthermore, we can apply a physically correct model to the 2D fields to compute the energy transfer in the interior of the voxel.

### 3.2 Correction Terms

- *Distance correction*: The distance between points on the projection plane and the center of projection has to be corrected. Due to the fact that the emission reduces by  $r^2$ , where  $r$  denotes the distance from the emitting point, we have to apply a correction term for each pixel on the projection plane. This correction term can be computed for each pixel at  $(x_{pixel}, y_{pixel}, z_{pixel})$  (position relative to the center of projection) by

$$Corr_{dist}(x_{pixel}, y_{pixel}, z_{pixel}) = \frac{1}{x_{pixel}^2 + y_{pixel}^2 + z_{pixel}^2} \quad (29)$$

These correction values are stored in a lookup table.

- *Directional correction*: The second correction term addresses the problem of the directional dependencies between  $side_j$  of volume element  $V_q$  that “sends” energy and a “receiving”  $side_i$  of volume element  $V_p$  as illustrated in figure 5. Since we have chosen the slicing planes of the 3D texture to be parallel to the projection plane, we have to correct for this effect manually. We observe that the exchange of energy is proportional to area that is “seen”, e.g. we have

$$Corr_{dir}(side_i, side_j) = \cos(\theta_p(side_i, side_j)) \cos(\theta_q(side_i, side_j)) \quad (30)$$

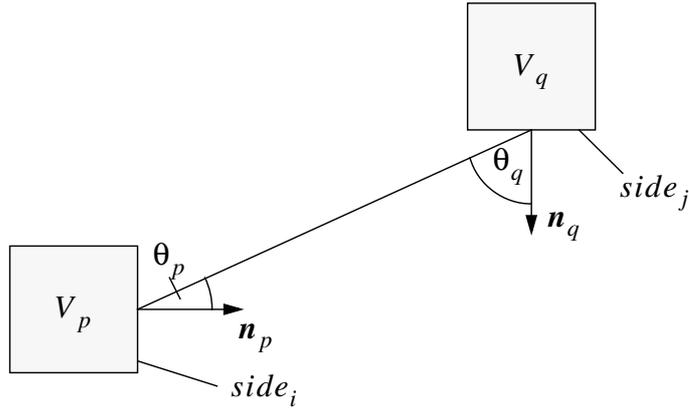


Figure 5: Directional correction.

- Positional correction:** Figure 6 shows the volume element  $V_p$  and its corresponding viewing frustum for the face  $side_i$ . We observe that the voxel  $V_q$  is not contained in the viewing frustum defined by  $side_i$  of voxel  $V_p$ , but since it sends energy towards  $side_i$  of the voxel  $V_p$  it has to be considered in the computation of the energy balance. This problem can easily be solved by a lookup table for  $side_j$ , as  $V_q$  appears within the viewing frustum of this face. Since the scene has to be rendered for  $side_j$  anyway, this additional energy contribution does not introduce any rendering overhead.

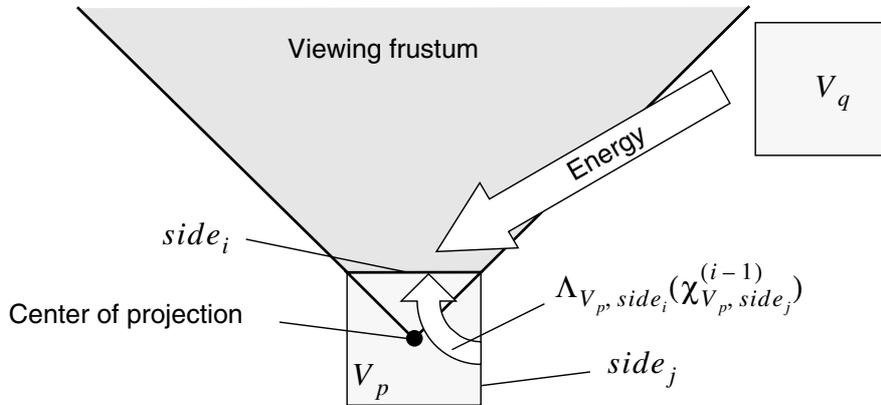


Figure 6: Energy contribution of a volume element that is not inside the viewing frustum.

- Scattering Correction:** In contrast to the distance and directional correction terms, the scattering correction does not adjust for an approximation error that occurs due to our computation scheme. It rather allows the consideration of directional in-scattering events and therefore extends the ability to simulate realistic reflectance behaviors. Recall that the BRDF  $\phi(\mathbf{x}, \omega', \omega)$  depends on the direction of the incident energy, the outgoing direction, and given material properties. For obvious reasons we want to avoid a fully transparent volume element to emit or reflect energy. Consequently we multiply this scattering function by its corresponding opacity coefficient  $\sigma(\mathbf{x})$ . We obtain a directional energy distribution over finite solid angles at pixel-resolution by

$$Corr_{scat}(V_p, x_{pixel}, y_{pixel}, z_{pixel}, side_i) = \frac{\varphi(V_p, x_{pixel}, y_{pixel}, z_{pixel}, \mathbf{n}_{side_i})}{4\pi} \cdot \sigma_{V_p} \quad (31)$$

which can be again stored in a lookup table.

- *Energy Exchange Correction:* This factor counts for the side-to-side transport of energy in the volume element. It also encompasses the absorption of the received energy within the volume element. This term can be formulated as

$$\Gamma(V_p, side_i, side_j) = \begin{cases} F_{side_j, side_i}(1 - \sigma_{V_p}) & , \text{ for } side_i \neq side_j \\ 1 & , \text{ otherwise} \end{cases} \quad (32)$$

where  $F_{side_j, side_i}$  describes the form factors as they were analytically given in [18, 25]. Note that the opacity  $\sigma_{V_p}$  of the volume element  $V_p$  equals one, the interior energy exchange factor equals zero. Thus, no energy flows from one face to another. This allows the generation of opaque objects and prevents leaking. The usage of the form factors guarantees that no additional energy is introduced in the scene and allows the computation of the energy exchange between two faces exactly.

Finally, the global cube algorithm can be written in pseudocode as

```
void GlobalCube (int radiosities[], int alpha[])
{
  while (radiosity solution not converged) {
    for all voxels p {
      for all side_p of voxel p // projection planes
        for all voxel q
          for all side_q of voxel q { // emitting directions of the scene
            compute framebuf[side_p] = project (scene_{side_q} onto side_p);
            for all pixel of framebuf
              newChiextern[side] += framebuf[side_p][pixel]
                * distance correction[pixel]
                * directional correction[side_p, side_q]
                * scattering correction[p, side_p, side_q]
          }
        for all side_q of voxel p
          for all side_p of voxel p
            newChiintern[side_q] += newChiextern[side_p]
              * positional correction[side_q, side_p]
              * Gamma(p, side_p, side_q)];
        Chi[p][side_p] += Chi_t[p] + newChiextern[side_p] + newChiintern[side_p]
      }
    }
  }
}
```

## 4 HIERARCHICAL GLOBAL CUBE

The wavelet transform generates a hierarchical representation of the emission and opacity terms. This operation is performed as a preprocessing step which will decompose the scalar coefficients that define the emission and the opacity. As a result, the preprocessing generates lowpass approximations and detail informations at different scales. The following relaxation algorithm balances the trade-off between an accurate solution, which will result in a visually acceptable image and a coarse approximation of a cluster which reduces the computational costs. The local spatial refinement is controlled by an oracle.

### 4.1 Approximate Solution

Since the first step of our algorithm is to find a solution of the coarse approximation of the data set, we solve the global illumination problem by using the approximation functions  $\tilde{\chi}^t$  and  $\tilde{\sigma}$  which are defined in (27). The solution is found by applying the global cube algorithm to the low-pass coefficients  $c_{Mpq\tau}$ . The algorithm terminates if the maximum energy update is less than a given threshold. It becomes obvious, that the solution costs depend strongly on the number of elements and therefore on the depth  $M$  and the initial size of the data set. We will next describe a refinement algorithm which will cut down efficiently the overall computation costs.

### 4.2 Refinement

The energy distribution in the lowpass filtered data set that was found as an approximate solution has to be refined selectively in order to find a more accurate solution. The prediction method that controls the refinement is denoted as an *oracle*. It decides whether a volume cluster (single basis function of level  $m$ ) has to be subdivided into eight sub-clusters (eight basis functions of level  $m-1$ ) or not. This refinement is simply performed by a spatially localized wavelet reconstruction. The associated oracle is driven by two heuristic functions and has proven to work effectively.

The first *static heuristic function* only considers information that is obtained from the wavelet transform. The wavelet transform extracts detail information from the lowpass signal. The details are encoded in the scalar coefficients  $d_{mpq\tau}^{type}$ . In general, large detail coefficients occur if the corresponding region exhibits high frequency changes of the source term, e.g. if the region is not homogeneous. Since we are interested in detecting these regions, high magnitude detail coefficients of either emission or opacity will force the algorithm to adaptively refine the corresponding area.

The static heuristic function can be formulated in pseudocode as follows:

```
boolean refine_static(voxel p, float threshold_emission,
                    float threshold_opacity,
                    octree radiosities[], octree alpha[])
{
  for m = m0 + 1 to M do // decomposition levels
    for type = 1 to 7 // wavelet types
      for all wavelet(m, type)
        if ((support(wavelet)  $\subseteq$  support (voxel p)) and
            ((abs(wavelet_coefficient_emission) > threshold_emission) or
             (abs(wavelet_coefficient_opacity) > threshold_opacity)))
          return(TRUE);
  return (FALSE);
}
```

Again, we want to point to the fact that the wavelet transform is only performed as an initial step and is based on the emission values and the opacity coefficients. Thus, the previously computed wavelet coefficients do not change during the computation of the energy distribution and therefore do not take the actual energy distribution into account.

In addition to the static heuristic function we introduce a second type of importance function. We denote these functions as *dynamic heuristic functions*. They consider progression of the energy transfer in the environment by comparing the received energy of two faces of a single volume element. If the energy differs more than a defined threshold, the element has to be further refined. Note that this criteria only forces volume elements to be refined if they receive energy. Hence, this function operates progressively as the energy is distributed in the volume data and leaves low energy regions unchanged. In pseudocode this function can be defined as:

```
boolean refine_dynamic(voxel p, float threshold_Chi,
                      octree radiosities[], octree alpha[])
{
  for all side of voxel p
    for all sidep of voxel p
      if(abs(Chi[p][side]-Chi[p][sidep]) > threshold_Chi)
        return(TRUE)
  return (FALSE);
}
```

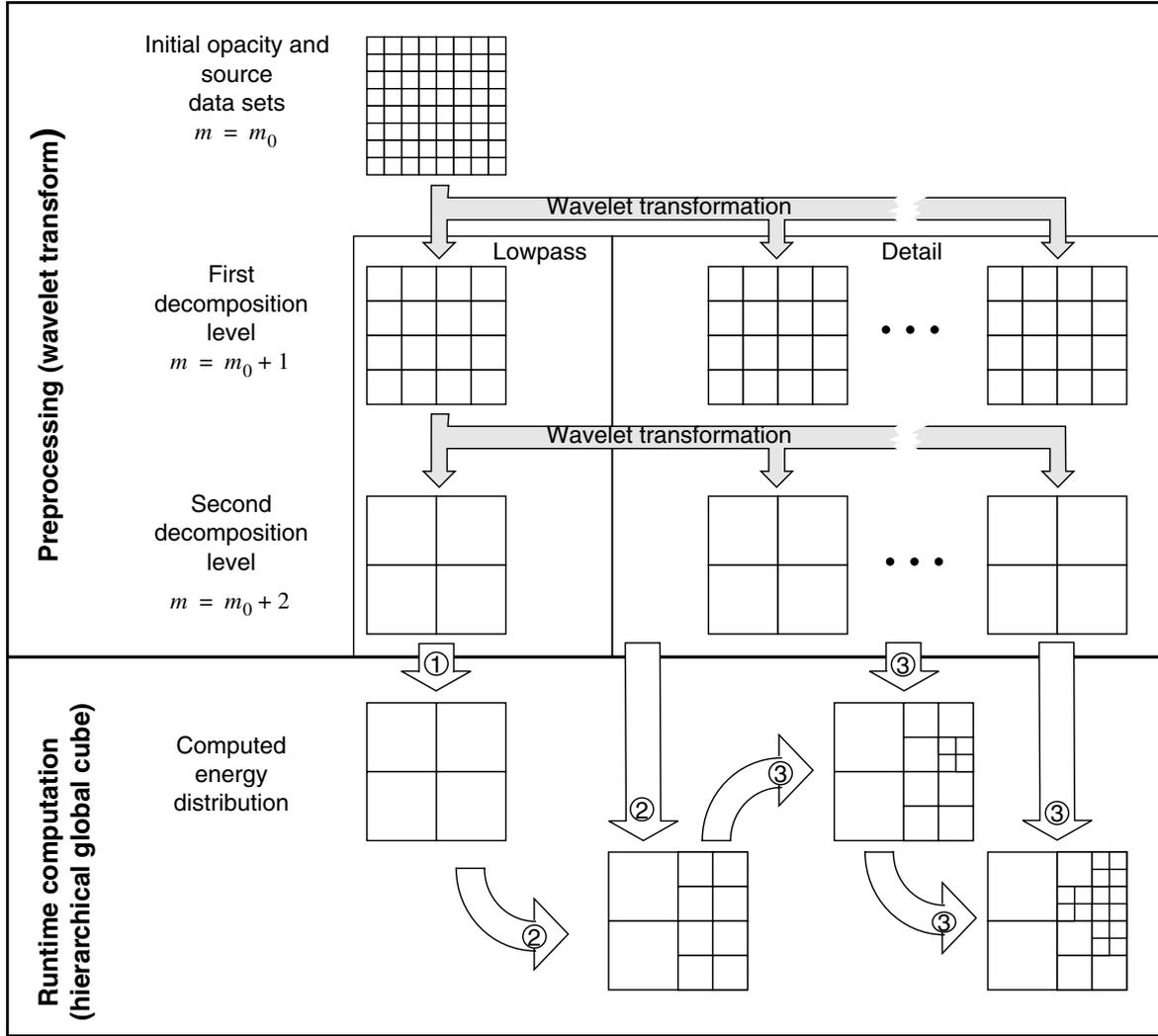
The conceptual components of the *hierarchical global cube algorithm* are illustrated in figure 7.

In the example of figure 7 the initial volumetric data sets ( $\chi^t$  and  $\sigma$ ) are represented as a single 2D grid. The preprocessing step computes a wavelet decomposition up to level  $m_0 + 2$ , whereas the hierarchical global cube algorithm is executed at runtime. The initial energy distribution computations only consider the lowpass information, whereas further refinement steps also take the detail information into account.

The accuracy of the global solution cannot be controlled directly since the correct energy distribution is not known in advance. In fact, enforcing a small error threshold in the reconstruction stage can obviously result in unnecessary fine structures, and does not even guarantee an accurate solution. Our implemented oracle has proven its usability for complex scenes. It also suppresses visual artifacts because it encourages refinement depending on visibility and extinction, since this corresponds to a shadow transition.

As mentioned earlier, the underlying concept of our approach refines the scene progressively. Thus, the previously computed solutions at decomposition level  $m$  will be used to calculate further refined versions at decomposition level  $m - 1$ . This coarse-to-fine strategy can thus be seen as an upward branch of a *V-cycle* in a multigrid scheme, that progressively solves the system. In terms of the multigrid method, the wavelet-driven reconstruction equals a prolongation operation that interpolates the correction to the finer grid [30].

For the refinement further considerations have to be made, as the refinement of a single voxel into its eight sub-voxels introduces new faces. This is depicted in figure 8. Therefore we suggest to initialize the emission value of the face side of the volume element  $V_r^{m-1}$  by a value that depends on the already computed emission value  $\chi_{V_p^m, side}$ , the opacity  $\sigma_{V_q^{m-1}}$ , and the source value  $\chi_{V_r^{m-1}}^t$ . The opacity and source values can be directly obtained from the inverse wavelet transform of the corresponding coefficients. As the adjacent volume element  $V_q^{m-1}$  shall be taken into account, we will also incorpo-



- ① Global cube algorithm applied to lowpass informations only.
- ② Refinement due to *static* and *dynamic* heuristic functions; computation of ener-
- ③ Refinement due to *dynamic* heuristic functions; computation of energy distribu-

Figure 7: Hierarchical global cube algorithm.

rate its weighted emission. The resulting term for the initial value can be formally seen as an *over* operation [29], which can be described as:

$$\chi_{V_r^{m-1}, side}^{(0)} = \chi_{V_r^{m-1}}^f + (1 - \sigma_{V_q}) \chi_{V_p^m, side} + \sigma_{V_q} \chi_{V_q^{m-1}}^f \quad (33)$$

Although this initial guess describes only an approximation of the energy distribution within the volume element, it turns out to be a good refinement strategy for the implementation. Furthermore, this term does not introduce additional energy to the system. Thus, the refinement ensures an energy balance between decomposition levels  $m$  and  $m-1$ . However, the exact emission value will be determined by the following iterative global cube procedure described in section 3. In contrast to the necessity for

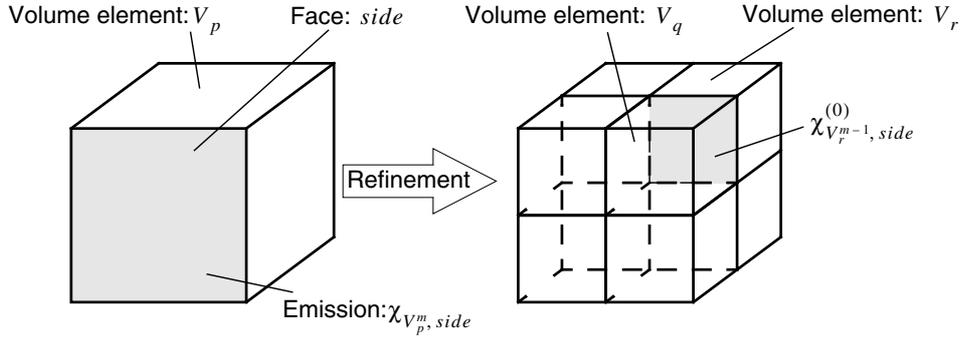


Figure 8: Propagation of radiosity in case of refinement.

the computation of intensity  $\chi_{V_r^{m-1}, side}^{(0)}$ , the emission  $\chi_{V_q^{m-1}, side}^{(0)}$  can be taken directly from the volume element  $V_p^m$  as we can set

$$\chi_{V_q^{m-1}, side}^{(0)} = \chi_{V_q^{m-1}}^t + \chi_{V_p^m, side} - \chi_{V_p^m}^t \quad (34)$$

## 5 RESULTS

In this section we will investigate the quality and accuracy of the *global cube algorithm* analyzing:

- quality of the approximations
- the convergence of the algorithm.
- the hierarchical solution of the radiosity problem.
- the speed of the method.

### Quality of the approximations

The global cube algorithm is applied on pure volumetric scenes: no 2D surfaces are used. This differs from the zonal method, where opaque objects are modeled as surfaces. The classic zonal method algorithm cannot be used with pure volumetric scenes, since opaque volume elements would “leak” energy collected on one side of the object onto the other side of the object. As a consequence it is not possible to compare the two algorithms directly. The quality of the approximations computed with the global cube algorithm can be quantified as follow:

- We guaranteed the energy balance for each operation performed on the energy distribution.
- We will show that the algorithm converges, if larger frame buffers are used in the projection step or if more voxels are refined during the hierarchical solution of the problem.
- We will show the radiosity distribution computed for two large scenes.

## Convergence

The first important result we show is the rate of convergence of the algorithm. Classic form factor based radiosity methods are very stable, and therefore iterative solvers converge to the solution fast. Since the global cube algorithm solve the same stable problem, we would expect it to converge as well.

The errors between different solutions have been computed using the two norm defined in (35)

$$Q_{square} = \frac{\sqrt{\sum_{p \in Scene} (\chi_{V_p}^{solution_1} - \chi_{V_p}^{solution_2})^2}}{|size(Scene)|} \quad (35)$$

We show the convergence of the algorithm by inspecting two parameters:

- The size of the *frame buffer* used during the projection. Larger frame buffers result in more accurate projection, since more information can be captured and aliasing effects are less noticeable, whereas small frame buffers result in faster computations.

As the size of the frame buffer increases towards infinity, we expect the projection operation to be computed more and more accurately, since the discretization is finer and finer. This type of convergence is in fact fulfilled by the global cube algorithm, as shown in figure 9:

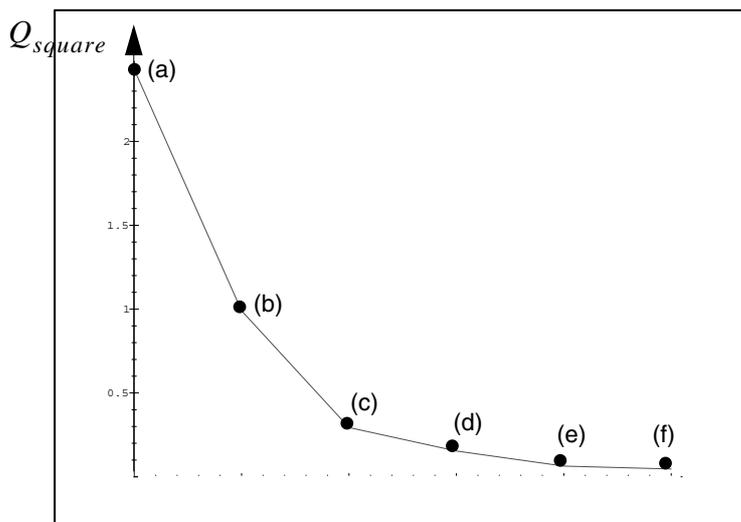


Figure 9: Convergence analysis: error between the reference solution computed with a frame buffer of size 128x128 and the solution computed with a frame buffer of size (a) 2x2, (b) 4x4, (c) 8x8, (d) 16x16, (e) 32x32, (f) 64x64

- The threshold used by the oracles in the *hierarchical global cube*. A low threshold will refine more volume elements, thus resulting in accurate solutions of the radiosity problem, whereas high threshold values will only refine the volume when strictly needed, and it will accelerate the computations.

As the size of the threshold converges towards zero we expect the solution of the *hierarchical global cube* to converge to the solution of the *standard global cube algorithm*. This is in fact the case as shown in figure 10:

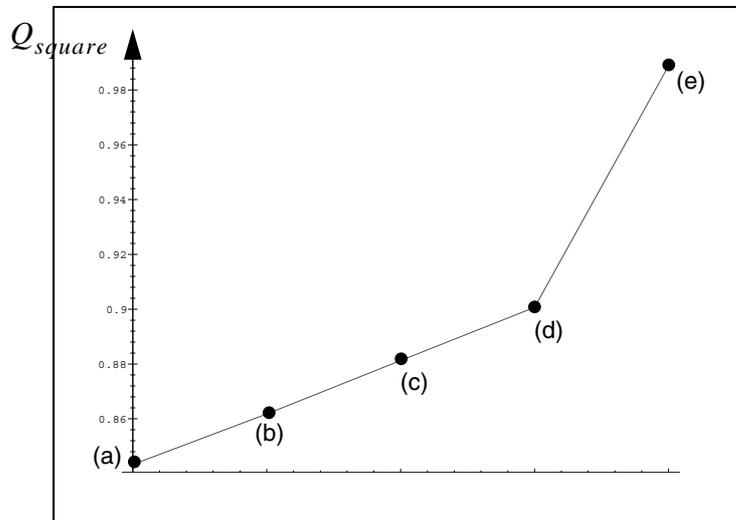


Figure 10: Convergence analysis: error between the reference solution computed with the standard global cube algorithm and the solution computed with the hierarchical global cube using a threshold of (a) 13, (b) 33, (c) 53, (d) 73, (e) 93

### Radiosity Distribution for a Large Scene

In this section we present two complex scenes that have been computed using the hierarchical global cube algorithm. The first scene we present consists of two sofas, a semi-transparent table, a light source, a book on the table and participating media; the size of the discretized volume is  $64 \times 64 \times 64$ . Fig. 11 illustrates the radiosity distribution for this scene. This scene will be used further in this section to illustrate the performance of the global cube algorithm and to show how the hierarchical global cube can significantly speed up the computation.

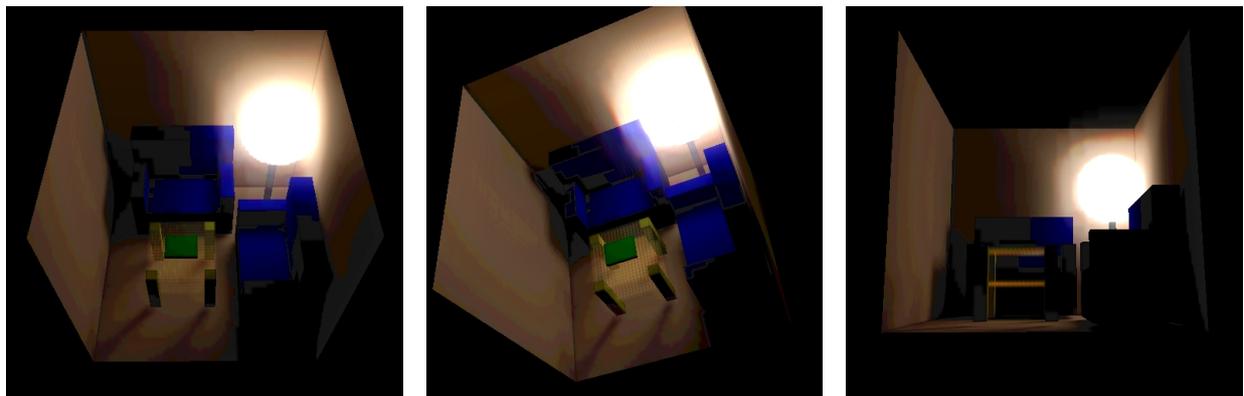


Figure 11: Solution for the first complex environment (3 x 262144 volume elements).

The second scene shows the entrance of a house. Three lights illuminate the scene: one square light on the ceiling, and two smaller lights on the side of the door hidden by two red panels; the size of the discretized volume is  $64 \times 64 \times 64$ . The radiosity distribution of the scene is shown in Fig. 12:



Figure 12: Solution for the second complex environment (3 x 262144 volume elements).

### Hierarchical Solution

In this subsection we show how the *hierarchical global cube* approximates the radiosity distribution in the scene. We use the same scene we showed in figure 11 to illustrate the radiosity distribution in a complex environment. The resolution of the modeled scene was chosen to be  $64^3$  voxels for R,G, and B and a two level decomposition was computed ( $M = m_0 + 2$ ). For reasons of optimization, the resolution of the projection plane adapts to the refinement step  $m$  of the hierarchy. For  $m = m_0 + 2$  the resolution equals  $16 \times 16$ , whereas for  $m = m_0 + 1$  and  $m = m_0$  the resolution is chosen as  $32 \times 32$  and  $64 \times 64$  respectively. The achieved solution has already been illustrated in figure 11 for different camera positions and viewing angles. We can observe very clearly direct and indirect illumination effects and shading. The hierarchical decompositions of the scene and the progressive computation of the energy distribution are depicted in figure 13. This figure shows the distribution of the red component of the lounge scene for different levels of the resolution and iteration steps.

The wavelet-decomposition of the emission and the opacity informations lead to a lowpass representation, as it is shown in (a). Due to the averaging character of the wavelet transform, we note that objects that are defined as opaque in the initial resolution appear as half-transparent objects in coarser levels. Starting from the solution that is computed for the  $16^3$  data set ( $m = m_0 + 2$ ), the oracle selectively refines the scene in order to obtain smaller errors and less visual artifacts. Image (b) displays the solution for the selectively refined data set at  $m = m_0 + 1$ . Images (c)-(f) finally illustrate the results of the

relaxation and oracle-controlled refinement steps of the global cube algorithm at the iteration level  $m = m_0$ .

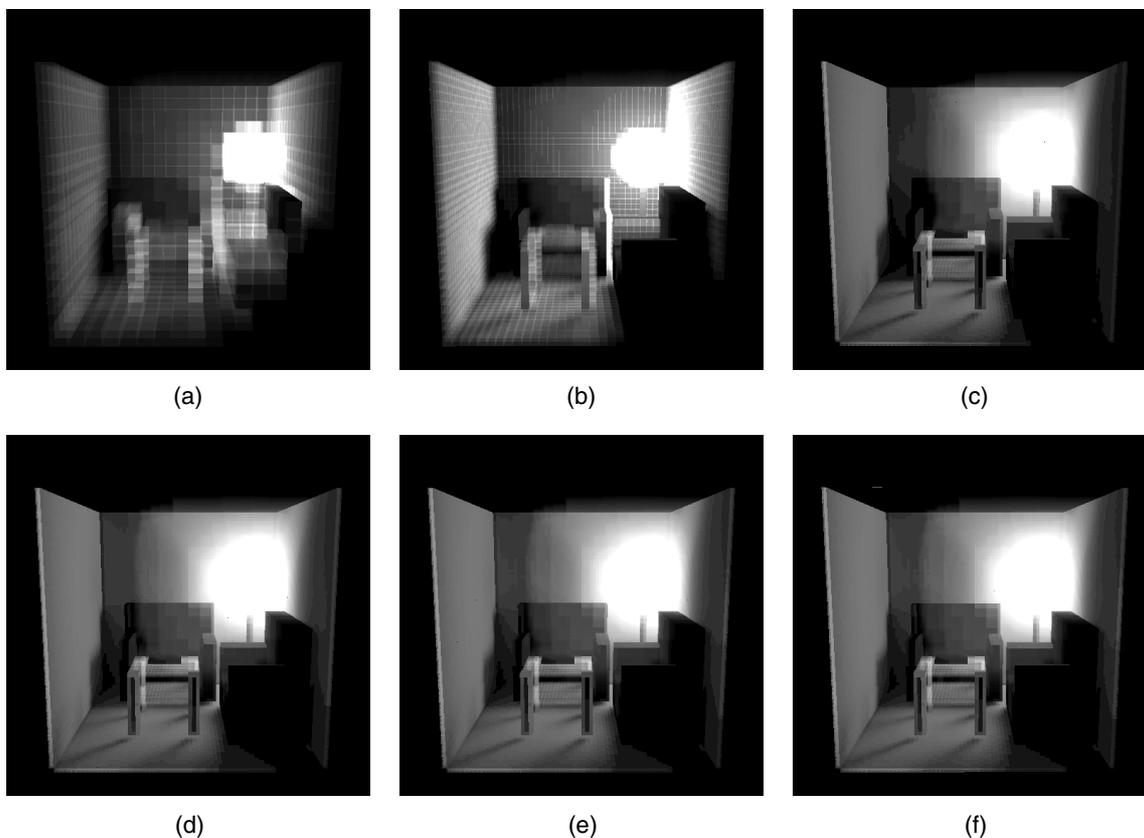


Figure 13: Progressive refinement of lounge scene: (a) solution for 163 volume elements ( $m=m_0+2$ ), (b) solution for reconstruction level  $m=m_0+1$ , (c) solution for the first iteration of the relaxation at  $m=m_0$ , (d) solution for the second iteration of the relaxation at  $m=m_0$ , (e) solution for the third iteration of the relaxation at  $m=m_0$ , (f) solution for the fourth iteration of the relaxation at  $m=m_0$ .

In order to analyze and visualize the influence of the dynamic heuristic function that controls the refinement oracle, we display a slice through the data and its corresponding hierarchy. The slice is taken near the back-wall. Figure 14 depicts the energy distribution and the hierarchy as it is computed in each relaxation step. The images correspond to the results that are displayed in figure 13 (c)-(f). As it can be taken from figure 14 (a), the static heuristic function forces the oracle to initially refine the opaque boundary regions of the sofa. The second relaxation step, illustrated in (b), also provides finer structures near the highly energized regions that are illuminated from the light source. This refinement is controlled by the dynamic oracle function and hence proving again its validity. This function also

increases the size of this region due to the energy flow in the data, as can be seen in (c) and (d).

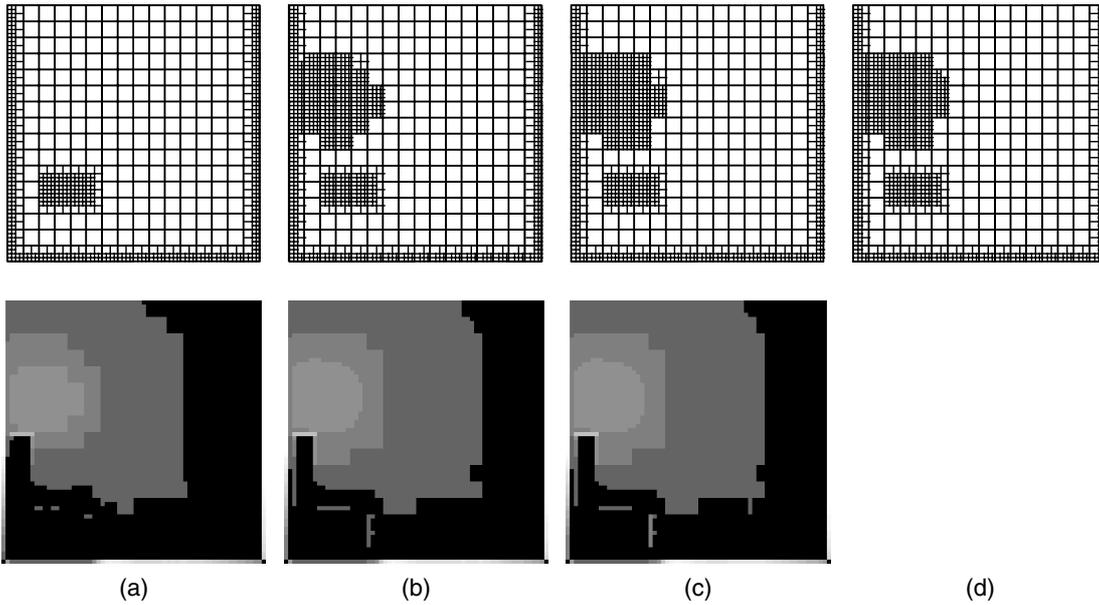


Figure 14: Hierarchy and intensity distribution of a slice through the lounge-scene: (a) solution for the first iteration of the relaxation at  $m=m_0$ , (b) solution for the second iteration of the relaxation at  $m=m_0$ , (c) solution for the third iteration of the relaxation at  $m=m_0$ , (d) solution for the fourth iteration of the relaxation at  $m=m_0$ .

### Computational time

In this last subsection we show the computational time needed to compute the radiosity distribution using the global cube. The time needed to compute the hierarchical solution of the “lounge” scene that we used in this section is presented in table 1:

Table 1: Performance of the global cube algorithm.

relaxation steps	$m=m_0+2$	$m=m_0+1$	$m=m_0$
1	4096 elements 1 min. (figure 13 (a))	13133 elements 7 min.	46768 elements 1 h 30 min. (figure 13 (c))
2	./.	14652 elements 9 min.	64177 elements 2 h 11 min. (figure 13 (d))
3	./.	14799 elements 8 min.	65605 elements 2 h 12 min. (figure 13 (e))
4	./.	14813 elements 8 min.	65738 2 h 12 min. (figure 13 (f))
5	./.	14813 elements 8min (figure 13 (b))	

This clearly demonstrate the fidelity and performance of our algorithm and the oracle we implemented. From this table we observe that the total time for the computation of the red component is of 8h 46 min. We also observe that rendering time can be reduced dramatically for lower image quality. Furthermore,

we see that the number of iterations varies significantly depending on the selected threshold and the computed energy-distribution.

A final remark: we do not have a fully optimized volume radiosity packet that is based on zonal method, so we can not compare these two approaches, but Rushmeier documents a computation time of 3 cpu hours for 1164 surfaces and 2744 volume elements on a VAX 8700 [31], whereas our global cube implementation requires for the 4096 volume elements only 186 seconds on a SGI Maximum Impact R10000 and a fraction of memory.

## 6 CONCLUSIONS

We have introduced the progressive global cube algorithm, a new concept for volume radiosity computations in a unified multiresolution framework. Our approach solves the energy distribution for the pure volumetric model and allows the visualization of realistically rendered volumetric data. This new physical model was directly developed from the transport theory by assuming non-reflecting and non-emitting boundary surfaces. The considered volumetric interactions can be used to augment a scene with effects such as haze, fog or smoke. The hardware-supported global cube technique evaluates the transfer of energy for complex anisotropic data sets. Our basic approach is founded on a combination of the volumetric Galerkin-radiosity approach and a low-albedo volume rendering technique. These concepts were further extended to hierarchical, wavelet-based structures which allow the computation of the energy equilibrium to any specified precision. The hierarchical global cube concept represents a departure from existing global illumination approaches in that it does not require explicit exchange factor computations and relaxation methods.

In both wavelet theory and volume rendering, however, some problems still remain unsolved. In wavelet theory, all concepts still suffer from the non-existence of a smooth function that features a small compact support. The current trade-off between blocky structures and high computation costs should be addressed in further research. The second important observation is that the wavelet transform on scattered data [7] also supports efficient hierarchical volume rendering strategies. The resulting radial basis functions may also bring up new volume radiosity concepts. Furthermore, other oracles be explored. Of special interest within this context are new oracles, which count for the visual perception. These oracles may lead to a more efficient refinement strategy and error estimation of the reconstructed volume.

In a broader context, it can be speculated that the global cube approach can be significantly accelerated by parallelism. The present approach are well-suited to this type of hardware architecture and can be easily implemented on distributed memory, parallel computers.

## 7 REFERENCES

- [1] J. R. Arvo. *Analytic Methods for Simulated Light Transport*. Ph.D. thesis, December 1995.
- [2] J. R. Arvo and D. B. Kirk. "Particle Transport and Image Synthesis." In *Computer Graphics (ACM SIGGRAPH '90 Proceedings)*, volume 24, pages 63–66, August 1990.
- [3] P. Blasi, B. L. Saec, and C. Schlick. "An Importance Driven Monte-Carlo Solution to the Global Illumination Problem." In *Fifth Eurographics Workshop on Rendering*, pages 173–183, Darmstadt, Germany, June 1994.
- [4] B. Cabral, N. Cam, and J. Foran. "Accelerated volume rendering and tomographic reconstruction using texture mapping hardware." In A. Kaufman and W. Krueger, editors, *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, Oct. 1994. ISBN 0-89791-741-3.

- [5] C. Chui and E. Quak. “Wavelets on a bounded interval.” In D. Braess and L. L. Schumaker, editors, *Numerical Methods of Approximation Theory*, pages 1–24. Birkhäuser Verlag, Basel, 1992.
- [6] C. K. Chui. *An Introduction to Wavelets*, volume 1 of *Wavelet Analysis and its Applications*. Academic Press, Inc., 1992.
- [7] C.K. Chui and J.D. Ward and K. Jetter and J. Stoeckler. “Wavelets for analyzing scattered data: An unbounded operator approach.” *Appl. Comput. Harmon. Anal.*, 3(3):254–267, 1996.
- [8] I. Daubechies. *Ten Lectures on Wavelets*. CBMS-NSF regional conference series in applied mathematics, no.61, SIAM, 1992.
- [9] D. L. DiLaura. “On the Simplification of Radiative Transfer Calculations.” *Journal of the Illuminating Engineering Society*, 12(1):12–16, October 1982.
- [10] E. Quak and N. Weyrich. “Decomposition and reconstruction algorithms for spline wavelets on a bounded interval.” *Appl. Comput. Harmon. Anal.*, 1(3):217–231, 1994.
- [11] A. S. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann, San Francisco, CA, 1995.
- [12] G. H. Golub and C. F. V. Loan. *Matrix Computations*. Johns Hopkins Series in the Mathematical Sciences. The Johns Hopkins University Press, 3.rd. edition, 1996.
- [13] C. M. Goral, K. E. Torrance, D. P. Greenberg, and B. Battaile. “Modelling the Interaction of Light Between Diffuse Surfaces.” In *Computer Graphics (ACM SIGGRAPH '84 Proceedings)*, volume 18, pages 212–222, July 1984.
- [14] S. J. Gortler, P. Schroder, M. F. Cohen, and P. Hanrahan. “Wavelet radiosity.” In *Computer Graphics Proceedings, Annual Conference Series, 1993*, pages 221–230, 1993.
- [15] M. H. Gross, L. Lippert, R. Dittrich, and S. Häring. “Two methods for wavelet-based volume rendering.” *Computers & Graphics*, 21(2):237–252, Mar. 1997. ISSN 0097-8493.
- [16] A. Haar. “Zur Theorie der orthogonalen Funktionen-Systeme.” *Math. Ann.*, 69:331–371, 1910.
- [17] P. S. Heckbert. “Finite Element Methods for Radiosity.” In *ACM SIGGRAPH '93 Course Notes - Global Illumination*, chapter 5, pages 1–7. 1993.
- [18] J. R. Howell. *A Catalog of Radiation Configuration Factors*. McGraw Hill, New York, NY, 1982.
- [19] A. Keller. “A Quasi-Monte Carlo Algorithm for the Global Illumination Problem in the Radiosity Setting.” Technical Report 260/94, University of Kaiserslautern, 1994.
- [20] W. Krüger. “The application of transport theory to visualization of 3d scalar data fields.” *Computers in Physics*, July/August 1991.
- [21] L. Lippert. *Wavelet Based Volume Rendering*. PhD thesis, Federal Institute of Technology (ETH), Zürich, 1998.
- [22] L. Lippert and M. H. Gross. “Fast wavelet based volume rendering by accumulation of transparent texture maps.” *Computer Graphics Forum*, 14(3):431–444, Aug. 1995. Proceedings of Eurographics '95. ISSN 1067-7055.

- [23] L. Lippert, M. H. Gross, and C. Kurmann. “Compression domain volume rendering for distributed environments.” *Computer Graphics Forum*, 16(3):95–108, Aug. 1997. Proceedings of Eurographics ’97. ISSN 1067-7055.
- [24] S. G. Mallat. “A theory for multiresolution signal decomposition: The wavelet representation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.
- [25] Y. Meyer. “Ondelettes sur l’intervalle.” *Rev. Mat. Iberoamericana*, 7:115–133, 1992.
- [26] Y. Meyer. *Wavelets and Operators*. Cambridge University Press, 1992.
- [27] R. G. Mistrick. “The Finite Fourier Equivalent of the Finite Element Method of Lighting Analysis.” M.Sc. thesis, Department of Civil, Environmental and Architectural Engineering, University of Colorado at Boulder, CO, 1985.
- [28] S. N. Pattanaik and S. P. Mudur. “Computation of Global Illumination in a Participating Medium by Monte Carlo Simulation.” *The Journal of Visualization and Computer Animation*, 4(3):133–152, July - September 1993.
- [29] T. Porter and T. Duff. “Compositing digital images.” In H. Christiansen, editor, *Computer Graphics (SIGGRAPH ’84 Proceedings)*, volume 18, pages 253–259, July 1984.
- [30] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing (2nd ed.)*. Cambridge University Press, Cambridge, 1992. ISBN 0-521-43108-5.
- [31] H. E. Rushmeier and K. E. Torrance. “The Zonal Method for Calculating Light Intensities in the Presence of a Participating Medium.” In *Computer Graphics (ACM SIGGRAPH ’87 Proceedings)*, volume 21, pages 293–302, July 1987.
- [32] F. Sillion. “A Unified Hierarchical Algorithm for Global Illumination with Scattering Volumes and Object Clusters.” *IEEE Transactions on Visualization and Computer Graphics*, 1(3), September 1995.
- [33] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, Wellesley, MA, 1996.
- [34] H. R. Zatz. “Galerkin Radiosity: A Higher Order Solution Method for Global Illumination.” In *Computer Graphics Proceedings, Annual Conference Series, 1993 (ACM SIGGRAPH ’93 Proceedings)*, pages 213–220, 1993.