Spherical Wavelets: Efficiently Representing Functions on the Sphere

Peter Schröder[*†] **Wim Sweldens**[*‡]
University of South Carolina

Abstract

Wavelets have proven to be powerful bases for use in numerical analysis and signal processing. Their power lies in the fact that they only require a small number of coefficients to represent general functions and large data sets accurately. This allows compression and efficient computations. Classical constructions have been limited to simple domains such as intervals and rectangles. In this paper we present a wavelet construction for scalar functions defined on the sphere. We show how biorthogonal wavelets with custom properties can be constructed with the *lifting scheme*. The bases are extremely easy to implement and allow *fully adaptive subdivisions*. We give examples of functions defined on the sphere, such as topographic data, bidirectional reflection distribution functions, and illumination, and show how they can be efficiently represented with spherical wavelets.

CR Categories and Subject Descriptors: I.3.0 [Computer Graphics]: *General*; G.1.0 [Numerical Analysis]: *General – Numerical Algorithms*; G.1.1 *Interpolation – Smoothing*; G.1.2 *Approximation – Nonlinear Approximation.*

Additional Key Words and Phrases: wavelets, sphere.

1 Introduction

1.1 Wavelets

Over the last decade wavelets have become an exceedingly powerful and flexible tool for computations and data reduction. They offer both theoretical characterization of smoothness, insights into the structure of functions and operators, and practical numerical tools which lead to faster computational algorithms. Examples of their use in computer graphics include surface and volume illumination computations [16, 29], curve and surface modeling [17], and animation [18] among others. Given the high computational demands and the quest for speed in computer graphics, the increasing exploitation of wavelets comes as no surprise.

While computer graphics applications can benefit greatly from wavelets, these applications also provide new challenges to the underlying wavelet technology. One such challenge is the construction of wavelets on general domains as they appear in graphics applications.

Classically, wavelet constructions have been employed on infinite domains (such as the real line **R** and plane $\mathbf{R}^2$). Since most practical computations are confined to finite domains a number of boundary constructions have also been developed [5]. However, wavelet type constructions for more general manifolds have only recently been attempted and are still in their infancy.

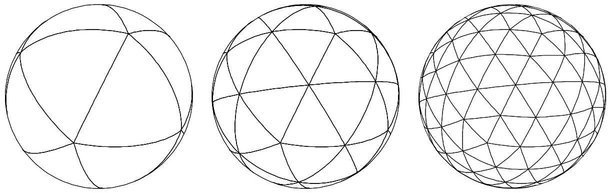

Figure 1: *The geodesic sphere construction starting with the icosahedron on the left (subdivision level 0) and the next 2 subdivision levels.*

Our work is inspired by the ground breaking work of Lounsbery *et al.*[20, 19] (hereafter referred to as LDW). While their primary goal was to efficiently represent *surfaces* themselves we examine the case of efficiently representing *functions defined on a surface*, and in particular the case of the sphere.

Although the sphere appears to be a simple manifold, techniques from $\mathbf{R}^2$ do not easily extend to the sphere. Wavelets are no exception. The first construction of wavelets on the sphere was introduced by Dahlke *et al.*[6] using a tensor product basis where one factor is an exponential spline. To our knowledge a computer implementation of this basis does not exist at this moment. A continuous wavelet transform and its semi-discretization were proposed in [13]. Both these approaches make use of a (φ, θ) parameterization of the sphere. This is the main difference with our method, which is parameterization independent.

Aside from being of theoretical interest, a wavelet construction for the sphere leading to efficient algorithms, has practical applications since many computational problems are naturally stated on the sphere. Examples from computer graphics include: manipulation and display of earth and planetary data such as topography and remote sensing imagery, simulation and modeling of bidirectional reflection distribution functions, illumination algorithms, and the modeling and processing of directional information such as environment maps and view spheres.

In this paper we describe a simple technique for constructing biorthogonal wavelets on the sphere with customized properties. The construction is an incidence of a fairly general scheme referred to as the *lifting scheme* [27, 28].

The outline of the paper is as follows. We first give a brief review of applications and previous work in computer graphics involving functions on the sphere. This is followed by a discussion of wavelets on the sphere. In Section 3 we explain the basic machinery of lifting and the fast wavelet transform. After a section on implementation, we report on simulations and conclude with a discussion and suggestions for further research.

1.2 Representing Functions on the Sphere

Geographical information systems have long had a need to represent sampled data on the sphere. A number of basic data structures originated here. Dutton [10] proposed the use of a geodesic sphere

[*] Department of Mathematics.

[†] Department of Computer Science.

[‡] Department of Computer Science, Katholieke Universiteit Leuven, Belgium.

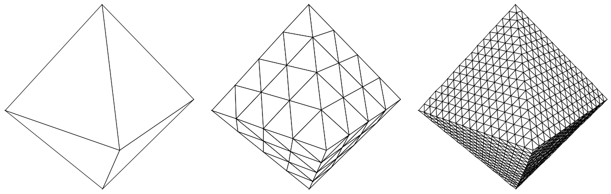

Figure 2: *Recursive subdivision of the octahedral base shape as used by LDW for spherelike surfaces. Level 0 is shown on the left followed by levels 2 and 4.*

construction to model planetary relief, see Figure 1 for a picture of the underlying subdivision. More recently, Fekete [12] described the use of such a structure for rendering and managing spherical geographic data. By using hierarchical subdivision data structures these workers naturally built sparse adaptive representations. There also exist many non-hierarchical interpolation methods on the sphere (for an overview see [22]).

An important example from computer graphics concerns the representation of functions defined over a set of directions. Perhaps the most notable in this category are bidirectional reflectance distribution functions (BRDFs) and radiance. The BRDF, $f_r(\vec{\omega}_i, \vec{x}, \vec{\omega}_o)$, describes the relationship at a point $\vec{x}$ on a surface between incoming radiance from direction $\vec{\omega}_i$ and outgoing radiance in direction $\vec{\omega}_o$. It can be described using spherical harmonics, the natural extension of Fourier basis functions to the sphere, see e.g. [30]. These basis functions are globally supported and suffer from some of the same difficulties as Fourier representations on the line such as ringing. To our knowledge, no fast (FFT like) algorithm is available for spherical harmonics. Westin *et al.* [30] used spherical harmonics to model BRDFs derived from Monte Carlo simulations of micro geometry. Noting some of the disadvantages of spherical harmonics, Gondek *et al.*[15] used a geodesic sphere subdivision construction [10, 12] in a similar context.

The result of illumination computations, the radiance $L(\vec{x}, \vec{\omega})$, is a function which is defined over all surfaces and all directions. For example, Sillion *et al.* [26] used spherical harmonics to model the directional distribution of radiance. As in the case of BRDF representations, the disadvantages of using spherical harmonics to represent radiance are due to the global support and high cost of evaluation. Similarly no locally controlled level of detail can be used.

In finite element based illuminations computations wavelets have proven to be powerful bases, see e.g. [24, 3]. By either reparameterizing directions over the set of visible surfaces [24], or mapping them to the unit square [3], wavelets defined on standard domains (rectangular patches) were used.

Mapping classical wavelets on some parameter domain onto the sphere by use of a parameterization provides one avenue to construct wavelets on the sphere. However, this approach suffers from distortions and difficulties due to the fact that no globally smooth parameterization of the sphere exists. The resulting wavelets are in some sense “contaminated” by the parameterization. We will examine the difficulties due to an underlying parameterization, as opposed to an *intrinsic* construction, when we discuss our construction.

We first give a simple example relating the compression of *surfaces* to the compression of functions defined *on surfaces.*

1.3 An Example

LDW constructs wavelets for surfaces of arbitrary topological type which are parameterized over a polyhedral base complex. For the case of the sphere they employed an octahedral subdivision domain (see Figure 2). In this framework a given goal surface such as the earth is parameterized over an octahedron whose triangular faces are successively subdivided into four smaller triangles. Each vertex can now be displaced radially to the limit surface. The resulting sequence of surfaces then represents the multiple levels of detail representation of the final surface.

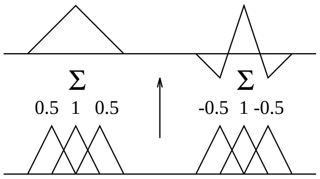

Figure 3: *A simple example of refinement on the line. The basis functions at the top can be expressed as linear combinations of the refined functions at the bottom.*

As pointed out by LDW compressing surfaces is closely related to compressing functions *on surfaces*. Consider the case of the unit sphere and the function $f(s) = f(\theta, \varphi) = \cos^2\theta$ with $s \in \mathrm{S}^2$. We can think of the graph of this function as a surface over the sphere whose height (displaced along the normal) is the value of the function f. Hence an algorithm which can compress surfaces can also compress the graph of a scalar function defined over some surface.

At this point the domain over which the compression is defined becomes crucial. Suppose we want to use the octahedron O. Define the projection $T : O \longrightarrow \mathrm{S}^2$, $s = T(p) = p/\|p\|$. We then have $\tilde{f}(p) = f(T(p))$ with $p \in O$. Compressing $f(s)$ with wavelets on the sphere is now equivalent to compressing $\tilde{f}(p)$ with wavelets defined on the octahedron. While f is simply a quadratic function over the sphere, $\tilde{f}$ is considerably more complicated. For example a basis over the sphere which can represent quadratics exactly (see Section 3.4) will trivially represent f. The same basis over the octahedron will only be able to *approximate* $\tilde{f}$.

This example shows the importance of incorporating the underlying surface correctly for any construction which attempts to efficiently represent functions defined *on that surface*. In case of compression of surfaces themselves one has to assume some canonical domain. In LDW this domain was taken to be a polyhedron.

By limiting our program to functions defined on a fixed surface (sphere) we can custom tailor the wavelets to it and get more efficiency. This is one of the main points in which we depart from the construction in LDW.

2 Wavelets on the Sphere

2.1 Second Generation Wavelets

Wavelets are basis functions which represent a given function at multiple levels of detail. Due to their local support in both space and frequency, they are suited for sparse approximations of functions. Locality in space follows from their compact support, while locality in frequency follows from their smoothness (decay towards high frequencies) and vanishing moments (decay towards low frequencies). Fast $O(n)$ algorithms exist to calculate wavelet coefficients, making the use of wavelets efficient for many computational problems.

In the classic wavelet setting, i.e., on the real line, wavelets are defined as the dyadic translates and dilates of one particular, fixed function. They are typically built with the aid of a *scaling function*. Scaling functions and wavelets both satisfy refinement relations (or two scale relations). This means that a scaling function or wavelet at a certain level of resolution (j) can be written as a linear combination of scaling basis functions *of the same shape but scaled* at one level finer (level $j + 1$), see Figure 3 for an example.

The basic philosophy behind *second generation wavelets* is to build wavelets with all desirable properties (localization, fast transform) adapted to much more general settings than the real line, e.g.,

Functions	
$\varphi_{j,k},\ k \in \mathcal{K}(j)$	primal scaling functions
$\tilde{\varphi}_{j,k},\ k \in \mathcal{K}(j)$	dual scaling functions
$\psi_{j,m},\ m \in \mathcal{M}(j)$	primal wavelets
$\tilde{\psi}_{j,m},\ m \in \mathcal{M}(j)$	dual wavelets
Biorthogonality relationships	
$\langle \varphi_{j,k}, \tilde{\varphi}_{j,k'} \rangle = \delta_{k,k'}$	$\varphi_{j,k}$ and $\tilde{\varphi}_{j,k'}$ are biorthogonal
$\langle \psi_{j,m}, \tilde{\psi}_{j',m'} \rangle = \delta_{m,m'} \delta_{j,j'}$	$\psi_{j,m}$ and $\tilde{\psi}_{j',m'}$ are biorthogonal
$\langle \varphi_{j,k}, \tilde{\psi}_{j,m} \rangle = 0$	$V_j \perp \tilde{W}_j$
$\langle \psi_{j,m}, \tilde{\varphi}_{j,k} \rangle = 0$	$W_j \perp \tilde{V}_j$
Vanishing moment relations	
$\tilde{\psi}_{j,m}$ has N vanishing moments	$\varphi_{j,k}$ reprod. polyn. degree $< N$
$\psi_{j,m}$ has $\widetilde{N}$ vanishing moments	$\tilde{\varphi}_{j,k}$ reprod. polyn. degree $< \widetilde{N}$
Refinement relations	
$\varphi_{j,k} = \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l}\, \varphi_{j+1,l}$	scaling function refinement eq.
$\tilde{\varphi}_{j,k} = \sum_{l \in \mathcal{K}(j+1)} \tilde{h}_{j,k,l}\, \tilde{\varphi}_{j+1,l}$	dual scaling function refinement eq.
$\psi_{j,m} = \sum_{l \in \mathcal{K}(j+1)} g_{j,m,l}\, \varphi_{j+1,l}$	wavelet refinement equation
$\tilde{\psi}_{j,m} = \sum_{l \in \mathcal{K}(j+1)} \tilde{g}_{j,m,l}\, \tilde{\varphi}_{j+1,l}$	dual wavelet refinement equation
$V_j = \text{clos span}\{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}$	with V_0 the coarsest space
$W_j = \text{clos span}\{\psi_{j,m} \mid m \in \mathcal{M}(j)\}$	with W_0 the coarsest space
$V_j \oplus W_j = V_{j+1}$	wavelets encode difference between levels of approximation
Wavelet transforms	
$\lambda_{j,k} = \langle f, \tilde{\varphi}_{j,k} \rangle$	scaling function coefficient
$\gamma_{j,m} = \langle f, \tilde{\psi}_{j,m} \rangle$	wavelet coefficient
Forward Wavelet Transform (Analysis)	
$\lambda_{j,k} = \sum_{l \in \mathcal{K}(j)} \tilde{h}_{j,k,l}\, \lambda_{j+1,l}$	scaling function coeff., fine to coarse
$\gamma_{j,m} = \sum_{l \in \mathcal{M}(j)} \tilde{g}_{j,m,l}\, \lambda_{j+1,l}$	wavelet coeff., fine to coarse
Inverse Wavelet Transform (Synthesis)	
$\lambda_{j+1,l} = \sum_{k \in \mathcal{K}(j)} h_{j,k,l}\, \lambda_{j,k}$ $+ \sum_{m \in \mathcal{M}(j)} g_{j,m,l}\, \gamma_{j,m}$	scaling function coeff., coarse to fine

Table 1: *Quick reference to the notation and some basic relationships for the case of second generation biorthogonal wavelets.*

wavelets on manifolds. In order to consider wavelets on a surface, we need a construction of wavelets which are *adapted to a measure* on the surface. In the case of the real line (and classical constructions) the measure is dx, the usual translation invariant (Haar) Lebesgue measure. For a sphere we will denote the usual area measure by $d\omega$. Adaptive constructions rely on the realization that translation and dilation are not fundamental to obtain the wavelets with the desired properties. The notion that a basis function can be written as a finite linear combination of basis functions at a finer, more subdivided level, is maintained and forms the key behind the fast transform. The main difference with the classical wavelets is that the filter coefficients of second generation wavelets are not the same throughout, but can change locally to reflect the changing (non translation invariant) nature of the surface and its measure.

Classical wavelets and the corresponding filters are constructed with the aid of the Fourier transform. The underlying reason is that translation and dilation become algebraic operations after Fourier transform. In the setting of second generation wavelets, translation and dilation can no longer be used, and the Fourier transform thus becomes worthless as a construction tool. An alternative construction is provided by the lifting scheme.

2.2 Multiresolution Analysis

We first introduce multiresolution analysis and wavelets and set some notation. For more mathematical detail the reader is referred to [9]. All relationships are summarized in Table 1.

Consider the function space $L_2 = L_2(S^2, d\omega)$, i.e., all functions of finite energy defined over S^2. We define a multiresolution analysis as a sequence of closed subspaces $V_j \subset L_2$, with $j \geqslant 0$, so that

I $V_j \subset V_{j+1}$, (finer spaces have higher index)

II $\bigcup_{j \geqslant 0} V_j$ is dense in L_2,

III for each j, scaling functions $\varphi_{j,k}$ with $k \in \mathcal{K}(j)$ exist so that $\{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}$ is a Riesz basis[1] of V_j.

Think of $\mathcal{K}(j)$ as a general index set where we assume that $\mathcal{K}(j) \subset \mathcal{K}(j+1)$. In the case of the real line we can take $\mathcal{K}(j) = 2^{-j}\mathbf{Z}$, while for an interval we might have $\mathcal{K}(j) = \{0, 2^{-j}, \ldots, 1 - 2^{-j}\}$. Note that, unlike the case of a classical multiresolution analysis, the scaling functions need not be translates or dilates of one particular function. Property (I) implies that for every scaling function $\varphi_{j,k}$, coefficients $\{h_{j,k,l}\}$ exist so that

$$\varphi_{j,k} = \textstyle\sum_l h_{j,k,l}\, \varphi_{j+1,l}. \tag{1}$$

The $h_{j,k,l}$ are defined for $j \geqslant 0$, $k \in \mathcal{K}(j)$, and $l \in \mathcal{K}(j+1)$. Each scaling function satisfies a different refinement relation. In the classical case we have $h_{j,k,l} = h_{l-2k}$, i.e., the sequences $h_{j,k,l}$ are independent of scale and position.

Each multiresolution analysis is accompanied by a dual multiresolution analysis consisting of nested spaces $\tilde{V}_j$ with bases given by dual scaling functions $\tilde{\varphi}_{j,k}$, which are biorthogonal to the scaling functions:

$$\langle \varphi_{j,k}, \tilde{\varphi}_{j,k'} \rangle = \delta_{k,k'} \text{ for } k, k' \in \mathcal{K}(j),$$

where $\langle f, g \rangle = \int f\, g\, d\omega$ is the inner product on the sphere. The dual scaling functions satisfy refinement relations with coefficients $\{\tilde{h}_{j,k,l}\}$.

In case scaling functions and dual scaling functions coincide, ($\varphi_{j,k} = \tilde{\varphi}_{j,k}$ for all j and k) the scaling functions form an *orthogonal* basis. In case the multiresolution analysis and the dual multiresolution analysis coincide ($V_j = \tilde{V}_j$ for all j but not necessarily $\varphi_{j,k} = \tilde{\varphi}_{j,k}$) the scaling functions are *semi-orthogonal*. Orthogonality or semi-orthogonality sometimes imply globally supported basis functions, which has obvious practical disadvantages. We will assume neither and always work in the most general biorthogonal setting (neither the multiresolution analyses nor the scaling functions coincide), introduced for classical wavelets in [4].

One of the crucial steps when building a multiresolution analysis is the construction of the wavelets. They encode the difference between two successive levels of representation, i.e., they form a basis for the spaces W_j where $V_j \oplus W_j = V_{j+1}$. Consider the set of functions $\{\psi_{j,m} \mid j \geqslant 0,\ m \in \mathcal{M}(j)\}$, where $\mathcal{M}(j) \subset \mathcal{K}(j+1)$ is again an index set. If

1. the set is a Riesz basis for $L_2(S^2)$,
2. the set $\{\psi_{j,m} \mid m \in \mathcal{M}(j)\}$ is the Riesz basis of W_j,

we say that the $\psi_{j,m}$ define a spherical wavelet basis. Since $W_j \subset V_{j+1}$, we have

$$\psi_{j,m} = \textstyle\sum_l g_{j,m,l}\, \varphi_{j+1,l} \text{ for } m \in \mathcal{M}(j). \tag{2}$$

An important property of wavelets is that they have vanishing moments. The wavelets $\psi_{j,m}$ have $\widetilde{N}$ vanishing moments if $\widetilde{N}$ independent polynomials P_i, $0 \leqslant i < \widetilde{N}$ exist so that

$$\langle \psi_{j,m}, P_i \rangle = 0,$$

for all $j \geqslant 0, m \in \mathcal{M}(j)$. Here the polynomials P_i are defined as the restriction to the sphere of polynomials on $\mathbf{R}^3$. Note that independent polynomials on $\mathbf{R}^3$ can become dependent after restriction to the sphere, e.g., $\{1, x^2, y^2, z^2\}$.

[1] A Riesz basis of some Hilbert space is a countable subset $\{f_k\}$ so that every element f of the space can be written uniquely as $f = \sum_k c_k\, f_k$, and positive constants A and B exist with $A\, \|f\|^2 \leqslant \sum_k |c_k|^2 \leqslant B\, \|f\|^2$.

For a given set of wavelets we have dual basis functions $\tilde{\psi}_{j,m}$ which are biorthogonal to the wavelets, or $\langle \psi_{j,m}, \tilde{\psi}_{j',m'} \rangle = \delta_{m,m'}\,\delta_{j,j'}$ for $j, j' \geqslant 0, m \in \mathcal{M}(j), m' \in \mathcal{M}(j')$. This implies $\langle \tilde{\psi}_{j,m}, \varphi_{j,k} \rangle = \langle \tilde{\varphi}_{j,k}, \psi_{j,m} \rangle = 0$ for $m \in \mathcal{M}(j)$ and $k \in \mathcal{K}(j)$, and for $f \in \mathrm{L}_2$ we can write the expansion

$$f = \textstyle\sum_{j,m} \langle \tilde{\psi}_{j,m}, f \rangle\, \psi_{j,m} = \sum_{j,m} \gamma_{j,m}\, \psi_{j,m} \qquad (3)$$

Given all of the above relationships we can also write the scaling functions $\varphi_{j+1,l}$ as a linear combination of *coarser* scaling functions and wavelets using the dual sequences (cf. Eqs. (1,2))

$$\varphi_{j+1,l} = \textstyle\sum_k \tilde{h}_{j,k,l}\, \varphi_{j,k} + \sum_m \tilde{g}_{j,m,l}\, \psi_{j,m}.$$

If not stated otherwise summation indices are understood to run over $k \in \mathcal{K}(j)$, $l \in \mathcal{K}(j+1)$, and $m \in \mathcal{M}(j)$.

Given the set of scaling function coefficients of a function f, $\{\lambda_{n,k} = \langle f, \tilde{\varphi}_{j,k} \rangle \mid k \in \mathcal{K}(n)\}$ where n is some finest resolution level, the fast wavelet transform recursively calculates the $\{\gamma_{j,m} \mid 0 \leqslant j < n,\ m \in \mathcal{M}(j)\}$, and $\{\lambda_{0,k} \mid k \in \mathcal{K}(0)\}$, i.e., the coarser approximations to the underlying function. One step in the fast wavelet transform computes the coefficients at a coarser level (j) from the coefficients at a finer level ($j+1$)

$$\lambda_{j,k} = \textstyle\sum_l \tilde{h}_{j,k,l}\, \lambda_{j+1,l} \ \text{ and } \ \gamma_{j,m} = \sum_l \tilde{g}_{j,m,l}\, \lambda_{j+1,l}.$$

A single step in the inverse transform takes the coefficients at the coarser levels and reconstructs coefficients at a finer level

$$\lambda_{j+1,l} = \textstyle\sum_k h_{j,k,l}\, \lambda_{j,k} + \sum_m g_{j,m,l}\, \gamma_{j,m}.$$

3 Wavelet Construction and Transform

We first discuss the *lifting scheme* [27, 28]. After the introducing of the algebra we consider two important families of wavelet bases, interpolating and generalized Haar. At the end of this section we give a concrete example which shows how the properties of a given wavelet basis can be improved by lifting it and lead to better compression.

Lifting allows us to build our bases in a fully biorthogonal framework. This ensures that all bases are of finite (and small) support and the resulting filters are small and easy to derive. As we will see it is also straightforward to incorporate custom constraints into the resulting wavelets.

3.1 The Lifting Scheme

The whole idea of the lifting scheme is to start from one basic multiresolution analysis, which can be simple or even trivial, and construct a new, more performant one, i.e., the basis functions are smoother or the wavelets have more vanishing moments. In case the basic filters are finite we will have lifted filters which are also finite.

We will denote coefficients of the original multiresolution analysis with an extra superscript ^{o} (from old or original), starting with the filters $h^o_{j,k,l}$, $\tilde{h}^o_{j,k,l}$, $g^o_{j,k,l}$, and $\tilde{g}^o_{j,k,l}$. The *lifting scheme* now states that a new set of filters can be found as

$$\begin{aligned} h_{j,k,l} &= h^o_{j,k,l}, & g_{j,m,l} &= g^o_{j,m,l} - \textstyle\sum_k s_{j,k,m}\, h_{j,k,l}, \\ \tilde{g}^o_{j,m,l} &= \tilde{g}_{j,m,l}, & \tilde{h}_{j,k,l} &= \tilde{h}^o_{j,k,l} + \textstyle\sum_m s_{j,k,m}\, \tilde{g}_{j,m,l}, \end{aligned}$$

and that, for any choice of $\{s_{j,k,m}\}$, the new filters will automatically be biorthogonal, and thus lead to an invertible transform. The scaling functions $\varphi_{j,l}$ are the same in the original and lifted multiresolution analysis, while the dual scaling function and primal wavelet change. They now satisfy refinement relations

$$\begin{aligned} \psi_{j,m} &= \textstyle\sum_l g^o_{j,m,l}\, \varphi_{j+1,l} - \sum_k s_{j,k,m}\, \varphi_{j,k} \qquad (4) \\ \tilde{\varphi}_{j,k} &= \textstyle\sum_l \tilde{h}^o_{j,k,l}\, \tilde{\varphi}_{j+1,l} + \sum_m s_{j,k,m}\, \tilde{\psi}_{j,m}. \end{aligned}$$

Note that the dual wavelet has also changed since it is a linear combination (with the old coefficients $\tilde{g}^o$) of a now changed dual scaling function. Equation (4) is the key to finding the $\{s_{j,k,m} \mid k\}$ coefficients. Since the scaling functions are the same as in the original multiresolution analysis, the only unknowns on the right hand side are the $s_{j,k,m}$. We can choose them freely to enforce some desired property on the wavelets $\psi_{j,m}$. For example, in case we want the wavelet to have vanishing moments, the condition that the integral of a wavelet multiplied with a certain polynomial P_i is zero can now be written as

$$0 = \textstyle\sum_l g^o_{j,m,l} \langle \varphi_{j+1,l}, P_i \rangle - \sum_k s_{j,k,m}\, \langle \varphi_{j,k}, P_i \rangle.$$

For a fixed j and m, this is a linear equation in the unknowns $\{s_{j,k,m} \mid k\}$. If we choose the number of unknown coefficients $s_{j,k,m}$ equal to the number of equations N, we need to solve a linear system for each j and m of size $N \times N$. A priori we do not know if this linear system can always be solved. We will come back to this later.

The fast wavelet transform after lifting can be written as

$$\begin{aligned} \gamma_{j,m} &= \textstyle\sum_l \tilde{g}^o_{j,m,l}\, \lambda_{j+1,l} \\ \lambda_{j,k} &= \textstyle\sum_l \tilde{h}^o_{j,k,l}\, \lambda_{j+1,l} + \sum_m s_{j,k,m}\, \gamma_{j,m}, \end{aligned}$$

i.e., as a sequence of two steps. First the old dual high and low pass filters. Next the update of the old scaling function coefficients with the wavelet coefficients using the $\{s_{j,k,m} \mid k\}$. The inverse transform becomes

$$\lambda_{j+1,l} = \textstyle\sum_k h^o_{j,k,l} \left(\lambda_{j,k} - \sum_m s_{j,k,m}\, \gamma_{j,m} \right) + \sum_m g^o_{j,m,l}\, \gamma_{j,m}.$$

Instead of writing everything as a sequence of two steps involving $\{s_{j,k,m} \mid k\}$ we could have formed the new filters h and $\tilde{g}$ first and then applied those in a single step. Structuring the new filters as two stages, however, simplifies the implementation considerably and is also more efficient.

Remarks:

1. The multiple index notation might look confusing at first sight, but its power lies in the fact that it immediately corresponds to the data structure of the implementation. The whole transform can also be written as one giant sparse matrix multiplication, but this would obscure the implementation ease of the lifting scheme.
2. Note how the inverse transform has a simple structure directly related to the forward transform. Essentially the inverse transform subtracts exactly the same linear combination of wavelet coefficients from $\lambda_{j,k}$ as was added in the forward transform.
3. It is also possible to keep the dual scaling function fixed and put the conditions on the dual wavelet. The machinery is exactly the same provided one switches primals and duals and thus toggles the tildes in the equations. We refer to this as the *dual lifting scheme*, which employs coefficients $\tilde{s}_{j,k,m}$. It allows us to improve the performance of the dual wavelet. Typically, the number of vanishing moments of the dual wavelet is important to achieve compression. Also, the lifting scheme and the dual lifting scheme can be alternated to bootstrap one's way up to a desired multiresolution analysis (cakewalk construction).
4. The construction in LDW can be seen as a special case of the lifting scheme. They use the degrees of freedom to achieve pseudo-orthogonality (i.e., orthogonality between scaling function and wavelets of one level within a small neighborhood) starting from an interpolating wavelet. The lifting scheme is more general in the sense that it uses a fully biorthogonal setting and that it can start from any multiresolution analysis with finite filters. The pseudo-orthogonalization requires the solution of linear systems which are of the size of the neighborhood (typically 24 by 24). Since many wavelets may in fact be the same caching of matrix computations is possible.
5. After finishing this work, the authors learned that a similar construction was obtained independently by Dahmen and collaborators. We refer to the original papers [2, 8] for details.

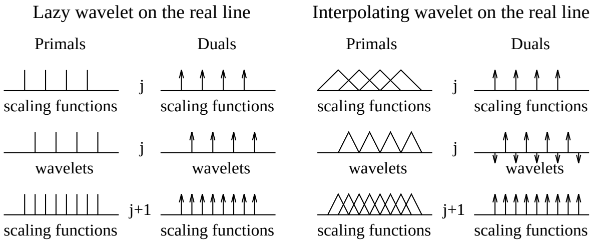

Figure 4: *For the Lazy wavelet all primals are Kronecker functions (1 at the origin, 0 otherwise), while all duals are unit pulses (Dirac distributions). Going from a finer to a coarser scale is achieved by subsampling with the missing samples giving the wavelet spaces* ($\mathcal{K}(j) = 2^{-j}\mathbf{Z}$, $\mathcal{M}(j) = 2^{-j}(\mathbf{Z} + 1/2)$, *and* $x_{j,k} = k$). *The well known linear B-splines as primal scaling and wavelet functions with Diracs as duals can be reached with dual lifting* ($\tilde{s}_{j,k,m} = 1/2\,\delta_{k-2^{-j-1},m} + 1/2\,\delta_{k+2^{-j-1},m}$)*, resulting in* $\tilde{\varphi}_{j,k} = \delta(\cdot - k)$ *and* $\tilde{\psi}_{j,m} = -1/2\,\delta(\cdot - m - 2^{-j-1}) + \delta(\cdot - m) - 1/2\,\delta(\cdot - m + 2^{-j-1})$.

6. Evidently, the lifting scheme is only useful in case one has an initial set of biorthogonal filters. In the following sections we will discuss two such sets.

3.2 Fast Lifted Wavelet Transform

Before describing the particulars of our bases we give the general structure of all transforms. Forward (analysis) and inverse (synthesis) transforms are always performed level wise. The former begins at the finest level and goes to the root while the latter starts at the root and descends to the leaf level. `AnalysisI` computes the unlifted wavelet coefficients at the parent level while `AnalysisII` performs the lifting if the basis is lifted otherwise it is empty. Similarly, `SynthesisI` performs the inverse lifting, if any, while `SynthesisII` computes the scaling function coefficients at the child level.

```
Analysis
    For level = leaflevel to rootlevel
        AnalysisI(level)
        AnalysisII(level)
Synthesis
    For level = rootlevel to leaflevel
        SynthesisI(level)
        SynthesisII(level)
```

The transforms come in two major groups: (A) Lifted from the Lazy wavelet: this involves interpolating scaling functions and a vertex based transform; (B) Lifted from the Haar wavelet: this involves a face based transform. We next discuss these in detail.

3.3 Interpolating Scaling Functions

We first give a trivial example of a wavelet transform: the Lazy wavelet [27, 28]. The Lazy wavelet transform is an orthogonal transform that essentially does not compute anything. However, it is fundamental as it is connected with interpolating scaling functions. The filters of the Lazy fast wavelet transform are given as

$$h^o_{j,k,l} = \tilde{h}^o_{j,k,l} = \delta_{k,l} \text{ and } g^o_{j,m,l} = \tilde{g}^o_{j,m,l} = \delta_{m,l}.$$

Consequently, the transform does not compute anything, it only subsamples the coefficients. Figure 4 (left) illustrates this idea for the case of the real line.

Scaling functions $\{\varphi_{j,k} \mid j \geqslant 0,\, k \in \mathcal{K}(j)\}$ are called *interpolating* if a set of points $\{x_{j,k} \mid j \geqslant 0, k \in \mathcal{K}(j)\}$ with $x_{j,k} = x_{j+1,k}$ exists, so that

$$\forall\, k, k' \in \mathcal{K}(j): \quad \varphi_{j,k}(x_{j,k'}) = \delta_{k,k'}.$$

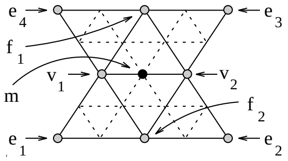

Figure 5: *Neighbors used in our bases. Members of the index sets used in the transforms are shown in the diagram* ($m \in \mathcal{M}(j)$, $\{v_1, v_2, f_1, f_2, e_1, e_2, e_3, e_4\} = \mathcal{K}_m$)*.*

An example for such functions on the real line is shown on the right side of Figure 4. In case of interpolating scaling functions, we can always take the dual scaling functions to be Dirac distributions, $\tilde{\varphi}_{j,k}(x) = \delta(x - x_{j,k})$, which are immediately biorthogonal (see the dual scaling functions on the right of Figure 4). This leads to trivial inner products with the duals, namely evaluation of the function at the points $x_{j,k}$.

The set of filters resulting from interpolating scaling functions and Diracs as their formal dual, can be seen as a *dual lifting of the Lazy wavelet*. This implies that $h_{j,k,k'} = \delta_{k,k'}$, $h_{j,k,m} = \tilde{s}_{j,k,m}$, $\tilde{g}_{j,m,k} = -\tilde{s}_{j,k,m}$, $\tilde{g}_{j,m,m'} = \delta_{m,m'}$. The wavelets are given by $\psi_{j,m} = \varphi_{j+1,m}$ and the dual wavelets by

$$\tilde{\psi}_{j,m} = \delta(\cdot - x_{j+1,m}) - \textstyle\sum_k \tilde{s}_{j,k,m}\,\delta(\cdot - x_{j,k}).$$

The linear B-spline (right side of Figure 4) can be seen to be the dual lifting of the Lazy wavelet. Since we applied dual lifting the primal wavelet does not yet have a vanishing moment.

Below we present other choices for the filter coefficients $h_{j,k,m}$. Typically one can choose the $\tilde{s}_{j,k,m}$ to insure that $\tilde{\psi}_{j,m}$ has vanishing moments (this will lead to the Quadratic scheme), or that $\varphi_{j,k}$ is smooth (this will lead to the Butterfly scheme).

At this point we have an interpolating multiresolution analysis, which was dually lifted from the Lazy wavelet. A disadvantage of this multiresolution analysis is that the functions cannot provide Riesz bases for L_2. The dual functions do not even belong to L_2. This is related to the fact that the wavelet does not have a vanishing integral since it coincides with a scaling function. Consequently, unconditional convergence of the expansion (3) is not guaranteed. One can now apply the primal lifting scheme to try to overcome this drawback by ensuring that the primal wavelet has at least 1 vanishing moment. Note that this is only a necessary and not a sufficient condition. This yields

$$\begin{aligned} \tilde{h}_{j,k,l} &= \delta_{k,l} + \textstyle\sum_m s_{j,k,m}\,\tilde{g}_{j,m,l} \\ g_{j,m,l} &= \delta_{m,l} - \textstyle\sum_k s_{j,k,m}\,h_{j,k,l}. \end{aligned}$$

The resulting wavelet can be written as

$$\psi_{j,m} = \varphi_{j+1,m} - \textstyle\sum_k s_{j,k,m}\,\varphi_{j,k}. \tag{5}$$

In the situation in Figure 4 setting $s_{j,k,m} = 1/4\,\delta_{m,k+2^{-j-1}} + 1/4\,\delta_{m,k-2^{-j-1}}$ results in $\psi_{j,m}$ having a vanishing integral. This choice leads us to the well known $(2,2)$ biorthogonal wavelet of [4].

3.4 Vertex Bases

Up to this point we have treated all index sets involved in the various filters as abstract sets. We now make these index sets more concrete. In order to facilitate our description we consider all index sets as defined locally around a given site $x_{j+1,m}$. A diagram is given in Figure 5. The index of a given site is denoted $m \in \mathcal{M}(j)$ and all the neighboring vertices ($x_{j,k}$ with $k \in \mathcal{K}(j)$) needed in the transform have indices v, f, and e respectively. To give some more intuition to these index sets recall wavelets on the real line as in Figure 4. In that case the set $\mathcal{K}(0) \ni l$ would consist of all integers, while $\mathcal{M}(-1) \ni m$ would contain the odd and $\mathcal{K}(-1) \ni k$ the even integers. For vertex based schemes we may think of the sites $m \in \mathcal{M}(j)$ as always living on the midpoint of some parent edge (these being the “odd” indices), while the endpoints of a given edge form the “even” indices ($k \in \mathcal{K}(j)$), and their union $l \in \mathcal{K}(j) \cup \mathcal{M}(j) = \mathcal{K}(j+1)$ gives the set of all indices. For each m the filters only range over some small neighborhood. We will refer to the elements in these neighborhoods by a local naming scheme (see Figure 5), $k \in \mathcal{K}_m \subset \mathcal{K}(j)$. For example, the site m lies in between the elements of $\mathcal{K}_m = \{v_1, v_2\}$.

For all vertex bases the unlifted scaling coefficients are simply subsampled during analysis and upsampled during synthesis, while the wavelet coefficients involve some computation.

```
AnalysisI(j):
  $\forall\, k \in \mathcal{K}(j)$ :   $\lambda_{j,k} := \lambda_{j+1,k}$
  $\forall\, m \in \mathcal{M}(j)$ :   $\gamma_{j,m} := \lambda_{j+1,m} - \sum_{k \in \mathcal{K}_m} \tilde{s}_{j,k,m}\, \lambda_{j,k}$
SynthesisII(j):
  $\forall\, k \in \mathcal{K}(j)$ :   $\lambda_{j+1,k} := \lambda_{j,k}$
  $\forall\, m \in \mathcal{M}(j)$ :   $\lambda_{j+1,m} := \gamma_{j,m} + \sum_{k \in \mathcal{K}_m} \tilde{s}_{j,k,m}\, \lambda_{j,k}$
```

We now give the details of the wavelet coefficient computations.

Lazy: As mentioned above the Lazy wavelet does nothing but subsampling. The resulting analysis and synthesis steps then become

$$\gamma_{j,m} := \lambda_{j+1,m} \text{ and } \lambda_{j+1,m} := \gamma_{j,m}.$$

respectively. The corresponding stencil encompasses no neighbors, i.e., the sums over $\tilde{s}_{j,k,m}$ are empty.

Linear: This basic interpolatory form uses the stencil $k \in \mathcal{K} = \{v_1, v_2\}$ (see Figure 5) for analysis and synthesis

$$\begin{aligned} \gamma_{j,m} &:= \lambda_{j+1,m} - 1/2(\lambda_{j+1,v_1} + \lambda_{j+1,v_2}) \\ \lambda_{j+1,m} &:= \gamma_{j,m} + 1/2(\lambda_{j,v_1} + \lambda_{j,v_2}), \end{aligned}$$

respectively. Note that this stencil does properly account for the geometry provided that the m sites at level $j+1$ have equal geodetic distance from the $\{v_1, v_2\}$ sites on their parent edge. Here $\tilde{s}_{j,v_1,m} = \tilde{s}_{j,v_2,m} = 1/2$.

Quadratic: The stencil for this basis is given by $\mathcal{K}_m = \{v_1, v_2, f_1, f_2\}$ (see Figure 5) and exploits the degrees of freedom implied to kill the functions x^2, y^2, and z^2 (and by implication the constant function [1]). Using the coordinates of the neighbors of the involved sites a small linear system results

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ x^2_{j,v_1} & x^2_{j,v_2} & x^2_{j,f_1} & x^2_{j,f_2} \\ y^2_{j,v_1} & y^2_{j,v_2} & y^2_{j,f_1} & y^2_{j,f_2} \\ z^2_{j,v_1} & z^2_{j,v_2} & z^2_{j,f_1} & z^2_{j,f_2} \end{pmatrix} \begin{pmatrix} \tilde{s}_{j,v_1,m} \\ \tilde{s}_{j,v_2,m} \\ \tilde{s}_{j,f_1,m} \\ \tilde{s}_{j,f_2,m} \end{pmatrix} = \begin{pmatrix} 1 \\ x^2_{j+1,m} \\ y^2_{j+1,m} \\ z^2_{j+1,m} \end{pmatrix}$$

Since $x^2 + y^2 + z^2 = 1$ this system is singular (but solvable) and the answer is chosen so as to minimize the l_2 norm of the resulting filter coefficients. Note that this is an instance of dual lifting with effective filters $\tilde{s}_{j,k,m} = h_{j,k,m} = -\tilde{g}_{j,m,k}$.

Butterfly: This is the only basis which uses other than immediate neighbors (all the sites $\mathcal{K}_m$ denoted in Figure 5). Here $\tilde{s}_{v_1} = \tilde{s}_{v_2} = 1/2$, $\tilde{s}_{f_1} = \tilde{s}_{f_2} = 1/8$, and $\tilde{s}_{e_1} = \tilde{s}_{e_2} = \tilde{s}_{e_3} = \tilde{s}_{e_4} = -1/16$. It is inspired by a subdivision scheme of Dyn *et al.* [11] for the construction of smooth surfaces.

3.5 Lifting Vertex Bases

All of the above bases, Lazy, Linear, Quadratic, and Butterfly can be lifted. In this section we use lifting to assure that the wavelet has at least one vanishing moment. It does not improve the ability of the dual wavelet to annihilate more functions. Consequently the ability of the bases to compress is not increased, but smaller error results when using them for compression (see the example in Section 3.8 and the results in Section 5). We propose wavelets of the form

$$\psi_{j,m} = \varphi_{j+1,m} - s_{j,v_1,m}\, \varphi_{j,v_1} - s_{j,v_2,m}\, \varphi_{j,v_2}. \quad (6)$$

In words, we define the wavelet at the midpoint of an edge as a linear combination of the scaling function at the midpoint $(j+1, m)$ and two scaling functions on the *coarser level* at the two endpoints of the parent edge $(j, v_{1,2})$. The weights $s_{j,k,m}$ are chosen so that the resulting wavelet has a vanishing integral

$$s_{j,k,m} = I_{j+1,m}/2\, I_{j,k} \text{ with } I_{j,k} = \int_{S^2} \varphi_{j,k}\, d\omega\,.$$

During analysis lifting is a second phase (at each level j) after the $\gamma_{j,m}$ computation, while during synthesis it is a first step followed by the regular synthesis step (Linear, Quadratic, or Butterfly as given above). The simplicity of the expressions demonstrates the power of the lifting scheme. Any of the previous vertex basis wavelets can be lifted *with the same expression*. The integrals $I_{j,k}$ can be approximated on the finest level and then recursively computed on the coarser levels (using the refinement relations).

```
AnalysisII(j):
  $\forall\, m \in \mathcal{M}(j)$ :  $\begin{cases} \lambda_{j,v_1} \mathrel{+}= s_{j,v_1,m}\, \gamma_{j,m} \\ \lambda_{j,v_2} \mathrel{+}= s_{j,v_2,m}\, \gamma_{j,m} \end{cases}$
SynthesisI(j):
  $\forall\, m \in \mathcal{M}(j)$ :  $\begin{cases} \lambda_{j,v_1} \mathrel{-}= s_{j,v_1,m}\, \gamma_{j,m} \\ \lambda_{j,v_2} \mathrel{-}= s_{j,v_2,m}\, \gamma_{j,m} \end{cases}$
```

For the interpolating case in the previous section, the scaling function coefficients at each level are simply samples of the function to be expanded (inner products with the $\tilde{\varphi}_{n,k}$). In the lifted case the coefficients are defined as the inner product of the function to be expanded with the (new) dual scaling function. This dual scaling

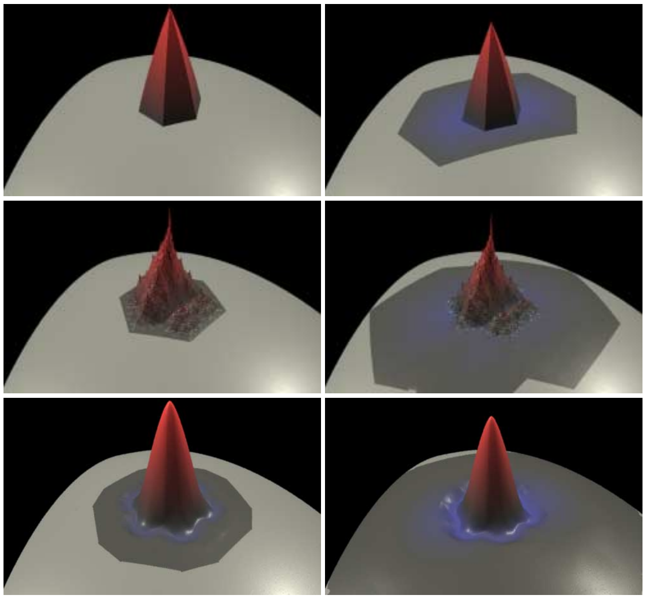

Figure 6: *Images of the graphs of all vertex based wavelets. On the left is the scaling function (or unlifted wavelet) while the right shows the lifted wavelet with 1 vanishing moment. From top to bottom: Linear, Quadratic, and Butterfly. Positive values are mapped to a linear red scale while negative values are shown in blue. The gray area shows the support.*

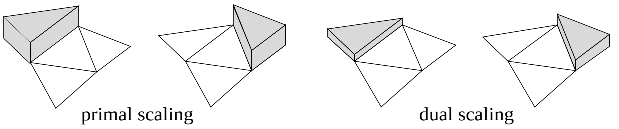

Figure 7: *Example Haar scaling functions on a triangular subdivision. On the left are primal functions each of height 1. On the right are the biorthogonal duals each of height $\alpha(T_i)^{-1}$. Disjoint bases have inner product of 0 while overlapping (coincident supports) lead to an inner product of 1. (For the sphere all triangles are spherical triangles.)*

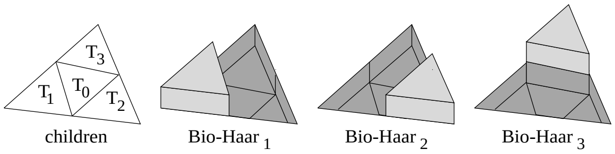

Figure 8: *The Bio-Haar wavelets. Note that the heights of the functions are not drawn to scale.*

function is only defined as the limit function of a non-stationary subdivision scheme. The inner products at the finest level therefore need to be approximated with a quadrature formula, i.e., a linear combination of function samples. In our implementation we use a simple one point quadrature formula at the finest level.

Figure 6 shows images of the graphs of all the vertex based ψ functions for the interpolating and lifted case.

3.6 The Generalized Haar Wavelets and Face Bases

Consider spherical triangles resulting from a geodesic sphere construction $T_{j,k} \subset \mathrm{S}^2$ with $k \in \mathcal{K}(j)$ (note that the face based $\mathcal{K}(j)$ are not identical to the vertex based $\mathcal{K}(j)$ defined earlier). They satisfy the following properties:

1. $\mathrm{S}^2 = \bigcup_{k\in\mathcal{K}(j)} T_{j,k}$ and this union is disjoint, i.e., the $T_{j,k}$ provide a simple cover of S^2 for every j,
2. for every j and k, $T_{j,k}$ can be written as the union of 4 "child" triangles $T_{j+1,l}$.

Let $\alpha(T_{j,k})$ be the spherical area of a triangle and define the scaling functions and dual scaling functions as

$$\varphi_{j,k} = \chi_{T_{j,k}} \text{ and } \tilde{\varphi}_{j,k} = \alpha(T_{j,k})^{-1}\chi_{T_{j,k}}.$$

Here χ_T is the function whose value is 1 for $x \in T$ and 0 otherwise. The fact that the scaling function and dual scaling function are biorthogonal follows immediately from their disjoint support (see Figure 7). Define the $V_j \subset \mathrm{L}_2$ as

$$V_j = \text{clos span}\{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}.$$

The spaces V_j then generate a multiresolution analysis of $\mathrm{L}_2(\mathrm{S}^2)$.

Now fix a triangle $T_{j,*}$. For the construction of the generalized Haar wavelets, we only need to consider the set of children $T_{j+1,l=0,1,2,3}$ of $T_{j,*}$. We call these bases the Bio-Haar functions (see Figure 8). The wavelets ($m = 1, 2, 3$) are chosen as

$$\psi_{j,m} = 2(\varphi_{j+1,m} - I_{j+1,m}/I_{j+1,0}\ \varphi_{j+1,0}),$$

so that their integral vanishes. A set of semi-orthogonal dual wavelets is then given by

$$\tilde{\psi}_{j,m} = 1/2(\tilde{\varphi}_{j+1,m} - \tilde{\varphi}_{j,*}).$$

These bases are inspired by the construction of *orthogonal* Haar wavelets for general measures, see [14, 21] where it is shown that the Haar wavelets form an unconditional basis.

The Bio-Haar wavelets have only 1 vanishing moment, but using the dual lifting scheme, we can build a new multiresolution analysis, in which the dual wavelet has more vanishing moments. Let $T_{j,k=4,5,6}$ be the neighboring triangles of $T_{j,*}$ (at level j), and $\mathcal{K}_m = \{*, 4, 5, 6\}$. The new dual wavelets are

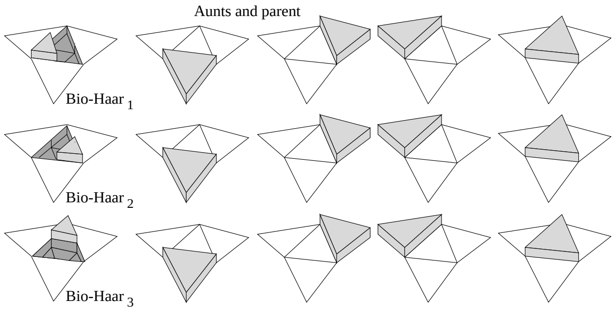

Figure 9: *Illustration of the dual lifting of the dual Bio-Haar wavelets. New dual wavelets can be constructed by taking linear combinations of the original dual Bio-Haar wavelets and parent level dual scaling functions. Each such linear combination is signified by a row. Solving for the necessary weights $\tilde{s}_{j,k,m}$ requires the solution to a small matrix problem whose right hand side encodes the desired constraints.*

$$\tilde{\psi}_{j,m} = 1/2(\tilde{\varphi}_{j+1,m} - \tilde{\varphi}_{j,*}) - \textstyle\sum_{k\in\mathcal{K}_m} \tilde{s}_{j,k,m}\,\tilde{\varphi}_{j,k}.$$

Note that this is a special case of Equation (4). The coefficients $\tilde{s}_{j,k,m}$ can now be chosen so that $\tilde{\psi}_{j,m}$ has vanishing moments. Figure 9 illustrates this idea. In the left column are the three dual Bio-Haar wavelets created before. The following four columns show the dual scaling functions over the parent and aunt triangles $T_{j,k=*,4,5,6}$. Each row signifies one of the linear combinations.

Similarly to the Quadratic vertex basis we construct dually lifted Bio-Haar wavelets which kill the functions x^2, y^2, z^2, and thus 1. This leads to the equations

$$\textstyle\sum_{k\in\mathcal{K}_m} \tilde{s}_{j,k,m}\langle\tilde{\varphi}_{j,k}, P\rangle = 1/2\langle\tilde{\varphi}_{j+1,m} - \tilde{\varphi}_{j,*}, P\rangle$$

with $P = x^2, y^2, z^2, 1$. The result is a 4×4 singular (but solvable) matrix problem for each $m = 1, 2, 3$. The unknowns are the $\tilde{s}_{j,k,m}$ with $k = *, 4, 5, 6$ and the entries of the linear system are moments of dual scaling functions. These can be computed recursively from the leaf level during analysis.

The Bio-Haar and lifted Bio-Haar transforms compute the scaling function coefficient during analysis at the parent triangle as a function of the scaling function coefficients at the children and possibly the scaling function coefficients at the neighbors of the parent triangle (in the lifted case). The three wavelet coefficients of the parent level are stored with the children T_1, T_2, and T_3 for convenience in the implementation. During synthesis the scaling function coefficient at the parent and the wavelet coefficients stored at children T_1, T_2, and T_3 are used to compute scaling function coefficients at the 4 children.

As before, lifting is a second step during analysis and modifies the wavelet coefficients. During synthesis lifting is a first step before the inverse Bio-Haar transform is calculated.

```
AnalysisII(j):
  $\forall\, m \in \mathcal{M}(j):$   $\gamma_{j,m}$  -=  $\sum_{k\in\mathcal{K}_m} \tilde{s}_{j,k,m}\lambda_{j,k}$
SynthesisI(j):
  $\forall\, m \in \mathcal{M}(j):$   $\gamma_{j,m}$  +=  $\sum_{k\in\mathcal{K}_m} \tilde{s}_{j,k,m}\lambda_{j,k}$
```

3.7 Basis Properties

The lifting scheme provides us with the filter coefficients needed in the implementation of the fast wavelet transform. To find the basis functions and dual basis functions that are associated with them, we use the *cascade algorithm*. To synthesize a scaling function φ_{j_0,k_0} one simply initializes the coefficient $\lambda_{j_0,k} = \delta_{k,k_0}$. The inverse wavelet transform starting from level j_0 with all wavelet coefficients $\gamma_{j,m}$ with $j \geqslant j_0$ set to zero then results in $\lambda_{j,k}$ coefficients which converge to function values of φ_{j_0,k_0} as $j \to \infty$. In case the cascade algorithm converges in L_2 for both primal and dual scaling functions, biorthogonal filters (as given by the lifting scheme) imply biorthogonal basis functions.

One of the fundamental questions is how properties, such as convergence of the cascade algorithm, Riesz bounds, and smoothness, can be related back to properties of the filter sequences. This is a very hard question and at this moment no general answer is available to our knowledge. We thus have no mathematical proof that the wavelets constructed form an unconditional basis except in the case of the Haar wavelets. A recent result addressing these questions was obtained by Dahmen [7]. In particular, it is shown there which properties in addition to biorthogonality are needed to assure stable bases. Whether this result can be applied to the bases constructed here needs to be studied in the future.

Regarding smoothness, we have some partial results. It is easy to see that the Haar wavelets are not continuous and that the Linear wavelets are. The original Butterfly subdivision scheme is guaranteed to yield a C^1 limit function provided the connectivity of the vertices is at least 4. The modified Butterfly scheme that we use on the sphere, will also give C^1 limit functions, provided a locally smooth (C^1) map from the spherical triangulation to a planar triangulation exists. Unfortunately, the geodesic subdivision we use here does not have this property. However, the resulting functions appear visually smooth (see Figure 6). We are currently working on new spherical triangulations which have the property that the Butterfly scheme yields a globally C^1 function.

In principle, one can choose either the tetrahedron, octahedron, or icosahedron to start the geodesic sphere construction. Each of them has a particular number of triangles on each level, and therefore one of them might be more suited for a particular application or platform. The octahedron is the best choice in case of functions defined on the hemisphere (cfr. BRDF). The icosahedron will lead to the least area imbalance of triangles on each level and thus to (visually) smoother basis functions.

3.8 An Example

We argued at the beginning of this section that a given wavelet basis can be made more performant by lifting. In the section on interpolating bases we pointed out that a wavelet basis with Diracs for duals and a primal wavelet, which does not have 1 vanishing moment, unconditional convergence of the resulting series expansions cannot be insured anymore. We now give an example on the sphere which illustrates the numerical consequences of lifting.

Consider the function $f(s) = \sqrt{|s_x|}$ for $s = (s_x, s_y, s_z) \in S^2$. This function is everywhere smooth except on the great circle $s_x = 0$, where its derivative has a discontinuity. Since it is largely smooth but for a singularity at 0, it is ideally suited to exhibit problems in bases whose primal wavelet does not have a vanishing moment. Figure 10 shows the relative l_1 error as a function of the number of coefficients used in the synthesis stage. In order to satisfy the same error threshold the lifted basis requires only approximately $1/3$ the number of coefficients compared to the unlifted basis.

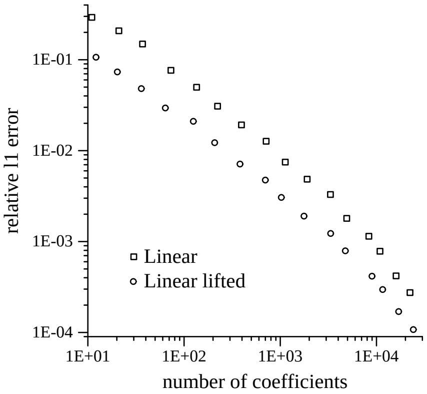

Figure 10: *Relative l_1 error as a function of the number of coefficients for the example function $f(s) = \sqrt{|s_x|}$ and (lifted) Linear wavelets. With the same number of coefficients the error is smaller by a factor of* 3 *or conversely a given error can be achieved with about* $1/3$ *the number of coefficients if the lifted basis is used.*

4 Implementation

We have implemented all the described bases in an interactive application running on an SGI Irix workstation. The basic data structure is a forest of triangle quadtrees [10]. The root level starts with 4 (tetrahedron), 8 (octahedron), or 20 (icosahedron) spherical triangles. These are recursively subdivided into 4 child triangles each. Naming edges after their opposite vertex, and children after the vertex they retain (the central child becomes T_0) leads to a consistent naming scheme throughout the entire hierarchy. Neighbor finding is a simple $O(1)$ (expected cost) function using bit operations on edge and triangle names to guide pointer traversal [10]. A vertex is allocated once and any level which contains it carries pointers to it. Each vertex carries a single λ and γ slot for vertex bases, while face bases carry a single λ and γ slot per spherical triangle. Our actual implementation carries other data such as surface normals and colors used for display, function values for error computations, and copies of all γ and λ values to facilitate experimentation. These are not necessary in a production system however.

Using a recursive data structure is more memory intensive (due to the pointer overhead) than a flat, array based representation of all coefficients as was used by LDW. However, using a recursive data structure enables the use of adaptive subdivision and results in simple recursive procedures for analysis and synthesis and a subdivision oracle. For interactive applications it is straightforward to select a level for display appropriate to the available graphics performance (polygons per second).

In the following subsections we address particular issues in the implementation.

4.1 Restricted Quadtrees

In order to support lifted bases and those which require stencils that encompass some neighborhood the quadtrees produced need to satisfy a restriction criterion. For the Linear vertex bases (lifted and unlifted) and the Bio-Haar basis no restriction is required. For Quadratic and lifted Bio-Haar bases no neighbor of a given face may be off by more than 1 subdivision level (every child needs a proper set of “aunts”). For the Butterfly basis a two-neighborhood must not be off by more than 1 subdivision level. These requirements are easily enforced during the recursive subdivision. The fact that we only need “aunts” (as opposed to “sisters”) for the lifting scheme allows us to have wavelets on *adaptively subdivided* hierarchies. This is a crucial departure from previous constructions, e.g., tree

Basis	Analysis	Synthesis	Lifted Basis	Analysis	Synthesis
Linear	3.59	3.55	Linear	5.85	5.83
Quadratic	21.79	21.00	Quadratic	24.62	24.68
Butterfly	8.43	8.42	Butterfly	10.64	10.62
Bio-Haar	4.31	6.09	Bio-Haar	42.43	36.08

Table 2: *Representative timings for wavelet transforms beginning with* 4 *spherical triangles and expanding to level* 9 *(*2^{20} *faces and* $2^{19}+2$ *vertices). All timings are given in seconds and measured on an SGI R4400 running at 150MHz. The initial setup time (allocating and initializing all data structures) took 100 seconds.*

wavelets employed by Gortler *et al.*[16] who also needed to support adaptive subdivision.

4.2 Boundaries

In the case of a hemisphere (top 4 spherical triangles of an octahedral subdivision), which is important for BRDF functions, the issues associated with the boundary need to be addressed. Lifting of vertex bases is unchanged, but the Quadratic and Butterfly schemes (as well as the lifted Bio-Haar bases) need neighbors, which may not exist at the boundary. This can be addressed by simply using another, further neighbor instead of the missing neighbor (across the boundary edge) to solve the associated matrix problem. It implicitly corresponds to adapting filter coefficients close to the boundary as done in interval constructions, see e.g. [5]. This construction automatically preserves the vanishing moment property even at the boundary. In the implementation of the Butterfly basis, we took a different approach and chose in our implementation to simply reflect any missing faces along the boundary.

4.3 Oracle

One of the main components in any wavelet based approximation is the oracle. The function of the oracle is to determine which coefficients are important and need to be retained for a reconstruction which is to meet some error criterion. Our system can be driven in two modes. The first selects a deepest level to which to expand all quadtrees. The storage requirements for this approach grow exponentially in the depth of the tree. For example our implementation cannot go deeper than 7 levels (starting from the tetrahedron) on a 32MB Indy class machine without paging. Creating full trees, however, allows for the examination of *all* coefficients throughout the hierarchies to in effect implement a perfect oracle. The second mode builds sparse trees based on a *deep refinement* oracle. In this oracle quadtrees are built depth first exploring the expansion to some (possibly very deep) finest level. On the way out of the recursion a local `AnalysisI` is performed and any subtrees whose wavelet coefficients are all below a user supplied threshold are deallocated. Once the sparse tree is built the restriction criterion is enforced and the (possibly lifted) analysis is run level wise.

The time complexity of this oracle is still exponential in the depth of the tree, but the storage requirements are proportional to the output size. With extra knowledge about the underlying function more powerful oracles can be built whose time complexity is proportional to the output size as well.

4.4 Transform Cost

The cost of a wavelet transform is proportional to the total number of coefficients, which grows by a factor of 4 for every level. For example, 9 levels of subdivision starting from 4 spherical triangles result in 2^{20} coefficients (each of λ and γ) for face bases and $2^{19}+2$ (each of λ and γ) for vertex bases. The cost of analysis and synthesis is proportional to the number of basis functions, while the constant of proportionality is a function of the stencil size. Table 2 summarizes timings of wavelet transforms for all the new bases. The initial setup took 100 seconds and includes allocation and initialization of all data structures and evaluation of the $\lambda_{9,k}$. Since the latter is highly dependent on the evaluation cost of the function to be expanded we used the constant function 1 for these timings. None of the matrices which arise in the Quadratic, and Bio-Haar bases (lifted and unlifted) was cached, thus the cost of solving the associated 4×4 matrices with a column pivoted QR (for Quadratic and lifted Bio-Haar) was incurred both during analysis and synthesis. If one is willing to cache the results of the matrix solutions this cost could be amortized over multiple transforms.

We make three main observations about the timings: (A) Lifting of vertex bases adds only a small extra cost, which is almost entirely due to the extra recursions; (B) the cost of the Butterfly basis is only approximately twice the cost of the Linear basis even though the stencil is much larger; (C) solving the 4×4 systems implied by Quadratic and lifted Bio-Haar bases increases the cost by a factor of approximately 5 over the linear case (note that there are twice as many coefficients for face bases as for vertex bases).

While the total cost of an entire transform is proportional to the number of basis functions, evaluating the resulting expansion at a point is proportional to the depth (log of the number of basis functions) of the tree times a constant dependent on the stencil size. The latter provides a great advantage over such bases as spherical harmonics whose evaluation cost at a single point is proportional to the total number of bases used.

5 Results

In this section we report on experiments with the compression of a planetary topographic data set, a BRDF function, and illumination of an anisotropic glossy sphere.

Most of these experiments involved some form of coefficient thresholding (in the oracle). In all cases this was performed as follows. Since all our bases are normalized with respect to the L_∞ norm, L_2 thresholding against some user supplied threshold ϵ becomes

$$\text{if } |\gamma_{j,m}|\sqrt{\text{supp}(\psi_{j,m})} < \epsilon,\ \gamma_{j,m} := 0.$$

Furthermore ϵ is scaled by $(\max(f) - \min(f))$ for the given function f to make thresholding independent of the scale of f.

5.1 Compression of Topographic Data

In this series of experiments we computed wavelet expansions of topographic data over the entire earth. This function can be thought of as both a surface, and as a scalar valued function giving height (depth) for each point on a sphere. The original data, ETOPO5 from the National Oceanographic and Atmospheric Administration gives the elevation (depth) of the earth from sea level in meters at a resolution of 5 arc minutes at the equator. Due to the large size of this data set we first resampled it to 10 arc minutes resolution. All expansions were performed starting from the tetrahedron followed by subdivision to level 9.

Figure 11 shows the results of these experiments (left and middle). After computing the coefficients of the respective expansions at the finest level of the subdivision an analysis was performed. After this step all wavelet coefficients below a given threshold were zeroed and the function was reconstructed. The thresholds were successively set to 2^{-i} for $i = 0, \ldots, 17$ resulting in the number of coefficients and relative l_1 error plotted (left graph). The error was computed with a numerical quadrature one level below the finest subdivision to insure an accurate error estimation. The results are plotted for all vertex and face bases (Linear, Quadratic, Butterfly, Bio-Haar, lifted and unlifted). We also computed l_2 and l_∞ error norms and the resulting graphs (not shown) are essentially identical

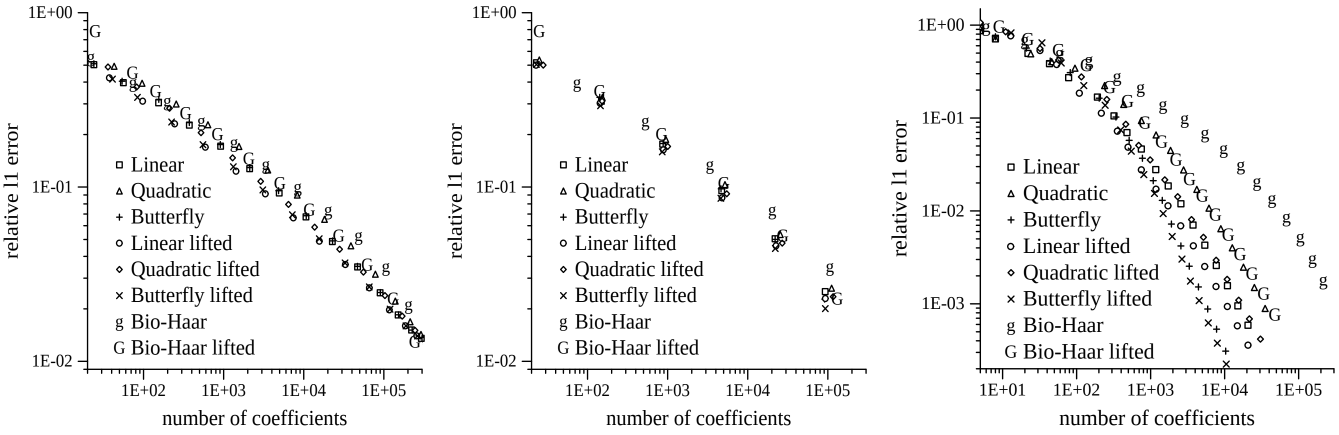

Figure 11: *Relative l_1 error as a function of the number of coefficients used during the reconstruction of the earth topographic data set (left and middle) and BRDF function (right). The six vertex bases and two face bases perform essentially the same for the earth. On the left with full expansion of the quadtrees to level 9 and thresholding. In the middle the results of the deep refinement oracle to level 10 with only a sparse tree construction. The curves are identical validating the refinement strategy. On the right the results of deep refinement to level 9 for the BRDF. Here the individual bases are clearly distinguished.*

(although the l_∞ error stays initially high before falling off due to deep canyon features). The plot reaches to about one quarter of all coefficients. The observed regime is linear as one would expect from the bases used.

The most striking observation about these error graphs is the fact that all bases perform similar. This is due to the fact that the underlying function is non-smooth. Consequently smoother bases do not perform any better than less performant ones. However, when drawing pictures of highly compressed versions of the data set the smoother bases produce visually better pictures (see Figure 12). Depending on the allowed error the compression can be quite dramatic. For example, 7 200 coefficients are sufficient to reach 7% error, while 119 000 are required to reach 2% error.

In a second set of experiments we used the deep refinement oracle (see Section 4.3) to explore the wavelet expansion to 10 levels (potentially quadrupling the number of coefficients) with successively smaller thresholds, once again plotting the resulting error in the middle graph of Figure 11. The error as a function of coefficients used is the same as the relationship found by the perfect oracle. This validates our deep refinement oracle strategy. Memory requirements of this approach are drastically reduced. For example, using a threshold of 2^{-9} during oracle driven refinement to level 10 resulted in 4 616 coefficients and consumed a total of 27MB (including 10MB for the original data set). Lowering the threshold to 2^{-10} yielded 10 287 coefficients and required 43MB (using the lifted Butterfly basis in both cases).

Finally Figure 12 shows some of the resulting adaptive data sets rendered with RenderMan using the Butterfly basis and a pseudo coloring, which maps elevation onto a piecewise linear color scale. Total runtime for oracle driven analysis and synthesis was 10 minutes on an SGI R4400 at 150MHz.

5.1.1 Comparison with LDW

The earth data set allows for a limited comparison of our results with those of LDW. They also compressed the ETOPO5 data set using pseudo orthogonalized (over a 2 neighborhood) Linear wavelets defined over the octahedron. They subdivide to 9 levels (on a 128MB machine) which corresponds to twice as many coefficients as we used (on a 180MB machine), suggesting a storage overhead of about 3 in our implementation. It is hard to compare the quality of the bases without knowing the exact basis used or the errors in the compressed reconstruction. However, LDW report the number of coefficients selected for a given threshold (741 for 0.02, 15 101 for 0.002, and 138 321 for 0.0005). Depending on the basis used we generally select fewer coefficients (6 000 – 15 000 for 0.002 and 28 000 – 65 000 for 0.0005). As timings they give 588 seconds (on a 100 MHz R4000) for analysis which is significantly longer than our smoothest basis (lifted Butterfly). Their reconstruction time ranges from 75 (741 coefficients) to 1 230 (138 058 coefficients) seconds which is also significantly longer than our times (see Table 2). We hypothesize that the timing and storage differences are largely due to their use of flat array based data structures. These do not require as much memory, but they are more compute intensive in the sparse polygonal reconstruction phase.

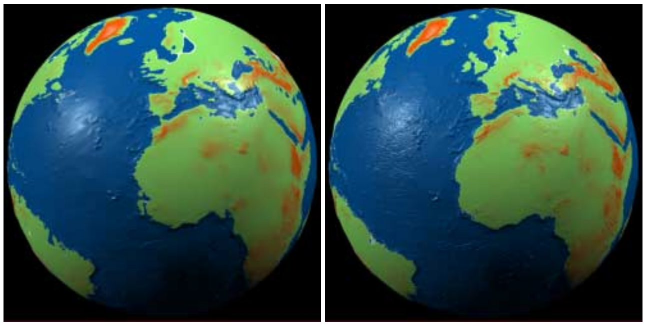

Figure 12: *Two views of the earth data set with* 15 000 *and* 190 000 *coefficients respectively using the Butterfly basis with l_1 errors of* 0.05 *and* 0.01 *respectively. In the image on the left coastal regions are rather smoothed since they contain little height variation (England and the Netherlands are merged and the Baltic sea is desertificated). However, such spiky features as the Cape Verde Islands off the coast of Africa are clearly preserved.*

5.2 BRDF Compression

In this series of experiments we explore the potential for efficiently representing BRDF functions with spherical wavelets. BRDF functions can arise from measurements, simulation, or theoretical models. Depending on the intended application different models may be preferable. Expanding BRDF functions in terms of locally supported hierarchical functions is particularly useful for wavelet based finite element illumination algorithms. It also has obvious applications for simulation derived BRDF functions such as those of Westin *et al.* [30] and Gondek *et al.* [15].

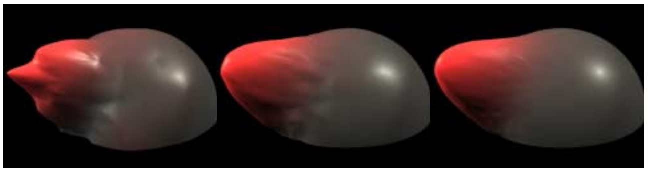

Figure 13: *Graphs of adaptive oracle driven approximations of the Schlick BRDF with* 19, 73, *and* 203 *coefficients respectively (left to right), using the lifted Butterfly basis. The associated thresholds were* 2^{-3}, 2^{-6}, *and* 2^{-9} *respectively, resulting in relative* l_1 *errors of* 0.35, 0.065, *and* 0.015 *respectively.*

The domain of a complete BRDF is a hemisphere times a hemisphere. In our experiments we consider only a fixed incoming direction and expand the resulting function over all outgoing directions (single hemisphere). To facilitate the computation of errors we used the BRDF model proposed by Schlick [23]. It is a simple Padé approximant to a micro facet model with geometric shadowing, a microfacet distribution function, but no Fresnel term. It has roughness ($r \in [0,1]$, where 0 is Dirac mirror reflection and 1 perfectly diffuse) and anisotropy ($p \in [0,1]$, where 0 is Dirac style anisotropy, and 1 perfect isotropy) parameters. To improve the numerical properties of the BRDF we followed the suggestion of Westin *et al.* [30] and expanded $\cos\theta_o f_r(\vec{\omega}_i, 0, \cdot)$.

In the experiments we used all 8 bases but specialized to the hemisphere. The parameters were $\theta_i = \pi/3$, $r = 0.05$, and $p = 1$. The results are summarized in Figure 11 (rightmost graph). It shows the relative l_1 error as a function of the number of coefficients used. This time we can clearly see how the various bases differentiate themselves in terms of their ability to represent the function within some error bound with a given budget of coefficients. We make several observations

- all lifted bases perform better than their unlifted versions, confirming our assertion that lifted bases are more performant;
- increasing smoothness in the bases (Butterfly) is more important than increasing the number of vanishing moments (Quadratic);
- dual lifting to increase dual vanishing moments increases compression ability dramatically (Bio-Haar and lifted Bio-Haar);
- overall the face based schemes do not perform as well as the vertex based schemes.

Figure 13 shows images of the graphs of some of the expansions. These used the lifted Butterfly basis with an adaptive refinement oracle which explored the expansion to level 9 (i.e., it examined 2^{19} coefficients). The final number of coefficients and associated relative l_1 errors were (left to right) 19 coefficients ($l_1 = 0.35$), 73 coefficients ($l_1 = 0.065$), and 203 coefficients ($l_1 = 0.015$). Total runtime was 170 seconds on an SGI R4400 at 150MHz.

5.3 Illumination

To explore the potential of these bases for global illumination algorithms we performed a simple simulation computing the radiance over a glossy, anisotropic sphere due to two area light sources. We emphasize that this is not a solution involving any multiple reflections, but it serves as a simple example to show the potential of these bases for hierarchical illumination algorithms. It also serves as an example of applying a finite element approach to a curved object (sphere) without polygonalizing it.

Figure 14 shows the results of this simulation. We used the lifted Butterfly basis and the BRDF model of Schlick with $r = 0.05$, $p = 0.05$, and an additive diffuse component of 0.005. Two area light sources illuminate the red sphere. Note the fine detail in the pinched off region in the center of the "hot" spot and also at the north pole where all "grooves" converge.

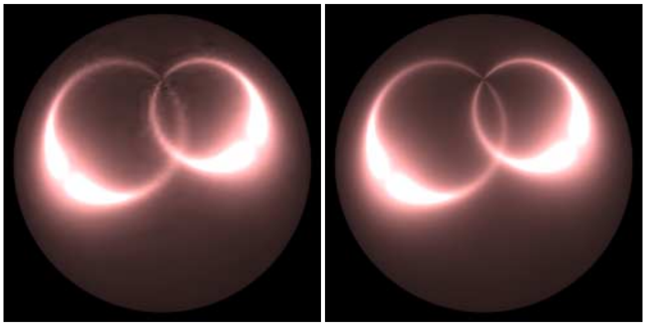

Figure 14: *Results of an illumination simulation using the lifted Butterfly basis. A red, glossy, anisotropic sphere is illuminated by 2 area light sources. Left: solution with* 2 000 *coefficients* ($l_1 = 0.017$)*. Right: solution with* 5 000 *coefficients* ($l_1 = 0.0035$)

6 Conclusions and Future Directions

In this paper we have introduced two new families of wavelets on the sphere. One family is based on interpolating scaling functions and one on the generalized Haar wavelets. They employ a generalization of multiresolution analysis to arbitrary surfaces and can be derived in a straightforward manner from the trivial multiresolution analysis with the lifting scheme. The resulting algorithms are simple and efficient. We reported on the application of these bases to the compression of earth data sets, BRDF functions and illumination computations and showed their potential for these applications. We found that

- for smooth functions the lifted bases perform significantly better than the unlifted bases;
- increasing the dual vanishing moments leads to better compression;
- smoother bases, even with only one vanishing moment, tend to perform better for smooth functions;
- our constructions allow non-equally subdivided triangulations of the sphere.

We believe that many applications can benefit from these wavelet bases. For example, using their localization properties a number of spherical image processing algorithms, such as local smoothing and enhancement, can be realized in a straightforward and efficient way [25].

While we limited our examination to the sphere, the construction presented here can be applied to other surfaces. In the case of the sphere enforcing vanishing *polynomial* moments was natural because of their connection with spherical harmonics. In the case of a general, potentially non-smooth (Lipschitz) surface, polynomial moments do not necessarily make much sense. Therefore, one might want to work with local maps from the surface to the tangent plane and enforce vanishing moment conditions in this plane.

Future research includes

- the generalization to arbitrary surfaces,
- the incorporation of smoother (C^2) subdivision schemes as recently introduced by Dyn *et al.* (personal communication, 1995),
- the use of these bases in applications such as the solution of differential and integral equations on the sphere as needed in, e.g., illumination or climate modeling.

Acknowledgments

The first author was supported by DEPSCoR Grant (DoD-ONR) N00014-94-1-1163. The second author was supported by NSF EPSCoR Grant EHR 9108772 and DARPA Grant AFOSR F49620-93-1-0083. He is also Senior Research Assistant of the National Fund of Scientific Research Belgium (NFWO). Other support came from Pixar Inc. We would also like to thank Princeton University and the GMD, Germany, for generous access to computing resources. Help with geometric data structures was provided by David Dobkin. Finally, the comments of the referees were very helpful in revising the paper.

References

[1] Alfeld, P., Neamtu, M., and Schumaker, L. L. Bernstein-Bézier polynomials on circles, sphere, and sphere-like surfaces. Preprint.

[2] Carnicer, J. M., Dahmen, W., and Peña, J. M. Local decompositions of refinable spaces. Tech. rep., Insitut für Geometrie und angewandete Mathematik, RWTH Aachen, 1994.

[3] Christensen, P. H., Stollnitz, E. J., Salesin, D. H., and DeRose, T. D. Wavelet Radiance. In *Proceedings of the 5th Eurographics Workshop on Rendering*, 287–302, June 1994.

[4] Cohen, A., Daubechies, I., and Feauveau, J. Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math. 45* (1992), 485–560.

[5] Cohen, A., Daubechies, I., Jawerth, B., and Vial, P. Multiresolution analysis, wavelets and fast algorithms on an interval. *C. R. Acad. Sci. Paris Sér. I Math. I*, 316 (1993), 417–421.

[6] Dahlke, S., Dahmen, W., Schmitt, E., and Weinreich, I. Multiresolution analysis and wavelets on S^2 and S^3. Tech. Rep. 104, Institut für Geometrie und angewandete Mathematik, RWTH Aachen, 1994.

[7] Dahmen, W. Stability of multiscale transformations. Tech. rep., Institut für Geometrie und angewandete Mathematik, RWTH Aachen, 1994.

[8] Dahmen, W., Prössdorf, S., and Schneider, R. Multiscale methods for pseudo-differential equations on smooth manifolds. In *Conference on Wavelets: Theory, Algorithms, and Applications*, C. K. C. et al., Ed. Academic Press, San Diego, CA, 1994, pp. 385–424.

[9] Daubechies, I. *Ten Lectures on Wavelets.* CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.

[10] Dutton, G. Locational Properties of Quaternary Triangular Meshes. In *Proceedings of the Fourth International Symposium on Spatial Data Handling*, 901–910, July 1990.

[11] Dyn, N., Levin, D., and Gregory, J. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *Transactions on Graphics 9*, 2 (April 1990), 160–169.

[12] Fekete, G. Rendering and Managing Spherical Data with Sphere Quadtrees. In *Proceedings of Visualization 90*, 1990.

[13] Freeden, W., and Windheuser, U. Spherical Wavelet Transform and its Discretization. Tech. Rep. 125, Universität Kaiserslautern, Fachbereich Mathematik, 1994.

[14] Girardi, M., and Sweldens, W. A new class of unbalanced Haar wavelets that form an unconditional basis for L_p on general measure spaces. Tech. Rep. 1995:2, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995. (ftp://ftp.math.scarolina.edu/pub/imi_95/imi95_2.ps).

[15] Gondek, J. S., Meyer, G. W., and Newman, J. G. Wavelength Dependent Reflectance Functions. In *Computer Graphics* Proceedings, Annual Conference Series, 213–220, 1994.

[16] Gortler, S., Schröder, P., Cohen, M., and Hanrahan, P. Wavelet Radiosity. In *Computer Graphics* Proceedings, Annual Conference Series, 221–230, August 1993.

[17] Gortler, S. J., and Cohen, M. F. Hierarchical and Variational Geometric Modeling with Wavelets. In *Proceedings Symposium on Interactive 3D Graphics*, 35–42, April 1995.

[18] Liu, Z., Gortler, S. J., and Cohen, M. F. Hierarchical Spacetime Control. *Computer Graphics* Proceedings, Annual Conference Series, 35–42, July 1994.

[19] Lounsbery, M. *Multiresolution Analysis for Surfaces of Arbitrary Topological Type*. PhD thesis, University of Washington, 1994.

[20] Lounsbery, M., DeRose, T. D., and Warren, J. Multiresolution Surfaces of Arbitrary Topological Type. Department of Computer Science and Engineering 93-10-05, University of Washington, October 1993. Updated version available as 93-10-05b, January, 1994.

[21] Mitrea, M. *Singular integrals, Hardy spaces and Clifford wavelets.* No. 1575 in Lecture Notes in Math. 1994.

[22] Nielson, G. M. Scattered Data Modeling. *IEEE Computer Graphics and Applications 13*, 1 (January 1993), 60–70.

[23] Schlick, C. A customizable reflectance model for everyday rendering. In *Fourth Eurographics Workshop on Rendering*, 73–83, June 1993.

[24] Schröder, P., and Hanrahan, P. Wavelet Methods for Radiance Computations. In *Proceedings 5th Eurographics Workshop on Rendering*, June 1994.

[25] Schröder, P., and Sweldens, W. Spherical wavelets: Texture processing. Tech. Rep. 1995:4, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995. (ftp://ftp.math.scarolina.edu/pub/imi_95/imi95_4.ps).

[26] Sillion, F. X., Arvo, J. R., Westin, S. H., and Greenberg, D. P. A global illumination solution for general reflectance distributions. *Computer Graphics (SIGGRAPH '91 Proceedings)*, Vol. 25, No. 4, pp. 187–196, July 1991.

[27] Sweldens, W. The lifting scheme: A construction of second generation wavelets. Department of Mathematics, University of South Carolina.

[28] Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. Tech. Rep. 1994:7, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1994. (ftp://ftp.math.scarolina.edu/pub/imi_94/imi94_7.ps).

[29] Westerman, R. A Multiresolution Framework for Volume Rendering. In *Proceedings ACM Workshop on Volume Visualization*, 51–58, October 1994.

[30] Westin, S. H., Arvo, J. R., and Torrance, K. E. Predicting reflectance functions from complex surfaces. *Computer Graphics (SIGGRAPH '92 Proceedings)*, Vol. 26, No. 2, pp. 255–264, July 1992.