THE LIFTING SCHEME:
A CONSTRUCTION OF SECOND GENERATION WAVELETS[*]

WIM SWELDENS[†]

Abstract. We present the lifting scheme, a simple construction of second generation wavelets, wavelets that are not necessarily translates and dilates of one fixed function. Such wavelets can be adapted to intervals, domains, surfaces, weights, and irregular samples. We show how the lifting scheme leads to a faster, in-place calculation of the wavelet transform. Several examples are included.

Key words. wavelet, multiresolution, second generation wavelet, lifting scheme

AMS subject classifications. 42C15

1. Introduction.

Wavelets form a versatile tool for representing general functions or data sets. Essentially we can think of them as data building blocks. Their fundamental property is that they allow for representations which are *efficient* and which can be computed *fast*. In other words, wavelets are capable of quickly capturing the essence of a data set with only a small set of coefficients. This is based on the fact that most data sets have correlation both in time (or space) and frequency. Because of the time-frequency localization of wavelets, efficient representations can be obtained. Indeed, building blocks which already reflect the correlation present in the data lead to more compact representations. This is the key to applications. Over the last decade wavelets have found applications in numerous areas of mathematics, engineering, computer science, statistics, physics, etc.

Wavelet functions $\psi_{j,m}$ are traditionally defined as the dyadic translates and dilates of one particular $L_2(\mathbf{R})$ function, the *mother wavelet* ψ: $\psi_{j,m}(x) = \psi(2^j x - m)$. We refer to such wavelets as *first generation wavelets*. In this paper we introduce a more general setting where the wavelets are not necessarily translates and dilates of each other but still enjoy all the powerful properties of first generation wavelets. These wavelets are referred to as *second generation wavelets*. We present the *lifting scheme*, a simple, but quite powerful tool to construct second generation wavelets.

Before we consider the generalization to the second generation case, let us review the properties of first generation wavelets which we would like to preserve.

P1: Wavelets form a Riesz basis for $L_2(\mathbf{R})$ and an unconditional basis for a wide variety of function spaces $\mathcal{F}$, such as Lebesgue, Lipschitz, Sobolev, and Besov spaces. If we denote the wavelet basis by $\{\psi_{j,m} \mid j,m\}$, we can represent a general function f in $\mathcal{F}$ as $f = \sum_{j,m} \gamma_{j,m}\, \psi_{j,m}$, with unconditional convergence in the norm of $\mathcal{F}$. Simple characterizations of the $\mathcal{F}$-norm of f in terms of the absolute value of its wavelet coefficients $\gamma_{j,m}$ exist.

P2: One has explicit information concerning the coordinate functionals $\widetilde{\psi}_{j,m}$ where $\gamma_{j,m} = \widetilde{\psi}_{j,m}(f)$. The wavelets are either orthogonal or the dual (biorthogonal) wavelets are known.

P3: The wavelets and their duals are local in space and frequency. Some wavelets are even compactly supported. The frequency localization follows from the smoothness of the wavelets (decay towards high frequencies) and the fact that they have vanishing polynomial moments (decay towards low frequencies).

[*] July 1995, Revised November 1997.

[†] Lucent Technologies, Bell Laboratories, Rm. 2C-175, 600 Mountain Avenue, Murray Hill NJ 07974, wim@bell-labs.com.

P4: Wavelets fit into the framework of multiresolution analysis. This leads to the *fast wavelet transform*, which allows us to pass between the function f and its wavelet coefficients $\gamma_{j,m}$ in linear time.

These properties result in the fact that, quoted from David Donoho in [58], "*wavelets are optimal bases for compressing, estimating, and recovering functions in $\mathcal{F}$.*" Roughly speaking, for a general class of functions, the essential information contained in a function is captured by a small fraction of the wavelet coefficients. Again this is the key to applications. Wavelets have proved to be useful in various application domains such as: signal and image processing, data compression, data transmission, the numerical solution of differential and integral equations, and noise reduction.

Many first generation wavelet families have been constructed over the last ten year. We refer to the work of (in alphabetical order) Aldroubi-Unser [2, 3, 108, 107], Battle-Lemarié [13, 78], Chui-Wang [19, 25, 24, 23], Cohen-Daubechies [28], Cohen-Daubechies-Feauveau [29], Daubechies [47, 49, 48], Donoho [57, 56], Frazier-Jawerth [65, 67, 66], Herley-Vetterli [73, 110], Kovačević-Vetterli [77, 111], Mallat [85, 84, 86], Meyer [87], and many more. Except for Donoho, they all rely on the Fourier transform as a basic construction tool. The reason is that translation and dilation become algebraic operations in the Fourier domain.

In fact, in the early 80's, several years before the above developments, Strömberg discovered the first orthogonal wavelets with a technique based on spline interpolation which does not rely on the Fourier transform [103].

The construction as initiated by Daubechies and co-workers essentially consists of three stages. The *algebraic stage* involves constructing the filters that are used in the fast wavelet transform; more precisely, it consists of finding certain polynomials and assuring that above property P4 is satisfied. In the *analytic stage*, one shows that wavelets associated with these filters exist, that they are localized (property P3), and that they form a basis for the proper function space (property P1). In the *geometrical stage*, one checks the smoothness of the basis functions (property P3). In this context, we mention the work of Collela and Heil [37, 38], Daubechies and Lagarias [50, 51], Eirola [63], Rioul [94], and Villemoes [113, 112].

Let us next consider applications which illustrate the need for generalizations of first generation wavelets.

G1: While first generation wavelets provided bases for functions defined on $\mathbf{R}^n$, applications such as data segmentation and the solution of partial differential and integral equations on general domains require wavelets that are defined on arbitrary, possibly non-smooth, domains of $\mathbf{R}^n$, as well as wavelets adapted to "life" on curves, surfaces or manifolds.

G2: Diagonalization of differential forms, analysis on curves and surfaces, and weighted approximation require a basis adapted to weighted measures; however, first generation wavelets typically provide bases only for spaces with translation invariant (Haar-Lebesgue) measures.

G3: Many real life problems require algorithms adapted to irregular sampled data, while first generation wavelets imply a regular sampling of the data.

A generalization of first generation wavelets to the settings G1-G3, while preserving the properties P1-P4 is needed. We refer to such wavelets as *second generation wavelets*. The key lies in the observations (A) that translation and dilation cannot be maintained in the settings G1-G3, and (B) that translation and dilation are not essential in obtaining the properties P1–P4. Giving up translation and dilation, however, implies that the Fourier transform can no longer be used as a construction tool.

A proper substitute is needed.

Several results concerning the construction of wavelets adapted to some of the cases in G1-G3 already exist. For example, we have wavelets on an interval [8, 10, 18, 30, 31, 88], wavelets on bounded domains [27, 74], spline wavelets for irregular samples, [15, 7, 45], and weighted wavelets [11, 12, 104]. These constructions are tailored toward one specific setting. Other instances of second generation wavelets have been reported in the literature, e.g., the construction of scaling functions through subdivision [41], basis constructions [43], as well as the development of stability criteria [41, 42].

In this paper, we present the *lifting scheme*, a simple, general construction of second generation wavelets. The basic idea, which inspired the name, is to start with a very simple or trivial multiresolution analysis, and gradually work one's way up to a multiresolution analysis with particular properties. The lifting scheme allows one to custom-design the filters, needed in the transform algorithms, to the situation at hand. In this sense it provides an answer to the algebraic stage of a wavelet construction. Whether these filters actually generate functions which form a stable basis (analytic stage) or have smoothness (geometric stage), remains to be checked in each particular case. The lifting scheme also leads to a fast in-place calculation of the wavelet transform, i.e. an implementation that does not require auxiliary memory.

The paper is organized as follows. We start out by discussing related work in Section 2. In Sections 3, 4, 5, and 6 we generalize respectively multiresolution analysis, cascade algorithm, wavelets, and the fast wavelet transform to the second generation setting. With the notation introduced in Section 7 we are able to state and prove the lifting scheme in Section 8. Section 9 discusses the lifted fast wavelet transform, while Section 10 covers the cakewalk construction, an enhanced version of the lifting scheme. Sections 11, 12, and 13 introduce three possible examples of an initial multiresolution analysis to start lifting: respectively generalized Haar wavelets, the Lazy wavelet, and biorthogonal Haar wavelets. Finally, Section 14 contains a discussion of applications and future research.

2. Related work. The idea of second generation wavelets and abandoning the Fourier transform as a construction tool for wavelets is not entirely new and, over the last few years, has been researched by several independent groups. In this section we discuss these development and their relationship with lifting.

The lifting scheme was originally inspired by the work of David Donoho on one side and Michael Lounsbery, Tony De Rose, and Joe Warren on the other. In [56, 57], Donoho presents the idea of interpolating and average-interpolating wavelets, a construction of first generation wavelets which relies on polynomial interpolation and subdivision as construction tools rather than the Fourier transform. It thus can be generalized to interval constructions [59] or weighed wavelets [104]. Lounsbery, De Rose, and Warren [79, 80], construct wavelets for the approximation of polyhedral surfaces of arbitrary genus. The wavelets are constructed by orthogonalizing scaling functions in a local neighborhood. We will show later how this can be seen as a special case of lifting.

The lifting scheme can also be used to construct first generation wavelets, see [105, 52]. Although in this setting, the lifting will never come up with wavelets which could not have been found using the Cohen-Daubechies-Feauveau machinery in [29], it leads to two new insights: a custom-design construction of wavelets, and a faster, in-place implementation of existing wavelet transforms [52]. In the first generation setting, lifting has many contacts with certain filter design algorithms used in signal processing. Those connections are pointed out in [105, 52].

Over the last few years Donovan, Hardin, Geronimo, and Massopust have developed techniques to construct wavelets based on fractal interpolation functions [60, 61, 62, 70]. They also introduced the concept of several generating functions (multi-wavelets). As this technique does not rely on the Fourier transform either, it too potentially can be used to construct second generation wavelets.

Several spatial constructions of spline wavelets on irregular grids have been proposed [15, 7]. In [45], Dahmen and Micchelli propose a spatial construction of compactly supported wavelets that generate complementary spaces in a multiresolution analysis of univariate irregular knot splines.

Dahmen already made use of a technique related to lifting in the first generation setting [40] and later introduced a multiscale framework related to second generation wavelets [41].

Finally, after finishing this work, the author learned of two other very similar techniques developed independent of each other and of lifting. Harten and Abgral developed a general multiresolution approximation framework based on prediction [71, 1], while Dahmen and co-workers [17, 46] develop a mechanism to characterize *all* stable biorthogonal decomposition. We will come back to this toward the end of the paper.

3. Multiresolution analysis. In this section we present the second generation version of multiresolution analysis. We keep most of the terminology and symbols of the first generation case, although their meaning can be quite different. For example, we maintain the name scaling function although it can be a little misleading since the scaling function can no longer be written as linear combinations of scaled versions of itself.

Consider a general function space $\mathrm{L}_2 = \mathrm{L}_2(X, \Sigma, \mu)$, with $X \subset \mathbf{R}^n$ being the spatial domain, Σ a σ-algebra, and μ a non-atomic measure on Σ. We do not require the measure to be translation invariant, so weighted measures are allowed. We assume (X, d) is a metric space.

Definition 3.1. *A* ***multiresolution analysis*** $\mathbf{M}$ *of* L_2 *is a sequence of closed subspaces* $\mathbf{M} = \{V_j \subset \mathrm{L}_2 \mid j \in \mathcal{J} \subset \mathbf{Z}\}$*, so that*

1. $V_j \subset V_{j+1}$,
2. $\bigcup_{j \in \mathcal{J}} V_j$ *is dense in* L_2,
3. *for each* $j \in \mathcal{J}$, V_j *has a Riesz basis given by* scaling functions $\{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}$.

One can think of $\mathcal{K}(j)$ as a general index set. We assume that $\mathcal{K}(j) \subset \mathcal{K}(j+1)$. We consider two cases:

I: $\mathcal{J} = \mathbf{N}$: This means there is one coarsest level V_0. This is the case if $\mu(X) < \infty$.

II: $\mathcal{J} = \mathbf{Z}$: We have a fully bi-infinite setting. This is typical when $\mu(X) = \infty$. We then add the condition that

$$\bigcap_{j \in \mathcal{J}} V_j = \{\mathbf{0}\}.$$

A *dual multiresolution analysis* $\widetilde{\mathbf{M}} = \{\widetilde{V}_j \mid j \in \mathcal{J}\}$ consists of spaces $\widetilde{V}_j$ with Riesz bases given by *dual scaling functions* $\widetilde{\varphi}_{j,k}$. These dual scaling functions are biorthogonal with the scaling functions, in the sense that

$$\langle\, \varphi_{j,k}, \widetilde{\varphi}_{j,k'} \rangle \;=\; \delta_{k,k'} \quad \text{for} \quad k,\, k' \in \mathcal{K}(j)\,. \tag{3.1}$$

For $f \in \mathrm{L}_2$, define the coefficients $\lambda_{j,k} = \langle f, \widetilde{\varphi}_{j,k} \rangle$ and consider the projections

$$P_j\, f \;=\; \sum_{k \in \mathcal{K}(j)} \lambda_{j,k}\, \varphi_{j,k}.$$

If the projection operators P_j are uniformly bounded in L_2, then

$$\lim_{j \to \infty} \| f - P_j\, f \| = 0.$$

First generation scaling functions reproduce polynomials up to a certain degree. To generalize this, consider a set of $\mathcal{C}^\infty$ functions on X, $\{P_p \mid p = 0, 1, 2, \ldots\}$, with $P_0 \equiv 1$ and so that the restrictions of a finite number of these functions to any ϵ-ball are linearly independent. We then say that the *order of the multiresolution analysis* is N, if for all $j \in \mathcal{J}$, each P_p with $0 \leqslant p < N$ can be represented point wise as a linear combination of the $\{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}$,

$$P_p(x) \;=\; \sum_{k \in \mathcal{K}(j)} c_{j,k}^{p}\, \varphi_{j,k}(x)\,.$$

We let $\widetilde{N}$ be the order of the dual multiresolution analysis, where we use a similar set of functions $\widetilde{P}_p$. In case X is a domain in $\mathbf{R}^n$ the functions P_p typically will be polynomials; in case X is a manifold, the functions P_p can, e.g., be parametric images of polynomials. However, in a practical situation one often has no explicit knowledge of the parameterization. This is why we use a very general definition of the order. Our definition obviously depends on the choice of P_p, but we do not include this dependency in the notation to avoid overloading. Most of the examples only have $N = 1$ in which case there is no dependency as $P_0 = 1$.

We assume that the dual functions are integrable and normalize them as

$$\int_X \widetilde{\varphi}_{j,k}\, d\mu \;=\; 1\,. \tag{3.2}$$

This implies that if $N > 0$,

$$\sum_{k \in \mathcal{K}(j)} \varphi_{j,k}(x) \;=\; 1\,. \tag{3.3}$$

4. Cascade algorithm. A question which immediately arises is how to construct scaling functions and dual scaling functions. As in the first generation case, there is often no analytic expression for them, and they are only defined through an iterative procedure, the *cascade algorithm*. In this section we present the second generation version of the cascade algorithm. To do so we need two things: a *set of partitionings* and a *filter*.

Let us start by defining a filter. The definition of multiresolution analysis implies that for every scaling function $\varphi_{j,k}$ ($j \in \mathcal{J}$, $k \in \mathcal{K}(j)$), coefficients $\{h_{j,k,l} \mid l \in \mathcal{K}(j{+}1)\}$ exist so that formally

$$\varphi_{j,k} \;=\; \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l}\, \varphi_{j+1,l}\,. \tag{4.1}$$

We refer to this equation as a *refinement relation*. Each scaling function can be written as a linear combination of scaling functions on the next finer level. To ensure that the summation in (4.1) is well defined we need to clearly state the definition of a *filter*. In this paper we only consider finite filters.

Definition 4.1. *A set of real numbers $\{h_{j,k,l} \mid j \in \mathcal{J}, k \in \mathcal{K}(j), l \in \mathcal{K}(j+1)\}$ is called a* ***finite filter*** *if:*

1. *For each j and k only a finite number of coefficients $h_{j,k,l}$ are non zero, and thus the set*
$$\mathcal{L}(j,k) = \{l \in \mathcal{K}(j+1) \mid h_{j,k,l} \neq 0\}$$
is finite.
2. *For each j and l only a finite number of coefficients $h_{j,k,l}$ are non zero, and thus the set*
$$\mathcal{K}(j,l) = \{k \in \mathcal{K}(j) \mid h_{j,k,l} \neq 0\}\,.$$
is finite.
3. *The size of the sets $\mathcal{L}(j,k)$ and $\mathcal{K}(j,l)$ is uniformly bounded for all j, k, and l.*

Note that in the first generation case $h_{j,k.l} = h_{l-2k}$, so if $\{h_k \mid k\}$ is a finite sequence the filter is finite according to the above definition. We will always choose our indices consistently so that $j \in \mathcal{J}$, $k \in \mathcal{K}(j)$, and $l \in \mathcal{K}(j+1)$, even though it will not always be explicitly mentioned. The above defined index sets indicate which elements are non-zero on each row (respectively column) of the (possibly infinite) matrix $\{h_{j,k,l} \mid k \in \mathcal{K}(j), l \in \mathcal{K}(j+1)\}$. We can think of them as adjoints of each other as

$$\mathcal{K}(j,l) = \{k \in \mathcal{K}(j) \mid l \in \mathcal{L}(j,k)\}\,.$$

The dual scaling functions satisfy refinement relations with coefficients $\{\widetilde{h}_{j,k,l}\}$. We can define similar index sets (denoted with tilde).

A set of partitionings $\{S_{j,k}\}$ can be thought of as the replacement for the dyadic intervals on the real line in the first generation case. Again each scaling function $\varphi_{j,k}$ is associated with exactly one set $S_{j,k}$. We use the following definition.

Definition 4.2. *A set of measurable subsets $\{S_{j,k} \in \Sigma \mid j \in \mathcal{J}, k \in \mathcal{K}(j)\}$ is called a* ***set of partitionings*** *if*

1. $\forall j \in \mathcal{J}\ :\ \operatorname{clos} \bigcup_{k \in \mathcal{K}(j)} S_{j,k} = X$ *and the union is disjoint,*
2. $\mathcal{K}(j) \subset \mathcal{K}(j+1)$,
3. $S_{j+1,k} \subset S_{j,k}$,
4. *For a fixed $k \in \mathcal{K}(j_0)$, $\bigcap_{j>j_0} S_{j,k}$ is a set which contains 1 point. We denote this point with x_k.*

The purpose now is to use a filter and a set of partitionings to construct scaling functions that satisfy (4.1). Assume we want to synthesize φ_{j_0,k_0}. First define a Kronecker sequence $\{\lambda_{j_0,k} = \delta_{k,k_0} \mid k \in \mathcal{K}(j_0)\}$. Then, generate sequences $\{\lambda_{j,k} \mid k \in \mathcal{K}(j)\}$ for $j > j_0$ by recursively applying the formula:

$$\lambda_{j+1,l} \;=\; \sum_{k \in \mathcal{K}(j,l)} h_{j,k,l}\, \lambda_{j,k}\,.$$

Next we construct the functions

$$f^{(j)}_{j_0,k_0} \;=\; \sum_{k \in \mathcal{K}(j)} \lambda_{j,k}\, \chi_{S_{j,k}} \qquad j \geqslant j_0\,. \tag{4.2}$$

These functions satisfy for $j > j_0$,

$$f^{(j)}_{j_0,k_0} \;=\; \sum_l h_{j_0,k_0,l}\, f^{(j)}_{j_0+1,l}\,. \tag{4.3}$$

If $\lim_{j\to\infty} f^{(j)}_{j_0,k_0}$ converges to a function in L_2, we define this function to be φ_{j_0,k_0}. This procedure is called the *cascade algorithm*. The limit functions satisfy

$$\lim_{j\to\infty} \lambda_{j,k} \;=\; \varphi_{j_0,k_0}(x_k) \quad \text{a.e.}$$

If the cascade algorithm converges for all j_0 and k_0, we get a set of scaling functions that satisfies the refinement equation (4.1). This can be seen by letting j go to infinity in (4.3). Note how the resulting functions depend both on the filter and the set of partitionings. If the scaling functions generate a multiresolution analysis, the cascade algorithm started with a sequence $\{\lambda_{j_0,k} \mid k \in \mathcal{K}(j)\}$ that belongs to $\ell^2(\mathcal{K}(j))$ converges to

$$\sum_k \lambda_{j_0,k}\, \varphi_{j_0,k}\,.$$

The dual scaling function are constructed similarly starting from a finite filter $\widetilde{h}$, the same set of partitionings, and an initial Kronecker sequence $\{\lambda_{j_0,k} = \delta_{k,k_0}/\mu(S_{j_0,k_0}) \mid k \in \mathcal{K}(j_0)\}$. The normalization of the initial sequences assures that $\langle\, \widetilde{\varphi}_{j,k}, \varphi_{j,k}\,\rangle = 1$.

An interesting question is now whether the biorthogonality condition (3.1) can be related back to the filters h and $\widetilde{h}$. By writing out the refinement relations we see that the biorthogonality (3.1) implies that

$$\sum_l h_{j,k,l}\, \widetilde{h}_{j,k',l} \;=\; \delta_{k,k'} \quad \text{for} \quad j \in \mathcal{J},\; k, k' \in \mathcal{K}(j)\,. \tag{4.4}$$

but the converse is not immediately true. More precisely, if the filter coefficients satisfy (4.4), and the cascade algorithm for the primal and dual scaling functions converges, then the resulting scaling functions are biorthogonal. This follows from the fact that (4.4) assures that the intermediate functions of the form $f^{(j)}_{j_0,k_0}$ in (4.2) (which converge to the scaling functions) are biorthogonal at each stage j.

It is important to note that not every filter corresponds to a set of scaling functions, i.e., the convergence of the cascade algorithm is not guaranteed. We would like to have a condition which relates convergence of the cascade algorithm and the Riesz basis property back to the filter coefficients, similar to the Cohen criterion in the first generation case [26] or the Cohen-Daubechies-Feauveau theorem [29] or [48, Theorem 8.3.1]. This result is part of the analysis phase of the construction. As we mentioned earlier, this paper is mostly concerned with the algebraic phase and the generation of the filter coefficients.

5. Wavelets.

First generation wavelets are defined as basis functions for spaces complementing V_j in V_{j+1}. The same idea remains in the second generation case. This leads to the following definition

DEFINITION 5.1. *A set of functions $\{\psi_{j,m} \mid j \in \mathcal{J},\, m \in \mathcal{M}(j)\}$, where $\mathcal{M}(j) = \mathcal{K}(j+1) \setminus \mathcal{K}(j)$ is a* ***set of wavelet functions*** *if*

1: The space $W_j = \operatorname{clos\,span}\{\psi_{j,m} \mid m \in \mathcal{M}(j)\}$ is a complement of V_j in V_{j+1} and $W_j - \widetilde{V}_j$.

2: If $\mathcal{J} = \mathbf{Z}$: The set $\{\psi_{j,m}/\|\psi_{j,m}\| \mid j \in \mathcal{J}, m \in \mathcal{M}(j)\}$ is a Riesz basis for L_2. *If $\mathcal{J} = \mathbf{N}$: The set $\{\psi_{j,m}/\|\psi_{j,m}\| \mid j \in \mathcal{J}, m \in \mathcal{M}(j)\} \cup \{\varphi_{0,k}/\|\varphi_{0,k}\| \mid k \in \mathcal{K}(0)\}$ is a Riesz basis for* L_2.

We always assume that the index m belongs to the set $\mathcal{M}(j)$. The dual basis is given by *dual wavelets* $\widetilde{\psi}_{j,m}$, which are biorthogonal to the wavelets,

$$\langle\, \psi_{j,m}, \widetilde{\psi}_{j',m'} \,\rangle \;=\; \delta_{m,m'}\, \delta_{j,j'}\,. \tag{5.1}$$

The dual wavelets span spaces $\widetilde{W}_j$ which complement $\widetilde{V}_j$ in $\widetilde{V}_{j+1}$ and $\widetilde{W}_j - V_j$. For $f \in \mathrm{L}_2$, define the coefficients $\gamma_{j,m} \;=\; \langle\, f, \widetilde{\psi}_{j,m} \,\rangle$. Then

$$f \;=\; \sum_{j,m} \gamma_{j,m}\, \psi_{j,m}\,.$$

Their definition implies that the wavelets satisfy refinement relations of the form

$$\psi_{j,m} \;=\; \sum_{l} g_{j,m,l}\, \varphi_{j+1,l}\,. \tag{5.2}$$

We assume that $g = \{g_{j,m,l} \mid j \in \mathcal{J}, m \in \mathcal{M}(j), l \in \mathcal{K}(j+1)\}$ is a finite filter according to Definition 4.1 with k substituted by m. This leads to the definition of the uniformly bounded finite sets

$$\mathcal{M}(j,l) \;=\; \{m \in \mathcal{M}(j) \mid g_{j,m,l} \neq 0\} \quad \text{and} \quad \mathcal{L}(j,m) \;=\; \{l \in \mathcal{K}(j{+}1) \mid m \in \mathcal{M}(j,l)\}\,.$$

The dual wavelets satisfy refinement relations with a finite filter $\widetilde{g}$.

Also, since $\varphi_{j+1,l} \in V_j \oplus W_j$, it holds that

$$\varphi_{j+1,l} \;=\; \sum_{k} \widetilde{h}_{j,k,l}\, \varphi_{j,k} + \sum_{m} \widetilde{g}_{j,m,l}\, \psi_{j,m}\,.$$

The biorthogonality (5.1) combined with (3.1) implies the following relations between the filters:

$$\begin{aligned} \sum_{l} g_{j,m,l}\, \widetilde{g}_{j,m',l} &\;=\; \delta_{m,m'} & \qquad \sum_{l} h_{j,k,l}\, \widetilde{g}_{j,m,l} &\;=\; 0 \\ \sum_{l} h_{j,k,l}\, \widetilde{h}_{j,k',l} &\;=\; \delta_{k,k'} & \qquad \sum_{l} g_{j,m,l}\, \widetilde{h}_{j,k,l} &\;=\; 0\,. \end{aligned} \tag{5.3}$$

Definition 5.2. *A set of filters $\{h, \widetilde{h}, g, \widetilde{g}\}$ is a* ***set of biorthogonal*** *filters if condition* (5.3) *is satisfied.*

Now given a set of biorthogonal filters and a set of partitionings, and assuming that the cascade algorithm converges, the resulting scaling functions, wavelets, dual scaling functions and dual wavelets are biorthogonal in the sense that

$$\begin{aligned} \langle\, \widetilde{\varphi}_{j,k}, \varphi_{j,k'} \,\rangle &= \delta_{k,k'} \\ \langle\, \widetilde{\psi}_{j,m}, \psi_{j,m'} \,\rangle &= \delta_{m,m'} \\ \langle\, \widetilde{\varphi}_{j,k}, \psi_{j,m} \,\rangle &= 0 \\ \langle\, \widetilde{\psi}_{j,m}, \varphi_{j,k} \,\rangle &= 0\,. \end{aligned}$$

Next we need to generalize the notion of vanishing polynomial moments. We therefore use the (non polynomial) functions P_p defined in Section 3. If the scaling functions $\varphi_{j,k}$ with $k \in \mathcal{K}(j)$ reproduce P_p, then

$$\int_X P_p \, \widetilde{\psi}_{j,m} \, d\mu \; = \; 0 \quad \text{for} \quad 0 \leqslant p < N, \; j \in \mathcal{J}, \; m \in \mathcal{M}(j) \,.$$

We say that the dual wavelets have N vanishing moments. Similarly, the wavelets have $\widetilde{N}$ vanishing moments.

6. Fast wavelet transform. The basic idea of a wavelet transform is the same as in the first generation case. Given the set of coefficients $\{\lambda_{n,k} \mid k \in \mathcal{K}(n)\}$, calculate the $\{\gamma_{j,m} \mid n_0 \leqslant j < n, \, m \in \mathcal{M}(j)\}$ and $\{\lambda_{n_0,k} \mid k \in \mathcal{K}(n_0)\}$. From the refinement relation of the dual scaling functions and wavelets, we see that a fast forward wavelet transform is given by recursive application of

$$\lambda_{j,k} = \sum_{l \in \widetilde{\mathcal{L}}(j,k)} \widetilde{h}_{j,k,l} \, \lambda_{j+1,l} \quad \text{and} \quad \gamma_{j,m} = \sum_{l \in \widetilde{\mathcal{L}}(j,m)} \widetilde{g}_{j,m,l} \, \lambda_{j+1,l} \,.$$

Similarly, the inverse transform follows from the recursive application of

$$\lambda_{j+1,l} \; = \; \sum_{k \in \mathcal{K}(j,l)} h_{j,k,l} \, \lambda_{j,k} \; + \sum_{m \in \mathcal{M}(j,l)} g_{j,m,l} \, \gamma_{j,m} \,.$$

The major difference with the first generation fast wavelet transform, and thus with traditional subband transforms, is that the filter coefficients are different for every coefficient. One has to be careful analyzing the complexity of the second generation fast wavelet transform. For general filters the complexity need not be linear as the number of terms in the above summation, albeit finite, can grow from level to level. This is precisely why Definition 4.1 of a finite filter requires the sizes of the index sets $\mathcal{L}$, $\mathcal{K}$, and $\mathcal{M}$ to be *uniformly* bounded. This leads to the following corollary.

Corollary 6.1. *In case the filters h, g, $\widetilde{h}$, and $\widetilde{g}$ are finite, the second generation fast wavelet transform is a linear time algorithm.*

Note that in a computer implementation the data structure for the filters can become much more complex than in the first generation case and therefore has to be designed carefully.

In case the wavelets form an unconditional basis, the condition number of the wavelet transform is bounded independent of the number of levels. Consequently the propagation of numerical round-off error in floating point calculations will be bounded. As we mentioned before, lifting does not guarantee stability and bounded condition numbers. However in a practical situation involving spherical wavelets [99] we numerically estimated the condition number and found it to vary little with the number of levels. For a spherical wavelet transform involving roughly 650 000 coefficients we found the condition number to be approximately 8.

7. A word on notation. So far we have been using a notation involving the filter coefficients $h_{j,k,l}$ and $g_{j,m,l}$. As one can see this leads to expressions involving many indices. We will refer to it as the *index notation*. In this section we introduce a new notation, which we refer to as the *operator notation*. The advantage is that both the statement and the proof of some results become more elegant. Statements in the operator notation will also formally look the same as in the first generation case. In this way it helps to shed light on why things work. The disadvantage is that the

$$\begin{tikzcd}
 & \ell^2(\mathcal{K}(j)) \arrow[dl, shift left, "H^*\ E^*"] \arrow[dd, shift left, "S^*"] \\
\ell^2(\mathcal{K}(j+1)) \arrow[loop left, "T"] \arrow[ur, shift left, "H\ E"] \arrow[dr, shift left, "G\ D"] & \\
 & \ell^2(\mathcal{M}(j)) \arrow[ul, shift left, "G^*\ D^*"] \arrow[uu, shift left, "S"]
\end{tikzcd}$$

FIG. 7.1. *Schematic representation of operators, their domain and range. This scheme can be used to verify that the order in which operators are applied is correct and which operators can be added.*

operator notation is not practical and that it obscures implementation. Therefore we always state results in the index notation as well.

First consider the spaces $\ell^2(\mathcal{K}(j+1))$, $\ell^2(\mathcal{K}(j))$, and $\ell^2(\mathcal{M}(j))$, with their usual norm and inner product. We denote elements of these spaces by respectively a, b, and c, so that

$$a \;=\; \{a_l \mid l \in \mathcal{K}(j+1)\} \in \ell^2(\mathcal{K}(j+1)),$$

and, similarly, mutatis mutandis, for $b \in \ell^2(\mathcal{K}(j))$ and $c \in \ell^2(\mathcal{M}(j))$. We always denote the identity operator on these spaces with 1. It should be clear from the context which one is meant. Next we introduce two operators (see also Figure 7.1):

1. $H_j : \ell^2(\mathcal{K}(j+1)) \rightarrow \ell^2(\mathcal{K}(j))$, where $b = H_j\, a$ means that

$$b_k \;=\; \sum_{l \in \mathcal{K}(j+1)} h_{j,k,l}\, a_l\,.$$

2. $G_j : \ell^2(\mathcal{K}(j+1)) \rightarrow \ell^2(\mathcal{M}(j))$, where $c = G_j\, a$ means that

$$c_m \;=\; \sum_{l \in \mathcal{K}(j+1)} g_{j,m,l}\, a_l\,.$$

The operators $\widetilde{H}_j$ and $\widetilde{G}_j$ are defined similarly. We refer to these operators as *filter operators* or sometimes simply as filters.

We can now write the fast wavelet transform in operator notation. Define the sequences $\lambda_j = \{\lambda_{j,k} \mid k\}$ and $\gamma_j = \{\gamma_{j,k} \mid m\}$. Then one step in the forward transform is given by

$$\lambda_j \;=\; \widetilde{H}_j\, \lambda_{j+1} \quad \text{and} \quad \gamma_j \;=\; \widetilde{G}_j\, \lambda_{j+1}\,,$$

and one step in the inverse transform is given by

$$\lambda_{j+1} \;=\; H_j^*\, \lambda_j + G_j^*\, \gamma_j\,.$$

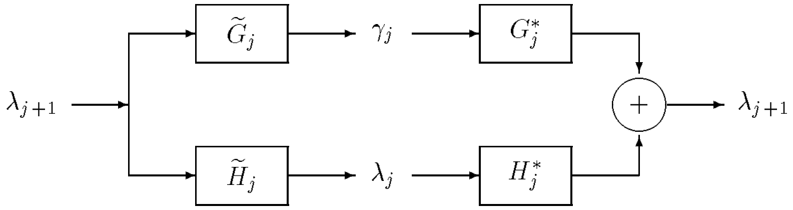

FIG. 7.2. *The fast wavelet transform. The major difference with the first generation fast wavelet transform is that the filters potentially are different for each coefficient. Observe that the subsampling is absorbed into the filters.*

One step of the transform is depicted as a block diagram in Figure 7.2. We use here a scheme similar to a subband transform. Note how the traditional subsampling is absorbed into the filter operators.

The conditions on the filter operators for exact reconstruction now readily follow:

$$\widetilde{H}_j\, H_j^* \;=\; \widetilde{G}_j\, G_j^* \;=\; 1\,, \qquad\qquad \widetilde{G}_j\, H_j^* \;=\; \widetilde{H}_j\, G_j^* \;=\; 0\,,$$

and

$$H_j^*\, \widetilde{H}_j + G_j^*\, \widetilde{G}_j \;=\; 1\,.$$

These we can write in matrix form as

$$\begin{bmatrix} \widetilde{H}_j \\ \widetilde{G}_j \end{bmatrix} \begin{bmatrix} H_j^* & G_j^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} H_j^* & G_j^* \end{bmatrix} \begin{bmatrix} \widetilde{H}_j \\ \widetilde{G}_j \end{bmatrix} = 1\,. \tag{7.1}$$

DEFINITION 7.1. *The set of filter operators $\{H_j, \widetilde{H}_j, G_j, \widetilde{G}_j\}$ is a* ***set of biorthogonal filter operators*** *if condition* (7.1) *is satisfied.*

With slight abuse of notation, i.e. by letting the operators work on sequences of functions, we can write the refinement relations. Define $\varphi_j = \{\varphi_{j,k} \mid k \in \mathcal{K}(j)\}$ and $\psi_j = \{\psi_{j,m} \mid m \in \mathcal{M}(j)\}$. Then

$$\varphi_j \;=\; H_j\, \varphi_{j+1} \quad \text{and} \quad \psi_j \;=\; G_j\, \varphi_{j+1}\,.$$

In the other direction we have

$$\varphi_{j+1} \;=\; \widetilde{H}_j^*\, \varphi_j + \widetilde{G}_j^*\, \psi_j\,.$$

Armed with this operator notation, we now can state the lifting scheme.

8. The lifting scheme.

In this section we state and prove the lifting scheme and show how it can be used to construct second generation wavelets.

THEOREM 8.1 (Lifting). *Take an initial set of biorthogonal filter operators $\{H_j^{\text{old}}, \widetilde{H}_j^{\text{old}}, G_j^{\text{old}}, \widetilde{G}_j^{\text{old}}\}$. Then a new set of biorthogonal filter operators $\{H_j, \widetilde{H}_j, G_j, \widetilde{G}_j\}$ can be found as*

$$\begin{aligned} H_j &= H_j^{\text{old}} \\ \widetilde{H}_j &= \widetilde{H}_j^{\text{old}} + S_j\, \widetilde{G}_j^{\text{old}} \\ G_j &= G_j^{\text{old}} - S_j^*\, H_j^{\text{old}} \\ \widetilde{G}_j &= \widetilde{G}_j^{\text{old}}\,, \end{aligned}$$

where S_j is an operator from $\ell^2(\mathcal{M}(j))$ to $\ell^2(\mathcal{K}(j))$.

Proof. We write the lifting scheme in matrix notation:

$$\begin{bmatrix} \widetilde{H}_j \\ \widetilde{G}_j \end{bmatrix} = \begin{bmatrix} 1 & S \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \widetilde{H}_j^{\text{old}} \\ \widetilde{G}_j^{\text{old}} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} H_j \\ G_j \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -S^* & 1 \end{bmatrix} \begin{bmatrix} H_j^{\text{old}} \\ G_j^{\text{old}} \end{bmatrix}$$

If we think of the biorthogonality conditions (7.1) in the matrix notation, the proof simply follows from the fact that

$$\begin{bmatrix} 1 & S \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -S \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

□

One can use Figure 7.1 to assert that the order of the operators H, G, and S is correct. The theorem in the index notation reads:

Theorem 8.2 (Lifting in index notation). *Take an initial set of biorthogonal filters $\{h^{\text{old}}, \widetilde{h}^{\text{old}}, g^{\text{old}}, \widetilde{g}^{\text{ old}}\}$, then a new set of biorthogonal filters $\{h, \widetilde{h}, g, \widetilde{g}\}$ can be constructed as*

$$\begin{aligned} h_{j,k,l} &= h_{j,k,l}^{\text{old}} \\ \widetilde{h}_{j,k,l} &= \widetilde{h}_{j,k,l}^{\text{old}} + \sum_m s_{j,k,m}\, \widetilde{g}_{j,m,l}^{\text{ old}} \\ g_{j,m,l} &= g_{j,m,l}^{\text{old}} - \sum_k s_{j,k,m}\, h_{j,k,l}^{\text{old}} \\ \widetilde{g}_{j,m,l} &= \widetilde{g}_{j,m,l}^{\text{ old}}\,. \end{aligned}$$

After lifting, the filters h and $\widetilde{g}$ remain the same, while the filters $\widetilde{h}$ and g change. As h remains the same, so do the primal scaling functions. The dual scaling functions and primal wavelets change since $\widetilde{h}$ and g change. The dual wavelets also changes because the dual scaling functions, from which they are built change. However, the coefficients $\widetilde{g}$ of the refinement equation of the dual wavelet remain the same. More precisely we have

$$\begin{aligned} \varphi_j &= \varphi_j^{\text{old}} \\ \widetilde{\varphi}_j &= \widetilde{H}_j^{\text{old}} \widetilde{\varphi}_{j+1} + S_j\, \widetilde{G}_j^{\text{old}}\, \widetilde{\varphi}_{j+1} \;=\; \widetilde{H}_j^{\text{old}} \widetilde{\varphi}_{j+1} + S_j\, \widetilde{\psi}_j \\ \psi_j &= G_j^{\text{old}}\, \varphi_{j+1} - S_j^*\, H_j^{\text{old}} \varphi_{j+1} \;=\; \psi_j^{\text{old}} - S_j\, \varphi_j^{\text{old}} \\ \widetilde{\psi}_j &= \widetilde{G}_j^{\text{old}}\, \widetilde{\varphi}_j\,, \end{aligned}$$

or

$$\varphi_{j,k} = \varphi_{j,k}^{\text{old}}$$

$$\widetilde{\varphi}_{j,k} = \sum_l \widetilde{h}_{j,k,l}^{\text{old}}\, \widetilde{\varphi}_{j+1,l} + \sum_m s_{j,k,m}\, \widetilde{\psi}_{j,m}. \tag{8.1}$$

$$\psi_{j,m} = \psi_{j,m}^{\text{old}} - \sum_k s_{j,k,m}\, \varphi_{j,k}^{\text{old}} \tag{8.2}$$

$$\widetilde{\psi}_{j,m} = \sum_l \widetilde{g}_{j,k,m}^{\text{old}}\, \widetilde{\varphi}_{j+1,l}\,. \tag{8.3}$$

Although formally similar, the expressions in (8.2) and (8.1) are quite different. The difference lies in the fact that in (8.2) the scaling functions on the right-hand side did not change after lifting, while in (8.1) the functions on the right-hand side did change after lifting. Indeed, the dual wavelets on the right-hand side of (8.1) already are the new ones.

The power behind the lifting scheme is that through the operator S we have full control over all wavelets and dual functions that can be built from a particular set of scaling functions. This means we can start from a simple or trivial multiresolution analysis and use (8.2) to choose S so that the wavelets after lifting have particular properties. This allows custom-design of the wavelet and it is the motivation behind the name "lifting scheme."

The fundamental idea behind the lifting scheme is that instead of using scaling functions on the *finer* level to build a wavelet, as in (5.2), we use an old, simple wavelet and scaling functions on the *same* level to synthesize a new wavelet, see (8.2). Thus instead of using "sister" scaling functions, we use "aunt" scaling functions of the family tree to build wavelets. As we will point out later, the "aunt" property is fundamental when building adaptive wavelets. The advantage of using (8.2) as opposed to (5.2) for the construction of $\psi_{j,m}$ is that in the former we have total freedom in the choice of S. Once S is fixed, the lifting scheme assures that all filters are biorthogonal. If we use (5.2) to construct ψ, we would have to check the biorthogonality separately.

Equation (8.2) is also the key to finding the S operator, since functions on the right-hand side do not change. Conditions on $\psi_{j,m}$ thus immediately translate into conditions on S. For example, we can choose S to increase the number of vanishing moments of the wavelet, or choose S so that $\psi_{j,m}$ resembles a particular shape.

If the original filters and S are finite filters, then the new filters will be finite as well. In such case define the (adjoint) sets

$$\mathcal{K}(j,m) \;=\; \{k \mid s_{j,k,m} \neq 0\} \quad \text{and} \quad \mathcal{M}(j,k) \;=\; \{m \mid k \in \mathcal{K}(j,m)\}\,.$$

If we want the wavelet to have vanishing moments, the condition that the integral of a wavelet multiplied with a certain function P_p is zero leads to,

$$\int_X P_p\, \psi_{j,m}\, d\mu \;=\; 0 \;\Rightarrow\; \int_X P_p\, \psi^{\text{old}}_{j,m}\, d\mu \;=\; \sum_{k \in \mathcal{K}(j,m)} s_{j,k,m} \int_X P_p\, \varphi^{\text{old}}_{j,k}\, d\mu\,.$$

For fixed indices j and m, the latter is a linear equation in the unknowns $\{s_{j,k,m} \mid k \in \mathcal{K}(j,m)\}$. All coefficients only depend on the old multiresolution analysis. If we choose the number of unknown coefficients $s_{j,k,m}$ equal to the number of equations, we simply need to solve a linear system for each j and m. Remember that the functions P_p had to be independent, so if the functions $\varphi^{\text{old}}_{j,k}$ are independent as well, the system will be full rank.

Notes:

1. Other constraints than vanishing moments can be used for the choice of S. For example one can custom-design the shape of the wavelet for use in feature recognition. Given the scaling functions, choose S so that $\psi_{j,m}$ resembles the particular feature we want to recognize. The magnitude of the wavelet coefficients is now proportional to how much the original signal at the particular scale and place resembles the feature. This has important applications in automated target recognition and medical imaging. Other ideas are fixing the value of the wavelet or the value of the derivative of the wavelet at a certain location. This is useful to accommodate boundary conditions.

2. In general it is not possible to use lifting to build orthogonal or semi-orthogonal wavelets using only finite lifting filters. In the semi-orthogonal case, the condition that a new wavelet $\psi_{j,m}$ is orthogonal to the V_j typically will require to use all $\varphi_{j,k}$ scaling functions of level j in the lifting (8.2). In [80] this was bypassed by *pseudo-orthogonalization,* a scheme where $\psi_{j,m}$ is only required to be orthogonal to the (interpolating) scaling functions in a certain neighborhood. As mentioned in the introduction, part of the inspiration of the lifting scheme came from generalizing this idea to a fully biorthogonal setting.
3. In [17] several examples, including splines with nonuniform knot sequences are given where semi-orthogonal wavelets are constructed. This construction uses all possible degrees of freedom for the construction of the wavelet, which is more than what lifting allows, but does not lead to finite primal and dual filters.
4. One of the appealing features of using the lifting scheme in the construction of second generation wavelets is that one gets the filters for the scaling functions *and* the wavelets together. Other constructions, such as non-stationary subdivision, only give the filters for the scaling functions, see for example [104, Chapter 5]. One then needs to use a technical trick to find the wavelet filters with the right biorthogonality properties. There is no guarantee that this is always possible.

9. Fast lifted wavelet transform. In this section we show how the lifting scheme can be used to facilitate and accelerate the implementation of the fast wavelet transform. The basic idea is to never explicitly form the new filters, but only work with the old filter, which can be trivial, and the S filter.

For the forward transform we get

$$\lambda_j \;=\; \widetilde{H}_j\,\lambda_{j+1} \;=\; \widetilde{H}_j^{\text{old}}\lambda_{j+1} + S_j\,\gamma_j\,.$$

In index notation this becomes

$$\lambda_{j,k} \;=\; \sum_l \widetilde{h}_{j,k,l}^{\text{old}}\,\lambda_{j+1,l} + \sum_m s_{j,k,m}\,\gamma_{j,m}\,.$$

This implies that if we first calculate the wavelet coefficients γ_j as $\widetilde{G}_j^{\text{old}}\lambda_{j+1}$, we can later *reuse* them in the calculation of the λ_j coefficients. The λ_j are first calculated as $\widetilde{H}_j^{\text{old}}\,\lambda_{j+1}$ and later updated (lifted) with the γ_j coefficients. This way we never have to form the (potentially large) filter $\widetilde{H}_j$. In other words, the lifting scheme makes optimal use of the similarities between the $\widetilde{H}$ and $\widetilde{G}$ filter. This both facilitates and accelerates the implementation.

For the inverse transform we find that

$$\lambda_{j+1} \;=\; H_j^*\,\lambda_j + G_j^*\,\gamma_j \;=\; H_j^{\text{old}*}(\lambda_j - S_j\,\gamma_j) + G_j^{\text{old}*}\gamma_j\,.$$

In index notation this becomes

$$\lambda_{j+1,l} \;=\; \sum_k h_{j,k,l}^{\text{old}}\left(\lambda_{j,k} - \sum_m s_{j,k,m}\gamma_{j,m}\right) + \sum_m g_{j,m,l}^{\text{old}}\gamma_{j,m}\,.$$

The inverse transform thus first undoes the lifting (between the parentheses) and then does an inverse transform with the old filters.

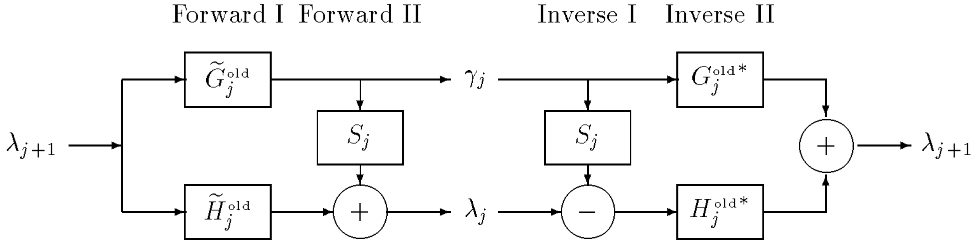

FIG. 9.1. *The fast lifted wavelet transform: The basic idea is to first perform a transform with the old, simple filters and later "lift" the scaling function coefficients with the help of wavelet coefficients. The inverse transform first undoes the lifting and then performs an inverse transform with the old filters*

This leads to the following algorithm for the *fast lifted wavelet transform* depicted in Figure 9.1. On each level the forward transform consists of two stages. Stage I is simply the forward transform with the old filters while stage II is the lifting. In the inverse transform, stage I simply undoes the lifting and stage II is an inverse transform with the old filters. In pseudo code this becomes:

```
Forward wavelet transform
    For j = n-1 downto 0
        Forward I(j)
        Forward II(j)
```

```
Inverse wavelet transform
    For level = 0 to n-1
        Inverse I(j)
        Inverse II(j)
```

`Forward I(j)`: Calculate the $\gamma_{j,m}$ and first stage of $\lambda_{j,k}$

$$\forall k \in \mathcal{K}(j) \; : \; \lambda_{j,k} := \sum_{l \in \widetilde{\mathcal{L}}(j,k)} \widetilde{h}^{\text{old}}_{j,k,l} \, \lambda_{j+1,l}$$

$$\forall m \in \mathcal{M}(j) \; : \; \gamma_{j,m} := \sum_{l \in \widetilde{\mathcal{L}}(j,m)} \widetilde{g}^{\text{old}}_{j,m,l} \, \lambda_{j+1,l}$$

`Forward II(j)`: Lift the $\lambda_{j,k}$ using the $\gamma_{j,m}$ calculated in Stage I

$$\forall k \in \mathcal{K}(j) \; : \; \lambda_{j,k} \mathrel{+}= \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m} \, \gamma_{j,m}$$

`Inverse I(j)`: Undo the lifting

$$\forall k \in \mathcal{K}(j) \; : \; \lambda_{j,k} \mathrel{-}= \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m} \, \gamma_{j,m}$$

`Inverse II(j)`: Calculate the $\lambda_{j+1,l}$ using the $\lambda_{j,k}$ from Stage I:

$$\forall l \in \mathcal{K}(j+1) \; : \; \lambda_{j+1,l} := \sum_{k \in \mathcal{K}(j,l)} h^{\text{old}}_{j,k,l} \, \lambda_{j,k} + \sum_{m \in \mathcal{M}(j,l)} g^{\text{old}}_{j,m,l} \, \gamma_{j,m}$$

As noted in [100], there are always two possibilities to implement these sums. For example, take the sum in the `Forward I` routine. We can either implement this as (after assigning 0 to $\gamma_{j,m}$)

$$\forall m \in \mathcal{M}(j) \, : \, \forall l \in \mathcal{L}(j,m) \, : \, \gamma_{j,m} \mathrel{+}= \widetilde{g}_{j,m,l}^{\,\mathrm{old}} \, \lambda_{j+1,l} \, ,$$

or as

$$\forall l \in \mathcal{K}(j+1) \, : \, \forall m \in \mathcal{M}(j,l) \, : \, \gamma_{j,m} \mathrel{+}= \widetilde{g}_{j,m,l}^{\,\mathrm{old}} \, \lambda_{j+1,l} \, .$$

The first option loops over all m, for each $\gamma_{j,m}$ identifies the $\lambda_{j+1,l}$ that determine its value, then calculates the linear combination and assigns it into $\gamma_{j,m}$. The second option loops over all l, identifies the $\gamma_{j,m}$ which are influenced by $\lambda_{j+1,l}$, and then adds on the right amount to each $\gamma_{j,m}$. Both options are theoretically equivalent, but often one of the two is much easier to implement than the other, see for example [100]. There one of the index sets always contains the same number of elements, while the cardinality of the other can vary depending on the mesh.

10. Cakewalk construction. In this section we discuss how one can iterate the lifting scheme to bootstrap one's way up to a multiresolution analysis with desired properties.

We first introduce the *dual lifting scheme*. The basic idea is the same as for the lifting scheme except that we now leave the dual scaling function and the $\widetilde{H}$ and G filters untouched. The H and $\widetilde{G}$ filters and the dual wavelet, scaling function, and wavelet (by refinement) change. We can use the dual lifting scheme to custom design the dual wavelet. If we denote the operator involved with $\widetilde{S}_j$, the new set of biorthogonal filter operators is given by

$$\begin{aligned} H_j &= H_j^{\mathrm{old}} + \widetilde{S}_j \, G_j^{\mathrm{old}} \\ \widetilde{H}_j &= \widetilde{H}_j^{\mathrm{old}} \\ G_j &= G_j^{\mathrm{old}} \\ \widetilde{G}_j &= \widetilde{G}_j^{\mathrm{old}} - \widetilde{S}_j^* \, \widetilde{H}_j^{\mathrm{old}} \, , \end{aligned}$$

where $\widetilde{S}_j$ is an operator from $\ell^2(\mathcal{M}(j))$ to $\ell^2(\mathcal{K}(j))$. Relationships like (8.2) and (8.1) can be obtained by simply toggling the tildes. In the second stage of the fast wavelet transform, the γ_j coefficients are now lifted with the help of the λ_j coefficients calculated in the first stage.

We now can alternate lifting and dual lifting. For example, after increasing the number of vanishing moments of the wavelet with the lifting scheme, one can use the dual lifting scheme to increase the number of vanishing moments of the dual wavelet. By iterating lifting and dual lifting, one can bootstrap one's way up to a multiresolution analysis with desired properties on primal and dual wavelets. This is the basic idea behind the *cakewalk* construction.

There is one issue that remains to be checked to allow cakewalk constructions. Suppose we first use dual lifting to increase the number of vanishing moments of the dual wavelet. How do we know that this will not be ruined by later lifting? Remember that lifting changes the dual scaling function and thus, by refinement, the dual wavelet. The answer is given by the following theorem.

THEOREM 10.1. *Consider a multiresolution analysis with order N. After lifting, the first N moments of the dual scaling function and dual wavelet do not change.*

Proof. The primal scaling functions do not change after lifting. This means that

$$P_p \;=\; \sum_k \langle\, P_p, \widetilde{\varphi}^{\,\mathrm{old}}_{j,k} \,\rangle \, \varphi_{j,k} \;=\; \sum_k \langle\, P_p, \widetilde{\varphi}_{j,k} \,\rangle \, \varphi_{j,k} \quad \text{for} \quad 0 \leqslant p < N\,.$$

This implies that the first N moments of the dual scaling functions do not change after lifting. Since the coefficients of the refinement relations of the dual wavelets do not change (8.3), neither do their moments. □

Thus lifting does not alter the number of vanishing moments of the dual wavelet obtained by prior lifting.

Suppose we use dual lifting to increase the number of dual vanishing moments from N^{old} to N. This involves solving a linear system of size N, independent of how many vanishing moments the old dual wavelets already had. This means that if we use a cakewalk construction the linear systems to be solved become larger and larger, and so do the S filters. Therefore we present a scheme which allows us to exploit the fact that the dual wavelets already have N^{old} moments and thus only solve a system of size $N - N^{\mathrm{old}}$. The basic idea is to lift an old dual wavelet ($\widetilde{\psi}^{\,\mathrm{old}}_{j,m}$) not with old dual *scaling functions* on the same level ($\widetilde{\varphi}^{\,\mathrm{old}}_{j,k}$), but with old dual *wavelets* on the coarser level ($\widetilde{\psi}^{\,\mathrm{old}}_{j-1,n}$). This leads to a new dual wavelet of the form:

$$\widetilde{\psi}_{j,m} \;=\; \widetilde{\psi}^{\,\mathrm{old}}_{j,m} - \sum_{n \in \mathcal{M}(j-1)} \widetilde{t}_{j,n,m}\, \widetilde{\psi}^{\,\mathrm{old}}_{j-1,n}\,.$$

Here the $\widetilde{t}_{j,n,m}$ are the coefficients of a filter operator $\widetilde{T}_j : \ell^2(\mathcal{M}(j)) \to \ell_2(\mathcal{M}(j-1))$. We always assume that the index n belongs to $\mathcal{M}(j-1)$. Note that the new dual wavelets, independent of $\widetilde{T}$ immediately have at least as many vanishing moments as the old ones (N^{old}). Expressing that the new dual wavelets have N vanishing moments leads to only $N - N^{\mathrm{old}}$ equations in the unknowns $\{\widetilde{t}_{j,n,m} \mid n\}$.

Let us try to find a fast wavelet transform associated with this. In operator notation we have

$$\widetilde{\psi}_j \;=\; \widetilde{\psi}^{\,\mathrm{old}}_j - \widetilde{T}^*_j\, \widetilde{\psi}^{\,\mathrm{old}}_{j-1} \;=\; \widetilde{\psi}^{\,\mathrm{old}}_j - \widetilde{T}^*_j\, \widetilde{G}^{\mathrm{old}}_{j-1}\, \widetilde{\varphi}^{\,\mathrm{old}}_j\,.$$

This construction thus corresponds to letting $\widetilde{S}^*_j = \widetilde{T}^*_j\, \widetilde{G}^{\mathrm{old}}_{j-1}$. The basic idea is to use only the old filters and the filter $\widetilde{T}$ and never construct the $\widetilde{S}$ filter or any of the new filters explicitly. The forward transform takes three stages:

I: Given the sequence λ_{j+1} calculate the forward transform with the old filters: $\lambda_j := \widetilde{H}^{\mathrm{old}}_j\, \lambda_{j+1}$ and $\gamma_j := \widetilde{G}^{\mathrm{old}}_j\, \lambda_{j+1}$.

II: Calculate another level with the old filters: $\lambda_{j-1} := \widetilde{H}^{\mathrm{old}}_{j-1}\, \lambda_j$ and $\gamma_{j-1} := \widetilde{G}^{\mathrm{old}}_{j-1}\, \lambda_j$.

III: Lift the γ_j with the γ_{j-1}: $\gamma_j - = \widetilde{T}^*_j\, \gamma_{j-1}$.

Note that the second stage on level j coincides with the first stage on level $j-1$, see Figure 10.1 for a block diagram. The inverse transform in a first stage undoes the lifting and then applies an inverse transform with the old filters.

We have seen how the lifting scheme can pass between an old and a new multiresolution analysis. To start the construction of second generation wavelets we therefore need an initial multiresolution analysis. In the following sections we will give three examples of an initial multiresolution analysis to start the lifting scheme.

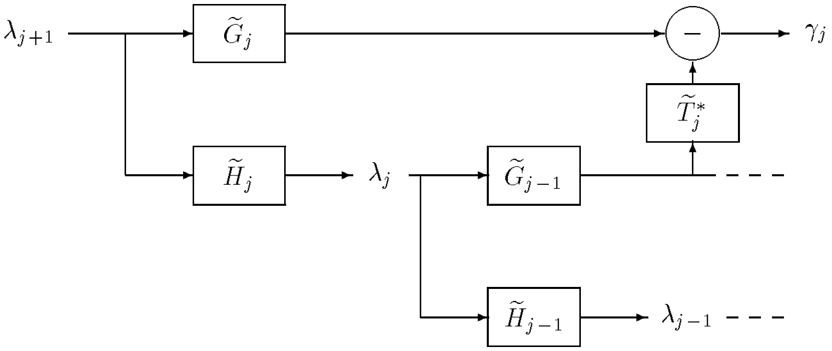

FIG. 10.1. *Part of a cakewalk construction. The basic idea is to lift the wavelet coefficients with wavelet coefficients on the coarser level. This way the fact that the old dual wavelets already have $N^{\rm old}$ vanishing moments can be exploited.*

11. Orthogonal Haar wavelets. In this section we present the generalized orthogonal Haar wavelets, which form a first example of an initial multiresolution analysis to start the lifting scheme. The idea was first introduced by Coifman, Jones, and Semmes for dyadic cubes in [33], generalized for Clifford-valued measures in [9, 91], and later generalized for arbitrary partitionings in [68].

We first introduce the notion of a *nested set of partitionings*.

DEFINITION 11.1. *A set of measurable subsets $\{X_{j,k} \mid j,k\}$ is a* ***nested set of partitionings*** *if it is a set of partitionings and if, for every j and k, $X_{j,k}$ can be written as a finite disjoint union of at least 2 sets $X_{j+1,l}$:*

$$X_{j,k} = \bigcup_{l \in \mathcal{L}(j,k)} X_{j+1,l} \,.$$

Note that because of the partition property ($X_{j,k} \subset X_{j+1,k}$) we have that $k \in \mathcal{L}(j,k)$. Let, according to our normalization (3.2), $\varphi_{j,k} = \chi_{X_{j,k}}$ and $\widetilde{\varphi}_{j,k} = \chi_{X_{j,k}}/\mu(X_{j,k})$. Define the $V_j \subset \mathrm{L}_2$ as

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

The spaces V_j generate a multiresolution analysis of L_2, see e.g. [68] for a proof. As the scaling functions are orthogonal, we let W_j be the orthogonal complement of V_j in V_{j+1} so that $\widetilde{V}_j = V_j$.

Now fix a scaling function $\varphi_{j,k}$. For the construction of the Haar wavelets, we only need to consider the set $X_{j,k}$. First we assume without loss of generality that $\mathcal{L}(j,k)$ contains either 2 or 3 elements. Indeed, if $\mathcal{L}(j,k)$ contains more elements, we can split them into two groups whose numbers of elements differ by at most one. For each group we can introduce (implicitly) a new corresponding $X_{j',k'}$. We can continue to do this until the number of elements is either 2 or 3. In case $\mathcal{L}(j,k) = \{k, m\}$ we let the wavelet $\psi_{j,m}$ be

$$\psi_{j,m} \; = \; \frac{\varphi_{j+1,k}}{2\,\mu(X_{j+1,k})} - \frac{\varphi_{j+1,m}}{2\,\mu(X_{j+1,m})} \,. \tag{11.1}$$

In case $\mathcal{L}(j,k) = \{k, m, m'\}$ we keep $\psi_{j,m}$ as above and let

$$\psi_{j,m'} \; = \; \frac{\varphi_{j+1,k} + \varphi_{j+1,m}}{2\,\mu(X_{j+1,k}) + 2\,\mu(X_{j+1,m})} - \frac{\varphi_{j+1,m'}}{2\,\mu(X_{j+1,m'})} \,.$$

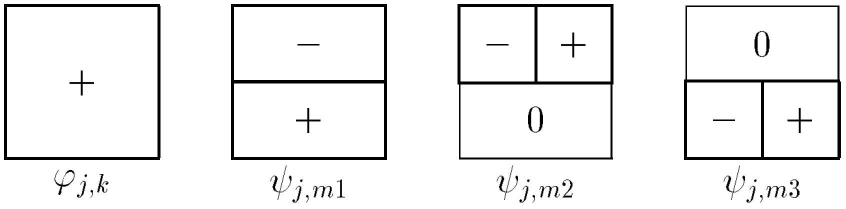

FIG. 11.1. *The generalized orthogonal Haar wavelets for square partitionings. The wavelets are piecewise constant and have a vanishing integral. The sign is indicated in the support. The orthogonality follows immediately from the support and the vanishing integral of the wavelets. Similar constructions apply to arbitrary partitionings.*

In case $\mathcal{L}(j,k) = \{k, m1, m2, m3\}$, we need two stages. Each stage involves two sets and a wavelet of the form (11.1), see Figure 11.1. The Haar wavelets are constructed so that

$$\int_X \psi_{j,m}\, d\mu \;=\; 0 \quad \text{and} \quad \int_X |\psi_{j,m}|\, d\mu \;=\; 1\,.$$

They are orthogonal to $\varphi_{j,k}$ because they have a vanishing integral. Two different wavelets are orthogonal, since either their supports are disjoint or one is constant on the support of the other.

These wavelets form an orthogonal basis for L_2. In fact, they also form an unconditional basis for L_p.

THEOREM 11.2 ([68]). *The generalized orthogonal Haar wavelets $\{\psi_{j,m} \mid j, m\}$ form an unconditional basis for L_p with $1 < p < \infty$, with unconditional basis constant $p^* - 1$, where $1/p + 1/p^* = 1$.*

This construction allows Haar wavelets adapted to the settings G1-G3 mentioned in the introduction. Their advantage is their generality. Their disadvantages are that they are non smooth and that they have only one vanishing moment. However, they form a perfect example of an initial multiresolution analysis to start the lifting scheme with. With the lifting scheme we can build wavelets with more vanishing moments and/or more smoothness.

In the case of the real line and the classical Haar wavelet, the dual lifting scheme corresponds to a technique called *average-interpolation* introduced by David Donoho in [56]. Here $\widetilde{\varphi}$ is the indicator function on $[0,1]$, while φ is constructed through a subdivision scheme which ensures that polynomials up to a certain order can be reproduced with the scaling functions. This condition is precisely the same as the vanishing moment condition of the dual wavelet as used in dual lifting. The average interpolating technique can be generalized to a second generation setting, see e.g. [104] for the construction of weighted wavelets. It generates primal and dual scaling functions which are biorthogonal. However, it is not immediately clear what the wavelets and dual wavelets are. In other words there is no immediate generalization for the Quadrature Mirror Filter construction where one takes $\widetilde{g}_k = (-1)^k\, h_{1-k}$. The dual lifting scheme provides a very simple solution to this problem. Again, the idea is to *first* construct the dual wavelets, and later check what happens to the scaling functions using the cascade algorithm.

12. Interpolating scaling functions and wavelets. In this Section we introduce the Lazy wavelet, another candidate to start the lifting scheme with, which is even simpler than the Haar wavelets. We show how it is connected with interpolating scaling functions.

12.1. The Lazy wavelet. One way to look at the general index sets $\mathcal{K}(j)$ and $\mathcal{M}(j)$ is to think of $\mathcal{K}(j)$ (respectively $\mathcal{M}(j)$) as the generalization of the even (respectively odd) indices. This inspires us to define two *subsampling operators* E (**e**ven) and D (o**d**d) as follows:

$E : \ell^2(\mathcal{K}(j+1)) \to \ell^2(\mathcal{K}(j))$, where $b = E\,a$ means $b_k = a_k$ for $k \in \mathcal{K}(j)$.

$D : \ell^2(\mathcal{K}(j+1)) \to \ell^2(\mathcal{M}(j))$, where $c = D\,a$ means $c_m = a_m$ for $m \in \mathcal{M}(j)$.

Although these operators depend on the level j we will not supply them with an extra subscript, since no confusion is possible. These operators provide a trivial orthogonal splitting, as

$$\begin{bmatrix} E \\ D \end{bmatrix} \begin{bmatrix} E^* & D^* \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} E^* & D^* \end{bmatrix} \begin{bmatrix} E \\ D \end{bmatrix} = 1\,.$$

We can now decompose any operator $W : \ell^2(\mathcal{K}(j)) \to \ell^2(\mathcal{K}(j))$ as

$$W = W_e\,E + W_d\,D\,, \quad \text{with} \quad W_e = W\,E^* \quad \text{and} \quad W_d = W\,D^*\,. \tag{12.1}$$

The filter operators of the *Lazy wavelet* are precisely these subsampling operators

$$H_j^{\text{Lazy}} = \widetilde{H}_j^{\text{Lazy}} = E \quad \text{and} \quad G_j^{\text{Lazy}} = \widetilde{G}_j^{\text{Lazy}} = D\,.$$

The Lazy wavelet transform thus is an orthogonal transform that essentially does nothing. It only resamples the coefficients into two groups each step and thus can be seen as the generalization of the polyphase transform to the second generation setting. However, it is important to consider since it is connected with interpolating scaling functions. The operators E and D are crucial when implementing the lifting scheme. Although they are mathematically trivial, the data structure in the program has to be designed carefully to make them easy to implement. With such a data structure, the implementation of the lifting scheme is straightforward.

Given a set of partitionings, one can formally associate scaling functions and dual scaling functions with the Lazy wavelet. By using the cascade algorithm point wise and respecting the normalization, one can see that $\widetilde{\varphi}_{j,k} = \delta(\cdot - x_k)$ and that $\varphi_{j,k}$ is zero everywhere except at x_k where it is one. Formally they are biorthogonal, but in the L_2 setting, $\widetilde{\varphi}_{j,k}$ does not belong to the space while $\varphi_{j,k}$ is zero. The wavelets and dual wavelets are given by $\psi_{j,m} = \varphi_{j+1,m}$ and $\widetilde{\psi}_{j,m} = \varphi_{j+1,m}$, and $N = \widetilde{N} = 0$.

12.2. Interpolating scaling functions. Next, we generalize the notion of an interpolating scaling function. We first need a *set of interpolation points* $\{x_k \mid j \in \mathcal{J}, k \in \mathcal{K}(j)\}$. Remember that such a set can be defined by a set of partitionings. In the other direction, we can associate a set of partitionings with a set of interpolating points as follows. Assume that

$$\forall k : \inf_{j\in\mathcal{J},\, k'\in\mathcal{K}(j)} d(x_k, x_{k'}) = 0\,,$$

for all k. Then let

$$S_{j,k} = \{x \in X \mid d(x, x_k) < d(x, x_{k'}) \text{ for } k' \in \mathcal{K}(j), k \neq k'\}\,.$$

The sets $S_{j,k}$ are the *Voronoi cells* of the set of points $\{x_k \mid k \in \mathcal{K}(j)\}$.

Definition 12.1. *A set of scaling functions $\{\varphi_{j,k} \mid j,k\}$ is* ***interpolating*** *if a set of interpolation points x_k exists, so that $\varphi_{j,k}(x_{k'}) = \delta_{k,k'}$ for $k, k' \in \mathcal{K}(j)$.*

As in the first generation case, the interpolating property can be characterized by means of the coefficients of the refinement relation. We state and prove the result in the index notation.

LEMMA 12.2. *If a set of second generation scaling functions is interpolating, then*

$$\forall\, k, k' \in \mathcal{K}(j) : h_{j,k,k'} \;=\; \delta_{k,k'}\,. \tag{12.2}$$

Proof.

$$\delta_{k,k'} = \varphi_{j,k}(x_{k'}) = \sum_m h_{j,k,l}\,\varphi_{j+1,l}(x_{k'}) = \sum_m h_{j,k,l}\,\delta_{l,k'} \;=\; h_{j,k,k'}\,.$$

□

Note that this lemma can be seen as a special case of Remark 4.2 in [41]. A filter h is called an *interpolating filter* if condition (12.2) holds. This condition can be written in operator notation as

$$H_j^{\text{int}}\,E^* \;=\; 1\,.$$

Note that this is the generalization of an *à trous* filter in the first generation case.

If we have an interpolating scaling function, we can always take Dirac functions as a formal dual

$$\widetilde{\varphi}_{j,k}^{\text{int}} \;=\; \delta(\cdot - x_k)\,.$$

The biorthogonality follows immediately from the interpolation property. The filter corresponding to the dual scaling function is

$$\widetilde{H}^{\text{int}} \;=\; E\,.$$

Now define $\widetilde{S}_j$ as $H_j^{\text{int}}\,D^*$. Then it follows from (12.1) that any interpolating filter can be written as $H_j^{\text{int}} = E + \widetilde{S}_j\,D$. But this expression can be seen as the result of applying the dual lifting scheme to the Lazy wavelet. We can then write a set of biorthogonal filters as

$$\begin{aligned} H_j^{\text{int}} &= E + \widetilde{S}_j\,D \\ \widetilde{H}_j^{\text{int}} &= E \\ G_j^{\text{int}} &= D \\ \widetilde{G}_j^{\text{int}} &= D - \widetilde{S}_j^*\,E\,. \end{aligned}$$

We have thus shown the following theorem.

THEOREM 12.3. *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.*

In index notation the filters become

$$\begin{aligned} h_{j,k,l}^{\text{int}} &= \begin{cases} \delta_{k,l} & \text{if } l \in \mathcal{K}(j) \\ \widetilde{s}_{j,k,l} & \text{if } l \in \mathcal{M}(j) \end{cases} \\ \widetilde{h}_{j,k,l}^{\text{int}} &= \delta_{k,l} \\ g_{j,m,l}^{\text{int}} &= \delta_{m,l} \\ \widetilde{g}_{j,m,l}^{\text{int}} &= \begin{cases} -\widetilde{s}_{j,l,m} & \text{if } l \in \mathcal{K}(j) \\ \delta_{m,l} & \text{if } l \in \mathcal{M}(j)\,. \end{cases} \end{aligned}$$

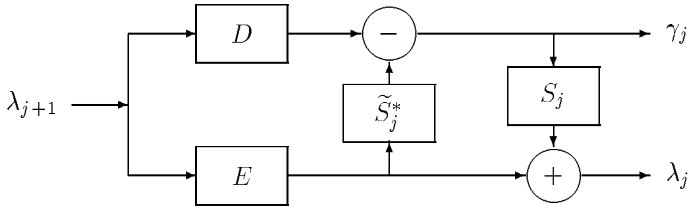

FIG. 12.1. *The fast wavelet transform for wavelets built from interpolating scaling functions. First apply a Lazy wavelet transform, then a dual lifting, and finally a regular lifting.*

Formally the dual wavelets are given by

$$\widetilde{\psi}_{j,m} \;=\; \delta(\cdot - x_m) - \sum_k h_{j,k,m}\delta(\cdot - x_k)\,.$$

The primal wavelets are $\psi_{j,m} \;=\; \varphi_{j+1,m}$. We have $\widetilde{N} = 0$ and N possibly > 0. These filters do not correspond to a multiresolution analysis of L_2, as the dual functions are Dirac distributions which do not even belong to L_2. In the case of linear interpolation, this examples corresponds to what is known in finite elements as "hierarchical basis functions" [116].

We next apply the lifting scheme to find wavelets which have $\widetilde{N} > 0$. This leads to new filters of the form

$$\begin{aligned}
H_j &= H_j^{\text{int}} \;=\; E + \widetilde{S}_j\, D\\
\widetilde{H}_j &= \widetilde{H}_j^{\text{int}} + S_j\, \widetilde{G}_j^{\text{int}} \;=\; (1 - S_j\, \widetilde{S}_j^*)\, E + S_j\, D\\
G_j &= G_j^{\text{int}} - S_j^*\, H_j^{\text{int}} \;=\; -S_j^*\, E + (1 - S_j^*\, \widetilde{S}_j)\, D\\
\widetilde{G}_j &= \widetilde{G}_j^{\text{int}} \;=\; -\widetilde{S}_j^*\, E + D\,.
\end{aligned}$$

This can be verified using Figure 12.1. For example, to find $\widetilde{H}_j$, follow the paths from λ_{j+1} to λ_j. There are three: one direct through E, one through D and then down through S_j, and one through E then up through $\widetilde{S}_j^*$ and down through S_j. Consequently $\widetilde{H}_j = (1 - S_j\, \widetilde{S}_j^*)\, E + S_j\, D$. In index notation this becomes

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

The new wavelet can be written as

$$\psi_{j,m} \;=\; \varphi_{j+1,m} - \sum_{k\in\mathcal{K}(j,m)} s_{j,m,k}\, \varphi_{j,k}. \tag{12.3}$$

One can find the $s_{j,k,m}$ in the same way as described above.

12.3. Algorithm. The algorithm for the wavelet transform associated with the wavelets constructed in the previous section consists of three stages. First a Lazy wavelet transform, then a dual lifting and finally a primal lifting, see Figure 12.1. The inverse transform can be derived immediately by simply inverting each step of the forward transform.

```
Forward(j):

    $\forall k \in \mathcal{K}(j): \quad \lambda_{j,k} := \lambda_{j+1,k}$
    $\forall m \in \mathcal{M}(j): \quad \gamma_{j,m} := \lambda_{j+1,m}$
    $\forall m \in \mathcal{M}(j): \quad \gamma_{j,m} \mathrel{-}= \sum_{k \in \widetilde{\mathcal{K}}(j,m)} \widetilde{s}_{j,k,m}\, \lambda_{j,k}$
    $\forall k \in \mathcal{K}(j): \quad \lambda_{j,k} \mathrel{+}= \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m}\, \gamma_{j,m}$
```

```
Inverse(j):

    $\forall k \in \mathcal{K}(j): \quad \lambda_{j,k} \mathrel{-}= \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m}\, \gamma_{j,m}$
    $\forall m \in \mathcal{M}(j): \quad \gamma_{j,m} \mathrel{+}= \sum_{k \in \widetilde{\mathcal{K}}(j,m)} \widetilde{s}_{j,k,m}\, \lambda_{j,k}$
    $\forall m \in \mathcal{M}(j): \quad \lambda_{j+1,m} := \gamma_{j,m}$
    $\forall k \in \mathcal{K}(j): \quad \lambda_{j+1,k} := \lambda_{j,k}$
```

One of the nice properties of the fast lifted wavelet transform is that all calculations can be done in-place, i.e., without auxiliary memory. It is sufficient to provide storage locations only for the coefficients $\lambda_{n,k}$ of the finest levels. No additional auxiliary memory is needed. A coefficient $\lambda_{j,k}$ with $j < n$ can be stored in the same location as $\lambda_{n,k}$, while a wavelet coefficient $\gamma_{j,m}$ with $j < n$ can be stored in the same location as $\lambda_{n,m}$. The Lazy wavelet transform now simply requires blinking your eyes. Lifting will only require updates with local neighboring coefficients (typically $+=$ or $-=$ operators in the implementation) and thus does not need extra storage.

13. Biorthogonal Haar wavelets. In this section we introduce a third example of an initial multiresolution analysis: the biorthogonal Haar wavelets. They were first used in triangular subdivision in [99]. On triangles, biorthogonal Haar wavelets have more symmetry than orthogonal Haar wavelets. We here show how the biorthogonal Haar wavelets themselves can be seen as a result of lifting from the Lazy wavelet.

Take a set of nested partitionings $X_{j,k}$. Note that this defines the index sets $\mathcal{L}(j,k)$. Consider the Lazy scaling function and wavelet,

$$\widetilde{\varphi}^{\text{Lazy}}_{j,k} = \delta(\cdot - x_k) \quad \text{and} \quad \widetilde{\psi}^{\text{Lazy}}_{j,m} = \delta(\cdot - x_m)\,.$$

Let us first apply dual lifting and denote the resulting functions with a superscript (1). Fix a $k^* \in \mathcal{K}(j)$ and let $\mathcal{M}(j,k^*) = \mathcal{L}(j,k^*) \backslash \{k^*\}$. In order for the new wavelet $\widetilde{\psi}^{(1)}_{j,m}$ with $m \in \mathcal{M}(j,k^*)$ to have one vanishing moment, we let

$$\widetilde{\psi}^{(1)}_{j,m} = \delta(\cdot - x_m) - \delta(\cdot - x_{k^*}) = \widetilde{\varphi}^{(1)}_{j+1,m} - \widetilde{\varphi}^{(1)}_{j+1,k^*}$$

so that $\mathcal{K}(j,m) = \{k^*\}$ and $\widetilde{s}_{j,k,m} = \delta_{k,k^*}$. This implies that the scaling function satisfies

$$\varphi^{(1)}_{j,k^*} = \varphi^{(1)}_{j+1,k^*} + \sum_{m \in \mathcal{M}(j,k^*)} \widetilde{s}_{j,k,m}\, \varphi^{(1)}_{j+1,m} = \sum_{l \in \mathcal{L}(j,k^*)} \varphi^{(1)}_{j+1,l} \tag{13.1}$$

which yields that $\varphi^{(1)}_{j,k} = \chi_{X_{j,k}}$ and thus $\psi^{(1)}_{j,m} = \chi_{X_{j+1,m}}$. We now have $N = 1$ and $\widetilde{N} = 0$ and could call this a *half Haar basis*. Note that this half Haar wavelet is used

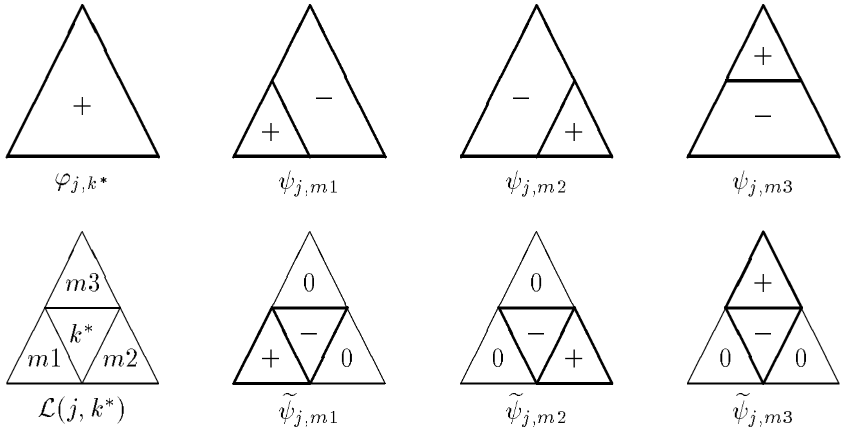

FIG. 13.1. *The biorthogonal Haar wavelets on triangles. Biorthogonality follows from the support and the vanishing integral of wavelets and dual wavelets. On triangles the biorthogonal Haar wavelets are more symmetric than the orthogonal Haar. This is another example to start the lifting scheme with.*

in the interlaced GIF format which is currently quite popular on the World Wide Web.

Next we use lifting to obtain a primal wavelet with a vanishing moment. We choose

$$\psi_{j,m} = \psi^{(1)}_{j,m} - s_{j,k^*,m}\,\varphi^{(1)}_{j,k*} \qquad \text{with} \qquad k^* \in \mathcal{K}(j,m)\,,$$

where $s_{j,k,m} = \mu(X_{j+1,m})/\mu(X_{j,k})$ if $m \in \mathcal{M}(j,k)$ and zero otherwise. In this way $\psi_{j,m}$ has one vanishing moment. The new dual scaling function becomes:

$$\begin{aligned}
\widetilde{\varphi}_{j,k} &= \widetilde{\varphi}_{j+1,k} + \sum_{m\in\mathcal{M}(j,k)} \mu(X_{j+1,m})/\mu(X_{j,k})\,\widetilde{\psi}_{j,m} \\
&= \widetilde{\varphi}_{j+1,k} + \sum_{m\in\mathcal{M}(j,k)} \mu(X_{j+1,m})/\mu(X_{j,k})\,(\widetilde{\varphi}_{j+1,m} - \widetilde{\varphi}_{j+1,k}) \quad \text{(see (13.1))} \\
&= \sum_{m\in\mathcal{M}(j,k)} \mu(X_{j+1,m})/\mu(X_{j,k})\,\widetilde{\varphi}_{j+1,m} + \\
&\qquad\qquad \left(1 - \sum_{m\in\mathcal{M}(j,k)} \mu(X_{j+1,m})/\mu(X_{j,k})\right)\,\widetilde{\varphi}_{j+1,k} \\
&= \sum_{l\in\mathcal{L}(j,k)} \mu(X_{j+1,l})/\mu(X_{j,k})\,\widetilde{\varphi}_{j+1,l} \\
&= \chi_{X_{j,k}}/\mu(X_{j,k})\,.
\end{aligned}$$

Summarizing we have the following basis functions, which generate the *biorthogonal Haar* multiresolution analysis:

$$\begin{aligned}
\varphi_{j,k} &= \chi_{X_{j,k}} \\
\widetilde{\varphi}_{j,k} &= \chi_{X_{j,k}}/\mu(X_{j,k}) \\
\psi_{j,m} &= \varphi_{j+1,m} - \mu(X_{j+1,m})/\mu(X_{j,k^*})\,\varphi_{j,k^*} \qquad \text{with} \qquad \{k^*\} = \mathcal{K}(j,m) \\
\widetilde{\psi}_{j,m} &= \widetilde{\varphi}_{j+1,m} - \widetilde{\varphi}_{j+1,k^*}\,.
\end{aligned}$$

Figure 13.1 shows the biorthogonal Haar wavelets for a triangular partitioning. Given that the scaling function and dual scaling function are multiples of each other, we actually have a semi-orthogonal setting. This means that the V_j and $\widetilde{V}_j$ spaces coincide. Consequently wavelets on different levels are orthogonal, but within one level the wavelets are not orthogonal. The biorthogonal Haar multiresolution analysis is another example of an initial multiresolution analysis with $N = \widetilde{N} = 1$. We here actually showed how it can be constructed by twice lifting the Lazy wavelet.

The algorithm for the biorthogonal Haar transform is given below. Again all calculations can be done in-place.

```
Forward(j):

    $\forall m \in \mathcal{M}(j):\quad \gamma_{j,m} := \lambda_{j+1,m} - \lambda_{j+1,k} \qquad (\{k^*\} = \mathcal{K}(j,m))$
    $\forall k \in \mathcal{K}(j):\quad \lambda_{j,k} := \lambda_{j+1,k} + \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m}\, \gamma_{j,m}$

Inverse(j):

    $\forall k \in \mathcal{K}(j):\quad \lambda_{j+1,k} := \lambda_{j,k} - \sum_{m \in \mathcal{M}(j,k)} s_{j,k,m}\, \gamma_{j,m}$
    $\forall m \in \mathcal{M}(j):\quad \lambda_{j+1,m} := \gamma_{j,m} + \lambda_{j+1,k} \qquad (\{k^*\} = \mathcal{K}(j,m))$
```

14. Applications and future research. Now that we understand the machinery of the lifting scheme, we can start to apply it in the settings described in the introduction. We discuss a few cases in more detail.

14.1. Wavelets on an interval. As pointed out in the introduction, many wavelet constructions on the interval already exist. They all involve modifying the wavelets and scaling functions close to the end point of the interval, which leads to special boundary filters. The derivation of the boundary filters is actually quite technical and it is not immediately clear to the user why they work. With the aid of the lifting scheme, the construction of interval wavelets and the implementation of the associated transform become much more transparent. The Haar and Lazy wavelet can be trivially defined on the interval. Lifting then only requires pulling in the right aunts (scaling functions on the coarser level) at the boundary of the interval. All calculations can be done in-place. For details we refer to [106].

A software package, `LIFTPACK`, to calculate the wavelet transformation of images is currently available [64]. Its properties are: in-place calculation, correct treatment of boundaries, arbitrary size images (not only powers of two), and a faster implementation of existing biorthogonal wavelet filters (speedup can be a factor of two).

14.2. Weighted wavelets. Let X be $\mathbf{R}$ and consider the weight function $w(x) = d\mu/dx$, where dx stands for the Lebesgue measure. The wavelets constructed with the lifting scheme are orthogonal with respect to a weighted inner product, where $w(x)$ is the weight function. We refer to them as weighted wavelets. They are useful for the approximation of functions with singularities. If a function f contains a singularity, then the approximation with first generation wavelets will be slow, independent of the number of dual vanishing moments N. If we can now choose a weight function w so that $w \cdot f$ is a smooth function, then the approximation with weighted wavelets will be again of the order of the number of vanishing moments. An example of this behavior is given in [106].

Weighted wavelets are also useful in the solution of boundary value ODEs, see [75, 104]. If the operator is of the form $-DaD$, then operator wavelets defined as the anti-derivative of weighted wavelets with weight function $w(x) = \sqrt{a(x)}$ diagonalize the operator. The solution algorithm is thus simply a forward and inverse wavelet transform. Future research involves the incorporation of the operator wavelets construction directly into the lifting scheme.

14.3. Wavelets on curves, surfaces, and manifolds. The only thing needed to construct wavelets on manifolds is either a set of interpolating points to define a Lazy wavelet or a set of nested partitionings to define Haar wavelets. Lifting will take care of the rest. The resulting wavelets are defined intrinsically on the manifold and do not depend on any parameterization or atlas.

In [99] the lifting scheme is used to construct wavelets on a sphere. Partitionings of the sphere were obtained by starting from a Platonic solid and alternating triangular subdivision and projection out to the sphere. This is known as a geodesic sphere construction. The Lazy wavelet is the starting point for a family of vertex-based wavelets, while the biorthogonal Haar wavelets lead to a family of face-based wavelets. In [100] these wavelets were used for the processing of spherical images. Current research involves the generalization of the construction and the applications to arbitrary surfaces.

14.4. Adaptive wavelets. The idea of adaptive wavelets was introduced in [69, 97, 98] in the context of the numerical solution of integral equations for illumination computations. The idea is the following. Assume the solution can be approximated with sufficient accuracy in a linear space V_n of dimension M. We know that out of the M^2 matrix entries representing the integral operator in the wavelet basis, only a fraction $\mathcal{O}(M)$ is relevant. If we have these entries, solving the matrix equation can be done in linear $(\mathcal{O}(M))$ time.

However, calculating all wavelet coefficients of the kernel from the finest level n to the coarsest level 0 with the fast wavelet transform requires $\mathcal{O}(M^2)$ operations and is thus a waste of CPU time and memory. Indeed the majority of all computations and memory use will be in vain. If we want an algorithm with linear complexity we can only afford to calculate the wavelets coefficients which we actually need or a slightly larger set.

Gortler, Schröder, et al. achieve this with the use of an *oracle* function. This function predicts, in a conservative fashion and based upon knowledge of the kernel of the integral equation, which wavelet coefficients need to be calculated. They were able to implement this with the use of what they call *tree wavelets*. Tree wavelets have the property that each wavelet of level j is supported within the support of only one scaling function of level j. Haar wavelets and Alpert wavelets [4, 5, 6] have this property. The advantage is that subdividing the support of a scaling function on level j, and thus constructing the wavelets of level j associated with it, does not imply subdividing any other support sets on level j. This way they can calculate the wavelet coefficients from the *coarsest* level to the *finest* level, thereby only subdividing (adding wavelets) where the oracle tells them to.

With traditional (non-tree) wavelets, subdividing a support set S on level j and constructing the wavelets associated with it (whose support may reach outside of S) will imply subdividing a neighboring set of S and dragging in the wavelets associated with that set. This process cascades out and would imply subdividing the *whole* level j and thus makes adaptive constructions awkward.

Tree wavelets are discontinuous and this is a drawback in many applications. As shown in [99], lifting provides a solution here. Indeed, because of the "aunt" property, subdividing a set S on level j only requires its neighbors to exist, but not necessarily requires them to be subdivided as well. The mesh only needs to satisfy a restriction criterion in the sense that neighboring sets only differ by at most one level. This does not cascade out. Lifting thus opens the door to smooth adaptive wavelets. Current research involves the incorporation of these wavelets in illumination computations.

A word of caution is needed here. In [46, 96] it is shown that adaptive wavelet algorithms require wavelets on manifolds satisfying specific conditions concerning stability, regularity, and norm equivalence. As pointed out earlier, lifting does not guarantee these conditions and they have to be verified in each particular case.

14.5. Recursive wavelets. The principle of *recursive wavelets* is explained in [8, 56]. The basic idea is not to use the cascade algorithm ad infinitum to construct the scaling functions, but instead fix the scaling functions on an arbitrary finest level n. This can be generalized easily to the second generation case. Consider a set of partitionings and let

$$\varphi_{n,k} \;=\; \chi_{S_{n,k}} \quad \text{for} \quad k \in \mathcal{K}(n)\,.$$

Next consider a general filter h (not necessarily a Haar filter) and define the scaling functions on the coarser level ($\varphi_{j,k}$ with $j < n$) through recursive applications of the refinement relation (4.1). By definition all scaling functions $\varphi_{j,k}$ are piecewise constant on the sets $\{S_{n,k} \mid k \in \mathcal{K}(n)\}$. This is precisely the advantage of recursive wavelets; no need to go through an infinite limit process to find the scaling functions, instead apply the refinement relation a finite number of times.

One can choose other functions than indicator functions as scaling functions on the finest level. The advantage of indicator functions is their generality, the disadvantage is that they are not smooth. If the topology admits it, smoother choices are piecewise linear (hat) functions or B/box-splines.

In the setting of recursive wavelets, there are L_2 functions associated with the Lazy wavelet. Indeed

$$\psi^{\text{Lazy}}_{j,m} \;=\; \varphi^{\text{Lazy}}_{j+1,m} \;=\; \varphi^{\text{Lazy}}_{n,m} \;=\; \chi_{S_{n,m}}\,.$$

In this paper, we have always assumed that the measure is non-atomic. This restriction, however, is not fundamental. Recursive wavelets allow for atomic measures. Indeed, there is no reason why any of the subsets of $S_{n,k}$ should be measurable. As is shown in [68], it is possible to build Haar wavelets on fully discrete sets such as the integers, or on sets which are of mixed continuous/discrete nature. Also the lifting scheme remains valid.

The idea of recursive wavelets can also be combined with the idea of adaptive wavelets of the previous section. Instead of fixing the scaling function on one finest level n, one can let the notion of finest level depend on the location. Indeed, the oracle of the previous subsection typically leads to finer subdivisions in certain locations and coarser subdivisions in other. The finest level $n(k)$ thus depends on the location k. We then fix the scaling functions

$$\varphi_{n(k),k} \;=\; \chi_{S_{n(k),k}}\,,$$

where

$$X = \bigcup_k S_{n(k),k}\,.$$

In other words, we are using an adaptive mesh.

It is important to note that even though recursive wavelets only use a finite number of levels, the stability issue does not go away but rather manifests itself as a problem concerning ill-conditioning.

14.6. Wavelet packets. *Wavelet packets* were introduced by Coifman, Meyer, and Wickerhauser [34, 35, 114, 115]. The idea is to also further split the W_j spaces with the help of the h and g filters. This way one obtains a better frequency localization. The splitting leads to a full binary tree of wavelet packets, which form a redundant set. For a given function, one can choose the *best-basis* with respect to a criterion such as the entropy of the basis coefficients. A fast tree algorithm to find the best basis was introduced in [36], see also [115].

This idea again carries over into the second generation setting and can be combined with lifting. The conditions for exact reconstruction have exactly the same algebraic structure as in the wavelet case. One can start with defining a Lazy wavelet packet or a generalized Haar wavelet packet (which could be called generalized Walsh functions). A new wavelet packet is now defined as an old wavelet packet plus a linear combination of wavelet packets that live on a coarser level. From a practical point of view, one extra index comes in, and proper data structures need to be designed to incorporate the new filters.

14.7. M-band wavelets. The idea of *M-band wavelets*, or *p-adic wavelets*, a name more common in the mathematical literature, is to split a space V_{j+1} into M (as opposed to 2) subspaces: $V_j \oplus W_j^1 \oplus \cdots \oplus W_j^{M-1}$. For each subspace a different filter is used. Several constructions were introduced in [72, 81, 102, 109]. In some sense the second generation wavelet setting already incorporates this. Indeed, it even allows for different filters for each individual wavelet. However, thinking of lifting combined with M-band wavelets can lead to new constructions. Let us start with the Lazy wavelet. An M-band Lazy wavelet is easily defined and again is the standard polyphase transform [109]. Now one can define a new wavelet as an old wavelet plus a linear combination of scaling functions on the coarser level. This would be ordinary lifting. But we could also define a new wavelet as an old wavelet plus a linear combination of scaling functions on a coarser level plus *wavelets belonging to another (lower index) subband.* This allows more flexibility in the construction. It requires that the M subband are calculated in ascending index order in the transform. In the extreme case one can let each subband contain precisely one wavelet. A new wavelet is now an old wavelet plus previously constructed new wavelets. This requires an ordering within the wavelets of one level. This way we can construct non compactly supported wavelets with only finite filters.

14.8. Non separable wavelets in $\mathbf{R}^n$. As mentioned earlier, the lifting scheme also leads to new insights in the construction of first generation wavelets. This was shown in the one-dimensional case in [105]. Higher-dimensional wavelets can always be constructed using tensor products, but this leads to severe axial directional dependencies. Instead one prefers to work with non separable wavelets which have more axial symmetry and which do not necessarily use a product lattice, see for example [22, 28, 32, 76, 77, 89, 90, 92, 93].

Here too lifting can help. Each lattice allows for the immediate definition of a Lazy wavelet or a Haar wavelet, either 2-band or M-band. Polynomial cancelation then leads to the filter coefficients. Whether lifting will actually lead to new wavelets in this context or rather provide faster implementations of already existing filters as in the one-dimensional case remains a topic for further study.

14.9. Wavelets on bounded domains and wavelet probing. One of the important application domains of wavelets is the solution of partial differential equations. In [44] it is shown that one can use wavelets to build multilevel preconditioners which result in stiffness matrices with uniformly bounded condition numbers. This leads to linear solution algorithms. To solve real life problems one needs wavelets constructed on non-smooth (Lipschitz) domains in $\mathbf{R}^n$. In [27, 74] such a construction is presented. In both cases, tensor product wavelets are used in the interior of the domain while at the boundary specially adapted wavelets are constructed. In this sense these constructions are the natural generalization of the interval constructions to higher dimensions. With the lifting scheme one can build non separable wavelets adapted to general domains. Again the only thing needed is a set of partitionings. One nice property here is that lifting allows for adaptive meshes.

Already in the simple case of the Laplace equation and a non-smooth domain it was recently shown that one cannot obtain an $\mathcal{O}(M^{-2})$ accuracy (where M is the number of elements) unless one uses non-linear approximation [39]. The underlying reason is that the solution does not belong to the second order Sobolev space but rather to a second order Besov space. In other words, one has to use adaptive grids to obtain the correct convergence order.

There is another important application of wavelets on domains. It is a technique called *wavelet probing* introduced independently in [59] and [8, 53]. Let us discuss the idea first on the real line. Consider a function which is smooth except for jump discontinuities at isolated points. We know that the decay of the wavelet coefficients is fast away from the jumps, and slow in the neighborhood of the jumps. Increasing the number of dual vanishing moments leads to faster decay away from the jumps but also to a larger set of coefficients affected by the jumps because of the larger support. It is thus not possible to obtain more efficient approximations and thus better compression by simply increasing the number of dual vanishing moments.

Suppose now we know the location of the jumps. If we use interval wavelets on each interval between two jumps, and thus segment the signal accordingly, we would get fast convergence *everywhere*. Wavelet probing is a technique which allows us to locate the jumps. It simply tries every location between two samples and checks whether it would pay off to segment at this location. The pay-off can be measured e.g. with the entropy of the wavelet coefficients. Probing one location only requires altering $\log M$ coefficients where M is the number of samples. The whole algorithm thus has a complexity of $M \log M$.

Wavelet probing has important applications in image compression. Indeed it allows quick localization of the edges in an image, and then builds wavelets on the segments defined by those edges. Considerably higher compression ratios can be obtained this way. Wavelet probing thus provides an alternative for the zero crossing representation introduced by Mallat and Zhong [82, 83]. The advantage is that no iterative reconstruction is needed. Wavelet probing interacts easily with lifting and the adaptive wavelets mentioned above.

14.10. Wavelets adapted to irregular samples. As we mentioned in the introduction, one of the motivation for the generalization to second generation wavelets, was the processing of irregularly sampled data. Let us discuss this first on the real line. Assume we are given irregular samples of a function $f(x)$: $\lambda_{n,k} = f(x_k)$ with $k \in \mathcal{K}(n)$. We first need to define a Lazy wavelet, i.e. we need the sets $\mathcal{K}(j)$ with $j < n$ (the coarser levels) and the sets $\mathcal{M}(j)$ with $j \geqslant n$, together with the x_m for

$m \in \mathcal{M}(j)$ (the finer levels). For the coarser levels we only need to decide *which* sample locations to retain, while for the finer levels we also have to decide *where* to put the new locations. Coarser levels are needed in the wavelet transform, finer in the cascade algorithm. A simple strategy, which was used in [106] is to retain every other sample on the coarser level and put a new sample in the middle of two old samples on the finer levels. Once a Lazy wavelet is defined, dual lifting provides interpolating scaling functions, and lifting yields wavelets with vanishing moments. The dual lifting can be seen as an instance of irregular Deslauriers-Dubuc subdivision [54, 55]. Current research involves the study of more advanced choices for the Lazy wavelet. One of the strategies is to choose the sample locations so that the ratio of the largest versus the smallest interval of a level becomes closer and closer to one on both the finer ($j > n$) and the coarser ($j < n$) levels. Future research also involves the study of these schemes in higher dimensions.

14.11. Integer to integer wavelet transforms. In [16] lifting is used to build reversible wavelets which map integers to integers for applications to lossless image coding. The idea is to introduce a non-linear round-off in each lifting step. This way the result is guaranteed to be integer while the lifting assure that the transform is invertible. The exact same idea works in the second generation setting and using this in second generation compression applications is another line of future research.

14.12. Conclusion. In this paper we presented the lifting scheme, a construction tool for wavelets adapted to general settings. We showed how one can start from a trivial multiresolution analysis and use lifting to work one's way up to a multiresolution analysis with particular properties. As we mentioned in the introduction, the lifting scheme provides an answer to the algebraic phase of a wavelet construction. For each of the applications mentioned in this section, one still has to verify whether the cascade algorithm converges, whether the resulting wavelets form a Riesz basis (analytic phase), and what their smoothness is (geometric phase).

Note. As mentioned in the introduction, we learned after finishing this work that Wolfgang Dahmen and collaborators independently obtained a construction of multiscale bases with a technique very similar to lifting [17]. We here go into more detail comparing the two approaches. Recall from Theorem 8.1 that the lifting involved an operator which can be written in matrix location as

$$\begin{bmatrix} 1 & S \\ 0 & 1 \end{bmatrix}.$$

The main advantage of this operator is that, independent of the choice of S, it is guaranteed to be invertible and the inverse can be found by flipping the sign on S. In the setting of [17] a more general operator of the form

$$\begin{bmatrix} 1 & S \\ 0 & K \end{bmatrix}$$

is used (which for $K = I$ becomes lifting). This operator is invertible if and only if K is invertible. The inverse is then given by

$$\begin{bmatrix} 1 & -S\,K^{-1} \\ 0 & K^{-1} \end{bmatrix}.$$

This setting is more general and allows to explore *all* degrees of freedom one has to generate new biorthogonal filters. In Section 3.3 of [17] a $K \neq I$ is construced which generates orthogonal decompositions. Section 4.1 of [17] discusses the case of compactly supported semi-orthogonal spline wavelets on irregular knot sequences. However, these settings do not allow both K and K^{-1} to be sparse.

There are certain advantages to the lifting ($K = I$) approach. One problem with the more general approach ($K \neq I$) is that it involves taking the inverse of K and does not guarantee that all primal and dual filters are finite. Infinitely supported filters are less useful practically and do not necessarily lead to fast transforms. Moreover K^{-1} might be difficult to compute numerically. Many of the attractive features of lifting such as in-place computation, no need for inverting operators, and adaptive transforms using “aunt” functions disappear when allowing a general $K \neq I$.

In the first generation setting it was shown recently [52] that in case of finite filters no generality is lost when restricting oneself to the lifting setting: all finite filters can be obtained using multiple alternate primal and dual lifting steps.

On several occasions in this paper we mentioned that lifting does not guarantee stable bases or convergence of the associated subdivision scheme. In fact, many of these issues have been addressed carefully in [17, 41, 42] and we refer to those papers for details.

Acknowledgement. While this paper was being written (spring 1995), the author worked in close collaboration with Peter Schröder on the application of the lifting scheme to the construction of spherical wavelets [99, 100] and on a tutorial paper [106]. The feedback from the numerous and stimulating discussions with Peter was a tremendous help in writing this paper and has led to several new insights. Peter also helped improving the exposition in this paper considerably. Also special thanks to Jelena Kovačević for pointing out the connection between the Lazy wavelet and the polyphase transform, and Thomas Barregren, Maurits Malfait, and Geert Uytterhoeven for carefully proofreading the text. Finally the author would like to thank the two referees for their thorough reports and for pointing out oversights in the original submission. Their constructive comments have led to a much improved paper.

While writing this paper the author was employed at the Katholieke Universiteit Leuven, Belgium and the University of South Carolina where he was partially supported by NSF EPSCoR Grant EHR 9108772 and DARPA Grant AFOSR F49620-93-1-0083. He is also on leave as Senior Research Assistant of the National Fund of Scientific Research Belgium (NFWO).

REFERENCES

[1] R. Abgral and A. Harten. Multiresolution representation in unstructured meshes. *SIAM J. Numer. Anal.*, (submitted).

[2] A. Aldroubi, M. Eden, and M. Unser. Discrete spline filters for multiresolutions and wavelets of ℓ_2. *SIAM J. Math. Anal.*, 25(5):1412–1432, 1994.

[3] A. Aldroubi and M. Unser. Families of multiresolution and wavelet spaces with optimal properties. *Numer. Funct. Anal. Optim.*, 14:417–446, 1993.

[4] B. Alpert. A class of bases in L_2 for the sparse representation of integral operators. *SIAM J. Math. Anal.*, 24(1):246–262, 1993.

[5] B. Alpert, G. Beylkin, R. Coifman, and V. Rokhlin. Wavelet-like bases for the fast solution of second-kind integral equations. *SIAM J. Sci. Comput.*, 14(1):159–184, 1993.

[6] B. K. Alpert. Wavelets and other bases for fast numerical linear algebra. In [20], pages 181–216.

[7] A. Amellaoui and P. G. Lemarié-Rieusset. Ondelettes splines sur grilles irregulieres. unpublished, 1994.

[8] L. Andersson, N. Hall, B. Jawerth, and G. Peters. Wavelets on closed subsets of the real line. In [101], pages 1–61.

[9] L. Andersson, B. Jawerth, and M. Mitrea. The Cauchy singular integral operator and Clifford wavelets. In [14], pages 525–546.

[10] P. Auscher. Wavelets with boundary conditions on the interval. In [20], pages 217–236.

[11] P. Auscher and Ph. Tchamitchian. Bases d'ondelettes sur les courbes corde-arc, noyau de Cauchy et espaces de Hardy associés. *Rev. Mat. Iberoamericana*, 5:139–170, 1989.

[12] P. Auscher and Ph. Tchamitchian. Ondelettes et conjecture de Kato. *C. R. Acad. Sci. Paris Sér. I Math.*, I(313):63–66, 1991.

[13] G. Battle. A block spin construction of ondelettes. *Comm. Math. Phys.*, 110:601–615, 1987.

[14] J. Benedetto and M. Frazier, editors. *Wavelets: Mathematics and Applications*. CRC Press, Boca Raton, 1993.

[15] M. D. Buhmann and C. A. Micchelli. Spline prewavelets for non-uniform knots. *Numer. Math.*, 61:455–474, 1992.

[16] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.*, to appear.

[17] J. M. Carnicer, W. Dahmen, and J. M. Peña. Local decompositions of refinable spaces. *Appl. Comput. Harmon. Anal.*, 3:127–153, 1996.

[18] 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.

[19] C. K. Chui. *An Introduction to Wavelets*. Academic Press, San Diego, CA, 1992.

[20] C. K. Chui, editor. *Wavelets: A Tutorial in Theory and Applications*. Academic Press, San Diego, CA, 1992.

[21] C. K. Chui, L. Montefusco, and L. Puccio, editors. *Conference on Wavelets: Theory, Algorithms, and Applications*. Academic Press, San Diego, CA, 1994.

[22] C. K. Chui, J. Stöckler, and J. D. Ward. Compactly supported box-spline wavelets. *Approx. Theory Appl.*, 8(3):77–100, 1992.

[23] C. K. Chui and J. Z. Wang. A cardinal spline approach to wavelets. *Proc. Amer. Math. Soc.*, 113:785–793, 1991.

[24] C. K. Chui and J. Z. Wang. A general framework of compactly supported splines and wavelets. *J. Approx. Theory*, 71(3):263–304, 1992.

[25] C. K. Chui and J. Z. Wang. On compactly supported spline wavelets and a duality principle. *Trans. Amer. Math. Soc.*, 330:903–915, 1992.

[26] A. Cohen. Ondelettes, analyses multiresolutions et filtres miroirs en quadrature. *Ann. Inst. H. Poincaré, Anal. Non Linéaire*, 7(5):439–459, 1990.

[27] A. Cohen, W. Dahmen, and R. DeVore. Multiscale decompositions on bounded domains. *Trans. Amer. Math. Soc.*, to appear.

[28] A. Cohen and I. Daubechies. Non-separable bidimensional wavelet bases. *Rev. Mat. Iberoamericana*, 9(1):51–137, 1993.

[29] A. Cohen, I. Daubechies, and J. Feauveau. Bi-orthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45:485–560, 1992.

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

[31] A. Cohen, I. Daubechies, and P. Vial. Multiresolution analysis, wavelets and fast algorithms on an interval. *Appl. Comput. Harmon. Anal.*, 1(1):54–81, 1993.

[32] A. Cohen and J.-M. Schlenker. Compactly supported bidimensional wavelet bases with hexagonal symmetry. *Constr. Approx.*, 9(2):209–236, 1993.

[33] R. R. Coifman, P. W. Jones, and S. Semmes. Two elementary proofs of the L_2 boundedness of Cauchy integrals on Lipschitz curves. *J. Amer. Math. Soc.*, 2(3):553–564, 1989.

[34] R. R. Coifman, Y. Meyer, S. Quake, and M. V. Wickerhauser. Signal processing and compression with wave packets. In Y. Meyer, editor, *Proceedings of the International Conference on Wavelets, Marseille, 1989*. Masson, Paris, 1992.

[35] R. R. Coifman, Y. Meyer, and V. Wickerhauser. Size properties of wavelet packets. In [95], pages 453–470.

[36] R. R. Coifman and M. L. Wickerhauser. Entropy based algorithms for best basis selection. *IEEE Trans. Inform. Theory*, 38(2):713–718, 1992.

[37] D. Colella and C. Heil. The characterization of continuous four-coefficient scaling functions and wavelets. *IEEE Trans. Inform. Theory*, 38(2):876–881, 1992.

[38] D. Colella and C. Heil. Characterizations of scaling functions: Continuous solutions. *SIAM J. Matrix Anal. Appl.*, 15:496–518, 1994.

[39] S. Dahlke and R. DeVore. Besov regularity for elliptic boundary value problems. *Comm. Partial Differential Equations*, 22:1–16, 1997.

[40] W. Dahmen. Decomposition of refinable spaces and applications to operator equations. *Nu-*

mer. Algor., 5:229–245, 1993.

[41] W. Dahmen. Some remarks on multiscale transformations, stability, and biorthogonality. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Wavelets, Images, and Surface Fitting*, pages 157–188, Wellesley, 1994. A. K. Peters.

[42] W. Dahmen. Stability of multiscale transformations. *J. Fourier Anal. Appl.*, 2:341–361, 1996.

[43] W. Dahmen, B. Kleemann, S. Prössdorf, and R. Schneider. Multiscale methods for the double layer potential equation on a polyhedron. In H. P. Dikshit and C. A. Micchelli, editors, *Advances in Computational Mathematics*, pages 15–57. World Scientific, 1994.

[44] W. Dahmen and A. Kunoth. Multilevel preconditioning. *Numer. Math.*, 63(2):315–344, 1992.

[45] W. Dahmen and C. A. Micchelli. Banded matrices with banded inverses II: Locally finite decompositions of spline spaces. *Constr. Approx.*, 9(2-3):263–281, 1993.

[46] W. Dahmen, S. Prössdorf, and R. Schneider. Multiscale methods for pseudo-differential equations on smooth manifolds. In [21], pages 385–424. 1994.

[47] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.

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

[49] I. Daubechies. Orthonormal bases of compactly supported wavelets II: Variations on a theme. *SIAM J. Math. Anal.*, 24(2):499–519, 1993.

[50] I. Daubechies and J. C. Lagarias. Two-scale difference equations I. Existence and global regularity of solutions. *SIAM J. Math. Anal.*, 22(5):1388–1410, 1991.

[51] I. Daubechies and J. C. Lagarias. Two-scale difference equations II. Local regularity, infinite products of matrices and fractals. *SIAM J. Math. Anal.*, 23(4):1031–1079, 1992.

[52] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[53] B. Deng, B. Jawerth, G. Peters, and W. Sweldens. Wavelet probing for compression based segmentation. In A. F. Laine, editor, *Wavelet Applications in Signal and Image Processing*, pages 266–276. Proc. SPIE 2034, 1993.

[54] G. Deslauriers and S. Dubuc. Interpolation dyadique. In *Fractals, dimensions non entières et applications*, pages 44–55. Masson, Paris, 1987.

[55] G. Deslauriers and S. Dubuc. Symmetric iterative interpolation processes. *Constr. Approx.*, 5(1):49–68, 1989.

[56] D. L. Donoho. Smooth wavelet decompositions with blocky coefficient kernels. In [101], pages 259–308.

[57] D. L. Donoho. Interpolating wavelet transforms. Preprint, Department of Statistics, Stanford University, 1992.

[58] D. L. Donoho. Unconditional bases are optimal bases for data compression and for statistical estimation. *Appl. Comput. Harmon. Anal.*, 1(1):100–115, 1993.

[59] D. L. Donoho. On minimum entropy segmentation. Preprint, Department of Statistics, Stanford University, 1994.

[60] G. C. Donovan, J. S. Geronimo, and D. P. Hardin. A class of orthogonal multiresolution analyses in 2D. *Mathematical Methods for Curves and Surfaces*, pages 99–110, 1995.

[61] G. C. Donovan, J. S. Geronimo, and D. P. Hardin. Intertwining multiresolution analyses and the construction of piecewise polynomial wavelets. *SIAM J. Math. Anal.*, 27:1791–1815, 1996.

[62] G. C. Donovan, J. S. Geronimo, D. P. Hardin, and P. R. Massopust. Construction of orthogonal wavelets using fractal interpolation functions. *SIAM J. Math. Anal.*, 27:1158–1192, 1996.

[63] T. Eirola. Sobolev characterization of solutions of dilation equations. *SIAM J. Math. Anal.*, 23(4):1015–1030, 1992.

[64] G. Fernández, S. Periaswamy, and W. Sweldens. LIFTPACK: A software package for wavelet transforms using lifting. In M. Unser, A. Aldroubi, and A. F. Laine, editors, *Wavelet Applications in Signal and Image Processing IV.* Proc. SPIE 2825, 1996.

[65] M. Frazier and B. Jawerth. Decomposition of Besov spaces. *Indiana Univ. Math. J.*, 34(4):777–799, 1985.

[66] M. Frazier and B. Jawerth. The φ-transform and applications to distribution spaces. In M. Cwikel et al., editor, *Function Spaces and Applications*, number 1302 in Lecture Notes in Math., pages 223–246, 1988.

[67] M. Frazier and B. Jawerth. A discrete transform and decompositions of distribution spaces. *J. Funct. Anal.*, 93:34–170, 1990.

[68] M. Girardi and W. Sweldens. A new class of unbalanced Haar wavelets that form an unconditional basis for L_p on general measure spaces. *J. Fourier Anal. Appl.*, 3(4), 1997.

[69] S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. *Computer*

Graphics (SIGGRAPH '93 Proceedings), pages 221–230, 1993.

[70] D. P. Hardin, B. Kessler, and P. R. Massopust. Multiresolution analyses based on fractal functions. *J. Approx. Theory*, 71(1):104–120, 1992.

[71] A. Harten. Multiresolution representation of data: A general framework. *SIAM J. Numer. Anal.*, 33(3):1205–1256, 1996.

[72] P. N. Heller. Rank M wavelets with N vanishing moments. *SIAM J. Math. Anal.*, 16, April 1995.

[73] C. Herley and M. Vetterli. Wavelets and recursive filter banks. *IEEE Trans. Signal Process.*, 41(8):2536–2556, 1993.

[74] B. Jawerth and G. Peters. Wavelets on non smooth sets of $\mathbf{R}^n$. Preprint, 1995.

[75] B. Jawerth and W. Sweldens. Wavelet multiresolution analyses adapted for the fast solution of boundary value ordinary differential equations. In N. D. Melson, T. A. Manteuffel, and S. F. McCormick, editors, *Sixth Copper Mountain Conference on Multigrid Methods*, pages 259–273. NASA Conference Publication 3224, 1993.

[76] R.-Q. Jia and C. A. Micchelli. Using the refinement equations for the construction of pre-wavelets II: Powers of two. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Curves and Surfaces*, New York, 1991. Academic Press.

[77] J. Kovačević and M. Vetterli. Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for $\mathbf{R}^n$. *IEEE Trans. Inform. Theory*, 38(2):533–555, 1992.

[78] P.-G. Lemarié. Ondelettes a localisation exponentielle. *J. Math. Pures Appl.*, 67(3):227–236, 1988.

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

[80] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution surfaces of arbitrary topological type. *ACM Trans. on Graphics*, 16(1):34–73, 1997.

[81] M. Lundberg and G. Welland. Construction of compact p-wavelets. *Constr. Approx.*, 9:347–370, 1993.

[82] S. Mallat and S. Zhong. Wavelet transform maxima and multiscale edges. In [95], pages 67–104.

[83] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Patt. Anal. Mach. Intell.*, 14:710–732, 1992.

[84] S. G. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoust. Speech Signal Process.*, 37(12):2091–2110, 1989.

[85] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbf{R})$. *Trans. Amer. Math. Soc.*, 315(1):69–87, 1989.

[86] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(7):674–693, 1989.

[87] Y. Meyer. *Ondelettes et Opérateurs,* I: *Ondelettes,* II: *Opérateurs de Calderón-Zygmund,* III: (with R. Coifman)*, Opérateurs multilinéaires*. Hermann, Paris, 1990. English translation of first volume, *Wavelets and Operators*, is published by Cambridge University Press, 1993.

[88] Y. Meyer. Ondelettes sur l'intervalle. *Rev. Mat. Iberoamericana*, 7:115–133, 1992.

[89] C. A. Micchelli. Using the refinement equations for the construction of pre-wavelets. *Numerical Algorithms*, 1(1):75–116, 1991.

[90] C. A. Micchelli, C. Rabut, and F. I. Utretas. Using the refinement equations for the construction of pre-wavelets III: Elliptic splines. *Numerical Algorithms*, 1(1):331–352, 1991.

[91] M. Mitrea. *Singular integrals, Hardy spaces and Clifford wavelets*. Number 1575 in Lecture Notes in Math. 1994.

[92] S. D. Riemenschneider and Z. Shen. Wavelets and pre-wavelets in low dimensions. *J. Approx. Theory*, 71(1):18–38, 1992.

[93] S. D. Riemenschneider and Z. W. Shen. Box splines, cardinal series and wavelets. In C. K. Chui, editor, *Approximation Theory and Functional Analysis*, New York, 1991. Academic Press.

[94] O. Rioul. Simple regularity criteria for subdivision schemes. *SIAM J. Math. Anal.*, 23(6):1544–1576, 1992.

[95] M. B. Ruskai, G. Beylkin, R. Coifman, I. Daubechies, S. Mallat, Y. Meyer, and L. Raphael, editors. *Wavelets and their Applications*. Jones and Bartlett, Boston, 1992.

[96] R. Schneider. *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösing großer vollbesetzter Gleichungssysteme*. Habilitationsschrift, TH Darmstadt, 1995.

[97] P. Schröder. *Wavelet Algorithms for Illumination Computations*. PhD thesis, Department of Computer Science, Princeton University, 1994.

[98] P. Schröder, S. J. Gortler, M. F. Cohen, and P. Hanrahan. Wavelet projections for radiosity. *Computer Graphics Forum*, 13(2):141–152, June 1994.

[99] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings, (SIGGRAPH 95)*, pages 161–172, 1995.

[100] P. Schröder and W. Sweldens. Spherical wavelets: Texture processing. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95*. Springer Verlag, Wien, New York, August 1995.

[101] L. L. Schumaker and G. Webb, editors. *Recent Advances in Wavelet Analysis.* Academic Press, New York, 1993.

[102] P. Steffen, P. Heller, R. A. Gopinath, and C. S. Burrus. Theory of regular m-band wavelets. *IEEE Trans. Signal Process.*, 41:3497–3511, 1993.

[103] J. O. Strömberg. A modified Franklin system and higher order spline systems on $\mathbf{R}^n$ as unconditional bases for Hardy spaces. In Beckner et al., editor, *Conference on Harmonic Analysis in Honor of Antoni Zygmund*, volume II, pages 475–494, Chicago, 1981. Univ. of Chicago Press.

[104] W. Sweldens. *Construction and Applications of Wavelets in Numerical Analysis.* PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, 1994.

[105] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[106] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.

[107] M. Unser, A. Aldroubi, and M. Eden. On the asymptotic convergence of B-spline wavelets to Gabor functions. *IEEE Trans. Inform. Theory*, 38:864–872, 1992.

[108] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms. *Signal Process.*, 30:141–162, 1993.

[109] P. P. Vaidyanathan, T. Q. Nguyen, Z. Doğanata, and T. Saramäki. Improved technique for design of perfect reconstruction FIR QMF banks with lossless polyphase matrices. *IEEE Trans. Acoust. Speech Signal Process.*, 37(7):1042–1055, 1989.

[110] M. Vetterli and C. Herley. Wavelets and filter banks: Theory and design. *IEEE Trans. Acoust. Speech Signal Process.*, 40(9):2207–2232, 1992.

[111] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[112] L. F. Villemoes. Energy moments in time and frequency for two-scale difference equation solutions and wavelets. *SIAM J. Math. Anal.*, 23(6):1119–1543, 1992.

[113] L. F. Villemoes. Wavelet analysis of refinement equations. *SIAM J. Math. Anal.*, 25(5):1433–1460, 1994.

[114] M. V. Wickerhauser. Acoustic signal compression with wavelet packets. In [20], pages 679–700.

[115] M. V. Wickerhauser. *Adapted Wavelet Analysis from Theory to Software.* A. K. Peters, Wellesley, MA, 1994.

[116] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.