FACTORING WAVELET TRANSFORMS INTO LIFTING STEPS

INGRID DAUBECHIES[†] AND WIM SWELDENS[*]

September 1996, revised November 1997

ABSTRACT. This paper is essentially tutorial in nature. We show how any discrete wavelet transform or two band subband filtering with finite filters can be decomposed into a finite sequence of simple filtering steps, which we call lifting steps but that are also known as ladder structures. This decomposition corresponds to a factorization of the polyphase matrix of the wavelet or subband filters into elementary matrices. That such a factorization is possible is well-known to algebraists (and expressed by the formula $\mathrm{SL}(n;\mathbf{R}[z,z^{-1}]) = \mathrm{E}(n;\mathbf{R}[z,z^{-1}])$); it is also used in linear systems theory in the electrical engineering community. We present here a self-contained derivation, building the decomposition from basic principles such as the Euclidean algorithm, with a focus on applying it to wavelet filtering. This factorization provides an alternative for the lattice factorization, with the advantage that it can also be used in the biorthogonal, i.e, non-unitary case. Like the lattice factorization, the decomposition presented here asymptotically reduces the computational complexity of the transform by a factor two. It has other applications, such as the possibility of defining a wavelet-like transform that maps integers to integers.

1. INTRODUCTION

Over the last decade several constructions of compactly supported wavelets originated both from mathematical analysis and the signal processing community. The roots of critically sampled wavelet transforms are actually older than the word "wavelet" and go back to the context of subband filters, or more precisely quadrature mirror filters [35, 36, 42, 50, 51, 52, 53, 57, 55, 59]. In mathematical analysis, wavelets were defined as translates and dilates of one fixed function and were used to both analyze and represent general functions [13, 18, 22, 34, 21]. In the mid eighties the introduction of multiresolution analysis and the fast wavelet transform by Mallat and Meyer provided the connection between subband filters and wavelets [30, 31, 34]; this led to new constructions, such as the smooth orthogonal, and compactly supported wavelets [16]. Later many generalizations to the biorthogonal or semiorthogonal (pre-wavelet) case were introduced. Biorthogonality allows the construction of symmetric wavelets and thus linear phase

[†]Program for Applied and Computational Mathematics, Princeton University, Princeton NJ 08544.
[*]Lucent Technologies, Bell Laboratories, Rm. 2C-175, 700 Mountain Avenue, Murray Hill NJ 07974.
ingrid@math.princeton.edu, wim@bell-labs.com.

filters. Examples are: the construction of semiorthogonal spline wavelets [1, 8, 10, 11, 49], fully biorthogonal compactly supported wavelets [12, 56], and recursive filter banks [25].

Various techniques to construct wavelet bases, or to factor existing wavelet filters into basic building blocks are known. One of these is lifting. The original motivation for developing lifting was to build second generation wavelets, i.e., wavelets adapted to situations that do not allow translation and dilation like non-Euclidean spaces. First generation wavelets are all translates and dilates of one or a few basic shapes; the Fourier transform is then the crucial tool for wavelet construction. A construction using lifting, on the contrary, is entirely spatial and therefore ideally suited for building second generation wavelets when Fourier techniques are no longer available. When restricted to the translation and dilation invariant case, or the "first generation," lifting comes down to well-known ladder type structures and certain factoring algorithms. In the next few paragraphs, we explain lifting and show how it provides a spatial construction and allows for second generation wavelets; later we focus on the first generation case and the connections with factoring schemes.

The basic idea of wavelet transforms is to exploit the correlation structure present in most real life signals to build a sparse approximation. The correlation structure is typically local in space (time) and frequency; neighboring samples and frequencies are more correlated than ones that are far apart. Traditional wavelet constructions use the Fourier transform to build the space-frequency localization. However, as the following simple example shows, this can also be done in the spatial domain.

Consider a signal $\mathbf{x} = (x_k)_{k\in\mathbf{Z}}$ with $x_k \in \mathbf{R}$. Let us split it in two disjoint sets which are called the polyphase components: the even indexed samples $\mathbf{x}_e = (x_{2k})_{k\in\mathbf{Z}}$, or "evens" for short, and the odd indexed samples $\mathbf{x}_o = (x_{2k+1})_{k\in\mathbf{Z}}$, or "odds." Typically these two sets are closely correlated. Thus it is only natural that given one set, e.g., the even, one can build a good predictor P for the other set. Of course the predictor need not be exact, so we need to record the difference or detail d:

$$\mathbf{d} = \mathbf{x}_e - P(\mathbf{x}_o).$$

Given the detail $\mathbf{d}$ and the odd, we can immediately recover the even as

$$\mathbf{x}_e = P(\mathbf{x}_o) + \mathbf{d}.$$

If P is a good predictor, then $\mathbf{d}$ approximately will be a sparse set; in other words, we expect the first order entropy to be smaller for $\mathbf{d}$ than for $\mathbf{x}_o$. Let us look at a simple example. An easy predictor for an odd sample x_{2k+1} is simply the average of its two even neighbors; the detail coefficient then is

$$d_k = x_{2k+1} - (x_{2k} + x_{2k+1})/2.$$

From this we see that if the original signal is locally linear, the detail coefficient is zero. The operation of computing a prediction and recording the detail we will call a *lifting step*. The idea of retaining $\mathbf{d}$ rather than $\mathbf{x}_o$ is well known and forms the basis of so-called DPCM methods [26, 27]. This idea connects naturally with wavelets as follows. The prediction steps can take care of some of the spatial correlation, but for wavelets we also want to get some separation in the frequency domain. Right now we have a transform from $(\mathbf{x}_e, \mathbf{x}_o)$ to $(\mathbf{x}_e, \mathbf{d})$. The frequency separation is poor since $\mathbf{x}_e$ is obtained by simply subsampling so that serious aliasing occurs. In particular, the running average of the $\mathbf{x}_e$ is not the same as that of the original samples $\mathbf{x}$. To correct this, we propose a second lifting step, which replaces the evens with smoothed values $\mathbf{s}$ with the use of an *update* operator U applied to the details:

$$\mathbf{s} = \mathbf{x}_e + U(\mathbf{d}).$$

Again this step is trivially invertible: given $(\mathbf{s}, \mathbf{d})$ we can recover $\mathbf{x}_e$ as

$$\mathbf{x}_e = \mathbf{s} - U(\mathbf{d}),$$

and then $\mathbf{x}_o$ can be recovered as explained earlier. This illustrates one of the built-in features of lifting: no matter how P and U are chosen, the scheme is always invertible and thus leads to critically sampled perfect reconstruction filter banks. The block diagram of the two lifting steps is given in Figure 1.

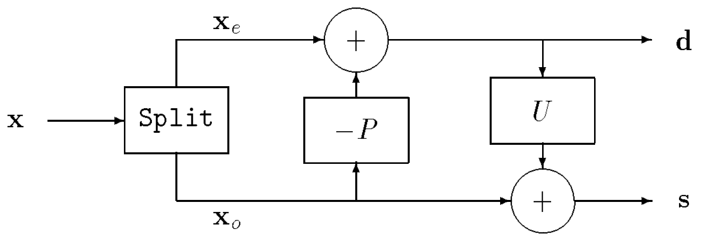

FIGURE 1. Block diagram of predict and update lifting steps.

Coming back to our simple example, it is easy to see that an update operator that restores the correct running average, and therefore reduces aliasing, is given by

$$s_k = x_{2k} + (d_{k-1} + d_k)/4.$$

This can be verified graphically by looking at Figure 2.

This simple example, when put in the wavelet framework, turns out to correspond to the biorthogonal (2,2) wavelet transform of [12], which was originally constructed using Fourier arguments. By the construction above, which did not use the Fourier transform but instead reasoned using only spatial arguments, one can easily work in a more general setting. Imagine for a moment that the samples were irregularly spaced. Using the same spatial arguments as

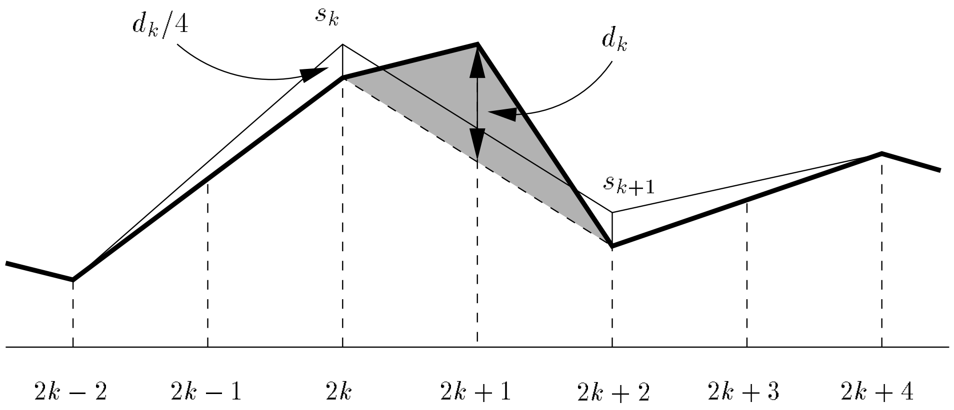

FIGURE 2. Geometric interpretation for piecewise linear predict and update lifting steps. The original signal is drawn in bold. The wavelet coefficient d_k is computed as the difference of an odd sample and the average of the two neighboring evens. This corresponds to a loss $d_k/2$ in area drawn in grey. To preserve the running average this area has to be redistributed to the even locations resulting in a coarser piecewise linear signal s_k drawn in thin line. Because the coarse scale is *twice* the fine scale and *two* even locations are affected, $d_k/4$, i.e, one *quarter* of the wavelet coefficient, has to be added to the even samples to obtain the s_k. Then the thin and bold lines cover the same area. (For simplicity we assumed that the wavelet coefficients d_{k-1} and d_{k+1} are zero.)

above we could then see that a good predictor is of the form $\beta\, x_{2k} + (1-\beta)\, x_{2k+1}$ where the β varies spatially and depends on the irregularity of the grid. Similarly spatially varying update coefficients can be computed [46]. This thus immediately allows for a (2,2) type transform for irregular samples. These spatial lifting steps can also be used in higher dimensions (see [45]) and leads e.g., to wavelets on a sphere [40] or more complex manifolds.

Note that the idea of using spatial wavelet constructions for building second generation wavelets has been proposed by several researchers:

- The lifting scheme is inspired by the work of Donoho [19] and Lounsbery et al. [29]. Donoho [19] shows how to build wavelets from interpolating scaling functions, while Lounsbery et al. build a multiresolution analysis of surfaces using a technique that is algebraically the same as lifting.
- Dahmen and collaborators, independently of lifting, worked on stable completions of multiscale transforms, a setting similar to second generation wavelets [7, 15]. Again independently, both of Dahmen and of lifting, Harten developed a general multiresolution approximation framework based on spatial prediction [23].

- In [14], Dahmen and Micchelli propose a construction of compactly supported wavelets that generates complementary spaces in a multiresolution analysis of univariate irregular knot splines.

The construction of the (2,2) example via lifting is one example of a 2 step lifting construction for an entire family of Deslauriers-Dubuc biorthogonal interpolating wavelets[1]. Lifting thus provides a framework that allows the construction of certain biorthogonal wavelets which can be generalized to the second generation setting. A natural question now is how much of the first generation wavelet families can be built with the lifting framework. It turns out that every FIR wavelet or filter bank can be decomposed into lifting steps. This can be seen by writing the transform in the polyphase form. Statements concerning perfect reconstruction or lifting can then be made using matrices with polynomial or Laurent polynomial entries. A lifting step then becomes a so-called elementary matrix, that is, a triangular matrix (lower or upper) with all diagonal entries equal to one. It is a well known result in matrix algebra that any matrix with polynomial entries and determinant one can be factored into such elementary matrices. For those familiar with the common notation in this field, this is written as $\mathrm{SL}(n; \mathbf{R}[z, z^{-1}]) = \mathrm{E}(n; \mathbf{R}[z, z^{-1}])$. The proof relies on the 2000 year old Euclidean algorithm. In the filter bank literature subband transform built using elementary matrices are known as ladder structures and were introduced in [5]. Later several constructions concerning factoring into ladder steps were given [28, 41, 48, 32, 33]. Vetterli and Herley [56] also use the Euclidean algorithm and the connection to diophantine equations to find all high pass filters that, together with a given low-pass filter, make a finite filter wavelet transform. Van Dyck et al. use ladder structures to design a wavelet video coder [20].

In this paper we give a self-contained constructive proof of the standard factorization result and apply it to several popular wavelets. We consider the Laurent polynomial setting as opposed to the standard polynomial setting because it is more general, allows for symmetry and also poses some interesting questions concerning non-uniqueness.

This paper is organized as follows. In Section 2 we review some facts about filters and Laurent polynomials. Section 3 gives the basics behind wavelet transforms and the polyphase representation while Section 4 discusses the lifting scheme. We review the Euclidean algorithm in Section 5 before moving to the main factoring result in Section 6. Section 7 gives several examples. In Section 8 we show how lifting can reduce the computational complexity of the wavelet transform by a factor two. Finally Section 9 contains comments.

[1] This family was derived independently, but without the use of lifting, by several people: Reissell [38], Tian and Wells [47], and Strang [43]. The derivation using lifting can be found in [44].

2. Filters and Laurent polynomials

A *filter* h is a linear time invariant operator and is completely determined by its *impulse response*: $\{h_k \in \mathbf{R} \mid k \in \mathbf{Z}\}$. The filter h is a Finite Impulse Response (FIR) filter in case only a finite number of filter coefficients h_k are non-zero. We then let k_b (respectively k_e) be the smallest (respectively largest) integer number k for which h_k is non-zero. The z-transform of a FIR filter h is a Laurent polynomial $h(z)$ given by

$$h(z) = \sum_{k=k_b}^{k_e} h_k \, z^{-k}.$$

In this paper, we consider only FIR filters. We often use the symbol h to denote both the filter and the associated Laurent polynomial $h(z)$. The *degree* of a Laurent polynomial h is defined as

$$|h| = k_e - k_b.$$

So the length of the filter is the degree of the associated polynomial plus one. Note that the polynomial z^p seen as a Laurent polynomial has degree zero, while as a regular polynomial it would have degree p. In order to make consistent statements, we set the degree of the zero polynomial to $-\infty$.

The set of all Laurent polynomials with real coefficients has a commutative ring structure. The sum or difference of two Laurent polynomials is again a Laurent polynomial. The product of a Laurent polynomial of degree l and a Laurent polynomial of degree l' is a Laurent polynomial of degree $l + l'$. This ring is usually denoted as $\mathbf{R}[z, z^{-1}]$.

Within a ring, exact division is not possible in general. However, for Laurent polynomials, division with remainder is possible. Take two Laurent polynomials $a(z)$ and $b(z) \neq 0$ with $|a(z)| \geqslant |b(z)|$, then there always exists a Laurent polynomial $q(z)$ (the quotient) with $|q(z)| = |a(z)| - |b(z)|$, and a Laurent polynomial $r(z)$ (the remainder) with $|r(z)| < |b(z)|$ so that

$$a(z) = b(z)\, q(z) + r(z).$$

We denote this as (C-language notation):

$$q(z) = a(z) \,/\, b(z) \quad \text{and} \quad r(z) = a(z) \,\%\, b(z).$$

If $|b(z)| = 0$ which means $b(z)$ is a monomial, then $r(z) = 0$ and the division is exact. A Laurent polynomial is invertible if and only if it is a monomial. This is the main difference with the ring of (regular) polynomials where constants are the only polynomials that can be inverted. Another difference is that the long division of Laurent polynomials is not necessarily unique. The following example illustrates this.

Example 1. Suppose we want to divide $a(z) = z^{-1} + 6 + z$ by $b(z) = 4 + 4\,z$. This means we have to find a Laurent polynomial $q(z)$ of degree 1 so that $r(z)$ given by

$$r(z) = a(z) - b(z)\,q(z)$$

is of degree zero. This implies that $b(z)\,q(z)$ has to match $a(z)$ in two terms. If we let those terms be the term in z^{-1} and the constant then the answer is $q(z) = 1/4\,(z^{-1} + 5)$. Indeed,

$$r(z) = (z^{-1} + 6 + z) - (4 + 4\,z)(1/4\,z^{-1} + 5/4) = -4\,z.$$

The remainder thus is of degree zero and we have completed the division. However if we choose the two matching terms to be the ones in z and z^{-1}, the answer is $q(z) = 1/4\,(z^{-1} + 1)$. Indeed,

$$r(z) = (z^{-1} + 6 + z) - (4 + 4\,z)(1/4\,z^{-1} + 1/4) = 4.$$

Finally, if we choose to match the constant and the term in z, the solution is $q(z) = 1/4\,(5\,z^{-1}+1)$ and the remainder is $r(z) = -4\,z^{-1}$.

The fact that division is not unique will turn out to be particularly useful later. In general $b(z)\,q(z)$ has to match $a(z)$ in at least $|a(z)| - |b(z)| + 1$ terms, but we are free to choose these terms in the beginning, the end, or divided between the beginning and the end of $a(z)$. For each choice of terms a corresponding long division algorithm exists.

In this paper, we also work with 2×2 matrices of Laurent polynomials, e.g.,

$$M(z) = \begin{bmatrix} a(z) & b(z) \\ c(z) & d(z) \end{bmatrix}.$$

These matrices also form a ring, which is denoted by $\mathrm{M}(2; \mathbf{R}[z, z^{-1}])$. If the determinant of such a matrix is a monomial, then the matrix is invertible. The set of invertible matrices is denoted $\mathrm{GL}(2; \mathbf{R}[z, z^{-1}])$. A matrix from this set is *unitary* (sometimes also referred to as *para-unitary*) in case

$$M(z)^{-1} = M(z^{-1})^t.$$

3. Wavelet transforms

Figure 3 shows the general block scheme of a wavelet or subband transform. The forward transform uses two analysis filters $\widetilde{h}$ (low-pass) and $\widetilde{g}$ (band pass) followed by subsampling, while the inverse transform first upsamples and then uses two synthesis filters h (low-pass) and g (high-pass). For details on wavelet and subband transforms we refer to [43] and [57]. In this paper we consider only the case where the four filters h, g, $\widetilde{h}$, and $\widetilde{g}$, of the wavelet transform are FIR filters. The conditions for perfect reconstruction are given by

$$\begin{aligned} h(z)\,\widetilde{h}(z^{-1}) + g(z)\,\widetilde{g}(z^{-1}) &= 2 \\ h(z)\,\widetilde{h}(-z^{-1}) + g(z)\,\widetilde{g}(-z^{-1}) &= 0. \end{aligned}$$

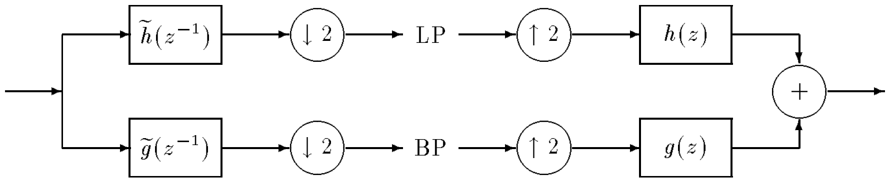

FIGURE 3. Discrete wavelet transform (or subband transform): The forward transform consists of two analysis filters $\widetilde{h}$ (low-pass) and $\widetilde{g}$ (high-pass) followed by subsampling, while the inverse transform first upsamples and then uses two synthesis filters h (low-pass) and g (high-pass).

We define the *modulation matrix* $M(z)$ as

$$M(z) = \begin{bmatrix} h(z) & h(-z) \\ g(z) & g(-z) \end{bmatrix}.$$

We similarly define the dual modulation matrix $\widetilde{M}(z)$. The perfect reconstruction condition can now be written as

$$\widetilde{M}(z^{-1})^t \, M(z) = 2\,\mathbf{I}, \tag{1}$$

where $\mathbf{I}$ is the 2×2 identity matrix. If all filters are FIR, then the matrices $M(z)$ and $\widetilde{M}(z)$ belong to $\mathrm{GL}(2; \mathbf{R}[z, z^{-1}])$.

A special case are *orthogonal* wavelet transforms in which case $h = \widetilde{h}$ and $g = \widetilde{g}$. The modulation matrix $M(z) = \widetilde{M}(z)$ is then $\sqrt{2}$ times a unitary matrix.

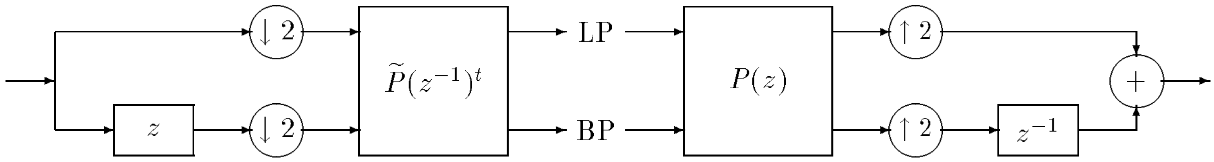

FIGURE 4. *Polyphase representation of wavelet transform: first subsample into even and odd, then apply the dual polyphase matrix. For the inverse transform: first apply the polyphase matrix and then join even and odd.*

The *polyphase representation* is a particularly convenient tool to express the special structure of the modulation matrix [3]. The polyphase representation of a filter h is given by

$$h(z) = h_e(z^2) + z^{-1} h_o(z^2),$$

where h_e contains the even coefficients, and h_o contains the odd coefficients:

$$h_e(z) = \sum_k h_{2k} z^{-k} \quad \text{and} \quad h_o(z) = \sum_k h_{2k+1}\, z^{-k},$$

or

$$h_e(z^2) = \frac{h(z) + h(-z)}{2} \quad \text{and} \quad h_o(z^2) = \frac{h(z) - h(-z)}{2\, z^{-1}}.$$

We assemble *the polyphase matrix* as

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix},$$

so that

$$P(z^2)^t = 1/2\, M(z) \begin{bmatrix} 1 & z \\ 1 & -z \end{bmatrix}.$$

We define $\widetilde{P}(z)$ similarly. The wavelet transform now is represented schematically in Figure 4. The perfect reconstruction property is given by

$$P(z)\, \widetilde{P}(z^{-1})^t = \mathbf{I}. \tag{2}$$

Again we want $P(z)$ and $\widetilde{P}(z)$ to contain only Laurent polynomials. Equation (2) then implies that det P(z) and its inverse are both Laurent polynomials; this is possible only in case $\det P(z)$ is a monomial in z: $\det P(z) = C z^l$; $P(z)$ and $\widetilde{P}(z)$ belong then to $\mathrm{GL}(2; \mathbf{R}[z, z^{-1}])$. Without loss of generality we assume that $\det P(z) = 1$, i.e, $P(z)$ is in $\mathrm{SL}(2; \mathbf{R}[z, z^{-1}])$. Indeed, if the determinant is not one, we can always divide $g_e(z)$ and $g_o(z)$ by the determinant. This means that for a given filter h, we can always scale and shift the filter g so that the determinant of the polyphase matrix is one.

The problem of finding an FIR wavelet transform thus amounts to finding a matrix $P(z)$ with determinant one. Once we have such a matrix, $\widetilde{P}(z)$ and the four filters for the wavelet transform follow immediately. From (2) and Cramer's rule it follows that

$$\tilde{h}_e(z) = g_o(z^{-1}), \qquad \tilde{h}_o(z) = -g_e(z^{-1}), \qquad \tilde{g}_e(z) = -h_o(z^{-1}), \qquad \tilde{g}_o(z) = h_e(z^{-1}).$$

This implies

$$\widetilde{g}(z) = z^{-1}\, h(-z^{-1}) \quad \text{and} \quad \widetilde{h}(z) = -z^{-1}\, g(-z^{-1}).$$

The most trivial example of a polyphase matrix is $P(z) = \mathbf{I}$. This results in $h(z) = \widetilde{h}(z) = 1$ and $g(z) = \widetilde{g}(z) = z^{-1}$. The wavelet transform then does nothing else but subsampling even and odd samples. This transform is called the polyphase transform, but in the context of lifting it is often referred to as the Lazy wavelet transform [44]. (The reason is that the notion of the Lazy wavelet can also be used in the second generation setting.)

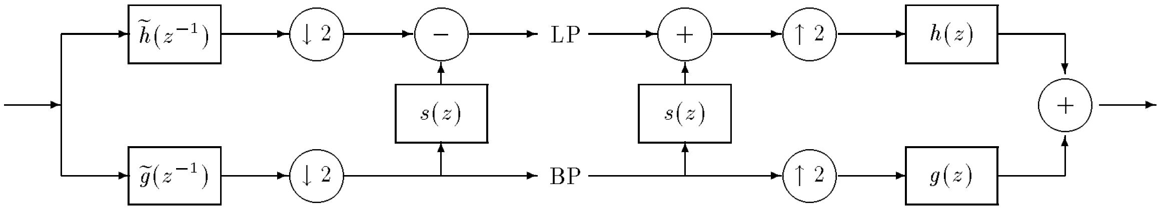

Figure 5. *The lifting scheme: First a classical subband filter scheme and then lifting the low-pass subband with the help of the high-pass subband.*

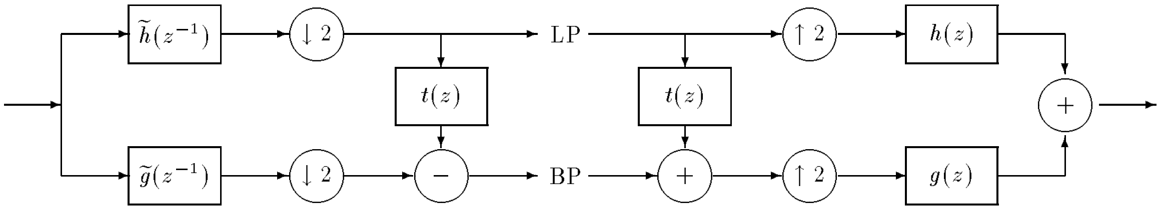

Figure 6. *The dual lifting scheme: First a classical subband filter scheme and later lifting the high-pass subband with the help of the low-pass subband.*

4. The Lifting Scheme

The lifting scheme [44, 45] is an easy relationship between perfect reconstruction filter pairs (h, g) that have the same low-pass or high-pass filter. One can then start from the Lazy wavelet and use lifting to gradually build one's way up to a multiresolution analysis with particular properties.

Definition 2. A filter pair (h, g) is *complementary* in case the corresponding polyphase matrix $P(z)$ has determinant 1.

If (h, g) is complementary, so is $(\widetilde{h}, \widetilde{g})$. This allows us to state the lifting scheme.

Theorem 3 (Lifting). *Let (h, g) be complementary. Then any other finite filter g^{new} complementary to h is of the form:*

$$g^{\text{new}}(z) = g(z) + h(z)\, s(z^2),$$

where $s(z)$ is a Laurent polynomial. Conversely any filter of this form is complementary to h.

Proof. The polyphase components of $h(z)\, s(z^2)$ are $h_e(z)\, s(z)$ for even and $h_o(z)\, s(z)$ for odd. After lifting, the new polyphase matrix is thus given by

$$P^{\text{new}}(z) = P(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}.$$

This operation does not change the determinant of the polyphase matrix. □

Figure 5 shows the schematic representation of lifting. Theorem 3 can also be written relating the low-pass filters h and $\widetilde{h}$. In this formulation, it is exactly the Vetterli-Herley lemma [56, Proposition 4.7]. The dual polyphase matrix is given by:

$$\widetilde{P}^{\text{new}}(z) = \widetilde{P}(z) \begin{bmatrix} 1 & 0 \\ -s(z^{-1}) & 1 \end{bmatrix}.$$

We see that lifting creates a new $\widetilde{h}$ filter given by

$$\widetilde{h}^{\text{new}}(z) = \widetilde{h}(z) - \widetilde{g}(z)\, s(z^{-2}).$$

Theorem 4 (Dual lifting). *Let (h, g) be complementary. Then any other finite filter h^{new} complementary to g is of the form:*

$$h^{\text{new}}(z) = h(z) + g(z)\, t(z^2),$$

where $t(z)$ is a Laurent polynomial. Conversely any filter of this form is complementary to g.

After dual lifting, the new polyphase matrix is given by

$$P^{\text{new}}(z) = P(z) \begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix}.$$

Dual lifting creates a new $\widetilde{g}$ given by

$$\widetilde{g}^{\text{new}}(z) = \widetilde{g}(z) - \widetilde{h}(z)\, t(z^{-2}).$$

Figure 6 shows the schematic representation of dual lifting. In [44] lifting and dual lifting are used to build wavelet transforms starting from the Lazy wavelet. There a whole family of wavelets is constructed from the Lazy followed by one dual lifting and one primal lifting step. All the filters h constructed this way are half band and the corresponding scaling functions are interpolating. Because of the many advantages of lifting, it is natural to try to build other wavelets as well, perhaps using multiple lifting steps. In the next section we will show that any wavelet transform with finite filters can be obtained starting from the Lazy followed by a finite number of alternating lifting and dual lifting steps. In order to prove this, we first need to study the Euclidean algorithm in closer detail.

5. The Euclidean Algorithm

The Euclidean algorithm was originally developed to find the greatest common divisor of two natural numbers, but it can be extended to find the greatest common divisor of two polynomials, see, e.g, [4]. Here we need it to find common factors of Laurent polynomials. The main difference with the polynomial case is again that the solution is not unique. Indeed the gcd of two Laurent polynomials is defined only up to a factor z^p. (This is similar to saying that the gcd of two polynomials is defined only up to a constant.) Two Laurent polynomials are relatively prime in case their gcd has degree zero. Note that they can share roots at zero and infinity.

Theorem 5 (Euclidean Algorithm for Laurent Polynomials). *Take two Laurent polynomials $a(z)$ and $b(z) \neq 0$ with $|a(z)| \geqslant |b(z)|$. Let $a_0(z) = a(z)$ and $b_0(z) = b(z)$ and iterate the following steps starting from $i = 0$*

$$a_{i+1}(z) = b_i(z) \tag{3}$$

$$b_{i+1}(z) = a_i(z) \,\%\, b_i(z). \tag{4}$$

Then $a_n(z) = gcd(a(z), b(z))$ where n is the smallest number for which $b_n(z) = 0$.

Given that $|b_{i+1}(z)| < |b_i(z)|$, there is an m so that $|b_m(z)| = 0$. The algorithm then finishes for $n = m+1$. The number of steps thus is bounded by $n \leqslant |b(z)|+1$. If we let $q_{i+1}(z) = a_i(z) \,/\, b_i(z)$, we have that

$$\begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} = \prod_{i=n}^{1} \begin{bmatrix} 0 & 1 \\ 1 & -q_i(z) \end{bmatrix} \begin{bmatrix} a(z) \\ b(z) \end{bmatrix}.$$

Consequently

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_n(z) \\ 0 \end{bmatrix},$$

and thus $a_n(z)$ divides both $a(z)$ and $b(z)$. If $a_n(z)$ is a monomial, then $a(z)$ and $b(z)$ are relatively prime.

Example 6. Let $a(z) = a_0(z) = z^{-1} + 6 + z$ and $b(z) = b_0(z) = 4 + 4z$. Then the first division gives us (see the example in Section 2):

$$a_1(z) = 4 + 4\,z$$

$$b_1(z) = 4$$

$$q_1(z) = 1/4\,z^{-1} + 1/4.$$

The next step yields

$$a_2(z) = 4$$

$$b_2(z) = 0$$

$$q_2(z) = 1 + z.$$

Thus, $a(z)$ and $b(z)$ are relatively prime and

$$\begin{bmatrix} z^{-1} + 6 + z \\ 4 + 4\,z \end{bmatrix} = \begin{bmatrix} 1/4\,z^{-1} + 1/4 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1+z & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 4 \\ 0 \end{bmatrix}.$$

The number of steps here is $n = 2 = |b(z)| + 1$.

6. The Factoring Algorithm

In this section, we explain how any pair of complementary filters (h, g) can be factored into lifting steps. First, note that $h_e(z)$ and $h_o(z)$ have to be relatively prime because any common factor would also divide $\det P(z)$ and we already know that $\det P(z)$ is 1. We can thus run the Euclidean algorithm starting from $h_e(z)$ and $h_o(z)$ and the gcd will be a monomial. Given the non-uniqueness of the division we can always choose the quotients so that the gcd is a constant. Let this constant be K. We thus have that

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix}.$$

Note that in case $|h_o(z)| > |h_e(z)|$, the first quotient $q_1(z)$ is zero. We can always assume that n is even. Indeed if n is odd, we can multiply the $h(z)$ filter with z and $g(z)$ with $-z^{-1}$. This does not change the determinant of the polyphase matrix. It flips (up to a monomial) the polyphase components of h and thus makes n even again. Given a filter h we can always find *a* complementary filter g^0 by letting

$$P^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix} = \prod_{i=1}^{n} \begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}.$$

Here the final diagonal matrix follows from the fact that the determinant of a polyphase matrix is one and n is even. Let us slightly rewrite the last equation. First observe that

$$\begin{bmatrix} q_i(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_i(z) & 1 \end{bmatrix}. \tag{5}$$

Using the first equation of (5) in case i is odd and the second in case i is even yields:

$$P^0(z) = \prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}. \tag{6}$$

Finally, the original filter g can be recovered by applying Theorem 3.

Now we know that the filter g can always be obtained from g^0 with one lifting or:

$$P(z) = P^0(z) \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix} \tag{7}$$

Combining all these observations we now have shown the following theorem:

Theorem 7. *Given a complementary filter pair (h, g), then there always exist Laurent polynomials $s_i(z)$ and $t_i(z)$ for $1 \leqslant i \leqslant m$ and a non-zero constant K so that*

$$P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}.$$

The proof follows from combining (6) and (7), setting $m = n/2 + 1$, $t_m(z) = 0$, and $s_m(z) = K^2\, s(z)$. In other words every finite filter wavelet transform can be obtained by starting with the Lazy wavelet followed by m lifting and dual lifting steps followed with a scaling.

The dual polyphase matrix is given by

$$\widetilde{P}(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & 0 \\ -s_i(z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -t_i(z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/K & 0 \\ 0 & K \end{bmatrix}.$$

From this we see that in the orthogonal case ($P(z) = \widetilde{P}(z)$) we immediately have two different factorizations.

Figures 7 and 8 represent the different steps of the forward and inverse transform schematically.

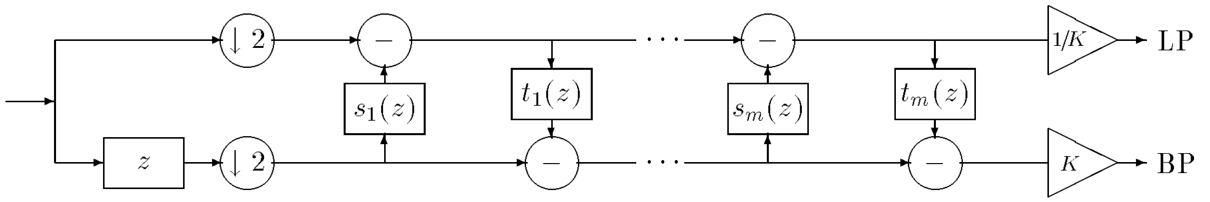

FIGURE 7. *The forward wavelet transform using lifting: First the Lazy wavelet, then alternating lifting and dual lifting steps, and finally a scaling.*

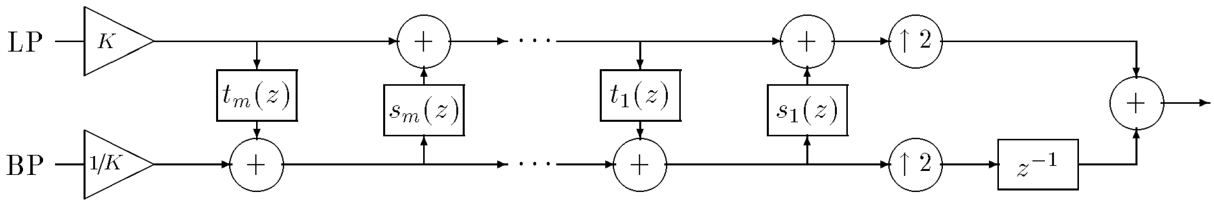

FIGURE 8. *The inverse wavelet transform using lifting: First a scaling, then alternating dual lifting and lifting steps, and finally the inverse Lazy transform. The inverse transform can immediately be derived from the forward by running the scheme backwards.*

7. EXAMPLES

We start with a few easy examples. We denote filters either by their canonical names (e.g. Haar), by $(N, \widetilde{N})$ where N (resp. $\widetilde{N}$) is the number of vanishing moments of $\widetilde{g}$ (resp. g), or by $(l_a - l_s)$ where l_a is the length of analysis filter $\widetilde{h}$ and l_s is the length of the synthesis filter h. We start with a sequence $x = \{x_l \mid l \in \mathbf{Z}\}$ and denote the result of applying the low-pass filter h (resp. high-pass filter g) and downsampling as a sequence $s = \{s_l \mid l \in \mathbf{Z}\}$ (resp. d).

The intermediate values computed during lifting we denote with sequences $s^{(i)}$ and $d^{(i)}$. All transforms are instances of Figure 7.

7.1. **Haar wavelets.** In the case of (unnormalized) Haar wavelets we have that $h(z) = 1 + z^{-1}$, $g(z) = -1/2 + 1/2z^{-1}$, $\widetilde{h}(z) = 1/2 + 1/2z^{-1}$, and $\widetilde{g}(z) = -1 + 1z^{-1}$. Using the Euclidean algorithm we can thus write the polyphase matrix as:

$$P(z) = \begin{bmatrix} 1 & -1/2 \\ 1 & 1/2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1/2 \\ 0 & 1 \end{bmatrix}.$$

Thus on the analysis size we have:

$$P(z)^{-1} = \widetilde{P}(1/z) = \begin{bmatrix} 1 & 1/2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}.$$

This corresponds to the following implementation of the forward transform:

$$\begin{aligned} s_l^{(0)} &= x_{2l} \\ d_l^{(0)} &= x_{2l+1} \\ d_l &= d_l^{(0)} - s_l^{(0)} \\ s_l &= s_l^{(0)} + 1/2\, d_l, \end{aligned}$$

while the inverse transform is given by:

$$\begin{aligned} s_l^{(0)} &= s_l - 1/2\, d_l \\ d_l^{(0)} &= d_l + s_l^{(0)} \\ x_{2l+1} &= d_l^{(0)} \\ x_{2l} &= s_l^{(0)}. \end{aligned}$$

7.2. **Givens rotations.** Consider the case where the polyphase matrix is a Givens rotation ($\alpha \neq \pi/2$). We then get

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \sin\alpha/\cos\alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & -\sin\alpha\cos\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 \\ 0 & 1/\cos\alpha \end{bmatrix}.$$

We can also do it without scaling with three lifting steps as (here assuming $\alpha \neq 0$)

$$\begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} = \begin{bmatrix} 1 & (\cos\alpha - 1)/\sin\alpha \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \sin\alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & (\cos\alpha - 1)/\sin\alpha \\ 0 & 1 \end{bmatrix}.$$

This corresponds to the well known fact in geometry that a rotation can always be written as three shears.

The lattice factorization of [51] allows the decomposition of any orthonormal filter pair into shifts and Givens rotations. It follows that any orthonormal filter can be written as lifting steps, by first writing the lattice factorization and then using the example above. This provides a different proof of Theorem 7 in the orthonormal case.

7.3. **Scaling.** These two examples show that the scaling from Theorem 7 can be replaced with four lifting steps:

$$P(z) = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} = \begin{bmatrix} 1 & K - K^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1/K & 1 \end{bmatrix} \begin{bmatrix} 1 & K - 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

or

$$P(z) = \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 - 1/K \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ K & 1 \end{bmatrix} \begin{bmatrix} 1 & 1/K^2 - 1/K \\ 0 & 1 \end{bmatrix}.$$

Given that one can always merge one of the four lifting steps with the last lifting step from the factorization, only three extra steps are needed to avoid scaling. This is particularly important when building integer to integer wavelet transforms in which case scaling is not invertible [6].

7.4. **Interpolating filters.** In case the low-pass filter is half band, or $h(z) + h(-z) = 2$, the corresponding scaling function is interpolating. Since $h_e(z) = 1$, the factorization can be done in two steps:

$$P(z) = \begin{bmatrix} 1 & g_e(z) \\ h_o(z) & 1 + h_o(z)\, g_e(z) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h_o(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & g_e(z) \\ 0 & 1 \end{bmatrix}.$$

The filters constructed in [44] are of this type. This gives rise to a family of $(N, \widetilde{N})$ (N and $\widetilde{N}$ even) symmetric biorthogonal wavelets built from the Deslauriers–Dubuc scaling functions mentioned in the introduction. The degrees of the filters are $|h_o| = N-1$ and $|g_e| = \widetilde{N}-1$. In case $\widetilde{N} \leqslant N$, these are particularly easy as $g_e^{(\widetilde{N})}(z) = -1/2\, h_o^{(\widetilde{N})}(z^{-1})$. (Beware: the normalization used here is different from the one in [44].)

Next we look at some examples that had not been decomposed into lifting steps before.

7.5. **4-tap orthonormal filter with two vanishing moments (D4).** Here the h and g filters are given by [16]:

$$h(z) = h_0 + h_1\, z^{-1} + h_2\, z^{-2} + h_3\, z^{-3}$$
$$g(z) = -h_3\, z^2 + h_2\, z^1 - h_1 + h_0\, z^{-1},$$

with

$$h_0 = \frac{1+\sqrt{3}}{4\sqrt{2}}, \quad h_1 = \frac{3+\sqrt{3}}{4\sqrt{2}}, \quad h_2 = \frac{3-\sqrt{3}}{4\sqrt{2}}, \text{ and } h_3 = \frac{1-\sqrt{3}}{4\sqrt{2}}.$$

The polyphase matrix is

$$P(z) = \widetilde{P}(z) = \begin{bmatrix} h_0 + h_2\, z^{-1} & -h_3\, z^1 - h_1 \\ h_1 + h_3\, z^{-1} & h_2 z^1 + h_0 \end{bmatrix}, \tag{8}$$

and the factorization is given by:

$$P(z) = \widetilde{P}(z) = \begin{bmatrix} 1 & -\sqrt{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4} z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix}. \tag{9}$$

As we pointed out in Section 6 we have two options. Because the polyphase matrix is unitary, we can use (9) as a factorization for either $P(z)$ or $\widetilde{P}(z)$. In the latter case the analysis polyphase matrix is factored as:

$$\widetilde{P}(1/z)^t = \begin{bmatrix} \frac{\sqrt{3}+1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}-1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{\sqrt{3}}{4} + \frac{\sqrt{3}-2}{4} z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\sqrt{3} & 1 \end{bmatrix}.$$

This corresponds to the following implementation for the forward transform:

$$\begin{aligned} d_l^{(1)} &= x_{2l+1} - \sqrt{3}\, x_{2l} \\ s_l^{(1)} &= x_{2l} + \sqrt{3}/4\, d_l^{(1)} + (\sqrt{3}-2)/4\, d_{l+1}^{(1)} \\ d_l^{(2)} &= d_l^{(1)} + s_{l-1}^{(1)} \\ s_l &= (\sqrt{3}+1)/\sqrt{2}\, s_l^{(1)} \\ d_l &= (\sqrt{3}-1)/\sqrt{2}\, d_l^{(2)}. \end{aligned}$$

The inverse transform follows from reversing the operations and flipping the signs:

$$\begin{aligned} d_l^{(2)} &= (\sqrt{3}+1)/\sqrt{2}\, d_l \\ s_l^{(1)} &= (\sqrt{3}-1)/\sqrt{2}\, s_l \\ d_l^{(1)} &= d_l^{(2)} - s_{l-1}^{(1)} \\ x_{2l} &= s_l^{(1)} - \sqrt{3}/4\, d_l^{(1)} - (\sqrt{3}-2)/4\, d_{l+1}^{(1)} \\ x_{2l+1} &= d_l^{(1)} + \sqrt{3}\, x_{2l}. \end{aligned}$$

The other option is to use (9) as a factorization for $P(z)$. The analysis polyphase matrix then is factored as:

$$P(z)^{-1} = \begin{bmatrix} \frac{\sqrt{3}-1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}+1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & -z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{\sqrt{3}}{4} - \frac{\sqrt{3}-2}{4} z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & \sqrt{3} \\ 0 & 1 \end{bmatrix},$$

and leads to the following implementation of the forward transform:

$$\begin{aligned} s_l^{(1)} &= x_{2l} + \sqrt{3}\, x_{2l+1} \\ d_l^{(1)} &= x_{2l+1} - \sqrt{3}/4\, s_l^{(1)} - (\sqrt{3}-2)/4\, s_{l-1}^{(1)} \\ s_l^{(2)} &= s_l^{(1)} - d_{l+1}^{(1)} \\ s_l &= (\sqrt{3}-1)/\sqrt{2}\, s_l^{(1)} \\ d_l &= (\sqrt{3}+1)/\sqrt{2}\, d_l^{(2)}. \end{aligned}$$

Given that the inverse transform always follows immediately from the forward transform, from now on we only give the forward transform.

One can also obtain an entirely different lifting factorization of D4 by shifting the filter pair corresponding to:

$$\begin{aligned} h(z) &= h_0\, z + h_1 + h_2\, z^{-1} + h_3\, z^{-2} \\ g(z) &= h_3\, z - h_2 + h_1\, z^{-1} - h_0\, z^{-2}, \end{aligned}$$

with

$$\widetilde{P}(z) = P(z) = \begin{bmatrix} h_1 + h_3\, z^{-1} & -h_2 - h_0\, z^{-1} \\ h_0\, z + h_2 & h_3 z + h_1 \end{bmatrix}$$

as polyphase matrix. This leads to a different factorization:

$$\widetilde{P}(z) = \begin{bmatrix} 1 & -\frac{1}{\sqrt{3}}\, z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\sqrt{3}}{4} z + \frac{6-3\sqrt{3}}{4} & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{1}{3} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{3+\sqrt{3}}{3\sqrt{2}} & 0 \\ 0 & \frac{3-\sqrt{3}}{3\sqrt{2}} \end{bmatrix},$$

and corresponds to the following implementation:

$$\begin{aligned} d_l^{(1)} &= x_{2l+1} - 1/\sqrt{3}\, x_{2l+2} \\ s_l^{(1)} &= x_{2l} + (6-3\sqrt{3})/4\, d_l^{(1)} + \sqrt{3}/4\, d_{l-1}^{(1)} \\ d_l^{(2)} &= d_l^{(1)} - 1/3\, s_l^{(1)} \\ s_l &= (3+\sqrt{3})/(3\sqrt{2})\, s_l^{(1)} \\ d_l &= (3-\sqrt{3})/(3\sqrt{2})\, d_l^{(2)}. \end{aligned}$$

This second factorization can also be obtained as the result of seeking a factorization of the original polyphase matrix (8) where the final diagonal matrix has (non-constant) monomial entries.

7.6. **6-tap orthonormal filter with three vanishing moments (D6).** Here we have

$$h(z) = \sum_{k=-2}^{3} h_k \, z^{-k},$$

with [16]

$$h_{-2} = \sqrt{2}\left(1+\sqrt{10}+\sqrt{5+2\sqrt{10}}\right)/32 \qquad h_{-1} = \sqrt{2}\left(5+\sqrt{10}+3\sqrt{5+2\sqrt{10}}\right)/32$$
$$h_{0} = \sqrt{2}\left(10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}}\right)/32 \qquad h_{1} = \sqrt{2}\left(10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}}\right)/32$$
$$h_{2} = \sqrt{2}\left(5+\sqrt{10}-3\sqrt{5+2\sqrt{10}}\right)/32 \qquad h_{3} = \sqrt{2}\left(1+\sqrt{10}-\sqrt{5+2\sqrt{10}}\right)/32.$$

The polyphase components are

$$h_e(z) = h_{-2}\, z + h_0 + h_2\, z^{-1} \qquad g_e(z) = -h_3\, z - h_1 - h_{-1}\, z^{-1}$$
$$h_o(z) = h_{-1}\, z + h_1 + h_3\, z^{-1} \qquad g_o(z) = h_2\, z + h_0 + h_{-2}\, z^{-1}.$$

In the factorization algorithm the coefficients of the remainders are calculated as:

$$\begin{aligned} r_0 &= h_{-1} - h_3 * h_{-2}/h_2 \\ r_1 &= h_1 - h_2 * h_0/h_2 \\ s_1 &= h_0 - h_{-2} * r_1/r_0 - h_2 * r_0/r_1 \\ t &= -h_3/h_{-2} * s_1^2. \end{aligned}$$

If we now let

$$\begin{aligned} \alpha &= h_3/h_1 \approx -0.4122865950 \\ \beta &= h_2/r_1 \approx -1.5651362796 \\ \beta' &= h_{-2}/r_0 \approx 0.3523876576 \\ \gamma &= r_1/s_1 \approx 0.0284590896 \\ \gamma' &= r_0/s_1 \approx 0.4921518449 \\ \delta &= -h_3/h_{-2} * s_1^2 \approx -0.3896203900 \\ \zeta &= s_1 \approx 1.9182029462, \end{aligned}$$

then the factorization is given by:

$$P(z) = \begin{bmatrix} 1 & 0 \\ \alpha & 1 \end{bmatrix} \begin{bmatrix} 1 & \beta z^{-1} + \beta' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma + \gamma' z & 1 \end{bmatrix} \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix}.$$

We leave the implementation of this filter as an exercise for the reader.

7.7. **(9-7) filter.** Here we consider the popular (9-7) filter pair. The analysis filter $\widetilde{h}$ has 9 coefficients, while the synthesis filter h has 7 coefficients. Both high-pass filters g and $\widetilde{g}$ have 4 vanishing moments. We choose the filter with 7 coefficients to be the synthesis filter because it gives rises to a smoother scaling function than the 9 coefficient one (see [17, p. 279, Table 8.3], note that the coefficients need to be multiplied with $\sqrt{2}$). For this example we run the factoring algorithm starting from the analysis filter:

$$\tilde{h}_e(z) = h_4\,(z^2 + z^{-2}) + h_2\,(z + z^{-1}) + h_0 \quad \text{and} \quad \tilde{h}_o(z) = h_3\,(z^2 + z^{-1}) + h_1\,(z + 1).$$

The coefficients of the remainders are computed as:

$$\begin{aligned} r_0 &= h_0 - 2\,h_4\,h_1/h_3 \\ r_1 &= h_2 - h_4 - h_4\,h_1/h_3 \\ s_0 &= h_1 - h_3 - h_3\,r_0/r_1 \\ t_0 &= r_0 - 2\,r_1. \end{aligned}$$

Then define

$$\begin{aligned} \alpha &= h_4/h_3 \approx -1.586134342 \\ \beta &= h_3/r_1 \approx -0.05298011854 \\ \gamma &= r_1/s_0 \approx 0.8829110762 \\ \delta &= s_0/t_0 \approx 0.4435068522 \\ \zeta &= t_0 = r_0 - 2\,r_1 \approx 1.149604398. \end{aligned}$$

Now

$$\widetilde{P}(z) = \begin{bmatrix} 1 & \alpha(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \beta(1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & \gamma(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \delta(1+z) & 1 \end{bmatrix} \begin{bmatrix} \zeta & 0 \\ 0 & 1/\zeta \end{bmatrix}.$$

Note that here too many other factorizations exist; the one we chose is symmetric: every quotient is a multiple of $(z+1)$. This shows how we can take advantage of the non-uniqueness to maintain symmetry. The factorization leads to the following implementation:

$$
\begin{aligned}
s_l^{(0)} &= x_{2l} \\
d_l^{(0)} &= x_{2l+1} \\
d_l^{(1)} &= d_l^{(0)} + \alpha\,(s_l^{(0)} + s_{l+1}^{(0)}) \\
s_l^{(1)} &= s_l^{(0)} + \beta\,(d_l^{(1)} + d_{l-1}^{(1)}) \\
d_l^{(2)} &= d_l^{(1)} + \gamma\,(s_l^{(1)} + s_{l+1}^{(1)}) \\
s_l^{(2)} &= s_l^{(1)} + \delta\,(d_l^{(2)} + d_{l-1}^{(2)}) \\
s_l &= \zeta\, s_l^{(2)} \\
d_l &= d_l^{(2)}/\zeta.
\end{aligned}
$$

7.8. **Cubic B-splines.** We finish with an example that is used frequently in computer graphics: the (4,2) biorthogonal filter from [12]. The scaling function here is a cubic B-spline. This example can be obtained again by using the factoring algorithm. However, there is also a much more intuitive construction in the spatial domain [46]. The filters are given by

$$
\begin{aligned}
h(z) &= 3/4 + 1/2\,(z + z^{-1}) + 1/8\,(z^2 + z^{-2}) \\
g(z) &= 5/4 z^{-1} - 5/32\,(1 + z^{-2}) - 3/8\,(z + z^{-3}) - 3/32\,(z^2 + z^{-4}),
\end{aligned}
$$

and the factorization reads:

$$
P(z) = \begin{bmatrix} 1 & 1/4(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ (1+z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -3/16(1+z^{-1}) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1/2 & 0 \\ 0 & 2 \end{bmatrix}.
$$

8. Computational complexity

In this section we take a closer look at the computational complexity of the wavelet transform computed using lifting. As a comparison base we use the standard algorithm, which corresponds to applying the polyphase matrix. This already takes advantage of the fact that the filters will be subsampled and thus avoids computing samples that will be subsampled immediately. The unit we use is the cost, measured in number of multiplications and additions, of computing one sample pair (s_l, d_l). The cost of applying a filter h is $|h| + 1$ multiplications and $|h|$ additions. The cost of the standard algorithm thus is $2(|h| + |g|) + 2$. If the filter is symmetric and $|h|$ is even, the cost is $3\,|h|/2 + 1$.

Let us consider a general case not involving symmetry. Take $|h| = 2N$, $|g| = 2M$, and assume $M \geqslant N$. The cost of the standard algorithm now is $4(N + M) + 2$. Without loss of generality we can assume that $|h_e| = N$, $|h_o| = N - 1$, $|g_e| = M$, and $|g_o| = M - 1$. In general the Euclidean algorithm started from the (h_e, h_o) pair now needs N steps with the degree of each quotient equal to one ($|q_i| = 1$ for $1 \leqslant i \leqslant N$). To get the (g_e, g_o) pair, one extra lifting step (7) is needed with $|s| = M - N$. The total cost of the lifting algorithm is:

scaling:	2
N lifting steps:	$4N$
final lifting step:	$2(M - N + 1)$
total	$2(N + M + 2)$

We have shown the following:

Theorem 8. *Asymptotically, for long filters, the cost of the lifting algorithm for computing the wavelet transform is one half of the cost of the standard algorithm.*

In the above reasoning we assumed that the Euclidean algorithm needs exactly N steps with each quotient of degree one. In a particular situation the Euclidean algorithm might need fewer than N steps but with larger quotients. The interpolating filters form an extreme case; with two steps one can build arbitrarily long filters. However, in this case Theorem 8 holds as well; the cost for the standard algorithm is $3(N + \widetilde{N}) - 2$ while the cost of the lifting algorithm is $3/2(N + \widetilde{N})$.

Of course, in any particular case the numbers can differ slightly. Table 1 gives the cost S of the standard algorithm, the cost L of the lifting algorithm, and the relative speedup $(S/L - 1)$ for the examples in the previous section.

Wavelet	Standard	Lifting	Speedup
Haar	3	3	0%
D4	14	9	56%
D6	22	14	57%
(9-7)	23	14	64%
(4,2) B-spline	17	10	70%
$(N, \widetilde{N})$ Interpolating	$3(N + \widetilde{N}) - 2$	$3/2(N + \widetilde{N})$	$\approx$ 100%
$\|h\| = 2N$, $\|g\| = 2M$	$4(N + M) + 2$	$2(N + M + 2)$	$\approx$ 100%

TABLE 1. *Computational cost of lifting versus the standard algorithm. Asymptotically the lifting algorithm is twice as fast as the standard algorithm.*

One has to be careful with this comparison. Even though it is widely used, the standard algorithm is not necessarily the best way to implement the wavelet transform. Lifting is only one idea in a whole tool bag of methods to improve the speed of a fast wavelet transform. Rioul and Duhamel [39] discuss several other schemes to improve the standard algorithm. In the case of long filters, they suggest an FFT based scheme known as the Vetterli-algorithm [56]. In the case of short filters, they suggest a "fast running FIR" algorithm [54]. How these ideas combine with the idea of using lifting and which combination will be optimal for a certain wavelet goes beyond the scope of this paper and remains a topic of future research.

9. Conclusion and Comments

In this tutorial presentation, we have shown how every wavelet filter pair can be decomposed into lifting steps. The decomposition amounts to writing arbitrary elements of the ring $\mathrm{SL}(2; \mathbf{R}[z, z^{-1}])$ as products of elementary matrices, something that has been known to be possible for a long time [2].

The following are a few comments on the decomposition and its usefulness. First of all, the decomposition of arbitrary wavelet transforms into lifting steps implies that we can gain, for *all* wavelet transforms, the traditional advantages of lifting implementations, i.e.

1. Lifting leads to a speed-up when compared to the standard implementation.
2. Lifting allows for an in-place implementation of the fast wavelet transform, a feature similar to the Fast Fourier Transform. This means the wavelet transform can be calculated without allocating auxiliary memory.
3. All operations within one lifting step can be done entirely parallel while the only sequential part is the order of the lifting operations.
4. Using lifting it is particularly easy to build non linear wavelet transforms. A typical example are wavelet transforms that map integers to integers [6]. Such transforms are important for hardware implementation and for lossless image coding.
5. Using lifting and integer-to-integer transforms, it is possible to combine biorthogonal wavelets with scalar quantization and still keep cubic quantization cells which are optimal like in the orthogonal case. In a multiple description setting, it has been shown that this generalization to biorthogonality allows for substantial improvements [58].
6. Lifting allows for adaptive wavelet transforms. This means one can start the analysis of a function from the coarsest levels and then build the finer levels by refining only in the areas of interest, see [40] for a practical example.

The decomposition in this paper also suggests the following comments and raises a few open questions:

1. Factoring into lifting steps is a highly non-unique process. We do not know exactly how many essentially different factorizations are possible, how they differ, and what is a good strategy for picking the "best one"; this is an interesting topic for future research.
2. The main result of this paper also holds in case the filter coefficients are not necessarily real, but belong to any field such as the rationals, the complex numbers, or even a finite field. However, the Euclidean algorithm does *not* work when the filter coefficients themselves belong to a ring such as the integers or the dyadic numbers. It is thus not guaranteed

that filters with binary coefficients can be factored into lifting steps with binary filter coefficients.

3. In this paper we never concerned ourselves with whether filters were causal, i.e., only had filter coefficients for $k \geqslant 0$. Given that all subband filters here are finite, causality can always be obtained by shifting the filters. Obviously, if both analysis and synthesis filters have to be causal, perfect reconstruction is only possible up to a shift. By executing the Euclidean algorithm over the ring of polynomials, as opposed to the ring of Laurent polynomials, it can be assured that then all lifting steps are causal as well.
4. The long division used in the Euclidean algorithm guarantees that, except for at most one quotient of degree 0, all the quotients will be at least of degree 1 and the lifting filters thus contain at least 2 coefficients. In some cases, e.g., hardware implementations, it might be useful to use *only* lifting filters with at *most* 2 coefficients. Then, in each lifting step, an even location will only get information from its two immediate odd neighbors or vice versa. Such lifting steps can be obtained by not using a full long division, but rather stopping the division as soon as the quotient has degree one. The algorithm still is guaranteed to terminate as the degree of the polyphase components will decrease by exactly 1 in each step. We are now guaranteed to be in the setting used to sketch the proof of Theorem 8.
5. In the beginning of this paper, we pointed out how lifting is related to the multiscale transforms and the associated stability analysis developed by Wolfgang Dahmen and co-workers. Although their setting looks more general than lifting since it allows for a non-identity operator K on the diagonal of the polyphase matrix, while lifting requires identities on the diagonal, this paper shows that, in the first generation or time invariant setting, no generality is lost by restricting oneself to lifting. Indeed, any invertible polyphase matrix with a non-identity polynomial $K(z)$ on the diagonal can be obtained using lifting. Note that some of the advantages of lifting mentioned above rely fundamentally on the $K = I$ and disappear when allowing a general K.
6. This factorization generalizes to the M-band setting. It is known that a $M \times M$ polyphase matrix with elements in a Euclidean domain and with determinant one can be reduced to an identity matrix using elementary row and column operations, see [24, Theorem 7.10]. This reduction, also known as the Smith normal form, allows for lifting factorizations in the M-band case. In [48] the discussion of the decomposition into ladder steps (which is the analog, in different notation, of what we have called here the factorization into lifting steps) is carried out for the general M-band case; please check this paper for details and applications.
7. Finally, under certain conditions it is possible to construct ladder like structures in higher dimensions using factoring of multivariate polynomials. For details, we refer to [37].

Acknowledgments. The authors would like to thank Peter Schröder and Boon-Lock Yeo for many stimulating discussions and for their help in computing the factorizations in the example section, Jelena Kovačević and Martin Vetterli for drawing their attention to reference [28], Paul Van Dooren for pointing out the connection between the M-band case and the Smith normal form, and Geert Uytterhoeven, Avraham Melkman, Mark Maslen, and Paul Abbott for pointing out typos and oversights in an earlier version.

Ingrid Daubechies would like to thank NSF (grant DMS-9401785), AFOSR (grant F49620-95-1-0290), ONR (grant N00014-96-1-0367) as well as Lucent Technologies, Bell Laboratories for partial support while conducting the research for this paper. Wim Sweldens is on leave as Senior Research Assistant of the Belgian Fund of Scientific Research (NFWO).

References

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

[2] H. Bass. *Algebraic K-theory.* W. A. Benjamin, Inc., New York, 1968.

[3] M. G. Bellanger and J. L. Daguet. TDM-FDM transmultiplexer: Digital polyphase and FFT. *IEEE Trans. Commun.*, 22(9):1199–1204, 1974.

[4] R. E. Blahut. *Fast Algorithms for Digital Signal Processing.* Addison-Wesley, Reading, MA, 1984.

[5] A. A. M. L. Bruekens and A. W. M. van den Enden. New networks for perfect inversion and perfect reconstruction. *IEEE J. Selected Areas Commun.*, 10(1), 1992.

[6] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal.*, 5(3):332–369, 1998.

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

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

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

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

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

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

[13] J. M. Combes, A. Grossmann, and Ph. Tchamitchian, editors. *Wavelets: Time-Frequency Methods and Phase Space.* Inverse problems and theoretical imaging. Springer-Verlag, New York, 1989.

[14] 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.

[15] W. Dahmen, S. Prössdorf, and R. Schneider. Multiscale methods for pseudo-differential equations on smooth manifolds. In [9], pages 385–424. 1994.

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

[17] 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.

[18] I. Daubechies, A. Grossmann, and Y. Meyer. Painless nonorthogonal expansions. *J. Math. Phys.*, 27(5):1271–1283, 1986.

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

[20] R. E. Van Dyck, T. G. Marshall, M. Chine, and N. Moayeri. Wavelet video coding with ladder structures and entropy-constrained quantization. *IEEE Trans. Circuits Systems Video Tech.*, 6(5):483–495, 1996.

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

[22] A. Grossmann and J. Morlet. Decompostion of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15(4):723–736, 1984.

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

[24] B. Hartley and T. O. Hawkes. *Rings, Modules and Linear Algebra.* Chapman and Hall, New York, 1983.

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

[26] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[27] N. S. Jayant and P. Noll. *Digital coding of waveforms.* Prentice Hall, Englewood Cliffs, NJ, 1984.

[28] T. A. C. M. Kalker and I. Shah. Ladder Structures for multidimensional linear phase perfect reconstruction filter banks and wavelets. In *Proceedings of the SPIE Conference on Visual Communications and Image Processing* (Boston), pages 12–20, 1992.

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

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

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

[32] T. G. Marshall. A fast wavelet transform based upon the Euclidean algorithm. In *Conference on Information Science and Systems*, Johns Hopkins, MD, 1993.

[33] T. G. Marshall. U-L block-triangular matrix and ladder realizations of subband coders. In *Proc. IEEE ICASSP*, volume III, pages 177–180, 1993.

[34] 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.

[35] F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE Trans. Acoust. Speech Signal Process.*, 33:626–630, 1985.

[36] T. Q. Nguyen and P. P. Vaidyanathan. Two-channel perfect-reconstruction FIR QMF structures which yield linear-phase analysis and synthesis filters. *IEEE Trans. Acoust. Speech Signal Process.*, 37:676–690, 1989.

[37] H.-J. Park. *A computational theory of Laurent polynomial rings and multidimensional FIR systems.* PhD thesis, University of California, Berkeley, May 1995.

[38] L.-M. Reissell. Wavelet multiresolution representation of curves and surfaces. *CVGIP: Graphical Models and Image Processing*, 58(2):198–217, 1996.

[39] O. Rioul and P. Duhamel. Fast algorithms for discrete and continuous wavelet transforms. *IEEE Trans. Inform. Theory*, 38(2):569–586, 1992.

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

[41] I. Shah and T. A. C. M. Kalker. On Ladder Structures and Linear Phase Conditions for Bi-Orthogonal Filter Banks. In *Proceedings of ICASSP-94*, volume 3, pages 181–184, 1994.

[42] M. J. T. Smith and T. P. Barnwell. Exact reconstruction techniques for tree-structured subband coders. *IEEE Trans. Acoust. Speech Signal Process.*, 34(3):434–441, 1986.

[43] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley, Cambridge, 1996.

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

[45] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

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

[47] J. Tian and R. O. Wells. Vanishing moments and biorthogonal wavelet systems. In *Mathematics in Signal Processing IV*. Institute of Mathematics and Its Applications Conference Series, Oxford University Press, 1996.

[48] L. M. G. Tolhuizen, H. D. L. Hollmann, and T. A. C. M. Kalker. On the realizability of bi-orthogonal M-dimensional 2-band filter banks. *IEEE Transactions on Signal processing*, 1995.

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

[50] P. P. Vaidyanathan. Theory and design of M-channel maximally decimated quadrature mirror filters with arbitrary M, having perfect reconstruction property. *IEEE Trans. Acoust. Speech Signal Process.*, 35(2):476–492, 1987.

[51] P. P. Vaidyanathan and P.-Q. Hoang. Lattice structures for optimal design and robust implementation of two-band perfect reconstruction QMF banks. *IEEE Trans. Acoust. Speech Signal Process.*, 36:81–94, 1988.

[52] 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.

[53] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Process.*, 10:219–244, 1986.

[54] M. Vetterli. Running FIR and IIR filtering using multirate filter banks. *IEEE Trans. Signal Process.*, 36:730–738, 1988.

[55] M. Vetterli and D. Le Gall. Perfect reconstruction FIR filter banks: Some properties and factorizations. *IEEE Trans. Acoust. Speech Signal Process.*, 37:1057–1071, 1989.

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

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

[58] Y. Wang, M. Orchard, A. Reibman, and V. Vaishampayan. Redundancy rate-distortion analysis of multiple description coding using pairwise correlation transforms. In *Proc. IEEE ICIP*, volume I, pages 608–611, 1997.

[59] J. W. Woods and S. D. O'Neil. Subband coding of images. *IEEE Trans. Acoust. Speech Signal Process.*, 34(5):1278–1288, 1986.