FACTORING WAVELET TRANSFORMS INTO LIFTING STEPS

INGRID DAUBECHIES† AND WIM SWELDENS*

September 1996

ABSTRACT. The lifting scheme is a new flexible tool for constructing wavelets and wavelet transforms. In this paper, we use the Euclidean algorithm to show how any discrete wavelet transform or two band subband transform with finite filters can be obtained with a finite number of lifting steps starting from the Lazy wavelet (or polyphase transform). We show a bound on the number of lifting steps that is proportional to the length of the filters. This factorization provides an alternative for the lattice factorization, with the advantage that it can also be used in the biorthogonal (non-unitary) case. The lifting factorization asymptotically reduces the computational complexity of the transform by a factor of two and allows for wavelet transforms that map 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 [36, 37, 41, 51, 52, 53, 54, 58, 56, 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. [15, 20, 25, 35, 24]. In the late eighties the introduction of multiresolution analysis and the fast wavelet transform by Mallat and Meyer provided the connection between subband filters and wavelets [33, 34, 35]; this led to the first construction of smooth, orthogonal, and compactly supported wavelets in 1987 [18]. Later many generalizations to the biorthogonal or semiorthogonal (pre-wavelet) case were introduced. Biorthogonality allows the construction of symmetric wavelets and thus linear phase filters. Examples are: the construction of semiorthogonal spline wavelets [1, 10, 12, 13, 50], fully biorthogonal compactly supported wavelets [14, 57], and recursive filter banks [28].

Recently a new angle to study these constructions was provided by the so-called "lifting scheme" [46]. The basic idea behind lifting is that it provides a simple relationship between all multiresolution analyses that share the same low-pass filter or high pass filter. The low-pass filter gives the coefficients of the refinement relation which entirely determines the scaling

†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.

function. The wavelet can be found as a linear combination of scaling functions where the coefficients are given by the high-pass filter. In [46] it is shown how lifting can be used for painless custom-design of wavelets. As an example a family of biorthogonal wavelets associated to the interpolating Deslauriers-Dubuc scaling functions [21] was derived. These wavelets can be thought of as biorthogonal Coiflets; they were also derived independently, but without the use of lifting, by several people: Reissell [38], Tian and Wells [48], and Strang [43]. Two software packages for this family are available: [23] uses lifting, while [5] does not. The advantages of lifting are numerous:

1. 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 [45].
2. Using lifting it is particularly easy to build non linear wavelet transforms. A typical example are wavelet transforms that map integers to integers [8]. Such transforms are important for hardware implementation and for lossless image coding.
3. Lifting allows the construction of wavelets without making use of the Fourier transform. This means that it can be used for building wavelets that are not necessarily translates and dilates of one function, so-called "second generation wavelets" [44]. In fact this was the original motivation behind the development of lifting. Typical examples are wavelets adjusted to weight functions, to irregular samples [47], or manifolds, see [40] for the construction of wavelets on a sphere. In this paper we use the term classical wavelets or first generation wavelets for wavelets formed by translation and dilation.
4. Every transform built with lifting is immediately invertible where the inverse transform has exactly the same computational complexity as the forward transform.
5. 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.
6. Lifting exposes the parallelism inherent in a wavelet transform. All operations within one lifting step can be done entirely parallel while the only sequential part is the order of the lifting operations.
7. Lifting does not rely on the Fourier transforms and can be introduced with using only arguments in the spatial domain. It thus allows an easy way to introduce wavelets which is particularly useful for people without a strong mathematical background [47].

The ideas behind lifting are not entirely new and have close connections with several earlier and/or independent developments.

- The lifting scheme is inspired by the work of Donoho [22] and Lounsbery et al. [32]. Donoho [22] shows how to built wavelets built from interpolating scaling functions, while Lounsbery et al. built a multiresolution analysis of surfaces using a technique that is algebraically the same as lifting.

- The technique of Vetterli en Herley [57] to build biorthogonal wavelet filters is another predecessor of lifting. Their Proposition 4.7 is the key behind lifting in the first generation setting. It turns out that the same lemma was also used for the construction of filter banks in [49] and in [31].
- Dahmen and collaborators, independently of lifting, worked on stable completions of multiscale transforms, a setting similar to second generation wavelets [9, 17]. Again independently, both of Dahmen and of lifting, Harten developed a general multiresolution approximation framework based on prediction [26].
- In [16], Dahmen and Micchelli propose a construction of compactly supported wavelets that generate complementary spaces in a multiresolution analysis of univariate irregular knot splines.
- There are also close similarities between lifting and so-called “ladder” structures in filter bank design [7, 30]. After finishing this paper we found that in this context a factorization result, similar to the one presented in this paper, was obtained earlier by Kalker and Shah in an unpublished manuscript [29]. While our work goes into more detail concerning the non-uniqueness, implementation, and computational complexity, their work considers also the more general M-band setting.
- Finally, on a more abstract abstract level, lifting connects with ideas introduced in other research areas. For example the notion of computations that are guaranteed to be invertible has been a topic of study in the theory of computation, and led to the definition of so-called reversible gates. In cryptology, ciphers are typically built by splitting data in two pieces and alternatively adding non-linear functions that depend on one piece to the other piece, see e.g. the DES standard [42]. These schemes again guarantee invertibility and can be thought of as far cousins of lifting.

Because of the many advantages of lifting, it is natural to ask how general lifting is. What is the set of wavelet filter banks that can be obtained using lifting? This paper answers this question in the first generation setting. For some time it was assumed that only particular wavelet transforms could be obtained with lifting, typically biorthogonal ones associated with interpolating scaling functions. We show here that *every* wavelet or subband transform with finite filters can be obtained with a finite number of lifting steps.

Our result is constructive in the sense that it gives an easy procedure to compute the filters needed in each lifting step; the key ingredient is simply an application of the Euclidean algorithm to the ring of Laurent polynomials. In fact, connections between the Euclidean algorithm and the wavelet transform are not entirely new. Vetterli and Herley [57] use the Euclidean algorithm and the connection to diophantine equations to find all high pass filters, given a low-pass filter, that make a finite filter wavelet transform. A different connection between constructing wavelets, Bezout's theorem, and the Euclidean algorithm is pointed out in [19]. In a more general context, the Euclidean algorithm also plays a role in using an inverse scattering framework for problems in signal processing [6].

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 of two. Finally Section 9 contains comments.

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 2 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 $r(z)\,q(z)$ 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

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

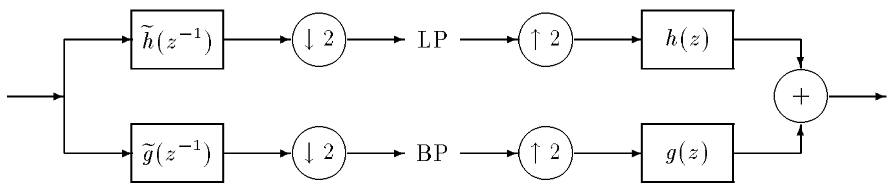

FIGURE 1. Discrete wavelet transform (or subband transform): The forward transform consists of two analysis filters $\widetilde{h}$ (low-pass) and $\widetilde{g}$ (high-pass) followed by subsampling, while the inverse transform first upsamples and then uses two synthesis filters h (low-pass) and g (high-pass).

3. Wavelet transforms

Figure 1 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 [58]. 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

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

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 $GL_2(\mathbf{R}[z, z^{-1}])$.

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.

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

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

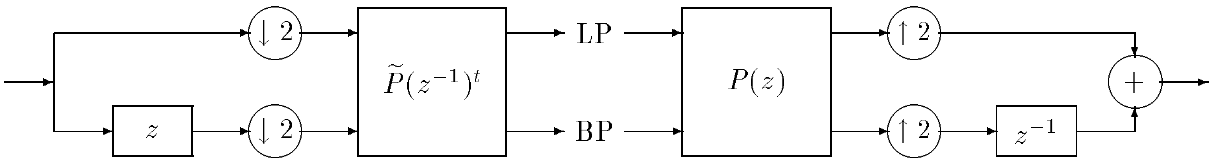

FIGURE 2. *Polyphase representation of wavelet transform: first subsample into even and odd, then apply the dual polyphase matrix. For the inverse transform: first apply the polyphase matrix and then join even and odd.*

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 2. 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 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 $GL_2(\mathbf{R}[z, z^{-1}])$. Without loss of generality we assume that $\det P(z) = 1$. Indeed, if the determinant is not one, we can always divide $g_e(z)$ and $\widetilde{g}(z)$ 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

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

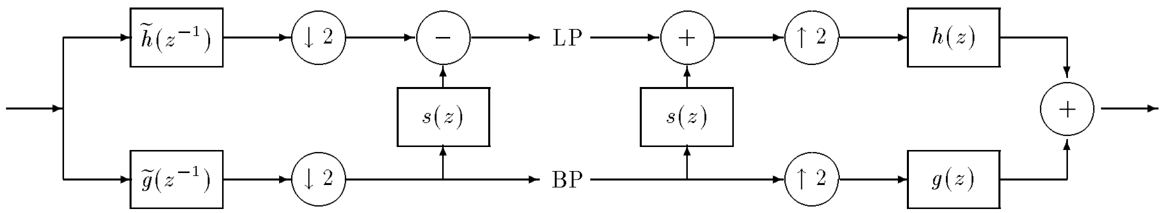

FIGURE 3. *The lifting scheme: First a classical subband filter scheme and then lifting the low-pass subband with the help of the high-pass subband.*

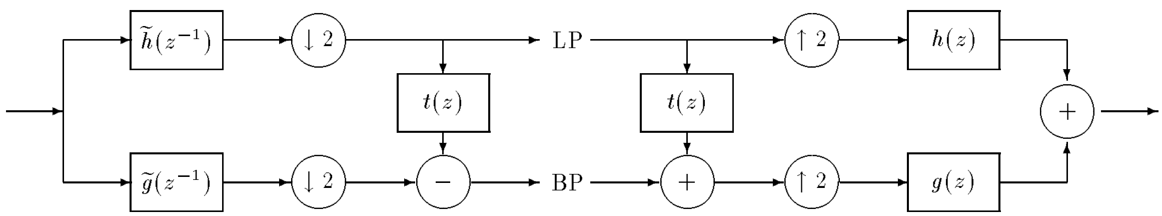

FIGURE 4. *The dual lifting scheme: First a classical subband filter scheme and later lifting the high-pass subband with the help of the low-pass subband.*

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 often is referred to as the Lazy wavelet transform [46]. (The reason is that the notion of the Lazy wavelet can also be used in the second generation setting.)

4. The Lifting Scheme

The lifting scheme [46, 44] 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

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

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

Figure 3 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 [57, Proposition 4.7]. The dual polyphase matrix is given by:

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

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

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

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

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

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

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

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

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

Figure 4 shows the schematic representation of dual lifting. In [46] 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):

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

$$b_1(z) = 4$$

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

Next step yields

$$\begin{aligned} a_2(z) &= 4 \\ b_2(z) &= 0 \\ q_2(z) &= 1 + z. \end{aligned}$$

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 doesn't 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 6 and 5 represent the different steps of the forward and inverse transform schematically.

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 6.

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:

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

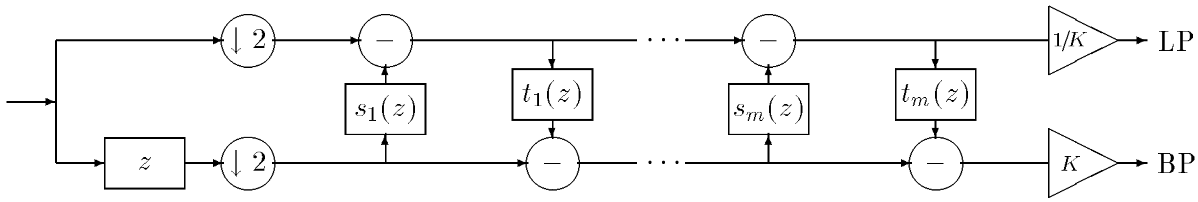

FIGURE 5. *The forward wavelet transform using lifting: First the Lazy wavelet, then alternating lifting and dual lifting steps, and finally a scaling.*

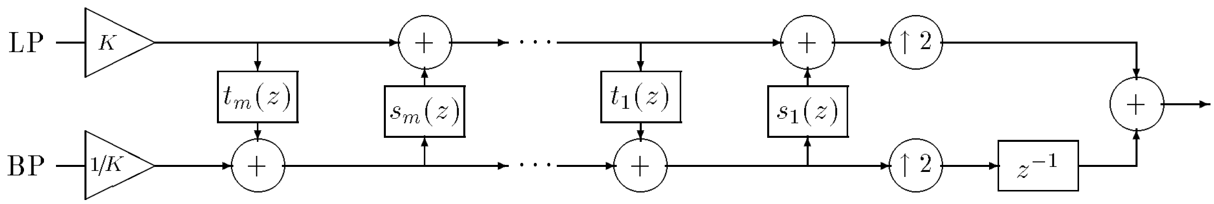

FIGURE 6. *The inverse wavelet transform using lifting: First a scaling, then alternating dual lifting and lifting steps, and finally the inverse Lazy transform. The inverse transform can immediately be derived from the forward by running the scheme backwards.*

This corresponds to the following implementation:

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

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 [52] allows the decomposition of any orthonormal filter pair into shifts and Givens rotations. It follows 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 transform in which case scaling is not invertible [8].

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 [46] 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 [46].)

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 [18]:

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

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

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

and the factorization is given by:

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

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:

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

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

One can obtain a 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

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

as polyphase matrix. This leads to a different factorization:

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

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 [18]

$$\begin{aligned}
h_{-2} &= \sqrt{2}\left(1+\sqrt{10}+\sqrt{5+2\sqrt{10}}\right)/32 & h_{-1} &= \sqrt{2}\left(5+\sqrt{10}+3\sqrt{5+2\sqrt{10}}\right)/32 \\
h_0 &= \sqrt{2}\left(10-2\sqrt{10}+2\sqrt{5+2\sqrt{10}}\right)/32 & h_1 &= \sqrt{2}\left(10-2\sqrt{10}-2\sqrt{5+2\sqrt{10}}\right)/32 \\
h_2 &= \sqrt{2}\left(5+\sqrt{10}-3\sqrt{5+2\sqrt{10}}\right)/32 & h_3 &= \sqrt{2}\left(1+\sqrt{10}-\sqrt{5+2\sqrt{10}}\right)/32.
\end{aligned}$$

The polyphase components are

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

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

$$\alpha = h_3/h_1 \approx -0.4122865950$$
$$\beta = h_2/r_1 \approx -1.5651362796$$
$$\beta' = h_{-2}/r_0 \approx 0.3523876576$$
$$\gamma = r_1/s_1 \approx 0.0284590896$$
$$\gamma' = r_0/s_1 \approx 0.4921518449$$
$$\delta = -h_3/h_{-2} * s_1^2 \approx -0.3896203900$$
$$\zeta = s1 \approx 1.9182029462,$$

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 [19]. 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:

$$r_0 = h_0 - 2\,h_4\,h_1/h_3$$
$$r_1 = h_2 - h_4 - h_4\,h_1/h_3$$
$$s_0 = h_1 - h_3 - h_3\,r_0/r_1.$$

Then define

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

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 [14]. 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 [47]. The filters are given by

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

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.

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 [57]. In the case of short filters, they suggest a "fast running FIR" algorithm [55]. 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.

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	$3(N+\widetilde{N})-2$	$3/2(N+\widetilde{N})$	$\approx 100\%$
$\lvert h\rvert = 2N$, $\lvert g\rvert = 2M$	$4(N+M)+2$	$2(N+M+2)$	$\approx 100\%$

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

9. COMMENTS

1. From an algebraic point of view the main result of this paper is not new. It has been known for a long time that elements of the ring $\mathrm{GL}_2(\mathbf{R}[z, z^{-1}])$ can be written as products of so called elementary matrices [2].
2. The main result of this paper also holds in case the filter coefficients are not necessarily real, but belong to another field such as the rationals, the complex numbers, or even a finite field. However, similar results do *not* hold in case 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. It is claimed that their setting is more general than lifting as 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, at least 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. Moreover, most of the advantages of lifting mentioned in the introduction rely fundamentally on the identity on the diagonal and would disappear when allowing a general K.
6. A natural question is how this factorization generalizes to the M-band setting. In algebra 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 [27, Theorem 7.10]. This reduction, also known as the Smith normal form, allows for lifting factorizations in the M-band case. For more details and applications of factorizations in the M-band case, we refer to [29].
7. The construction in this paper gives essentially a factorization of the polyphase matrix representation of the filter bank, by means of the Euclidean algorithm. The basic reason why these factorizations are possible can also be explained (albeit less elegantly) by the following argument. For simplicity, assume that the filter bank is orthogonal, with non-zero filter coefficients $h_0, \ldots, h_{2L+1}$, and $g_0, \ldots, g_{2L+1}$, with $L \geqslant 1$. The orthogonality relations imply both $h_0\, h_{2L} + h_1\, h_{2L+1} = 0$ and $h_0\, g_{2L} + h_1\, g_{2L+1} = 0$. It follows that $g_{2L} = \mu\, h_{2L}$, $g_{2L+1} = \mu\, h_{2L+1} = 0$ for some $\mu \in \mathbf{R}$, so that all the g_n can be written as $g_n = g'_n + \mu\, h_n$, using only $2L$ non-zero g'_n, $n = 0, \ldots, 2L-1$, instead of the $2L+2$ non-zero g_n, $n = 0, \ldots, 2L+1$. The g'_n inherit moreover many orthogonality relations from the g_n, so that they, in turn, can be used to whittle the $2L+2$ non-zero h_n down to $2L$ non-zero h'_n. Iteration of the whole process leads to a decomposition of the filter pair; the inverse procedure is exactly a lifting scheme.

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 the references [29] and [30], Paul Van Dooren for pointing out the connection between the M-band case and the Smith normal form, and Geert Uytterhoeven and Avraham Melkman for pointing out several typos in an earlier draft.

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] M. Bourges-Sévenier. Réalisation d'une bibliothèque C de fonctions ondelettes. Technical Report 2362, IRISA, Campus de Beaulieu, Rennes, France, 1994.

[6] A. Bruckstein and T. Kailath. An inverse scattering framework for several problems in signal processing. *IEEE Acoust. Speech Signal Process. Mag.*, 4(1):6–20, 1987.

[7] 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.

[8] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. Technical report, Department of Mathematics, Princeton University, 1996. `http://cm.bell-labs.com/who/wim/papers/papers.html#integer`.

[9] J. M. Carnicer, W. Dahmen, and J. M. Peña. Local decompositions of refinable spaces. *Journal of Appl. and Comput. Harmonic Analysis*, 3:127–153, 1996.

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

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

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

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

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

[15] 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.

[16] 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.

[17] W. Dahmen, S. Prössdorf, and R. Schneider. Multiscale methods for pseudo-differential equations on smooth manifolds. In [11], pages 385–424. 1994.

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

[19] 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.

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

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

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

[23] 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.

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

[25] 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.

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

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

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

[29] A. A. C. Kalker and I. A. Shah. On ladder structures and linear phase conditions for multidimensional biorthogonal filter banks. Preprint, Philips research laboratories, Eindhoven.

[30] A. A. C. Kalker and I. A. Shah. Ladder structures for multidimensional linear phase perfect reconstruction filter banks and wavelets. In *Visual Communications, Boston*, pages 711–722. Proc. SPIE, 1992.

[31] C. W. Kim, R. Ansari, and A. E. Cetin. A class of linear-phase regular biorthogonal wavelets. In *Proc. IEEE Int. Conference on Acoustics, Speech, Signal Processing*, pages 673–676, 1992.

[32] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution surfaces of arbitrary topological type. Department of Computer Science and Engineering 93-10-05, University of Washington, October 1993. Updated version available as 93-10-05b, January, 1994.

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

[34] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbf{R})$. *Trans. Amer. Math. Soc.*, 315(1):69–87, 1989.

[35] 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.

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

[37] 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.

[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] 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.

[42] D. R. Stinson. *Cryptography: Theory and practice*. CRC Press, Boca Raton, 1995.

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

[44] W. Sweldens. The lifting scheme: A construction of second generation wavelets. Technical Report 1995:6, Department of Mathematics, University of South Carolina, 1995.
`http://cm.bell-labs.com/who/wim/papers/papers.html#lift2`.

[45] W. Sweldens. In place calculations of wavelet transforms. Manuscript, 1996.

[46] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Appl. and Comput. Harmonic Analysis*, 3(2):186–200, 1996.

[47] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.
`http://cm.bell-labs.com/who/wim/papers/papers.html#athome`.

[48] 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.

[49] L. M. G. M. Tolhuizen, I. A. Shah, and A. A. C. Kalker. On constructing regular filter banks from domain bounded polynomials. *IEEE Trans. Signal Process.*, 42(2):451–456, 1994.

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

[51] 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.

[52] 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.

[53] 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.

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

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

[56] 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.

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

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

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