Wavelet Transforms by Nearest Neighbor Lifting

Wei ZHU and M. Victor WICKERHAUSER

Abstract We show that any discrete wavelet transform using finite impulse response filters may be factored into lifting steps that use only nearest-neighbor array elements. We then discuss the advantages and disadvantages of imposing this additional requirement.

1 Introduction

Our goal is to implement discrete wavelet transforms (DWT) efficiently. The recursive algorithms of Daubechies [3] and Mallat [5] offer an $O(n)$ algorithm for n-point time series. The *lifting* implementation of Daubechies and Sweldens [4] offers an alternative which is also $O(n)$ complex but which only requires about half as many arithmetic operations in the most common cases. Additionally, it acts on the input in such a way that requires just $O(1)$ auxilliary memory.

For DWT on an interval, artifacts may arise at the boundary if the input's periodization from that interval is discontinuous. *Symmetric extension* before periodization, thoroughly described in [1], reduces these artifacts and is easy to include within a lifting implementation.

In this paper we consider two further enhancements to the lifting method, with or without symmetric extension and periodization:

- **Nearest neighbor lifting** to reduce the number of distant memory accesses.
- **Lifting sequence choice** allowing some utility to be maximized.

Nearest neighbors in an input array are elements whose indices differ by one. The corresponding memory locations are thus close enough to ensure that both are very

Wei ZHU
Washington University in St. Louis, Missouri, e-mail: zhuwei@math.wustl.edu

M. Victor WICKERHAUSER
Washington University in St. Louis, Missouri, e-mail: victor@math.wustl.edu

likely to reside in the same physical cache and thus are quick to access. Different but equivalent *lifting sequences* all give the same filter transformation but may have different arithmetic complexity or propagation-of-error properties. In this paper, we will show how the existence and construction of these enhancements improves the efficiency of DWT.

2 Review of Discrete Wavelet Transforms

Recall that DWT consists of:

- **Signal:** $u \in \ell^2$, in practice finitely supported or periodic.
- **Analysis filters:** linear maps $\tilde{H}, \tilde{G} : \ell^2 \to \ell^2$, composed of convolution and downsampling.
- **Discrete wavelet transform:** for integer $J > 0$ levels, filter the signal into a collection of wavelet components

$$u \mapsto \{\tilde{H}^J u;\ \tilde{G}\tilde{H}^{J-1}u,\ \tilde{G}\tilde{H}^{J-2}u,\ \ldots,\ \tilde{G}\tilde{H}u,\ \tilde{G}u\}.$$

- **Synthesis filters:** linear maps $H, G : \ell^2 \to \ell^2$, composed of convolution and resampling and related to $\tilde{H}, \tilde{G}$.
- **Wavelet reconstruction:**

$$\begin{aligned} u &= G\tilde{G}u + H\tilde{H}u \\ &= G\tilde{G}u + H\left(G\tilde{G}\tilde{H}u + H\tilde{H}^2 u\right) \\ &= \cdots \\ &= G\tilde{G}u + H\left(G\tilde{G}\tilde{H}u + H\left(\cdots + H\left(G\tilde{G}\tilde{H}^{J-1}u + H\tilde{H}^J u\right)\right)\cdots\right). \end{aligned}$$

An example of DWT to depth $J = 4$ is depicted in Figure 1. In it the analysis filters are determined by sequences $h \leftrightarrow \tilde{H}$ and $g \leftrightarrow \tilde{G}$, while the synthesis filters are adjoints $H = \tilde{H}^* \leftrightarrow h^*$ and $G = \tilde{G}^* \leftrightarrow g^*$. Reconstruction of u is depicted as moving up and adding.

A *filter* $F : \ell^2 \to \ell^2$ is a linear transformation determined by a absolutely summable sequence $f = \{f_n : n \in \mathbf{Z}\}$:

$$Fx_m = \sum_n f_{2m-n} x_n, \qquad m \in \mathbf{Z}.$$

The *adjoint filter* F^* determined by the same sequence f is

$$F^* x_n = \sum_m \overline{f_{2m-n}} x_m, \qquad n \in \mathbf{Z}.$$

Thus $\langle Fx, y\rangle = \langle x, F^* y\rangle$ for all $x, y \in \ell^2$, using the Hermitean inner product in ℓ^2.

The *conjugate filter* $\dot{F}$ of F has sequence $\dot{f} = \{\dot{f}_n : n \in \mathbf{Z}\}$ defined by

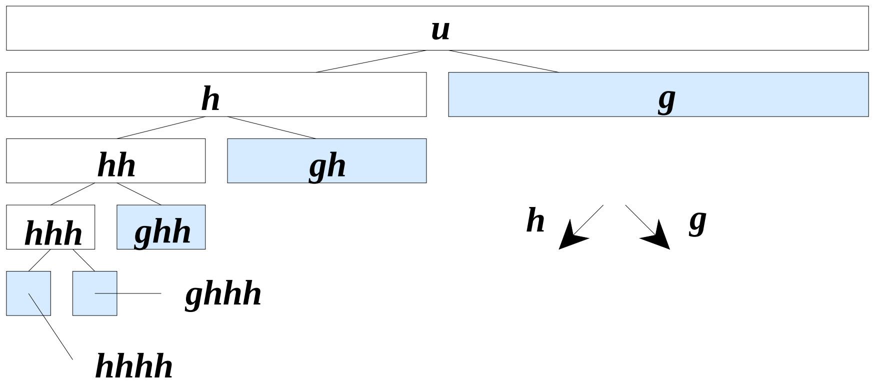

Fig. 1 Four-level discrete wavelet transform with filters h,g

$$\dot{f}_n = (-1)^n f_{1-n} \qquad \Rightarrow \qquad f_n = (-1)^{1-n}\dot{f}_{1-n}, \quad \Rightarrow \ddot{F} = -F$$

Filter F is called *finite*, equivalently *finite impulse response (FIR)*, if its sequence f is finitely-supported. Such filters have a *support interval* $I = [\min S, \max S]$ of finite length $|I|$, where $S = \{n \in \mathbf{Z} : f_n \neq 0\}$. If F is finite then $\dot{F}$ is also finite, with the same support length.

Filter H is called *orthogonal* if it and its conjugate filter $G = \dot{H}$ satisfy the *orthogonality conditions*:

$$HH^* = Id; \quad GG^* = Id; \quad GH^* = HG^* = 0; \quad H^*H + G^*G = Id.$$

Filters H,G form a *perfect reconstruction pair* if they and their conjugates $\tilde{H} = \dot{G}$ and $\tilde{G} = \dot{H}$ satisfy the weaker *biorthogonality conditions*:

$$\tilde{H}H^* = Id; \quad \tilde{G}G^* = Id; \quad \tilde{G}H^* + \tilde{H}G^* = 0; \quad H^*\tilde{H} + G^*\tilde{G} = Id.$$

These may be satisfied for some $G \neq \dot{H}$. However, since $\ddot{H} = -H$ and $\ddot{G} = -G$, any perfect reconstruction pair H,G also satisfies

$$H\tilde{H}^* = Id; \quad G\tilde{G}^* = Id; \quad G\tilde{H}^* + H\tilde{G}^* = 0; \quad \tilde{H}^*H + \tilde{G}^*G = Id.$$

Thus $(\tilde{H},\tilde{G}) = (\dot{G},\dot{H})$ form a perfect reconstruction pair whenever (H,G) are a perfect reconstruction pair.

Call H a *perfect reconstruction filter* if there exists a *complement* filter G such that (H,G) is a perfect reconstruction pair. Any orthogonal filter H is evidently a perfect reconstruction filter: we get a perfect reconstruction pair using $G = \dot{H}$ as its complement.

Equivalent perfect reconstruction conditions may be stated for filter sequences. For example, with $H \leftrightarrow h$ and its conjugate $\dot{H} = G \leftrightarrow g$, the orthogonality conditions become:

$$\sum_k h(k)\overline{h}(k+2n) = \mathbf{1}(n) = \sum_k g(k)\overline{g}(k+2n);$$

$$\sum_k g(k)\overline{h}(k+2n) = \quad 0 \quad = \sum_k h(k)\overline{g}(k+2n);$$

and

$$\sum_k h(2k+m)\overline{h}(2k+n) + \sum_k g(2k+m)\overline{g}(2k+n) = \mathbf{1}(n-m),$$

for all $n,m \in \mathbf{Z}$. Here

$$\mathbf{1}(n) = \begin{cases} 1, & \text{if } n=0, \\ 0, & \text{otherwise.} \end{cases} .$$

It is a straightforward exercise to rewrite the remaining conditions for biorthogonal perfect reconstruction pairs in terms of filter sequences.

3 Review of Lifting

Recall the definition of the Z *transform* of a sequence $x = \{x_n \in \mathbf{C} : n \in \mathbf{Z}\} \in \ell^2$:

$$x(z) = \sum_n x_n z^{-n}, \qquad \text{with} \quad \begin{cases} \textit{even part } x_e(z) \overset{\text{def}}{=} \sum_n x_{2n} z^{-n}; \\ \textit{odd part } \ x_o(z) \overset{\text{def}}{=} \sum_n x_{2n+1} z^{-n}. \end{cases}$$

We recover the Z-transform of x from the even and odd parts x_e, x_o:

$$x(z) = x_e(z^2) + z^{-1} x_o(z^2).$$

Likewise, we get the even and odd parts from the Z-transform:

$$x_e(z^2) = \frac{x(z) + x(-z)}{2}, \qquad x_o(z^2) = \frac{x(z) - x(-z)}{2z^{-1}}.$$

Any filter F determined by a sequence $\{f_n : n \in \mathbf{Z}\}$ is likewise determined by the Z transform $f(z) = \sum_n f_n z^{-n}$. We may denote the even and odd parts by $f_e(z)$ and $f_o(z)$, respectively, and write the actions of F and its adjoint F^* on x as pointwise multiplication of Z transforms:

$$\begin{aligned} Fx(z) &= \sum_m F x_m z^{-m} \ = \ \sum_m \sum_n f_{2m-n} x_n z^{-m} \\ &= \sum_m \sum_n f_{2m-2n} x_{2n} z^{-m} + \sum_m \sum_n f_{2m-2n-1} x_{2n+1} z^{-m} \\ &= \left(\sum_m f_{2m} z^{-m}\right)\left(\sum_n x_{2n} z^{-n}\right) + \left(\sum_m f_{2m-1} z^{-m}\right)\left(\sum_n x_{2n+1} z^{-n}\right) \end{aligned}$$

$$= \left(\sum_m f_{2m} z^{-m}\right)\left(\sum_n x_{2n} z^{-n}\right) + z^{-1}\left(\sum_m f_{2m+1} z^{-m}\right)\left(\sum_n x_{2n+1} z^{-n}\right)$$
$$= f_e(z)x_e(z) + z^{-1} f_o(z)x_o(z);$$

$$F^*x(z) = \sum_n F^*x_n z^{-n} = \sum_n \sum_m \bar{f}_{2m-n} x_m z^{-n}$$
$$= \sum_m \sum_n \bar{f}_n x_m z^{-n-2m} = \left(\sum_n \bar{f}_n z^{-n}\right)\left(\sum_m x_m z^{-2m}\right) = \bar{f}(z)x(z^2).$$

Remark 1. There is also a “correlation and downsampling” definition of filter and adjoint:

$$Fx_m = \sum_n f_{2m+n} x_n, \quad m \in \mathbf{Z}; \qquad F^*x_n = \sum_m \bar{f}_{2m+n} x_m, \quad n \in \mathbf{Z}.$$

Writing this in terms of Z transforms is straightforward and left to the reader. □

There are algebraic relations between the Z transforms of a filter F and its conjugate $\dot{F}$, respectively denoted by f and $\dot{f}$. Namely:

$$\dot{f}(z) = -z^{-1} f(-z^{-1}), \qquad \begin{cases} \dot{f}_e(z) = f_o(z^{-1}), \\ \dot{f}_o(z) = -f_e(z^{-1}). \end{cases}$$

Remark 2. It is possible to generalize to the *M-conjugate* for fixed $M \in \mathbf{Z}$:

$$\dot{f}_n = (-1)^n f_{2M+1-n} \qquad \Rightarrow \qquad f_n = (-1)^{1-n} \dot{f}_{2M+1-n}, \quad \Rightarrow \ddot{F} = -F$$

For the M-conjugate of filter F, compute

$$\dot{f}(z) = -z^{-2M-1} f(-z^{-1}), \qquad \begin{cases} \dot{f}_e(z) = z^{-2M} f_o(z^{-1}), \\ \dot{f}_o(z) = -z^{-2M} f_e(z^{-1}). \end{cases}$$ □

Using the relations just stated, perfect reconstruction conditions for filters may be written in terms of Z transforms. For filters $H, G, \tilde{H} = \dot{G}, \tilde{G} = \dot{H}$, these become:

$$h(z)\tilde{h}(z^{-1}) + g(z)\tilde{g}(z^{-1}) = 1; \qquad h(z)\tilde{h}(-z^{-1}) + g(z)\tilde{g}(-z^{-1}) = 0.$$

In terms of the even and odd parts:

$$h_e(z)\tilde{h}_e(z^{-1}) + g_e(z)\tilde{g}_e(z^{-1}) = 1; \; h_e(z)\tilde{h}_o(z^{-1}) + g_e(z)\tilde{g}_o(z^{-1}) = 0;$$

$$h_o(z)\tilde{h}_o(z^{-1}) + g_o(z)\tilde{g}_o(z^{-1}) = 1; \; h_o(z)\tilde{h}_e(z^{-1}) + g_o(z)\tilde{g}_e(z^{-1}) = 0.$$

We now turn our attention to finite filters. If $p \in \ell^2$ is finitely-supported, then its Z transform $p(z)$ is a *Laurent polynomial*:

$$p(z) = \sum_{n=a}^{b} p_n z^{-n}, \qquad a \leq b, \quad a,b \in \mathbf{Z}.$$

If $p \not\equiv 0$, then $S = \{n : p_n \neq 0\}$ is a finite nonempty set and the *degree* may be defined by $\deg p = \max S - \min S$, a nonnegative integer one less than the support length of the sequence p.

Laurent polynomials form the commutative ring $\mathbf{C}[z,z^{-1}]$ with multiplicative identity 1. Element $p \not\equiv 0$ is called a *unit*, if and only if p has a multiplicative inverse, if and only if p is a *monomial* $p(z) = Kz^n$ for some constants $K \neq 0$ and $n \in \mathbf{Z}$, if and only if $\deg p = 0$. Then $p^{-1}(z) = K^{-1}z^{-n}$.

We may also form matrices over the Laurent Polynomials. The case we will use is the matrix ring $\mathbf{Mat}\left(2\times 2, \mathbf{C}[z,z^{-1}]\right)$, with elements:

$$M(z) = \begin{bmatrix} a(z) & b(z) \\ c(z) & d(z) \end{bmatrix}, \qquad a,b,c,d \in \mathbf{C}[z,z^{-1}].$$

M is invertible if and only if $\det M = a(z)d(z) - b(z)c(z)$ is invertible in $\mathbf{C}[z,z^{-1}]$, namely is a nonzero monomial Kz^n. Then

$$M^{-1}(z) = K^{-1}z^{-n} \begin{bmatrix} d(z) & -b(z) \\ -c(z) & a(z) \end{bmatrix}.$$

Now, any pair H,G of finite filters determines a *polyphase matrix*:

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

Likewise, their conjugates $\tilde{H} = \dot{G}$, $\tilde{G} = \dot{H}$ determine the related polyphase matrix:

$$\tilde{P}(z) = \begin{bmatrix} \tilde{h}_e(z) & \tilde{g}_e(z) \\ \tilde{h}_o(z) & \tilde{g}_o(z) \end{bmatrix} = \begin{bmatrix} \dot{g}_e(z) & \dot{h}_e(z) \\ \dot{g}_o(z) & \dot{h}_o(z) \end{bmatrix}.$$

Both P and $\tilde{P}$ belong to $\mathbf{Mat}(2\times 2, \mathbf{C}[z,z^{-1}])$. A straightforward calculations now shows that the perfect reconstruction condition for (H,G) is equivalent to:

$$P(z)\tilde{P}(z^{-1})^t = Id.$$

Remark 3. In practice, Id may be replaced by any invertible diagonal matrix in $\mathbf{Mat}(2\times 2, \mathbf{C}[z,z^{-1}])$. The two units of $\mathbf{C}[z,z^{-1}]$ appearing on the diagonal will then be monomials Kz^n, or multiples by nonzero K of shifts by n indices. The original sequence x is easily reconstructed from such a shifted and multiplied version. □

Say that a Laurent polynomial $h(z)$ has a *complement* $g = g(z)$ if the polyphase matrix determined by h,g is invertible. It follows immediately that finite filter H is part of a perfect reconstruction pair, if and only if H has a complement, if and only if its Z-transform has a complement. This reduces part of filter design to algebra.

Now, $\mathbf{C}[z,z^{-1}]$ is a Euclidean domain, so the division lemma holds:

Lemma 1. *Suppose* $a,b \in \mathbf{C}[z,z^{-1}]$ *with* $\deg a \geq \deg b \geq 0$. *Then there exists a* quotient q *and a* remainder r *with* $\deg r < \deg b$ *so that*

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

Note that $\deg q = \deg a - \deg b$. □

Write $q = a/b$ and $r = a\%b$, as in the C programming language, but note that neither q nor r is unique.

Lemma 2. *There are at most* $2^{1+\deg a - \deg b}$ *different ways to divide* $a(z)/b(z)$, *among which at most* $2 + \deg a - \deg b$ *quotients are different.*

Proof. First note that division is a generalization of Gaussian elimination. The vector of b's coefficients is shifted, scaled, and added to the vector of a's coefficients to eliminate either the highest or lowest power of z, namely the leftmost or rightmost term. After at most $1 + \deg a - \deg b$ such eliminations, the remainder will have degree less than $\deg b$. Each sequence of eliminations is determined by its sequence of "left" and "right" directives, making at most $2^{1+\deg a - \deg b}$ different ways to find the quotient.

However, left and right eliminations commute as long as $\deg a > \deg b$. Hence, two quotients will be the same if their elimination sequences contain the same number of left and right eliminations, regardless of order. Thus, there can be no more distinct quotients than the number of sequences of length $1 + \deg a - \deg b$ with distinct numbers of "left" directives, which is $2 + \deg a - \deg b$. □

Example 1. Let $a(z) = 2z^{-1} + 4 + z$ and $b(z) = 1 + z$. Then $\deg a = 2$ and $\deg b = 1$, so there are three distinct quotients:

$$\begin{aligned} a(z) &= (2z^{-1}+2)b(z) + (-z) && \text{(left, left)} \\ a(z) &= (3z^{-1}+1)b(z) + (-z^{-1}) && \text{(right, right)} \\ a(z) &= (2z^{-1}+1)b(z) + 1 && \text{(left, right) or (right, left)} \end{aligned}$$

We may say "left division" to mean always eliminating the leftmost term, and "right division" to mean always eliminating the rightmost term. When $\deg a - \deg b$ is even, there will be an even number of terms to eliminate so we may say "symmetric division" to mean an equal number of left and right eliminations.

A Laurent polynomial $p = p(z)$ is said to be *symmetric* if it is unchanged by reversing the order of its coefficients. This is equivalent to the property

$$(\exists M)(\forall z)\ z^M p(z^{-1}) = p(z).$$

For symmetric p not identically zero, the *reflection index* M is unique. Monomials z^k are evidently symmetric with $M = 2k$. We may further distinguish whole or half index symmetry, depending upon whether M is even or odd. The parity of M will be the same as that of $\deg p$.

Lemma 3. *If $a,b \in \mathbf{C}[z,z^{-1}]$ are symmetric Laurent polynomials, then symmetric division results in a symmetric quotient a/b and symmetric remainder $a\%b$.*

Proof. The result holds if $\deg a < \deg b$, since then $a/b = 0$ and $a\%b = a$.

For all other cases, use induction on $n = \deg a - \deg b$.

If $n = 0$, left elimination and right elimination produce identical monomial quotients, which are trivially symmetric. The remainders are evidently identical and symmetric as well.

If $n = 1$, the quotient will have degree 1 with two identical coefficients, hence will be symmetric. The remainder will be the difference between two symmetric polynomials with the same reflection index M, hence will itself be symmetric with that same M.

The induction step follows from the observation that a symmetric (left, right) pair of elimination steps reduces the degree of the dividend by 2 while preserving its symmetry. This reduces n by 2 and contributes to the quotient a symmetric polynomial of the same reflection index M as a. □

We now recall some basic notions useful in Euclidean domains:

- Write $b|a$ (*b divides a*) if $a = qb + 0$ for some q. Thus $b|a \Rightarrow \deg b \leq \deg a$.
- Say that d is a *common divisor* of a and b if $d|a$ and $d|b$.
- Say that a common divisor d is a *greatest common divisor* of a and b if every common divisor c of a and b also divides d.

Lemma 4. *If d_1 and d_2 are greatest common divisors for a and b, then $d_1 = ud_2$ for some unit $u \in \mathbf{C}[z,z^{-1}]$.* □

Theorem 1. *Every pair $a,b \in \mathbf{C}[z,z^{-1}]$, not both zero, has a greatest common divisor that is unique up to multiplication by a unit.* □

Denote this set of greatest common divisors by $\gcd(a,b)$. Say that a,b are *coprime* if $\gcd(a,b)$ is contained in the set of units.

Assume a,b are Laurent polynomials with $\deg a \geq \deg b \geq 0$. Their greatest common divisor may be found by the Euclidean Algorithm for Laurent polynomials. Put $a_0 \overset{\text{def}}{=} a$ and $b_0 \overset{\text{def}}{=} b$, and define a_k, b_k recursively:

$$a_{k+1} = b_k; \qquad b_{k+1} = a_k - q_k b_k, \qquad\qquad k = 0,1,2,\ldots,$$

where q_k is one of the possible quotients a_k/b_k. It thus determines b_{k+1} as the corresponding one of the possible remainders $a_k\%b_k$.

Lemma 5. *Let n be the smallest positive integer for which $b_n = 0$. Then $a_n \in \gcd(a,b)$.* □

By Lemma 4, finding any representative in $\gcd(a,b)$ determines all the others.

Example 2. Consider $a(z) = a_0(z) = 2z^{-1} + 4 + z$ and $b(z) = b_0(z) = 1 + z$. Using symmetric division, the first step is

$$a_1(z) = 1 + z, \qquad b_1(z) = 1, \qquad q_1(z) = 2z^{-1} + 1.$$

The second step is

$$a_2(z) = 1, \qquad b_2(z) = 0, \qquad q_2(z) = 1 + z.$$

Therefore

$$\begin{bmatrix} 2z^{-1} + 4 + z \\ 1 + z \end{bmatrix} = \begin{bmatrix} 2z^{-1} + 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 + z & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

so $\gcd(a,b) = 1$.

Note that a_n is determined only up to a unit, defined by the sequence of quotients $q_0, \ldots, q_{n-1}$.

Theorem 2. *Laurent polynomial h has a complement g if and only if h_e and h_o are coprime.*

Proof. Apply the Euclidean Algorithm to find the polyphase matrix. Write the recursion in matrix form:

$$\begin{bmatrix} a_{k+1}(z) \\ b_{k+1}(z) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & -q_k(z) \end{bmatrix} \begin{bmatrix} a_k(z) \\ b_k(z) \end{bmatrix}, \quad \Rightarrow \begin{bmatrix} a_n(z) \\ 0 \end{bmatrix} = \prod_{k=1}^{n} \begin{bmatrix} 0 & 1 \\ 1 & -q_{n-k}(z) \end{bmatrix} \begin{bmatrix} a(z) \\ b(z) \end{bmatrix}.$$

Inverting the product of matrices gives

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

If n is odd, absorb the unit $(-1)^n$ term into a_n.

Put $a = h_e$ and $b = h_o$ and assume $\gcd(h_e, h_o) = Kz^m$, $K \neq 0$. Define g_e, g_o by

$$P(z) \overset{\text{def}}{=} \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \prod_{k=0}^{n-1} \begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} Kz^m & 0 \\ 0 & K^{-1}z^{-m} \end{bmatrix}.$$

Then $P(z)$ is evidently invertible. Get $\tilde{h}, \tilde{g}$ from $\tilde{P}(z^{-1})^t = P(z)^{-1}$. $\square$

This leads immediately to general implementations of DWT by lifting:

Theorem 3 (Daubechies and Sweldens). *For every perfect reconstruction finite filter pair (H,G) with polyphase matrix P, there exist finitely many Laurent polynomials $s_i(z)$ and $t_i(z)$, $1 \leq i \leq m < \infty$, and a non-zero constant K such that*

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

Remark 4. The matrix factors correspond to the following operations on sequences:

- *Prediction*: Unit upper triangular factor; $u_e \leftarrow u_e + Su_o$.
- *Updating*: Unit lower triangular factor, $u_o \leftarrow u_o + Tu_e$.
- *Scaling*: Last diagonal matrix, $u_e \leftarrow Ku_e$, $u_o \leftarrow K^{-1}u_o$.

Since u_e, u_e may be stored as disjoint arrays, this transform can be performed in place, without extra memory for temporary results. □

Proof. Observe first:

$$\begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & q_k(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_k(z) & 1 \end{bmatrix}.$$

The *flip* matrices $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ cancel if Predict and Update steps alternate.

Second, note that a leftover flip matrix may be factored into lifting steps in a number of ways:

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

Third, note that diagonal shift matrices may be factored into lifting steps:

$$\begin{aligned} \begin{bmatrix} z & 0 \\ 0 & z^{-1} \end{bmatrix} &= \begin{bmatrix} 1 & -z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ z^{-1} & 1 \end{bmatrix} \begin{bmatrix} 1 & 1-z \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1-z & 1 \end{bmatrix} \begin{bmatrix} 1 & z^{-1} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -z & 1 \end{bmatrix} \end{aligned}$$

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

Other factorizations exist, but at most five lifting steps are needed per shift. Thus, $\begin{bmatrix} z^m & 0 \\ 0 & z^{-m} \end{bmatrix}$ or $\begin{bmatrix} z^{-m} & 0 \\ 0 & z^m \end{bmatrix}$ factor into at most $5m$ lifting steps. Afterwards, only the constant diagonal matrix $\begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}$ remains. □

4 Nearest Neighbor Factorization

Assume that a smooth sampled signal $u \in \ell^2$ is finitely supported in the index interval $[0, N-1]$. Big endpoint values $|u(0)|$ or $|u(N-1)|$ may result in misleading large DWT coefficients. Similarly, a big difference $|u(N-1) - u(0)|$ may result in large periodized DWT coefficients. These undesirable effects are mitigated through the use of *symmetric extension*, as described in [1]. It requires symmetric H, G, defining

$$u(n) = \begin{cases} u(-n), & \text{if } -N < n < 0; \\ u(n) = u(2N-1-n), & \text{if } N < n < 2N, \end{cases}$$

and then treating u as $2N-2$-periodic. Several other extensions are possible, depending on the symmetry type of H, G.

It is easy to implement symmetric extension for certain special implementations. The lifting factorization

$$P(z) = \prod_{k=1}^{n} \begin{bmatrix} 1 & s_k(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_k(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix}.$$

uses only *nearest neighbors* if it satisfies the following conditions:

$$\begin{aligned} s_k(z) &= \alpha_k + \beta_k z^{-1}, \\ t_k(z) &= \gamma_k z + \delta_k, \end{aligned}$$

with $\alpha_k, \beta_k, \gamma_k, \delta_k \in \mathbf{C}$. Nearest neighbor action on sequences has the explicit forms:

- *Nearest neighbor predict:* $u_{2k} \leftarrow u_{2k} + \alpha_k u_{2k-1} + \beta_k u_{2k+1}$.
- *Nearest neighbor update:* $u_{2k+1} \leftarrow u_{2k+1} + \gamma_k u_{2k} + \delta_k u_{2k+2}$.

For nearest neighbor factorizations, symmetric extension becomes:

- *Symmetric extension nearest neighbor predict step:*

$$u_{2k} \leftarrow u_{2k} + \begin{cases} \alpha(u_{2k-1} + u_{2k+1}), & \text{if } 2k \neq 0; \\ 2\alpha\, u_{2k+1}, & \text{if } 2k = 0. \end{cases}$$

- *Symmetric extension nearest neighbor update step:*

$$u_{2k+1} \leftarrow u_{2k+1} + \begin{cases} \gamma(u_{2k} + u_{2k+2}), & \text{if } 2k+1 \neq N-1; \\ 2\gamma u_{N-2}, & \text{if } 2k+1 = N-1. \end{cases}$$

Hence the endpoints get almost the same treatment as the interior points.

5 All Lifting can be Nearest Neighbor Lifting

Unfortunately, not every perfect reconstruction filters factors into nearest neighbor lifting steps directly, even allowing for any choice of quotients in Euclid's algorithm. For example, let

$$h(z) = \frac{1}{\sqrt{2}}(1+z^{-9}) \qquad g(z) = \frac{1}{\sqrt{2}}(-z^{8}+z^{-1}).$$

This is similar to the Haar orthogonal filter pair. Then

$$h_e(z) = \frac{1}{\sqrt{2}}; \quad h_o(z) = \frac{1}{\sqrt{2}}z^{-4}; \quad g_e(z) = \frac{1}{\sqrt{2}}z^{4}; \quad g_o(z) = \frac{1}{\sqrt{2}}.$$

There are no division steps in Euclid's algorithm, so the (empty) sequence of quotients is unique. The ordinary lifting factorization gives:

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

This is not a nearest neighbor factorization because the off-diagonal nonconstant terms have powers other than z and z^{-1}. However, in common with nearest neighbor lifting steps, the off-diagonal terms $s(z), t(z)$ have $\deg s \le 1$ and $\deg t \le 1$, and further factorization is possible (see Lemma 7 below) to obtain nearest-neighbor steps.

We may obtain quotients of constrained degree through a modification of the division lemma:

Lemma 6 (Partial division). *Suppose $a, b \in \mathbf{C}[z, z^{-1}]$ with $\deg a \ge \deg b \ge 0$. Then there exists a* partial quotient *q and a* partial remainder *r with $\deg q \le 1$ and $\deg r < \deg a$ so that*

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

Proof. We limit ourselves to eliminating just one or two terms from a with a partial quotient of the form $q(z) = z^m(\gamma + \delta z)$. Then $\deg q \le 1$, but not both $\gamma = 0$ and $\delta = 0$, so $\deg r = \deg(a - qb) < \deg a$. $\square$

It is clear that any common divisor of $a(z)$ and $b(z)$ also divides the partial remainder $r(z) = a(z) - q(z)b(z)$, so we obtain lifting factors of degree one or less with Euclid's algorithm expanded to use partial division:

Theorem 4. *Assume a and b are two coprime nonzero Laurent polynomials. Then there exist Laurent polynomials $q_1, \ldots, q_n$ with $\deg q_k \le 1$ for all $k = 1, \ldots, n$, such that*

$$\begin{bmatrix} a(z) \\ b(z) \end{bmatrix} = \prod_{k=1}^{n} \begin{bmatrix} q_k(z) & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix},$$

where $n \le 2(\deg a + \deg b + 1)$. $\square$

We may now slightly strengthen Theorem 3:

Corollary 1. *For every perfect reconstruction finite filter pair (H,G) with polyphase matrix P, there is a lifting factorization*

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

where the Laurent polynomials $s_i(z)$ and $t_i(z)$, $1 \le i \le m < \infty$, each have degree one or less, and K is a non-zero constant. □

As shown earlier, the degree condition is not enough to guarantee that the factorization gives a nearest neighbor filter transform. To get from degree one or less to nearest neighbor polynomials requires additional factorization:

Lemma 7.

$$\begin{bmatrix} 1 & z^{2m}(\alpha z^{-1}+\beta) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} z^m & 0 \\ 0 & z^{-m} \end{bmatrix} \begin{bmatrix} 1 & \alpha z^{-1}+\beta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z^{-m} & 0 \\ 0 & z^m \end{bmatrix};$$

$$\begin{bmatrix} 1 & 0 \\ z^{2m}(\gamma+\delta z) & 1 \end{bmatrix} = \begin{bmatrix} z^{-m} & 0 \\ 0 & z^m \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \gamma+\delta z & 1 \end{bmatrix} \begin{bmatrix} z^m & 0 \\ 0 & z^{-m} \end{bmatrix},$$

where m is any integer and $\alpha,\beta,\gamma,\delta$ are constants. □

Factoring the z^m shifts into at most $5m$ nearest neighbor lifting steps each yields:

Corollary 2. *Any degree-one predict or update matrix factors further into a finite number of nearest neighbor lifting steps.* □

Remark 5. The conditions $\deg s_k \le 1$ and $\deg t_k \le 1$ may also be obtained from an ordinary lifting factorization by further decomposition:

$$\begin{bmatrix} 1 & s_1(z)+s_2(z) \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & s_1(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & s_2(z) \\ 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 \\ t_1(z)+t_2(z) & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ t_1(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_2(z) & 1 \end{bmatrix}.$$

However, this may create multiple successive predict factors and multiple successive update factors and ultimately requires more matrices. □

6 Backward Error Analysis for Lifting Factorizations

We now turn consider how various lifting factorizations affect the conditioning of DWT. In terms of the polyphase matrix, this may be computed as follows:

Lemma 8. *If $P(z)$ is the polyphase matrix of a perfect reconstruction filter pair, then*

$$\operatorname{cond}(P) \overset{\text{def}}{=} \frac{\sup\left\{\sqrt{\lambda_{\max}(P(z)^*P(z))} : |z|=1\right\}}{\inf\left\{\sqrt{\lambda_{\min}(P(z)^*P(z))} : |z|=1\right\}}.$$

Furthermore, if $P = P_1 \cdots P_n$, then $\operatorname{cond}(P) \le \operatorname{cond}(P_1)\cdots\operatorname{cond}(P_n)$. □

For a proof of this fact and extensive discussion of polyphase matrices, see [6].

We may use the special form of lifting step matrices to estimate the condition number of the factorization of P in another, simpler way. For definiteness, consider an "update" step in floating-point arithmetic with absolute truncation error ε. Its polyphase matrix G may be represented within the computer by something that differs by as much as δG, where

$$G(z) = \begin{bmatrix} 1 & 0 \\ t(z) & 1 \end{bmatrix} \qquad \Rightarrow \qquad \delta G(z) = \begin{bmatrix} \varepsilon & 0 \\ \delta t(z) & \varepsilon \end{bmatrix},$$

and Laurent polynomial $t(z) = \sum_k t_k z^{-k}$ has error $\delta t(z) = \sum_k \delta t_k z^{-k}$. Each polynomial coefficient δt_k is computed from the filter h by elimination and therefore satisfies

$$|\delta t_k| \le (1+\deg h)\varepsilon + O(\varepsilon^2).$$

For such G, define

$$|G|(z) \overset{\text{def}}{=} \begin{bmatrix} 1 & 0 \\ |t|(z) & 1 \end{bmatrix}, \qquad \text{where} \quad |t|(z) \overset{\text{def}}{=} \sum_k |t_k| z^{-k}.$$

Then the maximum matrix infinity norm of $G(z)$ may be computed as follows:

$$\|G\|_\infty \overset{\text{def}}{=} \sup_{|z|=1} \|G(z)\|_\infty \le \sup_{|z|=1} \|\,|G|(z)\,\|_\infty = 1 + \sup_{|z|=1} |t|(z) = 1 + \sum_k |t_k|.$$

The same estimate applies to "predict" steps as well.

Now assume that $P(z)$ is a polyphase matrix with lifting factorization

$$P(z) = \prod_i \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & K^{-1} \end{bmatrix} \overset{\text{def}}{=} \prod_j G_j(z),$$

where each G_j is a lifting step. Taking truncation errors into account, the computed results using floating-point arithmetic therefore satisfy

$$P(z) + \delta P(z) = \prod_j (G_j(z) + \delta G_j(z)).$$

By expanding the product and using the submultiplicativity of the matrix infinity norm, we get

$$\|\delta P\|_\infty \leq \varepsilon\,(1+\deg h)\sum_j ||G_j||_\infty + O(\varepsilon^2),$$

indicating that to obtain the smallest condition number, we should use a factorization with small lifting coefficients and not too many low-degree lifting steps.

Assuming the worst case, equality, we can estimate the condition number for three implementations of the filter bank:

1. the original polyphase matrix P,
2. the usual (shortest) lifting factorization of P, and
3. the nearest neighbor lifting factorization of P.

The results are displayed in Table 1, for a number of symmetric and nonsymmetric-orthogonal filters.

Filter	Cond of P(z)	Cond of Lifting	Cond of N-N
9-7	1.32	205	205
D4	1	77	77
D6	1	76	76
Cubic B-spline	4	56	56
CDF-1-1	1	8.59	8.59
CDF-1-3	1.28	8.72	3100
CDF-1-5	1.42	6.25	1200
CDF-2-2	2	8.59	8.59
CDF-2-4	2	99	1900
CDF-3-1	4	643	643
CDF-3-3	4	723	3200
CDF-4-2	8	111	111
CDF-4-4	8	113	2800

Table 1 Condition number bounds for nearest neighbor factorization versus ordinary lifting versus the polyphase matrix.

7 Applications and Examples

Because of nonuniqueness in the quotients in Euclid's algorithm for matrices over Laurent polynomials, we may choose a lifting factorization optimized for a minimal number of nearest neighbors.

In some cases, the original lifting steps yield a nearest neighbor algorithm. The filter indexing may need to be adjusted to eliminate or at least minimize the number of z^m shift matrices. In addition, the sequence of quotients $\{q_k(z) : k = 0, 1, \ldots, n-1\}$ may be chosen to minimize the condition number bound.

Example 3. To show how re-indexing may result in a nearest-neighbor factorization, consider Daubechies' `D4` filters, defined as follows:

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

with

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

Then the polyphase matrix is

$$P(z) = \tilde{P}(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} = \begin{bmatrix} h_2 z^{-1} + h_0 & -h_1 - h_3 z \\ h_3 z^{-1} + h_1 & h_0 + h_2 z \end{bmatrix}.$$

It has the following two factorizations, the first obtained by left division in Euclid's algorithm, the second by right division:

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

The forward transform corresponding to the first (left-division) factorization is not nearest neighbor:

$$\begin{aligned} x_{2m+1} &\leftarrow x_{2m+1} - \sqrt{3} x_{2m}; \\ x_{2m} &\leftarrow x_{2m} + \frac{\sqrt{3}}{4} x^{(1)}_{2m+1} + \frac{\sqrt{3}-2}{4} x^{(1)}_{2m+3}; \\ x_{2m+1} &\leftarrow x_{2m} + x_{2m-2}; \\ x_{2m} &\leftarrow \frac{\sqrt{3}+1}{\sqrt{2}} x_{2m}; \\ x_{2m+1} &\leftarrow \frac{\sqrt{3}-1}{\sqrt{2}} x_{2m+1}. \end{aligned}$$

The inverse transform for the left division factorization into lifting steps is similarly not nearest neighbor, as may be seen from its polyphase matrix:

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

It is left as an exercise to derive the predict and update steps from these matrices.

The second (right-division) factorization can be made nearest neighbor by factoring the leftover diagonal shift matrices into lifting steps. However, if the coefficients of h are first shifted so that $h(z) = \sum_{-2}^{1} h_{k+2} z^{-k}$, then right division in Euclid's algorithm yields a nearest neighbor transform directly:

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

It is again left as an exercise to find the corresponding predict and update steps.

Example 4. Not all orthogonal filters will give nearest neighbor factorization directly after a simple index shift. Consider the orthogonal filters defined earlier:

$$h(z) = \frac{\sqrt{2}}{2}(1+z^{-9}) \qquad g(z) = \frac{\sqrt{2}}{2}(-z^{8}+z^{-1}),$$

We cannot get a nearest neighbor factorization simply by using an index shift. For this filter, it is necessary to use Lemma 7 and pay the price of additional lifting steps.

Example 5. Not all filters offer a choice of lifting factorizations. The biorthogonal perfect reconstruction filters `CDF-2-4h` and `CDF-2g` have the following coefficients:

$$\begin{aligned} h(z) &= \sqrt{2}\left(\frac{3}{128}z^{-4} - \frac{3}{64}z^{-3} - \frac{1}{8}z^{-2} + \frac{19}{64}z^{-1} + \right. \\ &\qquad \left. + \frac{45}{64} + \frac{19}{64}z - \frac{1}{8}z^{2} - \frac{3}{64}z^{3} + \frac{3}{128}z^{4}\right) \\ g(z) &= \sqrt{2}\left(\frac{1}{4}z^{-2} - \frac{1}{2}z^{-1} + \frac{1}{4}\right). \end{aligned}$$

CDF biorthogonal filters [2] are symmetric, so there is a unique lifting factorization using the Euclidean algorithm. But since the degrees of $h(z)$ and $g(z)$ are very different, in most cases it does not yield a nearest neighbor factorization directly.

$$\begin{aligned} P(z) &= \begin{bmatrix} -\frac{1}{2}z^{-1} - \frac{1}{2} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{3}{64}z^{-1} + \frac{19}{64} + \frac{19}{64}z - \frac{3}{64}z^{2} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{2}z^{-1} - \frac{1}{2} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{3}{64}z^{-1} + \frac{19}{64} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \\ &\quad \begin{bmatrix} \frac{19}{64}z - \frac{3}{64}z^{2} & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} \end{bmatrix} \\ &= \begin{bmatrix} 1 & -\frac{1}{2}z^{-1} - \frac{1}{2} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -\frac{3}{64}z^{-1} + \frac{19}{64} \\ 0 & 1 \end{bmatrix} \\ &\quad \begin{bmatrix} z & 0 \\ 0 & z^{-1} \end{bmatrix} \begin{bmatrix} 1 & \frac{19}{64}z^{-1} - \frac{3}{64} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} z^{-1} & 0 \\ 0 & z \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}. \end{aligned}$$

An application of Lemma 7, Corollary 2, and Theorem 3 may be used to convert this into a nearest neighbor factorization.

Example 6. If the filter h has a perfect reconstruction complement, then its coefficients may be re-indexed and the proper quotients chosen in the division algorithm so that $\gcd(h_e, h_o)$ is a constant. However, this is not guaranteed to produce nearest neighbor lifting.

Consider Daubechies' orthogonal `D6` filter; it may have its indices shifted so that $h(z) = \sum_{k=-2}^{3} h_k z^{-k}$. The filter thus indexed, together with its complement $g = \dot{h}$, has the following polyphase matrix:

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

Then the lifting factorization by symmetric division yields a constant $\gcd(h_e, h_o) \approx 1.918$, entirely through nearest-neighbor predict and update steps:

$$\begin{aligned} P(z) = &\begin{bmatrix} 1 & 0 \\ -0.412 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1.565z^{-1} + 0.352 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.028 + 0.492z & 1 \end{bmatrix} \\ &\begin{bmatrix} 1 & -0.390 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.918 & 0 \\ 0 & 0.521 \end{bmatrix}. \end{aligned}$$

Alternatively, with the indexing $h(z) = \sum_{k=0}^{5} h_k z^{-k}$, we get a different polyphase matrix:

$$\begin{aligned} h_e(z) &= h_4 z^{-2} + h_2 z^{-1} + h_0 & g_e(z) &= -h_1 z^{-1} - h_3 z^{1} - h_5 z^{2} \\ h_o(z) &= h_5 z^{-2} + h_3 z^{-1} + h_1 & g_o(z) &= h_0 + h_2 z + h_4 z^2. \end{aligned}$$

Using the same division as for `D4` now gives nonconstant $\gcd(h_e, h_o) = 1.918z^{-1}$, but with nearest-neighbor predict and update steps:

$$\begin{aligned} P(z) = &\begin{bmatrix} 1 & 0 \\ -0.412 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1.565z^{-1} + 0.352 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.028 + 0.492z & 1 \end{bmatrix} \\ &\begin{bmatrix} 1 & -0.390 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.918z^{-1} & 0 \\ 0 & 0.521z \end{bmatrix} \end{aligned}$$

However, choosing right division so that the gcd comes out constant gives

$$\begin{aligned} P(z) = &\begin{bmatrix} 1 & 0 \\ -0.412 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1.565z^{-1} + 0.355 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0.001645z^{-1} - 0.028 & 1 \end{bmatrix} \\ &\begin{bmatrix} 1 & 607.65z - 116.5z^2 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} -33.172 & 0 \\ 0 & 0.03 \end{bmatrix}, \end{aligned}$$

which is evidently not a nearest neighbor factorization.

References

1. Christopher Brislawn. Classification of nonexpansive symmetric extension transforms for multirate filter banks. *Applied and Computational Harmonic Analysis*, 3(4):337–357, October 1996.
2. Albert Cohen, Ingrid Daubechies, and Jean-Christophe Feauveau. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45:485–500, 1992.
3. Ingrid Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41:909–996, 1988.
4. Ingrid Daubechies and Wim Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
5. Stéphane G. Mallat. Multiresolution approximation and wavelet orthonormal bases of $L^2(\mathbf{R})$. *Transactions of the AMS*, 315:69–87, 1989.
6. Gilbert Strang and Truong Nguyen. *Wavelets and Filter Banks*. Wellesley–Cambridge Press, Wellesley, Massachusetts, 1996.