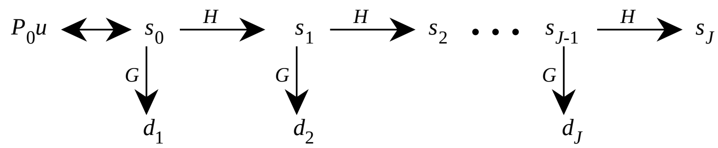

Figure 5.5: The pyramid algorithm for the discrete wavelet transform (DWT).

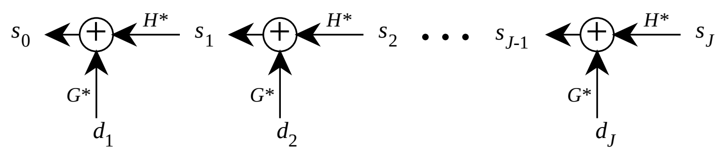

Figure 5.6: The pyramid algorithm for the inverse discrete wavelet transform (iDWT).

$$
\begin{aligned}
d_{j+1}(n) &= \sum_k g(k) s_j(2n+k) &= \sum_k g(k-2n) s_j(k); \qquad (5.59)\\
s_{j-1}(n) &= \sum_k \overline{h(n-2k)}\, s_j(k) + \sum_k \overline{g(n-2k)}\, d_j(k). \qquad (5.60)
\end{aligned}
$$

Equations 5.58 and 5.59 provide a recursive algorithm, the *Mallat pyramid algorithm* depicted in Figure 5.5, for computing expansions in any of the subspaces of the MRA. The finitely-supported sequence $s_0 = \{s_0(k) : k \in \mathbf{Z}\}$, defined by $s_0(k) = \langle \phi(t-k), u(t) \rangle$, completely determines the approximation $P_0 u \in V_0$. The *wavelet expansion of* $P_0 u$ *to level* $J > 0$ consists of the sequences $(d_1, d_2, \ldots, d_J; s_J)$. Sequence s_J determines the crude approximation $P_J u \in V_J$, while d_J, d_{J-1}, and so on contain extra information needed to refine it successively into the better approximation $P_0 u \in V_0$.

Reconstruction from the wavelet expansion is done by a similar pyramid algorithm, depicted in Figure 5.6. The arrows are reversed using adjoints, and the results are summed according to Equation 5.60.

In both pyramid algorithms, sequences $s_1, s_2, \ldots, s_{J-1}$ are computed along the way, even though they are not part of the discrete wavelet expansion or the reconstructed signal. Enough temporary storage to hold another copy of the signal may therefore be needed, depending upon how the arrow operations are implemented.

Filter transforms

A finitely-supported sequence $f = \{f(k) : k \in \mathbf{Z}\}$ defines a *filter transform*, acting on arbitrary sequences $u = \{u(k) : k \in \mathbf{Z}\}$ by either of the two equivalent formulas, related by the substitution $k \leftarrow k' + 2n$:

$$Fu(n) = \sum_k f(k-2n)u(k) = \sum_{k'} f(k')u(k'+2n), \qquad n \in \mathbf{Z}. \tag{5.61}$$

This F is a linear transformation on every vector space of complex-valued sequences, including the inner product space ℓ^2 of square-summable sequences with inner product $\langle u, v\rangle = \sum_k \overline{u(k)}v(k)$. In that space, F has an adjoint F^* that satisfies $\langle Fu, v\rangle = \langle u, F^*v\rangle$ for all u and v. But then,

$$\begin{aligned} \sum_n \overline{Fu(n)}v(n) &= \sum_n \left(\sum_k \overline{f(k-2n)}\bar{u}(k)\right) v(n) \\ &= \sum_k \left(\bar{u}(k) \sum_n \overline{f(k-2n)}v(n)\right) \quad \stackrel{\text{def}}{=} \quad \sum_k \bar{u}(k)F^*v(k), \end{aligned}$$

which defines two equivalent formulas for the adjoint filter transform:

$$\begin{aligned} F^*v(k) &= \sum_n \overline{f(k-2n)}\, v(n) \qquad (5.62) \\ &= \begin{cases} \sum_{n'} \overline{f(2n')}\, v(\frac{k}{2} - n'), & \text{if } k \in \mathbf{Z} \text{ is even,} \\ \sum_{n''} \overline{f(2n''+1)}\, v(\frac{k-1}{2} - n''), & \text{if } k \in \mathbf{Z} \text{ is odd.} \end{cases} \end{aligned}$$

These are related by the substitutions $n \leftarrow \frac{k}{2} - n'$ if k is even, and $n \leftarrow \frac{k-1}{2} - n''$ if k is odd.

Composing F and its adjoint gives

$$F^*Fu(j) = \sum_n \overline{f(2n-j)}Fu(n) = \sum_{n,k} \overline{f(2n-j)}f(2n-k)u(k). \tag{5.63}$$

Similarly,

$$FF^*u(m) = \sum_k f(2m-k)F^*u(k) = \sum_{k,n} f(2m-k)\overline{f(2n-k)}\, u(n). \tag{5.64}$$

Because of the dilation by 2, F typically shrinks the support of sequences, while F^* enlarges it:

Lemma 5.11 *Suppose that the sequence f is supported on the index interval $[a,b]$: $f = \{\dots, 0, f(a), f(a+1), \dots, f(b-1), f(b), 0, \dots\}$, since $f(n) = 0$ if $n < a$ or $n > b$. For any sequence u supported on $[x,y]$,*

- *Fu is supported on $[\lceil \frac{x-b}{2} \rceil, \lfloor \frac{y-a}{2} \rfloor]$;*
- *F^*u is supported on $[2x+a, 2y+b]$.*

Proof: Taking into account the support, the second version of the filter transform formula reduces to

$$Fu(n) = \sum_{k'=a}^{b} f(k')u(k'+2n). \tag{5.65}$$

Notice that the summand will be zero if $b + 2n < x$ or $a + 2n > y$. Only output values at indices $n \in [x', y']$ need to be computed, where $x' = \lceil (x-b)/2 \rceil$ and $y' = \lfloor (y-a)/2 \rfloor$.

On the other hand, the first version of the adjoint filter transform formula reduces to

$$F^*v(k) = \sum_{n=x}^{y} \overline{f(k-2n)}\, v(n). \tag{5.66}$$

The summand will be zero unless $a \le k - 2n \le b$ and $x \le n \le y$. Hence, output values need only be computed at indices $k \in [x'', y'']$, where $x'' = 2x + a$ and $y'' = 2y + b$. □

Lemma 5.11 illuminates two kinds of *spreading in support* that occur with filter transforms. Firstly, if F is one filter from an orthogonal CQF pair, then F^*F is an orthogonal projection on ℓ^2, but the support of F^*Fu may be greater than that of u. If u is finitely-supported on the index interval $[x, y]$, and f is supported on $[a, b]$, then F^*Fu will be finitely supported in the index interval $\left[2\left\lceil \frac{x-b}{2} \right\rceil + a, 2\left\lfloor \frac{y-a}{2} \right\rfloor + b\right]$. This contains $[x - (b-a-1), y + (b-a-1)]$, which in turn contains $[x, y]$ and is strictly larger if and only if $b - a > 1$. The only orthogonal CQF with $b - a \le 1$ is the Haar pair, with $a = 0, b = 1$ in the conventional indexing giving $b - a = 1$.

Secondly, a CQF pair H, G of filter transforms can produce more total output values than there are input values. Suppose the supports are $[a, b]$ for H and $[c, d]$ for G. For a finitely supported sequence $u = \{u(x), \dots, u(y)\}$ of length $N = 1 + y - x$, the high-pass and low-pass parts of the signal will be supported on the intervals $\left[\left\lceil \frac{x-b}{2} \right\rceil, \left\lfloor \frac{y-a}{2} \right\rfloor\right]$ and $\left[\left\lceil \frac{x-d}{2} \right\rceil, \left\lfloor \frac{y-c}{2} \right\rfloor\right]$, respectively, with total length

$$\left(1 + \left\lfloor \frac{y-a}{2} \right\rfloor - \left\lceil \frac{x-b}{2} \right\rceil\right) + \left(1 + \left\lfloor \frac{y-c}{2} \right\rfloor - \left\lceil \frac{x-d}{2} \right\rceil\right). \tag{5.67}$$

The total support length will be greater than $1 + y - x$ if and only if $b - a > 1$ or $d - c > 1$. The Haar CQF pair has $b - a = 1$ and $d - c = 1$, and is the only filter pair that does not cause spreading of the support.

Successive applications of H and G give the (nonperiodic) discrete orthogonal wavelet transform on finitely-supported infinite sequences. There are only finitely many finitely-supported sequences $d_1, d_2, \dots, d_J$, and s_J to compute, and each output coefficient costs only a finite number of operations since h, g are both finite sequences, say of length L. Since L must be even by Lemma 5.7, we can write $L = 2M$ for an integer M. If h is conventionally indexed, so that $h(k)$ is nonzero only for $0 \le k < L$, then we may choose[3] to define $g(k) = (-1)^k h(2M - 1 - k)$ to insure that $g(k)$ is also nonzero only for $0 \le k < L$.

[3]Work Exercise 12 to see why!

With these indexing conventions, if $s_j(k)$ is supported in $x \leq k \leq y$, then $d_{j+1}(n)$ and $s_{j+1}(n)$ may be nonzero for $\lceil(1+x-L)/2\rceil \leq n \leq \lfloor y/2 \rfloor$. Hence, the output sequences are of varying lengths:

Mallat's Discrete Wavelet Transform

```
dwt( u[], x, y, J, h[], g[], L ):
[0]  If J=0 then for n = x to y, print u[n]
[1]  Else do [2] to [9]
[2]     Let x1 = ceiling((1+x-L)/2), let y1 = floor(y/2)
[3]     For n=x1 to y1, do [4] to [8]
[4]        Let s[n] = 0, let d[n] = 0
[5]        For k=0 to L-1, do [6] to [7]
[6]           Accumulate s[n] += h[k]*u[k+2*n]
[7]           Accumulate d[n] += g[k]*u[k+2*n]
[8]        Print d[n]
[9]     Compute dwt( s[], x1, y1, J-1, h[], g[], L )
```

Of course, values `d[n]` do not have to be printed as soon as they are computed, they may be stored in an array. For fixed L and J, this array will require $O(N)$ total elements, and will cost $O(N)$ operations to fill.

Periodic filter transforms

If f_{2q} is a $2q$-periodic sequence with even period, then it can be used to define a *periodic filter transform* F_{2q} from $2q$-periodic to q-periodic sequences, and a *periodic adjoint* F_{2q}^* from q-periodic to $2q$-periodic sequences. These are, respectively, the transformations

$$F_{2q}u(n) = \sum_{k=0}^{2q-1} f_{2q}(k-2n)u(k) = \sum_{k'=0}^{2q-1} f_{2q}(k')u(k'+2n), \qquad 0 \leq i < q; \tag{5.68}$$

and

$$F_{2q}^* v(k) = \sum_{n=0}^{q-1} \overline{f_{2q}(k-2n)}\, v(n) \tag{5.69}$$

$$= \begin{cases} \displaystyle\sum_{n'=0}^{q-1} \overline{f_{2q}(2n')}\, v(\frac{k}{2} - n'), & \text{if } k \in [0, 2q-2] \text{ is even,} \\ \displaystyle\sum_{n''=0}^{q-1} \overline{f_{2q}(2n''+1)}\, v(\frac{k-1}{2} - n''), & \text{if } k \in [1, 2q-1] \text{ is odd.} \end{cases} \tag{5.70}$$

We have performed the same substitutions as in Equations 5.61 and 5.62. Except for the index ranges, the formulas are the same.

Periodization commutes with filter transforms: we get the same periodic sequence whether we first filter an infinite sequence and then periodizes the result, or first periodize both the sequence and the filter and then perform a periodic filter transform. To be precise:

Lemma 5.12 *$(Fu)_q = F_{2q}u_{2q}$ and $(F^*v)_{2q} = F^*_{2q}v_q$.*

Proof: Note that

$$\begin{aligned}
(Fu)_q(n) &= \sum_{j=-\infty}^{\infty} Fu(n+qj) \;=\; \sum_{j=-\infty}^{\infty}\sum_{k=-\infty}^{\infty} f(k-2[n+qj])u(k) \\
&= \sum_{k=-\infty}^{\infty}\left(\sum_{j=-\infty}^{\infty} f(k-2n-2qj)\right)u(k) \;=\; \sum_{k=-\infty}^{\infty} f_{2q}(k-2n)u(k) \\
&= \sum_{k_1=0}^{2q-1}\sum_{k_2=-\infty}^{\infty} f_{2q}(k_1+2qk_2-2n)u(k_1+2qk_2) \\
&= \sum_{k_1=0}^{2q-1} f_{2q}(k_1-2n)\sum_{k_2=-\infty}^{\infty} u(k_1+2qk_2) \\
&= \sum_{k_1=0}^{2q-1} f_{2q}(k_1-2n)u_{2q}(k_1).
\end{aligned}$$

Also,

$$\begin{aligned}
(F^*v)_{2q}(k) &= \sum_{j=-\infty}^{\infty} F^*v(k+2qj) \;=\; \sum_{j=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \overline{f([k+2qj]-2n)}\,v(n) \\
&= \sum_{n=-\infty}^{\infty}\left(\sum_{j=-\infty}^{\infty} \overline{f(k+2qj-2n)}\right)v(n) \;=\; \sum_{n=-\infty}^{\infty} \overline{f_{2q}(k-2n)}\,v(n) \\
&= \sum_{n_1=0}^{q-1}\sum_{n_2=-\infty}^{\infty} \overline{f_{2q}(k-2n_1-2qn_2)}\,v(n_1+qn_2) \\
&= \sum_{n_1=0}^{q-1} \overline{f_{2q}(k-2n_1)}\sum_{n_2=-\infty}^{\infty} v(n_1+qn_2) \\
&= \sum_{n_1=0}^{q-1} \overline{f_{2q}(k-2n_1)}\,v_q(n_1).
\end{aligned}$$

□

Thus, any redundancy introduced by periodizing an even-length input signal can be removed by ignoring all but one period of the output.

A $2q$-periodized pair of orthogonal CQFs h, g retain their orthogonal CQF properties. A $2q$-periodic input sequence u may be completely described by $2q$ coefficients, and H_{2q} and G_{2q} each produce q-periodic outputs that are completely