A detailed analysis of the Daubechies and Sweldens algorithm of wavelet filter factorization into lifting steps

PMS

October 13, 2011

Abstract

The factoring algorithm proposed by Daubechies and Sweldens, [**?**] is analyzed and decomposed to the step-by-step form enabling a direct implementation in *Mathematica* or *Maple* script.

Factoring into lifting steps. Below we analyze the algorithm in a detailed manner. Assume we have the initial filters (or, equivalently, their Z-transforms in a form of *Laurent polynomials*)

$$h(z) \text{ and } g(z) = z^{-1} h\left(-z^{-1}\right)$$

- The crucial operation consists in decomposition of the polyphase representation of $h(z)$, *viz.* $[h_e \; h_o]^T$ into lifting steps[1].
- Using a matrix notation the *Euclidean algorithm* can be written as

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

  where $\{q_i\}$ is a sequence of quotients produced by the algorithm for polynomials $h_e(z)$ and $h_o(z)$, and where K is their gcd().[2]

[1] The index e stands for even elements ("evens") of $h(z)$ and o for odd ones ("odds")

$$h_e\left(z^2\right) = \frac{h(z) + h(-z)}{2} \text{ and } h_o\left(z^2\right) = \frac{h(z) - h(-z)}{2z^{-1}}$$

[2] The algorithm is given recursively

$$q_i(z) = a_i(z) \operatorname{div} b_i(z)$$

where

$$a_{i+1}(z) = b_i(z) \text{ and } b_{i+1}(z) = a_i(z) \bmod b_i(z)$$

with $a_1(z) = h_e(z)$ and $b_1(z) = h_o(z)$, for $i = 1, \ldots, n \, (= |h_o(z)| + 1)$.

- The tricky identities are applied now (*i.e.* column and row swapping with the help of *permutation matrix*)

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

to yield the factorization which starts to mimic a lifting and dual lifting steps

$$\begin{bmatrix} h_e(z) \\ h_o(z) \end{bmatrix} = \prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} K \\ 0 \end{bmatrix}$$

- In general, n does not need to be an even number for a given filter $h(z)$. If it is odd, we can multiply $h(z)$ by z and $g(z)$ by z^{-1} (which, corresponds to shifting the filters h and g by 2).

- If we now replace $\begin{bmatrix} K & 0 \end{bmatrix}^T$ by $\begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$ we obtain the matrix

$$P^0(z) = \begin{bmatrix} h_e(z) & g_e^0(z) \\ h_o(z) & g_o^0(z) \end{bmatrix}$$

in which $g_e^0(z)$ and $g_o^0(z)$ are polyphase representations of a filter $g^0(z)$ (which is not known and does not need to be known, actually). Since $P^0(z)$ has (by its construction) the determinant 1 (note that for odd n the determinant would have been -1), then $g_o(z)$ is complementary to $h(z)$ and one step is sufficient to lift $g^0(z)$ to $g(z)$

$$g(z) = g^0 + h(z)\, s\left(z^2\right)$$

where $s(z)$ is some Laurent polynomial (not known either).

- Recall finally that, in a matrix form, this lifting step is given as

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

- All these facts eventually lead to the factorization

$$P(z) = \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & h_o(z) \end{bmatrix} = \left(\prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \right) \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix} \begin{bmatrix} 1 & s(z) \\ 0 & 1 \end{bmatrix}$$

which, in the article by Daubechies and Sweldens, [**?**], has the equivalent, but far less explicit, form

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

where $m = n/2 + 1$, $s_i(z) = q_{2i-1}(z)$, $t_i(z) = q_{2i}(z)$, for $i = 1, \ldots, n/2$, and where $s_m(z) = K^2 s(z)$ and $t_m(z) = 0$. To see this equivalence we rewrite it to the following form

$$P(z) = \left(\prod_{i=1}^{n/2} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \right) \begin{bmatrix} 1 & K^2 s(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}$$

and recall the trivial identity

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

- Note that the factorization (1) is not yet complete since the polynomial $s(z)$ remains unknown. To find it we merely need to perform the (right and left) multiplications by the inverses of the product

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

and by the rightmost diagonal matrix

$$\begin{bmatrix} 1 & K^2 \cdot s(z) \\ 0 & 1 \end{bmatrix} = \left[\prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \right]^{-1} \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix}$$

- Recalling that these inverses are of particularly simple forms

$$\begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \text{ and } \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ -q_{2i}(z) & 1 \end{bmatrix}$$

and applying another well known relation

$$(A_1 \cdot \cdots \cdot A_n)^{-1} = A_n^{-1} \cdot \cdots \cdot A_1^{-1}$$

we obtain the inverse of the product

$$\left[\prod_{i=1}^{n/2} \begin{bmatrix} 1 & q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ q_{2i}(z) & 1 \end{bmatrix} \right]^{-1} = \prod_{i=n/2}^{1} \begin{bmatrix} 1 & 0 \\ -q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -q_{2i-1}(z) \\ 0 & 1 \end{bmatrix}$$

- Finally, $s(z)$ can be computed from the formula

$$\begin{bmatrix} 1 & K^2 \cdot s(z) \\ 0 & 1 \end{bmatrix} = \left(\prod_{i=n/2}^{1} \begin{bmatrix} 1 & 0 \\ -q_{2i}(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -q_{2i-1}(z) \\ 0 & 1 \end{bmatrix} \right) \begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix} \begin{bmatrix} \frac{1}{K} & 0 \\ 0 & K \end{bmatrix}$$

after performing all multiplications on right-hand side and dividing the corresponding matrix entry by a scalar K^2. Substituting $s_m(z)$ by the just computed $s(z)$ in (1) accomplishes the factorization. ■

Remark 1 *One can use again the form*

$$\prod_{i=1}^{n}\left[\begin{array}{cc} q_i\left(z\right) & 1 \\ 1 & 0 \end{array}\right]$$

instead of the more elaborated

$$\prod_{i=1}^{n/2}\left[\begin{array}{cc} 1 & q_{2i-1}\left(z\right) \\ 0 & 1 \end{array}\right]\left[\begin{array}{cc} 1 & 0 \\ q_{2i}\left(z\right) & 1 \end{array}\right]$$

when computing $s\left(z\right)$*. This alternative formula is therefore simpler a bit*

$$\left[\begin{array}{cc} 1 & K^2\cdot s\left(z\right) \\ 0 & 1 \end{array}\right]=\left(\prod_{i=n}^{1}\left[\begin{array}{cc} 0 & 1 \\ 1 & -q_i\left(z\right) \end{array}\right]\right)\left[\begin{array}{cc} h_e\left(z\right) & g_e\left(z\right) \\ h_o\left(z\right) & g_o\left(z\right) \end{array}\right]\left[\begin{array}{cc} \frac{1}{K} & 0 \\ 0 & K \end{array}\right]$$