Gradient-based optimization of mother wavelets

Nicola Neretti   Nathan Intrator

Abstract— **We present a general framework for the design of a mother wavelet best adapted to a specific signal or to a class of signals. The filter's coefficients are obtained via optimization of a smooth objective function. We develop an unconstrained gradient-based optimization algorithm for a discrete wavelet transform. The algorithm is extended to the joint optimization of the mother wavelet and of the wavelet packets basis.**

Keywords— **Wavelets, Wavelet Pakets, Filter Banks, Polyphase Matrix, Lattice Decomposition, Optimization.**

I. Introduction

THE general problem we are trying to address is to find an invertible linear transform $\mathcal{L}$ that minimizes an objective function ϕ for a specific signal or for a class of signals. The linear transform depends in general on a set of parameters, $\vec{p} \in \mathcal{P} \subset \mathbf{R^q}$. Then, for a specific signal $\vec{x} = [x_0, \ldots, x_M]^T$, the minimum is obtained for:

$$\vec{p}_o(\vec{x}) = \mathrm{argmin}_{\vec{p}\in\mathcal{P}}\, \phi\left(\mathcal{L}_{\vec{p}}\left[\vec{x}\right]\right)$$
$$\mathcal{L}_{\vec{p}}\left[\vec{x}\right] : \mathbf{R^{M+1}} \mapsto \mathbf{R^k}\ , \mathbf{k} \leq \mathbf{M+1} \tag{1}$$
$$\mathcal{L}_{\vec{p}}\left[a\vec{x}+b\vec{y}\right] = a\mathcal{L}_{\vec{p}}\left[\vec{x}\right] + b\mathcal{L}_{\vec{p}}\left[\vec{y}\right]$$

For a class of signals, the objective function is redefined to compute some statistics from the collection of data. We concentrate on the class of linear transforms known as discrete wavelet transforms.

In is paper we describe a novel optimization for the purpose of wavelet filter decomposition or of more general basis function decompositions based on wavelet packets. The optimization is based on the lattice decomposition of filter banks and leads to a fast unconstrained algorithm.

II. Formulation of the discrete wavelet transform

In the discrete wavelet transform, a low-pass and a high-pass filter are applied to the signal, and the output is down-sampled by two. The high-pass coefficients are retained, while the process is repeated on the low-pass coefficients, until the length of the residual signal's coefficients equals that of the filter. In order for this transform to be invertible, the filters have to satisfy some constrains. In particular, orthonormality is required to obtain an orthonormal basis. These constrains can be expressed in different forms. Exact details of the various forms can be found in chapter 5 of [1]. Here, in particular, we use two formulations: 1) the time-domain method and 2) the lattice method. The first approach expresses the constrains directly on the filters' coefficients. This leads to a constrained optimization algorithm. The second approach is based on the lattice structure method. With this method it is possible to reparametrize the coefficients so that the constrains are automatically satisfied. This leads to an unconstrained optimization algorithm.

N. Neretti and N. intrator are with the Institute for Brain and Neural Systems, Brown University, Providence, RI 02912, US. E-mail: Nathan_Intrator@brown.edu

Without loss of generality, we concentrate on wavelet transforms with periodic boundaries. What follows can be extended to wavelets with different boundary conditions (e.g. Appendix C) .

A. The time-domain method

Consider an example which demonstrates how a discrete wavelet transform is computed and how some constrains on the filter's coefficients arise from the orthogonality condition. In what follows, the length of the signals will be a power of 2. The case of a filter of length 4 acting on a signal of length 8 is described in detail, as well as generalization to filters of even length. We use a notation from [1], where signals and filters are indexed from 0 to N, so that the total length is $N+1$.

Let $\vec{x} = [x_0, \ldots, x_7]^T$ be the original signal, $c_0, \ldots c_3$ and $d_0, \ldots d_3$ the low-pass and high-pass filters' coefficients respectively. After applying the two filters to $\vec{x}$, the result is decimated by two. These two operations, application of the filters and decimation, can be done more efficiently in a single step. In matrix form they are equivalent to the following:

$$[\sigma_0, \sigma_1, \sigma_2, \sigma_3, \delta_0, \delta_1, \delta_2, \delta_3]^T = C_1\vec{x}, \tag{2}$$

where

$$C_1 = \begin{bmatrix} c_3 & c_2 & c_1 & c_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_3 & c_2 & c_1 & c_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & c_2 & c_1 & c_0 \\ c_1 & c_0 & 0 & 0 & 0 & 0 & c_3 & c_2 \\ d_3 & d_2 & d_1 & d_0 & 0 & 0 & 0 & 0 \\ 0 & 0 & d_3 & d_2 & d_1 & d_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & d_3 & d_2 & d_1 & d_0 \\ d_1 & d_0 & 0 & 0 & 0 & 0 & d_3 & d_2 \end{bmatrix}. \tag{3}$$

Then the same process is repeated just on the s's, the low-pass coefficients:

$$\begin{aligned}&[\Sigma_0, \Sigma_1, \Delta_0, \Delta_1, \delta_0, \delta_1, \delta_2, \delta_3]^T \\ &= C_2\left[\sigma_0, \sigma_1, \sigma_2, \sigma_3, \delta_0, \delta_1, \delta_2, \delta_3\right]^T,\end{aligned} \tag{4}$$

where

$$C_2 = \begin{bmatrix} c_3 & c_2 & c_1 & c_0 & 0 & 0 & 0 & 0 \\ c_2 & c_0 & c_3 & c_1 & 0 & 0 & 0 & 0 \\ d_3 & d_2 & d_1 & d_0 & 0 & 0 & 0 & 0 \\ d_1 & d_0 & d_3 & d_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \quad (5)$$

Combining the two steps together, the transformed signal $\vec{y}$ is given by:

$$\vec{y} = C\vec{x} = C_2 C_1 \vec{x}, \quad (6)$$

It is easy to see that when C_1 is orthonormal, C_2 and hence C are orthonormal. The condition for orthonormality are:

$$\begin{array}{l} c_0^2 + c_1^2 + c_2^2 + c_3^2 = 1, \\ c_0 c_2 + c_1 c_3 = 0, \end{array} \quad (7)$$

$$\begin{array}{l} d_0^2 + d_1^2 + d_2^2 + d_3^2 = 1, \\ d_0 d_2 + d_1 d_3 = 0, \end{array} \quad (8)$$

$$\begin{array}{l} c_0 d_0 + c_1 d_1 + c_2 d_2 + c_3 d_3 = 0, \\ c_0 d_2 + c_1 d_3 = 0, \\ c_0 d_0 + c_3 d_1 = 0, \end{array} \quad (9)$$

If the c's satisfy (7), it is possible to solve all the equations in (8) and (9) with the following choice:

$$d_k = (-1)^k c_{3-k} \; , k = 0, \ldots, 3. \quad (10)$$

In general, a filter of even length $N+1$, acts on a signal of length M+1, where M+1 is a power of 2. The transform is given by:

$$\vec{y} = C\vec{x}, \quad (11)$$

where the matrix C is the product of Q orthonormal matrices:

$$C = C_Q C_{Q-1} \cdots C_2 C_1 \; , \quad Q \leq Q_{\max} \; , \quad (12)$$

where $Q_{\max}$, the maximum number of decomposition levels allowed, depends on both the length of the signal and that of the filter (MATLAB notation):

$$Q_{\max} = \text{floor}\left(\log_2 \frac{M+1}{N+1}\right) . \quad (13)$$

The orthonormality conditions can then be expressed in a compact form:

- Conditions on the c's:

$$\sum_{n=2k}^{N} c_n c_{n-2k} = \delta(k) \; , k = 0, \ldots, (N-1)/2 \quad (14)$$

- Conditions on the d's:

$$\sum_{n=2k}^{N} d_n d_{n-2k} = \delta(k) \; , k = 0, \ldots, (N-1)/2 \quad (15)$$

- Relations between the c's and the d's:

$$\begin{array}{lcll} \sum_{n=2k}^{N} c_n d_{n-2k} & = & \delta(k) & , k = 0, \ldots, (N-1)/2 \\ \sum_{n=2k}^{N} c_{n-2k} d_n & = & \delta(k) & , k = 0, \ldots, (N-1)/2 \end{array} \quad (16)$$

The high-pass coefficients can be computed from the low-pass ones just like in (10):

$$d_k = (-1)^k c_{N-k} \; , k = 0, \ldots, N. \quad (17)$$

With this choice, if (14) is satisfied, then (15) and (16) are.

Thus, summarizing all the constrains on the c's, we have:

$$\begin{cases} \textbf{normalization} & \\ c_0^2 + c_1^2 + \cdots + c_N^2 & = \; 1 \\ & \\ \textbf{orthogonality} & \\ c_0 c_2 + c_1 c_3 + \cdots + c_{N-2} c_N & = \; 0 \\ c_0 c_5 + c_1 c_6 + \cdots + c_{N-4} c_N & = \; 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ c_0 c_{N-1} + c_1 c_N & = \; 0 \end{cases} \quad (18)$$

B. The lattice method: reparametrization of the filter's coefficients

It is possible to reparametrize the coefficients $c_0, \ldots, c_N$ so that the constrains in (18) are automatically satisfied. Starting from the simple case of 4 coefficients, the constrains in (7) imply that:

$$[c_0 + c_2]^2 + [c_1 + c_3]^2 = 1, \quad (19)$$

which is automatically satisfied setting

$$\begin{cases} c_0 + c_2 & = & \cos(\theta_1 + \theta_2) \\ c_1 + c_3 & = & \sin(\theta_1 + \theta_2) \end{cases} \quad (20)$$

Using the trigonometric addition formulas

$$\begin{array}{lcl} \cos(\alpha+\beta) & = & \cos\alpha\cos\beta - \sin\alpha\sin\beta \\ \sin(\alpha+\beta) & = & \sin\alpha\cos\beta + \cos\alpha\sin\beta \end{array} \quad (21)$$

we get

$$\begin{cases} c_0 & = & \cos\theta_1 \cos\theta_2 \\ c_1 & = & \cos\theta_1 \sin\theta_2 \\ c_2 & = & -\sin\theta_1 \sin\theta_2 \\ c_3 & = & \sin\theta_1 \cos\theta_2 \end{cases} \quad (22)$$

It would be natural to extend this procedure to the general case. In fact, the orthonormality conditions in (18) imply that:

$$\left(\sum_{n \text{ even}} c_n\right)^2 + \left(\sum_{n \text{ odd}} c_n\right)^2 = 1, \quad (23)$$

so that we can set

$$\begin{aligned}\sum_{n=0}^{(N-1)/2} c_{2n} &= \cos\left(\sum_{n=0}^{(N-1)/2}\theta_n\right)\\ \sum_{n=0}^{(N-1)/2} c_{2n+1} &= \sin\left(\sum_{n=0}^{(N-1)/2}\theta_n\right)\end{aligned} \tag{24}$$

The right-hand side in both the above equations contains $(N+1)/2$ terms, but the expansions of the left-hand sides, obtained using the generalized trigonometric addition formula, contain $2^{(N-1)/2}$ terms. Distributing the trigonometric monomials to the various filter's coefficients is not straightforward. However, this problem can be solved in a systematic and very elegant way using the lattice factorization from the theory of filter banks. In fact, the polyphase matrix of any two channel orthogonal filter bank can be factorize as [1] [2] [3]:

$$H_p^{(K)} = \rho(\theta_1)\Lambda(z)\rho(\theta_2)\cdots\Lambda(z)\rho(\theta_K) \tag{25}$$

where $\rho(\theta)\in O(2)$ and

$$\Lambda(z) = \begin{bmatrix} 1 & 0 \\ 0 & z^{-1}\end{bmatrix}. \tag{26}$$

The factorization of the polyphase matrix leads naturally to a factorization of the wavelet transform in the time domain. This factorization process is illustrated below for a filter having six coefficients operating on a signal of length eight. Although illustrated for this example, the process can be extended to filters of even length and signals of arbitrary length (see Appendices A and C for details).

Equation (25) can be rewritten as:

$$H_p^{(K)} = H_p^{(K-1)}\Lambda(z)\rho(\theta_K) \tag{27}$$

which links the coefficients of a filter of length $2K$ to those of a filter of length $2(K-1)$, and allows an iterative procedure to build a filter of arbitrary lenght. In particular, the polyphase matrices for filters of length six, four and two are given by:

$$\begin{aligned} &H_p^{(3)} = \\ &\begin{bmatrix} c_0^{(3)} + z^{-1}c_2^{(3)} + z^{-2}c_4^{(3)} & c_1^{(3)} + z^{-1}c_3^{(3)} + z^{-2}c_5^{(3)} \\ d_0^{(3)} + z^{-1}d_2^{(3)} + z^{-2}d_4^{(3)} & d_1^{(3)} + z^{-1}d_3^{(3)} + z^{-2}d_5^{(3)}\end{bmatrix} \\ &H_p^{(2)} = \begin{bmatrix} c_0^{(2)} + z^{-1}c_2^{(2)} & c_1^{(2)} + z^{-1}c_3^{(2)} \\ d_0^{(2)} + z^{-1}d_2^{(2)} & d_1^{(2)} + z^{-1}d_3^{(2)}\end{bmatrix} \\ &H_p^{(1)} = \begin{bmatrix} c_0^{(1)} & c_1^{(1)} \\ d_0^{(1)} & d_1^{(1)}\end{bmatrix}\end{aligned} \tag{28}$$

where the othonormality is assured for $H_p^{(1)}$ and all polyphase matrices derived from it when

$$H_p^{(1)} = \rho(\theta_1) = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1\end{bmatrix} \tag{29}$$

Thus from equation (27), $H_p^{(2)}$ is given by:

$$H_p^{(2)} = H_p^{(1)}\Lambda(z)\rho(\theta_2) \tag{30}$$

which can be written as:

$$\begin{aligned} &\begin{bmatrix} c_0^{(2)} + z^{-1}c_2^{(2)} & c_1^{(2)} + z^{-1}c_3^{(2)} \\ d_0^{(2)} + z^{-1}d_2^{(2)} & d_1^{(2)} + z^{-1}d_3^{(2)}\end{bmatrix} = \\ &\begin{bmatrix} \cos\theta_1 & \sin\theta_1 \\ -\sin\theta_1 & \cos(\theta_1\end{bmatrix}\begin{bmatrix} 1 & 0 \\ 0 & z^{-1}\end{bmatrix}\times \\ &\times\begin{bmatrix} \cos\theta_2 & \sin\theta_2 \\ -\sin\theta_2 & \cos\theta_2\end{bmatrix}\end{aligned} \tag{31}$$

Multiplying the right-hand side and solving for the value of the various coefficients on the left-hand side can be determined using the reparameterized coefficients directly by equating powers of z. This leads to:

$$\begin{aligned} c_0^{(2)} &= \cos\theta_1\cos\theta_2 = c_0^{(1)}\cos\theta_2 \\ c_1^{(2)} &= \cos\theta_1\sin\theta_2 = c_0^{(1)}\sin\theta_2 \\ c_2^{(2)} &= -\sin\theta_1\sin\theta_2 = -c_1^{(1)}\sin\theta_2 \\ c_3^{(2)} &= \sin\theta_1\cos\theta_2 = c_1^{(1)}\cos\theta_2\end{aligned} \tag{32}$$

and similarly for the d's:

$$\begin{aligned} d_0^{(2)} &= -\sin\theta_1\cos\theta_2 = d_0^{(1)}\cos\theta_2 \\ d_1^{(2)} &= -\sin\theta_1\sin\theta_2 = d_0^{(1)}\sin\theta_2 \\ d_2^{(2)} &= -\cos\theta_1\sin\theta_2 = -d_1^{(1)}\sin\theta_2 \\ d_3^{(2)} &= \cos\theta_1\cos\theta_2 = d_1^{(1)}\cos\theta_2\end{aligned} \tag{33}$$

These relationships can be rewritten in matrix form as:

$$\begin{bmatrix} c_3^{(2)} \\ c_2^{(2)} \\ c_1^{(2)} \\ c_0^{(2)}\end{bmatrix} = \begin{bmatrix} \cos\theta_2 & 0 \\ -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 \\ 0 & \cos\theta_2\end{bmatrix}\begin{bmatrix} c_1^{(1)} \\ c_0^{(1)}\end{bmatrix} \tag{34}$$

$$\begin{bmatrix} d_3^{(2)} \\ d_2^{(2)} \\ d_1^{(2)} \\ d_0^{(2)}\end{bmatrix} = \begin{bmatrix} \cos\theta_2 & 0 \\ -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 \\ 0 & \cos\theta_2\end{bmatrix}\begin{bmatrix} d_1^{(1)} \\ d_0^{(1)}\end{bmatrix} \tag{35}$$

As noted above in equation (27), the next polyphase matrix $H_p^{(3)}$ can be found as a function of $H_p^{(2)}$. Multiplying the matrices and equating like powers of z as above yields:

$$\begin{aligned} &\left[c_5^{(3)} c_4^{(3)} c_3^{(3)} c_2^{(3)} c_1^{(3)} c_0^{(3)}\right]^T = \\ &\begin{bmatrix} \cos\theta_3 & 0 & 0 & 0 \\ -\sin\theta_3 & 0 & 0 & 0 \\ 0 & \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 & 0 \\ 0 & 0 & 0 & \sin\theta_3 \\ 0 & 0 & 0 & \cos\theta_3\end{bmatrix}\begin{bmatrix} c_3^{(2)} \\ c_2^{(3)} \\ c_1^{(2)} \\ c_0^{(2)}\end{bmatrix}\end{aligned} \tag{36}$$

Substituting equation (34) in (36) yields

$$\left[c_5^{(3)} c_4^{(3)} c_3^{(3)} c_2^{(3)} c_1^{(3)} c_0^{(3)}\right]^T =$$

$$\begin{bmatrix} \cos\theta_3 & 0 & 0 & 0 \\ -\sin\theta_3 & 0 & 0 & 0 \\ 0 & \sin\theta_3 & \cos\theta_3 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 & 0 \\ 0 & 0 & 0 & \sin\theta_3 \\ 0 & 0 & 0 & \cos\theta_3 \end{bmatrix} \times \tag{37}$$

$$\begin{bmatrix} \cos\theta_2 & 0 \\ -\sin\theta_2 & 0 \\ 0 & \sin\theta_2 \\ 0 & \cos\theta_2 \end{bmatrix} \begin{bmatrix} \sin\theta_1 \\ \cos\theta_1 \end{bmatrix}$$

A similar expression hold for the ds, so that combining the two together we have

$$\begin{bmatrix} c_5^{(3)} & c_4^{(3)} & c_3^{(3)} & c_2^{(3)} & c_1^{(3)} & c_0^{(3)} \\ d_5^{(3)} & d_4^{(3)} & d_3^{(3)} & d_2^{(3)} & d_1^{(3)} & d_0^{(3)} \end{bmatrix} =$$

$$\begin{bmatrix} \sin\theta_1 & \cos\theta_1 \\ \cos\theta_1 & -\sin\theta_1 \end{bmatrix} \times$$

$$\begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ 0 & 0 & \sin\theta_2 & \cos\theta_2 \end{bmatrix} \times$$

$$\begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sin\theta_3 & \cos\theta_3 & 0 & 0 \\ 0 & 0 & \cos\theta_3 & -\sin\theta_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix} \tag{38}$$

Thus, each coefficient can be found as a function of the reparameterized angle coefficients. Performing the matrix multiplication in equation (35) defines each coefficient as a linear combination of two or more trigonometric functions. The matrix C_1 that performs the first step in the wavelet cascade of filters is given by

$$E \begin{bmatrix} c_5^{(3)} & c_4^{(3)} & c_3^{(3)} & c_2^{(3)} & c_1^{(3)} & c_0^{(3)} & 0 & 0 \\ d_5^{(3)} & d_4^{(3)} & d_3^{(3)} & d_2^{(3)} & d_1^{(3)} & d_0^{(3)} & 0 & 0 \\ 0 & 0 & c_5^{(3)} & c_4^{(3)} & c_3^{(3)} & c_2^{(3)} & c_1^{(3)} & c_0^{(3)} \\ 0 & 0 & d_5^{(3)} & d_4^{(3)} & d_3^{(3)} & d_2^{(3)} & d_1^{(3)} & d_0^{(3)} \\ c_1^{(3)} & c_0^{(3)} & 0 & 0 & c_5^{(3)} & c_4^{(3)} & c_3^{(3)} & c_2^{(3)} \\ d_1^{(3)} & d_0^{(3)} & 0 & 0 & d_5^{(3)} & d_4^{(3)} & d_3^{(3)} & d_2^{(3)} \\ c_3^{(3)} & c_2^{(3)} & c_1^{(3)} & c_0^{(3)} & 0 & 0 & c_5^{(3)} & c_4^{(3)} \\ d_3^{(3)} & d_2^{(3)} & d_1^{(3)} & d_0^{(3)} & 0 & 0 & d_5^{(3)} & d_4^{(3)} \end{bmatrix} \tag{39}$$

where E separates the high-pass coefficients from the low-pass ones. Plugging (38) into equation (39), C_1 can be decomposed into the product of six matrices:

$$C_1 = ER(\theta_1)\, SR(\theta_2)\, SR(\theta_3) \tag{40}$$

where

$$R(\theta_j) = \begin{bmatrix} s_j & k_j & 0 & 0 & 0 & 0 & 0 & 0 \\ k_j & -s_j & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & s_j & k_j & 0 & 0 & 0 & 0 \\ 0 & 0 & k_j & -s_j & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_j & k_j & 0 & 0 \\ 0 & 0 & 0 & 0 & k_j & -s_j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & s_j & k_j \\ 0 & 0 & 0 & 0 & 0 & 0 & k_j & -s_j \end{bmatrix} \tag{41}$$

$$S = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{42}$$

and $s_j = \sin\theta_j$, $k_j = \cos\theta_j$. Notice that $R(\theta_j)$ are rotation matrices and S is an up-shift matrix. In general, for a filter of length $2K$ we have (see Appendix A for more details):

$$C_1 = ER(\theta_1)SR(\theta_2)S \cdots SR(\theta_K) \tag{43}$$

III. Extension to wavelet packets

A *library of wavelet packet bases* [4] [5] is defined to be a collection of orthonormal bases composed of functions of the form $W_f(2^{-s}x - p)$, where $s, p \in \mathbf{Z}, \mathbf{f} \in \mathbf{N}$. Therefore, each basis is determined by a subset of the indices: a scaling parameter s, a localization parameter p, and an oscillation parameter f. These are natural parameters, since the function $W_f(2^{-s}x - p)$ is roughly centered at $2^s k$, has support of size $\approx 2^s$, and oscillates $\approx f$ times.

The library can be obtained as follows. A low-pass filter H and a high-pass filter G are applied iteratively to the signal (see Figure 1, A). After the first step (second row from the top), we obtain two different representations of the signal: an averaged version and a detailed version. Each of the two representations has a number of coefficients equal to half the original one. The combination of the two represents a projection on an orthonormal basis. In the second iteration, the filters are applied to the two blocks to obtain a set of four blocks. Again, these new four blocks represent a projection on a new basis (Figure 1, C). The process is iterated until we reach blocks containing a single value.

It can be demonstrated that every collection of non-overlapping blocks that spans the entire length of the signal is an orthonormal basis. Figure 1 shows examples of bases that can be extracted. The library contains as a special case the standard wavelet basis (B), as well as a basis whose elements have the same scale (C); other bases can be extracted (D). The total number of basis functions for a signal of length n is $n\log(n)$, and they can be combined to create 2^n bases.

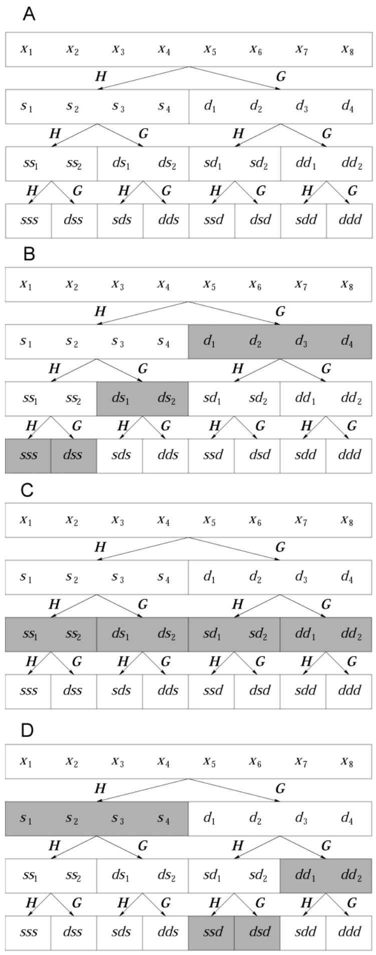

Fig. 1. Wavelet packet decomposition tree obtained applying recursively a high-pass filter G and a low-pass filter H (A); discrete wavelet basis (B); fixed-level wavelet packet basis (C); another wavelet packet basis (D).

The library of wavelet packet bases can be searched to find the best basis according to some criterion, for example to minimize an objective function. Coifman et al. [4] showed that, if the objective function is additive a divide and conquer formulation can be applied to the search, so that a library of 2^n bases can be searched in $n\log(n)$.

The wavelet transform formulation in terms of a product of matrices can be easily extended to compute any basis in the wavelet packets table. Expression (12) still holds: $C_l, 1 \leq l \leq Q$ is a block matrix; each block corresponds to either the identity, if that component of the signal is not decomposed any further, or a wavelet matrix. Equation (43) provides a factorization for a single step in the wavelet transform. Putting all the pieces together we obtain a factorization for a generic wavelet packets basis:

$$C = \underbrace{E_Q R_Q(\theta_1) S_Q R_Q(\theta_2) S_Q \cdots S_Q R_Q(\theta_K)}_{C_Q} \cdots \\ \cdots \underbrace{E_1 R_1(\theta_1) S_1 R_1(\theta_2) S_1 \cdots S_1 R_1(\theta_K)}_{C_1} \tag{44}$$

IV. Optimization

The formulation given in (44) is very natural for a gradient optimization. It thus serves our goal to find the optimal parameterization of the transform given a signal or a class of signals. This is obtained by minimizing an appropriate objective function. A general procedure for its minimization based on the computation of the derivatives with respect to the parameters is given below. Due to the linearity of the transform, these derivatives can be expressed in a simple form. Consider for example the optimization problem in (1), with $\vec{y} = \mathcal{L}_{\vec{p}}[\vec{x}]$. The gradient of the objective function with respect to the parameters of the transform is:

$$\nabla_p \phi = \mathbf{J_p^T} \nabla_{\mathbf{y}} \phi \ , \tag{45}$$

where $\mathbf{J_p}$ is the Jacobian of $\mathcal{L}$. For $\mathcal{L}_{\vec{p}}[\vec{x}] = C\vec{x}$

$$\mathbf{J_p} = \left[\left(\partial_{\mathbf{p_1}} \mathbf{C}\right) \vec{\mathbf{x}}, \left(\partial_{\mathbf{p_2}} \mathbf{C}\right) \vec{\mathbf{x}}, \ldots, \left(\partial_{\mathbf{p_q}} \mathbf{C}\right) \vec{\mathbf{x}} \right] \ . \tag{46}$$

An explicit form for the derivatives with respect to the lattice parameters can be obtained through equation (44):

$$\begin{array}{cl} & \partial_j C(\theta_1, \ldots, \theta_K) = \\ = & (\partial_j C_Q)\, C_{Q-1} \ldots\ldots\ldots\ldots\ldots\ldots C_1 \quad + \\ + & C_Q\, (\partial_j C_{Q-1}) \ldots\ldots\ldots\ldots\ldots\ldots C_1 \quad + \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ + & C_Q C_{Q-1} \ldots\ldots\ldots\ldots\ldots\ldots (\partial_j C_1) \quad = \\ = & E R_Q(\theta_1) S_Q \cdots \partial_j R_Q(\theta_j) \ldots\ldots S_1 R_1(\theta_K) \quad + \\ + & E R_Q(\theta_1) S_Q \cdots\cdots \partial_j R_{Q-1}(\theta_j) \ldots S_1 R_1(\theta_K) \quad + \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ + & E R_Q(\theta_1) S_Q \cdots\cdots\cdots \partial_j R_1(\theta_j) \ldots S_1 R_1(\theta_K) \quad = \\ = & E R_Q(\theta_1) S_Q \cdots R_Q(\theta_j + \frac{\pi}{2}) \ldots\ldots S_1 R_1(\theta_K) \quad + \\ + & E R_Q(\theta_1) S_Q \cdots\cdots R_{Q-1}(\theta_j + \frac{\pi}{2}) \ldots S_1 R_1(\theta_K) \quad + \\ \ldots & \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ + & E R_Q(\theta_1) S_Q \cdots\cdots\cdots R_1(\theta_j + \frac{\pi}{2}) S_1 R_1(\theta_K) \end{array} \tag{47}$$

where $\partial_j \equiv \frac{\partial}{\partial \theta_j}$, $j = 1, \ldots, K$, and in the last equality we used the fact that $\sin' \theta = \cos\theta = \sin(\theta + \pi/2)$ and $\cos' \theta = -\sin\theta = \cos(\theta + \pi/2)$.

It is possible to express the above derivative in terms of rotations of the original angles, making storage and computation more efficient. In fact, the derivative of a rotation block can be expressed as:

$$\tilde{\rho}\,'(\theta) = \tilde{\rho}(\theta + \pi/2) = \left[\begin{array}{cc} 0 & 1 \\ -1 & 0 \end{array} \right] \tilde{\rho}(\theta) \ , \tag{48}$$

so that

$$\partial_j R_l(\theta_j) = D_l R_l(\theta_j) \ , \tag{49}$$

where D_l is a block-diagonal matrix. Then (47) becomes:

$$
\begin{array}{rlr}
\multicolumn{3}{l}{\partial_j C(\theta_1, \ldots, \theta_K) =} \\
= & E R_Q(\theta_1) S_Q \cdots D_Q R_Q(\theta_j) \cdots S_1 R_1(\theta_K) & + \\
+ & E R_Q(\theta_1) S_Q \cdots\cdots D_{Q-1} R_{Q-1}(\theta_j) \cdots S_1 R_1(\theta_K) & + \\
\multicolumn{3}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\
+ & E R_Q(\theta_1) S_Q \cdots\cdots\cdots\cdots D_1 R_1(\theta_j) \cdots S_1 R_1(\theta_K) &
\end{array}
\tag{50}
$$

With the appropriate choice of the matrices $C_1, \ldots, C_Q$, the above expression allows the computation of the gradient with respect to the lattice angles for any basis in the wavelet packets table. It is then possible to combine a best basis algorithm with the optimization of the filters' coefficients in various ways (e.g. iterative two steps optimization). Moreover, given a basis in the wavelet packet table, the computation of the transform and of it's gradient with respect to the lattice angles can procede in parallel. Figure 2 shows the filter bank that computes two steps of the wavelet transform and its gradient for the case of two lattice angles.

It is possible to add additional contrains to the filter's coefficients – for example a certain number of vanishing moments can be required – but in general this leads to a constrained optimization. However, it is possible to impose a zero-mean conidition to the high-pass filter at very little cost, and recast the optimization into an unconstrained one. The details for this procedure are given in Appendix B.

V. Discussion

We have shown that it is possible to find an optimal mother wavelet with respect to a given objective function through an unconstrained optimization over a compact manifold, which guaranties the existence of a minimum. We explicitely reparametrized the filters' coefficients using the lattice decomposition so that the orthogonality constrains are automatically satisfied, and derived an expression for the gradient of the objective function with respect to the lattice angles. Any gradient based optimization can then be used to find the optimal solution. We showed that the framework we have developed is not constrained to wavelet transforms, but can be extended directly to a generic wavelet packets' basis. It is then possible to optimize the specific basis and the filters jointly.

Wavelet design is usually based on the optimization of wavelets according to some property of the filter itself, such as stopband attenuation, coding gain, or degree of smoothness. It has been shown that for some specific objective function it is convenient to optimize the filters coefficients in the time-domain [6]. This was possible because the optimization problem could be cast into a quadratic-constrained least-squares minimization one; moreover, the optimization was performed on a single step of the transform.

The constrained optimization leads to a polynomial with a degree proportional to the length of the filter and of the number of decomposition steps, which is in general linked to the length of the filter. Thus, realistic filters cannot be practically optimized due to the roughness of the polynomial error surface.

In our approach, the design is driven by the data themselves and the whole cascade of filters is taken into account. Preliminary work shows that there exist classes of signals for which the optimal solution is far from the classical wavelets developed with the above methods.

Furthermore, starting from the lattice formulation, it is simple to decompose the transform into lifting steps. In fact, every Givens rotation in the polyphase matrix (25) can be decomposed into three lifting steps [7]. Thus, once an optimal set of parameters are obtained through optimization, the corresponding lifting implementation can be used, making it possible to exploit all the advantages of the latter formulation.

Appendices

A. Generalization to arbitrary filter and signal lengths

This section describes how the factorization of the wavelet matrices can be extended to filters and signals of arbitrary even length, the length of the signal being a power of two. First, we rewrite $\Lambda(z)$ in (26) as:

$$\Lambda(z) = A + B z^{-1} \ , \tag{51}$$

where

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} , B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} , \tag{52}$$

and introduce

$$h_j^{(K)} = \begin{bmatrix} c_{2j}^{(K)} & c_{2j+1}^{(K)} \\ d_{2j}^{(K)} & d_{2j+1}^{(K)} \end{bmatrix} . \tag{53}$$

Then we can write the following series expansion of the polyphase matrix:

$$H_p^{(K)} = h_0^{(K)} + h_1^{(K)} z^{-1} + \cdots + h_{K-1}^{(K)} z^{-(K-1)}. \tag{54}$$

Case $K = 1$. This is the trivial case where:

$$h_0^{(1)} = \begin{bmatrix} c_0^{(1)} & c_1^{(1)} \\ d_0^{(1)} & d_1^{(1)} \end{bmatrix} = \rho_1 \ . \tag{55}$$

Case $K = 2$. Equation (25) gives:

$$
\begin{aligned}
h_0^{(2)} + h_1^{(2)} z^{-1} &= h_0^{(1)} \left(A + B z^{-1} \right) \rho_2 = \\
&= h_0^{(1)} A \rho_2 + h_0^{(1)} B \rho_2 z^{-1}
\end{aligned}
\tag{56}
$$

which is solved by equating the different powers of z separately:

$$
\begin{cases}
h_0^{(2)} = h_0^{(1)} A \rho_2 \\
h_1^{(2)} = h_0^{(1)} B \rho_2
\end{cases}
\tag{57}
$$

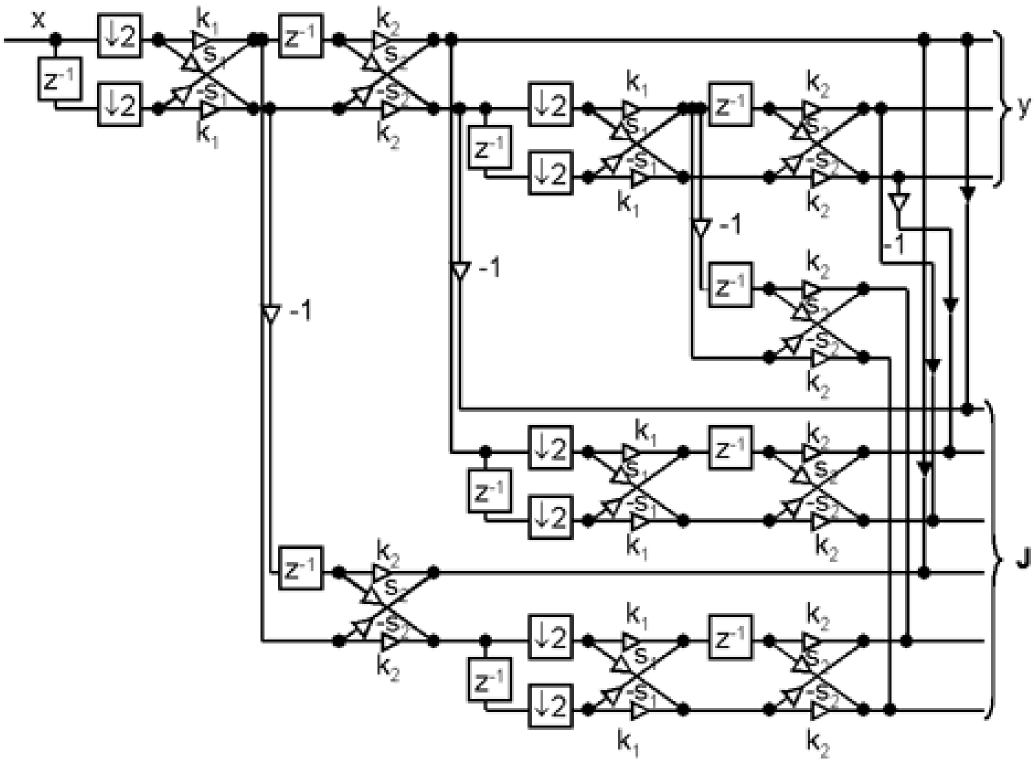

Fig. 2. Filter bank that computes two steps of the wavelet transform and its gradient for the case of two lattice angles.

or, in matrix form:

$$\left[h_0^{(2)}, h_1^{(2)}\right] = h_0^{(1)}[A, B]\begin{bmatrix} \rho_2 & 0 \\ 0 & \rho_2 \end{bmatrix}. \tag{58}$$

We introduce a flip operator for 2 by 2 matrices. This will allow us to build a convolution matrix for a causal FIR 2-filter bank. If

$$F = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \tag{59}$$

and V is a 2 by 2 matrix, then $\tilde{V} = VF$ is obtained by flipping each row of V, while $\hat{V} = FV$ is obtained by flipping each column of V. Since $F^2 = I$, equation (58) can be rewritten as:

$$\left[\tilde{h}_1^{(2)}, \tilde{h}_0^{(2)}\right] = \tilde{h}_0^{(1)}\left[\hat{B}, \hat{A}\right]\begin{bmatrix} \tilde{\rho}_2 & 0 \\ 0 & \tilde{\rho}_2 \end{bmatrix}. \tag{60}$$

Based on the preceding decomposition, it is easy to verify that the following holds:

$$\begin{bmatrix} \tilde{h}_1^{(2)} & \tilde{h}_0^{(2)} & & \\ & \tilde{h}_1^{(2)} & \tilde{h}_0^{(2)} & \\ & & & \ddots \end{bmatrix} = \begin{bmatrix} \tilde{h}_0^{(1)} & & \\ & \ddots & \\ & & \tilde{h}_0^{(1)} \end{bmatrix} \times \begin{bmatrix} \hat{B} & \hat{A} & & \\ & \hat{B} & \hat{A} & \\ & & & \ddots \end{bmatrix} \times \begin{bmatrix} \tilde{\rho}_2 & & \\ & \ddots & \\ & & \tilde{\rho}_2 \end{bmatrix} \tag{61}$$

The left hand side of equation (61) is the matrix that performs one step of the wavelet transform based on a filter with four coefficients. Moreover, the first matrix in the right hand side of the same equation is the wavelet matrix for a transform with filters of length 2.

Generic K**.** For K angles we have:

$$h_0^{(K)}+h_1^{(K)}z^{-1}+\cdots+h_{K-1}^{(K)}z^{-(K-1)}=$$
$$=\left[h_0^{(K-1)}+h_1^{(K-1)}z^{-1}+\cdots+h_{k-2}^{(K-1)}z^{-(K-2)}\right]\times$$
$$\times\left(A+Bz^{-1}\right)\rho_K=$$
$$=h_0^{(K-1)}A\rho_K+\left[h_0^{(K-1)}B+h_1^{(K-1)}A\right]\rho_K z^{-1}+\cdots$$
$$\cdots+h_{K-2}^{(K-1)}B\rho_K z^{-(K-1)} \tag{62}$$

which is true if

$$\left[h_0^{(K)},\ldots,h_{K-1}^{(K)}\right]=\left[h_0^{(K-1)},\ldots,h_{K-2}^{(K-1)}\right]\times \begin{bmatrix} A & B & & & \\ & A & B & & \\ & & & \ddots & \\ & & & A & B \end{bmatrix}\times \begin{bmatrix} \rho_K & & & \\ & \rho_K & & \\ & & \ddots & \\ & & & \rho_K \end{bmatrix}. \tag{63}$$

or, equivalently

$$\overbrace{\left[\tilde{h}_{K-1}^{(K)},\ldots,\tilde{h}_0^{(K)}\right]}^{2\times 2K}=\overbrace{\left[\tilde{h}_{K-2}^{(K-1)},\ldots,\tilde{h}_0^{(K-1)}\right]}^{2\times[2(K-1)]}\times \underbrace{\begin{bmatrix} \hat{B} & \hat{A} & & & \\ & \hat{B} & \hat{A} & & \\ & & & \ddots & \\ & & & \hat{B} & \hat{A} \end{bmatrix}}_{[2(K-1)]\times 2K}\underbrace{\begin{bmatrix} \tilde{\rho}_K & & & \\ & \tilde{\rho}_K & & \\ & & \ddots & \\ & & & \tilde{\rho}_K \end{bmatrix}}_{2K\times 2K}. \tag{64}$$

Using (64) as a building block it is possible to build a wavelet matrix for a signal of length 2^m:

$$\overbrace{\begin{bmatrix} \tilde{h}_K(K-1)\ldots\tilde{h}_K(0) \;\; 0\ldots\ldots\ldots 0 \\ 0 \;\; \tilde{h}_K(K-1)\ldots\tilde{h}_K(0) \;\; 0\ldots\ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \tilde{h}_K(K-2)\ldots\tilde{h}_{K-1}(0) \;\; 0\ldots 0 \;\; \tilde{h}_K(K-1) \end{bmatrix}}^{2^m\times 2^m}=$$
$$=\overbrace{\begin{bmatrix} \tilde{h}_{K-1}(K-2)\ldots\tilde{h}_{K-1}(0) \;\; 0\ldots\ldots\ldots 0 \\ 0 \;\; \tilde{h}_{K-1}(K-2)\ldots\tilde{h}_{K-1}(0) \;\; 0\ldots\ldots 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \tilde{h}_{K-1}(K-3)\ldots\tilde{h}_{K-1}(0) \;\; 0\ldots 0 \;\; \tilde{h}_{K-1}(K-2) \end{bmatrix}}^{2^m\times 2^m}\times$$
$$\times\underbrace{\overbrace{\begin{bmatrix} \hat{B} & \hat{A} & & \\ & \hat{B} & \hat{A} & \\ & & & \ddots \\ & & & \end{bmatrix}}^{2^m\times 2^m}}_{S}\underbrace{\overbrace{\begin{bmatrix} \tilde{\rho}_K & & & \\ & \tilde{\rho}_K & & \\ & & \ddots & \\ & & & \tilde{\rho}_K \end{bmatrix}}^{2^m\times 2^m}}_{R(\theta_K)}. \tag{65}$$

Notice that the above expression is nothing but a relation between two wavelet matrices with a different number of filter parameters. Thus, iterating (65), the first step of a wavelet transform, i.e. the analogous of matrix C_1 in (3), can be factorized as:

$$C_1=ER(\theta_1)SR(\theta_2)S\cdots SR(\theta_K), \tag{66}$$

where E separates the high-pass coefficients from the low-pass ones.

B. Zero-mean filters

It is convenient to require that the mean of the highpass filter be zero, so that no DC component is passed through the filter at any time. In the time-domain formulation this is equivalent to the following:

$$\sum_{n=0}^{N}(-1)^n c_{N-n}=0 \tag{67}$$

The above constrain can be easily translated into the lattice formulation. In fact, from the two equalities in (24) we get:

$$\textstyle\sum_{n=0}^{N}(-1)^n c_{N-n}=$$
$$=\textstyle\sum_{n=0}^{(N-1)/2}c_{2n}-\sum_{n=0}^{(N-1)/2}c_{2n+1}= \tag{68}$$
$$=\cos\left(\textstyle\sum_{j=1}^{K}\theta_j\right)-\sin\left(\textstyle\sum_{j=1}^{K}\theta_j\right)=0$$

which is true if

$$\sum_{j=1}^{K}\theta_j=\frac{\pi}{4}\,. \tag{69}$$

The search for the optimal filter coefficients is then a constrained optimization problem:

$$\vec{\theta}_{\text{min}} = \underset{\sum_{j=1}^{K}\theta_j=\pi/4}{\operatorname{argmin}} \ \phi(\theta_1,\ldots,\theta_K) \tag{70}$$

onto the plane defined by (69). It is however very simple to express the gradient of the objective function on this plane. Define the objective function on this plane as:

$$\Phi(\theta_1,\ldots,\theta_{K-1}) = \phi\left(\theta_1,\ldots,\frac{\pi}{4}-\sum_{j=1}^{K-1}\theta_j\right) \tag{71}$$

then its gradient is given by

$$\nabla\Phi(\theta_1,\ldots,\theta_{K-1}) = \begin{bmatrix} 1 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & -1 \end{bmatrix} \times \tag{72}$$
$$\times\nabla\phi\left(\theta_1,\ldots,\tfrac{\pi}{4}-\textstyle\sum_{j=1}^{K-1}\theta_j\right)$$

C. Boundary filters

Consider the case of a filter with four coefficients. Equation (38) can be rewritten as:

$$\overbrace{\begin{bmatrix} c_3^{(2)} & c_2^{(2)} & c_1^{(2)} & c_0^{(2)} \\ d_3^{(2)} & d_2^{(2)} & d_1^{(2)} & d_0^{(2)} \end{bmatrix}}^{H_2} = \begin{bmatrix} c_1^{(1)} & c_0^{(1)} \\ d_1^{(1)} & d_0^{(1)} \end{bmatrix} \times$$
$$\times \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} k_2 & s_2 & 0 & 0 \\ -s_2 & k_2 & 0 & 0 \\ 0 & 0 & k_2 & s_2 \\ 0 & 0 & -s_2 & k_2 \end{bmatrix} = \tag{73}$$
$$= \underbrace{\begin{bmatrix} c_1^{(1)} & c_0^{(1)} \\ d_1^{(1)} & d_0^{(1)} \end{bmatrix}}_{H_1} \underbrace{\begin{bmatrix} -s_2 & k_2 & 0 & 0 \\ 0 & 0 & k_2 & s_2 \end{bmatrix}}_{Q_2}$$

with $k_2^2+s_2^2=1$. It is easy to verify that $H_2H_2^T=I$, which simply expresses the orthonormality of the filters. However $H_2^TH_2 \neq I$ in general. In fact:

$$H_2^TH_2 = (H_1Q_2)^T(H_1Q_2) =$$
$$= Q_2^TH_1^TH_1Q_2 = Q_2^TQ_2 =$$
$$= \begin{bmatrix} -s_2 & 0 \\ k_2 & 0 \\ 0 & k_2 \\ 0 & s_2 \end{bmatrix} \begin{bmatrix} -s_2 & k_2 & 0 & 0 \\ 0 & 0 & k_2 & s_2 \end{bmatrix} = \tag{74}$$
$$= \begin{bmatrix} s_2^2 & -s_2k_2 & 0 & 0 \\ -s_2k_2 & k_2^2 & 0 & 0 \\ 0 & 0 & k_2^2 & s_2k_2 \\ 0 & 0 & s_2k_2 & s_2^2 \end{bmatrix}$$

where we used the fact that $H_1^TH_1=I$. If we define

$$P_2 = I - H_2^TH_2 =$$
$$= \begin{bmatrix} k_2^2 & s_2k_2 & 0 & 0 \\ s_2k_2 & s_2^2 & 0 & 0 \\ 0 & 0 & s_2^2 & -s_2k_2 \\ 0 & 0 & -s_2k_2 & k_2^2 \end{bmatrix} \tag{75}$$

then

$$H_2P_2 = H_2 - H_2H_2^TH_2 = H_2 - H_2 = 0 \tag{76}$$

This means that the columns of P_2 are orthogonal to the original filters in H_2. Notice that the first two columns are linearly dependent, and so are the last two. Thus, we have two vectors that are orthogonal to the original filters:

$$\begin{aligned} \vec{v}_{\text{left}} &= [k_2, s_2, 0, 0]^T \\ \vec{v}_{\text{right}} &= [0, 0, s_2, -k_2]^T \end{aligned} \tag{77}$$

For a filter of length six:

$$H_3^TH_3 = (H_2Q_3)^T(H_2Q_3) =$$
$$= (H_1Q_2Q_3)^T(H_1Q_2Q_3) = \left(Q_3^TQ_2^T\right)(Q_2Q_3) =$$
$$= \begin{bmatrix} -s_3 & 0 & 0 & 0 \\ k_3 & 0 & 0 & 0 \\ 0 & k_3 & -s_3 & 0 \\ 0 & s_3 & k_3 & 0 \\ 0 & 0 & 0 & k_3 \\ 0 & 0 & 0 & s_3 \end{bmatrix} \begin{bmatrix} -s_2 & 0 \\ k_2 & 0 \\ 0 & k_2 \\ 0 & s_2 \end{bmatrix} \times$$
$$\times \begin{bmatrix} -s_2 & k_2 & 0 & 0 \\ 0 & 0 & k_2 & s_2 \end{bmatrix} \times$$
$$\times \begin{bmatrix} -s_3 & k_3 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_3 & s_3 & 0 & 0 \\ 0 & 0 & -s_3 & k_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & k_3 & s_3 \end{bmatrix} = \tag{78}$$
$$= \begin{bmatrix} s_2s_3 & 0 \\ -s_2k_3 & 0 \\ k_2k_3 & -k_2s_3 \\ k_2s_3 & k_2k_3 \\ 0 & s_2k_3 \\ 0 & s_2s_3 \end{bmatrix} \times$$
$$\times \begin{bmatrix} s_2s_3 & -s_2k_3 & k_2k_3 & k_2s_3 & 0 & 0 \\ 0 & 0 & -k_2s_3 & k_2k_3 & s_2k_3 & s_2s_3 \end{bmatrix}$$

which gives

$$H_3^T H_3 =$$

$$\begin{bmatrix} s_2^2 s_3^2,\ -s_1^2 s_3 k_3,\ s_2 k_2 s_3 k_3,\ s_2 k_2 s_3^2,\ 0,\ 0 \\ -s_2^2 s_3 k_3,\ s_2^2 k_3^2,\ -s_2 k_2 k_3^2,\ -s_2 k_2 s_3 k_3,\ 0,\ 0 \\ s_2 k_2 s_3 k_3,\ -s_2 k_2 k_3^2,\ k_2^2,\ 0,\ -s_2 k_2 s_3 k_3,\ -s_2 k_2 s_3^2 \\ s_2 k_2 s_3^2,\ -s_2 k_2 s_3 k_3,\ 0,\ k_2^2,\ s_2 k_2 s_3 k_3,\ s_2 k_2 s_3^2 \\ 0,\ 0,\ -s_2 k_2 s_3 k_3,\ s_2 k_2 k_3^2,\ s_2^2 k_3^2,\ s_2^2 s_3 k_3 \\ 0,\ 0,\ -s_2 k_2 s_3^2,\ s_2 k_2 s_3 k_3,\ s_2^2 s_3 k_3,\ s_2^2 s_3^2 \end{bmatrix} \tag{79}$$

A matrix whose columns are orthogonal to the original filters can again be obtained as:

$$P_3 = I - H_3^T H_3 =$$

$$\begin{bmatrix} 1 - s_2^2 s_3^2,\ s_1^2 s_3 k_3,\ -s_2 k_2 s_3 k_3,\ -s_2 k_2 s_3^2,\ 0,\ 0 \\ s_2^2 s_3 k_3,\ 1 - s_2^2 k_3^2,\ s_2 k_2 k_3^2,\ s_2 k_2 s_3 k_3,\ 0,\ 0 \\ -s_2 k_2 s_3 k_3,\ s_2 k_2 k_3^2,\ s_2^2,\ 0,\ s_2 k_2 s_3 k_3,\ s_2 k_2 s_3^2 \\ -s_2 k_2 s_3^2,\ s_2 k_2 s_3 k_3,\ 0,\ s_2^2,\ -s_2 k_2 s_3 k_3,\ -s_2 k_2 s_3^2 \\ 0,\ 0,\ s_2 k_2 s_3 k_3,\ -s_2 k_2 k_3^2,\ 1 - s_2^2 k_3^2,\ -s_2^2 s_3 k_3 \\ 0,\ 0,\ s_2 k_2 s_3^2,\ -s_2 k_2 s_3 k_3,\ -s_2^2 s_3 k_3,\ 1 - s_2^2 s_3^2 \end{bmatrix} \tag{80}$$

This time, the first two columns of P_3 – and the last two as well – are linearly independent. Every linear combination will still be orthogonal to the filters. If we denote the first column of $H_3^T H_3$ as $\vec{u}$, the first two columns of P_3 can be rewritten as $\vec{v}_1 = [1,0,0,0,0,0]^T - \vec{u}$ and $\vec{v}_2 = [0,1,0,0,0,0]^T + k_3/s_3\,\vec{u}$ respectively. Then

$$\vec{v}_{\text{left}} = k_3\vec{v}_1 + s_3\vec{v}_2 = [k_3, s_3, 0, 0, 0, 0]^T \tag{81}$$

is orthogonal to the filters. The same argument can be applied to the last two columns obtaining another orthogonal vector:

$$\vec{v}_{\text{right}} = [0, 0, 0, 0, s_3, -k_3]^T \tag{82}$$

Acknowledgments

We would like to thank Raphi Coifman, Leon Cooper, Quyen Huynh and Truong Nguyen for many fruitful discussions. This work was partially supported by DARPA/ONR grants N00014-99-1-0009 and N00014-01-1-0306.

References

[1] Gilbert Strang and Truong Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA, 1996.

[2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice-Hall, Englewoods Cliffs, NJ, 1993.

[3] P. P. Vaidyanathan and P. Q. Hoang, "Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. 36, no. 1, pp. 81–94, 1988.

[4] Ronald R. Coifman and Mladen Victor Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[5] Mladen Victor Wickerhauser, *Adapted Wavelet Analysis from Theory to Software*, A K Peters, Wellesley, Massachusetts, 1994.

[6] T. Q. Nguyen, "Digital filter bank design – quadratic-constrained formulation," *IEEE Transactions on Signal Processing*, vol. 43, no. 9, pp. 2103–2108, 1995.

[7] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *The Journal of Fourier Analysis and Applications*, vol. 4, no. 3, pp. 247–269, 1998.