

Mathematical Foundations of Data Sciences



Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr
www.gpeyre.com
www.numerical-tours.com

November 12, 2017

Chapter 4

Wavelets

The reference for this chapter is [29].

4.1 Multi-resolution Approximation Spaces

A multiresolution approximation of $L^2(\mathbb{R})$ is a set of nested closed subspaces $(V_j)_j$

$$L^2(\mathbb{R}) \supset \dots \supset V^{j-1} \supset V_j \supset V_{j+1} \supset \dots \supset \{0\} \quad (4.1)$$

which must be related one from each other by dyadic scaling and must also be stable by dyadic translation

$$f \in V_j \iff f(\cdot/2) \in V_{j+1} \quad \text{and} \quad f \in V_j \iff \forall n \in \mathbb{Z}, f(\cdot + t2^j) \in V_j$$

So large j corresponds to coarse approximation spaces, and 2^j is often call the “scale”.

The limit on the left of (4.1) means that $\cup_j V_j$ is dense in $L^2(\mathbb{R})$, or equivalently that $P_{V_j}(f) \rightarrow f$ as $j \rightarrow -\infty$ where P_V is the orthogonal projector on V

$$P_V(f) = \operatorname{argmin}_{f' \in V} \|f - f'\|.$$

The limit on the right of (4.1) means that $\cap_j V_j = \{0\}$, or equivalently that $P_{V_j}(f) \rightarrow 0$ as $j \rightarrow +\infty$.

Scaling functions. We also require that there exists a scaling function $\varphi \in L^2(\mathbb{R})$ so that

$$\{\varphi(\cdot - n)\}_n \text{ is an Hilertian orthonormal basis of } V_0.$$

By the dilation property, this implies that

$$\{\varphi_{j,n}\}_n \text{ is an Hilertian orthonormal basis of } V_j \quad \text{where} \quad \varphi_{j,n} \stackrel{\text{def.}}{=} \frac{1}{2^{j/2}} \varphi\left(\frac{\cdot - 2^j n}{2^j}\right).$$

Note that one then has

$$P_{V_j}(f) = \sum_n \langle f, \varphi_{j,n} \rangle \varphi_{j,n}.$$

Figure 4.1 illustrates the translation and scaling effect.

A typical example is approximation by piecewise constant signals

$$V_j \stackrel{\text{def.}}{=} \{f \in L^2(\mathbb{R}) ; \forall n, f \text{ is constant on } [2^j n, 2^j(n+1)[\}, \quad (4.2)$$

in which case one can use $\varphi = 1_{[0,1[}$ and $\varphi_{j,n} = 2^{-j/2} 1_{[2^j n, 2^j(n+1)[}$.

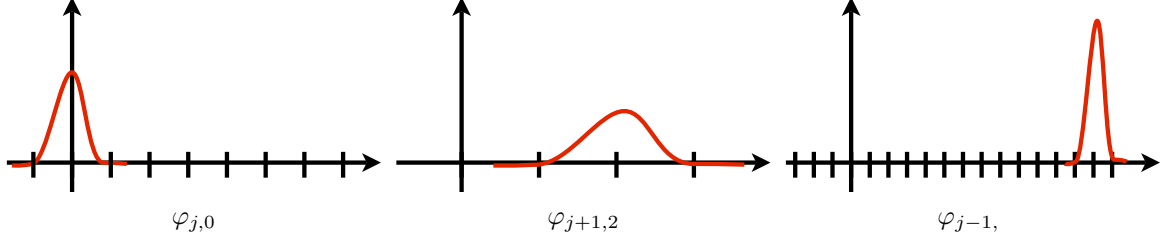


Figure 4.1: Translation and scaling to generate approximation spaces.

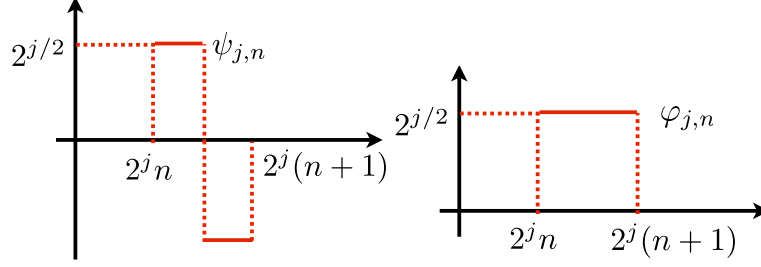


Figure 4.2: Haar scaling (left) and wavelet (right) functions.

Spectral orthogonalization. In many case of practical interest, the space V_j is describe by a translation-invariant basis which is not-orthogonal, $V_j = \text{Span}(\theta(\cdot - n))_{n \in \mathbb{Z}}$. The following proposition shows how to orthogonalize it using the Fourier transform.

Proposition 9. For $\theta \in L^2(\mathbb{R})$ (assumed regular and with fast enough decay), $\{\theta(\cdot - n)\}_{n \in \mathbb{Z}}$ is orthonormal if and only if

$$\forall \omega, \quad A(\omega) \stackrel{\text{def.}}{=} \sum_k |\hat{\theta}(\omega - 2\pi k)|^2 = 1.$$

If there exists $0 < a \leq b < +\infty$ such that $a \leq A(\omega) \leq b$, then φ defined by

$$\hat{\varphi}(\omega) = \frac{\hat{\theta}(\omega)}{\sqrt{A(\omega)}}$$

is such that $\{\varphi(\cdot - n)\}_{n \in \mathbb{Z}}$ is an Hilbertian basis of $\text{Span}\{\theta(\cdot - n)\}_{n \in \mathbb{Z}}$.

Proof. One has that $\{\theta(\cdot - n)\}_{n \in \mathbb{Z}}$ is orthonormal if and only if

$$\langle \theta, \theta(\cdot - n) \rangle = (\theta \star \bar{\theta})(n) = \delta_{0,n} \stackrel{\text{def.}}{=} \begin{cases} 0 & \text{if } n \neq 0, \\ 1 & \text{otherwise.} \end{cases}$$

where $\bar{\theta} = \theta(\cdot - \cdot)$ and δ_0 is the discrete Dirac vector. The Poisson summation formula (1.8) then reads

$$\sum_n \mathcal{F}(\theta \star \bar{\theta})(\omega - 2\pi n) = \sum_n \delta_{0,n} e^{-i\omega} = 1.$$

We conclude with the Fourier-convolution formula (2.6) shows that $\mathcal{F}(\theta \star \bar{\theta})(\omega) = \hat{\theta}(\omega) \hat{\theta}(\omega)^* = |\hat{\theta}(\omega)|^2$ which leads to the desired formula, and it is if and only if. Normalizing by $1/\sqrt{A(\omega)}$, which is a bounded function, shows that $\hat{\varphi}$ satisfies $\sum_k |\hat{\varphi}(\omega - 2\pi k)|^2 = 1$. \square

A typical example of application is spline (e.g. cubic ones) interpolations, which are generated by the box-spline function θ which is a piecewise polynomial with a compact support.

4.2 Multi-resolution Details Spaces

The details spaces are defined as orthogonal complement of $V_j \subset V_{j-1}$, which is legit because these are closed subspaces

$$\forall j, \quad W_j \text{ is such that } V_{j-1} = V_j \oplus^\perp W_j.$$

This leads to the following sequence of embedded spaces

$$L^2(\mathbb{R}) \longrightarrow \cdots \begin{array}{c} \searrow \\ \nearrow \end{array} \begin{array}{c} V_{j-1} \\ W_{j-1} \end{array} \begin{array}{c} \searrow \\ \nearrow \end{array} \begin{array}{c} V_j \\ W_j \end{array} \begin{array}{c} \searrow \\ \nearrow \end{array} \begin{array}{c} V_{j+1} \\ W_{j+1} \end{array} \begin{array}{c} \searrow \\ \nearrow \end{array} \cdots \longrightarrow \{0\}$$

Once again, we suppose that W_0 has an Hilbertian ortho-basis of the form $\{\psi(\cdot - n)\}_n$, so that

$$\{\psi_{j,n}\}_n \text{ is an Hilertian orthonormal basis of } V_j \quad \text{where} \quad \psi_{j,n} \stackrel{\text{def.}}{=} \frac{1}{2^{j/2}} \psi\left(\frac{\cdot - 2^j n}{2^j}\right).$$

Due to the orthogonal complementarity property, one has

$$L^2(\mathbb{R}) = \bigoplus_{j=-\infty}^{j=+\infty} W_j = V_{j_0} \bigoplus_{j \leq j_0} V_j.$$

This means that for all $f \in L^2(\mathbb{R})$, one has the following convergence in $L^2(\mathbb{R})$,

$$f = \lim_{(j_-, j_+) \rightarrow (-\infty, +\infty)} \sum_{j=j_-}^{j_+} P_{V_j} f = \lim_{j_+ \rightarrow +\infty} P_{j_0} f + \sum_{j=j_0}^{j_+} P_{V_j} f.$$

This decomposition shows that

$$\{\psi_{j,n} ; (j, n) \in \mathbb{Z}^2\}$$

is an Hilbertian orthogonal basis of $L^2(\mathbb{R})$, which is called a wavelet basis. One also have a “truncated” ortho-basis

$$\{\psi_{j,n} ; j \leq j_0, n \in \mathbb{Z}\} \cup \{\varphi_{j_0,n} ; n \in \mathbb{Z}\}.$$

A (forward) Wavelet transform corresponds to the computation of all the inner products of some function f with the elements of these basis.

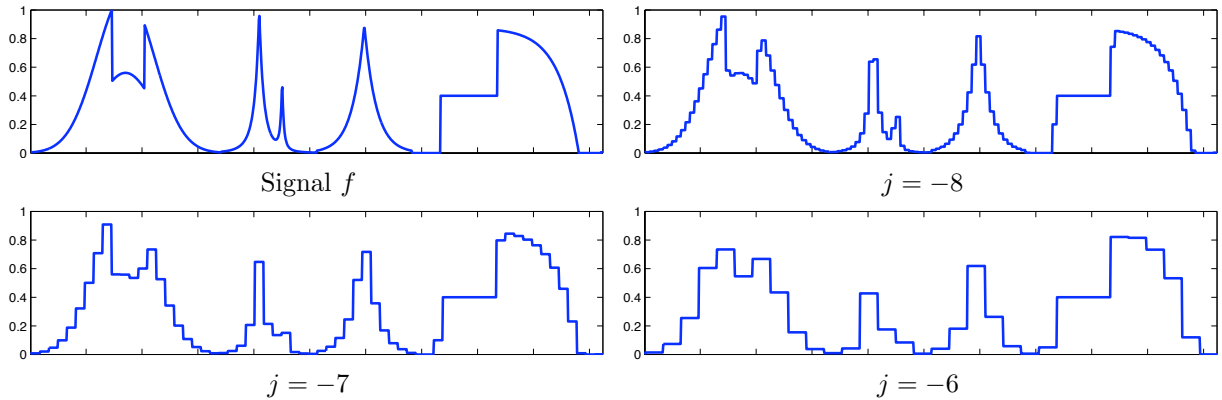


Figure 4.3: 1-D Haar multiresolution projection $P_{V_j} f$ of a function f .

Haar wavelets. For the Haar multiresolution (4.2), one has

$$W_j = \left\{ f ; \forall n \in \mathbb{Z}, f \text{ constant on } [2^{j+1}n, 2^{j+1}(n+1)) \text{ and } \int_{n2^j}^{(n+1)2^j} f = 0 \right\}. \quad (4.3)$$

A possible choice for a mother wavelet function is

$$\psi(t) = \frac{1}{\sqrt{2}} \begin{cases} 1 & \text{for } 0 \leq t < 1/2, \\ -1 & \text{for } 1/2 \leq t < 1, \\ 0 & \text{otherwise,} \end{cases}$$

as shown on Figure 4.2, right.

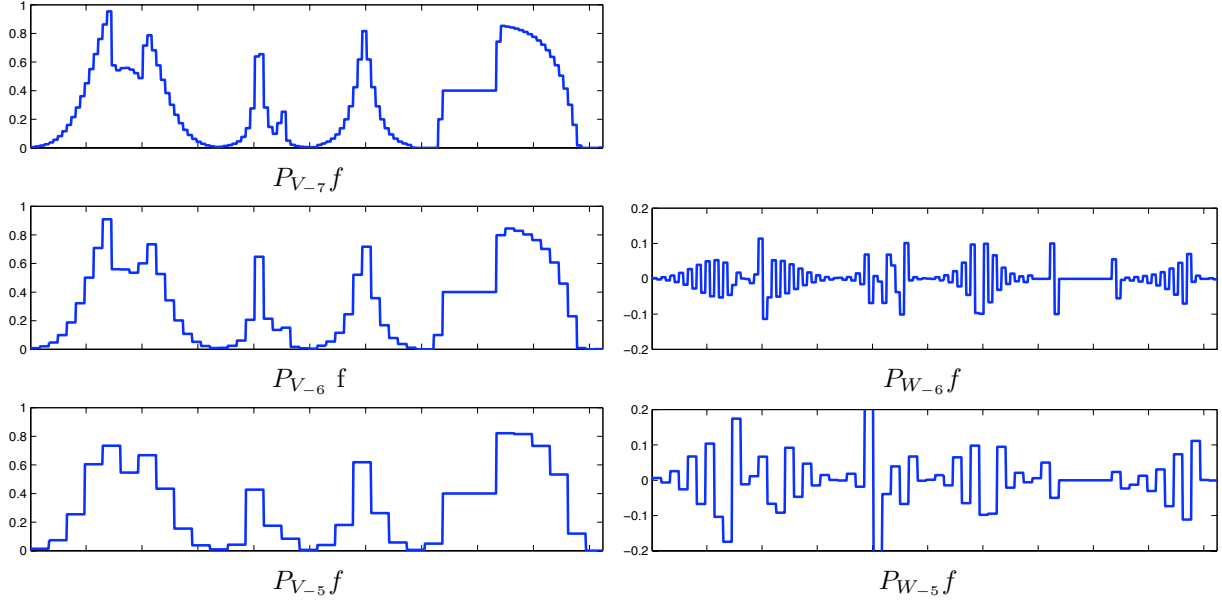


Figure 4.4: Projection on Haar approximation spaces (left) and detail spaces (right).

Figure 4.4 shows examples of projections on details spaces, and how they can be derived from projection on approximation spaces.

4.3 On Bounded Domains

On a periodic bounded domain $\mathbb{T} = \mathbb{R}/\mathbb{Z}$ (note that we use here 1-periodicity, in contrast to the convention we used for Fourier series of 2π -periodicity), one obtains an orthogonal wavelet basis of $L^2(\mathbb{T})$ by periodizing the original wavelets, and also restricting the translation points $2^j n$ to be in $[0, 1]$, i.e. $0 \leq n < 2^{-j}$. Similarly to (1.6), the periodization of a function $f \in L^1(\mathbb{R})$ is the function

$$f^P = \sum_{n \in \mathbb{Z}} f(\cdot - n) \in L^1(\mathbb{T}).$$

The wavelet basis is thus define as

$$\{\psi_{j,n}^P ; j \leq j_0, 0 \leq n < 2^{-j}\} \cup \{\varphi_{j_0,n}^P ; 0 \leq n < 2^{-j_0}\}.$$

and one verifies that it defines an Hilbertian ortho-basis of $L^2(\mathbb{T})$. It is possible to define wavelet basis using Neumann (mirror) boundary conditions, but this is more involved.

4.4 Fast Wavelet Transform

4.4.1 Discretization

We now work over \mathbb{R}/\mathbb{Z} . The modelling hypothesis is that one has access to a discrete signal $a_J \in \mathbb{R}^N$ with $N = 2^{-J}$ at some fixed scale 2^J , and that this signal exactly matches the inner-product with the scaling functions, i.e.

$$\forall n, \in \{0, \dots, N-1\}, \quad a_{J,n} = \langle f, \varphi_{J,n}^P \rangle \quad (4.4)$$

for some function of interest f we are sampling. This is equivalent to saying that the discretization process have exactly access to $P_{V_J}f$. This hypothesis is questionable, and similar to the Shannon bandlimit assumption. In practice, the scaling functions $\varphi_{j,n}$ are often quite close to the point-spread function of the acquisition device, so it is acceptable. One can however improves this by correcting the acquired values by the device to be closer to assumption (4.4).

The discrete wavelet transform then computes, from this input a_J , all the coefficients

$$\forall j \in \{J+1, J+2, \dots, 0\}, \quad \forall n \in \llbracket 0, 2^{-j} - 1 \rrbracket, \quad a_{j,n} \stackrel{\text{def.}}{=} \langle f, \varphi_{j,n}^P \rangle, \quad \text{and} \quad d_{j,n} \stackrel{\text{def.}}{=} \langle f, \psi_{j,n}^P \rangle$$

in this order (increasing values of j). During the algorithm, the previously computed vector a_j can be discarded, and only the d_j are kept.

The forward discrete wavelet transform on a bounded domain is thus the orthogonal finite dimensional map

$$a_J \in \mathbb{R}^N \mapsto \{d_{j,n} ; 0 \leq j < J, 0 \leq n < 2^{-j}\} \cup \{a_0 \in \mathbb{R}\}.$$

The inverse transform, which is thus the adjoint due to orthogonality, is the inverse map.

Figure 4.5 shows examples of wavelet coefficients. For each scale 2^j , there are 2^{-j} coefficients.

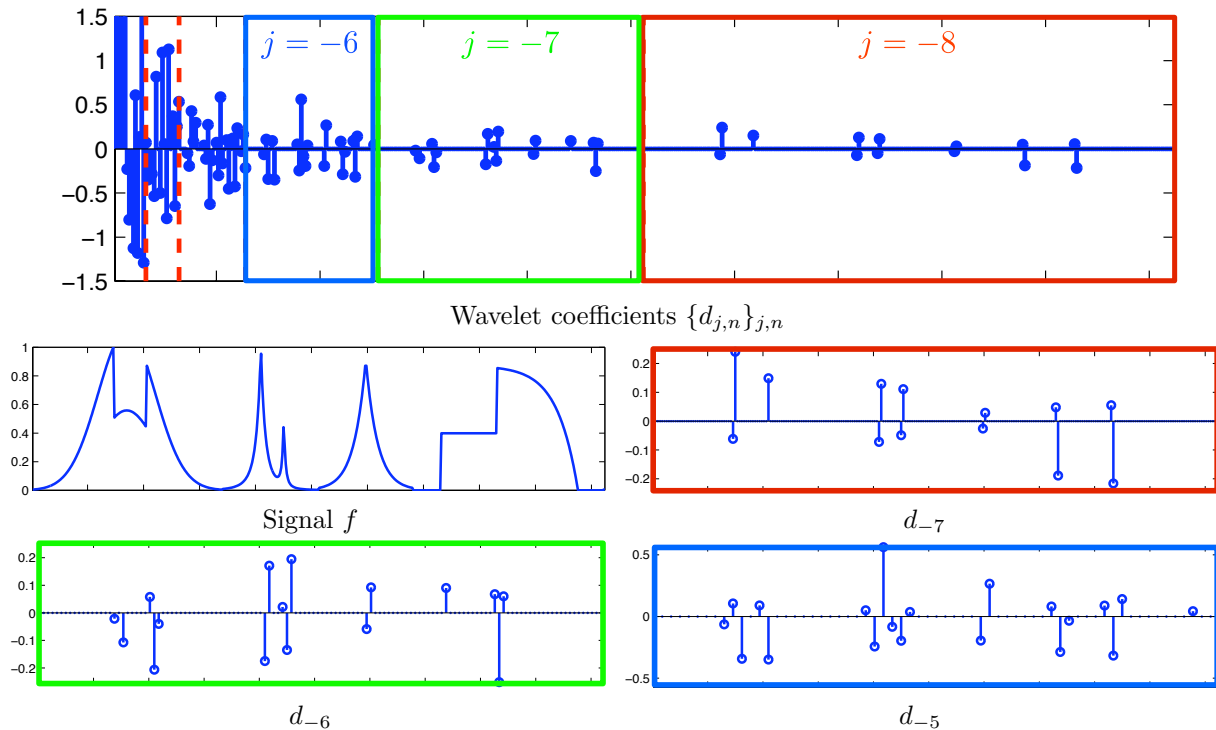


Figure 4.5: Wavelet coefficients. Top row: all the coefficients. Bottoms rows: zoom on the different scales

4.4.2 Forward Fast Wavelet Transform (FWT)

The algorithm proceeds by computing a series of simple operators

$$\forall j = J+1, \dots, 0, \quad (a_j, d_j) = \mathcal{W}_j(a_{j-1}) \quad \text{where} \quad \mathcal{W}_j : \mathbb{R}^{2^{-j+1}} \rightarrow \mathbb{R}^{2^{-j}} \times \mathbb{R}^{2^{-j}} \quad (4.5)$$

The number of such steps is thus $|J| = \log_2(N)$. Each \mathcal{W}_j is orthogonal since it corresponds to linear maps between coefficients in orthogonal bases.

In order to describe the algorithm that computes \mathcal{W}_j , we introduce the filters “filter” coefficients $f, g \in \mathbb{R}^{\mathbb{Z}}$

$$h_n \stackrel{\text{def.}}{=} \frac{1}{\sqrt{2}} \langle \varphi(\cdot/2), \varphi(\cdot - n) \rangle \quad \text{and} \quad g_n \stackrel{\text{def.}}{=} \frac{1}{\sqrt{2}} \langle \psi(\cdot/2), \varphi(\cdot - n) \rangle. \quad (4.6)$$

We denote as $\downarrow_2 : \mathbb{R}^K \rightarrow \mathbb{R}^{K/2}$ the subsampling operator by a factor of 2, i.e.

$$u \downarrow_2 \stackrel{\text{def.}}{=} (u_0, u_2, u_4, \dots, u_{K-2}, u_K).$$

In the following, we assume that these filters are decaying fast enough.

Proposition 10. *One has*

$$\mathcal{W}_j(a_j) = ((\bar{h} \star a_{j-1}) \downarrow_2, (\bar{g} \star a_{j-1}) \downarrow_2), \quad (4.7)$$

where \star denotes periodic convolutions on \mathbb{R}^{-j+1} .

Proof. We note that $\varphi(\cdot/2)$ and $\psi(\cdot/2)$ are in \mathcal{W}_j , so one has the decompositions

$$\frac{1}{\sqrt{2}} \varphi(t/2) = \sum_n h_n \varphi(t - n) \quad \text{and} \quad \frac{1}{\sqrt{2}} \psi(t/2) = \sum_n g_n \varphi(t - n) \quad (4.8)$$

Doing the change of variable $t \mapsto \frac{t-2^j p}{2^{j-1}}$ in (4.8), one obtains

$$\frac{1}{\sqrt{2}} \varphi\left(\frac{t-2^j p}{2^j}\right) = \sum_n h_n \varphi\left(\frac{t}{2^{j-1}} - (n+2p)\right)$$

(similarly for ψ) and then doing the change $n \mapsto n - 2p$, one obtains

$$\varphi_{j,p} = \sum_{n \in \mathbb{Z}} h_{n-2p} \varphi_{j-1,n} \quad \text{and} \quad \psi_{j,p} = \sum_{n \in \mathbb{Z}} g_{n-2p} \psi_{j-1,n}.$$

When working with periodized function $(\varphi_{j,n}^P, \psi_{j,p}^P)$, this formula is still valid, but the summation over $n \in \mathbb{Z}$ should be done modulo 2^{-j+1} . Taking inner product of both size with respect to f (which is legit if h, g are decaying fast enough), one obtains the fundamental recursion formula

$$a_{j,p} = \sum_{n \in \mathbb{Z}} h_{n-2p} a_{j-1,n} = (\bar{h} \star a_{j-1})_{2p} \quad \text{and} \quad d_{j,p} = \sum_{n \in \mathbb{Z}} g_{n-2p} a_{j-1,n} = (\bar{g} \star a_{j-1})_{2p} \quad (4.9)$$

where $\bar{u}_n \stackrel{\text{def.}}{=} u_{-n}$. One can show that this formula is still valid when working over a bounded interval $\mathbb{T} = \mathbb{R}/\mathbb{Z}$, but then \star denotes the periodic convolution over $\mathbb{Z}/2^{-j+1}\mathbb{Z}$. \square

Figure 4.6 shows two steps of application of these refinement relationships.

The FWT thus operates as follow:

- **Input:** signal $f \in \mathbb{C}^N$.
- **Initialization:** $a_J = f$.
- **For** $j = J, \dots, j_0 - 1$.

$$a_{j+1} = (a_j \star \tilde{h}) \downarrow_2 \quad \text{and} \quad d_{j+1} = (a_j \star \tilde{g}) \downarrow_2$$

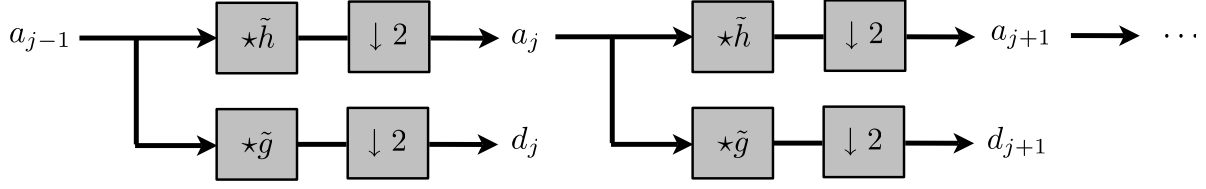


Figure 4.6: Forward filter bank decomposition.

– **Output:** the coefficients $\{d_j\}_{j_0 \leq j < J} \cup \{a_{j_0}\}$.

If $|h|, |g| \leq C$ so that both filter are compactly supported, then computing each \mathcal{W}_j is $(2C)2^{-j}$ operation, so that the complexity of the whole wavelet transform is

$$\sum_{j=J}^1 (2C)2^{-j} = (2C)2^{-J} = 2CN.$$

This shows that the fast wavelet transform is a linear time algorithm. Figure 4.7 shows the process of extracting iteratively the wavelet coefficients. Figure 4.8 shows an example of computation, where at each iteration, the coefficients of a_j and d_j are added to the left of the output vector.

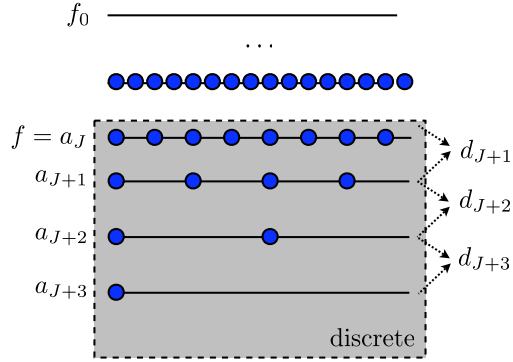


Figure 4.7: Pyramid computation of the coefficients.

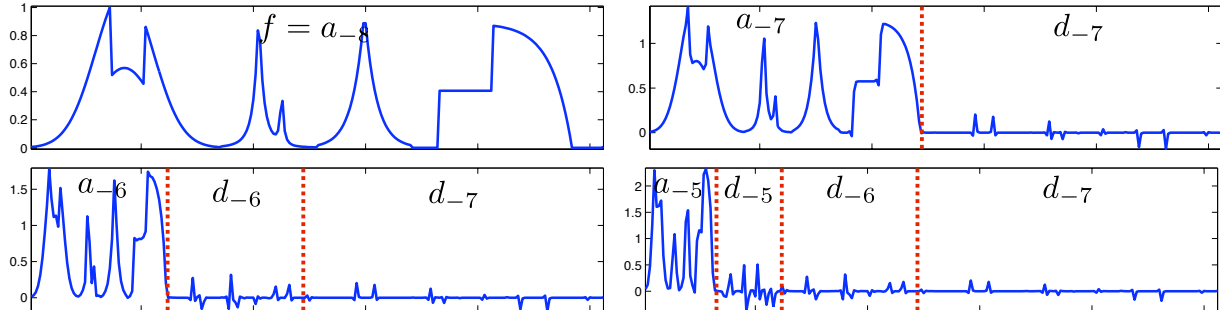


Figure 4.8: Wavelet decomposition algorithm.

Fast Haar transform. For the Haar wavelets, one has

$$\begin{aligned}\varphi_{j,n} &= \frac{1}{\sqrt{2}}(\varphi_{j-1,2n} + \varphi_{j-1,2n+1}), \\ \psi_{j,n} &= \frac{1}{\sqrt{2}}(\varphi_{j-1,2n} - \varphi_{j-1,2n+1}).\end{aligned}$$

This corresponds to the filters

$$\begin{aligned}h &= [\dots, 0, h[0] = \frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0, \dots], \\ g &= [\dots, 0, h[0] = \frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}, 0, \dots].\end{aligned}$$

The Haar wavelet transform algorithm thus processes by iterating averaging and differences:

- **Input:** signal $f \in \mathbb{C}^N$.
- **Initialization:** $a_J = f$.
- **For** $j = J, \dots, j_0 - 1$.

$$a_{j+1,n} = \frac{1}{\sqrt{2}}(a_{j-1,2n} + a_{j-1,2n+1}) \quad \text{and} \quad d_{j+1,n} = \frac{1}{\sqrt{2}}(a_{j-1,2n} - a_{j-1,2n+1}).$$

- **Output:** the coefficients $\{d_j\}_{j_0 \leq j < J} \cup \{a_{j_0}\}$.

4.4.3 Inverse Fast Transform (iFWT)

The inverse algorithm proceeds by inverting each step (4.10)

$$\forall j = 0, \dots, J+1, \quad a_{j-1} = \mathcal{W}_j^{-1}(a_j, d_j), \quad (4.10)$$

where \mathcal{W}_j^* is the adjoint for the canonical inner product on $\mathbb{R}^{2^{-j+1}}$, i.e. when viewed as a matrix, the transpose.

We denote $\uparrow_2: \mathbb{R}^{K/2} \rightarrow \mathbb{R}^K$ the up-sampling operator

$$a \uparrow_2 = (a_0, 0, a_1, 0, \dots, 0, a_{K/2}, 0) \in \mathbb{R}^K.$$

Proposition 11. *One has*

$$\mathcal{W}_j^{-1}(a_j, d_j) = (a_j \uparrow_2) \star h + (d_j \uparrow_2) \star g.$$

Proof. Since \mathcal{W}_j is orthogonal, $\mathcal{W}_j^{-1} = \mathcal{W}_j^*$. We write the whole transform as

$$\mathcal{W}_j = S_2 \circ C_{\bar{h}, \bar{g}} \circ \mathcal{D} \quad \text{where} \quad \begin{cases} \mathcal{D}(a) = (a, a), \\ C_{\bar{h}, \bar{g}}(a, b) = (\bar{h} \star a, \bar{g} \star a), \\ S_2(a, b) = (a \downarrow_2, b \downarrow_2). \end{cases}$$

One has the following adjoint operator

$$\mathcal{D}^*(a, b) = a + b, \quad C_{\bar{h}, \bar{g}}^*(a, b) = (h \star a, g \star b), \quad \text{and} \quad S_2(a, b) = (a \uparrow_2, b \uparrow_2).$$

Indeed, let us check this for the convolution, assuming involved sequences are in ℓ_1 (they are actually finite sums when considering periodic signals),

$$\langle f \star \bar{h}, g \rangle = \sum_n (f \star \bar{h})_n g_n = \sum_n \sum_k f_k h_{k-n} g_n = \sum_k f_k \sum_n h_{k-n} g_n = \langle f, h \star g \rangle,$$

for the copying

$$\langle \mathcal{D}(a), (u, v) \rangle = \langle a, u \rangle + \langle b, u \rangle = \langle a, u + v \rangle = \langle a, \mathcal{D}^*(u, v) \rangle,$$

and for the down-sampling

$$\langle f \downarrow_2, g \rangle = \sum_n f_{2n} g_n = \sum_n (f_{2n} g_n + f_{2n+1} 0) = \langle f, g \uparrow_2 \rangle.$$

This is shown using matrix notations in Figure 4.9. Putting everything together gives the desired formula. \square

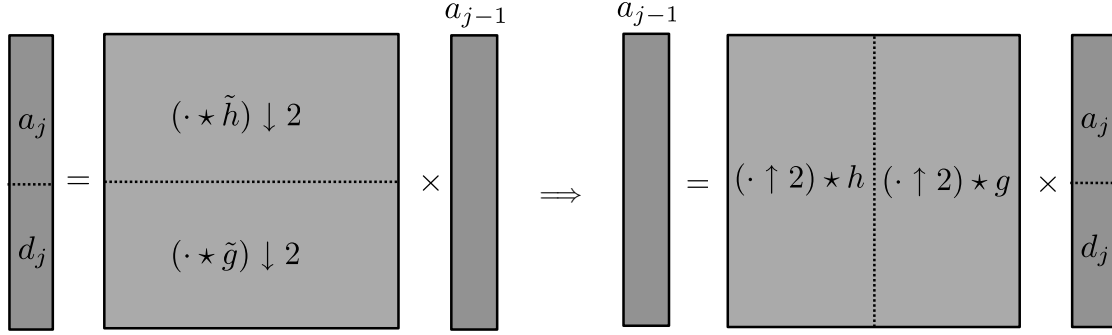


Figure 4.9: Wavelet inversion in matrix format.

The inverse Fast wavelet transform iteratively applies this elementary step

– **Input:** $\{d_j\}_{j_0 \leq j < J} \cup \{a_{j_0}\}$.

– **For** $j = j_0, \dots, J+1$.

$$a_{j-1} = (a_j \uparrow 2) \star h + (d_j \uparrow 2) \star g.$$

– **Output:** $f = a_J$.

This process is shown using a block diagram in Figure 4.10, which is the inverse of the block diagram 4.6.

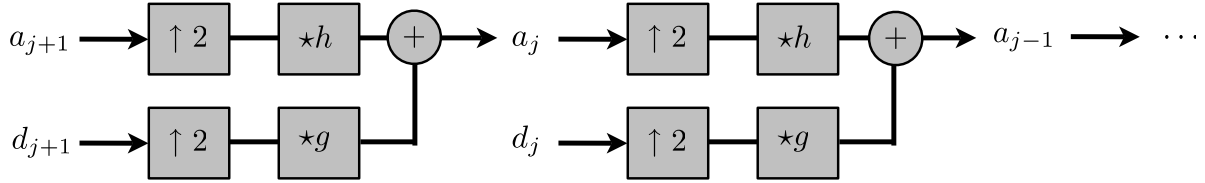


Figure 4.10: Backward filterbank recomposition algorithm.

4.5 2-D Wavelets

4.5.1 Anisotropic Wavelets

2-D anisotropic wavelets are defined using tensor product of Wavelet basis functions (2.12). The basis over \mathbb{T}^2 is thus of the form

$$\{\psi_{(j_1, j_2), (n_1, n_2)} ; (j_1, j_2) \in \mathbb{Z}^2, 0 \leq n_1 < 2^{-j_1}, 0 \leq n_2 < 2^{-j_2}\}$$

$$\text{where } \psi_{(j_1, j_2), (n_1, n_2)}(x_1, x_2) = \psi_{j_1, n_1}(x_1) \psi_{j_2, n_2}(x_2). \quad (4.11)$$

The computation of the fast anisotropic Wavelet transform in 2-D is similar to the 2-D FFT detailed in Section 2.5.2. Viewing the input image $a_J \in \mathbb{R}^{2^{-J} \times 2^{-J}}$ as a matrix, one first apply the 1-D FWT to each row, and then to each column, resulting in a linear time $O(N)$ algorithm, where $N = 2^{-2J}$.

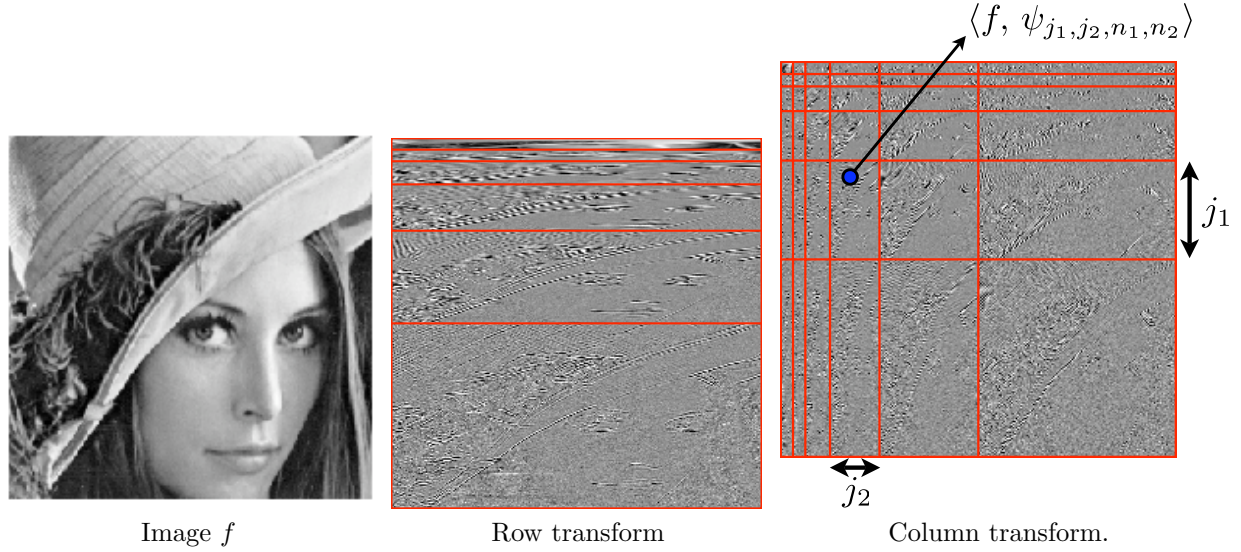


Figure 4.11: Steps of the anisotropic wavelet transform.

4.5.2 Isotropic Wavelets

A major issue with these anisotropic wavelet (4.11) is that a function $\psi_{(j_1, j_2), (n_1, n_2)}$ is scaled independently in each direction, leads to functions concentrated along an axis-oriented rectangle of size $2^{-j_1} \times 2^{j_2}$. This is not a very good features (since natural images usually do not exhibit such an anisotropy) and typically leads to visually unpleasant artifacts when used for processing (denoising or compression).

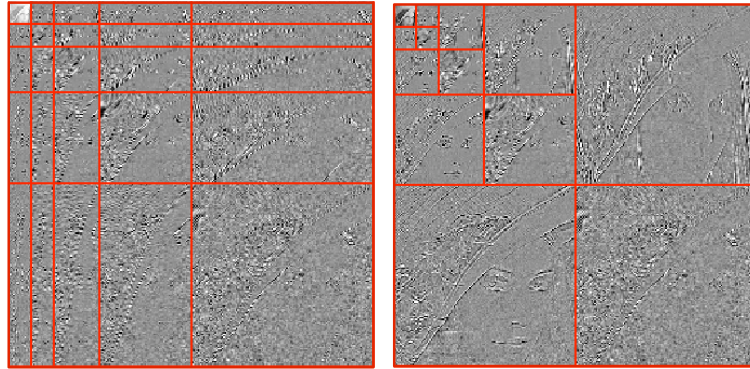


Figure 4.12: Anisotropic (left) versus isotropic (right) wavelet coefficients.

One rather use isotropic wavelet obtained by considering 2-D multi-resolution, obtained by tensor products of the 1-D approximation spaces

$$L^2(\mathbb{R}^2) \supset \dots \supset V^{j-1} \otimes V^{j-1} \supset V_j \otimes V_j \supset V_{j+1} \otimes V_{j+1} \supset \dots \supset \{0\}.$$

In the following, we denote

$$V_j^O \stackrel{\text{def.}}{=} V_j \otimes V_j$$

this isotropic 2-D multiresolution space.

Recall that the tensor product of two space $(V_1, V_2) \in L^2(\mathbb{R})^2$ is

$$V_1 \otimes V_2 = \text{Closure} \left(\text{Span} \{ f_1(x_1) f_2(x_2) \in L^2(\mathbb{R}^2) ; f_1 \in V_1, f_2 \in V_2 \} \right).$$

If $(\varphi_k^s)_k$ are Hilbertian bases for V_s , then one can show that $(\varphi_k^1(x_1) \varphi_k^2(x_2))_k$ is an Hilbertian basis for $V_1 \otimes V_2$.

One easily verify that one has the distributivity

$$(V_j \oplus^\perp W_j) \otimes (V_j \oplus^\perp W_j) = V_j^O \oplus^\perp W_j^V \oplus^\perp W_j^V \oplus^\perp W_j^V \quad \text{where} \quad \begin{cases} W_j^V \stackrel{\text{def.}}{=} (V_j \otimes W_j), \\ W_j^H \stackrel{\text{def.}}{=} (W_j \otimes V_j), \\ W_j^D \stackrel{\text{def.}}{=} (W_j \otimes W_j). \end{cases}$$

Here the letters $\{V, H, D\}$ stands for *Vertical, Horizontal, Diagonal* detail spaces. This leads to the following diagram of embedded spaces

$$\begin{array}{ccccccc} L^2(\mathbb{R}^2) & \rightarrow & \cdots & \rightarrow & V_{j-1} \otimes V_{j-1} & \rightarrow & V_j \otimes V_j & \rightarrow & V_{j+1} \otimes V_{j+1} & \rightarrow & \cdots & \rightarrow & \{0\} \\ & & & \searrow & & & \searrow & & \searrow & & & & \\ & & & & W_{j-1}^{(2)} & & W_j^{(2)} & & W_{j+1}^{(2)} & & & & \end{array}$$

For $j \in \mathbb{Z}$, each of the three wavelet spaces is spanned with a wavelet, where basis elements are indexed by $n = (n_1, n_2) \in \mathbb{Z}$ (or in $\{0, \dots, 2^{-j} - 1\}^2$ on the interval \mathbb{T}),

$$\forall \omega \in \{V, H, D\}, \quad W_j^\omega = \text{Span} \{ \psi_{j,n_1,n_2}^\omega \}_{n_1,n_2}$$

where

$$\forall \omega \in \{V, H, D\}, \quad \psi_{j,n_1,n_2}^\omega(x) = \frac{1}{2^j} \psi^\omega \left(\frac{x_1 - 2^j n_1}{2^j}, \frac{x_2 - 2^j n_2}{2^j} \right)$$

and where the three mother wavelets are

$$\psi^H(x) = \psi(x_1) \varphi(x_2), \quad \psi^V(x) = \varphi(x_1) \psi(x_2), \quad \text{and} \quad \psi^D(x) = \psi(x_1) \psi(x_2).$$

Figure 4.13 displays an examples of these wavelets.

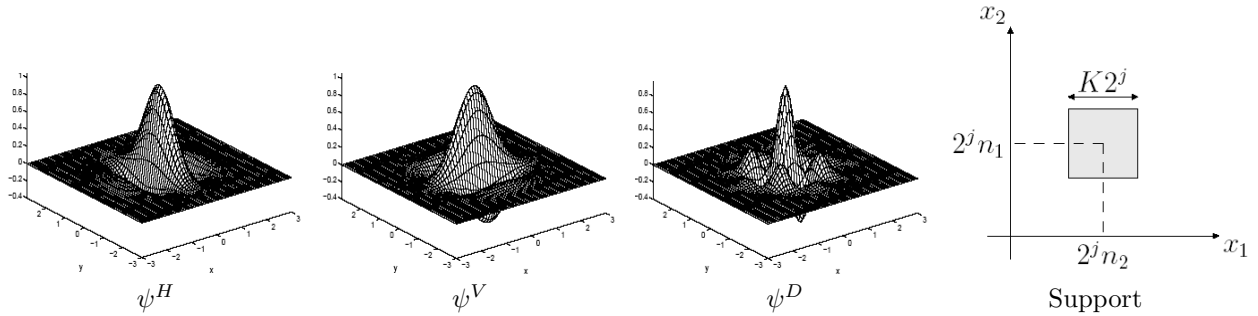


Figure 4.13: 2-D wavelets and their approximative support (right).

Haar 2-D multiresolution. For the Haar multiresolution, one obtains 2-D piecewise-constant Haar approximation. A function of $V_j \otimes V_j$ is constant on squares of size $2^j \times 2^j$. Figure 4.14 shows an example of projection of an image onto these 2-D Haar approximation spaces.



Figure 4.14: 2-D Haar approximation $P_{V_j^O} f$ for increasing j .

Discrete 2-D wavelet coefficients. Similarly to (4.4), we suppose that the sampling mechanism gives us access to inner product of the analog (continuous) signal f with the scaling function at scale $N = 2^{-J}$

$$\forall n, \in \{0, \dots, N-1\}^2, \quad a_{J,n} = \langle f, \varphi_{J,n}^P \rangle$$

Discrete wavelet coefficients are defined as

$$\forall \omega \in \{V, H, D\}, \forall J < j \leq 0, \forall 0 \leq n_1, n_2 < 2^{-j}, \quad d_{j,n}^\omega = \langle f, \psi_{j,n}^\omega \rangle.$$

(we use here periodized wavelets). Approximation coefficients are defined as

$$a_{j,n} = \langle f_0, \varphi_{j,n}^O \rangle.$$

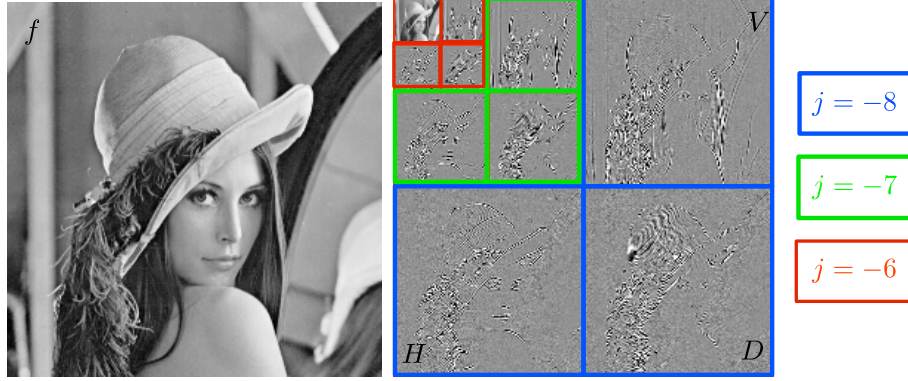


Figure 4.15: 2-D wavelet coefficients.

Figure 4.15 shows examples of wavelet coefficients, that are packed in an image of N pixels. Figure 4.16 shows other examples of wavelet decompositions.

Forward 2-D wavelet transform basic step. A basic step of the computation of the 2-D wavelet transform computes detail coefficients and a low pass residual from the fine scale coefficients

$$a_{j-1} \mapsto (a_j, d_j^H, d_j^V, d_j^D).$$

Similarly to the 1-D setting, this mapping is orthogonal, and is computed using the 1-D filtering and subsampling formula (4.7).

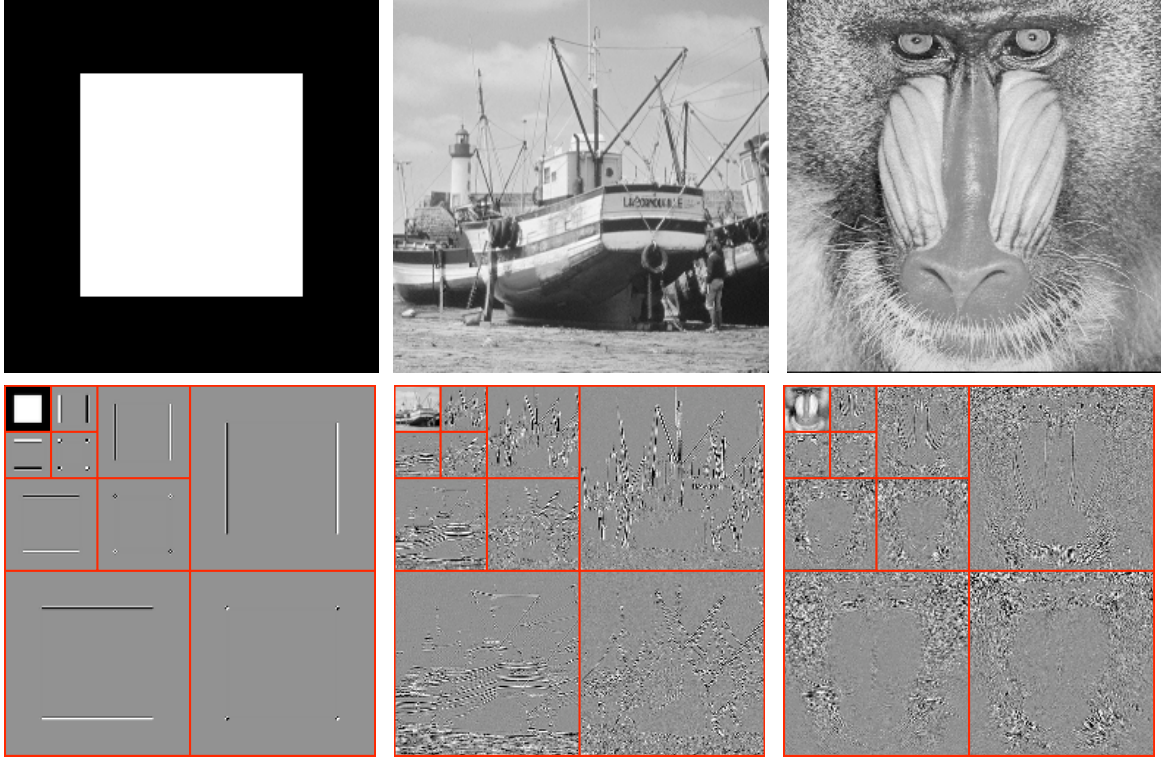


Figure 4.16: Examples of images (top row) and the corresponding wavelet coefficients (bottom row) .

One first applies 1-D horizontal filtering and sub-sampling

$$\begin{aligned}\tilde{a}_j &= (a_{j-1} \star^H \tilde{h}) \downarrow^H 2 \\ \tilde{d}_j &= (a_{j-1} \star^H \tilde{h}) \downarrow^H 2,\end{aligned}$$

where \star^H is the horizontal convolution, that applies the 1-D convolution to each column of a matrix

$$a \star^H b_{n_1, n_2} = \sum_{m_1=0}^{P-1} a_{n_1-m_1, n_2} b_{m_1}$$

where $a \in \mathbb{C}^{P \times P}$ and $b \in \mathbb{C}^P$ are matrix and vectors. The notation $\downarrow^H 2$ accounts for sub-sampling in the horizontal direction

$$(a \downarrow^H 2)_{n_1, n_2} = a_{2n_1, n_2}.$$

One then applies 1-D vertical filtering and sub-sampling to \tilde{a}_j and \tilde{d}_j to obtain

$$\begin{aligned}a_j &= (\tilde{a}_j \star^V \tilde{h}) \downarrow^V 2, & d_j^H &= (\tilde{d}_j \star^V \tilde{h}) \downarrow^V 2, \\ d_j^V &= (\tilde{a}_j \star^V \tilde{g}) \downarrow^V 2, & d_j^D &= (\tilde{d}_j \star^V \tilde{g}) \downarrow^V 2,\end{aligned}$$

where the vertical operators are defined similarly to horizontal operators but operating on rows.

These two forward steps are shown in block diagram in Figure 4.17. These steps can be applied in place, so that the coefficients are stored in an image of N pixels, as shown in Figure 4.18. This gives the traditional display of wavelet coefficients used in Figure 4.16.

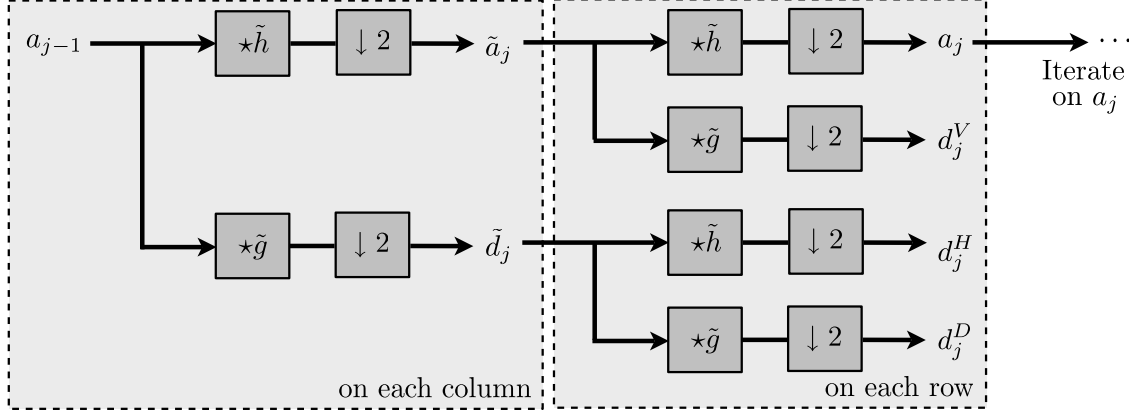


Figure 4.17: Forward 2-D filterbank step.



Figure 4.18: One step of the 2-D wavelet transform algorithm.

Fast 2-D wavelet transform. The 2-D FWT algorithm iterates these steps through the scales:

- **Input:** signal $f \in \mathbb{C}^N$.
- **Initialization:** $a_J = f$.
- **For** $j = J, \dots, j_0 - 1$.

$$\begin{aligned} \tilde{a}_j &= (a_{j-1} \star^H \tilde{h}) \downarrow^H 2, & d_j^V &= (\tilde{a}_j \star^V \tilde{g}) \downarrow^V 2, \\ \tilde{d}_j &= (a_{j-1} \star^H \tilde{h}) \downarrow^H 2, & d_j^H &= (\tilde{d}_j \star^V \tilde{h}) \downarrow^V 2, \\ a_j &= (\tilde{a}_j \star^V \tilde{h}) \downarrow^V 2, & d_j^D &= (\tilde{d}_j \star^V \tilde{g}) \downarrow^V 2. \end{aligned}$$

- **Output:** the coefficients $\{d_j^\omega\}_{j_0 \leq j < J, \omega} \cup \{a_{j_0}\}$.

Fast 2-D inverse wavelet transform. The inverse transform undo the horizontal and vertical filtering steps. The first step computes

$$\begin{aligned} \tilde{a}_j &= (a_j \star^V h) \uparrow^V 2 + (d_j^V \star^V g) \uparrow^V 2, \\ \tilde{d}_j &= (d_j^H \star^V h) \uparrow^V 2 + (d_j^D \star^V g) \uparrow^V 2, \end{aligned}$$

where the vertical up-sampling is

$$(a \uparrow^V 2)_{n_1, n_2} = \begin{cases} a_{k, n_2} & \text{if } n_1 = 2k, \\ 0 & \text{if } n_1 = 2k + 1. \end{cases}$$

The second inverse step computes

$$a_{j-1} = (\tilde{a}_j \star^H h) \uparrow^H 2 + (\tilde{d}_j \star^H g) \uparrow^H 2.$$

Figure 4.19 shows in block diagram this inverse filter banks, that is the inverse of the diagram 4.17.

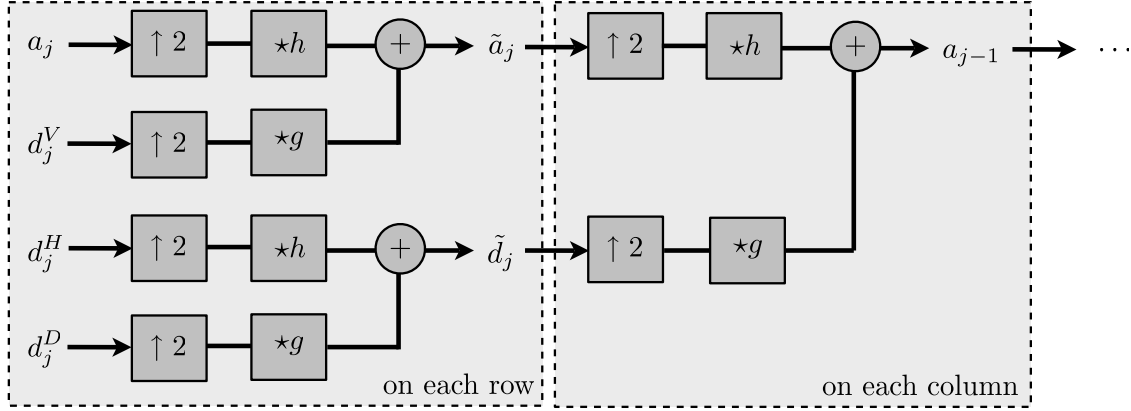


Figure 4.19: Backward 2-D filterbank step.

The inverse Fast wavelet transform iteratively applies these elementary steps

- **Input:** $\{d_j^\omega\}_{j_0 \leq j < J, \omega} \cup \{a_{j_0}\}$.
- **For** $j = j_0, \dots, J+1$.

$$\begin{aligned}\tilde{a}_j &= (a_j \star^V h) \uparrow^V 2 + (d_j^V \star^V g) \uparrow^V 2, \\ \tilde{d}_j &= (d_j^H \star^V h) \uparrow^V 2 + (d_j^D \star^V g) \uparrow^V 2, \\ a_{j-1} &= (\tilde{a}_j \star^H h) \uparrow^H 2 + (\tilde{d}_j \star^H g) \uparrow^H 2.\end{aligned}$$

- **Output:** $f = a_J$.

4.6 Wavelet Design

To be able to compute the wavelet coefficients using the FWT algorithm, it remains to know how to compute the scaling and wavelet functions. The FWT only makes use of the filters h and g , so instead of explicitly knowing the functions φ and ψ , one can only know these filters. Indeed, most of the known wavelets do not have explicit formula, and are implicitly defined through the cascade of the FWT algorithm.

This section shows what are the constraints h and g should satisfy, and gives practical examples. Furthermore, it shows that the knowledge of h determines g under the constraint of having quadrature filters, which is the most usual choice for wavelet analysis.

4.6.1 Low-pass Filter Constraints

We introduce the following three conditions on a filter h

$$\hat{h}(0) = \sqrt{2} \tag{C_1}$$

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2, \tag{C_2}$$

$$\inf_{\omega \in [-\pi/2, \pi/2]} |\hat{h}(\omega)| > 0. \tag{C^*}$$

Here we are using the Fourier series associated to a filter $h \in \mathbb{R}^{\mathbb{Z}}$

$$\forall \omega \in \mathbb{R}/2\pi\mathbb{Z}, \quad \hat{h}(\omega) \stackrel{\text{def.}}{=} \sum_{n \in \mathbb{Z}} h_n e^{-in\omega}.$$

If $h \in \ell^1(\mathbb{Z})$, this defines a continuous periodic function $\hat{h} \in C^0(\mathbb{R}/2\pi\mathbb{Z})$, and this definition can be extended to $h \in \ell^2(\mathbb{Z})$ and defines $\hat{h} \in L^2(\mathbb{R}/2\pi\mathbb{Z})$.

Theorem 23. *If φ defines multi-resolution approximation spaces, then (C_1) and (C_2) holds for h defined in (4.6). Conversely, if (C_1) , (C_2) and (C^*) holds, then there exists a φ defining multi-resolution approximation spaces so that associated filter is h as defined in (4.6).*

Proof. We only prove the first statement of the theorem. The converse statement is much more difficult to prove.

We now prove condition (C_1) . The refinement equation reads like a discrete-continuous convolution (or equivalently a convolution with a distribution)

$$\frac{1}{\sqrt{2}}\varphi\left(\frac{t}{2}\right) = \sum_{n \in \mathbb{Z}} h_n \varphi(t - n). \quad (4.12)$$

Denoting $h \star \varphi$ such a convolution, assuming $h \in \ell_1(\mathbb{Z})$ and $\varphi \in L^1(\mathbb{R})$, one check that one can apply Fubini and that $h \star \varphi \in L^1(\mathbb{R})$ and then

$$\begin{aligned} \mathcal{F}\left(\sum_{n \in \mathbb{Z}} h_n \varphi(t - n)\right)(\omega) &= \int_{\mathbb{R}} \sum_{n \in \mathbb{Z}} h_n \varphi(t - n) e^{-i\omega t} dt = \sum_{n \in \mathbb{Z}} h_n \int_{\mathbb{R}} \varphi(t - n) e^{-i\omega t} dt \\ &= \sum_{n \in \mathbb{Z}} h_n e^{-in\omega} \int_{\mathbb{R}} \varphi(x) e^{-i\omega x} dx = \hat{\varphi}(\omega) \hat{h}(\omega) \end{aligned}$$

where we made the change of variable $x = t - n$. Note that here, $h(\omega)$ is the 2π -periodic Fourier transform (i.e. Fourier series) of infinite filters defined in (4.6.1), whereas $\hat{\varphi}(\omega)$ is the Fourier transform of function. This is thus a product of a 2π -periodic function \hat{h} and a non-periodic function $\hat{\varphi}$. We recall that $\mathcal{F}(f(\cdot/s)) = s\hat{f}(s\cdot)$. Over the Fourier domain, equation (4.12) thus reads

$$\hat{\varphi}(2\omega) = \frac{1}{\sqrt{2}} \hat{h}(\omega) \hat{\varphi}(\omega). \quad (4.13)$$

One can show that $\hat{\varphi}(0) \neq 0$ (actually, $|\hat{\varphi}(0)| = 1$), so that this relation implies the first condition (C_1) .

We now prove condition (C_2) . The orthogonality of $\varphi(\cdot - n)\}_n$ is rewritten using a continuous convolution as (see also Proposition 9)

$$\forall n \in \mathbb{Z}, \quad \varphi \star \bar{\varphi}(n) = \delta_0$$

where $\bar{\varphi}(x) = \varphi(-x)$, and thus over the Fourier domain, using (4.13) which shows $\hat{\varphi}\omega = \frac{1}{\sqrt{2}}\hat{h}(\omega/2)\hat{\varphi}(\omega/2)$

$$1 = \sum_k |\hat{\varphi}(\omega + 2k\pi)|^2 = \frac{1}{2} \sum_k |\hat{h}(\omega/2 + k\pi)|^2 |\hat{\varphi}(\omega/2 + k\pi)|^2.$$

Since \hat{h} is 2π -periodic, one can split even and odd k and obtain

$$2 = |h(\omega/2)|^2 \sum_k |\hat{\varphi}(\omega/2 + 2k\pi)|^2 + |h(\omega/2 + \pi)|^2 \sum_k |\hat{\varphi}(\omega/2 + 2k\pi + \pi)|^2$$

This leads to condition (C_2) . Re-using the fact that $\sum_k |\hat{\varphi}(\omega + 2k\pi)|^2 = 1$ for $\omega' = \omega/2$ in place of ω , one thus has

$$|h(\omega')|^2 + |h(\omega' + \pi)|^2 = 2.$$

We do not prove the converse statement, which requires to “create” a function φ from the filter h . The intuition is that iterating (4.13) leads informally to

$$\varphi(\omega) = \prod_{k < 0} \frac{\hat{h}(\omega/2^k)}{\sqrt{2}}. \quad (4.14)$$

Condition (C*) can be shown to imply that this infinite product converge, and define a (non-periodic) function in $L^2(\mathbb{R})$. \square

Note that for the converse statement of this theorem to holds, condition (C*) imposes a control on the behavior of \hat{h} near 0.

4.6.2 High-pass Filter Constraints

We now introduce the following two conditions on a pair of filter (g, h)

$$|\hat{g}(\omega)|^2 + |\hat{g}(\omega + \pi)|^2 = 2 \quad (C_3)$$

$$\hat{g}(\omega)\hat{h}(\omega)^* + \hat{g}(\omega + \pi)\hat{h}(\omega + \pi)^* = 0. \quad (C_4)$$

Theorem 24. *If (φ, ψ) defines a multi-resolution analysis, then (C₃) and (C₄) holds for (h, g) defined in (4.6). Conversely, if (C₁) to (C₄) hold, then there exists a (φ, ψ) defining multi-resolution analysis so that associated filters are (h, g) as defined in (4.6). Furthermore,*

$$\hat{\psi}(\omega) = \frac{1}{\sqrt{2}}\hat{g}(\omega/2)\hat{\varphi}(\omega/2). \quad (4.15)$$

Proof. We prove condition (C₃). The refinement equation for the wavelet reads

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n \in \mathbb{Z}} g_n \varphi(t - n)$$

and thus over the Fourier domain

$$\hat{\psi}(2\omega) = \frac{1}{\sqrt{2}}\hat{g}(\omega)\hat{\varphi}(\omega). \quad (4.16)$$

The orthogonality of $\{\psi(\cdot - n)\}_n$ is re-written

$$\forall n \in \mathbb{Z}, \quad \psi \star \bar{\psi}(n) = \delta_0$$

and thus over the Fourier domain (using Poisson formula, see also Proposition 9)

$$\sum_k |\hat{\psi}(\omega + 2k\pi)|^2 = 1.$$

Using the Fourier domain refinement equation (4.16), similarly to the proof of Theorem 23 for (C₁), this is equivalent to condition (C₃). Figure 4.20 shows the Fourier transform of two filters that satisfy this complementary condition.

We now prove condition (C₄). The orthogonality between $\{\psi(\cdot - n)\}_n$ and $\{\varphi(\cdot - n)\}_n$ is written as

$$\forall n \in \mathbb{Z}, \quad \psi \star \bar{\varphi}(n) = 0$$

and hence over the Fourier domain (using Poisson formula, similarly to Proposition 9)

$$\sum_k \hat{\psi}(\omega + 2k\pi)\hat{\varphi}^*(\omega + 2k\pi) = 0.$$

Using the Fourier domain refinement equations (4.13) and (4.16), this is equivalent to condition (C₄). \square

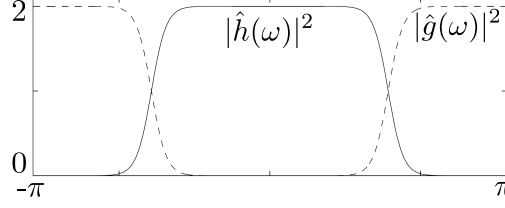


Figure 4.20: Complementarity between a low pass and a high pass wavelet filters h and g that satisfy condition (C_3) .

Quadrature mirror filters. Quadrature mirror filters (QMF) defines g as a function of h so that the conditions of Theorem 24 are automatically satisfy. This choice is the natural choice to build wavelet filters, and is implicitly assumed in most constructions (other choices leading to the same wavelet function anyway, since it satisfies (4.15)).

Proposition 12. For a filter $h \in \ell^2(\mathbb{Z})$ satisfying (C_1) , defining $g \in \ell^2(\mathbb{Z})$ as

$$\forall n \in \mathbb{Z}, \quad g_n = (-1)^{1-n} h_{1-n} \quad (4.17)$$

satisfies conditions (C_3) and (C_4) .

Proof. One indeed has that

$$\hat{g}(\omega) = e^{-i\omega} \hat{h}(\omega + \pi)^*, \quad (4.18)$$

so that

$$|\hat{g}(\omega)|^2 + |\hat{g}(\omega + \pi)|^2 = |e^{-i\omega} \hat{h}(\omega + \pi)^*|^2 + |e^{-i(\omega + \pi)} \hat{h}(\omega + 2\pi)^*|^2 = |\hat{h}(\omega + \pi)|^2 + |\hat{h}(\omega + 2\pi)|^2 = 2.$$

where we used the fact that \hat{h} is 2π -periodic, and also

$$\begin{aligned} \hat{g}(\omega) \hat{h}(\omega)^* + \hat{g}(\omega + \pi) \hat{h}(\omega + \pi)^* &= e^{-i\omega} \hat{h}(\omega + \pi)^* \hat{h}(\omega)^* + e^{-i(\omega + \pi)} \hat{h}(\omega + 2\pi)^* \hat{h}(\omega + \pi)^* \\ &= (e^{-i\omega} + e^{-i(\omega + \pi)}) \hat{h}(\omega + \pi)^* \hat{h}(\omega)^* = 0. \end{aligned}$$

□

4.6.3 Wavelet Design Constraints

According to the previous sections, the construction of a multi-resolution analysis (i.e. of functions (φ, ψ)) is obtained by designing a filter h satisfying conditions (C_1) and (C_2) . The function φ is obtained by an infinite cascade of filtering, or equivalently in the Fourier domain by (4.14), there is in general (put aside special case such as the Haar multiresolution) no closed form expression for φ . Once φ is defined, ψ is automatically defined by the relation (4.15) (and g can be defined as (4.18)).

There exists only one Fourier transform, but there is a large choice of different mother wavelet functions ψ . They are characterized by

- Size of the support.
- Number of oscillations (the so called number p of vanishing moments).
- Symmetry (only possible for non-orthogonal bases).
- Smoothness (number of derivatives).

We now detail how these constraints are integrated together with conditions (C_1) - (C_4) .

Vanishing moments. A wavelet ψ has p vanishing moments if

$$\forall k \leq p-1, \quad \int_{\mathbb{R}} \psi(x) x^k dx = 0. \quad (4.19)$$

This ensures that $\langle f, \psi_{j,n} \rangle$ is small if f is C^α , $\alpha < p$ on $\text{Supp}(\psi_{j,n})$.

This condition can be equivalently expressed over Fourier as followed.

Proposition 13. *Assuming enough regularity of ψ , and using the QMF construction (4.18), it has p vanishing moments if and only if*

$$\forall k \leq p-1, \quad \frac{d^k \hat{h}}{d\omega^k}(\pi) = \frac{d^k \hat{g}}{d\omega^k}(0) = 0. \quad (4.20)$$

Proof. Since ψ is regular, one has that $\hat{\psi}^{(k)} = \mathcal{F}((-i \cdot)^k \psi(\cdot))$, so that

$$(-i)^k \int_{\mathbb{R}} x^k \psi(x) dx = \hat{\psi}^{(k)}(0).$$

Relation (4.15) and (4.18) implies

$$\hat{\psi}(2\omega) = \hat{h}(\omega + \pi)^* \rho(\omega) \quad \text{where} \quad \rho(\omega) \stackrel{\text{def.}}{=} \frac{1}{\sqrt{2}} e^{-i\omega} \hat{\varphi}(\omega).$$

and differentiating this relation shows

$$2\hat{\psi}^{(1)}(2\omega) = \hat{h}(\omega + \pi)^* \rho^{(1)}(\omega) + \hat{h}^{(1)}(\omega + \pi)^* \rho(\omega)$$

which shows that, since $\rho(0) = \hat{\varphi}(0) \neq 0$, $\psi^{(1)}(0) = 0 \Leftrightarrow \hat{h}^{(1)}(\pi)^* = 0$. Recursing this argument and iterating the derivative, one obtains that $\psi^{(k)}(0) = 0 \Leftrightarrow \hat{h}^{(k)}(\pi)^* = 0$ (assuming this hold for previous derivatives). \square

Note that conditions (C_1) and (C_1) implies that $\hat{h}(\pi) = 0$, so that an admissible wavelet necessarily has 1 vanishing moment, i.e. $\int \psi = 0$. Condition (4.20) shows that having more vanishing moment is equivalent to having a Fourier transform \hat{h} which is “flatter” around $\omega = \pi$.

Support. Figure 4.21 shows the wavelet coefficients of a piecewise smooth signal. Coefficients of large magnitude are clustered near the singularities, because the wavelet ψ has enough vanishing moments.

To avoid that many wavelets create large coefficients near singularities, one should choose ψ with a small support. One can show that the size of the support of ψ is proportional to the size of the support of h . This requirement is however contradictory with the vanishing moment property (4.19). Indeed, one can prove that for an orthogonal wavelet basis with p vanishing moments

$$|\text{Supp}(\psi)| \geq 2p - 1,$$

where $\text{sup}(a)$ is the largest closed interval outside of which the function f is zero.

Chapter 6 studies in details the tradeoff of support size and vanishing moment to perform non-linear approximation of piecewise smooth signals.

Smoothness. In compression or denoising applications, an approximate signals is recovered from a partial set I_M of coefficients,

$$f_M = \sum_{(j,n) \in I_M} \langle f, \psi_{j,n} \rangle \psi_{j,n}.$$

This approximation f_M has the same smoothness as ψ .

To avoid visually unpleasant artifacts, one should thus choose a smooth wavelet function ψ . This is only for cosmetic reasons, since increasing smoothness does not leads to a better approximation. However, for most wavelet family, increasing the number of vanishing moments also increases the smoothness of the wavelets. This is for instance the case of the Daubechies family exposed in the next section.

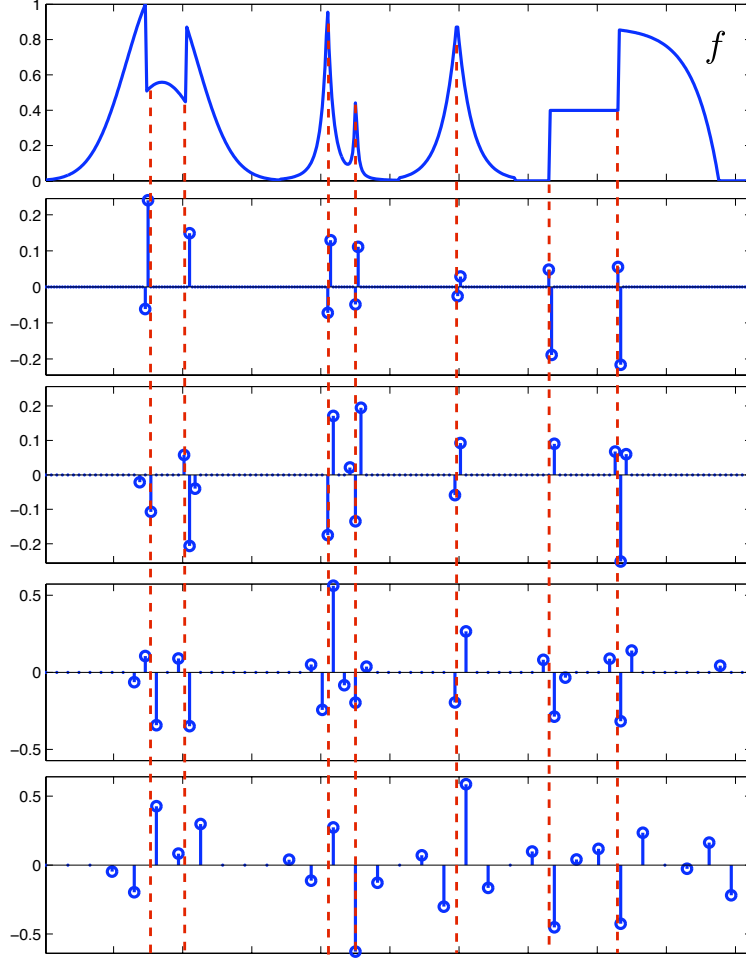


Figure 4.21: Location of large wavelet coefficients.

4.6.4 Daubechies Wavelets

To build a wavelet ψ with a fixed number p of vanishing moments, one designs the filter h , and use the quadrature mirror filter relation (4.18) to compute g . One thus look for h such that

$$|\hat{h}(\omega)|^2 + |\hat{h}(\omega + \pi)|^2 = 2, \quad \hat{h}(0) = \sqrt{2}, \quad \text{and} \quad \forall k < p, \quad \frac{d^k \hat{h}}{d\omega^k}(\pi) = 0.$$

This corresponds to algebraic relationships between the coefficients of h , and it turns out that they can be solved explicitly using the Euclidean division algorithm for polynomials.

This leads to Daubechies wavelets with p vanishing moments, which are orthogonal wavelets with a minimum support length of $2p - 1$.

For $p = 1$, it leads to the Haar wavelet, with

$$h = [h_0 = 0.7071; 0.7071].$$

For $p = 2$, one obtains the celebrated Daubechies 4 filter

$$h = [0.4830; h_0 = 0.8365; 0.2241; -0.1294],$$

and for $p = 3$,

$$h = [0; 0.3327; 0.8069; h_0 = 0.4599; -0.1350; -0.0854; 0.0352].$$

Wavelet display. Figure 4.22 shows examples of Daubechies mother wavelet functions with an increasing number of vanishing moments. These displays are obtained by computing in fact a discrete wavelet $\bar{\psi}_{j,n}$ defined in (??) for a very large number of samples N . This discrete wavelet is computed by applying the inverse wavelet transform to the coefficients $d_{j',n'} = \delta_{j-j'}\delta_{n-n'}$.

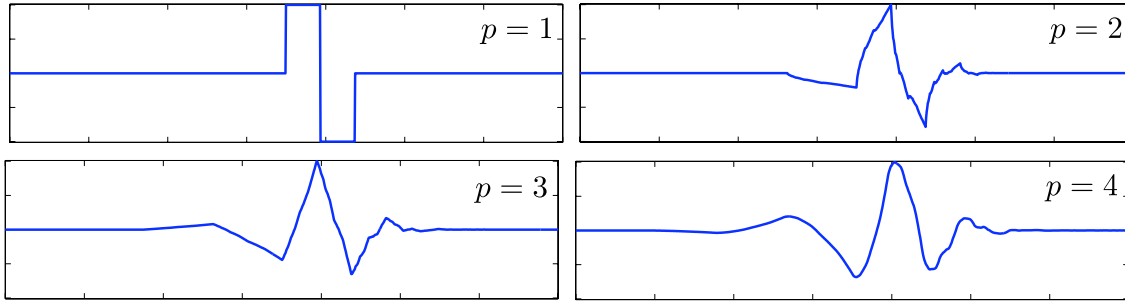


Figure 4.22: Examples of Daubechies mother wavelets ψ with an increasing number p of vanishing moments.

Bibliography

- [1] P. Alliez and C. Gotsman. Recent advances in compression of 3d meshes. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in multiresolution for geometric modelling*, pages 3–26. Springer Verlag, 2005.
- [2] P. Alliez, G. Ucelli, C. Gotsman, and M. Attene. Recent advances in remeshing of surfaces. In *AIM@SHAPE report*. 2005.
- [3] Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, 2014.
- [4] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.
- [5] Stephen Boyd and Lieven Vandenbergh. *Convex optimization*. Cambridge university press, 2004.
- [6] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004.
- [7] E. J. Candès. The restricted isometry property and its implications for compressed sensing. *Compte Rendus de l’Académie des Sciences, Serie I*(346):589–592, 2006.
- [8] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Modeling and Simulation*, 5:861–899, 2005.
- [9] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20:89–97, 2004.
- [10] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.
- [11] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.
- [12] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.
- [13] F. R. K. Chung. Spectral graph theory. *Regional Conference Series in Mathematics, American Mathematical Society*, 92:1–212, 1997.
- [14] Philippe G Ciarlet. Introduction à l’analyse numérique matricielle et à l’optimisation. 1982.
- [15] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, 4(4), 2005.

- [16] P. Schroeder et al. D. Zorin. Subdivision surfaces in character animation. In *Course notes at SIGGRAPH 2000*, July 2000.
- [17] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure and Appl. Math.*, 57:1413–1541, 2004.
- [18] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *J. Fourier Anal. Appl.*, 4(3):245–267, 1998.
- [19] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, Dec 1994.
- [20] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.
- [21] M. Figueiredo and R. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Proc.*, 12(8):906–916, 2003.
- [22] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In N. A. Dodgson, M. S. Floater, and M. A. Sabin, editors, *Advances in multiresolution for geometric modelling*, pages 157–186. Springer Verlag, 2005.
- [23] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.
- [24] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In Alyn Rockwood, editor, *Proceedings of the Conference on Computer Graphics (Siggraph99)*, pages 325–334. ACM Press, August8–13 1999.
- [25] A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In *Proceedings of the Computer Graphics Conference 2000 (SIGGRAPH-00)*, pages 271–278, New York, July 23–28 2000. ACM Press.
- [26] L. Kobbelt. $\sqrt{3}$ subdivision. In Sheila Hoffmeyer, editor, *Proc. of SIGGRAPH’00*, pages 103–112, New York, July 23–28 2000. ACM Press.
- [27] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997.
- [28] S. Mallat. *A Wavelet Tour of Signal Processing, 3rd edition*. Academic Press, San Diego, 2009.
- [29] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.
- [30] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. on Pure and Appl. Math.*, 42:577–685, 1989.
- [31] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.
- [32] Gabriel Peyré. *L’algèbre discrète de la transformée de Fourier*. Ellipses, 2004.
- [33] Gabriel Peyré and Marco Cuturi. Computational optimal transport. 2017.
- [34] J. Portilla, V. Strela, M.J. Wainwright, and Simoncelli E.P. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, 12(11):1338–1351, November 2003.
- [35] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics*, 22(3):340–349, July 2003.

- [36] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.
- [37] Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkhäuser, NY*, 2015.
- [38] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, Frank Lenzen, and L Sirovich. *Variational methods in imaging*. Springer, 2009.
- [39] P. Schröder and W. Sweldens. Spherical Wavelets: Efficiently Representing Functions on the Sphere. In *Proc. of SIGGRAPH 95*, pages 161–172, 1995.
- [40] P. Schröder and W. Sweldens. Spherical wavelets: Texture processing. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95*. Springer Verlag, Wien, New York, August 1995.
- [41] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
- [42] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.
- [43] Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse image and signal processing: Wavelets and related geometric multiscale analysis*. Cambridge university press, 2015.
- [44] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computation Harmonic Analysis*, 3(2):186–200, 1996.
- [45] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.