Numerical Issues When Using Wavelets

Jean-Luc Starck, Jalal M. Fadili

▶ **To cite this version:**

Jean-Luc Starck, Jalal M. Fadili. Numerical Issues When Using Wavelets. Encyclopedia of Complexity and System Science, 2008. hal-00256365

HAL Id: hal-00256365
https://hal.science/hal-00256365

Submitted on 15 Feb 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Numerical Issues When Using Wavelets

J.-L. Starck [*], J. Fadili [†]

May 23, 2007

Contents

1 Definition of the Subject and its Importance **2**

2 Introduction **2**

3 The Continuous Wavelet Transform **2**

4 The (Bi-)Orthogonal Wavelet Transform **5**

5 The Lifting Scheme **8**

6 The Undecimated Wavelet Transform **12**

7 The 2D Isotropic Undecimated Wavelet Transform **14**

8 Designing non-Orthogonal Filter Banks **18**

9 Iterative Reconstruction **20**

10 Future Directions **21**

Notations

For a real discrete-time filter whose impulse response is $h[n]$, $\bar{h}[n] = h[-n],\ n \in \mathbb{Z}$ is its time-reversed version. The hat $\hat{}$ notation will be used for the Fourier transform of square-integrable signals. For a filter h, its z-transform is written $H(z)$. The convolution product of two signals in $\ell^2(\mathbb{Z})$ will be written $*$. For the octave band wavelet representation, analysis (respectively, synthesis) filters are denoted h and g (respectively, $\tilde{h}$ and $\tilde{g}$). The scaling and wavelet functions used for the analysis (respectively, synthesis) are denoted ϕ ($\phi(\frac{x}{2}) = \sum_k h[k]\phi(x-k), x \in \mathbb{R}$ *and* $k \in \mathbb{Z}$) and ψ ($\psi(\frac{x}{2}) = \sum_k g[k]\phi(x-k), x \in \mathbb{R}$ *and* $k \in \mathbb{Z}$) (respectively, $\tilde{\phi}$ and $\tilde{\psi}$). We also define the scaled dilated and translated version of ϕ at scale j and position k as $\phi_{j,k}(x) = 2^{-j}\phi(2^{-j}x - k)$, and similarly for ψ, $\tilde{\phi}$ and $\tilde{\psi}$.

[*]J.-L. Starck is with the CEA-Saclay, DAPNIA/SEDI-SAP, Service d'Astrophysique, F-91191 Gif sur Yvette, France

[†]J. Fadili is with the GREYC CNRS UMR 6072, Image Processing Group, ENSICAEN 14050, Caen Cedex, France

Glossary

WT	Wavelet Trasnform
CWT	Continuous Wavelet Transform
DWT	Discrete (decimated) Wavelet Transform
UWT	Undecimated Wavelet Transform
IUWT	Isotropic Undecimated Wavelet Transform

1 Definition of the Subject and its Importance

Wavelets and related multiscale representations pervade all areas of signal processing. The recent inclusion of wavelet algorithms in JPEG 2000 – the new still-picture compression standard– testifies to this lasting and significant impact. The reason of the success of the wavelets is due to the fact that wavelet basis represents well a large class of signals, and therefore allows us to detect roughly isotropic elements occurring at all spatial scales and locations. As the noise in the physical sciences is often not Gaussian, the modeling, in the wavelet space, of many kind of noise (Poisson noise, combination of Gaussian and Poisson noise, long-memory $1/f$ noise, non-stationary noise, ...) has also been a key step for the use of wavelets in scientific, medical, or industrial applications [1]. Extensive wavelet packages exist now, commercial (see for example [2]) or non commercial (see for example [3, 4]), which allows any researcher, doctor, or engineer to analyze his data using wavelets.

2 Introduction

Over the last two decades there has been abundant interest in wavelet methods. In many hundreds of papers published in journals throughout the scientific and engineering disciplines, a wide range of wavelet-based tools and ideas have been proposed and studied. Background texts on the wavelet transform include [5, 6, 7, 1, 8]. The most widely used wavelet transform (WT) algorithm is certainly the decimated bi-orthogonal wavelet transform (DWT) which is used in JPEG2000. While the bi-orthogonal wavelet transform has led to successful implementation in image compression, results were far from optimal for other applications such as filtering, deconvolution, detection, or more generally, analysis of data. This is mainly due to the loss of the translation-invariance property in the DWT, leading to a large number of artifacts when an image is reconstructed after modification of its wavelet coefficients. Later efforts found that substantial improvements in perceptual quality could be obtained by translation invariant methods based on thresholding of an undecimated wavelet transform.

3 The Continuous Wavelet Transform

The continuous wavelet transform uses a single function $\psi(x)$ and all its dilated and shifted version to analyze signals. The Morlet-Grossmann definition [9] of the continuous wavelet transform (CWT) for a 1-dimensional real-valued function[1] $f(x) \in L^2(\mathbb{R})$, the space of all square-integrable functions,

[1] We only consider here real wavelets. This can be extended to complex wavelets without too much difficulty.

is:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x)\psi\left(\frac{x-b}{a}\right) dx \tag{1}$$

where:

- $W(a,b)$ is the wavelet coefficient of the function $f(x)$,
- $\psi(x)$ is the analyzing wavelet,
- a (> 0) is the scale parameter,
- b is the position parameter.

The inverse transform is obtained by:

$$f(x) = \frac{1}{C_\psi} \int_0^{+\infty} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{a}} W(a,b)\psi\left(\frac{x-b}{a}\right) \frac{da\ db}{a^2} \tag{2}$$

where:

$$C_\psi = \int_0^{+\infty} \frac{|\hat{\psi}|^2}{\nu} d\nu = \int_{-\infty}^{0} \frac{|\hat{\psi}|^2}{\nu} d\nu \tag{3}$$

Reconstruction is only possible if C_ψ is finite (admissibility condition) which implies that $\hat{\psi}(0) = 0$, i.e. the mean of the wavelet function is 0. The wavelet is said to have a zero moment property, and appears to have a band-pass profile. A closely related relation to the inverse given in Eq. 2, is an energy conservation formula, an analogue to Plancherel's formula [7].

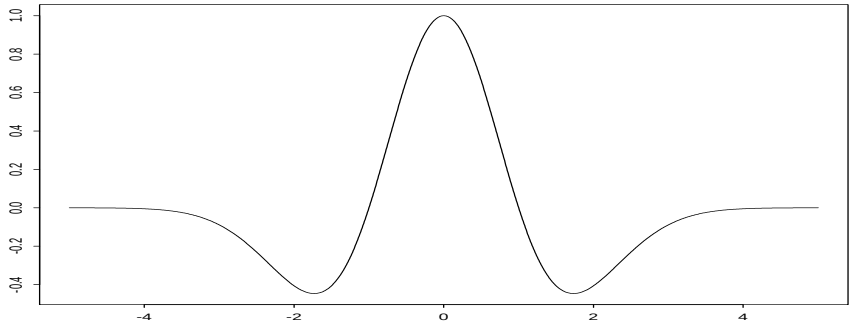

Figure 1: Mexican hat function.

Fig. 1 shows the Mexican hat wavelet function, which is defined by:

$$\psi(x) = (1 - x^2)e^{-x^2/2} \tag{4}$$

This is the second derivative of a Gaussian. The lower-part of Fig. 2 shows the CWT of a 1D signal (top plot of Fig. 2) computed with the Mexican Hat wavelet. This diagram is called a *s*calogram. Its y-axis represents the scale, and its x-axis represents the position parameter b.

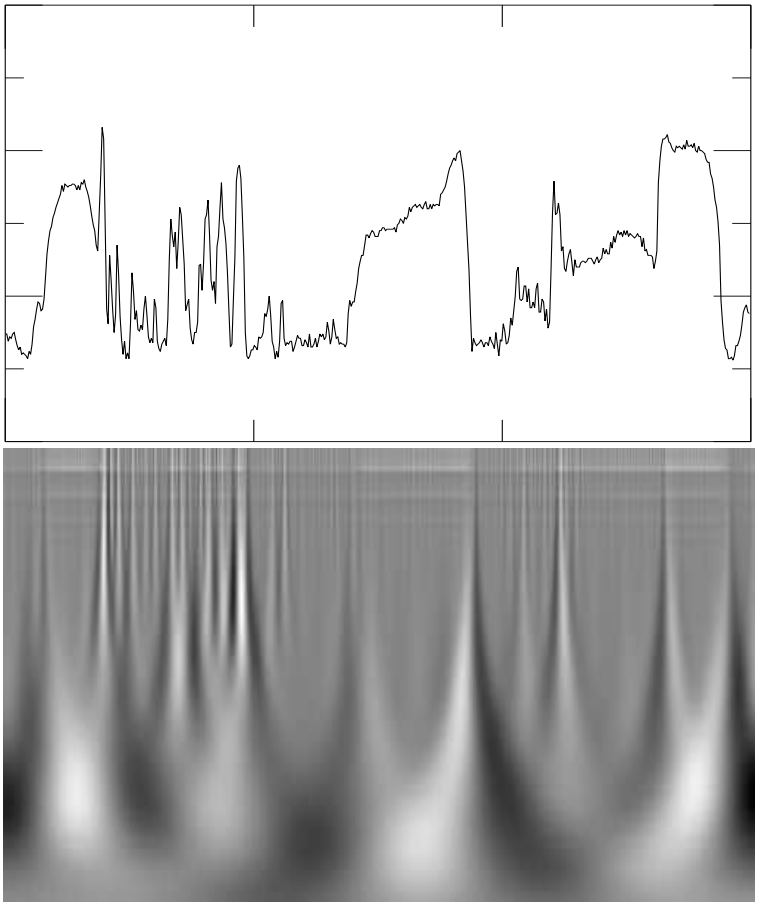

Figure 2: Top: 1D signal. Bottom: CWT computed with the Mexican Hat wavelet, the y-axis represents the scale and the x-axis represents the position parameter b.

In practice we need to discretize the scale space, and the CWT is computed for scales between a_{min} and a_{max} with a step δ_a. a_{min} must be chosen enough large to discretize properly the wavelet function, and a_{max} is limited by the number N of samples in the data. For the experiment shown in Fig. 2, a_{min} was set to 0.66 and since the dilated Mexican hat wavelet at scale a is approximately supported in $[-4a, 4a]$, we choose $a_{max} = \frac{N}{8}$. The number of scales J is defined as the number of voices per octave multiplied by the number of octaves (the number of octaves is the integral part of $\log_2\left(\frac{a_{max}}{a_{min}}\right)$. The number of voices per octave is generally chosen equal to 12, which guaranties a good resolution in scale and the possibility to reconstruct the signal from its wavelet coefficients. We then have $J = 12\log_2\left(\frac{a_{max}}{a_{min}}\right)$, and $\delta_a = \frac{a_{max}-a_{min}}{J-1}$.

The CWT algorithm is the following:

If the convolution is performed in the Fourier space (i.e. $\psi_a * D = \text{IFFT}(\text{FFT}(\psi_a)\text{FFT}(D))$, where FFT and IFFT denote respectively the Fourier transform and its inverse), the data is assumed to be periodic. In this case, the computation of the CWT requires $O\left(12N\left(\log_2 N\right)^2\right)$ operations [7]. If the convolution is done in the direct space, we can choose other ways to deal with the

1: Set the values a_{min}, a_{max}, J. These values depend on both the chosen wavelet function ψ and the number of samples N.
2: Set $\delta_a = \frac{a_{max}-a_{min}}{J-1}$ and $a = a_{min}$.
3: for $a = a_{min}$ to a_{max} with step δ_a do

- Compute $\psi_a = \psi(\frac{x}{a})/\sqrt{a}$.
- Convolve the input data D with $\bar{\psi_a}$ to get $W(a,.) = \frac{1}{\sqrt{(a)}}(\bar{\psi_a} * D)$. The convolution product can be done either in the direct space or in the Fourier space.
- $a = a + \delta_a$

4: W contains the CWT of D.

borders. For instance, we may prefer to consider mirror reflexive boundary conditions (i.e. for $k = 0, ..., N-1$ we have $D(-k) = D(k)$ and $D(N+k) = D(N-1-k)$).

The choice of the wavelet function is let to the user. As described above, the only constraint is to have a function with a zero mean (admissibility condition). Hence, a large class of functions verifies it and we can adapt the analyzing tool, i.e. the wavelet, to the data. For oscillating data such as audio signals or seismic data, we will prefer a wavelet function which oscillates like the Morlet wavelet. While for other kind of data such as spectra, it is better to choose a wavelet function with minimum oscillation and the Mexican hat would certainly be a good choice. The wavelet function can also be complex, in which case the wavelet transform will be complex. Both the modulus and the phase will carry information about the data.

Here, we have considered only 1D data. For higher dimensional data, we can apply exactly the same approach as above. For 2D data for example, the wavelet function will be defined as a function of five parameters (position (b_x, b_y), scale in the two directions (a_x, a_y) and orientation θ) and the wavelet transform of an image will be of dimension five. But The required memory and the computation time would not be acceptable in most applications. Considering an isotropic wavelet reduces significantly to only three the dimensionality. A even more efficient approach is the (bi-)orthogonal wavelet transform algorithm.

4 The (Bi-)Orthogonal Wavelet Transform

Many discrete wavelet transform algorithms have been developed [7, 1]. The most widely-known one is certainly the orthogonal transform, proposed by Mallat []wave:mallat89 and its bi-orthogonal version [5]. Here, we will introduce the bi-orthogonal through the two-channel iterated filter bank framework.

Using the bi-orthogonal wavelet transform, a signal s can be decomposed as follows:

$$s(x) = \sum_k c_J[k]\tilde{\phi}_{J,l}(x) + \sum_{j=1}^{J}\sum_k \tilde{\psi}_{j,k}(x)w_j[k] \tag{5}$$

with $\phi_{j,l}$ and $\psi_{j,l}$ are the scaled dilated and translated version of ϕ and ψ, which are are respectively the scaling function and the wavelet function. J is the number of resolution levels used in the

decomposition, w_j the wavelet (or detail) coefficients at scale j, and c_J is a coarse or smooth version of the original signal s. Thus, the algorithm outputs $J+1$ subband arrays. The indexing is such that, here, $j=1$ corresponds to the finest scale (high frequencies). Coefficients $c_j[k]$ and $w_j[k]$ are obtained by means of the analysis filters h and g:

$$\begin{aligned} c_{j+1}[l] &= \sum_k h[k-2l]c_j[k] \\ w_{j+1}[l] &= \sum_k g[k-2l]c_j[k] \end{aligned} \tag{6}$$

where h and g are such that:

$$\begin{aligned} \frac{1}{2}\phi(\frac{x}{2}) &= \sum_k h[k]\phi(x-k) \\ \frac{1}{2}\psi(\frac{x}{2}) &= \sum_k g[k]\phi(x-k) \end{aligned} \tag{7}$$

and the reconstruction of the signal is performed with:

$$c_j[l] = 2\sum_k \left(\tilde{h}[k+2l]c_{j+1}[k] + \tilde{g}[k+2l]w_{j+1}[k]\right) \tag{8}$$

where the filters $\tilde{h}$ and $\tilde{g}$ must verify the conditions of dealiasing and exact reconstruction:

$$\begin{aligned} \hat{h}^*\left(\nu+\frac{1}{2}\right)\hat{\tilde{h}}(\nu) + \hat{g}^*\left(\nu+\frac{1}{2}\right)\hat{\tilde{g}}(\nu) &= 0 \\ \hat{h}^*(\nu)\hat{\tilde{h}}(\nu) + \hat{g}^*(\nu)\hat{\tilde{g}}(\nu) &= 1 \end{aligned} \tag{9}$$

or equivalently, in the z-transform domain:

$$\begin{aligned} H(-z^{-1})\tilde{H}(z) + G(-z^{-1})\tilde{G}(z) &= 0 \\ H(z^{-1})\tilde{H}(z) + G(z^{-1})\tilde{G}(z) &= 1. \end{aligned}$$

Note that in terms of filter banks, the bi-orthogonal wavelet transform becomes orthogonal when $h=\tilde{h}$ and $g=\tilde{g}$, in which case h is a conjugate mirror filter.

In the decomposition, c_{j+1} and w_{j+1} are computed by successively convolving a_j with the filters $\bar{h}$ (low-pass) and g (high-pass). Each resulting channel is decimated by suppression of one sample out of two. The high-frequency channel w_{j+1} is left, and we iterate with the low-frequency part c_{j+1} (upper part of Fig. 3). In the reconstruction, we restore the sampling by inserting a 0 between each sample, then we convolve with the dual filters $\tilde{h}$ and $\tilde{g}$, we add the resulting coefficients and we multiply the result by 2. The procedure is iterated up to the smallest scale (lower part of Fig. 3).

Compared to the CWT, we have much less scales, because we consider only *dyadic scales*, i.e. scales a_j which are a power of two of the initial scale a_0 ($a_j = 2^j a_0$). Therefore, for a data set with N samples, we will typically use $J = \log(N) - 1$ scales. The algorithm is the following:

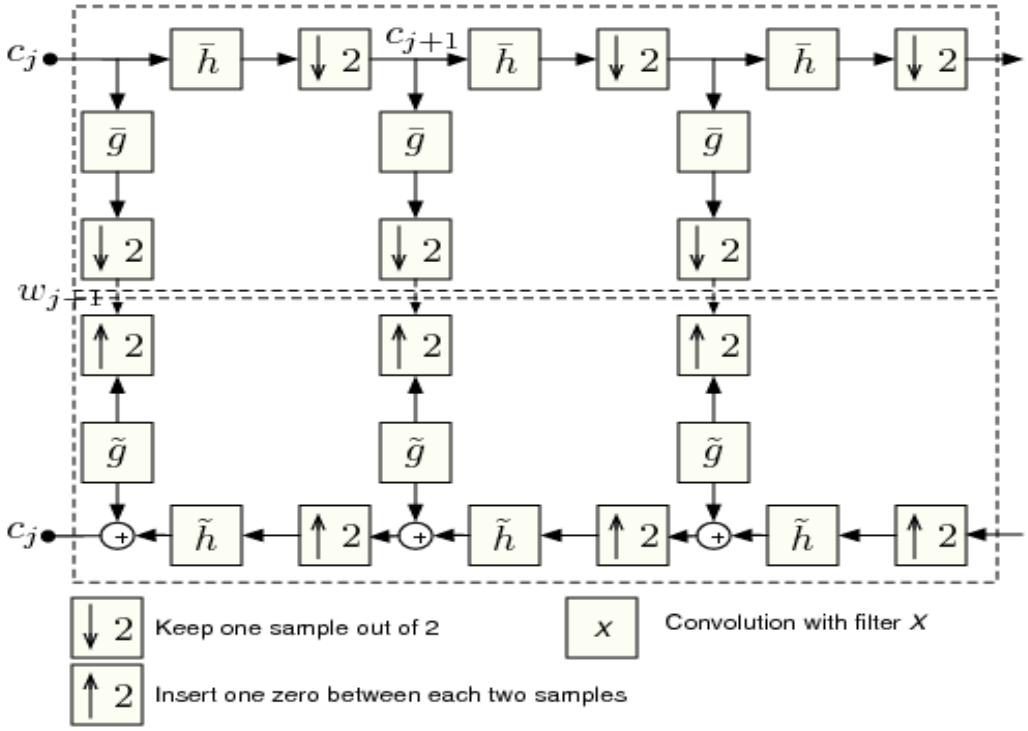

Figure 3: Fast pyramidal algorithm associated to the bi-orthogonal wavelet transform. Top: Fast analysis transform with a cascade of filtering with $\bar{h}$ and $\bar{g}$ followed by factor 2 subsampling. Bottom: Fast inverse transform by progressively inserting zeros and filtering with dual filters $\tilde{h}$ and $\tilde{g}$.

1: Set $c_0 = D$, $J = \log(N) - 1$.
2: for $j = 0$ to $J - 1$ do

- Compute $c_{j+1} = \bar{h} * c_j$, down-sample by a factor 2.
- Compute $w_{j+1} = \bar{g} * c_j$, down-sample by a factor 2.
- $j = j + 1$

3: The set $\mathcal{W} = \{w_1, ..., w_J, c_J\}$ represents the wavelet transform of the data.

The discrete bi-orthogonal wavelet transform (DWT) is also computationally very efficient,requiring $O(N)$ operations as compared to $O(N \log N)$ of the fast Fourier transform (N is the number of samples in data). The most used filters are certainly the 9/7 filters (by default in the JPEG 2000 norm), which are given in table 1.

In the literature, the filter bank can be given such that it is normalized to a unit mass $\sum_k h[k] = 1$, or to a unit ℓ_2-norm $\sum_k h[k]^2 = 1$.

The above DWT algorithm can be easily extended to any dimension by *separable* (tensor)

h	g	$\tilde{h}$	$\tilde{g}$
0	0.02674875741	0.02674875741	0
-0.04563588155	0.0168641184	-0.0168641184	0.04563588155
-0.02877176311	-0.0782232665	-0.0782232665	-0.02877176311
0.295635881557	-0.26686411844	0.26686411844	-0.295635881557
0.557543526229	0.60294901823	0.60294901823	0.557543526229
0.295635881557	-0.26686411844	0.26686411844	-0.295635881557
-0.02877176311	-0.0782232665	-0.0782232665	-0.02877176311
-0.04563588155	0.0168641184	-0.0168641184	0.04563588155
0	0.02674875741	0.02674875741	0

Table 1: 7/9 Filter bank (normalized to a unit mass).

products of a scaling function ϕ and a wavelet ψ. For instance, the two-dimensional algorithm is based on separate variables leading to prioritizing of horizontal, vertical and diagonal directions. The scaling function is defined by $\phi(x, y) = \phi(x)\phi(y)$, and the passage from one resolution to the next is achieved by:

$$c_{j+1}[k,l] = \sum_{m=-\infty}^{+\infty} \sum_{n=-\infty}^{+\infty} h[m-2k]h[n-2l]c_j[m,n] = \left(\bar{h}\bar{h} * c_j\right)[k,l] \tag{10}$$

The detail signal is obtained from three wavelets:

- vertical wavelet : $\psi^1(x, y) = \phi(x)\psi(y)$
- horizontal wavelet: $\psi^2(x, y) = \psi(x)\phi(y)$
- diagonal wavelet: $\psi^3(x, y) = \psi(x)\psi(y)$

which leads to three wavelet subimages at each resolution level. For three dimensional data, seven wavelet subcubes are created at each resolution level, corresponding to an analysis in seven directions.

For a $N \times N$ image D, the algorithm is the following:

5 The Lifting Scheme

A lifting is an elementary modification of perfect reconstruction filters, which is used to improve the wavelet properties. The lifting scheme [10] is a flexible technique that has been used in several different settings, for easy construction and implementation of traditional wavelets [10], and for the construction of wavelets on arbitrary domains such as bounded regions of $\mathbb{R}^d$ (second generation wavelets [11]) or surfaces (spherical wavelets [12]). To optimize the approximation and compression of signals and images, the lifting scheme has also been widely used to construct adaptive wavelet bases with signal-dependent liftings. For example, short wavelets are needed in the neighborhood of singularities, but long wavelets with more vanishing moments improve the approximation of regular regions.

1: Set $c_0 = D$, $J = \log(N) - 1$.
2: for $j = 0$ to $J - 1$ do

- Compute $c_{j+1} = \bar{h}\bar{h} * c_j$, suppress one sample out of two in each dimension.
- Compute $w^1_{j+1} = \bar{g}\bar{h} * c_j$, suppress one sample out of two in each dimension.
- Compute $w^2_{j+1} = \bar{h}\bar{g} * c_j$, suppress one sample out of two in each dimension.
- Compute $w^3_{j+1} = \bar{g}\bar{g} * c_j$, suppress one sample out of two in each dimension.
- $j = j + 1$

3: The set $\mathcal{W} = \{w^1_1, w^2_1, w^3_1, ..., w^1_J, w^2_J, w^3_J, c_J\}$ represents the wavelet transform of the data.

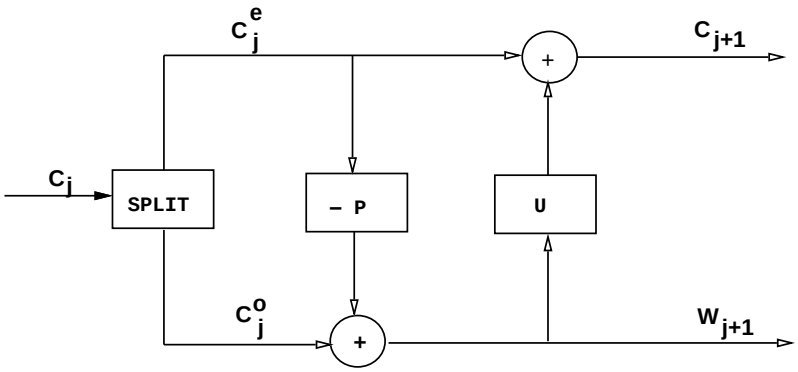

Figure 4: The lifting scheme – forward direction.

Its principle is to compute the difference between a true coefficient and its prediction:

$$w_{j+1}[l] = c_j[2l+1] - \mathcal{P}(c_j[2l-2L], ..., c_j[2l-2], c_j[2l], c_j[2l+2], ..., c_j[2l+2L]) \tag{11}$$

A pixel at an odd location $2l + 1$ is then predicted using pixels at even locations.

Computing the wavelet transform using lifting scheme consists of several stages. The idea is to first compute a trivial wavelet transform (the Lazy wavelet) and then improve its properties using alternating lifting and dual lifting steps. The transformation is done in three steps:

1. **Split:** This corresponds to Lazy wavelets which splits the signal into even and odd indexed samples:

$$\begin{aligned} c^e_j[l] &= c_j[2l] \\ c^o_j[l] &= c_j[2l+1] \end{aligned} \tag{12}$$

2. **Predict:** Calculate the wavelet coefficient $w_{j+1}[l]$ as the prediction error of $c^o_j[l]$ from $c^e_j[l]$ using the prediction operator $\mathcal{P}$:

$$w_{j+1}[l] = c^o_j[l] - \mathcal{P}(c^e_j[l]) \tag{13}$$

3. **Update:** The coarse approximation c_{j+1} of the signal is obtained by using $c_j^e[l]$ and $w_{j+1}[l]$ and the update operator $\mathcal{U}$:

$$c_{j+1}[l] = c_j^e[l] + \mathcal{U}(w_{j+1}[l]) \tag{14}$$

The lifting steps are easily inverted by:

$$\begin{aligned} c_j[2l] &= c_j^e[l] = c_{j+1}[l] - \mathcal{U}(w_{j+1}[l]) \\ c_j[2l+1] &= c_j^o[l] = w_{j+1}[l] + \mathcal{P}(c_j^e[l]) \end{aligned} \tag{15}$$

Some examples of wavelet transforms via the lifting scheme are:

- Haar wavelet via lifting: the Haar transform can be performed via the lifting scheme by taking the predict operator equal to the identity, and an update operator which halves the difference. The transform becomes:

$$\begin{aligned} w_{j+1}[l] &= c_j^o[l] - c_j^e[l] \\ c_{j+1}[l] &= c_j^e[l] + \frac{w_{j+1}[l]}{2} \end{aligned}$$

  All computation can be done in-place.

- Linear wavelets via lifting: the identity predictor used before is correct when the signal is constant. In the same way, we can use a linear predictor which is correct when the signal is linear. The predictor and update operators are now:

$$\begin{aligned} \mathcal{P}(c_j^e[l]) &= \frac{1}{2}\left(c_j^e[l] + c_j^e[l+1]\right) \\ \mathcal{U}(w_{j+1}[l]) &= \frac{1}{4}\left(w_{j+1}[l-1] + w_{j+1}[l]\right) \end{aligned}$$

  It is easy to verify that:

$$c_{j+1}[l] = -\frac{1}{8}c_j[2l-2] + \frac{1}{4}c_j[2l-1] + \frac{3}{4}c_j[2l] + \frac{1}{4}c_j[2l+1] - \frac{1}{8}c_j[2l+2]$$

  which is the bi-orthogonal Cohen-Daubechies-Feauveau []wave:cohen92 wavelet transform.

  The lifting factorization of the popular (9/7) filter pair leads to the following implementation [13]:

$$\begin{aligned} s^{(0)}[l] &= c_j[2l] \\ d^{(0)}[l] &= c_j[2l+1] \\ d^{(1)}[l] &= d^{(0)}[l] + \alpha(s^{(0)}[l] + s^{(0)})[l+1] \\ s^{(1)}[l] &= s^{(0)}[l] + \beta(d^{(1)}[l] + d^{(1)}[l-1]) \\ d^{(2)}[l] &= d^{(1)}[l] + \gamma(s^{(1)}[l] + s^{(1)}[l+1]) \\ s^{(2)}[l] &= s^{(1)}[l] + \delta(d^{(2)}[l] + d^{(2)}[l-1]) \\ c_{j+1}[l] &= us^{(2)}[l] \\ w_{j+1}[l] &= \frac{d^{(2)}[l]}{u} \end{aligned} \tag{16}$$

with

$$\begin{aligned} \alpha &= -1.586134342 \\ \beta &= -0.05298011854 \\ \gamma &= 0.8829110762 \\ \delta &= 0.4435068522 \\ u &= 1.149604398 \end{aligned} \tag{17}$$

Every wavelet transform can be written via lifting.

Integer wavelet transform.

When the input data consist of integer values, the wavelet transform is not necessarily integer-valued. For lossless coding and compression, it is useful to have a wavelet transform which produces integer values. We can build an integer version of every wavelet transform [14]. For instance, denoting $\lfloor x \rfloor$ as the largest integer not exceeding x, the integer Haar transform (also called "S" transform) can be calculated by:

$$\begin{aligned} w_{j+1}[l] &= c_j^o[l] - c_j^e[l] \\ c_{j+1}[l] &= c_j^e[l] + \lfloor \frac{w_{j+1}[l]}{2} \rfloor \end{aligned} \tag{18}$$

while the reconstruction is

$$\begin{aligned} c_j[2l] &= c_{j+1}[l] - \lfloor \frac{w_{j+1}[l]}{2} \rfloor \\ c_j[2l+1] &= w_{j+1}[l] + c_j[2l] \end{aligned} \tag{19}$$

More generally, the lifting operators for an integer version of the wavelet transform are:

$$\begin{aligned} \mathcal{P}(c_j^e[l]) &= \lfloor \sum_k p[k] c_j^e[l-k] + \frac{1}{2} \rfloor \\ \mathcal{U}(w_{j+1}[l]) &= \lfloor \sum_k u[k] w_{j+1}[l-k] + \frac{1}{2} \rfloor \end{aligned} \tag{20}$$

where p and u are appropriate filters associated to primal and dual lifting steps.

For instance, the linear integer wavelet transform[2] is given by

$$\begin{aligned} w_{j+1}[l] &= c_j^o[l] - \lfloor \frac{1}{2}\big(c_j^e[l] + c_j^e[l+1]\big) + \frac{1}{2} \rfloor \\ c_{j+1}[l] &= c_j^e[l] + \lfloor \frac{1}{4}(w_{j+1}[l-1] + w_{j+1}[l]) + \frac{1}{2} \rfloor \end{aligned} \tag{21}$$

More filters can be found in [14]. In lossless compression of integer-valued digital images, even if there is no filter that consistently performs better than all the other filters on all images, it was observed that the linear integer wavelet transform performs generally better than other integer wavelet transforms using other filters [14].

[2]This integer wavelet transform is based upon a symmetric, bi-orthogonal wavelet transform built from the interpolating Deslauriers-Dubuc scaling function where both the high-pass filter and its dual has 2 vanishing moments moments [7].

6 The Undecimated Wavelet Transform

The undecimated wavelet transform, UWT, consists of keeping the filter bank construction which provides a fast and dyadic algorithms, but eliminating the decimation step in the orthogonal wavelet transform [15, 16]: $c_1 = \bar{h} * c_0$ and $w_1 = \bar{g} * c_0$. By separating even and odd pixels in c_1 and w_1, we get (c_1^E, w_1^E) and (c_1^O, w_1^O), and both parts obviously allow us to reconstruct perfectly c_0. The reconstruction can be obtained by

$$c_0 = \frac{1}{2}(\tilde{h} * c_1^E + \tilde{g} * w_1^E + \tilde{h} * c_1^O + \tilde{g} * w_1^O). \tag{22}$$

For the passage to the next resolution, both c_1^E and c_1^O are decomposed, leading, after the splitting into even and odd pixels, to four coarse arrays associated with c_2. All of the four data sets can again be decomposed in order to obtain the third decomposition level, and so on.

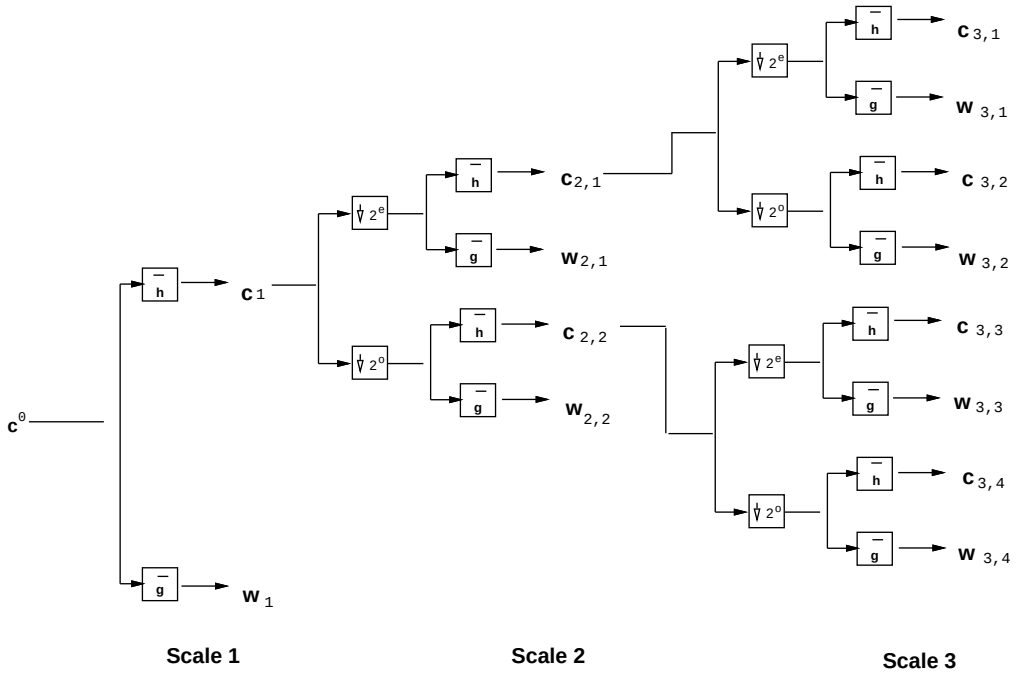

Figure 5: 1D undecimated wavelet transform.

Figure 5 shows the 1D UWT (UWT) decomposition. The decimation step is not applied and both w_1 and c_1 have the same size as c_0. c_1 is then split into c_1^E (even pixels) and c_1^O (odd pixels), and the same decomposition is applied to both c_1^E and c_1^O. c_1^E produces $c_{2,1}$ and $w_{2,1}$, while c_1^O produces $c_{2,2}$ and $w_{2,2}$. $w_2 = \{w_{2,1}, w_{2,1}\}$ contains the wavelet coefficients at the second scale, and is also of the same size as c_0. Figure 6 shows the 1D UWT reconstruction.

It is clear that this approach is much more complicated than the decimated bi-orthogonal wavelet transform. There exists, however, a very efficient way to implement it, called the "à trous" algorithm ("à trous", a French term, meaning *with holes*)[16, 17]. $c_{j+1}[l]$ and $w_{j+1}[l]$ can be

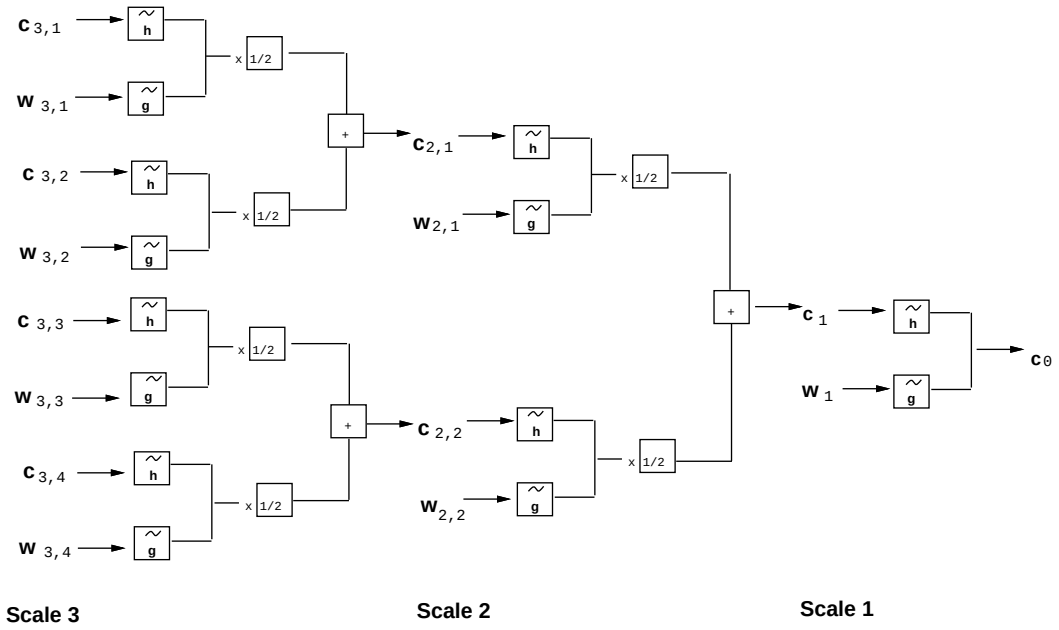

Figure 6: 1D undecimated wavelet reconstruction.

expressed as

$$\begin{aligned} c_{j+1}[l] &= (\bar{h}^{(j)} * c_j)[l] = \sum_k h[k] c_j[l + 2^j k] \\ w_{j+1}[l] &= (\bar{g}^{(j)} * c_j)[l] = \sum_k g[k] c_j[l + 2^j k], \end{aligned} \tag{23}$$

where $h^{(j)}[l] = h[l]$ if $l/2^j$ is an integer and 0 otherwise. For example, we have

$$h^{(1)} = (\ldots, h[-2], 0, h[-1], 0, h[0], 0, h[1], 0, h[2], \ldots)$$

The reconstruction is obtained by

$$c_j[l] = \frac{1}{2}\left[(\tilde{h}^{(j)} * c_{j+1})[l] + (\tilde{g}^{(j)} * w_{j+1})[l]\right]. \tag{24}$$

The filter bank $(h, g, \tilde{h}, \tilde{g})$ needs only to verify the exact reconstruction condition written in the z-transform domain:

$$H(z^{-1})\tilde{H}(z) + G(z^{-1})\tilde{G}(z) = 1. \tag{25}$$

This provides us with a higher degree of freedom when designing the synthesis prototype filter bank.

The à trous algorithm can be extended to 2D, by:

$$\begin{aligned} c_{j+1}[k,l] &= (\bar{h}^{(j)}\bar{h}^{(j)} * c_j)\ [k,l] \\ w^1_{j+1}[k,l] &= (\bar{g}^{(j)}\bar{h}^{(j)} * c_j)\ [k,l] \\ w^2_{j+1}[k,l] &= (\bar{h}^{(j)}\bar{g}^{(j)} * c_j)\ [k,l] \\ w^3_{j+1}[k,l] &= (\bar{g}^{(j)}\bar{g}^{(j)} * c_j)\ [k,l]. \end{aligned} \tag{26}$$

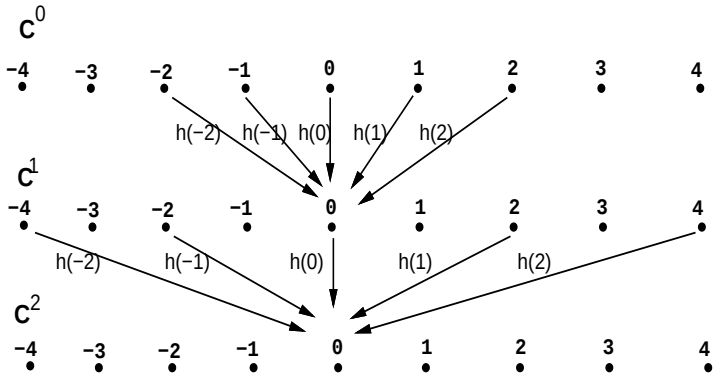

Figure 7: Passage from c_0 to c_1, and from c_1 to c_2 with the UWT à trous algorithm.

where $hg * c$ is the convolution of c by the separable filter hg (i.e. convolution first along the columns by h and then convolution along the rows by g). At each scale, we have three wavelet images, w^1, w^2, w^3, and each has the same size as the original image. The redundancy factor is therefore $3(J-1)+1$ [7].

7 The 2D Isotropic Undecimated Wavelet Transform

The Isotropic Undecimated Wavelet Transform, IUWT, algorithm is well known in the astronomical domain, because it is well adapted to astronomical data where objects are more or less isotropic in most cases [18]. Requirements for a good analysis of such data are:

- Filters must be symmetric ($\bar{h}[k] = h[k]$, and $\bar{g}[k] = g[k]$).
- In 2D or higher dimension, h, g, ψ, ϕ must be nearly isotropic.

Filters do not need to be orthogonal or bi-orthogonal and this lack of the need for orthogonality or bi-orthogonality is beneficial for design freedom. For computational reasons, we also prefer to have the separability; $h[k,l] = h[k]h[l]$. Separability is not a required condition, but it allows us to have a fast calculation, which is important for a large data set.

This has motivated the following choice for the analysis scaling and wavelet functions [18]:

$$\begin{aligned} \phi_1(x) &= \frac{1}{12}(\mid x-2 \mid^3 -4 \mid x-1 \mid^3 +6 \mid x \mid^3 -4 \mid x+1 \mid^3 + \mid x+2 \mid^3) \\ \phi(x,y) &= \phi_1(x)\phi_1(y) \\ \frac{1}{4}\psi\left(\frac{x}{2},\frac{y}{2}\right) &= \phi(x,y) - \frac{1}{4}\phi\left(\frac{x}{2},\frac{y}{2}\right) \end{aligned} \tag{27}$$

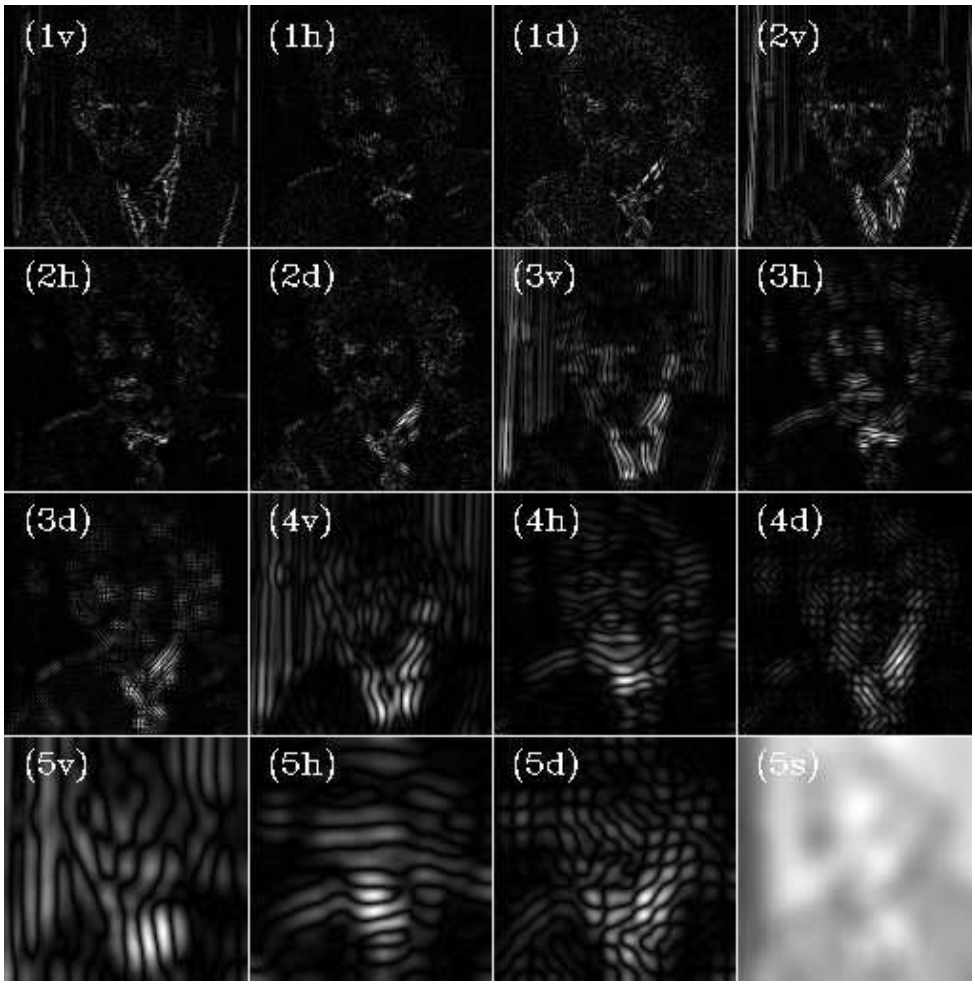

Figure 8: Undecimated wavelet transform of the `Einstein` image.

where $\phi_1(x)$ is the spline of order 3, and the wavelet function is defined as the difference between two resolutions. The related filters h and g are defined by:

$$\begin{aligned} h^{(1D)}[k] &= [1,4,6,4,1]/16,\ k=-2,\ldots,2 \\ h[k,l] &= h^{(1D)}[k]\ h^{(1D)}[l] \\ g[k,l] &= \delta[k,l] - h[k,l] \end{aligned} \tag{28}$$

where δ is defined as $\delta[0,0]=1$ and $\delta[k,l]=0$ for all (k,l) different from $(0,0)$.

The following useful properties characterize any pair of even-symmetric analysis FIR (finite impulse response) filters $(h, g=\delta-h)$ such as those of Eq. 28:

Property 1 *For any pair of even symmetric filters h and g such that $g=\delta-h$, the following holds:*

(i) This FIR filter bank implements a frame decomposition, and perfect reconstruction using FIR filters is possible.

(ii) The above filters can not implement a tight frame decomposition.

See [19] for a proof.

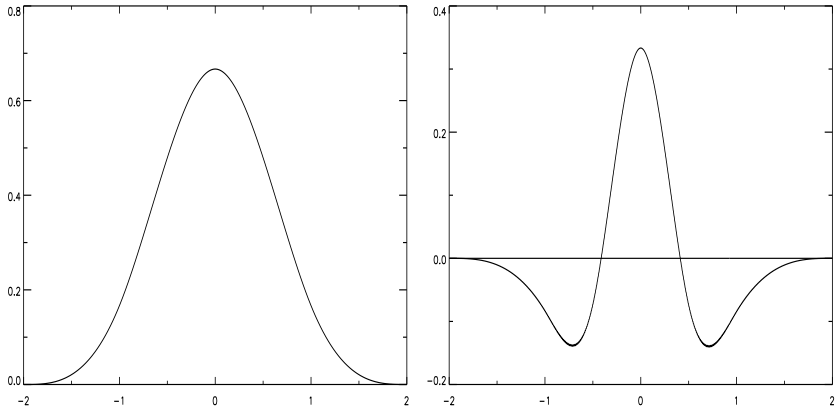

Figure 9: Left, the cubic spline function ϕ; right, the wavelet ψ. $\psi(x)$ is the difference between two resolutions.

Fig. 9 shows respectively the cubic spline scaling function ϕ and the wavelet ψ.

From the structure of g, it is easily seen that the wavelet coefficients are obtained just by taking the difference between two resolutions:

$$w_{j+1}[k,l] = c_j[k,l] - c_{j+1}[k,l] \tag{29}$$

where $c_{j+1}[k,l] = \left(\bar{h}^{(j)}\bar{h}^{(j)} * c_j\right)[k,l]$. At each scale j, we obtain one subband $\{w_j\}$ (and not three as in the undecimated WT, denoted UWT above) which has the same number of pixels as the input image.

The reconstruction is obtained by a simple co-addition of all wavelet scales and the final smoothed array, namely

$$c_0[k,l] = c_J[k,l] + \sum_{j=1}^{J} w_j[k,l] \tag{30}$$

That is, the synthesis filters are $\tilde{h} = \delta$ and $\tilde{g} = \delta$, which are indeed FIR as expected from Property 1(i). This wavelet transformation is very well adapted to the analysis of images which contain isotropic objects such as in astronomy [18] or in biology [20]. This construction has a close relation to the Laplacian pyramidal construction introduced by Burt and Adelson [21] or the FFT-based pyramidal wavelet transform [1].

Figure 10 shows the undecimated isotropic wavelet transform of the image `Einstein` using six resolution levels. This transformation contains 6 bands, each one being of the same size as the original image. The redundancy factor is therefore equal to 6. The simple addition of these six images reproduce exactly the original image.

Relation between the UWT and the IUWT

Since the dealiasing filter bank condition is not required anymore in the UWT decomposition, we can build the standard three-directional undecimated filter bank using the non-(bi-)orthogonal "Astro"

Figure 10: Undecimated isotropic wavelet transform of the `Einstein` image. The addition of these six images reproduce exactly the original image.

filter bank ($h^{1D} = [1, 4, 6, 4, 1]/16$, $g^{1D} = \delta - h^{1D} = [-1, -4, 10, -4, -1]/16$ and $\tilde{h} = \tilde{g} = \delta$). In two dimensions, this filter bank leads to a wavelet decomposition with three orientations w_j^1, w_j^2, w_j^3 at each scale j, but with the same property as for the IUWT, i.e. the sum of all scales reproduces the original image:

$$c_0[k,l] = c_J[k,l] + \sum_{j=1}^{J} \sum_{d=1}^{3} w_j^d[k,l] \tag{31}$$

Indeed, a straightforward calculation immediately shows that [19]:

$$w_j^1 + w_j^2 + w_j^3 = c_j - c_{j+1} \tag{32}$$

Therefore, the sum of the three directions reproduces the IUWT detail band at scale j. Figure 11 shows the UWT of the galaxy NGC2997. When we add the three directional wavelet bands at a given scale, we recover exactly the isotropic undecimated scale. When we add all bands, we recover exactly the original image. The relation between the two undecimated decompositions is clear.

8 Designing non-Orthogonal Filter Banks

A Surprising Result

Because the decomposition is non-subsampled, there are many ways to reconstruct the original image from its wavelet transform[3]. For a given filter bank (h,g), any filter bank ($\tilde{h}$,$\tilde{g}$) which satisfies the reconstruction condition of Eq. 25 leads to exact reconstruction. For instance, for isotropic h, if we choose $\tilde{h} = h$ (the synthesis scaling function $\tilde{\phi} = \phi$) we obtain a filter $\tilde{g}$ defined by [19]:

$$\tilde{g} = \delta + h$$

Again, as expected from Property 1, the analysis filter bank ($h, g = \delta - h$) implements a (non-tight) frame decomposition for FIR symmetric h, where $\tilde{h} = h$ and $\tilde{g} = \delta + h$ are also FIR filters. For instance, if $h = [1, 4, 6, 4, 1]/16$, then $\tilde{g} = [1, 4, 22, 4, 1]/16$. $\tilde{g}$ **is positive** [19]. This means that $\tilde{g}$ is no longer related to a wavelet function. The synthesis scaling function related to $\tilde{g}$ is defined by:

$$\frac{1}{2}\tilde{\psi}\left(\frac{x}{2}\right) = \phi(x) + \frac{1}{2}\phi\left(\frac{x}{2}\right) \quad (33)$$

Finally, note that choosing $\tilde{\phi} = \phi$, any synthesis function $\tilde{\psi}$ which satisfies

$$\hat{\tilde{\psi}}(2\nu)\hat{\psi}(2\nu) = \hat{\phi}^2(\nu) - \hat{\phi}^2(2\nu) \quad (34)$$

leads to an exact reconstruction [7] and $\hat{\tilde{\psi}}(0)$ can take any value. The synthesis function $\tilde{\psi}$ does not need to verify the admissibility condition (i.e. to have a zero mean).

Figure 12 shows the two scaling functions $\tilde{\phi}(x)$ $(= \phi)$ and $\tilde{\psi}(x)$ used in the reconstruction in 1D, corresponding to the synthesis filters $\tilde{h} = h$ and $\tilde{g} = \delta + h$. Figure 13 shows the backprojection of a wavelet coefficient in 2D (all wavelet coefficients are set to zero, except one), when the non-zero coefficient belongs to different bands. We can see that the reconstruction functions are positive.

Finally, we have an expansion of a 1D signal s,

$$s(x) = \sum_k c_J[k]\tilde{\phi}_{J,k}(x) + \sum_{j=1}^{J}\sum_k w_j[k]\tilde{\psi}_{j,k}(x) \quad (35)$$

where $\tilde{\phi}$ and $\tilde{\psi}$ are not wavelet functions (both of them have a non-zero mean and are positive), but the w_j are wavelet coefficients.

Reconstruction from the Haar Undecimated Coefficients

The Haar filters ($h = \tilde{h} = [1/2, 1/2], g = \tilde{g} = [-1/2, 1/2]$) are not considered as good filters in practice because of their lack of smoothness. They are however very useful in many situations such as denoising where their simplicity allows us to derive analytical or semi-analytical detection levels even when the noise does not follow a Gaussian distribution.

Adopting the same design approach as before, we can reconstruct a signal from its Haar wavelet coefficients choosing a smooth scaling function. For instance, if $\tilde{h} = [1, 4, 6, 4, 1]/16$, it is easy to derive that the z transforms of these three filters are respectively:

$$H(z) = \frac{1 + z^{-1}}{2}, \quad G(z) = \frac{z^{-1} - 1}{2}, \quad \tilde{H}(z) = \frac{z^2 + 4z + 6 + 4z^{-1} + z^{-2}}{16} \quad (36)$$

[3]In frame theory parlance, we would say that the UWT frame synthesis operator is not injective.

From the exact reconstruction condition in Eq. 25, we obtain:

$$\tilde{G}(z) = \frac{1 - \tilde{H}(z)H(z^{-1})}{G(z^{-1})} \tag{37}$$

In the case of the spline filter bank, this yields after some re-arrangement (where we used simple convolution properties of splines),

$$\tilde{G}(z) = -2\ \frac{1 - z^3\left(\frac{1+z^{-1}}{2}\right)^5}{1 - z^{-1}} = z^3\ \frac{1 + 6z^{-1} + 16z^{-2} - 6z^{-3} - z^{-4}}{16} \tag{38}$$

which is the z-transform of the corresponding filter $\tilde{g} = [1, 6, 16, -6, -1]/16$.

The Haar analysis filters fulfill the following property:

Property 2 *Haar analysis filters can implement a tight frame expansion (more precisely, one scale of the Haar wavelet UWT does). Perfect reconstruction with FIR synthesis filters is possible.*

Figure 14, upper left and right, depicts the coarsest scale and a wavelet scale of the Haar transform when the input signal contains only zero values except one sample (Dirac $\delta[k]$). Figure 14, bottom left, portrays the backprojection of a Dirac at the coarsest scale (all coefficients are set to zero) and Figure 14, bottom right, shows the backprojection of a Haar wavelet coefficient. Since the synthesis filters are regular, the backprojection of a Dirac does not produce any block staircase-like artifact. Finally, we would like to point out that other alternatives exist. For example the filter bank ($h = [1/2, -1/2]$,$g = [-1/4, 1/2, -1/4]$, $\tilde{h} = [1, 3, 3, 1]/8$ and $\tilde{g} = [1, 6, 1]/4$ leads also to an interesting solution where the synthesis filters are both positive.

Another Interesting Filter Bank

A particular case is obtained when $\hat{\tilde{\phi}} = \hat{\phi}$ and $\hat{\psi}(2\nu) = \frac{\hat{\phi}^2(\nu) - \hat{\phi}^2(2\nu)}{\hat{\phi}(\nu)}$, which leads to a filter g equal to $\delta - h * h$. In this case, the synthesis function $\tilde{\psi}$ is defined by $\frac{1}{2}\tilde{\psi}(\frac{x}{2}) = \phi(x)$ and the filter $\tilde{g} = \delta$ is the solution to Eq. 25. We end up with a synthesis scheme where only the smooth part is convolved during the reconstruction. Furthermore, for a symmetric FIR filter h, it can be easily shown that this filter bank fulfills the statements of Property 1.

Deriving h from a spline scaling function, for instance B_1 ($h_1 = [1, 2, 1]/4$) or B_3 ($h_3 = [1, 4, 6, 4, 1]/16$) (note that $h_3 = h_1 * h_1$), since h is even-symmetric (i.e. $H(z) = H(z^{-1})$), the z-transform of g is then:

$$\begin{aligned} G(z) = 1 - H^2(z) &= 1 - z^4\left(\frac{1 + z^{-1}}{2}\right)^8 \\ &= \left(-z^4 - 8z^3 - 28z^2 - 56z + 186 - 56z^{-1} - 28z^{-2} - 8z^{-3} - z^{-4}\right)/256 \end{aligned} \tag{39}$$

which is the z-transform of the filter $g = [-1, -8, -28, -56, 186, -56, -28, -8, -1]/256$. We get the following filter bank:

$$\begin{aligned} h = h_3 &= \tilde{h} = [1, 4, 6, 4, 1]/16 \\ g &= \delta - h * h = [-1, -8, -28, -56, 186, -56, -28, -8, -1]/256 \quad (40) \\ \tilde{g} &= \delta \quad (41) \end{aligned}$$

With this filter bank, there is a no convolution with the filter $\tilde{g}$ during the reconstruction. Only the low-pass synthesis filter $\tilde{h}$ is used. The reconstruction formula is:

$$c_j[l] = (h^{(j)} * c_{j+1})[l] + w_{j+1}[l] \tag{42}$$

and denoting $L^j = h^{(0)} * \cdots * h^{(j-1)}$ and $L^0 = \delta$, we have

$$c_0[l] = \left(L^J * c_J\right)[l] + \sum_{j=1}^{J} \left(L^{j-1} * w_j\right)[l] \tag{43}$$

Each wavelet scale is convolved with a low-pass filter.

Figure 15 shows the analysis scaling and wavelet functions. The synthesis functions $\tilde{\phi}$ and $\tilde{\psi}$ are the same as those in Figure 12.

9 Iterative Reconstruction

Denoting $\mathcal{W}$ the undecimated wavelet transform operator and $\mathcal{R}$ the reconstruction operator, and thanks to the exact reconstruction formulae, we have the relation: $\alpha_S = \mathcal{W}\mathcal{R}\alpha_S$, where S is a signal or image and α_S its wavelet coefficients (i.e. $\alpha_S = \mathcal{W}S$). But we loose one fundamental property of the (bi-)orthogonal WT. Indeed, the relation $\alpha = \mathcal{W}\mathcal{R}\alpha$ is not true for all α sets. For example, if we set all wavelet coefficients to zero except one at a coarse scale, there is no image such that its UWT would produce a Dirac at a coarse scale. Another way to understand this point is to consider the Fourier domain of a given undecimated scale. Indeed, wavelet coefficients α_j at scale j obtained using the wavelet transform operator will contain information only localised at a given frequency band. But any modification of the coefficients at this scale, such as a thresholding ($\alpha_T = \Delta_T(\alpha)$, where Δ_T is the thresholding operator with threshold T and α_T are the thresholded coefficients), will introduce some frequency components which should not exist at this scale j, and we have $\alpha_T \neq \mathcal{W}\mathcal{R}\alpha_T$.

Reconstruction from a Subset of Coefficients

Without loss of generality, we consider hereafter the case of 1D signals. If only a subset of coefficients (for instance after thresholding) is different from zero, we would like to reconstruct an image $\tilde{S}$ such that its wavelet transform reproduces the non-zero wavelet coefficients. This can be cast as an inverse problem. We want to solve the following optimization problem $\min_{\tilde{S}} \parallel M(\alpha_T - \mathcal{W}\tilde{S}) \parallel_2^2$ where $M_j[k]$ is the multiresolution support of α, i.e. $M_j[k] = 1$ if the wavelet coefficient $\alpha_j[k]$ at scale j and at position k is different from zero, and $M_j[k] = 0$ otherwise. A solution can be obtained using the Landweber iterative scheme [22, 1]:

$$\tilde{S}^{n+1} = \tilde{S}^n + \mathcal{R}M\left[\alpha_T - \mathcal{W}\tilde{S}^n\right] \tag{44}$$

If the solution is known to be positive, the positivity constraint can be introduced using the following equation:

$$\tilde{S}^{n+1} = P_+\left(\tilde{S}^n + \mathcal{R}M\left[\alpha_T - \mathcal{W}\tilde{S}^n\right]\right) \tag{45}$$

where P_+ is the projection on the cone of non-negative images. This iterative scheme can also be interpreted in terms of alternating projections onto convex sets (POCS). It has also proven very effective at many tasks such as image approximation and restoration when using the UWT [19].

10 Future Directions

For 2D or 3D data set, wavelet bases present some intrinsic limitations, because they are not adapted to the detection of highly anisotropic elements, such as lines or curvilinear structures in an image, or sheets in a cube. Recently, other multiscale systems like curvelets [23, 24, 25, 26] and ridgelets [27] which are very different from wavelet-like systems have been developed. Curvelets and ridgelets take the form of basis elements which exhibit very high directional sensitivity and are highly anisotropic. A digital implementation of both the ridgelet and the curvelet transform for image denoising has been described in [24]. These new data representations, combined with wavelets, have been used in many applications such as denoising [24, 28, 29], deconvolution [30], contrast enhancement [31], texture analysis [32, 33], detection [34], watermarking [35], component separation [36], inpainting [37] or blind source separation [38, 39].

To reach higher sparsity levels, the transforms just mentioned with a fixed geometry can be replaced by adaptive representations using an optimized basis. Geometric transforms such as wedgelets [40] or bandlets [41, 42] allow to define an adapted multiscale geometry. These transforms perform a non-linear search for an optimal representation. They offer geometrical adaptivity together with fast and stable algorithms. Recently, Mallat [43] proposed a more biologically inspired procedure named the grouplet transform, which defines a multiscale association field by grouping together pairs of wavelet coefficients.

Following Olshausen and Field [44], one can push one step forward the idea of adaptive sparse representation and requires that the dictionary is not fixed but rather optimized to sparsify a set of exemplar signals/images. Such a learning problem corresponds to finding a sparse matrix factorization as exposed in the K-SVD framework [45]. Explicit structural constraints such as translation invariance can also be enforced on the learned dictionary [46, 47].

References

[1] J.-L. Starck, F. Murtagh, and A. Bijaoui. *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press, 1998.

[2] MR/1. Multiresolution Image and Data Analysis Software Package, Version 3.0, Multi Resolutions Ltd., http://www.multiresolution.com, 2001.

[3] Wavelab 802. (http://www-stat.stanford.edu/∼wavelab/index_wavelab802.html), 2001.

[4] Wavelab 805. (http://www-stat.stanford.edu/∼wavelab/), 2005.

[5] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[6] G. Strang and T. Nguyen. *Wavelet and Filter Banks*. Wellesley-Cambridge Press, 1996.

[7] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[8] A. Cohen. *Numerical Analysis of Wavelet Methods*. Elsevier, 2003.

[9] A. Grossmann, R. Kronland-Martinet, and J. Morlet. Reading and understanding the continuous wavelet transform. In J.M. Combes, A. Grossmann, and Ph. Tchamitchian, editors, *Wavelets: Time-Frequency Methods and Phase-Space*, pages 2–20. Springer-Verlag, 1989.

[10] W. Sweldens and P. Schröder. Building your own wavelets at home. In *Wavelets in Computer Graphics*, pages 15–87. ACM SIGGRAPH Course notes, 1996.

[11] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, 29:511–546, 1997.

[12] Peter Schröder and Wim Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics Proceedings (SIGGRAPH 95)*, pages 161–172, 1995.

[13] I. Daubechies and W. Sweldens. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4:245–267, 1998.

[14] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. *Applied and Computational Harmonic Analysis*, 5:332–369, 1998.

[15] P. Dutilleux. An implementation of the "algorithme à trous" to compute the wavelet transform. In J.M. Combes, A. Grossmann, and Ph. Tchamitchian, editors, *Wavelets: Time-Frequency Methods and Phase-Space*. Springer New York, 1989.

[16] M. Holschneider, R. Kronland-Martinet, J. Morlet, and P. Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets: Time-Frequency Methods and Phase-Space*, pages 286–297. Springer-Verlag, 1989.

[17] M.J. Shensa. Discrete wavelet transforms: Wedding the à trous and Mallat algorithms. *IEEE Transactions on Signal Processing*, 40:2464–2482, 1992.

[18] J.-L. Starck and F. Murtagh. *Astronomical Image and Data Analysis*. Springer-Verlag, 2002.

[19] J.-L Starck, J. Fadili, and F. Murtagh. The undecimated wavelet decomposition and its reconstruction. *IEEE Transactions on Image Processing*, 16:297–309, 2007.

[20] A. Genovesio and J.-C. Olivo-Marin. Tracking fluorescent spots in biological video microscopy. In J.-A. Conchello, C.J. Cogswell, and T. Wilson, editors, *Three-Dimensional and Multidimensional Microscopy: Image Acquisition and Processing X*, volume 4964, pages 98–105. SPIE, 2003.

[21] P.J. Burt and A.E. Adelson. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31:532–540, 1983.

[22] J.-L. Starck, A. Bijaoui, and F. Murtagh. Multiresolution support applied to image filtering and deconvolution. *CVGIP: Graphical Models and Image Processing*, 57:420–431, 1995.

[23] E. J. Candès and D. L. Donoho. Curvelets – a surprisingly effective nonadaptive representation for objects with edges. In A. Cohen, C. Rabut, and L.L. Schumaker, editors, *Curve and Surface Fitting: Saint-Malo 1999*, Nashville, TN, 1999. Vanderbilt University Press.

[24] J.-L. Starck, E. Candès, and D.L. Donoho. The curvelet transform for image denoising. *IEEE Transactions on Image Processing*, 11(6):131–141, 2002.

[25] M. N. Do and M. Vetterli. Contourlets. In J. Stoeckler and G. V. Welland, editors, *Beyond Wavelets*. Academic Press, 2003.

[26] E.Candes, L.Demanet, D.Donoho, and L.Ying. Fast discrete curvelet transforms. *SIAM Multiscale Model. Simul.*, 5/3:861–899, 2006.

[27] E.J. Candès and D.L. Donoho. Ridgelets: the key to high dimensional intermittency? *Philosophical Transactions of the Royal Society of London A*, 357:2495–2509, 1999.

[28] B. Saevarsson, J. Sveinsson, and J. Benediktsson. Speckle reduction of sar images using adaptive curvelet domain. In *Proceedings of the IEEE International Conference on Geoscience and Remote Sensing Symposium, IGARSS '03*, volume 6, pages 4083–4085, 2003.

[29] G. Hennenfent and F.J. Herrmann. Seismic denoising with nonuniformly sampled curvelets. *IEEE Computing in Science and Engineering*, 8(3):16–25, May 2006.

[30] J.-L. Starck, M.K. Nguyen, and F. Murtagh. Wavelets and curvelets for image deconvolution: a combined approach. *Signal Processing*, 83(10):2279–2283, 2003.

[31] J.-L. Starck, F. Murtagh, E. Candes, and D.L. Donoho. Gray and color image contrast enhancement by the curvelet transform. *IEEE Transactions on Image Processing*, 12(6):706–717, 2003.

[32] J.-L Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representation and a variational approach. *IEEE Transactions on Image Processing*, 14(10):1570–1582, 2005.

[33] S. Arivazhagan, L. Ganesan, and T.S. Kumar. Texture classification using curvelet statistical and co-occurrence feature. In *Proceedings of the 18th International Conference on Pattern Recognition (ICPR 2006)*, volume 2, pages 938–941, 2006.

[34] J. Jin, J.-L. Starck, D.L. Donoho, N. Aghanim, and O. Forni. Cosmological non-gaussian signatures detection: Comparison of statistical tests. *Eurasip Journal*, 15:2470–2485, 2005.

[35] Z. Zhang, W. Huang, J. Zhang, H. Yu, and Y. Lu. Digital image watermark algorithm in the curvelet domain. In *Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP'06)*, pages 105–108, 2006.

[36] J.-L. Starck, M. Elad, and D.L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132, 2004.

[37] M. Elad, J.-L Starck, D. Donoho, and P. Querre. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Journal on Applied and Computational Harmonic Analysis*, 19:340–358, 2006.

[38] J. Bobin, Y. Moudden, J.-L. Starck, and M. Elad. Morphological diversity and source separation. *IEEE Trans. on Signal Processing*, 13(7):409–412, 2006.

[39] J. Bobin, J.-L Starck, J. Fadili, and Y. Moudden. Sparsity, morphological diversity and blind source separation. *IEEE Transactions on Image Processing*, 2007. submitted.

[40] D.L Donoho. Wedgelets: nearly-minimax estimation of edges. *Ann. Statist*, 27:859–897, 1999.

[41] E. Le Pennec and S. Mallat. Bandelet Image Approximation and Compression. *SIAM Multiscale Modeling and Simulation*, 4(3):992–1039, 2005.

[42] S. Mallat and G. Peyré. Orthogonal bandlet bases for geometric images approximation. *To appear in Com. Pure and Applied Mathematics*, 2006.

[43] S. Mallat. Geometrical grouplets. *Submitted to Applied and Computational Harmonic Analysis*, 2006.

[44] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive-field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, June 1996.

[45] M. Aharon, M. Elad, and A.M. Bruckstein. The K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. On Signal Processing*, 54(11):4311–4322, November 2006.

[46] B. A. Olshausen. Sparse coding of time-varying natural images. In *Int. Conf. Independent Component Analysis and Blind Source Separation (ICA)*, pages 603–608, Barcelona, Spain, 2000.

[47] T. Blumensath and M. Davies. Sparse and shift-invariant representations of music. *IEEE Transactions on Speech and Audio Processing*, 14(1):50–57, 2006.

A Selection of Wavelet Relative Books

- Stephane Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.
- Yves Meyer and Robert Ryan *Wavelets: Algorithms & Applications*, Society for Industrial and Applied Mathematic, 1993.
- Barbara Burke Hubbard, *The World According to Wavelets: The Story of a Mathematical Technique in the Making*, A.K. Peters, 1995.
- James S. Walker, *A Primer on Wavelets and Their Scientific Applications*, 1999.
- Ingrid Daubechies, *Ten Lectures on Wavelets*, Society for Industrial and Applied Mathematics Press, vol. 61 of CBMS-NSF Regional Conference Series in Applied Mathematics, Philadelphia, 1992.
- Martin Vetterli and Jelena Kovacevic, *Wavelets and Subband Coding*, Prentice-Hall, New Jersey, 1995.
- Brani Vidakovic, *Statistical Modeling by Wavelets*, Wiley Series in Probability and Statistics, 1999.
- Stephane Jaffard, Robert D. Ryan and Yves Meyer, *Wavelets: Tools for Science and Technology*, Society for Industrial and Applied Mathematics Press, 2001
- Tony Chan and James Shen, *Image Processing And Analysis: Variational, Pde, Wavelet, And Stochastic Methods*, 2005.

- Jean-Pierre Antoine, Romain Murenzi, Pierre Vandergheynst and Syed T. Ali , *Two-Dimensional Wavelets and their Relatives*, Cambridge University Press, 2004.
- Jean-Luc Starck, Fionn Murtagh and Albert Bijaoui, *Image Processing and Data Analysis : The Multiscale Approach*, Cambridge University Press, 1998.
- Albert Cohen, *Numerical Analysis of Wavelet Methods*, Elsevier, 2003.
- Jean-Luc Starck and Fionn Murtagh, *Astronomical Data Analysis*, second edition, Springer, 2006.

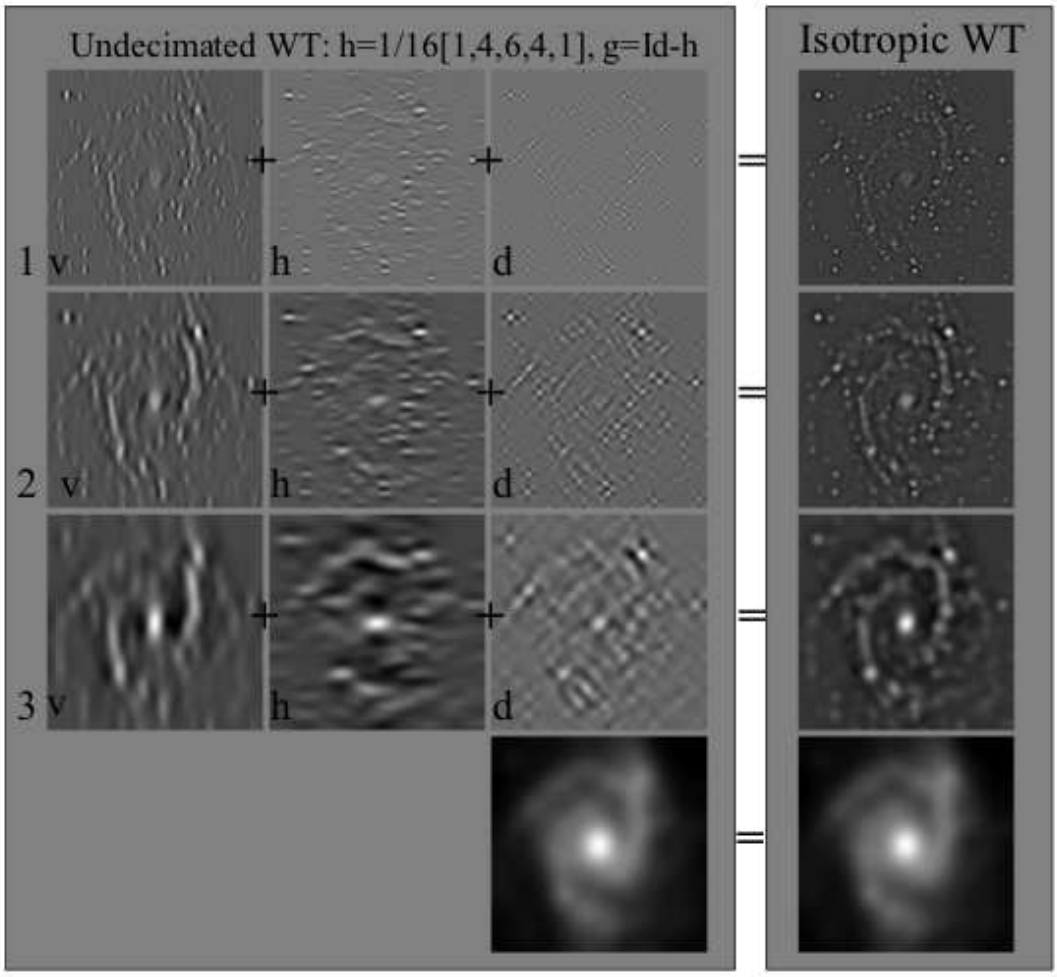

Figure 11: UWT of the galaxy NGC2997 using the Astro filter bank. The addition of three bands at a given scale is exactly the band related to the isotropic wavelet transform. Addition of all bands reproduces exactly the original image.

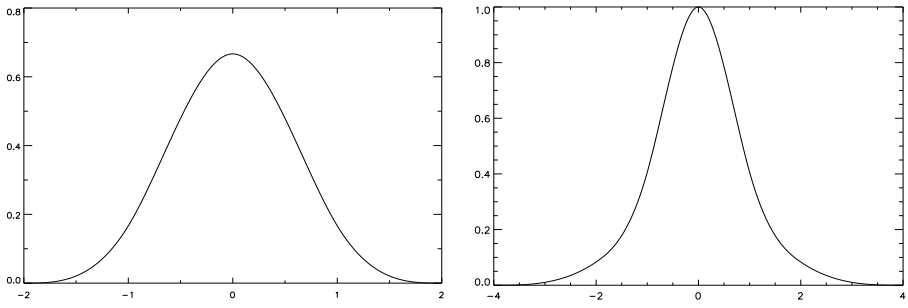

Figure 12: Left, the $\tilde{\phi}$ synthesis scaling function and right, the $\tilde{\psi}$ detail synthesis function.

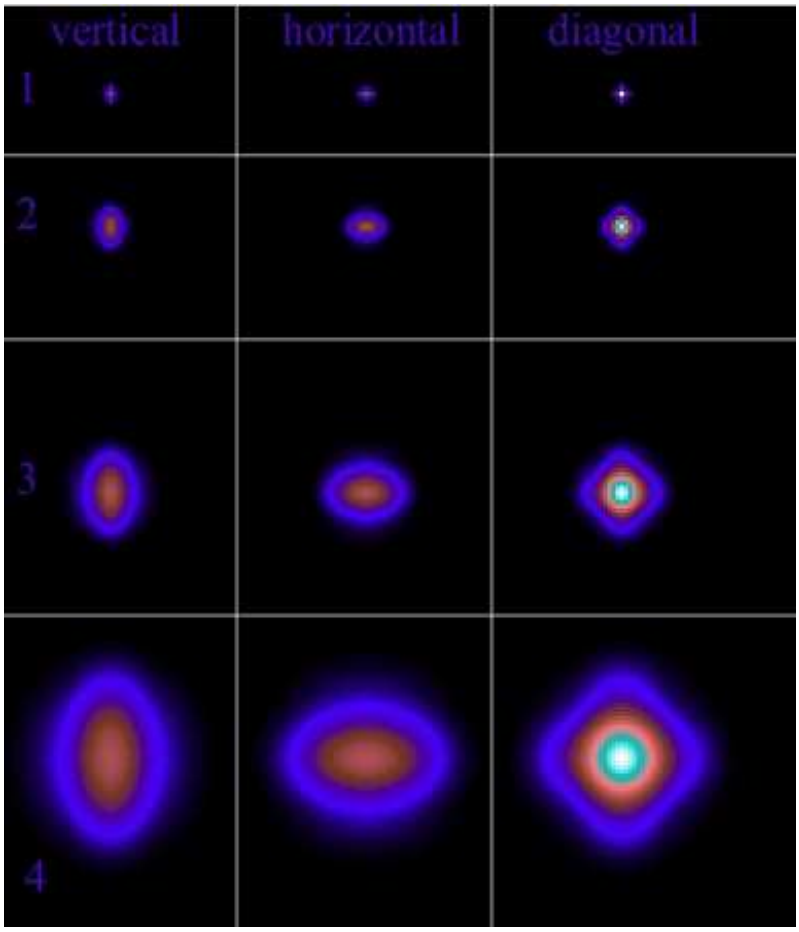

Figure 13: Back projection: Each image corresponds to the backprojection of one wavelet coefficient. All these reconstructed images are positive (no negative values). From left to right, the coefficient belongs to the vertical, horizontal and diagonal direction. From top to bottom, the scale index increases.

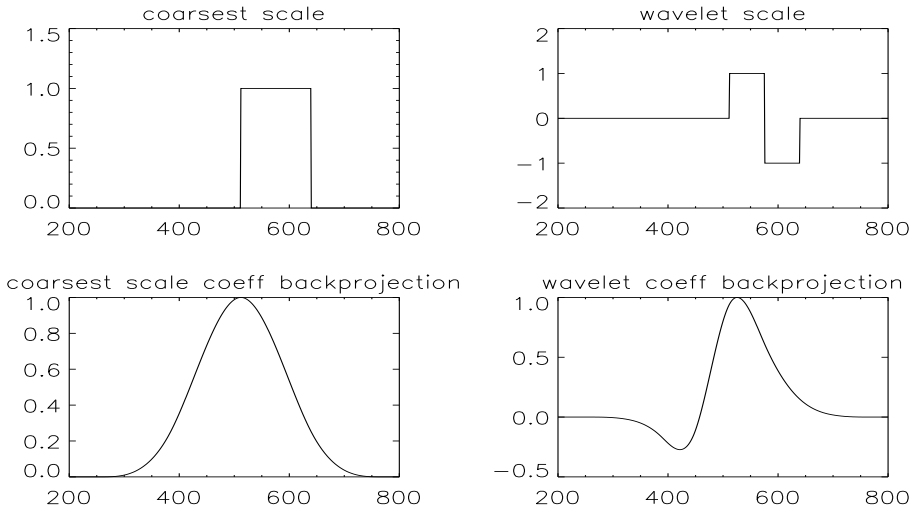

Figure 14: Haar Undecimated Transform: Upper Left, coarsest scale when the signal is $\delta[k]$. Upper right, one wavelet scale of the Dirac decomposition. Bottom left, backprojection of a Dirac at the coarsest scale. Bottom right, backprojection of a Haar wavelet coefficient.

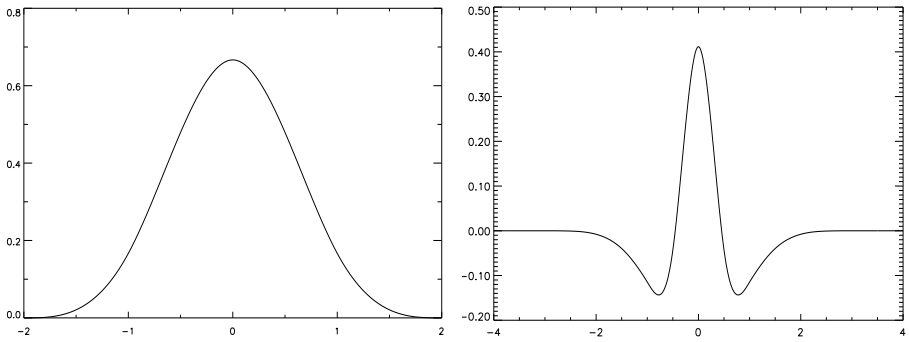

Figure 15: Left, the ϕ analysis scaling function and right, the ψ analysis wavelet function. The synthesis functions $\tilde{\phi}$ and $\tilde{\psi}$ are the same as those in Figure 12.