The à trous Algorithm

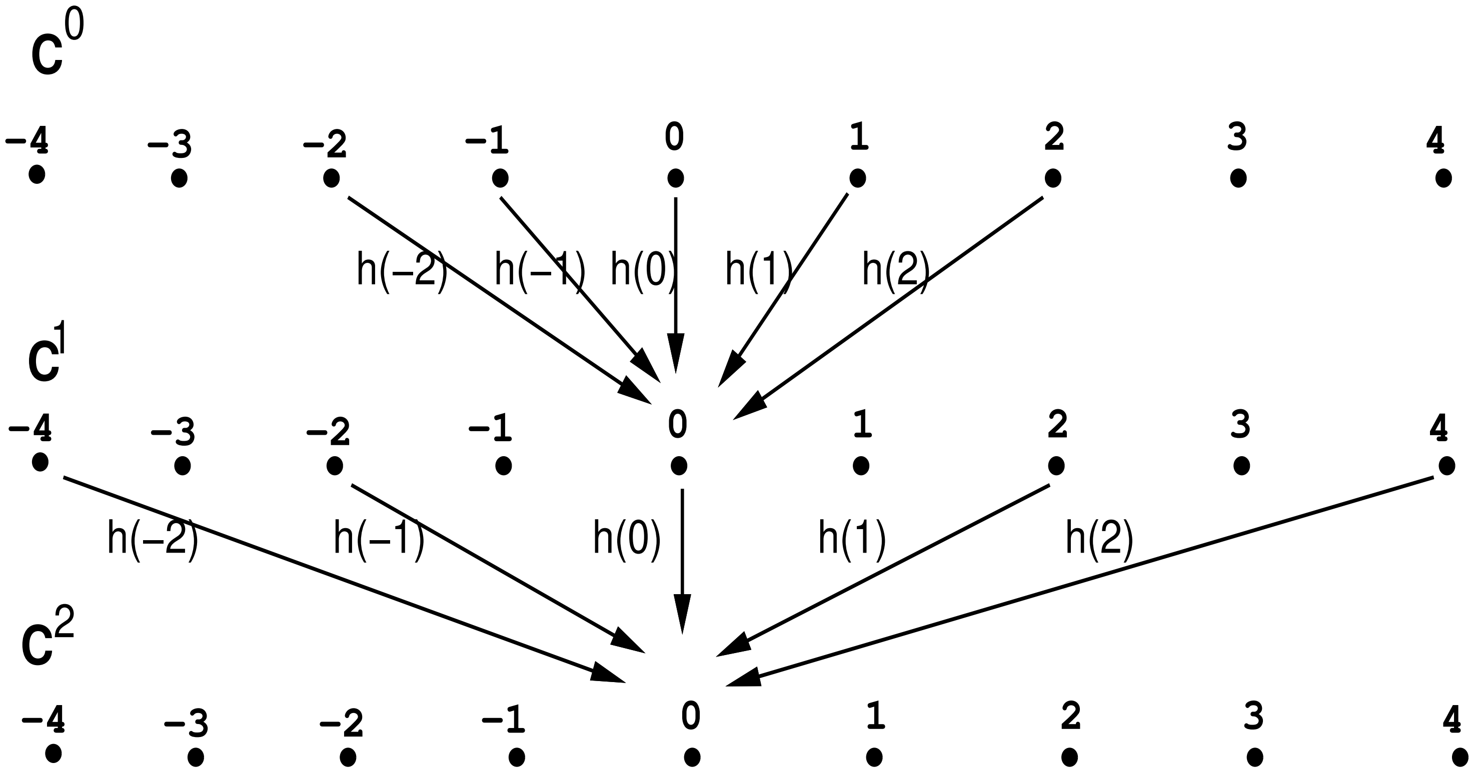
This is a “stationary” or redundant transform, i.e. decimation is not carried out. The distance between samples increasing by a factor 2 from scale j to the next. $c_{j+1,k}$ is obtained by:

$$c_{j+1,k} = \sum_l h(l) c_{j,k+2^j l}$$

and the discrete wavelet coefficients, with:

$$w_{j+1,k} = \sum_l g(l) c_{j,k+2^j l}$$

Passage from c_0 to c_1, and from c_1 to c_2



Generally, the wavelet resulting from the difference between two successive approximations is applied:

$$w_{j+1,k} = c_{j,k} - c_{j+1,k}$$

The associated wavelet is $\psi(x)$.

$$\frac{1}{2}\psi(\frac{x}{2}) = \phi(x) - \frac{1}{2}\phi(\frac{x}{2})$$

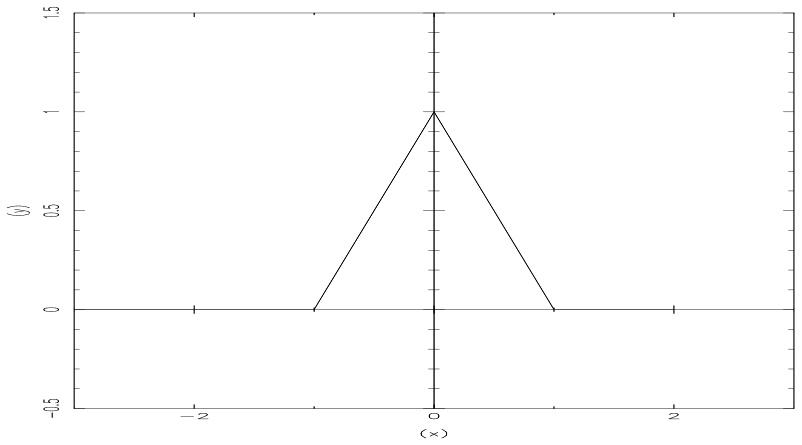
The reconstruction algorithm is immediate:

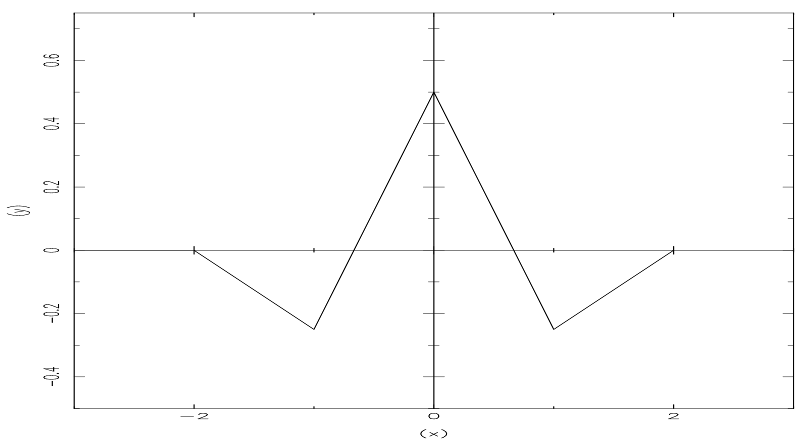
$$c_{0,k} = c_{J,k} + \sum_{j=1}^{J} w_{j,k}$$

The triangle function

Choosing the triangle function as the scaling function ϕ leads to piecewise linear interpolation:

$$\phi(x) = 1 - \mid x \mid \quad \text{if } x \in [-1, 1]$$
$$\phi(x) = 0 \quad \text{if } x \notin [-1, 1]$$





We have

$$\frac{1}{2}\phi(\frac{x}{2}) = \frac{1}{4}\phi(x+1) + \frac{1}{2}\phi(x) + \frac{1}{4}\phi(x-1)$$

c_{j+1} is obtained from c_j by:

$$c_{j+1,k} = \frac{1}{4}c_{j,k-2^j} + \frac{1}{2}c_{j,k} + \frac{1}{4}c_{j,k+2^j}$$

The wavelet coefficients at scale j are:

$$w_{j+1,k} = -\frac{1}{4}c_{j,k-2^j} + \frac{1}{2}c_{j,k} - \frac{1}{4}c_{j,k+2^j}$$

1. We initialize j to 0 and we start with the data $c_{j,k}$.
2. We increment j, and we carry out a discrete convolution of the data $c_{j-1,k}$ using the filter h. The distance between the central pixel and the adjacent ones is 2^{j-1}.
3. After this smoothing, we obtain the discrete wavelet transform from the difference $c_{j-1,k} - c_{j,k}$.
4. If j is less than the number J of resolutions we want to compute, then go to step 2.
5. The set $\mathcal{W} = \{w_1, ..., w_J, c_J\}$ represents the wavelet transform of the data.

A trous algorithm in 2D

In 2D, it leads to a convolution with a mask of 3×3 pixels for the wavelet associated with linear interpolation. The coefficients of the mask are:

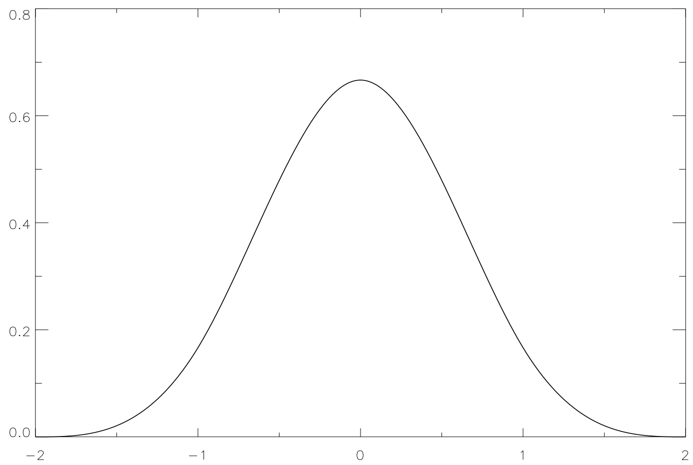
$$\left(\begin{array}{ccc} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{array} \right) \left(\begin{array}{c} 1/4 \\ 1/2 \\ 1/4 \end{array} \right)$$

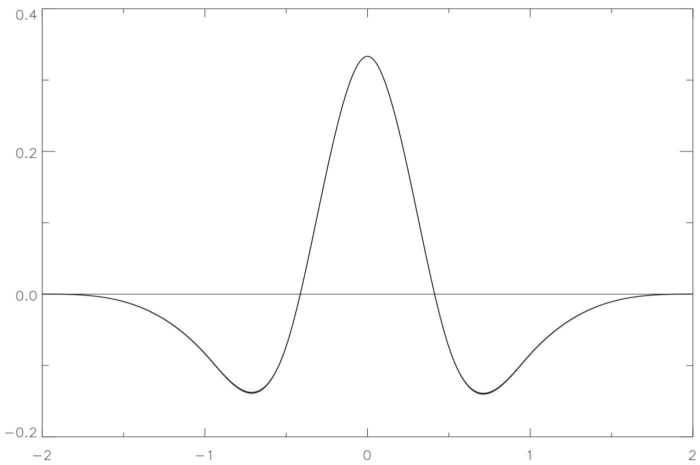
The cubic spline function

$$B_3(x) = \frac{1}{12}(\mid x-2 \mid^3 -4 \mid x-1 \mid^3 +6 \mid x \mid^3 -4 \mid x+1 \mid^3 + \mid x+2 \mid^3)$$

and

$$h(-2) = \frac{1}{16}; h(-1) = \frac{1}{4}; h(0) = \frac{3}{8}; h(1) = \frac{1}{4}; h(2) = \frac{1}{16};$$





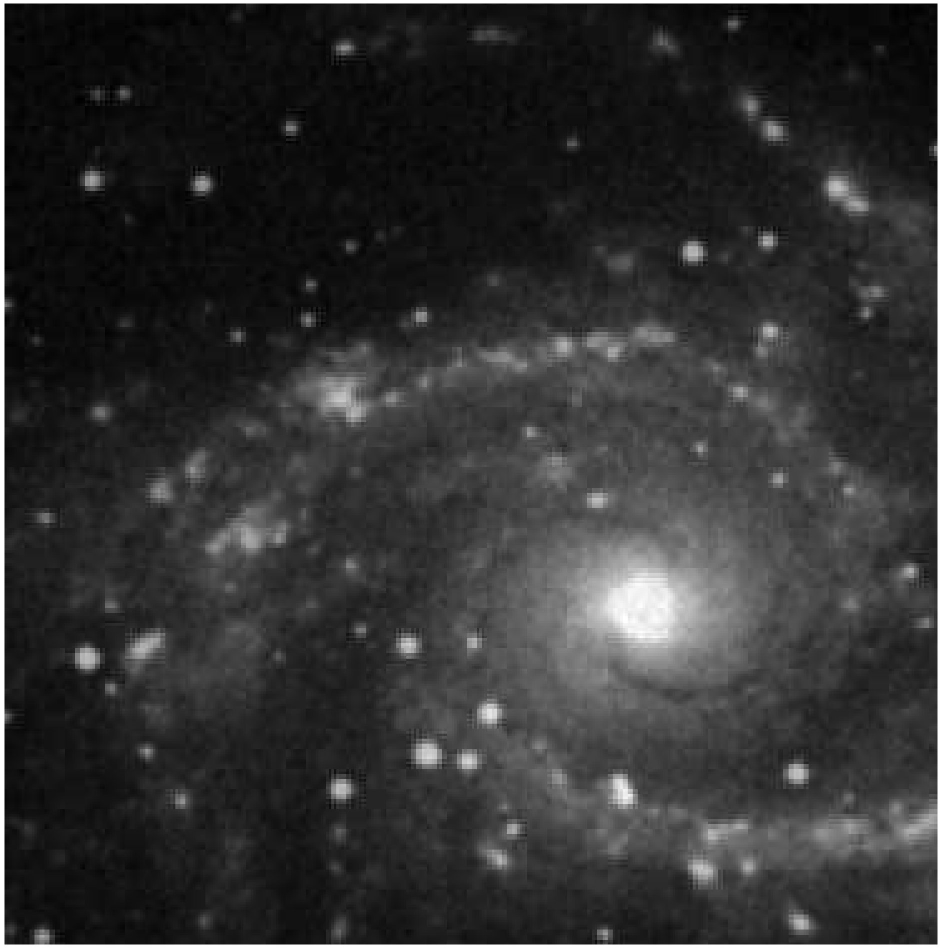
$$
\begin{pmatrix} \frac{1}{16} & \frac{1}{4} & \frac{3}{8} & \frac{1}{4} & \frac{1}{16} \end{pmatrix} \otimes \begin{pmatrix} 1/16 \\ 1/4 \\ 3/8 \\ 1/4 \\ 1/16 \end{pmatrix} = \begin{pmatrix} \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{3}{128} & \frac{3}{32} & \frac{9}{64} & \frac{3}{32} & \frac{3}{128} \\ \frac{1}{64} & \frac{1}{16} & \frac{3}{32} & \frac{1}{16} & \frac{1}{64} \\ \frac{1}{256} & \frac{1}{64} & \frac{3}{128} & \frac{1}{64} & \frac{1}{256} \end{pmatrix}
$$

In the 2-dimensional case, we assume the separability, which leads to a row-by-row convolution with $(\frac{1}{16}, \frac{1}{4}, \frac{3}{8}, \frac{1}{4}, \frac{1}{16})$; followed by column-by-column convolution.

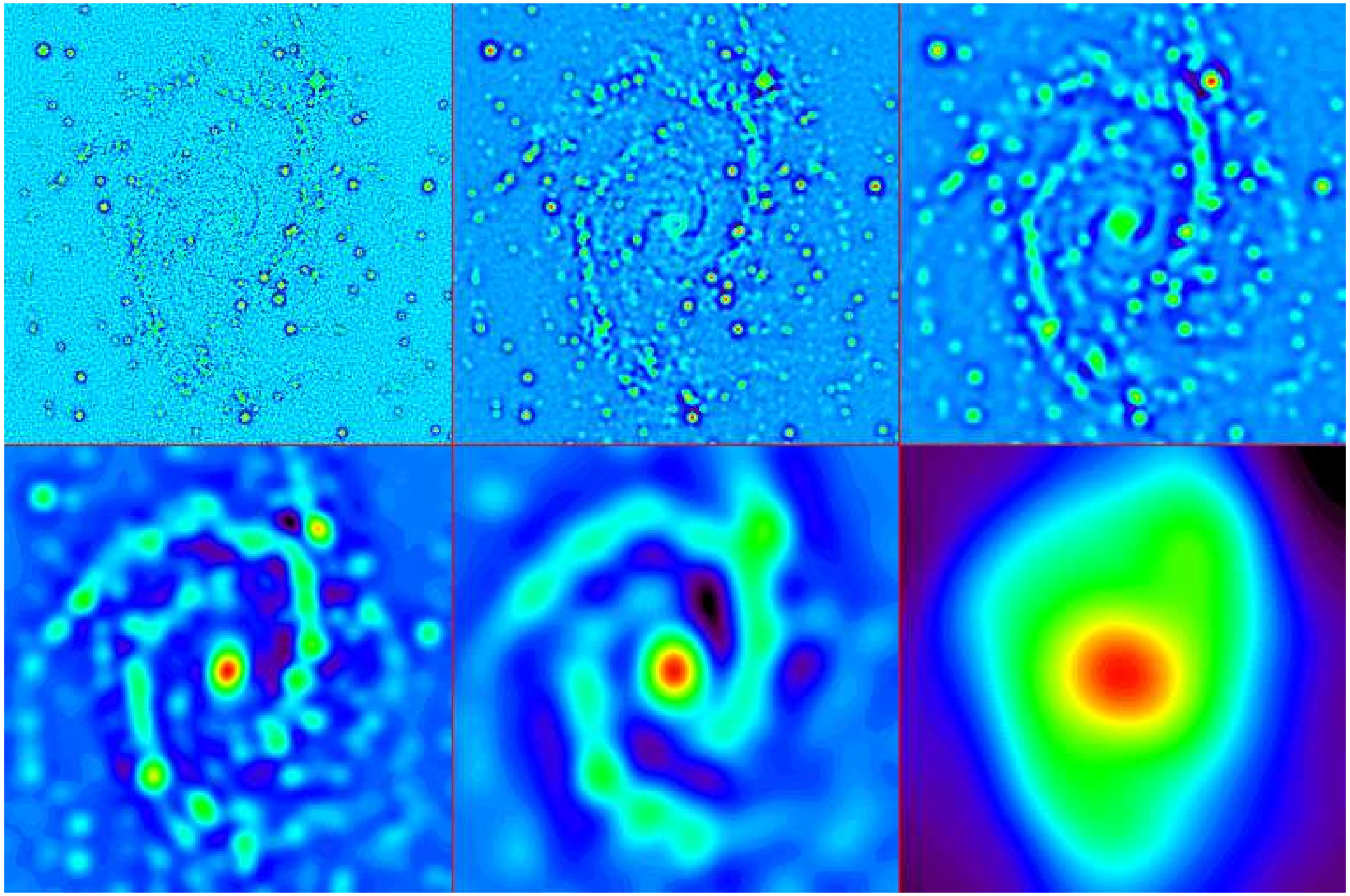
Boundaries

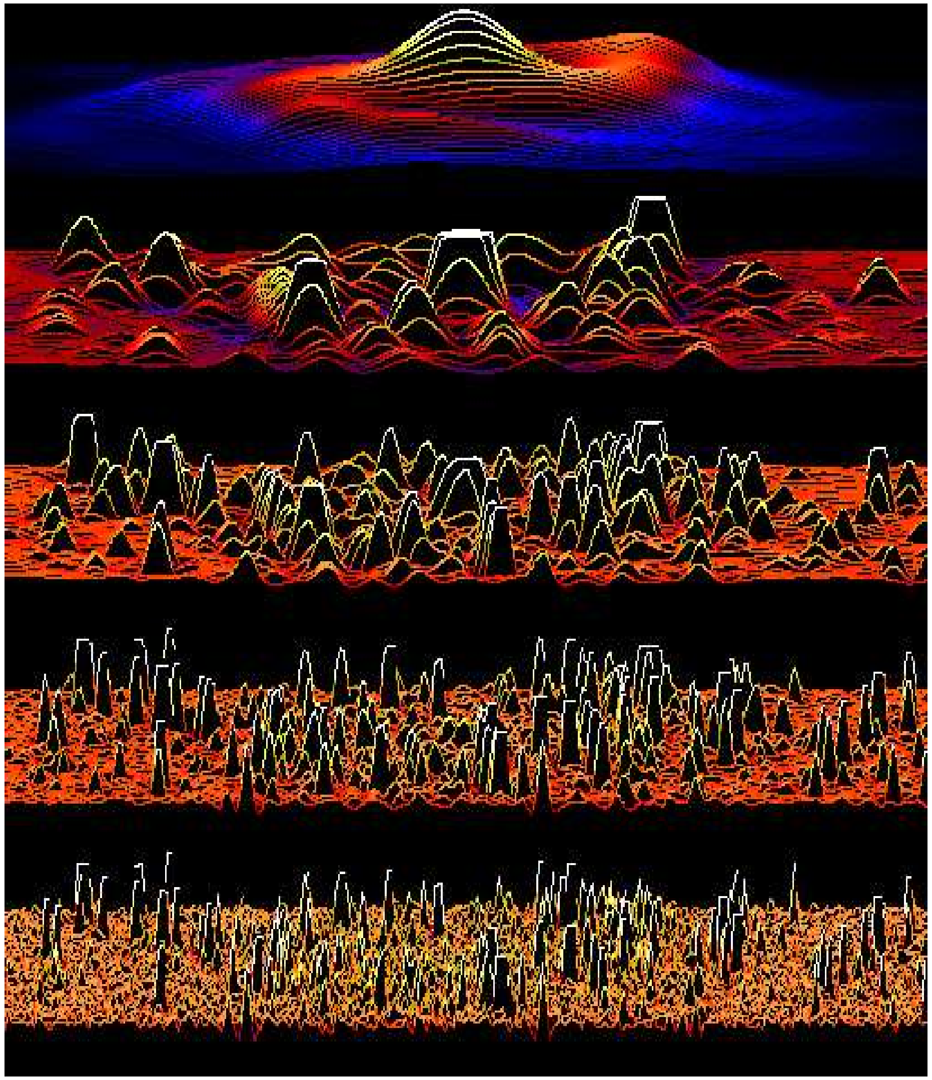
The most general way to handle the boundaries is to consider that $c_{k+N} = c_{N-k}$ (mirror). But other methods can be used such as periodicity ($c_{k+N} = c_k$), or continuity ($c_{k+N} = c_N$).

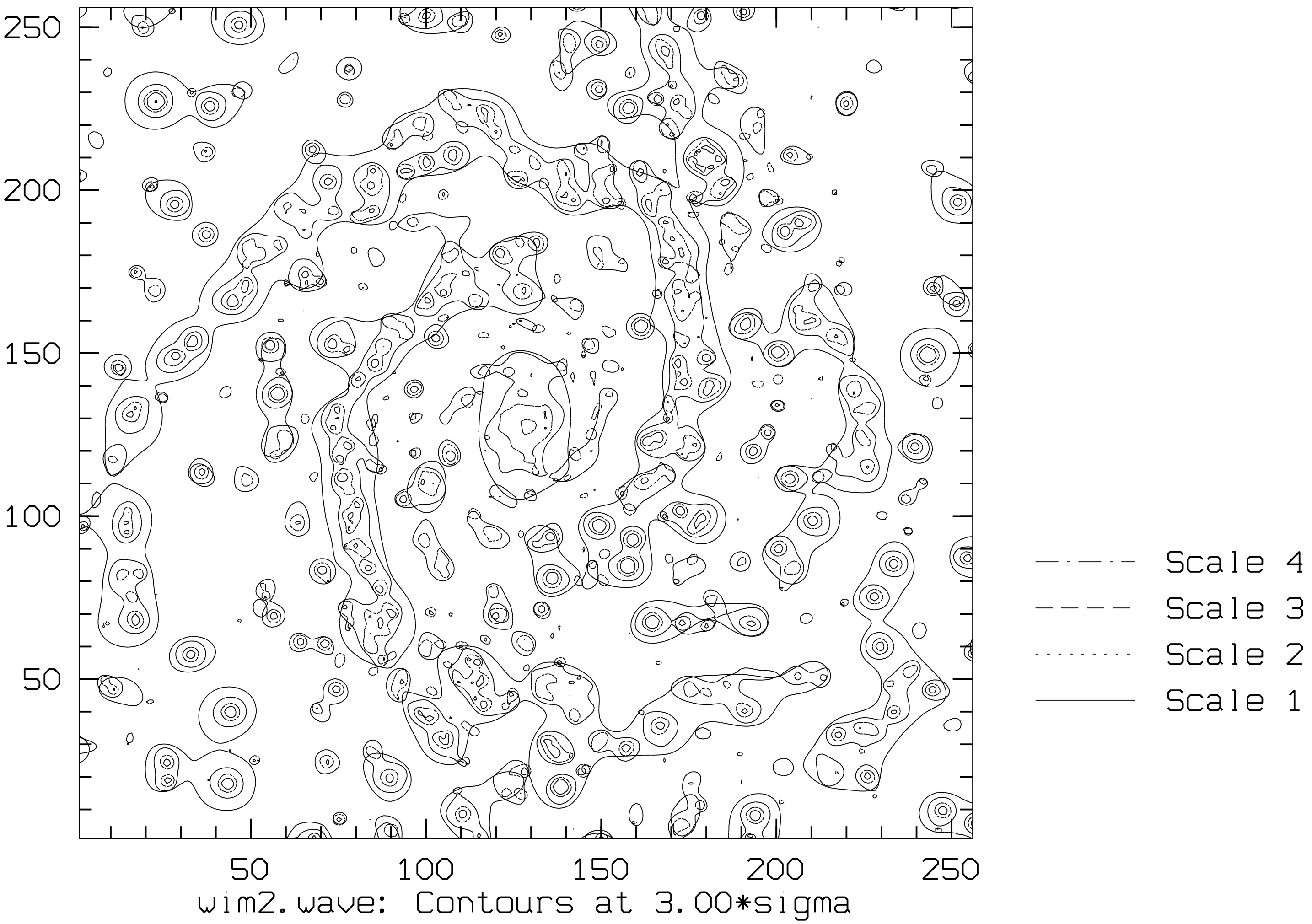
NGC2997

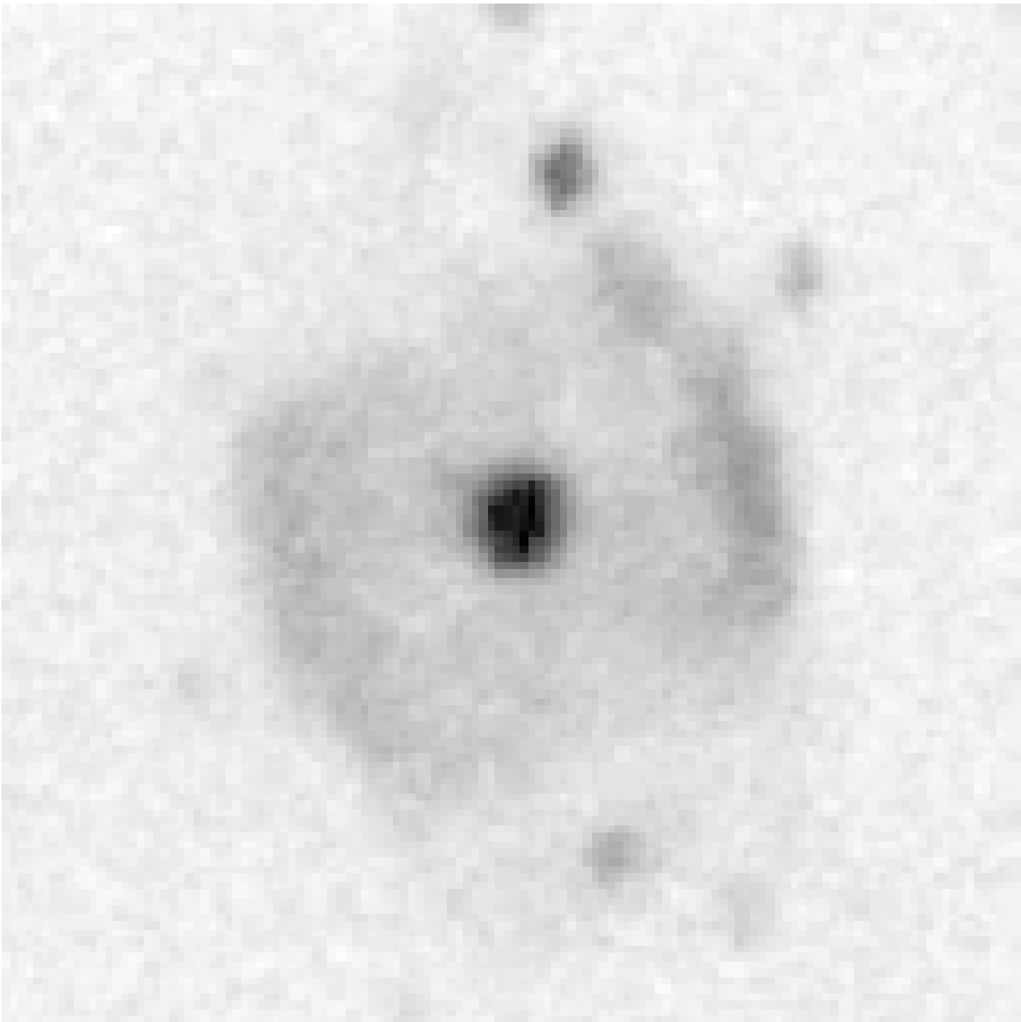


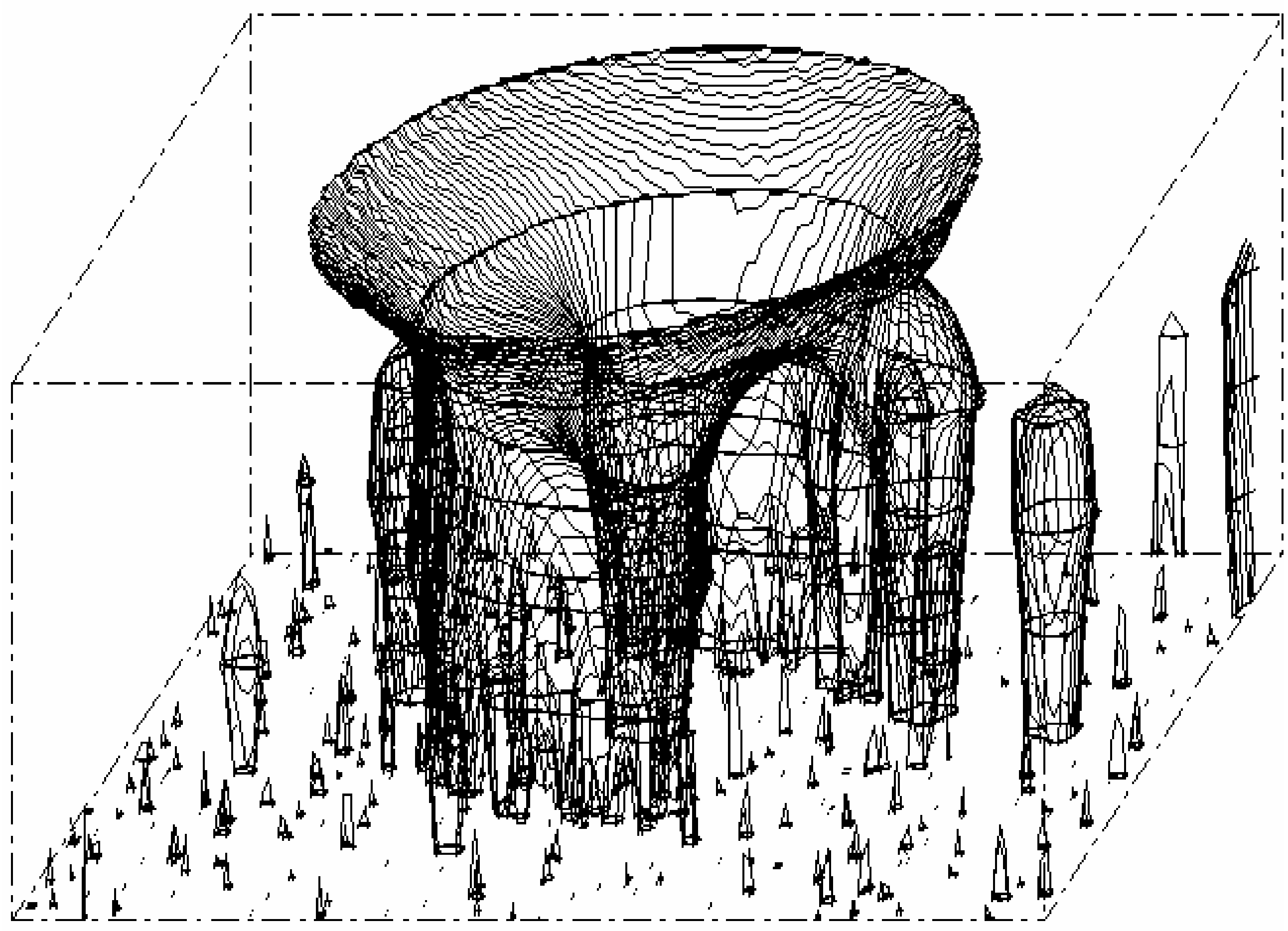
NGC2997 Wavelet Transform



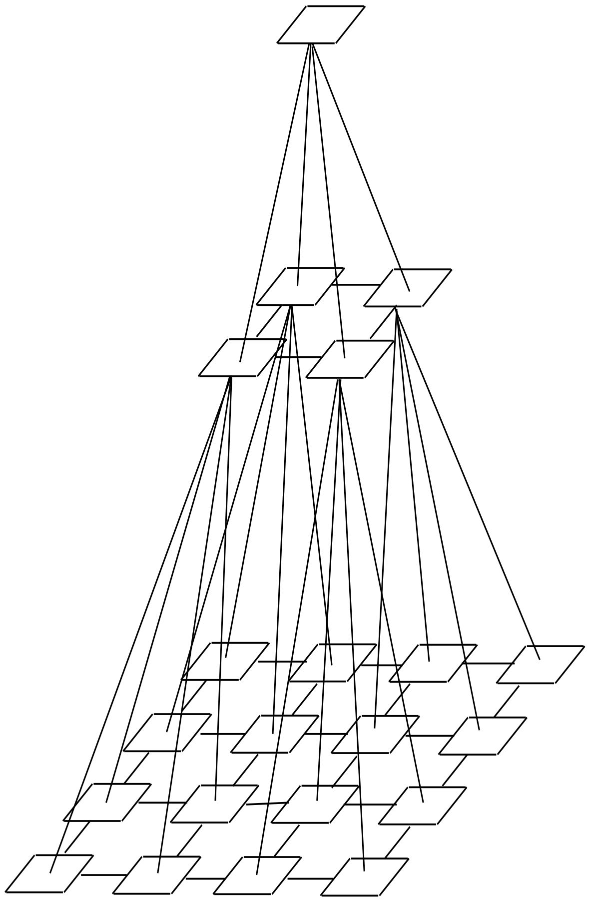



250
200
150
100
50
50
100
150
200
250
wim2.wave: Contours at 3.00*sigma
Scale 4
Scale 3
Scale 2
Scale 1





Pyramidal Algorithm



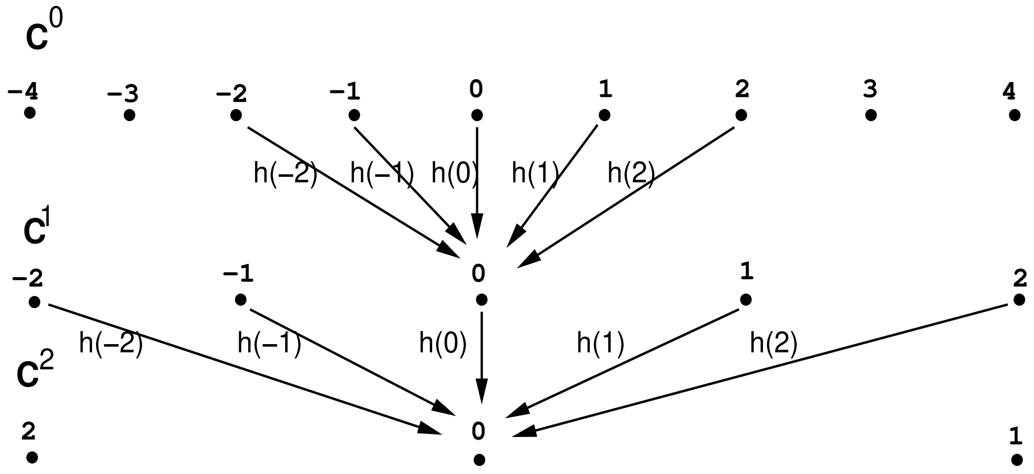
Several algorithms exist:

- The Laplacian Pyramid
- Pyramid with one wavelet
- Pyramidal wavelet transform using the FFT

The Laplacian Pyramid

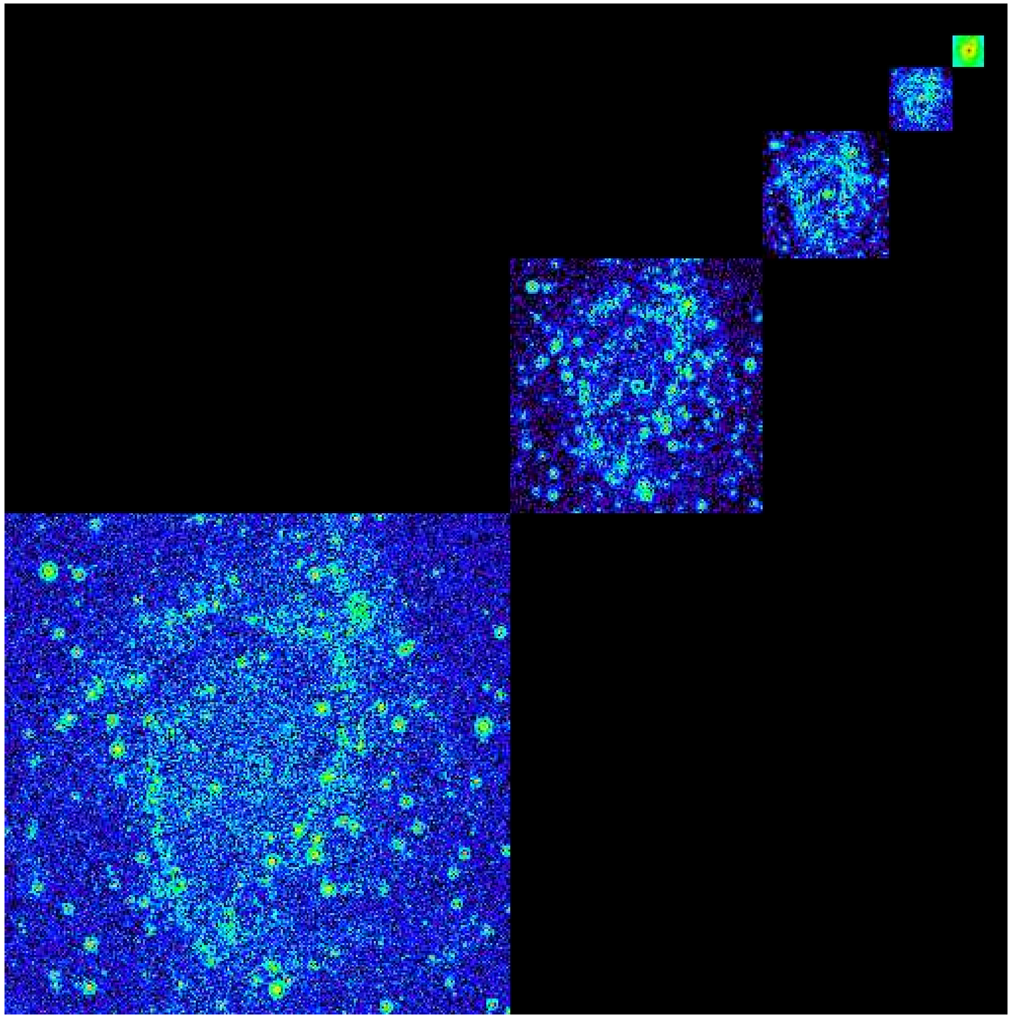
The Laplacian pyramid was developed by Burt and Adelson in 1981 in order to compress images. The convolution is carried out with the filter h by keeping one sample out of two:

$$
\begin{aligned}
c_{j+1,k} &= \sum_l h(l-2k) c_{j,l} \\
w_{j+1,k} &= c_{j,k} - \tilde{c}_{j,k} \\
\tilde{c}_{j,k} &= 2 \sum_l h(k-2l) c_{j+1,k}
\end{aligned}
$$



There is no invariance to translation.

NGC 2997 pyramidal coefficients



Pyramid with one wavelet

$$
\begin{aligned}
\tilde{c}_{j,k} &= \sum_l h(k-l) c_{j,k} \\
w_{j+1,k} &= c_{j,k} - \tilde{c}_{j,k} \\
\tilde{c}_{j,k} &= 2 \sum_l h(k-2l) c_{j+1,k}
\end{aligned}
$$

The reconstruction method is the same as with the Laplacian pyramid, but the reconstruction is not exact. However, the exact reconstruction can be performed by an iterative algorithm.

Wavelet Transform using the Fourier Transform

if $\phi(x)$ has a cut-off frequency $\nu_c \leq \frac{1}{2}$, we can compute the set $c_1(k)$ and $w_1(k)$ from $c_0(k)$ with the discrete filters $\hat{h}(\nu)$ and $\hat{g}(\nu)$, defined by:

$$\hat{h}(\nu) = \begin{cases} \frac{\hat{\phi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 0 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases}$$

$$\hat{g}(\nu) = \begin{cases} \frac{\hat{\psi}(2\nu)}{\hat{\phi}(\nu)} & \text{if } \mid \nu \mid < \nu_c \\ 1 & \text{if } \nu_c \leq \mid \nu \mid < \frac{1}{2} \end{cases}$$

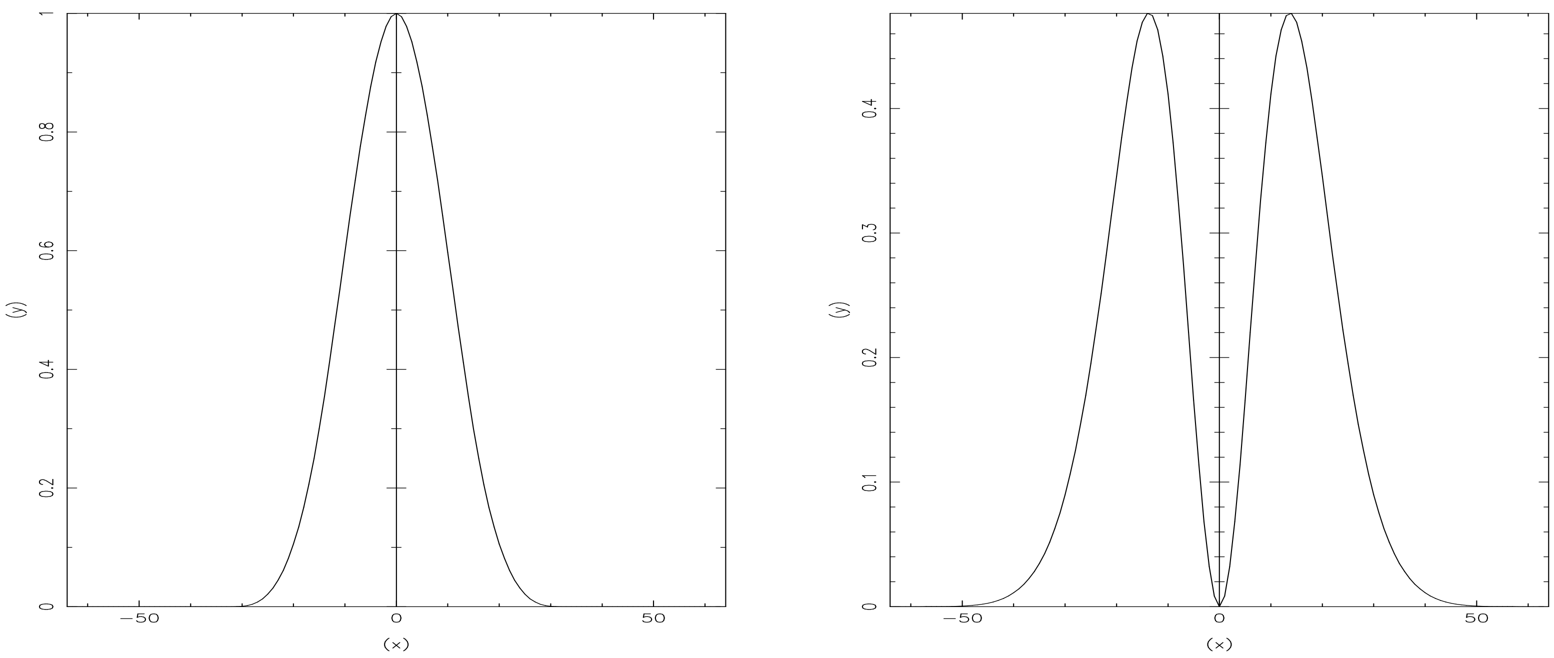
The transformation and the reconstruction are done in the Fourier space:

Transformation:

$$\begin{aligned}\hat{c}_{j+1}(\nu) &= \hat{c}_j(\nu)\hat{h}(2^j\nu)\\ \hat{w}_{j+1}(\nu) &= \hat{c}_j(\nu)\hat{g}(2^j\nu)\end{aligned}$$

Reconstruction:

$$\begin{aligned}\hat{c}_j(\nu) &= \hat{c}_{j+1}(\nu)\hat{\tilde{h}}(2^j\nu) + \hat{w}_{j+1}(\nu)\hat{\tilde{g}}(2^j\nu)\\ \hat{\tilde{h}}(\nu) &= \frac{\hat{h}^*(\nu)}{\mid \hat{h}(\nu)\mid^2 + \mid \hat{g}(\nu)\mid^2}\\ \hat{\tilde{g}}(\nu) &= \frac{\hat{g}^*(\nu)}{\mid \hat{h}(\nu)\mid^2 + \mid \hat{g}(\nu)\mid^2}\end{aligned}$$



On the left, the interpolation function $\hat{\phi}$ *and, on the right, the wavelet* $\hat{\psi}$.

Half pyramidal wavelet transform

At each iteration of the pyramidal transform, there is a smoothing and a decimation of the image. But the smoothing is not strong enough to reduce the cut-off frequency by a factor two. Then the decimation violates Shannon theorem. To avoid this point, which may create artifacts, Bijaoui (1997) proposed:

- not to decimate the image at the first iteration of the algorithm. It means that the two first scales have the same size as the original image.
- to decimate at the following iterations.