Chapter 4

Edge Detection on SAR Images: Multiscale Methods

The main disadvantage of statistical filters is that they strongly depend on the goodness of fit between the data and the statistical model considered by the filter. Then, even though in Section 3.3 we were able to develop the pdf of linearly filtered data (Gamma distributed), the knowledge of such pdf is generally very difficult to have. Furthermore, whereas the pdf in Eq. (3.83) depend on the parameter L alone, more complicated models could involve several parameters, which can be difficult or time-consuming to estimate. In this sense, the simplicity and mathematical treatability of the model is often preferred to formal correctness. Finally, non-linear combination of results, which are obtained by different parameter settings, are sometimes considered to improve the final performance [22]. In this last case, the final threshold is hard to be analytically computed throughout a theoretical model. Consequently, for all the previous reasons, classical linear filters relying on gradient operation are sometimes preferred since they neglect a statistical model of filtered data. Clearly, after applying these filters, the threshold is usually selected either heuristically or by means of data histogram [23]-[25].

Nevertheless, due to the multiplicative nature of the speckle, linear filtering applied to SAR data yields not CFAR result [8]. However, to retrieve such desired behavior, a simple logarithmic transformation can be applied to the input data (see Section 2.2.3) before filtering. Finally, since we want to report the general case of linear filtering, the problem of edge detection at different "scales" will be discussed and addressed. As a consequence, wavelet-based algorithms, which are strongly suggested in literature to improve final performance [26]-[31], will be treated in deep.

4.1 Wavelet and Edge Detection

Before introducing the mathematical formalism behind multiscale edge detectors, it was well known that edge information was linked to the scale concept. For example, in an image that contains a wall, edges can be constituted from the wall boundary. Yet, at finer scale, edges can be found in the brick contour and, decreasing scale again, edges can be due at the brick roughness. Nevertheless, at the beginning of multiscale edge detection, the theory was mainly focused on the combinations of the results at different scales. Parallel at the concept of multiscale edge detection, the mathematical formalism about decomposition of a signal in orthogonal bases with varying time support (wavelets) was getting developed [26], [32]. The first connection about the two fields was given by Mallat in [28], in which it proved how a wavelet can be expressed as the derivative of a smoothing function. Then, since the wavelet transform of a function can be expressed by a convolution, the wavelet transform of a signal is nothing but its derivative after a smoothing filtering. Before reporting this fundamental theorem, the concept of vanishing moments have to be introduced.

A function $\psi(t)$ has n **vanishing moments** if:

$$\int_{-\infty}^{\infty} t^k \psi(t) dt = 0, \qquad k = 0, \cdots, n-1 \tag{4.1}$$

Hence, if $\psi(t)$ has n vanishing moments it is orthogonal to any polynomial of degree $n-1$. Following Mallat's convention, indicating the wavelet transform at scale s and time-shift τ as:

$$\langle f(t), \psi_{\tau,s}(t) \rangle = \int_{-\infty}^{\infty} f(t) \psi_{\tau,s}^{*}(t) dt = Wf(\tau, s) \tag{4.2}$$

with:

$$\psi_{\tau,s}(t) = \frac{1}{\sqrt{s}} \psi\left(\frac{t - s\tau}{s}\right) \tag{4.3}$$

Mallat proved in [26] the following theorem, which is at the base of multiscale edge detector:

Theorem:

A wavelet $\psi(t)$ has n vanishing moments if and only if there exist a function $\theta(t)$ such that:

$$\psi(t) = (-1)^n \frac{d^n \theta(t)}{dt^n} \tag{4.4}$$

Consequently:

$$\langle f(t), \psi_{\tau,s} \rangle = Wf(\tau, s) = s^n \frac{d^n}{dt^n} [f * \bar{\theta}_s] \tag{4.5}$$

where:

$$\bar{\theta}_s(t) = \frac{1}{\sqrt{s}} \theta\left(-\frac{t}{s}\right) \tag{4.6}$$

Moreover, if $\psi(t)$ has exactly n vanishing moments (i.e. no more than n) then:

$$\int_{-\infty}^{\infty} \theta(t) dt \neq 0 \tag{4.7}$$

Among all, the important fact is that a wavelet $\psi(t)$ with n vanishing moments can be expressed as the *n*th order derivative of a function $\theta(t)$ and, consequently, the wavelet transform at scale s can be computed as the derivative of the function $f(t)$ filtered by $\bar{\theta}_s(t)$. As becomes clear from the previous formulas, **when n = 1, computing the wavelet transform is equal to applying an edge detector to the original signal.** Moreover, Mallat proved in [26] that singularity points of a function (i.e. also the edge points) are located at locally maxima points of $|Wf(\tau, s)|$, and the singularity type can be inferred by the decay of $|Wf(\tau, s)|$ in (τ, s) plane, when s is small. Moreover, if $\theta(t)$ is Gaussian (i.e. Canny edge detector), it proved that locally maxima of $|Wf(\tau, s)|$ belongs to connected curves in the plane (τ, s), which are never interrupted when the scale decrease. Anyway, this last theorem holds only for continuous scale parameter s and we lose this property when dyadic wavelet transform is used (i.e. $s = 2^j$ with $j \in \mathbb{Z}$). Anyway, Mallat proved that only from the values of the dyadic wavelet transform $\left|Wf\left(\tau, 2^j\right)\right|$ at the locally maxima points $\left\{\left(\tau, 2^{j_p}\right)\right\}_{p \epsilon N}$ and knowing the mean value of $f(t)$, it is possible to reconstruct the original signal $f(t)$ nearly perfectly. Actually, it can be proved that a family of function (and not only one) share the previous information (i.e. $\left|Wf\left(\tau, 2^j\right)\right|$at $\left\{\left(\tau, 2^{j_p}\right)\right\}_{p \epsilon N}$and the mean value of $f(t)$) and therefore the reconstruction is not perfect. Nevertheless, the functions of this family differ slightly from each other, so that the reconstructed copy is very near from the original one. Although the existence of a family that share the same properties for 2-D functions is still an open problem, a near copy of the original signal can be recovered exploiting inverse wavelet transform. Even in this case, this finding has a deep impact on edge detection. In fact, this means that **not only are edges (at dyadic scales) the most useful information contained in an image, but they are (practically) the whole one**.

4.1.1 Multiscale Edge Detector: 1-D

For the sake of clearness, we report the previous results when wavelets have only one vanishing moment, that is, the multiscale edge detector is derived.

Given a smoothing filter $\theta(t)$, at scale $s = 1$ we have:

$$\psi(t) = -\frac{d\theta(t)}{dt} \tag{4.8}$$

To limit the computation we set $s = \{2^j\}_{j\epsilon\mathbb{Z}}$ (dyadic decomposition). Writing:

$$\psi_{\tau,2^j} = \frac{1}{\sqrt{2^j}}\psi\left(\frac{t-2^j\tau}{2^j}\right) \tag{4.9}$$

the wavelet transform of $f(t)$ (where $f(t) \in L^2(\mathbb{R})$) is given by:

$$Wf(\tau,2^j) = \langle f(t), \psi_{\tau,2^j}(t)\rangle = f * \bar{\psi}_{\tau,2^j} \tag{4.10}$$

where $\bar{\psi}(t) = \psi(-t)$. Now, extending the Eq. (4.8) at the general scale $s = 2^j$ gives:

$$\psi_{\tau,2^j}(t) = -2^j \frac{d\theta_{\tau,2^j}(t)}{dt} \tag{4.11}$$

and putting this last result into Eq. (4.10) gives:

$$Wf(\tau,2^j) = f * \bar{\psi}_{\tau,2^j} = 2^j \frac{d}{dt}\left[f * \bar{\theta}_{\tau,2^j}\right] \tag{4.12}$$

Hence, the wavelet transform modulus is:

$$Mf(\tau,2^j) = \left|Wf(\tau,2^j)\right| = 2^j \frac{d}{dt}\left|f * \bar{\theta}_{\tau,2^j}\right| \tag{4.13}$$

which is proportional to the derivative of the smoothed image. Clearly, if we set unitary L_1-norm at the place of the previous L_2-norm, at scale change the multiplicative constant $1/\sqrt{2^j}$ becomes $1/2^j$, i.e.:

$$\psi_{2^j} = \frac{1}{2^j}\psi\left(\frac{t}{2^j}\right) \tag{4.14}$$

In this way, in Eq. (4.12) the coefficients $Wf(\tau,2^j)$ do not depend on the scale.

4.1.1.1 Example: Multiscale Gaussian Edge Detector

Presupposing the Gaussian smoothing kernel:

$$\theta(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{t^2}{2\sigma^2}} \tag{4.15}$$

the wavelet $\psi(t)$ is given by:

$$\psi(t) = -\frac{d\theta(t)}{dt} = \frac{t}{\sigma^2}\theta(t) = \frac{1}{\sqrt{2\pi\sigma^6}} e^{-\frac{t^2}{2\sigma^2}} \tag{4.16}$$

In Fig. 4.1 the previous functions are reported.

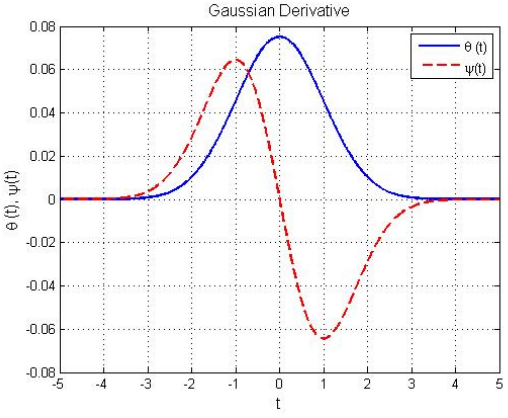
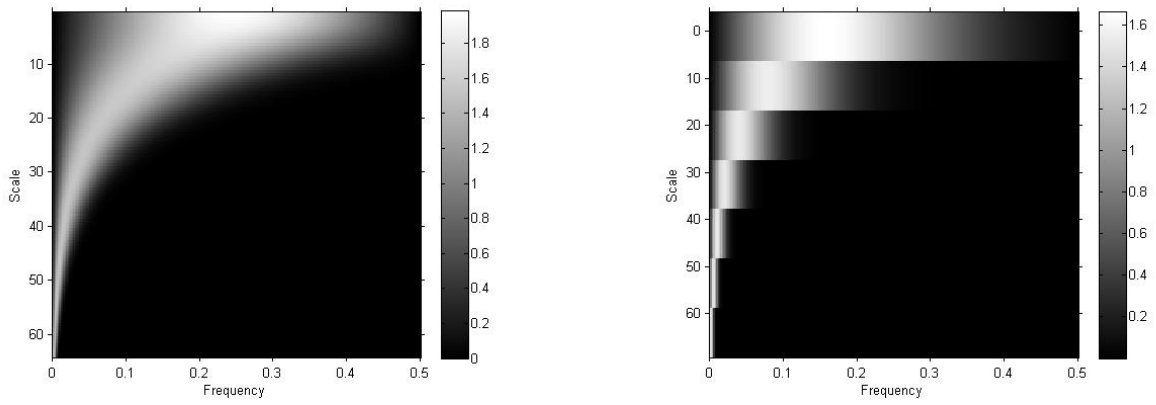


Fig. 4.1 - Gaussian smoothing and its derivative

In Fig. 4.2(a) we can see the frequency response of the wavelet filter at several scales. As can be seen from this figure, wavelets are nothing but **pass band filters** whose **frequency band goes down logarithmically increasing the scale**. It should be noticed that the scale decomposition is not a dyadic (orthogonal) decomposition. In fact, it is decomposed as $s = \{2^{j/v}\}_{(j,v)\epsilon\mathbb{Z}}$, with $0 \leq j \leq 6$ and $v = 16$, i.e. 7 scales (also called octaves) with 16 voices. Clearly, this last decomposition approximates well a continuous scale parameter. For the sake of comparison, in Fig. 4.2 (b) the frequency response for dyadic scales is reported. This figure makes clear how each dyadic scale manages to filter only a certain frequency band, differently from the "continuous" case where redundancy is introduced by nearly totally overlapping frequencies between subsequent scales.

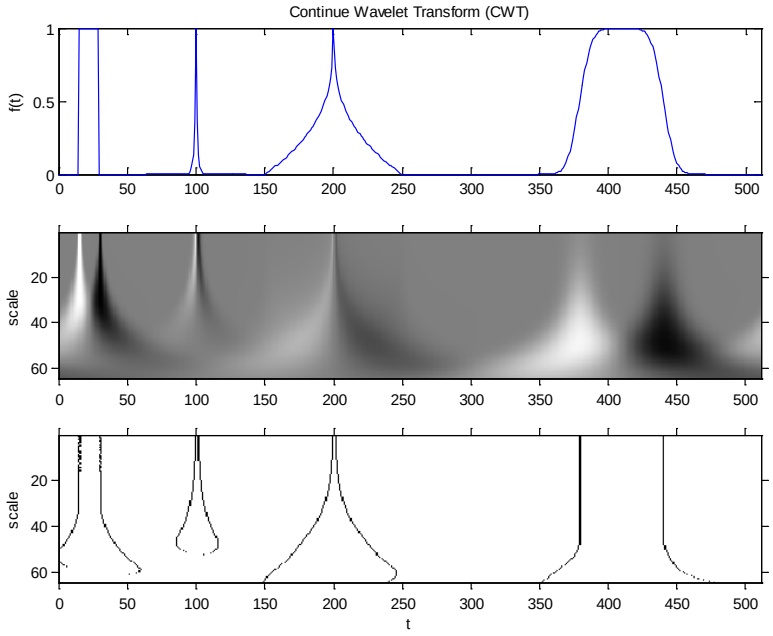
Nevertheless, as can be seen from these figures, the first scale does not cover the highest frequency components of the signal. This fact is due to the design choice of starting from the scale in which all filter coefficients are $h = \{-1,0,1\}$ at the place of $h = \{-1,1\}$ that is indicated by wavelet theory. Clearly, with this choice some information about high frequency components is lost (e.g. a chessboard edge with a frequency of exactly 0.5 would not be revealed by filters), but in this manner filter is less influenced by noise. Anyway, filtering with the latter finest scale can be added without any effort to the filter.

In Fig. 4.3(a)-(b) the result of the Multiscale Gaussian edge detector on a signal with several singularities is reported, respectively for continue and dyadic wavelet transform. From the non-maxima suppression result of both cases (reported at the bottom), we can notice the well-known **bias on edge position at large scales** of Canny (Gaussian) edge detector.

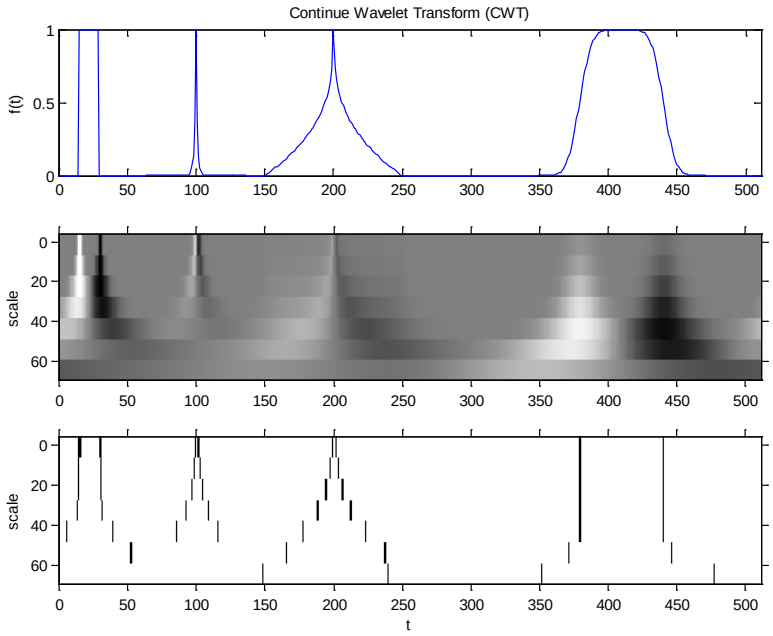


(a) (b)

Fig. 4.2 - Frequency response of the wavelet function (Gaussian derivative) at each scale. (a) Continue scale decomposition. (b) Dyadic scale decomposition



(a)



(b)

Fig. 4.3 - Multiscale edge detector result (bottom) when non-maxima suppression is applied at $Wf(\tau, 2^j)$ (in the middle) for a signal with several singularities (top). (a) Continue wavelet transform. (b) Dyadic wavelet transform.

4.1.2 Multiscale Edge Detector: 2-D

Extending the previous general results for 2-D signals, we can easily derive the 2-D Multiscale edge detector. Given a smoothing filter $\theta(\boldsymbol{x}) = \theta(x_1, x_2)$, at scale $s = 1$ we have:

$$\begin{cases} \psi^1(\boldsymbol{x}) = -\dfrac{\partial\theta(\boldsymbol{x})}{\partial x_1} \\ \psi^2(\boldsymbol{x}) = -\dfrac{\partial\theta(\boldsymbol{x})}{\partial x_2} \end{cases} \tag{4.17}$$

To limit the computation we set $s = \{2^j\}_{j\epsilon\mathbb{Z}}$ (dyadic decomposition). Writing :

$$\psi^k_{\boldsymbol{\tau},2^j}(\boldsymbol{x}) = \frac{1}{2^j}\psi^k_{\boldsymbol{\tau},2^j}\left(\frac{\boldsymbol{x} - 2^j\boldsymbol{\tau}}{2^j}\right);\ \ k = 1{,}2 \tag{4.18}$$

the wavelet transform of $f(\boldsymbol{x})$ (where $f(\boldsymbol{x})\ \epsilon\ L^2(\mathbb{R}^2)$) is given by 2 components:

$$W^k f(\boldsymbol{\tau}, 2^j) = \langle f(\boldsymbol{x}), \psi^k_{\boldsymbol{\tau},2^j}(\boldsymbol{x})\rangle = f * \bar{\psi}^k_{\boldsymbol{\tau},2^j};\quad k = 1{,}2; \tag{4.19}$$

where $\bar{\psi}(\boldsymbol{x}) = \psi(-\boldsymbol{x})$ and the convolution is over the two dimensions (x_1, x_2). Now, extending the Eq. (4.17) at the general scale $s = 2^j$ gives:

$$\begin{cases} \psi^1_{\boldsymbol{\tau},2^j}(\boldsymbol{x}) = -2^j\dfrac{\partial\theta_{\boldsymbol{\tau},2^j}(\boldsymbol{x})}{\partial x_1} \\ \psi^2_{\boldsymbol{\tau},2^j}(\boldsymbol{x}) = -2^j\dfrac{\partial\theta_{\boldsymbol{\tau},2^j}(\boldsymbol{x})}{\partial x_2} \end{cases} \tag{4.20}$$

and putting this last result into Eq. (4.19) gives:

$$W^k f(\boldsymbol{\tau}, 2^j) = 2^j\frac{\partial}{\partial x_k}\left[f * \bar{\theta}_{\boldsymbol{\tau},2^j}\right];\quad k = 1{,}2; \tag{4.21}$$

From Eq. (4.21) we can write:

$$\begin{pmatrix} W^1 f(\boldsymbol{\tau}, 2^j) \\ W^2 f(\boldsymbol{\tau}, 2^j) \end{pmatrix} = 2^j\begin{pmatrix} \dfrac{\partial}{\partial x_1}\left[f * \bar{\theta}_{\boldsymbol{\tau},2^j}\right] \\ \dfrac{\partial}{\partial x_2}\left[f * \bar{\theta}_{\boldsymbol{\tau},2^j}\right] \end{pmatrix} = 2^j\,\nabla\left[f * \bar{\theta}_{\boldsymbol{\tau},2^j}\right] \tag{4.22}$$

The modulus of the wavelet components is:

$$Mf(\boldsymbol{\tau}, 2^j) = \sqrt{|W^1 f(\boldsymbol{\tau}, 2^j)|^2 + |W^2 f(\boldsymbol{\tau}, 2^j)|^2} = 2^j\ \left|\nabla\left[f * \bar{\theta}_{\boldsymbol{\tau},2^j}\right]\right| \tag{4.23}$$

which is proportional to the gradient modulus of the smoothed image. Moreover, the angle:

$$Af(\boldsymbol{\tau}, 2^j) = \tan^{-1}\left(\frac{W^2 f(\boldsymbol{\tau}, 2^j)}{W^1 f(\boldsymbol{\tau}, 2^j)}\right) \tag{4.24}$$

indicates the direction of non-maxima suppression algorithm. It should be noticed that in 2-D, differently from 1-D, normalizing the L^2-norm is the correct choice to have $Mf(\boldsymbol{\tau}, 2^j)$ scale-independent.

4.1.2.1 Example: Multiscale Gaussian Edge Detector

Presupposing separable Gaussian smoothing function:

$$\theta(\boldsymbol{x}) = \theta(x_1)\theta(x_2) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x_1+x_2)^2}{2\sigma^2}} \tag{4.25}$$

we have:

$$\begin{cases} \psi^1(\boldsymbol{x}) = -\dfrac{\partial\theta(\boldsymbol{x})}{\partial x_1} = \psi(x_1)\theta(x_2) = \dfrac{x_1}{2\pi\sigma^4} e^{-\frac{(x_1+x_2)^2}{2\sigma^2}} \\ \psi^2(\boldsymbol{x}) = -\dfrac{\partial\theta(\boldsymbol{x})}{\partial x_2} = \theta(x_1)\psi(x_2) = \dfrac{x_2}{2\pi\sigma^4} e^{-\frac{(x_1+x_2)^2}{2\sigma^2}} \end{cases} \tag{4.26}$$

The frequency responses of the wavelet $\psi^1(\boldsymbol{x})$, which catches the HL (High in along f_{x_1} and Low along f_{x_2}) frequency part of the spectrum, and of the wavelet $\psi^2(\boldsymbol{x})$, which catches the LH part, are reported in .

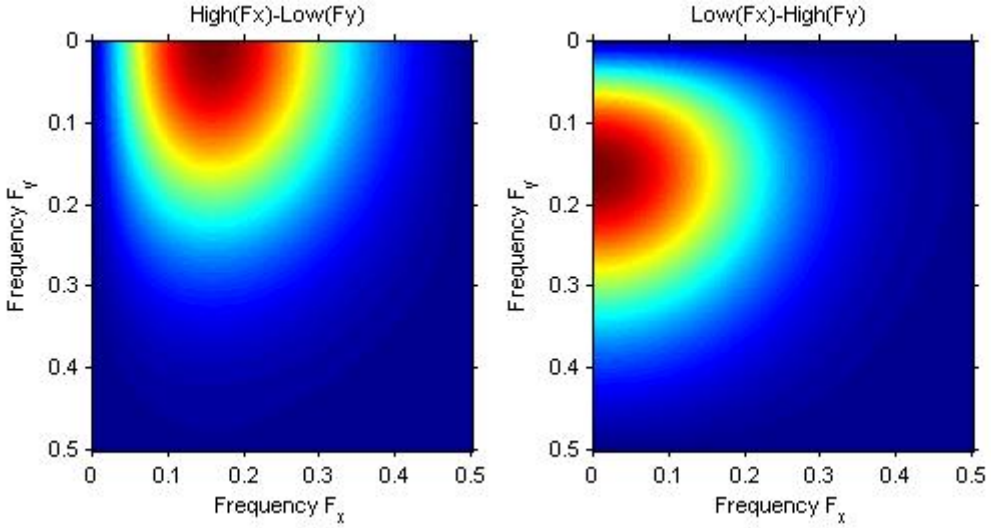


Fig. 4.4 - Frequency response of 2-D wavelet functions (Gaussian derivative): $\psi^1(\boldsymbol{x})$ (on the left), $\psi^2(\boldsymbol{x})$ (on the right). It should be noticed that the 2-D variables are here indicated as x and y.

Nevertheless, as can be understood from the decomposition theory, also the HH frequency part is needed to provide a perfect reconstruction. This means that, if the following wavelet is not considered:

$$\psi^3(\boldsymbol{x}) = \frac{\partial^2\theta(\boldsymbol{x})}{\partial x_1 \partial x_2} = \psi(x_1)\psi(x_2) \tag{4.27}$$

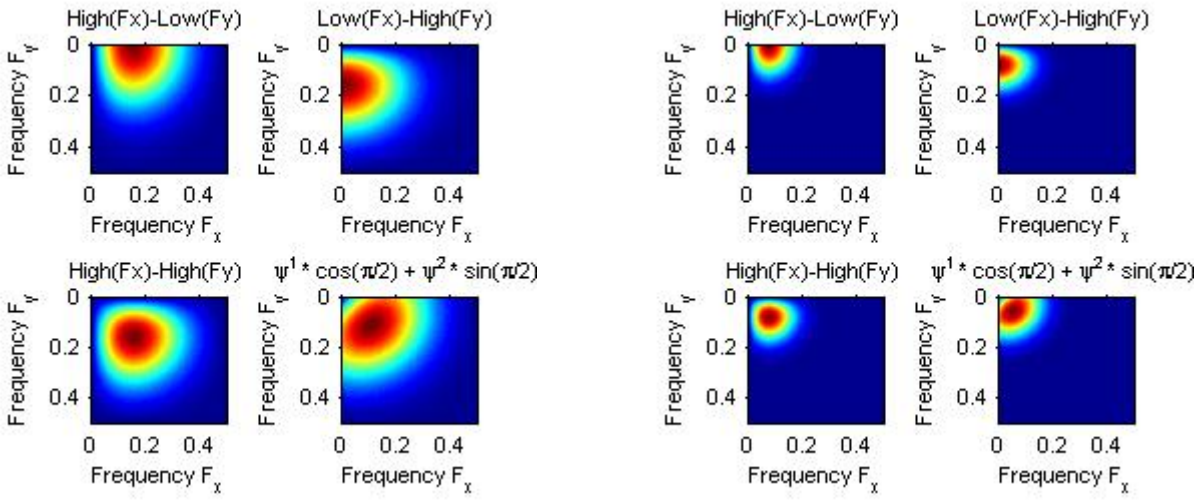
some information is lost in the decomposition process. In particular, the HH components represents a quickly variation of the image $f(\boldsymbol{x})$ along columns and rows jointly, i.e. the presence of a corner or a diagonal edge. Anyhow, presupposing the image $f(\boldsymbol{x})$ differentiable, the HH frequency components can be retrieved by a linear composition of previous two wavelets:

$$\psi^\alpha(\boldsymbol{x}) = \psi^1(\boldsymbol{x})\cos\alpha + \psi^2(\boldsymbol{x})\sin\alpha \tag{4.28}$$

In fact, given the gradient $\nabla f(\boldsymbol{x})$ of the image $f(\boldsymbol{x})$, the directional derivative along the direction $\hat{\boldsymbol{v}} = (\cos\alpha, \sin\alpha)$ is given by:

$$\nabla f(\boldsymbol{x}) \cdot \hat{\boldsymbol{v}} = \frac{\partial f(\boldsymbol{x})}{\partial x_1}\cos\alpha + \frac{\partial f(\boldsymbol{x})}{\partial x_1}\sin\alpha \tag{4.29}$$

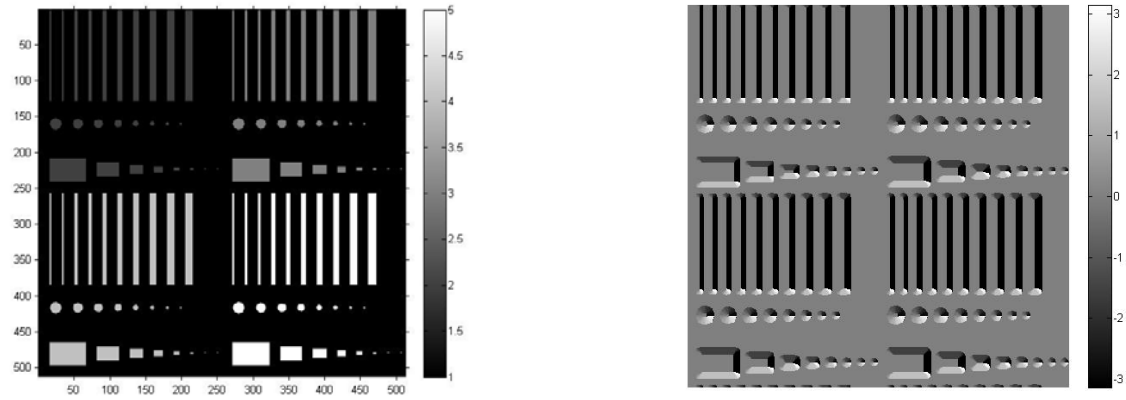
Anyway, it is fundamental to know that Eq. (4.28) (and consequently Eq. (4.29)) is valid only if the original image $f(\boldsymbol{x})$ is differentiable. Clearly, this requirement is a very strong request of regularity and it is surely not respected on SAR images for initial decomposition scales. In Fig. 4.5 the frequency response of the wavelet $\psi^1(\boldsymbol{x})$, $\psi^2(\boldsymbol{x})$, $\psi^3(\boldsymbol{x})$ and$\psi^\alpha(\boldsymbol{x})$ (with $\alpha = 45°$), are plotted for the initial two dyadic scales. As can be seen comparing the frequency responses of $\psi^3(\boldsymbol{x})$ and $\psi^\alpha(\boldsymbol{x})$ (with $\alpha = 45°$), only a small HH part cannot be totally reconstructed by this approximation. In particular, to the aim of edge detection, if only $\psi^1(\boldsymbol{x})$, $\psi^2(\boldsymbol{x})$ are used, we lost some information about diagonal edges at the initial (i.e. small) decomposition scales. Nevertheless, as will be shown subsequently in the document, the initial scales of SAR images are very noisy, so that this information loss could be considered nearly negligible.



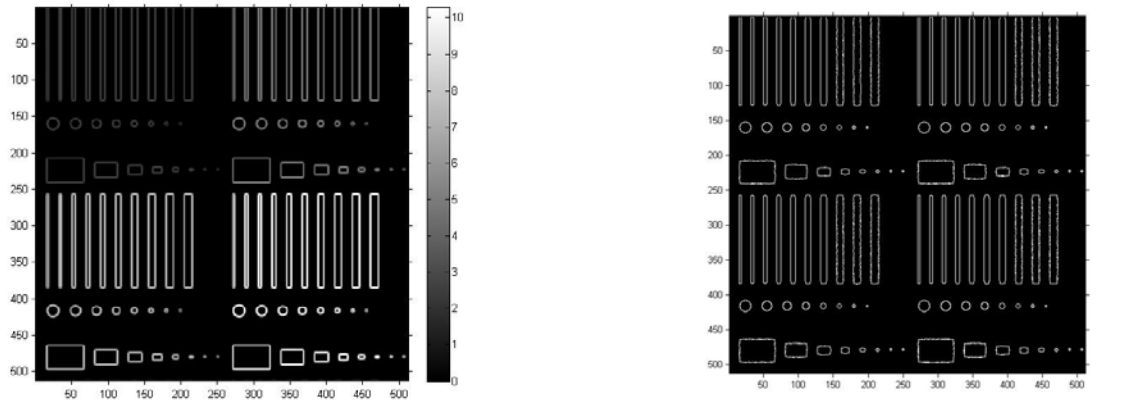
(a) (b)

Fig. 4.5- Frequency response of 2-D wavelet functions (Gaussian derivative) for several dyadic scales. (a) $s = 2^0$. (b) $s = 2^1$.

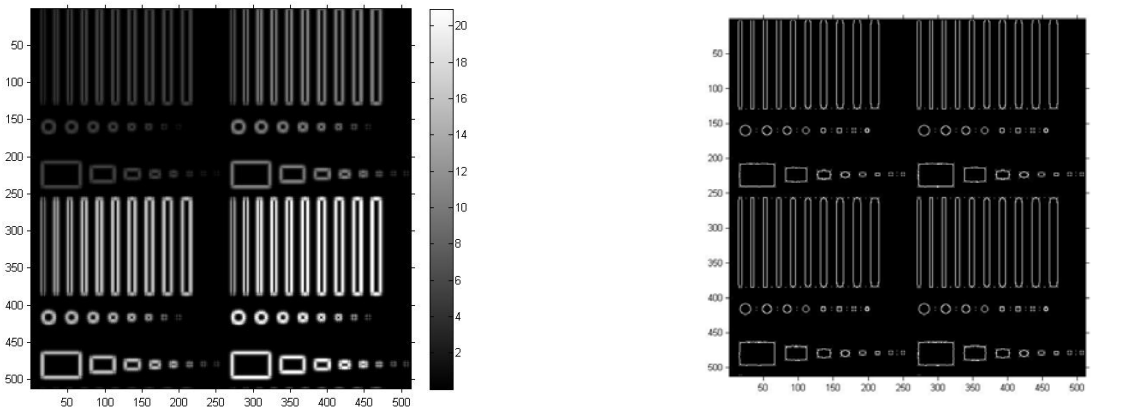
In Fig. 4.6(c),(d) gradient module $Mf(\boldsymbol{\tau}, 2^j)$ and its non-maxima suppression for $s = \{2^0, 2^1\}$ are reported. The original image is in Fig. 4.6 (a) and the angle $Af(\boldsymbol{\tau}, 2^j)$ for the scale $s = 1$ is in Fig. 4.6 (b).



(a) (b)



(c)



(d)

Fig. 4.6 - Multiscale Edge detection Result. (a) Original image. (b) Angle $Af(\tau, 2^j)$ for the smallest scale $s = 1$. (c) Gradient module $Mf(\tau, 2^j)$ (on the left) and its non-maxima suppression (on the right) for $s = 1$. (d) Gradient module $Mf(\tau, 2^j)$ (on the left) and its non-maxima suppression (on the right) for $s = 2$.

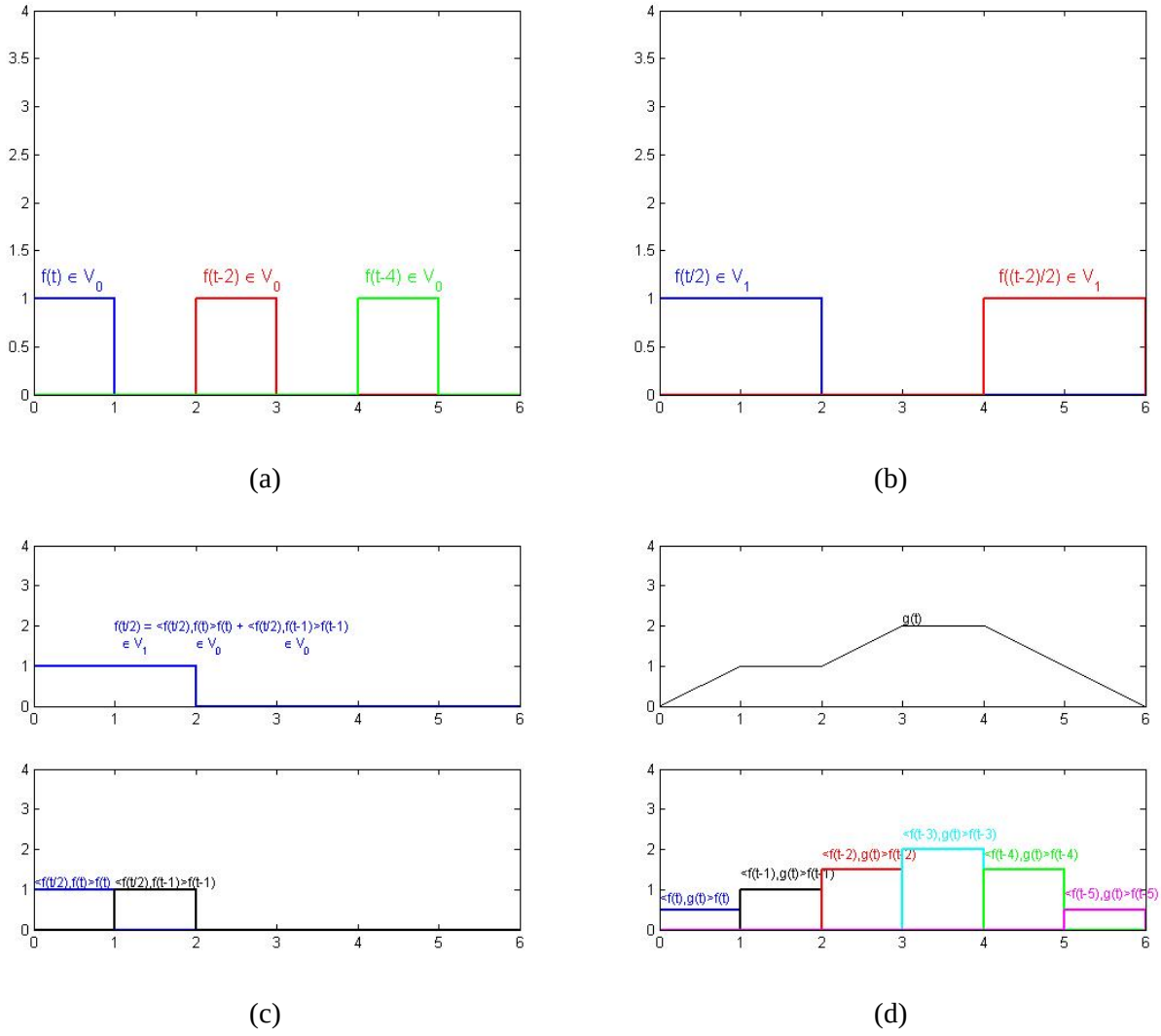
4.2 Multiresolution Analysis

In literature, wavelet concepts are usually presented with the scope of multiresolution analysis of signals. From the mathematical point of view, a formal definition of multiresolution approximation can be obtained in an axiomatic way [26]. A sequence $\{V_j\}_{j\in\mathbb{Z}}$ of closed subspaces of $L^2(\mathbb{R})$ (space of square integrable functions) is a multiresolution approximation if the following 6 properties hold:

1. $\forall(n,j)\in\mathbb{Z}^2,\ f(t)\in V_j \Leftrightarrow f(t-2^j n)\in V_j$, i.e. V_j is invariant by any translation proportional to the scale 2^j.
2. $\forall j\in\mathbb{Z},\ f(t)\in V_j \Leftrightarrow f(t/2)\in V_{j+1}$, i.e. a dilatation of $f(t)$ makes change the approximation scale.
3. $\forall j\in\mathbb{Z},\ V_{j+1}\subset V_j$, i.e. an approximation of $f(t)$ at scale 2^j contains all information to reconstruct the signal at coarser scale 2^{j+1}.
4. $\lim_{j\to\infty} V_j=\{0\}$, i.e. if the scale goes to infinite (the resolution goes to 0) we lost all information.
5. $\lim_{j\to-\infty} V_j=L^2(\mathbb{R})$, i.e. if the scale goes to 0 (the resolution goes to infinite) the approximation converges to the original signal.
6. There exist $\vartheta(t)$ such that $\{\vartheta(t-n)\}_{n\in\mathbb{Z}}$ is a Riesz Basis of V_0, i.e. any signal $f(t)\in V_0$ can be represented as $f(t)=\sum_{n=-\infty}^{\infty}\langle f(t),\vartheta(t-n)\rangle\vartheta(t-n)=\sum_{n=-\infty}^{\infty}a[n]\vartheta(t-n)$. For the sake of clearness, it should be noticed that a Riesz Basis is a general basis of linearly independent signals that avoids the growth to infinite of the energy coefficients ($\sum_{n=-\infty}^{\infty}|a[n]|^2<+\infty$).

The property 1 is visualized in Fig. 4.7(a) for piecewise constant functions belonging to the space V_0. As can be seen from this figure, the depicted signal translations belong to the same space of the original signal $f(t)$. The property 2 is reported in Fig. 4.7 (b) where can be noticed a scale increment by stretching of lower scale functions . InFig. 4.7 (c) we can see how a function of the upper scale can be recovered by composition of lower scale signals (property 3). Finally, in Fig. 4.7 (d) we can see how a general continuous function $g(t)$ can be decomposed by piecewise constant signals $f\in V_0$.

Another example can be provided taking the function $\vartheta(t)=sinc(t)=\sin(\pi t)/\pi t$, which is a function whose translation $\{\vartheta(t-n)\}_{n\in\mathbb{Z}}$ provides a particular class of Riesz Basis, i.e. an orthonormal basis of V_0, with V_0 the space of functions $f(t)$ whose normalized frequency support B is in $B\in[-1/2\ \ 1/2]$. Clearly, the generic family $\{\vartheta_{j,n}(t)=2^{-j/2}\vartheta(2^{-j}t-n)\}_{n\in\mathbb{Z}}$ provides an orthonormal basis for the generic space V_j of functions whose normalized frequency support B is included in $B\in[-1/2^{j+1}\ \ 1/2^{j+1}]$. It should be noticed that a function belonging to a space V_j has a multiplication factor of $2^{-j/2}$ in order to preserve unitary L_2-norm of each scale.



(a) (b)

(c) (d)

Fig. 4.7 - Multiresolution analysis. (a) Property 1. (b) Property 2. (c) Property 3. A function of the upper scale (top) can be recovered by composition of lower scale signals (bottom). (d) Decomposition of the signal $g(t)$ (top) by piecewise constant signals $f \in V_0$ (bottom)

Usually, when the Riesz Basis is formed by orthonormal functions, the translated and dilated function is called *scaling function* $\phi(t)$. Since from property 3 of the multiresolution approximation we know that $V_{j+1} \subset V_j$, a signal which belongs to V_{j+1} can expressed by the basis of the space V_j. Hence, the function $2^{-j/2}\,\phi(t/2)$ belonging to V_1 can be expressed by $\{\phi(t-n)\}_{n\in\mathbb{Z}}$, i.e. the bases of the space V_0:

$$\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} \langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-n)\rangle\, \phi(t-n) = \sum_{n=-\infty}^{\infty} h_0[n]\phi(t-n) \tag{4.30}$$

where the inner product is used to compute the projection along basis components. Moreover, defining W_j as an orthogonal complement of W_j in W_{j-1} (in formula $V_{j-1} = V_j \oplus W_j$)[2], we can prove that an orthonormal basis of W_j is formed by translations and dilatations of a *wavelet function* $\psi(t)$, $\{\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n)\}_{n\in\mathbb{Z}}$.

Roughly speaking, through the use of a wavelet function we can recover a lower scale function by an upper scale function (which contains less information than the upper scale counterpart), see Fig. 4.8.

[2] Any vector in $W_j \subset V_{j-1}$ is orthogonal to any vector in V_j.

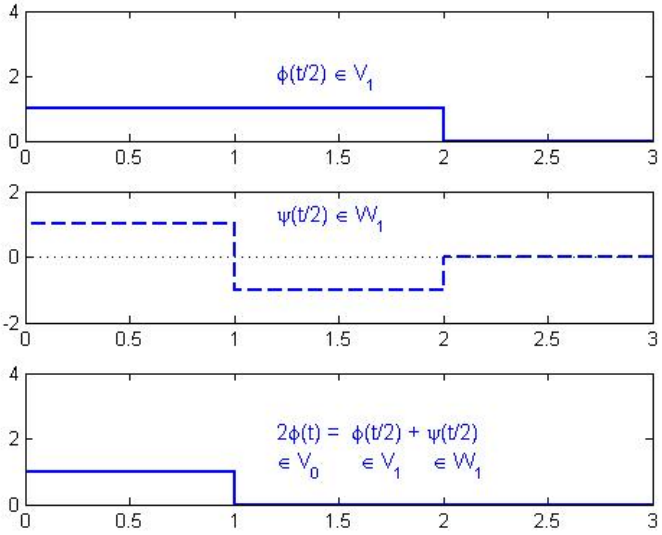


Fig. 4.8 - Wavelet function as orthogonal complement of scaling function, i.e. $V_{j-1} = V_j \oplus W_j$.

As previously, from subspaces property $W_j \subset V_{j-1}$ (a wavelet function $\psi_{j,n}(t) \in W_j$ can be recovered by composition of lower scale scaling functions $\phi_{j-1,n}(t) \in V_{j-1}$) we have:

$$\frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right) = \sum_{n=-\infty}^{\infty} \langle \frac{1}{\sqrt{2}}\psi\left(\frac{t}{2}\right), \phi(t-n)\rangle \, \phi(t-n) = \sum_{n=-\infty}^{\infty} h_1[n]\phi(t-n) \tag{4.31}$$

In order to study the frequency behaviour of the discrete filters $h_0[n]$ and $h_1[n]$, we can apply the FT respectively to Eq. (4.30) and (4.31). Anyway, since both discrete and continuous time signals are involved, we should pay attention about the correct FT to use. In particular, we should express the discrete signal as its continuous version multiplied by a Dirac comb $c(t) = \sum_{n=-\infty}^{\infty} \delta(t-n)$ before transforming it by the continuous FT. For example, the FT of the right term of Eq. (4.30) becomes:

$$\begin{aligned} &\mathfrak{F}\left[\sum_{n=-\infty}^{\infty} h_0(t)c(t)\phi(t-n)\right] = \mathfrak{F}[h_0(t)c(t) * \phi(t)] \\ &= \mathfrak{F}[h_0(t)c(t)]\,\mathfrak{F}[\phi(t)] = \left(H_0(f) * \sum_{k=-\infty}^{\infty} \delta(t-k)\right)\Phi(f) \\ &= \sum_{k=-\infty}^{\infty} H_0(f-k)\Phi(f) \end{aligned} \tag{4.32}$$

where the convolution of $H_0(f)$ with the Dirac comb FT $C(f) = \sum_{k=-\infty}^{\infty} \delta(f-k)$ indicates its periodicity of period equal to $f = 1$ in normalized frequency. Note that the whole transform is not periodic since the periodicity of $\sum_{k=-\infty}^{\infty} H_0(f-k)$ is deleted by the multiplication with the transform $\Phi(f)$[3]. Hence, applying the continuous time FT at Eq. (4.30) we have:

$$\sqrt{2}\Phi(2f) = H_0(f)\Phi(f) \tag{4.33}$$

and, exploiting multiresolution properties:

[3] Note that, being $h_0(t)c(t)$equal to $h_0(t)c(t) = \langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-n)\rangle \, c(t) = \left(\frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right) * \phi^*(-t)\right)c(t)$, its FT is: $\mathfrak{F}[h_0(t)c(t)] = H_0(f) * C(f) = \left(\sqrt{2}\Phi(2f)\Phi^*(f)\right) * \sum_{k=-\infty}^{\infty} \delta(f-k)$ therefore $\sum_{k=-\infty}^{\infty} H_0(f-k)$ is the sums of $\left(\sqrt{2}\Phi(2f)\Phi^*(f)\right)$ translated in frequency.

$$\begin{cases} |H_0(f)|^2 + \left|H_0\left(f+\frac{1}{2}\right)\right|^2 = 2 \\ H_0(f) = \sqrt{2} \end{cases} \tag{4.34}$$

where the first term is the Eq. (4.34) inserted in the **orthonormal relation** of the scaling functions given in Fourier space :

$$\langle \phi(t-n), \phi(t-p)\rangle = \delta[p-n] \quad \overset{FT}{\rightarrow} \quad \sum_{k=-\infty}^{\infty} |\Phi(f-k)|^2 = 1 \tag{4.35}$$

whereas the second term is derived from Eq. (4.33) inserting the equality $\Phi(0) = 1$, (equivalent to $\int_{-\infty}^{\infty} \phi(t)dt = 1$). Clearly, applying the same previous operations to Eq. (4.31) we have:

$$\sqrt{2}\Psi(2f) = H_1(f)\Psi(f) \tag{4.36}$$

and:

$$\begin{cases} |H_1(f)|^2 + \left|H_1\left(f+\frac{1}{2}\right)\right|^2 = 2 \\ H_1(f)H_0^*(f) + H_1\left(f+\frac{1}{2}\right)H_0^*\left(f+\frac{1}{2}\right) = 0 \end{cases} \tag{4.37}$$

where the first term is the **orthonormal relation** of wavelet functions given in Fourier space and the second term indicates that $\{\phi(t-n)\}_{n\in\mathbb{Z}}$ and $\{\psi(t-n)\}_{n\in\mathbb{Z}}$ are orthogonal, i.e. it is the FT of the relation $\langle \psi(t), \phi(t-n)\rangle = 0, \forall n \in \mathbb{Z}$.

Finally, combining relations (4.34) and (4.37) we obtain:

$$H_1(f) = H_0^*\left(f+\frac{1}{2}\right)e^{-j2\pi f} \tag{4.38}$$

which gives, in time domain:

$$h_1[n] = (-1)^{1-n}h_0[1-n] \tag{4.39}$$

4.2.1 Piecewise Constant Bases Computation

Let us see the piecewise constant bases as example. Given the function $\vartheta(t) = 1_{[0\ 1]}$, its translations $\{\vartheta(t-n)\}_{n\in\mathbb{Z}}$ is surely a Riesz basis of the space V_0 of the functions piecewise constant in the interval $t \in [n, n+1]$, i.e. the generic family $\{2^{j/2}\vartheta(2^{-j}t-n)\}_{n\in\mathbb{Z}}$ provides a basis for the generic space V_j of the functions piecewise constant in the interval $t \in [n2^j, (n+1)2^j]$. Moreover, $\{\vartheta(t-n)\}_{n\in\mathbb{Z}}$ are also an **orthonormal** family so that $\vartheta(t)$represents the scaling function $\phi(t)$ of the space V_0 ($\phi(t) = \vartheta(t)$). Therefore, we can compute $h_0[n]$ exploiting its definition:

exploiting its definition:

$$h_0[n] = \langle \frac{1}{\sqrt{2}}\phi\left(\frac{t}{2}\right), \phi(t-n)\rangle = \begin{cases} \frac{1}{\sqrt{2}} & ,n = 0,1 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.40}$$

then, by knowing $h_0[n]$ we can compute $h_1[n]$ from Eq. (4.39):

$$h_1[n] = (-1)^{1-n}h_0[1-n] = \begin{cases} -\frac{1}{\sqrt{2}} & ,n = 0 \\ \frac{1}{\sqrt{2}} & ,n = 1 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.41}$$

and $\psi(t)$can be computed from Eq. (4.31):

$$\psi\left(\frac{t}{2}\right) = \sqrt{2}\sum_{n=-\infty}^{\infty} h_1[n]\phi(t-n) = -\phi(t) + \phi(t-1) = \begin{cases} -1 & ,0 \le t < 1 \\ 1 & ,1 \le t < 2 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.42}$$

that gives:

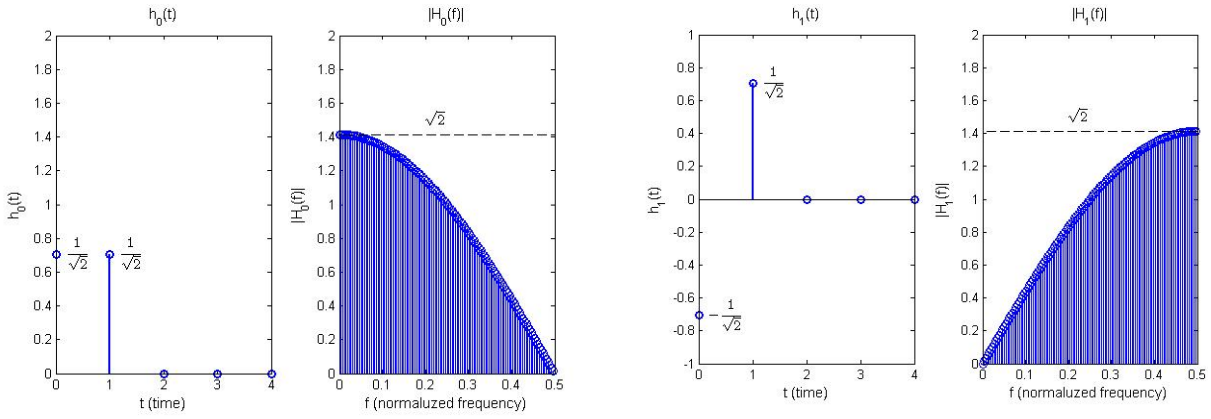
$$\psi(t) = \begin{cases} -1 & ,0 \le t < \frac{1}{2} \\ 1 & ,\frac{1}{2} \le t < 1 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.43}$$

Hereafter we report the previous **piecewise constant basis** (also called **Haar basis**) written in a compact way:

$$\phi(t) = 1_{[0\ 1]} \qquad h_0[n] = \begin{cases} \frac{1}{\sqrt{2}} & ,n = 0,1 \\ 0 & ,\text{otherwise} \end{cases}$$

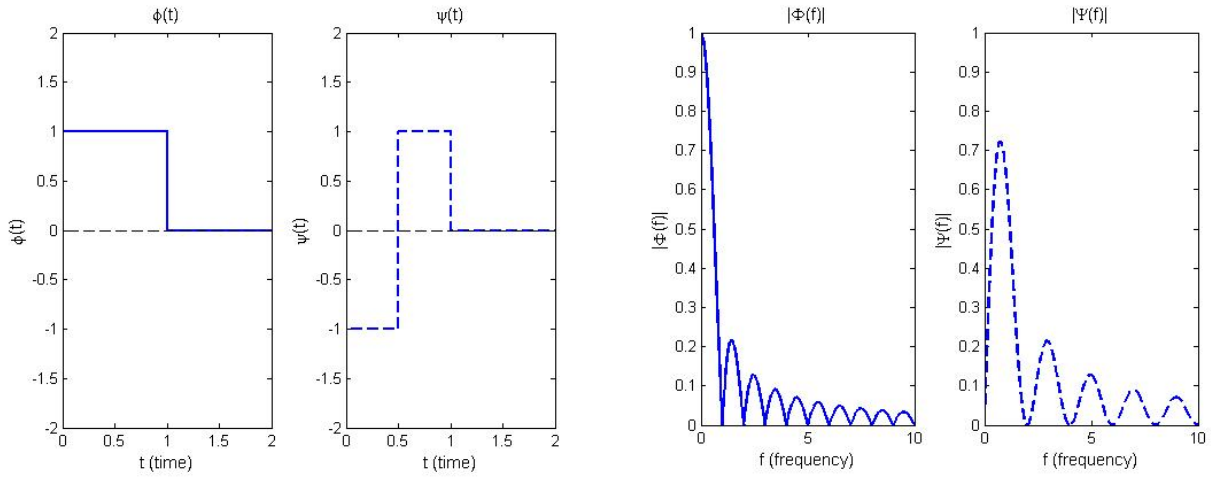
$$\psi(t) = \begin{cases} -1 & ,0 \le t < \frac{1}{2} \\ 1 & ,\frac{1}{2} \le t < 1 \\ 0 & ,\text{otherwise} \end{cases} \qquad h_1[n] = \begin{cases} -\frac{1}{\sqrt{2}} & ,n = 0 \\ \frac{1}{\sqrt{2}} & ,n = 1 \\ 0 & ,\text{otherwise} \end{cases} \tag{4.44}$$

In Fig. 4.9 we can see the responses, both time and normalized frequency, of the filters $h_0[n]$ and $h_1[n]$, whereas the time-frequency responses of the functions $\phi(t)$ and $\psi(t)$are depicted in Fig. 4.10.



(a) (b)

Fig. 4.9 - (a) $h_0[n]$ responses (time and normalized frequency). (b) $h_1[n]$ responses (time and normalized frequency)



(a) (b)

Fig. 4.10 - (a) $\phi(t)$ and $\psi(t)$time responses. (b) $\Phi(f)$and $\Psi(f)$ frequency responses

4.3 Edge Detection Filters

As seen in Section 4.1, any function expressed as a derivative of another function can be used as edge detector. In [25], Canny found a function that was the result of an optimization, i.e. an optimum function under some selected constraints. Nevertheless, following the Canny optimization method, which is only given for continuous signals, a lot of evolutions and improving have been presented in literature [33][34][35].

Briefly, let $x(t)$ be a 1-D noisy step-edge:

$$x(t) = Au(t) + n(t) \tag{4.45}$$

with A constant, $u(t)$ the unitary step function, and $n(t)$ a zero-mean white noise independent from the signal $u(t)$. Edge detection is performed applying a filter $h(t)$ to the signal $x(t)$ and looking for the maximum point of the output $y(t)$:

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \tag{4.46}$$

Clearly, for the presence of noise, at true edge point $t = 0$ the output $y(t)$ may not have any maximum or a lot of spurious maxima may exist near true edge location.

4.3.1 Canny Filter

In [25], Canny defined the following three quality criteria to be maximized by the filter $h(t)$.

1. *Insensibility to noise* (classical SNR):

$$\Sigma = \int_{-\infty}^{0} h(t)dt \Big/ \sqrt{\int_{-\infty}^{\infty} h^2(t)dt} \tag{4.47}$$

2. *Good edge localization* (inverse of the standard deviation between true edge point and maximum point of $y(t)$):

$$\Lambda = |h'(0)| \Big/ \sqrt{\int_{-\infty}^{\infty} h'^2(t)dt} \tag{4.48}$$

3. *Unique response to one edge* (it can be retrieved from the probability to have spurious maxima presupposing a Gaussian noise):

$$\mathrm{k} = |h'(0)| \Big/ \Sigma \sqrt{\int_{-\infty}^{\infty} h''^2(t)dt} \tag{4.49}$$

Imposing $h(t)$ with a finite support $[-w, w]$, Canny computed the function $h(t)$ that maximizes $\Sigma\Lambda$ under the constraint of having spurious maxima far than possible from the true edge point. This was done computing the standard deviation σ_d of the distance between maximum points of $y(t)$, which can be easily calculated presupposing Gaussian noise:

$$\sigma_d = 2\pi \left[\int_{-\infty}^{\infty} h'^2(t)dt \Big/ \int_{-\infty}^{\infty} h''^2(t)dt \right]^{\frac{1}{2}} \tag{4.50}$$

Then, he imposed the constraint $\sigma_d = |kw|$ in optimization method. It should be noticed that k should be bigger than possible and the maximum value of k is $\mathrm{k} = 2$, i.e. only one maximum, in mean, is inside the $y(t)$ support (which is $[-2w, 2w]$). Hence, performing the constrained optimization the following solution arises:

$$h(t) = a_1 e^{\alpha t} sin(\omega t) + a_2 e^{\alpha t} \cos(\omega t) + a_3 e^{-\alpha t} sin(\omega t) + a_4 e^{-\alpha t} \cos(\omega t) \tag{4.51}$$

Then, applying the boundary conditions[4]:

$$h(0) = 0;\ h(w) = 0;\ h'(0) = s;\ h'(w) = 0; \tag{4.52}$$

parameters $\{a_i\}_{i=1}^{4}$ are retrieved as function of α, $\omega$and s.

Anyway, imposing both $\Sigma\Lambda$ maximum and k bigger than possible gives rise to a too complex analytical optimization and numerical methods should be applied to provide the final solution. Applying this last methods, Canny showed that the maximum value of k ($\mathrm{k} = 0.58$) is reached for a filter that gives $\Sigma\Lambda = 1.12$ and whose shape resembles the first order derivatives of a Gaussian:

$$h(t) = -\frac{t}{\sigma^2} e^{-\frac{t^2}{2\sigma^2}} \tag{4.53}$$

which gives:

$$\Sigma = 1.03\sqrt{\sigma};\ \ \Lambda = \frac{0.89}{\sqrt{\sigma}};\ \ \Sigma\Lambda = 0.92;\ \ k = 0.51 \tag{4.54}$$

4.3.2 Deriche Filter

However, as Deriche pointed out in [35], with the approximation in Eq. (4.53) the filter performances ($\Sigma\Lambda = 0.92$) are worse by about 20% respect to the optimal operator ($\Sigma\Lambda = 1.12$). Anyway, using the criteria proposed by Canny, he performed the same passages but imposing an infinite support of $h(t)$ (i.e. $\to \infty$). In fact, it is well known that an infinite support enables to better remove the noise and avoid Gibbs effect of the filter FT $H(f)$ (the counterpart is a greater sensibility to near edges). Furthermore, many simplifications in the optimization method arise imposing $w \to \infty$ and the Eq. (4.51) in $[0, \infty]$ simply becomes:

$$h(t) = -ce^{-\alpha t} sin(\omega_0 t) \tag{4.55}$$

with α, c, and ω positive reals. Given the antisymmetry of $h(t)$, in $(-\infty, \infty)$ gives:

$$h(t) = -ce^{-\alpha|t|} sin(\omega_0 t) \tag{4.56}$$

and the performance indexes are:

[4] Being $h(t)$ antisymmetric ($h(-t) = h(t)$) the optimization is performed in $[0, w]$ and extended successively in $[-w, w]$, i.e. boundary conditions are in $[0, w]$. Moreover, parameter *s* sets the slope of $h(t)$ in $t = 0$.

$$\Sigma = \left[\frac{2\alpha}{\alpha^2 + \omega_0^2}\right]^{\frac{1}{2}};\ \Lambda = \sqrt{2\alpha};\ \Sigma\Lambda = \frac{2\alpha}{\sqrt{\alpha^2 + \omega_0^2}};\ k = \left[\frac{\alpha^2 + \omega_0^2}{5\alpha^2 + \omega_0^2}\right]^{\frac{1}{2}} \tag{4.57}$$

Setting $m = \alpha/\omega_0$, three cases can be deduced from these last formulas:

- $m \gg 1$:

$$\Sigma = \left[\frac{2}{\alpha}\right]^{\frac{1}{2}};\ \Lambda = \sqrt{2\alpha};\ \Sigma\Lambda \approx 2;\ k \approx 0.44 \tag{4.58}$$

- $m = 1$:

$$\Sigma = \left[\frac{1}{\alpha}\right]^{\frac{1}{2}};\ \Lambda = \sqrt{2\alpha};\ \Sigma\Lambda \approx \sqrt{2};\ k \approx 0.58 \tag{4.59}$$

- $m \ll 1$:

$$\Sigma \approx m\left[\frac{2}{\alpha}\right]^{\frac{1}{2}};\ \Lambda = \sqrt{2\alpha};\ \Sigma\Lambda \approx 2m;\ k \approx 1 \tag{4.60}$$

The case $m = 1$ shows that the final form used by Canny for his optimal operator ($\Sigma\Lambda = 1.12$ and $k = 0.58$) is worse than Deriche operator by more than 25%. Anyway, the best trade-off among performance indexes is reached for the first case, since the product $\Sigma\Lambda$ is maximum (more than the double of the Canny Gaussian derivative) without lowering too much the spurious maxima index *k*. The filter time response is shown in Fig. 4.11 for the three different cases previously discussed.

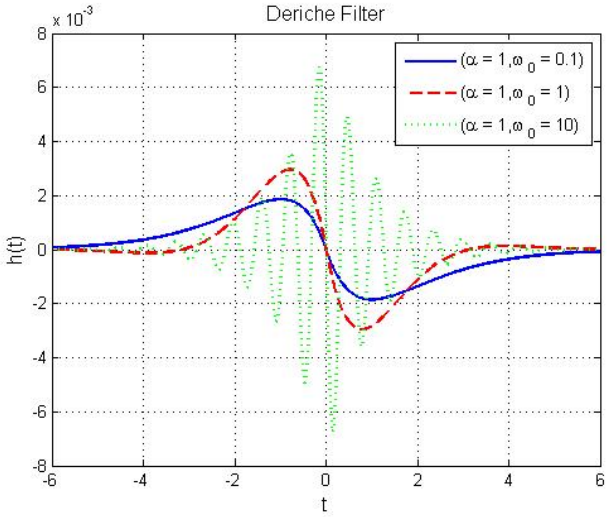


Fig. 4.11 - Deriche filter for three different parameter values ($m \gg 1$, $m = 1$, $m \ll 1$)

4.3.3 Paillou Filter

A dual form of Deriche operator was proposed by Paillou in [36]. It simply substituted the sine function in Eq. (4.56) with its hyperbolic version, i.e.:

$$h(t) = -ce^{-\alpha|t|}sin\mathrm{h}(\omega_0 t) \tag{4.61}$$

which respects boundary conditions if $\omega_0 \leq \alpha$, i.e. $m \geq 1$. In this way performance indexes become:

$$\Sigma = \left[\frac{2\alpha}{\alpha^2 - \omega_0^2}\right]^{\frac{1}{2}};\ \ \Lambda = \sqrt{2\alpha};\ \ \Sigma\Lambda = \frac{2\alpha}{\sqrt{\alpha^2 - \omega_0^2}};\ \ k = \left[\frac{\alpha^2 - \omega_0^2}{5\alpha^2 - \omega_0^2}\right]^{\frac{1}{2}} \tag{4.62}$$

consequently, $\Sigma\Lambda$ can go to infinite decreasing k to zero. Clearly, even in this last case a trade-off have to be chosen. Paillou proposed to set $k = 1/3$ (i.e. $m = 10/7$) in order to have:

$$\Sigma = 1.98;\ \ \Lambda = \sqrt{2};\ \ \Sigma\Lambda = 2.9;\ \ k = 1/3 \tag{4.63}$$

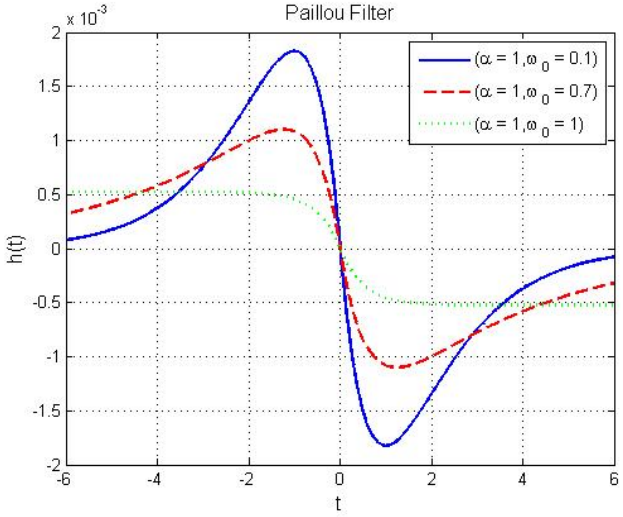


Fig. 4.12 - Paillou filter for three different parameter values ($m \gg 1$, $m \approx \sqrt{2}$, $m \ll 1$)

As can be seen from Fig. 4.12, when $m = 1$ the Paillou filter is a continuous version of the Box filter and in this last case $\Sigma\Lambda \to \infty$. Cleraly, it was already well-known that to maximize the SNR at the output of a step edge, the Box filter (i.e. the matched filter to the step edge) had to be used.

4.3.4 Shen-Castan Filter

Nevertheless, all the previous filter suffer from the trade-off problem between noise removal (high Σ) and edge localization (high Λ). An intuitive way to understand this fact can be easily described. The previous filters find an edge at the location where the signal first derivative is maximum. i.e. where the second order derivative crosses zero. In Canny notation, given the Gaussian function $G(t)$, the Canny filter is its derivative $h(t) = G'(t)$. This means that filtering a function $x(t)$ is equivalent to smoothing it with a Gaussian kernel and then applying an ideal derivative operator:

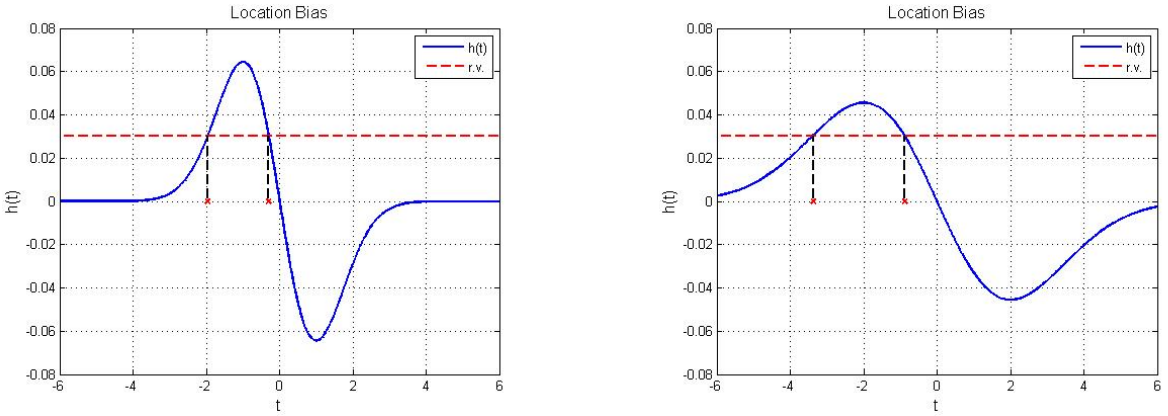
$$x * h = x * \frac{d}{dt}G = \frac{d}{dt}[x * G] \tag{4.64}$$

In order to see where the second order derivative crosses zero, we have to derive the function twice. Therefore, coming back to the original problem of finding the edge taken a nosy step $x(t) = Au(t) + n(t)$ as input:

$$\frac{d^2}{dt^2}[x * G] = \frac{d^2}{dt^2}[u * G] + \frac{d^2}{dt^2}[n * G] = u' * h(t) + n * h'(t) = h(t) + n * h'(t) \tag{4.65}$$

where the equivalence $u'(t) = \delta(t)$ has been exploited (in addition to the well-known property of the convolution, i.e. $\frac{d}{dt}[x * G] = [x' * G] = [x * G']$). This means that an edge is located in a point *t* where Eq. (4.65) is equal to zero, i.e. $h(t) = -n * h'(t)$. Clearly, the term $n * h'(t)$ is a random variable and, if its value in $t = 0$ (true edge location) is different from zero, a dislocation appears as illustrated in Fig. 4.13. In this figure can be clearly seen that the dislocation raises, in mean, increasing the Gaussian standard deviation σ.

Naturally, when σ raises, $h(t)$ has a flatter time-response, i.e. better noise removal. Moreover, always in Fig. 4.13 can be seen how the spurious maximum is kept further away from the detected edge when σ is greater.



(a) (b)

Fig. 4.13 - Location bias of Canny filter. Red dashed line indicates the mean of the r.v. $n * h'(t)$ (a) Gaussian standard deviation $\sigma = 1$. (b) Gaussian standard deviation $\sigma = 1$.

The intuition of Shen and Castan was to solve this problem using discontinuous functions. In particular, they used the derivative of a negative exponential, imposing the function zero at time $t = 0$, see Fig. 4.14(a)-(b):

$$\begin{cases} h(t) = \dfrac{d}{dt} e^{-\alpha|t|} \\ h(0) = 0 \end{cases} \tag{4.66}$$

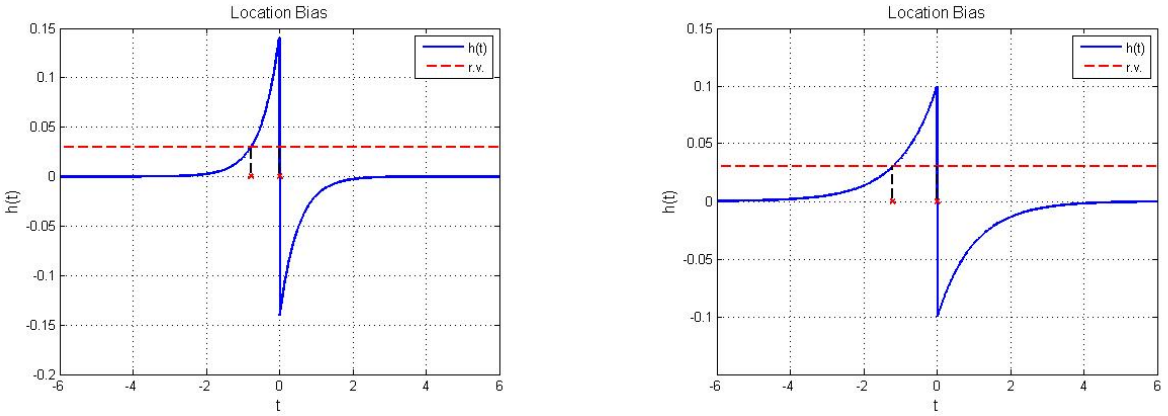
This last figure makes clear how, for Shen-Castan filter, the bias is completely avoided, so that only spurious maxima have to be solved. They derived this filter looking for a function with infinite support that optimizes the following functional:

$$\frac{(E_{x'}|_{t=0})^2}{E_n E_{n'}} \tag{4.67}$$

where $E_{x'}$ is the energy of the input signal after filtering, whereas E_n, $E_{n'}$ are the energies of the noise, respectively before and after ideal derivation, as illustrated in Fig. 4.15 In particular, for the negative exponential, maximizing the functional in Eq. (4.67) is equal to maximize the following functional:

$$\frac{E_x|_{t=0}}{E_n} \frac{E_{x'}|_{t=0}}{E_{n'}} \tag{4.68}$$

i.e. the product of the squared SNR at the exit of each system block.



(a) (b)

Fig. 4.14 - Location bias of Shen-Castan filter. Red dashed line indicates the mean of the r.v. $n * h'(t)$ (a) Negative exponential with $\alpha = 2$. (b) Negative exponential with $\alpha = 1$.

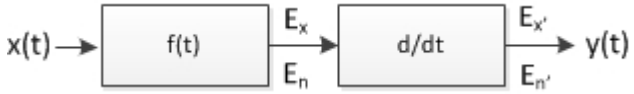


Fig. 4.15 - System outlook for Shen-Castan functional

Moreover, in [18] Shen and Castan proposed a very smart way to delete the most spurious maxima. In fact, for an edge whose first derivative is positive at edge point, the second order derivative should change sign from positive to negative (+/-). Instead, reversely, for an edge whose first derivative is negative at edge point, the second order derivative should change sign from negative to positive (-/+). Hence, only checking this simple behaviour, many false spurious edges can be easily deleted.

4.3.5 Theoretical Performance Summary

In Tab. 4.1 the theoretical performance of the previous filters are schematically reported.

		Quality Indexes (noise AWGN)				
Filter	**Impulse Response**	**Case study**	Σ	Λ	$\Sigma\Lambda$	k
Canny	$h(t) = -\frac{t}{\sigma^2} e^{-\frac{t^2}{2\sigma^2}}$	General	$1.03\sqrt{\sigma}$	$\frac{0.89}{\sqrt{\sigma}}$	0.92	0.51
Deriche	$h(t) = -ce^{-\alpha\|t\|} sin(\omega_0 t)$ $(c > 0, \alpha > 0, \omega_0 > 0)$	General	$\left[\frac{2\alpha}{\alpha^2 + \omega_0^2}\right]^{\frac{1}{2}}$	$\sqrt{2\alpha}$	$\frac{2\alpha}{\sqrt{\alpha^2 + \omega_0^2}}$	$\left[\frac{\alpha^2 + \omega_0^2}{5\alpha^2 + \omega_0^2}\right]^{\frac{1}{2}}$
		$\alpha/\omega_0 \gg 1$	$\left[\frac{2}{\alpha}\right]^{\frac{1}{2}}$	$\sqrt{2\alpha}$	≈ 2	≈ 0.44
Paillou	$h(t) = -ce^{-\alpha\|t\|} sin\mathrm{h}(\omega_0 t)$ $(c > 0, \alpha > 0, \omega_0 > 0)$	General	$\left[\frac{2\alpha}{\alpha^2 - \omega_0^2}\right]^{\frac{1}{2}}$	$\sqrt{2\alpha}$	$\frac{2\alpha}{\sqrt{\alpha^2 - \omega_0^2}}$	$\left[\frac{\alpha^2 - \omega_0^2}{5\alpha^2 - \omega_0^2}\right]^{\frac{1}{2}}$
		$\alpha/\omega_0 = 10/7$	1.98	$\sqrt{2}$	2.9	1/3
Shen-Castan	$h(t) = \begin{cases} -ce^{-\alpha\|t\|} & t \neq 0 \\ 0 & t = 0 \end{cases}$ $(c > 0, \alpha > 0,)$	General	$\left[\frac{1}{\alpha}\right]^{\frac{1}{2}}$	∞	-	1

Tab. 4.1 - Edge detection filter summary.

4.3.6 Filter Implementation

In this Section, the problem concerning the discrete implementation of continuous filter function will be discussed. Technical detail about filter functions and their Z-transformed version (when exists) for discrete implementation purpose is reported in Appendix I, even for the two dimensional case. Moreover, always in Appendix I, the implementation of wavelet transform, by both **Fast Fourier Transform** (FFT) and **recursive filtering method** is discussed**.**

From the Section 4.1it should be known that, given a smoothing continuous (and real) function $\theta(t)$, a wavelet function can be computed as $\psi(t) = -\theta'(t)$. Then, let's consider a dyadic multiscale approximation where translation happens with time shift $\tau \in \mathbb{R}$, and a change of scale happens through the usual formula $\psi_{\tau,2^j} = 2^{-j/2}\psi\left(2^{-j}t - \tau\right)$. In this case the wavelet transform of $f(t)$ is given by:

$$Wf\left(\tau, 2^j\right) = \langle f(t), \psi_{\tau,2^j}(t)\rangle = \int_{-\infty}^{\infty} f(t) 2^{-j/2} \psi\left(2^{-j}t - \tau\right) dt = f * \bar{\psi}_{\tau,2^j} \tag{4.69}$$

where $\bar{\psi}(t) = \psi(-t)$. Now, since:

$$\psi_{\tau,2^j}(t) = -2^j \frac{d\theta_{\tau,2^j}(t)}{dt} \tag{4.70}$$

Eq. (4.69) can be written as:

$$Wf(\tau, 2^j) = f * \bar{\psi}_{\tau,2^j} = 2^j \frac{d}{dt}\left[f * \bar{\theta}_{\tau,2^j}\right] \tag{4.71}$$

that is, wavelet transform is proportional to the derivative of the original function smoothed by $\bar{\theta}(t)$, i.e. it is proportional to the output of an edge detector.

Clearly, the function $f(t)$ is sampled in most practical applications, i.e. we have to use $f[n]$, with $n \in N$. Then, on sampled grid, it is straightforward to consider multiscale approximation where the time shift τ and the time variable *t* are not real but integer. In this case Eq. (4.71) becomes:

$$Wf(p, 2^j) = \langle f[n], \psi_{\tau,2^j}[n] \rangle = \sum_{n=-\infty}^{\infty} f[n] 2^{-j/2} \psi\left[2^{-j}n - p\right] = f * \bar{\psi}_{p,2^j} \tag{4.72}$$

It should be noticed that filters given in Section 4.3 (and Appendix I) can be directly inserted in this formula. In fact, the high-pass filter $h[n]$ given in Appendix I represents $h[n] = \bar{\psi}_{0,2^0}[n]$, i.e the discrete wavelet function of the space V_0, whereas the low-pass $h[n]$ represents $h[n] = \bar{\phi}_{0,2^0}[n]$. Implementing Eq. (4.72), the fast implementation relies on the use of recursive filtering (when possible). Naturally, a fast decomposition that uses $h_0[n]$ and $h_1[n]$ can also be given computing (numerically when no closed form exists) the following formulas:

$$\begin{aligned} h_0[n] &= \langle \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right), \phi(t-n) \rangle = \frac{1}{\sqrt{2}} \phi\left(\frac{t}{2}\right) * \phi(n-t) \\ h_1[n] &= \langle \frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right), \phi(t-n) \rangle = \frac{1}{\sqrt{2}} \psi\left(\frac{t}{2}\right) * \phi(n-t) \end{aligned} \tag{4.73}$$

where $\psi(t) = h(t)$ and $\phi(t) = -\int h(t)dt$, with $h(t)$ the continuous version of the filter $h[n]$ given in Section 4.3 (and Appendix I). It is worth noticing that in both implementation versions (Eq. (4.72) and Eq. (4.73)) the number of operations depends, respectively, on the length of the filter $h[n] = \bar{\psi}_{0,2^0}[n]$ ($h[n] = \bar{\phi}_{0,2^0}[n]$) and $h_1[n]$ ($h_0[n]$). Hence, theoretically, what method is the fastest relies on the edge detector filter and on the amount of discrete coefficients that are considered as meaningful (e.g. coefficients lesser than 10^{-4} can be considered zero so to shortening the filter length). Nevertheless, the results reported in this document, except when expressly mentioned, are obtained by Eq. (4.72). This choice has been done because Eq. (4.72) enables a direct implementation to be done. Nevertheless, whereas Eq. (4.73) is the theoretically sound way to implement wavelet decomposition, the version in Eq. (4.72) presupposes that, even at the first finest scale, the wavelet function can be discretized with the same sampling period of the original signal, i.e. the frequency support of the wavelet function is lesser or equal to that of the original signal. Naturally, this latter assumption is never respected, i.e. a little distortion (aliasing) arises in the computation of the wavelet coefficients at small scales. Anyway, in order to compare implementation of Eq. (4.72) with respect to Eq. (4.73), the Spline Filter has been added to the comparison. In fact, this filter is the Multiscale edge detector proposed by Mallat [28] to approximate Canny Filter and it enables implementation of Eq. (4.73) to be done.

4.3.7 Spline Filter (Mallat Edge Detector)

Quadratic Box Spline filter enables Eq. (4.73) to be implemented. In fact, such filter is devised choosing second order (quadratic) box spline $b_2(t)$ as scaling function:

$$\phi(t) = b_2(t) = rect(t) * rect(t) * rect(t) * \delta(t - 1/2) \tag{4.74}$$

whose FT is:

$$\Phi(f) = \left(\text{sinc}(f)\right)^3 e^{-j\pi f} \tag{4.75}$$

From Eq. (4.33) derives:

$$H_0(f) = \frac{\sqrt{2}\Phi(2f)}{\Phi(f)} = \sqrt{2}\left(\cos(\pi f)\right)^3 e^{-j\pi f} \tag{4.76}$$

whose IFT is:

$$h_0[n] = \{0.1768, 0.5303, 0.5303, 0.1768\}, n = -1,0,1,2. \tag{4.77}$$

Then, selecting $h_1[n] = \{-1, 1\}, n = 0,1$ (i.e. finite difference filter), the Decomposition Theorem can be applied [26]. This means that, at the output of the *j-th* scale, *approximation* (low-pass) coefficients $f_j^{(l)}[n] = \langle f, \phi_{j,n} \rangle$ and the *detail* (high-pass) sequence $f_j^{(h)}[n] = \langle f, \psi_{j,n} \rangle$ can be obtained by (see Fig. 4.16):

$$\begin{aligned} f_{j+1}^{(l)}[n] &= f_j^{(l)} * \bar{h}_0[2n] \\ f_{j+1}^{(h)}[n] &= f_j^{(l)} * \bar{h}_1[2n] \end{aligned} \tag{4.78}$$

where the original discrete signal $f[n]$ is considered equal to the approximation coefficients at level *0* (scale *1*), i.e. $f[n] = \langle f, \phi_{0,n} \rangle = f_0^{(l)}[n]$.

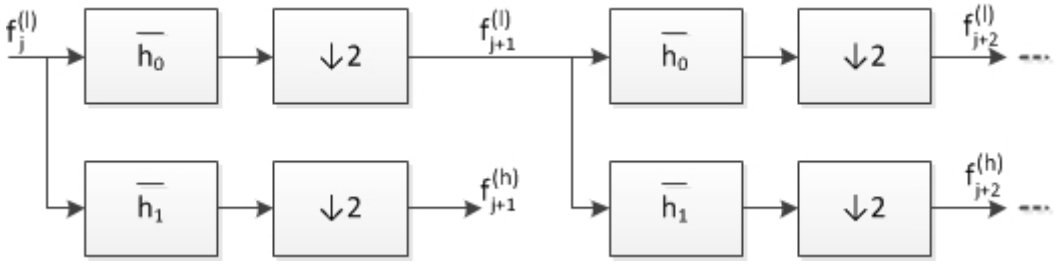


Fig. 4.16 - Decomposition Theorem Schema (two level)

Nevertheless, downsampling after filtering introduces some problems in edge detection issue since for certain scales the wavelet coefficients belonging to an edge can be dropped by downsampling operation. Hence, a *undecimated* wavelet decomposition theorem can be simply obtained by moving the downsamplers to the end of the chain and exploiting the *noble identitiy* showed in Fig. 4.17, i.e. each filter of the *j-th* level has been substituted with its upsampled version by a factor equal to 2^j (see Fig. 4.18). Spline filter results reported in this document have been obtained by implementing the undecimated wavelet decomposition previous explained.

Fig. 4.17 - Noble Identity

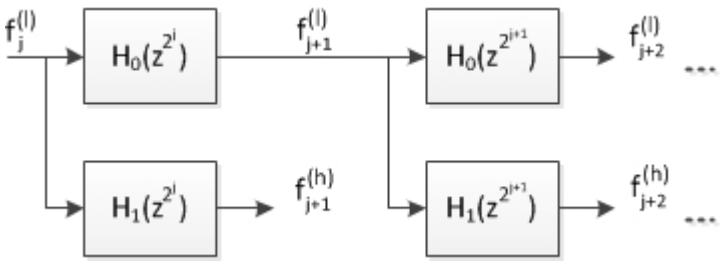


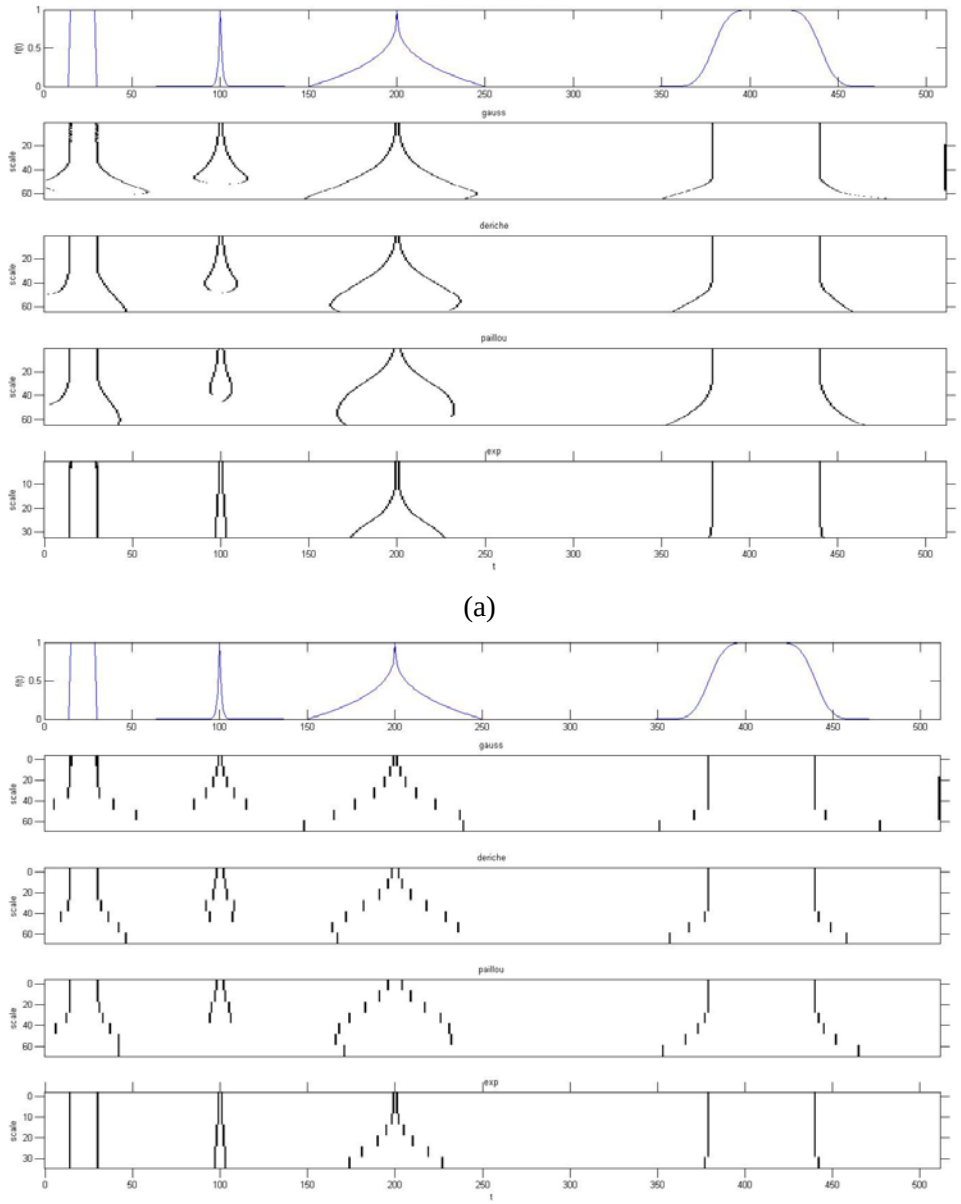
Fig. 4.18 - Undecimated Decomposition Schema (two level).

4.4 Edge Detection Filter: 1-D Results

In order to compare the filters discussed in Section 4.3, the results in applying such linear operators to a simple simulated signal with several singularities are reported in Fig. 4.19. The figure at the top show the original signal, whereas the remaining ones show, for each filter, the local maxima points at several scales (the scale is on ordinate axes). As can be seen from these figures, Deriche and Paillou obtain results very similar to Canny filter, yet a better edge localization is obtained at large scales (see results on the first step edge of the original signal). Nevertheless, as expected, only Shen-Castan filter keeps a perfect localization of all edges throughout the scales. It is worth noticing that the coarsest scale of each filter has been chosen in order to have the filter support lesser than the input signal to process. Furthermore, no threshold is applied to the modulus maxima found. Finally, dyadic decomposition of the Spline operator (see Fig. 4.20) gives results very close to the previous filters.

Filter behaviours can be appreciated also from respective frequency responses, both continuous and dyadic, reported in Fig. 4.21- Fig. 4.24. As can be seen from these figures (and previously discussed in Section 4.1), for all filters the first scale does not cover the highest frequency components of the signal. This fact is due to the design choice of starting from the scale in which $h_1[n]$filter is $h_1[n] = \{-1, 0, 1\}, n = -1,0,1$ at the place of $h_1[n] = \{-1, 1\}, n = 0,1$ that is needed to filter the highest frequencies. Clearly, some information about high frequency components is lost with our choice (e.g. a chessboard edge with a frequency of exactly 0.5 would not be revealed by filters), but in this way the initial wavelet coefficients are less influenced by noise. Anyway, filtering with the latter finest scale can be added without any effort to all filters. To see the different frequency behaviour of Spline filter at the first finest scale see Fig. 4.25, where frequency response of such filter is reported[5].

[5] For the sake of comparison among Spline filter FT and the previous frequency responses, at each scale only the frequency components $f \leq 1/2^j$ are reported for the Spline filter. In fact, because the Spline implementation relies on upsampling operations, the frequency response, which at scale 1 is in $f > 1/2$, arises in $1/2^j < f \leq 1/2$ at scales greater than 1.



(a)

(b)

Fig. 4.19 - Multiscale edge detector result when non-maxima suppression is applied at $Wf(\tau, 2^j)$ for a signal with several singularities (shown on top). The results are in the following order (from top to bottom): Canny, Deriche, Paillou and Shen-Castan. (a) Continue wavelet transform. (b) Dyadic wavelet transform

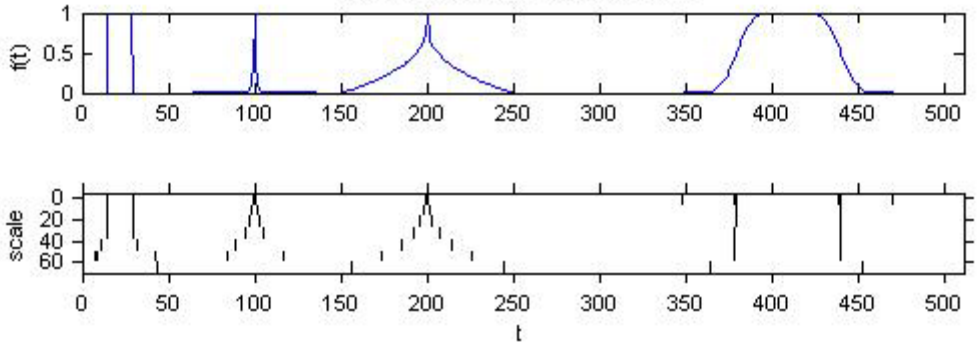
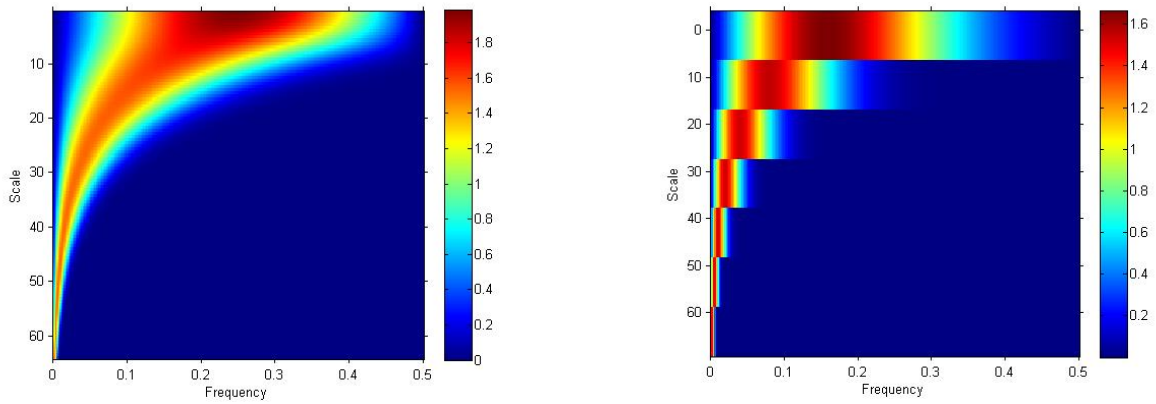
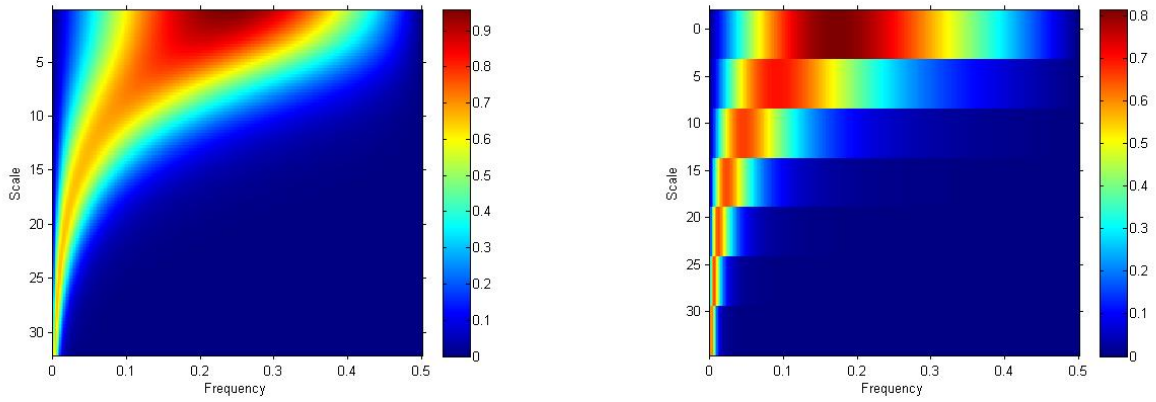


Fig. 4.20 - Spline multiscale edge detector result when non-maxima suppression (shown at the bottom) is applied at $Wf(\tau, 2^j)$ for a signal with several singularities (shown at the top)



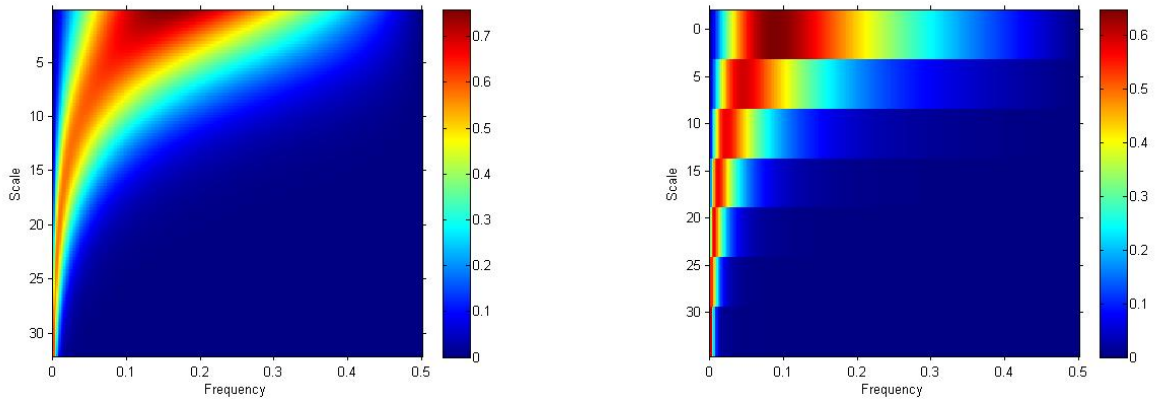
(a) (b)

Fig. 4.21 - Frequency response of the Canny wavelet function at each scale. (a) Continue scale decomposition. (b) Dyadic scale decomposition



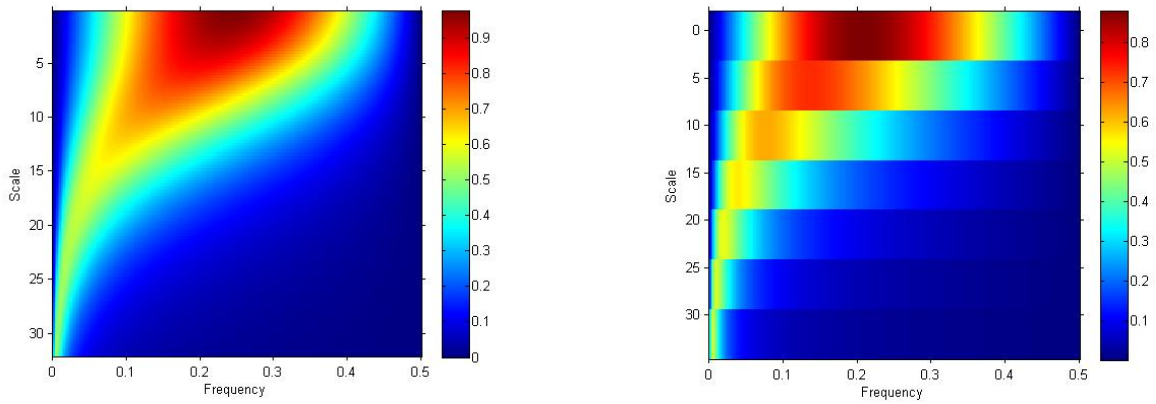
(a) (b)

Fig. 4.22 - Frequency response of the Deriche wavelet function at each scale. (a) Continue scale decomposition. (b) Dyadic scale decomposition



(a) (b)

Fig. 4.23- Frequency response of the Paillou wavelet function at each scale. (a) Continue scale decomposition. (b) Dyadic scale decomposition



(a) (b)

Fig. 4.24- Frequency response of the Shen-Castan wavelet function at each scale. (a) Continue scale decomposition. (b) Dyadic scale decomposition

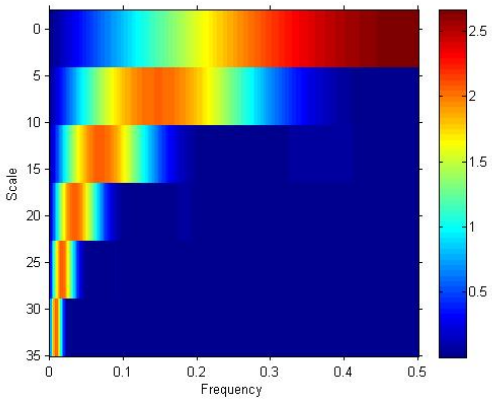


Fig. 4.25 - Frequency response of Spline filter at each dyadic scale

To see the differences among previous filters with respect to the well-known box filter[6], which compute the variation in mean between the left and right neighbourhood of a pixel, in Fig. 4.26 and Fig. 4.27 respectively the edge detection results and filter frequency responses of this latter filter are reported. As can be seen from these figures, the box filter (Haar basis) is very sensitive to neighbour singularities and, increasing the scale, a lot of spurious edge raise. Moreover, as expected, its frequency response gives rise to Gibbs effects due to the lack of smoothness in the filter shape.

[6] It is utilized in Ratio of Averages (RoA) filter paradigm described in Section 3.3. The output of the filter is the difference between the signal means on the left and on the right with respect to the reference pixel.

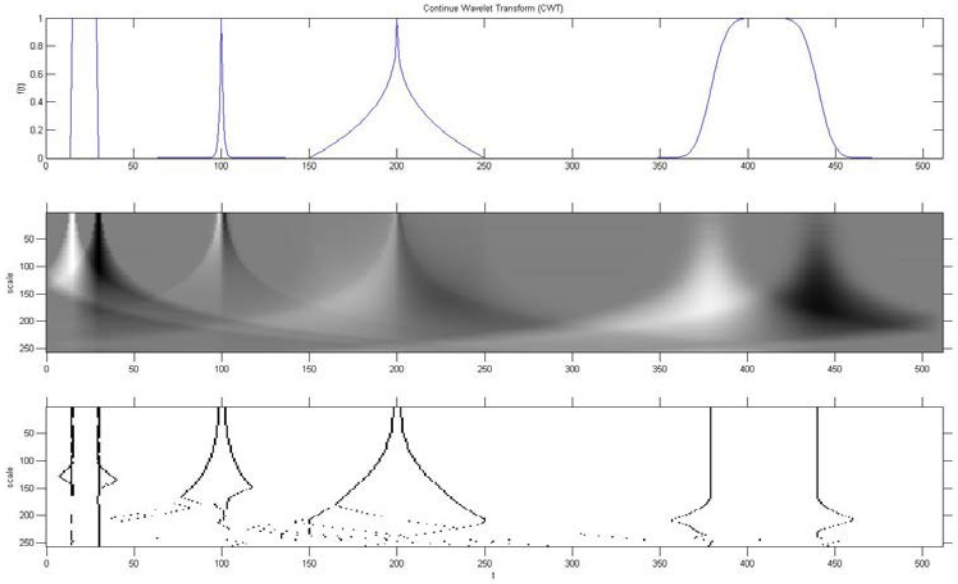


Fig. 4.26 - Box Filter (Haar) multiscale edge detector result when non-maxima suppression (shown at the bottom) is applied at $Wf(\tau, 2^j)$ (shown in the middle) for a signal with several singularities (shown at the top)

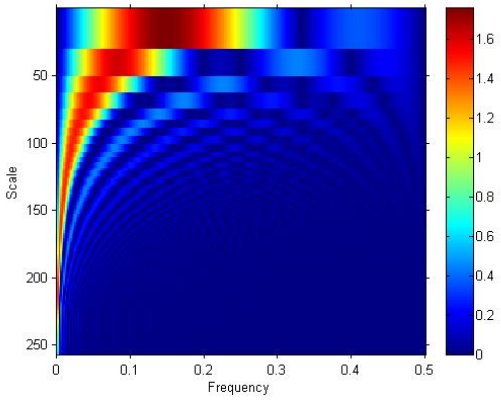
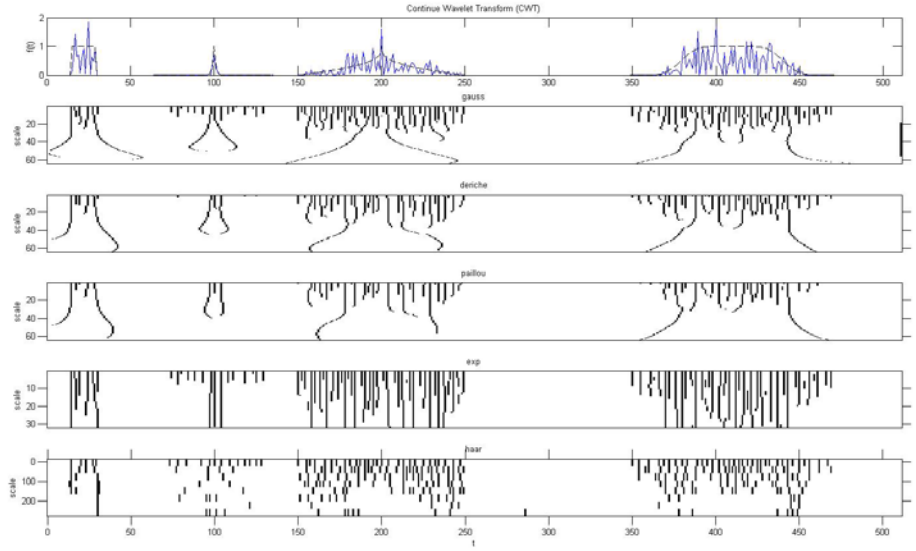
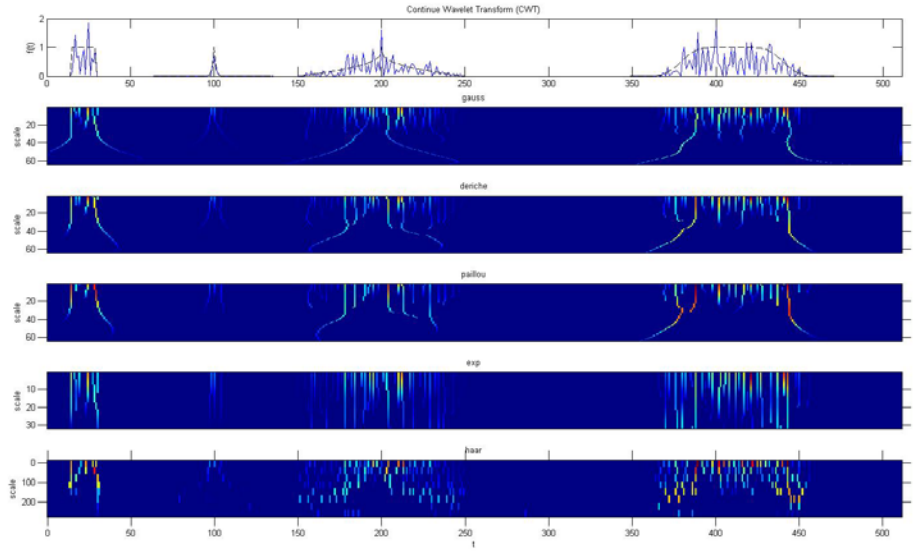


Fig. 4.27 - Frequency response of the Box filter (Haar) at each scale

Clearly, on noisy edges, results are very different and many spurious edges arise. In Fig. 4.28(a) results in applying wavelet filters on the previous signal corrupted by a **multiplicative Gamma noise with one number of looks** is reported. In order to show if a threshold is able to delete spurious edges, in Fig. 4.28 (b) at detected edge points the modulus value is depicted. As can be seen from these two images, Canny, Deriche and Paillou filter manage to remove the noise highlighting real edges (even though biased in position) raising the scale, whereas Shen-Castan and Harr are very sensitive to the noise.



(a)



(b)

Fig. 4.28 - Multiscale edge detector result when non-maxima suppression is applied at $Wf(\tau, 2^j)$ for a noisy signal (multiplicative Gamma noise with one number of look) with several singularities (shown on top). The results are in the following order (from top to bottom): Canny, Deriche, Paillou, Shen-Castan, Haar. (a) Non-maxima suppression. (b) Wavelet coefficients modulus $Mf(\tau, 2^j)$ at modulus maxima points

In order to compare previous results with those obtained by Spline filters, in Fig. 4.29 the non-maxima suppression (shown in the middle) and the value of $Mf(\tau, 2^j)$ in such points (shown at the bottom) is depicted. As can be seen, even Spline filter is very sensitive to such strong noise.

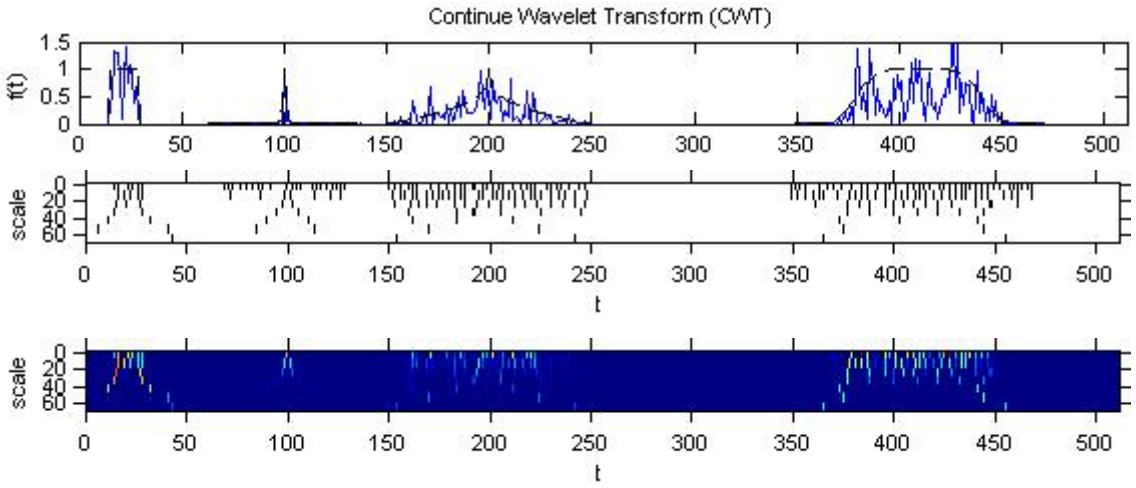


Fig. 4.29 - Spline multiscale edge detector result when non-maxima suppression (on the middle) is applied at $Mf(\tau, 2^j)$ (at the bottom only on maxima points) for a noisy signal (multiplicative Gamma noise with one number of look) with several singularities (shown on top)

4.5 Edge Detection Filter: 2-D Results

Relying on the previous results, Multiscale edge detectors seem useless in facing high variance (number of looks equal to one) speckle noise. Anyway, indicating with N_k the 1-D support (in time) of the filter $h[n]$ at scale *s*, the 2-D filter support at the same scale becomes $(N_k)^2$, so that filter sensitivity to the noise is greatly reduced.

The previous filters are firstly applied to the free-noise image in Fig. 4.30. In particular, each row is a pulse train with raising widths: 5, 15, 25, 35, 45, 55. Each rectangular pulse is expanded column-wise along 100 pixels with a free space of 25 columns from a subsequent pulse. The four pulse trains have a raising height of value: 200, 250, 300, 350, from top to bottom. As mentioned previously, those heights correspond to very low RCS ratios that can be frequently found on SAR images, e.g. between runways and nearby terrains.

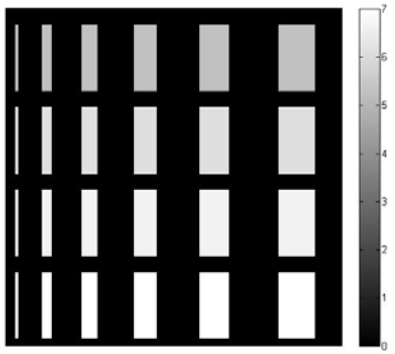


Fig. 4.30 - (Logarithm (base e) of the noise free signal (512x512 px).

In Fig. 4.31 the gradient module $Mf(\boldsymbol{\tau}, 2^j) = \sqrt{|W^1 f(\boldsymbol{\tau}, 2^j)|^2 + |W^2 f(\boldsymbol{\tau}, 2^j)|^2}$ of the noise free image at the first and fourth scale are reported beside their modulus maxima points (without thresholding). In Fig. 4.31 the frequency responses HL, LH and HH at scale 1 and 4 are depicted.

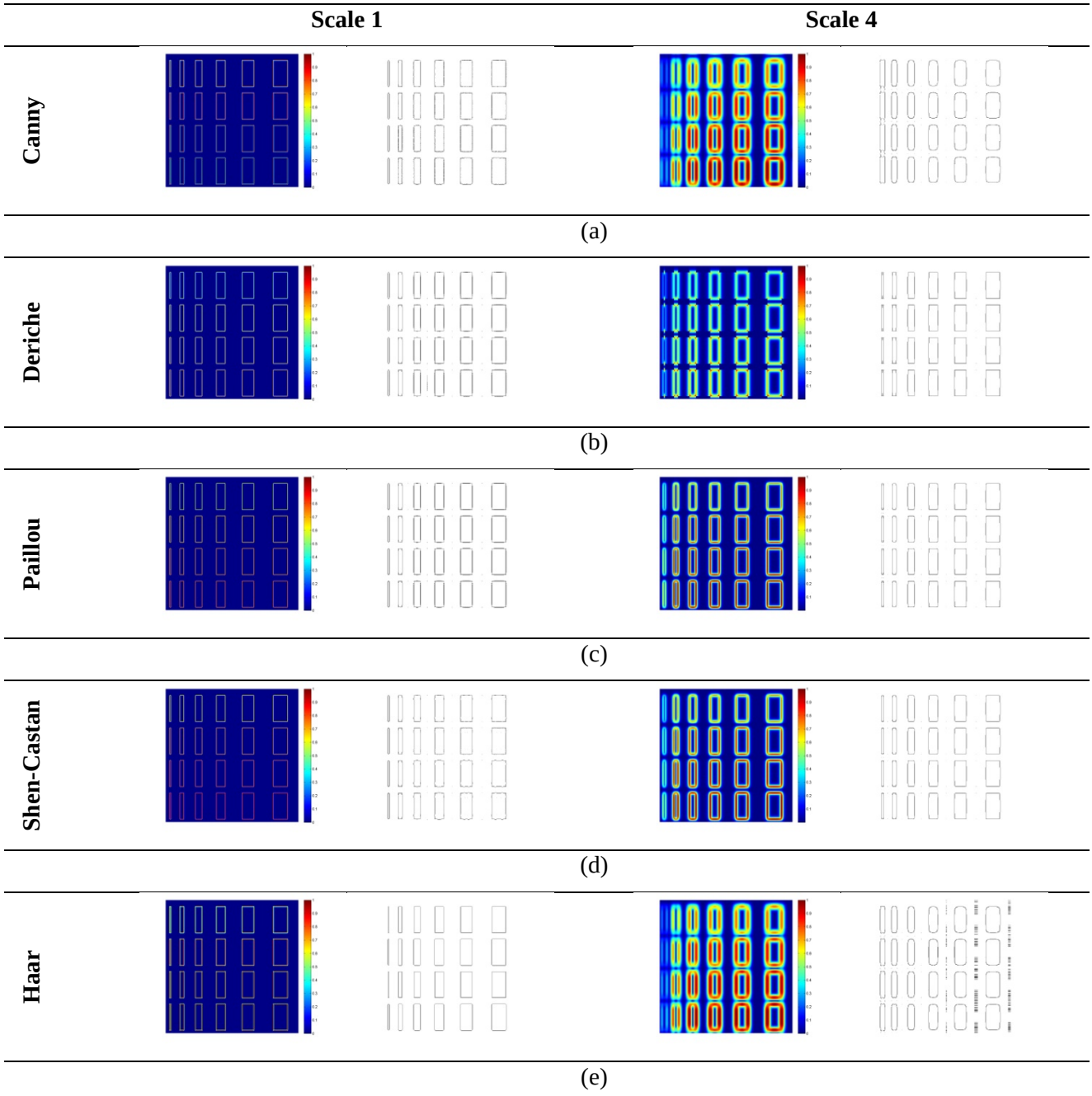


Fig. 4.31 - Normalized modulus $Mf(\tau, 2^j)$ and its local maxima values at scale 1 (left) and at scale 3 (right). (a) Canny. (b) Deriche. (c) Paillou. (d) Shen-Castan. (e) Haar

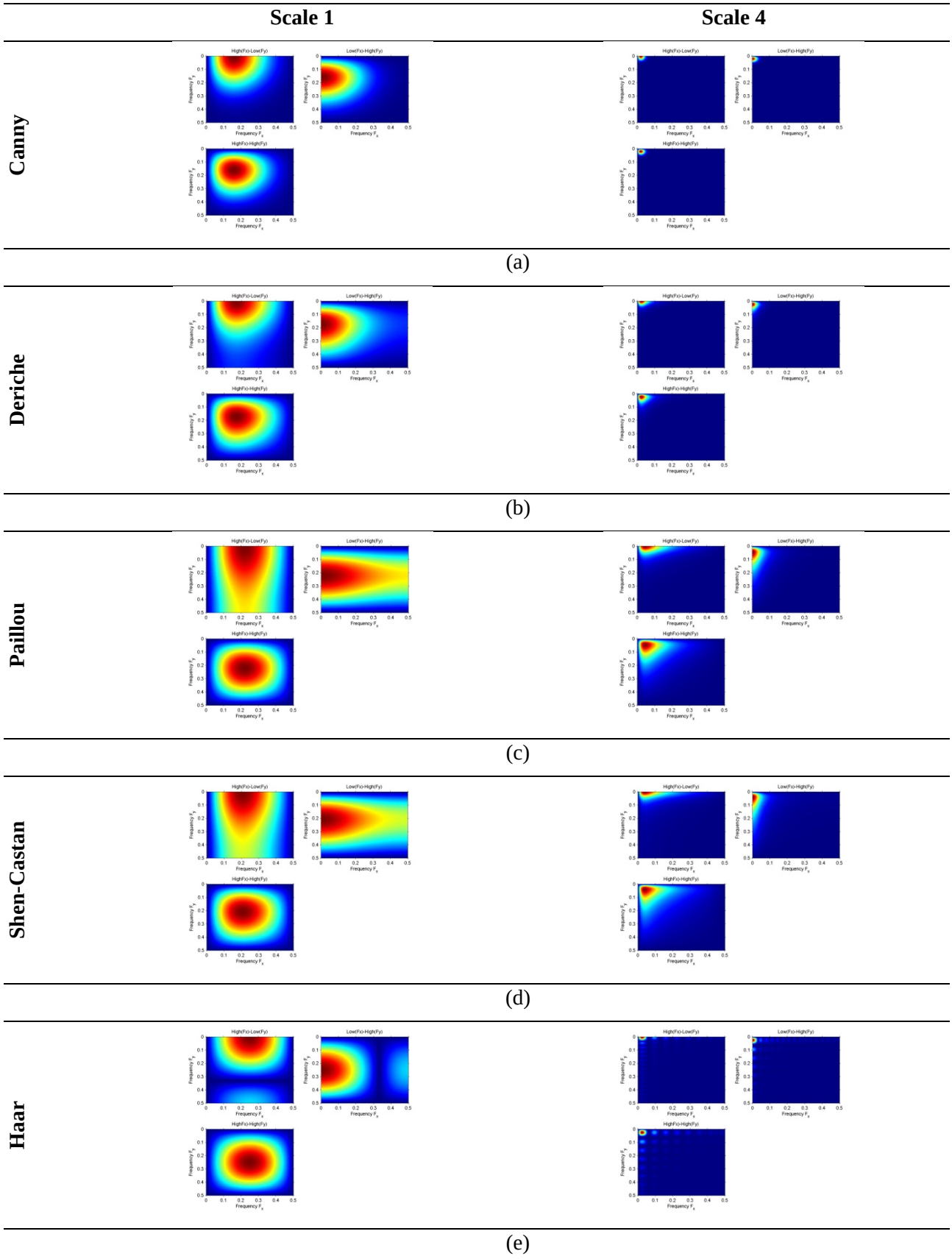


Fig. 4.32 - Frequency Responces HL, LH and HH at scale 1 (left) and at scale 4 (right). (a) Canny. (b) Deriche. (c) Paillou. (d) Shen-Castan. (e) Haar

4.5.1 Thresholding a Combination of Wavelet Coefficients

After filtering, we have measured detection rate (DR) and false alarm rate (FAR) after performing the following histogram thresholding methods:

- Otsu with two classes ("edges", "not edges") [23];
- Otsu with three classes ("strong edges", "weak edges", "not edges") [23];
- Entropy with two classes ("edges", "not edges") [24].

For each method, we compared the threshold selection both by using the wavelet coefficient modulus at each scale separately (i.e. one threshold for each scale) and by exploiting all scales jointly (i.e. the same threshold for all scales). It should be noticed that even the concise performance index "*idx*" has been measured. This index is simply computed as $idx = [DR + (1 - FAR)]/2$, and it is helpful to compare different methods together, without the necessity to see DR and FAR jointly. Furthermore, differently from the Pratt's Figure of Merit (FOM) [37], this index enable us to compute the FAR on an interval where the signal is constant, i.e. it is not biased by near edges as in FOM computation. In fact, FOM is computed as:

$$FOM = \frac{1}{\max\{N_{DE}, N_{ID}\}} \sum_{i=1}^{N_{DE}} \frac{1}{1 + \beta d_i^2} \tag{4.79}$$

with N_{DE} the number of detected points, N_{ID} the number of ideal edge points, d_i is the distance between the *i*th detected edge point and the closest true edge point, whereas β is a weight parameter. Instead, for *idx*, high values will indicate high DR and low FAR at the same time, whereas *idx* values around 0.5 indicates the worst case, i.e. low DR and low FAR contemporaneously. Analysing the obtained results (not reported for conciseness) many consideration can be done. Firstly, for all filters**, raising the scale implies to decrement DR**. Secondly, **only the initial small scales give acceptable results**. Finally, **using the same threshold for all scales gives the worst performance**.

After the previous thresholding methods, a way to combine edge maps at each scale has to be found. In order to solve this problem, a combination method of wavelet coefficients at each scale can be devised. In literature, the most promising results are the first two of the following ones:

- "scale multiplications" method [29][31];
- "scale sums" method [30];
- "scale minima" method.

In Scale Multiplication (SM) the coefficients of the initial scales are multiplied in order to enhance coefficients at edge position meanwhile neglecting noisy coefficients. Similar considerations hold for Scale Minima (SMin) method, where at each position, only the minima values throughout the considered scales are left. Instead, Scale Sums (SS) method simply computes the coefficient means of all scales for each position. In formulas, SM, SMin, and SS values can be respectively computed as:

$$\begin{aligned} SM\, f(t) &= \prod_{j=0}^{J_m - 1} Wf\left(\tau, 2^j\right) \\ SS\, f(t) &= \frac{1}{J_{ss}} \sum_{j=0}^{J_{ss} - 1} Wf\left(\tau, 2^j\right) \\ SM\text{in}\, f(t) &= \min_{j=0,\cdots,(J_{\min})} \left|Wf\left(\tau, 2^j\right)\right| \end{aligned} \tag{4.80}$$

where $J_m, J_{ss}, J_{\min}$ are the scale upper limit of each method.

4.5.1.1 Simulated Image

In this section the results in applying the previous filter to the image in Fig. 4.33 are reported. This last image is the simulated image in Fig. 4.30 onto which the **multiplicative noise of i.i.d. Gamma pixels with number of looks equal to one** has been applied.

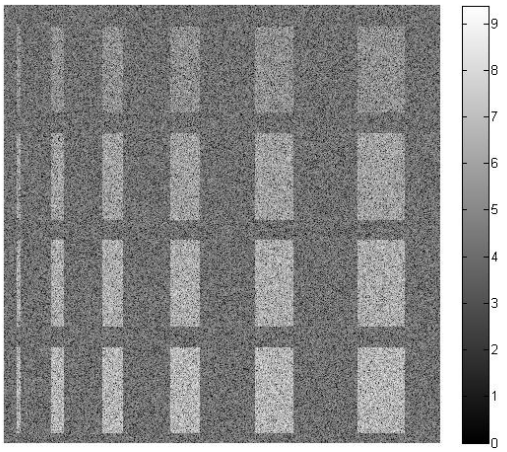


Fig. 4.33- Logarithm of the (512x512 px) signal in Fig. 4.30 with a multiplicative noise of i.i.d. Gamma pixels with number of looks equal to 1.

The performance measurements obtained after filtering the image in Fig. 4.33 and the combining wavelet coefficients is reported in Fig. 4.35 - Fig. 4.40. In these figures, a performance index value, fixed a thresholding method, is reported as a pixel with a color proportional to the value (red correspond to 1 and blue to 0), see Fig. 4.34 for the position legend. It is worth nothing that for each filter the best upper limit of the scale has been chosen among all possible values from 1 up to 9. Then, the weak threshold of the Canny-like double thresholding (also called hysteresis thresholding) method has been chosen as 40% of the strong threshold given by histogram thresholding methods [25], with exception of Otsu three classes that provides two thresholds directly. In particular, it should be noticed that in 2D case the information about gradient direction ("phase") is exploited to perform a non-maxima suppression along the gradient direction. Then, only the maxima that cross the threshold are left in the final edge map.

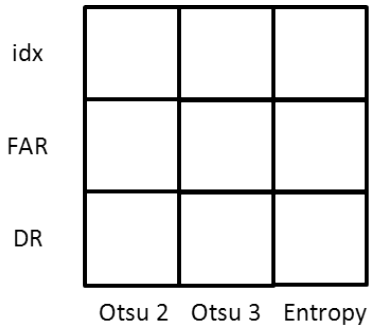
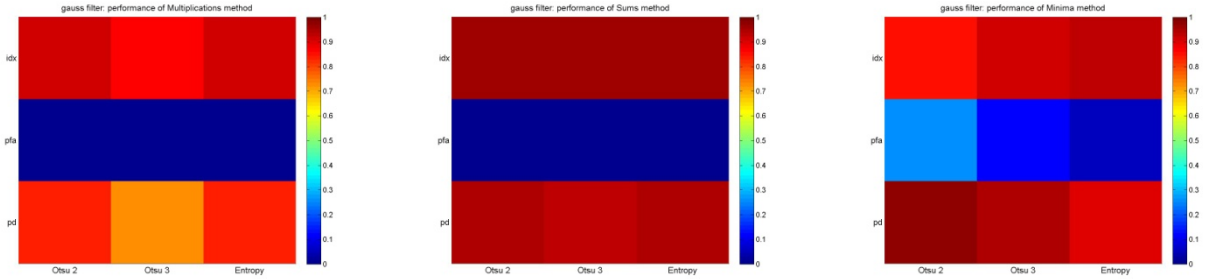


Fig. 4.34- Legend of the performance index results.

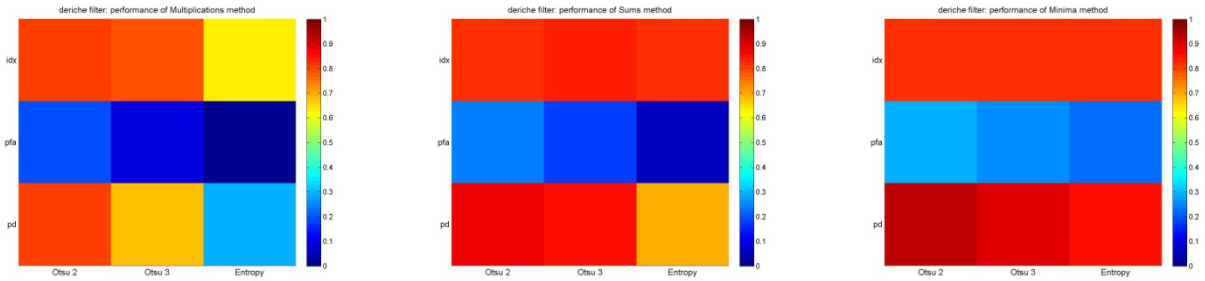
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 2$)**	**Minima ($J_{\min} = 2$)**



Canny

Fig. 4.35- Canny wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

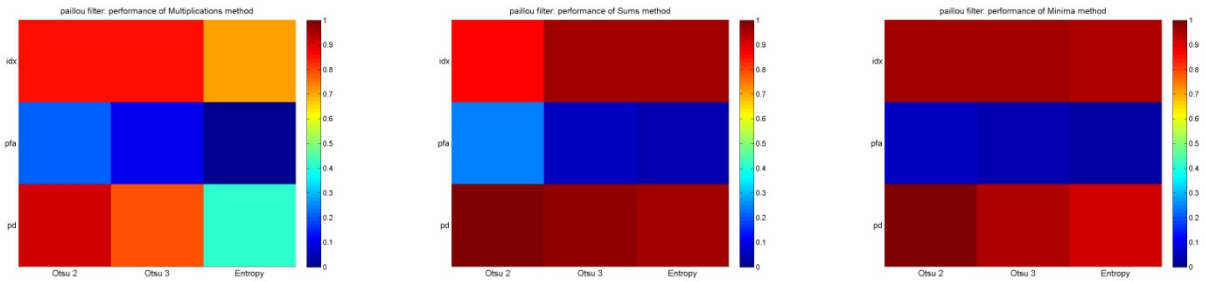
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 2$)**	**Minima ($J_{\min} = 2$)**



Deriche

Fig. 4.36 - Deriche wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

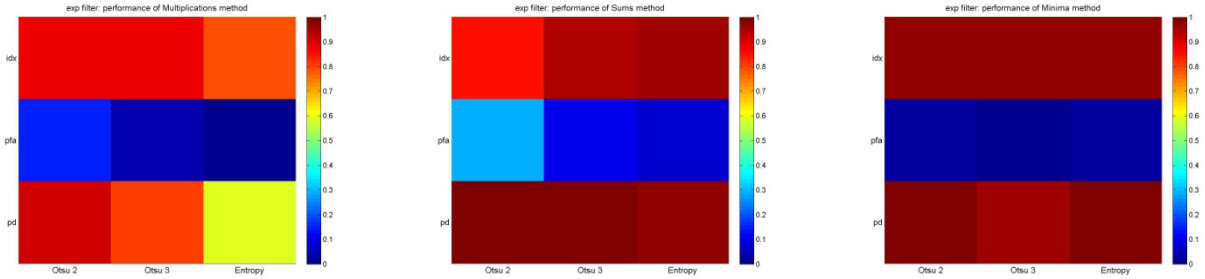
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 4$)**



Paillou

Fig. 4.37 - Paillou wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

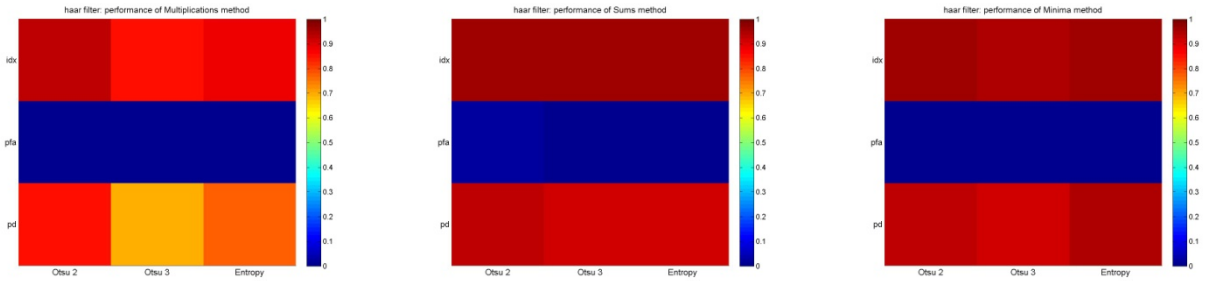
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 3$)**



Shen-Castan

Fig. 4.38 - Shen-Castan wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

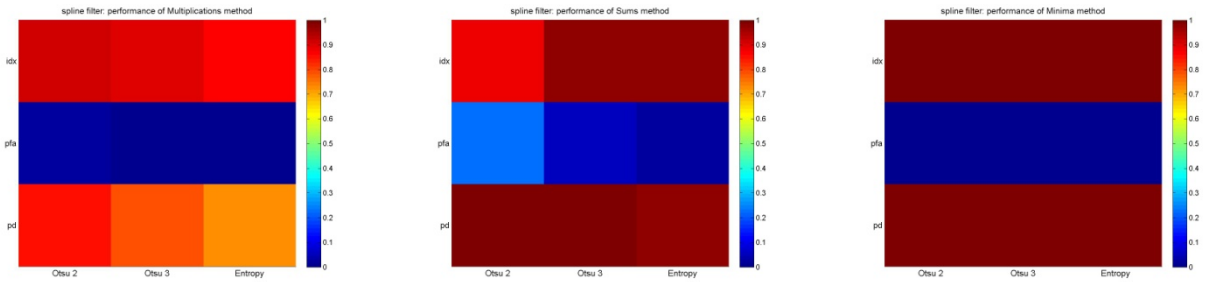
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 3$)**	**Minima ($J_{\min} = 2$)**



Haar

Fig. 4.39 - Haar wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 3$)**

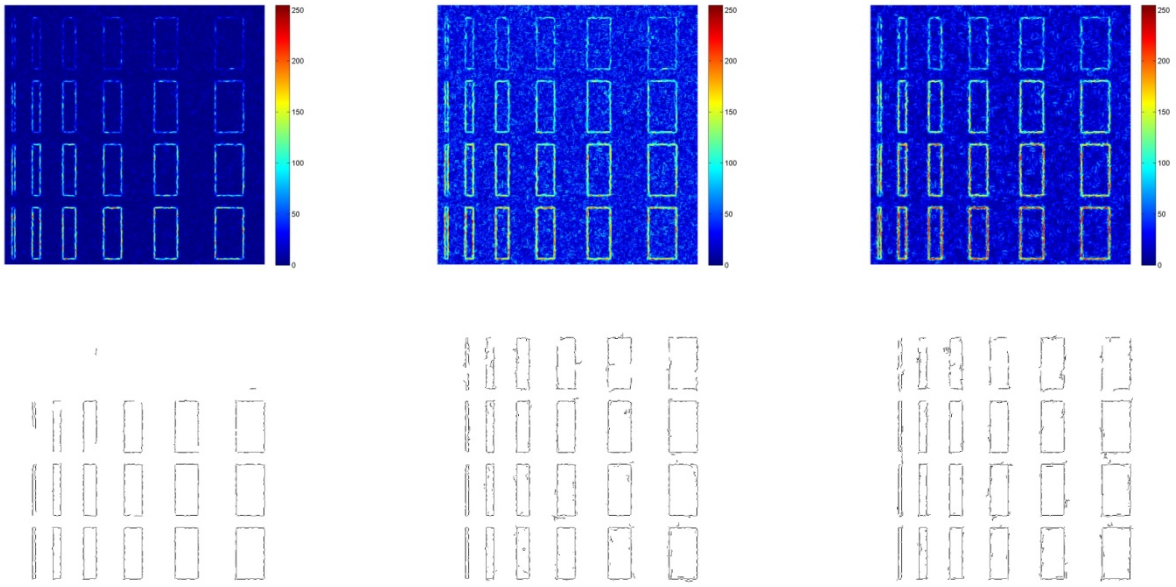


Spline

Fig. 4.40 - Spline wavelet combination, SM (on the left), SS (on the centre), SMin (on the right)

Analysing the previous results, we see that **SMin and SS** combination methods **enable the best performances to be obtained**. Then, **SM method shows a good capability to remove the noise**, even though only few scale combinations are allowed to preserve weak edges. Finally, **Haar filter yields nearly stable performance** for all combination methods, **followed in the rank by Spline, Canny, Shen-Castan, Paillou and Deriche operator.** In order visualize some results, in Fig. 4.41 and Fig. 4.42, the combined wavelet coefficients (normalized by its maxima value) and the corresponding edge map (using hysteresis thresolding with three classes Otsu method) are reported for Haar and Spline edge detectors.

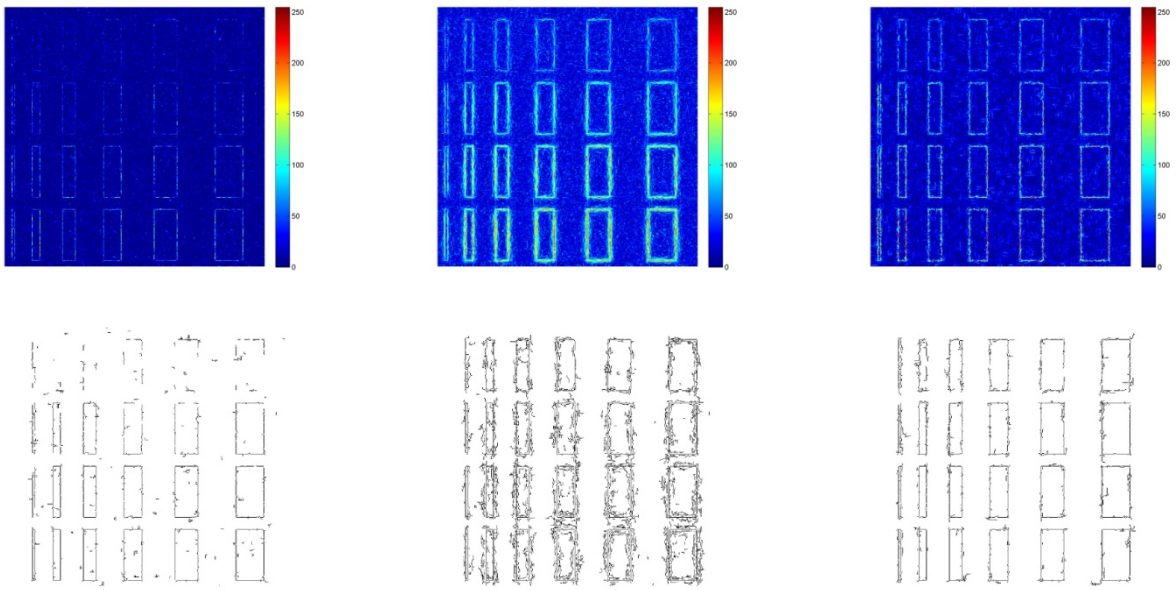
Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 3$)**	**Minima ($J_{\min} = 2$)**



Haar

Fig. 4.41 - Haar wavelet combination (top) and Otsu 3 classes hysteresis thresholding (bottom), SM (on the left), SS (on the centre), SMin (on the right)

Multiplications ($J_m = 2$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 3$)**

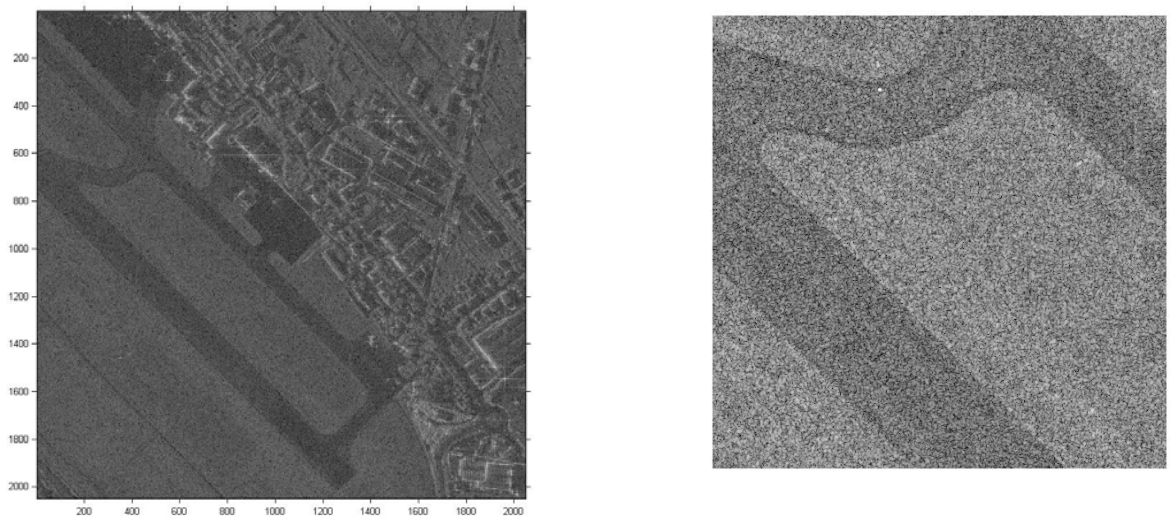


Spline

Fig. 4.42 - Spline wavelet combination (top) and Otsu 3 classes hysteresis thresholding (bottom), SM (on the left), SS (on the centre), SMin (on the right)

4.5.1.2 CSK Image (CSK1)

In order to appreciate filtering performance in real SAR images, the runway detail in Fig. 4.43 (b), which belongs to the "CSK1" image shown in Fig. 4.43 (a) has been used (see Section 1.9.2). This image, which depicts an airport, has been acquired in spotlight2 mode with polarization **HH** and incidence angle near/far of **55.9/56.3 (degrees)** and has a theoretical equivalent number of look equal to **1**. The selected image detail has been chosen because it contains linear (diagonal) and curvilinear edges jointly.



(a) (b)

Fig. 4.43 - (a) Logarithm (base e) of the Intensity "CSK1" image. (b) Logarithm (base e) of the runway detail (512x512 px)

In Fig. 4.44 - Fig. 4.49 the results in applying previous combination methods (SM, SMin, SS) are reported above the combined image. Since final results are very sensitive to the thresholding method, the threshold method reported in (4.81) and (4.82) has been added as further comparison (see the last column of all the results in Fig. 4.44 - Fig. 4.49). This method, which is referenced as "standard thresholding" method, selects the pixel (x_0, y_0) as strong edge pixel only if some conditions holds. Indicating for simplicity the 2-D wavelet transform modulus at scale 2^j and pixel (x, y) as $M_{2^j}f(x, y)$, the "standard thresholding" method, selects the pixel (x_0, y_0) as strong edge pixel only if:

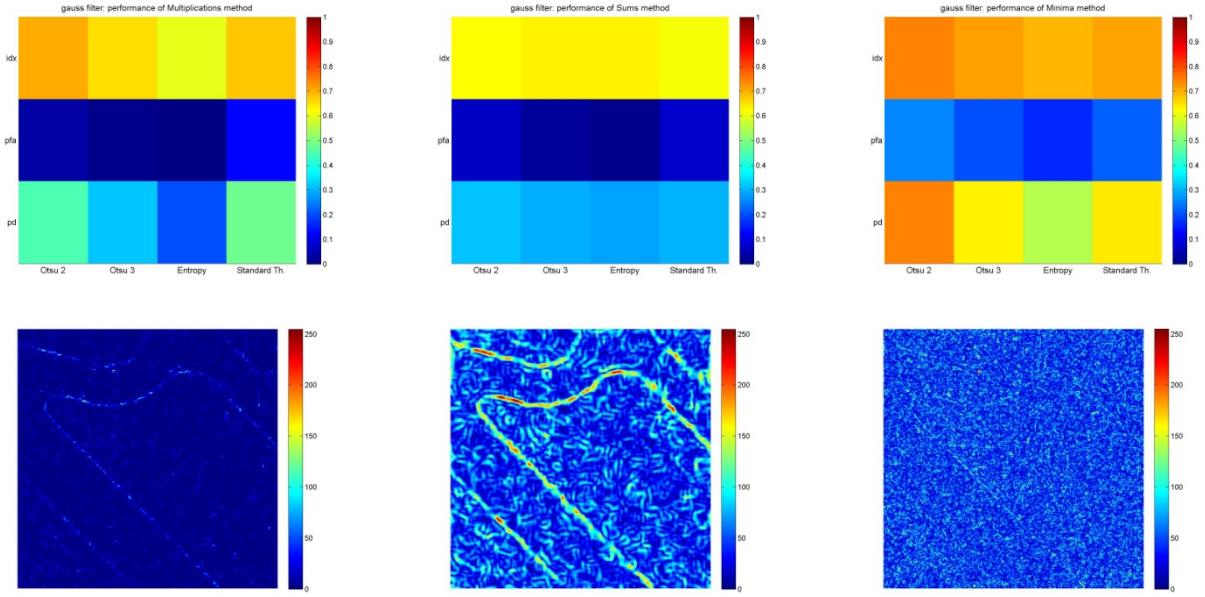
$$M_{2^j}f(x_0, y_0) > E\left[M_{2^j}f(x, y)\right] \;\; AND \;\; M_{2^j}f(x_0, y_0) > E\left[M_{2^j}f(x, y)\right] \Big|_{(x,y)\in N_{10}} \tag{4.81}$$

The equation can be clarified as follows: a pixel $(x_0, y_0)$is an edge pixel if and only if his wavelet module value is greater than both the global mean and the local mean of wavelet transform modulus computed in a $10x10$ neighbourhood (indicated as N_{10}) centred at that pixel. Then, the weak threshold for the hysteresis thresholding is computed as:

$$M_{2^j}f(x_0, y_0) > \frac{E\left[M_{2^j}f(x, y)\right]}{100} \;\; AND \;\; M_{2^j}f(x_0, y_0) > E\left[M_{2^j}f(x, y)\right] \Big|_{(x,y)\in N_{10}} \tag{4.82}$$

Anyway, even with this thresholding method the combined wavelet coefficients do not show any improvement.

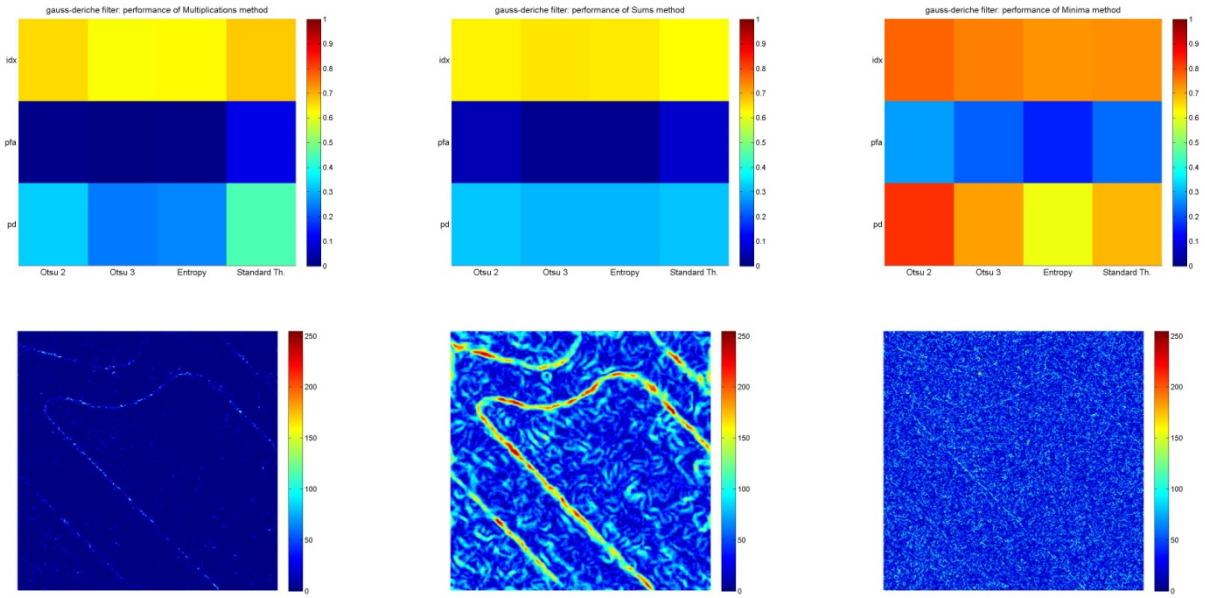
Multiplications ($J_m = 3$)	**Sums ($J_{ss} = 3$)**	**Minima ($J_{\min} = 2$)**



Canny

Fig. 4.44 - Canny wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

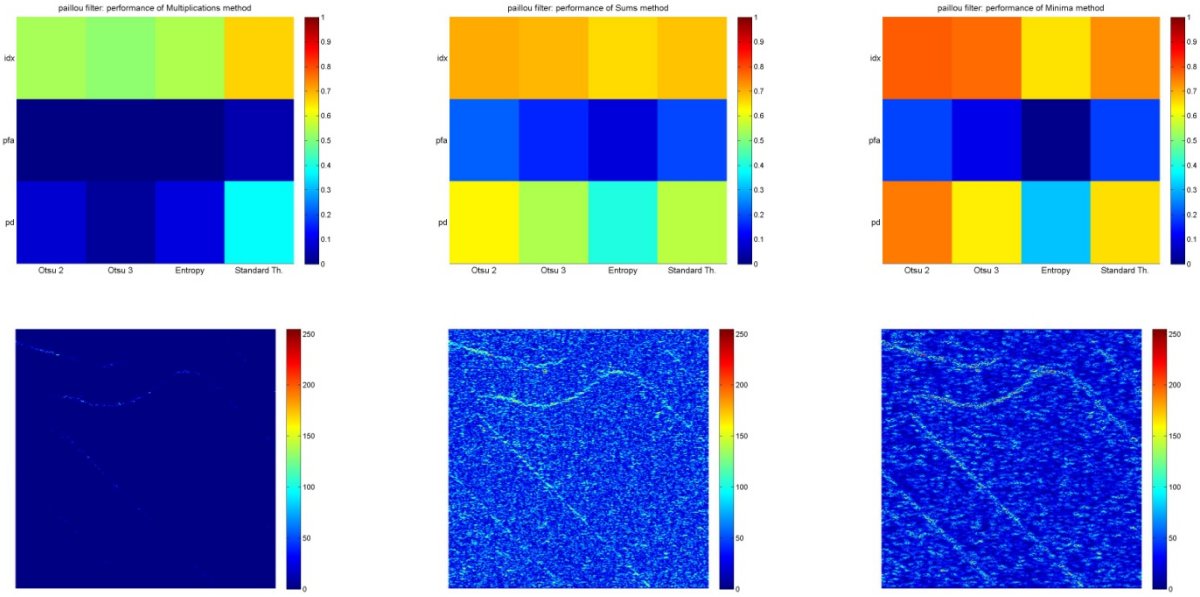
Multiplications ($J_m = 4$)	**Sums ($J_{ss} = 4$)**	**Minima ($J_{\min} = 3$)**



Gauss-Deriche

Fig. 4.45 - Deriche wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

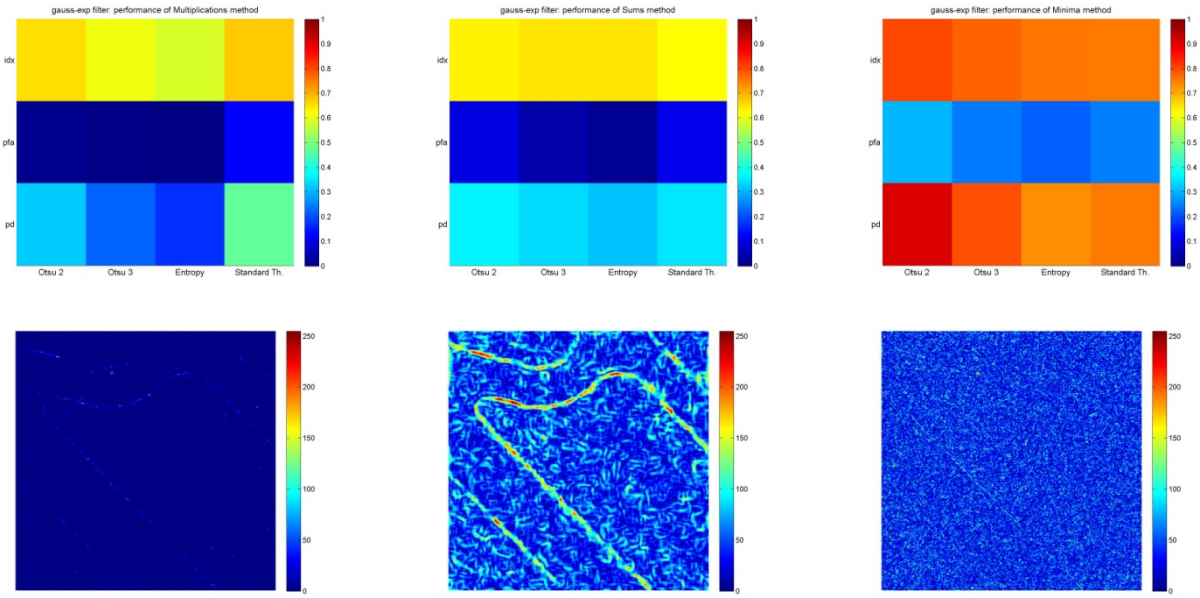
Multiplications ($J_m = 6$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 4$)**



Paillou

Fig. 4.46 - Paillou wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

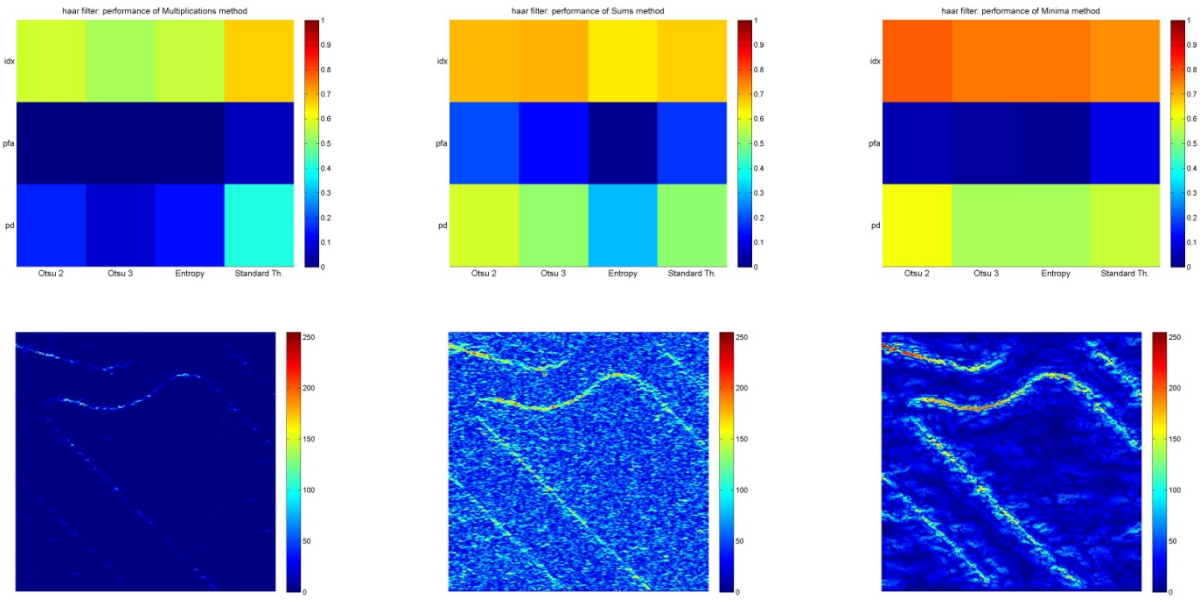
Multiplications ($J_m = 4$)	**Sums ($J_{ss} = 4$)**	**Minima ($J_{\min} = 3$)**



Gauss - Shen-Castan

Fig. 4.47- Gauss - Shen-Castan wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

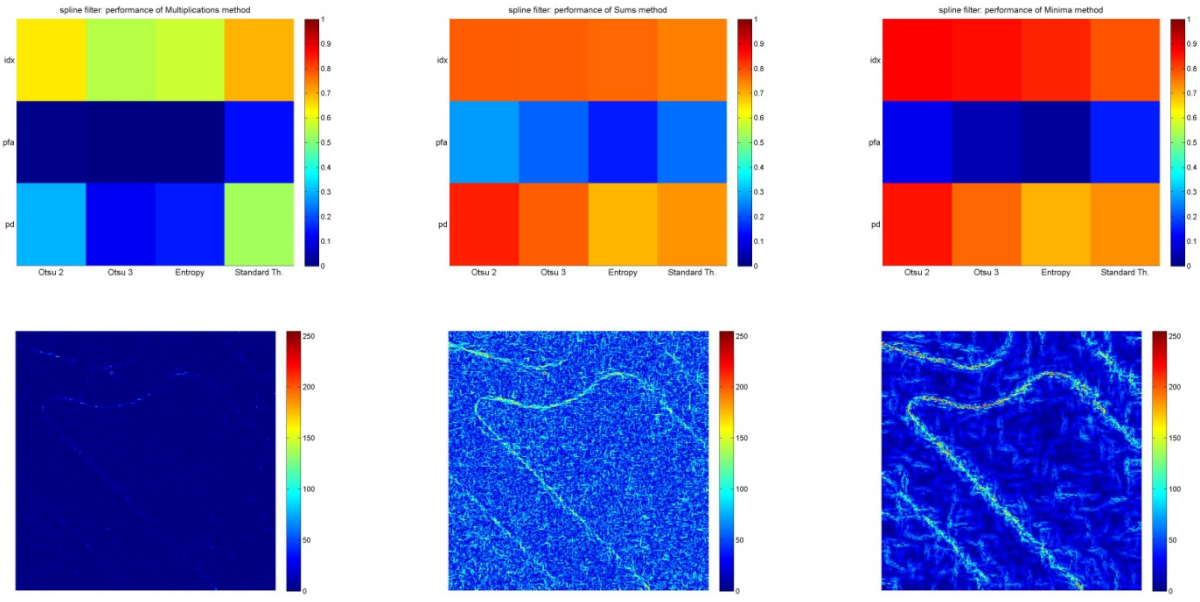
Multiplications ($J_m = 4$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 5$)**



Haar

Fig. 4.48 - Haar wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

Multiplications ($J_m = 4$)	**Sums ($J_{ss} = 5$)**	**Minima ($J_{\min} = 5$)**



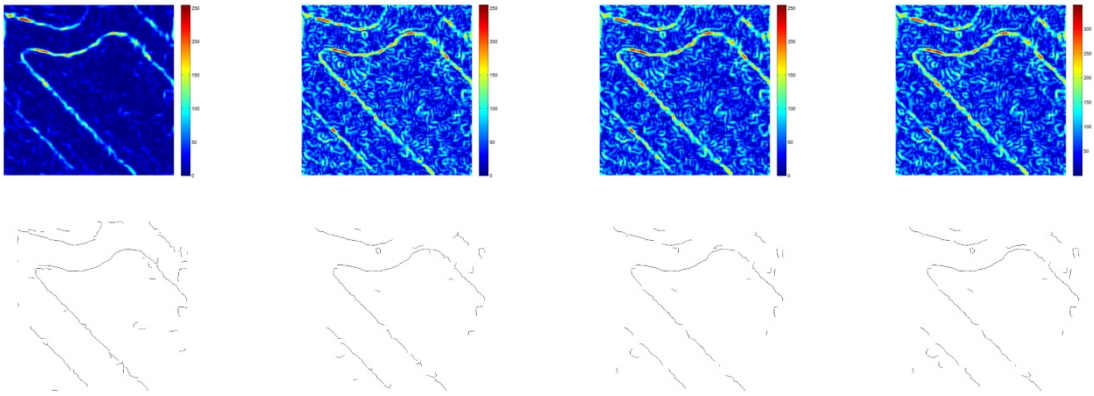
Spline

Fig. 4.49 - Spline wavelet combination results (top) combined image (bottom), SM (on the left), SS (on the centre), SMin (on the right).

Nevertheless, analysing the results obtained by filters at each scale, **we can see that only the central scales (third, fourth and fifth) contains the desired information about edges**. In fact, at smaller scales speckle noise is considered as edge and at larger scales only large distorted features are present. For these reasons, only the combinations among these scales are considered hereafter. Moreover, no one of the previous thresholding methods manage to select properly a threshold and final results strongly depend on the presence (or absence) of edges in the image (e.g. Otsu and Entropy algorithms always presuppose the existence of two or more classes). To solve this problem **we propose a semi-automatic thresholding method that relies on a homogeneous region in the image**. In particular, given a binary mask of a homogeneous region, at each scale the histogram of the wavelet coefficients relative to the homogeneous region (i.e. noise) can be easily computed. In this way, passing a percentile value Q as input, we can compute the corresponding value that enables us to have the desired false alarm rate (FAR) in output. Furthermore, even if no edges exist in the input image, the selected threshold would be correct. In Fig. 4.50 and Fig. 4.51 **we report the results of the two best filters: Canny and Spline,** when the semi-automatic thresholding method is used (with hysteresis between the two thresholds computed by the percentile $Q_1 = 0.999$ and $Q_2 = 0.9$) and only two scale J_1, J_2 are considered in the combination. From these results follows that **multiplication between the fourth and fifth scale** enable us to **remove most noise without degrading the important edges**. Furthermore, comparing the last column of each figure with the combination methods we can see that only through multiplication we can yield better performance than that obtained by a single scale. Moreover, while Canny shows good results, Spline filter gives more noisy results. Anyway, despite the Canny filter, for Spline edge detector a fast implementation exist, i.e. Spline can be considered a fast approximation of Canny filter.

In order to better visualize the best result, in Fig. 4.52 the canny multiplication method is reported beside the original CSK image. Nevertheless, form all experiments carried out in this study, it follows that **wavelet filters for edge detection are sensitive to noise**, and even using multiscale information does not produce a lot of improvement. In particular, **whereas on simulated image with one number of looks the results are very good, CSK images are very challenging for the edge detection issue and only high scales ($J \geq 3$) are not completely dominate by noise**. Anyway, at these high scales small details are not present any more, causing an appreciable loss of information. **For these reasons, a pre-processing step is needed before performing edge detection on high speckled SAR images, even when wavelet filters are considered**.

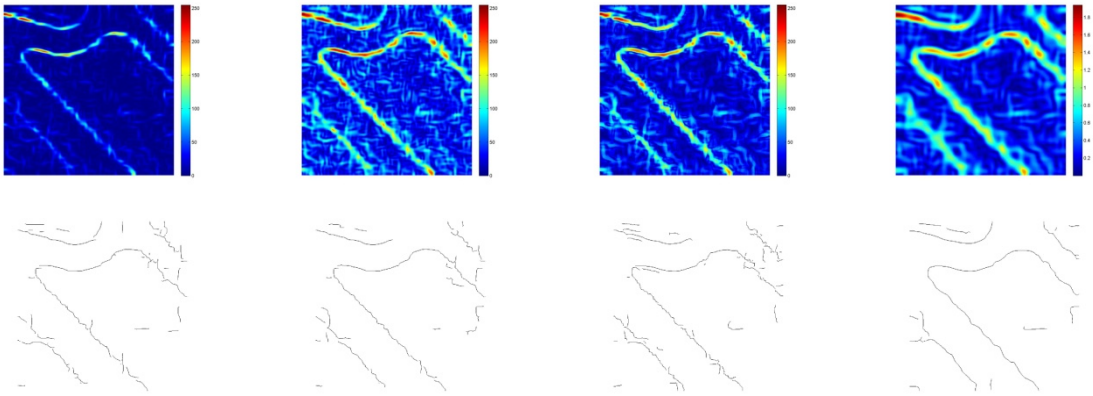
Multiplications :	**Sums :**	**Minima :**	**Best scale:**
$J_1 = 4; J_2 = 5$	$J_1 = 3; J_2 = 4$	$J_1 = 4; J_2 = 5$	$J = 4$



Canny

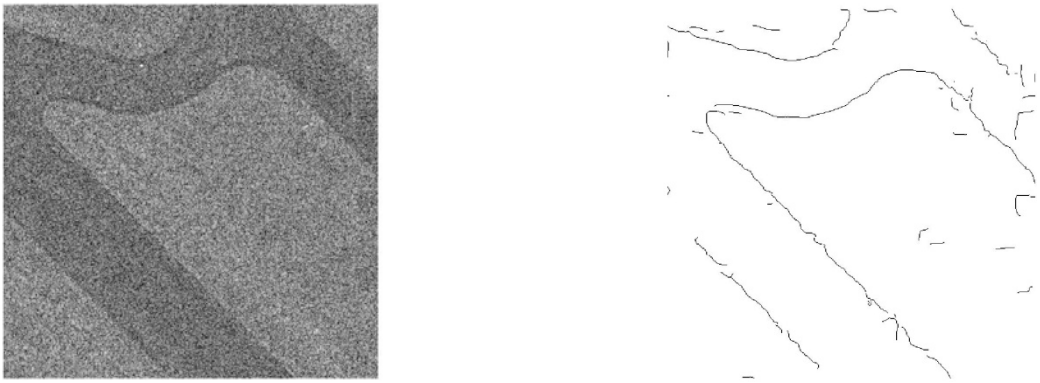
Fig. 4.50 - Canny wavelet combined image (top), thresholded image (bottom).

Multiplications :	**Sums :**	**Minima :**	**Best scale:**
$J_1 = 4; J_2 = 5$	$J_1 = 3; J_2 = 4$	$J_1 = 4; J_2 = 5$	$J = 5$



Spline

Fig. 4.51 - Spline wavelet combined image (top), thresholded image (bottom).



(a) (b)

Fig. 4.52 - (a) Original CSK image detail. (b) Canny multiplication method ($J_1 = 4; J_2 = 5$).

4.6 Computational Load

To perform a comparison among the various filter implementations, computational times for various image sizes are reported in Tab. 4.2. The values reported in such table are the execution time needed to compute the first three scale of the wavelet decomposition. The analysis has been performed on a computer **Intel Core 2 Duo, CPU T9400 @2.53GHz, 1.99GB di RAM**. As can be clearly seen from these results, IIR implementation is convenient for image size lesser or equal than 512x512 pixels, whereas circular convolution (CCONV) is more efficient for the other cases. Between the previous implementation versions, Shen-Castan filter is the fastest, even though Harr filter overcomes it on 1024x1024 images when CCONV is used. Next, as expected, "a trous" algorithm (Section 4.3.7) is the fastest in absolute.

		Computational Time (s)		
	Implementation	**Image 256x256 (px)**	**Image 512x512 (px)**	**Image 1024x1024 (px)**
Canny	*CCONV*	1.01	2.54	14.28
Deriche	*CCONV*	1.00	2.57	13.30
	IIR	0.60	1.73	20.42
Paillou	*CCONV*	0.99	2.46	13.16
	IIR	0.58	1.68	18.90
Shen-Castan	*CCONV*	0.98	2.49	14.27
	IIR	*0.52*	*1.56*	14.89
Haar	*CCONV*	1.01	2.57	*13.07*
	IIR	0.70	2.16	19.81
Spline (*m*=2)	*A TROUS*	**0.31**	**0.80**	**10.35**

Tab. 4.2 - Computational load of edge detection filters (mean time for compute the initial three scales). Bold font indicates the best value among all filters. Italic, underlined font indicates the best within the same "class"