Chapter 7

Despeckling Filters

In Chapter 4 we have proved that **wavelet filters for edge detection are sensitive to noise**, and even using multiscale information does not produce a lot of improvement on the challenging CSK images that have one number of looks. Before claiming the need of a pre-processing step, which would be useful to remove the noise afflicting SAR images, we should test statistical filter in Chapter 3 on these data.

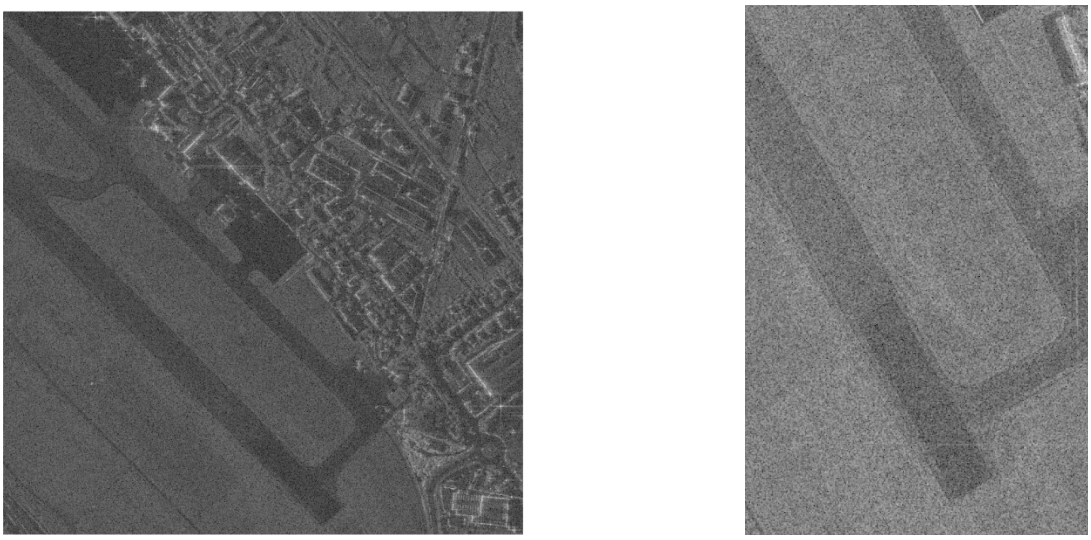
Two Cosmo-SkyMed (CSK) images, which depict the same airport, are shown in Fig. 7.3. Both the images in Fig. 7.3(a) and Fig. 7.3(c) have been acquired in spotlight2 mode with polarization **HH** and a theoretical equivalent number of look equal to **1.** The image in Fig. 7.3(a) has an incidence angle near/far of **55.9°/56.3°** whereas the image in Fig. 7.3(c) has an incidence angle near/far of **56.4°/56.8°.**

As we can see from Fig. 7.2(a)-(b), the image in Fig. 7.3(a) shows a pdf nearer to a Gamma than a LogNormal indicating the possibility to use RoA edge detector. Moreover, the image shows a correlation length of about 2 pixels in both directions, see Fig. 7.2(c), and therefore it needs to be subsampled of the same quantity before applying the edge detector. Actually, the downsampling is needed only to compute an exact threshold. The, even T-test and W-test were applied to these images. Since results were always worse than RoA ones, they are not reported in this document.

The results of RoA edge detector on the previous two images are shown in Fig. 7.3. As can be seen from these results, even allowing a lot of false alarms to be detected, some edges are missed .

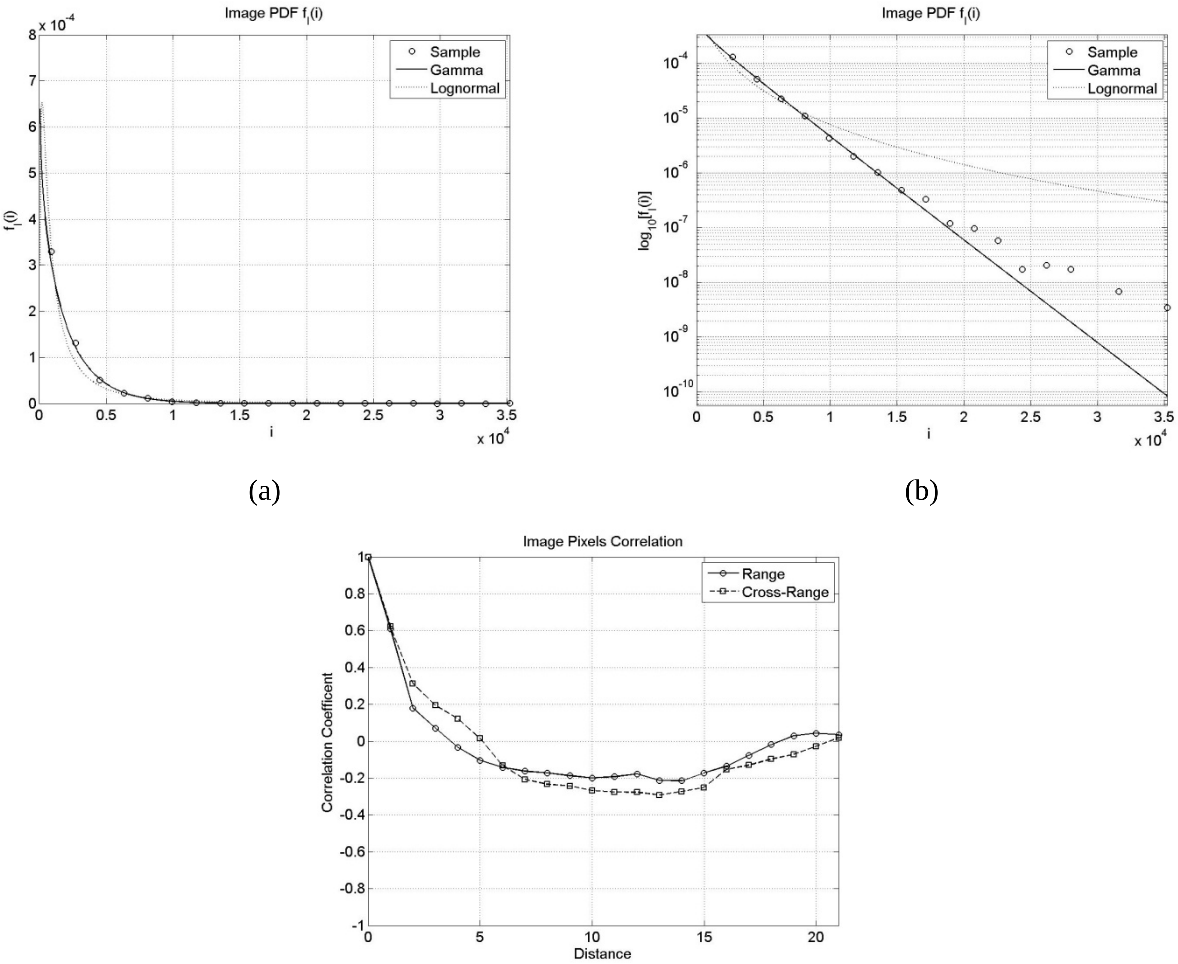
The **unsatisfactory results of statistical edge detectors** on these CSK images are mainly due to two reasons. First, the low signal to noise ratio (estimated number of look[11] ~ 0.8). Secondly, the low contrast between the boundary that divides the runway from the surrounding terrain. The first problem, i.e. the low signal to noise ratio, makes worse the RCS estimates made by statistical filters, that is, the estimates have a high variance. As consequence, the probability of false alarm (PFA) raises and the probability of detection (PD) decreases. The second question is purely theoretical, because to detect a lower contrast edge with a fixed PD we have to obliviously increase PFA. Nevertheless, the contrast between two materials with different roughness can be enhanced increasing the incidence angle in the image acquisition (see Appendix L for a detailed analysis). Nevertheless, even though the second problem is due to the sensor characteristics and, fixed a mode of acquisition, it is not solvable, the high speckle noise presents in these images can be profitably reduced exploiting the so called despeckling filters.

[11] The number of looks parameter has been estimated (by MoM assuming a multiplicative model) on an extended visually homogeneous region.



(a) (b)

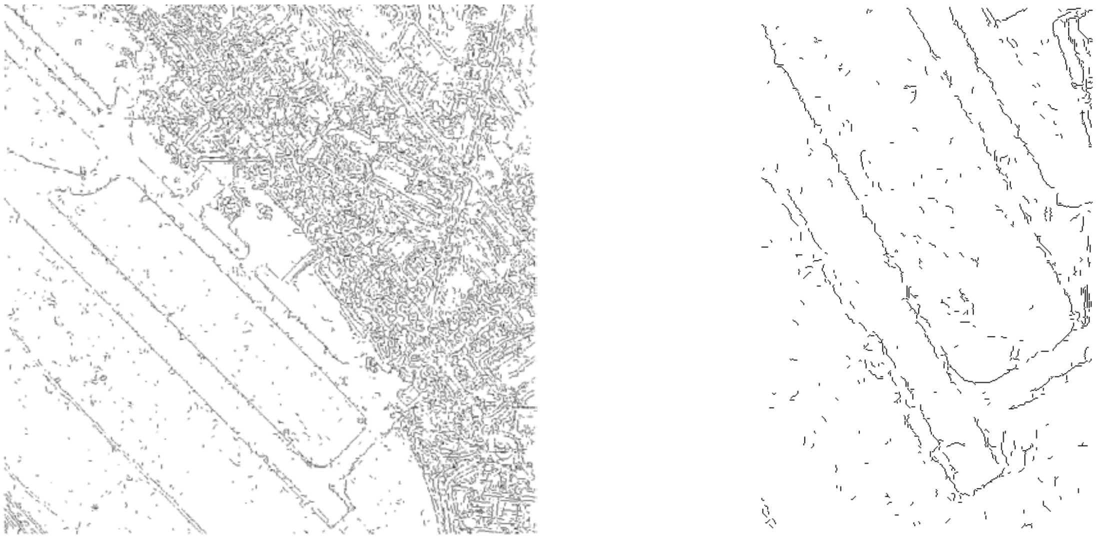
Fig. 7.1 - CSK images. (a) Image “CSK1”. (b) Image “CSK2”.



(a) (b)

(c)

Fig. 7.2 - (a) PDF of the image “CSK1” on a homogeneous region (terrain around runway). (b) Logarithmic scaled PDF of the image “CSK1” on a homogeneous region (terrain around runway). (c) Pixel correlation estimated on a homogeneous region (terrain around runway) of “CSK1” image



(a) (b)

Fig. 7.3 - Result of RoA filter on two different CSK images. (a) Result of RoA filter (window 11x11) on “CSK1”. (b) Result of RoA filter (window 11x11) on “CSK2”.

From the previous examples, we have seen that CSK images suffer from a high power speckle noise that even statistical edge detector do not manage to tackle successfully. For this reason, also considering that even multiscale linear filters are sensitive to noise, the need of introducing a despeckling filter before applying an edge detection step becomes clear.

Despeckling filters are also called **RCS reconstruction filters** [3]. In fact, their aim is of recovering the underlying RCS from the noisy data introducing the minimum amount of distortions. Criterions that drive the judgement on the filter quality are [3]:

- speckle reduction in homogeneous regions;
- feature preservations;
- no artefact introduction;
- radiometric data preservation.

Moreover, the choice of a filter is always driven by the **final objective**, i.e. it depends on applications. In our case, the filter has **to reduce the noise without blur the object boundaries**. As will be clear from the following discussions, satisfying this type of demand is strongly related to the quality of edge detection reachable from the initial noisy image. In the rest of the section a several classes of despeckling filter are described introducing, when possible, only the main concepts which underlying their theory.

7.1 “Classic” local Filters

With the adjective “classic” we want to point out the high degree of acceptance of this type of filter by the scientific community. This class of filters counts the Lee, Kuan, Frost, and Gamma MAP despeckling filters. All these filters exploit the pixel values inside a small window centered at a given pixel in order to make inference and to reconstruct its “true” value. In particular, they presuppose the image as an **ergodic random process** where statistical means can be substituted by spatial means. In the following some quantities are introduced to give exact reconstruction formulas.

7.1.1 Noise Model

In all the cases, if not expressively indicated, the noise model considered (on a homogeneous region) is a multiplicative one (see 2.2.1.1), the image I holds intensity values, the noise is indicated with n and it is independent from the Radar Cross Section (RCS) σ [12]:

$$I = \sigma n \tag{7.1}$$

where I indicates the observed intensity (square of the amplitude, i.e. received power), and it is further presupposed:

$$\mathrm{E}[n] = 1;\ \sigma_n^2 = 1/L \tag{7.2}$$

with $\mathrm{E}[n]$ and σ_n^2 respectively the noise mean and variance and L indicates the number of looks (or equivalent number of looks - ENL).

7.1.2 Lee and Kuan

Under the constraint to use a linear estimator (Kuan [58]) or using a first order Taylor expansion of image I (Lee [59]), the reconstructed minimum mean square error value $\hat{I}$ can be written as:

$$\hat{I} = \bar{I} + k(I - \bar{I}) \tag{7.3}$$

where I indicates the original noisy value, $\bar{I}$ is the mean of the values inside the window and k is a factor which depend on the SNR estimated through the pixels inside the window. In particular:

$$k_L = \frac{C_I^2 - C_n^2}{C_I^2 + C_n^4};\ k_k = \frac{C_I^2 - C_n^2}{C_I^2(1 + C_n^2)} \tag{7.4}$$

where k_L and k_k are the factors of Lee and Kuan filter respectively, $C_I^2$and C_n^2 are the coefficients of variation of the pixels inside the window and that of the noise, respectively computed as:

$$C_I^2 = \frac{\sigma_I^2}{{\mu_I}^2};\ C_n^2 = \frac{\sigma_n^2}{{\mu_n}^2} \tag{7.5}$$

with σ_I^2, σ_n^2 the two respective variances and μ_I, μ_n the respective means. Usually C_n^2 is not computed from the image pixels but it is taken from the noise model previously reported. It should be point out that the Lee formula in Eq. (7.4) is the corrected one. In fact, in literature, due to a mistake in the Lee original article a wrong version of the formula is sometimes reported (as in [3]), although some authors have underlined this mistake [60].

7.1.3 Frost

The frost filter $h(x)$ is the best unbiased *linear* filter which minimizes the mean square error between the original image and the reconstructed one [61], [62]:

$$\min\left[\left|\sigma(x) - \hat{I}(x)\right|^2\right] = \min\left[\left|\sigma(x) - \big(h(x) * I(x)\big)\right|^2\right] \tag{7.6}$$

It presupposes a multiplicative white noise and an underlying RCS inside a window as a wide sense stationary (WSS) stochastic process with an exponentially decay covariance (this type of decay has been extensively used in literature to model a large class of textured region [3], [62]). Given this assumption, a closed form of h can be computed as:

$$h(x) = \frac{\alpha}{2} e^{-\alpha x} \tag{7.7}$$

Now, exploiting the previous hypotheses, indicating with $C_{\sigma\sigma}(\Delta x)$, and $C_{nn}(\Delta x)$ the covariance of the reflectivity and of the noise respectively:

[12] Not to be confused with the variance of a general r.v. a that in the rest of the document will be indicated with the corresponding subscript, i.e σ_a.

$$C_{\sigma\sigma}(\Delta x) = \sigma_\sigma^2 e^{-\lambda|\Delta x|};\;\; C_{nn}(\Delta x) = \sigma_n^2 \delta(\Delta x) \tag{7.8}$$

with λ the correlation length of the reflectivity, we can express the filter parameter α as:

$$\alpha^2 = \frac{2\,\lambda/\sigma_n^2}{1 + (\mu_\sigma/\sigma_\sigma)^2} + \lambda^2 \tag{7.9}$$

Naturally, in the multiplicative noise model the previous quantities can be easily computed:

$$\mu_\sigma = \mu_I;\;\; \sigma_\sigma^2 = \frac{L\sigma_I^2 - \mu_I^2}{L+1} \tag{7.10}$$

Anyway, since the correlation length λ is computationally expensive to estimate, in literature a sub-optimal formulation is used, where λ is omitted, σ_n^2 is not used and a parameter β is added, so that α simply becomes [64]:

$$\alpha^2 = \beta C_I^2 \tag{7.11}$$

However, the results reported in this document are obtained with the optimal formula where λ is let as free parameter.

7.1.4 Gamma MAP

Differently from the previous filters, the Gamma MAP does not linearly operate to estimate the original value from the noisy data [3]. It presupposes the observed value of the intensity image as an occurrence of a random variable I with a conditional probability density function pdf $P(I|\sigma)$ modelled as a Gamma distribution. The Gamma pdf of the intensity, indicated as $I\sim\Gamma(L,\sigma/L)$, depends on the two parameters L and σ which are related to image first order statistics as:

$$\mu_I = \sigma;\;\; \sigma_I^2 = \frac{\sigma^2}{L} \tag{7.12}$$

Moreover, it models the pdf $P(\sigma)$ of the reflectivity σ, as another Gamma distribution $\sigma\sim\Gamma(v,\mu_\sigma/L)$ where:

$$\mu_\sigma = \mathrm{E}[\sigma];\; \sigma_\sigma^2 = \frac{\mu_\sigma^2}{v} \tag{7.13}$$

Under these hypotheses (see Section 2.3), the pdf of the image intensity is a K-distribution $I\sim\mathrm{K}(L,v,\mu_\sigma)$ with [5]:

$$\mu_I = \mu_\sigma;\;\; \sigma_I^2 = \mu_\sigma^2\left(\frac{1}{L} + \frac{1}{v} + \frac{1}{Lv}\right) \tag{7.14}$$

Naturally, in order to extract the Maximum a Posteriori (MAP) probability of the reflectivity R we can exploit the well-known Bayes formula:

$$P(I|\sigma) = \frac{P(I|\sigma)\,P(\sigma)}{P(I)} \propto\; P(I|\sigma)\,P(\sigma) \tag{7.15}$$

and the MAP estimate $\hat{\sigma}_{\mathrm{MAP}}$ is obtained as solution of : $\frac{d}{dR}\left[\log p(I|R) + \log p(R)\right] = 0$, i.e.:

$$\begin{aligned} &\frac{\partial}{\partial\sigma}[\,\ln P(I|\sigma) + \ln P(\sigma)] = 0 \\ &\rightarrow\; \hat{\sigma}_{\mathrm{MAP}}\frac{v}{\mu_\sigma} + \hat{\sigma}_{\mathrm{MAP}}(L + 1 - v) - LI = 0 \end{aligned} \tag{7.16}$$

where the parameter v is substituted which its estimate:

$$\hat{v} = \frac{1 + C_n^2}{C_I^2 - C_n^2} \tag{7.17}$$

7.1.5 “Classic” Refined Versions

Even though all the “Classic” filters described in Section 7.1 manage to reduce the speckle on homogeneous areas, they completely fail in RCS reconstruction every time an edge is inside the local window. The final result is an image whose sharp transitions have been completely smoothed out. In order to remove the noise without blurring the edges, the case of a sharp transition inside the local window has to be treated in a different way. The version of the classic filters where this fine consideration is applied are usually called “refined” or “enhanced” version. To implement this type of refinement Lee [65] suggest of using a simple gradient operator inside the local window. Anyway, as he admits, this type of operator suffers of the well-known problem of not being CFAR. In order to overcome this issue, [66] suggests using statistical edge detectors because, in addition to being CFAR, they enable us to easily compute a threshold fixing the desired PFA. In fact, under the hypothesis of independent samples, we can use RoA edge detector when the pdf of the image is supposed to be Gamma, the T-test if the image pdf is Gaussian, or eventually, the W-test if the previous hypotheses do not hold Section 3.2. Clearly, when there is some correlation between pixels we can still use these edge detectors even though the threshold parameter cannot be tied to the final PFA anymore.

7.2 Patch Based Filter

This class of filters exploits the redundancy that often exists in natural images. They are based on the principle that the reconstructed value has to be the weighted mean of all the image pixels which have a similar neighbourhood [67]. Given the image domain Ω, a neighbourhood $\Omega_{\boldsymbol{x}}$ of a pixel $\boldsymbol{x}$, is defined as a set of pixels $\{\boldsymbol{y}\} \in \Omega_{\boldsymbol{x}}$ around $\boldsymbol{x}$, e.g. contained in a window (also called “patch”). In a continuous case the formulation becomes [68]:

$$\begin{cases} \hat{I}(\boldsymbol{x}) = \dfrac{1}{C(\boldsymbol{x})}\displaystyle\int_{\Omega} I(\boldsymbol{y}) e^{-\frac{K(\sigma,\|I(\boldsymbol{x})-I(\boldsymbol{y})\|)}{h}} d\boldsymbol{y} \\ K(\sigma, \|I(\boldsymbol{x}) - I(\boldsymbol{y})\|) = \displaystyle\int_{\Omega} G_{\sigma}(\boldsymbol{t}) |I(\boldsymbol{x}+\boldsymbol{t}) - I(\boldsymbol{y}+\boldsymbol{t})|^2 d\boldsymbol{t} \\ C(\boldsymbol{x}) = \displaystyle\int_{\Omega} e^{-\frac{K(\boldsymbol{z})}{h}} d\boldsymbol{z} \end{cases} \tag{7.18}$$

where $K(\sigma, \|I(\boldsymbol{x}) - I(\boldsymbol{y})\|)$ is a measure of similarity among different patches and it depends on a norm between grey levels and the standard deviation σ of the Gaussian kernel $G_{\sigma}(\boldsymbol{t})$. $C(\boldsymbol{x})$ is a normalization constant and $exp\left(-\frac{K(\sigma,\|I(x)-I(y)\|)}{h}\right)$ express the weights used to compute the reconstructed value.

Hence, the working principle is slightly different from that previously enounced, i.e. the reconstructed value has to be the *exponentially negative* weighted mean of all the image pixels that have a similar *Gaussian smoothed* neighbourhood and with weights direct proportional to the similarity measure $K(\sigma, \|I(\boldsymbol{x}) - I(\boldsymbol{y})\|)$.

Since these filters use, in principle, all image pixels to define a singular weight they are also known as Non Local Means (NL-Means) [68]. Anyway, for implementation purpose, the pixels used in computations do not belong to all image domain Ω but at some restricted portion Ω_1. Therefore, Eq. (7.18) becomes:

$$\begin{cases} \hat{I}(\boldsymbol{x}) = \dfrac{1}{C(\boldsymbol{x})}\displaystyle\int_{\Omega_1} I(\boldsymbol{y}) e^{-\frac{K(\sigma,\|I(\boldsymbol{x})-I(\boldsymbol{y})\|)}{h}} d\boldsymbol{y} \\ K(\sigma, \|I(\boldsymbol{x}) - I(\boldsymbol{y})\|) = \displaystyle\int_{\Omega_2} G_{\sigma}(\boldsymbol{t}) |I(\boldsymbol{x}+\boldsymbol{t}) - I(\boldsymbol{y}+\boldsymbol{t})|^2 d\boldsymbol{t} \end{cases} \tag{7.19}$$

where Ω_1 is the so called “search window”, centred on $\boldsymbol{x}$ and usually with size 21x21 pixels, whereas Ω_2 is the so called “similarity window”, sliding on different $\boldsymbol{x} \in \Omega_1$ and with dimension 7x7 pixels.

7.2.1 Iterative Probabilistic Patch Based

In order to overcome some open questions on patch based filters, which mainly regard a theory to choose a similarity measure and a weight function appropriately, the authors in [69] propose to use the most common framework to assign weights: probability. The discrete version of the filter, considering the image as a vector which depends on the spatial location index i, can be written as:

$$\hat{I}(i) = \frac{\sum_{t\in\Omega_1} w(i,j)I(i)}{\sum_{t\in\Omega_1} w(i,j)} \tag{7.20}$$

where $w(i,j)$ is a weight function which depends on general discrete space location i and j inside the search window Ω_1. In the case of NL-Means:

$$w(i,j) = e^{-\frac{\sum_{t\in\Omega_2} g_t \left|I_{i,t}-I_{j,t}\right|^2}{h}} \tag{7.21}$$

where $\{g_t\}_{t\in\Omega_2}$ are the coefficients of a Gaussian smoothing and Ω_2 is the similarity window. In [69] $w(i,j)$ has been defined as:

$$w(i,j) = \left[P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|I\right)\right]^{\frac{1}{h}} \tag{7.22}$$

where $\Omega_{2,s}$ is the similarity window centered at pixel s. Hence, the weights are proportional to the probability that the "true" reflectivity σ is equal inside the two similarity windows, conditioned to the observed values I inside the windows themselves. Presupposing independence between pixels in the two windows with same dislocation t:

$$P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|I\right) = \prod_{t\in\Omega_2} P\left(\sigma_{i,t} = \sigma_{j,t}\middle|I_{i,t}, I_{j,t}\right) \tag{7.23}$$

and, using the Bayes formula:

$$\overbrace{P\left(\sigma_{i,t} = \sigma_{j,t}\middle|I_{i,t}, I_{j,t}\right)}^{Posteriori} = \frac{\overbrace{P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)}^{ML}\overbrace{P\left(\sigma_{i,t} = \sigma_{j,t}\right)}^{Priori}}{\underbrace{P\left(I_{i,t}, I_{j,t}\right)}_{Normalization}} \tag{7.24}$$

they put the a posteriori probability not proportional to the ML probability but at the normalization:

$$\overbrace{P\left(\sigma_{i,t} = \sigma_{j,t}\middle|I_{i,t}, I_{j,t}\right)}^{Posteriori} \propto \overbrace{P\left(I_{i,t}, I_{j,t}\right)}^{Normalization} = \int_S P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)P\left(\sigma_{i,t} = \sigma_{j,t}\right)d\sigma \tag{7.25}$$

with S the definition domain of the RCS. Practically, contrarily to what maintained in the article, they express the weights in Eq. (7.22) on ML:

$$w(i,j) = \left[P\left(I\middle|\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\right)\right]^{\frac{1}{h}} \tag{7.26}$$

and now, using the Bayes formula, ML is proportional to the normalization:

$$\overbrace{P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)}^{ML} \propto \overbrace{P\left(I_{i,t}, I_{i,t}\right)}^{Normalization} = \int_S P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)P\left(\sigma_{i,t} = \sigma_{j,t}\right)d\sigma \tag{7.27}$$

Finally, exploiting the independence between pixels in the two windows with same dislocation t, and leaving out the a priori probability $P\left(\sigma_{i,t} = \sigma_{j,t}\right)$ inside the integral:

$$\int_S P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)P\left(\sigma_{i,t} = \sigma_{j,t}\right)d\sigma \propto \int_S P\left(I_{i,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)P\left(I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)d\sigma \tag{7.28}$$

Hence, the final relations are:

$$\begin{cases} w(i,j) = \left[P\left(I\middle|\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\right)\right]^{\frac{1}{h}} = \left[\prod_{t\in\Omega_2} P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)\right]^{\frac{1}{h}} \\ P\left(I_{i,t}, I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right) \propto \int_S P\left(I_{i,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)P\left(I_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right)d\sigma \end{cases} \tag{7.29}$$

This weight function is used when only one iteration is computed. For the iterative version a little more complication is added. Practically, they use at step k the previous parameter estimation $\hat{\sigma}^{k-1}$ in the following way:

$$w(i,j) = \left[P\left(I\middle|\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\right)P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|\hat{\sigma}^{k-1}\right)\right]^{\frac{1}{h}} \tag{7.30}$$

where the new term $P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|\hat{\sigma}^{k-1}\right)$, which measures the probability to have equal RCS in the two similarity windows conditioned to the previous pixel estimate, is expressed as a Kullback-Leibler divergence over an exponential decay function[13] [69]:

$$P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|\hat{\sigma}^{k-1}\right) \propto e^{-\frac{1}{T}\int_S \left[P\left(t\middle|\hat{\sigma}_{i,t}^{k-1}\right)-P\left(t\middle|\hat{\sigma}_{j,t}^{k-1}\right)\right]\ln\frac{P\left(t\middle|\hat{\sigma}_{i,t}^{k-1}\right)}{P\left(t\middle|\hat{\sigma}_{j,t}^{k-1}\right)}dt} \tag{7.31}$$

where T is a free positive parameter.

7.2.1.1 Multiplicative Noise Model

For a multiplicative Gamma noise of the intensity, the amplitude $A = \sqrt{I}$ is a Nakagami distribution indicated as $A \sim \text{Nakagami}(L, \sigma/L)$:

$$f_A(a) = \frac{2a^{2L-1}}{\Gamma(L)(\sigma/L)^L} e^{-\frac{a^2}{\sigma/L}}; \quad a \geq 0 \tag{7.32}$$

and from this distribution we can compute the following weights:

$$\begin{cases} P\left(A_{i,t}, A_{j,t}\middle|\sigma_{i,t} = \sigma_{j,t}\right) \propto \left(\frac{A_{i,t}A_{j,t}}{A_{i,t}^2 + A_{j,t}^2}\right)^{2L-1} \\ P\left(\sigma_{\Omega_{2,i}} = \sigma_{\Omega_{2,j}}\middle|\hat{\sigma}^{k-1}\right) \propto e^{-\frac{L}{T}\frac{\left|\hat{\sigma}_{i,t}^{k-1}-\hat{\sigma}_{j,t}^{k-1}\right|^2}{\hat{\sigma}_{i,t}^{k-1}\hat{\sigma}_{j,t}^{k-1}}} \end{cases} \tag{7.33}$$

The interesting thing in this formulation, excluding the probability used to iteratively refining the estimation, is that the weight $w(i,j)$can be written as:

$$w(i,j) \propto e^{-\frac{1}{2L-1}\ln\left(\frac{A_{i,t}^2+A_{j,t}^2}{A_{i,t}A_{j,t}}\right)} = e^{-\frac{1}{2L-1}\ln\left(\frac{A_{i,t}}{A_{j,t}}+\frac{A_{j,t}}{A_{i,t}}\right)} \tag{7.34}$$

where can be seen how $w(i,j)$ depends on a sort of ratio measure (in amplitude) between singular pixels, in a similar way the RoA edge detector operates.

[13] This method is a common measure of distance between two pdfs.

7.3 Wavelet Based Filter

In common with other more diffused signal transforms used in image compression, wavelet based filters exploit powerful properties of wavelet transform in order to denoise images. In fact, it is well-known that a signal can be decomposed in a convenient base in order to have only few coefficients of it different from zero. This compression property can also be exploited in denoising ignoring coefficients lesser than a threshold in reconstruction phase (hard thresholding). However, this abrupt method yields ringing artefacts (Gibbs effects) in reconstructed image and more sophisticated thresholding are applied to the wavelet coefficients to limit these effects (soft thresholding, also called shrinking).

7.3.1 MMSE Denoising on Undecimated Wavelet Transform

As in image domain, one of the most powerful linear filtering to reduce noise is the minimum mean square error (MMSE) filter. For this reason in [70], linear MMSE filtering in wavelet domain is devised on images corrupted by speckle noise.

Considering the one-dimensional (1-D) case (2-D case is straightforward considering separable filters on rows and columns), it is well known that even multiplicative noise can be expressed as an additive, signal-dependent noise as:

$$\begin{aligned} I(x) &= \sigma(x)n(x) \\ &= \sigma(x) + \sigma(x)[n(x) - 1] \\ &= \sigma(x) + \sigma(x)n'(x) \\ &= \sigma(x) + v(x) \end{aligned} \tag{7.35}$$

where, as previously said, being $\mathrm{E}[n] = 1$; $\sigma_n^2 = 1/L$, considering σ and v independent, we have[14]:

$$\mathrm{E}[v] = 0;\ \sigma_v^2 = \mathrm{E}[\sigma^2]\sigma_n^2 + \sigma_\sigma^2\sigma_n^2 \tag{7.36}$$

Presupposing to arrange each sequence in a column vector, and imposing the use of a linear estimator, Kuan [58] expresses the Linear Minimum Square Error estimate of the noiseless reflectivity as:

$$\hat{\boldsymbol{\sigma}}_{LMMSE} = \mathrm{E}[\boldsymbol{\sigma}] + C_{\sigma I}C_I^{-1}(\boldsymbol{I} - \mathrm{E}[\boldsymbol{I}]) \tag{7.37}$$

where $C_{\sigma I}$ and C_I^{-1} are the cross-covariance matrix and covariance matrix respectively. From Eq. (7.37) can be seen as it is only required the knowledge of the second order statistics of $\boldsymbol{\sigma}$ and $\boldsymbol{v}$. Then, exploiting that $\mathrm{E}[\boldsymbol{v}] = 0$ (together with the independence between $\boldsymbol{\sigma}$ and $\boldsymbol{v}$):

$$C_{\sigma I} = C_\sigma;\ C_I = C_\sigma + C_v \tag{7.38}$$

and considering $\boldsymbol{\sigma}$ uncorrelated we have C_R and C_v diagonal, giving the same formula exploited by Kuan in Eq.(7.3)and (7.4) :

$$\hat{\sigma}_{LMMSE}(x) = \mathrm{E}[\sigma(x)] + \frac{\sigma_\sigma^2(x)}{\sigma_\sigma^2(x) + \sigma_v^2(x)}(I(x) - \mathrm{E}[I(x)]) \tag{7.39}$$

hence, only the first-order statistics are involved.

In order to exploit a multiresolution analysis (see Section 4.2 for a detailed discussion), the wavelets theory can be used. Briefly, multiresolution analysis at one level j can be thought as an approximation of a function $f(t)$ to the scale corresponding to that level. Presupposing the common dyadic decomposition, i.e. the scale corresponding to level j is the jth power of 2 (2^j), the projection of a function onto a subspace $V_j \subset L^2(\mathbb{R})$ (space of square integrable functions) can be performed "convolving" (actually is an internal product) $f(t)$ with a *scaling function* $\phi(t)$. The set of dilatations and translations of the scaling function $\left\{\phi_{j,n}(t) = 2^{-j/2}\phi\left(2^{-j}t - n\right)\right\}_{n\in\mathbb{Z}}$ constitutes a base of the subspace V_j. Since $V_j \subset V_{j-1}$, to obtain a projection on V_j from a projection on V_{j-1} we have to add an additional information which lies in a subspace W_j orthogonal to V_j. This latter subspace is called *Wavelet Subspace* and even in this case translations and dilatations of a

[14] It should be noted that in this section, ensemble means and spatial means are indicated differently. For ensemble means we use the expectation operator whereas spatial means are indicated with an upper bar.

wavelet function $\psi(t)$, $\{\psi_{j,n}(t) = 2^{-j/2}\psi(2^{-j}t - n)\}_{n\in\mathbb{Z}}$, constitutes a base of W_j. A digital signal $f[n]$ can be thought as the coefficients $\{f[n]\}_{n\in\mathbb{Z}}$obtained by the projection of a continuous signal $f(t)$ onto the lower resolution subspace V_0. Moreover, thanks to the linear properties of these subspace, projections of digital signal $f[n]$ onto V_j and W_j can be obtained convolving the original sequence with some digital filters $h_0[n]$ and $h_1[n]$, respectively lowpass and highpass (bandpass) filters. Another useful property of wavelet decomposition is that these projections can be obtained by recursively applying $h_0[n]$ and $h_1[n]$ to the original sequence $f[n]$, as shown in Fig. 7.4(a) (see also Section 4.3.7). This type of signal representation is known as *decimated*, or *critically subsampled*, wavelet decomposition. Let $f_j^{(l)}[n]$ and $f_j^{(h)}[n]$ respectively denote the *approximation* (lowpass) and *detail* (highpass) sequences at the output of the jth stage. An equivalent representation is given in Fig. 7.4 (a) obtained from that of Fig. 7.4 (b) shifting the downsamplers toward the output of the system and by using upsampled filters. Should be noticed that the undecimated wavelet (UDW) sequences, $\tilde{f}_j^{(l)}[n]$ and $\tilde{f}_j^{(h)}[n]$, can be obtained downsampling $f_j^{(l)}[n]$ and $f_j^{(h)}[n]$ by a factor 2^j.

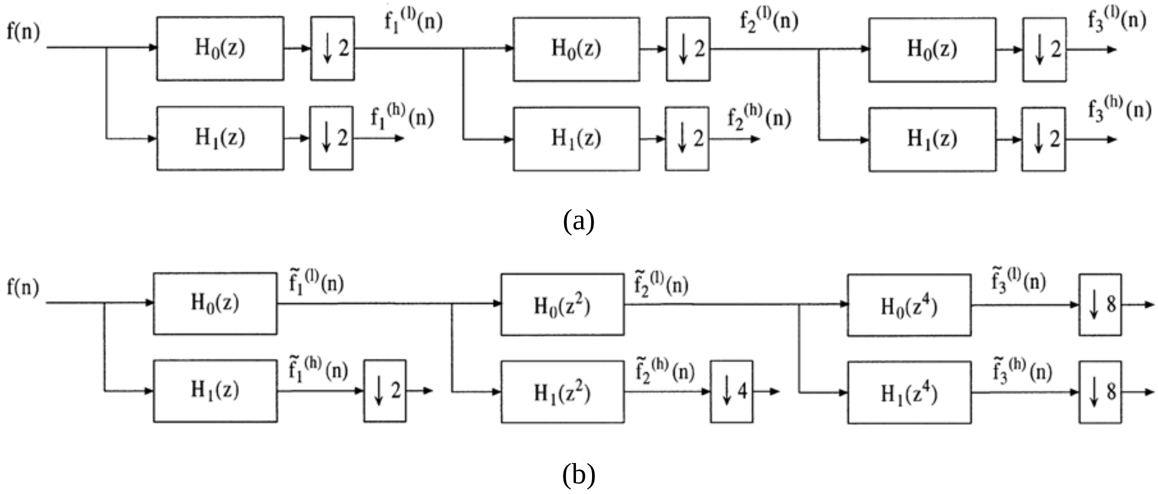


Fig. 7.4 - Three level decomposition. (a) Original schema. (b) Equivalent schema.

From Fig. 7.4 it results clear how the UWD coefficients at level j can be obtained convolving the original sequence $f[n]$ with equivalent filters:

$$\begin{aligned} H_{eq,j}^{(l)}(z) &= \prod_{m=0}^{j-1} H_0\left(z^{2^m}\right) \\ H_{eq,j}^{(h)}(z) &= \left[\prod_{m=0}^{j-2} H_0\left(z^{2^m}\right)\right] H_1\left(z^{2^{j-1}}\right) = H_{eq,j}^{(l)}(z) H_1\left(z^{2^{j-1}}\right) \end{aligned} \tag{7.40}$$

Getting back to the SAR image model of Eq. (7.35), considering the notation $f(x)$ at the place of $f[n]$ for discrete signals, the UWD coefficients of the reflectivity σ are given by:

$$\begin{aligned} \tilde{\sigma}_j^{(l)}(x) &= \sigma * h_{eq,j}^{(l)} = \sum_i h_{eq,j}^{(l)}(i)\,\sigma(x-i) \\ \tilde{\sigma}_j^{(h)}(x) &= \sigma * h_{eq,j}^{(h)} = \sum_i h_{eq,j}^{(h)}(i)\,\sigma(x-i) \end{aligned} \tag{7.41}$$

Analogously, the expression of $\tilde{I}_j^{(l)}$, $\tilde{I}_j^{(h)}$, $\tilde{v}_j^{(l)}$, and $\tilde{v}_j^{(h)}$ represent the projection of I and v on the UWD domain, where:

$$\begin{aligned} \tilde{I}_j^{(l)}(x) &= \tilde{\sigma}_j^{(l)}(x) + \tilde{v}_j^{(l)}(x) \\ \tilde{I}_j^{(h)}(x) &= \tilde{\sigma}_j^{(h)}(x) + \tilde{v}_j^{(h)}(x) \end{aligned} \tag{7.42}$$

In this domain, exploiting the properties $\mathrm{E}[v] = 0$ and $\sum_i h_{eq,j}^{(h)}(i) = 0$, we have:

$$\begin{cases} \mathrm{E}\left[\tilde{v}_j^{(l)}\right] = 0 \\ \mathrm{E}\left[\tilde{v}_j^{(h)}\right] = 0 \\ \mathrm{E}\left[\tilde{I}_j^{(l)}\right] = \mathrm{E}\left[\tilde{\sigma}_j^{(l)}\right] \neq 0 \\ \mathrm{E}\left[\tilde{I}_j^{(h)}\right] = \mathrm{E}\left[\tilde{\sigma}_j^{(h)}\right] \approx 0 \end{cases} \tag{7.43}$$

where $\mathrm{E}\left[\tilde{\sigma}_j^{(h)}\right]$ would be equally to zero if σ was a stationary random process. Now, leaving out the exponent (h) of the high-pass coefficients (for clarity), and presupposing the UWD coefficients $\tilde{I}_j$ as a random process corrupted by an additive zero-mean noise $\tilde{v}_j$, we can exploit the same theory in Eq. (7.39) which can be further simplified (exploiting $\mathrm{E}\left[\tilde{\sigma}_j^{(h)}\right] = 0$) in:

$$\hat{\tilde{\sigma}}_{LMMSE,j}(x) = \frac{\mathrm{E}\left[\tilde{I}_j^2(x)\right] - \mathrm{E}\left[v_j^2(x)\right]}{\mathrm{E}\left[\tilde{I}_j^2(x)\right]} \tilde{I}_j(x) \tag{7.44}$$

Practically speaking, Eq. (7.44) is the application of the Kuan filter to the undecimated wavelet coefficients. Equivalently to the Classic despeckling filters, if we assume that natural images are locally stationary and ergodic, we may substitute statistical expectations in (7.44) with spatial averages. However, while $\mathrm{E}\left[\tilde{I}_j^2(x)\right]$ can be directly computed, the term $\mathrm{E}\left[v_j^2(x)\right]$ can be derived as:

$$\mathrm{E}\left[v_j^2(x)\right] = \frac{\sigma_n^2}{1+\sigma_n^2} \sum_i h_{eq,j}(i)^2 \, \mathrm{E}[I^2(x-i)] \tag{7.45}$$

where the assumption of v incorrelated has been made. As previously said, even though all the previous computations are on 1-D signals, 2-D extensions is straightforward considering separable filters between row and column of the image.

The same authors, in [71] propose an improvement of this algorithm modelling the wavelet coefficients both of noiseless reflectivity $\tilde{\sigma}_j$ and of the noise $\tilde{v}_j$ as a zero-mean Generalized Gaussian (GG) which depends only on two parameters that are locally estimated. Even though GG is symmetric and the same authors admit that experiments on natural images have shown asymmetric histogram of wavelet coefficients, they use GG for its relative simplicity and because GG includes Normal distribution as special case. In fact, leaving out the first level of decomposition, the validity of the central limit theorem is expected for deeper levels. Anyway, a lot of complexity is introduced and the gain reported in the article is less of 2dB for 1-look images, and it decreases when the number of looks rise. For this reason, only the original LMMSE is used for comparison taking in mind that an improvement can be obtained.

7.4 Partial Differential Equation (PDE) Based Filter

A very powerful class of despeckling filters are those derived as a solution of a physical process represented by a partial differential equation (PDE). In this framework, filtering an image with a Gaussian kernel with standard deviation σ is equivalent to see the evolution of the system at the time $t = \sigma^2/2$, where each grey level in the original image represents a temperature measure [47], [72]. Formally, the image intensity I (grey level, color, etc.) is seen as a physical variable such as temperature. The existence of the concentration gradient ∇I creates a flux J (a vector) according to the *first Fick law* (diffusion law), in order to equilibrate the concentration differences in the diffusing medium Ω:

$$J = D\nabla I \tag{7.46}$$

where D is the diffusion tensor (a matrix) which characterizes the diffusion medium and which determines the type of diffusion process we apply on image. Since the mass conservation hypothesis has to be respected, the temporal variation of I (e.g. temperature) inside Ω is equal to the flux of I across the boundary of Ω (indicated with $d\Omega$ and $\hat{\boldsymbol{n}}$ is the unit vector normal to $d\Omega$):

$$\iint_{\Omega} \frac{dI}{dt} d\boldsymbol{x} = \int_{d\Omega} J \cdot \hat{\boldsymbol{n}}\, ds \tag{7.47}$$

Applying the Divergence Theorem:

$$\int_{d\Omega} J \cdot \hat{\boldsymbol{n}}\, ds = \iint_{\Omega} \operatorname{div}(J) d\boldsymbol{x} \tag{7.48}$$

and substituting all in Eq. (7.47):

$$\begin{gathered}\iint_{\Omega} \left(\frac{dI}{dt} - \operatorname{div}(J)\right) d\boldsymbol{x} = 0 \\ \rightarrow \iint_{\Omega} \left(\frac{dI}{dt} - \operatorname{div}(D\nabla I)\right) d\boldsymbol{x} = 0\end{gathered} \tag{7.49}$$

the *second Fick law* (diffusion law) is derived:

$$\frac{dI}{dt} = \operatorname{div}(D\nabla I) \tag{7.50}$$

so that the final PDE is:

$$\begin{cases}\dfrac{dI}{dt} = \operatorname{div}(D\nabla I) & , t > 0 \\ I = I_0 & , t = 0\end{cases} \tag{7.51}$$

with I_0 the initial image. Then in order to avoid any flux variation throughout the image margins it is usually imposed the following constraint:

$$\left.\frac{dI}{d\hat{\boldsymbol{n}}}\right|_{\partial\Omega} = 0 \tag{7.52}$$

with $\hat{\boldsymbol{n}}$ the usual unit vector normal to the image borders.

As an example, if D was scalar, the flux J would be parallel to ∇I, which means that the diffusion would be isotropic, and Eq. (7.51) would become:

$$\frac{dI}{dt} = D\, \nabla^2 I \tag{7.53}$$

where $\nabla^2 I$ is the laplacian of I. In this case the PDE to solve would be:

$$\begin{cases}\dfrac{dI}{dt} = D\, \nabla^2 I & , t > 0 \\ I = I_0 & , t = 0\end{cases} \tag{7.54}$$

which can be solved in a closed form and the solution at time t can be obtained as the initial image smoothed by a Gaussian kernel G_σ whose standard deviation is $\sigma = \sqrt{2t}$, i.e.:

$$\begin{cases}I = G_{\sqrt{2t}} * I_0 & , t > 0 \\ I = I_0 & , t = 0\end{cases} \tag{7.55}$$

Clearly, this type of evolution yields an image where all sharp transitions are smoothed out due to the Gaussian filtering.

7.4.1 Perona-Malik Anisotropic Diffusion

Without any modification to the PDE in Eq. (7.54), the natural evolution of the system leads to a completely blurred image (the solution is the initial image smoothed by a Gaussian kernel). For this reason, Perona and

Malik [72] proposed to change the scalar D with a function $g(\nabla I)$ reverse proportional to the image gradient. In particular, they suggested using one of the following functions:

$$g_1(\nabla I) = e^{-\frac{|\nabla I|^2}{k}};\ g_2(\nabla I) = \frac{1}{1 + \left(\frac{|\nabla I|}{k}\right)^2} \tag{7.56}$$

which belong to the interval $[0, 1]$, $g(\nabla I) \to 0$, with $|\nabla I| \to \infty$, and the parameter k acts as a threshold. In fact, can be easily proved that an edge with $|\nabla I| < k$ is smoothed whereas an edge with $|\nabla I| > k$ becomes sharper. We will see that this edge-enhancing property of the Perona-Malik filter is the fundamental key for the very impressive results of this filter. Furthermore, since this filter constitutes the starting point of all the subsequent PDE based filters, we report the proof of this property here after.

Suppose to have a 1-D smooth edge (here modeled as a step edge convolved with a Gaussian kernel) which varies only along the x direction, as showed in Fig. 7.5. In this case, indicating derivatives both explicitly and with a subscript, we have:

$$\text{div}(g(\nabla I)\nabla I) = \frac{\partial}{\partial x}[g(I_x)I_x] = \frac{\partial}{\partial x}[J(I_x)] \tag{7.57}$$

Then the 1-D version of the diffusion equation becomes:

$$I_t = \frac{\partial}{\partial x}[J(I_x)] = J'(I_x)I_{xx} \tag{7.58}$$

If we look at the variation in time of the slope of the edge, i.e. $\frac{\partial}{\partial t}[I_x]$, supposing I derivable, the order of differentiation can be inverted, so that:

$$\frac{\partial}{\partial t}[I_x] = \frac{\partial}{\partial x}[I_t] = \frac{\partial}{\partial x}[J'(I_x)I_{xx}] = J''(I_x)I_{xx}^2 + J'(I_x)I_{xxx} \tag{7.59}$$

As we can see in Fig. 7.5, since at edge point (in that case at $x = 64$), $I_{xx} = 0$ and $I_{xxx} \ll 0$, the term $J''(I_x)I_{xx}^2$ in Eq. (7.59) can be left out and, if $J'(I_x) > 0$ the slope decreases (edge smoothing case), otherwise, if $J'(I_x) < 0$ the edge is sharpened (edge enhancing case). Now, given one of the functions in Eq. (7.56), $J(I_x)$ is shown in Fig. 7.6 (a),(b). It becomes immediately clear that, when $|\nabla I| < k$ we have $J'(I_x) < 0$ and when $|\nabla I| > k$ we have $J'(I_x) > 0$.

Anyway, in the original formulation of the function $g_1(\nabla I)$ (which gives the best result along edges) there is a bias in the threshold parameter k that does not represent the actual point where the flux derivative change sign, see Fig. 7.6 (b). To solve this problem a unbiased version of this function has been proposed in [73], see Fig. 7.6 (c):

$$g_1(\nabla I) = 1 - e^{-\frac{C_m}{\left(\frac{|\nabla I|}{k}\right)^m}} \tag{7.60}$$

where the exponent power is a general integer number m and the parameter C_m can be determined as zero of the following equation:

$$\left.\frac{\partial}{\partial x}[xg_1(x)]\right|_{k=|\nabla I|} = 0 \tag{7.61}$$

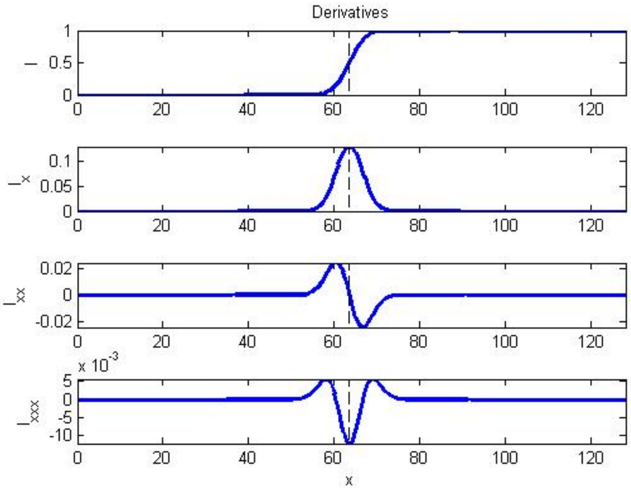
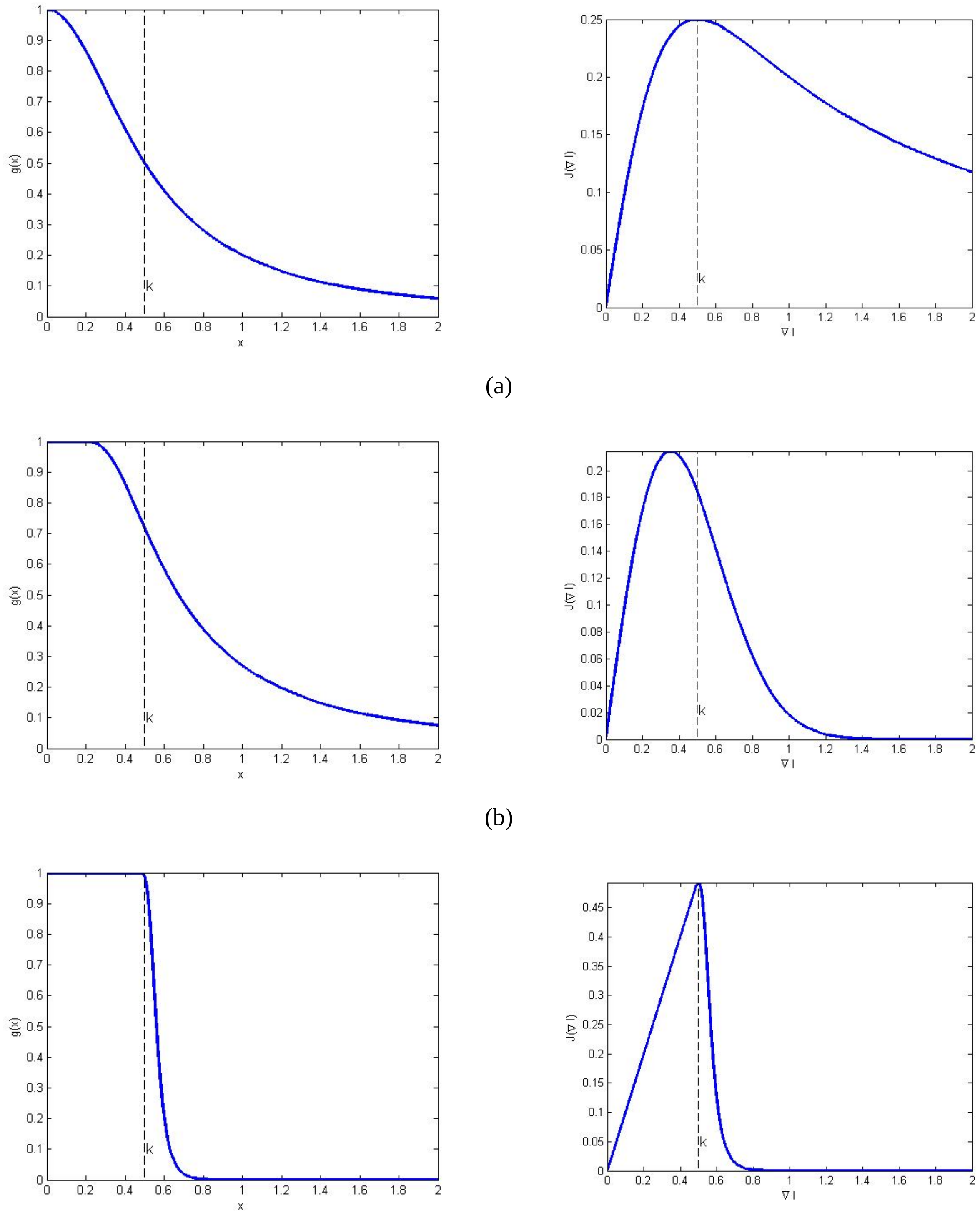


Fig. 7.5 - Smooth edge and its derivatives.

From the article of Perona-Malik [72], a lot of evolutions have been proposed in literature [73]-[81], anyway only the more relevant at our scopes will be treated. For a careful overview the reader is addressed to [73]. Cattè-Lion-Morel [77] assert that the Perona-Malik proposal of smoothing the initial noisy image with a Gaussian kernel (in order to make it differentiable), is a trick. Furthermore, they maintain that the original equation behaves as a forward-backward equation that is known to be bad-posed (a slight modification of the initial image yields a very different equation solution) .Therefore, they proposed the following corrected formula:

$$\frac{dI}{dt} = \mathrm{div}(g(\nabla I_\sigma)\nabla I) \tag{7.62}$$

where I_σ is the Gaussian smoothed image. In general, they proved that any low pass filtering applied only inside the diffusivity function g is enough. In [80] the Gaussian smoothing is substituted with a symmetric exponential filter in order to overcome the well-known problem of edge position error introduced by Gaussian smoothing (see Section 4.3). Anyway, regularizing only the gradient inside the diffusivity function yields results where edges are completely surrounded by noise. To avoid this drawback, as Perona-Malik have suggested, it suffices to filter the initial image with a Gaussian kernel (linear diffusion) before applying the algorithm (non-linear diffusion). It could seem that after the first linear diffusion, the algorithm cannot appropriately recover blurred edges. Yet, thanks to the edge-enhancing property of the Perona-Malik schema, this is not true and strong edges are correctly sharpened.



(a)

(b)

(c)

Fig. 7.6 - (a) Flux $J(I_x)$ (shown on the right) when function $g_2(\nabla I)$ (shown on the left) is used. (b) Flux $J(I_x)$ (shown on the right) when function $g_1(\nabla I)$ (shown on the left) is used. (c) Unbiased version of the flux (shown on the right) and diffusivity (shown on the left).

7.4.2 Biased Anisotropic Diffusion

Successively, another improvement has been proposed by Nordstrom [79]. He found a very close relation between the minimization of an energy functional based on the image Total Variation (TV) and Anisotropic Diffusion (AD) equation. From this sight, he proposed a “biased” AD (BAD) adding a term in the original equation in order to reach a steady state and avoid setting the final time parameter of the solution:

$$\frac{dI}{dt} = \operatorname{div}(g(\nabla I_\sigma)\nabla I) - (I_0 - I) \tag{7.63}$$

Moreover, he provided a physical interpretation of this equation as the temperature evolution of a slab leant on the initial image (always interpreted as a heat source).

7.4.3 Speckle Reducing Anisotropic Diffusion

Even though the PDE framework does not presuppose any model of the noisy image, authors in [75] have tried to account the multiplicative noise case, implementing a PDE of the type:

$$\begin{cases} \dfrac{dI}{dt} = \operatorname{div}(g(C_I^2)\nabla I) & ,t > 0 \\ I = I_0 & ,t = 0 \end{cases} \tag{7.64}$$

where the diffusivity function now depends on the coefficient of variation C_I^2 in place of the gradient ∇I. In particular, the diffusivity function may depend on the same factor k_L (or k_k) of Lee (or Kuan) filter in Eq. (7.4). However, in [75] the author suggest using:

$$g(C_I^2) = \frac{1}{1 + \left(\dfrac{C_I^2 - C_n^2}{C_n^2 + C_n^4}\right)^2} \tag{7.65}$$

with C_n^2computed at each iteration on a homogeneous region of the image passed as input. It should be noted that the coefficient k is different from both k_L and k_k, and it has been chosen to have a better edge-enhancement diffusion [75]. Moreover, some improvements on final performance can be obtained estimating differently and in a larger local neighborhood the coefficient of variation [83].

7.4.4 Coherence Enhancing Anisotropic Diffusion

Even though these filters are often referred to as anisotropic diffusion (AD) techniques, the first real anisotropic diffusion approach was proposed in [78], where the Coherence Enhancing Diffusion (CED) was devised. In fact, we recall that the term anisotropic is inappropriate because the only true anisotropic diffusion is that of CED filter, whereas the other ones are actually non-linear, isotropic, and inhomogeneous [73]. Briefly, if the diffusion tensor D is constant over the whole image domain, one speaks of homogeneous diffusion, and a space-dependent filtering is called inhomogeneous. Often the diffusion tensor is a function of the differential structure of the evolving image itself. Such a feedback leads to nonlinear diffusion filters. Diffusion which does not depend on the evolving image is called linear. Finally, if the flux $\boldsymbol{J}$ is parallel to the image gradient the diffusion is isotropic, otherwise is anisotropic.

Practically speaking, the author in [78] suggest of using the so called “structure tensor” at the place of the gradient to estimate the image variation directions. In fact, while smoothing the gradient components directly in order to reduce the noise gives rise to possible cancellation effects (i.e. near gradients with opposite direction are smoothed out), smoothing the structure tensor components does not produce this drawback. Structure tensor is also called “second order matrix” or “interest operator” [82] and is defined as:

$$S(\nabla I_\sigma) = \nabla I_\sigma \, \nabla I_\sigma^T \tag{7.66}$$

Once $S(\nabla I_\sigma)$ is smoothed element-wise by G_ρ to reduce the noise, the resultant matrix S_ρ (symmetric and positive semi-definite) can be decomposed to find the image principal components:

$$S_\rho = G_\rho * S(\nabla I_\sigma) = V\boldsymbol{\Lambda}V^T = [\boldsymbol{v_1} \quad \boldsymbol{v_2}]\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\begin{bmatrix}\boldsymbol{v_1^T}\\ \boldsymbol{v_2^T}\end{bmatrix} \tag{7.67}$$

where $\boldsymbol{v_1}$, $\boldsymbol{v_2}$ indicate respectively the eigenvectors parallel and orthogonal to the image gradient, with λ_1, λ_2 the corresponding eigenvalues. Considering $\lambda_2 \le \lambda_1$, the eigenvector $\boldsymbol{v_1}$ corresponds to the gradient direction and its eigenvalue λ_1 is high on edges. Moreover, the eigenvector $\boldsymbol{v_2}$ is orthogonal to the gradient direction and its eigenvalue λ_2 is high only on object corners and singularities. In [78] to prefer smoothing along the direction $\boldsymbol{v_2}$ and consequently link interrupted lines, the diffusion matrix D is derived from S_ρ as:

$$D = [\boldsymbol{v_1} \quad \boldsymbol{v_2}]\begin{bmatrix}g_{\lambda_1}(\lambda_1,\lambda_2) & 0\\ 0 & g_{\lambda_2}(\lambda_1,\lambda_2)\end{bmatrix}\begin{bmatrix}\boldsymbol{v_1^T}\\ \boldsymbol{v_2^T}\end{bmatrix} \tag{7.68}$$

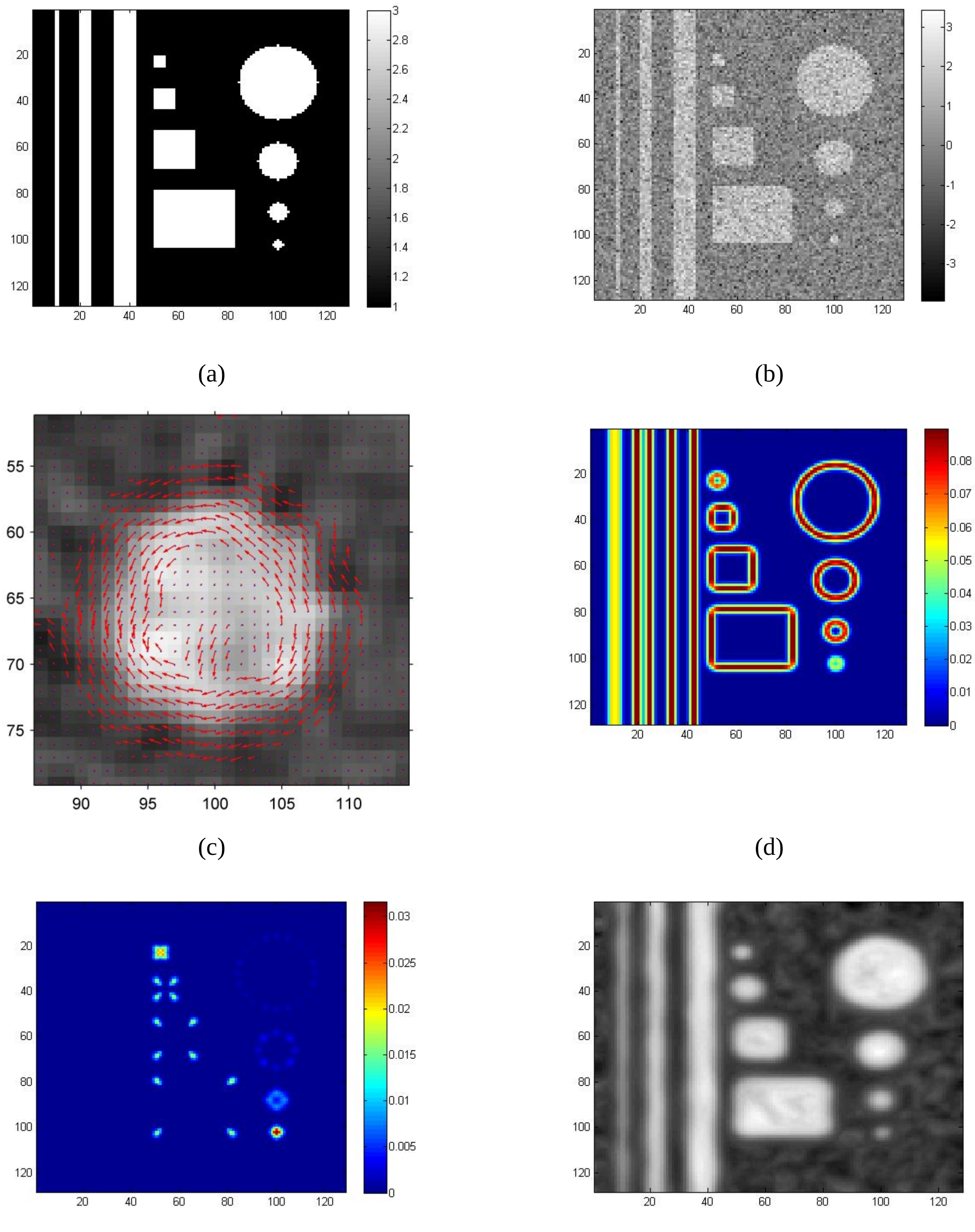
with:

$$\begin{cases} g_{\lambda_1}(\lambda_1,\lambda_2) = \alpha \\ g_{\lambda_2}(\lambda_1,\lambda_2) = \alpha + (1-\alpha)e^{\frac{-C_m}{(k/(\lambda_1-\lambda_2)^2)^m}} \end{cases} \tag{7.69}$$

where α permits a small diffusivity (usually $\alpha = 0.05$) even when no preferential direction exists, k acts as a threshold to the quantity $(\lambda_1 - \lambda_2)^2$ and c, C_m and m have the same role as in corrected Perona-Malik formula.

7.4.5 Improved Edge Enhancing Diffusion (IEED)

Since the aim of [78] was not to devise a denoise filter, applying CED on a speckle-corrupted image yields results with distorted details and a lot of noise left on homogeneous regions. In order to explain this behaviour let's recall how CED works. First, the structure tensor matrix for each point is computed. This matrix contains the same information that can be extracted from the gradient, although it groups this information in its eigenvectors and eigenvalues. The eigenvectors, here indicated as $\boldsymbol{v_1}, \boldsymbol{v_2}$, which are respectively parallel and orthogonal to the gradient ∇I, have eigenvalues λ_1, λ_2 whose value indicates the image variation along the respective direction. For this reason Weickert puts the diffusion matrix D of the classical PDE $I_t = \mathrm{div}(D\nabla I)$ equal to the structure tensor matrix but modifying each eigenvalue with a function $g_{\lambda_1}(\lambda_1,\lambda_2)$ and $g_{\lambda_2}(\lambda_1,\lambda_2)$, in order to adjust the smoothing direction. To see the behavior of CED diffusion matrix D, in Fig. 7.7(c) it is shown in blue the eigenvector $\boldsymbol{v_1}$ (parallel to the gradient) with a length proportional to its diffusivity coefficient $g_{\lambda_1}(\lambda_1,\lambda_2)$ and in red the eigenvector $\boldsymbol{v_2}$ (orthogonal to the gradient) with a length proportional to its diffusivity coefficient $g_{\lambda_2}(\lambda_1,\lambda_2)$. The reported values correspond to the first iteration, when CED is applied at the noisy image in Fig. 7.7 (b).



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7.7 - (a) Cartoon Model of the Simulated image “SIM”. (b) Image corrupted by an ideal Gamma noise with independent samples and number of look $L = 5$. (c) CED diffusivity matrix representation. In blue the eigenvector v_1 (parallel to the gradient) with a length proportional to its diffusivity coefficient $g_{\lambda_1}(\lambda_1, \lambda_2)$ and in red the eigenvector v_2 (orthogonal to the gradient) with a length proportional to its diffusivity coefficient $g_{\lambda_2}(\lambda_1, \lambda_2)$, at first iteration, when CED is applied on a magnification of the noisy image “SIM”. (d) Eigenvalue λ_1 on noiseless image “SIM”. (e) Eigenvalue λ_2 on noiseless image “SIM”. (f) Result of CED filter $\sigma = 1$; $\rho = 1$; $m = 2$; $\alpha = 0.05$; $Q = 0.9$.

As can be seen from Fig. 7.7 (c), only smoothing along edges is performed whereas on homogeneous region both modified eigenvalues μ_1, μ_2 become small (equal to parameter α, usually set to a small value near 0.05), where:

$$\begin{aligned} \mu_1 &= g_{\lambda_1}(\lambda_1, \lambda_2) \\ \mu_2 &= g_{\lambda_2}(\lambda_1, \lambda_2) \end{aligned} \tag{7.70}$$

To understand the role of each eigenvalue, in Fig. 7.7 (d)-(e) the original eigenvalues λ_1, λ_2 are shown when the structure tensor matrix is computed on the noiseless image. As can be seen λ_1 is high on edges whereas λ_2 is high only near corners. In Fig. 7.7 (f) the final smoothing result of CED is reported. It should be pointed out that the parameter Q reported in the caption is used to compute the threshold k. In particular, Q is the quantile of the experimental cumulative density function (ecdf) of the respective variables to which the threshold k has to be applied (see Appendix M for further details). That is, in CED Q is the ecdf quantile of the variable $(\lambda_1 - \lambda_2)^2$ computed on a homogeneous region passed as input.

From Fig. 7.7 (f) all previous discussion becomes clear and, moreover, we can see that CED has another lack, it does not preserve corners. In order to overcome CED drawbacks, a modification on diffusivity function can be done:

$$\begin{cases} g_{\lambda_1}(\lambda_1, \lambda_2) = 1 - e^{\frac{-C_{m_1}}{(k_1/\lambda_1)^{m_1}}} \\ g_{\lambda_2}(\lambda_1, \lambda_2) = 1 - e^{\frac{-C_{m_2}}{(k_2/\lambda_2)^{m_2}}} \end{cases} \tag{7.71}$$

The novelty of this filter not only is it in the use of an unbiased diffusivity formula in CED formulation but, above all, it is in the way eigenvalues information is used to have isotropic diffusion on homogeneous region, anisotropic diffusion along edges and corner preservation jointly.

The choice in Eq. (7.71) can be better appreciated exploiting a convenient decomposition of the flux $\boldsymbol{J}$, and an alternative decomposition of the diffusion matrix D. In fact, D can be expressed as:

$$D = [\boldsymbol{v_1} \quad \boldsymbol{v_2}] \begin{bmatrix} g_{\lambda_1}(\lambda_1, \lambda_2) & 0 \\ 0 & g_{\lambda_2}(\lambda_1, \lambda_2) \end{bmatrix} \begin{bmatrix} \boldsymbol{v_1^T} \\ \boldsymbol{v_2^T} \end{bmatrix} = g_{\lambda_1}(\lambda_1, \lambda_2) \boldsymbol{v_1 v_1^T} + g_{\lambda_2}(\lambda_1, \lambda_2) \boldsymbol{v_2 v_2^T} \tag{7.72}$$

Then, indicating with $\nabla I \cdot \boldsymbol{v}$ the scalar product between the gradient and another vector (that is also equal to a directional derivative of I along $\boldsymbol{v}$, indicated as I_v), and recalling that $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ have unitary module (i.e. they holds only directional information), the flux $\boldsymbol{J}$ can be written:

$$\begin{aligned} \boldsymbol{J} &= D \nabla I \\ &= \left[g_{\lambda_1}(\lambda_1, \lambda_2) \boldsymbol{v_1 v_1^T} + g_{\lambda_2}(\lambda_1, \lambda_2) \boldsymbol{v_2 v_2^T} \right] \nabla I \\ &= g_{\lambda_1}(\lambda_1, \lambda_2) \boldsymbol{v_1} [\nabla I \cdot \boldsymbol{v_1}] + g_{\lambda_2}(\lambda_1, \lambda_2) \boldsymbol{v_2} [\nabla I \cdot \boldsymbol{v_2}] \\ &= g_{\lambda_1}(\lambda_1, \lambda_2) I_{v_1} \boldsymbol{v_1} + g_{\lambda_2}(\lambda_1, \lambda_2) I_{v_2} \boldsymbol{v_2} \end{aligned} \tag{7.73}$$

Hence, the following proposition holds:

Prop (7.4.1): *the flux $\boldsymbol{J}$ can be decomposed in two orthogonal components, one directed along $\boldsymbol{v_1}$ and one along $\boldsymbol{v_2}$. Each component has a length proportional to the directional derivative along the respective direction and is multiplied by the corresponding diffusivity function.*

Therefore, using the diffusivities in Eq. (7.71), three different filter behaviors are highlighted by the novel IEED filter:

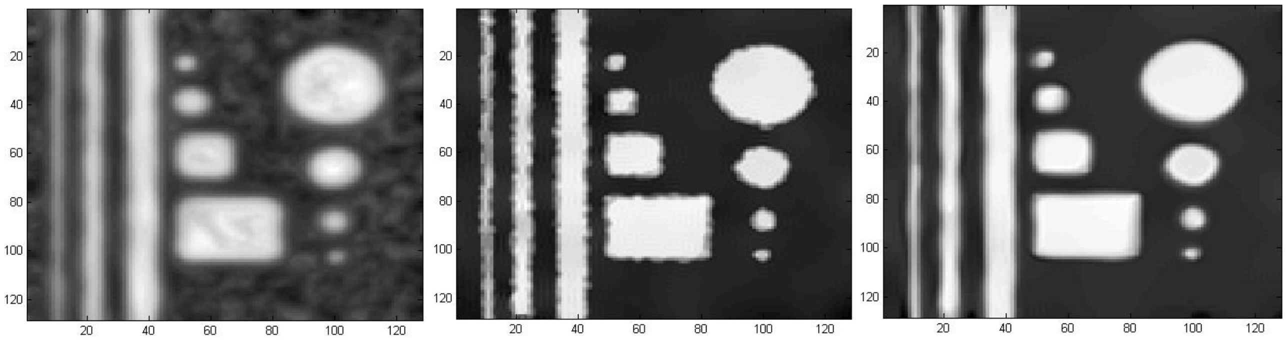
1) *On homogeneous areas* both λ_1 and λ_2 are low. Therefore, μ_1 and μ_2 are near 1, i.e. an isotropic diffusion case (speckle removal).

2) *On edges* λ_1 is high and λ_2 is low. Consequently, μ_1 is near 0 and μ_2 is near 1, i.e. an anisotropic diffusion case with smoothing along $\boldsymbol{v_2}$ (edge preservation).

3) *On corners* both λ_1 and λ_2 are high. As a consequence, μ_1 and μ_2 are near 0, i.e. a no diffusion case (corner preservation).

It should be noted that the last behavior avoids shape distortion of objects and hard targets smoothing. In fact, not only corners and singularities, but also small objects with high RCS are characterized by a high value of both λ_1 and λ_2, thus causing diffusion stop. In Fig. 7.8 we can see the result of IEED compared to IEED and PM (with unbiased diffusivity). Noise removal on edges is evident and also shape preservation is achieved. It should be pointed out that in IEED case Q_1 and Q_2 are the ecdf quantiles of the variables $\lambda_1$and λ_2, respectively computed on a homogeneous region passed as input, whereas for PM filter Q rules $|\nabla I|$. Finally, it is worth noting that the results in Fig. 7.8 are obtained by using the numerical schema optimized for rotational invariance descripted in detail in Appendix M, which enable to yield the best performance.

Summarizing, with the diffusivity in Eq. (7.71) we manage to effectively combine the respective advantages of the PM and CED filter. In the PM schema, the speckle on homogeneous regions is removed but there is no diffusion along edges, so they remain noisy in the final result. Conversely, the CED filter provides little speckle removal on homogeneous areas, but edges are correctly reconstructed.



(a) (b) (c)

Fig. 7.8 - (a) Result of CED filter $\boldsymbol{\sigma = 1;\ \rho = 1; m = 2;\ \alpha = 0.05; Q = 0.9}$. (b) Result of PM filter $\boldsymbol{\sigma = 1;\ m = 2;\ Q = 0.9}$. (c) Result of IEED filter $\boldsymbol{\sigma = 1;\ \rho = 1; m = 2;\ Q_1 = 0.9, Q_2 = 0.999}$.

7.4.6 PDE Filter Summary

In Tab. 7.1, the discussed PDE filter schemas are schematically reported, whit $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ the canonical bases of the space $\mathbb{R}^2$. Discussion about implementation and its connected problems are reported in Appendix M.

	PDE $I_t = \mathrm{div}(D\nabla I)$	**Diffusivity Function**
PM	$I_t = \mathrm{div}\left([\boldsymbol{e_1} \quad \boldsymbol{e_2}]\begin{bmatrix} g(\nabla I) & 0 \\ 0 & g(\nabla I) \end{bmatrix}\begin{bmatrix} \boldsymbol{e_1^T} \\ \boldsymbol{e_2^T} \end{bmatrix}\nabla I\right)$	$g(\nabla I) = 1 - e^{-\frac{C_m}{\left(\frac{\lvert\nabla I\rvert}{k}\right)^m}}$
SRAD	$I_t = \mathrm{div}\left([\boldsymbol{e_1} \quad \boldsymbol{e_2}]\begin{bmatrix} g\left(C_I^2\right) & 0 \\ 0 & g\left(C_I^2\right) \end{bmatrix}\begin{bmatrix} \boldsymbol{e_1^T} \\ \boldsymbol{e_2^T} \end{bmatrix}\nabla I\right)$	$g(C_I^2) = \dfrac{1}{1 + \left(\dfrac{C_I^2 - C_n^2}{C_n^2 + C_n^4}\right)^2}$
CED	$I_t = \mathrm{div}\left([\boldsymbol{v_1} \quad \boldsymbol{v_2}]\begin{bmatrix} g_{\lambda_1}(\lambda_1, \lambda_2) & 0 \\ 0 & g_{\lambda_2}(\lambda_1, \lambda_2) \end{bmatrix}\begin{bmatrix} \boldsymbol{v_1^T} \\ \boldsymbol{v_2^T} \end{bmatrix}\nabla I\right)$	$\begin{cases} g_{\lambda_1}(\lambda_1, \lambda_2) = \alpha \\ g_{\lambda_2}(\lambda_1, \lambda_2) = \alpha + (1-\alpha)e^{\frac{-C_m}{(k/(\lambda_1 - \lambda_2)^2)^m}} \end{cases}$
IEED	$I_t = \mathrm{div}\left([\boldsymbol{v_1} \quad \boldsymbol{v_2}]\begin{bmatrix} g_{\lambda_1}(\lambda_1, \lambda_2) & 0 \\ 0 & g_{\lambda_2}(\lambda_1, \lambda_2) \end{bmatrix}\begin{bmatrix} \boldsymbol{v_1^T} \\ \boldsymbol{v_2^T} \end{bmatrix}\nabla I\right)$	$\begin{cases} g_{\lambda_1}(\lambda_1, \lambda_2) = 1 - e^{\frac{-C_{m_1}}{(k_1/\lambda_1)^{m_1}}} \\ g_{\lambda_2}(\lambda_1, \lambda_2) = 1 - e^{\frac{-C_{m_2}}{(k_2/\lambda_2)^{m_2}}} \end{cases}$

Tab. 7.1 - PDE filter schema summary.

7.5 Results

In this document, both simulated and real SAR images are used in order to evaluate despeckling algorithms. Simulated images belong to a unique category: piecewise constant images. Even natural optic images have been considered. Anyway, corrupting them with a multiplicative Gamma noise with a number of look equal to 1, the performances of all filters are very poor (natural edges have a low contrast ratio). For this reason these results are not reported in this document. Instead, for real SAR images the **CSK images (L1B product)** are used.

For each filter, when no other specification is made, the best parameter set is chosen and their values are reported in Appendix N.

7.5.1 Evaluation Indexes

Evaluation of a reconstructed image $\hat{I}$ is performed by the following measures, where the original noiseless image (when it is known) is indicated as I:

1. Mean Square Error (MSE):

$$MSE = 10\log_{10}\overline{\overline{(\hat{I}-I)^2}} \tag{7.74}$$

2. Signal to Noise Ratio (SNR):

$$SNR = 10\log_{10}\overline{(I-\bar{I})^2}/\overline{\overline{(\hat{I}-I)^2}} \tag{7.75}$$

3. Coefficient of variation of a homogeneous region:

$$C_{\hat{I}}^2 = \frac{\sigma_{\hat{I}}^2}{{\mu_{\hat{I}}}^2} \tag{7.76}$$

4. Ratio between original and reconstructed image with computation of first order statistics of the “ratio noise” $\hat{n}$:

$$\begin{aligned} &\hat{n} = \frac{I}{\hat{I}} = \frac{\sigma n}{\hat{\sigma}} \\ &\mathrm{E}[\hat{n}] \\ &\sigma_{\hat{n}}^2 \end{aligned} \tag{7.77}$$

5. Local contrast around an edge;

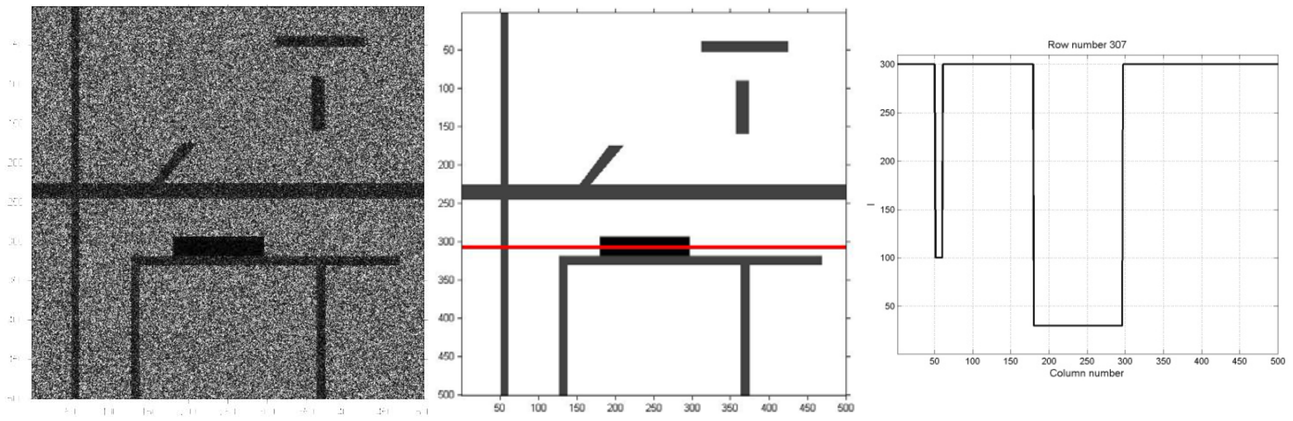
6. Visual inspection;

7. Computational time.

Naturally, to have a perfect reconstruction *MSE* should tend to zero, *SNR* should be maximum, and $C_{\hat{I}}^2$ should tend to zero. Moreover, the ratio noise image should look like a white noise and no structures should be visible. Finally, its first order statistics have to be near to their corresponding parameters used in noise simulation.

7.5.2 Simulated Image (SIM1)

This image is shown in Fig. 7.9(a). It is a piecewise constant image corrupted by an ideal Gamma noise with independent samples and number of look $L = 1$. In Fig. 7.10 the result of the best filter of each class is reported, respectively: Classic Mean filter (CM), PM, iterated Probabilistic Patch Based (PPBi), and Undecimated Wavelet Denoising (UWD), Perona-Malik with optimized schema (PM*) and Improved Edge Enhancing Diffusion with optimized schema (IEED*). It should be pointed out that in Fig. 7.10 we reported both PM* and IEED* to show the very impressive improving on the final results when the optimized numerical schema is used.



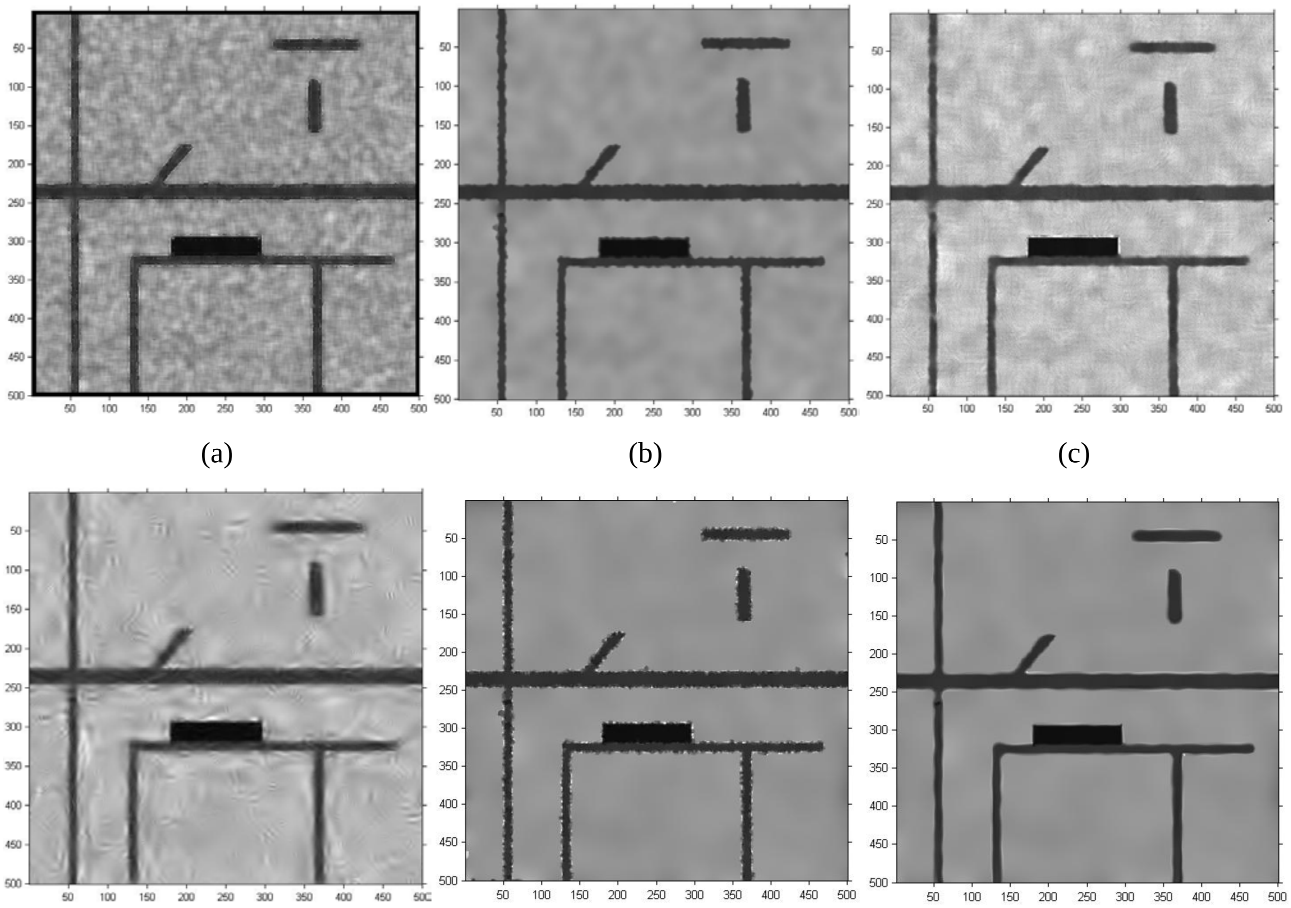
(a) (b) (c)

Fig. 7.9 - (a) Simulated image (SIM1) corrupted by an ideal Gamma noise with independent samples and number of look $L = 1$. (b) Sampled row (row number 307) superimposed to the noiseless image (SIM1) used to display results in 1-D format. (b) Intensity of the sampled row.

	SNR dB/ MSE dB	$C_{\hat{I}}^{2}$	**E**$[\hat{n}]$ / $\sigma_{\hat{n}}^{2}$
Lee	5.22/ 31.31	0.0124	0.97/ 0.82
Kuan	5.55/ 30.98	0.0114	0.98/ 0.83
Frost	3.69/ 32.84	0.0209	0.95/ 0.72
Gamma	4.29/ 32.24	0.0157	1.00/ 0.87
Mean	*6.57/ 29.95*	*0.0079*	*1.00/ 0.99*
PM	10.41/ 26.11	9.5108e-004	***1.01***/ ***1.01***
SRAD	9.58/ 26.95	7.1526e-004	1.02/ 1.03
PM*	9.05/ 27.47	4.7187e-004	1.04/ 1.07
IEED*	***10.62/ 25.90***	***4.0999e-004***	1.04/ 1.07
PPB1	5.96/ 30.56	0.0072	0.96/ 0.81
PPB10	*7.91/ 28.61*	*0.0052*	*0.96/ 0.83*
UWD	0.10/ 36.42	0.0027	1.27/ 1.59

Tab. 7.2 - Evaluation indexes for SIM1 image. Bold font indicates the best parameter value among all filters. Italic, underlined font indicates the best within each class. The asterisk indicates the use of the optimized numerical schema.

As we can see from Tab. 7.2, the best class of despeckling filters is the **Anisotropic Diffusion** (AD) one. Moreover, the optimized numerical schema enables the best performance to be obtained. Within this class, and **Improved Edge Enhancing Diffusion** (IEED*) **filter gives the best results** since it manages to have the best SNR (and of consequence MSE) with a very low residual coefficient of variation inside homogeneous regions. Moreover, the “ratio noise” has first order statistics very near to the simulated multiplicative noise. It should be noticed that this latter result indicates a nearly perfect RCS reconstruction (mean and variance preservation).



(a) (b) (c)

(d) (e) (f)

Fig. 7.10 - Despeckle results of the best filters of each class. (a) Classic Mean filter (CM). (b) Perona-Malik (PM). (c) Iterated Probabilistic Patch Based (PPB10). (d) Undecimated Wavelet Denoising (UWD). (e) Perona-Malik with optimized schema (PM*). (f) Improved Edge Enhancing Diffusion with optimized schema (IEED*).

In Fig. 7.10 the result of the best filter of each class is reported, respectively: Classic Mean filter (CM), PM, iterated Probabilistic Patch Based (PPBi), and Undecimated Wavelet Denoising (UWD).

From Fig. 7.10 we can see that IEED* manages to reconstruct the “cartoon model” nearly ideally, PM* obtains a similar result except along edges that are a little waved, whereas the others filters fail in the reconstruction. **Classic filters** that use the estimated coefficient of variation yields the worse result because it suffices an outlier value to stop the smoothing on a homogeneous region. It is worth noting that this behaviour is very useful on low resolution SAR images because high value pixels which could be targets are preserved. **Patch based filters** behaves better than Classical one, even though they introduce some artefacts along strong edges and at some isolated pixel. Then, as descends from theory**, UWD** reconstruction presents a lot of ringing artefacts.

In Fig. 7.11 we can see the “ratio noise” for these filters. While CM presents some correlation, PM, PPBi and UWD filters show a nearly ideal white noise (except along edges). Nevertheless, the best results is always the one of IEED* where nearly no structures seem present.

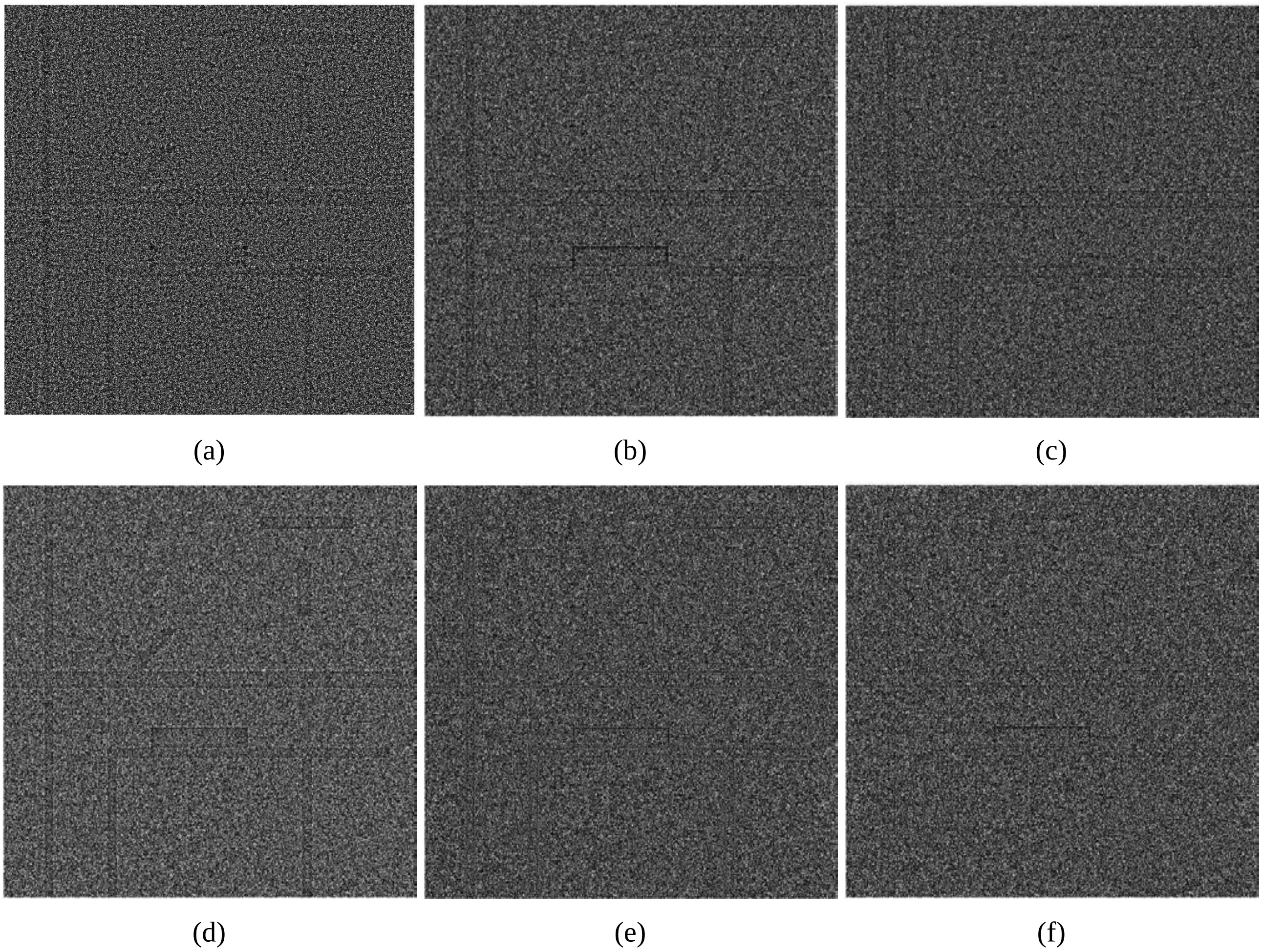
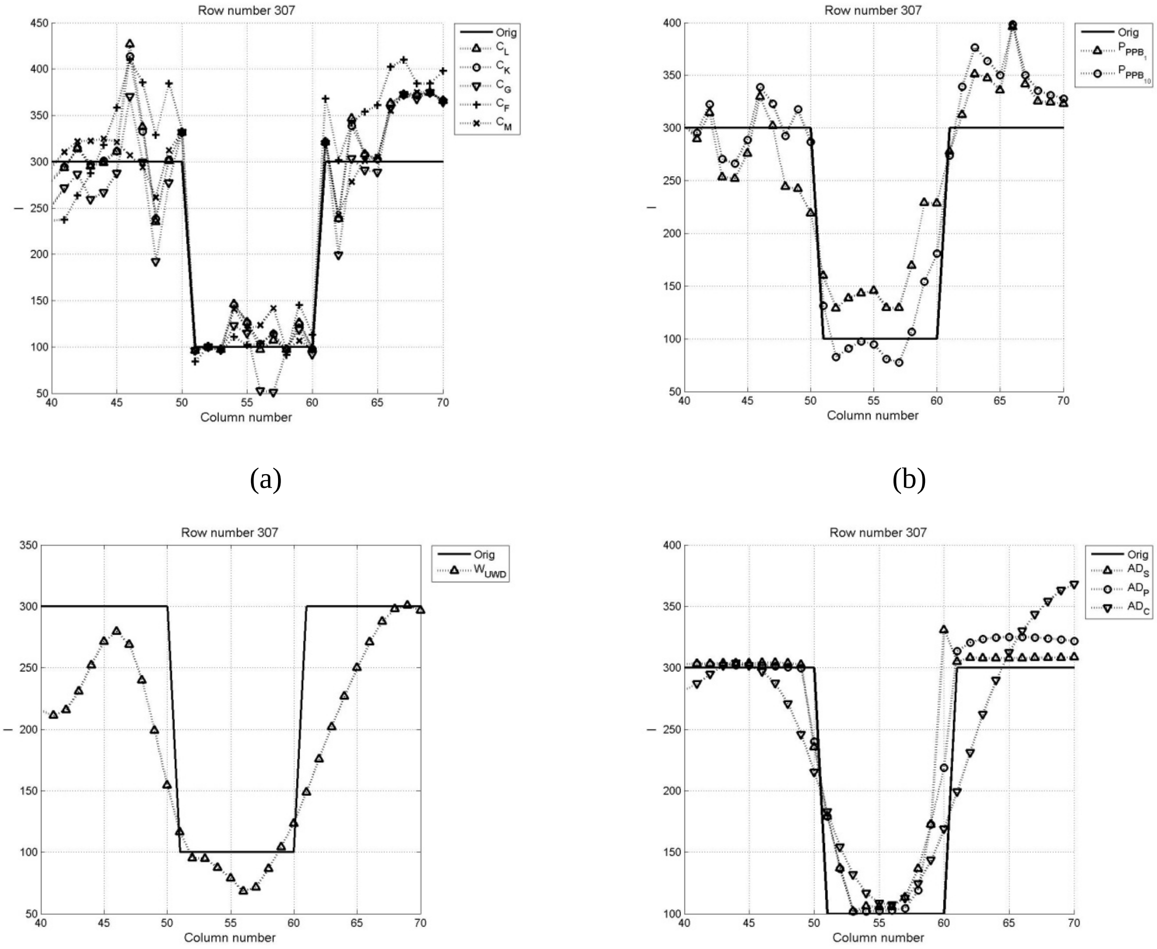


Fig. 7.11 - Ratio noise of the best filters of each class. (a) Classic Mean filter (CM). (b) Perona-Malik (PM). (c) Iterated Probabilistic Patch Based (PPB10). (d) Undecimated Wavelet Denoising (UWD). (e) Perona-Malik with optimized schema (PM*). (f) Improved Edge Enhancing Diffusion with optimized schema (IEED*).

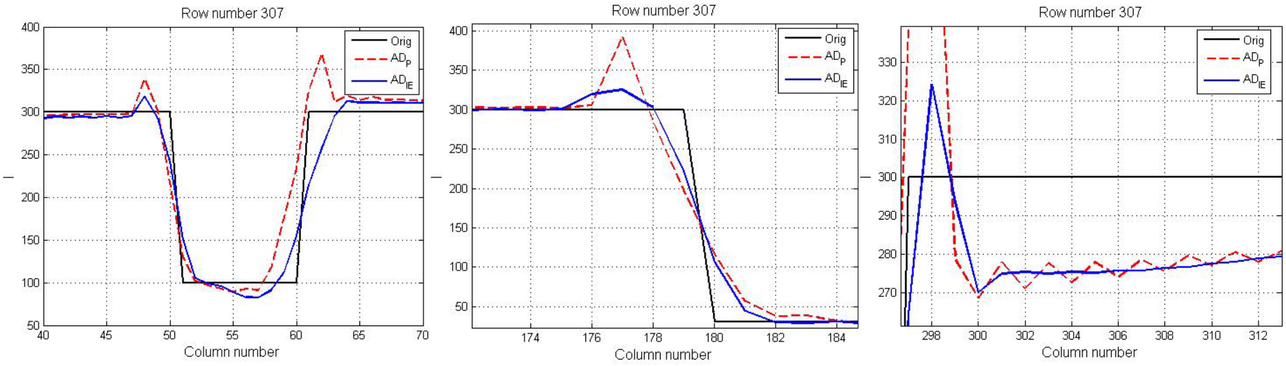
In Fig. 1.13 the sampled row around the first edge is shown for all the filters, except PM* and IEED*, which are reported on Fig. 7.13. As we can see PDE based filters are the only ones which manage to completely remove the noise. It is worth noting that in the legend of the figure the capitalized letter indicates the class (i.e. "C" is classic, "P" patch, "W" wavelet, "AD" anisotropic diffusion), and the subscript indicates the first letters of the filter name (e.g. "C_L" means Classic Lee filter). In order to better visualize the PM* drawback of leaving some noise along the edges, the magnification around some other borders along the sampled row are reported in Fig. 7.13(b),(c).



(a) (b)

(c) (d)

Fig. 7.12 - Despeckling results of all filters on sampled row (row number 307). (a) Classic filters. (b) PDE based filters. (c) Patch Based filters. (d) Undecimated Wavelet Denoising (UWD).

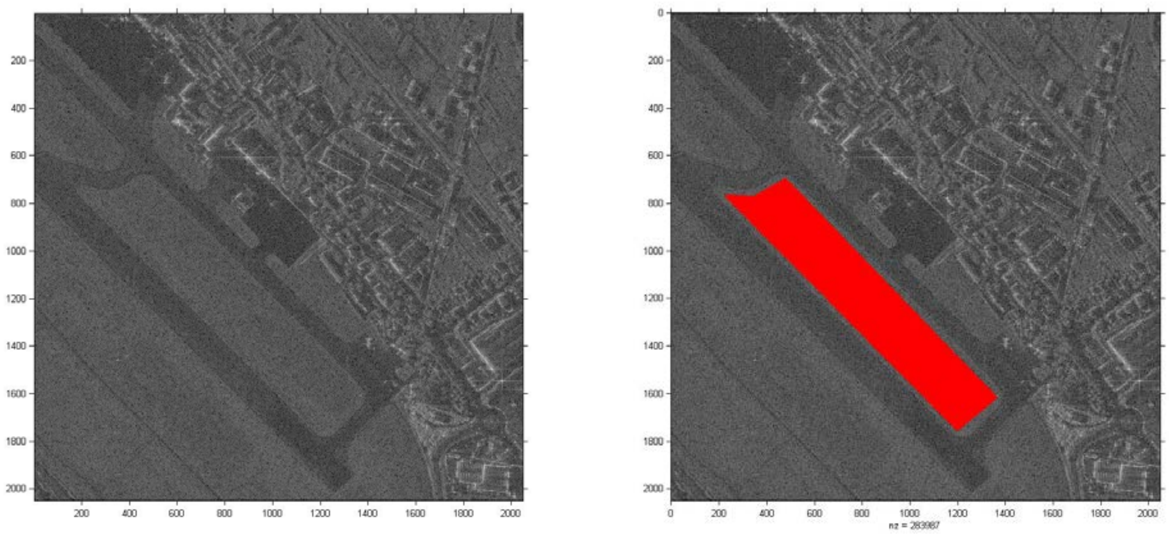


(a) (b) (c)

Fig. 7.13 - Despeckling results on sampled row (row number 307). (a)-(b)-(c) Magnification around an edge.

7.5.3 CSK Image (CSK1)

The image shown in Fig. 7.14 (a) belongs to the “CSK1” image that depicts an airport (see Section 1.9.2). This image has been acquired in Spotlight 2 mode with polarization **HH** and incidence angle near/far of **55.9°/56.3°** and has a theoretical equivalent number of look equal to **1**.



(a)

(b)

Fig. 7.14 - (a) Original CSK image (CSK1). (b) Homogeneous region used to compute evaluation indexes.

	$C_{\hat{I}}^2$	$E[\hat{n}]$ / $\sigma_{\hat{n}}^2$
Original	1.2198	-
Lee	0.1496	0.98/0.79
Kuan	0.1429	0.98/0.80
Frost	0.2245	0.91/0.50
Gamma	0.1461	1.01/0.84
Mean	*0.1318*	*1.01/0.97*
PM	0.0408	*1.05/ 1.01*
SRAD	0.0360	0.89/ 0.55
PM*	0.0220	1.06/ 1.27
IEED*	***0.0211***	1.06/ 1.28
PPB1	*0.1042*	*0.96*/ *0.81*
PPB10	0.2304	0.90/ 0.50
UWD	0.2885	1.06 / 0.54

Tab. 7.3 - Evaluation indexes for CSK1 image. The bold font indicates the best parameter value whereas italic, underlined font indicates the best within each class. The asterisk indicates the use of the optimized numerical schema.

In Tab. 7.3 we can see that, even on real images, PDE based filters manage to reduce speckle noise on homogeneous region more heavily (lower $C_{\hat{I}}^2$), preserving presupposed speckle statistics[15].

The magnifications of the results obtained by the best filters of each class are reported in Fig. 7.15.

For **Classic filters**, as happens on simulated images, the images present noisy reconstruction due to the variance of the final estimation computed only through pixels inside the local window. Before applying these filters, even downsampling the original image by a factor 2 does not produce a quality improvement. In this latter case homogeneous regions are better reconstructed but targets are completely distorted.

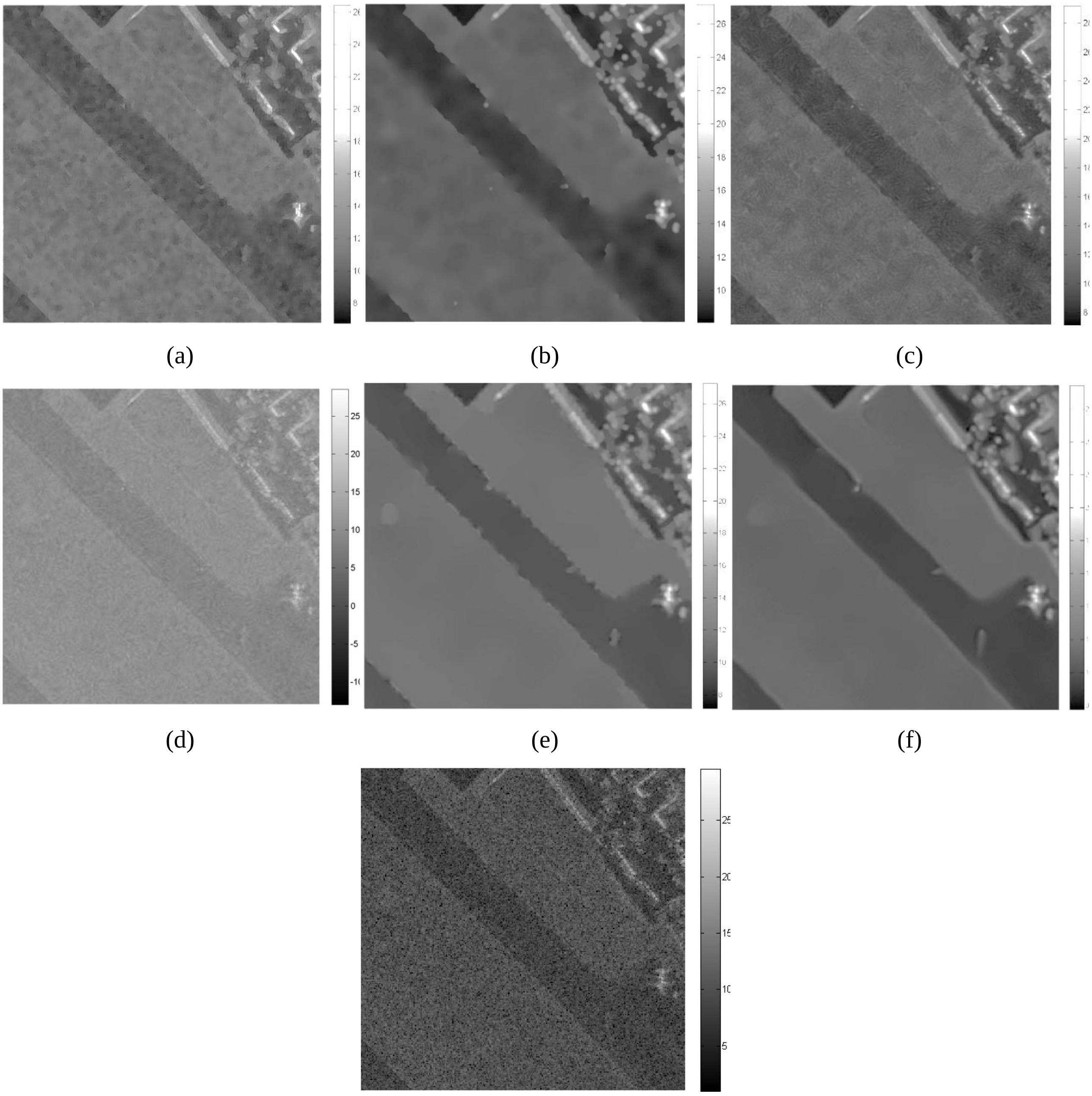
About **Patch based filters**, only one iteration of the PPB filter (PPB1) improves the performance of classic filters, but it does not remove completely the original speckle. However, at each iteration, PPB manage to enhance the previous edges although it introduces very strong, unacceptable, artifacts (small segments composed by low gray values).

As we can see from the colorbar of **UWD filter**, the final dynamic is incremented. This means that low values are lowered and high values are increased. This is not properly the behavior of the filter we look for.

As in simulated images, **PDE based filters** show the best result. From an accurate visual inspection, we can see that SRAD cleans homogeneous region but it smoothes the edges too much. Instead, PM manages to reduce speckle strongly on homogeneous regions and even most weak edges are preserved. Nevertheless, only PM* and IEED* manage to obtain very impressive results.

Nevertheless, in order to acquire more insights about edge preservation and the possibility to detect them after filtering, we have applied at all final results the Canny edge detector. In Fig. 7.16 the gradient magnitude of the Canny filter is reported. As can be seen from these figures the edge detection is greatly improved by PM* and IEED* filters.

[15] On real images the multiplicative speckle noise is only a model. Hence, indexes E[$\hat{n}$] and $\sigma_{\hat{n}}^2$ do not provide the same indication as on simulated images.

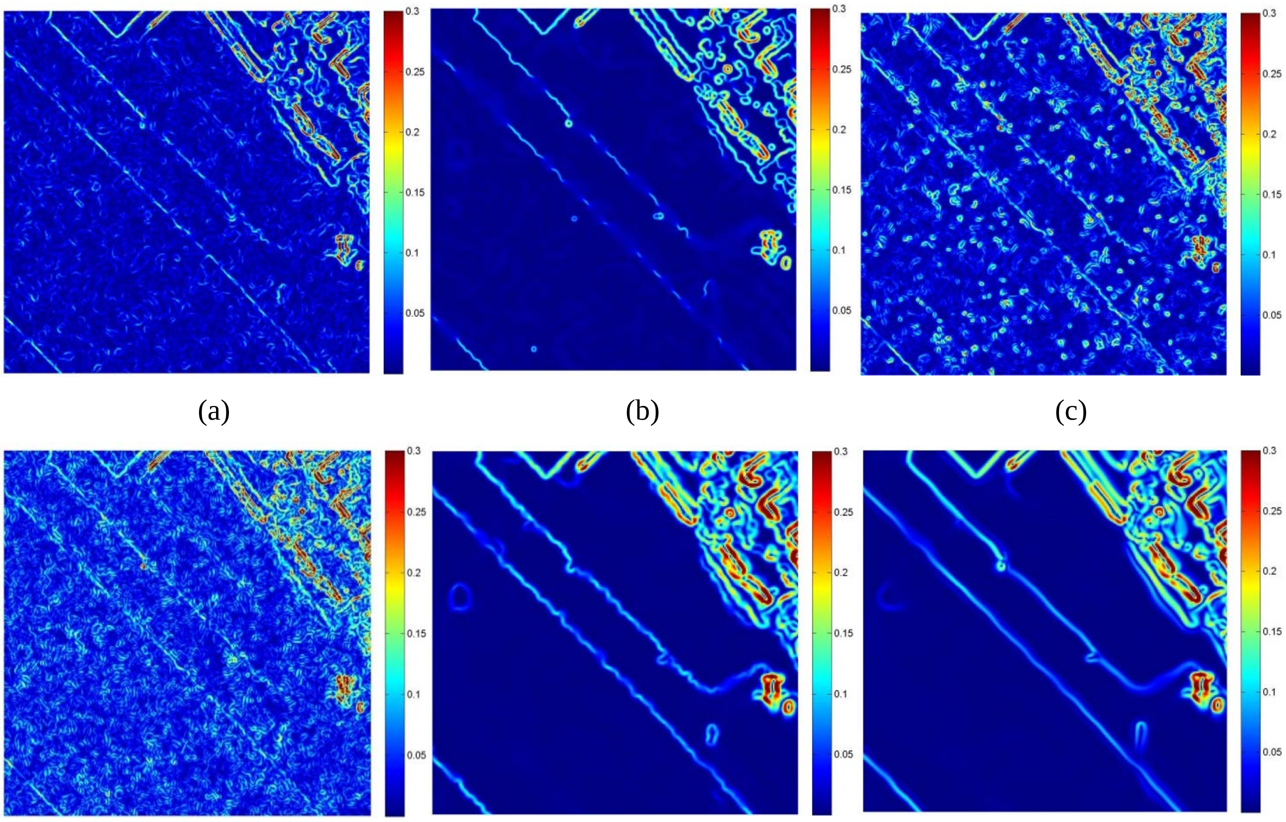


(a) (b) (c)

(d) (e) (f)

(g)

Fig. 7.15 - Despeckle results of the best filters of each class. (a) Classic Mean filter (CM). (b) Perona-Malik (PM). (c) Iterated Probabilistic Patch Based (PPB10). (d) Undecimated Wavelet Denoising (UWD). (e) Perona-Malik with optimized schema (PM*). (f) Improved Edge Enhancing Diffusion with optimized schema (IEED*). (g) Magnification of the original image.



(a) (b) (c)

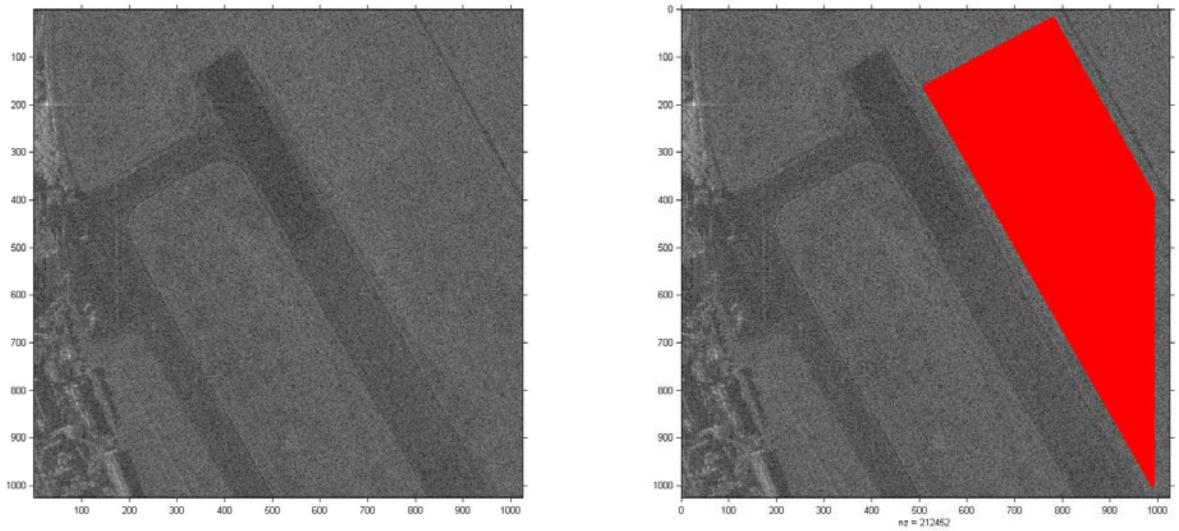
(d) (e) (f)

Fig. 7.16 - Canny gradient magnitude on despeckled results of the best filters of each class. (a) Classic Mean filter (CM). (b) Perona-Malik (PM). (c) Iterated Probabilistic Patch Based (PPB10). (d) Undecimated Wavelet De-noising (UWD). (e) Perona-Malik with optimized schema (PM*). (f) Improved Edge Enhancing Diffusion with optimized schema (IEED*).

7.5.4 CSK Image (CSK2)

The image shown in Fig. 7.17 (a) belongs to the “CSK2” image that depicts an airport (see Section 1.9.2). This image has been acquired in Spotlight 2 mode with polarization **HH** and incidence angle near/far of **51.1°/51.5°** and it has a theoretical equivalent number of look equal to **1**.

The equal behavior of the filters on this image can be noted by both visual inspection in Fig. 7.18 and from the indexes reported in Tab. 7.4. Consequently, even in this case, PM* and IEED* filters ménage to obtain the best results.

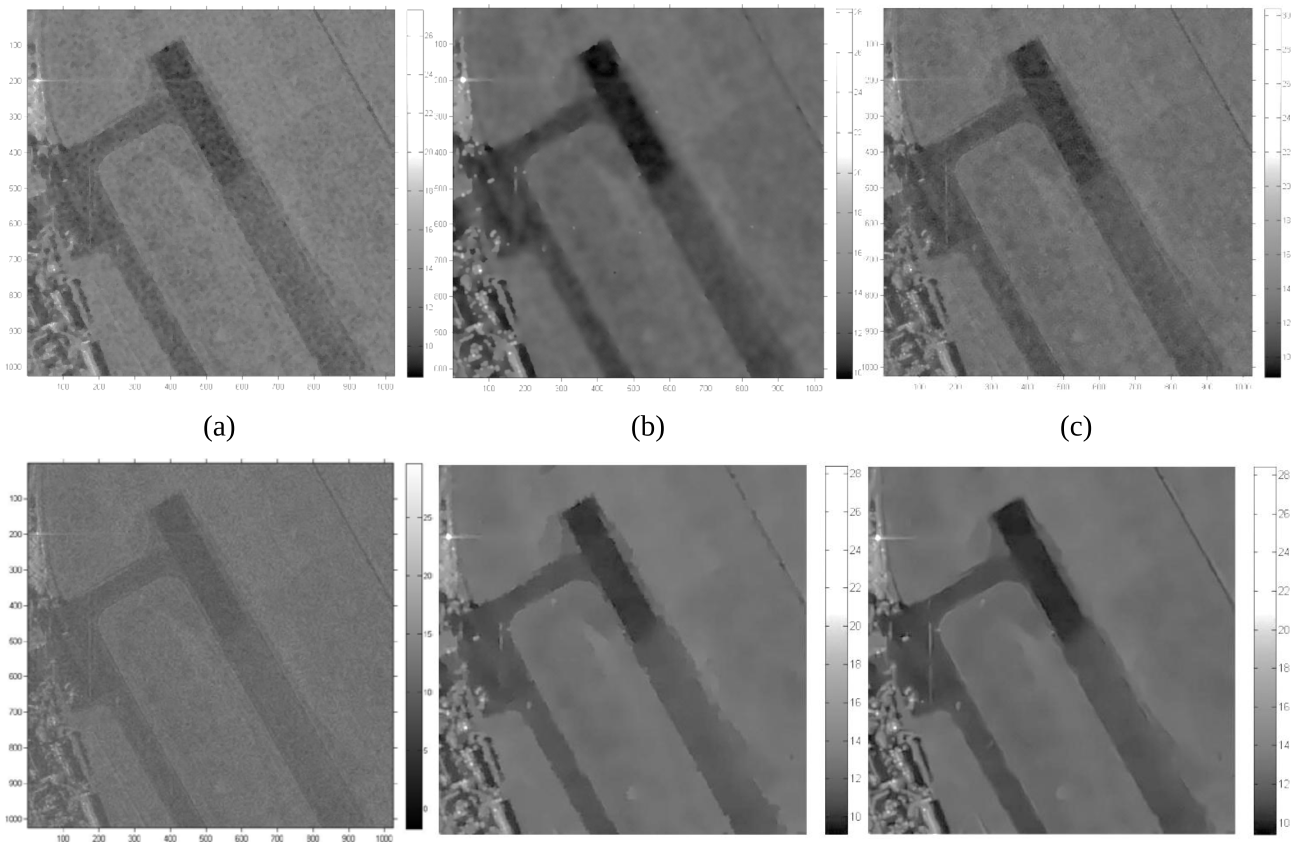


(a) (b)

Fig. 7.17 - (a) Original CSK image (CSK2). (b) Homogeneous region used to compute evaluation indexes.

	$C_{\hat{I}}^2$	$E[\hat{n}]$ / $\sigma_{\hat{n}}^2$
Original	1.2393	-
Lee	0.1650	0.98/0.78
Kuan	0.1599	0.98/0.79
Frost	0.2464	0.91/0.49
Gamma	0.1632	1.01/0.83
Mean	*0.1501*	*1.01/0.952*
PM	0.0452	1.05/ 1.18
SRAD	0.0500	*1.04*/ *1.14*
PM*	0.0347	1.06/ 1.26
IEED*	**0.0334**	1.06/ 1.26
PPB1	*0.1160*	*0.88*/ *0.53*
PPB10	0.2512	0.89/ 0.47
UWD	0.2885	1.06 / 0.54

Tab. 7.4 - Evaluation indexes for CSK2 image. The bold font indicates the best parameter value whereas italic, underlined font indicates the best within each class. The asterisk indicates the use of the optimized numerical schema.



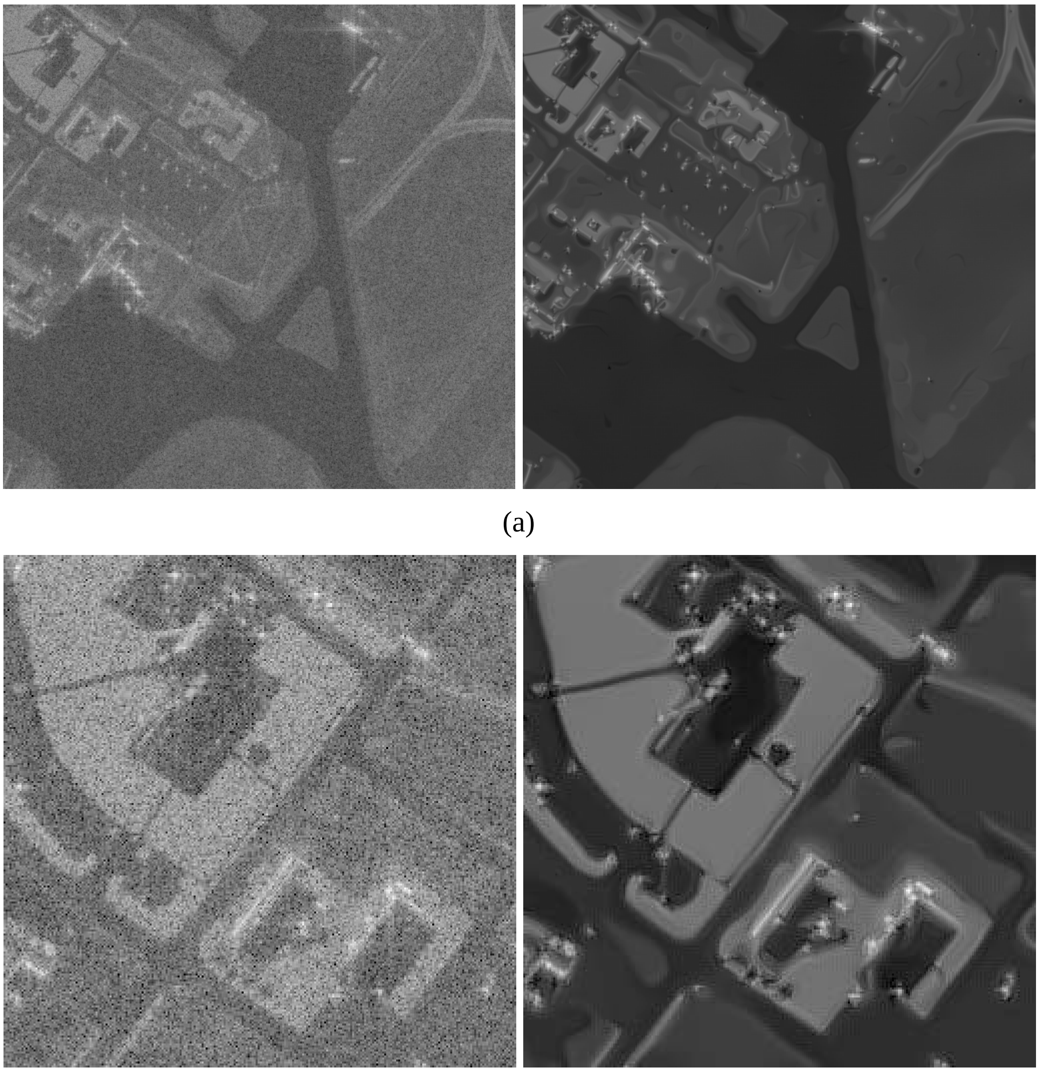
(a) (b) (c)

(d) (e) (f)

Fig. 7.18 - Despeckle results of the best filters of each class. (a) Classic Mean filter (CM). (b) Perona-Malik (PM). (c) Iterated Probabilistic Patch Based (PPB10). (d) Undecimated Wavelet Denoising (UWD). (e) Perona-Malik with optimized schema (PM*). (f) Improved Edge Enhancing Diffusion with optimized schema (IEED*).

7.5.5 CSK Image (Tucson)

The image shown in Fig. 7.19 belongs to the “Tucson” image that depicts an airport (see Section 1.9.2). This image has been acquired in Spotlight 2 mode with polarization **HH** and incidence angle near/far of **23.9°/24.8°** and has a theoretical equivalent number of look equal to **1**. In order to highlight the very impressive results of IEED* filter, only the result of such operator are reported in Fig. 7.19.



(a)

(b)

Fig. 7.19 - Despeckle results of the Improved Edge Enhancing Diffusion with optimized schema (IEED*). (a) Amplitude of “Tucson” image (on the left) and its despeckled version (on the right).
(b) Respective image magnification around airport buildings.

7.5.6 Computational Load

To deepen our analysis of despeckling algorithms computational aspects are as essential as final performance. For what concerns computational load, in Tab. 7.5 execution time of the algorithms is reported. The analysis has been performed on a computer **Intel Core 2 Duo, CPU T9400 @2.53GHz, 1.99GB di RAM**.

Before analysing the computational time it has to be said that **Classical filters** and **Anisotropic Diffusion filter** are not optimized and the straightforward implementation has been used. **UWD filter** utilizes the MATLAB functions *filter* and *filter2* whereas **PPB filters** have their main subroutines even implemented in C.

As can be caught from Tab. 7.5, **UWD filter** shows the minor computational load. Among **Anisotropic Diffusion filters,** PM and SRAD have a quite similar behaviour while IEED is the worst of its category. Finally, **PPB1** has clearly a lower load of its iterative counterpart **PPB10**.

Even for what concerns memory resources some discussions can be carried on. In general, for not optimized codes, a computational improvement can be always obtained utilizing more memory allocation. As example it suffices to think about moving window processing which can be quickly performed saving in memory neighbour pixel values for each pixel instead to really move a window on image pixels. Given an image whose memory allocation is M bytes and considering straightforward implementation, the available memory needed is reported in the last column of Tab. 7.5.

	Computational Time (µs/px)		**Memory (bytes)**
	Image 512x512 (px)	**Image 1024x1024 (px)**	**Image M (bytes)**
Lee	164	173	**M**
Kuan	152	161	M
Frost	174	183	M
Gamma	152	163	M
Mean	*149*	*159*	M
PM	986	126	8M
SRAD	*70*	*67*	*6M*
PM*	74	70	8M
IEED*	171	191	14M
PPB1	*157*	*146*	**M**
PPB10	330	308	2M
UWD	**38**	**23**	9M

Tab. 7.5- Computational load of despeckling filters.