Chapter 3

Edge Detection on SAR Images: Statistical Methods

Many authors, who were influenced from the statistical theory underlying the target detection in early surveillance radars, tried to adapt the same theory to generalize information extraction in SAR data. From this point of view, detection means firstly presupposing a model of the pixels belonging to the background (clutter). Secondly, it means to verify, for each pixel in the image, if it has an acceptable probability of belonging to the background (i.e. if its value is over a certain selected threshold). Clearly, in that case, the speckle is not seen as a noise but its statistical model is exploited to have a desired probability of false alarms P_{FA} at the end of the processing chain. Nevertheless, due to the multiplicative nature of speckle, both amplitude and intensity data (see Eq. (2.19) and (2.13)) has a variance which depends to the underlying RCS of the acquired area. Therefore, in order to have the same performance in different RCS areas, we would have to change the threshold accordingly. CFAR operators are developed to avoid such adaptive thresholding.

3.1 Constant False Alarms (CFAR) Operator

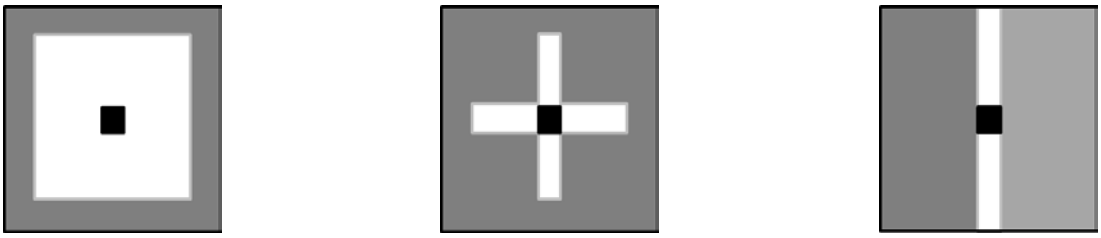
Further in deep, CFAR operators are nothing but statistical hypotheses tests applied at each pixel of the image useful to distinguish from two hypothesis, e.g. the pixels belongs (or not) to a target. Quickly, indicating with H_0 the hypothesis of not having a target at a pixel t, and with H_1 the hypothesis of having a target at the same pixel, the Likelihood Ratio (LR) to verify is:

$$\frac{p(H_1|t)}{p(H_0|t)} = \frac{p(t|H_1)p(H_1)}{p(t|H_0)p(H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{3.1}$$

so that, since a priori probabilities are nearly never known, Eq.(3.1) it is replaced by:

$$\frac{p(t|H_1)}{p(t|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} 1 \tag{3.2}$$

Moreover, even the pdf parameters are often unknown, so that the Maximum Likelihood Ratio (MLR) in Eq. (3.2) becomes a Generalized Likelihood Ratio (GLR), where the unknown pdf parameters are replaced by their estimates. In general, CFAR operators work in a limited spatial area around the pixel under test, i.e. a DxD window is used to define the pixels useful for the computation. In Fig. 3.1 three different kinds of windows used from CFAR operators are shown. In particular, the window used in Fig. 3.1(a) is applied to find anomaly pixels in a homogeneous background, the window in Fig. 3.1(b) is applied for strong target detection and we resort to the window in Fig. 3.1(c) to find vertical edges separating two different regions.



(a) (b) (c)

Fig. 3.1 - Three different kinds of windows. In black the pixel under test, in gray the pixels used for the estimation and in white the neglected pixels. (a) Window used in anomaly detections. In gray are shown the pixels used to estimate the background. (b) Windows used in target detections. The white part is used to exclude pixels belonging to the SAR impulse response. (c) Window used in edge detection. In different gray levels are indicated the two regions across the possible vertical edge (in white).

For what concerns edge detection, since the edges can be oriented in different directions, windows with specific orientation are used (see Fig. 3.2). Usually, applying these oriented windows, a pixel is considered belonging to an edge if it is detected by at least one of the four windows.

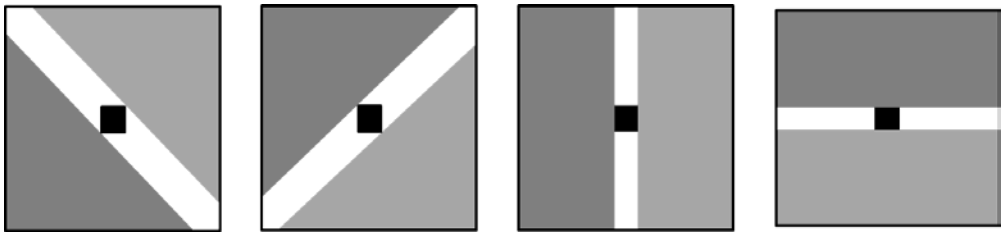


Fig. 3.2 - Four orientations of the edge detector windows. From left to right they are: oblique at 135°, oblique at 45°, vertical (90°) and horizontal (0°).

3.1.1 Generalized Likelihood Ratio

The GLR method to find the presence of an edge between two regions (see Fig. 3.1(c)) was first proposed in [7], where multi-looks intensity data distributed as in Eq. (2.12) are presupposed. In particular, let us consider a DxD sliding window divided by two homogeneous regions indicated by the pixel sets $\{t_i\}_{i=1}^{N}$, $\{t_i\}_{i=N+1}^{2N}$, of N pixels each and with mean RCS σ_1 and σ_2 respectively. Given the hypotheses:

- $H_0: \sigma_1 \neq \sigma_2$;
- $H_1: \sigma_1 = \sigma_2$;

the MLR is given by:

$$\Lambda = \frac{p\left(\{t_i\}_{i=1}^{N}, \{t_i\}_{i=N+1}^{2N} \middle| H_1\right)}{p\left(\{t_i\}_{i=1}^{N}, \{t_i\}_{i=N+1}^{2N} \middle| H_0\right)} = \frac{p\left(\{t_i\}_{i=1}^{N}, \{t_i\}_{i=N+1}^{2N} \middle| H_1\right)}{p\left(\{t_i\}_{i=1}^{2N} \middle| H_0\right)} \tag{3.3}$$

so that, applying the natural logarithm:

$$\begin{aligned} \ln \Lambda = {} & N(L-1)\,\overline{\ln} I_1 + N(L-1)\,\overline{\ln} I_2 - 2N(L-1)\,\overline{\ln} I_0 - NL \ln \sigma_1 - NL \ln \sigma_2 \\ & + 2NL \ln \sigma_0 - \frac{N\bar{I}_1}{\left(\frac{\sigma_1}{L}\right)} - \frac{N\bar{I}_2}{\left(\frac{\sigma_2}{L}\right)} + \frac{N\bar{I}_0}{\left(\frac{\sigma_0}{L}\right)} \end{aligned} \tag{3.4}$$

with $\overline{\ln} I = 1/N \sum_{i=1}^{N} \ln t_i$, $\bar{I} = 1/N \sum_{i=1}^{N} t_i$ and σ_0 the mean RCS presupposing all the pixels belonging to the same homogeneous region. Presupposing $L = 1$ Eq. (3.4) becomes:

$$\ln \Lambda = -N \ln \sigma_1 - N \ln \sigma_2 + 2N \ln \sigma_0 - \frac{N\bar{I}_1}{\sigma_1} - \frac{N\bar{I}_2}{\sigma_2} + \frac{N\bar{I}_0}{\sigma_0} \tag{3.5}$$

so that, replacing σ_1, σ_2 by their Maximum Likelihood (ML) estimates given by the sample means $\bar{I}_1$, $\bar{I}_2$ we have:

$$\ln \Lambda = -N \ln \bar{I}_1 - N \ln \bar{I}_2 + 2N \ln \bar{I}_0 \tag{3.6}$$

with $\bar{I}_0 = (\bar{I}_1 + \bar{I}_2)/2$. Now, in order to fix a certain P_{FA} we need to know how $\ln \Lambda$ is distributed. For this purpose we can express Eq. (3.6) as function of the variable $= \bar{I}_1/\bar{I}_2$:

$$\ln \Lambda = -N \ln r + 2N \ln(z+1) - 2N \ln 2 \tag{3.7}$$

and then we can exploit the formula in Eq. (2.5) to find the $\ln \Lambda$ distribution. Nevertheless, after carrying out Monte Carlo experiments, the author in [7] found that the r.v. $r = \bar{I}_1/\bar{I}_2$ alone manages to obtain the same performances (i.e. same P_D fixing a certain P_{FA}) of the GLR in Eq. (3.7). This is only one of the reasons making the r.v. $r = \bar{I}_1/\bar{I}_2$ so important for the edge detection in SAR images. We shall see in Section 3.1.2 that this operator takes the name Ratio of Averages (RoA) and, due to its good performance in presence of speckle noise, it is one of the most used CFAR edge detector on SAR images.

3.1.2 Ratio of Averages (RoA)

The RoA operator proposed in [8], which computes the ratio between the sample means (i.e. averages) computed in the two opposite part of the sliding window, manages to obtain as good performance as the GLR method in Eq. (3.7), but with a lower computational load. Moreover, in [9] was proved its better performance with respect to the Wilcoxon-Mann-Whitney (WMW), which is one of the best non-parametric CFAR edge detector [9]. In general, given a DxD sliding window that divides two homogeneous regions (see Fig. 3.1(c)) of N pixels each and with mean RCS σ_1 and σ_2 respectively, the RoA edge detector is computed as:

$$r = \frac{\bar{I}_1}{\bar{I}_2} \tag{3.8}$$

where r is greater than 1 when $\bar{I}_1 > \bar{I}_2$ and lower than 1 conversely. Presupposing i.i.d. pixels distributed as Eq. (2.12), the pdf of r has the following closed form [8]:

$$f_r(r) = \frac{\Gamma(2NL)}{\Gamma(NL)^2}\left(\frac{\sigma_1}{\sigma_2}\right)^{NL} \frac{r^{NL-1}}{\left(r + \frac{\sigma_1}{\sigma_2}\right)^{2NL}} \tag{3.9}$$

which can be related to a **Fisher-Snedecor distribution**, also indicated as **F-distribution** or **Variance Ratio distribution**. In particular, the function Γ indicates the Gamma function, so that the same pdf can be written as:

$$f_r(r) = \frac{(\sigma_1/\sigma_2)^{NL}}{B(NL, NL)} \frac{r^{NL-1}}{\left(r + \frac{\sigma_1}{\sigma_2}\right)^{2NL}} \tag{3.10}$$

with B the beta function:

$$B(a,b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \tag{3.11}$$

However, having r limited in $[0,1]$ is helpful so that only one threshold has to be set to have a desired P_{FA} in output. For this reason the RoA is usually expressed as:

$$r = \min\left\{\frac{\bar{I}_1}{\bar{I}_2}, \frac{\bar{I}_2}{\bar{I}_1}\right\} \tag{3.12}$$

whose pdf becomes [8]:

$$f_r(r) = \frac{\Gamma(2NL)}{\Gamma(NL)^2}\left[\frac{(\sigma_1/\sigma_2)^{NL}}{\left(r + \frac{\sigma_1}{\sigma_2}\right)^{2NL}} + \frac{(\sigma_2/\sigma_1)^{NL}}{\left(r + \frac{\sigma_2}{\sigma_1}\right)^{2NL}}\right] r^{NL-1} \tag{3.13}$$

Now, considering the usual hypotheses:

- $H_0: \sigma_1 \neq \sigma_2$;
- $H_1: \sigma_1 = \sigma_2$;

and fixed a general threshold T, the evaluation test is the following:

$$r \underset{H_1}{\overset{H_0}{\gtrless}} T \tag{3.14}$$

see Fig. 3.3 for the pdf of r under the two hypotheses.

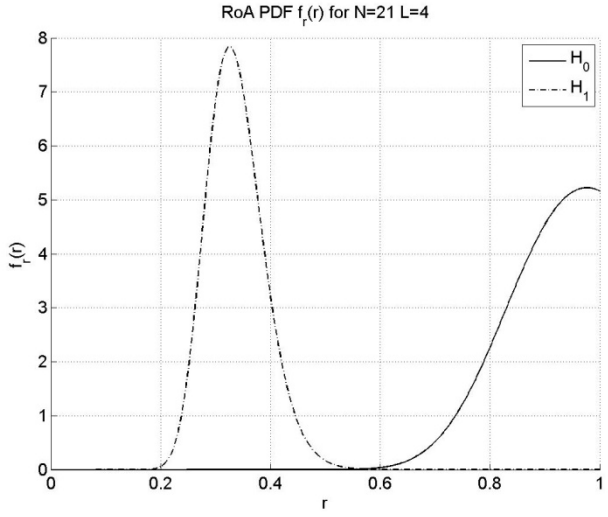


Fig. 3.3 - RoA pdf under the two hypotheses H_0 and H_1 ($\sigma_1 = 3\sigma_2$).

Clearly, the P_{FA} performance of the RoA operator can be computed as:

$$P_{FA} = P\{r < T|H_0\} = \int_0^T f_r\,(r|H_0)dr = \int_0^T \frac{2\Gamma(2NL)}{\Gamma(NL)^2}\frac{r^{NL-1}}{(r+1)^{2NL}}dr \tag{3.15}$$

whereas the P_D is:

$$P_D = P\{r < T|H_1\} = \int_0^T f_r\,(r|H_1)dr = \int_0^T \frac{\Gamma(2NL)}{\Gamma(NL)^2}\left[\frac{(\sigma_1/\sigma_2)^{NL}}{\left(r+\frac{\sigma_1}{\sigma_2}\right)^{2NL}} + \frac{(\sigma_2/\sigma_1)^{NL}}{\left(r+\frac{\sigma_2}{\sigma_1}\right)^{2NL}}\right] r^{NL-1}\,dr \tag{3.16}$$

3.1.2.1 Threshold Computation

First of all, it should be noted that the number of looks L in Eq. (3.13) is nearly always considered known in literature [3], [7]-[9] and equal to the number of incoherent image sums performed in the multi-looks process. However, in applying the multi-looks process described in Section 1.3, correlated looks are actually averaged so that the total effect is always smaller than averaging the same number of independent looks. For this reason, L parameter is often called equivalent number of looks (ENL), i.e. the final image is considered as it were averaged by an equivalent number of independent looks. Estimation of ENL parameter (i.e. the variable L in Eq. (3.13)) can be done by local or global methods. Usually, global methods estimate L parameter form a homogeneous region of the image (computing the inverse of the squared coefficient of variation in Eq. (2.13)), or by using all image pixels so to have an unsupervised method [11]. Instead, in local methods, L is estimated for each pixel of the image resorting to the pixels inside the local window. Moreover, the estimation can be done exploiting the moments of the data pdf, i.e. using the Method of Moments (MoM), or throughout the ML method. Nevertheless, differently from the MoM, the ML estimate of L does not exist in closed form and iterative methods have to be applied. It is worth noting that this last fact has a great impact

on the computational load of local methods, since the iteration (ML estimation) has to be repeated for each image pixel. For the sake of simplicity, in the rest of the document with the parameter L we consider the equivalent number of look, either estimated or known a priori. In general, the computation of the threshold T needs the inversion of the following equation:

$$P_{FA} = \int_0^T \frac{2\Gamma(2NL)}{\Gamma(NL)^2} \frac{r^{NL-1}}{(r+1)^{2NL}} dr \tag{3.17}$$

which last about $5\, ms/px$ utilizing a classical bisection method implemented on MATLAB language and operating on a $32\, bit$ Intel Core 2Duo CPU at $2.52\, GHz$, with $1.99\, GB$ of RAM. This means that, taking into account only the inversion operation, the processing of an $1000x1000$ image lasts about 83 minutes if local estimation methods are concerned (i.e. different L for each pixel). Moreover, for big values of NL ($NL > 500$), due to the finite computer precision, computation problems arise in the Beta function term. Nevertheless, this problem is not considered in the literature and none of the previous articles [3], [7]-[9] treat this computational issue. To solve this problem we related the RoA cumulative density function (cdf) to the **Incomplete Beta function** (see Appendix A) for which many optimized inversion methods exist. Exploiting such relation, the $1000x1000$ image is now filtered in only 10 minutes for local methods. In the rest of the document we use as global method the estimate performed on a homogeneous region selected by the user. Instead, the local method actually refers to the estimations of parameter L by using, separately, the pixels of each region of the local window (i.e. we may have two different values of L, one for each region). However, in order to use such method we have to know the pdf of the ratio between two Gamma r.v. with different shape parameter L. Since that, from the knowledge of the author, this computation is not reported in literature, we made the computation by ourselves (see Section 3.3).

3.1.3 Student T-test (Welch Correction)

The Student T-test relies on the hypothesis of pixel distributed as a Normal (μ, σ^2) , with μ indicates the distribution mean and σ^2 its variance, which does not have to be confused with the RCS parameter of the Gamma distribution in Eq. (2.12). Indicating with $N(\mu_1, \sigma_1^2)$ the pdf of the first region inside the sliding window and with $N(\mu_2, \sigma_2^2)$ the second one, the classical Student T-test needs the following hypothesis to hold:

$$\sigma_1^2 = \sigma_2^2 \tag{3.18}$$

This hypothesis, which is not satisfied for intensity data, holds for logarithmically transformed data. Nevertheless, we decided to use the Welch T-test at the place of the classical one, so that Eq. (3.18) may not hold without compromise the significance of the test. It should be noted that in the rest of the document we always refer to the T-test indicating its Welch generalization. The T-test r.v. is the following:

$$t = \frac{|\bar{I}_1 - \bar{I}_2|}{\sqrt{\frac{S_1^2 + S_2^2}{N}}} \tag{3.19}$$

with N the pixel number of each window region (see Fig. 3.1(c)) and S_1^2, S_2^2, $\bar{I}_1$, $\bar{I}_2$ respectively the unbiased sample variances ($S^2 = \sum_i (t_i - \bar{I})^2/(N-1)$) and sample means of the regions. We can prove [12], that when all pixels inside the window belong to the same distribution (H_0 hypothesis), the t r.v. has a Student pdf with $\hat{v}$ degrees of freedom:

$$f_t(t|H_0) = 2\frac{\left(1 + \frac{t^2}{\hat{v}}\right)^{-\left(\frac{\hat{v}+1}{2}\right)}}{\sqrt{\hat{v}}B\left(\frac{1}{2}, \frac{\hat{v}}{2}\right)} \tag{3.20}$$

with B the usual Beta function:

$$B\left(\frac{1}{2},\frac{\hat{v}}{2}\right)=\frac{\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{\hat{v}}{2}\right)}{\Gamma\left(\frac{1}{2}+\frac{\hat{v}}{2}\right)}=\frac{\pi\Gamma\left(\frac{\hat{v}}{2}\right)}{\Gamma\left(\frac{1}{2}+\frac{\hat{v}}{2}\right)} \tag{3.21}$$

whereas the **degrees of freedom** are computed as:

$$\hat{v}=\frac{(S_1^2+S_2^2)^2}{(S_1^2)^2+(S_2^2)^2}(N-1) \tag{3.22}$$

An example of the T-test under the two hypotheses is reported in Fig. 3.4.

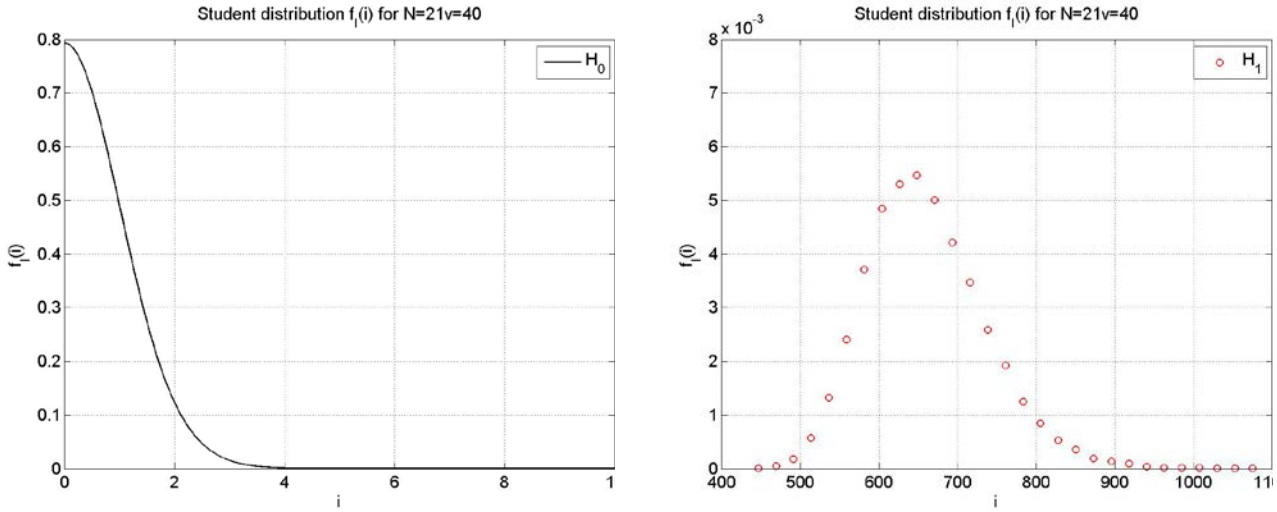


Fig. 3.4 - T-test pdf under the two hypotheses H_0 (on the left), and H_1 with $\mu_1 = 3\mu_2$ (on the right). This last pdf is computed throughout Monte Carlo simulations since it is not known in closed form.

Clearly, computed t in Eq. (3.19) the evaluation test is the following:

$$t \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{3.23}$$

where T is the threshold for which:

$$P_{FA}=P\{t>T|H_0\}=\int_T^{\infty} f_t\,(t|H_0)dt=\int_T^{\infty}2\frac{\left(1+\frac{t^2}{\hat{v}}\right)^{-\left(\frac{\hat{v}+1}{2}\right)}}{\sqrt{\hat{v}}B\left(\frac{1}{2},\frac{\hat{v}}{2}\right)}dt \tag{3.24}$$

3.1.3.1 Threshold Computation

To compute the threshold T so to have a desired P_{FA} we have to reverse the Eq. (3.24). Fortunately, equally to the RoA edge detector, even in this case we can map the integrand function $f_t(t|H_0)$ with the **Incomplete Beta Function** (see Appendix B). Therefore, the inversion requires the same passages as in the RoA case, though the involved variables are different, so that the computational cost of the inversion operation is the same as RoA. Nevertheless, to compute $\hat{v}$ we need to calculate two sample variances whereas in the RoA the computational cost depends on the estimation method of L.

3.1.4 Wilcoxon-Mann-Whitney (WMW) test

Differently from the CFAR operators described so far, the WMW-test[1] is independent from the pdf of the data under test (i.e. it is a **non-parametric test**). As said before, it is one of the best non-parametric CFAR edge detector [9], and it works in the following way. Firstly, it sorts the data within the sliding window (see Fig. 3.1(c)) in ascending order and assigns at each pixel a position index (the lowest value has index 1 the greater the index $2N$), where N is the pixel number of each region. Secondly, for each region it computes the sums of the position indexes of the respective pixels. Next, it compare the two just computed values to decide if the data of the two regions belongs or not to the same distribution. Moreover, the data of one region can be shifted by a value δ in order to find out the distribution difference greater than δ. Even though we describe the WMW test in its general case, in our experiments we have fixed $\delta = 0$ to detect even the finest shift of the distribution. Mathematically, given the two regions R_1 and R_2, i.e. two sets of pixels t_i:

$$R_1 = \{t_i\}_{i=1}^{N};\ R_2 = \{t_i\}_{i=N+1}^{2N} \tag{3.25}$$

we create another set A of pixels $A_i \in A$:

$$A_i = \begin{cases} t_i - \delta & , t_i \in R_1 \\ t_i & , t_i \in R_2 \end{cases} \tag{3.26}$$

Now, indicating with P_i the position index of the pixel t_i retrieved by sorting the set A in ascending order, we compute the sums of the indexes P_i of the pixel in R_1:

$$W_A = \sum_{i=1}^{N} P_i \tag{3.27}$$

The same reasoning applies for the second region R_2, but creating another set B of pixels $B_i \in B$:

$$B_i = \begin{cases} t_i + \delta & , t_i \in R_1 \\ t_i & , t_i \in R_2 \end{cases} \tag{3.28}$$

and computing the sums of the indexes P_i of the pixel in R_2 as:

$$W_B = \sum_{i=N+1}^{2N} P_i \tag{3.29}$$

Now, we compute the following r.v.:

$$W' = \max\{W_A, W_B\} \tag{3.30}$$

Under the hypothesis H_0, which means equal pdfs of the two regions (i.e. no edge inside the window), for not too low N ($N > 20$), it can be proved [13] that the r.v. has a Normal distribution $W' \sim N(\mu, \sigma^2)$ with:

$$\begin{aligned} \mu &= N\,(2N+1)/2 \\ \sigma^2 &= N^2\,(2N+1)/12 \end{aligned} \tag{3.31}$$

so that, normalizing as:

$$W = \frac{W' - \mu}{\sigma^2} \tag{3.32}$$

the r.v. W has a Standard Normal pdf $W \sim N(0,1)$:

$$f_w(w|H_0) = \frac{1}{\sqrt{2\pi}} e^{-\frac{w^2}{2}} \tag{3.33}$$

Even in this case we have considered the absolute value of W at the place of W itself so to have only one threshold to compute. In this case we have to multiply by 2 the pdf in Eq. (3.33). The pdf of $|W|$ under the two hypotheses is depicted in Fig. 3.5.

[1] Also called W-test, Wilcoxon-test Mann-Whitney-test or MW-test.

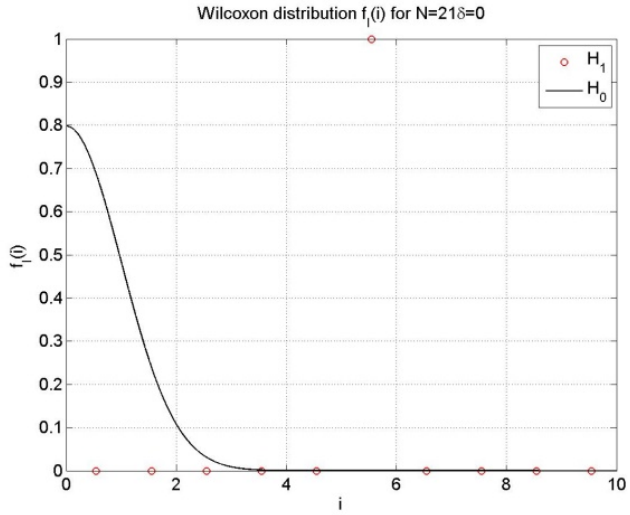


Fig. 3.5 - WMW-test pdf under the two hypotheses H_0 and H_1 with $\mu_1 = 3\mu_2$. This last pdf is computed throughout Monte Carlo simulations since it is not known in closed form.

Clearly, after computing $|W|$ the evaluation test is:

$$|W| \underset{H_0}{\overset{H_1}{\gtrless}} T \tag{3.34}$$

where T is the threshold for which:

$$P_{FA} = P\{|W| > T|H_0\} = \int_T^{\infty} f_{|W|}(w|H_0)dt = \int_T^{\infty} \frac{2}{\sqrt{2\pi}} e^{-\frac{w^2}{2}} dt \tag{3.35}$$

3.1.4.1 Threshold Computation

To compute T in Eq. (3.35) we can use one of the several existing inversion methods from the Normal pdf to the corresponding quantile (see Appendix C). In the case of WMW-test the inversion computation is negligible ($\sim 0.67\ ms$) because it is performed only once since the same T is used for all image pixels.

3.2 CFAR operators on Real SAR Images

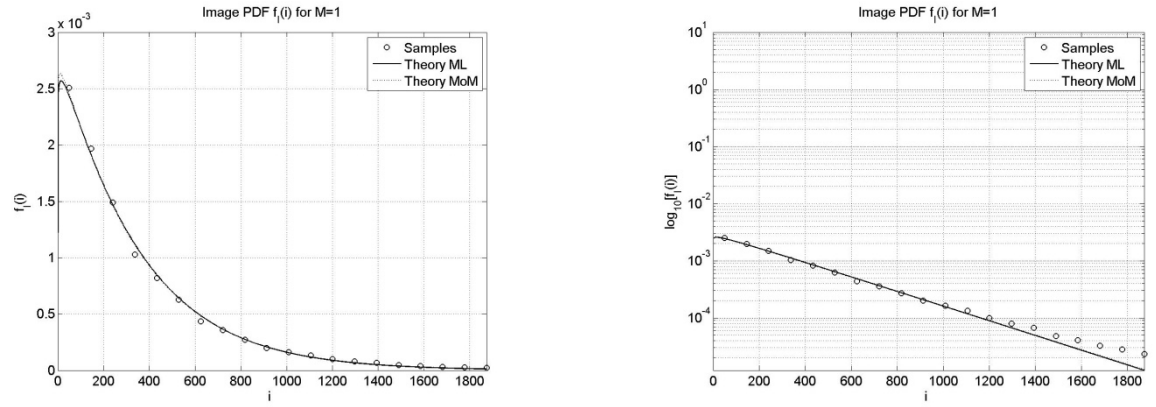
To analyze the results of the CFAR operators descripted so far, we used the SAR images belonging to the public dataset **CLUTTER MSTAR** (see Section 1.9). This choice is firstly due to the existence of several statistical studies on these images and to their diffusion in the scientific community so that some comparisons may be possible.

3.2.1 Effects of Hypothesis Violations

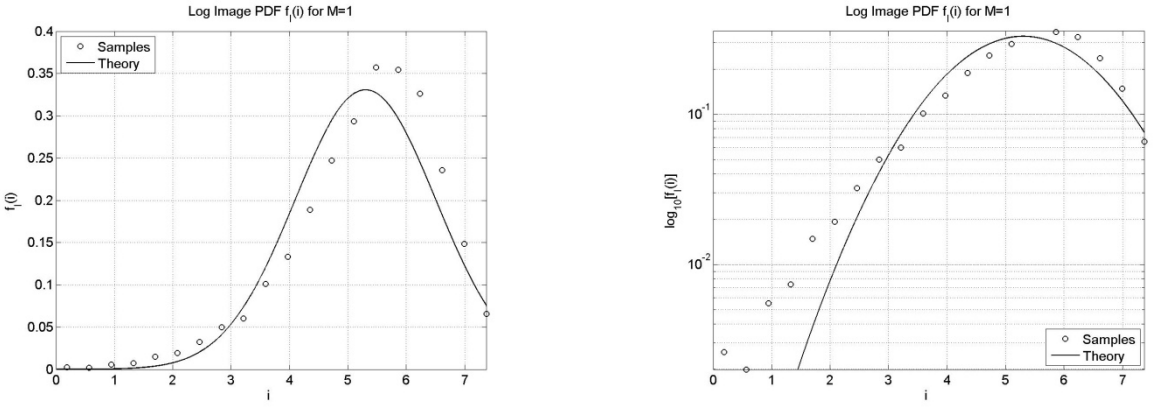
The CFAR operators rely on two fundamental hypotheses, which have to be strictly respected:

- **Hp. 1**: pixels inside the sliding window must be independent;
- **Hp**. **2**: pixels inside the same region of the window must belong to the same pdf.

An important question arising now is what happens when these hypotheses are violated. Let us consider the image “HB06173” in Fig. 3.7(a). Even though the physical formation of the speckle with Gamma pdf (see Chapter 2) is more justified in low resolution images (huge number or scatterers inside the resolution cell), previous study [15] showed how well the Gamma density function fits the pdf of homogeneous clutter in very high resolution as MSTAR data (resolution $\sim 0.3\ m$ in both directions).

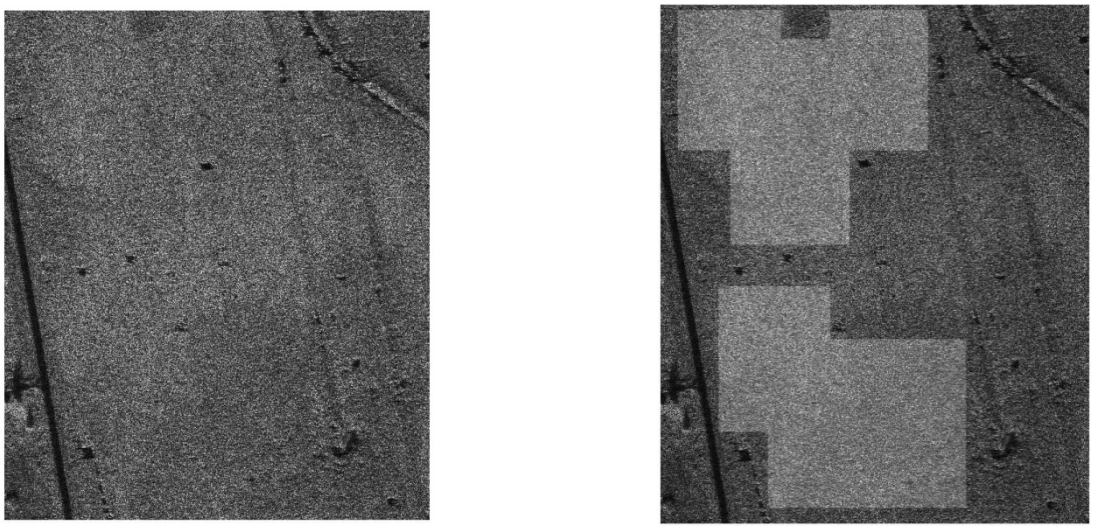


(a)



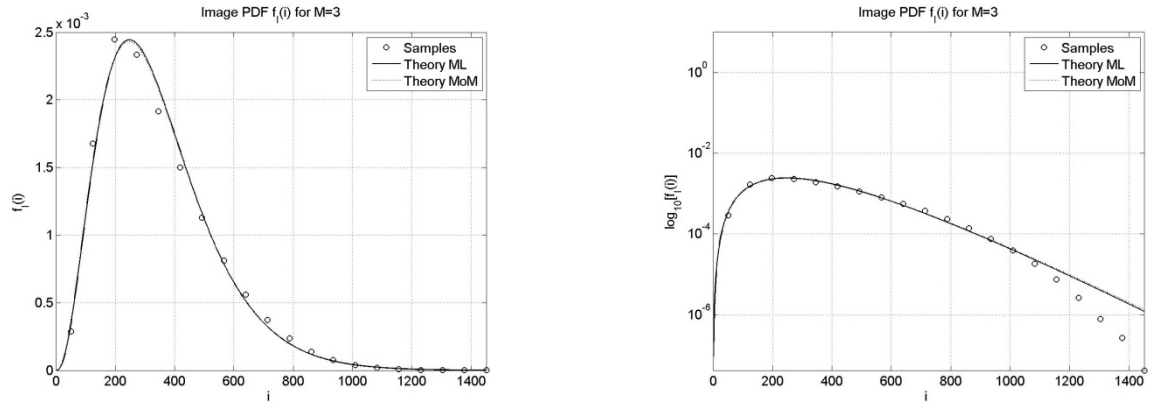
(b)

Fig. 3.6 - Histogram and theoretical pdf of intensity and log-intensity data of the image "HB06173". (a) Histogram of intensity data to which a Gamma pdf is superimposed (with parameter *L* estimated both Mom and ML from data). Both in normal and logarithmic scale on abscissa. (b) Histogram of log-intensity data to which a Normal pdf (with mean and variance estimated from data) is superimposed. Both in normal and logarithmic scale on abscissa.

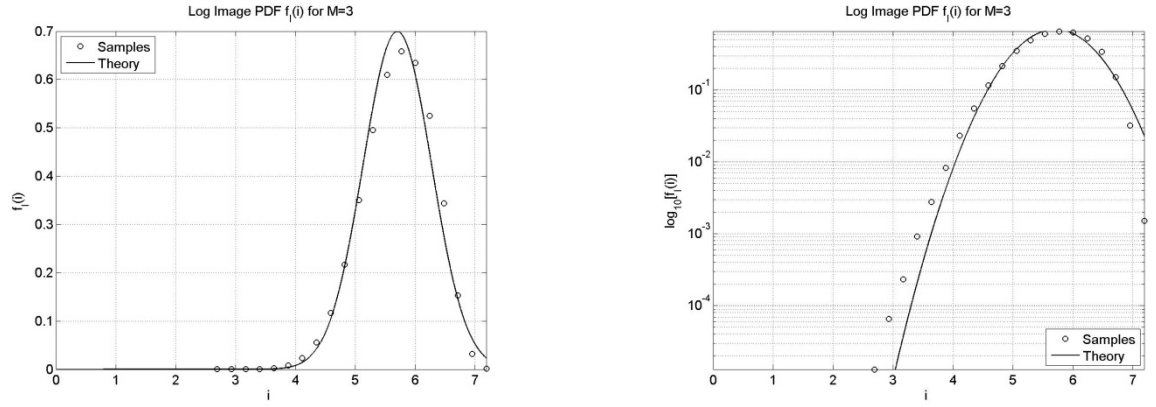


(a) (b)

Fig. 3.7 - (a) Image "HB06173". (b) Image "HB06173" where the homogeneous areas considered to compute the statistical measurements are superimposed.



(a)



(b)

Fig. 3.8 - Histogram and theoretical pdf of intensity and log-intensity data of the image "HB06173" after applying a MxM mean filtering with $M = 3$. (a) Histogram of intensity data to which a Gamma pdf is superimposed (with parameter L estimated both MoM and ML from data). Both in normal and logarithmic scale on abscissa. (b) Histogram of log-intensity data to which a Normal pdf (with mean and variance estimated from data) is superimposed. Both in normal and logarithmic scale on abscissa.

In general, it is worth noting that whereas the K-distribution fits well even high texture data (plowed terrains and woods), the Gamma distribution is more indicated to fit areas with no texture (grass, plain terrain, water basins and general extended targets) [15]. Nevertheless, we can always think to describe areas with high texture by Gamma distribution after applying a sort of mean filtering on that data (i.e. after reducing the spatial RCS variation). To see which CFAR operator can be applied on these images, we report in Fig. 3.6 the histograms of intensity and log-intensity data belonging to the homogeneous area shown in Fig. 3.7. Then, the same measurements are reported in Fig. 3.8 but after applying a mean filtering with a window MxM with $M = 3$.

As can be seen from Fig. 3.6 and Fig. 3.8, the Gamma pdf, which is needed for **RoA** work hypotheses, fits well the intensity data even applying a mean filtering. Instead, the Normal pdf, which is useful for the **T-test**, follows the log-intensity data only after applying a mean filtering. As said before, the **WMW-test**, which is not parametric, does not need any assumption on data distribution so that it can be applied in all previous cases. Nevertheless, the CFAR operators cannot be applied to their respective data directly. In fact, applying the edge detector to the data directly, the results are equal to the one in Fig. 3.9. This is due to the correlation of the data (see Fig. 3.10), which is present even in the original images without mean filtering. This last fact can be appreciated also comparing pixel spacing and resolution entries of Tab. 1.1.

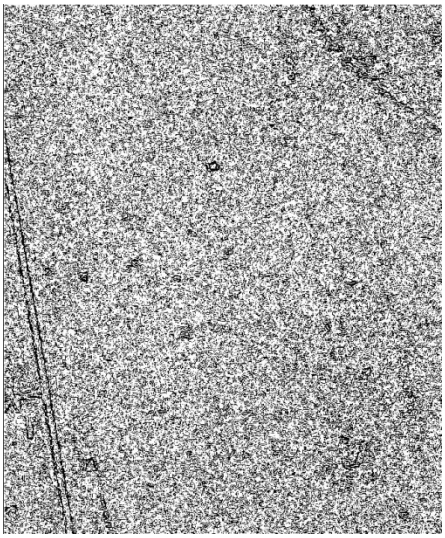
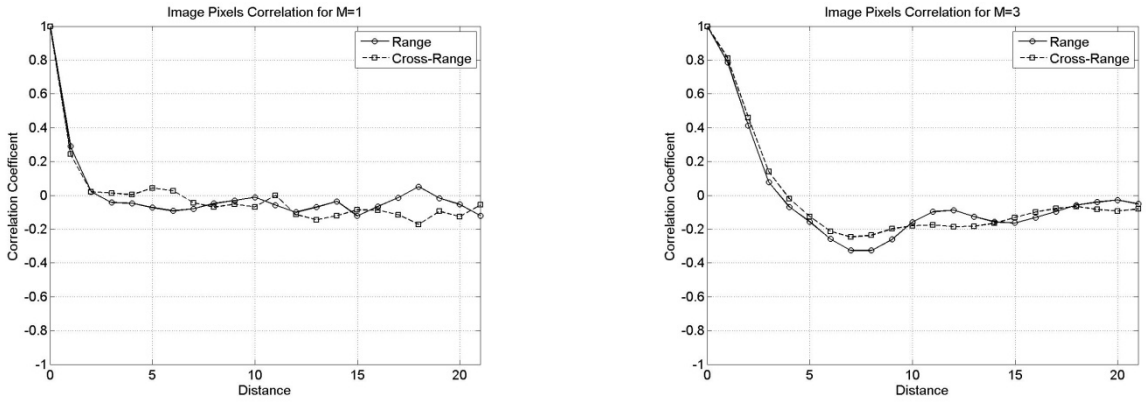
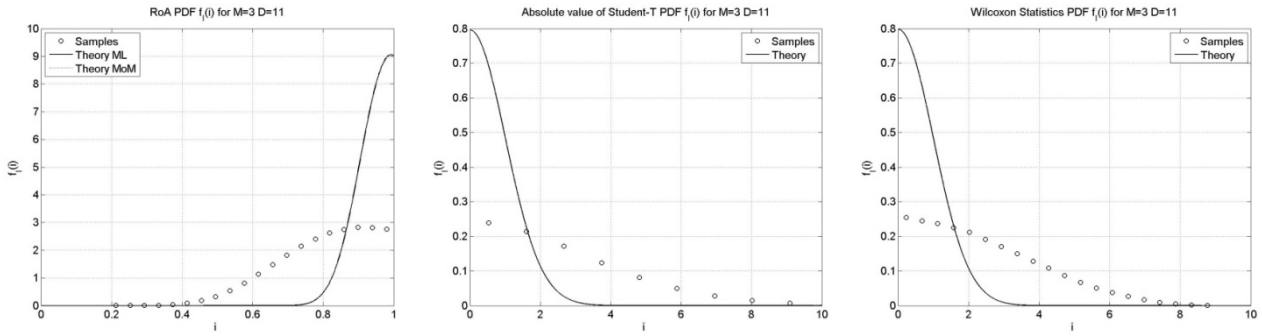


Fig. 3.9 - Application of RoA operator to "HB06173" mean filtered with *M=3*. The RoA filter was applied with the following parameters: global MoM estimation of the parameter *L*, $P_{FA}=10^{-7}$ and filtering window *D=11*.



(a) (b)

Fig. 3.10 - Correlation coefficient of the image "HB06173". (a) Intensity data. (b) Intensity data after applying a MxM mean filtering with $M = 3$.

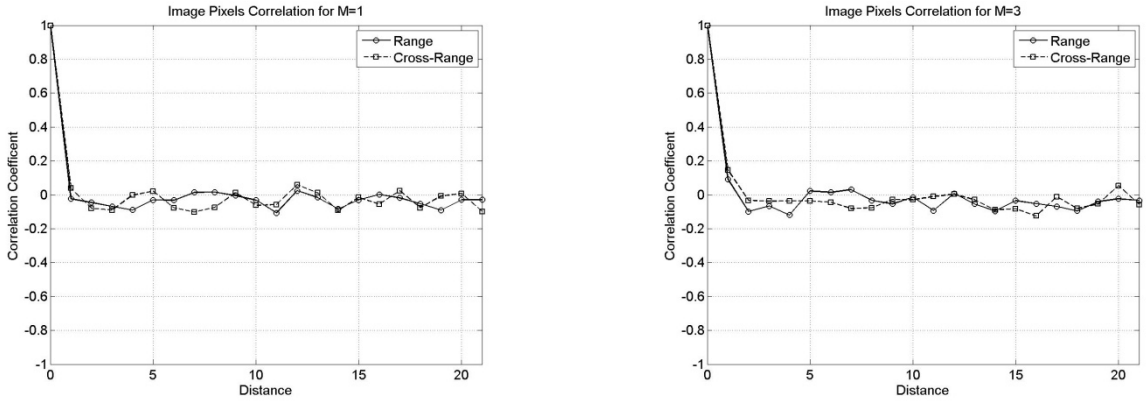


(a) (b) (c)

Fig. 3.11 - Histogram of the various operators with *D=11* applied to "HB06173" mean filtered with *M=3*. (a) RoA filter with global MoM and ML estimation of the parameter *L*. (b) T-test. (c) WMW-test.

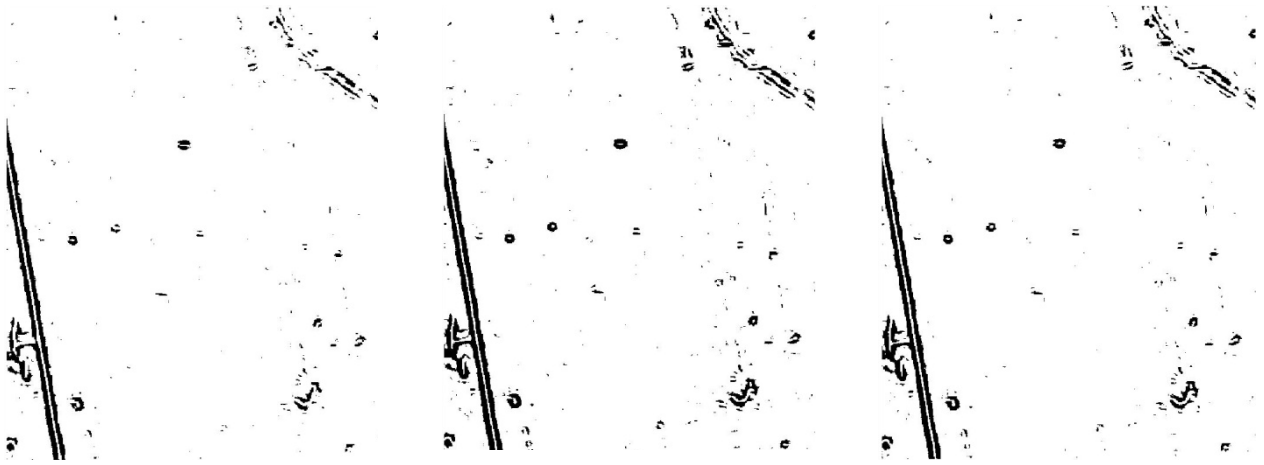
As can be seen from Fig. 3.9, the obtained False Alarm Rate (FAR) is very far from the P_{FA} we set for that case ($P_{FA} = 10^{-7}$). Practically speaking, the correlation of data makes the r.v. of each test have a different pdf with respect to the theoretical one. This fact can be better appreciated in Fig. 3.11, where the theoretical pdf of each operator is superimposed to the respective histogram computed from filtered data. Therefore, due to the data correlation, the **Hp.1** is violated and the consequent effect is that the experimental pdf of CFAR operators is far away from the theoretical one. Consequently, set a threshold relying on the theoretical model is absolutely pointless since the obtained FAR is hugely greater than expected. Clearly, cancel the correlation between near pixels can be done at the expense of resolution in several way [3]. The correct way to lose the minimum information (in a digital processing sense), is that of exploiting the frequency spectrum of the Single Look Complex (SLC) data. In fact, after computing the FFT of SLC data, a weighting can be applied to the frequency samples in order to compensate the system PSF. In this case, the final spectrum can be considered constant (i.e. uncorrelated in time), and a correct downsampling can be performed to respect the Nyquist sampling theorem. Nevertheless, as seen in Chapter 1, constant PSF in frequency enables the maximum spatial resolution to be obtained but at the expense of spreading a lot of point target energy in the PSF side lobes. This means that a trade-off between spatial resolution and radiometric resolution always exists and should be considered for the application. Moreover, equally to the ENL concept explained in Section 3.1.2.1, since CFAR operator pdfs rely on the sum of N independent r.v. (see Section 3.3), the data correlation can also be taken into account considering the model coming out from the sum of N' independent r.v., with $N' < N$. Nevertheless, this fact introduces the additional problem of estimating the correct value of N' from data. However, for the extraction of extended target boundaries in very high resolution data, a simple data downsampling with a factor equal to the estimated correlation length solves the problem without compromise the final detection performance. In fact, differently from target detection on low resolution data, in which fine details and small objects are of fundamental interest, in edge detection issue the boundaries detected in subsampled data can be simply reconstructed as in the original data in a subsequent phase. In the rest of the document, to indicate the performed processing, we report only the values of M (side length of the mean filtering window) and the downsampling factor. For example, the processing $[M = 3, (1:3)]$ indicates an image mean filtered with $M = 3$ and downsampled by a factor $(1:3)$, i.e. taking a sample every three in both dimensions. From Fig. 3.10 we can see that the maximum correlation length occurs for the case $M = 3$ so that, after downsampling by a factor $(1:3)$, we obtain the cancellation of the data correlation, as can be seen in Fig. 3.12. In particular, in Fig. 3.12(b) we can see that a little correlation still remains for $[M = 3, (1:3)]$ data.

The results of CFAR operators on downsampled images are shown in Fig. 3.13 (image $[M = 3, (1:3)]$) and Fig. 3.14 (image $[M = 1, (1:3)]$). As can be seen from these results, all results are much better than the one previously presented in Fig. 3.9. However, the final measured FAR is different in the two images even though the CFAR operator parameters ($P_{FA}$and D) are equal. This fact can be attributed to the correlation that still exist in $[M = 3, (1:3)]$ data. In fact, if we compare the histograms of the operators in the two cases, which are reported in Fig. 3.15 and Fig. 3.16, we can see that only with the processing $[M = 1, (1:3)]$ the theoretical model fits well the experimental one.



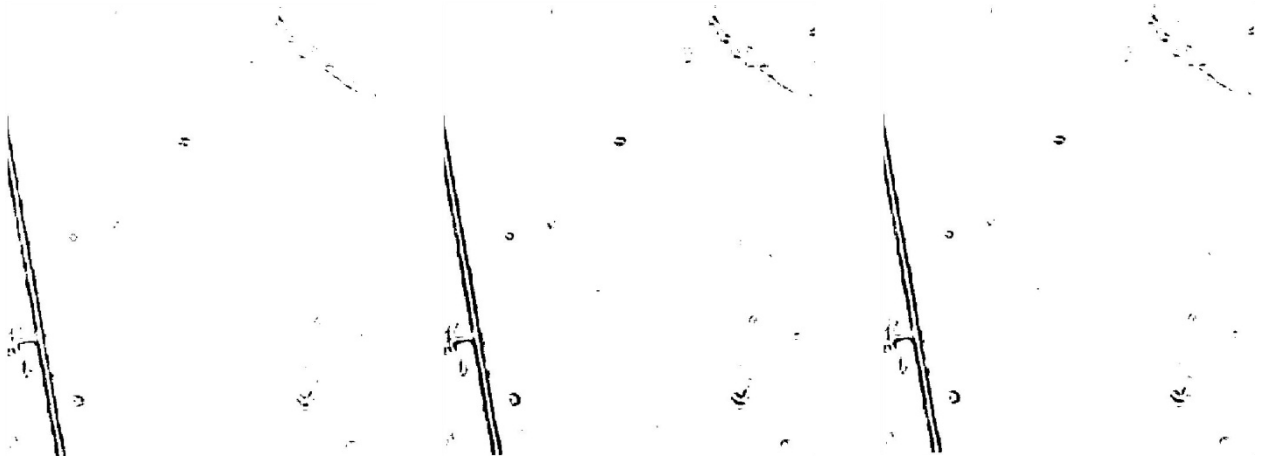
(a) (b)

Fig. 3.12 - Correlation coefficient of the image "HB06173". (a) Intensity data [*M=1,(1:3)*]. (b) Intensity data [*M=3,(1:3)*].



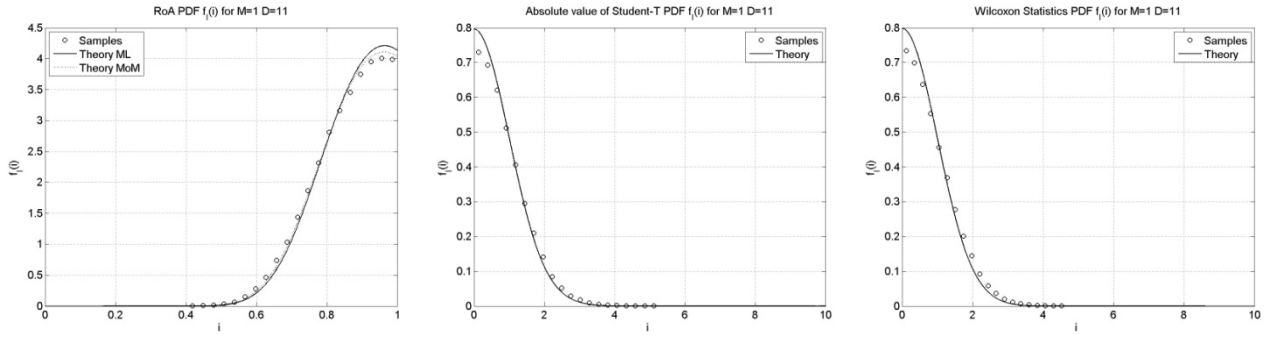
(a) (b) (c)

Fig. 3.13 - CFAR operator results on "HB06173" [*M=3,(1:3)*] with $P_{FA}=10^{-7}$ and *D=11*. (a) RoA with local MoM estimation of the parameter *L* (b) T-test. (c) WMW.



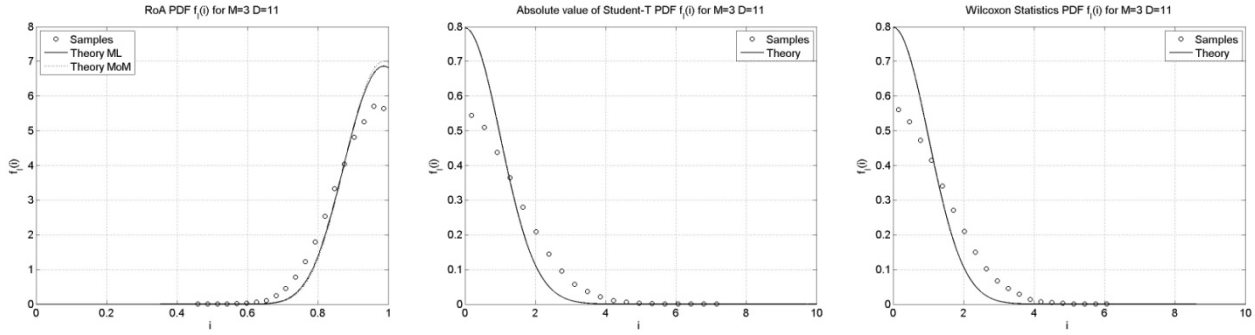
(a) (b) (c)

Fig. 3.14 - CFAR operator results on "HB06173" [*M=1,(1:3)*] with $P_{FA}=10^{-7}$ and *D=11*. (a) RoA with local MoM estimation of the parameter *L* (b) T-test. (c) WMW.



(a) (b) (c)

Fig. 3.15 - Histogram of the various operators with *D=11* applied to "HB06173" [*M=1,(1:3)*]. (a) RoA filter with global MoM and ML estimation of the parameter *L*. (b) T-test. (c) WMW-test.



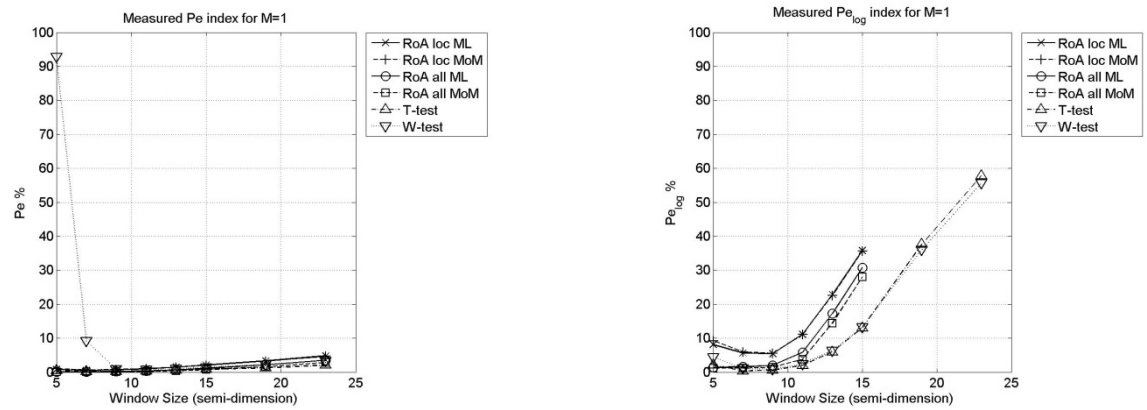
(a) (b) (c)

Fig. 3.16 - Histogram of the various operators with *D=11* applied to "HB06173" [*M=3,(1:3)*]. (a) RoA filter with global MoM and ML estimation of the parameter *L*. (b) T-test. (c) WMW-test.

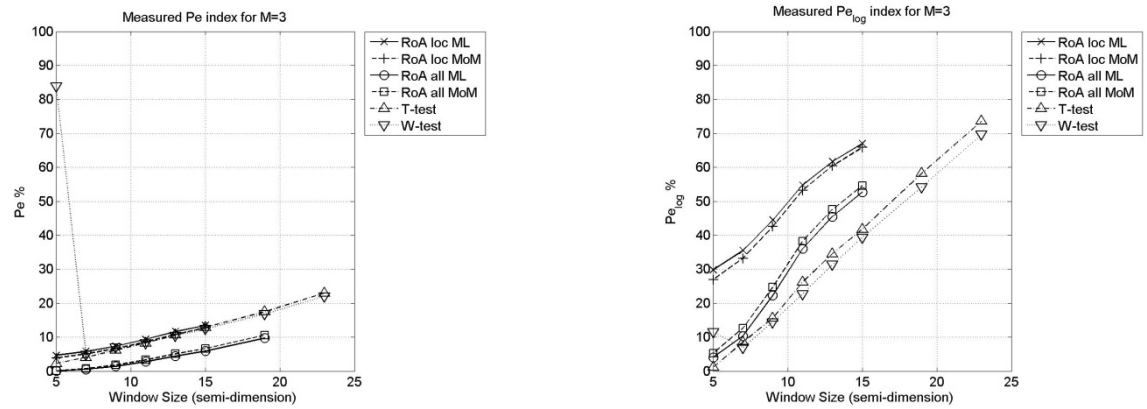
To see what happens varying the side length D of the sliding window used by CFAR edge detectors, we analyze two indexes helpful to measure the distance between a theoretical pdf and an experimental one. In this particular case we use the square error (p_e) to measure the fit between the bodies of the distribution whereas we use the square error in log scale ($p_{e_{log}}$) to measure the fit between the tails. Indicating with $f(t)$ the theoretical pdf and with $\hat{f}(i)$the experimental one (i.e. data histogram), where i is the central value of the histogram bin, the indexes are computed as:

$$p_e = \frac{\sum_i \left[\hat{f}(i) - f(i)\right]^2}{\sum_i f(i)^2}$$
$$p_{e_{log}} = \frac{\sum_i \left[\ln \hat{f}(i) - \ln f(i)\right]^2}{\sum_i \ln f(i)^2} \tag{3.36}$$

In Fig. 3.17 we can see p_e and $p_{e_{log}}$ values of each edge detector, varying D and for the two processing $[M = 1, (1:3)]$ and $[M = 3, (1:3)]$. It is worth noting that the label "RoA loc" refers to the RoA with L estimated locally whereas the label "RoA all" refers to the global method. Moreover, some values of p_e and $p_{e_{log}}$ are not reported for the RoA because for those values of D and L, the computation of the Beta function in Eq. (3.11) gives values outside the memory bound. Comparing Fig. 3.17 (a) and Fig. 3.17 (b) we can notice the strong impact of correlation on the goodness of fit between theoretical and experimental pdf. In fact, as previously noticed in the specific case $D = 11$, the processing $[M = 1, (1:3)]$ has a better fits than $[M = 3, (1:3)]$ for every D. However, raising D the fitness of the theoretical model get worse in both cases, and this can be attributed to the violation of the **Hp.2**, i.e. increasing the size of the sliding window is more probable to pick up pixels belonging to different pdfs. Finally, in both the body and tail part, the estimation of L with the global method gives better fits of data than local one. It should be noted that since in the global method the L estimation is performed by the great number of pixels in the homogeneous areas of Fig. 3.7, the ML and MoM methods yields similar estimated value. Therefore, due to its lower computational burden, the MoM method is preferred (see Fig. 3.18).



(a)



(b)

Fig. 3.17 - Pe (on the left) and Pe_{log} (on the right) - in percentage - varying *D* on "HB06173" for the various CFAR operators. (a) Processing [*M=3,(1:3)*]. (b) Processing [*M=1,(1:3)*].

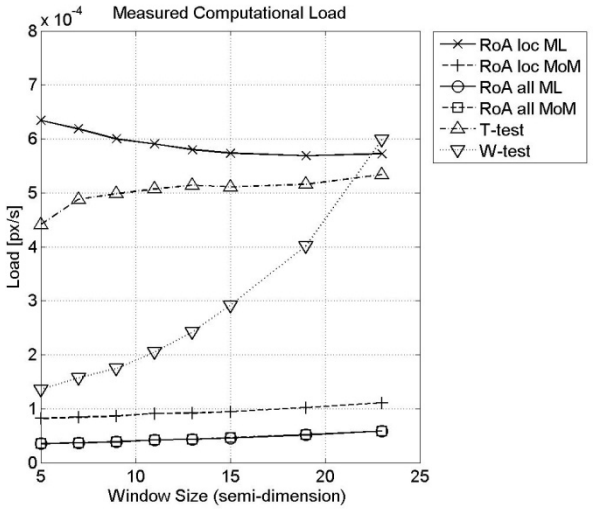
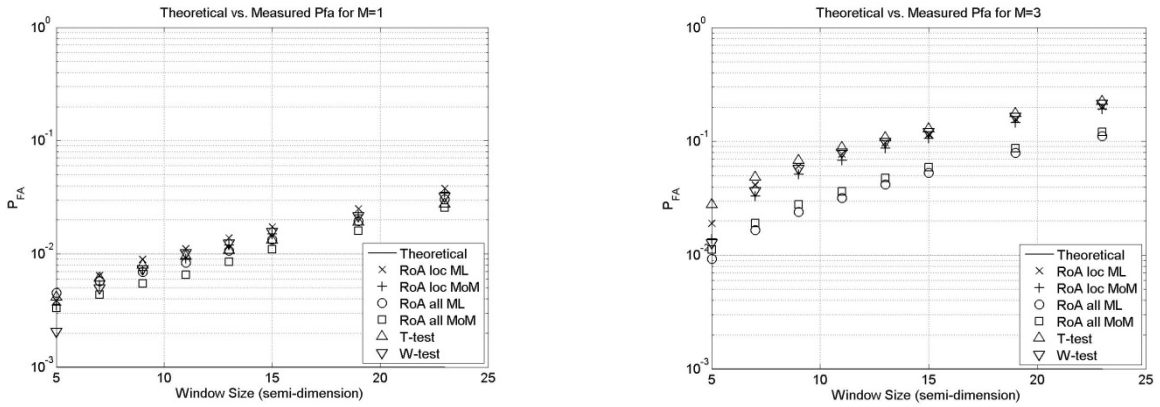


Fig. 3.18 - Computational load of the various CFAR operators varying *D*.

Summarizing, if the **Hp.1** is not respected we make an error in threshold computation because the theoretical model is distant from the real one. This error give rises at a greater FAR with respect to the desired one. Instead, the **Hp.2** is more likely violated raising *D* because the probability to have pixels belonging to a different region increases. Finally, the best estimation method of parameter *L* for the RoA edge detector is the MoM global, where *L* is computed one time from pixels of a homogeneous region of the image. It should be noticed that the values found for *L* by this method is very near to 1 in most images. Since $L = 1$ is the nomi-

nal ENL of MSTAR images, the performance loss setting L by this known value can be considered nearly negligible. Therefore, in this way, the RoA operator can be used in a fully automatic way. Finally, even though performance comparisons among the various operators can be led, this type of comparison suffers from many practical problems. First of all, since the involved P_{FA} are in the order of about $[10^{-9}, 10^{-7}]$, the respective FAR measured on real images will be affected by a great error because the homogeneous region selected could have small inhomogeneous areas that influence the final computation. However, more important than ground truth for FAR computation is the limited number of pixels of the homogeneous regions. In fact, occurrence frequency measured by FAR is different from the corresponding probability generating it (e.g. P_{FA}). As a rule of thumb, in order to have a measure affected by a limited relative error, the FAR should be measured on a number of pixels equal to 100 times the inverse of the probability of occurrence[16] (i.e. on a pixel number of $100/P_{FA}$). Actually, in our experiments we found that, in order to have a relative error lesser than 10% we should measure the FAR on a number of pixel equal to $250/P_{FA}$. Nevertheless, even in the optimistic case of $100/P_{FA}$ occurrences, with a $P_{FA} = 10^{-7}$ we should have an homogeneous area of 10^9 pixels. For example, joining the homogeneous areas shown in Fig. 3.7 (b), which are very extended respect to the most of MSTAR images, we rich a pixel number of about 10^5, so that only $P_{FA} \geq 10^{-3}$ can be safely represented by the FAR index. As confirmed by the analysis in Fig. 3.17, comparing the FAR index for the various operators in both $[M = 1, (1:3)]$ and $[M = 3, (1:3)]$ processing, we found that in the mean filtered case the obtained FAR is more distant from the set P_{FA} ($P_{FA} = 10^{-3}$) and it increases rising the side of the processing window D.

In order to compare the performance of the various CFAR operator in term of probability of detection P_D (or detection rate DR indeed), fixed the same P_{FA} (or FAR), we have generated the semi-simulated image in Fig. 3.20. In particular, to keep the image characteristics (correlation and texture) as in the original image, we have multiplied both $[M = 1, (1:3)]$ and $[M = 3, (1:3)]$ images for a known mask with variable RCS edges.



(a) (b)

Fig. 3.19 FAR index of the various CFAR operator with $P_{FA}=10^{-3}$ on the image "HB06173". (a) Intensity data [$M=1,(1:3)$]. (b) Intensity data [$M=3,(1:3)$].

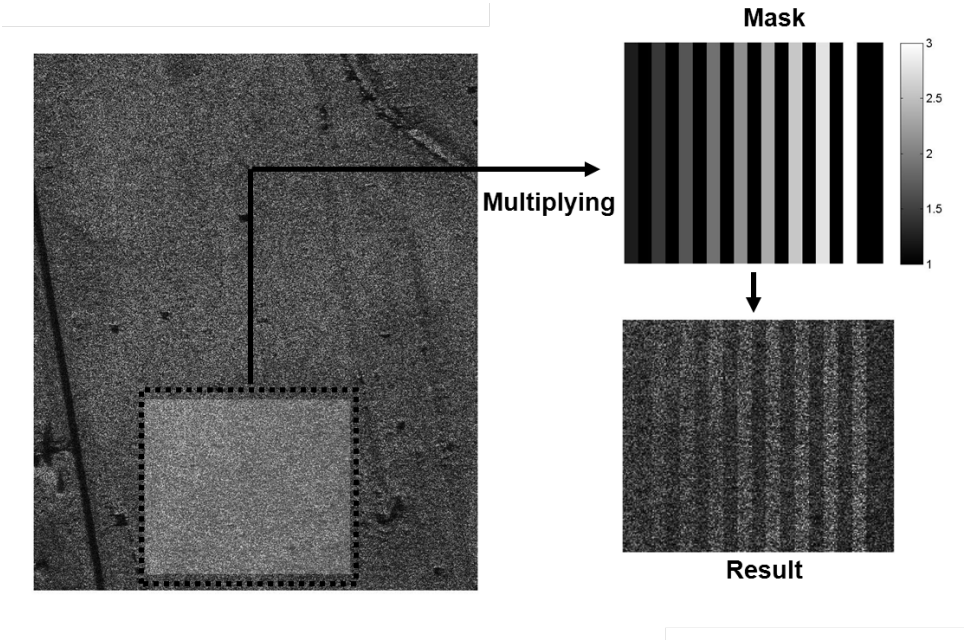
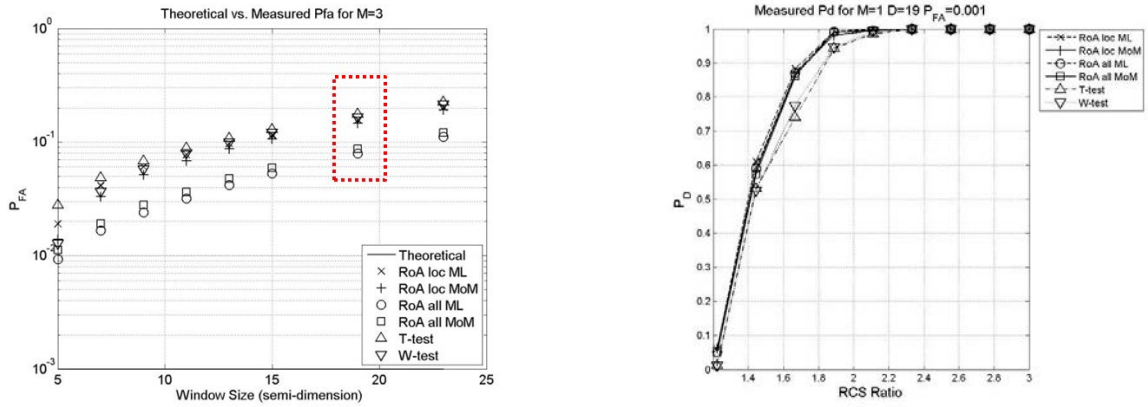


Fig. 3.20 - Semi-simulated image on which DR index varying the RCS ratio is measured.



(a) (b)

Fig. 3.21 Sample of the ROC curve. (a) Highlighted in red the FAR with $D = 19$. (b) DR of the corresponding operators.

In Fig. 3.21 (b) we report the DR index for the various operator with $D = 19$. This point of the receiving operator characteristic (ROC) has been chosen because for this value of D we have near values of FAR but very different values of DR. In particular, we can see that the RoA with L globally estimated has the minimum FAR but an higher DR than T-test and WMW-test. Then, even for the other estimation method the RoA edge detector shown similar FAR but higher DR. Moreover, this type of behavior is kept for all ROC points. This means that on this image, the statistical power of the RoA operator is higher than T-test and WMW-test. Even neglecting this kind of theoretical analysis we can see, by visual inspection, that for the processed MSTAR images, the RoA with L estimated globally yields the best performance.

3.3 RoA Analysis: a New Theoretical Approach

The distribution of the RoA was firstly computed in [8]. Quickly, let us presuppose the pixels t_i as i.i.d Gamma r.v.s with different shape parameter α_i and equal scale parameter β, i.e. $t_i \sim \Gamma(\alpha_i, \beta)$. It can be proved that the sum of such N Gamma r.v. (same shape parameter β and different scale parameter α_i), it is a Gamma r.v. itself, with shape parameter $\alpha = \sum_{i=1}^{N} \alpha_i$ equal to the sum of the single shape parameters and with the same scale parameter β. Therefore, in formulas:

$$\begin{aligned} & t = \sum_{i=1}^{N} t_i \,; \\ & t_i \sim \Gamma(\alpha_i, \beta); \\ & t \sim \Gamma\left(\sum_{i=1}^{N} \alpha_i \,, \beta\right) \end{aligned} \tag{3.37}$$

or, with a little abuse of notation helpful to develop next analyses:

$$\sum_{i=1}^{N} \Gamma(\alpha_i, \beta) \;=\; \Gamma\left(\sum_{i=1}^{N} \alpha_i \,, \beta\right) \tag{3.38}$$

In the next section we will see how this simple consideration helps our analyses.

3.3.1 Distribution Varying Window and Edge Orientations

The main idea behind this study is of retrieving the RoA pdf when, fixed a window orientation (e.g. the last configuration of Fig. 3.2, that is an horizontal filtering window), the edge direction varies (e.g. horizontal, vertical, oblique at 45° and oblique at 135°). Even though this analysis may seem pointless, we will see that from the reached findings we will retrieve many useful considerations other than a performance improving on an edge linking algorithm (see Section 5.5).

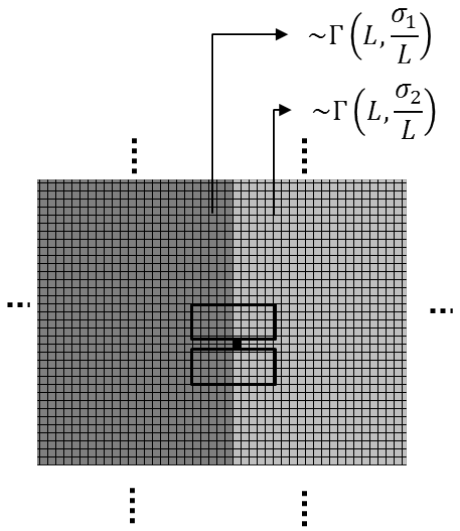


Fig. 3.22 - Horizontal filtering window overlapped to an ideal vertical edge separating two homogeneous regions.

Let us presuppose a horizontal filtering window across a pixel belonging to a vertical edge (see Fig. 3.22). Then, considering the vertical edge across two ideal homogeneous regions R_1 and R_2, whose pixels t_i' and t_i'' respectively are Gamma i.i.d. r.v.s distributed as:

$$\begin{aligned} & t_i' \sim \Gamma\left(L, \frac{\sigma_1}{L}\right); \\ & t_i'' \sim \Gamma\left(L, \frac{\sigma_2}{L}\right) \end{aligned} \tag{3.39}$$

where σ_1, σ_2 are the respective RCS and L the image ENL. At this point, computing the ratio between the sample means of each part of the window, we have:

$$r = \frac{\bar{I}_1}{\bar{I}_2} = \frac{\frac{1}{N}\sum_{i=1}^{N} t_i^{(1)}}{\frac{1}{N}\sum_{i=1}^{N} t_i^{(2)}} \tag{3.40}$$

where we have considered each part of the window composed by the N pixels $t_i^{(1)}$ and $t_i^{(2)}$ respectively. Now, dividing the r.v. with same scale parameter σ_1/L and σ_2/L ,we have:

$$r = \frac{\bar{I}_1}{\bar{I}_2} = \frac{\frac{1}{N}\sum_{i=1}^{N_1} t_i' + \frac{1}{N}\sum_{i=1}^{N_2} t_i''}{\frac{1}{N}\sum_{i=1}^{N_1} t_i' + \frac{1}{N}\sum_{i=1}^{N_2} t_i''} = \frac{X_\Sigma' + X_\Sigma''}{X_\Sigma' + X_\Sigma''} = \frac{Y_1}{Y_2} \tag{3.41}$$

where the X_Σ' and X_Σ'' are distributed as:

$$\begin{aligned} X_\Sigma' &\sim \Gamma\left(N_1 L, \frac{\sigma_1}{NL}\right); \\ X_\Sigma'' &\sim \Gamma\left(N_2 L, \frac{\sigma_2}{NL}\right) \end{aligned} \tag{3.42}$$

with N_1 and N_2 the number of pixel with the same scale parameter inside the same part of the window, i.e.:

$$N_1 + N_2 = N \tag{3.43}$$

Therefore, in order to know the pdf of r in Eq. (3.41), we have to know the pdf of the numerator and denominator. Taking into account the numerator only, we have to know the following pdf:

$$Y_1 = X_\Sigma' + X_\Sigma'' \sim \Gamma\left(N_1 L, \frac{\sigma_1}{NL}\right) + \Gamma\left(N_2 L, \frac{\sigma_2}{NL}\right) \tag{3.44}$$

Nevertheless, when the Gamma r.v.s have different shape parameters σ_i/L, their sum has a pdf containing an infinite sum of terms [17], i.e. it is not easy to manipulate. To solve this problem we can use the following approximation: the sums of generic Gamma r.v.s has pdf Gamma with mean and variance of the real pdf. In [17] was proved that this approximation, even though can seems simple, it is very accurate for a wide range of parameter values and its accuracy improves with the number of r.v.s summed. As an example the computation of the percentile 95% made by the approximate pdf leads to an absolute error of only 0.1% on the desired percentile. Following this approximation (see Appendix D), the numerator pdf becomes:

$$Y_1 \sim \Gamma\left(\frac{L(N_1\sigma_1 + N_2\sigma_2)^2}{N_1\sigma_1^2 + N_2\sigma_2^2}, \frac{N_1\sigma_1^2 + N_2\sigma_2^2}{NL(N_1\sigma_1 + N_2\sigma_2)}\right) = \Gamma(\alpha_1, \beta_1) \tag{3.45}$$

At this point, to compute the pdf of r in Eq. (3.41) we need to know the distribution of the ratio between generic Gamma r.v.:

$$r = \frac{Y_1}{Y_2} \sim \frac{\Gamma(\alpha_1, \beta_1)}{\Gamma(\alpha_2, \beta_2)} \tag{3.46}$$

which can be computed solving the following integral (see Appendix E):

$$\begin{aligned} f_r(r) &= \int_0^{+\infty} f_{Y_1}(r y_2) f_{Y_2}(y_2) dy_2 = \cdots \\ &= \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \left(\frac{\beta_1}{\beta_2}\right)^{\alpha_2} \frac{r^{\alpha_1 - 1}}{\left(r + \frac{\beta_1}{\beta_2}\right)^{(\alpha_1 + \alpha_2)}} \end{aligned} \tag{3.47}$$

and considering the RoA $r = \min\ \{Y_1/Y_2\,, Y_2/Y_1\}$ we have (see Appendix E):

$$f_r(r) = \frac{\Gamma(\alpha_1+\alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)}\left[\left(\frac{\beta_1}{\beta_2}\right)^{\alpha_2}\frac{r^{\alpha_1-1}}{\left(r+\frac{\beta_1}{\beta_2}\right)^{(\alpha_1+\alpha_2)}}+\left(\frac{\beta_2}{\beta_1}\right)^{\alpha_1}\frac{r^{\alpha_2-1}}{\left(r+\frac{\beta_2}{\beta_1}\right)^{(\alpha_1+\alpha_2)}}\right] \tag{3.48}$$

Having done the computation using generic parameters α and β we have freed the result from the case reported as example (horizontal filtering window across a pixel belonging to a vertical edge). Introducing some notation, indicating with $r_{H|V}$ the r.v. of the RoA computed by an horizontal window on a vertical edge, with $r_{H|H}$ the r.v. computed by an horizontal window on a horizontal edge, $r_{H|45}$ indicates horizontal window on 45° edge and $r_{H|135}$ indicates horizontal window on 135° edge. Now, let us consider $N_1^{(i)}$ and $N_2^{(i)}$ the generic number of pixels having different scale parameter inside the part i of the filtering window ($i = 1,2$). Then, indicating with D the side length of the filtering window and with $W = (D-1)/2$ the smaller side of each sub-window (see Fig. 3.23), we can compute $N_1^{(i)}$ and $N_2^{(i)}$ for each particular configuration.

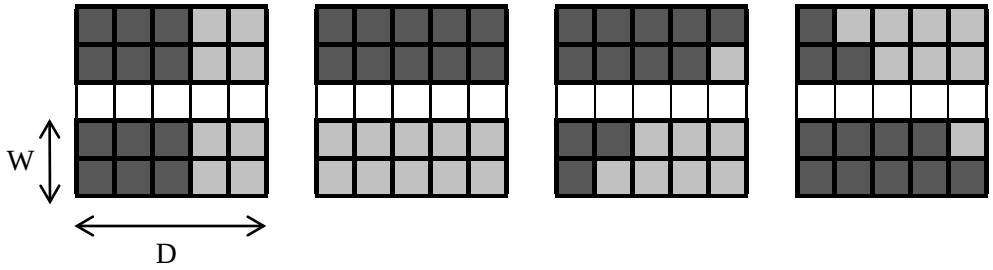


Fig. 3.23 - Horizontal filtering window overlapped to differently oriented edges. The RoA r.v.s are, from left to right: $r_{H|V}$, $r_{H|H}$, $r_{H|45}$ and $r_{H|135}$.

Therefore, as can be seen from Fig. 3.23, for the r.v. $r_{H|V}$ we have:

$$\begin{cases} N_1^{(1)} = (W+1)W \\ N_2^{(1)} = N - N_1^{(1)} \\ N_1^{(2)} = N_1^{(1)} \\ N_2^{(2)} = N_2^{(1)} \end{cases} \tag{3.49}$$

for the $r_{H|H}$ we have:

$$\begin{cases} N_1^{(1)} = DW \\ N_2^{(1)} = 0 \\ N_1^{(2)} = N_1^{(1)} \\ N_2^{(2)} = N_2^{(1)} \end{cases} \tag{3.50}$$

and for $r_{H|45}$ and $r_{H|135}$:

$$\begin{cases} N_1^{(1)} = \sum_{i=1}^{W} i \\ N_2^{(1)} = N - N_1^{(1)} \\ N_1^{(2)} = \sum_{i=W+2}^{D} i = 3N_1^{(1)} \\ N_2^{(2)} = N - N_1^{(2)} = N_2^{(1)} - 2N_1^{(1)} \end{cases} \tag{3.51}$$

As said before, every aforementioned r.v.s can be expressed with the pdf in Eq. (3.48). Since from Eq. (3.45) the generic α and β are:

$$\alpha_1 = \frac{L\left(N_1^{(1)}\sigma_1 + N_2^{(1)}\sigma_2\right)^2}{N_1^{(1)}\sigma_1^2 + N_2^{(1)}\sigma_2^2} \quad \beta_1 = \frac{N_1^{(1)}\sigma_1^2 + N_2^{(1)}\sigma_2^2}{NL\left(N_1^{(1)}\sigma_1 + N_2^{(1)}\sigma_2\right)}$$
$$\alpha_2 = \frac{L\left(N_1^{(2)}\sigma_1 + N_2^{(2)}\sigma_2\right)^2}{N_1^{(2)}\sigma_1^2 + N_2^{(2)}\sigma_2^2} \quad \beta_2 = \frac{N_1^{(2)}\sigma_1^2 + N_2^{(2)}\sigma_2^2}{NL\left(N_1^{(2)}\sigma_1 + N_2^{(2)}\sigma_2\right)} \tag{3.52}$$

the specific pdfs can be obtained by replacing $N_1^{(i)}$ and $N_2^{(i)}$ with their corresponding values in Eq. (3.49), (3.50) and (3.51). For example, in Fig. 3.24 and Fig. 3.25 we can see the previous pdfs with two different values of the RCS ratio $R = \sigma_1/\sigma_2$ and same values of N and L.

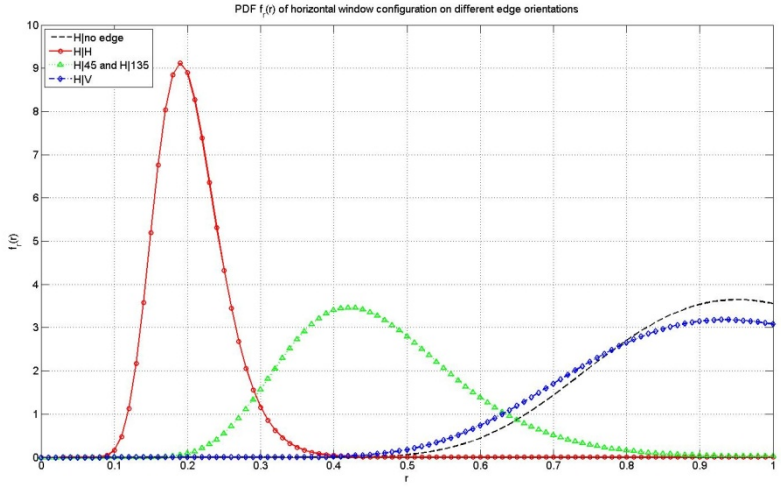


Fig. 3.24 -The RoA pdfs in the case of horizontal filtering window overlapped to differently oriented edges ($N = 10, L = 4, \sigma_1 = 5\sigma_2$).

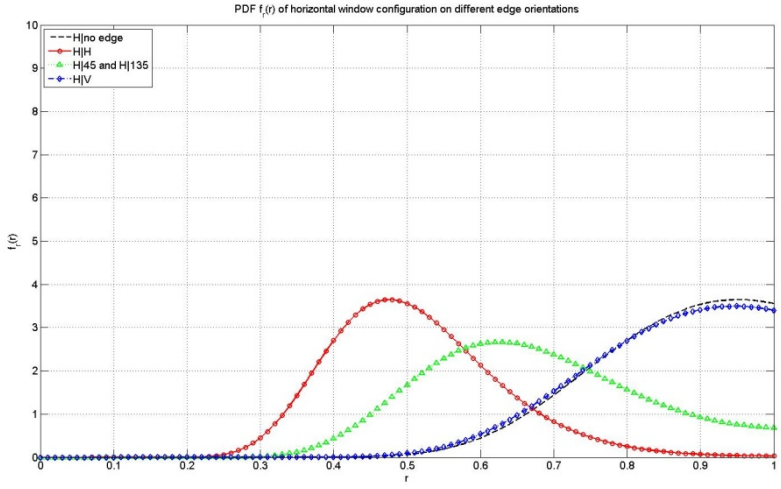


Fig. 3.25 -The RoA pdfs in the case of horizontal filtering window overlapped to differently oriented edges ($N = 10, L = 4, \sigma_1 = 2\sigma_2$).

Clearly, one could think that we missed to compute $N_1^{(i)}$ and $N_2^{(i)}$ of remaining combination between window and edge orientations. Nevertheless, we can prove that exploiting some properties of these pdfs we can retrieve all the remaining configurations from the only one reported in Eq. (3.49), (3.50) and (3.51). More specifically, sorting the windows orientation by grouping in the same category windows with orthogonal orientation (see Fig. 3.26), we surely have:

$$\begin{aligned} &A_i \perp A_j;\ B_i \perp B_j \quad \forall i,j \text{ with } j \neq i \\ &A_i \angle B_j \qquad\qquad\quad \forall i,j \end{aligned} \tag{3.53}$$

we can prove (see Appendix F) that the following properties hold:

Property 1 $f_{r_{A_i|A_i}} = f_{r_{B_i|B_i}} \quad \forall i$ (3.54)

Property 2 $f_{r_{A_i|A_j}} = f_{r_{B_i|B_j}} \quad \forall i,j \text{ with } j \neq i$ (3.55)

Property 3 $f_{r_{A_i|B_j}} = f_{r_{B_j|A_i}} \quad \forall i,j$ (3.56)

Property 4 $f_{r_{A_i|B_j}} = f_{r_{A_k|B_j}} \quad \forall i,j,k \text{ with } k \neq i$ (3.57)

These properties make possible to compute the pdf of each combination between window and edge orientations throughout the knowledge of the distribution of: $r_{H|V}$, $r_{H|H}$, $r_{H|45}$ and $r_{H|135}$. In fact, if we want to know the RoA pdf when a filtering window with orientation 45° is over a vertical edge ($r_{45|V}$), exploiting Eq. (3.56) we have $f_{r_{45|V}} = f_{r_{V|45}}$ and with Eq, (3.57) we have $f_{r_{V|45}} = f_{r_{H|45}}$. Therefore, as stated before, all configurations window/edge can be computed only knowing Eq. (3.49), (3.50), (3.51) and exploiting the aforementioned properties.

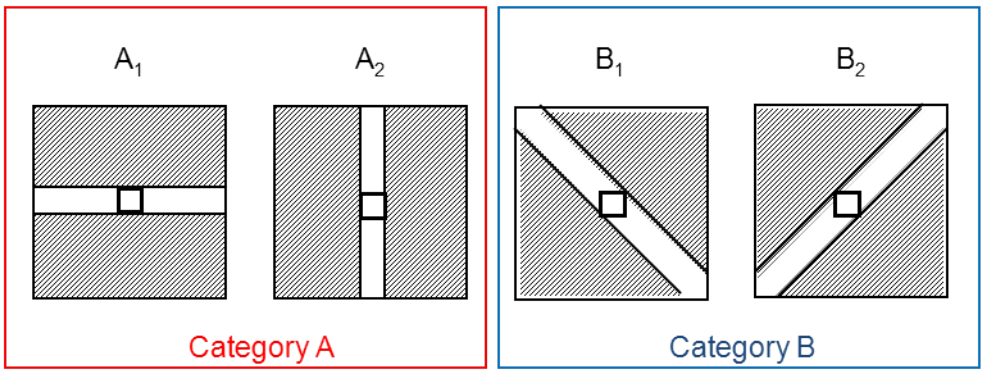


Fig. 3.26 - Grouping in the same category orthogonal orientation of the filtering window.

3.3.2 Window Side Length

In literature [9], [13], following the mathematical relation between the window side length D and the theoretical P_D ,it is often showed how high D values are useful to detect even weak edges, though this implies a loss of small details. Nevertheless, this last consideration was a little misunderstood since it is true only when target detection (and not edge detection) is considered. In fact, since we are interested to linear features (e.g. roads) we have to study the performance of windows with different D on a border with a general width W (which has not to be confused with the side length of each window part used in Section 3.3.1). Presupposing an ideal situation, as the one depicted in Fig. 3.27, where a road of width W and RCS σ_1 is surrounded by a uniform background of RCS σ_2.

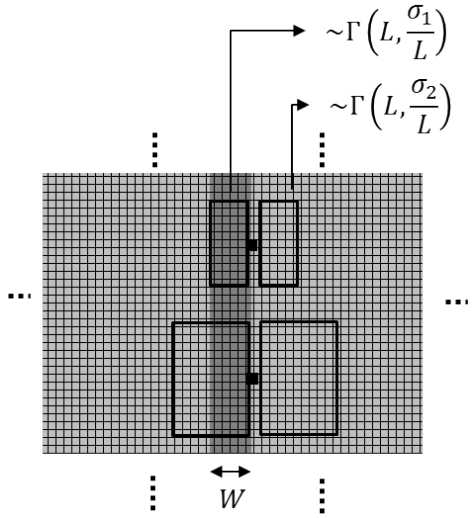


Fig. 3.27 - Vertical windows of different size D on a vertical edge of width W.

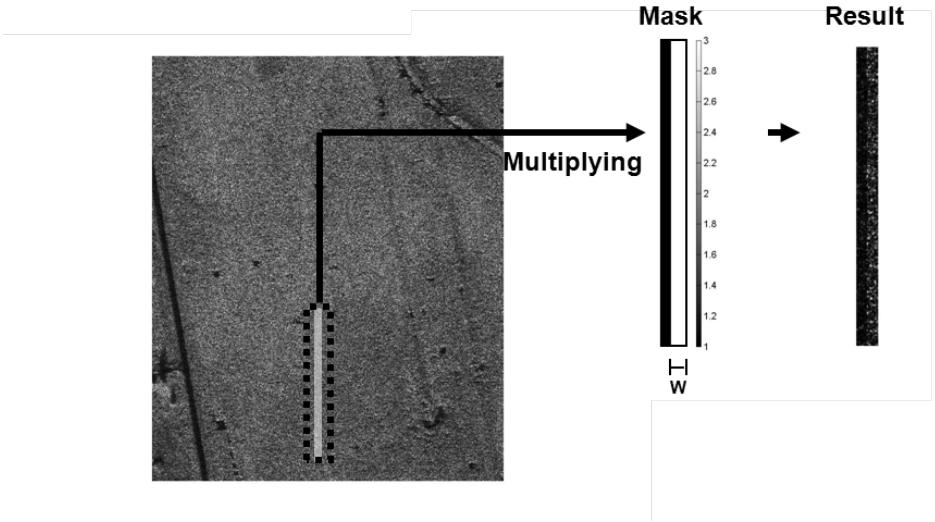
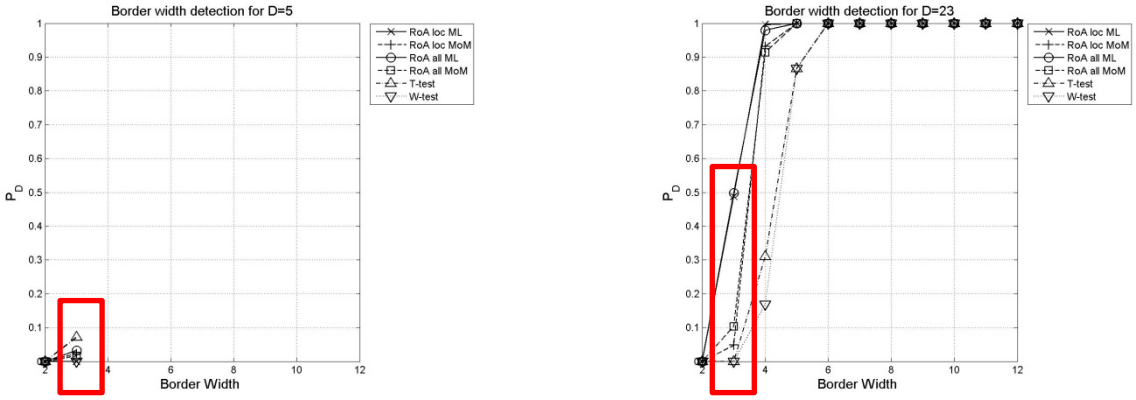


Fig. 3.28 - Simulation of an edge with width W and known RCS ratio.



(a) (b)

Fig. 3.29 - P_D varying W on a simulated image with i.i.d. Gamma pixels. (a) $D = 5$. (b) $D = 23$.

One can expect that the smaller window, which has a part completely overlapped to the road and the other completely filled by background, will have the highest P_D fixed the same P_{FA}. Actually, it happens that the window with highest D detect better even the edge of small width. Both simulating the image in Fig. 3.27 and using the method shown in Fig. 3.28, we noticed a rise of P_D increasing D, for any border width W. Moreover, this fact is very marked for the RoA whereas it is not noticed for the T-test and WMW-test.

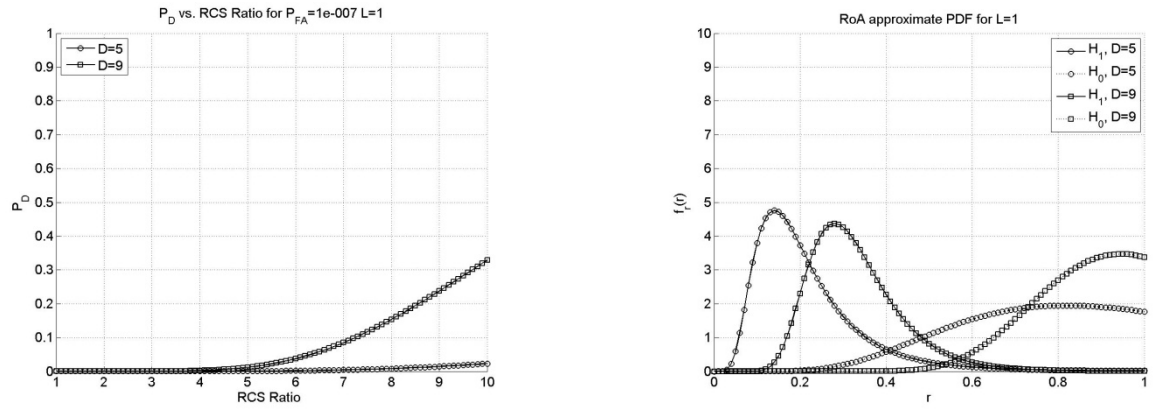
As an example, we can see in Fig. 3.29 that for a road of with $W = 3$ the P_D values of the various RoA in Fig. 3.29 (b) is bigger than the ones in Fig. 3.29 (a). Nevertheless, in the previous simulation the RCS ratio between road and background was equal to 3 and we would be confident of this behavior for any RCS ratio. Once again, our statistical model developed in Section 3.3.1 helps us so that we can retrieve the RoA pdf when the window is overlapped to a border with width W. It can be proved that the RoA r.v. $r = \min\{\bar{I}_1/\bar{I}_2\,, \bar{I}_2/\bar{I}_1\}$ becomes (see Appendix G):

$$f_r(r) = \frac{1}{B(NL_{Eq}, NL)}\left[\frac{(C_\beta)^{NL_{Eq}} r^{NL_{Eq}-1}}{(r + C_\beta)^{NL_{Eq}+NL}} + \frac{(1/C_\beta)^{NL_{Eq}} r^{NL-1}}{\left(r + \frac{1}{C_\beta}\right)^{NL_{Eq}+NL}}\right] \tag{3.58}$$

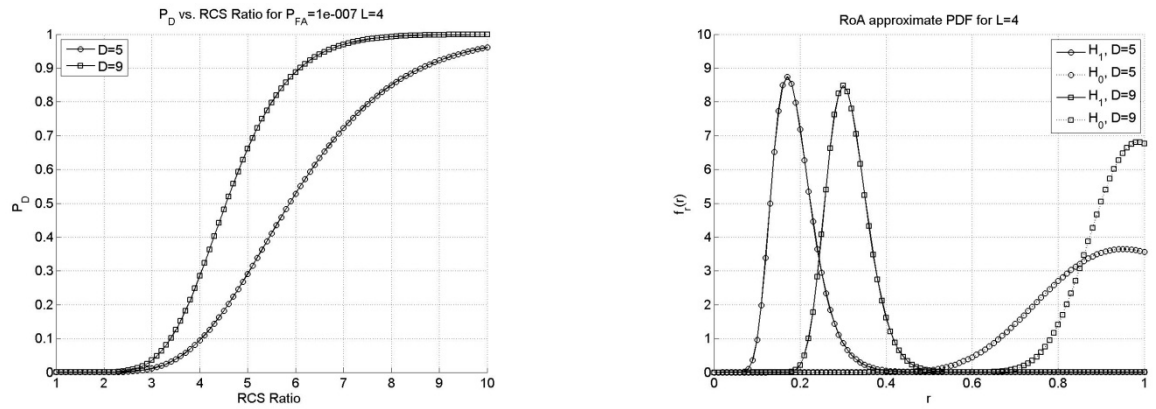
with:

$$\begin{aligned} &R = \sigma_2/\sigma_1\,; \quad C_\alpha = (C_1)^2/C_2\,; \quad C_1 = WDR + (N - WD); \\ &L_{Eq} = LC_\alpha; \quad C_\beta = C_2/(C_1)^3\,; \quad C_2 = WDR^2 + (N - WD); \end{aligned} \tag{3.59}$$

We report in Fig. 3.30the graphs of the P_D varying the RCS ratio for various values of D and L, with $W = 1$. As can be seen, theoretical results confirm the best detection for windows with higher values of D.



(a)



(b)

Fig. 3.30 - P_D varying the RCS ratio with $W = 1$ and$P_{FA} = 10^{-7}$ (on the left) and the respective RoA pdfs when the RCS ratio is equal to 10. (a) $L = 1$. (b) $L = 4$.

3.3.3 Number of Windows and Their Orientations

Until now, to detect an edge we have used four windows with different orientations (see Fig. 3.2), and a pixel was considered belonging to an edge if it was detected by one of the four windows at least (i.e. a logical OR among edge maps was performed). Naturally, using such method the final P_{FA} is different from the case of single window reported in Eq. (3.16). Presupposing that the false alarm events are independent among the windows (i.e. a false alarm due to a window does not imply a false alarm in another one) then:

$$\begin{aligned} &\bar{P}_{FA} = 1 - P_{FA}; \\ &\bar{P}_{FA_4} = \left(\bar{P}_{FA_1}\right)^4; \\ &P_{FA_4} = 1 - \bar{P}_{FA_4} = 1 - \left(\bar{P}_{FA_1}\right)^4 = 1 - \left(1 - P_{FA_1}\right)^4; \end{aligned} \tag{3.60}$$

where the subscript to the P_{FA} indicates the case of single or four windows. It should be repeated that P_{FA_1} is the one reported in Eq. (3.16). Clearly, the hypothesis of independent false alarm events is never respected since the windows are partially overlapped and if a window detects a false alarm it is very probable to have the same false alarm in another window. In [8] has been experimentally verified that applying the four oriented windows in Fig. 3.26 we have:

$$P_{FA_4} = 1 - \left(1 - P_{FA_1}\right)^3; \tag{3.61}$$

It should be noticed that comparing Eq. (3.61) to Eq. (3.60), the case of independent false alarm events among the windows is worse than the correlated case and, fixed P_{FA_1} the final P_{FA_4} becomes greater than the one obtained in the other case. Another question concerns the number of orientation to use to have the best performances. In literature [8], [9], and [13], the four windows in Fig. 3.26 are often used, even though sometimes eight orientations are applied without any justification. In order to address this problem, we have studied the theoretical improving applying a greater number of orientations. Considering applying a vertical window on an edge with a displacement α form the ideal vertical orientation, see Fig. 3.31. The idea is to compute the RoA r.v. when the general displacement α is concerned and then comparing the final performance when four and eight orientation are applied.

.

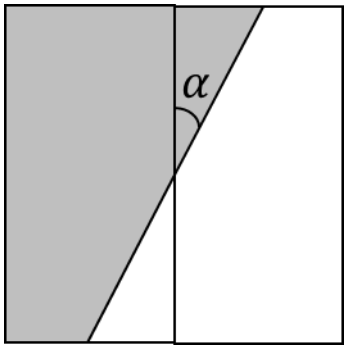


Fig. 3.31 - Vertical window on an edge with displacement α from the ideal orientation.

Once again, exploiting the model developed in Section 3.3.1, presupposing the ideal window in Fig. 3.31 composed by i.i.d. r.v.s Gamma distributed with number of looks L and RCS σ_1 and σ_2 on the different regions, the sample mean in one sub- window would be:

$$\bar{I}_1 \sim \Gamma\left(\frac{L\left(N_1^{(1)}\sigma_1 + N_2^{(1)}\sigma_2\right)^2}{N_1^{(1)}\sigma_1^2 + N_2^{(1)}\sigma_2^2}, \frac{N_1^{(1)}\sigma_1^2 + N_2^{(1)}\sigma_2^2}{NL\left(N_1^{(1)}\sigma_1 + N_2^{(1)}\sigma_2\right)}\right) = \Gamma(\alpha_1, \beta_1) \tag{3.62}$$

with;

$$N_1^{(1)} = \frac{A_\alpha}{A_T} N; \quad N_2^{(1)} = \frac{A_T - A_\alpha}{A_T} N; \tag{3.63}$$

where we indicated with A_α, A_T, the area of the triangle due to the displacement α and the total area of the sub-windows respectively. In particular, considering $0 \leq \alpha \leq \pi/4$, on a DxD window we have:

$$A_T = \frac{D^2}{2}; \quad A_\alpha = \frac{D^2}{8} \tan \alpha; \tag{3.64}$$

which gives:

$$\frac{A_\alpha}{A_T} = 4 \cot \alpha \tag{3.65}$$

so that replacing these relations in Eq. (3.62) gives:

$$\bar{I}_1 \sim \Gamma\left(\frac{NL[\sigma_1 4 \cot \alpha + \sigma_2(1 - 4 \cot \alpha)]^2}{4 \cot \alpha\, \sigma_1^2 + (1 - 4 \cot \alpha)\sigma_2^2}, \frac{4 \cot \alpha\, \sigma_1^2 + (1 - 4 \cot \alpha)\sigma_2^2}{NL\big(\sigma_1 4 \cot \alpha + \sigma_2(1 - 4 \cot \alpha)\big)}\right) = \Gamma(\alpha_1, \beta_1) \tag{3.66}$$

From Eq. (3.66) we can see that though this model the mean value of $\bar{I}_1$ is:

$$\mathrm{E}[\bar{I}_1] = \sigma_1 4 \cot \alpha + \sigma_2(1 - 4 \cot \alpha) = \sigma_1 \frac{A_\alpha}{A_T} + \frac{A_T - A_\alpha}{A_T} \sigma_2 \tag{3.67}$$

that is a weighted mean between the RCS, with weights proportional to the respective areas inside the sub-window. At this point, since $\bar{I}_2$ has the same pdf of $\bar{I}_1$ but with swapped RCS, the ratio betweens sample means becomes:

$$r = \frac{\bar{I}_1}{\bar{I}_2} \sim \frac{\Gamma(\alpha_1, \beta_1)}{\Gamma(\alpha_2, \beta_2)} \tag{3.68}$$

distributed as:

$$f_r(r) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \left(\frac{\beta_1}{\beta_2}\right)^{\alpha_2} \frac{r^{\alpha_1 - 1}}{\left(r + \frac{\beta_1}{\beta_2}\right)^{(\alpha_1 + \alpha_2)}} \tag{3.69}$$

and considering the RoA $r = \min \{\bar{I}_1/\bar{I}_2\,, \bar{I}_2/\bar{I}_1\}$ we have:

$$f_r(r) = \frac{\Gamma(\alpha_1 + \alpha_2)}{\Gamma(\alpha_1)\Gamma(\alpha_2)} \left[\left(\frac{\beta_1}{\beta_2}\right)^{\alpha_2} \frac{r^{\alpha_1 - 1}}{\left(r + \frac{\beta_1}{\beta_2}\right)^{(\alpha_1 + \alpha_2)}} + \left(\frac{\beta_2}{\beta_1}\right)^{\alpha_1} \frac{r^{\alpha_2 - 1}}{\left(r + \frac{\beta_2}{\beta_1}\right)^{(\alpha_1 + \alpha_2)}}\right] \tag{3.70}$$

The case of oblique filtering window on a displaced edge (see Fig. 3.32) is equal to the one computed so far but with a different value of A_α.

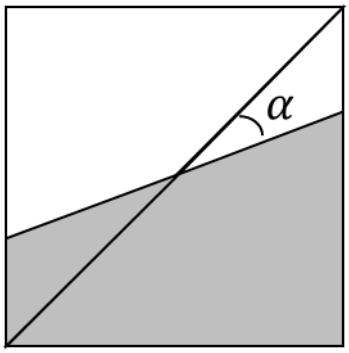


Fig. 3.32 - Oblique window on an edge with displacement α from the ideal orientation.

In fact, it can be immediately retrieved that:

$$A_\alpha = \frac{D^2}{8}\left[1 - \tan\left(\frac{\pi}{4} - \alpha\right)\right] \tag{3.71}$$

Now, fixing a ratio between σ_1 and σ_2 we can compute the final probability of detection due to four windows (indicated as P_{D_4}), which will be function of the displacement α, i.e. $P_{D_4}(\alpha)$. A this point, indicating with $P_{D_1}^{(V)}(\alpha)$ the probability of detection of a vertical windows on an edge displaced by an angle α, and with $P_{D_1}^{(45)}(\alpha)$ the detection probability of an oblique window at 45° on an edge displaced by that orientation of α, it is straightforward to prove that:

$$\begin{aligned} P_{D_1}^{(V)}(\alpha) &= P_{D_1}^{(H)}(\alpha); \\ P_{D_1}^{(45)}(\alpha) &= P_{D_1}^{(135)}(\alpha) \end{aligned} \tag{3.72}$$

where we want to point out that the displacement is always referred to the applied windows. In fact, if an angle is displaced by an angle α from an orientation (e.g. vertical), then it differs by $\pi/2 - \alpha$ from the orthogonal orientation (e.g. horizontal) and by $\pi/4 \pm \alpha$ from the remaining orientations (e.g. oblique at 45° and 135°). Now, considering a border with displacement α from the vertical orientation we have:

$$\begin{aligned} &P\{\text{no detection}|\text{edge exist}\} = \bar{P}_{D_4} = 1 - P_{D_4}; \\ &\bar{P}_{D_4} = \bar{P}_{D_1}^{(V)}(\alpha)\bar{P}_{D_1}^{(H)}\left(\frac{\pi}{2} - \alpha\right)\bar{P}_{D_1}^{(45)}\left(\frac{\pi}{4} - \alpha\right)\bar{P}_{D_1}^{(135)}\left(\frac{\pi}{2} + \alpha\right) \end{aligned} \tag{3.73}$$

where:

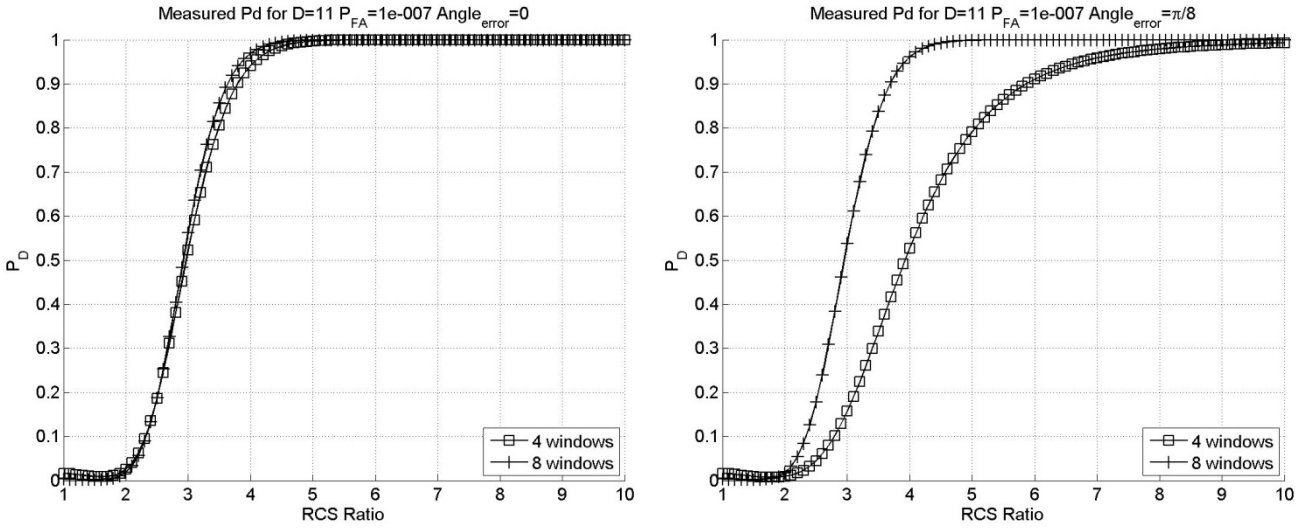
$$\bar{P}_{D_1}^{(W)}(\alpha) = P\{r^{(W)} > T|H_1\} = \int_T^1 f_{r^{(W)}}(r|H_1)dr; \;\; W = H, V, 45°, 135°; \tag{3.74}$$

and $f_{r^{(W)}}(r|H_1)$ the pdf in Eq. (3.48) with the respective parameters for the different four cases ($W = H, V, 45°, 135°$), remembering that:

$$\begin{aligned} f_{r^{(V)}}(r|H_1) &= f_{r^{(H)}}(r|H_1); \\ f_{r^{(45)}}(r|H_1) &= f_{r^{(135)}}(r|H_1) \end{aligned} \tag{3.75}$$

At this point we can compare P_{D_4} to the case of perfect orientation, and even to the case of eight filtering windows, where we added the orientations at $\pm\pi/8$ from the vertical and horizontal one. Clearly, in the case of independent false alarm events, applying eight windows we have:

$$P_{FA_8} = 1 - \left(1 - P_{FA_1}\right)^8; \tag{3.76}$$



(a) (b)

Fig. 3.33 - P_D varying the RCS ratio for the case of 4 and 8 windows, with $P_{FA}=10^{-7}$, $D=11$. (a) $\alpha=0$. (b) $\alpha=\pi/8$.

A comparison of the final P_D in the case of four and eight windows (P_{D_4} and P_{D_8}), fixing the same P_{FA} in output ($P_{FA_4} = P_{FA_8}$), with angular error (displacement) $\alpha = 0$ (i.e. correct alignment) and $\alpha = \pi/8$ is reported in Fig. 3.33.

The results show a better probability of detection for the case of eight oriented windows (for small RCS values the four windows obtain a better P_D, but being very small it is outside our interest). Next, for both cases, the P_D behavior varying the angular displacement $0 \leq \alpha \leq \pi/8$ and fixing a RCS ratio equal to 3.5 (selected to have a P_D of 0.8, which is of our interest) is reported in Fig. 3.34.

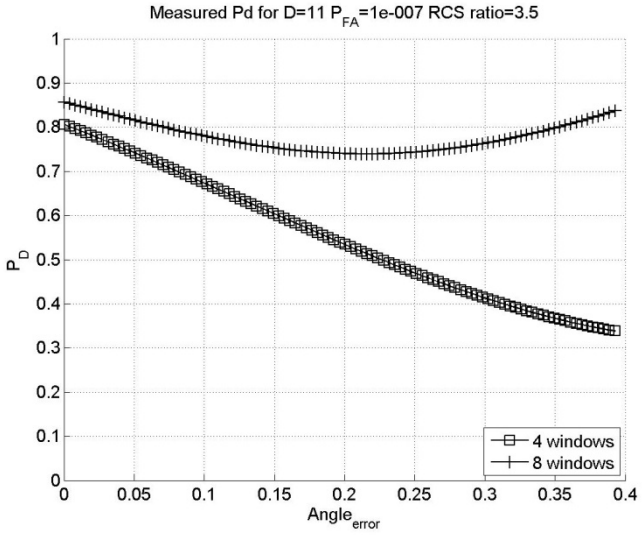
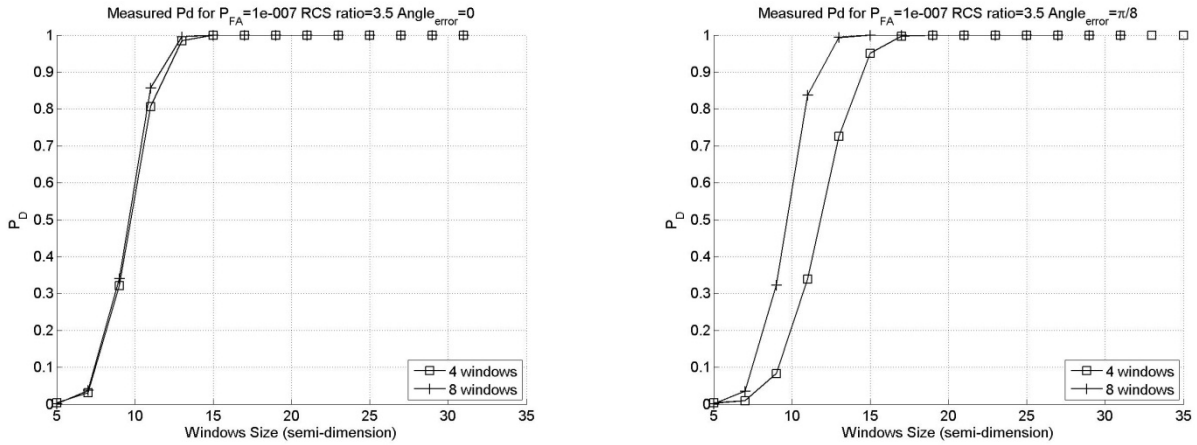


Fig. 3.34 - P_D varying angular displacement $0 \leq \alpha \leq \pi/8$ for the case of 4 and 8 windows, with $P_{FA}=10^{-7}$, $D=11$, and RCS ratio equal to 3.5.

As can be seen from this figure, the P_D is always higher in the case of eight windows, and being periodic varying α, only the value $0 \leq \alpha \leq \pi/8$ are reported. Moreover, as suggested by intuition, the worst case to detect is due to an edge with a displacement which is in-between of two near orientations of the respective cases (i.e. $\alpha = \pi/8$ for four windows and $\pi/16$ for eight windows). Finally, since A_α depends on the window side length D we have computed in Fig. 3.35 the P_D varying D, fixing $P_{FA} = 10^{-7}$, RCS ratio equal to 3.5 and considering the two cases $\alpha = 0$ (i.e. correct alignment) and $\alpha = \pi/8$.



(a) (b)

Fig. 3.35 - P_D varying D for the case of 4 and 8 windows, with $P_{FA}=10^{-7}$ and RCS ratio equal to 3.5. (a) $\alpha=0$. (b) $\alpha=\pi/8$.

Once again, the results show a better P_D for eight windows. Summarizing, using an higher number of windows joining the result by a logical OR operation gives raise to better performance. Naturally, eight windows means a double computational burden with respect to use four orientations alone. It should be noticed that the results are computed considering the false alarm among the windows as independent, i.e.:

$$\begin{aligned} P_{FA_4} &= 1-\left(1-P_{FA_1}\right)^4; \\ P_{FA_8} &= 1-\left(1-P_{FA_1}\right)^8 \end{aligned} \tag{3.77}$$

which can be rewritten, for low values of P_{FA_1} ($P_{FA_1} \leq 10^{-3}$), as:

$$\begin{aligned} P_{FA_4} &\approx 4P_{FA_1}; \\ P_{FA_8} &\approx 8P_{FA_1} \end{aligned} \tag{3.78}$$

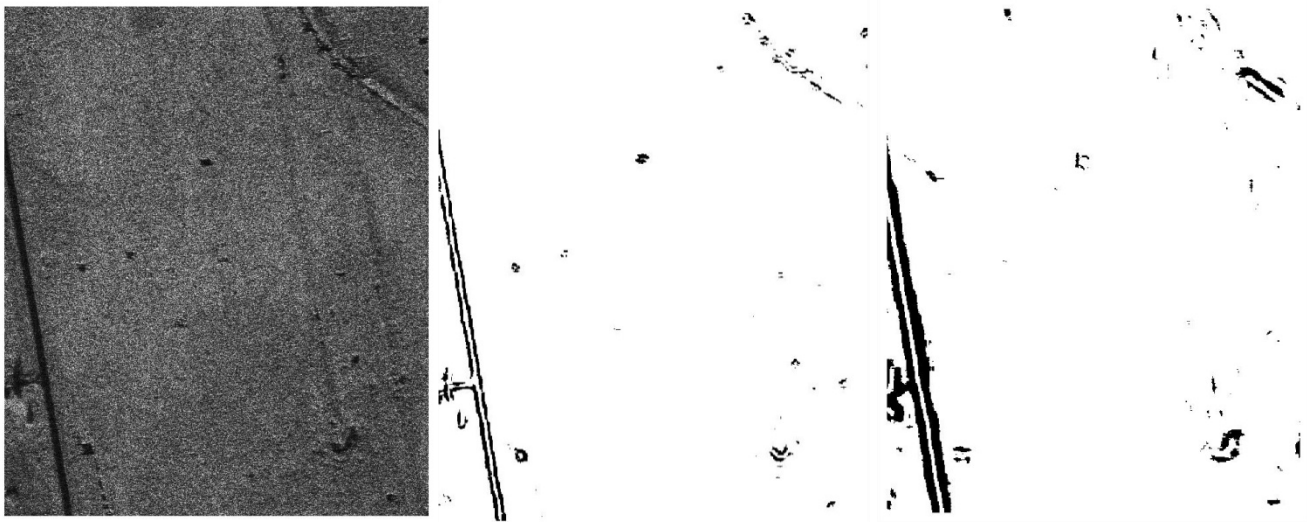
Nevertheless, considering dependent events (as in the real case), it was proved experimentally that:

$$\begin{aligned} P_{FA_4} &\approx 3P_{FA_1}; \\ P_{FA_8} &\approx 5P_{FA_1} \end{aligned} \tag{3.79}$$

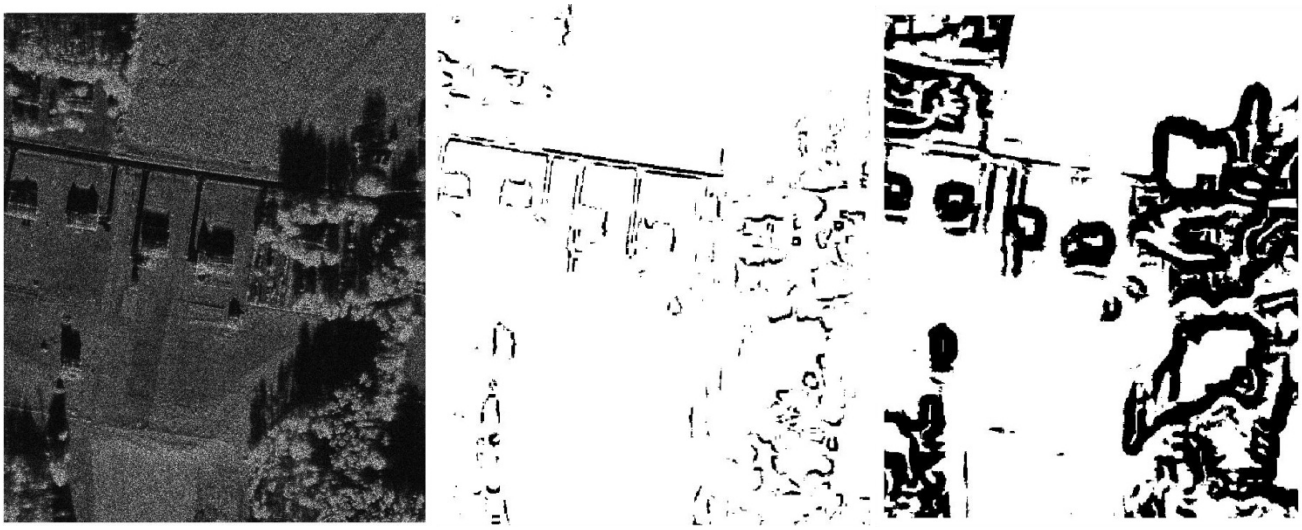
Therefore, respect to the four windows case whit the multiplicative constant passing from 4 to 3, in the eight windows case it passes from 8 to 5. This means that the P_{FA_1} has to be set higher to obtain the same P_{FA_8} and therefore there will be a further increment on the final P_D that will be greater for the eight windows case.

3.3.4 Multiple Borders

Until now, we have seen that increasing the window side length D gives rise to a greater P_D and, for what concern linear features (e.g. roads, rivers, runways, etc.), not only are not there any loss, but they are detected better, for any border width. Nevertheless, the studied case is ideal, with only one border inside the filtering window. Unfortunately, with high D we can have more than one edge inside the window and therefore a minor final P_D. Moreover, if no post processing is concerned, even the thickness of the edge map could be a problem, see Fig. 3.36.



(a)



(b)

Fig. 3.36 - RoA edge detector applied to MSTAR images with $P_{FA}=10^{-7}$ and $D=7$ (in the middle) and $D=23$ (on the right). (a) Image "HB06173". (b) Image "HB06210".

To solve this problem, one can think to take account of the multiple borders in the theoretical model. This consideration was done by the author in [18], who proved that a filter with negative exponential kernel can be seen as the best linear minimum square error estimator (LMMSE) for a wide sense stationary (WSS) stochastic process used to model a signal composed by step edges statistically controlled by a Poisson pdf. It is worth noting that a Poisson pdf of the signal, which is often used to characterize statistical events happening in a certain interval, gives rise to a negative exponential autocorrelation, whose LMMSE estimator was already proved to be a negative exponential function itself (see the Frost despeckling filter in [19], [20]). Nevertheless, claiming the novelty of their findings, the author of ROEWA (ratio of exponential weighted averages) edge detector repeated this computation many years after [18], [19], and [20].

Since the goal of statistical edge detector is to control the P_{FA} in output, the operator in [18] will be discussed in Chapter 4, where we neglect the need to compute an analytical threshold. Instead, in this chapter, to keep the statistical control of filtered data without neglecting the multiple borders problem, we propose two different solutions. The **first solution** concern the application of a filtering window as that reported in Fig. 3.37 (a), which is different from the classical square window reported in Fig. 3.37 (b).

(a) (b)

Fig. 3.37 - Different shapes of a vertical filtering window. (a) Negative exponential shape. (b) Classical square shape.

Clearly, applying the window with the exponential negative shape in Fig. 3.37 (a) means approximating the weights as ratio between integer numbers (i.e. positive rational number). However, this type of approach enables us to know exactly the final pdf of the RoA operator. The **second solution** is to apply the approximate model developed in Section 3.3.1, i.e. computing the distribution of sums of weighted Gamma r.v.s. In fact, after multiplying a Gamma r.v. $t \sim \Gamma(L, \sigma/L)$ with a positive constant c, i.e. $t' = ct$, the resulting r.v. is Gamma distributed as:

$$t' \sim \Gamma\left(L, \frac{c\sigma}{L}\right) \tag{3.80}$$

so that the multiplication modifies the scale parameter. Therefore, using space-varying c_i inside the filtering window gives rise to Gamma r.v.s with different scale parameter whose sum is not Gamma. Nevertheless, using the approximate model developed in Section 3.3.1, the usual RoA operator would be (see Appendix H):

$$r = \frac{\bar{I}_1}{\bar{I}_2} \sim \frac{\Gamma(\alpha_1, \beta_1)}{\Gamma(\alpha_2, \beta_2)} \tag{3.81}$$

where:

$$\begin{aligned} \alpha_1 &= L\frac{(C_\Sigma)^2}{C_{\Sigma^2}} \quad \beta_1 = \frac{\sigma_1}{L}\frac{(C_\Sigma)^2}{C_{\Sigma^2}} \\ \alpha_2 &= L\frac{(C_\Sigma)^2}{C_{\Sigma^2}} \quad \beta_2 = \frac{\sigma_2}{L}\frac{(C_\Sigma)^2}{C_{\Sigma^2}} \end{aligned} \tag{3.82}$$

and:

$$C_\Sigma = \sum_{i=1}^{N} c_i\,; \quad C_{\Sigma^2} = \sum_{i=1}^{N} c_i^2\,; \tag{3.83}$$

In general, neglecting the research of optimal values of weights, let us investigate the pros and cons of applying lower weighs to the pixel farther away from the central one. With equal N, if there is not any border, the result is equal. If the window is perfectly overlapped to a border, there is no difference again. Instead, if the border has a width smaller than the smaller side length of the sub-window, we have two advantages. The first is that, presupposing the background RCS smaller than the border one, the weighted sample mean computed in the sub-window overlapped to the border is greater than the one computed on a classical shape window. This means that the final P_D is greater than in the classical case. The second is that moving the window from the edge to the whole background area, the edge pixel contributes will weigh less and so the thickness on the edge map is kept down. The disadvantage comes out when the window is overlapped to an edge with a displacement. In fact, in that case some background pixels will have high weights whereas some edge ones will have small weights, so that the final sample mean will be lower than in the classical case, i.e. it will have a lower final P_D.