

Interval edge estimation in SAR images

Laércio Dias, Francisco Cribari-Neto, and Raydonal Ospina*

January 31, 2018

Abstract

This paper considers edge interval estimation between two regions of a Synthetic Aperture Radar (SAR) image which differ in texture. This is a difficult task because SAR images are contaminated with speckle noise. Different point estimation strategies under multiplicative noise are discussed in the literature. It is important to assess the quality of such point estimates and to also perform inference under a given confidence level. This can be achieved through interval parameter estimation. To that end, we propose bootstrap-based edge confidence interval. The relative merits of the different inference strategies are compared using Monte Carlo simulation. The results show that interval edge estimation can be used to assess the accuracy of an edge point estimate. They also show that interval estimates can be quite accurate and that they can indicate the absence of an edge. In order to illustrate interval edge estimation, we also analyze a real dataset.

Keywords: Synthetic aperture radar (SAR), edge detection, confidence interval, bootstrap.

1 Introduction

Imagery obtained through the use of coherent illumination suffers from a noise known as speckle. This is the case, for instance, of synthetic aperture radar (SAR) images [1]. The noise is usually modeled in a multiplicative fashion and different inference strategies are used in order to extract useful information from data subject to such speckle noise. Such sensors are particularly useful because they do not require external sources of illumination and their wavelength is not affected by weather conditions.

There are different types of texture for a given region of a SAR image. Textures range from homogeneous (i.e., surfaces with little texture such as lakes, deforestation areas, crop fields, deserts and areas covered by snow), to extremely heterogeneous (i.e., areas that have very intense texture, such as urban areas) with heterogeneous targets, as forests, in between. Such a classification depends on several factors: signal frequency, polarization, and angle of incidence, among others (see [1], section 13.2.3).

One of the most important image processing tools is edge detection, which allows one to detect the position of the edge between two regions. Many edge detection strategies that have been proposed in the literature yield edge point estimates. Our main goal in this

*All authors are with the Departamento de Estatística, Universidade Federal de Pernambuco, Cidade Universitária, Recife/PE, 50740-540, Brazil.

paper is to consider interval edge estimation. We introduce different strategies that can be used to produce a confidence interval for an edge that separates two regions of a SAR image with different textures. The interval estimators are obtained by setting the desired confidence level and their lengths are indicative of how certain one can be that the edge has been located. Interval edge estimation yields important information since point edge estimation can be a difficult task due to speckle noise and since it can signal the absence of an edge, as we shall see.

Approaches of several kinds for detecting an edge between neighboring regions with different textures have been proposed in the literature. Nascimento et al. [2] used stochastic entropies and distances as a criterion for locating edges. Baseline and Ferraioli [3] carried out edge detection based on the ground properties of urban areas using Markov chains to jointly model the amplitudes and interferometric phases of two SAR complex images. Fu et al. [4] proposed an edge detector which uses ‘square successive differences of averages’ as an indicator of ‘edge strength’. Singhal and Singh [5] locate edges by removing of speckle noise using mathematical morphology. Alonso et al. [6] use a robust and unsupervised edge enhancement algorithm based on a combination of wavelet coefficients at different scales. Frery et al. [7] consider polarimetric SAR image region boundary detection based on B-spline active contours. Oliver et al. [8] propose a maximum likelihood method that can be used to detect an edge within a window and determine its location. Touzi et al. [9] provide a constant-false-alarm-rate (CFAR) edge detector based on the ratio between pixel values. Fjørtoft et al. [10] propose a step-edge detector which is optimal in the minimum mean square error under a stochastic multiedge model. Melgar et al. [11] evaluate and compare different edge-detection algorithms developed for high-speckle SAR images. Baseline et al. [12] provide an approach based on the exploitation of real and imaginary parts of single-look complex acquired data. Their technique is developed exploiting Markov random fields. Qian et al. [13] propose an automatic local thresholding algorithm (ALTA) to determine threshold and improve ratio-typical synthetic aperture radar (SAR) edge detector.

Gambini et al. [14, 15] proposed a parametric point estimator (i.e., detector) for the edge location using an objective function that is maximal at the transition points that lie on strip of pixels. Nonparametric detectors have also been proposed. Bovik et al. [16] used nonparametric methods to detect edges under additive Gaussian noise. Beauchemin et al. [17] used an alternative approach based on the Wilcoxon-Mann-Whitney statistic to detect changes in adjacent sets of pixels. Lim and Jang [18] used two sample tests to detect edges in images subject to noise. Girón et al. [19], following the main ideas in [14], introduced detectors based on the Mann-Whitney, squared ranks to variances, Kruskal-Wallis and empirical distribution nonparametric test statistics. The authors used Monte Carlo simulation to compare the small sample performances of their detectors to that of Gambini’s estimator. Their numerical evidence shows that the Kruskal-Wallis and Gambini’s detectors behave similarly, the former however being considerably less costly from a computational viewpoint. Indeed, their numerical results show that edge detection using the Kruskal-Wallis detector is approximately one thousand times faster than that based on Gambini’s maximum likelihood detector.

We chose to construct confidence intervals based on the Kruskal-Wallis point estimator because the available numerical evidence shows that it performs as well as the maximum

likelihood detector (Gambini's detector) at a much lower computational cost. We use bootstrap resampling in order to obtain an estimate of its distribution, which is unknown. The bootstrap methods usually delivers accurate estimates [20]. We also consider different bootstrap interval estimators: basic bootstrap, percentile and two variants percentile- t method that we propose. For details on bootstrap interval estimation, the reader is referred to Efron and Tibishirani [21].

The remainder of the article unfolds as follows. Section 2 introduces the statistical model we shall use to model SAR data. Section 3 presents the Gambini [14, 15] and Kruskal-Wallis detectors. Section 4 considers edge bootstrap interval estimation. Section 5 presents Monte Carlo evidence. Section 6 contains an application that uses real (not simulated) data. Finally, Section 7 offers some concluding remarks.

2 The SAR image model

The multiplicative statistical model is widely used in SAR image data analyses. According to this model, the data are described by a random variable Z which can be viewed as the product of the independent random variables X and Y , where X models the properties of the imaged area (backscatter) and Y models the multiplicative noise (speckle) introduced by the use of coherent illumination which degrades the image quality.

According to the model proposed by Frery et al. [22], the speckle noise is gamma distributed, denoted by $Y \sim \Gamma(L, L)$, and the backscatter can be modeled using the inverse gamma distribution, $X \sim \Gamma^{-1}(\alpha, \gamma)$. Thus, $Z = XY$ is distributed as $\mathcal{G}_{\mathcal{J}}^0$, $Z \sim \mathcal{G}_{\mathcal{J}}^0(\alpha, \gamma, L)$, whose density function is

$$f_Z(z) = \frac{L^L \Gamma(L - \alpha)}{\gamma^\alpha \Gamma(L) \Gamma(-\alpha)} \frac{z^{L-1}}{(\gamma + Lz)^{L-\alpha}}, \quad L \geq 1, -\alpha, \gamma, z > 0. \quad (1)$$

Images represented in the intensity format can be described by the distribution $\mathcal{G}_{\mathcal{J}}^0$, denoted by $Z \sim \mathcal{G}_{\mathcal{J}}^0$. The parameters that index such a distribution are: (i) the number of looks $L \geq 1$, which is a measure of the signal-to-noise ratio, (ii) the scale parameter $\gamma > 0$, which is related to the relative strength between the incident and reflected signals, and (iii) the roughness parameter $\alpha < 0$, which relates to the land type texture. The larger the value of α , the more heterogeneous the area: when $\alpha < -10$ the area is very homogeneous (e.g., pastures), when $-10 < \alpha < -4$ the area is heterogeneous (e.g., forests) and when $-4 < \alpha < 0$ the area is extremely heterogeneous (e.g., urban areas). In what follows we shall assume that the number of looks is known.

Using (1), it can be easily shown that the r th noncentral moment is

$$E[Z^r] = \left(\frac{\gamma}{L}\right)^r \frac{\Gamma(-\alpha - r) \Gamma(L + r)}{\Gamma(-\alpha) \Gamma(L)}, \quad (2)$$

if $-\alpha > r$, and ∞ otherwise. Notice that the value of γ that corresponds to $E[Z] = 1$ is

$$\gamma = \frac{\Gamma(-\alpha) \Gamma(L) L}{\Gamma(-\alpha - 1) \Gamma(L + 1)}. \quad (3)$$

Gambini's edge detection method requires estimation of α and γ . Several estimators for these parameters are available in the literature. For instance, [23] and [24] consider robust

parameter estimation, and [25–27] propose bias-corrected estimators. The work [28] numerically evaluate the accuracy of parameter estimation in small samples. We shall use a Kruskal-Wallis nonparametric detector which requires no distribution assumption and also no parameter estimation.

3 Edge detection in SAR imagery

3.1 The Gambini parametric detector.

The Gambini algorithm is based on the fact that if a point belongs to the edge, then a sample taken from its neighborhood is expected to exhibit a noticeable change in the parameter values which are used to describe the pixel distribution on either side of the edge. Let s be a line segment (detection line) such that, in principle, one point that belongs to s also belongs to the edge that separates two regions. The interest lies in determining which point detection line is a transition boundary. The segment $s = (z_1, \dots, z_N)$ is a strip of pixels obtained by discretization of a straight line on the image. Assume that there is a nonnull intersection between the edge and the detection line and let z_j be a point such that $(z_1, \dots, z_j)$ comes from a $\mathcal{G}_{\mathcal{J}}^0$ distribution with parameters $(\alpha_\ell, \gamma_\ell)$ and $(z_{j+1}, \dots, z_N)$ comes from another $\mathcal{G}_{\mathcal{J}}^0$ distribution with parameters (α_r, γ_r) . Then, the value of j indicates the edge location. Be L the number of looks. To find the transition point at s , consider the likelihood function given by

$$\mathbb{L} = \mathbb{L}(\alpha_\ell, \gamma_\ell, \alpha_r, \gamma_r) = \prod_{k=1}^j Pr(z_k; \alpha_\ell, \gamma_\ell) \quad (4)$$

$$\times \prod_{k=j+1}^N Pr(z_k; \alpha_r, \gamma_r). \quad (5)$$

The log-likelihood function

$$\mathcal{L} = \ln \mathbb{L} = \sum_{k=1}^j \ln f_{\mathcal{G}_{\mathcal{J}}^0}(z_k; \alpha_\ell, \gamma_\ell) + \sum_{k=j+1}^N \ln f_{\mathcal{G}_{\mathcal{J}}^0}(z_k; \alpha_r, \gamma_r),$$

where $j \in \{1, \dots, N-1\}$, is maximized when the index j is the edge. Using Equation (1), it follows that

$$\begin{aligned} \mathcal{L} = & \sum_{k=1}^j \ln \frac{L^L \Gamma(L - \hat{\alpha}_\ell) z_k^{L-1}}{\hat{\gamma}_\ell^{\hat{\alpha}_\ell} \Gamma(L) \Gamma(-\hat{\alpha}_\ell) (\hat{\gamma}_\ell + L z_k)^{L-\hat{\alpha}_\ell}} \\ & + \sum_{k=j+1}^N \ln \frac{L^L \Gamma(L - \hat{\alpha}_r) z_k^{L-1}}{\hat{\gamma}_r^{\hat{\alpha}_r} \Gamma(L) \Gamma(-\hat{\alpha}_r) (\hat{\gamma}_r + L z_k)^{L-\hat{\alpha}_r}}. \end{aligned} \quad (6)$$

Thus, the estimated transition point at s is given by

$$\hat{j}_{GE} = \arg \max_j \mathcal{L}. \quad (7)$$

Edge detection using the Gambini estimator uses information on the pixels that lie in a neighborhood of a rectangular region (detection window) around detection line. Figure 1

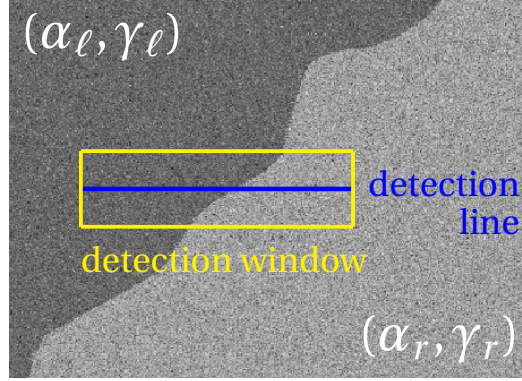


Figure 1: Using pixels in a detection window (yellow), one can, using the Gambini estimator, locate the edge between two distinct regions along a detection line (blue).

shows a simulated SAR image that contains two distinct regions and also a detection window and an inner detection line.

The Gambini's estimator $\hat{J}_{GE}$ requires the estimation of $(\alpha_\ell, \gamma_\ell)$ and (α_r, γ_r) for all possible values of $j \in \{1, \dots, N-1\}$. Such estimations are carried out using rectangular windows in which the detection line coincides with the major axis of the window. Note that for extreme values of j estimation is performed using a small sample, and as a consequence the resulting estimate can be poor. The papers [25] and [27] present estimation approaches that can yield more accurate inferences.

3.2 Nonparametric edge detection.

Edge detection using nonparametric methods as alternative to the Gambini's estimator was proposed by Girón et al. [19]. In what follows we shall use their best performing estimator, namely: Kruskal-Wallis.

Consider k independent random samples (with possibly different sizes) from k identical populations or from k populations such that at least one of them tends to produce observations with larger values, the i th sample containing n_i observations. Let $N = \sum_{i=1}^k n_i$ be the total number of observations. The Kruskal-Wallis statistic was developed to test the null hypothesis that the k -samples come from the same population. If the null hypothesis is false, then at least one of the k samples will tend to produce observations with larger values. In order to compute the Kruskal-Wallis statistic we have to consider N observations and assign ranks to them. Let $R(z_{ij})$ be the rank of z_{ij} , the j th observation of the i th sample, and let $R_i = \sum_{j=1}^{n_i} R(z_{ij})$ be the sum of ranks to the i th sample. The test statistic is given by

$$T = \frac{1}{S^2} \left(\sum_{i=1}^k \frac{R_i^2}{n_i} - \frac{N(N+1)^2}{4} \right),$$

where

$$S^2 = \frac{1}{N-1} \left(\sum_{i=1}^k \sum_{j=1}^{n_i} R(z_{ij}) - \frac{N(N+1)^2}{4} \right).$$

If the number of ties is sufficiently small (or null), it can be shown that the statistic T can be expressed as

$$T_{KW} = \frac{12}{N(N+1)} \sum_{i=1}^k \frac{R_i^2}{n_i} - 3(N+1). \quad (8)$$

Edge estimation using T_{KW} is straightforward. Consider a strip of pixels $s = (z_1, \dots, z_N)$ and assume that the j th element is the edge. The edge thus splits the data into two samples, $(z_1, \dots, z_j)$ and $(z_{j+1}, \dots, z_N)$, whose sample sizes are $n_1 = j$ and $n_2 = N - j$, respectively, where $j \in \{1, \dots, N-1\}$. We conclude that the two samples come from different populations when T_{KW} is large, the corresponding edge estimator being

$$\hat{j}_{KW} = \underset{j}{\operatorname{argmax}} T_{KW}. \quad (9)$$

This edge estimator is computationally less costly than Gambini's parametric estimator.

4 Bootstrap-based edge confidence intervals

It is important for practitioners to be able to compute not only edge point estimates but also edge interval estimates, that is, confidence intervals for SAR image edges. We note that the distribution of KW point estimator is unknown and hence the usual corresponding confidence intervals cannot be obtained. An alternative is to use bootstrap resampling to construct interval estimates [20,21,29]. The main idea is to use data resampling to construct pseudo-samples, to compute the point estimate using each artificial sample and then to use all point estimates to produce a confidence interval.

In what follows we shall consider bootstrap edge interval estimation. In particular, the following bootstrap interval estimators shall be considered: bootstrap basic method (BBM), percentile (PERC), and Studentized (or percentile- t , bootstrap- t) (ST). Given the high computational burden of the percentile- t method, we propose two new variants of such a confidence interval that are much less computer-intensive.

Consider again a strip of pixels $s = (z_1, \dots, z_N)$, with an edge between two regions located at $j \in \{1, \dots, N\}$. Let $\hat{j}_{kw}$ be the KW edge estimate. Using the estimate $\hat{j}_{kw}$ we can obtain pseudo-samples $s^* = (z_1^*, \dots, z_N^*)$, where $(z_1^*, \dots, z_{\hat{j}_{kw}}^*)$ is obtained by randomly sampling with replacement from $(z_1, \dots, z_{\hat{j}_{kw}})$. Similarly, $(z_{\hat{j}_{kw}+1}^*, \dots, z_N^*)$ is obtained by random sampling with replacement from $(z_{\hat{j}_{kw}+1}, \dots, z_N)$. Let $\hat{j}^*$ be the edge estimate obtained by resampling a strip of pixels s^* . After executing this scheme B times, we obtain B edge estimates $(\hat{j}_1^*, \dots, \hat{j}_B^*)$. This subsampling process is known as nonparametric bootstrap. Let $\hat{j}_{(1)}^*, \dots, \hat{j}_{(B)}^*$ the ordered values of $\hat{j}_1^*, \dots, \hat{j}_B^*$. The empirical distribution of $\hat{j}_{KW}$ is given by

$$\hat{F}_B^*(j') = \frac{\#\{\hat{j}_{(b)}^* \leq j'\}}{B}, \quad b \in \{1, \dots, B\}. \quad (10)$$

It can be used for constructing confidence intervals for the true edge location.

Assuming that $\hat{j}_{KW}$ is a consistent estimator of the edge and given that the true distribution of $\hat{j}_{KW}$ is unknown, one can construct an edge confidence interval based on the bootstrap approximation to the distribution of $\hat{j}_{KW} - j$. Here, the quantiles are obtained

by using the ordered values of $\hat{j}_{KW}^* - \hat{j}_{kw}$. The $100(1 - a)\%$ basic bootstrap method (BBM) confidence interval [20] is

$$\begin{aligned} & [\hat{j}_{a/2}^*, \hat{j}_{1-a/2}^*]_{BBM} = \\ & [\hat{j}_{kw} - (\hat{j}_{(B(1-a/2))}^* - \hat{j}_{kw}), \hat{j}_{kw} - (\hat{j}_{(B(a/2))}^* - \hat{j}_{kw})]. \end{aligned} \quad (11)$$

An alternative bootstrap confidence interval is the Percentile interval. The $100(1 - a)\%$ percentile interval [20] is given by

$$[\hat{j}_{a/2}^*, \hat{j}_{1-a/2}^*]_{PERC} = [\hat{j}_{(B(a/2))}^*, \hat{j}_{(B(1-a/2))}^*]. \quad (12)$$

It is noteworthy that this interval may be asymmetric and only contains proper values of the edge.

An alternative approach is to construct a bootstrap studentized confidence interval: the percentile- t or bootstrap- t interval. To that end, in each bootstrap replication we compute the quantity

$$Z_b^* = \frac{\hat{j}_b^* - \hat{j}_{kw}}{\hat{v}_b^{*1/2}}, \quad (13)$$

where $\hat{v}_b^{*1/2}$ is the standard error $\hat{j}^*$ obtained in the b th bootstrap replication. The a th sample quantile, $\hat{t}_{(a)}$, is obtained as follows:

$$\frac{\#\{Z_b^* \leq \hat{t}_{(a)}\}}{B} = a,$$

$b \in \{1, \dots, B\}$. The $100(1 - a)\%$ studentized bootstrap confidence interval [20] is then given by

$$[\hat{j}_{a/2}^*, \hat{j}_{1-a/2}^*]_{ST} = [\hat{j}_{kw} - \hat{v}^{1/2} \hat{t}_{(B(1-a/2))}, \hat{j}_{kw} - \hat{v}^{1/2} \hat{t}_{(B(a/2))}]. \quad (14)$$

The sample variance ν_b^* can be computed by using a sub-bootstrap (i.e., a second level bootstrap) with B' replications, where $B' = 50$ is the minimum number of resamples that yields reasonable estimates. Efron and Tibshirani [21] recommend using $B' = 200$. However, they also recommend using $B = 1000$. This implies that estimation of ν_b^* using a second level bootstrap adds considerable computational burden, even when one uses $B' = 50$.

In what follows we shall propose two alternative approximate bootstrap- t interval estimators, denoted ST1 and ST2. By using them one can compute ν_b^* without having to resort to a second level bootstrap.

The idea behind our ST1 bootstrap method is to estimate ν_b^* using a subset of size B' from the replications $\hat{j}_1^*, \dots, \hat{j}_B^*$ imposing the restriction that the sample variance is positive. Let $\hat{j}^{*'} = \{\hat{j}_1^{*'}, \dots, \hat{j}_{B'}^{*'}\}$ be the randomly selected subset of B bootstrap estimates. Our algorithm can be outlined as follows:

It is noteworthy that Algorithm 1 must be executed for each $b \in \{1, \dots, B\}$. This algorithm is considerably less computationally intensive than the standard bootstrap- t algorithm in which a second level bootstrapping scheme is used for variance estimation.

At Algorithm 2 is introduced a second approximation to the bootstrap- t method.

Notice that ST1 is less computationally expensive than the ST2 and that they are both less computationally costly than the standard bootstrap- t , in which a second level (inner) bootstrap is carried out for variance estimation.

Algorithm 1 ST1 Method

```
1: if  $\hat{j}_b^* = \hat{j}_{kw}$  then
2:    $Z_b^* \leftarrow 0$ 
3: else
4:   if  $\text{Var}(\hat{j}_1^*, \dots, \hat{j}_B^*) \neq 0$  then
5:     Obtain  $\hat{j}_*'^* = \{\hat{j}_1^{*'}, \dots, \hat{j}_{B'}^{*'}\}$  by drawing with replacement from  $\{\hat{j}_1^*, \dots, \hat{j}_B^*\}$ .
6:      $\hat{v}_b^* \leftarrow \text{Var}(\hat{j}_*'^*)$ 
7:     if  $\hat{v}_b^* = 0$  then
8:       Go back to line 5 (do this at most  $B'' > 0$  times).
9:     end if
10:    if line 8 was repeated  $B''$  times then
11:       $\hat{v}_b^* \leftarrow \text{Var}(\hat{j}_1^*, \dots, \hat{j}_B^*)$ 
12:    end if
13:  else
14:    repeat
15:      Obtain  $\hat{j}_*'^*$  by estimating the edge using a new bootstrap replication of pixels
      strip  $s$ .
16:    until  $\hat{j}_*'^* \neq \hat{j}_1^*$  (do this at most  $B'' > 0$  times)
17:    if  $\hat{j}_*'^* = \hat{j}_1^*$  then
18:       $\hat{j}_*'^* \leftarrow \hat{j}_1^* + 1$ 
19:       $\hat{v}_b^* \leftarrow \text{Var}(\hat{j}_1^*, \dots, \hat{j}_{B'-1}^*, \hat{j}_*'^*)$ 
20:    end if
21:  end if
22: end if
```

5 Numerical results and discussion

In what follows we shall report the results of several Monte Carlo simulations which were performed to assess the finite sample merits of different edge interval estimates in SAR images. All simulations were carried out using Ox matrix programming language [30]. The hardware used was an Intel(R) Core(TM)2 Quad CPU Q6600 2.40GHz computer running on Ubuntu Linux. Graphics were produced using R programming environment [31].

The simulated data have been generated according to the statistical distribution $\mathcal{G}_{\mathcal{J}}^0$ given in (1) and random number generation can be easily performed. It suffices to use the fact if $W \sim \Gamma^{-1}(k, \theta)$ then $1/W \sim \Gamma(k, 1/\theta)$. Thus, in order to generate a pseudo-random number from $\mathcal{G}_{\mathcal{J}}^0$ we only need to generate a pseudo-random number from $\Gamma(L, L)$ and then divide it by a pseudo-random number obtained from $\Gamma(-\alpha, 1/\gamma)$. For instance, the following Ox [30] function can be used for $\mathcal{G}_{\mathcal{J}}^0$ random number generation:

```
// A - texture parameter
// G - scale parameter
g0i_generation (const A, const G)
{
  decl x = rangamma(rowss, cols, -A, 1/G);
  decl y = rangamma(rowss, cols, L, L);
  return y./x;
}
```

Algorithm 2 ST2 Method

```
1: Obtain estimates  $\hat{j}_x^* \leftarrow \{\hat{j}_{B+1}^*, \dots, \hat{j}_{B+B_x}^*\}$  of the edge based on  $B_x$  new resamples from the
   pixels strip  $s$ .
2: if  $\text{Var}(\hat{j}_x^*) \neq 0$  then
3:   Obtain  $\hat{j}^{*'} = \{\hat{j}_1^{*'}, \dots, \hat{j}_{B'}^{*'}\}$  ( $B' < B_x$ ) by drawing with replacement from  $\hat{j}_x^*$ .
4:    $\hat{v}_b^* \leftarrow \text{Var}(\hat{j}^{*'})$ .
5:   if  $\hat{v}_b^* = 0$  then
6:     Go back to line 3. This step is to be executed at most  $B''$  times ( $0 < B'' < B_x$ ).
7:   end if
8:   if line 6 was repeated  $B''$  times then
9:      $\hat{v}_b^* \leftarrow \text{Var}(\hat{j}_x^*)$ .
10:  end if
11: else
12:  repeat
13:    Obtain  $\hat{j}_*^{*'}$  by estimating the edge using a new bootstrap replication of pixels strips
     $s$ .
14:  until  $\hat{j}_*^{*'} \neq \hat{j}_{B+1}^*$  (do this at most  $B'' > 0$  times)
15:  if  $\hat{j}_*^{*'} = \hat{j}_{B+1}^*$  then
16:     $\hat{j}_*^{*'} \leftarrow \hat{j}_{B+1}^* + 1$ 
17:  end if
18:   $\hat{v}_b^* \leftarrow \text{Var}(\{\hat{j}_{B+1}^*, \dots, \hat{j}_{B+B'-1}^*, \hat{j}_*^{*'}\})$ 
19: end if
```

In all simulations, $L = 1$. Recall that this is the most challenging situation. All images we generated are rectangular and contain 20×100 pixels ($N = 100$) with an edge located at $j = 50$. Thus, each image contains two distinct regions of equal size. In the region to the left of the edge $\alpha = \alpha_\ell$, and otherwise $\alpha = \alpha_r$, with $\alpha_\ell, \alpha_r \in \{-2, -3, \dots, -15\}$. For each region, the scale parameter γ was chosen according to (3). The most challenging situation as far as edge detection is concerned takes place when $|\alpha_r - \alpha_\ell| = 1$; the higher the absolute difference between α_r and α_ℓ , the easier it is to locate the edge. We notice that the case in which $|\alpha_r - \alpha_\ell| = 1$ becomes particularly more challenging when the values α_r and α_ℓ are large (in absolute values) because we are then dealing with two very homogeneous regions. Accordingly, [32] shows that when $\alpha_\ell, \alpha_r \rightarrow -\infty$ the stochastic distance between the two $\mathcal{G}_{\mathcal{G}}^0$ diminishes, even when $|\alpha_r - \alpha_\ell|$ remains constant.

The number of Monte Carlo replications is $R = 5000$ and the number of bootstrap replications is $B = 1000$. For each configuration, we constructed 5000 confidence intervals using the basic bootstrap method, the percentile method, and also the ST1 and ST2 methods. The nominal coverage is $1 - \alpha = 0.95$ (95%). The results for the ST1 and ST2 methods were obtained using $B' = 50$, and $B'' = 200$, and for ST2 we used $B_x = 200$.

We computed the exact coverage ($\bar{C}$) of each interval estimator corresponding to the 95% nominal level, which was done for each configuration (α_ℓ, α_r) . The best performing estimator is the one whose exact coverage is closest to the nominal coverage. Figure 2 plots the distance $D = |\bar{C} - 0.95| \times 100\%$ against α_r (the roughness parameter value to the right of the edge). The six panels in the figure correspond to $\alpha_\ell \in \{-2, -5, -8, -11, -13, -15\}$. Situa-

tions in which $\alpha_\ell = \alpha_r$ (i.e., there is no edge) are not considered.

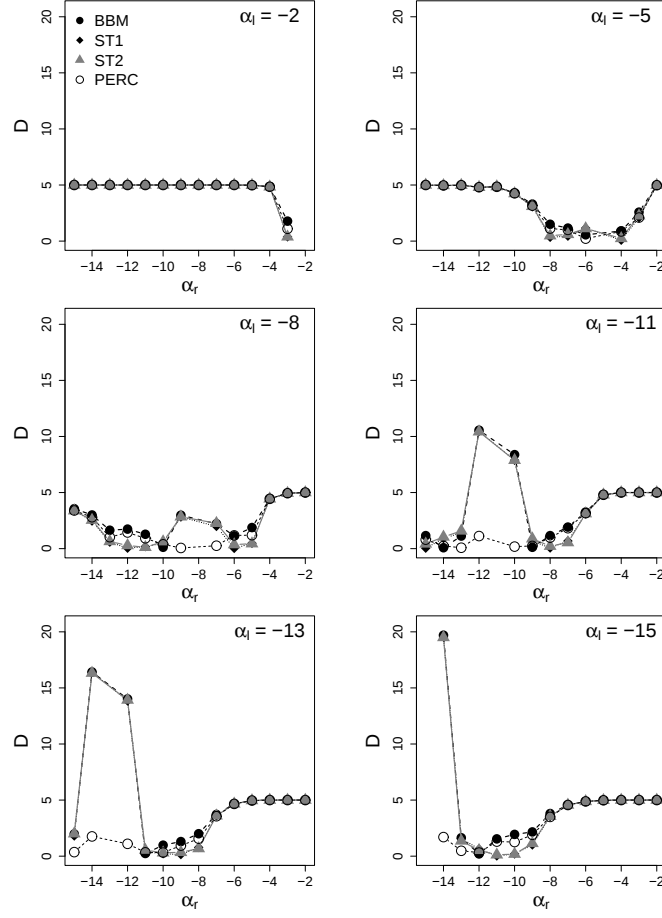


Figure 2: Distances between empirical and nominal coverages.

Visual inspection of Figure 2 reveals that all interval estimators perform equally well when $\alpha_\ell = -2$ and $\alpha_r \leq -4$; their coverages are 100%. We also notice that as $|\alpha_\ell - \alpha_r|$ increases the differences between the different coverages tend to decrease, i.e., the different intervals display similar coverages. That happens because the distribution of $\hat{j}_{KW}$ tends to be more concentrated around of the true value $j = 50$ as $|\alpha_\ell - \alpha_r|$ increases, i.e., the estimator $\hat{j}_{KW}$ is more accurate when the two regions have very distinct textures. Such a tendency is not, however, uniform. Also, when the regions on both sides of the edge are very heterogeneous (for example, $\alpha_\ell = -2$, and $\alpha_r = -3$, or $\alpha_\ell = -5$ and $\alpha_r = -4$), the different intervals behave similarly, but when the two regions are very homogeneous (for example $\alpha_\ell = -13$, and $\alpha_r = -12$), the percentile estimator outperforms all other interval estimators. Figure 2 shows that the percentile estimator is the best performing interval estimator in most situations. It is also noteworthy that the percentile estimator is the best performing estimator in the most challenging situations, i.e., whenever the two regions are very homogeneous and have similar textures.

In Figure 3, a symbol indicates which is the best performing interval estimator for each configuration (α_ℓ, α_r) ; the quantity below the symbol is

$$\Delta = D_{\text{percentile}} - D_{\text{best method}},$$

which measures the method performance relative to the percentile method. The symbols used in the figure are the same as the ones used in Figure 2, except for the symbol used to represent BBM, which is now $\otimes$. The symbol $\boxtimes$ is used whenever all methods produced the same coverage. We observe that in configurations in which the two absolute values of roughness parameter are large and similar (the most challenging situations), the percentile estimator is the best performer, closely followed by BBM.

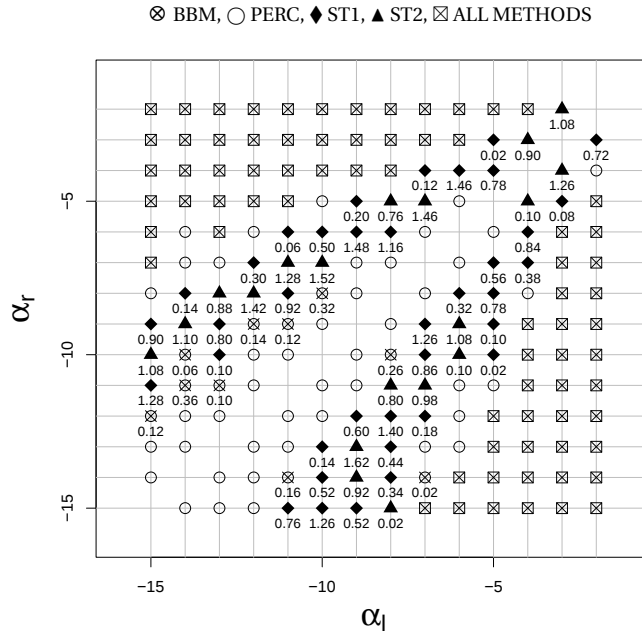


Figure 3: Best method and difference performance Δ between the percentile and the best performer.

The two studentized estimators, ST1 and ST2, outperformed the competition whenever $|\alpha_\ell - \alpha_r| \leq 5$. In these situations, the values of Δ are between 1% and 2%. Since these two estimators behave similarly, we recommend the use of ST1, which is less computationally intensive. It should be noted that in such situations the percentile estimator is quite competitive with the studentized estimators.

All methods behave similarly when the absolute difference between α_ℓ and α_r becomes very large; see Figure 2.

When taken together, our numerical results show that the percentile estimator is either the best performing estimator or quite competitive with the best performer. Therefore, we conclude that the percentile edge estimator is to be preferred.

We have also carried out simulations in which there is no edge, i.e., $\alpha_\ell = \alpha_r$. Here, we shall focus not on coverages but on the average lengths of the different confidence intervals.

The estimated edge locations are expected to be randomly and uniformly distributed along the detection line since there is no edge. We thus expect the average interval length to be approximately equal to the detection line length. Figure 4 displays average interval lengths (AIL).

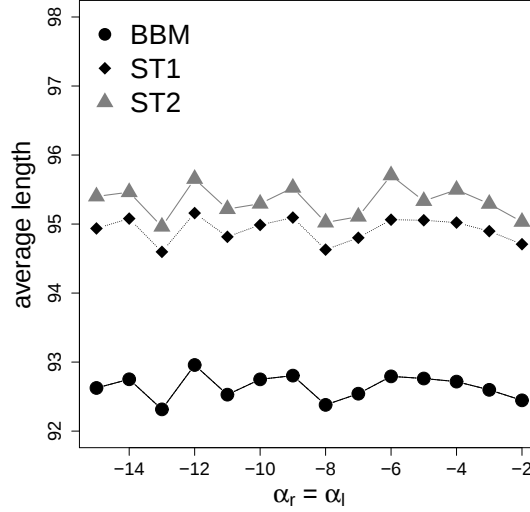


Figure 4: Average length for situations where there is not edge.

We notice that all methods produce similar average interval lengths, regardless of the configuration (α_ℓ, α_r) , as expected. The percentile and BBM interval estimators have the same average length: approximately 93 pixels. Notice that an open circle is used to represent the percentile method and a filled circle is used for the BBM method, therefore, only the filled circle (BBM) is visible. The ST1 and ST2 estimators displayed similar average lengths (approximately 95 pixels), the former always being slightly narrower. Overall, we notice that all methods yield very wide intervals when there is no edge to be located. A very wide confidence interval can then be taken as evidence that there is no edge.

Table 1 displays selected average interval lengths. Notice that when $|\alpha_\ell - \alpha_r|$ is large, the average lengths are null or very small. On the other hand, when $|\alpha_\ell - \alpha_r|$ is small, the average lengths tend to be large. percentile and BBM intervals yields the smallest average lengths whenever the average lengths are not all the same. The largest average lengths correspond to the studentized interval estimators, ST1 and ST2. The average lengths of the studentized and percentile intervals are very close.

We have also considered the case in which the edge lies not in the middle of the detection window. In particular, we carried out simulations using $j = 20$. The number of looks ($L = 1$) and the detection window size (20×100 pixels) were kept constant. The results corresponding to low contrast situations, which are the most challenging cases, are presented in Table 2. Such results are in agreement with those previously reported; see Table 1 and Figure 2.

The execution times of the different interval edge estimators for each Monte Carlo replication are given in Table 3. Since the computational cost does not depend on the parameter configuration, we use $(\alpha_\ell, \alpha_r) = (-2, -3)$ in order to measure the different execution times.

Table 1: Average interval lengths (AIL) for $\alpha_\ell \neq \alpha_r$ (selected configurations).

α_r	$\alpha_\ell = -3$			$\alpha_\ell = -8$			$\alpha_\ell = -13$		
	BBM	ST1	ST2	BBM	ST1	ST2	BBM	ST1	ST2
-2	1.04	1.07	1.10	0.00	0.00	0.00	0.00	0.00	0.00
-3				0.00	0.00	0.00	0.00	0.00	0.00
-4	3.40	3.77	3.84	0.05	0.04	0.04	0.00	0.00	0.00
-5	0.50	0.47	0.49	0.99	1.02	1.05	0.00	0.00	0.00
-6	0.04	0.03	0.03	3.96	4.35	4.40	0.02	0.02	0.02
-7	0.01	0.00	0.00	24.39	25.90	26.16	0.20	0.18	0.19
-8	0.00	0.00	0.00				0.96	0.99	1.03
-9	0.00	0.00	0.00	32.57	34.23	34.65	2.29	2.59	2.63
-10	0.00	0.00	0.00	7.28	7.89	8.00	5.09	5.54	5.62
-11	0.00	0.00	0.00	3.21	3.55	3.60	14.59	15.72	15.91
-12	0.00	0.00	0.00	1.79	2.00	2.05	62.06	63.61	64.14
-13	0.00	0.00	0.00	0.97	1.00	1.02			
-14	0.00	0.00	0.00	0.45	0.43	0.45	67.00	68.57	69.14
-15	0.00	0.00	0.00	0.19	0.17	0.18	21.58	23.01	23.34

Table 2: Edge lies not in the middle of the detection window – Percentile Method (selected configurations).

Texture	α_ℓ	α_r	AIL	coverage (%)
Extremely Heterogeneous	-2	-3	1.18	96.40
	-2	-4	0.04	99.68
Heterogeneous	-7	-8	33.70	94.74
	-7	-9	6.00	94.90
Homogeneous	-13	-15	29.40	94.16
	-14	-15	78.76	96.22

We report both the total time (in seconds) it takes to compute the interval estimator and that relative to the bootstrap- t estimator (%). The bootstrap- t estimator was computed using 50 replications in the inner bootstrap, which is carried out for variance estimation. The results show that the Bootstrap- t is approximately 50 times more computationally intensive than all competing methods. Percentile and BBM methods have nearly the same computational cost. ST1 and ST2 are slightly more costly, ST1 being less computationally intensive than ST2. It should be noted that the percentile method is not only the best performer as far as coverage is concerned, but it is also the least computationally intensive method.

Table 3: Computational cost and computational cost relative to bootstrap- t .

Method	Execution time (s)	Relative to bootstrap- t (%)
Bootstrap- t	81.422	100.00
BBM	1.578	1.94
ST1	1.630	2.00
ST2	2.191	2.69
PERC	1.565	1.92

The sample size affects the point estimates accuracy. Interval estimates constructed using smaller samples tend to be wider, thus reflecting the increased uncertainty. We observed such a behavior in Monte Carlo simulations (results not included here for brevity).

6 An application

In this section we apply the different edge interval estimators to a real image, i.e., to observed (not simulated) data. An E-SAR single look image of Weiß ling (Bayern, Germany) was used [33]. This image is displayed in Figure 5.

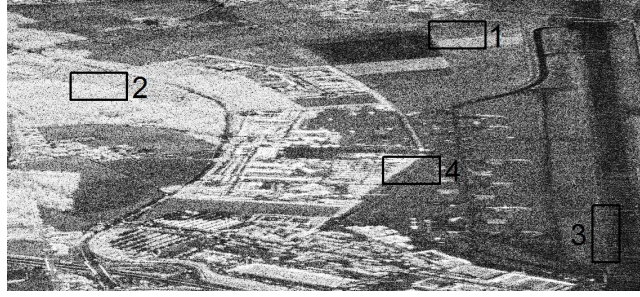


Figure 5: E-SAR image with 1100×2400 pixels, and delimited regions.

It was obtained in L band, and it exhibits part of an airport, urban areas, and pastures. The rectangles in the image are the regions chosen for edge detection. Rectangles (regions) 1, 2 and 4 have 210×100 pixels whereas rectangle (region) 3 has 100×210 pixels. In the edge detection process, we divided rectangle 3 in ten horizontal windows, each of 21×100 pixels, the detection line being horizontal for each window and dividing the window in equal parts of 10×100 pixels. Similarly, each of the other rectangles was divided in ten windows of 100×21 pixels, the detection line being vertical and dividing the window in equal parts of 100×10 pixels. Therefore, for each rectangle we compute ten interval edge estimates. In the figures that follow, we use green lines to connect the lower interval limits and also the upper interval limits; red lines are used to connect the different edge point estimates. Each detection window is delimited by yellow lines, the windows being numbered from left to right or from top to bottom (for example, the first window in rectangle 1 corresponds to the window located in the extreme left).

Figure 6 contains results relative to edge detection in rectangle 1.

We observe that the confidence intervals are quite narrow (the smallest interval length equals 3 pixels), except for the first two detection windows. In the second window there are less pixels in light regions than in dark regions, in which there is more noise, hence the wider confidence interval. This happens because edge detection becomes quite challenging under complex, rich textures. We also note that the estimated edge is typically located in areas in which the differences in texture are largest; see, e.g., the first window.

A challenging situation for edge detection can be seen in rectangle (region) 2 of Figure 5. There we notice two fairly different textures, but it is not easy to identify where exactly lies the frontier. The results of edge detection for this region are presents in Figure 7. Note that, in all detection windows, the confidence intervals are fairly wide; the smallest interval length is 5 pixels (last detection window) and the widest interval covers 26 pixels (next to last detection window). In challenging situations such as this interval estimation becomes quite useful since it signals that point edge detection may not be reliable.

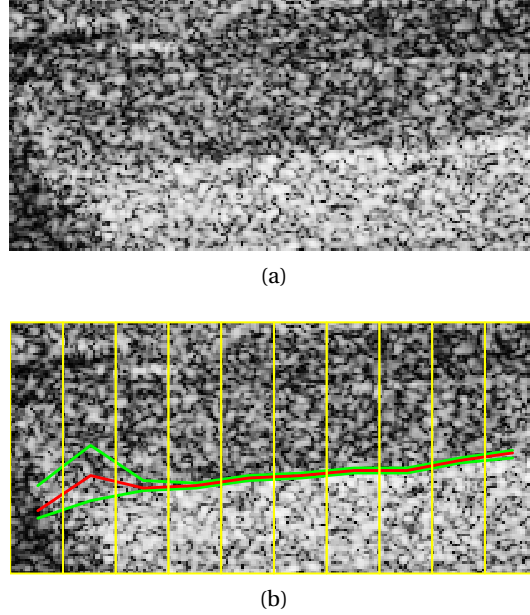


Figure 6: Results for interval detection in region 1. (a) Region 1 magnification. (b) Graphical presentation of confidence intervals to edges in region 1. Green lines links interval limits for each detection window. The windows are drawn by yellow lines. The red line links edge point estimates for each detection window.

Edge detection in rectangle (region) 3 of Figure 5 is even more challenging than in rectangle (region) 2 because the edge divides the image in two regions with textures that are visually very diffuse. The detection results are displayed in Figure 8.

We note that the interval lengths are nearly the same in all detection windows. None of interval lengths exceeds 11 pixels. We conclude that, despite the aforementioned difficulty, the KW detector yields accurate edge estimates.

A limitation of the point edge detection methods we consider is that an edge will be always located even when there is none. This happens because T_{KW} in (8) always achieves a maximum value. Interval estimation can, however, be used to assess whether an edge indeed exists. Such a situation is explored in rectangle (region) 4 of Figure 5. A confidence interval that covers 94 pixels is obtained, thus suggesting that there is no edge. The results of edge detection for this region are presents in Figure 9. In the last window detection, we obtain a fairly wide interval (41 pixels) which can be taken as evidence that the point edge estimate is not accurate. A similar situation takes place at the first detection window. Here, however, the dark regions are most likely not the result of noise but reflect the terrain characteristics.

Samples 2 and 3 are particularly interesting since they involve neighboring regions that are very similar. The edge is thus difficult to be located and the lengths of the corresponding interval estimates signal such uncertainty. They are wider in areas where the contrast between the two regions is small.

In Figures 6, 7, 8 and 9 we connected the different confidence interval limits using

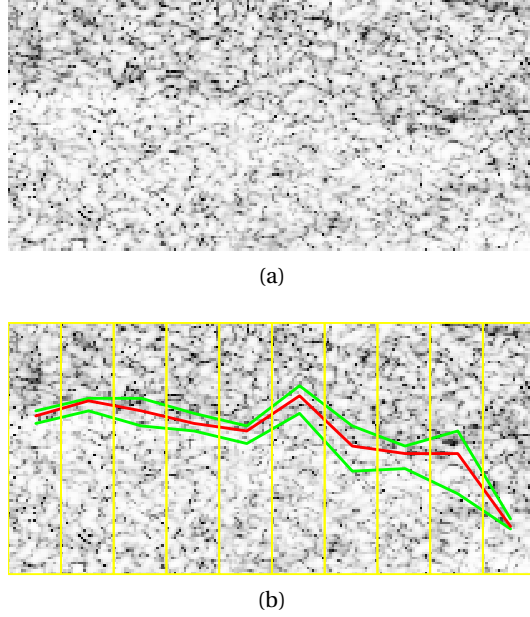


Figure 7: Results for interval detection in region 2. (a) Region 2 magnification. (b) Graphical presentation of confidence intervals to edges in region 2. Green lines links interval limits for each detection window. The windows are drawn by yellow lines. The red line links edge point estimates for each detection window.

straight lines. As noted by a referee, when using interval edge detection, the user can specify the kind of edge she is willing to retrieve. For instance, provided a wide enough interval, she can choose between linear features and straight angles between them, as is the case of many agricultural fields, or smoothly varying edges, as in shores. This is, we believe, a novel feature of our approach. In particular, we note that smoothing methods, such as kernel and splines smoothing [34–36], can be used for constructing the lower and upper curves. This was done in Figure 10. We note that the user can choose the amount of smoothing she wants to use.

In Figure 11 we compare the interval estimation approach proposed in this paper at the 95% confidence level (panels 11b, 11f and 11j) to the *gPb* estimator proposed in [38] (panels 11c, 11g and 11k) and also to Sobel’s [39] approach after applying the speckle-reducing filter proposed by Lee [40] (panels 11d, 11h and 11l). Our goal is to verify whether the different methods are able to locate an edge that separates two regions of similar textures. We thus used simulated images. This was done so that we would have control over the true parameters. The edge is located between regions of equal sizes and textures are (1) extremely heterogeneous, $\alpha = -2$ and $\alpha = -3$ (panel 11a), (2) heterogeneous, $\alpha = -7$ and $\alpha = -8$ (panel 11e) and (3) homogeneous, $\alpha = -14$ and $\alpha = -15$ (panel 11i). Here, $L = 1$ and $\gamma = -\alpha - 1$. Each image has 200×400 pixels. In order to perform interval estimation, we divided each image into 10 detection windows, each containing 20×400 pixels. In all situations, the point KW detector (red line) was able to locate the edge with high precision, if not perfectly (see panels 11b, 11f and 11j). The interval lengths indicate that such detection

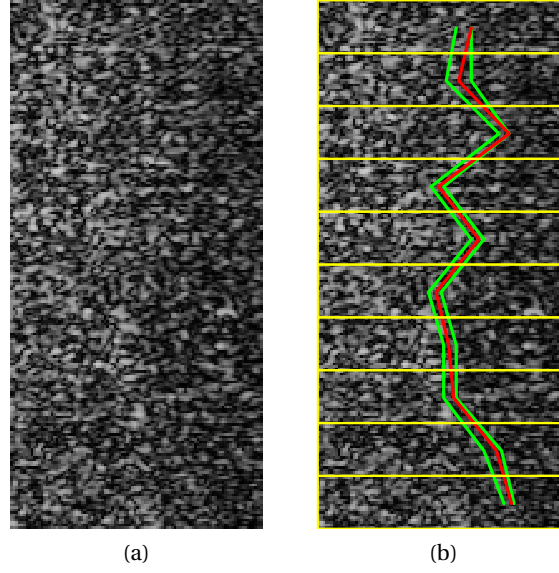


Figure 8: Results for interval detection in region 3. (a) Region 3 magnification. (b) Graphical presentation of confidence intervals to edges in region 3. Green lines links interval limits for each detection window. The windows are drawn by yellow lines. The red line links edge point estimates for each detection window.

is reliable when the regions the regions are extremely heterogeneous or homogenous. The confidence intervals obtained for heterogeneous regions are, however, wider and thus indicate that point edge estimation based on the KW detector may not be as reliable. The *gPb* detector yields a hierarchical representation for the edges in the image. Such representation is called Ultrametric Contour Map (UCM). The UCM at level k yields a set of curves that are the segmentation edges at scale k . The scale is related to the probability $P(x)$ that a given pixel x lies on the edge. The white lines in panels 11c, 11g and 11k are the pixels for which $P(x) \geq k$; here, $k = 0.07$. In all cases, no line emerged when we used $P(x) \geq 0.09$. Hence, each pixel of all images lies on an edge with very low probability according to the *gPb* detector. Additionally, none of the white lines fully agree with the true edge. Likewise, Sobel's method was not able to locate the edges; see panels 11d, 11h and 11l. We obtained results that are similar to those obtained using Sobel's filter (not presented here) by using the gradient filter [41] and also the method introduced by Touzi [9]. The *gPb* detector was computed using the Matlab code available at <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>. The Lee filter was computed using the code available at Matlab Exchange (<http://www.mathworks.com/matlabcentral/fileexchange>). Finally, other threshold based edges were computed using the Monteverdi-Orfeo Toolbox (<http://www.orfeo-toolbox.org/otb/monteverdi.html>).

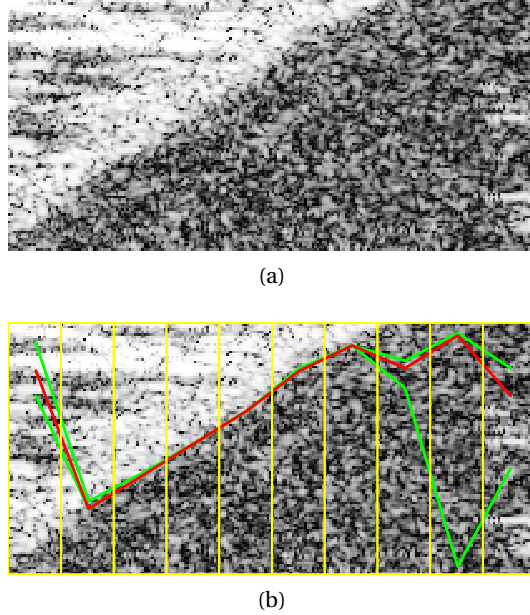


Figure 9: Results for interval detection in region 4. (a) Region 4 magnification. (b) Graphical presentation of confidence intervals to edges in region 4. Green lines links interval limits for each detection window. The windows are drawn by yellow lines. The red line links edge point estimates for each detection window.

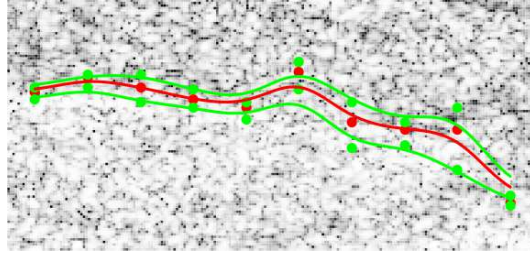
7 Concluding remarks

We addressed the issue of edge detection in SAR images. As pointed out in Section 1, the distributions of point edge estimators are usually unknown and can be estimated using data resampling. We used the bootstrap method to estimate the KW detector distribution. This allowed us to obtain confidence intervals for the edge location. Several bootstrap-based interval estimates were described and numerically evaluated. We addressed situations in which there is an edge and also situations in which there is none. The case of multiple edges will be addressed in future research.

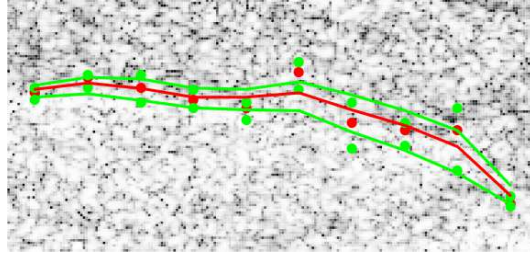
Overall, the percentile confidence interval proved to be most reliable, especially in the challenging situation in which the regions on both sides of the edge have similar textures. The percentile interval typically displayed the best coverage but it is typically wider than alternative intervals. There is thus a trade-off between coverage and length. Percentile and basic bootstrap method delivered intervals with the smallest lengths.

We proposed two variants of the bootstrap- t method for edge interval estimation, denoted ST1 and ST2. They are less computationally costly than the standard Bootstrap- t method. They tend to work equally well, ST1 being more computationally efficient.

We have considered situations in which edge detection is carried out in an image region in which there is no edge. Point estimation will always locate an edge, even when there is none. In such situations, the resulting interval estimates tend to be fairly wide, thus signaling that the detected edge is not to be trusted. Very wide intervals can be taken as an



(a) Gaussian kernel smoothing with bandwidth equal to 23.



(b) Cubic Splines smoothing with degrees of freedom equal to 8.

Figure 10: Smooth confidence bands, region 2. Point estimates are denoted by red dots and the different confidence intervals are marked using green dots.

indication that most likely there is no edge in that region of the SAR image and we can use this measure in an unsupervised system of edge detection. Real (not simulated) data were analyzed. We performed edge detection in several regions of a SAR image.

It is important to remark that the edge interval estimators presented in this paper can be used with other types of image. Edge location in SAR images is particularly challenging due to the existence of speckled noise. Such noise can make the usual detection strategies to take noise for edges. Interval estimation can add useful information to the task of locating edges in such images. It can even signal that there is no edge in the region of the image under scrutiny.

An interesting extension of the interval estimation methodologies considered in this paper involves edge detection in polarimetric SAR images, which are complex-valued. That would add important information to the edge point estimates that were proposed in the literature; see, e.g., [37] and [7]. We shall address that in future research.

Acknowledgements

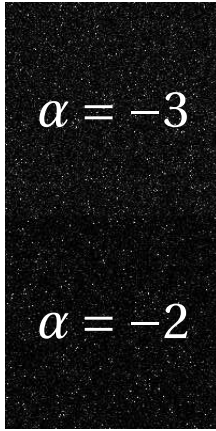
We gratefully acknowledge partial financial support from CAPES and CNPq. We thank Alejandro C. Frery for providing us with the data analyzed in Section 6. Finally, we thank three anonymous referees for their thoughtful comments, suggestions and criticisms.

References

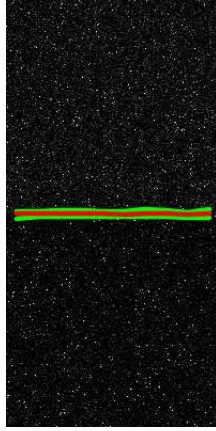
- [1] C. Oliver and S. Quegan *Understanding Synthetic Aperture Radar Images*. Artech House, Boston, 2004.
- [2] A. D. C. Nascimento, M. M. Horta, A. C. Frery, and R. J. Cintra. Comparing Edge Detection Methods Based on Stochastic Entropies and Distances for PolSAR Imagery. *Accepted for publication in IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing*.
- [3] F. Baselice, and G. Ferraioli. Statistical Edge Detection in Urban Areas Exploiting SAR Complex Data. *IEEE Geoscience And Remote Sensing Letters*, 9(2):185–189, 2012.
- [4] X. Fu, H. You, and K. Fu. A Statistical Approach to Detect Edges in SAR Images Based on Square Successive Difference of Averages. *IEEE Geoscience And Remote Sensing Letters*, 9(6):1094–1098, 2012.
- [5] A. Singhal, and M. Singh. Speckle Noise Removal and Edge Detection Using Mathematical Morphology. *International Journal of Soft Computing and Engineering*, 1(5):146–149, 2011.
- [6] M. T. Alonso, C. López-Martínez, J. J. Mallorquí, and P. Salembier. Edge Enhancement Algorithm Based on the Wavelet Transform for Automatic Edge Detection in SAR Images. *IEEE Transactions On Geoscience And Remote Sensing*, 49(1):222–235, 2011.
- [7] A. C. Frery, J. Jacobo-Berlles, J. Gambini, and M. Mejail. Polarimetric SAR image segmentation with B-splines and a new statistical model. *Multidimensional Systems and Signal Processing*, 21:319–342, 2010.
- [8] C. J. Oliver, D. Blacknell, and R. G. White. Optimum edge detection in SAR. *IEE Proceedings Radar, Sonar and Navigation*, 143(1):31–40, 1996.
- [9] R. Touzi, A. Lopes, and P. Bousquet, A statistical and geometrical edge detector for SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 26:764–773, 1988
- [10] R. Fjortoft, A. Lopes, P. Marthon, and E. Cubero-Castan. An optimal multiedge detector for SAR image segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 36:793–802, 1998
- [11] I. Melgar, J. Fombellida, J. Gomez, and J. Seijas, Edge detection algorithms in intensity SAR images: Empirical comparison of performances and proposed improvements. *In Proc. WAC 2008 World. Automation Congress*, 2:1–7 Sept. 28 2008-Oct, 2008
- [12] F. Baselice, G. Ferraioli, and D. Reale, Edge Detection Using Real and Imaginary Decomposition of SAR Data. *IEEE Transactions on Geoscience and Remote Sensing*, press:, 2013
- [13] Q-R. Wei; D.-Z. Feng; M-D. Yuan, Automatic local thresholding algorithm for SAR image edge detection. Radar Conference 2013, IET International 1–5, 14-16 April 2013.

- [14] J. Gambini, M. E. Mejail, J. Jacobo-Berlles, and A. C. Frery. Feature extraction in speckled imagery using dynamic B-spline deformable contours under the G(0) model. *International Journal of Remote Sensing*, 27(22):5037–5059, 2006.
- [15] J. Gambini, M. E. Mejail, J. Jacobo-Berlles, and A. C. Frery. Accuracy of edge detection methods with local information in speckled imagery. *Statistics and Computing*, 18(1):15–26, 2008.
- [16] A. C. Bovik, T. S. Huang, and D. C. Munson. Nonparametric tests for edge-detection in noise. *Pattern Recognition*, 19(3):209–219, 1986.
- [17] M. Beauchemin, K. P. B. Thomson, and G. Edwards. On nonparametric edge detection in multilook SAR images. *IEEE Transactions on Geoscience and Remote Sensing*, 36(5):1826–1829, 1998.
- [18] D. H. Lim and S. J. Jang. Comparison of two-sample tests for edge detection in noisy images. *Journal of the Royal Statistical Society Series D*, 51:21–30, 2002.
- [19] E. Girón, A. C. Frery, and F. Cribari-Neto. Nonparametric edge detection in speckled imagery. *Mathematics and Computers in Simulation*, 82(11):2182–2198, 2012.
- [20] A. C. Davison, and D. V. Hinkley. *Bootstrap Methods and Their Application*. Cambridge University Press, 1997
- [21] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.
- [22] A. C. Frery, H. J. Muller, C. D. F. Yanasse, and S. J. S. Sant’Anna. A model for extremely heterogeneous clutter. *IEEE Transactions on Geoscience and Remote Sensing*, 35(3):648–659, MAY 1997.
- [23] O. H. Bustos, M. M. Lucini, and A. C. Frery. M-Estimators of Roughness and Scale for GA0-Modelled SAR Imagery. *EURASIP Journal on Advances in Signal Processing*, 1:105–114, 2002.
- [24] H. Allende, A. C. Frery, J. Galbiati, and L. Pizarro. M-Estimators with Asymmetric Influence Functions: the GA0 distribution case. *Journal of Statistical Computation and Simulation*, 76(11):941–956, 2006.
- [25] M. Silva, F. Cribari-Neto, and A. C. Frery. Improved likelihood inference for the roughness parameter of the GA0 distribution. *Environmetrics*, 19(4):347–368, 2008.
- [26] K. L. P. Vasconcellos, A. C. Frery, and L. B. Silva. Improving estimation in speckled imagery. *Computational Statistics*, 20:503–519, 2005.
- [27] F. Cribari-Neto, A. C. Frery, and M. F. Silva. Improved estimation of clutter properties in speckled imagery. *Computational Statistics & Data Analysis*, 40(4):801–824, 2002.
- [28] A. C. Frery, F. Cribari-Neto, and M. O. Souza. Analysis of minute features in speckled imagery with maximum likelihood estimation. *EURASIP Journal on Applied Signal Processing*, 16:2476–2491, 2004.

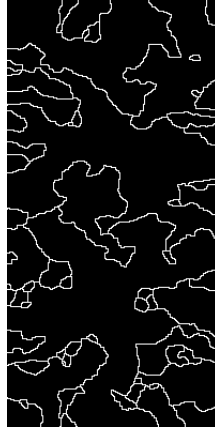
- [29] P. Good. *Permutation, Parametric, and Bootstrap Tests of Hypothesis*. Springer Series in Statistics. Springer-Verlag, 2005
- [30] J. A. Doornik. *Object-Oriented Matrix Programming Using Ox*. 2 Timberlake Consultants Press & Oxford, London, 4rd ed, URL <http://www.doornik.com>, 2006.
- [31] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Statistics and Computing, Springer, New York, 2002.
- [32] A. D. C. Nascimento, R. J. Cintra, and A. C. Frery. Hypothesis testing in speckled data with stochastic distances. *IEEE Transactions On Geoscience And Remote Sensing*, 48(1):373–385, 2010.
- [33] R. Horn. E-SAR: The experimental airborne L/C-band SAR system of DFVLR. *Proceedings IGARSS '98: IEEE International Geoscience and Remote Sensing Symposium*, 2:1025–1026, 1998.
- [34] M. G. Schimek. *Smoothing and Regression*. Wiley, 2000.
- [35] S. Efromovich. *Nonparametric Curve Estimation*. Springer-Verlag, 1999.
- [36] W. Härdle. *Smoothing Techniques with Implementation in S*. Springer-Verlag, 1991.
- [37] J. Schou, H. Skriver, A. A. Nielsen, and K. Conradsen. CFAR Edge Detector for Polarimetric SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 41:20–32, 2003
- [38] P. Arbelaez, M. Maire, C. Fowlkes and J. Malik, Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:898–916, 2011
- [39] W.K. Pratt *Digital Image Processing*. New York: Wiley, 1978.
- [40] J. S. Lee. Speckle suppression and analysis for synthetic aperture radar images. *Optical Engineering*, 25:636–643, 1986.
- [41] L. S. Davis. A survey of edge detection techniques. *Computer Graphics and Image Process*, 24:248–270, 1975.



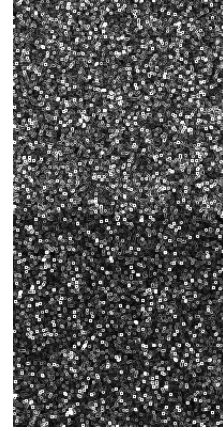
(a) Extremely Heterogeneous Regions.



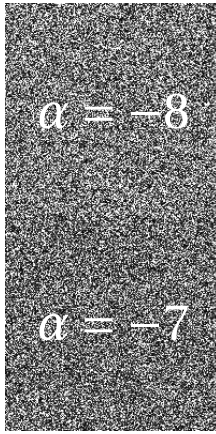
(b) Edge confidence interval for (a).



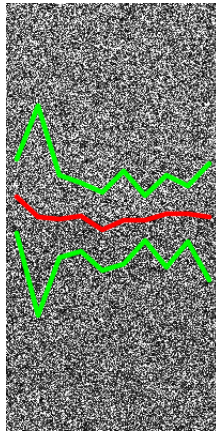
(c) *gPb* method applied to (a).



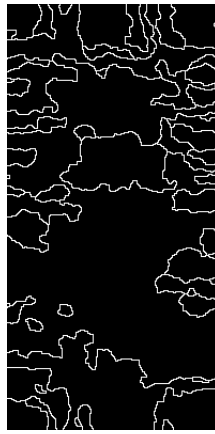
(d) Sobel method applied to (a).



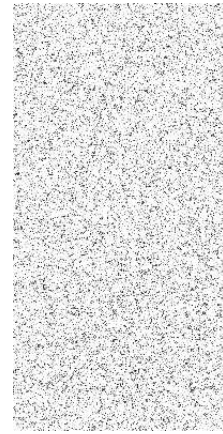
(e) Heterogeneous Regions.



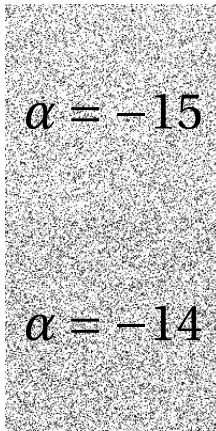
(f) Edge confidence interval for (e).



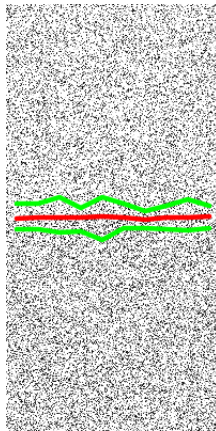
(g) *gPb* method applied to (e).



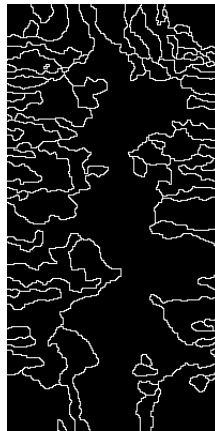
(h) Sobel method applied to (e).



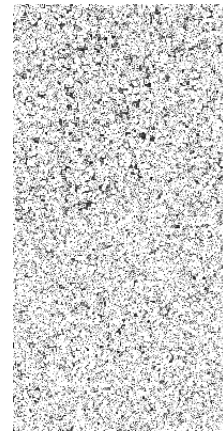
(i) Homogeneous Regions.



(j) Edge confidence interval for (i).



(k) *gPb* method applied to (i).



(l) Sobel method applied to (i).

Figure 11: Edge location comparison: the *gPb* method and the classical Sobel method coupled with the Lee filter.