Interval edge estimation in SAR images
Laércio Dias, Francisco Cribari-Neto, and Raydonal Ospina

Abstract
This paper considers edge interval estimation between two regions of a Synthetic Aperture Image which differ in texture. 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. We also analyze a real dataset.

Index Terms
Synthetic aperture radar (SAR), edge detection, confidence interval, bootstrap.

I. 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. 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. Much attention has been devoted to statistical inference using SAR data because the use of SAR sensors has become widespread. 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 three possible types of texture for each region of a SAR image, namely: homogeneous (i.e., surfaces with little texture such as lakes, deforestation areas, crop fields, deserts and snow covered areas), heterogeneous (i.e., surfaces with more texture than homogeneous areas, but with not as much texture as forests at regular relief) and extremely heterogeneous (i.e., areas that have very intense texture, such as urban areas).

Several approaches for detecting an edge between neighboring regions with different textures have been proposed in the literature. Recent development in edge detection and segmentation in speckled imagery can be found in [1], [2], [3], [4], [5], [6] and [7].

Gambini et al. [8] proposed a parametric point estimator (i.e., detector) for the edge location. It has been shown that their estimator typically outperforms alternative detectors [9], [10], [11] in a number of different settings [12]. Nonparametric detectors have also been proposed. Bovik et al. [13] used nonparametric methods to detect edges under additive Gaussian noise. Beauchemin et al. [14] used an alternative approach based on the Wilcoxon-Mann-Whitney statistic to detect changes in adjacent sets of pixels. Lim and Jang [15] used two sample tests to detect edges in images subject to noise. Girón et al. [16] 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.

All edge detection strategies that have been proposed in the literature yield edge point estimates. Our main goal in this 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.

More specifically, we wish to detect the position of the edge between two regions along a line segment (detection line). To that end we use information on the pixels that lie in a neighborhood of a rectangular shaped detection line (window). Figure 1 shows a simulated SAR image that contains two distinct regions and also a detection window divided by the detection line.

As pointed by Gambini et al. [8], their detector can deliver poor estimates when the detection line is not orthogonal to the boundary, as it is the case in Figure 1. In contrast, edge interval estimation does not rely on the orthogonality between the detection line and the boundary. This reinforces the importance of interval edge detection.

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 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 Tibshirani [17]. Our numerical results show that the bootstrap percentile estimator is the overall best performer in the sense that it typically has the best coverage amongst all estimators.
The remainder of the article unfolds as follows. Section III introduces the statistical model we shall use to model SAR data. Section IV presents the Gambini [8], [12] and Kruskal-Wallis detectors. Section V considers edge bootstrap interval estimation. Section VI presents Monte Carlo evidence. Section VII contains an application that uses real (not simulated) data. Finally, Section VII offers some concluding remarks.

II. 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.

Images represented in the intensity format can be described by the distribution $G_0^I$, denoted by $Z \sim G_0^I$. 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 incoming and reflected signals, and (iii) the roughness parameter $\alpha < 0$, which relates to the land type texture. The larger the value of $\alpha$, 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.

According to the model proposed by Frery et al. [18], the speckle noise is gamma distributed, denoted by $Y \sim \Gamma(L, L)$, its density being

$$f_Y(y) = \frac{L^L}{\Gamma(L)} y^{L-1} \exp(-L y), \quad L \geq 1, y > 0. \quad (1)$$

The backscatter can be modeled using the inverse gamma distribution, $X \sim \Gamma^{-1}(\alpha, \gamma)$, whose density function is

$$f_X(x) = \frac{1}{\gamma^\alpha \Gamma(-\alpha)} x^{-\alpha-1} \exp\left\{-\frac{\gamma}{x}\right\}, \quad -\alpha, \gamma, x > 0. \quad (2)$$

Thus, $Z = XY$ is distributed as $G_0^I$, $Z \sim G_0^I(\alpha, \gamma, L)$, whose density function is

$$f_Z(z) = \frac{L^L \Gamma(L-\alpha)}{\gamma^\alpha \Gamma(L) \Gamma(-\alpha)} \frac{z^{L-1}}{(\gamma + L z)^{L-\alpha}}, \quad L \geq 1, -\alpha, \gamma, z > 0. \quad (3)$$

Figure 2 shows the $G_0^I$ density for $L \in \{1, 3\}$ and $\alpha \in \{-2, -5, -15\}$.

It is noteworthy that $G_0^I$ 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 $G_0^I$ we only need to generate a pseudo-random number from $\Gamma(L, L)$ and then divide it by a pseudo-random number obtained from $\Gamma(-\alpha, 1/\gamma)$. 
Using (3), it can be easily shown that the \( r \)th noncentral moment is
\[
E[Z^r] = \left( \frac{\gamma}{L} \right)^r \frac{\Gamma(-\alpha - r) \Gamma(L + r)}{\Gamma(-\alpha) \Gamma(L)},
\]
if \( -\alpha > r \), and \( \infty \) otherwise. Notice that the value of \( \gamma \) that corresponds to \( E[Z] = 1 \) is
\[
\gamma = \frac{\Gamma(-\alpha) \Gamma(L)}{\Gamma(-\alpha - 1) \Gamma(L + 1)}.
\]

When \( \alpha < -1 \), parameter estimation can be performed using the method of moments. Let \((Z_1, \ldots, Z_n)\) be a vector of independent and identically distributed random variables with common distribution \( \mathcal{G}_\alpha^0(\alpha, \gamma, L) \). The \( r \)th sample moment is given by
\[
\hat{m}_r = \frac{1}{n} \sum_{i=1}^{n} z_i^r.
\]
Using (4) we can obtain expressions for the moments of order \( \frac{1}{2} \) and 1 as a function of \( \alpha \) and \( \gamma \). Replacing the population moments by their sample counterparts and replacing the parameters \( \alpha \) and \( \gamma \) by their respective estimators \( \hat{\alpha} \) and \( \hat{\gamma} \), we obtain the following system of equations:
\[
\begin{align*}
\hat{m}_1 &= \frac{\hat{\gamma}}{L} \frac{\Gamma(-\hat{\alpha} - 1) \Gamma(L + 1)}{\Gamma(-\hat{\alpha}) \Gamma(L)}, \\
\hat{m}_{\frac{1}{2}} &= \frac{\hat{\gamma}^{\frac{1}{2}}}{L} \frac{\Gamma(-\hat{\alpha} -\frac{1}{2}) \Gamma(L + \frac{1}{2})}{\Gamma(-\hat{\alpha}) \Gamma(L)}.
\end{align*}
\]

It then follows that
\[
\frac{\hat{m}_1 \Gamma(-\hat{\alpha}) \Gamma(L) L}{\Gamma(-\hat{\alpha} - 1) \Gamma(L + 1)} = \frac{\hat{m}_{\frac{1}{2}}^2 \Gamma^2(-\hat{\alpha}) \Gamma^2(L) L}{\Gamma^2(-\hat{\alpha} - \frac{1}{2}) \Gamma^2(L + \frac{1}{2})},
\]
which can be numerically solved. After obtaining the moments estimate of \( \alpha \) we can then compute \( \hat{\gamma} \).

In what follows, we consider that there is an edge when it is surrounded by two regions with textures. In this case, one of the regions is represented by \( \mathcal{G}_\alpha^0(\alpha_\ell, \gamma_\ell, L) \) whereas the other region is represented by \( \mathcal{G}_\alpha^0(\alpha_r, \gamma_r, L) \).
III. EDGE DETECTION IN SAR IMAGERY

A. 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 a 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 \( s = (z_1, \ldots, z_N) \) 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 \( (z_1, \ldots, z_j) \) comes from the \( G_{\alpha, \gamma}^0(\alpha_\ell, \gamma_\ell, L) \) distribution and \( (z_{j+1}, \ldots, z_N) \) comes from the \( G_{\alpha, \gamma}^0(\alpha_r, \gamma_r, L) \) distribution. Then, the value of \( j \) indicates the edge location. To find the transition point at \( s \), consider the likelihood function given by

\[
L = L(\alpha_\ell, \gamma_\ell, \alpha_r, \gamma_r) = \prod_{k=1}^{j} Pr(z_k; \alpha_\ell, \gamma_\ell) \times \prod_{k=j+1}^{N} Pr(z_k; \alpha_r, \gamma_r). 
\]

The log-likelihood function

\[
\mathcal{L} = \ln L = \sum_{k=1}^{j} \ln f_{G_{\alpha, \gamma}^0}(z_k; \alpha_\ell, \gamma_\ell) + \sum_{k=j+1}^{N} \ln f_{G_{\alpha, \gamma}^0}(z_k; \alpha_r, \gamma_r),
\]

where \( j \in \{1, \ldots, N-1\} \), is maximized when the index \( j \) is the edge. Using Equation (3), it follows that

\[
\mathcal{L} = \sum_{k=1}^{j} \ln \frac{L^L \Gamma(L-\hat{\alpha}_\ell)z_k^{L-1}}{\hat{\gamma}_\ell \Gamma(L)\Gamma(-\hat{\alpha}_\ell)(\hat{\gamma}_\ell + Lz_k)^{L-\hat{\alpha}_\ell}} + \sum_{k=j+1}^{N} \ln \frac{L^L \Gamma(L-\hat{\alpha}_r)z_k^{L-1}}{\hat{\gamma}_r \Gamma(L)\Gamma(-\hat{\alpha}_r)(\hat{\gamma}_r + Lz_k)^{L-\hat{\alpha}_r}}.
\]

Thus, the estimated transition point at \( s \) is given by

\[
\hat{j}_{GE} = \arg \max_j \mathcal{L}.
\]

The Gambini’s estimator \( \hat{j}_{GE} \) requires the estimation of \((\alpha_\ell, \gamma_\ell)\) and \((\alpha_r, \gamma_r)\) for all possible values of \( j \in \{1, \ldots, N-1\} \). 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. We also note that the moments estimator can be quite biased.
B. Nonparametric edge detection:

Edge detection using nonparametric methods as alternative to the Gambini’s estimator was proposed by Girón et al. [16]. 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 $n$ 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

$$T_{KW} = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1). \quad (11)$$

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

$$\hat{j}_{KW} = \arg\max_j T_{KW}. \quad (12)$$

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

IV. 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.

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 \(s = (z_1, \ldots, z_N)\), with an edge between two regions located at \(j \in \{1, \ldots, N\}\). Let \(\hat{j}_{kw}\) be the KW edge estimate. Using the estimate \(\hat{j}_{kw}\) we can obtain pseudo-samples \(s^* = (z_1^*, \ldots, z_N^*)\), where \((z_1^*, \ldots, z_N^*)\) is obtained by randomly sampling with replacement from \((z_1, \ldots, z_{\hat{j}_{kw}})\).

Similarly, \((z_{\hat{j}_{kw}+1}^*, \ldots, z_N^*)\) is obtained by randomly sampling with replacement from \((z_{\hat{j}_{kw}+1}, \ldots, z_N)\). 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 \((\hat{j}_{1}^*, \ldots, \hat{j}_{B}^*)\). This subsampling process is known as nonparametric bootstrap. Let \(\hat{j}_{(1)}^*, \ldots, \hat{j}_{(B)}^*\) the ordered values of \(\hat{j}_{1}^*, \ldots, \hat{j}_{B}^*\). The empirical distribution of \(\hat{j}_{KW}\) is given by

\[
\hat{F}_B(j') = \frac{\#\{\hat{j}_{(b)}^* \leq j'\}}{B}, \quad b \in \{1, \ldots, B\}.
\] (13)

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 is

\[
\left[ \hat{j}_{a/2}, \hat{j}_{1-a/2} \right]_{BBM} = \left[ \hat{j}_{kw} - (\hat{j}_{(B(1-a/2))} - \hat{j}_{kw}), \right.
\left. \hat{j}_{kw} - (\hat{j}_{(B(a/2))} - \hat{j}_{kw}) \right].
\] (14)

An alternative bootstrap confidence interval is the Percentile interval. The 100(1 – \(a\))% percentile interval is given by

\[
\left[ \hat{j}_{a/2}, \hat{j}_{1-a/2} \right]_{PERC} = \left[ \hat{j}_{(B(a/2))}^*, \hat{j}_{(B(1-a/2))}^* \right].
\] (15)

It is noteworthy that this interval is 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

\[
Z_b^* = \frac{\hat{j}_b^* - \hat{j}_{KW}}{\hat{\sigma}_b^{1/2}},
\] (16)

where \(\hat{\sigma}_b^{1/2}\) 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:

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

\(b \in \{1, \ldots, B\}\). The 100(1 – \(a\))% studentized bootstrap confidence interval is then given by

\[
\left[ \hat{j}_{a/2}, \hat{j}_{1-a/2} \right]_{ST} = \left[ \hat{j}_{KW} - \hat{\sigma}_b^{1/2} \hat{t}_{(B(1-a/2))}, \right.
\left. \hat{j}_{KW} - \hat{\sigma}_b^{1/2} \hat{t}_{(B(a/2))} \right].
\] (17)
The sample variance $v_b^*$ 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 \cite{17} recommend using $B' = 200$. However, they also recommend using $B = 1000$. This implies that estimation of $v_b^*$ 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 $v_b^*$ without having to resort to a second level bootstrap.

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

**Algorithm 1: ST1 Method**
1) If $\hat{j}_b^* = \hat{j}_{kw}$, then $v_b^* = 0$ (see Equation (16)) and the algorithm ends.
2) If the variance $\hat{j}_1^*, \ldots, \hat{j}_B^*$ is not equal to zero:
   a) Obtain $\hat{j}^* = \{\hat{j}_1^*, \ldots, \hat{j}_B^*\}$ by drawing with replacement from $\{\hat{j}_1^*, \ldots, \hat{j}_B^*\}$.
   b) Compute
   $$\tilde{v}_b^* = \frac{1}{B' - 1} \sum_{b'=1}^{B'} \left( \hat{j}_{b'}^* - \frac{1}{B'} \sum_{b''=1}^{B'} \hat{j}_{b''}^* \right)^2.$$  
   c) If $\tilde{v}_b^* = 0$, go back to step 2(a). Otherwise, the algorithm ends. This step is to be executed at most $B" > 0$ times.
   d) If the limiting number of repetitions for step 2(c) $(B")$ is reached, set $\hat{v}_b^*$ equal to the sample variance of $\hat{j}_1^*, \ldots, \hat{j}_B^*$.
3) If the sample variance of $\hat{j}^* = \{\hat{j}_1^*, \ldots, \hat{j}_B^*\}$ is equal to zero:
   a) Obtain $\hat{j}^*$ by estimating the edge using a new bootstrap replication of pixels strip $s$. Repeat this step until $\hat{j}^* \neq \hat{j}_1^*$. This step is to be executed at most $B" > 0$ times.
   b) If $\hat{j}^* = \hat{j}_1^*$, do $\tilde{v}_b^* = \hat{j}_1^* + 1$.
   c) Set $\tilde{v}_b^*$ equal to the sample variance of $\{\hat{j}_1^*, \ldots, \hat{j}_{B'}^*, \hat{j}_{B'}^*\}$.

It is noteworthy that Algorithm 1 must be executed for each $b \in \{1, \ldots, B\}$. 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.

We shall now introduce a second approximation to the bootstrap-$t$ method.

**Algorithm 2: ST2 Method**
1) Obtain estimates of the edge based on $B_x$ new resamples from the pixels strip $s$, thus obtaining $\hat{j}_{s}^* = \{\hat{j}_{B+1}^*, \ldots, \hat{j}_{B+B_s}^*\}$.
2) If the sample variance of the values $\hat{j}_{s}^*$ is not equal to zero:
   a) Obtain $\hat{j}^* = \{\hat{j}_1^*, \ldots, \hat{j}_{B'}^*\}$ by drawing with replacement from $\hat{j}_{s}^*$.
   b) Set $\tilde{v}_b^*$ equal to the sample variance of $\hat{j}^*$.
   c) If $\tilde{v}_b^* = 0$, go back to step 2(a). Otherwise, the algorithm ends. This step is to be executed at most $B"$ times ($0 < B" < B_x$).
   d) If the limiting number of repetitions set for step 2(c) $(B")$ is reached, set $\hat{v}_b^*$ equal to the sample variance of $\hat{j}_s^*$.
3) If the sample variance of the $\hat{j}_{s}^*$ values is equal to zero:
   a) Obtain $\hat{j}^*$ by estimating the edge using a new bootstrap replication of pixels strips $s$. Repeat this step until that $\hat{j}^* \neq \hat{j}_{B+1}^*$. This step is not to be executed more than $B"$ times.
   b) If $\hat{j}^* = \hat{j}_{B+1}^*$, do $\tilde{v}_b^* = \hat{j}_{B+1}^* + 1$.
   c) Estimate $\tilde{v}_b^*$ by sample variance of $\{\hat{j}_{B+1}^*, \ldots, \hat{j}_{B+B"-1}^*, \hat{j}_{B"}^*\}$. 
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.

V. 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 ([19]). 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 [20].

In all simulations, $L = 1$. Recall that this is the most challenging situation. All images we generated are rectangular and contain $20 \times 100$ 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 $\alpha_\ell, \alpha_r \in \{-2, -3, \ldots, -15\}$. For each region, the scale parameter $\gamma$ was chosen according to (5). 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 $\alpha_r$ and $\alpha_\ell$, the easier it is to locate the edge.

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 $1 - a = 0.95$ (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 $(\alpha_\ell, \alpha_r)$. The best performing estimator is the one whose exact coverage is closest to the nominal coverage. Figure 3 plots the distance $D = |\bar{C} - 0.95| \times 100\%$ against $\alpha_r$ (the roughness parameter value to the right of the edge). The six panels in the figure correspond to $\alpha_\ell \in \{-2, -5, -8, -11, -13, -15\}$. Situations in which $\alpha_\ell = \alpha_r$ (i.e., there is no edge) are not considered.

Visual inspection of Figure 3 reveals that all interval estimators perform equally well when $\alpha_\ell = -2$ and $\alpha_r \leq -4$; their coverages is 100%. We also notice that as $|\alpha_\ell - \alpha_r|$ increases the differences between the different coverages tend decrease, i.e., the different intervals display similar coverages. That happens because the distribution of $\hat{j}_{K\ell}$ tends to be more concentrated around of the true value $j = 50$ as $|\alpha_\ell - \alpha_r|$ increases, i.e., the estimator $\hat{j}_{K\ell}$ 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 $\alpha_\ell = -4$), the different intervals behave similarly, but when the two regions of are very homogeneous (for example $\alpha_\ell = -13$, and $\alpha_r = -12$), the percentile estimator outperforms all other interval estimators. Figure 5
show 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 4, a symbol indicates which is the best performing interval estimator for each configuration \((\alpha_l, \alpha_r)\); 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 3, except for the symbol used to represent BBM, which is now \(\otimes\). The symbol \(\boxdot\) 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.

The two studentized estimators, ST1 and ST2, outperformed the competition whenever \(|\alpha_l - \alpha_r| \leq 5\). In these situations, the values of \(\Delta\) 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 $\alpha_\ell$ and $\alpha_r$ becomes very large; see Figure 4.

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 5 displays average interval lengths. We notice that all methods produce similar average interval lengths, regardless of the configuration \((\alpha_\ell, \alpha_r)\), as expected. The percentile and BBM interval estimators have the same average length: approximately 93 pixels. 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 I

| \(\alpha_\ell\) | \(-3\) | \(-8\) | \(-13\) |
|-----------------|-------|-------|-------|
| \(\alpha_r\)    | 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  | 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  | 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  | 3.66  | 14.59 | 15.72 | 15.91 |
| -12             | 0.00  | 0.00  | 0.00  | 1.79  | 2.00  | 2.05  | 62.06 | 63.01 | 64.14 |
| -13             | 0.00  | 0.00  | 0.00  | 0.97  | 1.00  | 1.02  | 14.59 | 15.72 | 15.91 |
| -14             | 0.00  | 0.00  | 0.00  | 0.45  | 0.43  | 0.44  | 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 I 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.

The execution times of the different interval edge estimators for each Monte Carlo replication are given in Table III. 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. 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 II
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                            |

Fig. 6. E-SAR image with 1100 × 2400 pixels, and delimited regions.

VI. An application

In this section we apply the different edge interval estimators to a real image, i.e., to observed (not simulated) data. The SAR image is displayed in Figure 6. 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 7 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 6. 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 presented in Figure 8. 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.

Edge detection in rectangle (region) 3 of Figure 6 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 9. 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 (11) 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 6. A confidence interval that covers 94 pixels is obtained, thus suggesting that there is no edge. 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.
Fig. 8. Results for interval detection in region 2.

Fig. 9. Results for interval detection in region 3.
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.

VII. CONCLUDING REMARKS

We addressed the issue of edge detection in SAR images. Several bootstrap-based interval estimates were described and numerically evaluated. 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 indication that most likely there is no edge in that region of the SAR image.
Finally, real (not simulated) data were analyzed. We performed edge detection in several regions of a SAR image.

**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 VI.

**REFERENCES**

[1] 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.

[2] 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.

[3] R. J. Cintra, A. C. Frery, and A. D. C. Nascimento. Parametric and nonparametric tests for speckled imagery. *Pattern Analysis and Applications*, 16:141–161, 2013.

[4] P. Shui, and D. Cheng. Edge Detector of SAR Images Using Gaussian-Gamma-Shaped Bi-Windows. *IEEE Geoscience and Remote Sensing Letters*, 9(5):846–850, 2012.

[5] F. Baseline, and G. Ferraioi. Statistical Edge Detection in Urban Areas Exploiting SAR Complex Data. *IEEE Geoscience And Remote Sensing Letters*, 9(2):185–189, 2012.

[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] W. Jiang, K. Lam, and T. Shen. Efficient Edge Detection Using Simplified Gabor Wavelets *IEEE Transactions On Systems, Man, And Cybernetics – part B: Cybernetics*, 39(4):1036–1047, 2009.

[8] J. Gambini, M. E. Mejail, J. Jacobo-Berlies, 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, NOV 20 2006.

[9] A. Blake and M. Isard. *Active Contours*. Springer, Berlin, 1998.

[10] S. S. Chen, J. M. Keller, and R. M. Crownover. On the calculation of fractal features from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1087–1090, October 1993.

[11] P. Perona and J. Malik. Scale-space and edge-detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[12] J. Gambini, M. E. Mejail, J. Jacobo-Berlies, and A. C. Frery. Accuracy of edge detection methods with local information in speckled imagery. *Statistics and Computing*, 18(1):15–26, 2008.

[13] A. C. Bovik, T. S. Huang, and D. C. Munson. Nonparametric tests for edge-detection in noise. *Pattern Recognition*, 19(3):209–219, 1986.

[14] 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.

[15] 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.

[16] 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.

[17] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993.

[18] 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.

[19] J. A. Doornik. *Object-Oriented Matrix Programming Using Ox*. Timberlake Consultants Press & Oxford, London, 4rd ed, URL http://www.doornik.com, 2006.

[20] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Statistics and Computing, Springer, New York, 2002.

[21] K. L. P. Vasconcellos, A. C. Frery, and L. B. Silva. Improving estimation in speckled imagery. *Computational Statistics*, 20:503–519, 2005.