LSDSAR, a Markovian a contrario framework for line segment detection in SAR images

Chenguang Liu, Rémy Abergel, Yann Gousseau, Florence Tupin

LSDSAR, a Markovian a contrario framework for line segment detection in SAR images

Chenguang Liu[a,*], Rémy Abergel[b], Yann Gousseau[a], Florence Tupin[a]

[a] *LTCI, Télécom ParisTech, Université Paris-Saclay, France*
[b] *Laboratoire MAP5 (CNRS UMR 8145), Université Paris Descartes, France*

Abstract

In this paper, we propose a generic method for the detection of line segments in SAR images. The approach relies on an *a contrario* framework and is inspired by the state-of-the art LSD detector. As with all *a contrario* approaches, false detections are controlled through the use of a background model, whose development is especially challenging in the framework of SAR images. Indeed, statistical characteristics of SAR images strongly differ from those of optical images, making the use of existing background models intrinsically inadequate. In order to circumvent this problem, we proceed in two steps. First, the building blocks of the detector, namely the local orientations, are computed carefully to avoid any spatial bias. Second, we propose a new background model, in which the spatial dependency between local orientations are modeled with a Markov chain. This is in strong contrast with most existing *a contrario* methods who heavily rely on independence assumptions. We provide a complete and detailed algorithm for our line segment detector, and perform experiments on synthetic and real images demonstrating its efficiency.

Keywords: Line segments, SAR images, a contrario models, Markov chain, local orientations.

*Corresponding Author.
Email addresses: chenguang.liu@telecom-paristech.fr (Chenguang Liu), Remy.Abergel@parisdescartes.fr (Rémy Abergel), yann.gousseau@telecom-paristech.fr (Yann Gousseau), florence.tupin@telecom-paristech.fr (Florence Tupin)

1. Introduction

Two kinds of linear structures can be distinguished in Synthetic Aperture Radar (SAR) images: narrow bands of pixels having a given width [1], usually simply called *linear features*, and straight subparts of edges, that from now on we call *line segments*. The detection of linear features in SAR images has received a lot of attention, typically in view of the detection of road networks [2, 3, 4]. The detection of line segments in SAR images has received less attention, in contrast with the many such works that have been developed for optical images. The main reason for this situation is that the estimation of a reliable gradient information remains a difficult task in the presence of strong speckle noise, a noise that is inherent to all coherent imaging systems. Nevertheless, line segments are very important features in SAR images, mostly because many man-made objects like buildings, farmlands or airports can be described by line segments. Besides, most geometric structures can be approximated by line segments. In addition, line segments can be extracted as low level features and then be used for tasks such as image registration and target recognition.

Due to the strong speckle noise, methods that are effective for optical images cannot be straightforwardly applied to SAR images. First, the usual assumption that noise is additive and Gaussian is wrong. Second, and more importantly, the strong level of noise encountered in SAR images makes most optical approaches inefficient. Taking the logarithm of the amplitude or intensity of SAR images can change multiplicative noise to additive noise but this does not allow the plain application of optical methods, as we will see in the experimental section in the case of the state-of-the-art LSD detector [5].

The usual way to detect line segments in SAR images is as follows. First, a constant false alarm rate edge detector, such as that described in [6], is applied to the image, followed by a Hough transform [7, 8] to detect lines. Then, post processing steps are applied to localize Hough lines into line segments. Many methods of this kind have been proposed for SAR images following the early work [9], in the context of different applications. In [10], line segments are ex-

tracted by the Hough transform and then used to reconstruct buildings from meter-resolution multi-aspect SAR images. An optical-to-SAR image registration method is proposed in [11], relying on line segments that are detected using a ratio-based gradient and the Hough transform. The same idea was previously explored in [12]. In [13], edge detection using phase symmetry [14] and wavelet correlations is followed by a Hough transform in order to detect ship wakes.

A common limitation of the aforementioned approaches is that the performance of the Hough transform critically relies on both a preliminary edge detection and on the selection of parameters. The input of the Hough transform is usually a binary edge map. Many dedicated methods have been proposed for SAR images to compute gradients [6, 15, 16], but extracting a binary edge map necessitates a difficult compromise between suppressing false alarms due to speckle and preserving edges of low contrast. Besides, the corresponding threshold choices are strongly image-dependent. An interesting approach, which was recently proposed in [17, 18], detects lines from the magnitude field instead of a binary edge map, but the subsequent detection tasks still require non-trivial parameter tunings.

Among the various approaches to the automatic setting of parameters for low-level vision tasks, the *a contrario* methodology was initially proposed for the detection of line segments [19] in optical images. This powerful approach, relying on a statistical control of the number of false detections, was then applied to many detection and grouping tasks [20] and, in particular, a state-of-the-art line segment detector, named LSD, was proposed in [5]. The goal of the present paper is to investigate the applicability of this approach to the case of SAR images, which results in a very challenging task. Indeed the LSD detector, as most *a contrario* approaches, relies on a null hypothesis against which segments are detected. Unfortunately, this null hypothesis is completely inadequate for SAR images. More precisely, local orientations (each of which being defined as the direction perpendicular to the gradient orientation) are grouped against the hypothesis that they are 1) uniformly distributed and 2) mutually independent. Both these hypotheses appear to be structurally wrong in SAR

images. First, classical ways [6] to compute the gradient in SAR images yield a non uniform distribution of the orientation, even in the absence of geometrical structures. Second, and more importantly, the speckle noise imposes the use of strong filtering schemes, implying strong structural dependencies between nearby orientations. In the proposed LSDSAR approach, we replace the gradient computation with a ratio based method [16], which yields robust and unbiased local orientations at each pixel. Further, we replace the crucial independence hypothesis between local orientations by a first order Markov chain modeling, which in practice is enough to counterbalance the effect of filtering and yields an efficient control of the number of false detections. The result is a generic line-segment detector adapted to the specific structure of SAR images.

This paper is organized as follows: in Section 2, we give a description of the original LSD algorithm. In Section 3, the new line segment detector is provided. In particular, the computation of the gradient is detailed, as well as the use of a first order Markov chain in the *a contrario* model. Line segment detection results on both synthetic and real SAR images are given in Section 4 and are compared to the results of state-of-the-art line segment detectors. We finally conclude in Section 5 and present some perspectives.

2. The LSD algorithm

The LSD algorithm is a state-of-the-art line segment detector for optical images, originally introduced in [5], and also described in [22]. As in the precursor work from [19], segments are detected by grouping local orientations. The groups are validated by computing the so-called *number of false alarms* (NFA). In contrast with the original approach [19], the LSD detector is very fast thanks to an efficient search heuristic that was proposed in [23]. The LSD algorithm relies on three steps: the *region growing* step, the *rectangular approximation* step and the *line segment validation* step. For self-containedness, we detail each of these steps in the following.

2.1. Region growing

The first step of LSD is region growing. First, the image gradient is computed at each pixel. The local orientation is defined at each pixel as the orientation perpendicular to the gradient orientation. Then, starting from the pixel having larger gradient magnitude, pixels having the same local orientation as the region up to a certain angle tolerance are gathered together until no more pixel can be added, yielding a candidate region. Then the process starts again from the next remaining pixel with higher gradient magnitude. Each time a pixel is added into the region, the angle of the region is updated as the average local orientation of the already gathered pixels.

2.2. Rectangular approximation

After region growing, we are given a set of candidate regions. The next step is to approximate each region with a rectangle. Four parameters are used to describe the rectangles: center, angle, length and width. The center of the rectangle is defined as the center of mass (the mass being the gradient magnitude) of the region. The angle of the rectangle is defined as the angle between the horizontal axis and the first inertia axis of the region. The length and width of the rectangle are then defined as the smallest values allowing a complete covering of pixels of the considered region.

2.3. Line segment validation

In order to decide whether or not a line segment is present in the rectangle, an *a contrario* criterion is used to evaluate how unlikely a similar structure is to happen by chance in a random image. The general idea of the *a contrario* detectors is that no meaningful structure should happen by chance in a random configuration according to the so-called Helmholtz principle [20]. What is considered as a *random configuration* is more formally defined in a $\mathcal{H}_0$ model, which describes a kind of data where no detection should occur. In the case of LSD, an image I_0 follows the $\mathcal{H}_0$ model if it satisfies the following requirements:

a. The local orientations of the pixels in I_0 are independent random variables.

b. Those random variables follow a uniform distribution over $[0, 2\pi]$.

The likeliness of a given rectangular configuration is evaluated through the number of its aligned pixels. In the following, we say that a pixel of a rectangle is *aligned* if it has the same local orientation as the rectangle, up to a given angle tolerance. For a given rectangle r in the observed image I, we write $n(r)$ for its total number of pixels and $k(r)$ for its number of aligned pixels. In order to evaluate the unlikeliness of a given rectangular configuration, we compute the expected number of rectangles having the same configuration that would appear in a random image I_0 following $\mathcal{H}_0$. We consider a rectangle with size $n(r)$ in this image and write $k_0(r)$ for the random variable equals to the number of aligned points in this rectangle. The distribution of this variable only depends on the size of r and not on its shape, as we will see shortly. Then, we define a Number of False Alarms (NFA) [19, 5] associated to r as

$$\mathrm{NFA}(r) = N_{\mathcal{R}} \cdot \mathbb{P}(k_0(r) \geq k(r)) \,, \tag{1}$$

where $N_{\mathcal{R}}$ is the total number of possible rectangles, which is approximated by the value $11 \cdot (MN)^{5/2}$ for an image containing $M \times N$ pixels in [5].

In practice, the rectangle r will be detected as a segment if $\mathrm{NFA}(r) \leq \varepsilon$, for a given threshold ε set by the user. The interest of such an approach is that the NFA defined in (1) satisfies the so-called NFA-property [19, 5], which can be written as

$$\forall \varepsilon > 0, \quad \mathbb{E}_{\mathcal{H}_0}\left[\#\left\{r,\ \mathrm{NFA}(r) \leq \varepsilon\right\}\right] \leq \varepsilon \,, \tag{2}$$

where # stands for the cardinality of a set. This property means that using the threshold $\mathrm{NFA}(r) \leq \varepsilon$, no more than ε rectangles will be detected on the average on a random image I_0 following $\mathcal{H}_0$, assuming that we consider all possible rectangles. Again, the idea behind this detection rule is that bounding the total number of false detections under model $\mathcal{H}_0$ is enough to yield efficient detection thresholds. In general we set $\varepsilon = 1$, which guarantees that, on average, at most one false detection could be done. The rectangles validated as true line segments because their NFA is less than ε are called ε-*meaningful* segments, or

simply *meaningful* segments when $\varepsilon = 1$.

In order to get an explicit detection rule, it remains to compute $\mathbb{P}_{H_0}(k_0(r) \geq k(r))$. According to the second assumption b) of $\mathcal{H}_0$ (uniform distribution), the probability that a pixel is aligned is

$$p = \frac{\tau}{\pi}, \tag{3}$$

where τ is the angle tolerance. Moreover, because of the independence assumption a) of $\mathcal{H}_0$, the number of aligned pixels in r follows a Binomial distribution. Thus, we have

$$\mathbb{P}_{\mathcal{H}_0}(k_0(r) \geq k(r)) = \mathcal{B}(n(r), k(r), p), \tag{4}$$

where $\mathcal{B}$ denotes the tail of the Binomial distribution,

$$\mathcal{B}(n(r), k(r), p) = \sum_{i=k(r)}^{n(r)} \binom{n(r)}{i} p^i (1-p)^{n(r)-i}.$$

Therefore, the NFA in (1) can be explicitly computed using

$$\mathrm{NFA}(r) = N_{\mathcal{R}} \cdot \mathcal{B}(n(r), k(r), p)\,. \tag{5}$$

The main difficulty to develop a similar line segment detector for SAR images is that the two crucial assumptions of model $\mathcal{H}_0$ are in general not suited to such images. First, the most usual ways to compute local orientations in SAR images are more or less biased toward certain directions, so that the assumption of a uniform distribution does not hold, even in a pure noise image. Second, and more important, SAR images are strongly impacted by speckle noise, so that a strong filtering is necessary to compute reliable local orientations, which in turn implies that the independence assumption does not hold, even when local orientations are evaluated on a pure noise image. In the next section, we will address both these issues. First, in Section 3.2, we explain how to compute accurate and non-biased local orientations. Then, in Section 3.3, we propose a relaxation of the independence assumption of the $\mathcal{H}_0$ model, by making use of a first order Markov chain.

-

3. LSDSAR, a line segment detector for SAR images

3.1. Statistics of SAR images

SAR sensors send electro-magnetic waves which are then backscattered by the elements lying on the ground and processed to form a SAR image. Let $a\exp(j\phi)$ be the complex backscattered signal and a the amplitude image. Notice that the SAR images considered in this paper are all amplitude images.

SAR images are corrupted by a strong multiplicative noise known as speckle. The probability density function (pdf) f_a of the amplitude a for a region with mean reflectivity μ (the physical parameter characterizing the imaged area) and $L-look$ under the hypothesis of fully developed speckle [2, 24, 25] is:

$$f_a(x|\mu) = \frac{2}{\Gamma(L)}\left(\frac{L}{\mu}\right)^L x^{2L-1}e^{-(Lx^2/\mu)}, \tag{6}$$

where $\Gamma : x \mapsto \int_0^{+\infty} t^{x-1}e^{-t}dt$ is the Gamma function. This pdf is a Nakagami distribution. The speckle noise s follows a Nakagami distribution with the parameter $\mu = 1$ and the multiplicative noise model is given by $a = s.\mu$. Original SAR images are 1-look images, but a pre-processing of multi-looking can be applied to reduce the signal fluctuations. It corresponds to an incoherent averaging of the backscattered values in the power domain to reduce the noise.

Because of the speckle, detecting line segments in SAR images is a completely different situation from detecting line segments in optical images since optical images are contaminated by additive noise. LSD relies on the computation of the gradient in each pixel. However, usual gradients based on finite differences schemes do not have a constant false alarm rate when used to detect edges in SAR images [6]. This part is thus replaced with a ratio based method, better appropriate to SAR images. In the following we discuss the impact of this step on the previously mentioned $\mathcal{H}_0$ hypotheses.

3.2. Gradient computation in SAR images

The first such constant false alarm rate edge detector for SAR images has been proposed in [6]. The Ratio of Average values (ROA) instead of difference of

average values is used to compute the gradient, ensuring a constant false alarm rate whatever the underlying reflectivity R of an area. For a given pixel located at position (x, y) in the image I, the ratio of averages $R(x, y)$ is computed along P directions as the ratio of the arithmetic mean of pixel values of two opposite side windows. In the i-th direction, where $1 \leq i \leq P$, this ratio is given by

$$R^i(x, y) = \frac{M_1^i(x, y)}{M_2^i(x, y)}\,, \tag{7}$$

where $M_1^i(x, y)$ and $M_2^i(x, y)$ denote the arithmetic means of the image computed over two opposite side windows, separated by an axis with direction i. This ratio is then normalized as

$$T^i(x, y) = \max\left(R^i(x, y), \frac{1}{R^i(x, y)}\right). \tag{8}$$

The magnitude $|ROA|$ of the gradient at position (x, y) is then defined as

$$|ROA(x, y)| = \max_{1 \leq i \leq P} T^i(x, y)\,, \tag{9}$$

and its direction is defined as the direction having the highest value of ratio. Once the gradient is computed, a threshold on its magnitude can be used for the purpose of edge extraction. Pixels having magnitude values higher than the threshold are considered as edges, while others are not.

Although ROA works well on isolated step edges, it is less efficient in the presence of multiple edges, For that reason, a multiple edge detector, ROEWA, was proposed in [15]. Instead of computing the ratio along P directions, only the ratio along the horizontal and vertical directions are computed in ROEWA, and the arithmetic means are replaced by weighted averages. In the horizontal direction, the ratio $R^h(x, y)$ and its normalization $T^h(x, y)$ are computed as

$$R^h(x, y) = \frac{m_1^h(x, y)}{m_2^h(x, y)}\,,$$

$$T^h(x, y) = \max\left(R^h(x, y), \frac{1}{R^h(x, y)}\right),$$

where

$$\begin{aligned} m_1^h(x,y) &= \sum_{x'=-W}^{W} \sum_{y'=1}^{W} I(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}}, \\ m_2^h(x,y) &= \sum_{x'=-W}^{W} \sum_{y'=-W}^{-1} I(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}}, \end{aligned}$$

and where W is the upper integer part of $\log(10) \times \alpha$. A complete study of the influence of α is beyond the scope of this paper but can be found in [15]. The ratio along the vertical direction $R^v(x,y)$ and its normalization $T^v(x,y)$ are computed in the same way. The normalized ratios $T^h(x,y)$ and $T^v(x,y)$ being considered as the horizontal and vertical components of the ROEWA gradient, its magnitude is simply given by

$$|ROEWA(x,y)| = \sqrt{T^h(x,y)^2 + T^v(x,y)^2}\,. \tag{10}$$

Although ROEWA gives an efficient and accurate way to compute the magnitude in each pixel, this method does not give a precise measure of the edge orientations. Increasing the number of directions, as in ROA, yields the so-called *multi-directional ratio-based methods*, which are however quite time consuming.

In [16], a new gradient named Gradient by Ratio (GR) was proposed. Different from ROEWA, the horizontal and vertical gradient components are defined as

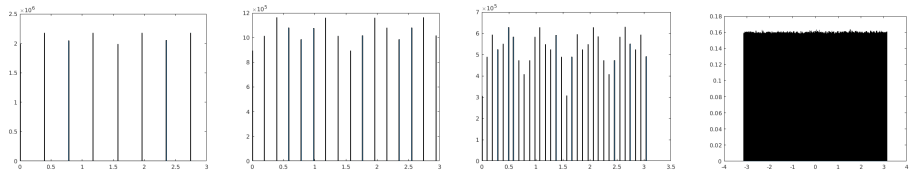
$$G^h(x,y) = \log(R^h(x,y)),$$

$$G^v(x,y) = \log(R^v(x,y)).$$

The magnitude $|GR(x,y)|$ and orientation $ang(GR(x,y))$ of GR at position (x,y) are defined by

$$\begin{aligned} |GR(x,y)| &= \sqrt{G^h(x,y)^2 + G^v(x,y)^2}, \\ ang(GR(x,y)) &= \arctan \frac{G^v(x,y)}{G^h(x,y)}. \end{aligned}$$

GR gives a very efficient and effective way to compute both magnitude and orientation of the gradient [16]. Therefore a first way to adapt LSD to SAR images is to replace the gradient computation method of LSD with GR.



(a) 8 directions (b) 16 directions (c) 32 directions (d) GR

Fig. 1: Histograms of the local orientations computed over a 1-look pure speckle noise image of size 4096×4096 pixels, using the multi-directional ratio based methods (a,b,c) or GR (d), with the same setting $\alpha = 4$ for all methods.

As explained before, the *a contrario* framework of LSD relies on a uniform distribution over $[0, 2\pi]$ of the local orientations computed in a pure noise image. To check whether this assumption is compatible or not with the above-mentioned ways to evaluate local orientations, we plot in Fig. 1 the histograms of the local orientations obtained using GR over a pure speckle noise image of size 4096×4096 and compare it with those obtained using the multi-directional ratio-based methods with 8-directions, 16-directions and 32-directions. We used the same exponential weight function and the same weight parameter $\alpha = 4$ for GR and the multi-directional methods. In order to reduce the influence of rotation for the multi-directional ratio-based methods, we use circle-shaped windows to improve invariance by rotation. We also guarantee that the circle-shaped windows for the multi-directional ratio-based methods and rectangle windows for GR have the same number of pixels.

From Fig. 1 (a), (b) and (c), we can see that multi-directional ratio-based methods all deviate from a uniform distribution, with stronger deviation when the number of directions is increased. This is all the more problematic that using a large number of orientations (16 or 32) is needed to avoid strong orientation quantization. From Fig. 1 (d), we can see that the local orientations computed using GR follow the hypothesis $\mathcal{H}_0$-b of the *a contrario* model. Moreover, GR does not yield the orientation quantization which is inherent to multi-directional ratio-based methods. For those reasons, we will use GR to compute the magnitude and local orientations of the gradient.

3.3. Modeling weak dependency between local orientations

Thanks to the gradient computation technique detailed in the previous paragraph, the assumption of uniform distribution for the local orientations $\mathcal{H}_0$-b approximately holds true in a pure noise image. However, the independence assumption of local orientations $\mathcal{H}_0$-a is not satisfied because of the local averaging needed to compute gradients in speckle noise. Consequently, the usual *a contrario* framework badly fails to control the number of false detections when the local orientations are computed with a method suited to SAR, as we will show our experiments. In order to account for the strong dependencies between nearby pixels introduced by filtering, we draw from the ideas in [26], generalizing the *a contrario* methodology to the grouping of events that are not mutually independent. Specifically, the distribution of local orientations at nearby pixels will be modeled by a first order Markov chain, which appears as a good compromise between accuracy and tractability. Notice that, contrarily to the application cases in [26], the necessity to take dependency into account in our case is a consequence of the way the gradient is computed (using large averaging windows) and not of the structure of the noise.

We consider a rectangle r in the observed image I, containing $n(r)$ pixels. We assume that pixels within the rectangle are ordered so that two consecutive pixels are neighbors. For $t = 1, \ldots, n(r)$, let the variable X_t be equal to 1 if the t-th pixel is aligned, and 0 otherwise. Then $k(r) = \sum_{t=1}^{n(r)} X_t$ represents the number of aligned pixels within rectangle r.

Instead of assuming that X_1, X_2, ..., $X_{n(r)}$ are mutually independent, we assume that they follow a Markov chain of order one, i.e., we assume that, for all $1 < t \leq n(r)$,

$$\mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, \ldots, X_1 = x_1) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}). \quad (11)$$

Therefore, the distribution of aligned pixels is characterized by the four transi-

tion probabilities

$$\mathbb{P}(X_1 = x_1 | X_0 = x_0)\,, \quad \text{for } (x_0, x_1) \in \{0,1\}^2\,.$$

For brevity, they may be written $\mathbb{P}(1|1)$, $\mathbb{P}(1|0)$, $\mathbb{P}(0|1)$, and $\mathbb{P}(0|0)$ in the following. In practice, those transition probabilities will be estimated by computing the local orientation using GR on a pure speckle noise image. Now, let us introduce a modified background model, that we denote by $\mathcal{H}_0'$.

Definition 1 ($\mathcal{H}_0'$ model)**.** *We say that an image I_0 follows the $\mathcal{H}_0'$ model when its local orientations computed using GR satisfy the following properties:*

a. The local orientations follow a Markov chain of order one.

b. Each local orientation follows a uniform distribution over $[0, 2\pi]$.

The new definition of the NFA follows from this modified model. As before we consider a rectangle r containing $n(r)$ pixels and $k(r)$ aligned pixels in the observed image I. We write $k_0(r)$ for the number of aligned points within r in a random image I_0 following model $\mathcal{H}_0'$. The initial NFA formula (1) of LSD is naturally changed into

$$\mathrm{NFA}'(r) = N_{\mathcal{R}} \cdot \mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r)) \tag{12}$$

which also satisfies the NFA-property (2) [27].

Now, let us focus on the practical evaluation of (12). Because of the Markov chain assumption $\mathcal{H}_0'$-b, the probability of having $k_0(r) \geq k(r)$ involved in (12) is given by

$$\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r)) = \sum_{x_1 + \ldots + x_{n(r)} \geq k(r)} \mathbb{P}(X_1 = x_1) \cdot \prod_{t=2}^{n(r)} \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1})\,. \tag{13}$$

The probabilities $\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r))$, for all possible values of $k(r)$ and $n(r)$, are heavy to compute using a straightforward implementation of (13). In order to overcome this limitation, we proceed as in [26] and compute $\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r))$

using a dynamic programming algorithm and a descending induction. Indeed, writing $n = n(r)$ and $k = k(r)$, letting $Y_t = \sum_{j=t}^{n} X_j$, one observes that, for $t \leq n-1$, we have

$$\mathbb{P}(Y_t \geq k) = \mathbb{P}(Y_{t+1} \geq k | X_t = 0) \cdot \mathbb{P}(X_t = 0) + \mathbb{P}(Y_{t+1} \geq k-1 | X_t = 1) \cdot \mathbb{P}(X_t = 1)\,. \quad (14)$$

Besides, one has, for $x \in \{0,1\}$ and $k' \geq 1$,

$$\begin{aligned}\mathbb{P}(Y_{t+1} \geq k' | X_t = x) &= \sum_{y \in \{0,1\}} \mathbb{P}(Y_{t+2} \geq k' - y, X_{t+1} = y | X_t = x) \\ &= \mathbb{P}(Y_{t+2} \geq k' | X_{t+1} = 0) \cdot \mathbb{P}(X_{t+1} = 0 | X_t = x) \\ &\quad + \mathbb{P}(Y_{t+2} \geq k' - 1 | X_{t+1} = 1) \cdot \mathbb{P}(X_{t+1} = 1 | X_t = x)\,, \quad (15)\end{aligned}$$

and $\mathbb{P}(Y_n \geq k' | X_{n-1} = x)$ is simply given by

$$\mathbb{P}(Y_n \geq k' | X_{n-1} = x) = \begin{cases} 1 & \text{if } k' = 0\,, \\ \mathbb{P}(1|x) & \text{if } k' = 1\,, \\ 0 & \text{otherwise}\,. \end{cases} \quad (16)$$

It follows that (14) can be computed in polynomial time [26], and so does the probability $\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r))$ needed in (12).

In practice, we can precompute the values of $\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r))$ for all $(n_0(r), k(r)) \in \{1, \ldots, N_{\max}\}^2$, with N_{max} a large enough value. In our implementation we used $N_{\max} = 5000$. In the case the computation of $\mathbb{P}_{\mathcal{H}_0'}(k_0(r) \geq k(r))$ is needed for $n(r)$ greater than $N_{\max}$, though theoretically it can still be done using (14), (15) and (16), we accept the rectangles directly to save memory space and computation time. The reason is that a rectangle with large enough number of pixels should be a meaningful structure in the random image under the hypothesis of the background model. Notice that, as mentioned above, the transition probabilities involved in (15) and (16) are estimated from the local orientations computed using GR in a pure Nakagami noise. Contrarily

to [26], the probabilities can be computed in advance for each possible parameter choice in the gradient computation, namely for each choice of the α value. Given an image of 1-look pure Nakagami noise, we consider all horizontal and vertical lines. This is because local orientations are computed by the 2-direction method (horizontal and vertical directions), which imposes the strongest correlations along the horizontal and vertical directions. For each line, we consider the empirical frequency of X_t and X_{t-1} over all pixel pairs to estimate $\mathbb{P}(x_1|x_0)$ for $x_0, x_1 = 0, 1$. In practice, we only need to estimate $\mathbb{P}(x_1|x_0)$ for $x_0 \neq x_1$, since we have $\mathbb{P}(1|1) = 1 - \mathbb{P}(0|1)$ and $\mathbb{P}(0|0) = 1 - \mathbb{P}(1|0)$. In Table 1, we display the transition probabilities estimated using a 1-look pure Nakagami noise image of size 4096×4096 pixels, for different values of parameter α. The angle tolerance was set to 22.5 degrees, as in the original LSD algorithm. One can see that, as α increases, more dependencies are introduced between adjacent pixels.

3.4. The complete LSDSAR algorithm

The proposed LSDSAR algorithm relies on three main steps, namely the *region growing*, *rectangular approximation* and *line segment validation* steps, on which we perform several modifications.

(i) *Region growing*: in this step we replace the finite-differences based gradient of LSD by GR, as described in Section 3.2;

(ii) *Rectangular approximation*: this step is identical to that of the original LSD algorithm;

(iii) *Line segment validation*: in this step, we compute the number of false alarms using (12) instead of (1), as we described in Section 3.3.

A pseudocode description of LSDSAR is proposed in Algorithm 1. In this pseudocode description, the routines *RegionGrow* and *Rectangle* correspond to the *Region growing* and *Rectangular approximation* steps mentioned above. Up to the modification we gave in (i) for gradient computation, those routines are

Table 1: estimated transition probabilities of the first order Markov chain for $\tau = 22.5°$ and different values of α.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
$\mathbb{P}(1\|1)$	0.2444	0.4078	0.5144	0.5874	0.6356
$\mathbb{P}(1\|0)$	0.1076	0.0846	0.0694	0.0590	0.0521

Table 2: influence of the setting of α on the number of (false) detections obtained using LSDSAR on a 1-look pure Nakagami noise of size 4096×4096 pixels.

Parameter	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Number of detections	0	2	10	51	97

Table 3: number of false detections obtained using LSD+GR (second column) and LSDSAR (third column) with the setting $\varepsilon = 1$ in a pure speckle noise image with size 4096×4096.

Parameter α	**LSD+GR**	**LSDSAR**
$\alpha = 1$	2	0
$\alpha = 2$	2710	2
$\alpha = 3$	16602	10
$\alpha = 4$	29771	51
$\alpha = 5$	35916	97

Table 4: average number of (false) detections obtained using LSDSAR over one hundred single look images with size 1024×1024, for different values of the NFA threshold ε.

NFA threshold ε	0.01	0.1	1	10	100
Average number of false detections	5.2	8.6	14.5	24.1	1074

Table 5: Stability of the detection performances.

F_1-score	minimal	maximal	average	median
LSDSAR (default parameters)	0.71	0.82	0.78	0.78
IEFA (tuned parameters)	0.73	0.81	0.76	0.77

Table 6: real SAR images.

Satellite	Sentinel 1	TerraSAR-X
Place	Lelystad (Netherlands)	San Francisco (United States)
Date	06/10/2015	02/10/2011
Image mode	Stripmap	-
Image size (pixels)	1024×3072	2048×2048
pixel-spacing (azimuth)	4m	-
pixel-spacing (range)	14m	-

the same as those used in LSD, which are carefully described in [22]. The routine *AlignedPixelDensity*, also explicitly defined in [22], is used to compute the density of aligned pixels in the rectangle. When the density of aligned pixels in the rectangle is smaller than the threshold D, the routine *CutRegion* is used to refine the rectangle. In the original LSD, this *CutRegion* routine involves the use of a modified angle tolerance, which is computed adaptively to the orientations of pixels in the rectangle. In our implementation of the *CutRegion* procedure, we decided to set this modified tolerance parameter always equal to $\tau/2$, in order to be able to use some pre-tabulated values of the transition probabilities of the Markov chain for that particular setting of τ. Last, the *ImprovedRectangle* routine defined in LSD, which also involves several refinements of the angle tolerance is again adapted in order to make the modified angle tolerance always equal to $\tau/2$ (step 1 of the *ImprovedRectangle* routine) and $\tau/4$ (step 5 of the *ImprovedRectangle* routine). Last, it should be noticed that, contrarily to the original LSD algorithm, we do not need to pre-filter the image (rescaling, Gaussian filtering) and we do not need the threshold for the gradient magnitude because SAR images are mostly free of aliasing and quantization effects.

3.5. Parameters setting

Algorithm 1 relies on four parameters, ε, α, τ and D. As usual with the *a contrario algorithms*, the NFA-property (2) provides a handy meaning for the NFA threshold parameter ε, even for non-expert users. This threshold ε represents an upper-bound on the average number of detections that we allow in pure noise input data. As mentioned before, a common setting for this parameter is $\varepsilon = 1$. Besides, from our simulations, we found appropriate the setting $\tau = 22.5°$ proposed in the original LSD algorithm. In this section we propose to discuss the influence of the two remaining parameters α and D on the detection results, and we propose a default setting for them.

First, let us focus on the setting of α, the regularization parameter used in the GR computation. Increasing the value of α helps to suppress the speckle, but in turn introduces more dependencies between adjacent pixels. Choosing a

Algorithm 1: LSDSAR

Inputs: a SAR (amplitude) image I with size $M \times N$, the NFA threshold ε, the regularization parameter α, the angle tolerance τ and the density threshold D.

Output: the list L containing the detected line segments.

Initialization:

1) Apply GR with parameter α on the input image.

2) Compute *OrderedList*, the sorting in descending order of the pixels of I according to their gradient magnitudes

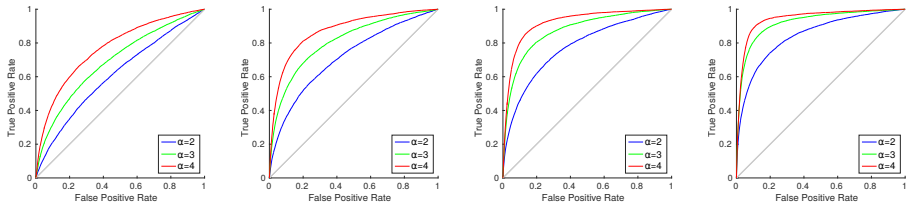
3) Precompute all values of $\mathbb{P}_{\mathcal{H}'_0}(k(r_0) \geq k)$ (for $(n(r_0), k) \in \{0, \ldots, N_{\max}\}^2$), needed in (12).

4) Define NFA$'$ as in (12), using $N_{\mathcal{R}} = 3 \cdot (MN)^{5/2}$.

for $P \in$ *OrderedList* **do**
- *region* $\leftarrow$ *RegionGrow*(P, τ)
- $r \leftarrow$ *Rectangle(region)*
- **while** *AlignedPixelDensity(r,*τ*)¡D* **do**
  - *region* $\leftarrow$ *CutRegion(region)*
  - $r \leftarrow$ *Rectangle(region)*
- nfa $\leftarrow$ NFA$'(r)$
- **if** nfa $\leq \varepsilon$ **then** $L \leftarrow L \cup \{r\}$
- **else**
  - $r \leftarrow$ *ImproveRectangle*(r)
  - nfa $\leftarrow$ NFA$'(r)$
  - **if** nfa $\leq \varepsilon$ **then** $L \leftarrow L \cup \{r\}$

return L

The reason of the setting $N_{\mathcal{R}} = 3 \cdot (MN)^{5/2}$ (instead of $N_{\mathcal{R}} = 11 \cdot (MN)^{5/2}$ in LSD) is that the total number of segments is potentially multiplied by three (instead of 11 in LSD) because each rectangle may be modified only two times (instead of 10 in LSD) during the refinement step (routine *ImproveRectangle*). Notice that, step 2) can be accelerated by using a linear-time pseudo-ordering algorithm, as in LSD. Note also that, in step 3), one precomputation is needed per considered value of the angle tolerance (τ, $\tau/2$ and $\tau/4$).



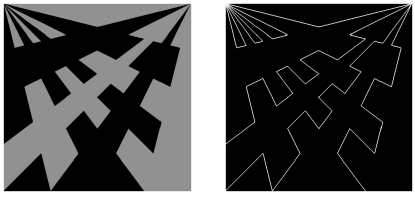
(a) contrast=1.2 (b) contrast=1.3 (c) contrast=1.4 (d) contrast=1.5

Fig. 2: ROC curves [28, 29] obtained using GR as an edge detector over several 1-look images with different edge contrasts values, and different values of α.

proper value for α should rely on a reasonable trade-off between these two effects. We will evaluate the influence of α under two aspects: on the one hand, α must be set in such a way that GR efficiently highlights the edges of the input image. On the other hand, the amount of dependencies between local orientations must be correctly taken into account using our Markov Chain approach, which can be implicitly checked by looking whether or not, our modified NFA$'$ formula (12) provides an effective control of the number of false detections.

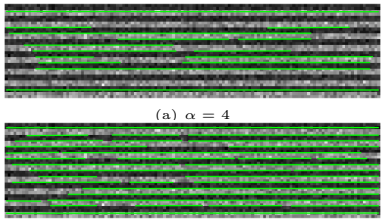
In order to evaluate the performances of GR in terms of edge detection, we used the ROC-curve strategy described in [28, 29]. Those ROC curves are obtained by evaluating the number of true positive and false positive edge pixels detected by thresholding the GR magnitude with a threshold λ. The ROC curve is obtained by varying the λ parameter from the minimal to the maximal gradient amplitude value. In Fig. 2, we display the ROC curves computed over 1-look synthetic edge images with different contrast values and with sizes 512×512 pixels. The reference image corresponding to the synthetic edge with contrast 1.5 is displayed in Fig. 3 (a), and the associated ground truth is displayed in Fig. 3 (b).

From Fig. 2 we can see that, when $\alpha = 4$, GR gives the best performance since the corresponding ROC curve remains above those obtained with lower values of α. On the other hand, Table 2 gives the number of false detections obtained by applying Algorithm 1 on a 1-look pure Nakagami noise of size 4096×4096 pixels with $\varepsilon = 1$ and for different values of α. We can see in Table 2



(a) clean image (b) edge ground truth

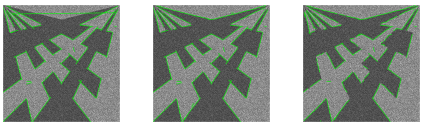
Fig. 3: (a) synthetic image with a contrast amplitude of 1.5, (b) the corresponding edge ground truth.



(a) $\alpha = 4$

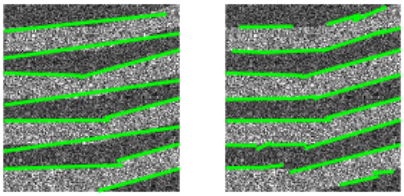
(b) $\alpha = 2$

Fig. 4: Line segment detection on 20-looks synthetic image of size 32×128 pixels containing horizontal edges with 2-pixel width and amplitude contrast of 1.8.



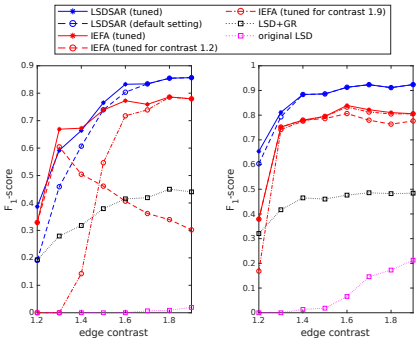
(a) $D = 0$ (b) $D = 0.4$ (c) $D = 0.5$

Fig. 5: Line segment detection on 3-look synthetic edge image (contrast 1.6) for different value of the density threshold.



(a) $D = 0.4$ (b) $D = 0.6$

Fig. 6: Line segment detection results on 3-look synthetic edge images (contrast 1.6) for two different values of D. The angle difference of the problematic line segments is 16 degrees, while the angle tolerance is 22.5 degrees.



(a) 1-look images (b) 3-look images

Fig. 7: Performances evaluations in terms of F_1-score for different algorithms, using 1-look (a) and 3-look (b) synthetic edge images with contrast values ranging from 1.2 to 1.9.

that the proposed Markov chain model succeeds in controlling the number of false detections in the case $\alpha = 1$ and $\alpha = 2$, but as α becomes larger, the number of false detections may be significantly above ε, especially when $\alpha = 5$. In the case $\alpha = 4$, the number of false detections remains reasonably comparable to ε, and considering that this value gives the best performance on the ROC curves displayed in Fig. 2, we suggest to use $\alpha = 4$ as a default setting. However, one must be aware that, with the setting $\alpha = 4$, LSDSAR may not be able to distinguish line segments that are very close to each other, as illustrated in Fig. 4. In situations where the accurate detection of close line segments is required, a smaller value, such as $\alpha = 2$, can be used.

Now, let us focus on the setting of D, the threshold for the density of aligned pixels involved in LSDSAR. The density of aligned pixels within rectangle r is defined as the ratio $k(r)/(w(r) \cdot \ell(r))$, noting $w(r)$ and $\ell(r)$ the width and length of r. In Algorithm 1, after the region growing step, the rectangles are refined until their density of aligned pixels becomes larger than D. As explained in [5], this threshold aims to avoid situations where two straight edges are present in the region with an angle between them smaller than the tolerance τ, leading to inconsistent detections. Again, a tradeoff must be found, since low values of D lead to inconsistent detections, while too large values of D will have the effect of over-cutting the line segments into small subsegments. This phenomenon is illustrated in Fig. 5, where we can see that one spurious line segment appears when $D = 0$, line segments are fragmented when $D = 0.5$, while a better satisfying detection is obtained with $D = 0.4$, that we recommend as default setting. Although we observed that the setting $D = 0.4$ is generally safe, larger values may sometimes be needed, as illustrated in Fig. 6 which shows that a larger density threshold may be necessary when we need to distinguish two line segments with an angle difference significantly smaller than the angle tolerance τ.

4. Experiments

In this Section, we study the performances of the proposed LSDSAR algorithm over synthetic and real SAR images, and compare this algorithm with the following concurrent methods:

a. the original LSD algorithm applied to the logarithm of the amplitude of the SAR image;

b. what we refer to as the LSD+GR algorithm, which roughly corresponds to the trivial adaptation of LSD where the finite-difference based gradient suited to optical images is replaced by GR. More precisely, our LSD+GR implementation is a variant of Algorithm 1 where we avoid the use of Markov chains, and use the usual NFA formula (1) instead of NFA' as defined in formula (12);

c. a state-of-the-art Hough-transform based method, named IEFA, recently proposed in [17]. IEFA relies on a ratio-based gradient [30] and a threshold to suppress the pixels with small gradient magnitude. Then, non-maximum suppression [31, 32, 33] is used to extract the local maxima of the magnitude field. Instead of using a binary edge map as the input of the Hough transform, the gradient magnitude field is selected as the input. The accumulation weight in the Hough transform is the gradient magnitude weighted by the gradient orientation with a Gaussian function, as described in [17, 34, 35].

Unless explicitly mentioned, we will use the default setting $\varepsilon = 1$, $\alpha = 4$, $\tau = 22.5°$, and $D = 0.4$ in our LSDSAR algorithm. In particular, we will show that LSDSAR performances are robust with respect to the parameter settings, and that the proposed default parameters yields essentially near optimal performances in most situations. In the LSD+GR variant, we set $\alpha = 4$ and keep the other parts of the LSD algorithm unchanged. We also keep the default setting of the original LSD [22] algorithm unchanged. In the case of IEFA, which relies

on many parameters, we will use a specific parameter tuning procedure in our numerical experiments. This procedure will be detailed below.

4.1. Ability to control the NFA

One of the requirements of the proposed LSDSAR algorithm is its ability to control the number of false detections. As discussed before, the filtering provided by GR introduces important dependencies between local orientations, so that the LSD+GR approach does not fulfill the independence assumption required by the original LSD algorithm. This phenomenon is clearly demonstrated in Table 3 (column 2), where we indicate, for several values of α, the number of false detections obtained using LSD+GR over a pure speckle image of size 4096×4096 pixels. On the contrary, we see in Table 3 (column 3) that, thanks to the Markov chain modeling of weak dependencies between local orientations, the number of false detections obtained with LSDSAR is better controlled and remains comparable to ε, even for large values of α. The ability of LSDSAR to provide a reasonable control of false detections is further confirmed in Table 4, for different values of the NFA threshold ε, where we can see that the average number of false detections done in pure speckle noise images is roughly comparable to ε.

4.2. Detection performances on synthetic data

In order to provide a quantitative evaluation of the detection quality of the considered algorithms, we propose to use the F_1-score, which is defined as

$$F_1\text{-score} = 2 \cdot \frac{\mathit{precision} \times \mathit{recall}}{\mathit{precision} + \mathit{recall}},$$

where *precision* denotes the ratio between the number of pixels correctly detected as line segments and the total number of pixels detected as line segment. Thus, $1 - \mathit{precision}$ measures the proportion of false positive pixels among detected pixels. *Recall* denotes the ratio between the number of pixels correctly detected as line segments and the actual number of pixels corresponding to line segments and $1 - \mathit{recall}$ measures the proportion of false negative pixels among

actual line segment pixels. Those quantitative measures of F_1-score will be performed over some noisy single or multilook observation of synthetic images for which a ground-truth is available (see Fig. 3).

We computed the performances in terms of F_1-score for LSDSAR, IEFA, LSD+GR and LSD, over 1-look and 3-look synthetic edge images with different contrast values. In the case of IEFA, the algorithm was tested with a large range of parameters. More precisely, the threshold that we used for the gradient magnitude ranges from 0.1 to 0.40 with step 0.05. The number of peaks in the Hough transform was set to 5000 so that all the line segments on the image could be detected. The smallest value of a bin which can be considered as a peak was set to 3. The minimum gap between two line segments extracted from the same bin ranges from 3 to 15 with step 2. The minimum length of a line segment was set to 13. In Fig. 7, the red plain curve (referred to as *IEFA (tuned)* in the legend) represents the best F_1-score value obtained among all tested parameters for each contrast value. Therefore, this curve represents the best F_1-score performance that can be achieved by IEFA. Now, let us stress that the parameters leading to this optimal performances are different from one contrast value to the other. This observation is confirmed by the red dashed curves of Fig. 7, where we used the optimal parameter setting found for contrast 1.2 and 1.9 to process the images with other contrast values (see the curves referred to as *IEFA (tuned for contrast 1.2)* ans *IEFA (tuned for contrast 1.9)*). One can see that the performances of IEFA can be very different from the optimal performances when the input parameters are fixed (and even optimized for a given contrast value). In the case of LSDSAR, except for the default parameter settings, we also tuned the parameters to demonstrate that the F_1-score obtained by LSDSAR can be improved. The value of ε ranges over 1, 10^2, 10^4 and 10^6. The value of α ranges from 4 to 7 with step 1. The value of τ can be 22° or 33.75°. We keep the parameter $D = 0.4$ unchanged. LSDSAR with tuned parameter values leads to the blue plain curve, referred as *LSDSAR (tuned)* in the legend of Fig. 7. This optimal F_1-score can be compared to the F_1-score obtained using the default setting of LSDSAR (blue dashed curve)

that we recommend in this paper ($\varepsilon = 1$, $\alpha = 4$, $\tau = 22°$ and $D = 0.4$). We can see that the default recommended setting for LSDSAR leads to F_1-score performances that are similar to that obtained when tuning the parameters of LSDSAR, which demonstrates the robustness of the algorithm with respects to the setting of its parameters. Besides, we can see that, in most situations, the F_1-score achieved by LSDSAR with default parameter setting is comparable or better than the F_1-score achieved by IEFA with tuned parameters, and also significantly better than the F_1-score obtained using LSD+GR and the original LSD. Indeed, in the 3-look experiments displayed in Fig. 7 (b), the performance of LSDSAR is clearly above that of IEFA. Some of the images corresponding to this experiments are displayed in Fig. 8. In the 1-look experiment displayed in Fig. 7 (b), IEFA with tuned parameters may lead to slightly better results than LSDSAR in low contrast situations (in fact, this is only clear for contrast 1.3), but it should be noted that, in this particular case, the IEFA algorithm's performance is very dependent on the parameter setting and may collapse if the parameters are slightly changed. Last, it must be noted that, in the case of IEFA, multiple responses may be obtained for the same line segment, which may be an issue for practical applications, while it is not the case for the LSDSAR algorithm.

4.3. Stability comparison between LSDSAR and IEFA

The stability of a detector, that is, its ability to produce stable performances for different noisy observations of the same image, is important in practical applications. In order to test whether LSDSAR and IEFA yield stable performances, both algorithms were used to process fifty single-look noisy observations of a synthetic edge image with an amplitude edge contrast of 1.6. We display in Table 5 the minimal, maximal, average and median values of F_1-score achieved by LSDSAR (with default parameters) and IEFA (with tuned parameters) for this experiment. We can see from this experiment that the stability of the detection performance provided by both algorithms is similar.

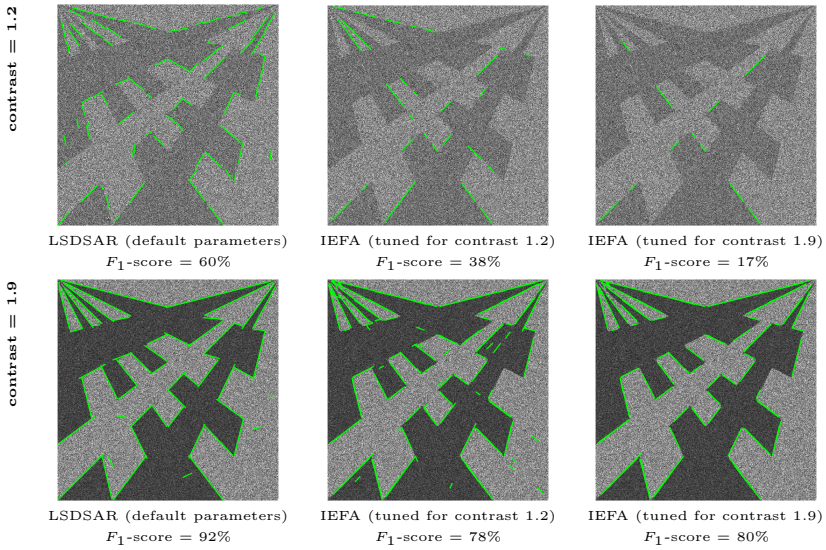
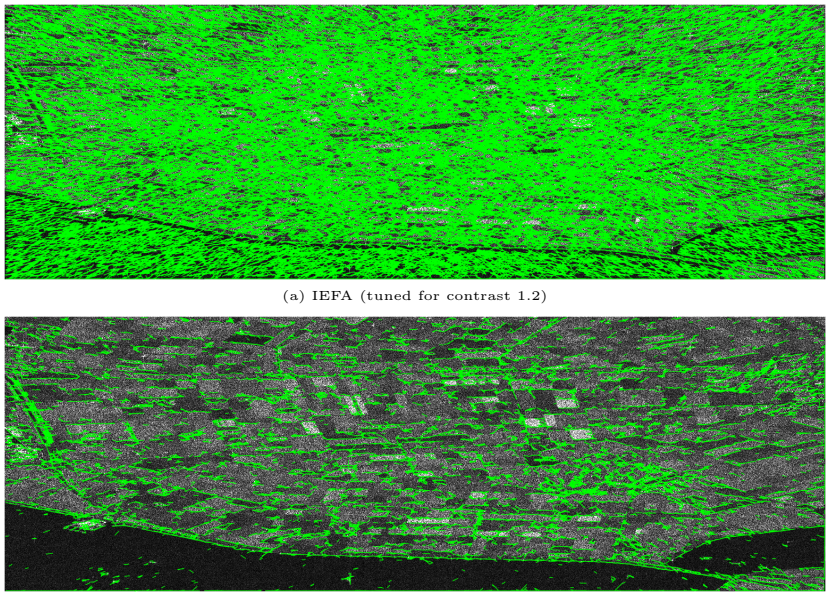


Fig. 8: Detection results obtained using LSDSAR (with default parameters) and IEFA (tuned for contrast 1.2 or 1.9) over 3-look synthetic images with contrast values equal to 1.2 (first row) or 1.9 (second row). We can see that, in both situations, LSDSAR with defaults parameters yields better performances in terms of F_1-score than IEFA (even with tuned parameters).



(a) IEFA (tuned for contrast 1.2)

(b) IEFA (tuned for contrast 1.9)

Fig. 9: Line segment detection obtained using IEFA (tuned for contrast 1.2) and IEFA (tuned for contrast 1.9) over a 1-look real SAR image.

4.4. Comparison between LSDSAR and IEFA on real SAR images

In this section, we focus on the performances of IEFA and LSDSAR on real SAR images. Details about the SAR images that we used are provided in Table 6. We observed that, for IEFA, the parameters yielding the best F_1-scores in the 1-look synthetic experiment were not appropriate to process real 1-look SAR images, as we show in Fig. 9. Again, this illustrates the difficulty of using IEFA in practical applications. In our experiments on real data, we manually tuned the IEFA parameters on each considered image, by means of a visual inspection of the detection result. The parameters of the IEFA algorithm were explored as follows: the threshold of the gradient magnitude was set equal to 0.25 (the gradient magnitude of our SAR images varies from 0 to 0.93), the number of peaks extracted in the Hough transform was set equal to 5000, the smallest value of a bin that can be considered as a peak was set equal to 3, the minimum gap between two line segments extracted from the same bin was set equal to 7 and the smallest length of a line segment was set equal to 13. In Fig. 10 and Fig. 11, we display the line segment detection results obtained using LSDSAR and IEFA over two different single-look SAR images.

From the images displayed in Fig. 10 and Fig. 11, we can see that LSDSAR detects more correct line segments than IEFA while the number of false detections is better controlled. There are nevertheless some line segments detected by IEFA that are not detected by LSDSAR. Better performances could be obtained with LSDSAR by increasing either α or ε, at the cost of increasing the number of false detections. We also performed experiments on real SAR images with better signal-to-noise ratio, by running the two algorithms on the temporal averaging of 3 registered Sentinel images. We can see in Fig. 12 that LSDSAR detects much more correct line segments in the multi-look situations while the number of false detections is well controlled. Again, we must underline that the parameter setting of IEFA is very dependent on the image content and all those experiments involved a careful tuning of its parameters by means of a visual inspection, while LSDSAR always provided satisfying results using its default parameter setting.

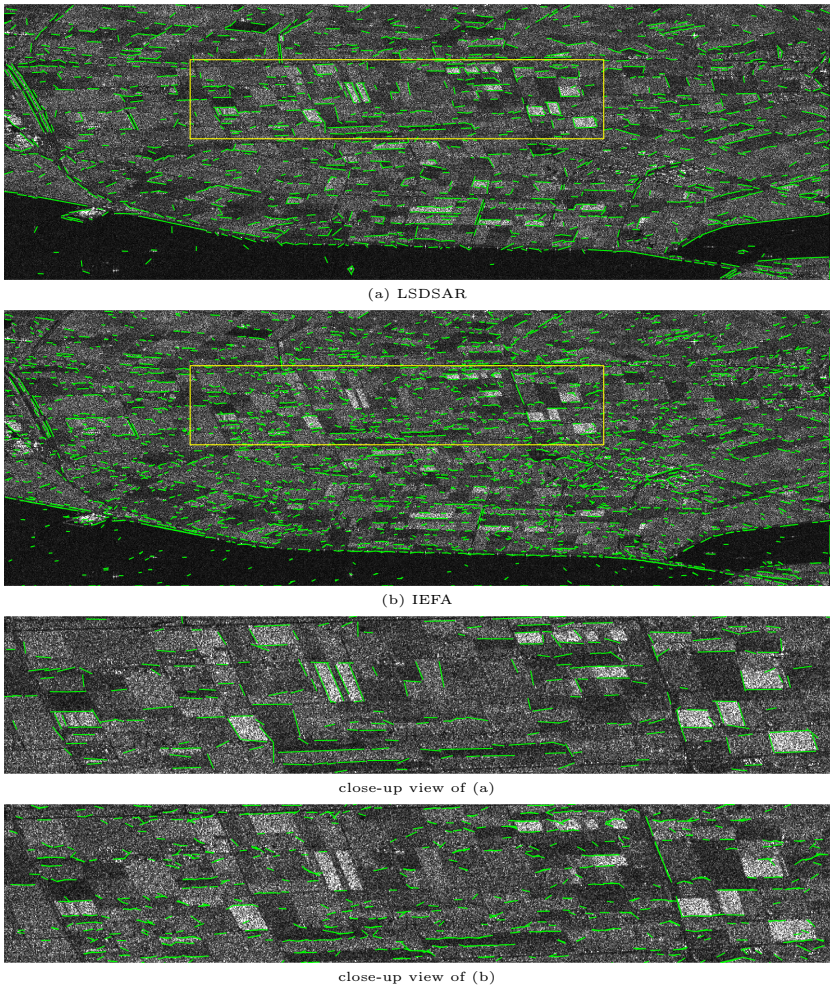


Fig. 10: Line segment detection using LSDSAR (a) and IEFA (b) over a 1-look Sentinel 1 SAR image (Leystad).

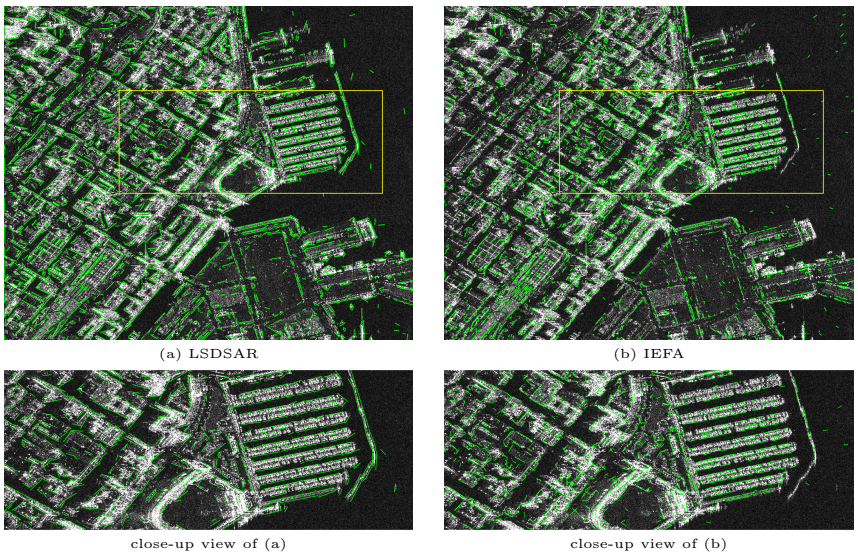
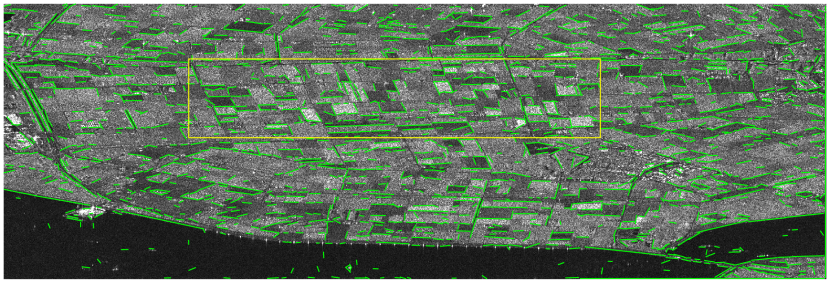


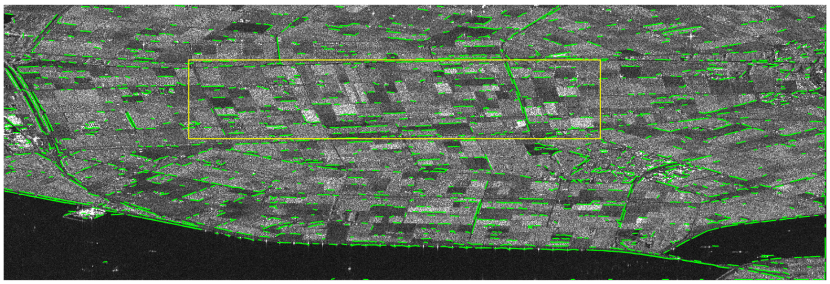
Fig. 11: Line segment detection using LSDSAR (a) and IEFA (b) over a 1-look TerraSAR-X SAR image (San Francisco).

5. Conclusion

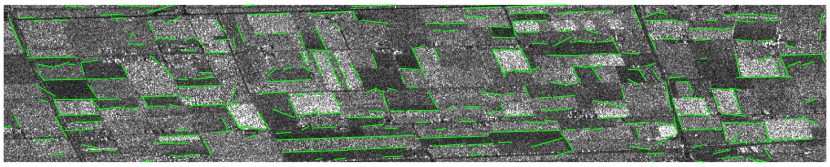
In this paper, we proposed a line segment detector for SAR images, inspired from the LSD detector for optical images. Our experiments on synthetic and real SAR images showed the ability of LSDSAR to detect correct line segments even in 1-look situations while offering a valuable control of the number of false detections. This LSDSAR algorithm only relies on few parameters which are reasonably easy to set. We proposed a default setting that achieved satisfying results in all our experiments. We also demonstrated that a first order Markov chain can be efficiently used to handle the dependencies between the local orientations computed by GR, yielding a reasonably accurate control of the number of false detections for LSDSAR. An interesting perspective for this work would be to consider higher order Markov chains, or two dimensional Markov models, in order to even better take into account all the dependencies and achieve a strict control of the NFA, but in this case the complexity of the algorithm may increase rapidly. More generally, we believe that the use of the first order



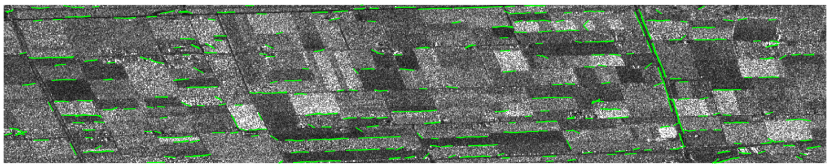
(a) LSDSAR



(b) IEFA



close-up view of (a)



close-up view of (b)

Fig. 12: Line segment detection using LSDSAR (a) and IEFA (b) over a 3-look Sentinel 1 SAR image (Leystad).

Markov chains is a simple and effective way to take into account of the dependencies in the *a contrario* framework, and that it is interesting to study how such an approach can be further used to extend the *a contrario* methodology to practical situations where dependencies are difficult to handle.

Funding

This work is supported by the China Scholarship Council (File No. 201606270202) and the National Natural Science Foundation of China (NSFC) (Grant No.61771014).

Source Code

The source code of LSDSAR is available at : `https://perso.telecom-paristech.fr/cheliu/co_sou_LSDSAR/`.

References

[1] O. Hellwich, H. Mayer, Extracting line features from synthetic aperture radar (SAR) scenes using a Markov random field model, in: Proceedings of 3rd IEEE International Conference on Image Processing, Vol. 3, 1996, pp. 883–886 (1996).

[2] F. Tupin, H. Maître, J.-F. Mangin, J.-M. Nicolas, E. Pechersky, Detection of linear features in SAR images: application to road network, IEEE Transactions on Geoscience and Remote Sensing 36 (1998) 434–453 (1998). doi:10.1109/36.662728.

[3] J. Chanussot, G. Mauris, P. Lambert, Fuzzy Fusion Techniques for Linear Features Detection in Multitemporal SAR Images, IEEE Transactions on Geoscience and Remote sensing 37 (1999) 1292–1305 (1999).

[4] F. Medeiros, R. Costa, R. Marques, C. Laprano, Multiscale detection of linear features in speckled imagery, in: 16th Brazilian Symposium on Computer Graphics and Image Processing 2003, 2003, pp. 371–375 (2003).

[5] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: A Fast Line Segment Detector with a False Detection Control, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 722–732 (2010). doi:10.1109/TPAMI.2008.300.

[6] R. Touzi, A. Lopès, P. Bousquet, A statistical and geometrical edge detection for SAR images, IEEE Transactions on Geoscience and Remote Sensing 26 (1988) 764–773 (1988). doi:10.1109/36.7708.

[7] R. O. Duda, P. E. Hart, Use of the Hough transformation to detect lines and curves in pictures, Communications of the ACM 15 (1972) 11–15 (1972).

[8] J. Illingworth, J. Kittler, The adaptive Hough Transform, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-9 (1987) 690–698 (1987).

[9] J. Skingley, A. Rye, The Hough transform applied to SAR images for thin line detection, Pattern Recognition Letters 6 (1987) 61–67 (1987).

[10] F. Xu, Y. Jin, Automatic Reconstruction of Building Objects from Multiaspect Meter-Resolution SAR Images, IEEE Transactions on Geoscience and Remote Sensing 45 (2007) 2336–2353 (2007).

[11] H. Sui, C. Xu, J. Liu, F. Hua, Automatic Optical-to-SAR Image Registration by Iterative Line Extraction and Voronoi Integrated Spectral Point Matching, IEEE Transactions on Geoscience and Remote Sensing 53 (2015) 6058–6072 (2015).

[12] C. Palmann, S. Mavromatis, J. Sequeira, SAR image registration using a new approach based on the generalized Hough transform, in: ISPRS 2008, Vol. XXXVII, 2008 (2008).

[13] A. Arnold-Bos, A. Khenchaf, A. Martin, et al., An evaluation of current ship wake detection algorithms in SAR images, Caractérisation du milieu marin, Brest, France (2006).

[14] P. Kovesi, et al., Symmetry and asymmetry from local phase, in: Tenth Australian joint conference on artificial intelligence, Vol. 190, Citeseer, 1997, pp. 2–4 (1997).

[15] R. Fjørtoft, A. Lopès, P. Marthon, E. Cubero-Castan, An optimal multiedge detector for SAR image segmentation, IEEE Transactions on Geoscience and Remote Sensing 36 (1998) 793–802 (1998). doi:10.1109/36.673672.

[16] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, F. Tupin, SAR-SIFT: A SIFT-Like Algorithm for SAR Images, IEEE Transactions on Geoscience and Remote Sensing 53 (2015) 453–466 (2015). doi:10.1109/TGRS.2014.2323552.

[17] Q. Wei, D. Feng, Extracting Line Features in SAR Images Through Image Edge Fields, IEEE Geoscience and Remote Sensing Letters 13 (2016) 540–544 (2016).

[18] Q. Wei, D. Feng, W. Zheng, J. Zheng, Rapid Line-Extraction Method for SAR Images Based on Edge-Field, IEEE Geoscience and Remote Sensing Letters 14 (2017) 1865–1869 (2017).

[19] A. Desolneux, L. Moisan, J.-M. Morel, Meaningful Alignments, International Journal of Computer Vision 40 (2000) 7–23 (2000).

[20] A. Desolneux, L. Moisan, J.-M. Morel, From Gestalt Theory to Image Analysis. A Probabilistic Approach, Springer, 2008 (2008).

[21] A. Robin, L. Moisan, S. Le Hegarat-Mascle, An a-contrario approach for subpixel change detection in satellite imagery, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (11) (2010) 1977–1993 (2010).

[22] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, G. Randall, LSD: a Line Segment Detector, Image Processing On Line 2 (2012) 35–55 (2012).

[23] J. B. Burns, A. R. Hanson, E. M. Riseman, Extracting straight lines, IEEE Transactions on Pattern Analysis and Machine Intelligence (4) (1986) 425–455 (1986).

[24] J. Goodman, Statistical properties of laser speckle patterns, Vol. ch. 2, Laser Speckle and Related Phenomena, 1975 (1975).

[25] E. Jakeman, R. Tough, Generalized K distribution: a statistical model for weak scattering, Journal of the optical Society of America 4 (1987) 1764–1772 (1987).

[26] A. Myaskouvskey, Y. Gousseau, M. Lindenbaum, Beyond Independence: An Extension of the *a Contrario* Decision, International journal of Computer Vision 101 (2013) 22–44 (2013). doi:10.1007/s11263-012-0543-6.

[27] B. Grosjean, L. Moisan, A-contrario detectability of spots in textured backgrounds, Journal of Mathematical Imaging and Vision 33 (3) (2009) 313–337 (2009).

[28] S. Dougherty, K. W. Bozyer, C. Kranenburg, ROC Curves Evaluation of Edge Detector Performance, in: Proceedings 1998 International Conference on Image Processing, Vol. 2, 1998, pp. 525–529 (1998).

[29] K. Bowyer, C. Kranenburg, S. Dougherty, Edge detector evaluation using empirical ROC curves, in: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 1999, p. 359 (1999).

[30] P.-L. Shui, D. Cheng, Edge Detector of SAR images Using Gaussian-Gamma-Shapped Bi-Windows, IEEE Geoscience and Remote Sensing Letters 9 (2012) 846–850 (2012). doi:10.1109/LGRS.2012.2184521.

[31] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8 (1986) 679–698 (1986).

[32] W. Jiang, K.-M. Lam, T. zhi Shen, Efficient Edge Detection Using Simplified Gabor Wavelets, IEEE Transactions on Systems, Man, and Cybernetics 39 (2009) 1036–1047 (2009).

[33] S. Yi, D. Labate, G. R. Easley, H. Krim, A Shearlet Approach to Edge Analysis and Detection, IEEE Transactions on Image Processing 18 (2009) 929–941 (2009).

[34] F. O'Gorman, M. Clowes, Finding Picture Edges Through Collinearity of Feature Points, IEEE Transactions on Computers C-25 (1976) 449–456 (1976).

[35] A. Bonci, T. Leo, S. Longhi, A Bayesian Approach to the Hough Transform For Line Detection, IEEE Transactions on Systems, Man, and Cybernetics 35 (2005) 945–955 (2005).

Chenguang Liu received the Bachelor degree in the School of Electronic and Information in Wuhan University, China in 2014 and he is currently working toward a Ph.D degree in Télécom ParisTech, Paris, France.

Rémy Abergel received the master degree from the French Institut Supérieur de l'Aéronautique et de l'Espace, France in 2012, and the Ph.D degree in applied mathematics from the Université Paris Descartes, France in 2016. He is currently a permanent Research Engineer at CNRS/MAP5 Laboratory in Université Paris Descartes, France.

Yann Gousseau received the Engineer degree from the École Centrale de Paris in 1995 and the Ph.D degree in applied mathematics from the University of Paris Dauphine, Paris, France in 2000. He is currently a professor in Télécom ParisTech, France.

Florence Tupin received the Engineer degree and the Ph.D degree in signal and image processing from the École Nationale Supérieure des Télécommunications, Paris, Frnace, in 1994 and 1997. respectively. She is currently a professor in Télécom ParisTech, France.