A novel Pixel Orientation Estimation based line segment detection framework, and its applications to SAR images

Chenguang Liu, Cuiling Liu, Chisheng Wang, Wu Zhu, Qingquan Li

A novel Pixel Orientation Estimation based line segment detection framework, and its applications to SAR images

Chenguang Liu,Cuiling Liu, Chisheng Wang, Wu Zhu, Qingquan Li

***Abstract*—In this paper we propose a new line segment detection framework based on a novel Pixel Orientation Estimation (POE) method, which detects line segments from binary edge maps and can be combined with any edge detectors. We show its efficiency by testing it in 1-look SAR images. The novel Pixel Orientation Estimation method estimates the orientation of each edge pixel by counting the number of edge pixels along a set of orientations. As the most edge pixels exist along the orientation of the line segment, the orientation of the edge pixel is given by the orientation that gives the maximum number of counts. Counting the number of edge pixels along different orientations is equivalent to convolving the local neighbourhood of an edge pixel with a set of carefully designed window functions with each window function corresponding to a fixed orientation. With the estimated orientations of pixels in the edge map, pixels can be grouped with a region growing step to form line support regions. Regions with their size larger than a size threshold will be accepted. Finally, rectangles are used to approximate the accepted regions and those rectangles are detected line segments. Experiments in both simulated SAR dataset and three 1-look Sentinel-1 images demonstrate the efficiency of the proposed method. In particular, we advance the state-of-the-art performances by 18 percent (F1-score) on the 1-look dataset simulated from YorkUrban-LineSegment Dataset.**

***Index Terms*—Pixel Orientation Estimation (POE), line segment detection framework, binary edge maps, 1-look SAR images.**

I. Introduction

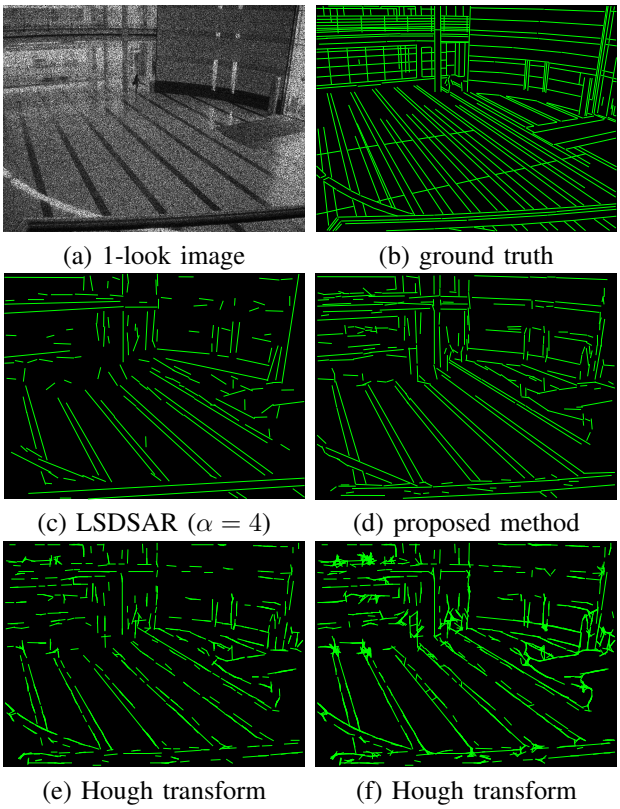
LINE segments, which we define as straight parts of edges here, are important features of images. Line segment detection has been a long standing yet challenging problem in computer vision. Many methods have been proposed for line segment detection in optical images, including traditional detectors [1]–[6] and deep learning based methods [7]–[11], while there are much fewer line segment detectors dedicated to SAR images, mostly because of the strong multiplicative noise. Due to the strong differences in the statistics between optical and SAR images, methods developed for optical images are not suitable for detecting line segments in SAR images.

Chenguang Liu, Cuiling Liu, Chisheng Wang and Qingquan Li are with Shenzhen University, Shenzhen City, Guangdong Province, China (email: chenguangl@whu.edu.cn;sherwoodwang88@gmail.com). Wu Zhu is with Chang'an university, Xi'an City, Shanxi Province, China. This research is funded by Chang'an University (Xi'an, China) through the National Key Research and Development Program of China (2020YFC1512001), the National Natural Science Foundation of China (41974006), the Shenzhen Scientific Research and Development Funding Program (KQJSCX20180328093453763 and JCYJ20180305125101282), the Research Program from the Department of Education of Guangdong Province (grant No. 2018KTSCX196), and the National Natural Science Foundation of China (NSFC) (Grant No.61771014).

The classical ways to detect line segments in SAR images are based on Hough transform [1], [3], [12]–[16]. A ratio operator [17], [18] is first applied to detect the edges in SAR images, and then Hough transform is applied to detect lines in SAR images. Finally, line segments are obtained with a post-processing step. More specifically, Hough transform tries to find the cues about the existence of a line segment by transforming the pixel coordinates into a parameter space. According to Hough transform, a line in the image space corresponds to a point in the parameter space. Therefore, the local maxima in the parameter space correspond to candidate lines. The maximum number of lines to be detected has to be set beforehand for Hough transform based methods. In addition, local maxima with their number of pixels smaller than a given threshold will be discarded. Besides, two more thresholds are required to find the end points of line segments and set the length of the shortest line segment that is allowed. These thresholds usually have to be tuned according to the images at hand and it is difficult to find a set of threshold values that works globally. It can be seen from Figure 1-(e) and Figure 1-(f) that when the parameter settings change, line segment detection results computed by Hough transform may be quite different. Except for the threshold problem, Hough transform based methods may suffer from the problem of multiple responses due to the digitization of pixel coordinates in images. This phenomenon is shown in Figure 1-(e) and Figure 1-(f) that multiple line segments are detected from the same location. The performances of Hough transform based method can be improved by using the gradient magnitude field as input rather than using binary edge maps [19], [20], but the drawbacks of Hough transform remain.

A recent line segment detector for SAR images, which is named as LSDSAR [21], [22], has been shown to be able to detect line segments with a number of false detection control. An appealing point of this method is that it requires few parameter tunings. LSDSAR is based on a Markovian $a\ contrario$ model and the Helmholtz principle [23]. It first computes the gradient orientation of each pixel with a ratio based gradient computation method [24], and then groups pixels sharing the same gradient orientation to get candidate line segments. Candidate line segments are finally validated by measuring the unlikeliness of a similar segment to appear by chance in a random image. The more unlikely a line segment is to appear by chance in a random image, the more likely it is to be a meaningful detection [25]. Even though LSDSAR has been proven to be effective to detect line segments in

SAR images, its performances in 1-look SAR images are not satisfying enough. It can be seen from Figure 1-(c) that many true line segments are missing for LSDSAR.



(a) 1-look image (b) ground truth

(c) LSDSAR ($\alpha = 4$) (d) proposed method

(e) Hough transform (f) Hough transform

Fig. 1: Line segment detection results obtained by LSDSAR, Hough transform based methods and our POE based method in a 1-look simulated SAR image. The size of the image is 480×640 pixels. The same edge detector is used for the Hough transform based method and our POE based method. For (e) and (f), the parameter settings for Hough transform are different.

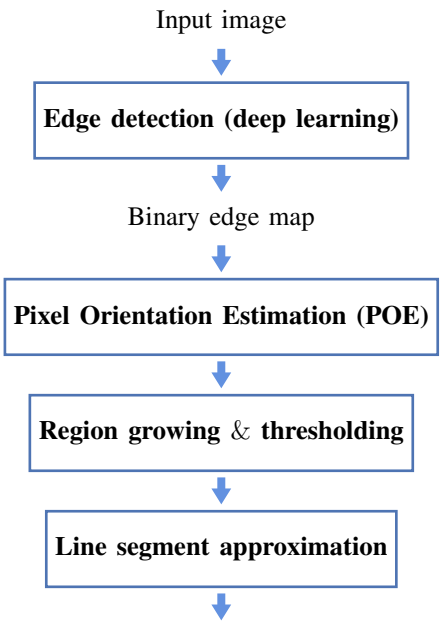
On the other hand, deep learning techniques have been successfully used for edge detection in SAR images. A deep learning based edge detector named GRHED [26] has shown to be much more efficient for detecting edges in 1-look real SAR images while the performances of it are much less sensitive to the choice of parameter values (compared to the traditional ratio based edge detectors for SAR images). GRHED is an edge detection framework constituted by a hand-crafted layer and fully convolutional layers, enabling models trained using datasets simulated from optical datasets to be efficient for edge detection in SAR images. Motivated by the success of applying deep learning based methods for edge detection in 1-look real SAR images, we propose a new line segment detection framework, which detects line segments from binary edge maps, to improve line segment detection results in SAR images.

All line segment detectors need to group edge pixels belonging to the same line segment. Hough transform based methods transform the coordinates of edge pixels into a parameter space to find the cues for the existence of line segments [1], [3]. In [25], the authors first compute the gradient orientation of each pixel, and then search for meaningful line segments from all lines starting from the sides of images. Methods based on perceptual grouping typically used the gradient orientation of pixels [2], [4], [21] by grouping pixels sharing the same gradient orientation to generate candidate line segments. In [6], edge pixels along the same line segment are grouped by exploiting the intrinsic property of line segments in image space, where line segments are represented by a set of linelets (a linelet is a set of either horizontally or vertically connected pixels). Despite the existence of these edge grouping methods, they can not be used to detect line segments from the results computed by deep learning based edge detectors except for Hough transform. Gradient orientations of pixels are used during either edge grouping or line segment validation process [2], [4], [6], [21], [25] while deep learning based edge detector [26]–[28] computes only the edge strength map without the edge orientation information. Hough transform seems to be the best choice to detect line segments from binary edge maps. (we do not discuss on those deep learning based line segment detectors for two reasons: first, to the best of our knowledge, there is still no real SAR dataset for line segment detection in SAR images; second, most deep learning based line segment detectors combined the detection of line segments and junctions generating candidate line segments by connecting two junctions [9]–[11]. Therefore, the strategies used in deep learning based methods are not suitable for our situation).

Since Hough transform based methods suffer from the problem of parameter tuning and multiple responses, we propose a brand-new edge grouping method based on a Pixel Orientation Estimation (POE) method, which contributes to a line segment detection framework that can be combined with any edge detectors. More precisely, the orientation of edge pixels can be obtained by counting the number of edge pixels along a set of orientations. The orientation of the edge pixel is then given by the orientation that gives the maximum number of counts as most edge pixels exist along the line segment. Counting the number of edge pixels along different orientations is equivalent to convolving the local neighbourhood centering at the pixel with a series of circle-shaped window functions (with each window function corresponding to a fixed orientation). By estimating the orientation of each edge pixel, we are able to obtain a set of line support regions with a region growing algorithm by grouping pixels having the same orientation. Generated regions will be checked by comparing their size with a threshold based on the Helmholtz principle [23]. Rectangles are finally used to describe accepted regions and those rectangles are detected line segments. The pipeline of the proposed line segment detection framework is shown in Figure 2.

The contributions of this paper can be summarized as follows:

- we propose a brand-new edge grouping method based on a novel Pixel Orientation Estimation (POE) method, which enables us to generate line segments by grouping edge pixels with the estimated orientations. The resulting line segment detection framework requires few parameter



Overview of the line segment detection framework

Fig. 2: Framework of the method

tunings;

- the proposed line segment detection framework can be easily combined with any edge detectors, which means that any improvements in the performances of edge detectors will contribute to a better line segment detector;
- the proposed line segment detection framework advances the state-of-the-art performances by a large margin in 1-look SAR images.

The rest of the paper is organized as follows: in Section II, we briefly introduce the edge detectors we use to detect edges of 1-look SAR images; in Section III, we will give details about our novel Pixel Orientation Estimation (POE) method, and introduce the region growing, region size thresholding, as well as the line segment approximation step; experiments in both 1-look simulated SAR dataset and three 1-look real SAR images will be provided in Section IV to demonstrate the efficiency of the proposed method; some conclusions will finally be given in Section V.

II. Edge detection using deep learning based edge detectors

In this paper, we propose to follow the traditional fashion of detecting line segments in images by grouping edge pixels, based on a novel Pixel Orientation Estimation method. This is mainly to fully exploit the success of deep learning based method for edge detection in SAR images [26]. In [26], the authors proposed a framework composed of a hand-crafted layer and fully convolutional layers, which aims at enabling models trained using simulated dataset to work in real SAR images. In the absence of a real SAR dataset for the task of edge detection, GRHED [26] proposed a solution to the application of deep learning models to real SAR images.

Contrarily to the traditional fully learnable deep models, there is a hand-crafted layer before learnable layers in GRHED. The hand-crafted layer, which is defined by a ratio based gradient computation method GR [24], ensures that the distributions of feature maps computed by GR in simulated SAR and real SAR images are similar. Thus, the learnable layers trained on the feature maps of simulated SAR images will be able to perform efficient edge detection in the feature maps computed by GR in real SAR images.

In GRHED [26], the convolutional layers are identical to those in HED [27], [29] and this is the reason why it is called GRHED. However, GRHED is actually a framework, both the ratio operator (GR) and convolutional layers (HED) are replaceable (we can replace GR with any other ratio operators [17], [18] and replace HED layers with other convolutional layers). In this work, we also replace the convolutional layers with those of RCF [28] and keep the ratio operator GR intact. We call it as GRRCF. During training, the loss function of GRHED is identical to that in HED [27] and the loss function of GRRCF is identical to the one in RCF [28].

For the training of GRHED and GRRCF, we use two datasets. The first one is the same as the one used in [26], which is simulated by multiplying each image in the augmented BSDS500 [27], [30] with 1-look noise (it contains 28800 images for training and validation). We train GRHED on this dataset and keep all the settings for training in [26], except that we set the training iterations to be 30000 (more details about the training settings can be found in [26]). GRHED trained with this dataset is called GRHED. The other dataset is simulated from a dataset that mixes the augmented BSDS500 with the flipped VOC Context dataset [28], [31]–[33] (it contains 49000 images in total), by multiplying each image with 1-look noise. Both GRHED and GRRCF will be trained using this dataset and we call them as GRHED-aug and GRRCF-aug respectively. The feature maps computed by GR with multiple α values ($\alpha = 2, 3, 4, 5$, α is the exponential weight parameter used in the ratio operator GR) will be combined and used as the input of learnable layers, as done in [26].

Different from GRHED, which is implemented in Tensorflow, both GRHED-aug and GRRCF-aug are implemented in PyTorch. GRHED-aug and GRRCF-aug are trained from scratch with Xavier initialization [34]. Both models are trained using Adam optimizer with learning rate 0.0001. The batch size is set to 10 and the weight decay is set to 0.0002. GRHED is trained for 20 epochs and the learning rate will be divided by 10 after 30000 iterations. GRRCF is trained for 10 epochs (we tried to train GRRCF for 20 epochs and divide the learning rate by 10 after 10 epochs, but it does not improve the performances). Non-maxima suppression [35] and plain thresholding are applied to GRHED, GRHED-aug and GRRCF-aug to obtain the binary edge map.

We compare the performances of GRHED, GRHED-aug and GRRCF-aug on the two hundred 1-look images, which are simulated from BSDS500 dataset, in terms of three criteria: the Optimal Dataset Scale (ODS) F1-score (one threshold for all the images), Optimal Image Scale (OIS) F1-score (one threshold for each image) and Average Precision (AP). The

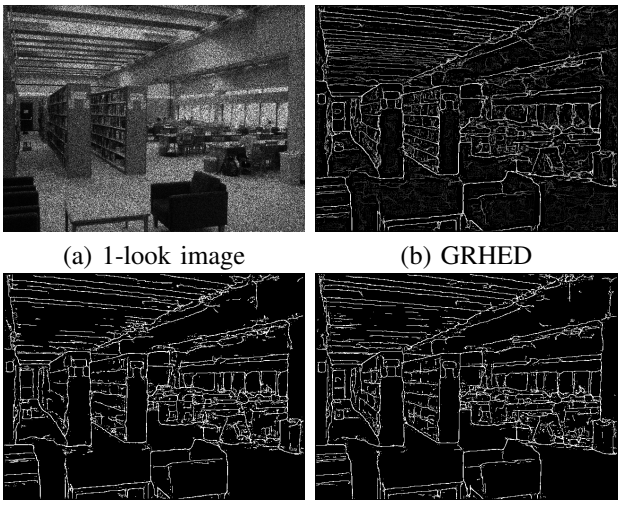
results are shown in Table I.

TABLE I: Comparison of GRHED, GRHED-aug and GRRCF-aug on two hundred 1-look images simulated from the testing images of BSDS500.

Methods	ODS F1	OIS F1	AP
GRHED	0.667	0.685	0.690
GRHED-aug	0.694	0.710	0.724
GRRCF-aug	0.693	0.710	0.716

From Table I for the comparison between GRHED and GRHED-aug we can see that we push forward the ODS F1-score from 0.667 to 0.694 by using more training data. From the comparison between GRHED-aug and GRRCF-aug we can see that for these training settings, they show comparable detection capabilities, even though RCF [28] outperforms HED [29] in BSDS500 dataset (0.806 compared to 0.790). Conclusions may be different if we try more training settings for GRHED-aug and GRRCF-aug, but we leave it for future work as the main contribution of this work is that we propose a new line segment detection framework based on the Pixel Orientation Estimation method, which will enable us to detect line segments from binary edge maps produced by any edge detectors.

Edge maps computed by GRHED, GRHED-aug and GRRCF-aug in a 1-look SAR image (simulated with an image from YorkUrban-LineSegment dataset) are shown in Figure 3. Similar to the observation in Table I, GRHED-aug and GRRCF-aug detects slightly more true edge pixels than GRHED. In this work we will detect line segments from binary edge maps (as those shown in Figure 3) computed by GRHED, GRHED-aug and GRRCF-aug, and compare the performances of the proposed methods with the state-of-the-art line segment detectors for SAR images.



(a) 1-look image (b) GRHED

(c) GRHED-aug (d) GRRCF-aug

Fig. 3: Edge maps computed by GRHED, GRHED-aug and GRRCF-aug in a 1-look simulated SAR image.The size of the image is 480×640 pixels.

III. Line segment detection with a new edge grouping method based on the Pixel Orientation Estimation method

Deep learning based edge detectors compute only binary edge maps, while missing the information of edge orientations. Hough transform is the most commonly used method to detect line segments from binary edge maps. However, the drawbacks of Hough transform make it difficult to be used in general applications. A better edge grouping method is expected to detect line segments from binary edge maps.

Perceptual grouping methods, which typically use the edge orientation to group edge pixels, do not suffer from the problems faced by Hough transform. Candidate line segments are generated by searching and grouping pixels which have the same edge orientation, and the end points of candidate line segments are naturally identified. The aim of this paper is to propose a Pixel Orientation Estimation method which estimates the edge orientation of pixels from binary edge maps. The proposed method should force edge pixels along the same line segment to have the same orientation, and output different angles for pixels belonging to different line segments. Candidate line segments can then be generated through a region growing algorithm. In the following, we will first describe the novel Pixel Orientation Estimation (POE) method we propose in this work, and give details about the subsequent procedures of the line segment detection framework, which includes region growing, region size thresholding and line segment approximation.

A. Pixel Orientation Estimation (POE)

In this work we propose a novel Pixel Orientation Estimation method using contextual information of each pixel in the binary edge map. Consider the binary edge maps shown in the second column of Figure 5, it is natural that the orientation of an edge pixel should be aligned with the orientation of the line segment passing through it. Besides, all edge pixels along a line segment should have the same orientation. Estimating the orientation of an edge pixel can thus be transformed as estimating the orientation of the line segment passing through it. In order to find the orientation of the line segment passing through an edge pixel, we can count the number of edge pixels along a series of orientations. The maximum number of counts should appear when the orientation coincides with the orientation of the line segment. Therefore, the orientation of the line segment passing through an edge pixel can be found by counting the number of edge pixels toward a set of orientations. The orientation of the line segment is given by the orientation that gives the maximum number of counts.

More formally, counting the number of edge pixels along a set of orientations is equivalent to computing the convolution between the local region of an edge pixel with a set of carefully designed window functions. For an edge pixel $u(x, y)$ which locates at (x, y) in the binary edge map (u) of a SAR image, we consider a circle-shaped region which centers at (x, y), and compute the convolution of this region with a set of circle-shaped window functions respectively (with each window function corresponding to a fixed orientation), the

orientation corresponding to the window function which gives the maximum response will be considered as the orientation of the edge pixel.

Assuming that we have P window functions, and the pixel values of these windows are all zero except for one diameter of each window. Pixel values along the exclusive diameter of each window are 1. The angles of these diameters are uniformly distributed over $[0, \pi)$ and the orientation of each window is defined as the orientation of its exclusive diameter. For the i-th window ($i = 1, 2, \cdots, P$, defining the angle of its exclusive diameter θ_i as $\frac{(i-1)*\pi}{P}$), then the convolution between the circle-shaped region centering at pixel (x, y) and the i-th window can be computed as

$$S_i(x, y) = \sum_{x'=-w}^{w} \sum_{y'=-w}^{w} u(x + x', y + y') * f(x', y'), \quad (1)$$

where w is the radius of the circle-shaped region and circle-shaped windows, and

$$f(x', y') = \begin{cases} 1.0 & \text{if } |-x' \sin(\theta_i) + y' \cos(\theta_i)| \\ & < 0.5 \text{ and } x'^2 + y'^2 <= w^2, \\ 0.0 & \text{otherwise.} \end{cases} \quad (2)$$

Assuming that

$$S_j(x, y) = \max_{1 \le i \le P} S_i(x, y), \quad (3)$$

then, the edge orientation of pixel (x, y) is defined as

$$O(x, y) = \frac{(j-1) * \pi}{P}, \quad (4)$$

which corresponds to the direction of the diameter of the j-th window.

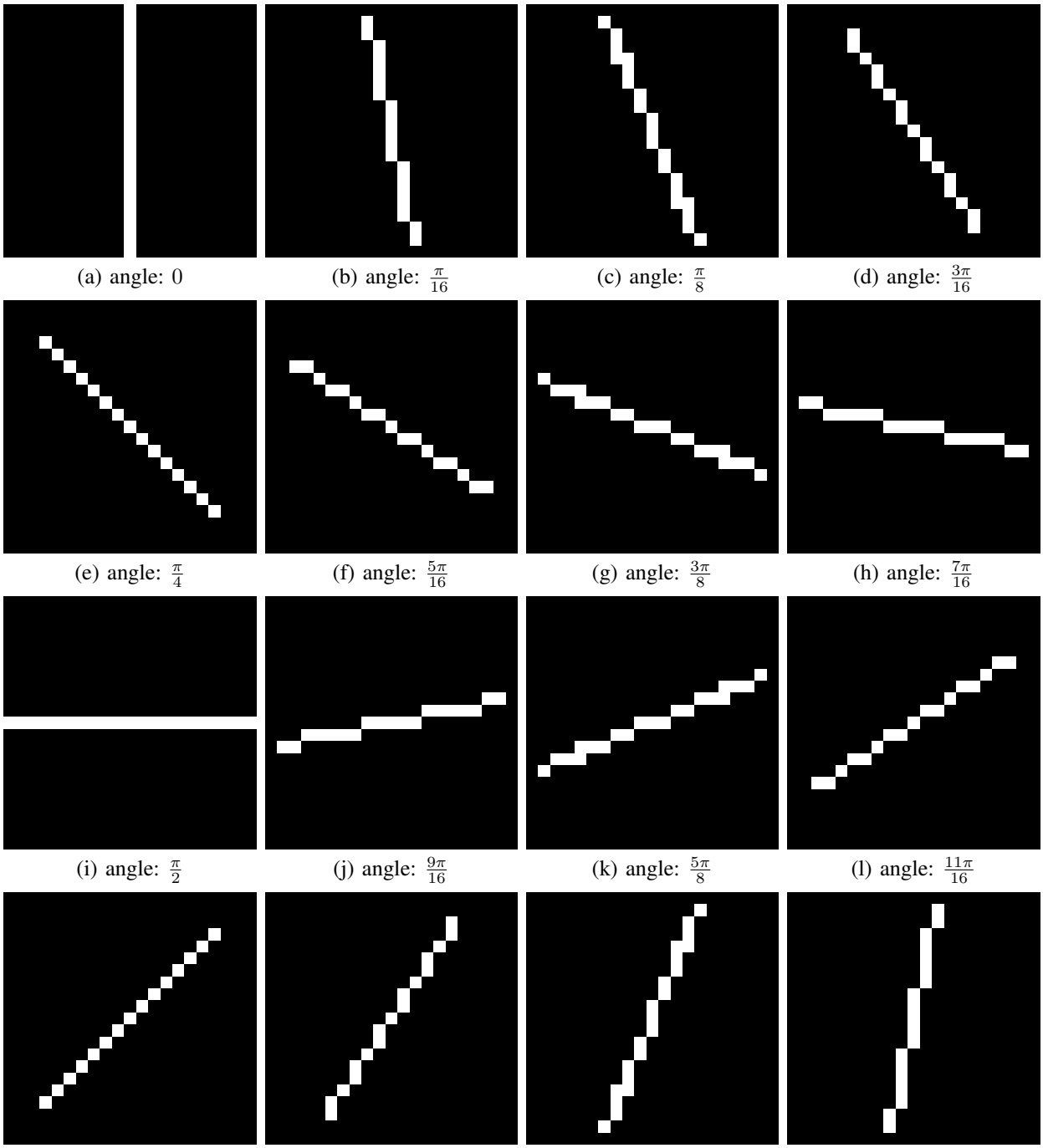
One important point here is that we need to choose an appropriate number of windows (the value of P). Several factors have to be taken into account: first, the number of windows should be large, so that we can distinguish two line segments with close orientations; second, because of the digitization of pixel location in the image space, when the number of directions is too large, two windows with their diameters orienting to two adjacent directions may overlap significantly; third, with the increase of the number of directions, the computation load increases rapidly. In our experiments, we found 16 directions to be a nice choice. It means that for each window, the angle of its diameter is one of the following values: 0, $\frac{\pi}{16}$, $\frac{\pi}{8}$, $\frac{3\pi}{16}$, $\cdots$, $\frac{15\pi}{16}$. The other variable that matters here is the radius value w of these circle-shaped windows. We found the value of 10 appropriate as with a larger radius, the estimation of orientations is more stable for each edge pixel. The settings of these parameter values will be discussed further in the experimental section.

The 16 windows with radius size $w = 10$ are shown in Figure 4. It is clear that convolving the binary edge maps as shown in the second column of Figure 5 with the window functions shown in Figure 4 is equivalent to counting the number of edge pixels along a set of orientations. Estimating the orientation of an edge pixel (or equivalently estimating the orientation of the line segment passing through it) can thus be transformed as finding the direction that gives the maximum response. This new method, which is named as Pixel Orientation Estimation (POE) is designed to estimate the angles of edge pixels from binary edge maps. Based on the estimated angles, line segments can be identified by grouping edge pixels sharing the same orientation. To show the efficiency of POE, we display the histograms of the estimated angles using POE in the third column of Figure 5, where the angles are estimated from the binary edge maps shown in the second column of Figure 5. It can be easily observed from the first three rows of Figure 5 that nearly all edge pixels have the same angle as the line segments in those images are parallel. Although the angles of some edge pixels differ from the most frequent one, the angle difference is pretty small (just $\frac{\pi}{16}$). Besides, it can be seen from the last row of Figure 5 that as line segments with two different angles exist, two peaks appear in the histogram of the estimated angles. Line segments detected by grouping the edge pixels having the same estimated angle up to an angle tolerance $\frac{\pi}{16}$ are shown in the last column of Figure 5. It is clearly shown that nearly all line segments in the binary edge maps are perfectly identified. In the following, we will describe the region growing algorithm, region size thresholding and subsequent line segment approximation step.

B. Region growing and region size thresholding

After estimating the orientation of each edge pixel in the edge map, we use region growing method to group edge pixels. In other words, we will group adjacent pixels whose edge orientations are the same up to a certain angle tolerance τ ($\tau = \frac{\pi}{P}$ in our experiments). The region growing step will go through all the edge pixels in the edge map. More precisely, we start from one edge pixel and use it as the seed pixel of a region, then we search in its 3×3 neighbourhood, if the angle difference between one pixel and the seed pixel is not larger than τ, we will add this pixel into the region. Each time a pixel is added into the region, the searching process will loop over the 3×3 neighbourhoods of all the pixels in the region searching for the pixels having the same angle as the seed pixel, until no more pixels can be added into the region. At the end of one searching process, we will check the size of the region (the number of pixels in the region). If the number of pixels in the region is larger than λ, it will be accepted and all the pixels will be labeled as $USED$ so that they will not be visited again. The region will be rejected otherwise and the pixels in the region are released so that they can be added to other regions. The region growing step then starts from another edge pixel until it goes through all the edge pixels in the edge map.

The natural question is how to set the region size threshold λ. We set this threshold according to the Helmholtz principle, which claims that *no meaningful structure should be observed by chance in a random image*. More precisely, according to the definition in [25], a line segment is meaningful if the expected number of its occurrences in a random image is small. The meaningfulness of a line segment is checked through the use of an $a\ contrario$ model. According to the usual $a\ contrario$ models [4], [21], [25], a line segment is considered to be ε-meaningful if its Number of False Alarms (NFA)



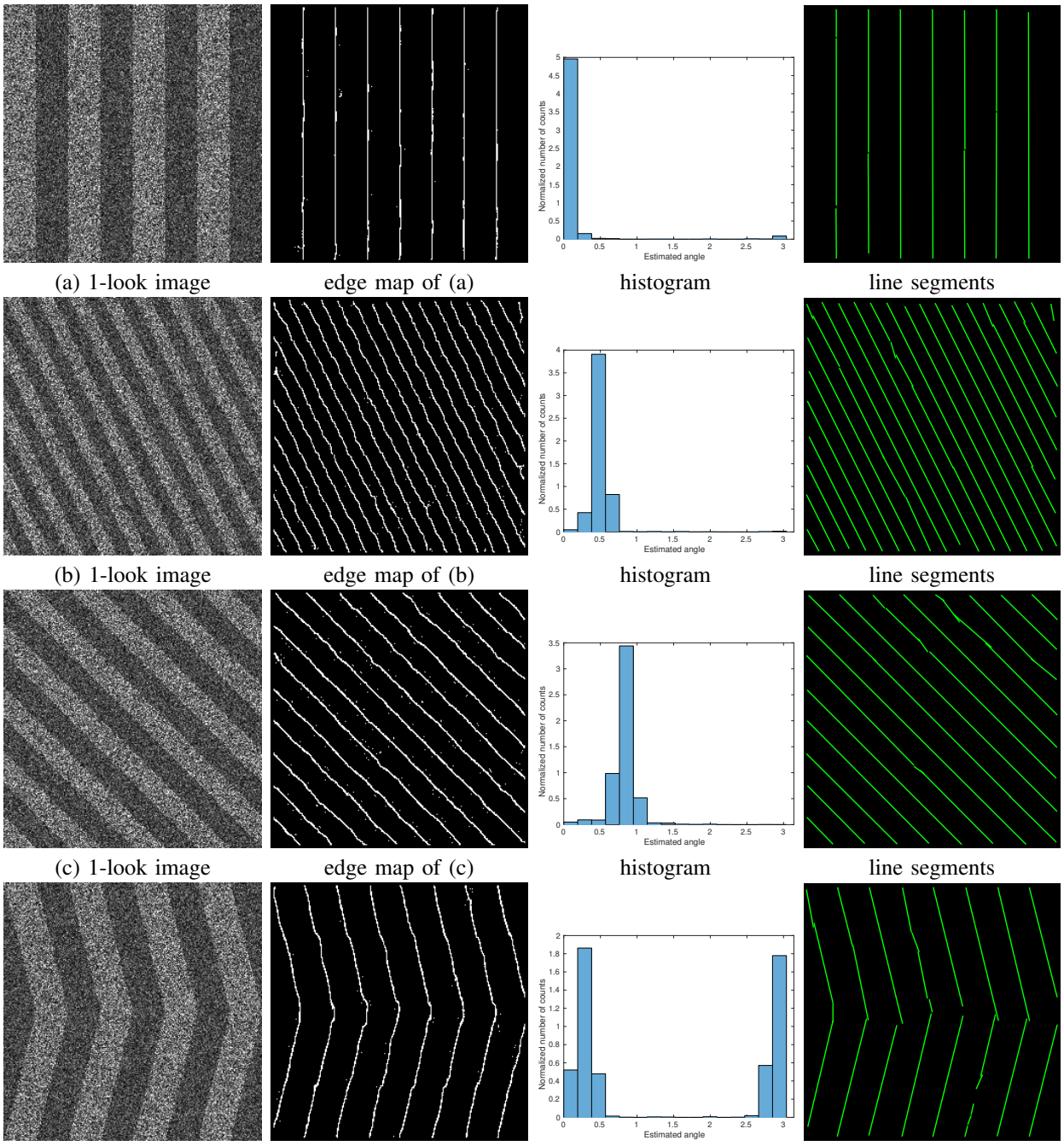
(a) angle: 0 (b) angle: $\frac{\pi}{16}$ (c) angle: $\frac{\pi}{8}$ (d) angle: $\frac{3\pi}{16}$

(e) angle: $\frac{\pi}{4}$ (f) angle: $\frac{5\pi}{16}$ (g) angle: $\frac{3\pi}{8}$ (h) angle: $\frac{7\pi}{16}$

(i) angle: $\frac{\pi}{2}$ (j) angle: $\frac{9\pi}{16}$ (k) angle: $\frac{5\pi}{8}$ (l) angle: $\frac{11\pi}{16}$

(m) angle: $\frac{3\pi}{4}$ (n) angle: $\frac{13\pi}{16}$ (o) angle: $\frac{7\pi}{8}$ (p) angle: $\frac{15\pi}{16}$

Fig. 4: The 16 window functions with their diameters orienting to a certain direction. The radius of the windows are 10 pixels. The angle of the diameter for a window is one of the following: 0, $\frac{\pi}{16}$, $\frac{\pi}{8}$, $\frac{3\pi}{16}$, $\cdots$, $\frac{15\pi}{16}$.



(a) 1-look image | edge map of (a) | histogram | line segments

(b) 1-look image | edge map of (b) | histogram | line segments

(c) 1-look image | edge map of (c) | histogram | line segments

(d) 1-look image | edge map of (d) | histogram | line segments

Fig. 5: The histograms of angles (third column) estimated by the proposed POE method from binary edge maps (second column) computed by GRHED-aug in several 1-look synthetic edge images (first column). In all the synthetic edge images, the amplitude ratio contrast of the edge is 1.6. The size of the images are 256×256 pixels. LIne segment detection results computed by the proposed POE based method are shown in the last column.

is smaller than ε. Computing the Number of False Alarms (NFA) to check the meaningfulness of a line segment implies a minimum length of line segment [25]. Here we use the hypothesis in the usual *a contrario* model [4], [25]: first, the orientation of each pixel follows a uniform distribution over $[0, \pi)$ (with minor modifications as it is $[0, 2\pi]$ in the original version): second, the orientations of pixels are independent, random variables. Then, for the minimum meaningful line segment r with size l_{min}, all pixels should have the same orientation as the line segment, in an image u of size (M, N), its Number of False Alarms should satisfy [4], [25]

$$\mathrm{NFA}(r) = N_R * p^{l_{min}} \leq \varepsilon, \tag{5}$$

where N_R represents the total number of possible line segments in a random image, and p represents the probability that a pixel has the same angle as the line segment. In our case $N_R = \sqrt{(MN)^5}$ as done in [4], [21]. As the angle tolerance τ is set to $\frac{\pi}{16}$, $p = \frac{3}{P}$ because of the uniform distribution hypothesis.

Then we have

$$l_{min} \geq \frac{\log(\varepsilon) - \log(N_R)}{\log(p)}. \tag{6}$$

Therefore, only regions with their size larger than $\frac{\log(\varepsilon)-\log(N_R)}{\log(p)}$ could be corresponding to a meaningful line segment. We set $\lambda = \frac{\log(\varepsilon)-\log(N_R)}{\log(p)}$ and keep only those regions with their size (number of pixels) larger than $\frac{\log(\varepsilon)-\log(N_R)}{\log(p)}$ (letting $\varepsilon = 1$ as done in [4], [21], [25]).

C. Line segment approximation

After the region growing and region size thresholding step, we have a set of regions which correspond to detected line segments. Here we use rectangles to approximate those regions and the rectangles are considered to be detected line segments, as done in [4], [21]. Four parameters are used to describe the rectangle: center, angle, length and width.

We first describe the way to compute the center location of the region. Given a region, let $(x(i), y(i))$ be the coordinates of the i-th pixel in the region, then the coordinates of the center pixel (c_x, c_y) can be computed as

$$c_x = \frac{\sum_{i \in region} x(i)}{N_{region}}, \tag{7}$$

$$c_y = \frac{\sum_{i \in region} y(i)}{N_{region}}, \tag{8}$$

where N_{region} is the number of pixels in the region.

Even though we have estimated the edge orientation of each pixel, and assume that their orientations are the same as the line segment, their orientations are not accurate enough. We compute the angle of the line segment as the angle between the horizontal axis and the first inertia axis of the region. The angle of the line segment can be computed as the angle of the eigenvector corresponding to the smallest eigenvalue of the following matrix:

$$\mathbb{M} = \begin{pmatrix} \mathbb{M}^{xx} & \mathbb{M}^{xy} \\ \mathbb{M}^{xy} & \mathbb{M}^{yy} \end{pmatrix}, \tag{9}$$

where

$$\mathbb{M}^{xx} = \frac{\sum_{i \in region}(x(i) - c_x)^2}{N_{region}}, \tag{10}$$

$$\mathbb{M}^{yy} = \frac{\sum_{i \in region}(y(i) - c_y)^2}{N_{region}}, \tag{11}$$

$$\mathbb{M}^{xy} = \frac{\sum_{i \in region}(x(i) - c_x) \cdot (y(i) - c_y)}{N_{region}}. \tag{12}$$

The length and width of the rectangle are then selected as the smallest values that can cover the considered region.

IV. Experiments

In this section we will demonstrate the efficiency of our proposed line segment detection framework by comparing it with the Hough transform based methods and the state-of-the-art line segment detector LSDSAR [21] in both 1-look simulated SAR dataset and several 1-look real SAR images. We focus on the 1-look situation as it is the most challenging case for line segment detection in SAR images. As we use GRHED, GRHED-aug and GRRCF-aug to compute the edge maps of images, we name the combination of these edge detectors with our Pixel Orientation Estimation (POE) based line segment detection framework as GRHED-POE, GRHED-aug-POE and GRRCF-aug-POE. We will also term them as POE based line segment detectors in the following. The efficiency of the proposed line segment detection framework will be demonstrated by these three line segment detectors.

A. Comparison in 1-look simulated SAR dataset

In order to compare different line segment detectors quantitatively in complex situations, we created a 1-look simulated SAR dataset using YorkUrban-LineSegment dataset proposed in [6], by multiplying each image in the dataset with 1-look noise. YorkUrban-LineSegment dataset is created by three annotators [6] who spent around two months labeling 102 images of size 480×640 pixels (45 indoor images, 57 outdoor images) . These images are taken from scenes of York University campus and downtown Toronto, Canada [36]. As an example to show how difficult it is to detect line segments from 1-look images, we display an image from YorkUrban-LineSegment dataset and the 1-look version of it in Figure 6. Our aim is to detect line segments in images with noise level similar to the one in Figure 6-(b).

Evaluation criterion

We compare the performances of different methods in terms of F1-scores by transforming line segments into a binary line segment map, where pixel values are one if they belong to a certain line segment, and zero otherwise. The F1-score is computed as follows:

$$F1 = \frac{2 \times precision \times recall}{precision + recall}, \tag{13}$$

where $precision$ is the fraction of correctly detected line segment pixels among all detected line segment pixels, while $recall$ is the fraction of correctly detected line segment pixels among all the line segment pixels in the ground truth.

(a) clean image (b) 1-look version

Fig. 6: An image from YorkUrban-LineSegment dataset and the 1-look version (multiply the image with 1-look noise) of it.

1) Comparison between Hough transform based methods and our POE based methods: In this part we compare our POE based methods with Hough transform based methods to demonstrate the efficiency of our new edge grouping method. Line segments will be detected by Hough transform and our POE based method from edge maps computed by GRHED-aug. The combination of GRHED-aug with Hough transform will be called as GRHED-aug-Hough. Before comparing the performances of Hough transform based methods with our POE based methods, we first evaluate the influences of parameter settings on the performances of Hough transform based methods and POE based methods.

The influences of parameter settings on the performances of Hough transform based methods

For Hough transform, four parameters have to be set by users:

- t_1, the number of peaks (or lines) to be detected;
- t_2, the smallest number of pixels in a bin, or in other words, the smallest number of pixels in a line;
- t_3, the minimum gap between two line segments along a line;
- t_4, the length of the shortest line segment that is allowed.

As no default setting is proposed for the parameters in Hough transform, we choose five sets of values as displayed in Table II to analyze the influence of parameter settings on the performances of Hough transform based method. t_1 is always set to 5000 as it corresponds to the maximum number of lines to be detected. By setting it to be 5000, we aim to detect all possible lines in the image.

TABLE II: Five sets of parameter values used for Hough transform.

Methods	t_1	t_2	t_3	t_4
GRHED-aug-Hough-1	5000	18	1	13
GRHED-aug-Hough-2	5000	18	3	13
GRHED-aug-Hough-3	5000	18	5	13
GRHED-aug-Hough-4	5000	18	5	18
GRHED-aug-Hough-5	5000	30	5	13

F1-scores computed by Hough transform for each set of parameter values in the simulated SAR dataset are displayed in Table III. From the comparison between the second column and third column of Table III we can see that the setting of ρ does not influence much on the performances of Hough transform (ρ is used to binarize the edge strength map computed by GRHED-aug). F1-scores obtained by Hough transform with $\rho = 0.2$ is generally two or three percent higher than those obtained by Hough transform with $\rho = 0.4$. Besides, it can be easily observed from the first three rows of Table III that the performances of Hough transform is very sensitive to the setting of t_3, which corresponds to the minimum gap between two line segments in the same bin. It is clearly shown that with a minor change of t_3, F1-scores obtained by Hough transform change a lot. From the last three rows of Table III we can see that the settings of t_2 and t_4 are of minor influence on the performances of Hough transform.

TABLE III: F1-scores computed by Hough transform in the simulated SAR dataset with different parameter settings.

Methods	F1-score ($\rho = 0.2$)	F1-score ($\rho = 0.4$)
GRHED-aug-Hough-1	0.2064	0.0.2031
GRHED-aug-Hough-2	0.5430	0.5283
GRHED-aug-Hough-3	0.6646	0.6445
GRHED-aug-Hough-4	0.6220	0.5922
GRHED-aug-Hough-5	0.6656	0.6442

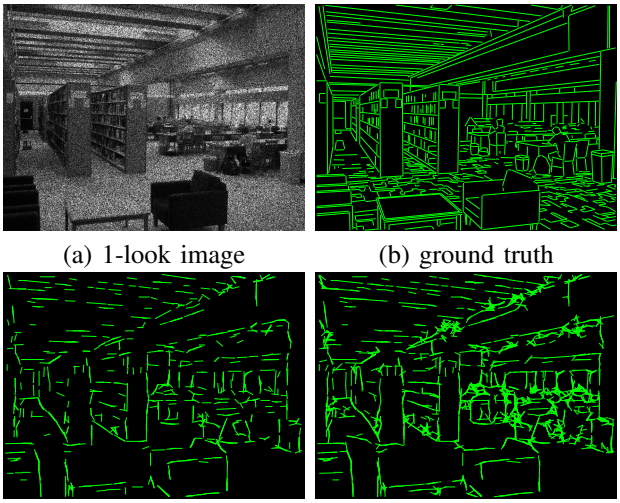
From the observation in Table III we can see that setting 3 and setting 5 seem to be good choices for Hough transform. However, when we visualize the line segment detection results shown in Figure 7, we find that many false line segments are detected by GRHED-aug-Hough-5. Besides, the phenomenon that several line segments are detected from the same location appears frequently. Similar problems have been observed for GRHED-aug-Hough-3. This is probably because the value of t_3 is set too large (with $t_3 = 5$), which means that for pixels along a line, they will be connected to form a line segment if their distance is not larger than 5 pixels, while they are assumed to belong to different line segments if their distance is larger than 5 pixels. Line segment detection results computed by GRHED-aug-Hough-2 (with $t_3 = 3$) seems to be better as much fewer false detections are produced, even though the F1-score computed by it is approximately 12 percent lower than that computed by GRHED-aug-Hough-3 and GRHED-aug-Hough-5. The problem of multiple responses is also relieved when t_3 is set to 3.

Observing the performances of Hough transform based method with different parameter settings as shown in Table III and Figure 7, we will use the settings as displayed in the second row and third row of Table II. Therefore, we will compare the performances of GRHED-aug-Hough-2 and GRHED-aug-Hough-3 with other methods.

The influences of parameter settings on the performances of our POE based method

For our Pixel Orientation Estimation (POE) based line segment detectors, three parameters have to be set:

- w: the radius of the circle-shaped windows;
- P: the number of orientations;
- τ: the angle tolerance used in the region growing step.



(a) 1-look image (b) ground truth

(c) GRHED-aug-Hough-2 (d) GRHED-aug-Hough-5

Fig. 7: Line segment detection results obtained by Hough transform based method with two sets of parameter values in a 1-look simulated SAR image. The size of the image is 480×640 pixels.

We first evaluate the influence of the radius size on the performances of our POE based method in the simulated SAR dataset. During the evaluation, the other parameters are fixed with $\rho = 0.4$, $P = 16$ and $\tau = \frac{\pi}{16}$. F1-scores computed by GRHED-aug-POE with different settings of radius size are shown in Table IV. It is clearly shown in Table IV that the influences of radius size on the performances of our POE based method are minor.

TABLE IV: F1-scores computed by GRHED-aug-POE with different radius size. ρ is set to 0.4, P is set to 16 and τ is set to $\frac{\pi}{16}$.

radius size (w)	5	8	10	12	15
F1-score	0.6068	0.6122	0.6178	0.6206	0.6216

The influences of the setting of P on the performances of the POE method are further evaluated with $\rho = 0.4$, $\tau = \frac{\pi}{16}$ and the radius size $w = 10$. F1-scores computed by GRHED-aug-POE with different settings of P is displayed in Table V. From the F1-scores displayed in Table V we can see that with different settings of P, F1-scores obtained by GRHED-aug-POE do not change a lot. From the comparison between the third column and fourth column of Table V we can see that the performances of GRHED-aug-POE are not improved by considering a larger number of orientations. From the comparison between the second column and third column of Table V we can see that the performances of GRHED-aug-POE decrease with smaller number of orientations considered. Two line segments with their angle close to each other may not be distinguished by considering a smaller number of orientations. In addition, when P is set to 8 and τ is set to $\frac{\pi}{16}$, the POE method groups only pixels with exactly the same angle, while for $P = 16$ and $\tau = \frac{\pi}{16}$, pixels with their angle differences no larger than $\frac{\pi}{16}$ will be gathered together. This could be another reason for the decrease of performances of GRHED-aug-POE with $P = 8$.

TABLE V: F1-scores computed by GRHED-aug-POE on the simulated SAR dataset with different settings of P. ρ is set to 0.4, τ is set to $\frac{\pi}{16}$ and the radius size w is set to 10 pixels.

Number of Orientations	8	16	32
F1-score	0.5717	0.6178	0.6149

The influences of the setting of τ on the performances of GRHED-aug-POE are evaluated in the simulated SAR dataset as shown in Table VI. It is clearly shown in Table VI that F1-scores obtained by GRHED-aug-POE become higher when a larger value of τ is used. The F1-score obtained by GRHED-aug-POE with $\tau = \frac{\pi}{32}$ is more than 10 percent lower than GRHED-aug-POE with other settings of τ. This could be because with this setting of τ, only pixels with exactly the same angle will be grouped. Even though GRHED-aug-POE with $\tau = \frac{\pi}{8}$ obtains the highest F1-score, the setting of $\tau = \frac{\pi}{16}$ is prefered as when τ is set to $\frac{\pi}{8}$, GRHED-aug-POE may not be able to distinguish two line segments with their angle difference smaller than $\frac{\pi}{8}$.

TABLE VI: The influences of the setting of τ on the performances of GRHED-aug-POE. The radius of the circle-shaped windows is set to 10 pixels, ρ is set to 0.4 and the number of orientations P is set to 16.

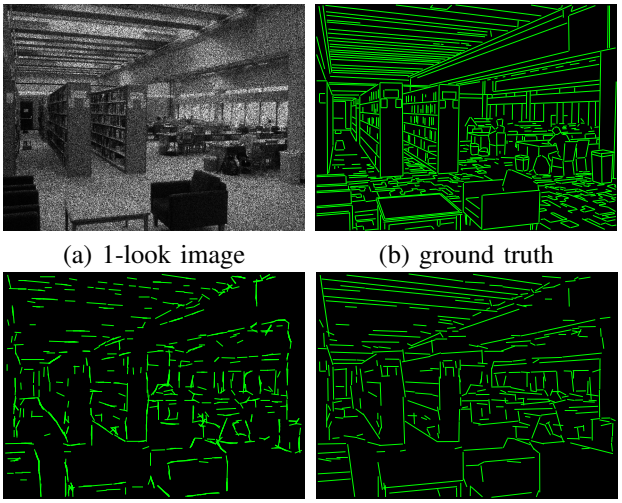
τ	$\frac{\pi}{8}$	$\frac{\pi}{16}$	$\frac{\pi}{32}$
F1-score	0.6406	0.6178	0.4957

From the observations in Table IV, Table V and Table VI, we propose to use the following setting as the default setting of our POE based method: the radius size w of the circle-shaped windows is set to 10 pixels, the number of orientations P is set to 16, and the angle tolerance τ is set to $\frac{\pi}{16}$.

For a visual inspection, we compare the performances of GRHED-aug-Hough-2 and GRHED-aug-POE with the prosed setting in a 1-look simulated SAR image as shown in Figure 8. It is clearly shown in Figure 8 that GRHED-aug-POE detects much more true line segments than GRHED-aug-Hough-2, while fewer false detections are produced by our POE based method. Besides, line segments detected by GRHED-aug-Hough-2 tend to be fragmented. The completeness of line segments detected by Hough transform based methods can be improved with a larger value of t_3, as displayed in Figure 7-(d), but more false detections are produced. Furthermore, it can be seen from both Figure 7 and Figure 8 that the problem of multiple responses exists for Hough transform based methods.

From all the experiments above, the advantages of our POE based methods over Hough transform based methods are clear:

- Parameter values for our POE based methods are much easier to choose, and the performances of our POE based methods are much less sensitive to the choice of parameter values;
- the problem of multiple responses exist for Hough transform based methods, but it is not the case for our POE based methods;



(a) 1-look image (b) ground truth

(c) GRHED-aug-Hough-2 (d) GRHED-aug-POE

Fig. 8: Line segment detection results obtained by GRHED-aug-Hough-2 and GRHED-aug-POE in a 1-look simulated SAR image. The size of the image is 480×640 pixels.

- the proposed POE based methods detect more true line segments while producing fewer false detections, the line segments detected by our POE based method are more complete.

More comparisons will be done between Hough transform based methods and our POE based methods in the following to show the efficiency of our new edge grouping method based on POE.

2) Comparison between LSDSAR and our POE based methods: In this part we compare the performances of our POE based methods with the state-of-the-art line segment detector for SAR images LSDSAR.

Parameter settings

For our Pixel Orientation Estimation (POE) based line segment detectors, the radius of the circle-shaped windows is set to 10 pixels, the number of orientations is set to 16, the angle tolerance τ is set to $\frac{\pi}{16}$, the region size threshold λ is automatically computed according to the Helmholtz principle with $\varepsilon = 1$.

For deep learning based edge detectors that we will use to compute the binary edge maps, namely GRHED-POE, GRHED-aug-POE and GRRCF-aug-POE, one important factor that may influence the performances of the resulting detectors is the threshold ρ used to binarize the edge strength map during plain thresholding. We vary this threshold ρ from 0.1 to 0.5 with step 0.1 for all the deep learning based edge detectors and test its influences on the F1-scores for each method (the range of pixel values in the edge probability maps computed by deep learning based methods is $[0, 1]$).

For LSDSAR, we vary the exponential weight parameter α used in the gradient computation step (GR) from 1 to 5 with step 1 and keep the other parameter settings unchanged [21]. We compute F1-scores for LSDSAR with each α value.

Evaluation results

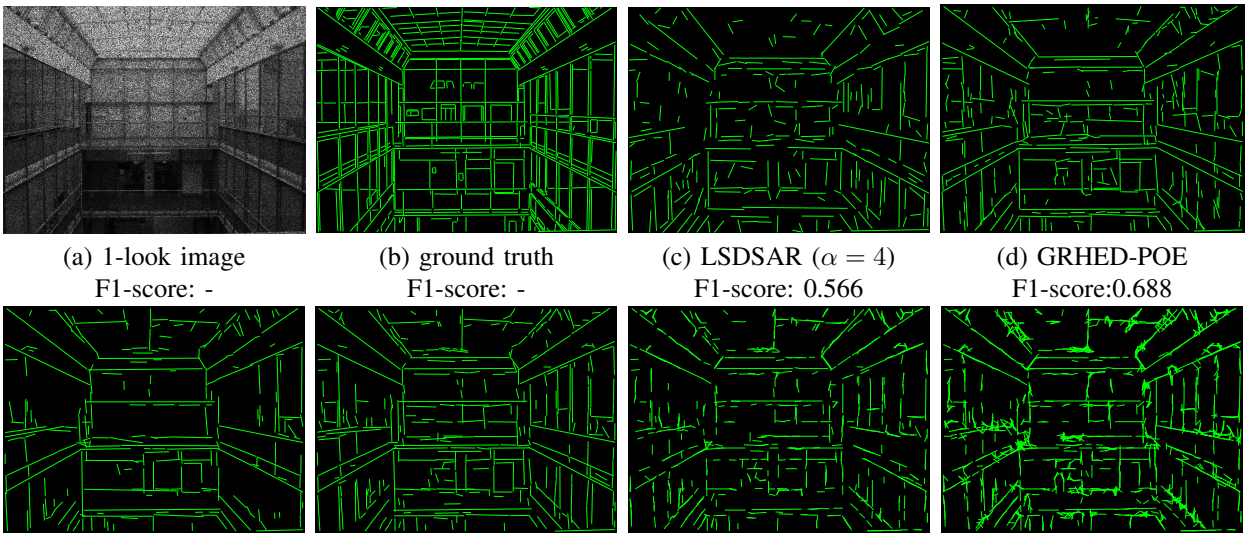
The F1-scores computed by LSDSAR, GRHED-POE, GRHED-aug-POE and GRRCF-aug-POE in the 1-look simulated SAR dataset are shown in Table VII. The best F1-score obtained by each method is highlighted in larger bold font.

From Table VII we can see that the highest F1-score obtained by LSDSAR is 0.4810 with $\alpha = 4$. It is much lower (more than 10 percent) than the other methods, which shows the efficiency of the proposed line segment detection framework. In addition, GRHED-aug-POE with $\rho = 0.1$ advances the performances of the state-of-the-art LSDSAR with $\alpha = 4$ by approximately 18 percent (0.6677 compared to 0.4810). An interesting phenomenon is that with the value of ρ increasing, it seems that the detection capabilities of GRHED-POE, GRHED-aug-POE and GRRCF-aug-POE become weaker. The main reason is that we may discard true edge pixels when we use a larger value of ρ, even though we detect fewer noise pixels with it. It can also be seen from Table VII that with all settings of ρ, the F1-scores obtained by POE based line segment detectors are almost always much higher than the best F1-score obtained by LSDSAR.

Although the POE based line segment detectors with $\rho = 0.1$ or 0.2 seem to have better detection capability, more false line segments could be detected as more false edge pixels will be produced by deep learning based edge detectors. It has been shown in [26] that the performances of deep learning based edge detectors are not sensitive to the choice of ρ. In general applications, we propose to set ρ to 0.4 so that the edge maps computed by deep learning based edge detectors are cleaner.

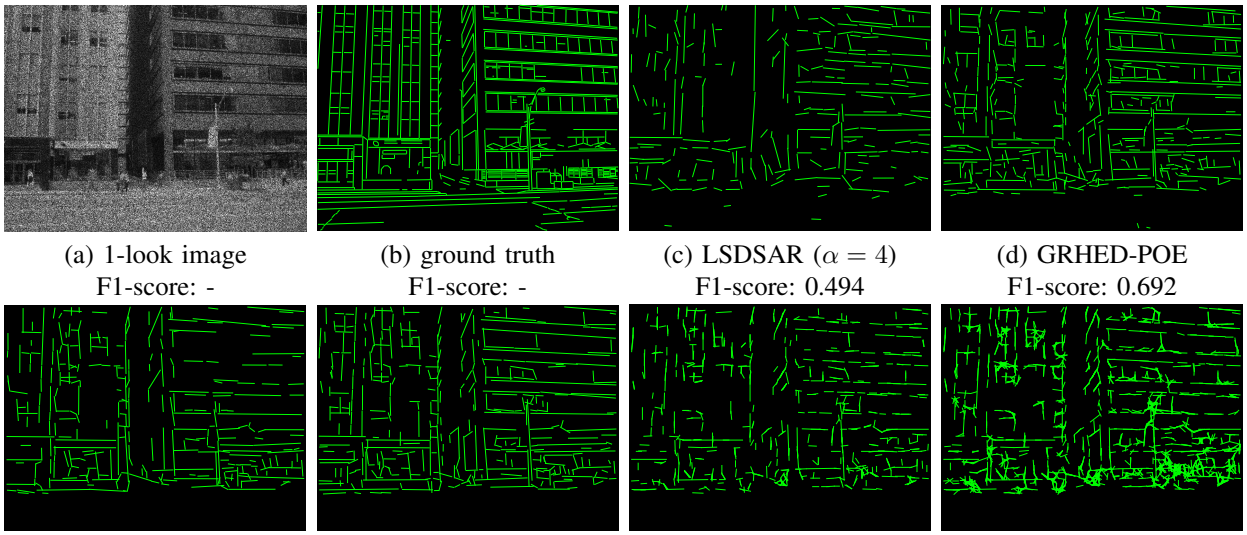
3) Line segment detection results computed by Hough transform based methods, LSDSAR and the POE based methods in two 1-look simulated SAR images: For a visual comparison, we show the line segment detection results obtained by GRHED-aug-Hough-2 ($\rho = 0.4$), GRHED-aug-Hough-3 ($\rho = 0.4$), LSDSAR ($\alpha = 4$), GRHED-POE ($\rho = 0.4$), GRRCF-aug-POE ($\rho = 0.4$) and GRHED-aug-POE ($\rho = 0.4$) in two 1-look simulated SAR images (one indoor image and one outdoor image) in Figure 9 and Figure 10. It is clearly shown from the second row and third row of both figures that the proposed POE based line segment detectors detect much more true line segments than LSDSAR while they produce only a few false detections. In both figures, the F1-scores computed by GRHED-POE and GRHED-aug-POE are at least 12 percent higher than that of LSDSAR. GRRCF–aug-POE achieves lower F1-scores than GRHED-POE and GRHED-aug-POE, but its F1-scores are still much higher than those of LSDSAR. Compared to the line segments detected by the POE based methods, the line segments detected by LSDSAR tend to be fragmented. For the comparison among those POE based methods, the performances of GRHED-POE and GRHED-aug-POE are similar, while both of them detect more true line segments than GRRCF-aug-POE. This conforms with the observation in Table VII that GRRCF-aug-POE obtains the lowest F1-score with $\rho = 0.4$.

From the comparison among Figure 9-(f), Figure 9-(g) and Figure 9-(h) we can see that with the edge maps computed by the same edge detector, the proposed POE based method obtains much better line segment detection results as we detect more true line segments and less false detections. Although GRHED-aug-Hough-3 seems to obtain the highest F1-score, line segments detected by GRHED-aug-POE are



(a) 1-look image F1-score: -
(b) ground truth F1-score: -
(c) LSDSAR ($\alpha = 4$) F1-score: 0.566
(d) GRHED-POE F1-score:0.688
(e) GRRCF-aug-POE F1-score: 0.603
(f) GRHED-aug-POE F1-score: 0.7
(g) GRHED-aug-Hough-2 F1-score: 0.614
(h) GRHED-aug-Hough-3 F1-score: 0.734

Fig. 9: Line segment detection results obtained by Hough transform based method, LSDSAR and POE based line segment detectors in a 1-look simulated SAR image (indoor). The size of the image is 480×640 pixels.



(a) 1-look image F1-score: -
(b) ground truth F1-score: -
(c) LSDSAR ($\alpha = 4$) F1-score: 0.494
(d) GRHED-POE F1-score: 0.692
(e) GRRCF-aug-POE F1-score: 0.627
(f) GRHED-aug-POE F1-score: 0.684
(g) GRHED-aug-Hough-2 F1-score: 0.561
(h) GRHED-aug-Hough-3 F1-score: 0.688

Fig. 10: Line segment detection results obtained by Hough transform based method, LSDSAR and POE based line segment detectors in a 1-look simulated SAR image (outdoor). The size of the image is 480×640 pixels.

more complete. For Hough transform based methods, multiple responses exist for the same line segment and detected line segments tend to be fragmented. Besides, many true line segments are missing for GRHED-aug-Hough-2 while too many false line segments are detected by GRHED-aug-Hough-3. It is also shown from Figure 9-(g) and Figure 9-(h) that the performances of Hough transform based methods are sensitive to the choice of thresholds. Similar conclusions can be obtained from Figure 10-(f), Figure 10-(g) and Figure 10-(h)).

4) The influence of spatial correlations on the performances of LSDSAR and GRHED-aug-POE: Indeed, spatial correlations exist among neighbouring pixels of real SAR images. For 1-look complex data, these correlations are mainly introduced by two factors: over-sampling and weighting of the azimuth and range spectrum [37], [38]. However, these factors can change depending on the data providers, even for images with similar resolutions. For instance, the weighting functions of TerraSAR-X and CSK images are different [39].

In order to show the influence of spatial correlations on the performances of LSDSAR and GRHED-aug-POE, we simulate three 1-look synthetic images with different amplitude ratio contrast of the edge and a 1-look SAR dataset using YorkUrban-LineSegment dataset. The spatially correlated

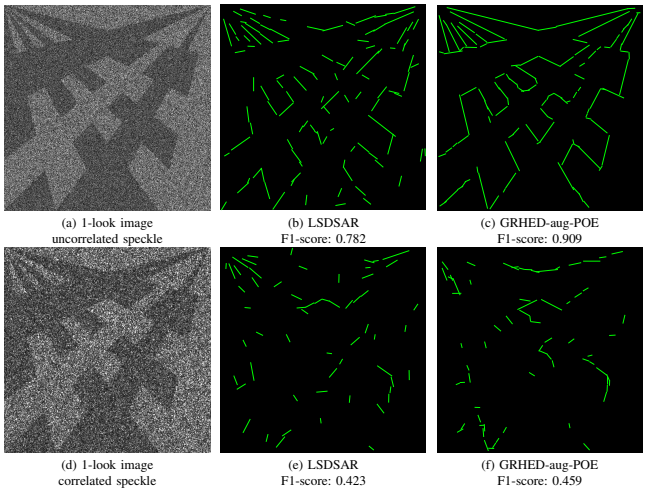
TABLE VII: F1-scores computed for LSDSAR with different α values and the POE ;based line segment detectors with different settings of ρ. ρ is used to binarize the Edge Strength Map (ESM) computed by deep learning based edge detectors. The range of pixel values in the ESM is $[0.0, 1.0]$).

Methods	F1-score	Methods	F1-score
LSDSAR ($\alpha = 1$)	0.2160	GRRCF-aug-POE ($\rho = 0.1$)	**0.6556**
LSDSAR ($\alpha = 2$)	0.4001	GRRCF-aug-POE ($\rho = 0.2$)	0.6429
LSDSAR ($\alpha = 3$)	0.4616	GRRCF-aug-POE ($\rho = 0.3$)	0.6161
LSDSAR ($\alpha = 4$)	**0.4810**	GRRCF-aug-POE ($\rho = 0.4$)	0.5695
LSDSAR ($\alpha = 5$)	0.4742	GRRCF-aug-POE ($\rho = 0.5$)	0.4962
GRHED-POE ($\rho = 0.1$)	0.6497	GRHED-aug-POE ($\rho = 0.1$)	**0.6677**
GRHED-POE ($\rho = 0.2$)	**0.6546**	GRHED-aug-POE ($\rho = 0.2$)	0.6623
GRHED-POE ($\rho = 0.3$)	0.6451	GRHED-aug-POE ($\rho = 0.3$)	0.6459
GRHED-POE ($\rho = 0.4$)	0.6216	GRHED-aug-POE ($\rho = 0.4$)	0.6178
GRHED-POE ($\rho = 0.5$)	0.5870	GRHED-aug-POE ($\rho = 0.5$)	0.5690

speckle noise is simulated using the parameters of TerraSAR-X images. In StripMap mode, the spatial correlations of the speckle noise mainly come from two points: apodization applied after synthesis and image over-sampling by zero-padding interpolation. Please refer to [39]–[41] for more details about these steps.

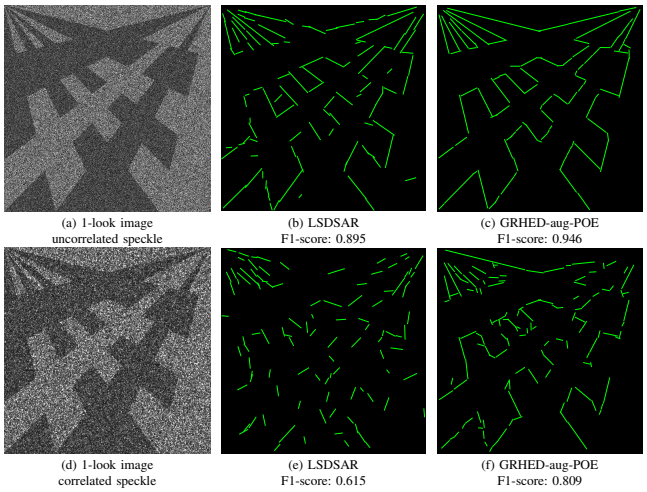
We simulate three synthetic images of size 512×512 pixels with the amplitude ratio contrast of the edge 1.4, 1.6 and 1.8. These images are corrupted by both uncorrelated speckle and correlated speckle noise to produce the simulated SAR images. We compare the performances of LSDSAR with $\alpha = 4$ and GRHED-aug-POE with $\rho = 0.4$ in these simulated SAR images. Line segment detection results produced by LSDSAR and GRHED-aug-POE are displayed in Figure 11, Figure 12, and Figure 13. It is clearly shown from these figures that GRHED-aug-POE obtains consistently better line segment detection results than LSDSAR. Our POE based method detects more true line segments with very few false detections. The F1-scores obtained by GRHED-aug-POE are usually much higher than those obtained by LSDSAR.

It can also be seen from Figure 11, Figure 12 and Figure 13 that the spatial correlations of speckle noise decrease the performances of both LSDSAR and GRHED-aug-POE. Both methods are much more efficient in images with uncorrelated speckle noise. This is reasonable as both methods assume spatial independence of the speckle noise.



(a) 1-look image uncorrelated speckle
(b) LSDSAR F1-score: 0.782
(c) GRHED-aug-POE F1-score: 0.909
(d) 1-look image correlated speckle
(e) LSDSAR F1-score: 0.423
(f) GRHED-aug-POE F1-score: 0.459

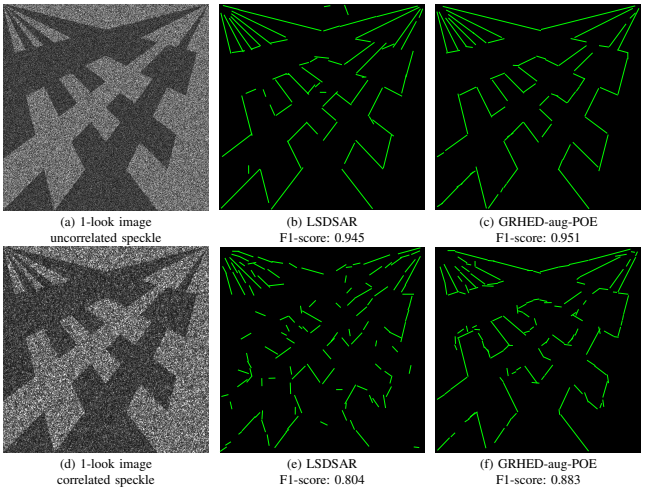
Fig. 11: Line segment detection results obtained by LSDSAR and GRHED-aug-POE in a 1-look synthetic SAR image with the amplitude ratio contrast of the edge 1.4. The size of the image is 512×512 pixels.



(a) 1-look image uncorrelated speckle
(b) LSDSAR F1-score: 0.895
(c) GRHED-aug-POE F1-score: 0.946
(d) 1-look image correlated speckle
(e) LSDSAR F1-score: 0.615
(f) GRHED-aug-POE F1-score: 0.809

Fig. 12: Line segment detection results obtained by LSDSAR and GRHED-aug-POE in a 1-look synthetic SAR image with the amplitude ratio contrast of the edge 1.6. The size of the image is 512×512 pixels.

We also simulate a SAR dataset using YorkUrban-LineSegment dataset by corrupting the images with spatially correlated speckle noise, and compare the performances of LSDSAR (with $\alpha = 4$) and GRHED-aug-POE (with $\rho = 0.4$) on it. The F1-scores obtained by LSDSAR and GRHED-aug-POE in the simulated dataset are displayed in Table VIII. It can be seen from Table VIII that the F1-score obtained by GRHED-aug-POE is 12 percent higher than that of LSDSAR. It is also shown in Table VIII that the F1-scores obtained by both methods in the images corrupted by spatially correlated speckle noise are around 10 percent lower than those obtained by them in the dataset simulated using uncorrelated speckle noise. We display the line segment detection results computed by LSDSAR and GRHED-aug-POE in a 1-look simulated



(a) 1-look image uncorrelated speckle
(b) LSDSAR F1-score: 0.945
(c) GRHED-aug-POE F1-score: 0.951
(d) 1-look image correlated speckle
(e) LSDSAR F1-score: 0.804
(f) GRHED-aug-POE F1-score: 0.883

Fig. 13: Line segment detection results obtained by LSDSAR and GRHED-aug-POE in a 1-look synthetic SAR image with the amplitude ratio contrast of the edge 1.8. The size of the image is 512×512 pixels.

SAR image with spatially correlated noise in Figure 14. It is clearly shown in Figure 14 that our proposed method is much more efficient than LSDSAR. More true line segments are detected with only a few false detections. The F1-score obtained by GRHED-aug-POE is around 22 percent higher than that of LSDSAR, which demonstrates the efficiency of the proposed method.

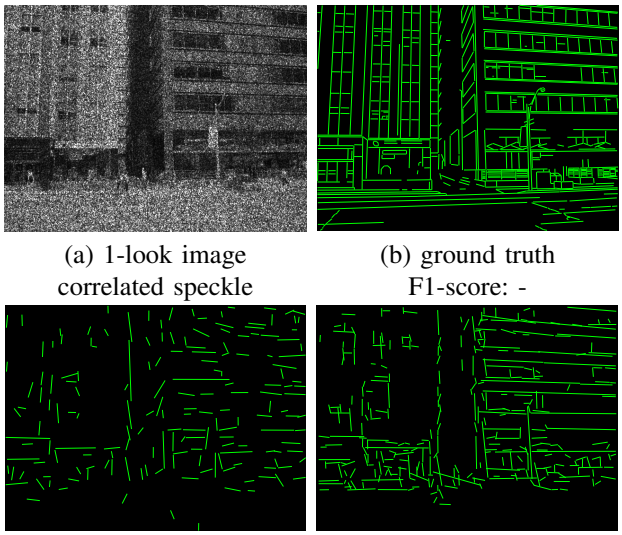
TABLE VIII: F1-scores obtained by LSDSAR with $\alpha = 4$ and GRHED-aug-POE with $\rho = 0.4$ in the 1-look YorkUrban-LineSegment datasets which are corrupted by correlated and uncorrelated speckle noise.

Methods	LSDSAR	GRHED-aug-POE
F1-score (correlated)	0.3809	0.5028
F1-score (uncorrelated)	0.4810	0.6178

LSDSAR uses a ratio operator to compute the local orientations of pixels, while the justification of the ratio operators for SAR images relies on the assumption of uncorrelated speckle noise [17]. GRHED-aug-POE also contains a ratio operator, and the edge detector GRHED is trained using spatially uncorrelated speckle noise. Therefore, it is reasonable to see the decrease of performances when they are tested in images contaminated by spatially correlated speckle noise. As almost all line segment detectors and edge detectors for SAR images depend on the ratio operation, all of them assume spatial independence of the speckle noise. Considering the decrease of performances of LSDSAR and GRHED-aug-POE, we will work on how to tackle the spatial correlations of real SAR images in the future (some researches dedicated to speckle decorrelation of real SAR images can be found in [40]–[44]).

B. Testing in three 1-look real SAR images

The efficiency of the proposed line segment detection framework has been demonstrated in simulated SAR images,



(a) 1-look image correlated speckle
(b) ground truth F1-score: -
(c) LSDSAR F1-score: 0.347
(d) GRHED-aug-POE F1-score: 0.569

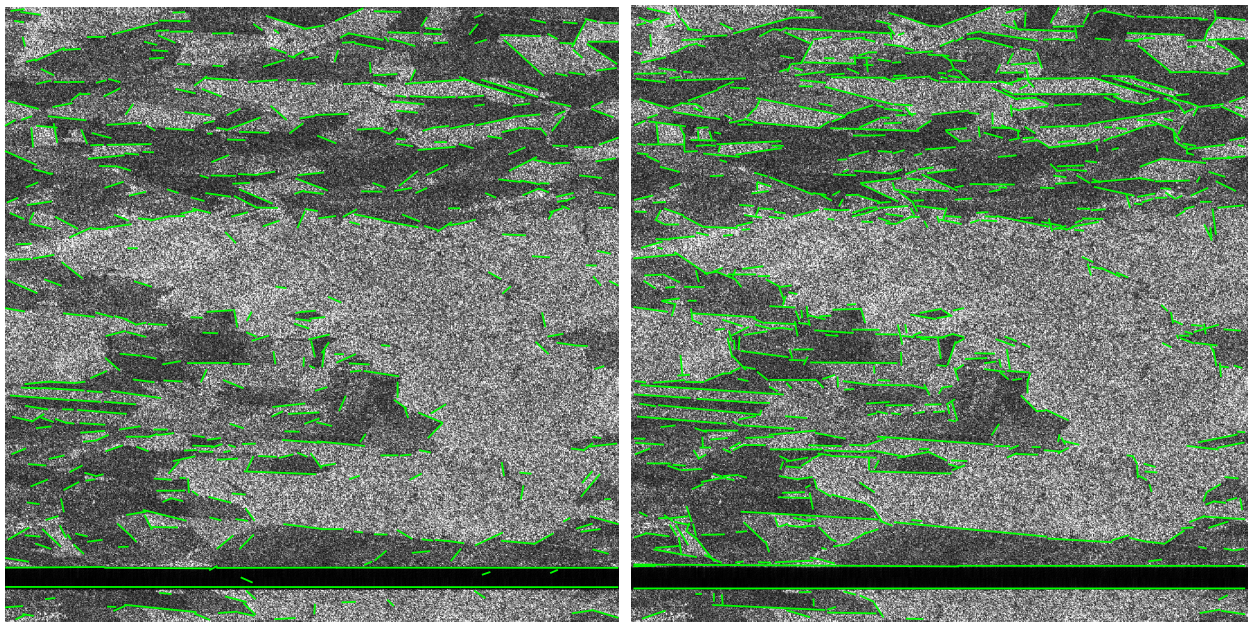
Fig. 14: Line segment detection results obtained by LSDSAR and GRHED-aug-POE in a 1-look simulated SAR image with spatially correlated speckle noise (outdoor). The size of the image is 480×640 pixels.

but it is still necessary to check its efficiency in real SAR images. We compare the performances of Hough transform based method, LSDSAR and our POE based method in three Sentinel-1 (ESA) Single Look Complex images of different scenes. The size of the image is 1024×1024 pixels. The resolution ($rg \times az$) of the images are from 2.7×22 meters to 3.5×22 meters. Pixel spacing ($rg \times az$) are 2.3×14.1 meters. More details about these images can be found in Table IX.

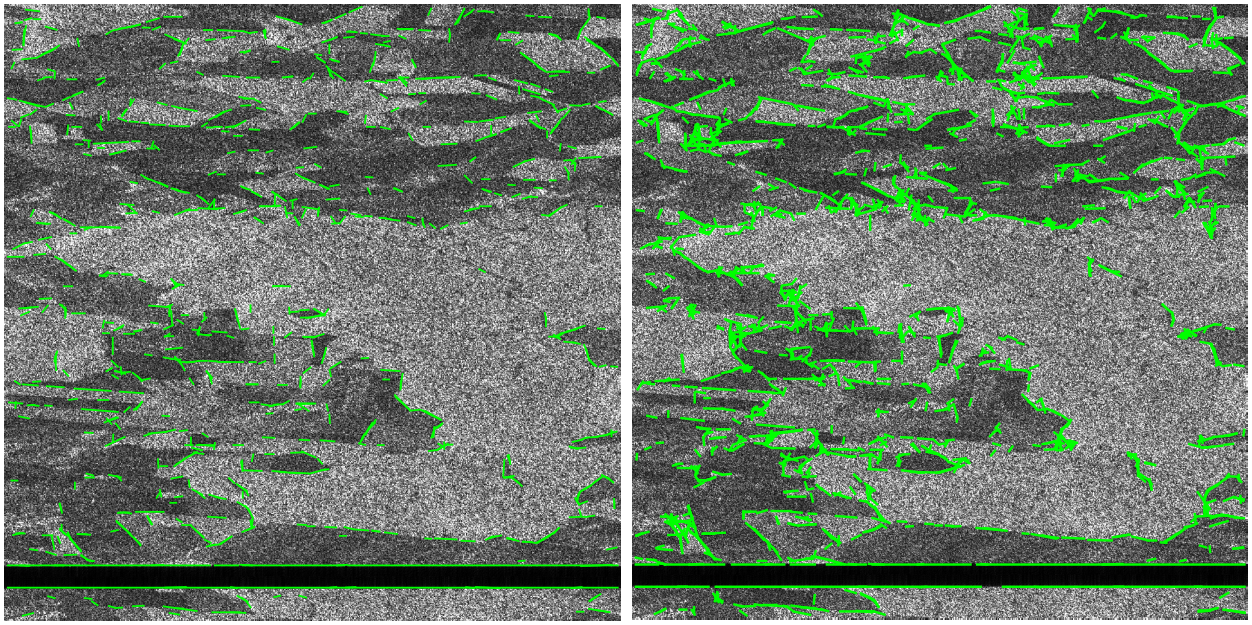
TABLE IX: Three Sentinel-1 Single Look Complex images that are used to compare the performances of different methods.

Area	Number of looks	mode	date
Paris, France	1	IW mode	May 12, 2021
Toronto, Canada	1	IW mode	June 27, 2021
Lelystad, Netherlands	1	IW mode	June 29, 2021

Edge maps computed by GRHED-aug with $\rho = 0.4$ are used as the input of Hough transform and our POE based method. We use the default parameter settings in LSDSAR with $\alpha = 4$. Two sets of parameter values are used for Hough transform and we use the proposed setting for our POE based method. Line segment detection results computed by different methods in the three 1-look Sentinel-1 images are shown in Figure 15, Figure 16 and Figure 17. It is clearly shown from all these figures that the proposed POE based line segment detector detects much more true line segments than the other methods, while detecting only a few false line segments. In addition, line segments detected by GRHED-aug-POE are more complete compared to those detected by LSDSAR and GRHED-aug-Hough-2. The proposed POE based method fully exploits the efficiency of the deep learning based edge detector



(a) LSDSAR (b) GRHED-aug-POE



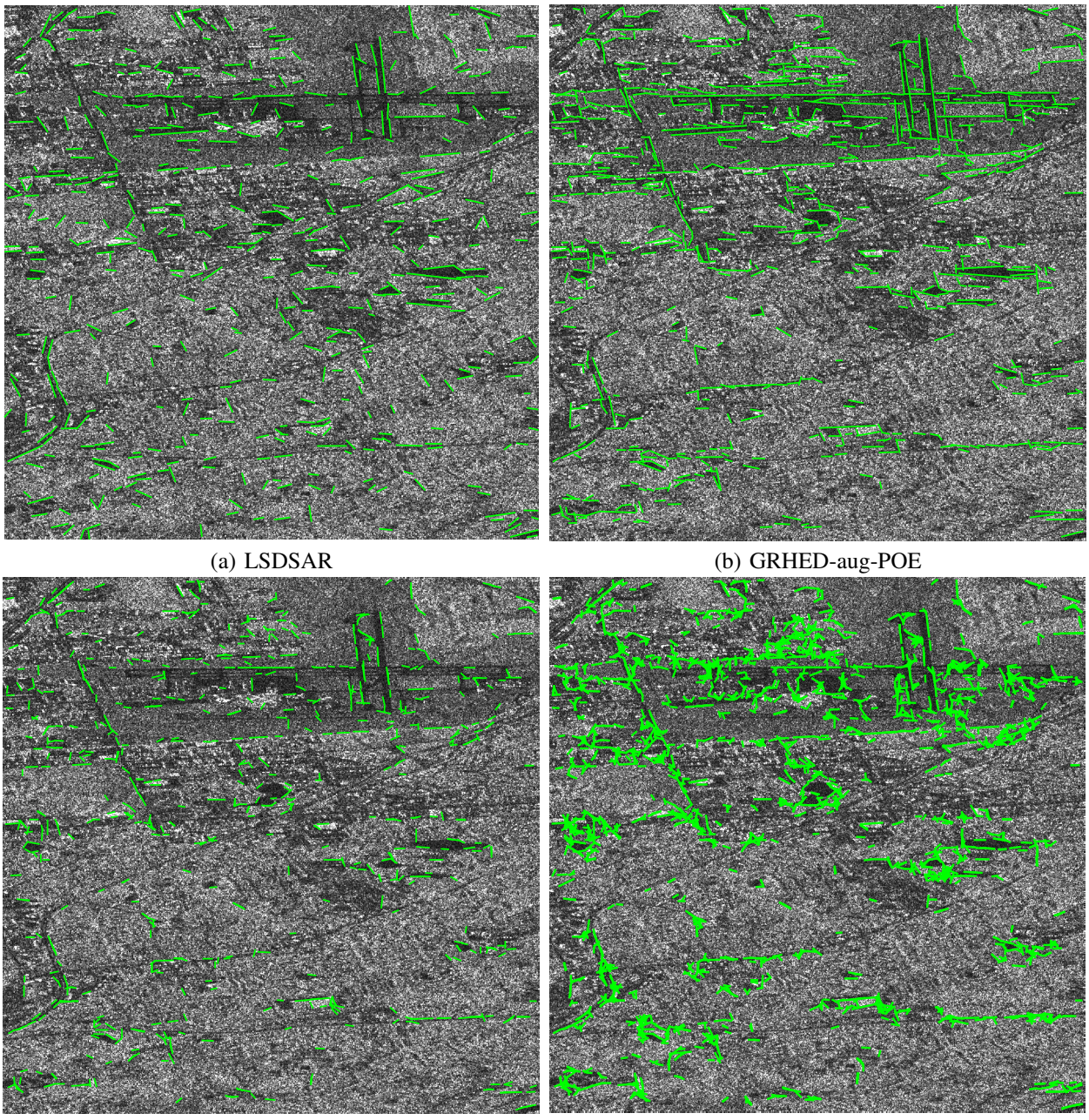
(c) GRHED-aug-Hough-2 (d) GRHED-aug-Hough-3

Fig. 15: Line segment detection results (shown in green lines) computed by Hough transform based method, LSDSAR and the POE based method in a 1-look Sentinel-1 image (Paris, France) of size 1024×1024 pixels.

(GRHED-aug) to emphasize true edge pixels and suppress false edge pixels. Due to the inefficiency of Hough transform to group edge pixels, it detects either a small number of fragmented line segments or too many false line segments, as can be seen from the second row of all these figures. In the meantime, the new edge grouping strategy based on our POE method shows its efficiency. GRHED-aug-POE fully exploits the detection capability of the deep learning based edge detector and outperforms the existing methods a lot.

V. Conclusions

In this work, we propose a novel Pixel Orientation Estimation (POE) method, which is able to estimate the orientation of pixels in binary edge maps. On the basis of POE, we propose a new line segment detection framework that can be combined with any edge detectors. Compared to Hough transform, which is the most commonly used method to detect line segments from binary edge maps, the benefits of the proposed method are clear: first, few parameter tunings are required in POE based methods; second, the end points of line segments will be naturally identified upon the generation of candidate line segments, while in Hough transform, a threshold has to be set



(a) LSDSAR (b) GRHED-aug-POE

(c) GRHED-aug-Hough-2 (d) GRHED-aug-Hough-3

Fig. 16: Line segment detection results (shown in green lines) computed by Hough transform based method, LSDSAR and the POE based method in a 1-look Sentinel-1 image (Toronto, Canada) of size 1024×1024 pixels.

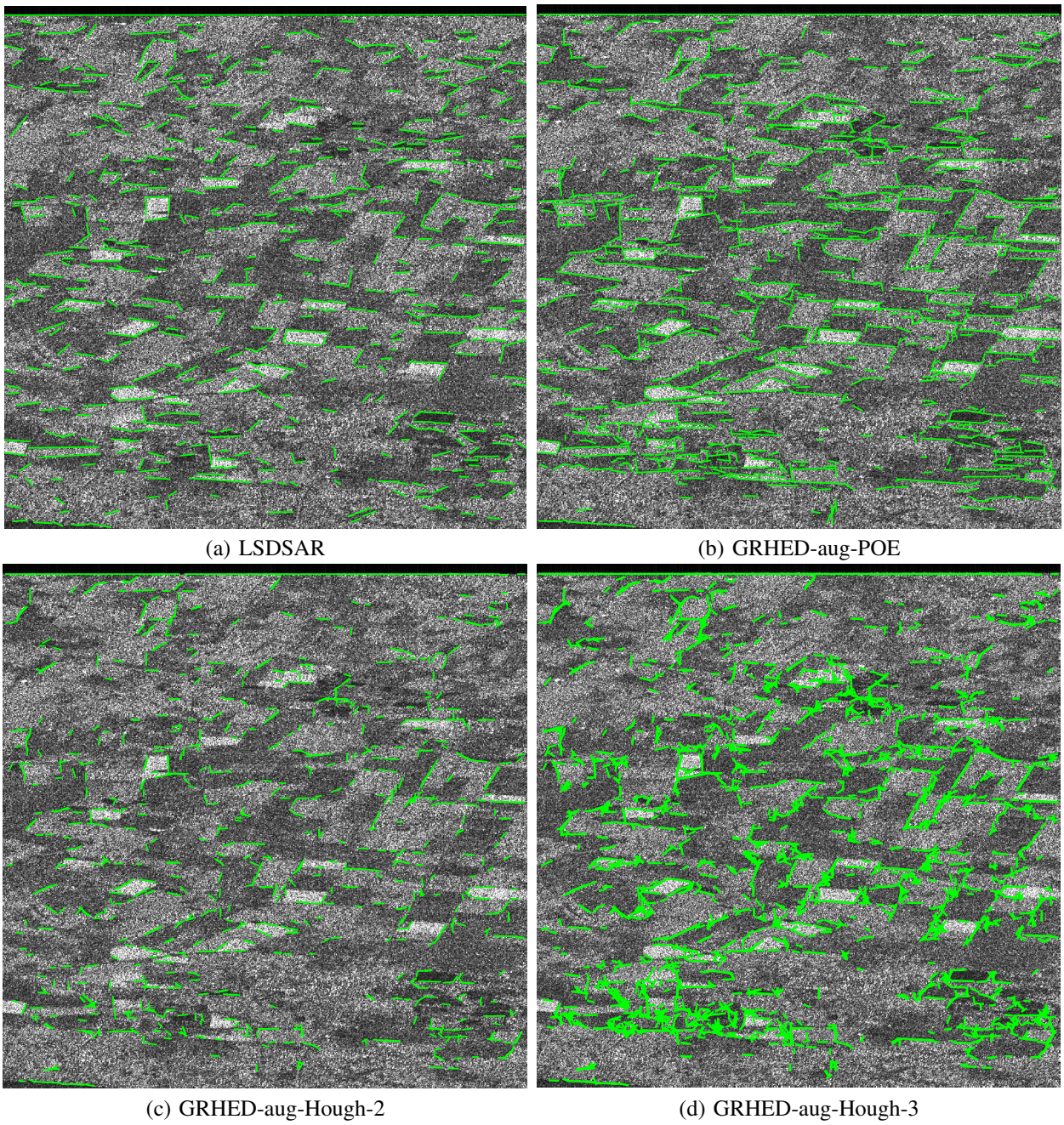
and it is difficult to choose one that works well globally; third, as line segments are described with rectangles, we do not face the problem of multiple responses faced by Hough transform. In addition, the proposed line segment detection framework enables us to leverage the success of deep learning models for edge detection in SAR images. The efficiency of the proposed line segment detection framework is demonstrated in both simulated SAR and real SAR images. However, it should be noted that many false line segments could be detected if lots of false edge pixels are detected during edge detection, which means that the performances of the proposed line segment detection framework relies a lot on the performances of edge detectors. Thanks to the success of deep learning models for edge detection in SAR images, we advance the performances of the state-of-the-art method by a large margin in both simulated SAR and real SAR images.

SOURCE CODE

The source code of this work will be released upon the acceptance of the paper.

FUNDINGS

This research is funded by Chang'an University (Xi'an, China) through the National Key Research and



(a) LSDSAR (b) GRHED-aug-POE

(c) GRHED-aug-Hough-2 (d) GRHED-aug-Hough-3

Fig. 17: Line segment detection results (shown in green lines) computed by Hough transform based method, LSDSAR and the POE based method in a 1-look Sentinel-1 image (Lelystad, Netherlands) of size 1024×1024 pixels.

Development Program of China (2020YFC1512001), the National Natural Science Foundation of China (41974006), the Shenzhen Scientific Research and Development Funding Program (KQJSCX20180328093453763 and JCYJ20180305125101282), the Research Program from the Department of Education of Guangdong Province (grant No. 2018KTSCX196), and the National Natural Science Foundation of China (NSFC) (Grant No.61771014).

REFERENCES

[1] R. O. Duda and P. E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, pp. 11–15, 1972.

[2] J. B. Burns, A. R. Hanson, and E. M. Riseman, "Extracting straight lines," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, pp. 425–455, 1986.

[3] J. Skingley and A. Rye, "The hough transform applied to sar images for thin line detection," *Pattern Recognition Letters*, vol. 6, pp. 61–67, 1987.

[4] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: A fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 722–732, 2010.

[5] E. J. Almazan, R. Tal, Y. Qian, and J. H. Elder, "Mcmlsd: A dynamic programming approach to line segment detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2031–2039.

[6] N.-G. Cho, A. Yuille, and S.-W. Lee, "A novel linelet-based representation for line segment detection," *IEEE Transactions on Pattern Analysis*

and Machine Intelligence, vol. 40, pp. 1195–1208, 2018.

[7] K. Huang, Y. Wang, Z. Zhou, T. Ding, S. Gao, and Y. Ma, "Learning to parse wireframes in images of man-made environments," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018, pp. 626–635.

[8] N. Xue, S. Bai, F. Wang, G. Xia, T. Wu, and L. Zhang, "Learning attraction field representation for robust line segment detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1595–1603.

[9] Z. Zhang, Z. Li, N. Bi, J. Zheng, J. Wang, K. Huang, W. Luo, Y. Xu, and S. Gao, "PPGNet: Learning Point-Pair Graph for Line Segment Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7105–7114.

[10] Y. Zhou, H. Qi, and Y. Ma, "End-to-End Wireframe Parsing," in *IEEE Conference on Computer Vision (ICCV) 2019*, 2019, pp. 962–971.

[11] N. Xue, T. Wu, S. Bai, F.-D. Wang, G.-S. Xia, L. Zhang, and P. H. Torr, "Holistically-Attracted Wireframe Parsing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] J. Illingworth and J. Kittler, "The adaptive hough transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, pp. 690–698, 1987.

[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] F. Xu and Y. Jin, "Automatic reconstruction of building objects from multiaspect meter-resolution sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, pp. 2336–2353, 2007.

[15] C. Palmann, S. Mavromatis, and J. Sequeira, "Sar image registration using a new approach based on the generalized hough transform," in *ISPRS 2008*, vol. XXXVII, 2008.

[16] H. Sui, C. Xu, J. Liu, and 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*, vol. 53, pp. 6058–6072, 2015.

[17] R. Touzi, A. Lopes, and P. Bousquet, "A statistical and geometrical edge detection for sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 26, pp. 764–773, 1988.

[18] R. Fjørtoft, A. Lopes, P. Marthon, and E. Cubero-Castan, "An optimal multiedge detector for sar image segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 36, pp. 793–802, 1998.

[19] Q. Wei and D. Feng, "Extracting line features in sar images through image edge fields," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, pp. 540–544, 2016.

[20] Q. Wei, D. Feng, W. Zheng, and J. Zheng, "Rapid line-extraction method for sar images based on edge-field," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, pp. 1865–1869, 2017.

[21] C. Liu, R. Abergel, Y. Gousseau, and F. Tupin, "Lsdsar, a markovian a contrario framework for line segment detection in sar images," *Pattern Recognition*, vol. 98, 2020.

[22] ——, "A line segment detector for sar images with controlled false alarm rate," in *2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018.

[23] A. Desolneux, L. Moisan, and J.-M. Morel, *From Gestalt Theory to Image Analysis. A Probabilistic Approach*. Springer, 2008.

[24] F. Dellinger, J. Delon, Y. Gousseau, J. Michel, and F. Tupin, "Sar-sift: A sift-like algorithm for sar images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, pp. 453–466, 2015.

[25] A. Desolneux, L. Moisan, and J.-M. Morel, "Meaningful alignments," *International Journal of Computer Vision*, vol. 40, pp. 7–23, 2000.

[26] C. Liu, F. Tupin, and Y. Gousseau, "Training cnns on speckled optical dataset for edge detection in sar images," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 170, pp. 88–102, 2020.

[27] S. Xie and Z. Tu, "Holistically nested edge detection," in *2015 IEEE International Conference on Computer Vision*, 2015, pp. 1395–1403.

[28] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutioonal features for edge detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1939–1946.

[29] S. Xie and Z. Tu, "Holistically-nested edge detection," *International Journal of Computer Vision*, vol. 125, pp. 3–18, 2017.

[30] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 898–916, 2011.

[31] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 193–202.

[32] I. Kokkinos, "Pushing the boundaries of boundary detection using deep learning," in *4th International Conference on Learning Representations (ICLR 2016)*, vol. 4, 2016.

[33] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 891–898.

[34] X. Glorot and Y. B. and, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[35] P. Dollar and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 1558–1570, 2015.

[36] P. Denis, J. H. Elder, and F. J. Estrada, "Efficient edge-based methods for estimating manhattan frames in urban imagery," in *Proceedings of 10th European Conference on Computer Vision (ECCV) 2008*, 2008, pp. 197–210.

[37] D. Massonnet and J. C. Souyris, "Imaging with synthetic aperture radar," *EPFL Press*, 2008.

[38] C. Wang, X. Wang, Y. Xu, B. Zhang, M. Jiang, S. Xiong, Q. Zhang, W. Li, and Q. Li, "A new likelihood function for consistent phase series estimation in distributed scatterer interferometry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, 2022.

[39] R. Abergel, S. Ladjal, F. Tupin, and J.-M. Nicolas, "A complex spectrum based sar image resampling method with restricted target sidelobes and statistics preservation," in *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2017.

[40] R. Abergel, L. Denis, S. Ladjal, and F. Tupin, "Subpixellic methods for sidelobes suppression and strong targets extraction in single look complex sar images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, pp. 759–776, 2018.

[41] R. Abergel, L. Denis, F. Tupin, S. Ladjal, and C.-A. Deledalle, "Resolution-preserving speckle reduction of sar images: The benefits of speckle decorrelation and targets extraction," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019.

[42] A. Lapini, T. Bianchi, F. Argenti, and L. Alparone, "Blind speckle decorrelation for sar image despeckling," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, pp. 1044–1058, 2014.

[43] A. Garzelli, B. Aiazzi, L. Alparone, F. Argenti, A. Arienzo, and C. Zoppetti, "Impact of a spatial decorrelation of the noise on the estimation accuracy of temporal changes in the scene from a couple of single-look sar images," in *Proceedings: Image and Signal Processing for Remote Sensing, SPIE Remote Sensing*, vol. 11533, 2020.

[44] E. Dalsasso, L. Denis, and F. Tupin, "How to handle spatial correlations in sar despeckling? resampling strategies and deep learning approaches," in *EUSAR 2021; 13th European Conference on Synthetic Aperture Radar*, 2021.