



HAL
open science

Low level feature detection in SAR images

Chenguang Liu

► **To cite this version:**

Chenguang Liu. Low level feature detection in SAR images. Computer Vision and Pattern Recognition [cs.CV]. Institut Polytechnique de Paris, 2020. English. NNT : 2020IPPAT015 . tel-02861903

HAL Id: tel-02861903

<https://theses.hal.science/tel-02861903>

Submitted on 9 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT
POLYTECHNIQUE
DE PARIS

NNT : 2020IPPAT015

Thèse de doctorat



Low level feature detection in SAR images

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 Informatique, données, IA
Spécialité de doctorat : Signal, Images, Automatique et robotique

Thèse présentée et soutenue à Paris, le Date, par

CHENGUANG LIU

Composition du Jury :

Julie Delon Professor, Université Paris Descartes	Président
Sylvie Le Hégarat-Masclé Professor, Université de Paris Sud	Rapporteur
Antoine Roueff Maitre de Conférence, École Centrale de Marseille	Rapporteur
Julie Delon Professor, Université Paris Descartes	Examineur
Florence Tupin Professor, Télécom Paris	Directeur de thèse
Yann Gousseau Professor, Télécom Paris	Co-directeur de thèse
Rémy Abergel Research Engineer, Université Paris Descartes	Invité

Résumé

Dans cette thèse, nous développons des détecteurs de caractéristiques de bas niveau pour les images radar à synthèse d'ouverture (SAR) afin de faciliter l'utilisation conjointe des données SAR et optiques. Les segments de droite et les bords sont des caractéristiques de bas niveau importantes dans les images qui peuvent être utilisées pour de nombreuses applications comme l'analyse ou le stockage d'images, ainsi que la détection d'objets. Alors qu'il existe de nombreux détecteurs efficaces pour les structures bas-niveau dans les images optiques, il existe peu de détecteurs de ce type pour les images SAR, principalement en raison du fort bruit multiplicatif. Dans cette thèse, nous développons un détecteur de segment de droite générique et un détecteur de bords efficace pour les images SAR.

Le détecteur de segment de droite proposé, nommé LSDSAR, est basé sur un modèle Markovien *a contrario* et le principe de Helmholtz, où les segments de droite sont validés en fonction d'une mesure de significativité. Plus précisément, un segment de droite est validé si son nombre attendu d'occurrences dans une image aléatoire sous l'hypothèse du modèle Markovien *a contrario* est petit. Contrairement aux approches habituelles *a contrario*, le modèle Markovien *a contrario* permet un filtrage fort dans l'étape de calcul du gradient, car les dépendances entre les orientations locales des pixels voisins sont autorisées grâce à l'utilisation d'une chaîne de Markov de premier ordre. Le détecteur de segments de droite basé sur le modèle Markovien *a contrario* proposé LSDSAR, bénéficie de la précision et l'efficacité de la nouvelle définition du modèle de fond, car de nombreux segments de droite vraie dans les images SAR sont détectés avec un contrôle du nombre de faux détectés. De plus, peu de réglages de paramètres sont requis dans les applications pratiques de LSDSAR.

Dans la deuxième partie de cette thèse, nous proposons un détecteur de bords basé sur l'apprentissage profond pour les images SAR. Les contributions du détecteur de bords proposé sont doubles : 1) sous l'hypothèse que les images optiques et les images SAR réelles peuvent être divisées en zones constantes par morceaux, nous proposons de simuler un ensemble de données SAR à l'aide d'un ensemble de données optiques ; 2) Nous proposons d'appliquer un réseaux de neurones convolutionnel classique, HED, directement sur les champs de magnitude des images. Ceci permet aux images de test SAR d'avoir des statistiques semblables aux images optiques en entrée du réseau. Plus précisément, la distribution du gradient pour toutes les zones homogènes est la même et la distribution du gradient pour deux zones homogènes à travers les frontières ne dépend que du rapport de leur intensité moyenne valeurs. Le détecteur de bords proposé, GRHED permet d'améliorer significativement l'état de l'art, en particulier en présence de fort bruit (images 1-look).

Abstract

In this thesis we develop low level feature detectors for Synthetic Aperture Radar (SAR) images to facilitate the joint use of SAR and optical data. Line segments and edges are important low level features in images which can be used for many applications like image analysis, image registration and object detection. Contrarily to the availability of many efficient low level feature detectors dedicated to optical images, there are few efficient line segment detector and edge detector for SAR images mostly because of the strong multiplicative noise. In this thesis we develop a generic line segment detector and an efficient edge detector for SAR images.

The proposed line segment detector which is named as LSDSAR, is based on a Markovian *a contrario* model and the Helmholtz principle, where line segments are validated

according to their meaningfulness. More specifically, a line segment is validated if its expected number of occurrences in a random image under the hypothesis of the Markovian *a contrario* model is small. Contrarily to the usual *a contrario* approaches, the Markovian *a contrario* model allows strong filtering in the gradient computation step, since dependencies between local orientations of neighbouring pixels are permitted thanks to the use of a first order Markov chain. The proposed Markovian *a contrario* model based line segment detector LSDSAR benefit from the accuracy and efficiency of the new definition of the background model, indeed, many true line segments in SAR images are detected with a control of the number of false detections. Moreover, little parameter tuning is required in the practical applications of LSDSAR.

The second work of this thesis is that we propose a deep learning based edge detector for SAR images. The contributions of the proposed edge detector are two fold : 1) under the hypothesis that both optical images and real SAR images can be divided into piecewise constant areas, we propose to simulate a SAR dataset using optical dataset ; 2) we propose to train a classical CNN (convolutional neural network) edge detector, HED, directly on the gradient fields of images. This, by using an adequate method to compute the gradient, enables SAR images at test time to have statistics similar to the training set as inputs to the network. More precisely, the gradient distribution for all homogeneous areas are the same and the gradient distribution for two homogeneous areas across boundaries depends only on the ratio of their mean intensity values. The proposed method, GRHED, significantly improves the state-of-the-art, especially in noisy cases such as 1-look images.

Table des matières

Notations	7
1 Context	13
1.1 Introduction	13
1.2 Motivations	14
1.3 Contributions	14
1.3.1 LSDSAR, a Markovian <i>a contrario</i> model for line segment detection in SAR images	14
1.3.2 GRHED, a CNN based edge detector for SAR images	15
1.4 Organization of the thesis	16
2 SAR basics, statistics of SAR images and data used in this thesis	19
2.1 Principles of SAR and its applications	19
2.2 Statistics of SAR images	19
2.3 Data used in this thesis	21
I Line segment detection in SAR images	23
3 Introduction	25
3.1 Line segment detectors for SAR images	25
3.2 Line segment detectors for optical images	26
3.3 Adapting LSD to SAR images	27
4 LSD, a line segment detector with false detection control	29
4.1 Region growing	29
4.2 Rectangular approximation	30
4.3 Line segment validation	31
5 LSDSAR, a generic line segment detector for SAR images	35
5.1 Gradient computation in SAR images	35
5.2 The choice of the gradient computation methods	37
5.2.1 The property of constant false alarm rate (CFAR)	37
5.2.1.1 Check the CFAR property in pure noise images	37
5.2.1.2 Check the CFAR property in images of two homogeneous areas across boundaries	39
5.2.1.3 Conclusion on the CFAR property	41

5.2.2	Detection capability of the gradient magnitude : a comparison of existing gradient computation methods	41
5.2.2.1	The influence of the format of SAR images : amplitude or intensity	42
5.2.2.2	Choice of the weighting function for 2-direction methods	44
5.2.2.3	The comparison of 2-direction and multi-direction methods	47
5.2.3	Accuracy and distribution of the local orientation	49
5.2.3.1	Comparison in the accuracy of local orientation	50
5.2.3.2	Check the distribution of local orientations in noise	51
5.3	A Markovian <i>a contrario</i> model	52
5.4	The complete LSDSAR algorithm	56
5.5	Parameter settings	58
6	Experiments	63
6.1	Ability to control NFA	63
6.1.1	The number of false detections in pure noise images	63
6.1.2	The influence of image size on the Number of False Alarms (NFA)	64
6.1.3	The influence of the ε value on the ability of LSDSAR to control the Number of False Alarms (NFA)	64
6.2	Comparison in synthetic edge images	65
6.3	Comparison between LSDSAR and IEFA on real SAR images	71
6.4	Summary of LSDSAR	75
6.5	Inefficiency of LSDSAR in 1-look and low-contrasted situations	76
II	Edge detection in SAR images using CNNs	79
7	Introduction	81
7.1	Existing approaches	81
7.2	Training CNNs for edge detection in SAR images	82
8	Presentation of the HED method and of the training dataset	83
8.1	Details of the HED method	83
8.1.1	Network architecture	83
8.1.2	Training and testing phases	83
8.2	Speckled optical dataset	85
8.3	What HED has learned through training	86
8.3.1	Testing HED and HED-log in pure noise images	86
8.3.2	Testing HED and HED-log in synthetic edge images	90
8.3.2.1	Synthetic edge images with different amplitude ratio contrasts	91
8.3.2.2	Synthetic edge images with the same amplitude ratio contrast but with different mean intensity values.	94
8.3.2.3	Conclusion	98
9	GRHED, introducing a hand-crafted layer before the usual CNNs	99
9.1	The differences between speckled optical images and real SAR images	99
9.2	GRHED, training HED on the gradient magnitude field computed by GR	102
9.2.1	Gradient by Ratio (GR)	103
9.2.2	Interest of using GR to feed the network	103

9.2.3	Benefits of GRHED	106
10	Experiments	109
10.1	Normalization of the SAR data	110
10.2	Comparison in 1-look synthetic edge images	110
10.3	Comparison of different algorithms on the speckled optical images in BSDS500- speckled	113
10.4	Comparison of different methods in 1-look SAR images	115
10.4.1	Setting thresholds according to a given probability of false alarms . .	115
10.4.2	Comparison of different algorithms in realistic SAR images	116
10.4.3	Comparison of different algorithms in two 1-look real SAR images .	119
10.5	Summary of GRHED	122
10.6	Improving line segment detection results with GRHED	122
11	Summary of the thesis	127
11.1	Conclusion	127
11.2	Perspectives	128
	Bibliographie	135

Notations

$z = z e^{j\phi}$	SAR data, a complex backscattered electro-magnetic field with $ z $ representing its amplitude and ϕ representing its phase.
L	the number of looks for a SAR image.
$\langle \mathbb{I} \rangle$	the mean intensity value of a homogeneous area.
$\Gamma(\cdot)$	the Gamma function.
s	the speckle noise.
(x, y)	pixel location.
ε	the threshold for NFA.
I	an image.
I_0	a pure random image.
$G^h(x, y)$	horizontal components of gradient for pixel located at (x, y) .
$G^v(x, y)$	vertical components of gradient for pixel located at (x, y) .
$G(x, y)$	gradient magnitude for pixel located at (x, y) .
ang_i	the angle of the local orientation for the i -th pixel.
(c_x, c_y)	the center location of the approximated rectangle r .
r	the rectangle r , the candidate line segment.
\mathbb{M}	the matrix used to estimate the angle of the rectangle.
\mathcal{H}_0	the <i>a contrario</i> model used in LSD.
$n(r)$	the total number of pixels in rectangle r .
$k(r)$	the number of aligned pixels in rectangle r .

$NFA(r)$	the number of false alarms associated with rectangle r .
$N_{\mathcal{R}}$	the total number of possible rectangles.
$\mathbb{P}(k_0(r) \geq k(r))$	the probability of rectangles with $n(r)$ pixels having equal or greater number of aligned pixels than $k(r)$.
$M \times N$	the size of the image.
τ	the angle tolerance.
p	the probability of a pixel to be aligned to the rectangle up to a certain angle tolerance.
κ	the number of aligned pixels in rectangles with n pixels.
k_n	the minimum number of aligned pixels for a rectangle with n pixels to be ε -meaningful.
m_n	the number of rectangles with n pixels.
N_{max}	the maximum possible number of pixels in a rectangle.
r_i	the i -th rectangle in a random image.
\mathcal{X}_{r_i}	a random variable representing whether rectangle r_i is ε -meaningful.
$m(\mathcal{R})$	the number of ε -meaningful rectangles in a realization.
P	the number of directions used to compute the gradient.
$R^i(x, y)$	the ratio of average at pixel located at (x, y) along the i -th direction.
$M_1^i(x, y)$	the arithmetic mean on one side of windows of pixel located at (x, y) , along the i -th direction.
$M_2^i(x, y)$	the arithmetic mean on the other side of windows of pixel located at (x, y) , along the i -th direction.
$T^i(x, y)$	the normalized ratio along the i -th direction for pixel located at (x, y) .
$G_{roa}(x, y)$	the gradient magnitude computed by ROA in pixel located at (x, y) .
$R^h(x, y)$	the ratio of exponentially weighted average computed along the horizontal direction for pixel located at (x, y) .
$R^v(x, y)$	the ratio of exponentially weighted average computed along the vertical direction for pixel located at (x, y) .
$m_1^h(x, y)$	exponentially weighted average on one side of windows along the horizontal direction.
$m_2^h(x, y)$	exponentially weighted average on the other side of windows along the horizontal direction.
$T^h(x, y)$	the normalized ratio of exponentially weighted average for pixel located at (x, y) along the horizontal direction.

$T^v(x, y)$	the normalized ratio of exponentially weighted average for pixel located at (x, y) along the vertical direction.
$G_{roewa}(x, y)$	the gradient magnitude computed by ROEWA.
$G_{gr}(x, y)$	the gradient magnitude computed by GR.
$ang_{gr}(x, y)$	the angle of the gradient orientation computed by GR.
α	the weighting parameter for the exponential weight function in ROEWA and GR.
X_t	the random variable representing whether the t -th pixel is aligned.
$\mathbb{P}(X_1 = x_1 X_0 = x_0)$	the transition probability of the first order Markov chain.
\mathcal{H}'_0	the Markovian <i>a contrario</i> model.
$NFA'(r)$	the number of false alarms associated with rectangle r in a pure random image following \mathcal{H}'_0 .
$\mathbb{P}_{\mathcal{H}'_0}(k_0(r) \geq k(r))$	the probability of a rectangle with $n(r)$ pixels having equal or greater number of aligned pixels than $k(r)$.
$Y_t = \sum_{j=t}^n X_j$	the number of aligned pixels in the tail of the pixel sequence starting from the t -th pixel.
$\mathbb{P}(Y_t \geq k)$	the probability of the tail of the pixel sequence starting from the t -th pixel having more than k aligned pixels.
$\mathbb{P}(Y_{t+1} \geq k' X_t = x)$	the conditional probability of the tail of the sequence starting from the $(t + 1)$ -th pixel having more than k' aligned pixels.
$w(r)$	the width of rectangle r
$\ell(r)$	the length of rectangle r
u	an image in the training dataset.
u_j	the j -th pixel in image u .
$ u $	the total number of pixels in image u .
\mathbb{G}	the edge ground truth associated with u .
\mathbb{G}_j	the edge ground truth of the j -th pixel in image u .
$ \mathbb{G} $	the total number of pixels in \mathbb{G} .
\mathbb{W}	the collection of parameters in standard network layers excluding those of side layers.
\mathbb{W}_{side}	the collection of parameters in all side layers.
\mathbb{W}_{side}^m	the collection of parameters in the m -th side layer.

$\mathcal{L}_{side}(\mathbb{W}, \mathbb{W}_{side})$	the weighted sum of the losses from side outputs.
β_m	the weights for the loss of the m -th side output.
$\ell_{side}^{(m)}(\mathbb{W}, \mathbb{W}_{side}^{(m)})$	the loss of the m -th side output.
λ	the balance weight between the loss from positive and negative classes.
$ \mathbb{G}_+ $	the number of edge pixels in ground truth.
$ \mathbb{G}_- $	the number of non-edge pixels in ground truth.
$\hat{\mathbb{G}}_{side}^{(m)}$	the edge prediction map of the m -th side layer.
$\hat{a}_j^{(m)}$	the activation value of the j -th pixel in the output of the m -th side layer.
$\hat{A}_{side}^{(m)}$	the activations of the output in the m -th side layer.
$\sigma(\cdot)$	the sigmoid function.
$\hat{\mathbb{G}}_{fuse}$	the edge prediction map of the fused output.
h_m	the fusion weight of the output in the m -th side layer.
$\mathcal{L}_{fuse}(\mathbb{W}, \mathbb{W}_{side}, h)$	the loss of the fused output.
$\text{Dist}(\cdot)$	the distance between fused output and ground truth.
$(\mathbb{W}, \mathbb{W}_{side}, h)^*$	the optimized parameters after training.
$\hat{\mathbb{G}}_{HED}$	the final output of HED.
$\langle \mathbb{I}_{opt} \rangle$	the mean intensity value of the speckled optical dataset.
$\langle \mathbb{I}_{SAR} \rangle$	the mean intensity value of a real SAR image.
\hat{u}	SAR image with normalized mean intensity value.
t_{mode}	the mode of a distribution.

Chapitre 1

Context

1.1 Introduction

Remote sensing is the process of acquiring information about an object, an area or a phenomenon without any physical contact. Remote sensing has many advantages including the ability to acquire information from inaccessible areas such as oceans and deep valleys, allowing the coverage of large areas, allowing repetitive coverage, the ability to provide data in all weather conditions and so on. Due to the valuable properties of remote sensing, they can be used in many applications such as land use and land cover (classification, change detection), agriculture (crop health monitoring, crop acreage estimation), water resources (distributed hydrological modelling, rainfall monitoring), oceanographic applications (identification of potential fishing zones, study of the sea surface temperature). urban and regional applications (urban sprawl analysis), environmental applications (study of urban heat islands, landfill site identification), disaster management applications (landslide hazard zonation, flood modelling and inundation studies). Many missions have been launched recently for various applications, for example, the Aeolus mission launched by ESA in 2018 to improve the quality of weather forecasts, and to advance our understanding of atmospheric dynamics and climate processes, the Sentinel-5P mission launched by ESA in 2017 for atmospheric monitoring, the Sentinel-3 mission launched in 2016 for marine observation, the Sentinel-2 mission launched by ESA in 2015 for land monitoring, the Sentinel-1 mission launched by ESA in 2014 for land and ocean monitoring.

The new generation of remote sensing sensors provide data with improved resolution and with high temporal frequency. One of the main challenges is the joint use and combination of such data, especially the combination of Synthetic Aperture Radar (SAR) and optical data. SAR is well known for its all-weather and all-time property since it is able to penetrate darkness, clouds and rain. It provides valuable information for the applications like urban monitoring, military surveillance, digital elevation model computation, ground movement monitoring, etc. In comparison, optical data are available only with good weather conditions but are easier to understand for human being since they are close to natural images. These data have nowadays high temporal repetition frequency and multitemporal datasets are available. One of the major challenge for data combination is the reliable extraction of common features. Low level features like line segments or edges are specially difficult to extract from SAR data because of the speckle noise. The aim of this PhD work is to develop useful methods to extract reliable features on SAR data.

1.2 Motivations

Line segments and edges are important low level features in images. Both line segments and edges can be used as low level features for tasks like object detection/recognition, image segmentation and image analysis. What is more important, both line segments and edges can be used for image registration Chen et al. [2014], Zhang et al. [2015], Zhao et al. [2018]. Image registration is a usual preprocessing step for the joint use of information from images across views, especially for the combination of images from different sensors, like SAR and optical images. Many researches have been dedicated to SAR and optical image registration using line segments and edges Sui et al. [2015], Xiong et al. [2016], Shen et al. [2017], Zhang et al. [2017], while the accuracy of registration relies a lot on the robustness and reliability of the line segment detectors and the edge detectors. Many methods have been proposed for line segment detection and edge detection in natural images and these methods can be easily applied to remote sensing optical images due to their similar statistics. However, line segment detection and edge detection in SAR images remain challenging tasks, mostly because of the strong multiplicative noise. Those methods developed for optical images can not be directly applied to SAR images because of the difference in the nature of the noise. The optical images are usually contaminated by the additive Gaussian noise, while SAR images are usually contaminated by multiplicative noise. In order to facilitate the combination of SAR and optical data, a reliable and generic line segment detector as well as an edge detector will be very helpful.

Challenges

Line segment detectors and edge detectors usually depend on the computation of the gradient (or some similar operators) in each pixel. Because of the strong multiplicative noise in SAR images, the gradient magnitudes in homogeneous areas usually have large variations, even for a method suitable for SAR images. Thus, it is difficult to distinguish between noise pixels (which are corresponding to pixels in homogeneous areas), and edge pixels (especially for low-contrasted edges). Important edge pixels could be lost if a high threshold is used, though a high threshold could be efficient for suppressing the speckle noise. If the threshold is low, post-processing will be challenging since the number of false detections is difficult to control. The most challenging problem for developing line segment detectors and edge detectors is how to ensure reasonable amount of correct detections while controlling the number of false detections.

1.3 Contributions

The main contributions of this thesis lie in two parts : 1) we propose a generic line segment detector for SAR images, which is based on a Markovian *a contrario* model and the Helmholtz principle Desolneux et al. [2008]; 2) we propose a CNN-based edge detector for SAR images leveraging a SAR dataset simulated from an optical dataset. We briefly describe them in the following.

1.3.1 LSDSAR, a Markovian *a contrario* model for line segment detection in SAR images

The first work of the Ph.D is to develop a generic line segment detector for SAR images. Borrowing the idea from the famous line segment detector (LSD Grompone von Gioi et al.

[2010]) for natural images, we propose to develop a LSD-like line segment detector for SAR images.

LSD relies on the grouping of pixels sharing the same local orientations (the local orientation is the direction perpendicular to the gradient orientation), after which a set of line support regions are obtained. All line support regions are then approximated using rectangles. Therefore, all rectangles are candidate line segments. Finally, all candidate line segments are validated based on the *a contrario* model Desolneux et al. [2000] and Helmholtz principle Desolneux et al. [2008]. Specifically, a candidate line segment is validated according to its meaningfulness. If a line segment is unlikely to happen in a random image, then it is considered as a meaningful one and is accepted, otherwise it will be rejected since it is common in random images. More precisely, according to the definition of meaningfulness in Desolneux et al. [2000], a line segment is considered to be ε -meaningful if the expected number of its occurrences in a random image is less than ε . Thus, for each candidate line segment, its Number of False Alarms Desolneux et al. [2000], Grompone von Gioi et al. [2010] is computed under the hypothesis of the *a contrario* model and compared with a chosen threshold ε . If its NFA is smaller than ε , then it is accepted as a meaningful one, and it is rejected otherwise. The advantages of LSD are as follows : first, the null hypothesis (the *a contrario* model) against which a line segment is detected does not depend on the observed image, and thus it is a generic one. Second, ε can be safely set to 1 which ensures that all detected segments are 1-meaningful. What's more important, using $\varepsilon = 1$ ensures that on average, no more than 1 false detection could be done by chance in a random image.

The advantages of the LSD detector are obvious since it ensures all detected line segments are meaningful and the number of false detections is controlled. What is more important, no parameter tuning is required in general applications of LSD, except for some specific situations where one parameter (among six) may need to be tuned, thanks to the generic priori null hypothesis as well as the insensitiveness of the NFA associated with a rectangle to the value of ε . The aim of our work is to develop a similar line segment detector for SAR images. First, the gradient computation step is replaced with a gradient computation method suitable for SAR images because the usual gradient computation methods dedicated to natural images do not have a constant false alarm rate for SAR images Touzi et al. [1988]. However, a large averaging window (with size 21×21 pixels for example, much larger than the 2×2 mask used in LSD) is usually necessary to reduce the influence of speckle noise and helpful for detecting important line segments in SAR images. The independence hypothesis in the usual *a contrario* model is not followed. By introducing a first order Markov chain in the *a contrario* model to take into account the dependencies between the local orientations of neighbouring pixels, we propose a Markovian *a contrario* model, against which a meaningful line segment is detected. The proposed line segment detector LSDSAR is a generic line segment detector for SAR images and has the same advantages as LSD.

1.3.2 GRHED, a CNN based edge detector for SAR images

Edge detection in SAR images remains a challenging task, especially in the most challenging 1-look situation. A global threshold which is able to give good counterbalance in preserving important edges and suppressing the influence of speckle noise seems inaccessible in general case. Therefore, a gradient computation method, which can separate the edge pixels and noise pixels more efficiently is required. Observing the success of CNN (Convolu-

tional Neural Network) based edge detectors in natural images, we study the applicability of these CNN based edge detectors to SAR images. One crucial factor that contributes to the success of CNNs is a large dataset for training, while there is still no training dataset for edge detection in SAR images. However, edges correspond to changes in local brightness or textures in grayscale images and not to the kinds of targets in images. The definition of edges in optical images should be similar to that in SAR images. Therefore, we propose to simulate a SAR dataset using an optical dataset BSDS500 Arbelaez et al. [2011] to train CNNs for edge detection in SAR images. Under the hypothesis that both optical and SAR images can be divided into piecewise constant areas, the main gap between simulated SAR images and real SAR images lies in the differences in the possible mean intensity values of homogeneous areas, where some homogeneous areas with high mean intensity values (higher than 255^2) in real SAR images do not exist in the simulated SAR dataset. In order to cope with the gap, we propose to train CNNs on the gradient magnitude fields of images and apply the trained models to the gradient magnitude fields of testing images. By computing the gradient with a ratio based method, the gradient distribution of homogeneous areas will be the same regardless of their mean intensity values. The gradient distribution of two homogeneous areas across boundaries depends only on the ratio of their mean intensity values. Therefore, the differences between the gradient magnitude fields of speckled optical images and real SAR images are small. The proposed CNN based edge detector GRHED is more efficient than the traditional methods in highlighting true edge pixels and suppressing noise pixels. With the gradient computation done by GRHED, a simple non-maxima suppression and a plain thresholding are enough to obtain reasonable edge maps, even in the most challenging 1-look situations. What is more important, the models trained on the gradient magnitude field of speckled optical images can be directly applied to the gradient magnitude fields of real SAR images while detecting many true edge pixels with a probability of false alarms control. Furthermore, the performances of GRHED are insensitive to the choice of the threshold. GRHED with a threshold chosen from a relatively large range are all able to obtain near-optimal edge maps. The proposed CNN edge detector GRHED largely outperforms existing edge detectors for SAR images.

1.4 Organization of the thesis

This thesis can be divided into two parts : in part I, we describe the Markovian *a contrario* model based line segment detector LSDSAR for SAR images ; in part II, we provide details about the CNN based edge detector GRHED. Before these two parts, we first provide the basic principles of SAR, the statistics of SAR images and the details about the real SAR data that will be used in this thesis in chapter 2. Part I consists of four chapters. Chapter 3 describes the existing line segment detectors for both optical and real SAR images, and describe the motivation of adapting LSD to SAR images. Chapter 4 describes the details of LSD detector. In chapter 5, we describe the proposed LSDSAR line segment detector. A thorough analysis and comparison of existing gradient computation methods is provided. In addition, the definition of the proposed Markovian *a contrario* model is also given in this chapter. In chapter 6, we provide a comprehensive demonstration of the proposed line segment detector in pure noise images, synthetic edge images and real SAR images. Some conclusions for the work of line segment detector are given at the end of this chapter. Part II is also constituted by four chapters. In chapter 7, we introduce the existing edge detectors for SAR images and propose to develop a CNN based edge detector for SAR images. Chapter 8 gives a brief description of HED network architecture

and its formulation. The details about the simulated SAR dataset using optical dataset are also provided in this chapter. Besides, we give an experimental analysis of what HED can learn from training and deduce that the convolutional layers in HED mainly learn to process samples drawn from many distributions. In chapter 9, we first analyze the differences between speckled optical images and real SAR image. In order to cope with the gap between training and testing data, we propose to train HED on the gradient magnitude fields of speckled optical images and apply the trained model to the gradient magnitude fields of testing images, and give the motivation and benefits to do this. In chapter 10, the advantages of GRHED are demonstrated in several 1-look synthetic edge images, 200 1-look speckled optical images, one realistically simulated SAR image and two 1-look real SAR images. The summary of the work GRHED are given at the end of this chapter. We summarize the thesis and give some perspectives in chapter 11.

Chapitre 2

SAR basics, statistics of SAR images and data used in this thesis

2.1 Principles of SAR and its applications

Synthetic Aperture Radar (SAR) is an active sensor providing illumination in the form of microwaves. It is typically mounted on aircraft or spacecraft and its origins use an advanced form of side looking airborne radar. SAR sensors send successive pulses of electromagnetic waves and the echoes reflected from the target scene for each pulse are recorded. Since the location of antenna relative to the objects in the target scene is changing, echoes from multiple antenna positions can be combined through signal processing techniques. In this way, a large synthetic antenna aperture can be created and thus images with higher spatial resolution can be obtained. Pulses of electromagnetic waves are emitted in the direction perpendicular to the flight path. The direction perpendicular to the flight path is the range direction and the direction parallel to the flight path is the azimuth direction.

The ground resolution is defined as the minimum distance between two object points on the ground that can be distinguished in an image. In the range direction, the range resolution depends on the frequency bandwidth of the pulse and the incidence angle of the beam (usually defined as the angle between the midpoint of the swath and nadir). Therefore, better range resolution can be achieved by increasing the frequency bandwidth. In the azimuth direction, the azimuth resolution depends on the aperture length. Because of the motion of the radar antenna, a longer aperture can be synthesized using the Doppler spread of the echo signal. Therefore, the azimuth resolution of SAR can be much better than a conventional one.

Since illumination is provided by the emitting antenna, SAR is independent of day time. It has day and night imaging capability. In addition, pulse frequencies can be selected so that SAR is independent of weather. Therefore, SAR is widely used in remote sensing applications such as terrain discrimination, subsurface imaging, volcano and earthquake monitoring. Besides, SAR is also a very useful tool for applications like environment monitoring, urban growth and military surveillance.

2.2 Statistics of SAR images

In a SAR image, each pixel (a single resolution cell) corresponds to an area on the ground with a certain size, while many different scatterers may exist in this area. Com-

pared to the size of the area, the wavelength of the electromagnetic wave is very small and thus it allows the electromagnetic wave to interact with each individual scatterer. The echoes are reflected by those scatterers and they interfere each other either constructively or destructively. Constructive interferences produce a bright pixel in the image while destructive interferences produce a dark pixel.

Each pixel in a SAR image can be represented as a complex value. Due to the coherent imaging system, SAR data presents the well known speckle phenomenon. In the case of rough surfaces (roughness being defined compared to the wavelength) Goodman has developed a model of backscattering Goodman [1975]. The real and imaginary components can both be modeled as independently and identically Gaussian distributed variables as a consequence of central limit theorem, under the assumption in each resolution cell that : 1) no dominant scatterer exist ; 2) the scatterers are statistically identical and independent and there are a large number of scatterers ; 3) the phases of echoes are uniformly distributed. For each pixel value z , the probability can thus be modeled as a 2-dimensional multivariate normal distribution. Another way to describe the complex value for each pixel is $z = |z|e^{j\phi}$ ($j^2 = -1$), with $|z|$ representing the amplitude, $|z|^2$ representing the intensity, and ϕ representing the phase. In the case of fully developed speckle model of Goodman, ϕ is uniformly distributed. Therefore, when dealing with a single image, the phase does not carry information and only amplitude or intensity information is exploited.

Original SAR images are 1-look images, but a pre-processing of multi-looking can be applied to reduce the signal fluctuations. This multi-looking operator is given by :

$$y = \frac{1}{L} \sum_{i=1}^L |z_i|^2, \quad (2.1)$$

L being the number of looks and z_i representing the pixel in the i -th 1-look image. It corresponds to an incoherent averaging of the backscattered values in the power domain to reduce the noise.

Following the model of Goodman [1975] of fully developed speckle previously mentioned, it can be shown that the amplitude of the backscattered electro-magnetic field of a homogeneous area with mean intensity $\langle \mathbb{I} \rangle$ follows a Nakagami distribution :

$$f(t|\langle \mathbb{I} \rangle) = \frac{2}{\Gamma(L)} \left(\frac{L}{\langle \mathbb{I} \rangle} \right)^L t^{2L-1} e^{-(Lt^2/\langle \mathbb{I} \rangle)}, \quad (2.2)$$

L being a sensor or data parameter and $\Gamma : x \mapsto \int_0^{+\infty} t^{x-1} e^{-t} dt$ is the Gamma function. For images with the best resolution, $L = 1$ and amplitude of a homogeneous area follows a Rayleigh distribution. Another way of modeling the 1-look data is the multiplicative model : $t = \sqrt{\langle \mathbb{I} \rangle} s$, s representing the speckle noise and following the Rayleigh distribution given in eq. (2.2) with $\langle \mathbb{I} \rangle = 1$ and $L = 1$. This model is not exactly verified by real SAR data since the pixels are usually spatially correlated (due to over-sampling and spectrum apodization for lobe reduction).

The intensity of the backscattered electromagnetic field of a homogeneous area with mean intensity $\langle \mathbb{I} \rangle$ follows a Gamma distribution :

$$f(y|\langle \mathbb{I} \rangle) = \frac{1}{\Gamma(L)} \left(\frac{\langle \mathbb{I} \rangle}{L} \right)^{-L} y^{L-1} e^{-\frac{yL}{\langle \mathbb{I} \rangle}}. \quad (2.3)$$

In this thesis, unless explicitly mentioned, we will work on amplitude SAR images. We will show by experiments that the usual gradient computation methods for SAR images

keep its valuable CFAR property (constant false alarm rate) in both amplitude and intensity SAR images, without influencing their accuracy.

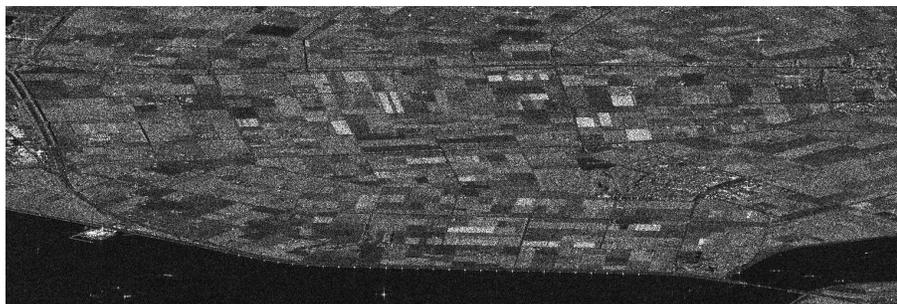
2.3 Data used in this thesis

In order to test the performances of the developed low level feature detectors, several kinds of images are used to give a comprehensive analysis of the proposed algorithms. Details about the synthetic SAR images (simulated data) will be described when they are used.

In this section, we give details about the real SAR images as shown in table 2.1. We have used a middle resolution image acquired by Sentinel-1 (ESA, European Space Agency) on the Netherlands (Lelystad city). This image is on a rather flat area with many fields, roads and channels and some small urban areas as can be seen Figure 2.1. The other image is a very high resolution image acquired by TerraSAR-X on San-Francisco (USA) it is a very dense urban areas with a harbor where boats can be seen (figure 2.2).

TABLE 2.1 – 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	1m
pixel-spacing (range)	14m	1m



Sentinel-1 image (Lelystad)

FIGURE 2.1 – The Sentinel-1 image (Lelystad) used in this thesis. The size of the image are 1024×3072 pixels.



TerraSAR-X image (San Francisco)

FIGURE 2.2 – The TerraSAR-X image (San Francisco) used in this thesis. The size of the image are 2048×2048 pixels.

PART I

Line segment detection in SAR images

Chapitre 3

Introduction

Two kinds of linear structures can be distinguished in Synthetic Aperture Radar (SAR) images : narrow bands of pixels having a given width, usually simply called linear features, and straight subparts of edges, that we will call line segments. The detection of linear features (narrow bands of pixels having a given width Hellwich and Mayer [1996]) in SAR images has received a lot of attention, typically in view of the detection of road networks Tupin et al. [1998], Chanussot et al. [1999], Medeiros et al. [2003]. In contrast, reliably detecting line segments (straight subpart of edges) is still an open problem. Nevertheless, line segments are 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. Since the aim of the thesis is to develop useful tools for the joint use of SAR and optical data, and line segments are important low level features which can be useful for the combination of them, we first address the problem of line segment detection in SAR images.

3.1 Line segment detectors for SAR images

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 LSD detector Grompone von Gioi et al. [2010] and with a recent line segment detector AFM Xue et al. [2019] which is based on deep learning.

The usual way to detect line segments in SAR images is global and relies on the Hough transform Xu and Jin [2007], Sui et al. [2015], Wei and Feng [2016], Wei et al. [2017]. First, a constant false alarm rate edge detector, such as that described in Touzi et al. [1988], is applied to the image, followed by a Hough transform Duda and Hart [1972], Illingworth and Kittler [1987] 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 Skingley and Rye [1987], in the context of different applications. In Xu and Jin [2007], line segments are extracted 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 Sui et al. [2015], relying on line segments that are detected using a ratio-based gradient and the Hough transform. The same idea was previously explored in Palmann et al. [2008]. In Arnold-Bos et al. [2006], edge detection using phase symmetry Kovési et al. [1997] and wavelet correlations are followed by a Hough transform in order to detect ship wakes.

A common limitation of these 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 Touzi et al. [1988], Fjørtoft et al. [1998], Dellinger et al. [2015], 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 image-dependent. An interesting approach, which was recently proposed in Wei and Feng [2016], Wei et al. [2017], detects lines from the magnitude field instead of a binary edge map, but the subsequent detection tasks still require non-trivial parameter tunings.

In order to develop an efficient and reliable line segment detector for SAR images, we first give a brief description of existing line segment detectors for optical images in the next section and then propose to adapt one of those methods to SAR images.

3.2 Line segment detectors for optical images

The LSD Grompone von Gioi et al. [2010] detector, which is based the *a contrario* model Desolneux et al. [2000] and the Helmholtz principle Desolneux et al. [2008], is one of the most used line segment detector in the field of computer vision. Line segment candidates are obtained through a first grouping of pixels sharing the same local orientations and then the approximation with a rectangle for regions constituted by those pixels. Candidate line segments are finally verified according to the Helmholtz principle, that *no meaningful structure should be detected by chance in a random image*. A line segment is validated as ε -meaningful if the expected number of its occurrences in a random image following the *a contrario* model is less than ε . All ε -meaningful line segments are accepted as true line segments. Even though LSD is famous for its ability to control the number of false detections while requiring little parameter setting, and the performances of LSD is good in most situations, it fails to obtain satisfying results in some challenging cases. Motivated by this observation, several line segment detectors are proposed afterwards in order to achieve higher accuracy and location precision.

A Markov Chain Marginal Line Segment Detector (MCMLSD) Almazan et al. [2017] was proposed combining the advantages of both perceptual grouping and global Hough analysis. More specifically, perceptual grouping based methods are based on the geometric grouping of local cues, therefore, local decisions of line segments are made without taking global information into account. The Hough approaches are able to integrate information globally, while identifying the endpoints of line segments is not straightforward. The proposed MCMLSD approach aims at combining the advantages of both approaches. In the first step, a probabilistic Hough transform method is used to identify the lines in images. In the second step, by limiting the search for line segments from image space to lines, the problem of identifying maximum probability line segments is modeled as a hidden states labelling, where the hidden states represent the existence of line segments in each pixel and they follow a first order Markov Chain. Thanks to the assumption of Markov

chain, the hidden states labelling process can be computed in linear time using a dynamic programming algorithm.

A novel linelet-based line segment detector Cho et al. [2018] was addressed recently which exploits the intrinsic properties of the line segment. Specifically, line segments are assumed to be constituted by linelets where linelets are defined to be a set of horizontally or vertically connected pixels, which is caused by the digitization of lines in real images. During the detection process, linelets are first detected from the horizontally or vertically non-maxima suppressed gradient magnitude (in the horizontal direction, horizontally connected pixels are considered as linelets and for vertical direction, linelets are vertically connected pixels). Candidated lines are obtained through the grouping of neighbouring linelets which share the same intrinsic properties. For each candidate line segment, its angle is estimated through the use of an undirected graphical model for lines. The line segments are then validated by a probabilistic model using the cues from the gradient magnitude and gradient orientation. Finally, an aggregation step is used to connect disjoint line segments if they are likely from the same line.

A deep learning based line segment detector AFM Xue et al. [2019] is proposed leveraging the accuracy, robustness and efficiency of deep convolutional neural networks. The attraction field representation of line segments reformulates the line segment detection problem as region coloring problem. Each pixel in an image is assigned to one and only one line segment and thus the image is divided into many regions with each region corresponding to a line segment. Each pixel in the attraction field maps is defined as the attraction vector between the pixel and projected pixel in the corresponding line segment. The deep convolutional neural networks are trained to predict the attraction field maps for testing images. After obtaining the attraction field map for a testing image, a squeeze module is used to recover the original line segments. The detected line segment are then verified by an aspect ratio check to preserve only 'thin' line segment.

3.3 Adapting LSD to SAR images

Compared to line segment detection in optical images, line segment detection in SAR images remains challenging due to the strong multiplicative noise. In this work, we aim at developing a LSD-like line segment detector for SAR images, motivated by its ability to strictly control the number of false detections while requiring little parameter tuning. However, special care should be taken to do this. Before describing how to develop a LSD-like line segment detector for SAR images, we first describe the details about LSD in the next chapter and then describe the way to adapt it to SAR images.

Chapitre 4

LSD, a line segment detector with false detection control

The LSD detector, which is based on the *a contrario* model Desolneux et al. [2000] and the Helmholtz principle Desolneux et al. [2008], is a widely used state-of-the-art line segment detector. The LSD detector finds candidate line segments through a local grouping of pixels which share the same local orientation (the direction perpendicular to that of the gradient orientation), and then validates each candidate line segment according to its meaningfulness. The decision of the meaningfulness for a line segment is based on the Helmholtz principle which claims that *no meaningful structure should be observed by chance in a random image*. Therefore, if a candidate line segment is unlikely to be observed in a random image, then it should be a meaningful one. According to Desolneux et al. [2000], a line segment is accepted as ε -meaningful if the expectation of the number of its occurrences in a random image is smaller than ε . The validation step is done by comparing the Number of False Alarms Desolneux et al. [2000], Grompone von Gioi et al. [2010] corresponding to the candidate line segment with a chosen ε value. This validation step ensures that the expected number of false detections done by LSD is smaller than ε while ensuring that all detected line segments are ε -meaningful segments, that is to say, all detected line segments are not common in a random image.

The LSD detector relies on three steps : the region growing step, the rectangular approximation step and the line segment validation step. We detail each of these steps in the following.

4.1 Region growing

The first step of LSD is region growing. The aim of the region growing step is to group pixels sharing the same local orientation (the local orientation is perpendicular to the gradient orientation). Therefore, the image gradient should be computed at each pixel. For a pixel located at (x, y) in an image I , the horizontal and vertical components of the gradient are computed using a 2×2 mask as :

$$G^h(x, y) = \frac{I(x, y + 1) - I(x, y) + I(x + 1, y + 1) - I(x + 1, y)}{2} \quad (4.1)$$

$$G^v(x, y) = \frac{I(x + 1, y) - I(x, y) + I(x + 1, y + 1) - I(x, y + 1)}{2} \quad (4.2)$$

The gradient magnitude is then computed as

$$G(x, y) = \sqrt{G^h(x, y)^2 + G^v(x, y)^2} \quad (4.3)$$

The local orientation of the pixel located at (x, y) is defined to be perpendicular to the gradient orientation and computed as

$$\text{atan2}\left(G^v(x, y), -G^h(x, y)\right) \quad (4.4)$$

Since pixels with higher gradient magnitude are usually corresponding to more contrasted edges, the region growing step starts with pixels having higher gradient magnitude. In order to obtain a fast line segment detector, all pixels are sorted in descending order according to their gradient magnitude using a pseudo-ordering algorithm. Pixels are classified into 1024 bins of equal size between zero and the largest gradient magnitude. The region growing step starts from one seed pixel as a region, and initialize the region angle to be the local orientation of the seed pixel. Then the 8-connected neighbouring pixels of the region are tested. All tested pixels having the same orientation as the region up to a certain angle tolerance τ will be added to the region. Each time a pixel is added to the region, the region angle will be updated by :

$$\text{atan2}\left(\sum_i \sin(\text{ang}_i), \sum_i \cos(\text{ang}_i)\right), \quad (4.5)$$

where the index i goes through all pixels in the region and ang_i represents the angle of the local orientation of the i_{th} pixel. A line support region is obtained until no more pixels can be added into the region. Then the region growing step starts from another seed pixel searching for line support regions. The seed pixels are first chosen from the bin with the largest magnitude, and then from the second bin and so on until all the pixels have been visited. Notice that a pixel will be labeled as *USED* if it belongs to a line support region and will not be visited any more during the region growing step.

4.2 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. Given a line support region, noting $(x(i), y(i))$ the location of the i_{th} pixel and $G(x(i), y(i))$ its gradient magnitude, the center location (c_x, c_y) of the approximated rectangle r is computed as

$$c_x = \frac{\sum_{i \in \text{region}} G(x(i), y(i)) \cdot x(i)}{\sum_{i \in \text{region}} G(x(i), y(i))} \quad (4.6)$$

$$c_y = \frac{\sum_{i \in \text{region}} G(x(i), y(i)) \cdot y(i)}{\sum_{i \in \text{region}} G(x(i), y(i))} \quad (4.7)$$

The angle of the rectangle is defined as the angle between the horizontal axis and the first inertia axis of the region. Specifically, the angle of the rectangle r can be computed as the

angle of the eigenvector corresponding to the smallest eigenvalue of the matrix

$$\mathbb{M} = \begin{pmatrix} \mathbb{M}^{xx} & \mathbb{M}^{xy} \\ \mathbb{M}^{xy} & \mathbb{M}^{yy} \end{pmatrix}, \quad (4.8)$$

where

$$\mathbb{M}^{xx} = \frac{\sum_{i \in \text{region}} G(x(i), y(i)) \cdot (x(i) - c_x)^2}{\sum_{i \in \text{region}} G(x(i), y(i))} \quad (4.9)$$

$$\mathbb{M}^{yy} = \frac{\sum_{i \in \text{region}} G(x(i), y(i)) \cdot (y(i) - c_y)^2}{\sum_{i \in \text{region}} G(x(i), y(i))} \quad (4.10)$$

$$\mathbb{M}^{xy} = \frac{\sum_{i \in \text{region}} G(x(i), y(i)) \cdot (x(i) - c_x) \cdot (y(i) - c_y)}{\sum_{i \in \text{region}} G(x(i), y(i))} \quad (4.11)$$

The length and width of the rectangle are then defined as the smallest values allowing a complete covering of the pixels of the considered region.

4.3 Line segment validation

All approximated rectangles are candidate line segments and the last step of LSD is to decide whether a candidate rectangle is a true line segment or not according to its meaningfulness. The meaningfulness of the rectangle is checked through the use of an *contrario* method Desolneux et al. [2000] which evaluates how unlikely a similar structure is to happen by chance in a random image consisting of white noise, that from now on we will simply call a noise image. According to the Helmholtz principle, very few meaningful structures should be observed by chance in a noise image. Therefore, the more unlikely the rectangle is to be observed in a noise image, the more meaningful it is. Based on the definition in Desolneux et al. [2000], a rectangle is considered to be ε -meaningful if the expected number of its occurrence in a noise image is smaller than ε . What is considered as a *noise image* 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. Each random variable follows a uniform distribution over $[0, 2\pi]$.

Given the definition of the background model \mathcal{H}_0 , and for a given rectangle with a certain pixel configuration, it is possible to compute the expected number of similar rectangles that would appear in a noise image. 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. We consider a rectangle with size $n(r)$ in a random image and write $k_0(r)$ for the number of aligned points in this rectangle. The Number of False Alarms (NFA) Desolneux et al. [2000], Grompone von Gioi et al. [2010] associated to r is computed as

$$\text{NFA}(r) = N_{\mathcal{R}} \cdot \mathbb{P}(k_0(r) \geq k(r)), \quad (4.12)$$

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 Grompone von Gioi et al. [2010]. Following Desolneux et al. [2000], a rectangle r is said ε -meaningful if $\text{NFA}(r) \leq \varepsilon$.

The interest of such an approach is that the NFA defined in (4.12) satisfies the so-called NFA-property Desolneux et al. [2000], Grompone von Gioi et al. [2010], which can be written as

$$\forall \varepsilon > 0, \quad \mathbb{E}_{\mathcal{H}_0} [\#\{r, \text{NFA}(r) \leq \varepsilon\}] \leq \varepsilon, \quad (4.13)$$

where $\#$ stands for the cardinality of a set. This property means that using the threshold $\text{NFA}(r) \leq \varepsilon$, on average no more than ε rectangles will be detected in a random image I_0 following \mathcal{H}_0 . In general, ε can be set to 1, which guarantees that, on average at most one false detection should be done.

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

$$p = \frac{\tau}{\pi}, \quad (4.14)$$

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

$$\mathbb{P}_{\mathcal{H}_0}(k_0(r) \geq k(r)) = \mathcal{B}(n(r), k(r), p), \quad (4.15)$$

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

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

Therefore, the NFA in (4.12) can be explicitly computed using

$$\text{NFA}(r) = N_{\mathcal{R}} \cdot \mathcal{B}(n(r), k(r), p). \quad (4.16)$$

Proof of the NFA-property(Eq. (4.13)) :

Consider a rectangle r with n pixels in a random image I_0 following \mathcal{H}_0 , noting $k_0(r)$ its number of aligned pixels. This rectangle r is ε -meaningful if it contains at least k_n pixels, with

$$k_n = \min \{\kappa \in \mathbb{N}, N_{\mathcal{R}} \cdot \mathbb{P}(k_0(r) \geq \kappa) \leq \varepsilon\} \quad (4.17)$$

This is equivalent to say that a rectangle r with n pixels is ε -meaningful if its number of aligned pixels satisfies $k_0(r) \geq k_n$.

Noting m_n for the total number of rectangles with n pixels and N_{max} the maximal possible number of pixels for a rectangle in the image I_0 , we have

$$\sum_{n=0}^{N_{max}} m_n = N_{\mathcal{R}}. \quad (4.18)$$

Consider all possible rectangles $r_i \in \mathcal{R}$ ($1 \leq i \leq N_{\mathcal{R}}$) in a random image I_0 following the \mathcal{H}_0 model. Let the random variable \mathcal{X}_{r_i} be equal to 1 if rectangle r_i is a ε -meaningful rectangle during a realization, and 0 otherwise. Then $m(\mathcal{R}) = \sum_{r_i \in \mathcal{R}} \mathcal{X}_{r_i}$ represents the

number of ε -meaningful rectangles in a realization. The expected number of ε -meaningful rectangles in a random image I_0 following \mathcal{H}_0 models is

$$\begin{aligned}
 \mathbb{E}(m(\mathcal{R})) &= \mathbb{E}(\sum_{r_i \in \mathcal{R}} \mathcal{X}_{r_i}) \\
 &= \sum_{n=0}^{N_{max}} m_n \cdot \mathbb{P}_{\mathcal{H}_0}(k_0(r) \geq k_n) \\
 &\leq \sum_{n=0}^{N_{max}} m_n \cdot \frac{\varepsilon}{N_{\mathcal{R}}} \\
 &= N_{\mathcal{R}} \cdot \frac{\varepsilon}{N_{\mathcal{R}}} = \varepsilon,
 \end{aligned} \tag{4.19}$$

which implies that on average at most one false detection could be detected by chance in the entire image.

Chapitre 5

LSDSAR, a generic line segment detector for SAR images

The aim of this work is to develop an LSD-like line segment detector for SAR images. In order to do this, two aspects of the original algorithm have to be modified. First, LSD relies on the grouping of pixels sharing the same local orientation, therefore, the gradient computation step has to be replaced with a gradient computation method suitable for SAR images. Second, the meaningfulness of a candidate rectangle is verified in an *a contrario* model which is based on two hypothesis : 1) the local orientations of pixels are independent random variables ; 2) each random variable follows a uniform distribution over $[0, 2\pi]$. In what follows, we will see that these hypotheses are not satisfied for SAR images, so that the *a contrario* model has to be strongly modified.

5.1 Gradient computation in SAR images

Due to the multiplicative nature of speckle noise in SAR images, the usual gradient computation methods which are based on pixel value differences tend to produce more false edges in homogeneous areas with high mean intensity values than those with low mean intensity values Touzi et al. [1988]. Contrarily to those gradient computation methods, it has been proven that the ratio based method is more suitable for SAR images because it yields a constant false alarm rate. More precisely, the probability of false alarms do not depend on the underlying mean intensity values of the homogeneous areas. In Touzi et al. [1988], the Ratio of Average (ROA) operator is introduced to compute the gradient (notice that in Touzi et al. [1988] and Fjørtoft et al. [1998], the authors only named the operators as *ratio detector* but we assume they are both ratio-based gradient detector, so we will mention them as gradient computation methods in this thesis). For a given pixel located at position (x, y) in the image I , the ratio of averages $R(x, y)$ is computed along P ($P = 4, 8, 16, \dots$) directions. For each direction, the ratio of average is defined as the ratio of the arithmetic mean of pixel values in the opposite side windows. In the i -th direction, where $1 \leq i \leq P$, the ratio is computed as

$$R^i(x, y) = \frac{M_1^i(x, y)}{M_2^i(x, y)}, \quad (5.1)$$

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 used to

compute

$$T^i(x, y) = \max \left(R^i(x, y), \frac{1}{R^i(x, y)} \right). \quad (5.2)$$

The magnitude $G_{roa}(x, y)$ of the gradient at position (x, y) is then defined as

$$G_{roa}(x, y) = \max_{1 \leq i \leq P} T^i(x, y), \quad (5.3)$$

and its direction is defined as the direction having the highest value of ratio. Once the gradient is computed, a threshold corresponding to a given probability of false alarm rate can be used for the purpose of edge extraction.

ROA works well on isolated step edges, when there is no other edge in the window used for the computation. However, due to the strong multiplicative noise in real SAR images, large averaging windows are usually needed to reduce the influence of noise, in which case some edges could exist in the averaging windows. The problem for the existence of multiple edges was addressed in Fjørtoft et al. [1998] under the hypothesis of a stochastic multiedge model, where the exponentially weighted average was proven to be optimal for estimating the local mean intensity values in the sense of minimum mean square error (MMSE) when the averaging windows cover edges with random intervals. The horizontal and vertical components of the gradient computed by ROEWA Fjørtoft et al. [1998] were defined as the ratios of exponentially weighted average in the opposite side windows along the horizontal and vertical directions. 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

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

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

and where W is the upper integer part of $\log(10) \times \alpha$. 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

$$G_{roewa}(x, y) = \sqrt{T^h(x, y)^2 + T^v(x, y)^2}. \quad (5.4)$$

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 what we refer to as the *multi-direction ratio-based methods*, which are however quite time consuming.

In Dellinger et al. [2015], 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 $G_{gr}(x, y)$ and orientation $ang_{gr}(x, y)$ of GR at position (x, y) are defined by

$$\begin{aligned} G_{gr}(x, y) &= \sqrt{G^h(x, y)^2 + G^v(x, y)^2}, \\ ang_{gr}(x, y) &= \text{atan2}(G^h(x, y), G^v(x, y)). \end{aligned}$$

5.2 The choice of the gradient computation methods

In order to choose the most suitable gradient computation method that can be used in LSD, we give a thorough analysis and comparison of different gradient computation methods in SAR images. Furthermore, we describe the criterion and reason for the choice of the method.

5.2.1 The property of constant false alarm rate (CFAR)

An important advantage of the ratio based edge detectors is that they have a constant false alarm rate for SAR images. Specifically, for an image of homogeneous areas with a certain number of looks, the gradient magnitude fields computed by the ratio based methods follow exactly the same distribution (for a given size of averaging windows), regardless of the mean intensity values of the homogeneous areas. In addition, the probability density functions (pdf) of two homogeneous areas across boundaries depend only on the contrast of the edge, and do not depend on the mean intensity values on the two sides. The complete theoretical analysis of the CFAR property is done in Touzi et al. [1988] for the intensity format of SAR images. We analyze here the CFAR property for both amplitude and intensity SAR images and illustrate that the ratio based gradient computation methods also have CFAR in amplitude SAR images. We study the influence of the mean intensity values on the pdf of the gradient magnitude fields computed in both pure noise images and synthetic edge images, while ensuring that all the images have the same number of looks. In addition, the size of the windows as well as the weight function should be the same for all experiments.

5.2.1.1 Check the CFAR property in pure noise images

In order to compare the distribution of the gradient magnitude fields computed by the ratio based methods in homogeneous areas, we simulate eight 1-look pure noise images of size 4096×4096 pixels. The mean intensity values of these homogeneous areas are 60^2 , 90^2 , 120^2 , 150^2 , 180^2 , 210^2 , 240^2 and 270^2 , respectively. The gradient computation methods we use here are ROEWA and GR, both with the weighting parameter $\alpha = 4$. The gradient computation is done in both amplitude and intensity format of these pure noise images. The histograms (normalized pdf) of the gradient magnitude fields of homogeneous areas computed by ROEWA and GR can be found in figure 5.1. From figure 5.1 we can see that the distribution of the gradient magnitude fields computed by both ROEWA and GR does

not depend on the mean intensity values of the homogeneous areas in both amplitude and intensity format of SAR images : their histograms overlap very well. It implies that for a given probability of false alarms, the choice of threshold does not depend on the mean intensity values of the homogeneous areas.

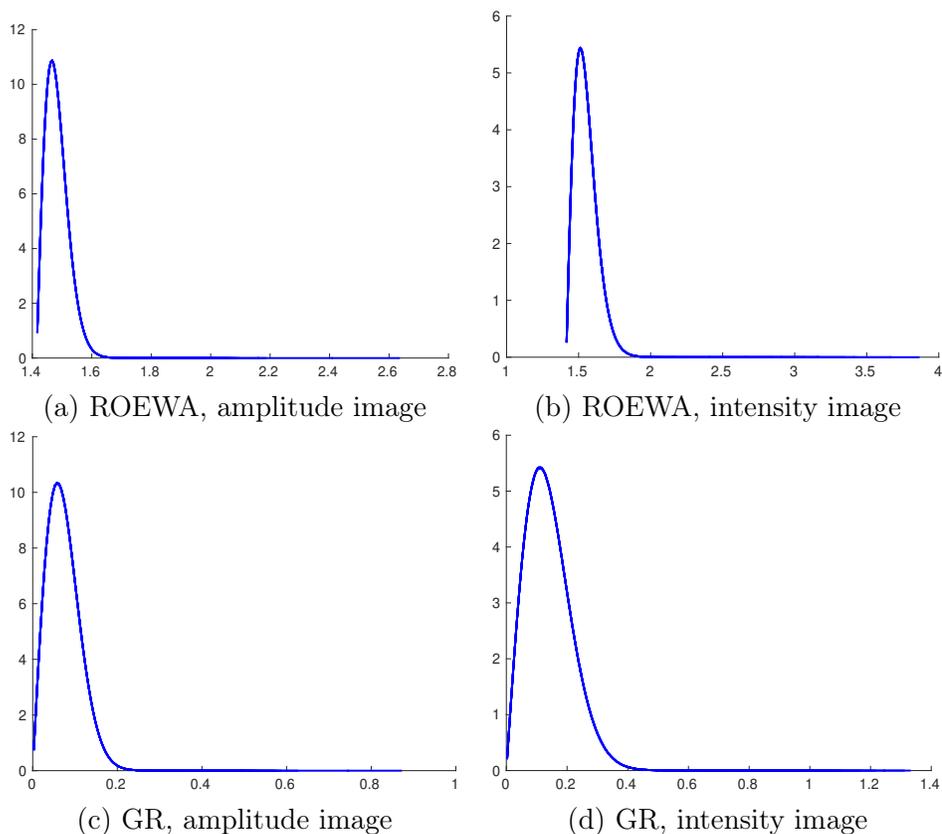


FIGURE 5.1 – The histograms (normalized pdf) of gradient magnitude fields computed by ROEWA and GR in eight 1-look pure noise images of size 4096×4096 pixels. The mean intensity of the homogeneous areas are 60^2 , 90^2 , 120^2 , 150^2 , 180^2 , 210^2 , 240^2 and 270^2 , respectively. The gradient computation is done in both amplitude and intensity SAR images. The weighting parameter α in both ROEWA and GR is 4. The size of the bin is 0.005.

5.2.1.2 Check the CFAR property in images of two homogeneous areas across boundaries

In this part, we compare the distribution of gradient magnitude fields computed by ROEWA and GR in two homogeneous areas across boundaries. We simulate two kinds of synthetic edge images : one with amplitude ratio contrast 1.5 and one with amplitude ratio contrast 3.0. For each kind of synthetic edge image, there are 8 realizations and the homogeneous areas for one side of the edge are with mean intensity values : 20^2 , 50^2 , 70^2 , 90^2 , 120^2 , 150^2 , 180^2 , 200^2 , respectively. The mean intensity values for the homogeneous areas in the other side of the edges can be computed as follows : for example, if the mean intensity value for the homogeneous area in one side of the edges is 90^2 , and the amplitude ratio contrast of the edge is 3.0, then the mean intensity value for the homogeneous area in the other side of the edges is $(90 \times 3.0)^2$. The size of the edge images are all 1024×1024 pixels. One synthetic edge image with its associated edge ground truth can be found in figure 5.2.

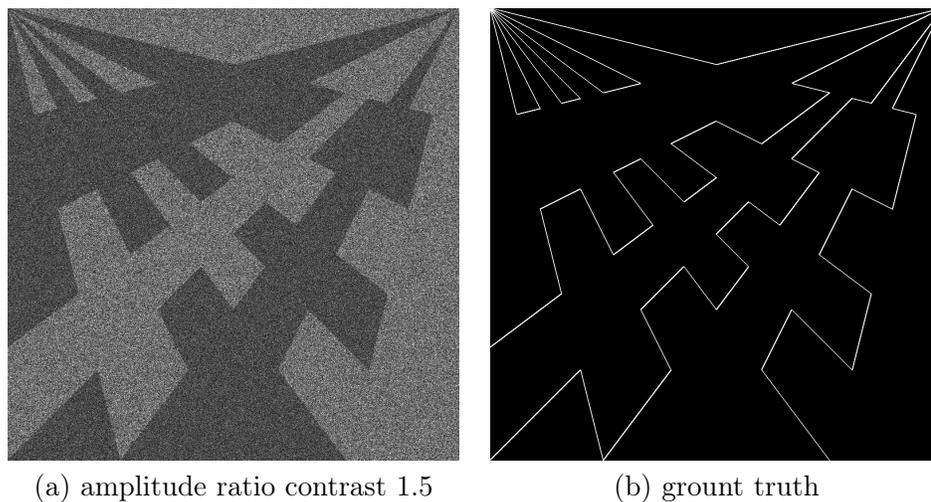


FIGURE 5.2 – A 1-look synthetic edge image with amplitude ratio contrast 1.5 and its associated edge ground truth. The size of images are 1024×1024 pixels.

The histograms (normalized pdf) of the gradient magnitude fields computed by ROEWA and GR can be found in figure 5.3 and figure 5.4. From figure 5.3 and figure 5.4 we can see that the mean intensity values of the homogeneous areas on both sides of the edges do not influence the distribution of the gradient magnitude fields, which implies that the gradient distribution for two homogeneous areas across boundaries depends only on the ratio of the mean intensity values.

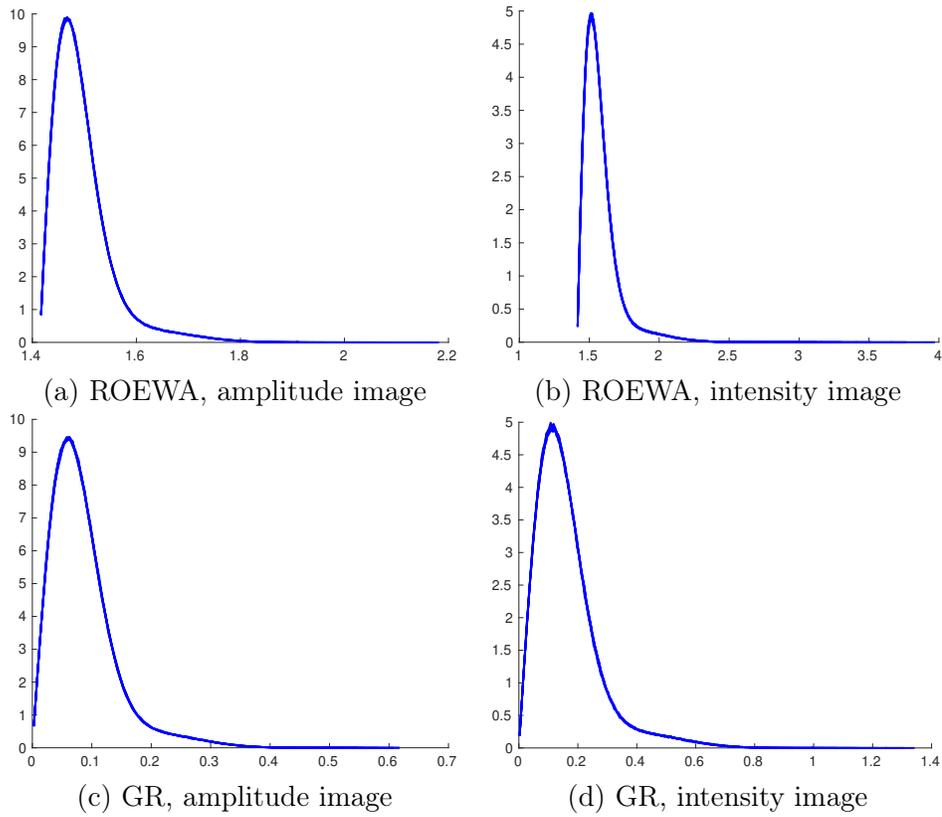


FIGURE 5.3 – The histograms (normalized pdf) of the gradient magnitude fields computed by ROEWA and GR in eight 1-look synthetic edge images with amplitude ratio contrast 1.5. The size of the images are 1024×1024 pixels. The value of the weighting parameter α is 4. The size of the bin is 0.005.

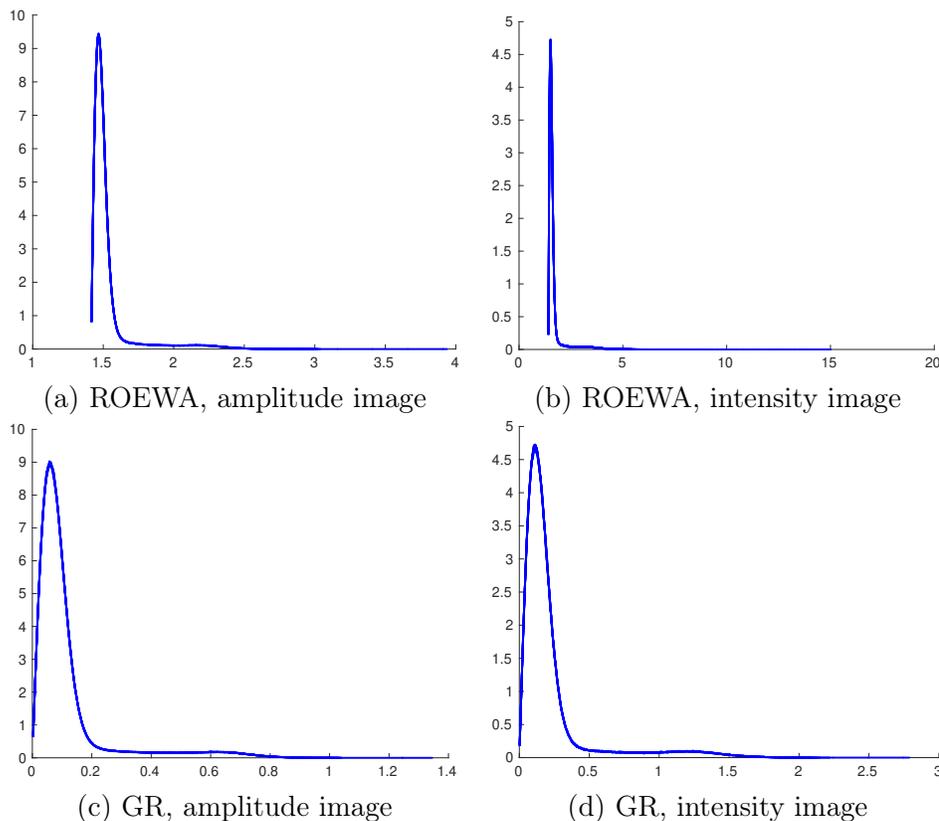


FIGURE 5.4 – The histograms (normalized pdf) of the gradient magnitude fields computed by ROEWA and GR in eight 1-look synthetic edge images with amplitude ratio contrast 3. The size of the images are 1024×1024 pixels. The value of the weighting parameter α is 4. The size of the bin is 0.005.

5.2.1.3 Conclusion on the CFAR property

With the observation in figure 5.1, figure 5.3 and figure 5.4, we assume that the mean intensity values do not influence the pdf of the gradient magnitude fields computed by both ROEWA and GR. In addition, ROEWA and GR has a constant false alarm rate in both amplitude and intensity SAR images.

5.2.2 Detection capability of the gradient magnitude : a comparison of existing gradient computation methods

In order to choose the best gradient computation method for SAR images, we compare the existing approaches in terms of two criterions : 1) we compare the ability of existing approaches to highlight true edge pixels using ROC curves Dougherty et al. [1998], Bowyer et al. [1999]; 2) we compare the accuracy of gradient orientation computed by existing approaches and check whether those methods satisfy the hypothesis in the *a contrario* model that each local orientation follows a uniform distribution over $[0, 2\pi]$. In this section, we first analyze the ability of existing approaches to distinguish between edge pixels and noise pixels.

5.2.2.1 The influence of the format of SAR images : amplitude or intensity

It has been studied in section 5.2.1 that ROEWA and GR have a constant false alarm rate for both amplitude and intensity format of SAR images. What should also be studied is the influence of the format of SAR images on the ability of the gradient computation method to highlight the edge pixels. Notice that in Fjørtoft et al. [1998], the exponential weight function is proved to be optimal when applied to the intensity SAR images. We will show in the following that the format of SAR images has minor influences on the performances of ROEWA and GR. We simulate eight 1-look synthetic edge images and eight 3-look synthetic edge images with size 1024×1024 pixels. The amplitude ratio contrast of the edge varies from 1.2 to 1.9 with step 0.1. The edge image with amplitude ratio contrast 1.5 and the associated ground truth (GT) can be found in figure 5.2. The gradient computation method we are using is GR with weighting parameter $\alpha = 4$.

ROC curves of the gradient magnitude fields computed by GR in eight 1-look synthetic edge images and eight 3-look synthetic edge images are shown in figure 5.5 and figure 5.6. From these figures we can see that the format of SAR images has little influence on the performances of GR. In addition, considering that GR has CFAR for both amplitude and intensity SAR images, in all the following experiments, we provide experiments using only amplitude SAR images.

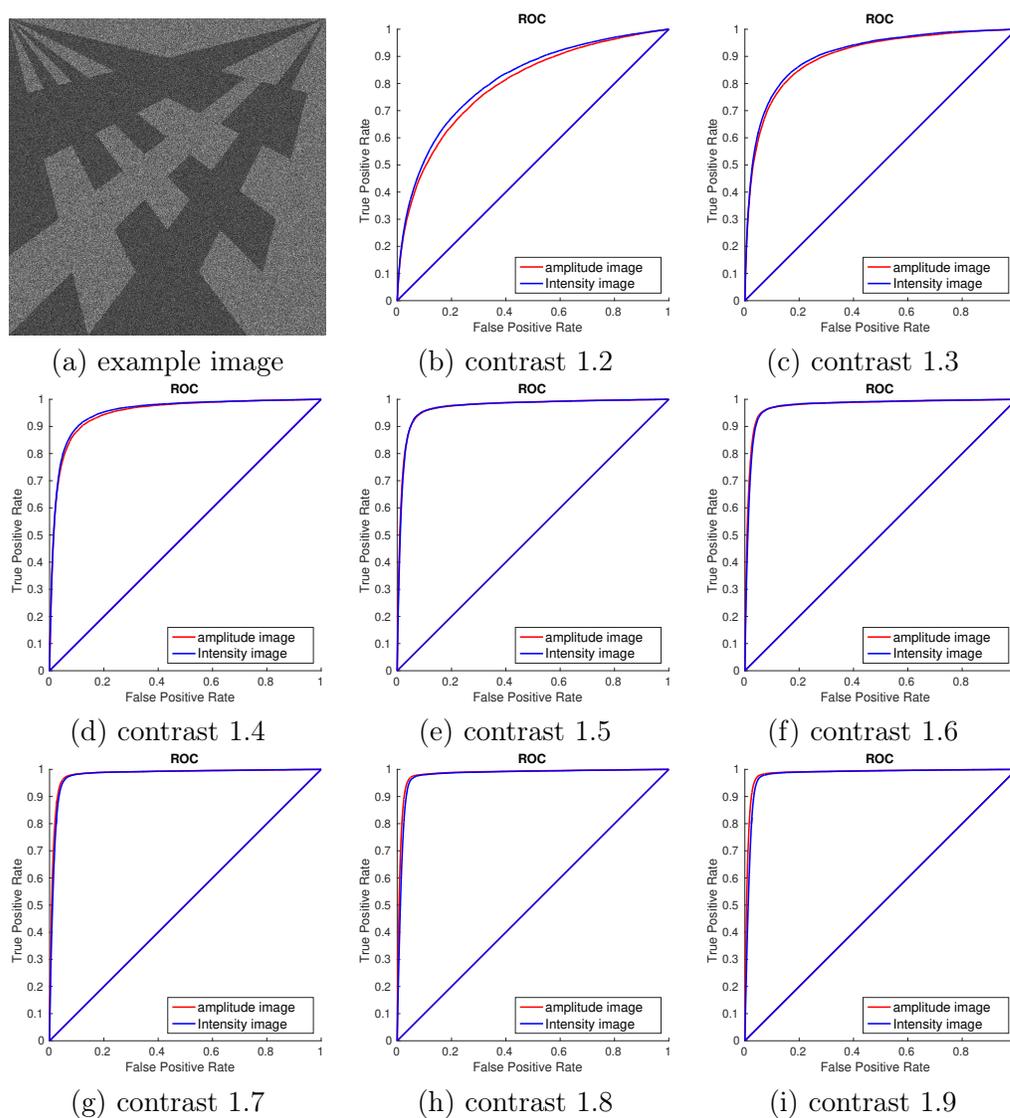


FIGURE 5.5 – An example edge image with contrast 1.5 and ROC curves of gradient magnitude fields computed by GR in eight 1-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

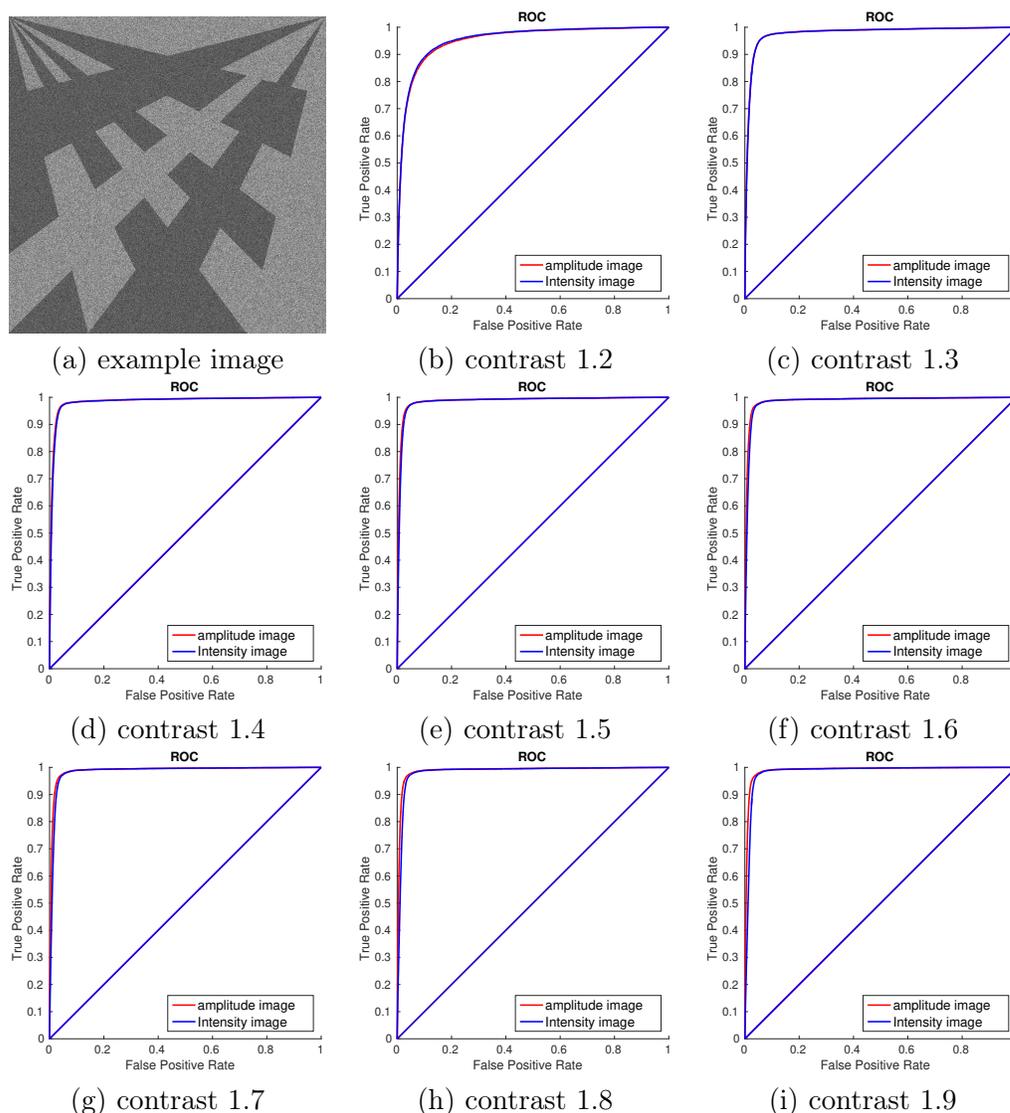


FIGURE 5.6 – An example image with contrast 1.5 and ROC curves of gradient magnitude fields computed by GR in eight 3-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

5.2.2.2 Choice of the weighting function for 2-direction methods

What we refer to as the 2-direction methods are those who compute the horizontal and vertical components of the gradient, as done by ROEWA and GR. We add one 2-direction method which is named as ROA-2, though it has been studied in Fjørtoft et al. [1998] that it gives worse performances than ROEWA under the hypothesis of a multi-edge model. For ROA-2, we use the same size of windows as ROEWA and GR to compute the gradient (the size of averaging windows in ROEWA and GR is decided by the weighting parameter α). The difference between ROA-2 and ROEWA is that ROA-2 is defined as the ratio of the arithmetic mean. For ROEWA and GR, the weighting parameter α is set to 4. We simulate eight 1-look and eight 3-look synthetic edge images with the amplitude ratio contrast of the edge ranging from 1.2 to 1.9 with step 0.1. The size of the synthetic edge images are

all 1024×1024 pixels. A synthetic edge image with ratio contrast 1.5 and its associated ground truth can be found in figure 5.2.

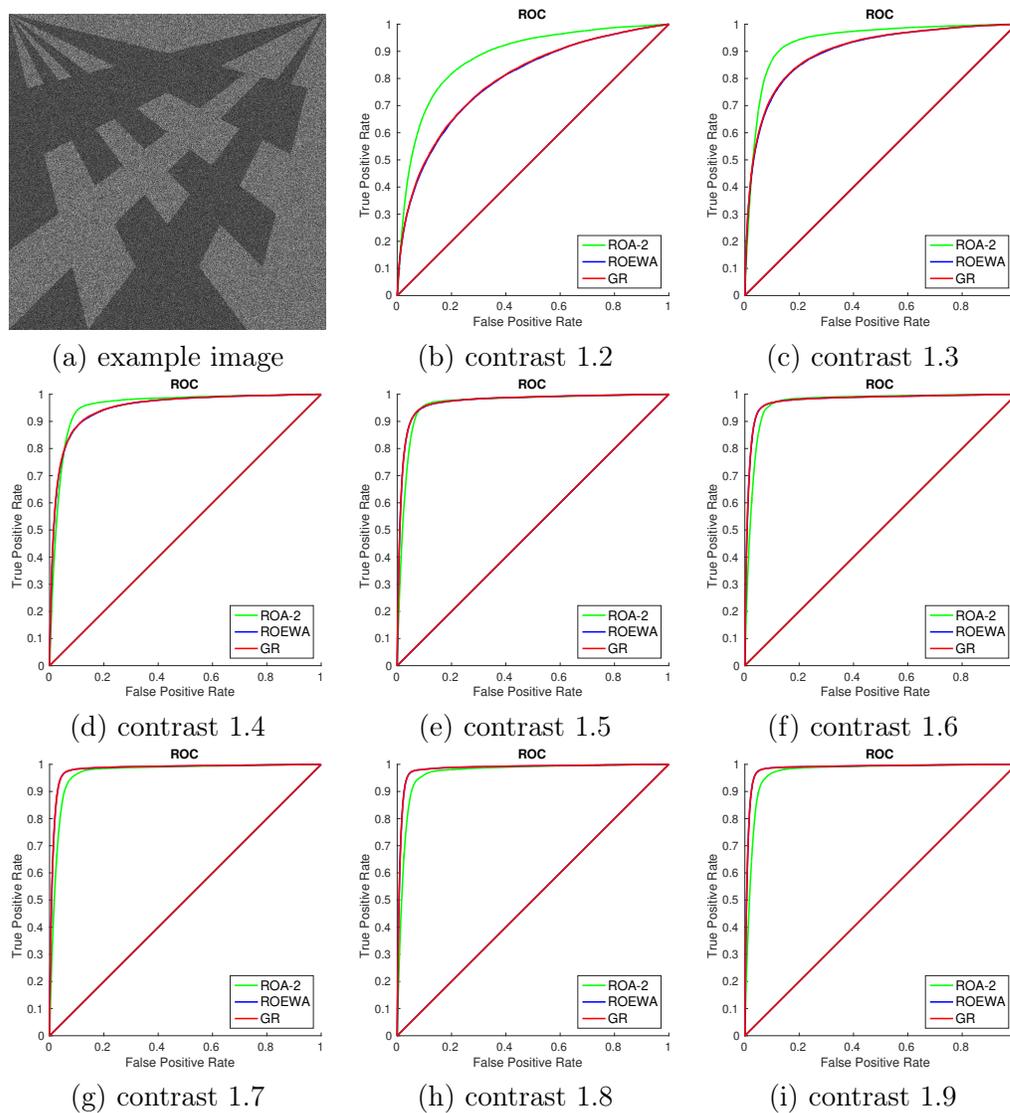


FIGURE 5.7 – An example of 1-look edge image and ROC curves of gradient magnitude fields computed by ROA-2, ROEWA and GR in eight 1-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

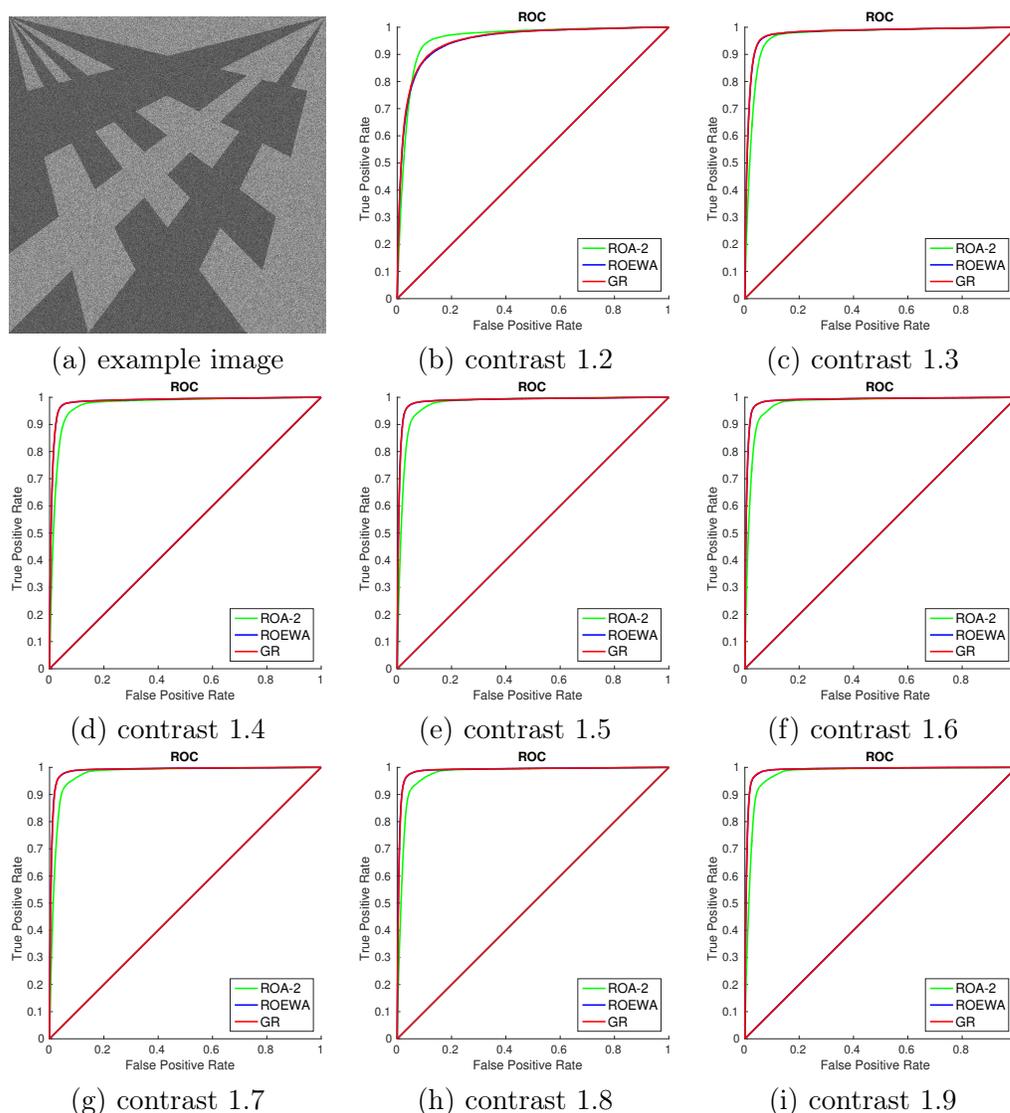


FIGURE 5.8 – An example of 3-look edge image and ROC curves of gradient magnitude fields computed by ROA-2, ROEWA and GR in eight 3-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

The ROC curves computed by ROA-2, ROEWA and GR in eight 1-look synthetic edge images and eight 3-look synthetic edge images can be found in figure 5.7 and figure 5.8. From these figures we can see that in 1-look and low contrast situation, ROA-2 are more powerful in emphasizing edge pixels because its ROC curves are above those of ROEWA and GR, while in less challenging cases (like 1-look high contrast situation, multi-look situation), ROEWA and GR show slightly superior performances than ROA-2. What can be assumed from figure 5.7 and figure 5.8 is that for isolated step edges, the choice of the weighting function has little influence on the results.

However, it is quite common in practical case that there exists several edges in the averaging windows, so that the hypothesis of isolated step edges does not hold in the general case. We simulate eight 1-look edge images of such kind with amplitude ratio

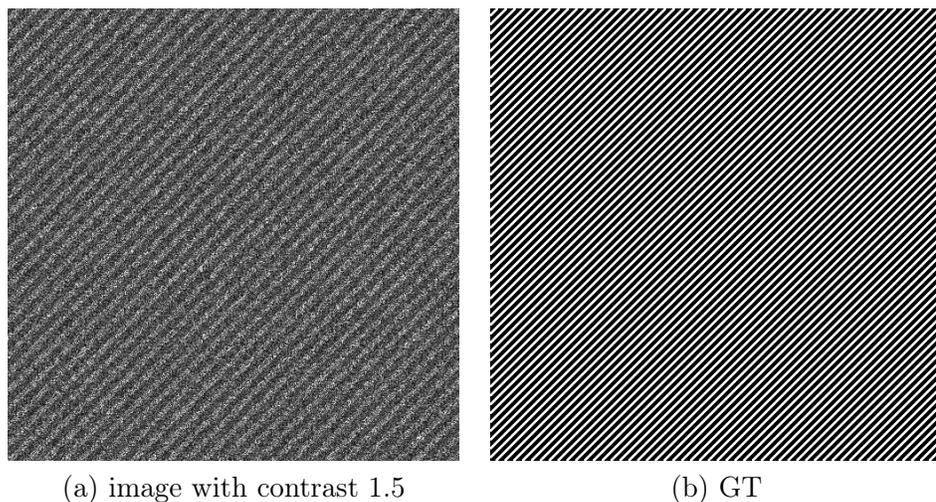


FIGURE 5.9 – images with very close edges to demonstrate the efficiency of exponential weighting function compared to the arithmetic mean.

contrast varying from 1.2 to 1.9 with step 0.1 (an example with amplitude ratio contrast 1.5 can be found in figure 5.9). The ROC curves of the gradient magnitude field computed by ROA-2, ROEWA, and GR can be found in figure 5.10. From these ROC curves we can see that under the hypothesis of a multiedge model, the performances of ROEWA and GR are much better than those of ROA-2, which follows the analysis in Fjørtoft et al. [1998]. In addition, it must be noted that in all cases the performances of ROEWA and GR are comparable.

5.2.2.3 The comparison of 2-direction and multi-direction methods

Following the analysis in section 5.2.2.2, we decide to use the exponential weight function in the following. There are two usual ways to compute the gradient in SAR images : 1) compute the ratio along multiple directions, the maximal normalized ratio being then defined as the gradient magnitude, as discussed in Touzi et al. [1988]; 2) compute the ratio along the horizontal and vertical directions to obtain the horizontal and vertical gradient components, as those in Fjørtoft et al. [1998] and Dellinger et al. [2015]. In this section, we compare the detection capability of the gradient magnitude obtained by the two methods. The 2-direction method we choose is GR, and for multi-direction method, we use 8 directions, 16 directions and 32 directions. For all of them, we use the exponential weight function with weighting parameter $\alpha = 4$. For the multi-direction method, we use the circle-shaped windows to reduce the influence of rotation and ensure that the number of pixels in the window is approximately the same as that of GR. The comparison is done on eight 1-look synthetic edge images of size 1024×1024 pixels with amplitude ratio contrast from 1.2 to 1.9. One example of the synthetic edge images can be found in figure 5.2. ROC curves computed by GR and the multi-direction method can be found in figure 5.11. It can be seen from figure 5.11 that the multi-direction methods gives slightly better performances than GR because their ROC curves remain on the top-left side of those of GR. In addition, the number of directions seems to have minor influences on the performances for the multi-direction methods. Even though the multi-direction methods show higher detection capability in terms of the gradient magnitude, we should also check the accuracy

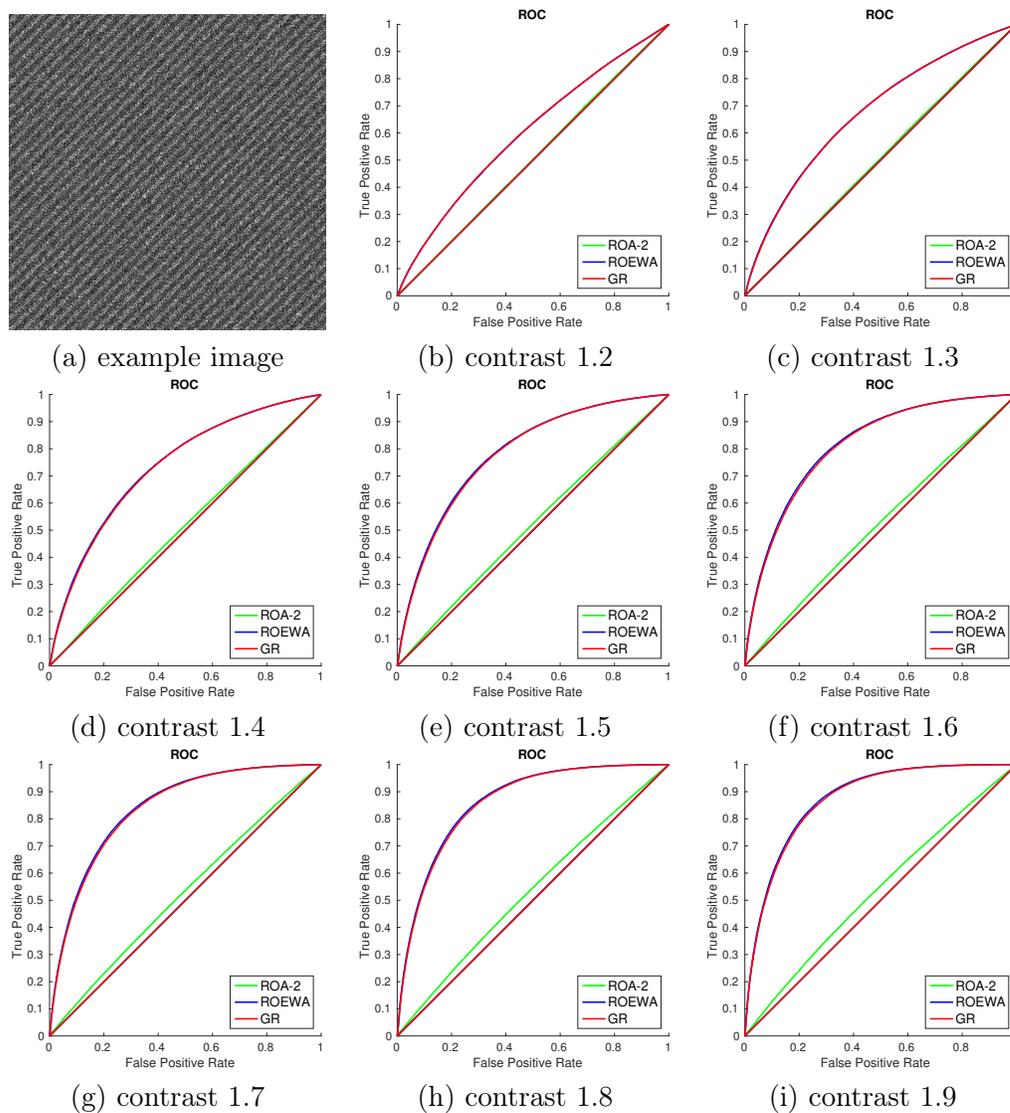


FIGURE 5.10 – An example image and ROC curves of gradient magnitude fields computed by ROA-2, ROEWA and GR in eight 1-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 512×512 pixels.

of the gradient orientation for both of these two kinds of method. What's more important, we need to check whether the local orientation computed by them satisfies the hypothesis in the *a contrario* model. In the next section, we are going to check these points.

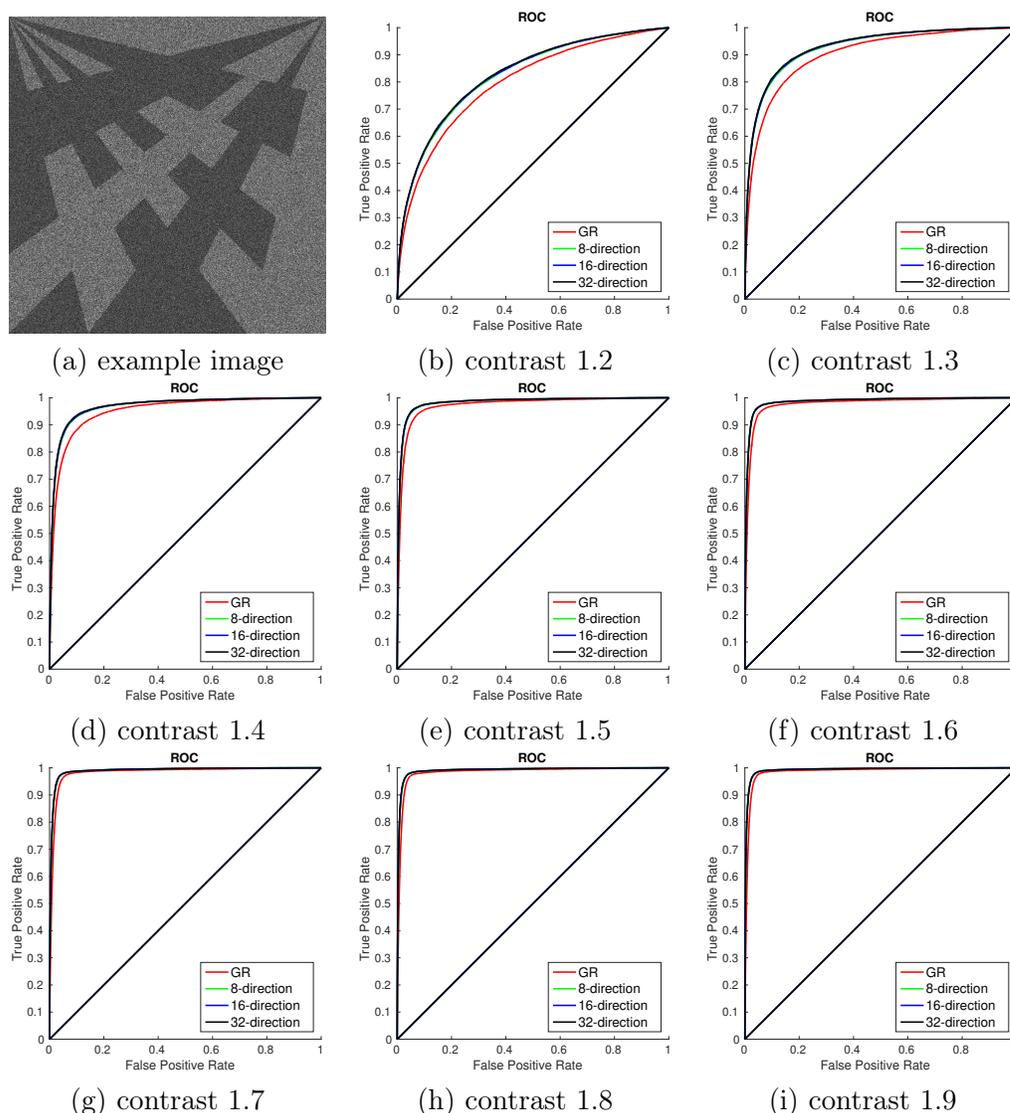


FIGURE 5.11 – An example 1-look image and ROC curves of gradient magnitude fields computed by GR, 8-direction, 16-direction and 32-direction method in eight 1-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

5.2.3 Accuracy and distribution of the local orientation

In order to choose the most suitable gradient computation method to be used in the region growing step of LSD, it is also of vital importance to check the accuracy of the computed local orientations. More importantly, we need to check whether the local orientation of each pixel follows a uniform distribution over $[0, 2\pi]$. We address these two points in the following.

5.2.3.1 Comparison in the accuracy of local orientation



FIGURE 5.12 – The image with horizontal edges used to compare the accuracy of the local orientation computed by GR, 8-direction method, 16-direction method and 32-direction method. The size of the image are 50×1024 pixels.

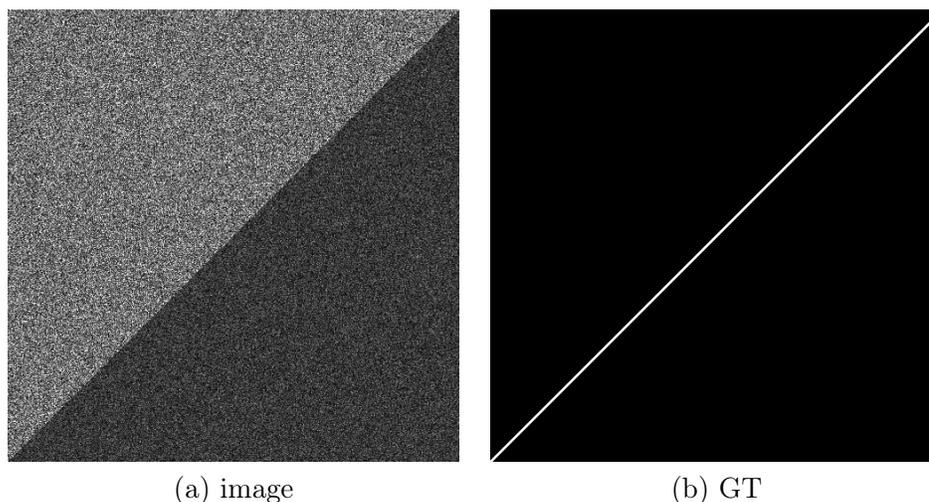


FIGURE 5.13 – The image with diagonal edges used to compare the accuracy of the local orientation computed by GR, 8-direction method, 16-direction method and 32-direction method. The size of the image are 512×512 pixels.

It has been studied in section 5.2.2.3 that the detection capability of the gradient magnitude computed by the multi-direction methods is slightly better than that of GR. In this part, we compare the accuracy of the local orientation. The comparison is done in two kinds of 1-look images : one contains a horizontal edge as shown in figure 5.12, the size of the image are 50×1024 pixels; one contains a diagonal edge as shown in figure 5.13, the size of the image are 512×512 pixels. For both kinds of images, eight contrast values varying from 1.2 to 1.9 are evaluated. For each contrast value, there are 10 realizations for each kind of image. The angle of a pixel along the edge is considered to be accurate if this angle is equal to the real one up to an angle tolerance $\tau = \frac{\pi}{8}$. The probability to obtain the accurate orientation of a pixel for each method can be estimated as the ratio between the number of pixels having the correct angles and the total number of pixels along the edges. The probability of accuracy in the local orientation computed by GR, 8-direction, 16-direction and 32-direction method can be found in figure 5.14-(a) and figure 5.14-(b). The observations from figure 5.14 can be summarized as follows :

- with the increase of contrast, the accuracy probabilities of the local orientation computed by all methods are increasing ;
- the local orientations computed along the horizontal edges are more accurate than those along the diagonal ones ;
- with the increase in the number of directions for multi-direction methods, the accuracy of the local orientations is improved ;

- the accuracy of local orientations computed by GR is better than 8-direction and 16-direction methods, and slightly worse than the 32-direction method.

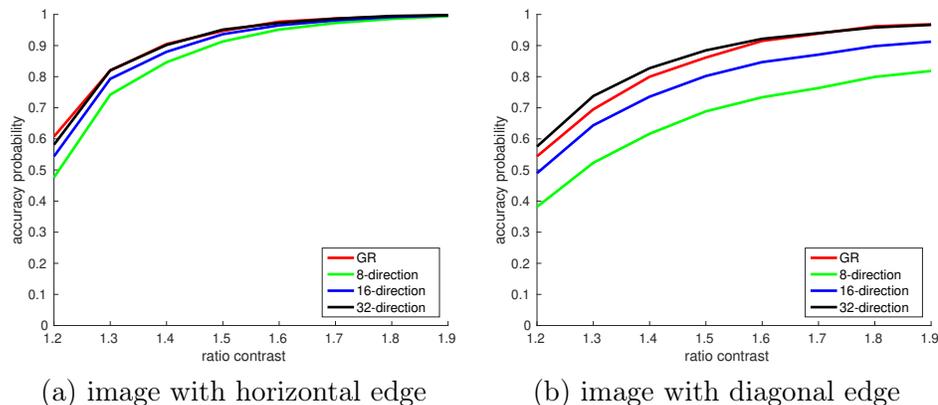


FIGURE 5.14 – The accuracy probability curves computed by GR, 8-direction, 16-direction and 32-direction methods computed in 1-look images with ratio contrast varying from 1.2 to 1.9 with step 0.1. For images with each contrast of edge, there are 10 random realizations.

5.2.3.2 Check the distribution of local orientations in noise

We have compared the accuracy of local orientations computed by GR and the multi-direction methods in section 5.2.3.1, but one important point we have to study is whether the local orientation computed by them follows a uniform distribution over $[0, 2\pi]$. Indeed, this is one of the two crucial hypothesis in the *a contrario* model. We analyze the distribution of the local orientation in 1-look images of constant areas. Since all pixels in the image are drawn from a Rayleigh distribution with the same underlying mean intensity value, the local orientations of all pixels can be considered as random realizations of the same random variable. We compute the local orientation using GR, 8-direction, 16-direction and 32-direction methods in a 1-look pure noise image of size 8192×8192 pixels. The weighting parameter α is set to 4. The histograms of local orientations computed by these methods can be found in figure 5.15. From this figure we can see that, among the compared methods, only GR roughly yields a uniform distribution over $[0, 2\pi]$. In the case of multi-direction methods, only a limited number of angles is accessible, and with the increase number of directions used to compute the gradient, the distribution deviates further from a uniform distribution. Combining the distribution property with the detection capability of gradient magnitude described in section 5.2.2.3 and the accuracy of the local orientation described in section 5.2.3.1, we decide to replace the gradient computation step in LSD with the GR method.

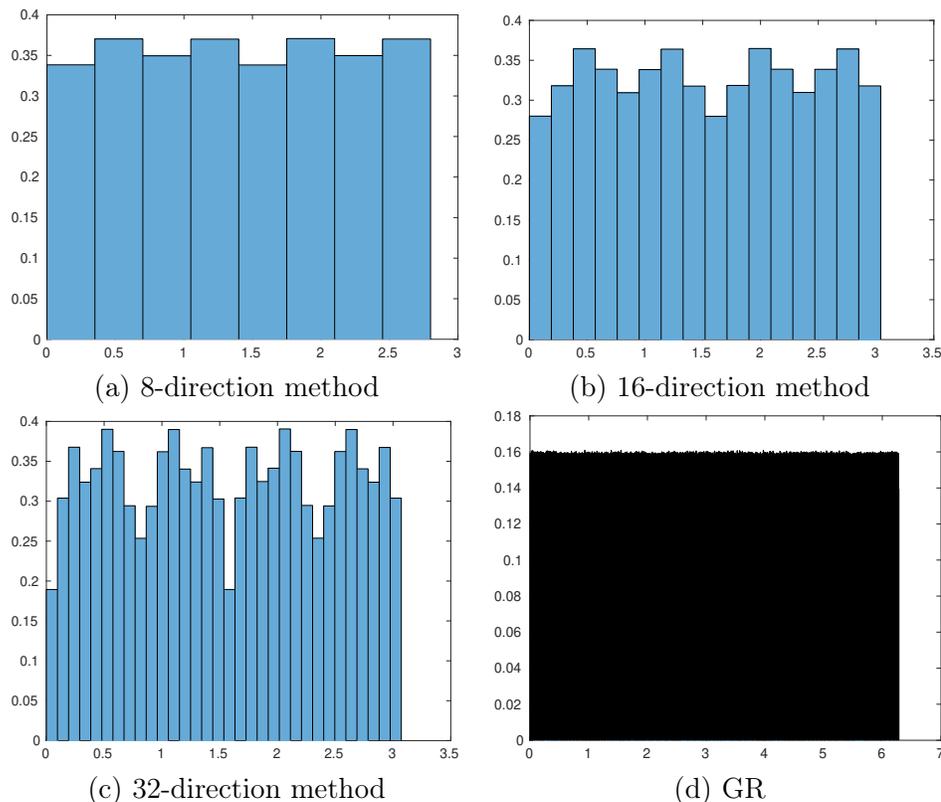


FIGURE 5.15 – The histograms of local orientations computed by GR, 8-direction, 16-direction and 32-direction methods in a 1-look pure noise image of size 8192×8192 pixels. The exponential weighting parameter α is set to 4.

5.3 A Markovian *a contrario* model

Following to the analysis in section 5.2, we have decided to use GR to compute the gradient. It has been checked that the local orientation computed by GR satisfies a uniform distribution over $[0, 2\pi]$. The next question is whether the local orientations are independent random variables. In the paper Desolneux et al. [2000], in order to ensure the independence of local orientations, only pixels at a distance larger than 2 are taken into account along a line when the size of windows used to compute the gradient are 2×2 pixels. In the case of gradient computation for SAR images, much larger averaging windows (with size 21×21 pixels for example) are required to reduce the influence of strong multiplicative noise and to detect edges between homogeneous areas with close mean intensity values Touzi et al. [1988].

Consequently, the usual *a contrario* framework badly fails to control the number of false detections due to the dependencies introduced between local orientations of neighbouring pixels when computing the local orientation with a method suited to SAR, as we will show in our experiments. In order to account for the strong dependencies between nearby pixels introduced by filtering, we draw from the ideas in Myaskouvskey et al. [2013], 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 Myaskovskey et al. [2013], 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 $t = 1, \dots, n(r)$, 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 $X_1, X_2, \dots, X_{n(r)}$ are mutually independent, we assume that they follow a Markov chain of order one, i.e., we assume that, for all $1 < t \leq n(r)$,

$$\mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}, \dots, X_1 = x_1) = \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}). \quad (5.5)$$

Therefore, the distribution of aligned pixels is characterized by the four transition 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 as $\mathbb{P}(1|1)$, $\mathbb{P}(1|0)$, $\mathbb{P}(0|1)$, 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.

The modified background model, that we denote by \mathcal{H}'_0 is defined as follows.

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 (4.12) of LSD is naturally changed into

$$\text{NFA}'(r) = N_{\mathcal{R}} \cdot \mathbb{P}_{\mathcal{H}'_0}(k_0(r) \geq k(r)), \quad (5.6)$$

which also satisfies the NFA-property (4.13) Grosjean and Moisan [2009].

The next question would be how to compute (5.6). Because of the Markov chain assumption \mathcal{H}'_0 -b, the probability of having $k_0(r) \geq k(r)$ involved in (5.6) is given by

$$\mathbb{P}_{\mathcal{H}'_0}(k_0(r) \geq k(r)) = \sum_{x_1 + \dots + x_{n(r)} \geq k(r)} \mathbb{P}(X_1 = x_1) \cdot \prod_{t=2}^{n(r)} \mathbb{P}(X_t = x_t | X_{t-1} = x_{t-1}). \quad (5.7)$$

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 (5.7). In order to overcome this limitation, we proceed as in Myaskovskey et al. [2013] 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

$$\begin{aligned} \mathbb{P}(Y_t \geq k) &= \mathbb{P}(Y_{t+1} \geq k | X_t = 0) \cdot \mathbb{P}(X_t = 0) \\ &\quad + \mathbb{P}(Y_{t+1} \geq k - 1 | X_t = 1) \cdot \mathbb{P}(X_t = 1). \end{aligned} \quad (5.8)$$

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

$$\begin{aligned}
\mathbb{P}(Y_{t+1} \geq k' | X_t = x) &= \sum_{y \in \{0,1\}} \mathbb{P}(Y_{t+2} \geq k' - y, X_{t+1} = y | X_t = x) \\
&= \mathbb{P}(Y_{t+2} \geq k' | X_{t+1} = 0) \cdot \mathbb{P}(X_{t+1} = 0 | X_t = x) \\
&\quad + \mathbb{P}(Y_{t+2} \geq k' - 1 | X_{t+1} = 1) \cdot \mathbb{P}(X_{t+1} = 1 | X_t = x), \quad (5.9)
\end{aligned}$$

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

$$\mathbb{P}(Y_n \geq k' | X_{n-1} = x) = \begin{cases} 1 & \text{if } k' = 0, \\ \mathbb{P}(1|x) & \text{if } k' = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (5.10)$$

It follows that (5.8) can be computed in polynomial time Myaskouvskey et al. [2013], and so does the probability $\mathbb{P}_{\mathcal{H}'_0}(k_0(r) \geq k(r))$ needed in (5.6).

In practice, we can precompute the values of $\mathbb{P}_{\mathcal{H}'_0}(k_0(r) \geq k(r))$ for all $(n_0(r), k(r)) \in \{1, \dots, N_{\max}\}^2$, 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 (5.8), (5.9) and (5.10), 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.

The next question is how to estimate the transition probabilities involved in (5.9) and (5.10). Since the dependencies are introduced by the strong filtering during the gradient computation, the dependencies between local orientations should be estimated in a pure random image (a structureless image). In addition, the dependencies introduced between local orientations are determined by the size of averaging windows used to compute the gradient, while the size of averaging windows is decided by the value of the weighting parameter α in GR. Therefore, the transition probabilities should be estimated according to a value of α . The larger the value of α is, the more the dependencies are introduced.

The influence of α values on the amount of dependencies introduced between neighbouring pixels

The fact that computing GR with large α values will introduce dependencies between local orientations of neighbouring pixels can be easily verified by computing the gradient in a pure speckle noise image, as can be seen from figure 5.16. This figure shows the local orientation field computed by GR with $\alpha = 4$ in a 1-look pure noise image of size 30×30 pixels. What can be observed from figure 5.16 is that strong dependencies are introduced between local orientations of neighbouring pixels, so that neighbouring pixels tend to have the same local orientation.

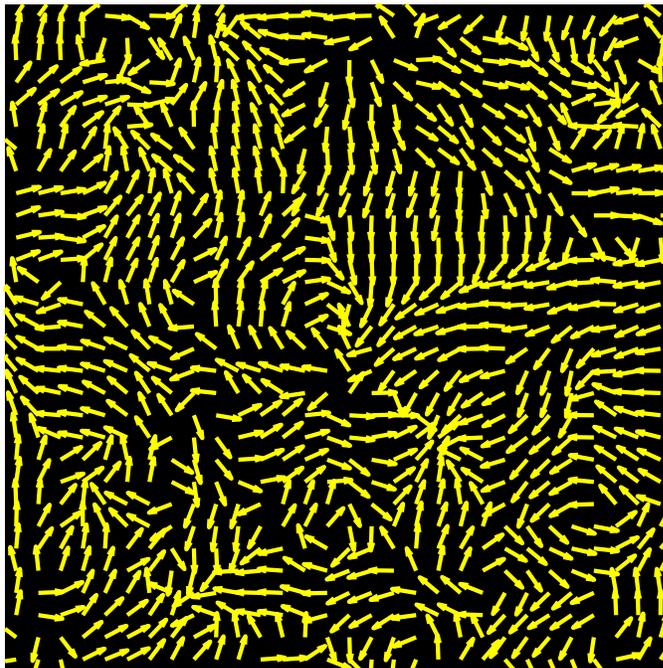


FIGURE 5.16 – The local orientation field computed by GR with $\alpha = 4$ in a 1-look pure noise image of size 30×30 pixels.

TABLE 5.1 – Comparison of transition probabilities estimated along horizontal and diagonal lines in the local orientation fields computed by GR with $\alpha = 4$ in eight 1-look pure noise images of size 4096×4096 pixels.

	horizontal	diagonal
$\mathbb{P}(1 1)$	0.586310 ± 0.000060	0.395011 ± 0.000084
$\mathbb{P}(1 0)$	0.059153 ± 0.000011	0.086333 ± 0.000018

Since GR computes the gradient in the discrete spatial coordinates of images, the averaging windows of neighbouring pixels overlap the most along the horizontal and vertical directions. Therefore, it can be deduced that GR introduces the most dependencies along the horizontal and vertical directions. In order to demonstrate this, we compare the transition probabilities estimated from horizontal and diagonal lines. 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 $\mathbb{P}(0|0) = 1 - \mathbb{P}(1|0)$. The transition probabilities estimated from the local orientation fields computed by GR with $\alpha = 4$ in eight 1-look Nakagami noise images of size 4096×4096 pixels can be found in table 5.1, where the angle tolerance was set to 22.5 degrees, as in the original LSD algorithm. From table 5.1 we can see that much more dependencies are introduced between pixel pairs along horizontal lines compared to those pixel pairs along the diagonal lines. Consequently, in the definition of our Markovian

a contrario model, the transition probabilities will be estimated from the local orientations of pixel pairs along the horizontal and vertical lines.

The transition probabilities estimated from the local orientation fields computed by GR with different values of α in eight 1-look pure Nakagami noise images of size 4096×4096 pixels can be found in table 5.2, where all pixel pairs along the horizontal and vertical lines are considered to estimate the transition probabilities. One can see that, as α increases, more dependencies are introduced between adjacent pixels. It should be noticed that the dependencies introduced between local orientations are consequences of strong filtering, the transition probabilities do not rely on the number of looks of the images, as can be seen from those transition probabilities estimated from eight 3-look pure Nakagami noise images of size 4096×4096 pixels shown in table 5.3, where the transition probabilities for all α values approximately remain the same as those estimated in 1-look pure Nakagami noise images. Therefore, for the Markovian *a contrario* model defined with a certain value of α , it is applicable to images of any number of looks.

TABLE 5.2 – Estimated transition probabilities of the first order Markov chain for $\tau = 22.5^\circ$ and different values of α in the local orientation fields of eight 1-look pure noise images of size 4096×4096 pixels.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
$\mathbb{P}(1 1)$	0.245866 ± 0.000052	0.406489 ± 0.000060	0.512974 ± 0.000061	0.586310 ± 0.000060	0.634831 ± 0.000059
$\mathbb{P}(1 0)$	0.109585 ± 0.000014	0.085222 ± 0.000013	0.069719 ± 0.000012	0.059153 ± 0.000011	0.052201 ± 0.000010

TABLE 5.3 – Estimated transition probabilities of the first order Markov chain for $\tau = 22.5^\circ$ and different values of α in the local orientation fields of eight 3-look pure noise images of size 4096×4096 pixels.

	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
$\mathbb{P}(1 1)$	0.243331 ± 0.000052	0.406063 ± 0.000060	0.513112 ± 0.000061	0.586526 ± 0.000060	0.635020 ± 0.000059
$\mathbb{P}(1 0)$	0.108522 ± 0.000014	0.084959 ± 0.000013	0.069600 ± 0.000012	0.059092 ± 0.000011	0.052151 ± 0.000010

5.4 The complete LSDSAR algorithm

The proposed LSDSAR algorithm relies on three main steps, namely *region growing*, *rectangular approximation* and *line segment validation*, 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 5.1 ;

-
- (ii) *Rectangular approximation* : this step is identical to that of the original LSD algorithm ;
 - (iii) *Line segment validation* : in this step, we use the Markovian *a contrario* model as the background model to validate the candidate line segments.
-

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 , the density threshold D and the estimated transition probabilities for the first order Markov chain with different α values.

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 $(n(r_0), k) \in \{0, \dots, N_{\max}\}^2$), needed in (5.6).
- 4) Define NFA' as in (5.6), using $N_{\mathcal{R}} = 3 \cdot (MN)^{5/2}$.

for $P \in \text{OrderedList}$ **do**

```

region ← RegionGrow(P, τ)
r ← Rectangle(region)
while AlignedPixelDensity(r, τ) < D do
  region ← CutRegion(region)
  r ← Rectangle(region)
nfa ← NFA'(r)
if nfa ≤ ε then L ← L ∪ {r}
else
  r ← ImproveRectangle(r)
  nfa ← NFA'(r)
  if nfa ≤ ε then L ← L ∪ {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*). Note that, in step 3), one precomputation is needed per considered value of the angle tolerance (τ , $\tau/2$ and $\tau/4$).

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 the same as those used in LSD, which are carefully described in Grompone von Gioi et al. [2012]. The routine *AlignedPixelDensity*, also explicitly defined in Grompone von Gioi et al. [2012], 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.

5.5 Parameter settings

Algorithm 1 relies on four parameters, ε , α , τ and D . As usual with the *a contrario* algorithms, the NFA-property (4.13) 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 $\tau = 22.5^\circ$ 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.

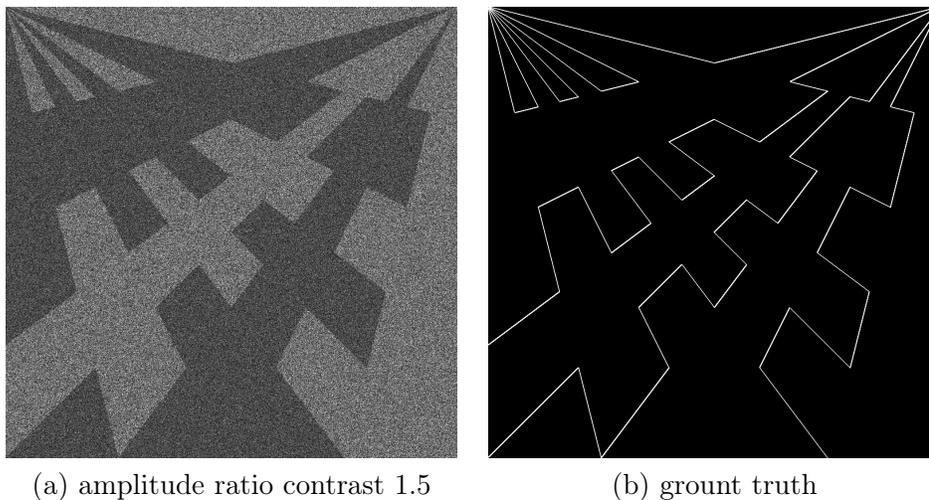


FIGURE 5.17 – A 1-look synthetic edge image with amplitude ratio contrast 1.5 and its associated edge ground truth. The size of images are 1024×1024 pixels.

First, we 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 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 (5.6) provides an effective control of the number of false detections.

In order to study the influences of α values on the performances of GR, we use ROC curves to evaluate the detection capability of gradient magnitude computed by GR in simulated edge images of size 1024×1024 pixels. Since 1-look SAR images are with the best resolution, we evaluate the detection capability of GR in 1-look images. The ratio contrasts of the edges in the simulated images vary from 1.2 to 1.9 with step 0.1. One example of the simulated edge images with contrast 1.5 and its associated ground truth can be found in figure 5.17.

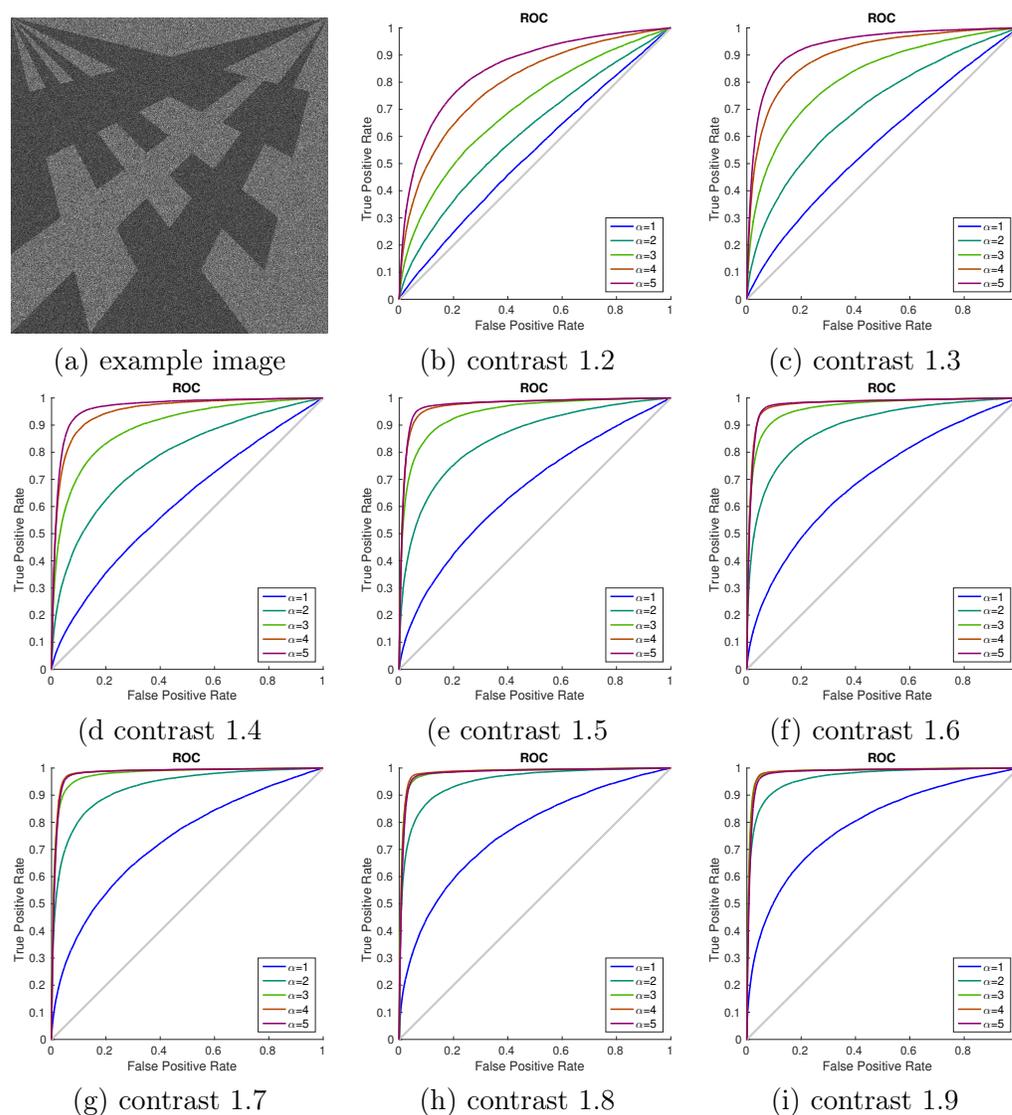


FIGURE 5.18 – A 1-look example image and ROC curves of gradient magnitude fields computed by GR with different values of α in eight 1-look synthetic edge images with amplitude ratio contrast of the edge varying from 1.2 to 1.9 with step 0.1. The size of images are 1024×1024 pixels.

The ROC curves computed using GR with different α values (the values of α vary from 1 to 5 with step 1) can be found in figure 5.18. From figure 5.18 we can see that

with the increase of α values, the detection capability of the magnitude field computed by GR becomes higher. Besides, the differences between GR with different α values in the detection capability of the magnitude field becomes smaller when the ratio contrasts of the edge become large. From figure 5.18 we can assume that it is better to choose a large α value.

As mentioned before, the choice of α values should be a trade-off between the detection capability of the gradient magnitude and the efficiency of the first order Markov chain to take into account of dependencies. The efficiency of the first order Markov chain can be checked through the number of false detections by LSDSAR in pure noise images. Table 5.4 gives the average number of false detections obtained by applying Algorithm 1 on eight 1-look pure Nakagami noise of size 4096×4096 pixels with $\varepsilon = 1$ and for different values of α . We can see in Table 5.4 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 reasonable, and considering that the performances of GR with $\alpha = 4$ and $\alpha = 5$ seem comparable and both are better than those with smaller α values, 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 figure 5.19. In situations where the accurate detection of close line segments is required, a smaller value, such as $\alpha = 2$, can be used.

TABLE 5.4 – Influence of the setting of α on the number of (Average number of false) detections obtained using LSDSAR on eight 1-look pure Nakagami noise images of size 4096×4096 pixels.

Parameter	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
Number of detections	0 ± 0	0.63 ± 0.26	7.38 ± 0.94	50.88 ± 2.33	129.63 ± 2.89

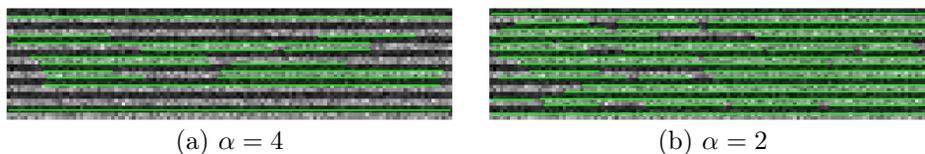


FIGURE 5.19 – Line segment detection on 20-look synthetic image of size 32×128 pixels containing horizontal edges with 2-pixel width and amplitude contrast of 1.8.

It remains to choose a proper value 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 Grompone von Gioi et al. [2010], 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

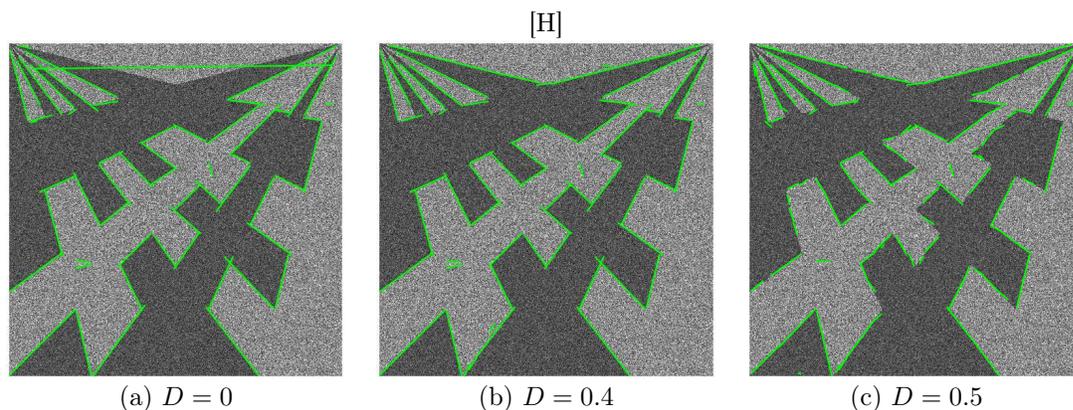


FIGURE 5.20 – Line segment detection on 3-look synthetic edge image (contrast 1.6) for different value of the density threshold.

of D will have the effect of over-cutting the line segments into small subsegments. This phenomenon is illustrated in figure 5.20, 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 figure 5.21 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 .

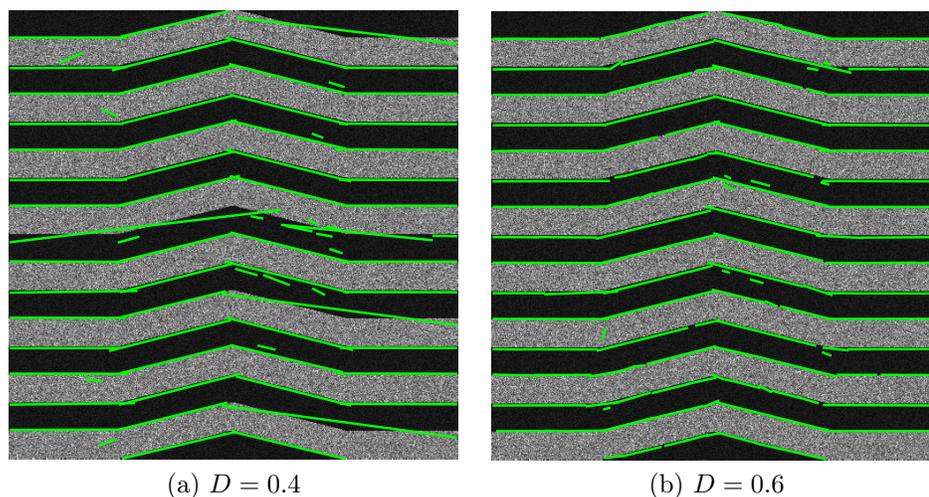


FIGURE 5.21 – Line segment detection results on 3-look synthetic edge images (contrast 1.4) for two different values of D . The angle differences of the problematic line segments are smaller than the angle tolerance is 22.5 degrees.

Chapitre 6

Experiments

6.1 Ability to control NFA

6.1.1 The number of false detections in pure noise images

One of the requirements of the proposed LSDSAR algorithm is its ability to control the number of false detections, more precisely, the ability of the new Markovian *a contrario* model to control the number of false detections in random images. To demonstrate this, we compare LSDSAR with 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 (4.12) instead of NFA' as defined in formula (5.6).

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 6.1 (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 6.1 (column 3) that, thanks to the Markov chain modeling of 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 .

TABLE 6.1 – 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 pixels.

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

6.1.2 The influence of image size on the Number of False Alarms (NFA)

Though theoretically on average, at most one false detection is allowed in a random image, regardless of the size of images. However, we observe that the size of images influences the number of false detections. LSDSAR with different α values is applied to pure noise images of different size and the average number of false detections is computed for each size of images. Specifically, we simulate eight 1-look pure noise images of size 8192×8192 pixels, eight 1-look pure noise images of size 4096×4096 pixels, 128 1-look pure noise images of size 1024×1024 pixels and 512 1-look pure noise images of size 512×512 pixels. The average number of false detections computed for each size of images, and each value of α can be found in table 6.2. From table 6.2 we can conclude that when $\alpha = 1$ and $\alpha = 2$, the Markovian *a contrario* model is efficient enough to model the dependencies between local orientations, the number of false detections detected by LSDSAR is strictly controlled regardless of the size of images. When α values become larger, since the hypothesis of the first order Markov chain is not strictly accurate, the number of false detections of LSDSAR is not strictly controlled and the number of false detections becomes larger when the size of images becomes larger. What can also be observed is that with the increase of α values, more dependencies are introduced and the LSDSAR method is less efficient to control the number of false detections.

TABLE 6.2 – The average number of false detections by applying LSDSAR with different values of α to images of different sizes.

Parameter	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
NFA (8192×8192)	$0 \pm$	$1.25 \pm$	$11.13 \pm$	$90.13 \pm$	$277.38 \pm$
	0	0.45	1.04	3.12	5.62
NFA (4096×4096)	$0 \pm$	$0.63 \pm$	$7.38 \pm$	$50.88 \pm$	$129.63 \pm$
	0	0.26	0.94	2.33	2.89
NFA (1024×1024)	$0 \pm$	$0.54 \pm$	$4.15 \pm$	$14.23 \pm$	$29.20 \pm$
	0	0.06	0.19	0.34	0.47
NFA (512×512)	$0 \pm$	$0.48 \pm$	$2.72 \pm$	$7.67 \pm$	$14.10 \pm$
	0	0.03	0.07	0.12	0.16

6.1.3 The influence of the ε value on the ability of LSDSAR to control the Number of False Alarms (NFA)

In order to study the influence of the NFA threshold ε values on the performances of LSDSAR, we apply LSDSAR with different ε values on 128 1-look pure noise images of size 1024×1024 pixels and the average number of false detections is computed as shown in table 6.3. What can be concluded is that when ε values are smaller than 100, the number of false detections for LSDSAR with $\alpha = 1$ and $\alpha = 2$ are strictly controlled. For LSDSAR with larger α values, though the number of false detections is not strictly controlled, the number of false detections remains in the reasonable scope. When ε is set to 100, there are too many false detections for LSDSAR with all α values.

TABLE 6.3 – Average number of (false) detections obtained using LSDSAR over 128 single look images with size 1024×1024 , for different values of α and different values of the NFA threshold ε .

NFA threshold ε	0.01	0.1	1	10	100	1000
NFA ($\alpha = 1$)	$0 \pm$	$0 \pm$	$0 \pm$	$0 \pm$	$242.43 \pm$	$349.91 \pm$
	0	0	0	0	1.37	1.73
NFA ($\alpha = 2$)	$0.11 \pm$	$0.23 \pm$	$0.54 \pm$	$1.20 \pm$	$775.01 \pm$	$1077.02 \pm$
	0.03	0.04	0.06	0.09	2.38	2.89
NFA ($\alpha = 3$)	$1.05 \pm$	$2.11 \pm$	$4.15 \pm$	$7.83 \pm$	$1015.69 \pm$	$1493.55 \pm$
	0.10	0.12	0.19	0.27	2.72	3.25
NFA ($\alpha = 4$)	$5.20 \pm$	$8.62 \pm$	$14.23 \pm$	$23.66 \pm$	$1062.63 \pm$	$1375.73 \pm$
	0.20	0.28	0.34	0.43	2.93	3.20
NFA ($\alpha = 5$)	$12.70 \pm$	$19.59 \pm$	$29.20 \pm$	$45.03 \pm$	$975.86 \pm$	$1259.23 \pm$
	0.31	0.38	0.47	0.62	2.42	2.86

6.2 Comparison in synthetic edge images

In this section, we study the performances of the proposed LSDSAR algorithm over synthetic edge 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, as mentioned above ;
- c. a recent line segment detector, called AFM, introduced in Xue et al. [2019], which is based on deep learning. We use the pretrained model provided by the author, and apply it to the logarithm of the amplitude of the SAR images ;
- d. a state-of-the-art Hough-transform based method, named IEFA, proposed in Wei and Feng [2016]. IEFA relies on a ratio-based gradient Shui and Cheng [2012] and a threshold to suppress the pixels with small gradient magnitude. Then, non-maximum suppression Canny [1986], Jiang et al. [2009] 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 Wei and Feng [2016], O’Gorman and Clowes [1976], Bonci et al. [2005]. We use our own implementation since the code by the authors is not available online.

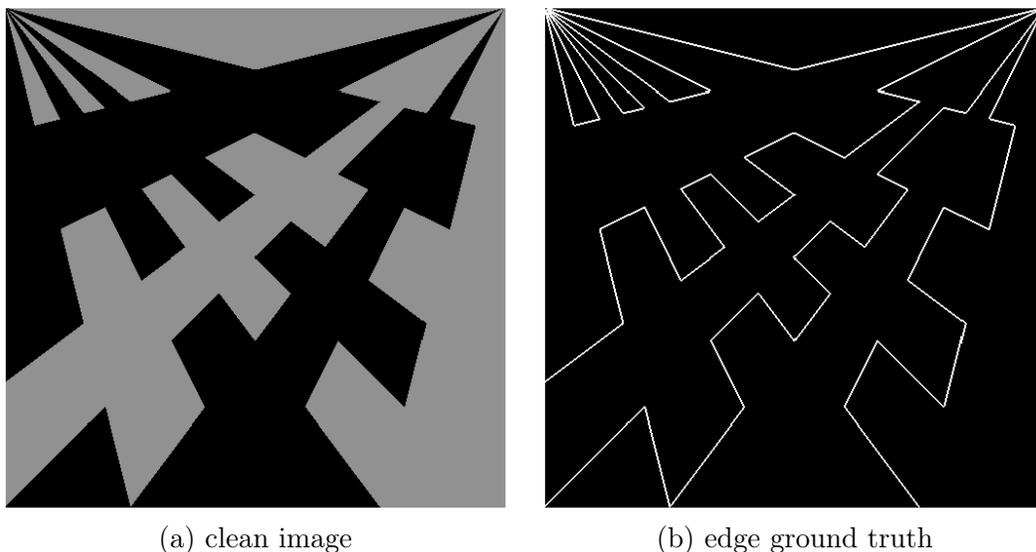


FIGURE 6.1 – (a) synthetic image with a contrast amplitude of 1.5, (b) the corresponding edge ground truth.

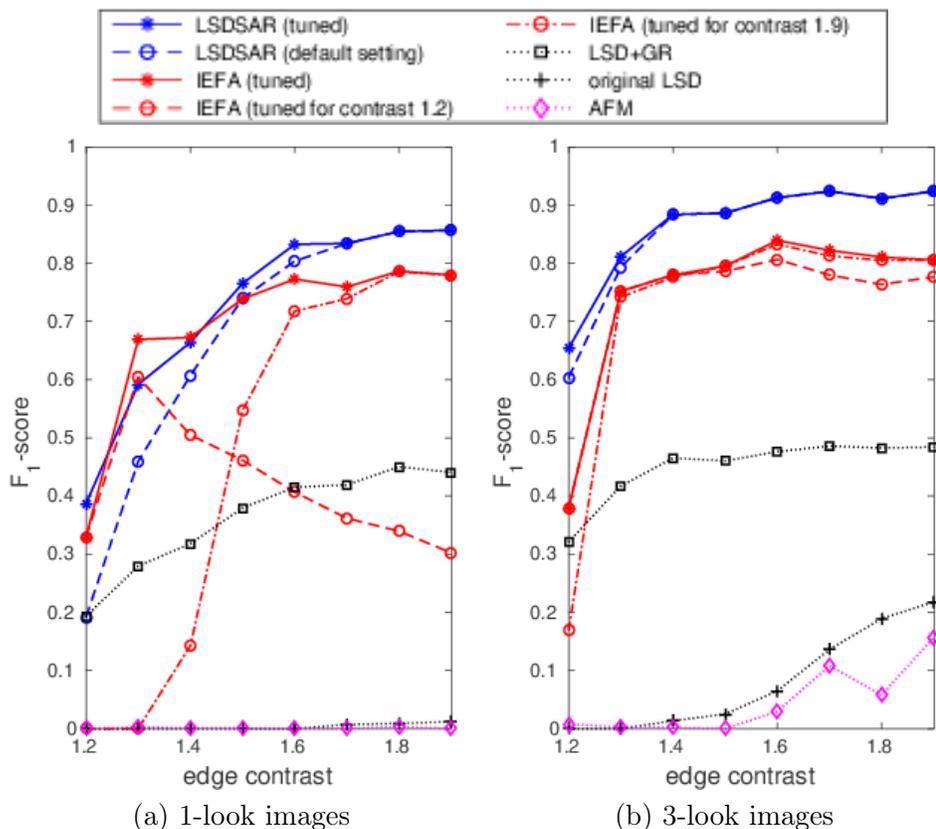


FIGURE 6.2 – Performances evaluations in terms of F_1 -score for different algorithms, using 1-look (a) and 3-look (b) synthetic edge images with contrast values ranging from 1.2 to 1.9.

Unless explicitly mentioned, we will use the default setting $\varepsilon = 1$, $\alpha = 4$, $\tau = 22.5^\circ$, 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 Grompone von Gioi et al. [2012] 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.

In order to provide a quantitative comparison of the considered algorithms, we propose to use the F_1 -score computed in the binary line maps (in the line map, pixel value is set to 1 if the pixel belongs to a line segment, and 0 otherwise). The comparisons will be performed over some noisy single look or multi-look observation of synthetic images for which a ground-truth is available (see Fig. 6.1).

We computed the performances in terms of F_1 -score for LSDSAR, IEFA, LSD+GR, LSD and AFM, 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 figure 6.2, 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. However, 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 figure 6.2, 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)* and *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 figure 6.2. 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 ($\varepsilon = 1$, $\alpha = 4$, $\tau = 22^\circ$ 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, the original LSD and AFM. Indeed, in the 3-look experiments displayed in figure 6.2 (b), the performance of LSDSAR is clearly above that of IEFA. Some of the images corresponding to this experiments are displayed in figure 6.3. In the 1-look experiment displayed in figure 6.2 (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 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. Concerning the failure of LSD and AFM, the main reason is that taking logarithm of the images will flatten the edges, while the additive noise changed from multiplicative noise is relatively strong. Therefore, both LSD and AFM obtain very bad results. For AFM, another reason is that it is pretrained from an optical dataset, and it is trained to process approximately clean images. If AFM is trained using a dataset which holds similar statistics to SAR images, the performances of AFM could be improved significantly. However, the lack of training dataset is a crucial difficulty, but it could be an interesting point which can be investigated in the future work.

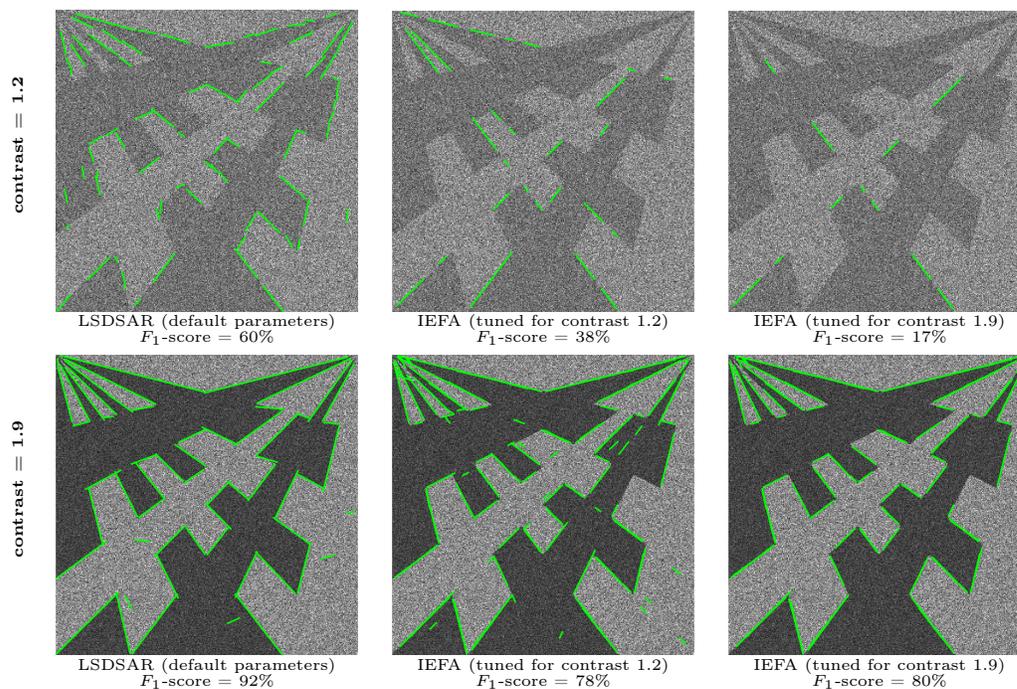


FIGURE 6.3 – Detection results obtained using LSDSAR (with default parameters) and IEFA (tuned for contrast 1.2 or 1.9) over 3-look synthetic images with contrast values equal to 1.2 (first row) or 1.9 (second row). We can see that, in both situations, LSDSAR with defaults parameters yields better performances in terms of F_1 -score than IEFA (even with tuned parameters).

The influence of α value and ε value on the performances of LSDSAR in synthetic data

To test the influence of α value on the performances, we test LSDSAR with different α values in both 1-look and 3-look synthetic edge images and display the corresponding F_1 -score in figure 6.4. It can be seen that in low contrast and very noisy situation, LSDSAR with larger α values is preferred, while for high contrast and less noisy cases, LSDSAR with smaller α values would be a better choice. LSDSAR with different ε settings is applied to

both 1-look and 3-look synthetic images. The F_1 -score curves can be found in figure 6.5. We can see that the performances of LSDSAR is not sensitive to ε values.

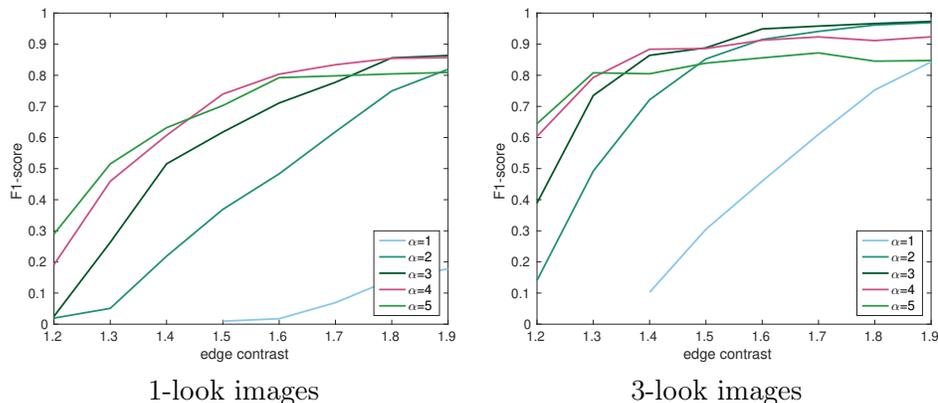


FIGURE 6.4 – F_1 -score curves obtained by applying LSDSAR with different α values to 1-look and 3-look synthetic images.

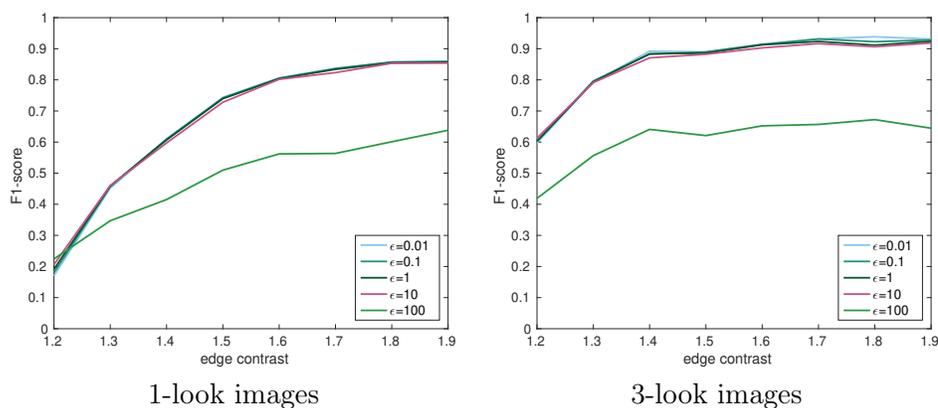


FIGURE 6.5 – F_1 -score curves obtained by applying LSDSAR with different ε values to 1-look and 3-look synthetic images.

Line segment detection results by LSDSAR with default parameter settings in 1-look and 3-look synthetic edge images

In order to further analyze the performances of LSDSAR with default parameter settings, we display the line segment detection results by LSDSAR in those 1-look and 3-look synthetic edge images in figure 6.6 and figure 6.7. The line segment detection results are corresponding to the F_1 -score curves of LSDSAR with default settings in figure 6.2. From figure 6.6 (b)-(c) we can see that the performances of LSDSAR in 1-look and low contrast situations are bad, since most true line segments are not detected though the number of false detections is controlled. The performances of LSDSAR could be improved by increasing the values of α and ε , but the performances of LSDSAR remain bad as can be seen from the F_1 -score curves of LSDSAR (tuned) in figure 6.2. The main reason for the poor performances of LSDSAR in these situations is that the gradient computation method GR is not efficient enough to highlight true edge pixels and suppress noise pixels in such

noisy and such low contrast situation. It can also be seen from figure 6.6 that in 1-look situations, the performances of LSDSAR are improved with the increase of contrast values. In 3-look situations as shown in figure 6.7, the performances of LSDSAR in low contrast situation (figure 6.7 (b)-(c)) seem reasonable, which is probably because the images are less noisy. For 3-look and high contrast situations, LSDSAR detects almost all line segments. Furthermore, in all experiments, the number of false detections is controlled and only a few false line segments are detected. However, LSDSAR is not able to detect close line segments (in the corner of the images) because of the large α values. If the images are less noisy and close line segments are required to be detected, LSDSAR with smaller α values could be tested.

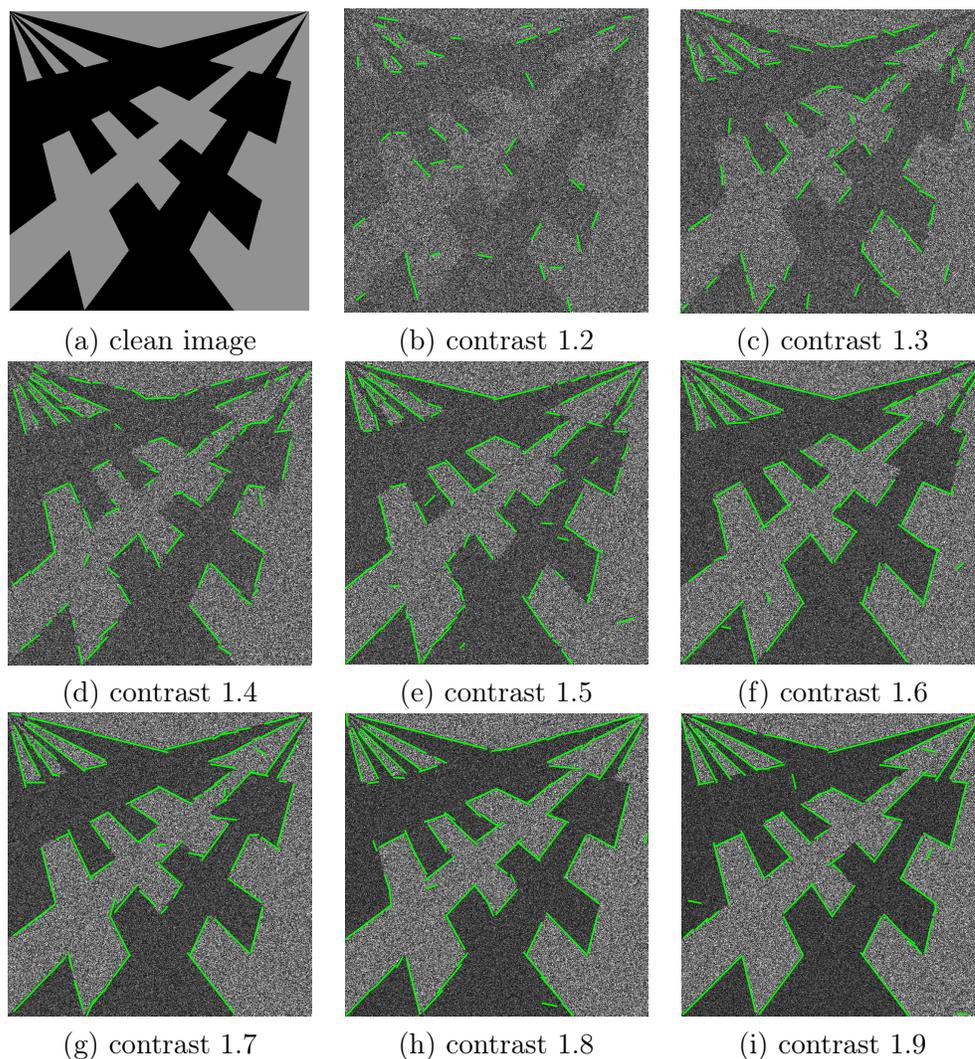


FIGURE 6.6 – An example of clean image and line segment detection results by LSDSAR with default parameter settings in eight 1-look synthetic edge images of size 512×512 pixels. The amplitude ratio contrasts of the images vary from 1.2 to 1.9.

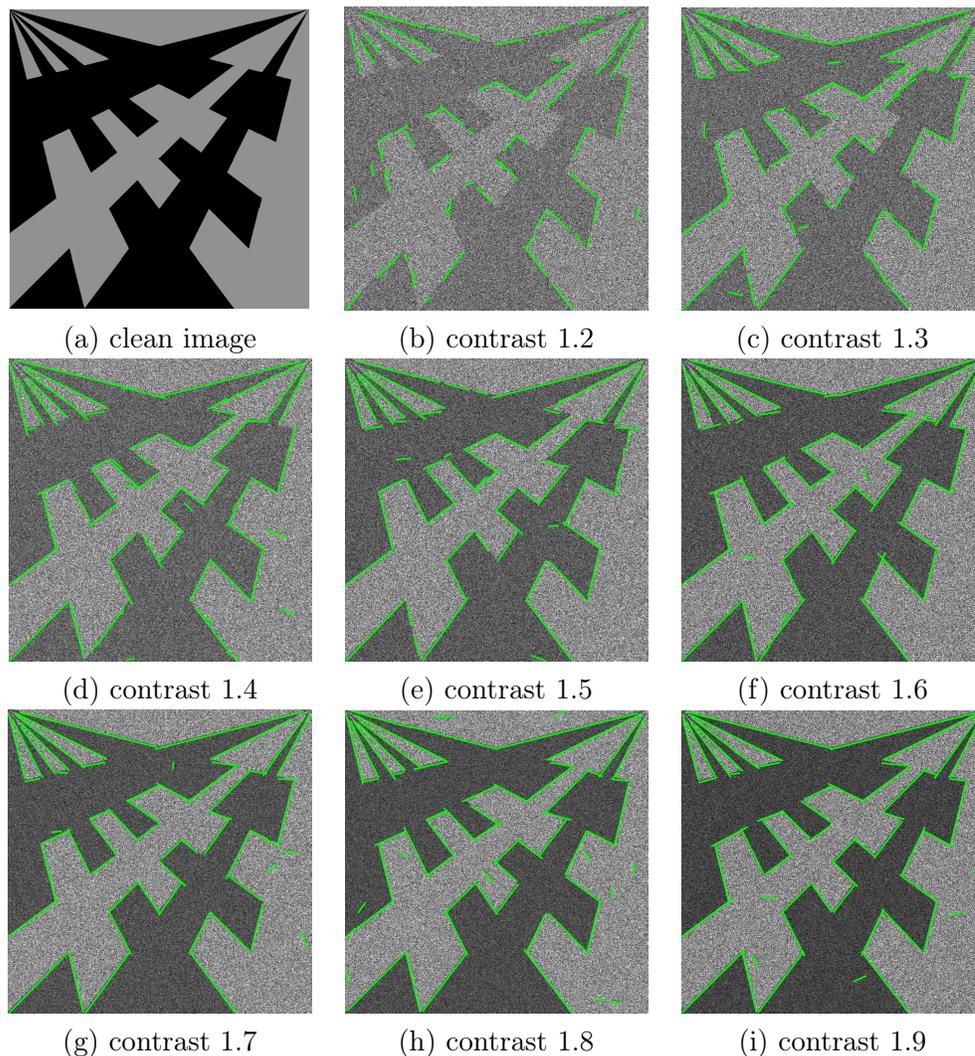
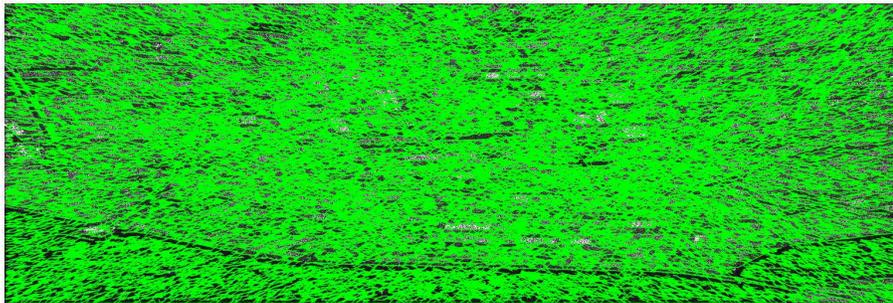


FIGURE 6.7 – An example clean image and line segment detection results by LSDSAR with default parameter settings in eight 3-look synthetic edge images of size 512×512 pixels. The amplitude ratio contrasts of the images vary from 1.2 to 1.9.

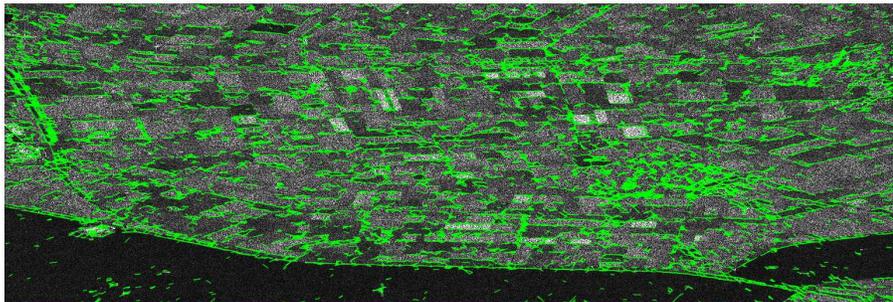
6.3 Comparison between LSDSAR and IEFA on real SAR images

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 section 2.3. 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 figure 6.8. 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 figure 6.9 and figure 6.10, we display the line segment detection results obtained using LSDSAR and IEFA over two different single-look SAR images.

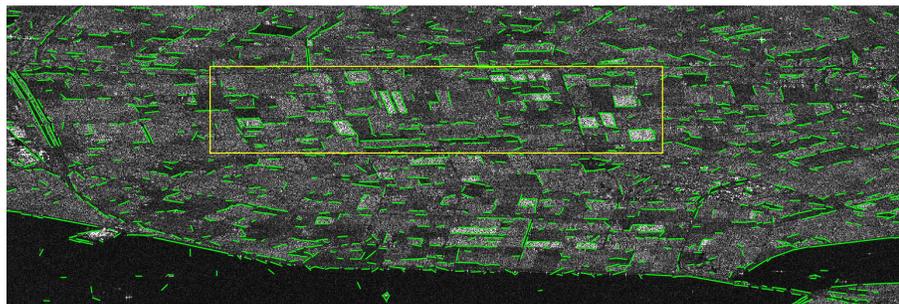


(a) IEFA (tuned for contrast 1.2)

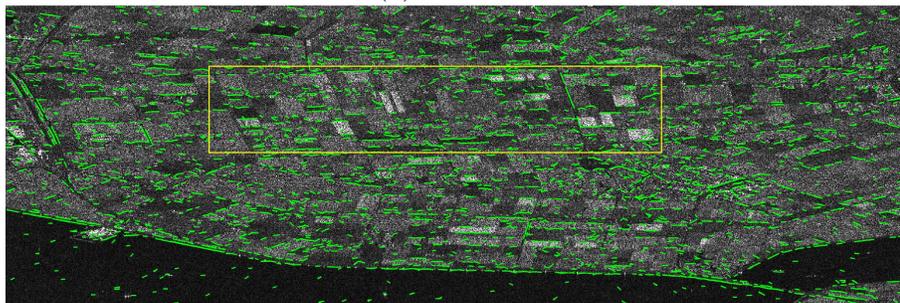


(b) IEFA (tuned for contrast 1.9)

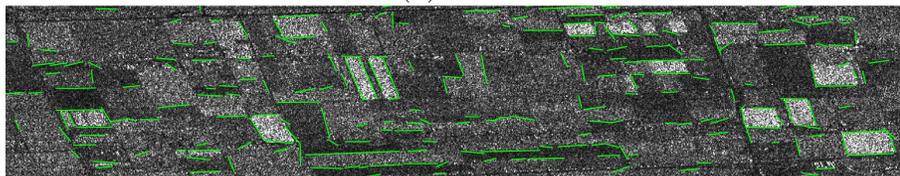
FIGURE 6.8 – Line segment detection obtained using IEFA (tuned for contrast 1.2) and IEFA (tuned for contrast 1.9) over a 1-look real SAR image.



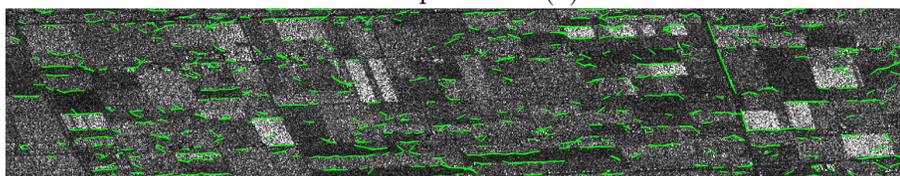
(a) LSDSAR



(b) IEFA



close-up view of (a)



close-up view of (b)

FIGURE 6.9 – Line segment detection using LSDSAR (a) and IEFA (b) over a 1-look Sentinel 1 SAR image (Lelystad).

From the images displayed in figure 6.9 and figure 6.10, 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 figure 6.11 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.

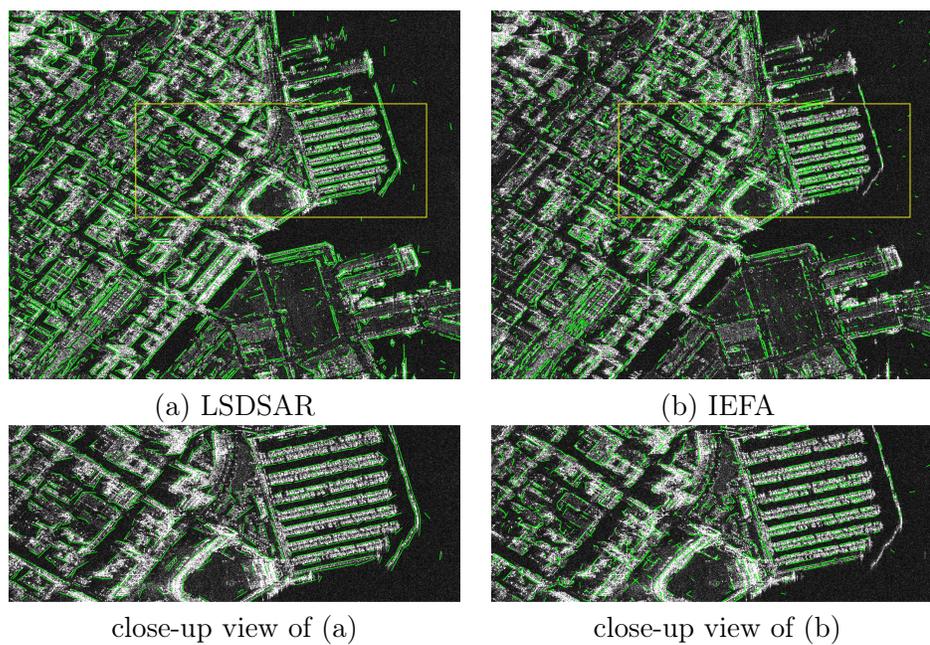
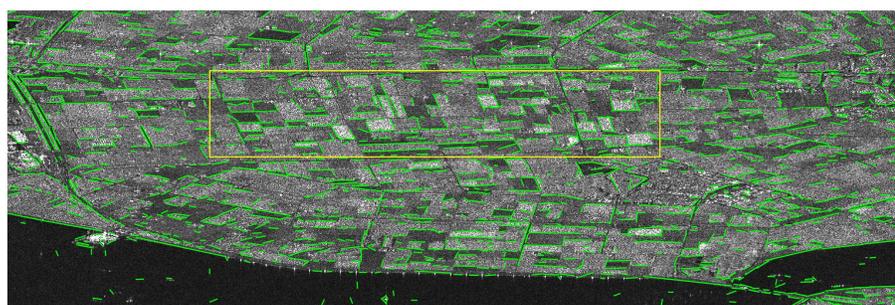
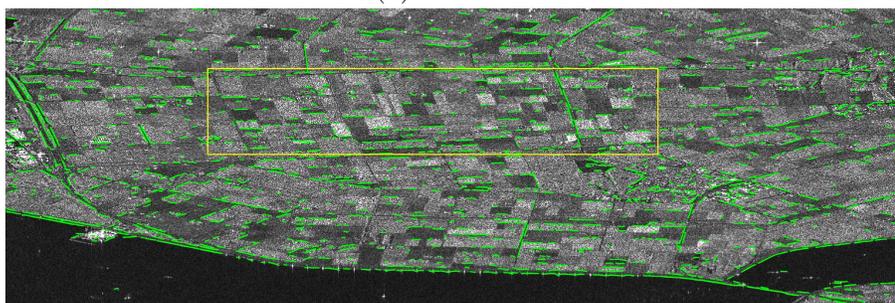


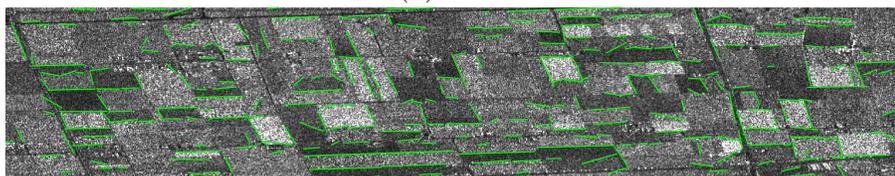
FIGURE 6.10 – Line segment detection using LSDSAR (a) and IEFA (b) over a 1-look TerraSAR-X SAR image (San Francisco).



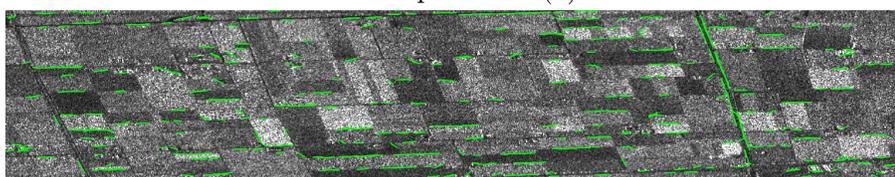
(a) LSDSAR



(b) IEFA



close-up view of (a)



close-up view of (b)

FIGURE 6.11 – Line segment detection using LSDSAR (a) and IEFA (b) over a 3-look Sentinel 1 SAR image (Lelystad).

6.4 Summary of LSDSAR

In this part, 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 four parameters which are 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 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.

6.5 Inefficiency of LSDSAR in 1-look and low-contrasted situations

Except for the imperfectness of the Markovian *a contrario* model, the main reason for the failure of LSDSAR to detect many true line segments in some situations is that the gradient computation methods are not efficient enough to highlight true edge pixels and suppress noise pixels, thus the local orientations will not be accurate enough. The gradient magnitude fields computed by GR with $\alpha = 4$ in four 1-look synthetic edge images with contrasts varying from 1.2 to 1.5 with step 0.1 can be found in figure 6.12 (corresponding to line segment detection results in figure 6.6 (b)-(e)). From the gradient magnitude fields computed by GR with $\alpha = 4$ shown in figure 6.12 we can deduced that the accuracy of local orientations (the angle of the direction along the edges) is pretty low for 1-look and low contrast situations, especially for contrast 1.2 (figure 6.12-(a)) and contrast 1.3 (figure 6.12-(b)), which gives an explanation for the poor performances of LSDSAR in those images as can be seen in figure 6.6-(b) and figure 6.6-(c). A quantitative proof of the poor performances of LSDSAR in those images can also be seen in figure 6.2. What can also be observed from figure 6.12 is that with the increase of edge contrasts, the edge pixels are better highlighted (figure 6.12) and thus the line segment detection results by LSDSAR are improved (figure 6.6). The gradient magnitude fields computed by GR in 3-look realizations of those edge images with contrast varying from 1.2 to 1.5 with step 0.1 can be found in figure 6.13 (corresponding to line segment detection results in figure 6.7 (b)-(e)). From figure 6.13 we can see that GR is more efficient in less noisy situations. In 3-look images, GR is able to highlight true edge pixels even in low contrast situations (figure 6.13 (a)-(b)), the corresponding line segment detection results by LSDSAR has improved a lot (figure 6.7) compared to those in 1-look images (figure 6.6).

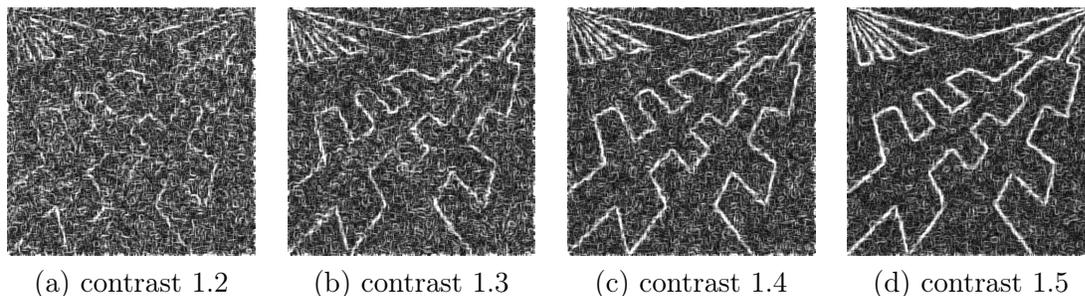


FIGURE 6.12 – The gradient magnitude fields computed by GR with $\alpha = 4$ in four 1-look synthetic edge images (corresponding to line segment detection results in figure 6.6 (b)-(e)).

In order to propose a more efficient gradient computation method, which is able to distinguish between true edge pixels and noise pixels even in 1-look and low contrast

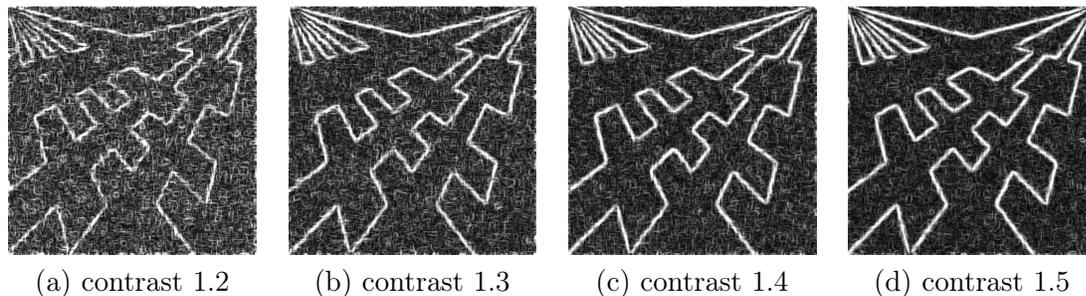


FIGURE 6.13 – The gradient magnitude fields computed by GR with $\alpha = 4$ in four 3-look synthetic edge images (corresponding to line segment detection results in figure 6.7 (b)-(e)).

situations, we propose to use Convolutional Neural Networks (CNNs) since it has shown its efficiency to highlight edge pixels and suppress noise pixels in natural images Xie and Tu [2015, 2017], Yang et al. [2016], Xu et al. [2017], Liu et al. [2017b, 2019], as what we will describe in the next part. Due to the efficiency of the CNN based gradient computation method, a simple non-maxima suppression and a plain thresholding are enough to obtain reasonable edge maps even in 1-look and low contrast situations, as we will also see in the next part. Furthermore, it will be shown that the CNN-based gradient computation method may help to improve the line segment detection results.

PART II

Edge detection in SAR images using CNNs

Chapitre 7

Introduction

Edges are important features in Synthetic Aperture Radar (SAR) images. The edge features can be used as low level features for many applications like line segment detection Wei and Feng [2016], Wei et al. [2017], SAR image segmentation Yu et al. [2012], Song et al. [2013], coastline detection Lee and Jurkevich [1990], Liu et al. [2017a], image registration Chen et al. [2014], Zhang et al. [2015] and SAR image despeckling Dai et al. [2004]. What is more important, due to the strong differences between images of different modalities, like SAR and optical data, detecting low level features is usually a preprocessing step for subsequent applications like image registration. As one of the most important low level features, edges will be helpful for the joint use of SAR and optical images. In this second part of the thesis, we develop a CNN based edge detector for SAR images, which addresses the most challenging cases for edge detection task, namely the edge detection in 1-look SAR images. In addition, we will show that the gradient magnitude fields computed by the proposed CNN based methods outperform the existing gradient computation methods and can be used to improve the line segment detection results.

7.1 Existing approaches

Due to the strong multiplicative speckle noise in SAR images, methods developed for optical images, which are usually based on pixel value differences, produce more false edges in brighter areas and thus are not suitable for SAR images. Many researches have been dedicated to edge detection in SAR images in the past years. In Touzi et al. [1988], the Ratio of Average (ROA) was proved to have a constant false alarm rate for SAR images (CFAR property). The ratio operator is applied along four directions and the minimum normalized ratio is used to compute the edge maps. The direction corresponding to the minimum normalized ratio is regarded as the edge orientation. A threshold determined by a given probability of false alarms and a morphological operator are then applied to obtain a binary thin edge map. However, ROA is optimal only for isolated step edges. An efficient multiedge detector, ROEWA, was proposed afterwards in Fjørtoft et al. [1998]. The Ratio of Exponentially Weighted Average was shown to be optimal in terms of minimum mean square error (MMSE) under the hypothesis of a stochastic multi-edge model. The method is based on ROA, but averages are weighted by a decreasing exponential function, allowing a better detection of multiple edges close to each other. Besides, instead of computing the ratio along four directions, the ratio is computed along the horizontal and vertical directions and the normalized ratios are considered as the horizontal and vertical

components of the gradient magnitude. A modified watershed algorithm is then used to threshold the Edge Strength Map and a region merging algorithm is used to eliminate the false edge pixels. Edge detectors using different shape of window functions were introduced later in Shui and Cheng [2012] and Wei et al. [2016]. Non-maxima suppression Canny [1986] and hysteresis thresholding are applied to obtain the binary thin edge map. An edge compensation strategy was also introduced in Wei et al. [2016] to extract weak edge pixels. To reduce the influence of isolated strong bright points in real SAR images, an Anisotropic Morphological Directional Ratio (AMDR) Shui and Fan [2018] was proposed by replacing the weighted average filter with the weighted median filter. The edge localization accuracy in the Edge Strength Map (ESM) and the Edge Direction Map (EDM) was then improved by a multiplicative spatial and directional matching filter. By introducing the improved ESM and EDM into the routine of Canny edge detector Canny [1986], the resulting edge detector is able to obtain a binary thin edge map. The connectivity of edges is finally improved by an edge remedy strategy.

7.2 Training CNNs for edge detection in SAR images

The performances of the edge detectors developed for SAR images are still not fully satisfying, especially in the challenging 1-look situation. On the other hand, convolutional neural networks (CNNs) have proven to be successful for edge detection in natural images, with approaches such as Deepedge Bertasius et al. [2015], DeepContour Shen et al. [2015], HED Xie and Tu [2015, 2017], CEDN Yang et al. [2016], AMH-Net Xu et al. [2017] or RCF Liu et al. [2017b, 2019], which have permitted to improve significantly traditional edge detectors like Sobel Kittler [1983], Canny Canny [1986], Statistical Edge Konishi et al. [2003], Pb Martin et al. [2004], gPb Arbelaez et al. [2011] and Structured Edge Dollar and Zitnick [2015], at least on databases similar to the training sets. Motivated by this fact, we study the possibility to apply CNN-based methods to SAR images. We will pay a special attention on developing methods that are as CFAR as possible.

One crucial factor that contributes to the success of CNNs is the availability of training datasets with ground truth. A first difficulty for the present work is that, to the best of our knowledge, there is still no available training dataset for edge detection in SAR images. Considering that edges mostly correspond to changes in local brightness and textures (and color for color images), we assume that the problem of edge detection should not rely much on the specific content of the image. In this case, datasets of natural images used to train CNN could be used to train a SAR edge detector, provided images are modified using a reasonable noise model. On this basis, we propose to simulate a SAR-like training set by multiplying by speckle noise datasets of natural images. The natural images multiplied by speckle are called speckled optical images in the following.

Another difficulty in applying CNN-based method to SAR images is their specific dynamic ranges. The range of values is indeed much higher for SAR images than for natural images. To cope with this problem and ease the training of the network, we propose in this work to apply a pre-processing step to the SAR images by applying a first low-level edge detection step. This step is done using the Gradient by Ratio (GR) proposed in Delinger et al. [2015]. These pre-processed features are then given as input to the network to be trained (HED Xie and Tu [2015, 2017]). This strategy will be called GRHED in the following. In the next chapter, we will give the details about HED and the details about the simulated speckled optical dataset. The details about GRHED and its benefits will be discussed in subsequent chapters.

Chapitre 8

Presentation of the HED method and of the training dataset

8.1 Details of the HED method

8.1.1 Network architecture

The HED method, introduced in Xie and Tu [2015, 2017], relies on a fully convolutional network Long et al. [2015], Shelhamer et al. [2017], which is trained end-to-end to perform image-to-image prediction. The network architecture of HED is shown in table 8.1. The HED network is trimmed from the VGG-16 Simonyan and Zisserman [2015] net by discarding the last max-pooling layer and the 3 fully connected layers. Motivated by the deeply-supervised nets Lee et al. [2015], five side outputs are added to the convolutional layers just before the five max-pooling layers in the original VGG-16 net. The loss function of the HED network is composed of the loss function from the side layers and the loss function from the fused output. The final output of HED is an average of the side outputs and the fused output. In-network bilinear interpolation Long et al. [2015], Shelhamer et al. [2017] is used to upsample the side outputs so that they all have the same size as the edge ground truth. Notice that in table 8.1, all convolutional layers are constituted by convolutional filters with each filter being followed by a ReLU activation function (the ReLU function is the function $f(x) = \max(0, x)$) Krizhevsky et al. [2012]. The receptive field size of all convolutional filters are 3×3 pixels and the role of ReLU activation function is to introduce non-linearity. The max-pooling layer applies a 2×2 max filter with stride 2 over the input. Each pixel in the output of a max-pooling layer is the maximum value of a region with size 2×2 pixels in the input.

8.1.2 Training and testing phases

Training Given an image u in the training dataset, noting \mathbb{G} for its associated edge ground truth, where $u = \{u_j, j = 1, \dots, |u|\}$ denotes the input image and $\mathbb{G} = \{\mathbb{G}_j, j = 1, \dots, |\mathbb{G}|\}$ denotes the edge ground truth. The goal of training HED is to produce an output approaching the edge ground truth for each input image. Noting \mathbb{W} for the collection of parameters in all network layers (excluding those corresponding to side output layers), and noting $\mathbb{W}_{side} = (\mathbb{W}_{side}^{(1)}, \mathbb{W}_{side}^{(2)}, \dots, \mathbb{W}_{side}^{(5)})$ for the collection of parameters in those side

output layers, the objective function for the side output layers is defined as :

$$\mathcal{L}_{side}(\mathbb{W}, \mathbb{W}_{side}) = \sum_{m=1}^5 \beta_m \ell_{side}^{(m)}(\mathbb{W}, \mathbb{W}_{side}^{(m)}), \quad (8.1)$$

where ℓ_{side} denotes the loss corresponding to the side outputs. In HED, all loss functions are computed over all pixels in a training image and its associated ground truth.

TABLE 8.1 – The network architecture of HED. The convolutional layers are denoted as "conv(receptive field size)-(number of channels)". For brevity, we do not show the ReLU activation function following the convolutional filters in each convolutional layer.

conv3-64	
conv3-64	→ side output 1
MAX-POOLING	
conv3-128	
conv3-128	→ side output 2
MAX-POOLING	
conv3-256	
conv3-256	
conv3-256	→ side output 3
MAX-POOLING	
conv3-512	
conv3-512	
conv3-512	→ side output 4
MAX-POOLING	
conv3-512	
conv3-512	
conv3-512	→ side output 5

In order to balance the loss between positive/negative classes, a class-balanced cross-entropy loss function is used in formula (8.1) :

$$\begin{aligned} \ell_{side}^{(m)}(\mathbb{W}, \mathbb{W}_{side}^{(m)}) &= -\lambda \sum_{j \in \mathbb{G}_+} \log \mathbb{P}(\mathbb{G}_j = 1 | u : \mathbb{W}, \mathbb{W}_{side}^{(m)}) \\ &\quad - (1 - \lambda) \sum_{j \in \mathbb{G}_-} \log \mathbb{P}(\mathbb{G}_j = 0 | u : \mathbb{W}, \mathbb{W}_{side}^{(m)}), \end{aligned} \quad (8.2)$$

where $\lambda = \frac{|\mathbb{G}_-|}{|\mathbb{G}|}$, and $1 - \lambda = \frac{|\mathbb{G}_+|}{|\mathbb{G}|}$, $|\mathbb{G}_-|$ and $|\mathbb{G}_+|$ represent the non-edge and edge label sets. $\mathbb{P}(\mathbb{G}_j = 1 | u : \mathbb{W}, \mathbb{W}_{side}^{(m)}) = \sigma(a_j^{(m)}) \in [0, 1]$ with $\sigma(\cdot)$ representing the sigmoid function and $a_j^{(m)}$ representing the activation value at pixel j . The edge probability map produced by each side layer is thus defined as :

$$\hat{\mathbb{G}}_{side}^{(m)} = \sigma(\hat{A}_{side}^{(m)}), \quad (8.3)$$

where $\hat{A}_{side}^{(m)} \equiv \{a_j^{(m)}, j = 1, 2, \dots, |\mathbb{G}|\}$ represents the activations of the output of the side layer.

The final fused output is a weighted fusion of those side outputs :

$$\hat{\mathbb{G}}_{fuse} \equiv \sigma\left(\sum_{m=1}^5 h_m \hat{A}_{side}^{(m)}\right), \quad (8.4)$$

and the corresponding loss function for the fused output \mathcal{L}_{fuse} is defined as :

$$\mathcal{L}_{fuse}(\mathbb{W}, \mathbb{W}_{side}, h) = \text{Dist}(\mathbb{G}, \hat{\mathbb{G}}_{fuse}), \quad (8.5)$$

where h denotes the fusion weights which are learned during training. $\text{Dist}(\cdot, \cdot)$ represents the distance between the fused output and the edge ground truth, which is measured by the cross entropy loss.

The objective function that needs to be minimized during training using stochastic gradient descent is thus defined as :

$$(\mathbb{W}, \mathbb{W}_{side}, h)^* = \arg \min(\mathcal{L}_{side}(\mathbb{W}, \mathbb{W}_{side}) + \mathcal{L}_{fuse}(\mathbb{W}, \mathbb{W}_{side}, h)). \quad (8.6)$$

Testing phase Given a testing image u , six edge probability maps (five side outputs and one fused output) are obtained from HED :

$$(\hat{\mathbb{G}}_{fuse}, \hat{\mathbb{G}}_{side}^{(1)}, \dots, \hat{\mathbb{G}}_{side}^{(5)}) = \text{CNN}(u, (\mathbb{W}, \mathbb{W}_{side}, h)^*). \quad (8.7)$$

The final output is computed as the average of all the outputs :

$$\hat{\mathbb{G}}_{HED} = \text{Average}(\hat{\mathbb{G}}_{fuse}, \hat{\mathbb{G}}_{side}^{(1)}, \dots, \hat{\mathbb{G}}_{side}^{(5)}). \quad (8.8)$$

8.2 Speckled optical dataset

Since the aim of this work is to train CNNs for edge detection in SAR images, the training dataset is of crucial importance for the performances of the edge detector. Instead of doing the tedious job of edge labeling, we leverage the available optical dataset for edge detection in natural images. This is motivated by the fact that the edges are mainly corresponding to the changes in brightness and textures so that the task of edge detection should not be influenced a lot by the image contents.

In order to simulate a SAR dataset, we should take into account the statistics of SAR images. Recall from section 2.2 that the amplitude of the backscattered electro-magnetic field of a homogeneous area with mean intensity $\langle \mathbb{I} \rangle$ follows a Nakagami distribution :

$$f(t|\langle \mathbb{I} \rangle) = \frac{2}{\Gamma(L)} \left(\frac{L}{\langle \mathbb{I} \rangle}\right)^L t^{2L-1} e^{-(Lt^2/\langle \mathbb{I} \rangle)}, \quad (8.9)$$

L being the number of looks of the image. For 1-look images, $L = 1$ and the amplitude of a physically homogeneous area follows a Rayleigh distribution. Another way of modeling 1-look data is the multiplicative model : $t = \sqrt{\langle \mathbb{I} \rangle} s$, s representing the speckle noise and following the Rayleigh distribution given in eq. (8.9) with $\langle \mathbb{I} \rangle = 1$ and $L = 1$.

Using the multiplicative noise model it is therefore easy to generate speckled data by multiplying an image by s . Although not truly verified (for real data the speckle is spatially

correlated and the fully developed model of Goodman is verified only for rough surfaces Goodman [1975]), this model is widely used to generate simulated data.

In this work, we use the BSDS500 Arbelaez et al. [2011] dataset with the same data augmentation as in HED Xie and Tu [2017] for training and validation, to simulate a SAR dataset. Specifically, each image in the training and validation set is rotated by 16 angles, flipped horizontally, and rescaled to the 50%, 100%, 150% of its original size. There is no data augmentation for the testing images. The resulting speckled optical dataset, which we will call BSDS500-speckled in the following, is formed by multiplying the grayscale intensity channel of each color image with 1-look speckle noise. It contains $300 \times 16 \times 2 \times 3 = 28800$ images for training (80%) and validation (20%) and 200 images for testing. After creating a simulated dataset for training, the next question is how to train HED using this dataset, especially how to enable the trained model to work well in real SAR images. In order to fulfill this aim, we will first analyze what HED has learned during training, and then address the way to tackle the gap between speckled optical images and real SAR images.

8.3 What HED has learned through training

It is usually difficult to rigorously explain what CNNs have learned during training. In this section, we try to understand what HED has learned through training using the simulated SAR dataset. We try to demonstrate experimentally our assumption that HED is learning to process samples drawn from many different distributions : it learns to suppress all pixels located in homogeneous areas and it learns to emphasize pixels located on the boundaries between two homogeneous areas. In addition, we try to show that HED directly trained on the images (or logarithm of images) does not have CFAR since the distribution of the output produced by HED in homogeneous areas depends on their mean intensity values.

Under the hypothesis that both speckled optical images and real SAR images can be divided into homogeneous areas with many different mean intensity values, HED mainly learns to do two jobs from the speckled optical dataset : first, HED learns to ensure that there is no detection of edges in homogeneous areas ; second, HED learns to detect the boundaries between any two different homogeneous areas. According to formula (8.9), the amplitude pdf of an homogeneous area in real SAR images follows a Rayleigh distribution in 1-look situations, therefore, HED learns to process samples drawn from many different distributions. In the case of real SAR images, taking the logarithm of SAR images is a usual preprocessing step because it can change multiplicative noise into additive noise. We will use two ways of training using the speckled optical dataset : 1) training HED on the images directly, that we will notate as HED (the output of HED is notated as \hat{G}_{HED}) ; 2) training HED on the logarithm of images in the dataset, which we will notate as HED-log (the output of HED-log will be notated as $\hat{G}_{HED-log}$). Global mean subtraction is the preprocessing step that we use before feeding the data into convolutional layers, for both HED and HED-log. In order to test HED and HED-log which are trained using the dataset, we evaluate their behaviours on two kinds of images : pure noise images with homogeneous areas of different mean intensity values, and synthetic edge images.

8.3.1 Testing HED and HED-log in pure noise images

The first kind of cases that HED and HED-log have to learn to process are homogeneous areas, where no edge should be detected. In order to analyze the way of HED and HED-log

to process homogeneous areas, we simulate eight kinds of 1-look pure noise images of size 1024×1024 pixels. For each kind of pure noise image, the underlying mean intensity value of the homogeneous area is one of these eight possibilities : 50^2 , 80^2 , 120^2 , 150^2 , 180^2 , 200^2 , 230^2 , and 250^2 . For pure noise images with each mean intensity value, there are 20 random realizations.

Figure 8.1 gives the histograms of the original pure noise images and the histograms of the gradient magnitude fields produced by HED. Figure 8.2 gives the histograms for the logarithm of pure noise images and the histograms of the corresponding output produced by HED-log. From figure 8.1 and figure 8.2 we can see that only if the input data of HED layers follows the same distribution (with the same mean intensity values for pure noise images), the output of HED layers follows the same distribution. If the input data of HED layers follows different distributions, it is likely that the output of HED layers follows different distributions. In addition, the edge probabilities for almost all pixels in the pure noise images are small (smaller than 0.5 for example), for both HED and HED-log, which means that both HED and HED-log ensures almost no detection in pure noise images.

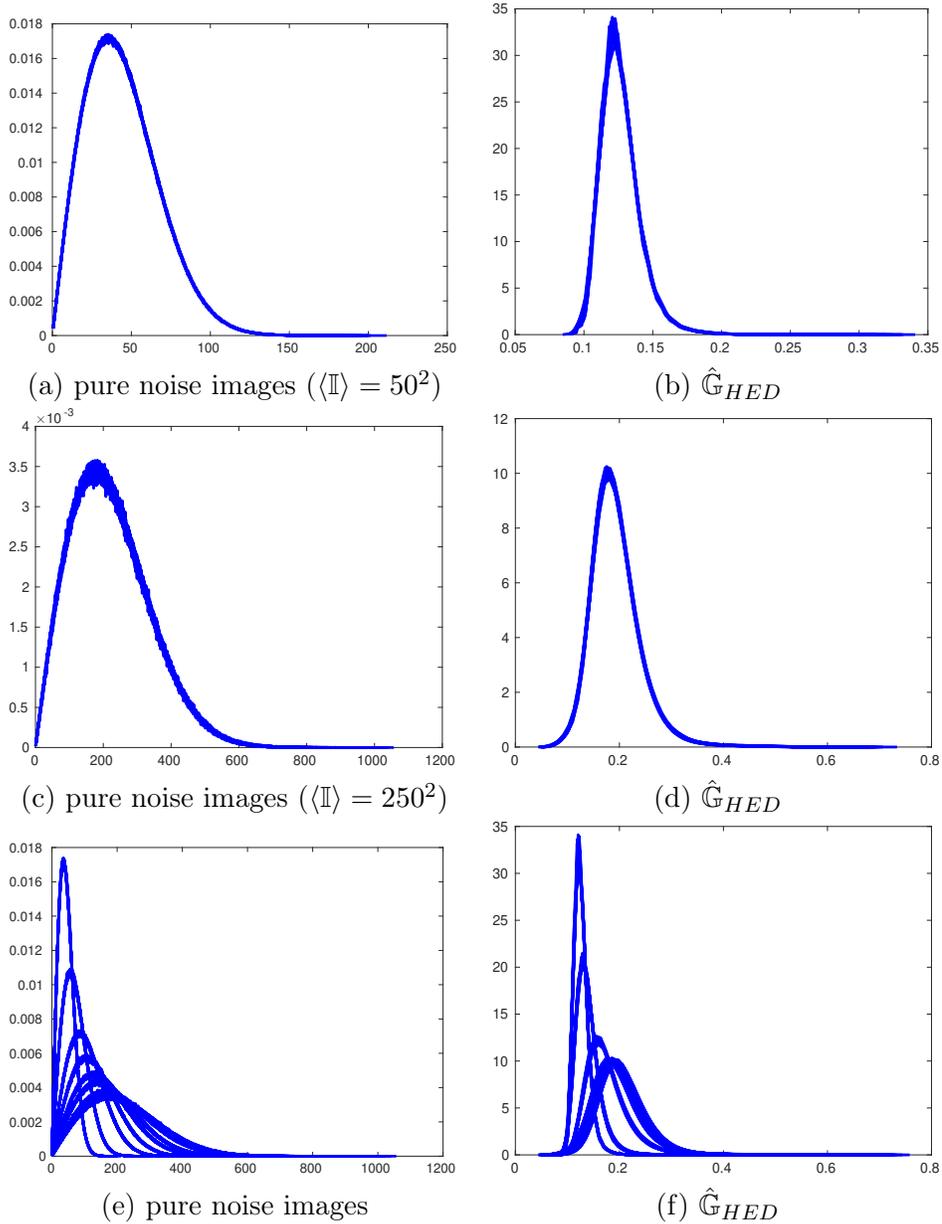


FIGURE 8.1 – The histograms of the original pure noise images with different mean intensity values and the histograms for the corresponding output of HED. (a) histograms of twenty 1-look pure noise images with the underlying mean intensity value 50^2 ; (b) histograms of the gradient magnitude field produced by HED in the twenty images of (a); (c) histograms of twenty 1-look pure noise images with the underlying mean intensity value 250^2 ; (d) histograms of the gradient magnitude field produced by HED in the twenty images of (c); (e) histograms of 160 1-look pure noise images with eight possible underlying mean intensity values, for each mean intensity value, there are 20 random realizations; (f) the histograms of the gradient magnitude field produced by HED in the 160 images of (e). The size of images are 1024×1024 pixels.

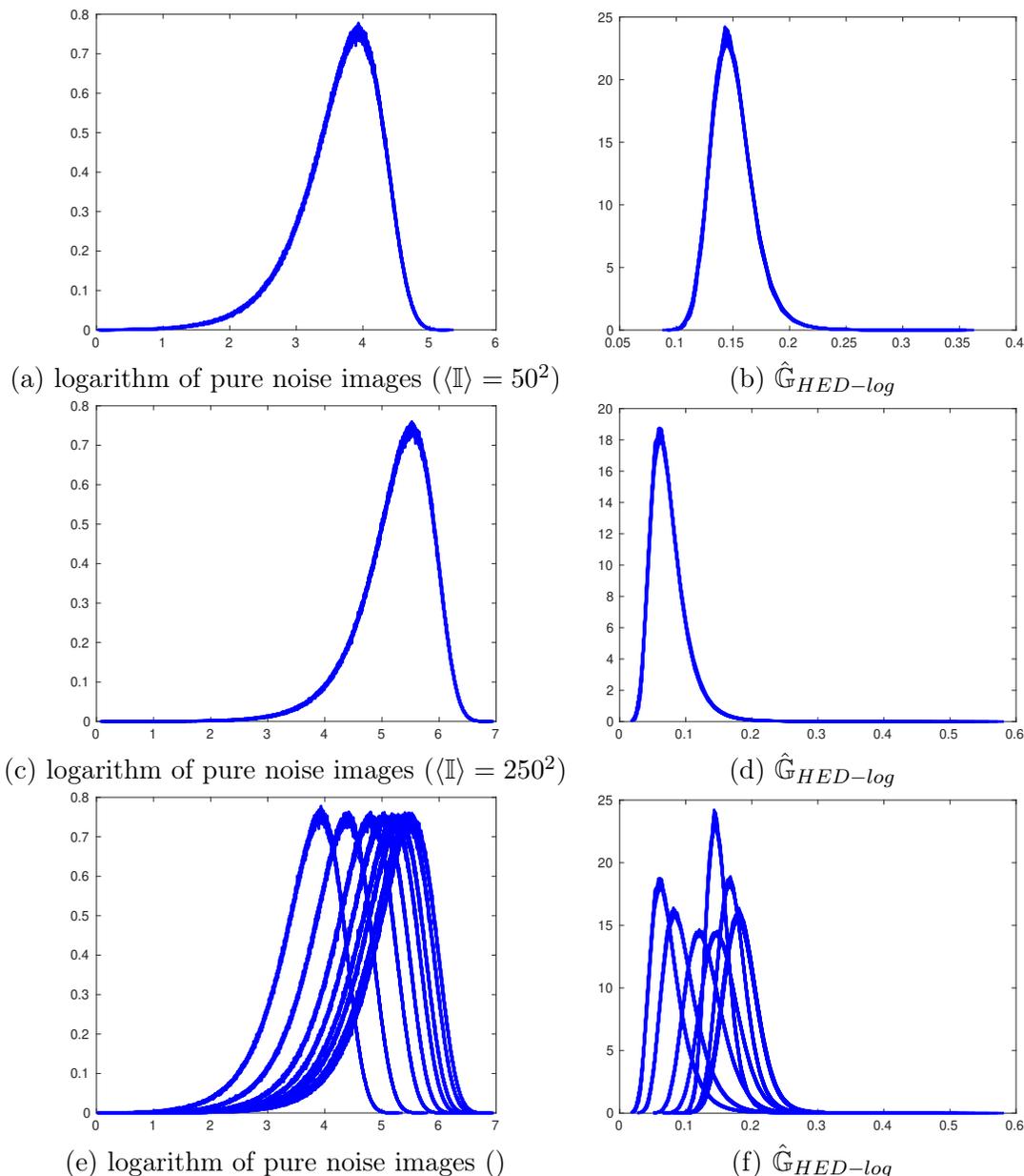


FIGURE 8.2 – The histograms for the logarithm of original pure noise images with different mean intensity values and histograms for the corresponding output of HED-log. (a) histograms for the logarithm of twenty 1-look pure noise images with the underlying mean intensity value 50^2 ; (b) the histograms of the gradient magnitude field produced by HED-log in the twenty images of (a); (c) histograms for the logarithm of twenty 1-look pure noise images with the underlying mean intensity value 250^2 ; (d) the histograms of the gradient magnitude field produced by HED-log in the twenty images of (c); (e) histograms for the logarithm of 160 1-look pure noise images with eight possible underlying mean intensity values, for each mean intensity value, there are 20 random realizations; (f) histograms of the gradient magnitude field produced by HED-log in the 160 images of (e). The size of images are 1024×1024 pixels.

8.3.2 Testing HED and HED-log in synthetic edge images

The other situation we have to analyze is the way of HED and HED-log to process areas with edges. In order to do this, we simulate many 1-look synthetic edge images having only one kind of contrast. One example of the synthetic edge images with amplitude ratio contrast 1.5 can be found in figure 8.3. The size of the synthetic edge images are 512×512 pixels. We address two cases : 1) for images with edges of different contrasts ; 2) for images with edges of the same contrast but with different mean intensity values for the homogeneous areas on both sides of the edges.

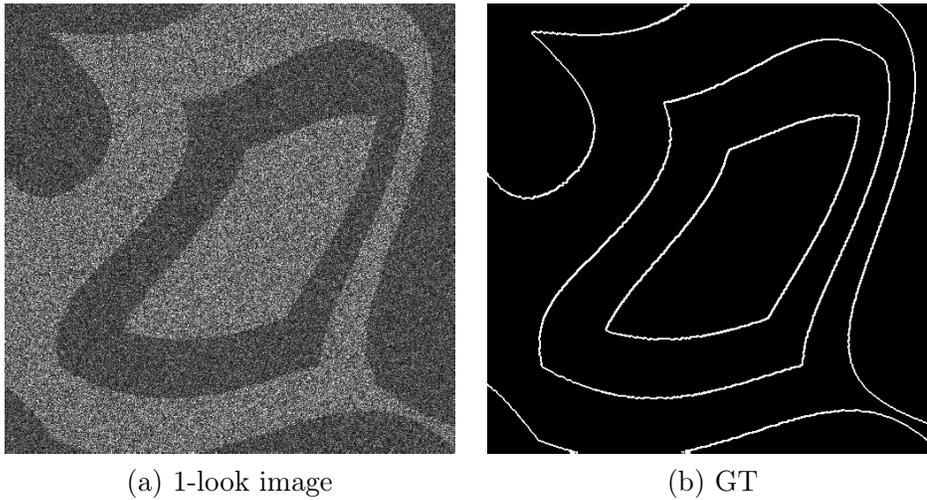


FIGURE 8.3 – A 1-look edge image with ratio contrast 1.5 and the associated ground truth. The size of the image are 512×512 pixels.

8.3.2.1 Synthetic edge images with different amplitude ratio contrasts

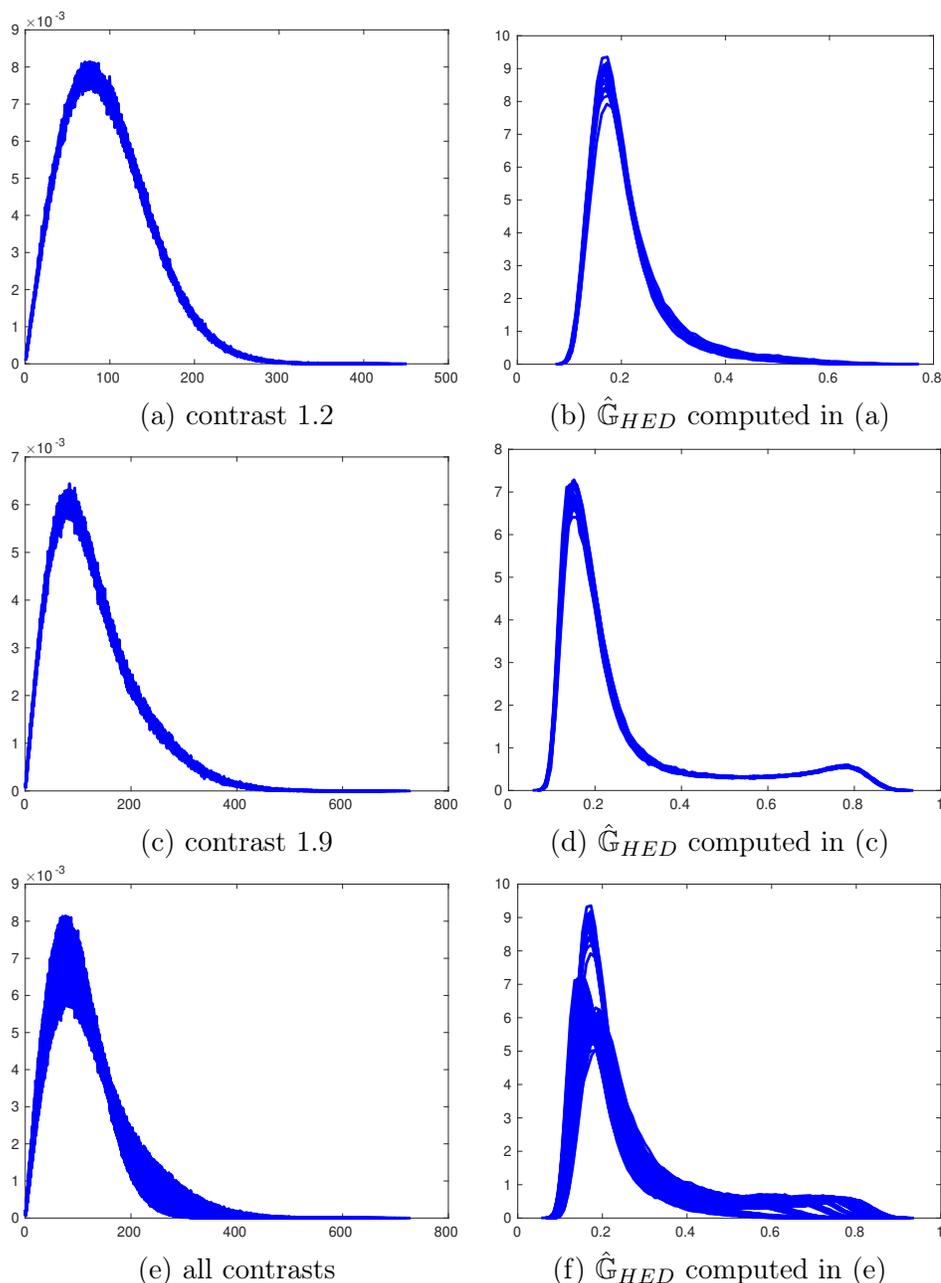


FIGURE 8.4 – The histograms of the 1-look synthetic edge images and the histograms of the output of HED. (a) the histograms for twenty random realizations of 1-look synthetic edge images with contrast 1.2; (b) histograms for the output of HED in the twenty images of (a); (c) the histograms for twenty random realizations of 1-look synthetic edge images with contrast 1.9; (d) the histograms for the output of HED in the twenty images of (c); (e) histograms for 160 1-look synthetic edge images with contrasts varying from 1.2 to 1.9 with step 0.1, for each contrast, there are 20 random realizations; (f) histograms for the output of HED in the 160 images of (e). The size of the images are 512×512 pixels.

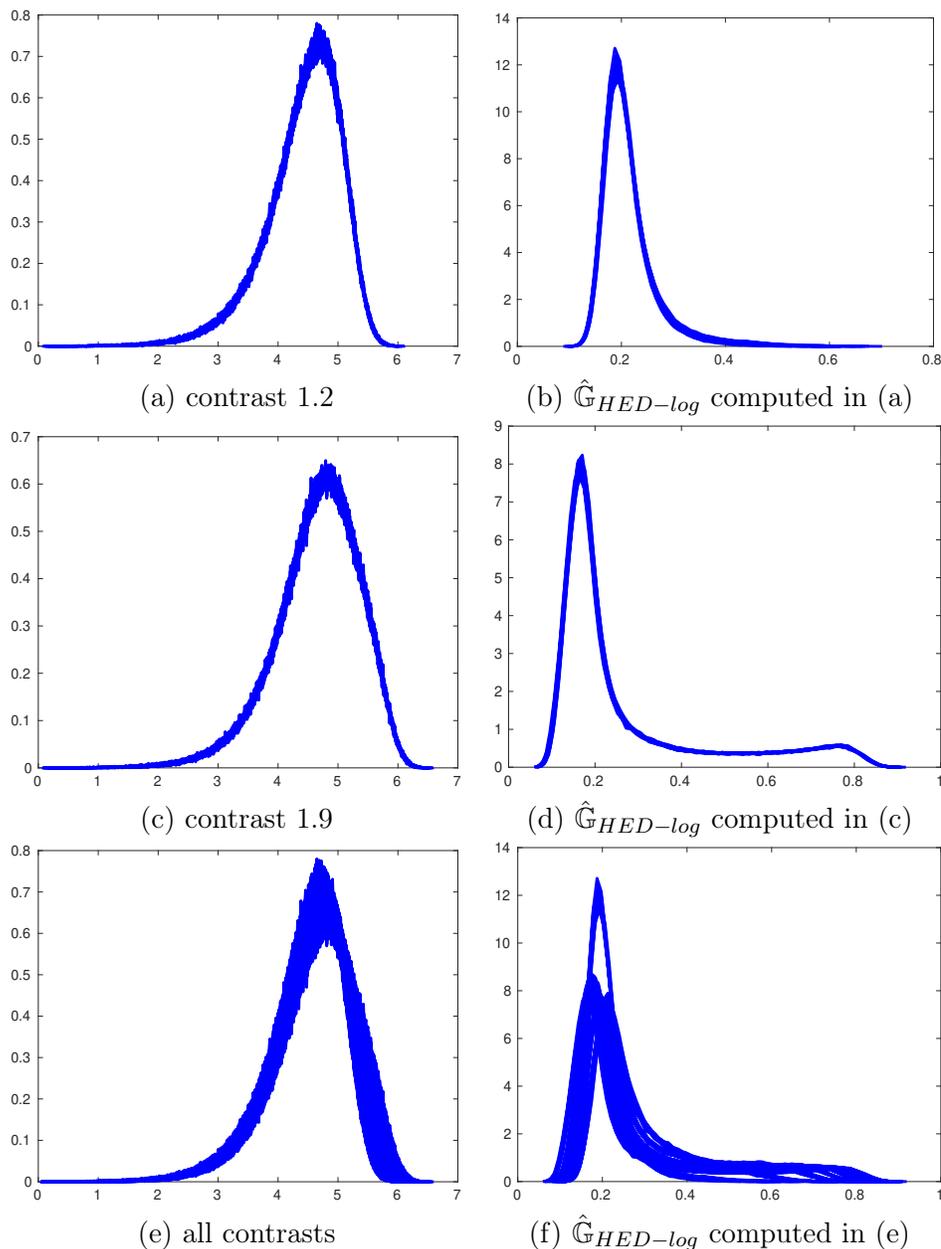


FIGURE 8.5 – The histograms for the logarithm of 1-look synthetic edge images and the histograms for the output of HED-log. (a) histograms for the logarithm of twenty random realizations of 1-look synthetic edge images with contrast 1.2; (b) the histograms for the output of HED-log in the twenty images of (a); (c) histograms for the logarithm of twenty random realizations of 1-look synthetic edge images with contrast 1.9; (d) histograms for the output of HED-log in the twenty images of (c); (e) the histograms for the logarithm of 160 1-look synthetic edge images with contrasts varying from 1.2 to 1.9 with step 0.1, for each contrast, there are 20 random realizations; (f) histograms for the output of HED-log in the 160 images of (e). The size of the images are 512×512 pixels.

We first test HED and HED-log in synthetic edge images with different amplitude ratio contrasts. The contrasts of the edges vary from 1.2 to 1.9 with step 0.1. The mean intensity

value of one homogeneous areas is fixed to be 100^2 , and the mean intensity value of the other homogeneous areas is decided according to the amplitude contrasts, for example for contrast 1.6, the mean intensity value of the homogeneous area in the other side is $(100 \times 1.6)^2$. For edge images with each contrast, there are 20 random realizations.

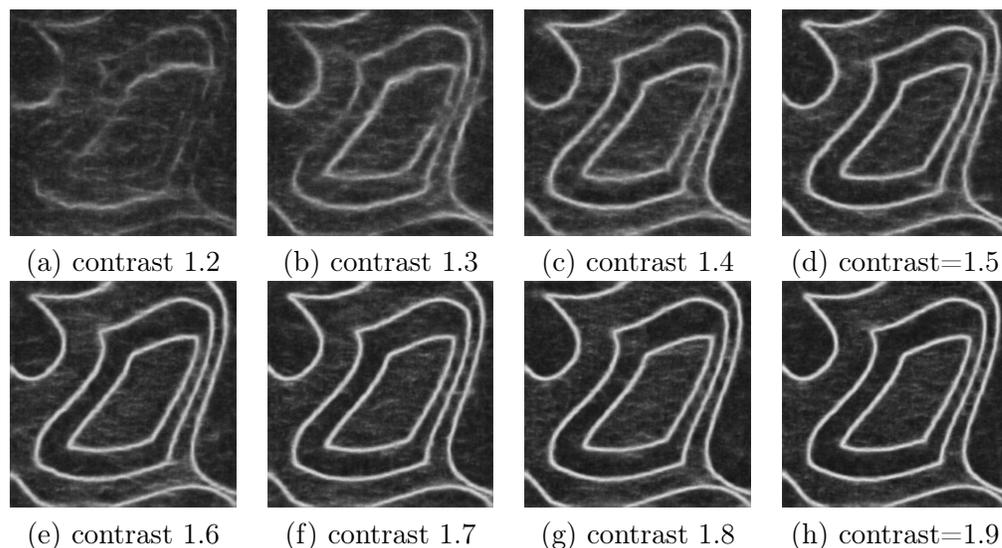


FIGURE 8.6 – The gradient magnitude field computed by HED in eight 1-look synthetic edge images with amplitude ratio contrasts of the edge varying from 1.2 to 1.9 with step 0.1. The size of the images is 512×512 pixels.

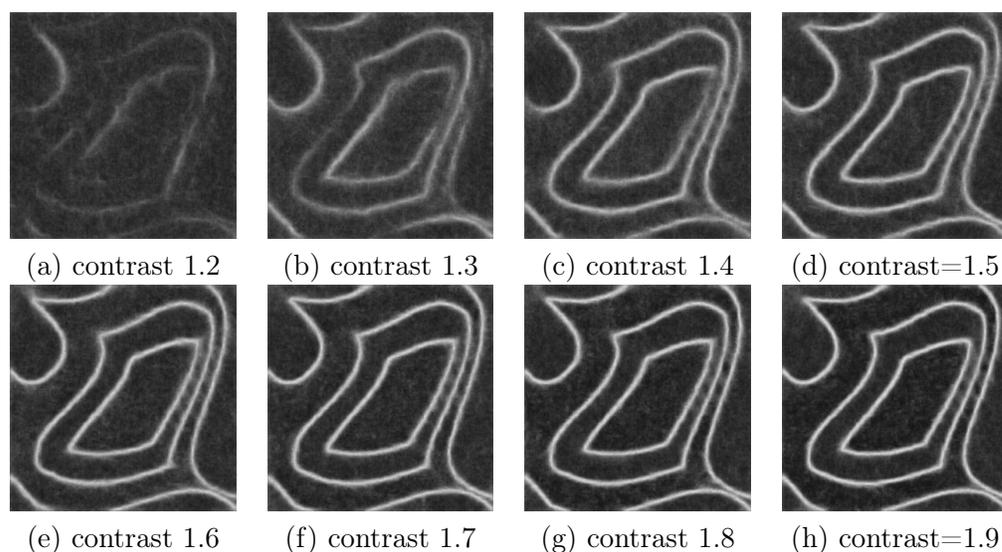


FIGURE 8.7 – The gradient magnitude field computed by HED-log in the logarithm of eight 1-look synthetic edge images with amplitude ratio contrasts of the edge varying from 1.2 to 1.9 with step 0.1. The size of the images is 512×512 pixels.

The histograms of the synthetic edge images and the histograms for the output of HED can be found in figure 8.4. The histograms for the logarithm of synthetic edge images and the histograms for the output of HED-log can be found in figure 8.5. From both figure 8.4

and figure 8.5 we can assume that for samples drawn from the same underlying distribution (the mean intensity values of both homogeneous areas remain unchanged), the output of HED and HED-log follows the same distribution, while when samples drawn from different distributions are fed into CNN layers, the distribution of the output are not guaranteed to be the same.

The gradient magnitude field produced by HED and HED-log in eight 1-look synthetic edge images with contrasts varying from 1.2 to 1.9 can be found in figure 8.6 and figure 8.7. It can be seen from these figures that both HED and HED-log learn to emphasize the edge pixels from the images and learn to suppress the noise pixels. In addition, when the contrasts of the edges become higher, it is easier for both HED and HED-log to distinguish between edge pixels and noise pixels.

8.3.2.2 Synthetic edge images with the same amplitude ratio contrast but with different mean intensity values.

In this part, we try to explore one more situation that HED and HED-log have to learn to process during training, namely images with edges of the same contrast, but with homogeneous areas having different underlying mean intensity values on both sides of the edges. We will see that in this case, both HED and HED-log fail to produce the same output for different mean intensity. This may be explained by the fact that the diversity of distributions is important for the networks to correctly learn what to output. We simulate edge images with contrast 1.4 and six mean intensity values. For each mean intensity value, there are 20 random realizations. One example of the synthetic edge images can be found in figure 8.3. We fix the mean intensity value of one homogeneous area in each image to be one of the six values : 20^2 , 50^2 , 80^2 , 100^2 , 120^2 and 150^2 . The mean intensity value of the other homogeneous areas can be computed according to the contrast of the edges. For example, for edges of contrast 1.4, and the mean intensity value of one homogeneous area is 80^2 , the mean intensity value of the other homogeneous area is $(80 \times 1.4)^2$. For images with a certain contrast and certain mean intensity values, there are 20 random realizations. Therefore, for images with a certain contrast, the total number of images is 120.

The histograms for the 120 1-look synthetic edge images with contrast 1.4 and the histograms for the gradient magnitude field computed by HED can be found in figure 8.8. The histograms for the logarithm of those edge images and the histograms for the gradient magnitude field computed by HED-log in the logarithm of images can be found in figure 8.9. It can be seen from these two figures that even for images with edges of the same contrast, there are many different distributions that HED and HED-log have to learn to process, while the distributions of the output are likely to be different when the input data follows different distributions.

Figure 8.10 and figure 8.11 gives the gradient magnitude field computed by HED and HED-log in six edge images with contrast 1.4 and with different mean intensity values. According to these two figures and the histograms shown in figure 8.8 and figure 8.9, it can be deduced that the performances of HED and HED-log depends on the mean intensity values of the two homogeneous areas on the opposite side of the edges, even when the ratio of the mean intensity values remains unchanged.

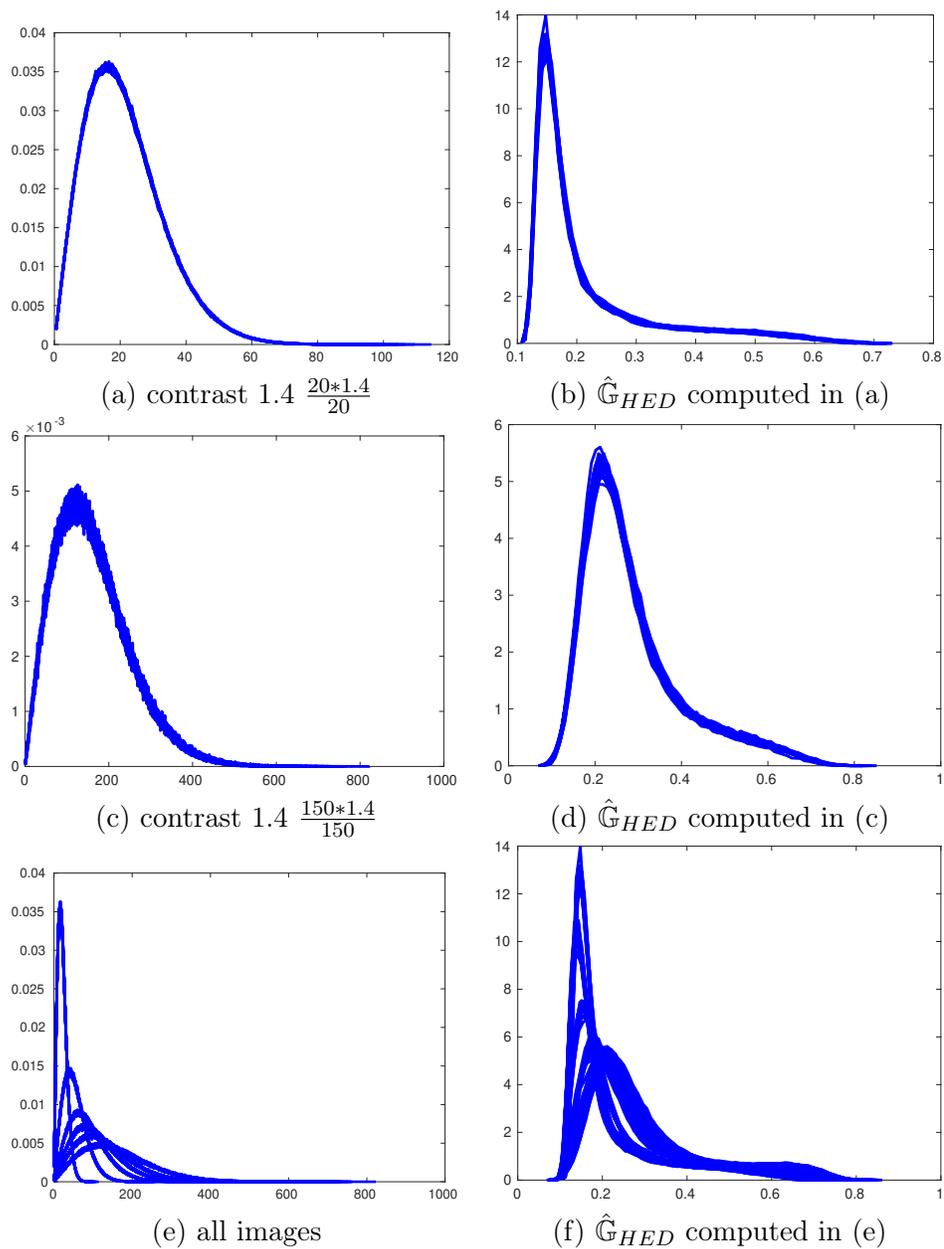


FIGURE 8.8 – The histograms for the 1-look synthetic edge images with amplitude ratio contrast 1.4 and the histograms for the gradient magnitude field computed by HED in those images.

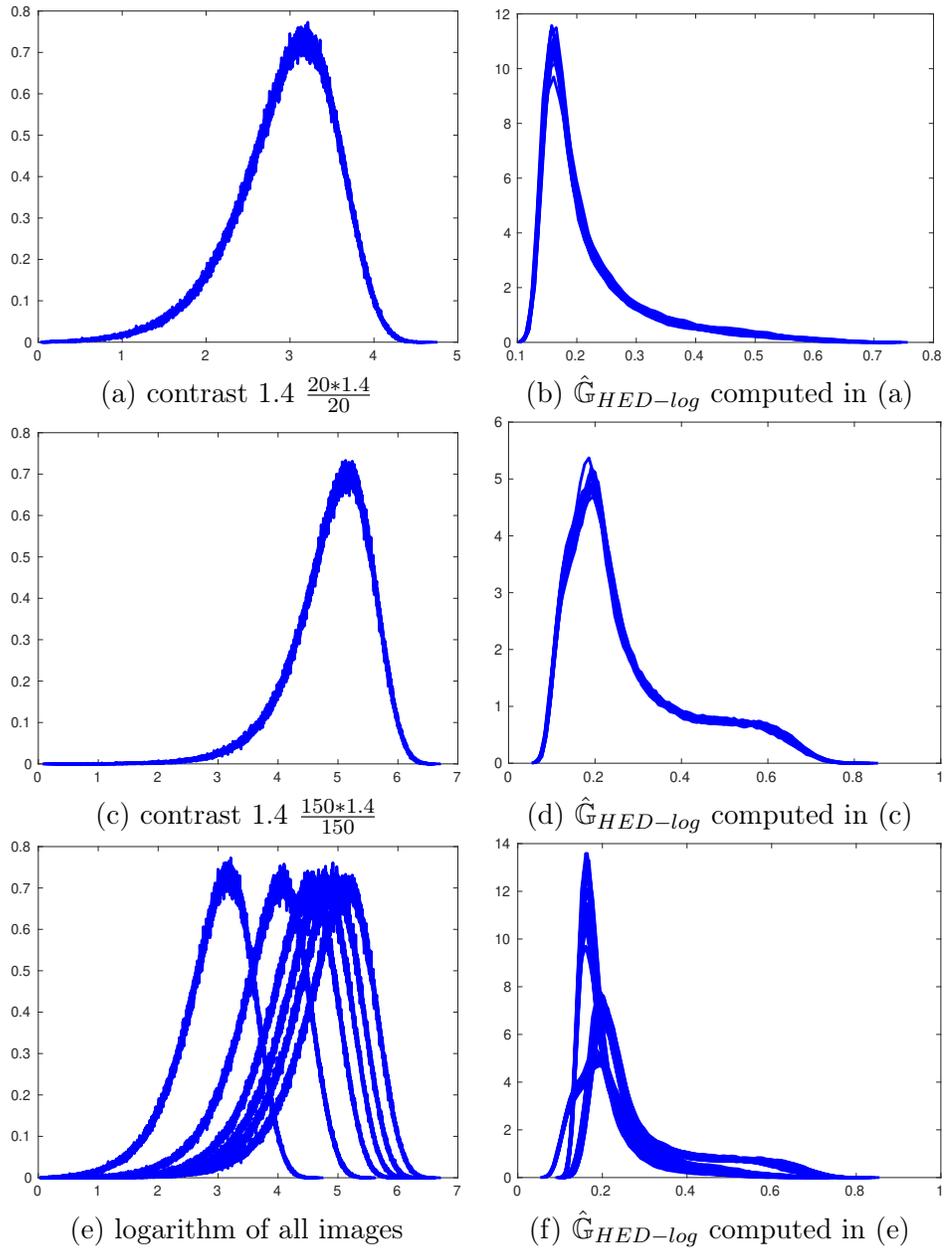


FIGURE 8.9 – The histograms for the logarithm of 1-look synthetic edge images with amplitude ratio contrast 1.4 and the histograms for the gradient magnitude field computed by HED-log in those images.

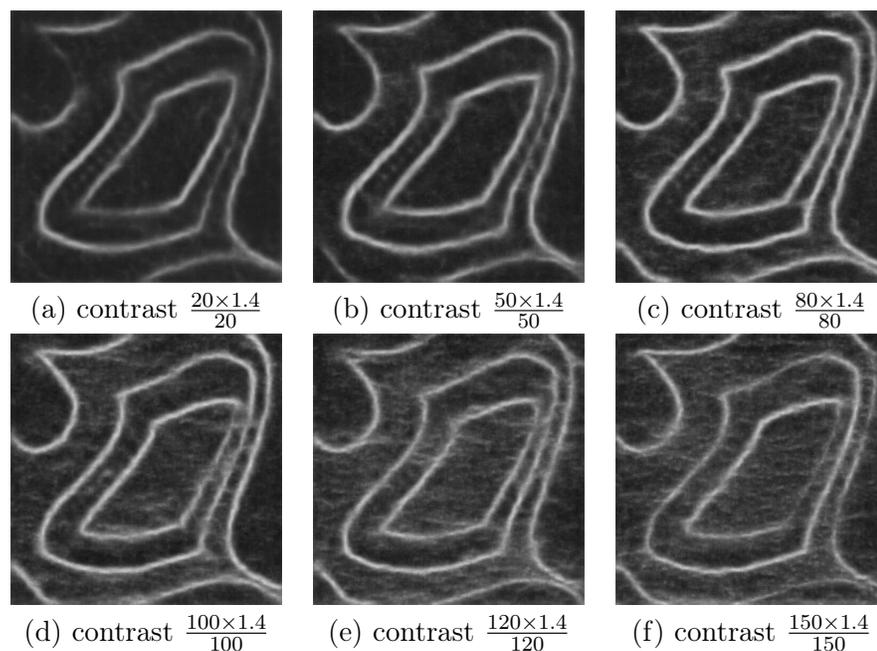


FIGURE 8.10 – The gradient magnitude field computed by HED in six 1-look synthetic edge images with amplitude ratio contrasts of the edge 1.4. The mean intensity values of two homogeneous areas in each image vary from image to image.

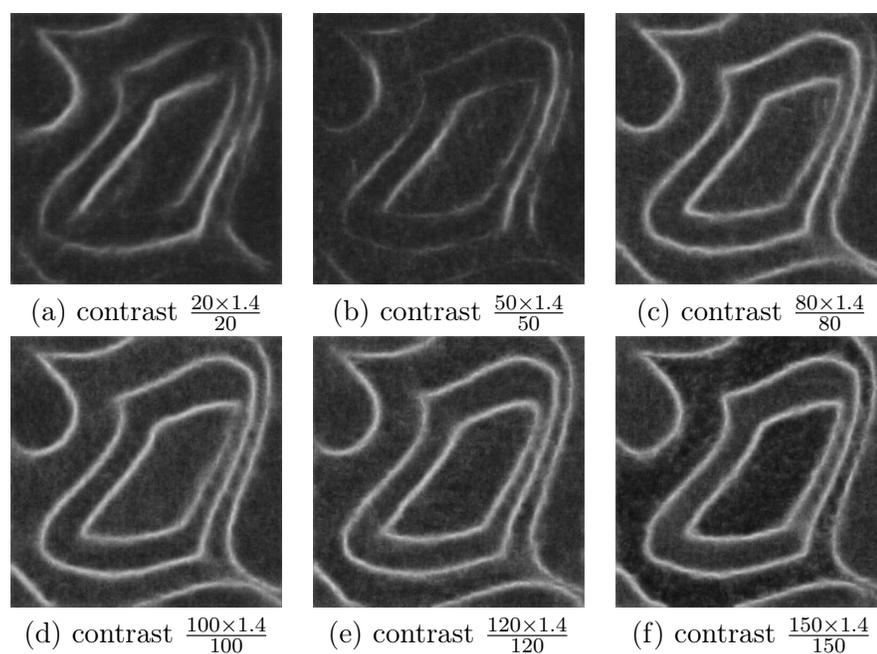


FIGURE 8.11 – The gradient magnitude field computed by HED-log in the logarithm of six 1-look synthetic edge images with amplitude ratio contrasts of the edge 1.4. The mean intensity values of two homogeneous areas in each image vary from image to image.

8.3.2.3 Conclusion

According to the observations in section 8.3.1, section 8.3.2.1 and section 8.3.2.2, What HED (or HED-log) have learned during training can be summarized as follows :

- HED and HED-log learn to ensure that there is no detection of edges in homogeneous areas, the underlying mean intensity values of which are within $[0, 255^2]$ because the maximum pixel value in natural images is 255 ;
- HED and HED-log learn to detect the edges between any two kinds of homogeneous areas, the mean intensity values of them being different, but all belonging to $[0, 255^2]$.
- HED and HED-log do not have a constant false alarm rate for SAR images since they produce different distributions in homogeneous areas with different mean intensity values. Observing that they produce the same distribution when the input data follows the same distribution, we can deduce that the only way to develop a CFAR edge detector is to feed CFAR features into the convolutional layers of HED.

Knowing what HED (or HED-log) has learned from the training dataset, the question is how to enable the models trained using the speckled optical dataset to work well in real SAR images. In order to do this, we need to know the differences between speckled optical images and real SAR images. In the next chapter, we will analyze the difference between speckled optical images and real SAR images, and try to propose a solution to overcome it.

Chapitre 9

GRHED, introducing a hand-crafted layer before the usual CNNs

One crucial hypothesis in this work is that both optical and real SAR images are composed of many homogeneous areas. Under this hypothesis, the main possible gap between speckled optical images and real SAR images lie in the differences in the mean intensity values of homogeneous areas. In this chapter, we will show that some bright homogeneous areas (with mean intensity values larger than 255^2) in real SAR images may not exist in speckled optical dataset (in the following, those bright areas will be called as bright areas or homogeneous areas with high mean intensity values). In order to cope with the gap, we propose a strategy to train the network so that the trained models are robust to the mean intensity values of homogeneous areas.

9.1 The differences between speckled optical images and real SAR images

In order to compare the differences between speckled optical images and real SAR images, we plot the histograms for both of them. We display the histograms of five speckled optical images which are randomly selected from the training dataset in figure 9.1-(a) and the histogram of the entire training dataset in figure 9.1-(b) and compare them with the histograms of two 1-look real SAR images (Lelystad and San Francisco) as shown in figure 9.2. From figure 9.1 and figure 9.2 we can see that the main differences between histograms of speckled optical images and those of real SAR images are the range of pixel values. Though most of pixels in speckled optical and real SAR images lie in a similar range, some pixel values in real SAR images are much larger than those of speckled optical images.

Formula (8.9) is the most usual way to describe the statistics of real SAR images, which gives the way of modelling the homogeneous areas in real SAR images. In addition, most existing gradient computation methods, such as ROA Touzi et al. [1988], ROEWA Fjørtoft et al. [1998] and GR Dellinger et al. [2015], are justified under the hypothesis that real SAR images are composed of piecewise constant areas and that edges exist between two homogeneous areas with different mean intensity values. In this work, we proceed in the same way and assume that both speckled optical images and real SAR images are mixtures of homogeneous areas with many different mean intensity values. The edges are defined as boundaries between any two homogeneous areas.

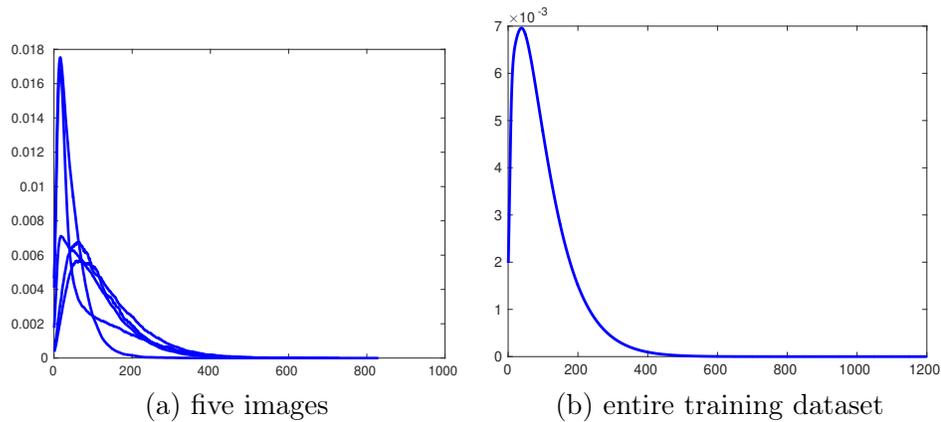


FIGURE 9.1 – (a) histograms of five speckled optical images which are randomly chosen from the training dataset ; (b) the histogram of the entire training dataset. The size of the bin is 1.0.

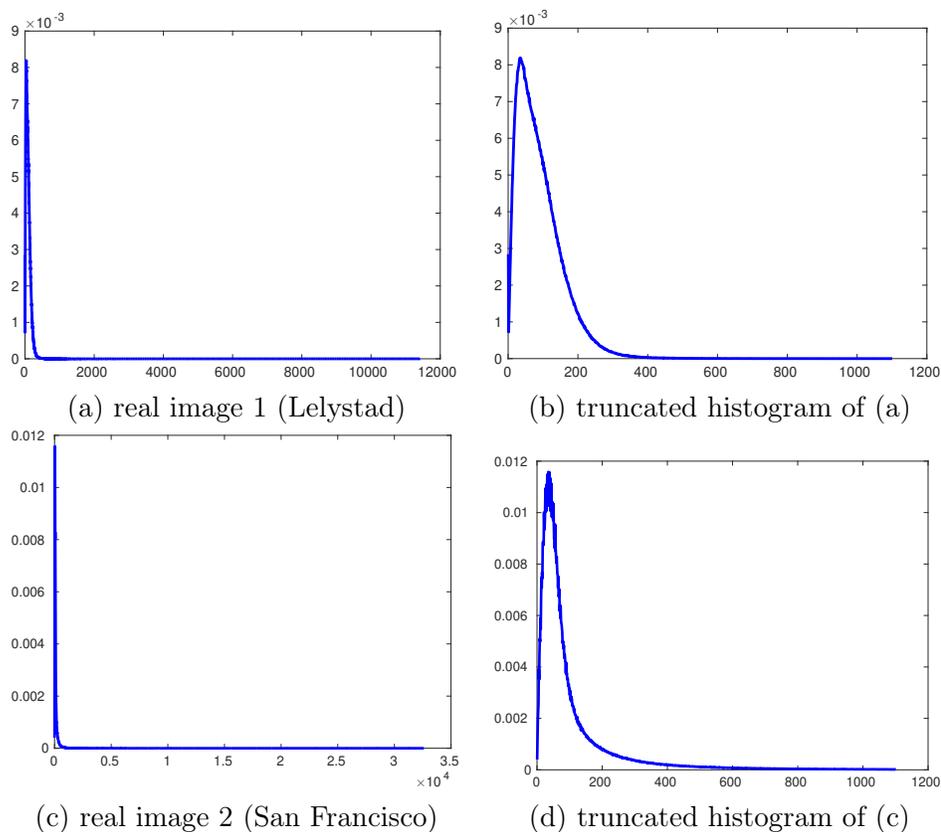


FIGURE 9.2 – Histograms of two 1-look real SAR images and the truncated histogram for each image. The size of the bin is 1.0.

Under the hypothesis that both speckled optical images and real SAR images are composed of piecewise constant areas, we try to evaluate the possible underlying mean intensity values for both speckled optical images and real SAR images. We use windows of size 20×20 pixels and windows of size 50×50 pixels to estimate the underlying mean intensity values

for areas whose size is equal to the size of the windows, in both speckled optical images and real SAR images. The curves of the square root of the mean intensity values (in ascending order) estimated from two randomly selected speckled optical images and two 1-look real SAR images can be found in figure 9.3 (the size of windows are 20×20 pixels) and figure 9.4 (the size of windows are 50×50 pixels). For the estimation in speckled optical images, we use the maximum likelihood estimator to estimate the square root of the mean intensity values, while for real SAR images, we use the mode to estimate the mean intensity values in order to reduce the influence of strong bright points in real SAR images (It should be pointed out that both maximum likelihood estimator and mode estimator are under the hypothesis that samples in the windows follow a Rayleigh distribution). More details about the mode estimator can be found in section 10.1.

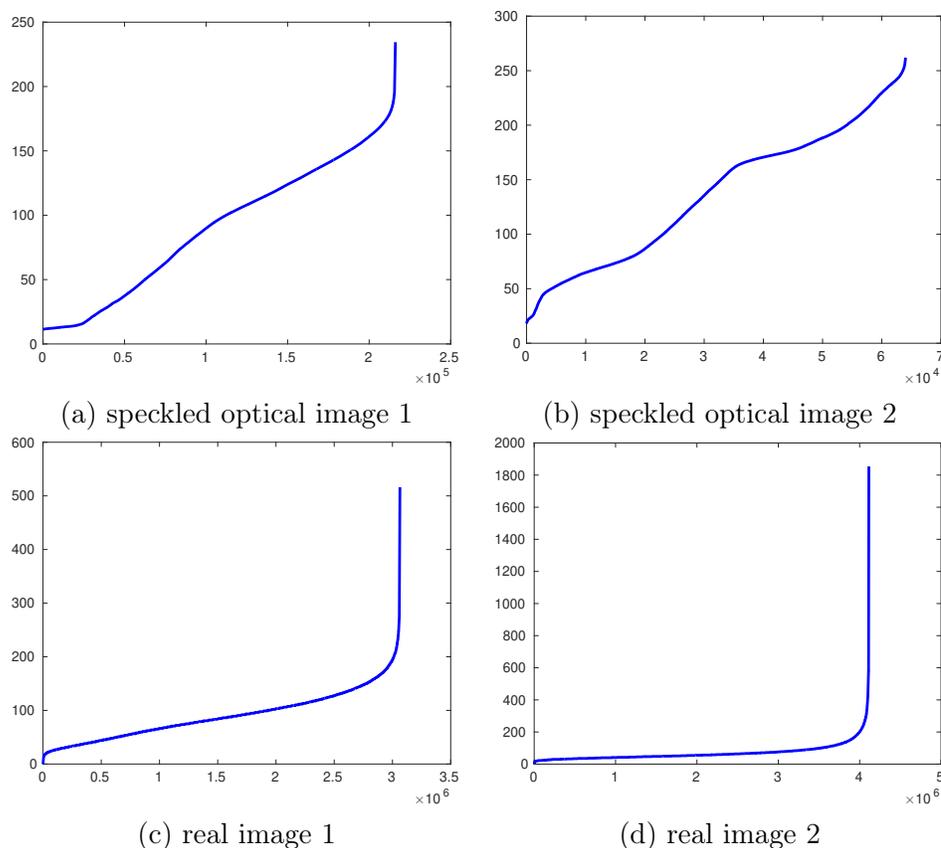


FIGURE 9.3 – The curves for square root of mean intensity values estimated from two speckled optical images and two real SAR images, using windows of size 20×20 pixels. (a) speckled optical image 1; (b) speckled optical image 2; (c) real image 1 (Leystad); (d) real image 2 (San Francisco).

From both figure 9.3 and figure 9.4 we can see that the underlying mean intensity values in homogeneous areas of real SAR images can be much larger than those in speckled optical images (it should be noted that the underlying mean intensity values for all homogeneous areas in the training dataset will not be larger than 255^2 because the maximum possible value in natural images before multiplying the speckle noise is 255). Therefore, the main differences between speckled optical and real SAR images are that some areas with high mean intensity values exist in real SAR images but not in speckled optical images. What

can also be assumed from figure 9.3 and figure 9.4 is that those areas occupy only a small part of the image.

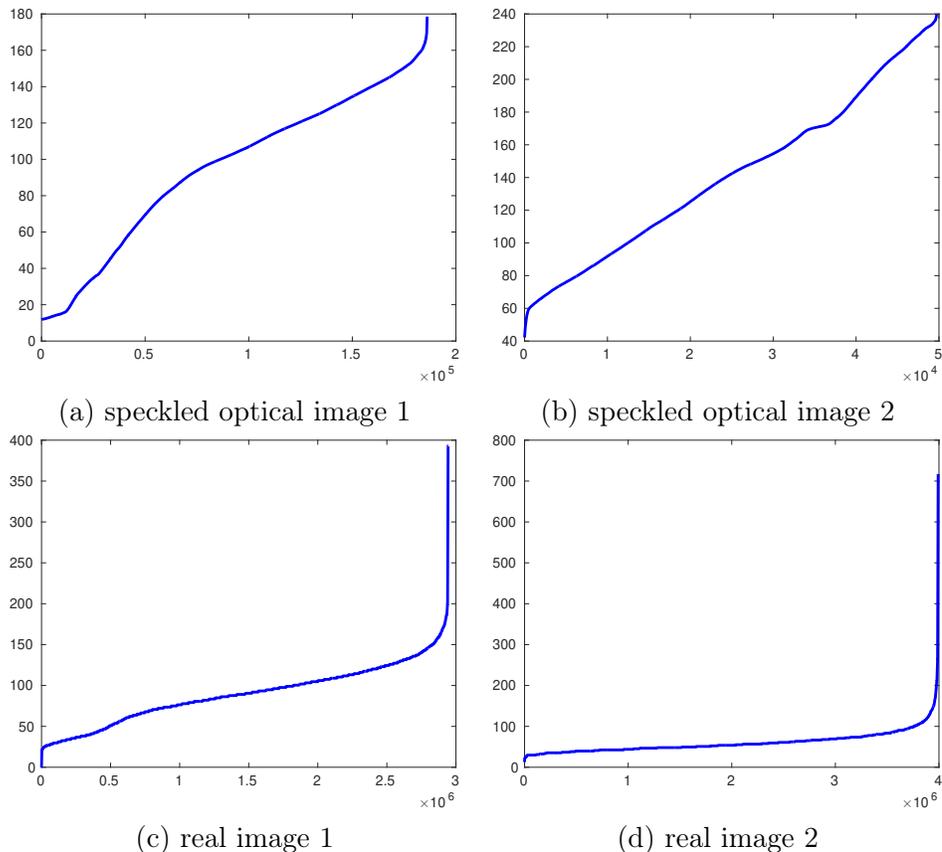


FIGURE 9.4 – The curves for square root of mean intensity values estimated from two speckled optical images and two real SAR images, using windows of size 50×50 pixels. (a) speckled optical image 1; (b) speckled optical image 2; (c) real image 1 (Leystad); (d) real image 2 (San Francisco).

Furthermore, since the maximum possible pixel values in homogeneous areas with higher mean intensity values are larger than those in homogeneous areas with lower mean intensity values, we deduce that those high pixel values in real SAR images are samples in bright homogeneous areas. The main gap between speckled optical images and real SAR images is that there exists some bright areas in real SAR images which do not exist in the training dataset.

9.2 GRHED, training HED on the gradient magnitude field computed by GR

In order to deal with the differences in the range of pixel values between speckled optical images and real SAR images, but also to ease the training of the network, we propose to train HED on the gradient magnitude fields of the training images and to apply the trained network to the gradient magnitude fields of images at test time. The main reason for computing the gradient is that the gradient distribution of speckled optical

images and SAR images are similar for the usual ratio based methods Touzi et al. [1988], Fjørtoft et al. [1998], Dellinger et al. [2015]. The interest of this strategy will be discussed in the following and be demonstrated in the experimental part, chapter 10.

The gradient computation is achieved by the Gradient by Ratio method Dellinger et al. [2015] and the HED network is trained on the gradient magnitude fields computed by GR. We therefore call GRHED the resulting architecture. In the following we will describe the details about the GR and the benefits of computing GR.

9.2.1 Gradient by Ratio (GR)

For a given pixel located at position (x, y) in the image u , the horizontal and vertical gradient components (GR) are defined as

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

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

where $R^h(x, y)$ and $R^v(x, y)$ is the ratio of exponentially weighted average in the opposite side windows of pixel located at (x, y) , along the horizontal and vertical directions. In the horizontal direction, $R^h(x, y)$ can be computed as

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

where

$$m_1^h(x, y) = \sum_{x'=-W}^W \sum_{y'=1}^W u(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}},$$

$$m_2^h(x, y) = \sum_{x'=-W}^W \sum_{y'=-W}^{-1} u(x+x', y+y') \times e^{-\frac{|x'|+|y'|}{\alpha}},$$

and where W is the upper integer part of $\log(10) \times \alpha$. $R^v(x, y)$ can be computed in a similar way.

The magnitude $G_{gr}(x, y)$ and orientation $ang_{gr}(x, y)$ of GR at position (x, y) are defined by

$$G_{gr}(x, y) = \sqrt{G^h(x, y)^2 + G^v(x, y)^2},$$

$$ang_{gr}(x, y) = \text{atan2}(G^v(x, y), G^h(x, y)).$$

9.2.2 Interest of using GR to feed the network

Under the hypothesis that both optical and SAR images are composed of piecewise constant areas, they can be divided into two parts : homogeneous areas and boundaries (boundaries exist between two homogeneous areas). As discussed in section 8.3, another hypothesis in this work is that HED is trained to process samples drawn from many different distributions and the way of HED to process those samples depends on their corresponding distributions. In 1-look SAR images, the amplitude of all homogeneous areas follows a Rayleigh distribution depending on their mean intensity values. For two homogeneous areas across boundaries, the total distribution of them depends on the mean intensity

values of both two homogeneous areas. Computing the ratio based gradient thus ease the training process, reducing the number of situations to learn, since for all homogeneous areas, their gradient magnitude fields follow the same distribution, regardless of their mean intensity values, and the gradient distribution computed over two homogeneous areas across boundaries depends only on the ratio of the mean intensity values. What's more important, the main gap between speckled optical images and real SAR images is that SAR images may contain homogeneous areas with high mean intensity values. Since the gradient distribution will not be influenced by the mean intensity values of any homogeneous areas, feeding the gradient features enables the network trained on speckled optical dataset to work well in real SAR images. Besides, it ensures a CFAR property for the network result since only CFAR features are given as input.

In order to give a better explanation on the benefits of computing GR, we compare the distribution of the data before and after computing the gradient for both homogeneous areas and two homogeneous areas across boundaries. We simulate eight 1-look pure noise images of size 4096×4096 pixels and eight 1-look synthetic images of size 512×512 pixels with amplitude ratio contrast 1.5. The square root of the mean intensity values of the homogeneous images (proportional to an amplitude value) are 60, 90, 120, 150, 180, 210, 240 and 270. For two homogeneous areas across boundaries, the amplitude ratio is chosen as 1.5 for mean values (for the smallest value along the edge) of 20, 50, 70, 90, 110, 130, 150, 200. One example of the synthetic edge image can be found in figure 8.3-(a).

The histograms of the eight 1-look pure noise images and the histograms of their gradient magnitude fields computed by GR with weighting parameter $\alpha = 4$ can be found in figure 9.5. The histograms of eight 1-look synthetic edge images as well as the histograms of their gradient magnitude fields can be found in figure 9.6. From figure 9.5 and figure 9.6 we can deduce that the number of distributions that has to be learned by HED when trained on the gradient magnitude fields is much smaller than that of a direct training on the images. Since the amount of data is unchanged, the amount of training data for each distribution is therefore increased.

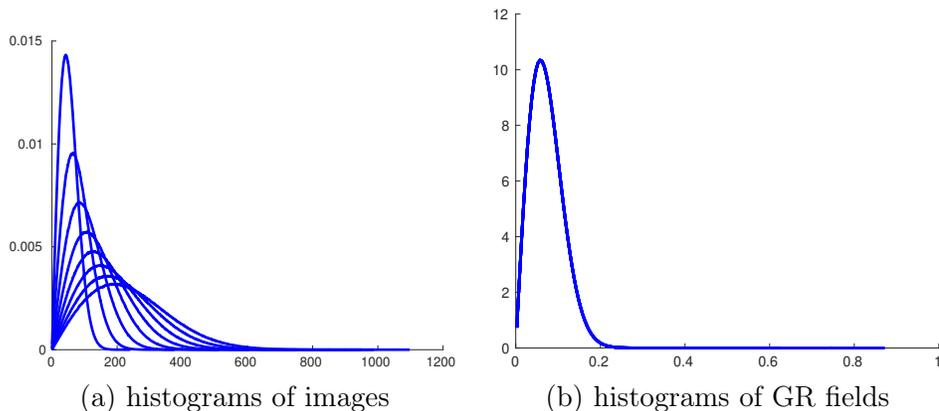


FIGURE 9.5 – Histograms of the eight 1-look pure noise images and histograms of their gradient magnitude fields computed by GR with $\alpha = 4$.

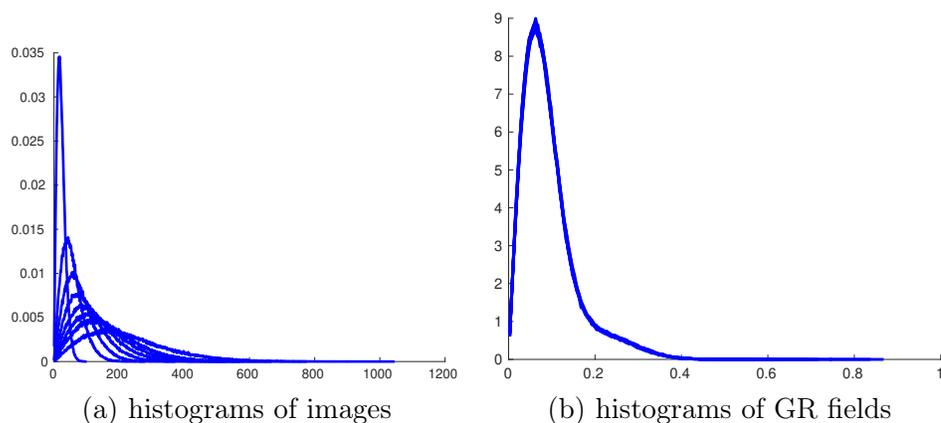


FIGURE 9.6 – Histograms of the eight 1-look synthetic edge images and histograms of their gradient magnitude fields computed by GR with $\alpha = 4$.

A further illustration of the interest to compute GR can be done by comparing the histograms of GR fields in both speckled optical images and real SAR images. The histograms of the gradient magnitude fields computed by GR in 5 randomly selected speckled optical images and the histograms of the GR field computed in the entire training dataset can be found in figure 9.7. The histograms of the gradient magnitude fields computed by GR in two 1-look real SAR images can be found in figure 9.8. The weighting parameter α is set to 4.0. The size of the bin is 0.01. From figure 9.7 and figure 9.8 we can see that even though the range of pixel values in the GR fields of those two 1-look real SAR images may be larger than that of some speckled optical images, nearly all pixel values in the GR fields of real SAR images are within the range of GR field computed in the entire training dataset (The maximum value in the GR field of the TerraSAR-X image (San Francisco) is 3.7, the maximum pixel value in the GR field of the Sentinel-1 image (Leystad) is 5.66, and the maximum pixel value in the GR field of the entire training dataset is 5.36). Compared to the huge differences in the range of pixel values of images shown in figure 9.1 and figure 9.2, this observation illustrates the benefits of computing GR.

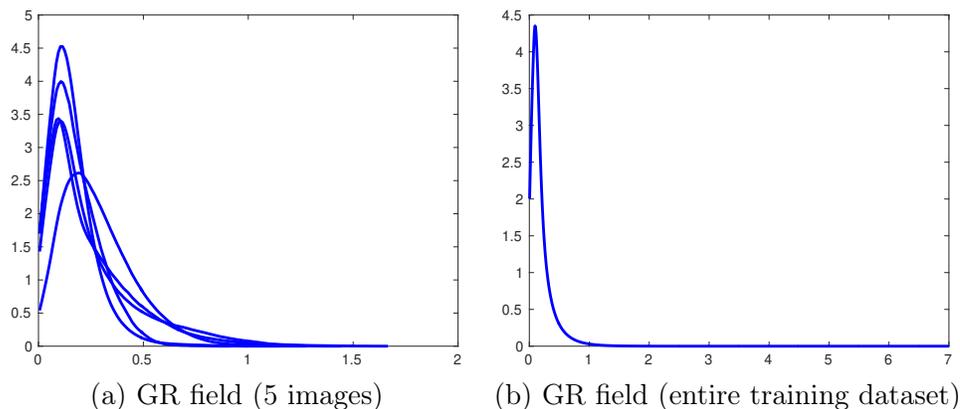


FIGURE 9.7 – (a) histograms of the gradient magnitude fields computed by GR in five speckled optical images which are randomly chosen from the training dataset; (b) the histogram of the gradient magnitude field computed by GR in the entire training dataset. The weighting parameter α is set to 4.0. The size of the bin is 0.01. The maximum pixel value in the GR field of the entire training dataset is 5.36. The tail of the histogram in (b) goes up to 7.0 because we set the maximum value to be 7.0 when we plot the histogram.

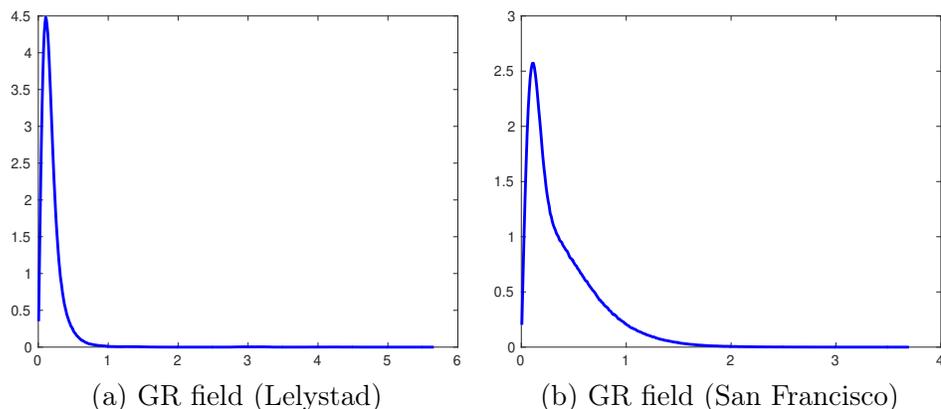


FIGURE 9.8 – Histograms of the gradient magnitude fields computed by GR in two 1-look real SAR images. The weighting parameter α is set to 4.0. The size of the bin is 0.01. The maximum pixel value in the GR field of the Sentinel-1 image (Lelystad) is 5.66 and the maximum pixel value in the GR field of the TerraSAR-X image (San Francisco) is 3.7.

9.2.3 Benefits of GRHED

Since the gradient distribution of GR is the same for homogeneous areas with all possible mean intensity values, and the gradient distribution in two homogeneous areas across boundaries is influenced by the ratio and not by their mean intensity values, we assume that the possible distributions in the gradient feature space of real SAR images are included in those of the training dataset. Therefore, we propose to train HED on the gradient magnitude field of GR. This can be seen as the addition of a hand-crafted layer before the usual HED layers. The resulting GRHED has the following advantages :

- the distribution of feature map values obtained by GR depends only on the ratio of the mean intensity values on the opposite side windows of each pixel (ratio

being 1 for pixels located in homogeneous areas), therefore, homogeneous areas with high mean intensity values in real SAR images will not impair the performances of GRHED;

- constant false alarm rate (CFAR) is ensured for SAR images because of the ratio operation;
 - by using multiple weighting parameter α values in GR Dellinger et al. [2015] we can combine diverse informations from the image by concatenating together gradient magnitude fields produced by GR with different α values (the input of HED will have multiple channels with each channel being the gradient magnitude field computed by GR with a certain value of α). It has been studied in Liu et al. [2020] that GR with different α values can capture complementary informations.
-

Chapitre 10

Experiments

In this chapter we study the performances of the method we propose, GRHED, using several 1-look synthetic edge images, two hundred 1-look speckled optical images in BSDS500-speckled, one 1-look realistically simulated SAR image and two 1-look real SAR images. In order to show the efficiency of GRHED, we compare it with the original HED algorithm and to the result of training HED on the logarithm of images, as defined below, an algorithm that we will call HED-log. Observe that for GRHED, HED and HED-log, the convolutional layers that need to be trained are the same, the difference between these method being the input of the network : gradient magnitude fields of images for GRHED, plain images for HED and logarithm of images for HED-log. To take into account the distribution of the training set, the SAR data are first normalized to have a similar distribution as explained in section 10.1. The training strategy for those convolutional layers is as follows : we use Adam optimizer to train the network from scratch on the speckled optical dataset BSDS500-speckled. The number of iterations for training is 10000 and the learning rate is 0.001. The size of the batch is 10. The preprocessing step before feeding the data into the learnable layers is global mean subtraction, as done in VGG. Since the outputs of all methods are probability edge maps, they should be processed further to obtain the binary edge maps. In order to obtain the binary edge map, we use the same Non-maxima Suppression procedure as the one in Structured Edge Dollar and Zitnick [2015] and use a threshold to discard pixels with low values in the probability edge map.

The following methods will be used in the comparison of the next sections :

- GR with $\alpha = 4$ (this choice of $\alpha = 4$ being adapted to GR in 1-look situations, as discussed in Liu et al. [2020]). For the GR magnitude field, we use the same postprocessing steps as for the magnitude field produced by HED, HED-log and GRHED ;
 - HED : HED is trained on the original speckled images and tested on normalized images (see 10.1) ;
 - HED-log : HED is trained on the logarithm of the images in BSDS500-speckled, and is applied to the logarithm of testing images after normalization (see 10.1) ;
 - GRHED with multiple α values, $\alpha = 2, 3, 4, 5$: HED is trained on the gradient feature maps which are obtained by concatenating the gradient magnitude fields produced by GR with different α values.
-

10.1 Normalization of the SAR data

As already mentioned, speckled optical data have a much narrower dynamic range than real SAR images. This is a problem at test time if we want the method to be efficient on real SAR images. We therefore need to define a strategy to normalize images at test time, ensuring a relatively stable dynamic range. In order to do so, a given a test image (potentially a real SAR image), we divide its amplitude values by the square root of their mean intensity value, before multiplying them by the square root of the mean intensity value of the optical (training) dataset. Let $\langle \mathbb{I}_{opt} \rangle$ be the mean intensity of the global training set and $\langle \mathbb{I}_{SAR} \rangle$ the mean intensity value of the SAR data, the normalization formula of the SAR amplitude value u is the following :

$$\hat{u} = \frac{u \cdot \sqrt{\langle \mathbb{I}_{opt} \rangle}}{\sqrt{\langle \mathbb{I}_{SAR} \rangle}} \quad (10.1)$$

It is easy to check that the the mean value of \hat{u} in intensity is equal to $\langle \mathbb{I}_{opt} \rangle$ and thus corresponds to the training distribution.

We use the maximum likelihood estimator to estimate $\langle \mathbb{I}_{opt} \rangle$. To avoid being too much influenced by the strong backscattered values in real SAR images (especially for urban areas), we propose to use a robust estimate of $\langle \mathbb{I}_{SAR} \rangle$, supposing a global Rayleigh distribution of the scene. Under this hypothesis it can easily be shown that the link between the mode of the distribution t_{mode} and the mean intensity is the following :

$$\langle \mathbb{I}_{SAR} \rangle = 2t_{mode}^2.$$

In the following, for each test image, t_{mode} is computed to estimate $\langle \mathbb{I}_{SAR} \rangle$ and the data are normalized using equation (10.1).

10.2 Comparison in 1-look synthetic edge images

In order to give a fair comparison of different methods, we compare GR with $\alpha = 4$, HED, HED-log and GRHED with multiple α values ($\alpha = 2, 3, 4, 5$), in terms of ROC curves and F1-score curves in 1-look synthetic edge images with different ratio contrasts (for GR we use the same postprocessing steps as the magnitude field produced by HED, HED-log and GRHED).

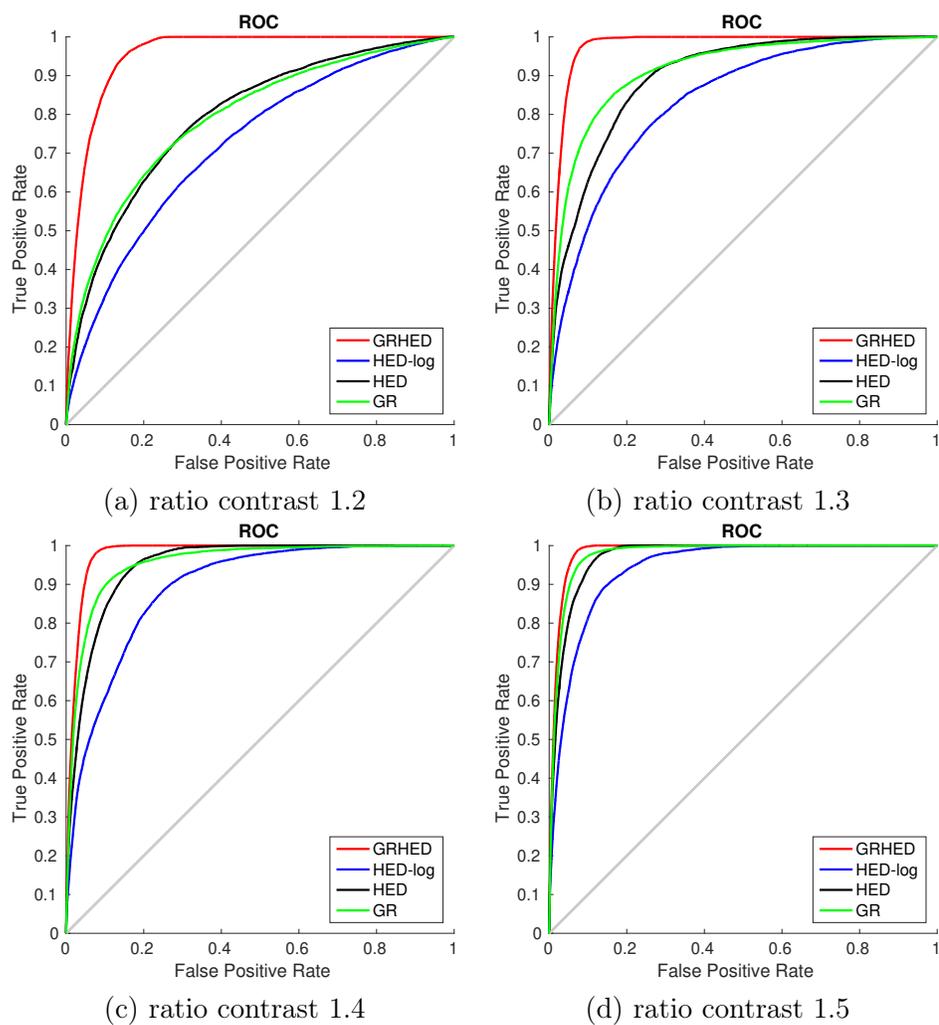


FIGURE 10.1 – ROC curves computed in the magnitude field produced by GR, HED, HED-log and GRHED in 1-look simulated images with contrast 1.2, 1.3, 1.4 and 1.5. The size of the images are 512×512 pixels.

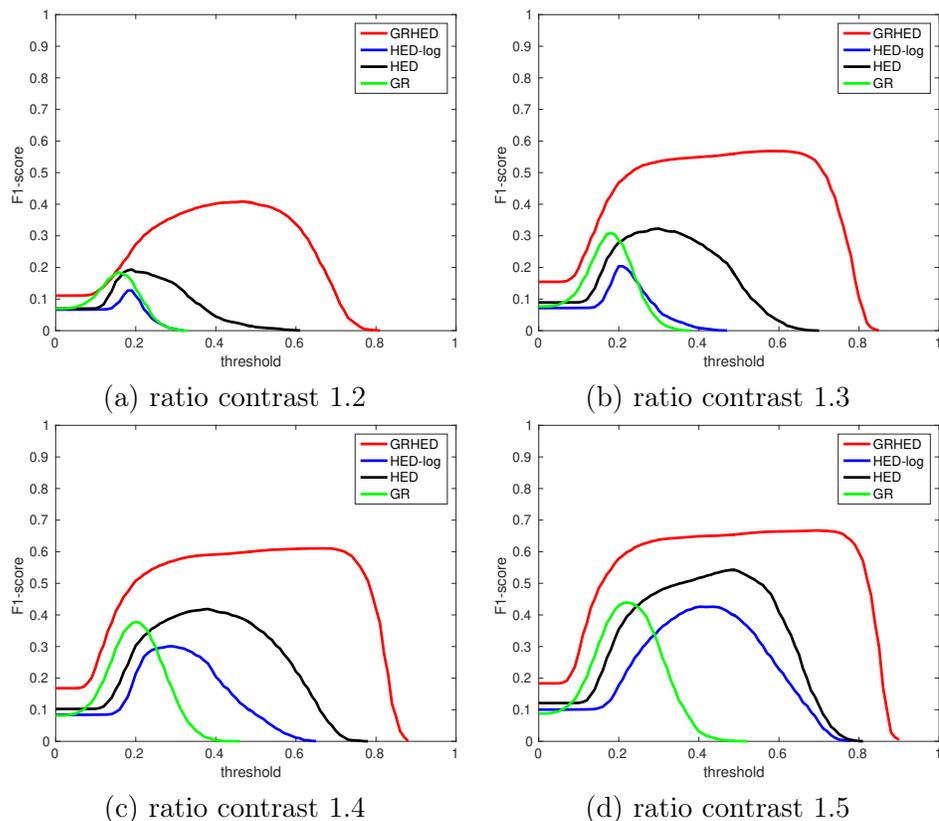


FIGURE 10.2 – F1-score curves obtained by GR, HED, HED-log and GRHED in synthetic edge images with ratio contrast 1.2, 1.3, 1.4 and 1.5. The F1-score curves are obtained by varying the threshold from 0.0 to 0.9 with step 0.01.

One example of the simulated 1-look edge images with contrast 1.5 and the corresponding ground truth can be found in figure 8.3. The ROC curves computed in the magnitude fields produced by GR, HED, HED-log and GRHED in 1-look simulated images with ratio contrast 1.2, 1.3, 1.4 and 1.5 can be found in figure 10.1. It can be seen from figure 10.1 that GRHED shows the best performances in all situations, especially in 1-look and low contrast situation. The performances of GR and HED appears comparable in terms of ROC curves, and both are better than HED-log.

In order to give a more clear comparison of different methods, we apply the same Non-maxima suppression step for all methods and vary the threshold from 0.0 to 0.9 with step 0.01 for each method. The F1-score is computed for each threshold and the corresponding F1-score curves for each method in the 4 simulated edge images can be found in figure 10.2. From figure 10.2 we can see that the F1-score curves of GRHED are above the F1-score curves of all the other methods. In addition, the large flat areas of F1-score curves of GRHED show that the performances of GRHED are insensitive to the chosen threshold. In addition, it should also be pointed out that although the best F1-scores that can be obtained by GR and HED seem comparable, the performances of HED are less sensitive to the choice of threshold.

The optimal edge maps obtained by GR, HED, HED-log and GRHED in those simulated edges with contrast 1.2 and 1.4 using the threshold which gives the best F1-score can be found in figure 10.3 and figure 10.4. From figure 10.3 and figure 10.4 we can see

that the ability of GRHED to preserve true edge pixels and suppress noise pixels are much more powerful than all the other methods. GRHED detects the most true edge pixels while detecting the least number of false detections. It should be noted that although these edge maps are obtained using the optimal threshold, the edge maps obtained by HED and GRHED will remain comparable to their optimal ones with the threshold in a relatively large range as can be deduced from figure 10.2, especially for GRHED.

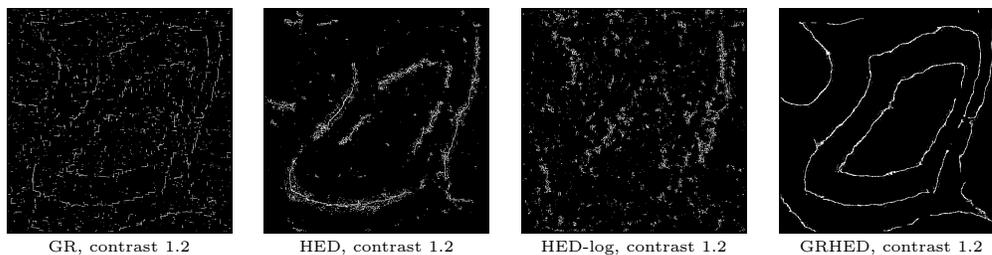


FIGURE 10.3 – Optimal edge maps obtained by GR, HED, HED-log and GRHED in 1-look simulated edge images with contrast 1.2. For each method, the chosen threshold gives the best F1-score in this image. The size of the images are 512×512 pixels.

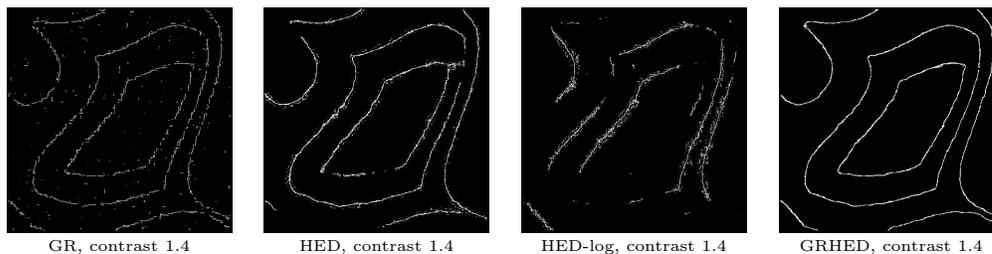


FIGURE 10.4 – Optimal edge maps obtained by GR, HED, HED-log and GRHED in 1-look simulated edge images with contrast 1.4. For each method, the chosen threshold gives the best F1-score in this image. The size of the images are 512×512 pixels.

10.3 Comparison of different algorithms on the speckled optical images in BSDS500-speckled

In order to give a more comprehensive comparison in more general situations between different methods, we compare GR, HED, HED-log and GRHED in the two hundred 1-look speckled optical images in BSDS500-speckled. For GR, we use $\alpha = 2$ and $\alpha = 4$ to show its performance. For GRHED, GRHED with a single α value and GRHED combining multiple α values are all provided to give a clearer demonstration of the efficiency for choosing the combination of α values. Three criterions are used to compare different algorithms : ODS F1 (fixed contour threshold for 200 images), OIS F1 (best threshold for each image), and average precision (AP). The quantitative comparison can be found in table 10.1. From table 10.1 we can see that compared to GR, the CNN-based methods give much higher values in all three criterions. In addition, GRHED gives at least comparable or even better performances than HED and HED-log in the 200 1-look images, especially in the case of combining multiple α values of GR. It should also be noted that, GRHED combining multiple α values give better performances than GRHED using a single α value, which is

probably due to the richer information which is provided to HED convolutional layers.

TABLE 10.1 – The performances of different methods in the 200 speckled optical images in BSDS500-speckled.

methods	ODS (F1)	OIS (F1)	AP
GR ($\alpha = 2$)	0.5658	0.5852	0.5094
GR ($\alpha = 4$)	0.5894	0.6151	0.5286
HED	0.6461	0.6671	0.6981
HED-log	0.6258	0.6466	0.6838
GRHED ($\alpha = 1$)	0.6427	0.6523	0.6949
GRHED ($\alpha = 2$)	0.6603	0.6762	0.7208
GRHED ($\alpha = 3$)	0.6570	0.6783	0.7018
GRHED ($\alpha = 4$)	0.6552	0.6729	0.7050
GRHED ($\alpha = 5$)	0.6492	0.6687	0.6897
GRHED ($\alpha = 6$)	0.6463	0.6660	0.6893
GRHED ($\alpha = 2, 3, 4, 5$)	0.6643	0.6826	0.7109
GRHED ($\alpha = 1, 2, 3, 4, 5, 6$)	0.6643	0.6832	0.7070

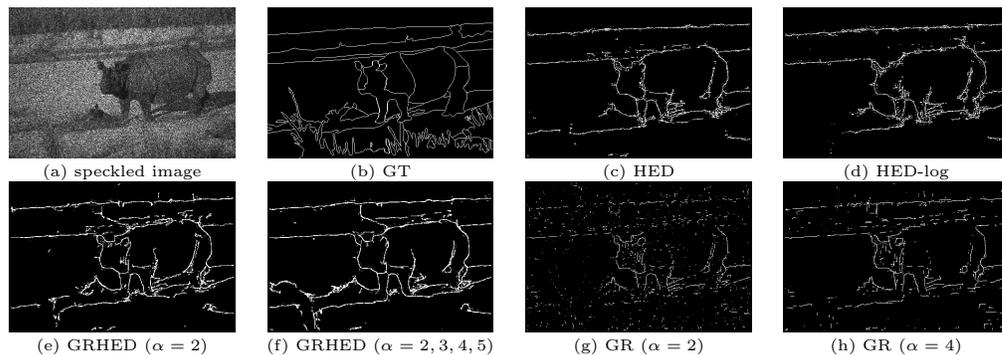


FIGURE 10.5 – Comparison of different edge detectors on a 1-look speckled optical images.

The edge maps obtained by GR ($\alpha = 2$), GR ($\alpha = 4$), HED, HED-log, GRHED ($\alpha = 2$) and GRHED ($\alpha = 2, 3, 4, 5$) on one speckled optical image can be found in figure 10.5. For each method, the threshold is chosen to be the one corresponding to the ODS F1, which gives the best results in the 200 images. Specifically, the threshold used for each method can be found in table 10.2. It should be noted that using the threshold corresponding to the ODS F1, GRHED ($\alpha = 2, 3, 4, 5$) will obtain near-optimal edge maps in the simulated edge images as shown in figure 10.2. The ground truth of this images is the one provided by 1 labeler (there are usually five labelers for each image in BSDS500). From figure 10.5 we can see that GRHED detects a bit more true edge pixels than HED, and both of them preserve more true edge pixels while detecting less false detections. What can also be observed from the figure is that a direct processing of the GR magnitude field can not obtain satisfying

results. Using HED as a postprocessing of GR field, the GRHED is able to strengthen the true edge pixels and suppress the false detections.

TABLE 10.2 – The threshold corresponding to the ODS F1-score for each method.

methods	HED	HED-log	GRHED ($\alpha = 2$)	GRHED ($\alpha = 2, 3, 4, 5$)	GR ($\alpha = 2$)	GR ($\alpha = 4$)
threshold (ODS F1)	0.5666	0.5306	0.5686	0.5516	0.3137	0.2745

10.4 Comparison of different methods in 1-look SAR images

Though the efficiency of GRHED has been demonstrated in both simulated edge images and speckled optical images, demonstrating its ability to detect edges in SAR images is the most important point in practice.

10.4.1 Setting thresholds according to a given probability of false alarms

For the gradient computation method dedicated to SAR images, it is possible to set the threshold according to a chosen probability of false alarms (pfa) only if the method has a constant false alarm Rate (CFAR). Therefore, it is important to study whether HED, HED-log and GRHED have CFAR for SAR images. The CFAR property of these methods can be checked experimentally by plotting the histograms of their gradient magnitude fields computed in pure noise images (homogeneous areas with different mean intensity values). A method is considered to have CFAR if the histograms of its gradient magnitude fields computed in all images overlap well. We plot the histograms of the magnitude fields produced by GR, HED, HED-log and GRHED in 160 1-look pure noise images of size 1024×1024 pixels in figure 10.6. The mean intensity values of these pure noise images have eight possibilities, 50^2 , 80^2 , 120^2 , 150^2 , 180^2 , 200^2 , 230^2 and 250^2 . For pure noise images with a certain mean intensity value, there are 20 random realizations. From figure 10.6 we can see that GR and GRHED both hold CFAR, but HED and HED-log do not.

TABLE 10.3 – The threshold corresponding to a given pfa for each method.

pfa	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
threshold (GR)	0.18	0.22	0.26	0.29	0.31
threshold (HED)	0.3	0.38	0.48	0.57	0.63
threshold (HED-log)	0.23	0.27	0.3	0.34	0.39
threshold (GRHED)	0.17	0.26	0.37	0.45	0.52

Even though HED and HED-log do not have CFAR for SAR images, we still assume they have CFAR since we desire to set the threshold from a given pfa. The threshold corresponding to a given probability of false alarms for GR, HED, HED-log and GRHED is estimated from 160 pure noise images as shown in table 10.3. Notice that for different methods, we will use the testing threshold corresponding to the pfa (10^{-5} , fifth column in table 10.3).

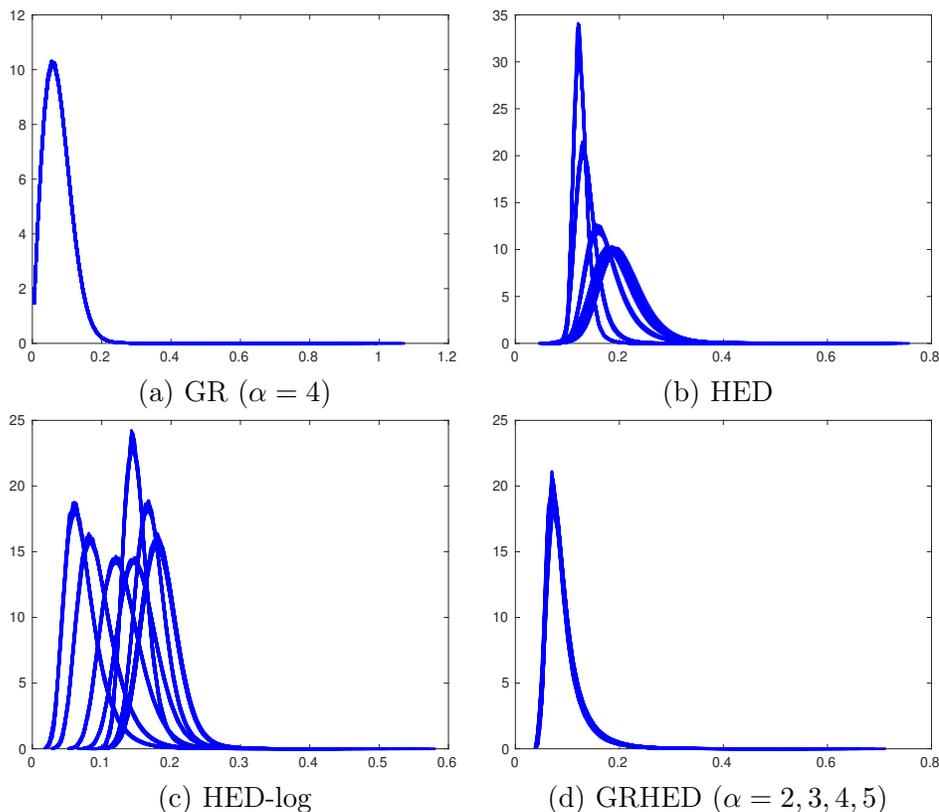
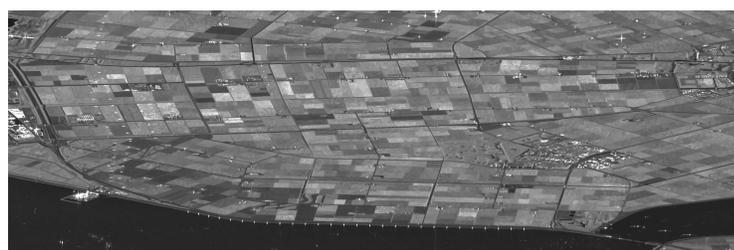


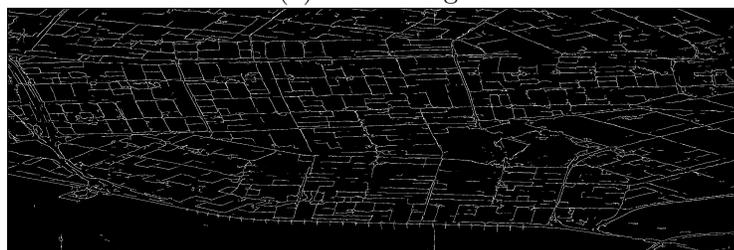
FIGURE 10.6 – The histograms of the gradient magnitude fields produced by GR, HED, HED-log and GRHED in 160 1-look pure noise images of size 1024×1024 pixels. There are 8 possibilities for the mean intensity values of these images, namely, 50^2 , 80^2 , 120^2 , 150^2 , 180^2 , 200^2 , 230^2 and 250^2 . For noise images with each mean intensity value, there are 20 random realizations.

10.4.2 Comparison of different algorithms in realistic SAR images

It is usually difficult to annotate the edges in real SAR images due to strong multiplicative noise. In order to give a quantitative evaluation on the performances of HED, HED-log, GRHED, and GR in images with targets similar to those in real SAR images, we obtain a 'ground truth' by applying HED-clean (HED trained on the grayscale images converted from clean natural images in BSDS500) to a SAR image almost without speckle noise. This SAR image with reduced speckle is obtained by averaging a large amount of well registered Sentinel-1 images (equivalent to temporal multi-looking). Although not justified in changing areas, this temporal multi-looking allows a strong reduction of the speckle in stable ones like roads, urban areas, etc. To overcome the varying residual noise, a final despeckling step is applied Deledalle et al. [2017]. The image is then converted to an 8-bits image using a clipping between $[0, 255]$ with a threshold given by the mean value of the image plus three times of its standard deviation. This image is similar to a natural image and an "edge ground truth" is obtained by using HED-clean on it. The multi-temporal despeckled SAR image and its associated ground truth can be found in figure 10.7.



(a) clean image



(b) GT

FIGURE 10.7 – The denoised multi-look image (Lelystad, Sentinel 1) and its ground truth. The size of the image are 1024×3072 pixels.

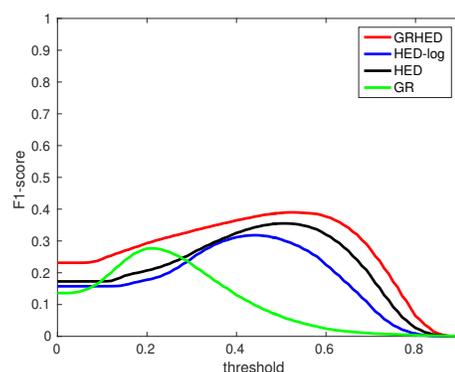
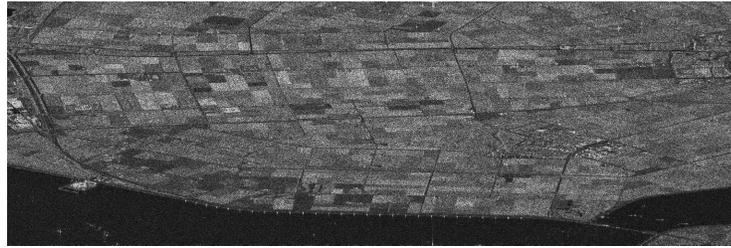
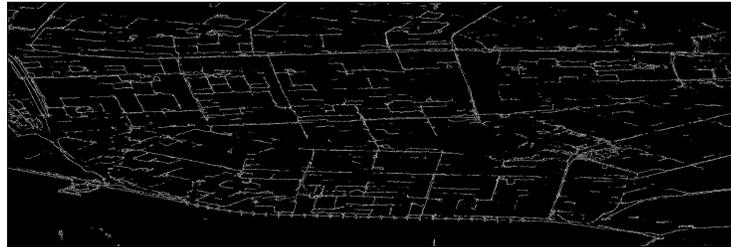


FIGURE 10.8 – F1-score curves computed for GR, HED, HED-log and GRHED in a 1-look synthetic SAR image of size 1024×3072 pixels. The 1-look synthetic SAR image is obtained by multiplying the clean SAR image in figure 10.7 with 1-look speckle noise.

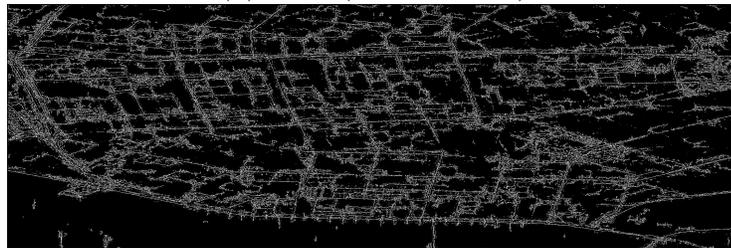
F1-score curves computed for GR, HED, HED-log and GRHED in the synthetic 1-look SAR image can be found in figure 10.8, where 1-look SAR image is obtained by multiplying the clean SAR image with 1-look speckle noise. What can be seen from figure 10.8 is that the F1-score curve of GRHED is above the F1-score curves of all the other methods and the performances of GRHED is not sensitive to the choice of threshold. In addition, both HED and HED-log are shown to be more efficient than GR to detect edges in complex situations.



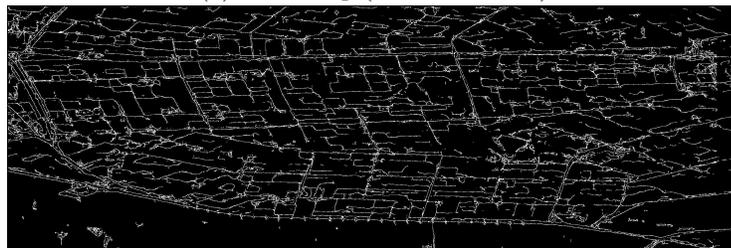
(a) 1-look image



(b) HED (F1-score 0.34)



(c) HED-log (F1-score 0.28)



(d) GRHED (F1-score 0.38)



(e) GR (F1-score 0.28)

FIGURE 10.9 – edge maps computed with a threshold corresponding to $\text{pfa} (10^{-5})$ in a synthetic 1-look image (Lelystad, Sentinel 1) for different methods. For GR, we use the threshold corresponding to $\text{pfa} (10^{-3})$, which is very close to the threshold corresponding to the best F1-score.

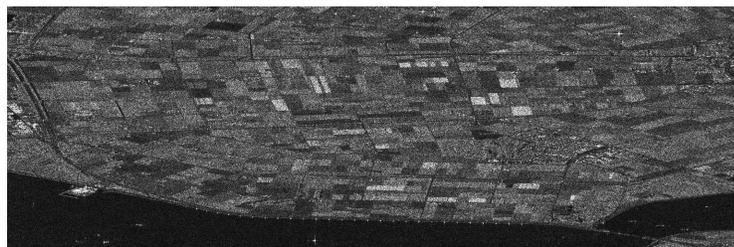
The edge maps obtained with the threshold corresponding to $\text{pfa} (10^{-5})$ for different methods (for GR, the threshold is corresponding to $\text{pfa} 10^{-3}$ in order to preserve more

true edges) can be found in figure 10.9. From figure 10.9 we can see that GRHED detects more true edge pixels than the other methods and the F1-scores of GRHED is higher than all the other methods. In addition, HED also provides competitive edge detection results. GR detects more false detections probably because of the threshold chosen from a higher pfa, but it should be noted that the F1-score of GR is close to its optimal one according to figure 10.8. There are too many false detections for HED-log, this could probably be because HED-log is too far from CFAR, the strategy to choose threshold is not suitable for it. Due to the poor performances of HED-log, we do not compare with it in the following.

10.4.3 Comparison of different algorithms in two 1-look real SAR images

In this part, we test the efficiency of GRHED in two 1-look real SAR images : a 1-look Sentinel-1 image (Lelystad) of size 1024×3072 pixels and a 1-look TerraSAR-X image (San Francisco) of size 2048×2048 pixels. Details about these two 1-look images can be found in section 2.3. We first test the performances of the state-of-art edge detectors AMDR Shui and Fan [2018] and find that AMDR detects too many false detections and thus is not usable in such noisy situations (1-look). We do not provide experiments with it. We compare HED and GRHED with GR in a 1-look real SAR image (Lelystad, Sentinel 1) of size 1024×3072 pixels as shown in figure 10.10. From figure 10.10 we can see that though GR is able to detect many true edges, it is not able to provide a good separation between true edge pixels and noise pixels. Therefore, there are also many false detections in the edge maps produced using GR. In comparison, both HED and GRHED detect many true edge pixels while the number of false detections smaller. However, HED produces spurious detections in some bright areas, this could be probably because the distributions of these areas are not included in those of the training dataset, HED does not learn the way to process them. Compared to HED, GRHED does not suffer from a similar problem. The detections of GRHED in all areas seem reasonable, since the number of false detections is small, and it detects many true edges. In addition, the connectivity of edges detected by GRHED is better than that of HED.

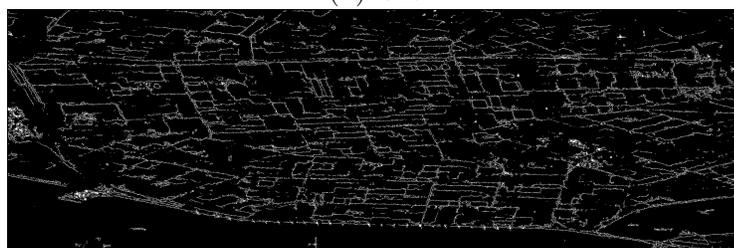
We compare GR, HED, and GRHED in another 1-look real SAR image (San Francisco, TerraSAR-X) of size 2048×2048 pixels. The edge maps obtained by these methods can be found in figure 10.11. Again, the weakness of GR is that it is not able to separate the true edge pixels and noise pixels efficiently. For HED, since many homogeneous areas with high mean intensity values exist in the image, and these kinds of areas do not exist in the training images, it is difficult for HED to perform effective edge detection on them. Therefore, the edge map outputed by HED has spurious detections in many bright areas. In comparison, those bright areas do not cause troubles to GRHED because the gradient distribution computed by GR in homogeneous areas will not be influenced by their mean intensity values. What is more important, GRHED is able to separate true edge pixels and false edge pixels efficiently.



(a) 1-look image



(b) GR



(c) HED

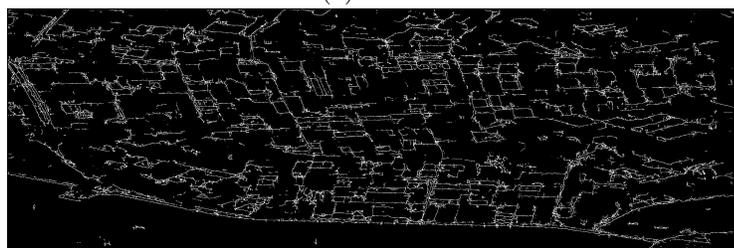
(d) GRHED ($\alpha = 2, 3, 4, 5$)

FIGURE 10.10 – Comparison of different methods in a 1-look real SAR image (Lelystad, Sentinel 1). The size of the image are 1024×3072 pixels.

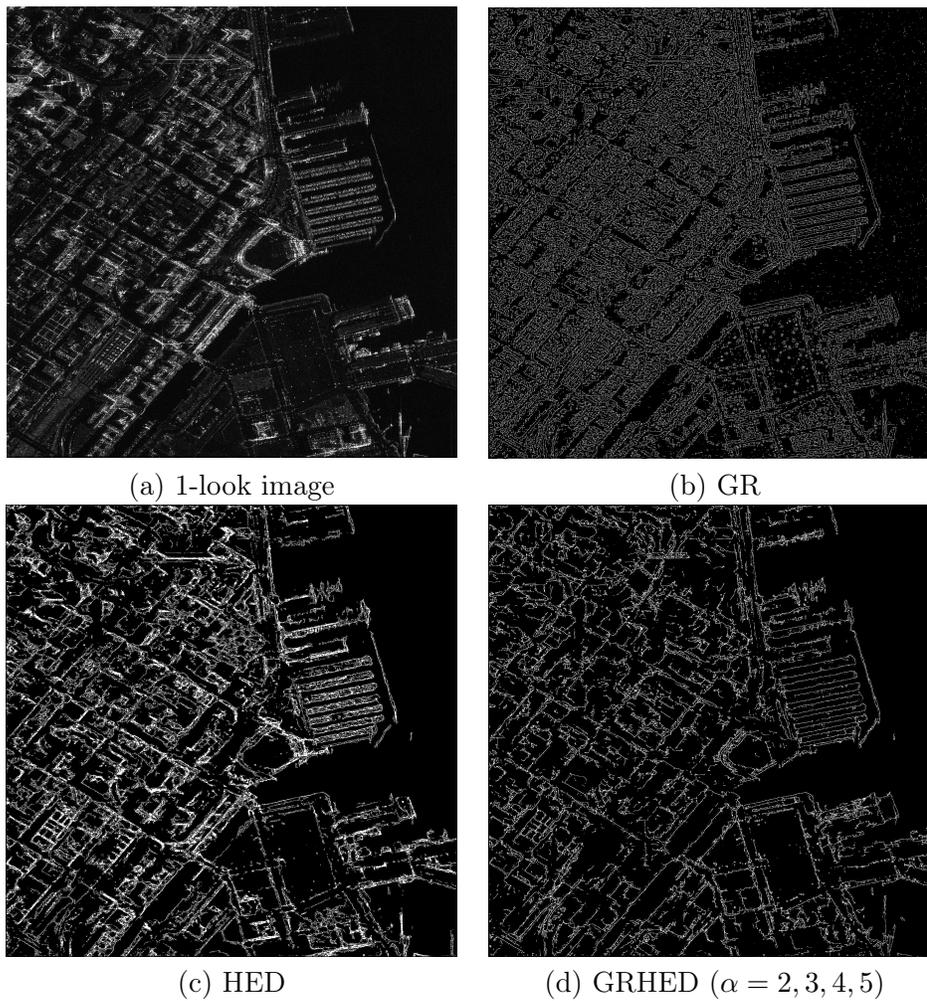


FIGURE 10.11 – Comparison of different methods in a 1-look real SAR image (San Francisco, TerraSAR-X). The size of the image are 2048×2048 pixels.

10.5 Summary of GRHED

In this part we addressed the challenging task of edge detection in 1-look real SAR images. Leveraging the available optical dataset, we proposed to develop a CNN-based edge detector for SAR images by training the CNN models on the gradient magnitude fields of speckled optical images. By introducing a fixed hand-crafted layer (GR) instead of a learnable one, the proposed CNN edge detector GRHED is much less influenced by the differences in the range of pixel values between speckled optical and real SAR images. GR ensures more similarity between the distribution of real SAR images and speckled optical ones, but it should be pointed out that there may be some gradient distributions that do not exist in those of the training dataset. For example, when some edges with extremely high contrast appear in real SAR images, the ratio between the mean intensity values of the homogeneous areas on the opposite side of the edges is too high. From all the experiments we can see that GRHED is able to obtain stable and reliable detection results. This is especially interesting in the case of one-look images.

10.6 Improving line segment detection results with GRHED

As described in chapter 6.4, the main reason for the failure of LSDSAR to detect line segments in 1-look and low contrast situations (with contrast 1.2 and 1.3 for example) is caused by the inefficiency of the gradient computation method GR. It has been demonstrated that GRHED is much more efficient than GR to distinguish edge pixels and noise pixels, as can be seen from the gradient magnitude fields computed by both GR and GRHED in 1-look synthetic edge images with contrast 1.2 to 1.5 with step 0.1. The size of images is 512×512 pixels. For GR, the weighting parameter α is set to 4, as in LSDSAR. For GRHED, multiple α values are combined, specifically, $\alpha = 2, 3, 4, 5$. The gradient magnitude fields computed by GR and GRHED in those 1-look edge images can be found in figure 10.12. It can be easily seen from figure 10.12 that GRHED is much more powerful than GR in both highlighting true edge pixels and suppressing noise pixels.

With the gradient magnitude field computed by GRHED ($\alpha = 2, 3, 4, 5$), we apply the Non-maxima suppression and use the threshold 0.45 (corresponding to pfa 10^{-5}), we can obtain the binary edge map as shown in figure 10.13. The edge maps of figure 10.13 are then used as the input of Hough transform to detect lines in the images. Post-processing steps are then applied to localize lines into line segments. The parameter settings during these steps are as follows : 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 7. The minimum gap between two line segments extracted from the same bin is set to 3. The minimum length of a line segment was set to 7. The comparison between LSDSAR and the method combining Hough transform and GRHED can be found in figure 10.14. What can be seen from this figure is that with the help of GRHED to obtain reasonable edge maps, much more line segments are detected by GRHED plus Hough transform than with LSDSAR. In addition, the number of false detections is also small for GRHED plus Hough thanks to the powerful ability of GRHED to distinguish between edge and non-edge pixels. However, It should be noted that due to the weakness of the Hough transform, there are multiple responses for the same line segments.

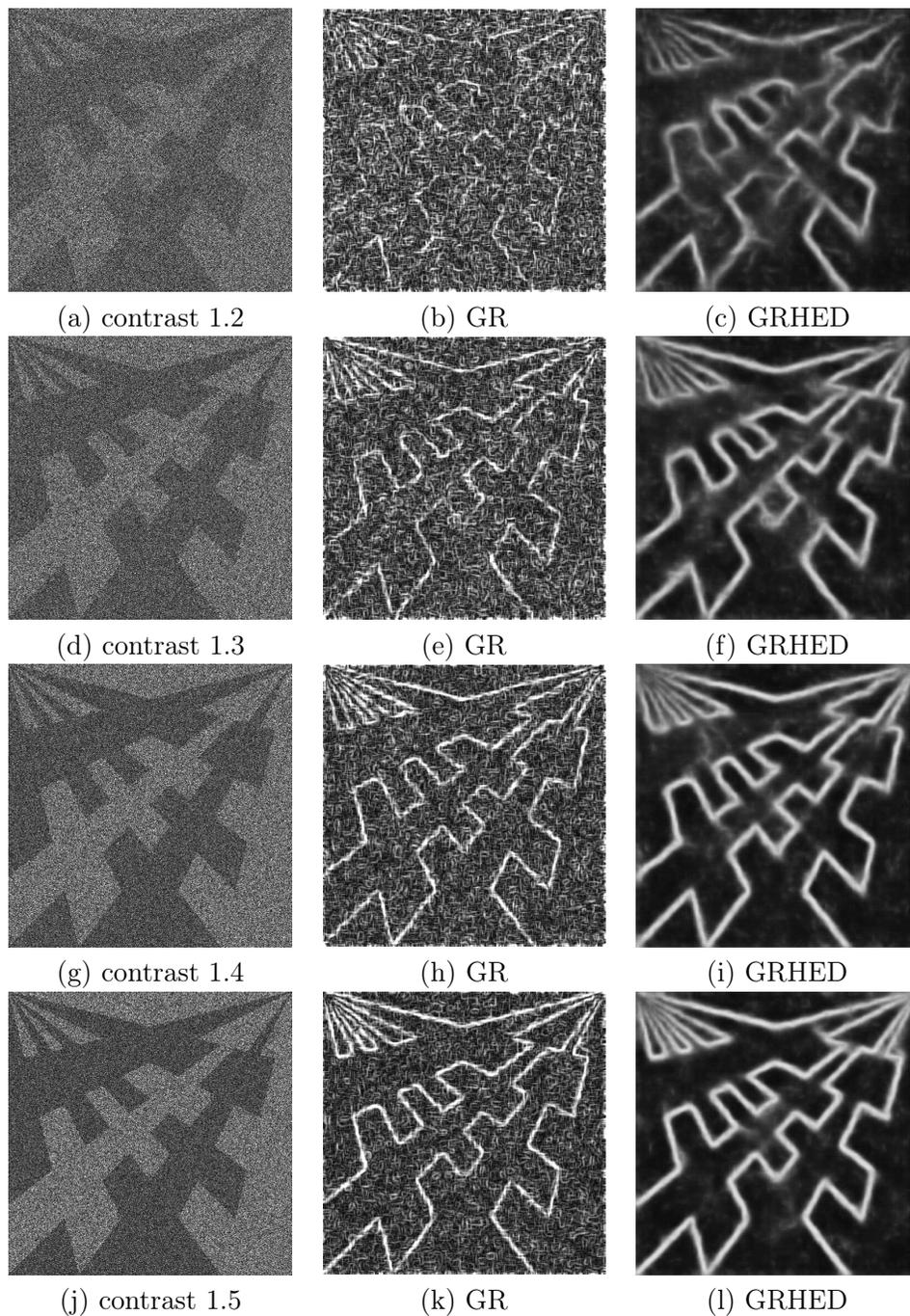


FIGURE 10.12 – The gradient magnitude field computed by GR ($\alpha = 4$) and GRHED ($\alpha = 2, 3, 4, 5$) in 1-look synthetic edge images with amplitude ratio contrasts varying from 1.2 to 1.5 with step 0.1. The size of the images are 512×512 pixels.

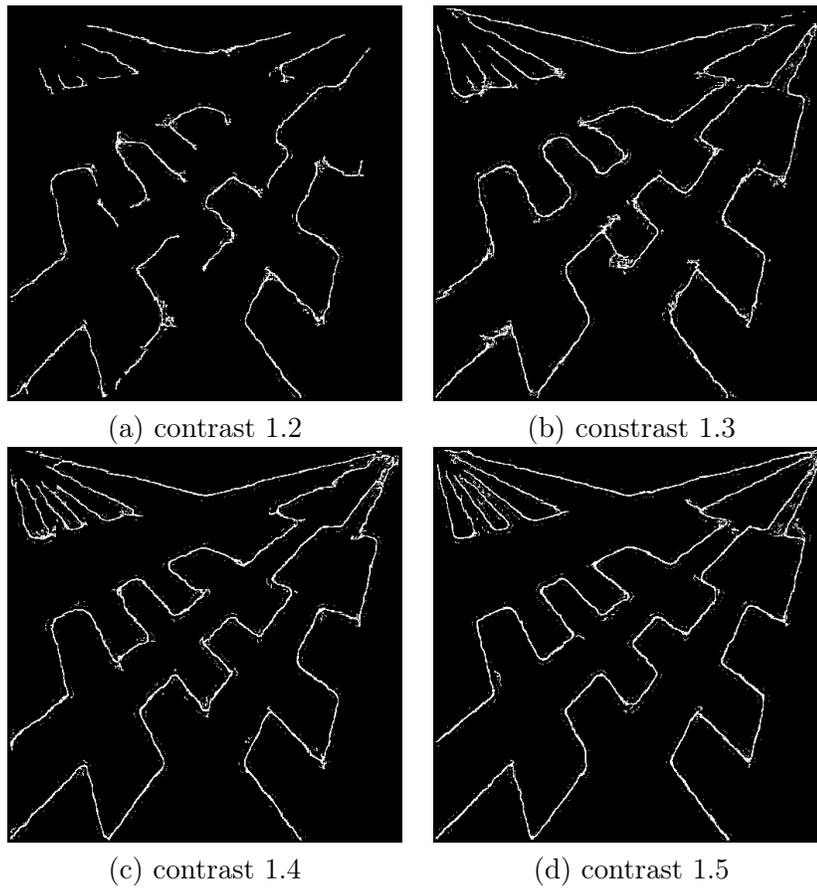


FIGURE 10.13 – Edge maps obtained by applying Non-maxima suppression and thresholding on the gradient magnitude field computed by GRHED ($\alpha = 2, 3, 4, 5$) in four 1-look synthetic edge images with edge contrasts varying from 1.2 to 1.5 with step 0.1. The threshold for all results is the same and set to 0.45. It corresponds to the probability false alarm rate 10^{-5} .

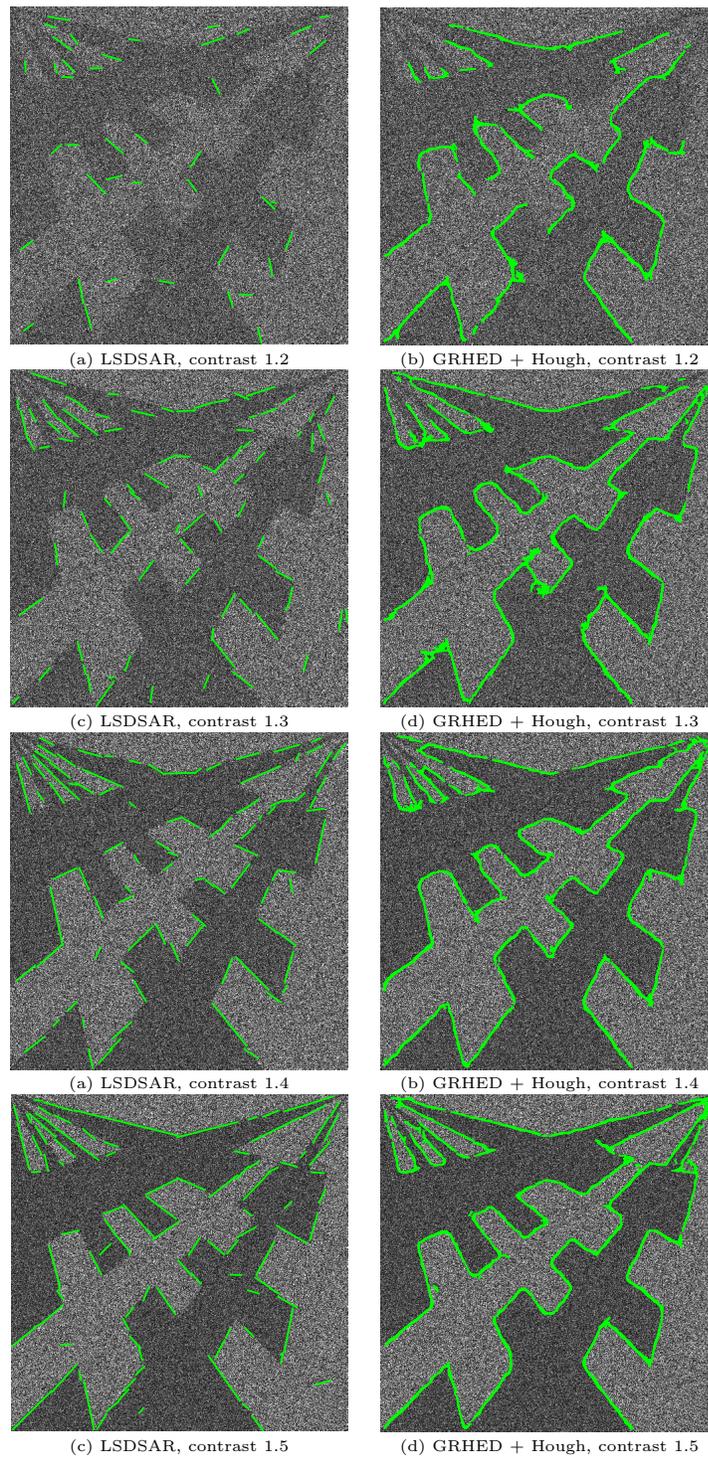


FIGURE 10.14 – Comparison of LSDSAR and GRHED + Hough transform in four 1-look synthetic edge images with edge contrasts varying from 1.2 to 1.5 with step 0.1. The size of the images are 512×512 pixels.

Chapitre 11

Summary of the thesis

11.1 Conclusion

The aim of the thesis is to design low level feature detectors to facilitate the joint use of SAR and optical data. Due to the significant differences between the sources of data, detecting low level features is usually an important preprocessing step for the combination of them. In this thesis, we propose a generic line segment detector and a deep learning based edge detector for SAR images. Indeed, it is still challenging to detect line segments and edges in SAR images, mostly because of the strong multiplicative noise. The line segment detector is based on a Markovian *a contrario* model and the Helmholtz principle. The main advantages of the proposed method is that it is able to detect reasonable amount of line segments and control the number of false detections while requiring little parameter tuning. The number of false detections is controlled through a null hypothesis against which meaningful line segments are detected. The deep learning based edge detector is composed of a hand-crafted layer and learnable convolutional layers. The hand-crafted layer ensures that the feature maps computed by it in real SAR images have minor differences from those of speckled optical images and thus it enables the models trained on the speckled optical images to work well in real SAR images.

A Markovian *a contrario* based line segment detector. We propose a new definition of the background model against which meaningful line segments are detected. The new background model that we call as the Markovian *a contrario* model, assumes that local orientations of pixels follow a first order Markov chain, supposing dependencies between local orientations of neighbouring pixels. It differs a lot from most existing *a contrario* based methods since they are usually based on the independence hypothesis. The Markovian *a contrario* model allows strong filtering during gradient computation, which is necessary to obtain accurate gradient magnitude and local orientation when images are noisy. The proposed Markovian *a contrario* based line segment detector LSDSAR is one of the main contributions of the Ph.D and has been published in Liu et al. [2020].

GRHED, a CNN based edge detector for SAR images. Training dataset with labeled ground truth is of crucial importance for supervised learning based methods. The lack of training dataset for edge detection in SAR images makes it difficult to develop a CNN based edge detector while edge labeling is tedious and time consuming. Under the hypothesis that both SAR and optical images can be divided into piecewise constant areas, we propose to simulate a SAR dataset using optical dataset. Leveraging the valuable CFAR

property of the ratio based gradient computation methods for SAR images, which ensures that the gradient distribution for all homogeneous areas is the same, and the gradient distribution for two homogeneous areas across boundaries depends only on the ratio of their mean intensity values, we propose to train CNNs on the gradient magnitude fields of speckled optical images and apply the trained models to the gradient magnitude fields of real SAR images. The proposed training strategy enables the models trained on the speckled optical dataset to work well in real SAR images. The proposed CNN based edge detector GRHED is the second main contribution of the Ph.D and has been submitted to IEEE Transactions on Geoscience and Remote Sensing.

11.2 Perspectives

Line segment detection in SAR images. Even though the performances of LSDSAR are generally reasonable, many true line segments are missing, especially in 1-look SAR images. The main reason is that the gradient magnitude and gradient orientation are not accurate enough for the traditional gradient computation methods. We have shown the powerful ability of CNN based methods to distinguish between true edge pixels and noise pixels in 1-look images. The next step is to develop a line segment detector which is able to detect much more true line segments in 1-look real SAR images leveraging CNNs. It has been studied in section 10.6 that with a post-processing of the edge maps produced by GRHED using Hough transform, much more true line segments are detected in 1-look images with low contrasted edges, even though the performances are not fully satisfying because of the multiple responses for the same line segment caused by the weakness of the Hough transform based method. A better grouping and validation step can be developed to obtain a much better line segment detector for SAR images. Another strategy could be developing a CNN based line segment detector directly without the step of edge detection, as the deep learning based line segment detector described in Huang et al. [2018], Xue et al. [2019]. Under the hypothesis that both optical images and SAR images can be divided into piecewise constant areas, the dataset for line segment detection in optical images should be applicable to SAR images, with the first layer of CNNs being a ratio based gradient computation method, such as GR Dellinger et al. [2015].

Taking spatial correlations of noise into account, and developing exclusive structure detector for SAR images. the spatial correlation of the noise on real images has not been addressed and probably leads to a decrease of the CNN performances. The method of Lapini et al. [2014] could be an interesting approach to take it into account. Secondly, the specific features of SAR images like bright points and lines due to strong backscatterings of diedral or triedral structures do not exactly correspond to edges. Therefore dedicated detectors should be developed for these specific structures to be combined with edge detectors.

Image segmentation. Under the hypothesis that both optical images and real SAR images are composed of piecewise constant areas, developing a CNN based image segmentation method leveraging the optical dataset should also be possible.

Image registration. For the joint use of SAR and optical data, a preliminary registration is usually required. With robust line segment detector and edge detector for both optical and real SAR images, it is possible to develop reliable feature based image registration

method. A preliminary idea is to first detect line segments and edges from both optical and SAR images and find the correspondence between the low level features, such as those described in Zhang et al. [2015]. However, special care should be taken because there exists geometric distortions in SAR images, especially for the urban areas. In addition, the resolution of SAR images and optical images may be different, this should also be taken into account during the matching of low level features. After the registration of optical and SAR images, the information from both of them can be combined for many applications such as change detection.

Funding

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

Bibliographie

- 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.
- P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 : 898–916, 2011.
- 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.
- G. Bertasius, J. Shi, and L. Torresani. Deepedge : A multi-scale bifurcated deep network for top-down contour detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4389, 2015.
- A. Bonci, T. Leo, and S. Longhi. A bayesian approach to the hough transform for line detection. *IEEE Transactions on Systems, Man, and Cybernetics*, 35 :945–955, 2005.
- K. Bowyer, C. Kranenburg, and S. Dougherty. Edge detector evaluation using empirical roc curves. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, page 359, 1999.
- J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8 :679–698, 1986.
- J. Chanussot, G. Mauris, and P. Lambert. Fuzzy fusion techniques for linear features detection in multitemporal sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 37 :1292–1305, 1999.
- T. Chen, L. Chen, and Y. Su. A sar image registration method based on pixel migration of edge-point features. *IEEE Geoscience and Remote Sensing Letters*, 11 :906–910, 2014.
- 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*, 40 :1195–1208, 2018.
- M. Dai, C. Peng, A. Chan, and D. Loguinov. Bayesian wavelet shrinkage with edge detection for sar image despeckling. *IEEE Transactions on Geoscience and Remote Sensing*, 42 :1642–1648, 2004.
- C. Deledalle, L. Denis, S. Tabti, and F. Tupin. MuLoG, or how to apply gaussian denoisers to multi-channel sar speckle reduction ? *IEEE Transactions on Image Processing*, 26(9) : 4389–4403, Sept 2017. ISSN 1057-7149. doi : 10.1109/TIP.2017.2713946.
-

- 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*, 53 :453–466, 2015. doi : 10.1109/TGRS.2014.2323552.
- A. Desolneux, L. Moisan, and J.-M. Morel. Meaningful alignments. *International Journal of Computer Vision*, 40 :7–23, 2000.
- A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis. A Probabilistic Approach*. Springer, 2008.
- P. Dollar and C. L. Zitnick. Fast edge detection using structured forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37 :1558–1570, 2015.
- S. Dougherty, K. W. Bozyer, and C. Kranenburg. Roc curves evaluation of edge detector performance. In *Proceedings 1998 International Conference on Image Processing*, volume 2, pages 525–529, 1998.
- R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15 :11–15, 1972.
- 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*, 36 :793–802, 1998. doi : 10.1109/36.673672.
- J. Goodman. *Statistical properties of laser speckle patterns*, volume ch. 2. Laser Speckle and Related Phenomena, 1975.
- 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*, 32 :722–732, 2010. doi : 10.1109/TPAMI.2008.300.
- R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd : a line segment detector. *Image Processing On Line*, 2 :35–55, 2012.
- B. Grosjean and L. Moisan. A-contrario detectability of spots in textured backgrounds. *Journal of Mathematical Imaging and Vision*, 33(3) :313–337, 2009.
- O. Hellwich and 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*, volume 3, pages 883–886, 1996.
- 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.
- J. Illingworth and J. Kittler. The adaptive hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9 :690–698, 1987.
- W. Jiang, K.-M. Lam, and T. zhi Shen. Efficient edge detection using simplified gabor wavelets. *IEEE Transactions on Systems, Man, and Cybernetics*, 39 :1036–1047, 2009.
- J. Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1 :37–42, 1983.
-

-
- S. Konishi, A. L. Yuille, J. M. Coughlan, and S. Zhu. Statistical edge detection : Learning and evaluating edge cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 :57–74, 2003.
- P. Kovesei et al. Symmetry and asymmetry from local phase. In *Tenth Australian joint conference on artificial intelligence*, volume 190, pages 2–4. Citeseer, 1997.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS) 2012*, pages 1106–1114, 2012.
- A. Lapini, T. Bianchi, F. Argenti, and L. Alparone. Blind speckle decorrelation for SAR image despeckling. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2), 2014.
- C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.
- J.-S. Lee and I. Jurkevich. Coastline detection and tracing in sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 28 :662–668, 1990.
- C. Liu, Y. Xiao, and J. Yang. A coastline detection method in polarimetric sar images mixing the region-based and edge-based active contour models. *IEEE Transactions on Geoscience and Remote Sensing*, 55 :3735–3747, 2017a.
- C. Liu, R. Abergel, Y. Gousseau, and F. Tupin. Lsdsar, a markovian a contrario framework for line segment detection in sar images. *Pattern Recognition*, 98, 2020. doi : <https://doi.org/10.1016/j.patcog.2019.107034>.
- Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. Richer convolutional features for edge detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017b.
- Y. Liu, M.-M. Cheng, X. Hu, J.-W. Bien, L. Zhang, X. Bai, and J. Tang. Richer convolutional features for edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 :1939–1946, 2019.
- J. Long, E. Shelhamer, and T. Darrel. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.
- D. R. Martin, C. C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 :530–549, 2004.
- F. Medeiros, R. Costa, R. Marques, and C. Laprano. Multiscale detection of linear features in speckled imagery. In *16th Brazilian Symposium on Computer Graphics and Image Processing 2003*, pages 371–375, 2003.
- A. Myaskouvsky, Y. Gousseau, and M. Lindenbaum. Beyond independence : An extension of the *a contrario* decision. *International journal of Computer Vision*, 101 :22–44, 2013. doi : [10.1007/s11263-012-0543-6](https://doi.org/10.1007/s11263-012-0543-6).
-

- F. O’Gorman and M. Clowes. Finding picture edges through collinearity of feature points. *IEEE Transactions on Computers*, C-25 :449–456, 1976.
- C. Palmann, S. Mavromatis, and J. Sequeira. Sar image registration using a new approach based on the generalized hough transform. In *ISPRS 2008*, volume XXXVII, 2008.
- E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 :640–651, 2017.
- D. Shen, J. Zhang, J. Yang, D. Feng, and J. Li. Sar and optical image registration based on edge features. In *2017 4th International Conference on Systems and Informatics*, 2017.
- W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. Deepcontour : A deep convolutional feature learned by positive-sharing loss for contour detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3982–3991, 2015.
- P. Shui and S. Fan. Sar image edge detection robust to isolated strong scatterers using anisotropic morphological directional ratio test. *IEEE Access*, 6 :37272–37285, 2018.
- P.-L. Shui and D. Cheng. Edge detector of sar images using gaussian-gamma-shaped bi-windows. *IEEE Geoscience and Remote Sensing Letters*, 9 :846–850, 2012. doi : 10.1109/LGRS.2012.2184521.
- K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- J. Skingley and A. Rye. The hough transform applied to sar images for thin line detection. *Pattern Recognition Letters*, 6 :61–67, 1987.
- H. Song, B. Huang, and K. Zhang. A globally statistical active contour model for segmentation of oil slick in sar imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6 :2402–2409, 2013.
- 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*, 53 :6058–6072, 2015.
- R. Touzi, A. Lopes, and P. Bousquet. A statistical and geometrical edge detection for sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 26 :764–773, 1988. doi : 10.1109/36.7708.
- F. Tupin, H. Maître, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky. Detection of linear features in sar images : application to road network. *IEEE Transactions on Geoscience and Remote Sensing*, 36 :434–453, 1998. doi : 10.1109/36.662728.
- Q. Wei and D. Feng. Extracting line features in sar images through image edge fields. *IEEE Geoscience and Remote Sensing Letters*, 13 :540–544, 2016.
- 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*, 14 :1865–1869, 2017.
-

-
- Q.-R. Wei, D.-Z. Feng, and H. Xie. Edge detector of sar images using crater-shaped window with edge compensation strategy. *IEEE Geoscience and Remote Sensing Letters*, 13 : 38–42, 2016.
- S. Xie and Z. Tu. Holistically nested edge detection. In *2015 IEEE International Conference on Computer Vision*, pages 1395–1403, 2015.
- S. Xie and Z. Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125 :3–18, 2017.
- B. Xiong, W. Li, L. Zhao, J. Lu, X. Zhang, and G. Kuang. Registration for sar and optical images based on straight line features and mutual information. In *2016 IEEE International Geoscience and and Remote Sensing Symposium*, 2016.
- D. Xu, W. Ouyang, X. Mamedea-Pineda, E. Ricci, X. Wang, and N. Sebe. Learning deep structured multi-scale features using attention-gated crfs for contour prediction. In *2017 Conferce on Neural Information Processing Systems (NIPS 2017)*, 2017.
- F. Xu and Y. Jin. Automatic reconstruction of building objects from multiaspect meter-resolution sar images. *IEEE Transactions on Geoscience and Remote Sensing*, 45 : 2336–2353, 2007.
- 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*, pages 1595–1603, 2019.
- 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*, pages 193–202, 2016.
- P. Yu, A. Qin, and D. A. Clausi. Unsupervised polarimetric sar image segmentation and classification using region growing with edge penalty. *IEEE Transactions on Geoscience and Remote Sensing*, 50 :1302–1317, 2012.
- G. Zhang, H. Sui, Z. Song, F. Hua, and L. Hua. Automatic registration method of sar and optical image based on line features and spectral graph theory. In *2017 2nd International Conference on Multimedia and Image Processing*, 2017.
- H. Zhang, W. Ni, W. Yan, J. Wu, and S. Li. Robust sar image registration based on edge matching and refined coherent point drift. *IEEE Geoscience and Remote Sensing Letters*, 12 :2115–2119, 2015.
- M. Zhao, Y. Wu, S. Pan, F. Zhou, B. An, and A. Kaup. Automatic registration of images with inconsistent content through line support region segmentation and geometrical outlier removal. *IEEE Transactions on Image Processing*, 27 :2731–2746, 2018.
-

Titre : Détection de caractéristiques de bas niveau dans les images SAR

Mots clés : détection de segments de droite, détection de contours, images SAR, modèle Markovien, méthode a contrario, réseau de neurones convolutionnels.

Résumé : Nous développons un détecteur de segments de droite et un détecteur de bords pour les images SAR. Le détecteur de segments de droite proposé, LSDSAR, est basé sur un modèle a contrario Markovien et le principe de Helmholtz, où les segments de droite sont validés en fonction d'une mesure de significativité. Contrairement aux approches a contrario habituelles, le modèle Markovien a contrario permet un fort filtrage lors du calcul du gradient. LSDSAR bénéficie de la précision et de l'efficacité du nouveau modèle de fond. Le détecteur de bords proposé GRHED est basé sur l'apprentissage profond. Les contributions du travail sont doubles: 1) nous proposons de simuler un ensemble de données SAR à l'aide d'un ensemble de données optiques; 2) nous proposons d'entraîner des réseaux convolutionnels classiques sur les champs de magnitude du gradient des images pour assurer la similitude de la distribution du gradient entre les images SAR et optiques. GRHED surpasse largement les méthodes concurrentes.

Title : Low level feature detection in SAR images

Keywords : line segment detection, edge detection, SAR images, Markovian model, a contrario model, convolutional neural network.

Abstract : We develop a line segment detector and an edge detector for SAR images. The proposed line segment detector LSDSAR is based on a Markovian a contrario model and the Helmholtz principle, where line segments are validated according to their meaningfulness. Contrarily to the usual a contrario approaches, the Markovian a contrario model allows strong filtering during gradient computation. LSDSAR benefits from the accuracy and efficiency of the new background model. The proposed edge detector GRHED is based on deep learning. The contributions of the work are two fold: 1) we propose to simulate a SAR dataset using optical dataset; 2) we propose to train usual CNNs on the gradient magnitude fields of images to ensure similarity of gradient distribution between SAR and optical images. GRHED largely outperforms concurrent methods.