Exploiting Multi-temporal Information for SAR Images Segmentation

François Aspert

Assistant : Dr. Meritxell Bach Cuadra
Director : Prof. Dr. Jean-Philippe Thiran

Signal Processing Institute (ITS),
Swiss Federal Institute of Technology (EPFL)

May 10, 2006
Abstract

Earth observation is taking more and more importance in our society. Indeed, recent technologies march allows for satellite images to get ground resolution up to less than one meter. Thus, most of satellite image information obtained begins now to be widely used for multiple purposes ranging from cartography and mapping to agricultural management and security applications. Among all the satellite imaging systems, Synthetic Aperture Radar (or SAR) are very interesting for global earth surveillance and monitoring purposes as they can provide data at any time under all weather conditions. However only few work have been done to take advantage of such time robustness properties for retrieving the images main structures.

In this study we focused one the particular issue of obtaining an efficient closed region segmentation using multi-temporal SAR image series. To this end, several tools have developed. In order to remove SAR inherent noise we filtered the image sequences using a multi-temporal anisotropic non-linear diffusion algorithm. Then a new approach of Canny edge detection have been introduced to extract edge features from these vector-valued images. Finally, a region segmentation step has been applied to get closed regions.

We demonstrated that for all the methods, exploiting multi-temporal information can improve the results accuracy. Moreover, since some radiometric changes can occur within region due to man intervention or natural diseases we also presented a direct application of region segmentation to retrieve such changes across time series.
Acknowledgments

This work was carried out between September 2005 and March 2006 at the Signal Processing Institute of the Swiss Federal Institute of Technology in Lausanne, Switzerland and in collaboration with the Sarmap company, Purasca, Switzerland.

First of all, I would like to thank the director of this Master thesis, Professor Jean-Philippe Thiran for giving me the possibility to work at LTS and for the valuable advices he gave me throughout this work. I would like to thank also my supervisor Doctor Meritxell Bach Cuadra for her availability, good mood and highly enriching talk we had along this project. As well I am really grateful to Doctor Francesco Holecz and Alessio Cantone from Sarmap for sharing with me their expertise and enthusiasm about SAR image processing and for their worth comments and ideas.

I would also like to give my warmest thank to all the members of my family for its unconditional support which is so important for me. Another thanks surely goes to Blanchon, the family cat, for always understanding me.

Finally, I have a special thought for Naline, who have always been there for me. Thank you for always giving me your joy for life and infallible help and support.
# Contents

1 Introduction ......................................................... 1
   1.1 Background ..................................................... 2
       1.1.1 SAR image formation ...................................... 2
       1.1.2 Speckle ..................................................... 3
       1.1.3 Data sets .................................................. 4
   1.2 Objectives ...................................................... 6
   1.3 Organization of the text ...................................... 6

2 Speckle Filtering .................................................. 7
   2.1 State of the art ................................................. 7
   2.2 Methodology .................................................... 11
       2.2.1 Non-linear diffusion ...................................... 11
       2.2.2 Single image anisotropic nonlinear diffusion .............. 12
       2.2.3 Multi-temporal anisotropic nonlinear diffusion .......... 13
       2.2.4 Diffusion parameters ...................................... 16
       2.2.5 Discretization schemes .................................... 18
   2.3 Results ........................................................ 19
       2.3.1 Nonlinear diffusion ........................................ 19
       2.3.2 Anisotropic nonlinear diffusion ........................... 19
       2.3.3 Influence of parameters .................................... 21
       2.3.4 Method comparison ......................................... 21
   2.4 Discussion ...................................................... 25

3 Edge detection ..................................................... 27
   3.1 State of the art ................................................. 28
   3.2 Methodology .................................................... 29
       3.2.1 Single image ............................................... 30
       3.2.2 Multi-temporal images ..................................... 33
   3.3 Results ........................................................ 34
   3.4 Discussion ...................................................... 37

4 Region segmentation ............................................... 38
   4.1 State of art ..................................................... 39
   4.2 Methodology .................................................... 40
       4.2.1 Boundary constrained agglomerative segmentation ....... 40
       4.2.2 Unsupervised statistical segmentation .................... 43
   4.3 Results ........................................................ 47
4.3.1 Unsupervised statistical segmentation 47
4.3.2 Region merging 47
4.3.3 Watershed 51
4.4 Discussion 55

5 Time varying region segmentation 57
5.1 Methodology 58
5.1.1 FGMM fitting 59
5.1.2 MDL criterion 60
5.1.3 Agglomerative Fitting 61
5.1.4 Time-varying segmentation 63
5.2 Results 63
5.3 Discussion 67

6 Conclusion and future work 69

A Bilinear interpolation 74
Introduction

Nowadays, earth observation is taking an ever-growing importance in our society. Indeed, recent technology march allows for satellite imaging to get ground resolution up to less than one meter resolution cells for optical image devices. Thus, most of satellite image information obtained begins now to be widely used in Geographic Information Systems (GIS) for multiple purposes ranging from cartography and mapping to agricultural management and security applications.

Among all the satellite imaging systems, Synthetic Aperture Radar (or SAR) and its intrinsic properties are very interesting for global earth surveillance and monitoring purposes. Indeed, in contrast to passive optical imaging systems that require the Sun’s illumination, SAR is an active and coherent remote sensing method. Hence it is possible to proceed to a sensing independently from weather and day/night conditions. Therefore, it can provide reliable information at any time which is particularly useful for risk detection programs. However, if SAR images can be very useful, they are subject to annoying drawbacks compared to optical images. Indeed, the maximum ground resolution that can be achieved (for ERS1) is 12.5x12.5 meters resolution cells. Moreover, coherent interference between backscatterers also known as speckle noise introduce high level of perturbation for such images, reducing the final quality of images and preventing use of classical techniques for efficient image processing.

Although many image processing tools have been successfully developed in the past few years for optical images, there is a real need for a SAR specific treatment which would take into account SAR characteristics (especially speckle) and perform classical tasks for segmentation, classification and more generally feature extraction. Particularly, when multi-temporal information is available which is likely to be the case in SAR images, the use of appropriate techniques could add some extra information and enhance accuracy of feature extraction process in order to be fully integrated to GIS softwares.
CHAPTER 1. INTRODUCTION

1.1 Background

This section presents basic theory for SAR image understanding. A more complete information can be found in [12] and [28].

1.1.1 SAR image formation

SAR is a coherent, active, microwave imaging method that improves natural radar resolution by focusing the image through a process known as synthetic aperture processing. SAR takes advantage of the Doppler history of the radar echoes generated by the forward motion of the spacecraft to synthesize a large antenna, enabling high azimuthal resolution in the resulting image despite a physically small antenna. As the radar moves, a pulse is transmitted at each position. The return echoes pass through the receiver and are recorded in an echo store (see Figure 1.1).

In essence, return signals from the center portion of the beamwidth are discriminated by detecting Doppler frequency shifts, which is a change in wave frequency resulting from the relative velocities of a transmitter and a reflector. Within the wide antenna beam, returns from features in the area ahead of the platform will have upshifted, or higher, frequencies resulting from the Doppler effect. Conversely, returns from features behind the platform will have downshifted, or lower, frequencies. Returns from features near the centreline of the beamwidth (the so-called Zero-Doppler line) will experience no frequency shift.

If achieving a good azimuth resolution is important, another characteristic that strongly influence the final SAR image quality is the range resolution which measure the resolution in perpendicularly to the azimuth direction. The range or across-track resolution is dependent on the length of the pulse ($P$). Two distinct targets on the surface will be resolved in the range dimension if their separation is greater than half the pulse length. Slant range resolution remains constant, independent of range. However, when projected into ground range coordinates, the resolution in ground range will be dependent of the incidence angle. Thus, for fixed slant range resolution, the ground range resolution will decrease with increasing range.

Radar images are composed of many picture elements referred to as pixels. Each pixel in the radar image represents an estimate of the radar backscatter for that area on the ground also called radar cross section (RCS) or backscattering coefficient. Darker areas in the image represent low backscatter, while brighter areas represent high backscatter. Bright features mean that a large fraction of the radar energy was reflected back to the radar, while dark features imply that very little energy was reflected. Backscatter for a target area at a particular wavelength will vary for a variety of conditions, such as the physical size of the scatterers in the target area, the target’s electrical properties and the moisture content, with wetter objects appearing bright, and drier targets appearing dark. (the exception to this is a smooth body of water, which will act as a flat surface and reflect incoming pulses away from a target, leading to a dark appearance). The wavelength and polarization of the Radar pulses, and the observation angles will also affect backscatter.
1.1.2 Speckle

This a crucial issue for SAR image processing as it damage seriously images. Speckle occurs in all types of imagery acquired from coherent imaging systems, such as laser, acoustic and SAR imagery. This is caused by random constructive and destructive interference of the de-phased but coherent return waves scattered by the elementary scatterers within each resolution cell [10] (see Fig. 1.2).

Goodman gives an extensive examination of the origins and statistical properties of image speckle. The first- and second-order statistical properties of speckle are well known. Fully developed speckle is characterized by a negative exponential distribution in a single look intensity SAR image or equivalently a Rayleigh distribution in an amplitude image. Speckle effects can be reduced
using multi-looking (see next section). The probability density function of the intensity follows then a Gamma distribution.

The formulation of the SAR image as a deterministic RCS modulating a random stationary process has led to the most frequently assumed model for speckle interference:

$$ I = \sigma \cdot n $$ \hspace{1cm} (1.1)

where $I$ is the observed intensity, $\sigma$ the noise-free image and $n$ the speckle component. SAR-specific techniques for speckle reduction will use this model and probability density function (PDF) to estimate and reduce the inherent speckle noise.

1.1.3 Data sets

This section present the two datasets that will be used in this study for testing and validation. Figure 1.3 is a sample from a set of 16 images from South-Africa agricultural land taken between 13/11/2004 and 16/07/2005. We will further refer to this set as South-Africa. Figure 1.4 also represents sample image from a set of 12 images from Philippine agricultural land taken between the 20/11/2003 and 08/04/2005. We will further refer to this set as Philippines.

Images from both sequences are obtained from different sensors (Envisat ASAR and Radarsat1). Then these RAW data have been processed using the chain presented in Figure 1.5.

![South-Africa original multi-temporal sequence](image1.png)

(a) 22 Jan. 05  (b) 23 May 05

**Figure 1.3** — South-Africa original multi-temporal sequence
CHAPTER 1. INTRODUCTION

Figure 1.4 — Rice original multi-temporal sequence

(a) 3 Jan. 04
(b) 4 Mar. 04

Figure 1.5 — SAR pre-processing chain

SAR raw data

- **Focusing**
  - Focusing is the process of gathering all information acquired during scanning to synthesize the effective aperture antenna. The obtained data is also called single look complex data (SLC).
  
  Since SLC is very speckled, multi-look images can be generated by averaging values over range and/or azimuth (with respect to pixel spacing) leading to a speckle reduction and a degradation in spatial resolution.

- **Multi-looking**

- **Co-registration**
  - Co-registration is the process of superimposing in the same range geometry two or more images that have the same orbit and acquisition mode.

- **Speckle filtering**
  - A first speckle filtering step is taken taking advantage of multi-temporal series to perform a multi-channel filtering.

- **Geocoding**
  - Geocoding is the process of converting each pixel from the slant range geometry to a cartographic reference system (i.e. UTM 32)
CHAPTER 1. INTRODUCTION

1.2 Objectives

The main goal of this work is to provide efficient and robust tools for multi-temporal SAR image segmentation. By segmentation, we here refer to the delineation of closed homogenous regions within input images. Such a segmentation is for instance required proceeding to a further classification in order to take advantage of a region-based classification.

To this end, a multi-image speckle filtering/smoothing method will be developed for enabling some reliable feature extraction. Then, for enhancing the final segmentation result, an original extension of Canny \cite{8} algorithm for multi-temporal edge detection will be designed using the previously filtered data. Closed regions will be then obtained using a two-part coding scheme using an edge map as side information and a region growing technique. Hence, we will study how an efficient edge detection can contribute to provide an accurate region segmentation. Moreover it is also highly interesting to detect intra-region changes over time for a potential classification step involving for instance crop growth profile. Thus, and in order to retrieve the backscattering changes relative to natural scene fluctuation, the multi-temporal segmentation model will be refined to a single image segmentation model to retrieve the regions time profiles.

In this report, we will also review the different techniques already existing in all the investigated fields and discussing about their advantages/drawbacks when applied to multi-temporal SAR images. Then, we will present the different chosen methods for each step and have quantitative (when possible) and qualitative assessment concerning the attained results. Finally, we will compare these results with those obtained with an existing commercial software InfoPACK version 1.2.1 \cite{1} performing the same operations when available.

In order to summarize the structure of this work let us recall the different steps taken:

- Speckle Filtering/Smoothing
- Edge Detection
- Region Segmentation
- Time varying region segmentation

1.3 Organization of the text

The report is organized as follow. Chapter 2 introduce an al. Chapter 3 will deal with an original multi-temporal global edge detection algorithm. Chapter 4 will use results previously obtained to get the final region segmentation with a vector-valued region merging algorithm using side information and minimum description length. In chapter 5 we will refine the region algorithm to obtain single image segmentation rather than a global cartoon model. Finally, conclusions and future work will be discussed in chapter 6.
The first task to process SAR image is a regularization task. Indeed, due to the residual amount of speckle noise that corrupt such images, there is a crucial need to perform an adaptive smoothing of the images in order to eliminate the noise effect while preserving the contours as much as possible for enhancing results when applying feature extraction tools (e.g. edge detection). For this purpose, Oliver [12] is suggesting four criteria in that a despeckling filter (also known as RCS reconstruction filter) should fulfill in order to be optimal:

1. Speckle reduction in extended uniform regions
2. Feature preservation
3. Absence of artifacts
4. Radiometric preservation

If point 1 and 3 are obvious quality criteria, the second point is a fundamental issue for further processing as it implies that image relevant features such as edges should appear in the filtered image. In the same way, point 4 is also extremely important. Indeed, such a constraint will enable the future establishment of radiometric-temporal profiles of the homogenous filtered zones in order to proceed for example to a final comparison with known crop-growth profile.

In this chapter we will first review and discuss some existing methods widely used to perform such a speckle filtering. In section 2.2 we will present three despeckling method for SAR images. Then, in section 2.3 and 2.4, we will measure the accuracy of the different filtering results obtained with our method related to the criteria cited above and discuss about their respective impact for a potential segmentation.

2.1 State of the art

Several algorithms have been developed in order to reduce SAR inherent speckle noise using both single image and multi-image information. This section present
different methods that have been extensively used over the past few years for such a task.

**Lee filter** Based on a linear speckle model and the minimum mean square error approach (MMSE), the Lee filter, first described in [11], make use of local statistics in a sliding window of size \(N \times N\) to eliminate the speckle part from image. The despeckled image is computed in the following way:

\[
y(i,j) \approx x(i,j) \cdot n(i,j),
\]

where \(y(i,j)\) is the corrupted image, \(x(i,j)\) the noise-free image and \(n(i,j)\) the speckle component. Taking the logarithm, we then have:

\[
\log(y(i,j)) = \log(x(i,j)) + \log(n(i,j)).
\]

Then, using the MMSE approach, we can estimate the uncorrupted image using:

\[
\hat{x}(i,j) = \bar{I}(i,j) + k(i,j)(y(i,j) - \bar{y}(i,j)).
\]

Where \(\bar{y}(i,j)\) corresponds to the mean value within the window centered in \((i, j)\), \(\eta(i,j)\), \(k(i,j)\) is the adaptative filter coefficient.

\[
k(i,j) = 1 - \frac{C^2_u}{C^2_{(i,j)}},
\]

where,

\[
C^2_{(i,j)} = \left(1 / |\eta(i,j)| \right) \cdot \frac{\sum_{p \in \eta(i,j)} y_p - \bar{y}^2_{(i,j)}}{\bar{y} - \bar{y}^2_{(i,j)}},
\]

and \(C^2_u\) is a constant for a given image. The filtering is then controlled by the local statistics \(C(i,j)\), the value of \(k(i,j) \rightarrow 0\) in uniform areas, leading to a uniform blurring and \(k(i,j) \rightarrow 1\) near edge, leading to little pixel changes.

**Frost filter** The Frost filter ([13]) use also local neighborhood statistics and sliding window to estimate the original image. In this filter, the output pixel of interest is computed as a weighted sum of its neighbors:

\[
\hat{x}(i,j) = \sum_{p \in \eta} m_p \cdot y_p,
\]

where,

\[
m_p = \frac{\exp(-K \cdot C^2_{(i,j)} \cdot d_{(i,j),p})}{\sum_{p \in \eta(i,j)} \exp(-K \cdot C^2_{(i,j)} \cdot d_{(i,j),p})},
\]

\[
d_{(i,j),p} = \sqrt{(i - i_p)^2 + (j - j_p)^2}.
\]

Here the filtering is controlled by \(K \cdot C_{(i,j)}^2\): as it approaches 0 in homogenous regions, the smoothing will be high and tend to be inhibited at edges locations where \(K \cdot C_{(i,j)}^2\) becomes large.
CHAPTER 2. SPECKLE FILTERING

Gamma-MAP filtering  The goal of this filter is to estimate the original image by performing a maximum a posteriori measure on the scene probability density function. Considering that the observed distribution follow a gamma PDF, the expression of the original image PDF becomes [14]:

\[
P_{\Gamma}(\sigma) = \left(\frac{\nu}{\mu}\right)^{\nu} \frac{\sigma^{\nu-1}}{\Gamma(\nu)} \exp\left(-\frac{\nu\sigma}{\mu}\right), \quad (2.9)
\]

where \( \mu \) and \( \nu \) are the mean RCS and order parameter respectively. \( \mu \) and \( \nu \) cannot be measured directly and have to be estimated from the data. The PDF of \( \sigma \) with given intensity \( I \) is given by:

\[
P_{MAP}(\sigma | I) = \left(\frac{L}{\sigma}\right)^L \frac{I^{L-1}}{\Gamma(L)} \exp\left(-\frac{LI}{\sigma}\right) \frac{\nu^{\nu-1}}{\Gamma(\nu)} \exp\left(-\frac{\nu\sigma}{\mu}\right), \quad (2.10)
\]

The corresponding likelihood is then:

\[
\lambda = P_{MAP}(\sigma | I) \cdot P_{\Gamma}(\sigma). \quad (2.11)
\]

One has then to solve the following equation for obtaining an estimation of the noise free image:

\[
\frac{\nu\sigma^2}{\mu} + (L + 1 - \nu)\sigma - LI = 0. \quad (2.12)
\]

Correlated Gamma-MAP filtering  So far, we reconstructed the RCS using single pixels values. This can be improved by introducing a relation between the pixel and its neighborhood via a correlated neighborhood model that selects the smoothest possible noise-free estimation consistent with observed intensity variations in a local region. The probability of observing intensities within neighborhood of the central pixel of RCS \( \sigma \) is:

\[
P_{loc}(I | \sigma) = \left(\frac{\alpha}{\sigma}\right)^{I^{-1}} \frac{\Gamma(\alpha)}{\Gamma(\alpha)} \exp\left(-\frac{\alpha I}{\sigma}\right), \quad (2.13)
\]

where \( \alpha \) is again an order parameter (different from \( \nu \)) that can be estimated from local neighborhood. We can then use this an 8-neighborhood and prior PDF for \( \sigma \) to derive a joint PDF :

\[
P_{CMAP}(\sigma | I_0 \ldots I_8) = P(I_0 | \sigma)P_{\Gamma}(\sigma) \prod_{j=1}^{8} P_{loc}(I_j | \sigma). \quad (2.14)
\]
Therefore, the corresponding MAP estimation becomes:

$$\frac{\nu \sigma_{\text{MAP}}^2}{\mu} + \sigma_{\text{MAP}}(m\alpha + L + 1 - \nu) - LI_0 - m\alpha I = 0.$$  \hspace{1cm} (2.15)

**Structured Gamma-MAP filtering** An enhancement of the local neighborhood model described in [26] is to reduce the size of the window size (3x3) and add some structure-adaptive filtering by using small directional windows of size 3-pixels and applying the Gamma-MAP framework.

![Gamma-MAP estimation windows](image)

**Figure 2.2 — Gamma-MAP estimation windows**

**Discussion** Table 2.1 summarize the different characteristics of the filters regarding criterions Oliver criterions. The main advantage of all the described methods is the fact that they are all based on a local coefficient of variation or RCS estimation, leading to a local sensibility for intra-image intensity changes without additional parameter tuning to retrieve the adequate level of sensibility to changes.

Correlated and structured Gamma-MAP can be used into a nonlinear iterated scheme such as simulated annealing in order to get enhanced results with a large amount of smoothing compared to Lee, Frost, non-iterative Gamma-Map and correlated Gamma-MAP (see [26]).

But, if these methods are performing well, such algorithms have no special behavior when processing edges i.e. when a sufficient variation indicating an edge is encountered, due to the linearity of filtering, no specific processing is done letting edge zone being intact. This is particularly annoying as any following edge detection will be then disturbed by this remaining noise (especially with gamma-MAP). Structured gamma-MAP improve this situation with the use of structural filters.

Another problem lies in the fact that none of the filter is providing some edge enhancement properties. However, even with the structured Gamma-MAP, no significant improvement concerning highly perturbed edges is noticed with an additional loss of small structure details. Yet, this is an important property since such an improvement in contrast could heighten a future segmentation task when using these processed images.
All these filters possess their corresponding multichannel extensions using either multivariate statistical description [29] or linear combination of multi-temporal images sequence as in [15], improving the global speckle reduction. However the most significative point is the introduction of iterated speckle filtering which results in highly smoothed images. It have been used in combination with multi-temporal filter descriptions in order to enhance image quality and speckle smoothing.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Lee</th>
<th>Frost</th>
<th>GMAP filters</th>
<th>GMAP iterated filters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speckle reducing</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>High smoothing properties</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Feature preservation</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Radiometric invariance</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>Edge enhancing</td>
<td>no</td>
<td>no</td>
<td>no</td>
<td>no</td>
</tr>
</tbody>
</table>

Table 2.1 — Filters properties

### 2.2 Methodology

In order to improve the results obtained with the state-of-the-art filters mentioned in the previous section and to combine an efficient behavior for both Oliver criterions and edge enhancing quality, an anisotropic nonlinear diffusion filtering algorithm using multi-temporal images sequences has been developed. In this section, we will see how the intrinsic properties of this technique such as anisotropy and conservation will fulfill the different requirements seen in section 2.1.

First used by Perona in [2] in the case of single optical images, diffusion has been successfully applied to the color images [7] and in multi-spectral case by Acton [5] and Weickert [16]. But, as far as we know, diffusion has not been applied to the multi-temporal images yet. Section 2.2.1 will exhibit the nonlinear diffusion process details for single images whereas section 2.2.2 and 2.2.3 introduces the non-linear anisotropic algorithm for both the mono-image case and multi-temporal sequences as an original tool to filter SAR data sets. Sections 2.2.4 and 2.2.5 will analyze in more details the different aspects and parameters intervening in the diffusion process. Finally, the obtained results will be presented and discussed in section 2.3 and 2.4.

#### 2.2.1 Non-linear diffusion

For a continuous image, diffusion on image $I$ may be enacted by the following partial differential equation (PDE):

$$\frac{\partial I}{\partial t} = \text{div}[c \cdot \nabla I],$$  \hspace{1cm} (2.16)

where $\nabla$ is the gradient, $\text{div}$ is the divergence operator, and $c$, the conduction coefficient is a matrix of diffusion coefficients of the same size as $I$. We can notice
that this equation reduces to the isotropic heat diffusion equation \( I_t = c \cdot \Delta I \) if \( c \) is constant.

In order to keep data consistency and avoid irregular results, the original image is first smoothed with a gaussian circular kernel of fixed size and variance \( \sigma \). This step is also called the regularization step. Our anisotropic PDE thus becomes:

\[
\frac{\partial I}{\partial t} = \text{div}[c \cdot \nabla I],
\]  
(2.17)

with \( I \) being the convolution between \( I \) and the gaussian kernel. Diffusion is achieved by setting \( c \) as a function of the image gradient and of an edge threshold parameter \( \lambda \). By taking \( c = c(\nabla \| I \|, \lambda) \), we allow the diffusion to vary according to the strength of the smoothed image gradient. Therefore, by choosing an appropriate \( c \) function, we can have strong diffusion (regularization) in regions with low gradient (inferior to \( \lambda \)) and low diffusion in strong gradient zones.

Equation 2.17 which holds for continuous images can now be discretized on a square lattice and solved using forward Euler differences, ending up with the following equation:

\[
I_{t+1} = I_t + \nu \cdot dI_t,
\]  
(2.18)

\[
I_{t=0} = I_{\text{original}},
\]  
(2.19)

where \( I_{\text{original}} \) is our discrete original image. \( dI \), the image update, corresponds to the discretization of the divergence operator and \( \nu \) is the discretization scheme step size.

We can summarize the global nonlinear diffusion algorithm in the following way:

**Algorithm 1: nonlinear diffusion process**

<table>
<thead>
<tr>
<th>input</th>
<th>Original image</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters</td>
<td>( c ) function variables (( \lambda, \sigma, \ldots )), stopping time ( t )</td>
</tr>
<tr>
<td>output</td>
<td>Filtered image</td>
</tr>
</tbody>
</table>

\[I^0 = \text{original image}\]

\[\text{for } t = 0 \text{ to time do}\]

\[I'_t = I^t * G(\sigma)\]

\[
\text{grad} = \text{Gradient}(I'_t)\]

\[
\text{diffusion amount} = \text{Cfunction(\text{grad,}\lambda)}\]

\[I^{t+1} = I^t + \text{step size} \cdot dI^t(\text{diffusion amount})\]

\[\text{end}\]

However, if the use of an appropriate \( c \) function can enhance edges contrast and, the filtering process is still stopped at edges locations, leaving them potentially very noisy.

### 2.2.2 Single image anisotropic nonlinear diffusion

With the diffusion algorithm presented in algorithm 1, the problem of edge regularization and enhancing is still unsolved. In order to overcome this situation,
Weickert [16], [18], [19] introduced edge-direction sensitive diffusion. Indeed, the amount of diffusion will no more be a unique scalar value but rather a matrix $D$ (also called diffusion tensor) of values specifying the diffusion importance in several directions. Moreover, this matrix can be designed to take into account the gradient direction for defining diffusion direction. Therefore, at a single pixel location we can choose the smoothing degree along the gradient direction and perpendicular to this direction. We will see now and in section 2.2.3 how the definition of $D$ can allow us to achieve both isotropic filtering in homogenous regions and structure adapted filtering in high gradient regions.

The definition of anisotropic filtering leads to the following PDEs:

$$\frac{\partial I}{\partial t} = \text{div}[D(\nabla I) \cdot \nabla I],$$  \hspace{1cm} (2.20)

$$I^{t+1} = I_t + \nu \cdot dI_t(D).$$ \hspace{1cm} (2.21)

**D construction**

$D$ is set-up using the zero-order structure tensor from the matrix

$$S_0 = \nabla I \otimes \nabla I^T,$$ \hspace{1cm} (2.22)

where $\otimes$ stands for the tensor product. Here, the structure tensor eigenvectors $\overrightarrow{v}_1 = \frac{\nabla I}{\| \nabla I \|}$ and $\overrightarrow{v}_2 \perp \overrightarrow{v}_1$ corresponds to the gradient direction and eigenvalues $(\lambda_1 = \| \nabla I \|, \lambda_2 = 0)$ to the amount of changes in these directions. Hence, $S_0$ can be diagonalized as :

$$S_0 = ( \begin{array}{cc} \overrightarrow{v}_1 & \overrightarrow{v}_2 \end{array} ) \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \begin{array}{c} \overrightarrow{v}_1 \\ \overrightarrow{v}_2 \end{array}.$$ \hspace{1cm} (2.23)

We construct $D$ with the same eigenvectors that $S_0$ but changing the eigenvalues. This means that now, the new eigenvalues should corresponds to diffusion amount in the eigenvector direction. Let $f_1, f_2$ be two continuous functions with $f : \mathbb{R}^2 \rightarrow [0, 1]$. If we choose:

$$\phi_1 = f_1(\lambda_1),$$ \hspace{1cm} (2.24)

$$\phi_2 = f_2(\lambda_2),$$ \hspace{1cm} (2.25)

where $\phi_1$ and $\phi_2$ are the diffusion tensor new eigenvalues, $\phi_1$ will control diffusion along the gradient whereas $\phi_2$ will be in charge of the filtering process perpendicular to this gradient. Therefore, $f_1$ has to satisfy all requirements stated in section 2.2.4 and will be choose to be the same as $c$ in nonlinear diffusion. $\phi_2$ will be fixed to a constant value as we require edges to be smoothed uniformly. This leads to the final computation for $D$.

Algorithm 2 shows the details of the nonlinear anisotropic diffusion method implementation.

### 2.2.3 Multi-temporal anisotropic nonlinear diffusion

The previous method can be extended to vector-valued images in order to take advantage of the redundant information of the multi-temporal sequences...
Algorithm 2: Anisotropic nonlinear diffusion process

\begin{algorithm}
\textbf{input}: Original image
\textbf{parameters}: \( f_1 \) function variables (\( \lambda, \sigma, \ldots \)), stopping time \( t \)
\textbf{output}: Filtered image
\textbf{I}^0 = \text{original image}
\textbf{for} \( t = 0 \) \textbf{to} \text{time} \textbf{do}
\quad \textbf{I}^t = \text{I}^{t-1} \ast G(\sigma)
\quad \text{grad} = \text{Gradient}(\text{I}^t)
\quad \phi_1 = \text{Cfunction(\text{grad},\lambda)}
\quad \phi_2 = 1
\quad D = \begin{pmatrix}
\phi_1 & 0 \\
0 & \phi_2
\end{pmatrix}
\quad \text{I}^{t+1} = \text{I}^t + \text{step size} \cdot d\text{I}^t(D)
\textbf{end}
\end{algorithm}

and improve the filtering quality. As in [5], the difference lies in the diffusion amount computation which is no longer varying with a single image gradient. Indeed, the choice is made to make \( \phi_1 \) value in \( D \) dependent on a gradient measure computed using the whole set of images. Several choices exist for such a measure: one can simply take euclidian norm of the single images or use Acton Median and average based dissimilarity measures. However, it is not clear on how one should combine the particular gradient expression for each image into a global gradient to get optimal results. Therefore, the most natural choice is to use the reliable formulation for gradient computation with vector data stated in [6] which takes the gradient as a two dimensional manifold embedded in \( \mathbb{R}^m \).

Gradient of a multi-image As in [7], let \( f(u_1, u_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^m \) be a multi-valued image with components \( f_i(u_1, u_2) : \mathbb{R}^2 \rightarrow \mathbb{R} \) with \( i \in [1 \ldots m] \). Therefore, each image value at a pixel location is a vector in \( \mathbb{R}^m \).

Hence, since we are interested by the gradient of the image, we can compute a difference between two pixels locations \( P \) and \( Q \) distant of an infinitesimal displacement \( dP \). The difference can be expressed as:

\[ \Delta f(P,Q) = f(Q) - f(P), \]  \( (2.26) \)

with \( P = (u_1, u_2) \) and \( Q = (v_1, v_2) \). Then, we get the differential:

\[ df = \sum_{i=1}^{2} \frac{\partial f}{\partial x_i} dx_i. \]  \( (2.27) \)

The square norm is then:

\[ df^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{\partial f}{\partial x_i} \frac{\partial f}{\partial x_j} dx_i dx_j. \]  \( (2.28) \)

Here, the dot corresponds to the vector scalar product in \( \mathbb{R}^m \). Note that this implies that \( \mathbb{R}^m \) possess a metric; although we will use Euclidian metric, the main results hold for any nonsingular Riemannian metric. The quadratic
CHAPTER 2. SPECKLE FILTERING

form obtained in (2.28) is called the first fundamental form. This form can be expressed into a matrix representation. Indeed, taking \( g_{ij} = \frac{\partial f}{\partial x_i} \cdot \frac{\partial f}{\partial x_j} \) lead to:

\[
df^2 = \left( \begin{array}{cc} dx_1 & dx_2 \\ \end{array} \right) \left( \begin{array}{cc} g_{11} & g_{12} \\ g_{21} & g_{22} \end{array} \right) \left( \begin{array}{c} dx_1 \\ dx_2 \end{array} \right). \tag{2.29}
\]

Practically, in our case, we get for the multi-temporal image sequences the following first fundamental form:

\[
df^2 = \left( \begin{array}{c} dx \\ dy \end{array} \right) \left( \sum_{i=1}^m \nabla I_{i,x}^2 \sum_{i=1}^m \nabla I_{i,x} \nabla I_{i,y} \sum_{i=1}^m \nabla I_{i,x} \nabla I_{i,y} \sum_{i=1}^m \nabla I_{i,x}^2 \right) \left( \begin{array}{c} dy \\ dy \end{array} \right). \tag{2.30}
\]

For a unit vector \( w = (\cos \theta, \sin \theta) \), the first fundamental form, \( df^2(w) \) is the measure of the rate of change in the \( w \) direction. In our case, we notice that \( g_{12} = g_{21} \), therefore, finding the direction of minimum and maximum rate of changes which corresponds then to our vector-valued gradient computation yield to minimizing/maximizing the following equation:

\[
F(\theta) = g_{11} \cos^2 \theta + 2g_{12} \cos \theta \sin \theta + g_{22} \sin^2 \theta. \tag{2.31}
\]

The solution (which corresponds to Eq. 2.30 eigenvectors) is of the form \((\cos \theta_+, \sin \theta_+)\) and \((\cos \theta_-, \sin \theta_-)\) with:

\[
\theta_+ = \frac{1}{2} \arctan \frac{2g_{12}}{g_{11} - g_{22}}, \tag{2.32}
\]

\[
\theta_- = \theta_+ + \frac{\pi}{2}. \tag{2.33}
\]

The maximum and minimum rate of change corresponding to the computed gradient directions can be then extracted from equation 2.30 using simple linear algebra as the matrix eigenvalues \( \lambda_+ \) and \( \lambda_- \) yielding:

\[
\lambda_+ = \frac{g_{11} + g_{22} + \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}, \tag{2.34}
\]

\[
\lambda_- = \frac{g_{11} + g_{22} - \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}. \tag{2.35}
\]

**Diffusion process** Now that the gradient directions and intensity is known for our sequences, one can proceed to the diffusion process which implies the same transformations as in section 2.2.2 for the \( D \) computation. The diffusion equation can be written as:

\[
\frac{\partial I_1}{\partial t} = div[D(\nabla \overrightarrow{I}) \cdot \nabla \overrightarrow{I}],
\]

\[
\frac{\partial I_k}{\partial t} = div[D(\nabla \overrightarrow{I}) \cdot \nabla \overrightarrow{I}],
\]

\[
\vdots
\]

Where \( \overrightarrow{I} \) corresponds to the whole multi-temporal image sequence and \( I_i \) is the \( i^{th} \) image in the sequence. Therefore, each image is filtered using global sequence information, taking into account features from all images. Details about practical implementation can be found in Algorithm 3.
CHAPTER 2. SPECKLE FILTERING

Algorithm 3: Anisotropic Multi-temporal nonlinear diffusion process

<table>
<thead>
<tr>
<th>input</th>
<th>Original image</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameters</td>
<td>$f_1$ function variables ($\lambda$, $\sigma$, ...), stopping time $t$</td>
</tr>
<tr>
<td>output</td>
<td>Filtered image</td>
</tr>
</tbody>
</table>

$I^0 = \text{original image stack}$

for $t = 0$ to time do
  for $i = 1$ to $N$ do
    $I^t_{\sigma,i} = I^t_i \ast G(\sigma)$
    fundamental form $= \text{UpdateFundamentalForm}(\nabla I^t_{\sigma,i})$
  end
  $[\phi_1, \psi_1, \psi_2] = \text{Parameters(fundamental form)}$
  for $i = 1$ to $N$ do
    $I^{t+1}_i = I^t_i + \text{step size} \cdot dI^t_i(D)$
  end
end

2.2.4 Diffusion parameters

c function

Also called diffusivity function, it plays a crucial role of edge detector and is closely related to the edge threshold $\lambda$. With $c : \mathbb{R}^+ \rightarrow [0, 1]$ being non increasing function, it will control the diffusion amount over the image, taking close to 1 values near edges and dropping to 0 in homogenous areas. Such diffusivity function are designed such that the flux magnitude $|\psi(\|\nabla I\|) = \|\nabla I\| \cdot c(\|\nabla I\|)$ is increasing on the interval $[0, \lambda]$ and decreasing (or at least nonincreasing) on $[\lambda, \infty)$. In [2], Perona introduces two $c$ functions which gives similar perceptual results:

$$c(\|\nabla I\|) = \exp(-\|\nabla I\|^2/\lambda^2), \quad (2.37)$$

$$c(\|\nabla I\|) = \frac{1}{1 + (\|\nabla I\|/\lambda)^2}, \quad (2.38)$$

where equation (2.37) privileges high-contrast edges whereas (2.38) privileges wide regions.

Black [17] also studied extensively the design of $c$ and link the Perona diffusivity to the weighting functions of robust statistical estimation. The alternative diffusivity function of equation, developed from the Tukeys biweight, is:

$$c(\|\nabla I\|, \lambda) = \begin{cases} \frac{1}{2} \left[ 1 - \left( \|\nabla I\|/\lambda \right)^2 \right]^2 & \|\nabla I\| \leq \sigma \\ 0 & \text{otherwise} \end{cases} \quad (2.39)$$

Weickert refined the design of $c$ [16] in the following way:

$$c(\|\nabla I\|) = 1 - \exp \left( -\frac{C_m}{\|\nabla I\|/\lambda} \right)^m. \quad (2.40)$$

In Weickert function, the values of $m$ and $C_m$ are chosen in order that the flux $\Phi(\|\nabla I\|) = \|\nabla I\| \cdot c(\|\nabla I\|)$ is increasing in the interval $[0, \lambda]$ and exponentially decreasing in $[\lambda, \infty]$. The user can then choose by varying $\lambda$ the strength of
gradient sensitivity to take into consideration. Here, \(C_m\) can be computed with respect to the value of \(m\) using some simple mathematical processing. The shape of the controlling function and its sensitivity is then ruled by respectively \(m\) and \(\lambda\).

Figure 2.3 shows the three different possible choice for the \(c\) and corresponding flux functions.

\[\text{Figure 2.3 — Diffusivity (left) and flux (right) functions for fixed } \lambda = 2\]

From diffusivity function observation, the Weickert \(c\) formulation combines the theoretical advantages of a diffusivity positive everywhere with a potentially fast decline of \(c\) around \(\lambda\) controlled by a shape parameter \(m\), implying that strong edges are well preserved for a long time period of the diffusion process. It seem to give much more precision than the other two formulations where the diffusivity drops down more slowly. Moreover, the fact that \(c\) is identically equal to zero for \(c > \lambda\) does not satisfy the criteria for mathematical consistency definition of scale spaces stated in [16]. Hence, in order to get the best results, with high sensitivity and low time sensibility, we recommend the use of Weickert function when proceeding to the diffusion filtering process.

\(\lambda\) parameter Several methods are used to estimate the optimal value of the edge threshold parameter \(\lambda\). Perona [2] state that \(\lambda\) can be fixed using local "noise estimator" described by Canny [8]. Black [17] prosed a global method derived from robusts statistic to guess this robust scale using the following:

\[
\lambda = 1.4826 \cdot \text{MAD}(\nabla I),
\]

where MAD stands for Median absolute deviation. However, for scaling purposes, we decided not to have recourse on these estimation and to let the user the possibility to chose his appropriate scale for filtering.

\(\sigma\) parameter This parameter denotes the variance and width of the Gaussian kernel used to pre-smooth the image gradient before the diffusivity function computation. Also called noise scale, its effects have the two advantages of ensuring and to mitigate the discretization artifacts are mitigated. It also have an impact on the diffusion time since high width and variance smoothing will speed-up the regularization process. But if the diffusion is done faster, it is done
at the expense of several image details loss since it will add more blur and hence reducing the gradient discriminant power.

2.2.5 Discretization schemes

Functional diffusion Since the equation presented in Equation 2.16 holds for continuous images., one have to discretize in space and in time the resolution of the diffusion equation in order to apply it to digital images while preserving the consistency and accuracy of the solution.

Here, for both space and time, finite difference methods will be used as they are designed for implementation purposes in the digital images domain. For discretization in time, finite differences approximates \( \frac{\partial I}{\partial t} \) by \( \frac{I^{t+1} - I^t}{\nu} \) where \( \nu \) stands for the discrete time step. The space discretization relies on also on a simple finite difference method to finally get:

\[
\frac{I^{t+1} - I^t}{\nu} = \text{div}[c(\|\nabla I\|) \cdot \nabla I], \\
I^{t+1} = I^t + \nu \cdot \text{div}(c(\|\nabla I\|) \nabla I). 
\] (2.42)

(2.43)

The explicit discretization scheme used in this section is the most straightforward one but requires a small time step \( \nu \) in order to be stable. It as been shown [16] that the stability condition (assuming \( \Delta x = 1 \) and \( c \leq 1 \)) is \( T < \frac{1}{2N} \) with \( N \) being the number of dimensions of the data. This is a severe limitation, implying that more iterations are needed to reach a fixed stopping time \( T \). Therefore, in the testing part we will work with \( \nu = 0.2 \) in order to reduce the number of iterations.

Tensor diffusion The same conditions for time discretization scheme step size holds for tensor diffusion. However, for discretization in space, some additional finite element processing is necessary. Scharr [20] proved that for tensor diffusion, the 8-element discretization stencil was consistent and accurate. If the concerned pixel is \((i,j)\). Then on a square grid, we can extract the space discretization stencil shown in Figure 2.2.5.

![Figure 2.4 — 3x3 discretization stencil](image)
CHAPTER 2. SPECKLE FILTERING

2.3 Results

This section presents the different results obtained with the algorithms described in the previous section. First, results obtained using nonlinear diffusion (NLD) will be shown in section 2.3.1, and then those produced with anisotropic nonlinear diffusion (ANLD) will be presented in section 2.3.2. We will also study the particular differences between the different algorithms and the influence of corresponding parameters for these techniques. Finally, we will compare our method to the one used for the same task within the InfoPACK [1] software.

Estimated number of looks (or ENL) will be used to measure the smoothing effects of despeckling methods. It is defined as $ENL = \frac{\mu^2}{\sigma^2}$, where $\mu$ and $\sigma$ are the mean and the variance of intensity value over a uniform image area. The value of ENL theoretically equals the number of look of an intensity image. The speckle reduction will be quantified by the measure of ENL on two 30x30 homogenous regions (see Fig 2.11(c)).

2.3.1 Nonlinear diffusion

Figure 2.5 and 2.6 show the smoothing for South-Africa and Philippines using NLD. The parameters used in order to get the figures were chosen empirically for best quality results with help of Weickert diffusivity function.

![South-Africa NLD with $\sigma = 2$, $\lambda = 0.01$, $\nu = 0.2$, $T = 100$](image)

Figure 2.5 — South-Africa NLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2, T = 100$

2.3.2 Anisotropic nonlinear diffusion

Figure 2.7 and 2.8 shows the smoothing for South-Africa and Philippines using ANLD. The parameters used in order to get the figures were chosen empirically for best quality results with help of Weickert diffusivity function.
CHAPTER 2. SPECKLE FILTERING

Figure 2.6 — Philippines NLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2$

Figure 2.7 — South-Africa ANLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2, T = 100, \phi_2 = 1$
2.3.3 Influence of parameters

This section illustrate the impact of the different parameters on the diffusion. Figure 2.9(a) and 2.9(b) present the diffusion of South-Africa 22. Jan image using the multi-temporal diffusion scheme with scale parameter (normalized) $\lambda = 0.05$ and $\lambda = 0.1$.

Figure 2.10(a) and 2.10(b) present the diffusion of South-Africa 22. Jan image using the multi-temporal diffusion scheme with respectively edge diffusion $\phi_2 = 0.2$ and $\phi_2 = 0.5$.

Finally, Figure 2.11 gives an objective evaluation of the smoothing quality with respect to time stopping criterion (Fig 2.11(b)) and to the size of the SAR dataset (Fig 2.11(a)).

2.3.4 Method comparison

This section presents the results that can be obtained with the different solutions proposed in section 2.2. Figure 2.12 underline the difference of regularization between NLD (Fig. 2.12(a)) and ANLD (Fig. 2.12(b)) with multi-temporal information.

Figure 2.13 gives the vector-valued despeckling attained with the InfoPACK v 1.2.1 software for South-Africa data set.
Figure 2.9 — South-Africa 22 Jan. ANLD with $\sigma = 2, \nu = 0.2, T = 100, \phi_2 = 1$

Figure 2.10 — South-Africa 22 Jan. ANLD with $\sigma = 2, \nu = 0.2, T = 100$
Figure 2.11 — South-Africa 22 Jan. Estimation of the number of Look
Figure 2.12 — South-Africa 22 Jan. regularization process
2.4 Discussion

It is clear that with NLD, a good filtering amount is achieved in homogenous regions even when single image is used as presented in Figures 2.5. Indeed, due to the algorithm definition, in low gradient zones, filtering will be isotropic and maximal. However, Figure 2.5 and 2.6 shows that when processing image edges, the original image is left unfiltered leading to very irregular zones. Yet, this highly undesirable for further feature extraction.

Conversely, with ANLD, we see that a very good despeckling is attained with the suppression of edge filtering problem for both sets (Fig. 2.7 and 2.8). Moreover, Figure 2.11(a) shows the impact of the number of images taken into account when proceeding to diffusion. Indeed, we see that ENL grows quickly with this number. This is very important since it shows the larger the input data set is, the more efficient will be the despeckling. In addition to the effective ENL improvement. One can also notice that using multi-temporal image gradient will give more importance to redundant structures within the image and therefore will lead to a high blurring effect with less iterations than required with a smaller number of images.

Another crucial parameter is the stopping time criterion $T$. We see in Figure 2.11 that ENL is growing with $T$. Indeed we can observe an improvement of more than 90% in term of ENL after 100 iterations of the algorithm. However, one has to be careful since over time since Weickert diffusivity function will lead to the progressive blurring of edges over the images due to its design. Hence, a tradeoff have to be set up between the value of $T$ and the scale of details we want to preserve. We can notice that some tentative have been made [21] to automatically select an optimal value for $T$.

Figure present the influence of $\lambda$ parameter on the final result. We observe that this parameter is very important for defining the adequate scale of filtering. Choosing $\lambda$ too high will lead an over-blurring situation where most of the...
details are lost. On the contrary, choosing $\lambda$ too small will disable the image smoothing. Therefore, it is very important for the user to choose the right value for $\lambda$ with the tuning constraint that it represents. In our case, $\lambda$ has been chosen empirically to preserve redundant structures in the image sequence.

The effect of $\phi_2$ is also important. Indeed, it enables diffusion to regularize edges. However, the counterpart for such a regularization is the introduction of a rounding effect: the higher is $\phi_2$, the more rounded will be edges. Therefore, user has to make a tradeoff between high accuracy and edge smoothing. Such an effect could be counterbalanced by choosing a small value for $\phi_2$ (typically 0.1 or 0.2) and iterating diffusion during a longer time.

The comparison with InfoPACK v1.2.1 [1] despeckling scheme is interesting. InfoPACK is using the annealed multi-temporal series statistical speckle reducing method described in section 2.1. In Figure 2.13, we observe a large amount of incorrectly filtered patches due to the structural filtering framework which might not perform well when encountering noise related non-coherent structures (i.e. with high gradients in several directions). It follows that even if edge filtering is better than with NLD, it still shows some irregularities that do not appear when using ANLD.

However we can see that in homogenous regions, smoothing appear to be better using NLD. This is explained by the fact that number of small insignificant features are enhanced in their coherence direction at the beginning of iterations with ANLD. Most of these structures are then blurred but some of them can be further enhanced if connected during the iteration process with other structures. This problem could be solved using an hybrid method alternating NLD and ANLD iteration steps.

As a conclusion, we can say that ANLD shows very interesting properties in term of filtering, feature preservation and feature enhancing. Indeed, even if some improvements could be introduced in this original method, it already seem to be very efficient compared to existing algorithms.
In the process of finding homogenous regions in SAR image, if regions mean and variance provide useful information, one has to pay careful attention to the particular zone of transition between distinct regions which is the main error source when proceeding to segmentation.

Therefore, there is a need in performing special treatment to locate and characterize accurately these zones within our image as it is a crucial matter when integrated to a global segmentation algorithm and provide valuable extra information. As it focuses on region dissimilarity, such a process of finding transitions can be assimilated as an edge detection task. This chapter will focus on this problem and its application to the particular case of SAR images. Indeed, such images have their own characteristics that make traditional processing obsolete and need specific edge detector. J. Canny [8] stated that the criterions for an edge detector to be optimal are:

1. Good Detection: the ability to locate and mark all real edges.
2. Good Localization: minimal distance between the detected edge and real edge.
3. Clear Response: only one response per edge.

The first criterion implies that only real edges should be detected, avoiding as much as possible false edge detection. The second criterion adds the constraint that edges should be detected at the right place or with minimum distance from the real edge location. This is extremely important in order not to truncate or incorrectly reconstruct the final regions. The last criterion is set up in order to avoid that a nonsignificant edge satisfying the first two conditions could be detected.

Section 3.1 will describe some classical methods used in edge detection for both SAR and optical images found in the literature. In Section 3.2, we will develop an original and optimal (in the Canny sense) edge detector for multi-temporal SAR images sequences. The results obtained with this new approach will be presented in 3.3 discussed in section 3.4. In chapter 4 will show how one
can use this as input on a region segmentation algorithm to drastically improve the results.

### 3.1 State of the art

This section presents three classical approaches used for finding edges in an image for SAR and optical images. Note that these filters have been chosen for their relevance, accuracy, and degree of use but that various other types of edge detectors can be found in the literature.

**Ratio edge detector** This technique described in [23] assumes that for L-look images, the power received from homogenous area PDF is:

\[ P(x \mid P_0) = \frac{1}{\Gamma(L)} \left( \frac{P_0}{L} \right)^{-Lx^{L-1}} \exp \left( -\frac{xL}{P_0} \right) \]  

(3.1)

Where \( x \) is a random pixel value and \( P_0 \) the mean power of the distributed target. The ratio edge detector is then defined as the ratio of the average of pixels values of two non-overlapping neighborhoods on opposite side of the point. If we take into account two neighborhood of size \( N \), the corresponding PDF for the ratio \( R = x_1/x_2 \) is:

\[ P(R \mid P_1, P_2) = n\frac{\Gamma(2NL)}{\Gamma(NL)^2} \left( \frac{(P_1/P_2)^{NL}}{(R_n + P_1/P_2)^{2NL}} R_n^{NL-1} \right) \]  

(3.2)

In practice, two decision thresholds \( T_1, T_2 \) are fixed. The considered point is assigned to:

1. the edges if \( R < T_1 \) or \( R > T_2 \)
2. the homogeneous class in the other case

**Linear features extraction** This method [22] consists in determining edge magnitude and direction by convolution of an image with a number of edge masks (see Fig. 3.1). Then, the obtained results are fused to provide a global edge information. Another step consisting of thinning and thresholding these edges magnitudes is taken. The resulting edges are then linked based on proximity and orientation criterions. Finally, the linked elements are approximated by piecewise linear segments to refine the edge detection.

**Marr-Hildreth edge detector** From neurophysiological experiments, Marr [24] concluded that object boundaries are the most important cues that link an intensity image with its interpretation. He proposed the use of zero crossings of the second derivative for accurate edge detection. The first derivative of the image function should have extremum at the position corresponding to the edge in the image, so the second derivative should be zero at the same position, however, it is much easier and more precise to find a zero crossing position than an extremum. The operator can be represented by the following:

\[ \text{LoG}(x, y) = (\nabla^2 \cdot G(x, y, \sigma)) \ast f(x, y) \]  

(3.3)

Where \( \text{LoG} \) stands for Laplacian of Gaussian, \( G(x, y, \sigma) \) is a 2D Gaussian and \( \ast \) is the convolution operator. Then edges are detected as \( \text{LoG}(x, y) = 0 \).
**Discussion**  Only the Ratio edge detector is tuned for SAR image processing. However, it is subject to various parameters so that the performance of the ratio edge detector must be discussed as a function of the size of neighborhoods, the number of looks, and the ratio of the mean powers which is not suitable for an automatic edge detection. If the Linear feature extraction seem to provide interesting results, the main difficulty lies in the fact that several filters are needed to be able to detect all edges orientation within an image (see Figure 3.1). Again, this is not a desirable property since confusion may rise between the different orientations filters. The same observation holds for the Ratio edge detector. So As a conclusion, none of these filters are designed to take into account multi-temporal information.

For the Marr-Hildreth filter its main drawback reside in the fact that it is unadapted to SAR speckle noise. Indeed, such a Laplacian of gaussian computing is highly noise-sensitive and is designed for gaussian noise. Therefore, applying such a filter will lead to many false-edges detection.

As a conclusion, besides none of the pre-cited filters have been designed for the use of multi-temporal image sequences, they do not provide sufficiently reliable and accurate way to detect all real edges in an image (as it will be shown in section 3.3).

![Figure 3.1 — directional filters for Ratio and Linear Feature edge detectors](image)

**3.2 Methodology**

The method chosen here is derived from the well-known Canny edge detector [8] and is designed to fulfill the quality criterions previously mentioned. Therefore, this algorithm can be considered as optimal in this sense. Regarding to the particular case of SAR images sequences, the main reasons for choosing Canny detection as model can be stated as:
• Low sensitivity to noise (important for speckle corrupted images)
• High performance for edge detection
• Possible extension to multi-temporal sequences

In opposition to Ratio edge detector (RED), Canny detection has the advantage of not relying on statistical a-priori on regions distributions and to be a purely analytical scheme for detecting contours. Moreover, it is less sensitive to noise than other analytical methods such as the Marr-Hildreth operator. Finally, such a detector is valid for all possible edge orientations and subject to a limited number parameter estimation.

We will first review the technical aspects of the algorithm for single image processing. Then, we will explain how one can apply it to vector-values data sets.

3.2.1 Single image

For single image, the Canny edge detector consist in the sequential execution of five steps. First the image id smoothed to eliminate and noise. It then finds the image gradient to highlight regions with high spatial derivatives. The algorithm then tracks along these regions and suppresses any pixel that is not at the maximum (nonmaximum suppression). The gradient array is now further reduced by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed which ones are insignificant by using two thresholds $T_1$ and $T_2$.

Step 1 : smoothing In order to implement the canny edge detector algorithm, a series of steps must be followed. The first step is to filter out any noise in the original image before trying to locate and detect any edges. Once a suitable mask has been calculated, the Gaussian smoothing can be performed using standard convolution methods. A convolution mask is usually much smaller than the actual image. As a result, the mask is slid over the image, manipulating a square of pixels at a time. The larger the width of the Gaussian mask, the lower is the detector’s sensitivity to noise. The localization error in the detected edges also increases slightly as the Gaussian width is increased.

Step 2 : edge magnitude After smoothing the image and eliminating the noise, the next step is to find the edge strength by taking the gradient of the image. A finite difference operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. Such an operator uses discretization stencils, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows).

Step 3 : gradient direction We now have to find the edge direction using the formula :

$$\theta = \tan^{-1} \frac{G_Y}{G_X} (\Pi)$$

An example for gradient direction results can be found in Figure 3.3(c).
Step 4: nonmaximum suppression  After the edge directions are known, nonmaximum suppression now has to be applied. Nonmaximum suppression is used to trace along the edge in the gradient direction and suppress any pixel value (sets it equal to 0) that is not maximum along the gradient direction. First two pixels values are interpolated on a 3x3 grid along the gradient direction \( \theta \) as shown in Figure 3.2. Using a Bilinear interpolation algorithm, we can then estimate \( i_1 \) and \( i_2 \) values with subpixel resolution (Appendix A). If the center pixel has a gradient magnitude superior (see Figure 3.3(b)) to the two interpolated pixels, it is considered as maximum and marked as an edge. Otherwise, it marked as non edge. As a result, this will give a thin line in the output image and ensure that no multiple response will be given for an existing edge.

![Figure 3.2 — Pixel interpolation for nonmaximum suppression](image)

Step 5: hysteresis thresholding  Finally, hysteresis is used as a means of eliminating streaking. Streaking is the breaking up of an edge contour caused by the operator output fluctuating above and below the threshold. If a single threshold, \( T_1 \) is applied to an image, and an edge has an average strength equal to \( T_1 \), then due to noise, there will be instances where the edge dips below the threshold. Equally it will also extend above the threshold making an edge look like a dashed line. To avoid this, hysteresis uses 2 thresholds, a high (\( T_1 \)) and a low (\( T_2 \)). Any pixel in the image that has a value greater than \( T_1 \) is presumed to be an edge pixel, and is marked as such immediately. Then, any pixels that are connected to this edge pixel and that have a value greater than \( T_2 \) are also selected as edge pixels. Pixels with edge magnitude less than \( T_2 \) are marked...
CHAPTER 3. EDGE DETECTION

Figure 3.3 — Required data for nonmaximum suppression step
immediately as non-edge. Figure 3.4 sketch how hysteresis thresholding can improve the final detection. In the left part, white segments have magnitude above $T_1$, red segments have magnitude between $T_1$ and $T_2$ and yellow segments have a strength below $T_2$.

![Before hysteresis thresholding](image1)

![After hysteresis thresholding](image2)

Figure 3.4 — hysteresis thresholding conditions

### 3.2.2 Multi-temporal images

The main idea is to extend the algorithm presented for single image using the work done in section 2.2.3. Although some partial applications of Canny technique to vector-valued data has been found in [22] (regarding hysteresis thresholding) and [9] (regarding nonmaximum suppression), as far as we know, the whole Canny algorithm has not been applied to multi-image. In the vector-valued data case, the main difference with original Canny lies in the computation of the direction and magnitude of gradient. For multi-image, a global gradient map can be extracted from image sequences using the following

$$
\lambda = \frac{g_{11} + g_{22} + \sqrt{(g_{11} - g_{22})^2 + 4g_{12}^2}}{2}.
$$

(3.4)

And the gradient orientation can be derived as

$$
\theta = \frac{1}{2} \arctan \frac{2g_{12}}{g_{11} - g_{22}}.
$$

(3.5)

Therefore, applying smoothing, nonmaximum suppression and hysteresis thresholding with the gradient magnitude and directions defined by equations 3.4 and 3.5 lead to the definition of a multi-temporal Canny edge detector. It is important to notice that this approach still satisfy the criterions of detection, localization and clear response defining an optimal edge detector which are crucial for the final detection accuracy.
3.3 Results

Figure 3.5 and 3.6 present the results of a vector-valued edge detection for both South-Africa and Philippines with hysteresis thresholds set to $T_1 = 200$ and $T_2 = 30$ for South-Africa and $T_1 = 400$ and $T_2 = 30$ for the latter. These thresholds have been set up experimentally to provide good results. The smoothing parameter has been fixed deliberately to 1 in order to not over-smooth, provided that ANLD step already done that during filtering. This edge detection have been performed on the ANLD filtered time series (see Figure 2.7 and 2.8). For visualization purposes, resulting contours have been superimposed fo filtered images.

Figures 3.7 and 3.8 compares the edge detection obtained using single image methods Canny (Fig. 3.8(a)), Marr-Hildreth (Fig. 3.7(b)) and multi-temporal Canny detection (Fig. 3.7(a))

Finally, in Figure 3.9, we illustrate the impact of filtering by performing a Canny edge detection over the NLD filtered South-Africa 22. Jan image.

![Figure 3.5](image1.png) ![Figure 3.6](image2.png)

(a) 22. Jan    (b) 23 May

Figure 3.5 — Multi-temporal edge detection for South-Africa
CHAPTER 3. EDGE DETECTION

Figure 3.6 — Multi-temporal edge detection for Philippines

(a) Multi-temporal image Canny detection

Figure 3.7 — Canny versus Marr-Hildreth edge detection
CHAPTER 3. EDGE DETECTION

Figure 3.8 — Zoom on upper left corner edge detection for South-Africa 22 January edge map

Figure 3.9 — Edge detection for South-Africa 22 Jan. with NLD filtering
3.4 Discussion

First of all we must notice here that since ground-truth is highly difficult to obtain for multi-temporal data, no objective evaluation could be done for the different edge map. Therefore, all further comments are only visual comments. A future work could surely to find a technique for objective contour map accuracy quantification.

From Figures 3.5 and 3.6, we observe that good results are obtained with the multi-temporal Canny edge detection. Indeed, for both sets, most of the important structures are detected with few false-edge detections. The use of a vector-valued gradient computation helps to reduce considerably the number of non-edge artifact detection. One can also notice that for the Philippines set, global detection accuracy is much lower than for South-Africa. This is due to a higher intra-series variability which leads to a reduction of the discriminant power of the multi-temporal gradient. Yet, when this multi-temporal gradient is enough significant as with the South-Africa data set (lower intra variability), the detection is greatly improved.

From Figure 3.7, we can see the advantage for vector-valued Canny edge detection to Marr-Hildreth detector. Comparing Figure 3.7(a) and 3.7(b) underline the fact that the Marr-Hildreth is much less precise with a lot of false-edge detection. We can also state that the lack for hysteresis thresholding lead to dashed contours which is not the case with Canny detection. Compared to the Single image Canny detection, we see in Figure 3.8, which compares details of the upper-left corner edge maps for single and multi-temporal Canny algorithm, that multi-temporal data clearly adds some valuable information in the task of fields contours retrieval and provide a well defined edge map. Hence, is also reduce the original Canny noise sensitivity, which is a crucial matter for speckle-corrupted SAR images.

However, the edge detection for image sequences overall accuracy is still highly dependant on the speckle filtering. Indeed, Figure 3.9 illustrate this fact. We can observe that when performing a multi-temporal Canny detection over a nonlinear diffused data sets where some residual noise is present near edges, the detection accuracy is highly irregular and hence is not sufficient for a further use in a region discriminating task.

As a conclusion, we can say that the multi-temporal Canny edge detector seems to be very promising since achieving strong accuracy without any post-processing needs, even if it is strongly sensitive to speckle filtering. Therefore, it can be further provided as side information with great confidence to different segmentation algorithms. Moreover, the multi-temporal detector is optimal regarding the Canny criterions as edges well localized and that only one response can be observed for each edges.
Region segmentation

Region segmentation is the process of obtaining well-defined closed regions corresponding to the original image homogenous backscattering coefficient areas (i.e., lands with same crop type). These closed regions will be also denoted by the term *segments*. Human eye and brain are able to group and distinguish objects within an image, using thousand of criterions ranging from color and shape to some prior knowledge about the given scene. Therefore, it is a very challenging task for an algorithm to retrieve a good segmentation. So far, three main types of segmentation can be spotted out:

- Histogram-based
- Edge-based
- Region-based

Histogram based algorithms try to take advantage of the scene distribution to estimate an optimal test and perform a threshold (for instance the Maximum a Posteriori test) operation for segmenting the image. Edge-based techniques are searching for discontinuities in the image, connect the corresponding borderlines together to achieve the segmentation. Finally, region-based methods tries to recursively group pixels with same properties (mean, variance etc) until a satisfactory result is achieved.

All these methods are extensible to multi-temporal series of images resulting in a global map (the cartoon model) with relevant structures appearing in most of images. These extensions can lead to an improved segmentation due to the redundancy introduced by multi-temporal information.

In this chapter, we will first review some basic techniques used in the field of region segmentation. Then, we will see how the combination of edge and region-based methods in addition to multi-temporal series can add some valuable information in the final segmentation accuracy improvement. Then we will present and discuss the results obtained with the derived technique and compare them with those attained with some widely used algorithm.
4.1 State of art

This section presents two widely used algorithms for region segmentation. The first one is watershed segmentation and has been widely used for all images types whereas the second one, known as annealed segmentation, is typically designed for vector-valued SAR image segmentation.

Watershed segmentation In this method [32], the intensity gradient is first computed. The regions start growing from the local minima of this gradient. This is called the zero level. Then the neighboring pixels are merged, increasing the size of regions. If two different regions are going to merge a wall is constructed, preventing them to merge. The segmentation consists of the resulting boundaries. The principal problem with the watershed algorithm is that we get an oversegmentation which can be explained by the presence of spurious local minima. The main issue is then to eliminate the influence of these insignificant minima. The first modification consists in introducing an initial flooding level in the Watershed algorithm. In this way, all the edges which are present in the image produced by the original Watershed algorithm, and which have one or more gradient values below the threshold, will disappear (Fig. 4.1).

\[ P_z(z) = \prod_{n=1}^{N} \prod_{i=1}^{M} \frac{1}{\Gamma(\nu)} \left( \frac{\nu}{\mu_i} \right)^\nu \frac{\nu^\nu - 1}{\nu^\nu} \exp \left( -\frac{\nu}{\mu_i} z_{n,i} \right), \]  

where the intensity \( z_{n,i} \) is defined at pixel \( n \) for image \( i \), \( \mu_i \) is the mean of the considered segment with ENL \( \nu \). Since \( \mu_i \) and \( \nu \) are unknown, Eq. 4.1 cannot be directly evaluated. Therefore, we resort to the generalized ML solution with respect to the unknown parameters whose logarithmic form is given by:

\[ \max_{\mu_i} \ln P_z(z) = N M \ln \left( \frac{(\nu N/e)^\nu}{\Gamma(\nu)} \right) \]

\[ + \sum_{i=1}^{M} \left[ \nu - 1 \right] \sum_{n=1}^{N} \ln z_{n,i} - \nu N \ln \left( \sum_{n=1}^{N} z_{n,i} \right) \]
The optimal image segmentation can then be obtained by jointly maximizing the generalized likelihood function of all the pixels of the $M$ images. If we assume that each of the $M$ images can be decomposed into the union of the same set of $Q$ segments, the optimum segmentation is obtained by choosing that configuration of $Q$ segments with $N_q$ pixels that maximizes the joint generalized likelihood (JGL):

$$JGL = \sum_{q=1}^{Q} \max_{\mu_i} \ln p_z(z^{(q)}),$$

with some development, it yields to:

$$JGL = \text{const} - \nu \sum_{q=1}^{Q} \sum_{i=1}^{M} N_q \ln \left( \frac{1}{N_q} \sum_{n=1}^{N_q} z_{n,i}^{(q)} \right).$$

Eq.4.3 can be maximized by means of a joint simulated annealing technique [12].

Discussion  The main drawback for the described methods is the fact that the output result (even with some variations) is strongly oversegmented. That is why, most of time, these techniques are not sufficient to retrieve direct the natural image boundaries. Hence, one need to perform an additional merging stage in order to reduce the number of segments. Moreover, for annealed segmentation, especially in the vector-valued image case, the computational cost is extremely high as the maximization of the $JGL$ with simulated annealing requires a large number of iterations and evaluation of the latter.

4.2 Methodology

In order to get a suitable segmentation, two methods have been implemented and tested in this study. Section 4.2.1 presents a region-based technique for segmentation using multi-temporal information. Section 4.2.2 introduce an histogram-based statistical methods based on scene distribution estimation and local spatial priors.

4.2.1 Boundary constrained agglomerative segmentation

Found in [3], this segmentation algorithm regards the task of finding regions within the image as a coding task. The approach is based on an hypothetical image coding scheme that decomposes a given image into homogeneous segments and encodes these segments independently of each other. The approach also supports situations in which the encoder and the decoder share some reference information about the image, i.e., we suppose there exists a map indicating some tentative hypotheses about possible segment boundaries (for instance an edge map) to improve segmentation.

Thus, in essence, the entire coding procedure can be then described as a two-part source channel coding with side information. First, the reference edge map will be encoded. Then the pixels values will be transmitted to the decoder. Hence, image segmentation can be stated as the minimization of the expected code length with respect to the given segmentation. In our case, the final region
delineation is achieved by a final region growing step satisfying the minimum code length constraint described.

**Encoding side information** A segment \( \nu \) boundary can be described as a set of consecutive linelets enclosing this segment. Each linelet correspond to an horizontal or vertical step along a single pixel. The encoder defines the boundary by gradually pacing ahead until arriving back to its starting position. Therefore, it successively moves on a virtual grid that is composed of horizontal and vertical linelets between neighboring pixels. While proceeding in this fashion, the encoder must permanently decide among three options: it can turn either clockwise or counterclockwise, or it can move straight on. We regard the outcomes of these decisions as realizations of a random variable obtained by statistically independent sampling \( D \in \{1, 2, 3\} \) (Fig 4.2).

![Figure 4.2 — Encoder scanning scheme (left) and segment overview(right)](image)

Two different cases, which are described by a variable \( R \in \{0, 1\} \) substantially affect the distribution from which \( D \) is drawn. \( R = 0 \) indicates that none of the three options for the current boundary step is marked as a reference edge, whereas \( R = 1 \) indicates that at least one reference edge is among the options. In order to estimate how often the event occurs while iterating along the boundary, we make two assumptions:

- If the encoder has made its previous step along a reference edge, then the probability of having another reference edge in its direct neighborhood is assumed to be approximately equal to 1.
- If the previous step does not coincide with a reference edge, we will assume that the probability of arriving next to a reference edge is equal to a value \( p^R \) which is considerably smaller than 1.

In practice, \( p^R \) is estimated from the fraction of reference edges in the entire image.

Hence, if \( \nu \) is bounded by \( n^B_{\nu} \) linelets with \( n^B_{\nu \cap R} \) of them corresponding to reference edges, we can then conclude that the event \( R = 1 \) occurs approximately \( n^R_{\nu} \) times while iterating along the boundary with:

\[
    n^R_{\nu} = n^B_{\nu \cap R} + (n^B_{\nu} - n^B_{\nu \cap R}) p^R.
\] (4.4)
So, the encoder has estimated probability $p^R_\nu$ chances to select a reference edge for its next step with

$$p^R_\nu = \frac{n^R_{B\cap R}}{n^R_\nu}. \quad (4.5)$$

If $R = 0$, we then assume that any of the three alternatives is equally likely to be chosen as there is no prior information. The average costs for encoding the following steps for are,

$$S_\nu(D \mid R = 0) = \log 3, \quad (4.6)$$

$$S_\nu(D \mid R = 1) = -p^R_\nu \log p^R_\nu - (1 - p^R_\nu) \log \frac{1 - p^R_\nu}{2}. \quad (4.7)$$

Therefore, the description length for encoding a complete boundary for a segment $\nu$ is given by:

$$H_{BND}^\nu = n^R_\nu S_\nu(D \mid R = 1) + (n^R_\nu - n^R_{B\cap R}) S_\nu(D \mid R = 0), \quad (4.8)$$

with $n^R_\nu$ being the number of reference edge along $\nu$. Hence, within whole images, the cost for encoding all segments becomes:

$$H_{BND} = \frac{1}{2} \sum_{\nu \in \gamma} H_{BND}^\nu, \quad (4.9)$$

if $\gamma$ is the set of all image segments and where the $1/2$ factor comes from the fact that each reference edge will be counted twice. With such a code length function, the more there will be reference edges on boundaries, the smaller will be the resulting code length.

**Encoding pixels value** Once the boundary is encoded, one have to encode the pixels values inside each segment to complete the image model. From [12] we know that for SAR images image can be either represented using a Gaussian distribution or a Gamma PDF. Hermes and Buhmann [3] proved that in the case of two-part coding, these two representations are equivalent. Therefore, for the sake of simplicity, we will further assume that data will be described with a Gaussian PDF:

$$P(x_\nu \mid \sigma_\nu, \mu_\nu) = \frac{1}{\sqrt{2\pi}\sigma_\nu} \exp \left( -\frac{(x - \mu_\nu)^2}{2\sigma^2_\nu} \right), \quad (4.10)$$

where $\mu$ and $\sigma$ stand for respectively the segment mean and variance. Its entropy can be derived as:

$$S_{P_\nu} = \text{const} + \frac{1}{2} \log \sigma^2_\nu. \quad (4.11)$$

Dropping the constant term, the remaining costs for encoding pixels becomes:

$$H^{PIX} = \frac{1}{2} \sum_{\nu \in \gamma} n_\nu \log \sigma^2_\nu. \quad (4.12)$$
Cost function  Finally, the complete cost function measures the expected code length according to equations 4.9 and 4.12. We then get:

$$H^{TPC} = \frac{1}{2} \left( \sum_{\nu \in \gamma} n_\nu \log \sigma^2_\nu + H^{BND}_\nu \right).$$  (4.13)

For multi-channel data, the cost function can be extended. If we have $M$ channels and $\sigma_{\nu,i}$ denotes the variance of segment $\nu$ in image $i$, we have

$$H^{TPC}_{MC} = \frac{1}{2} \left( \sum_{\nu \in \gamma} \frac{n_\nu}{d} \sum_{i \leq M} \log \sigma^2_{\nu,i} + H^{BND} \right).$$  (4.14)

Region growing  The final step for achieving the required segmentation is to find the segmentation corresponding to a minimal cost with equation 4.14. To this end, a region growing stage is taken: First, images are divided into one-pixel regions. Then, all the merging costs between 4-connected regions are computed and stored in a list. Then, the merging operation corresponding to the lower cost is chosen. Finally, the merge list is updated. The complete process is summarized in algorithm 4.

<table>
<thead>
<tr>
<th>Algorithm 4: Boundary constraint agglomerative segmentation process</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>input</strong>: Speckle filtered set of images, multi-temporal edge map</td>
</tr>
<tr>
<td><strong>output</strong>: Segmented image</td>
</tr>
<tr>
<td><strong>parameters</strong>: Scale stopping criterion $S$</td>
</tr>
<tr>
<td>Break up image into one pixels regions</td>
</tr>
<tr>
<td>Merge list ← Compute Original merging costs</td>
</tr>
<tr>
<td>for $i = 1$ to $S$ do</td>
</tr>
<tr>
<td>chosen merge operation = $\min_{i,j}$(merge list)</td>
</tr>
<tr>
<td>Update region map</td>
</tr>
<tr>
<td>Update merge list</td>
</tr>
<tr>
<td>end</td>
</tr>
</tbody>
</table>

The only parameter intervening here is the number of merge operations to perform. This is a crucial matter as it defines the final scale for segmentation. Several studies have been made in order to automatically select an optimal number of merge operation [33]-[35]. However, no particular threshold has been set up in this work, we rather decided to generate the whole hierarchy of segmentations so that user can scale it to fit his particular needs in term of quality and accuracy.

4.2.2 Unsupervised statistical segmentation

Let us index $N$ data points to be segmented by $i \in S = \{1, 2, ..., N\}$. Let us furthermore denote each data feature vector (observed data) by $x_i \in \mathbb{R}^d$ (where $d$ is the number of images in the data set). The segmentation process aims to classify the data $S$ into one (hidden) underlying spectral classes, $M$, present in the image, that is, to split the image into homogeneous regions. The family of
random variables $y$ represents these classes, where $y = \{y_1, y_2, ..., y_N\} \in \mathcal{M}^N$ denotes a possible configuration of $Y$.

Now, let us suppose that all the random variables $x_i$ are identically and independently distributed. Then, the probability density function of the intensity pixel can be defined by:

$$P(x_i) = \sum_{\forall y_i \in \mathcal{M}} P(y_i)P(x_i|y_i). \quad (4.15)$$

A density function of this form is called finite mixture (FM) density. $P(y_i)$ is the prior probability of each class $y_i$, also called mixing parameter. $P(x_i|y_i)$ is the probability density function of $x_i$ given the class $y_i$, also named transition or conditional probability. A reasonably assumption for the probability density function of the observing data $x_i$ given the underlying class $y_i$ is the Gaussian function:

$$P(x_i|y_i, \theta_{y_i}) = \frac{1}{\sigma_{y_i}\sqrt{2\pi}} \exp \left[-\frac{(y_i - \mu_{y_i})^2}{2\sigma_{y_i}^2}\right], \quad y_i \in \mathcal{M}. \quad (4.16)$$

where the model parameters $\theta_{y_i} = \{\mu_{y_i}, \sigma_{y_i}\}$ are respectively the mean and standard deviation of the Gaussian function.

Another common assumption for prior PDF taken in the particular case of SAR images is the gamma PDF [12,30]:

$$P(x_i|y_i, \theta_{y_i}) = \frac{L^{L}x_i^{L-1}}{\mu_{y_i}^L}\Gamma(L), \quad \mu_{y_i} \in \mathcal{M}. \quad (4.17)$$

where the model parameters $\theta_{y_i} = \{\mu_{y_i}, L\}$ are respectively the mean of the gamma function and the estimated number of look for the original image. The prior probability encodes, in both cases (gamma and gaussian) the spatial information and it is treated in the following subsection. However, we observed that no noticeable difference between the two distribution models can be observed for image segmentation. Therefore, for the sake of simplicity, we will take the Gaussian distribution model for the rest of the work.

**Spatial distribution model** The simplest spatial distribution model that can be assumed is that where $P(y_i) = \omega_{y_i}$, that is, a prior probability constant for each class. A more evolved method consists in modelling spatial interactions among pixels by means of a Markov Random Field (MRF) model. MRF theory [37] asserts that the total image information can be reduced to a local information according to a neighborhood system. Then, all the sites in the image $S$ are related with a neighborhood system $N = \{N_i, i \in S\}$, where $N_i$ is the set of sites neighboring $i$, with $i \notin N_i$, and $i \in N_j \Rightarrow j \in N_i$. In this work a 8-neighborhood system is considered [4,36].

A random field $y$ is said to be a MRF on $S$ with respect to a neighborhood system $N$ if and only if

$$P(y) > 0, \quad y \in \mathcal{Y}, \text{ and,} \quad (4.18)$$

$$P(y_i|y_{S-\{i\}}) = P(y_i|y_{N_i}), \quad (4.19)$$
where $y_i$ denotes the current estimate at location $i$, and $y_{S-i}$ denotes those at all the locations of $S$ except at $i$. According to the Hammersley-Clifford theorem [38,39], an MRF can be equivalently characterized by a Gibbs distribution,

$$P(y) = Z^{-1} e^{-U(y,\beta)},$$

(4.20)

that has several free parameters to be determined: the normalization factor $Z$, the spatial parameter $\beta$, and the energy function $U(y)$.

The energy function used here is

$$U(i,y_i) = \beta \sum_{j \in N_i} V_{i,j}(y_i, y_j),$$

(4.21)

where

$$V_{i,j}(y_i, y_j) = \begin{cases} 1, & \text{if } y_i = y_j; \\ 0, & \text{otherwise}. \end{cases}$$

(4.22)

This is the well-known as Potts model [41]. Intuitively, the equation above encourages one pixel to be classified as the spectral class that the majority of its neighbors belongs to.

The normalization factor of the Gibbs distribution is theoretically well-defined as

$$Z(U) = \sum_y e^{-U(y,\beta)},$$

(4.23)

but it requires a high computational cost or it is even intractable since the sum among all the possible configurations of $x$ is usually not known [40]. Instead, the conditional probability is normalized by forcing

$$\sum_{\forall y_i \in M} P(y_i|y_{N_i}) = 1.$$  

(4.24)

The spatial parameter $\beta$ controls the relative influence of the spatial prior over the intensity model. $\beta = 0$ corresponds to a uniform distribution over the $M$ possible states so that only on the conditional distribution of the observed data $P(y|x)$ is considered. On the other hand, with $\beta \to \infty$ the spatial information is dominant over the intensity information and one tends to classify all pixels to a single class. The value of $\beta$ is sometimes determined by maximum likelihood estimation although the complexity of the MRF model require the use of approximations [42]. $\beta$ can also be determined empirically as proposed in [43] by gradually increasing its value through the algorithm iterations. In this work, the value of $\beta$ will be chosen empirically to be 2 in order to get a high level of regularization and thus provide coherent regions.

**Maximum a posteriori (MAP) estimation** A twofold problem is being solved in the segmentation step. On one side, a parameter estimation problem (the estimation of the Gaussian distribution parameters $\theta_y$), and, on the other side, the estimation of the underlying spectral classes, that is, segmentation of the image into homogeneous regions ($\hat{y}$). Here, only intensity feature (pixels values) of all images within data sets is used in combination with local spatial priors computed for again for the whole batch of images.
GHMRF implementation follows a scheme that iterates between solving the parameter estimation and the segmentation problems. For Gaussian model, the $\theta_y$ parameters are estimated using an adapted version of the EM algorithm called HMRF-EM [41]:

$$
\hat{P}(t)(k|x_i, \theta) = P(x_i|\hat{\theta}^{(t-1)}_k) \cdot \hat{P}^{(t-1)}(k|k_{N_i}) \sum_{m, \forall m \in M} P(x_i|m, \hat{\theta}^{(t-1)}_m) \hat{P}^{(t-1)}(m|m_{N_i}), (4.25)
$$

$$
\hat{\mu}^{(t)}_k = \sum_{i \in S} \hat{P}(t)(k|x_i, \theta)x_i \sum_{i \in S} \hat{P}^{(t)}(k|x_i, \theta), (4.26)
$$

$$
(\hat{\sigma}^{(t)}_k)^2 = \sum_{i \in S} \hat{P}(t)(k|x_i, \theta)(x_i - \hat{\mu}^{(t)}_k)^2 \sum_{i \in S} \hat{P}^{(t)}(k|x_i, \theta), (4.27)
$$

where $t$ and $k$ denote the $t$-th iteration and a components of the mixture respectively. The GHMRF algorithm stops when the relative difference between estimated means is lower than the stopping criterion $\epsilon$ (in this work it is 0.1%).

The actual segmentation of the image is done by maximizing the a posteriori probability (MAP):

$$
\hat{y} = \arg\max_{y \in Y} \{ P(x|y)P(y) \}. (4.28)
$$

Note that this is a computationally infeasible maximization problem. However, solutions can be computed locally in the image using the marginal probabilities $P(x_i|y_i)P(y_i)$.

Practical implementation Note in Eq. 4.25 that initial values are required by the HMRF-EM: initial estimates of the distribution (Gamma or Gaussian) parameters, $(\hat{\theta}^{(t-1)}_k)$ and an initial segmentation to compute $\hat{P}^{(t-1)}(k|k_{N_i})$. So, we can proceed by first segmenting the image with a Finite Gaussian Mixture Model (FGMM) (with no contextual information) and a MAP estimation. This is solved by the traditional EM algorithm which initial values are obtained by a k-means algorithm. The steps of the algorithm are summarized in algorithm 5.

**Algorithm 5:** The GHMRF segmentation

**Input:** ANLD multi-temporal data set. Number of classes for the segmentation $M$.

**Output:** Segmented image into homogeneous regions.

**Procedure:**
1. K-means algorithm,
2. EM-algorithm to solve a Finite Gaussian Mixture Model,
3. Compute Eq. 4.25, Eq. 4.26 and Eq. 4.27,
4. MAP from step 3,
5. Go back to 3 till convergence ($\epsilon < 0.1\%$).
CHAPTER 4. REGION SEGMENTATION

4.3 Results

This section presents the results of the region segmentation step. This segmentation has been performed over 200x200 pixels wide zones extracted from the ANLD filtered images obtained in chapter 2 (see Fig. 2.7 and 2.8) with 100 iterations and parameters $\lambda = 0.01$, $\sigma = 2$ and $\phi_2 = 1$ for both sets of images.

Section 4.3.1 enlightens the segmentation performed using the unsupervised statistical method described in 4.2.2 whereas section 4.3.2 shows the segmentation level attained with the agglomerative segmentation technique given in section 4.2.1. Finally, section gives an idea about results that can be expected when using the watershed algorithm presented in section 4.1.

4.3.1 Unsupervised statistical segmentation

Figures 4.3(a) to 4.3(c) shows three samples from ANLD filtered South-Africa temporal series and Figures 4.4(a) to 4.4(c) three samples from ANLD filtered Philippines data sets. The GHMRF segmentations represented in Figures 4.3(d) to 4.3(f) and 4.4(d) to 4.4(f) are obtained using whole South-Africa and Philippines sets and three different number of final spectral classes for each batch of images. To produce the results, the following parameters have been chosen in an heuristic way: $\beta = 2$ and $\epsilon = 0.1$ for both sets.

In order to measure the effect of using multi-temporal information when performing classes parameter estimation, Figures 4.5(a), 4.5(c) and 4.5(e) presents the results obtained when using a single image (22 January) to perform a GHMRF segmentation. Here, the same parameters utilized when performing the multi-temporal segmentation described were used. Figures 4.5(b), 4.5(d) and 4.5(f) shows the same results for the South-Africa 23 May image.

4.3.2 Region merging

First of all, Figure 4.6 recall the edge maps obtained in Figure 3.5 and 3.6 with thresholds respectively fixed to $\{T_1 = 200, T_2 = 30\}$ and $\{T_1 = 400, T_2 = 30\}$ that will be used in this section together with the agglomerative segmentation algorithm.

Again, Figures 4.7(a) to 4.7(c) shows three samples from ANLD filtered South-Africa temporal series and Figures 4.8(a) to 4.8(c) three samples from ANLD filtered Philippines data sets. Figures 4.7(d) to 4.7(f) presents the results achieved using the region growing technique with side information at three different scales for South-Africa set. Figures 4.8(d) to 4.8(f) gives the same results attained for the Philippines batch of images.

In figures 4.7(g) to 4.7(i) and 4.8(g) to 4.8(i), the same algorithm as in the preceding paragraph has been tested on both sets without any edge information in order to quantify the impact of side information an the final segmentation accuracy.

Finally, Figures 4.9(a), 4.9(c) and 4.5(e) presents the results obtained when using a single image (22 January) to perform a region growing segmentation with
CHAPTER 4. REGION SEGMENTATION

(a) ANLD Filtering of January
(b) ANLD Filtering of May
(c) ANLD Filtering of December

(d) global segmentation using 6 spectral classes
(e) global segmentation using 7 spectral classes
(f) global segmentation using 8 spectral classes

Figure 4.3 — South-Africa multi-temporal GHMRF segmentation versus filtered images
CHAPTER 4. REGION SEGMENTATION

Figure 4.4 — Philippines GHMRF segmentation versus filtered images
CHAPTER 4. REGION SEGMENTATION

(a) ANLD Filtering of 22 January

(b) ANLD Filtering of 23 May

(c) 22 January segmentation using 6 spectral classes

(d) 23 May segmentation using 6 spectral classes

(e) 22 January segmentation using 7 spectral classes

(f) 23 May segmentation using 7 spectral classes

Figure 4.5 — South-Africa single image GHMRF segmentation versus filtered images
same parameters taken as with multi-temporal segmentation described before. Figures 4.9(b), 4.9(d) and 4.9(f) shows the same results for the South-Africa 23 May image.

![Edge maps for South-Africa and Philippines](image)

*(a) South-Africa  (b) Philippines*

**Figure 4.6** — Edge maps for South-Africa and Philippines

### 4.3.3 Watershed

Figure 4.10 presents the results obtained by running the Matlab implementation of the watershed algorithm on South-Africa 22. January.
CHAPTER 4. REGION SEGMENTATION

(a) ANLD Filtering of 22 January
(b) ANLD Filtering of 23 May
(c) ANLD Filtering of 16 December
(d) global segmentation with 30 segments and edge map
(e) global segmentation with 20 final segments and edge map
(f) global segmentation with 15 final segments and edge map
(g) global segmentation with 30 segments without edge map
(h) global segmentation with 20 final segments without edge map
(i) global segmentation with 15 final segments without edge map

Figure 4.7 — South-Africa multi-temporal region growing segmentation versus filtered images
Chapter 4. Region Segmentation

(a) ANLD Filtering of January
(b) ANLD Filtering of March
(c) ANLD Filtering of December
(d) global segmentation with 30 segments and edge map
(e) global segmentation with 25 final segments and edge map
(f) global segmentation with 20 final segments and edge map
(g) global segmentation with 30 segments without edge map
(h) global segmentation with 25 final segments without edge map
(i) global segmentation with 20 final segments without edge map

Figure 4.8 — Philippines multi-temporal region growing segmentation with edge map versus filtered images
(a) ANLD Filtering of 22 January
(b) ANLD Filtering of 23 May
(c) 22 January segmentation with 15 final segments
(d) 23 May segmentation with 15 final segments
(e) 22 January segmentation with 20 final segments
(f) 23 May segmentation with 20 final segments

Figure 4.9 — South-Africa single image region growing segmentation with edge map versus filtered images
4.4 Discussion

From Figures 4.3 and 4.4 we observe that visually satisfactory results are achieved with the statistical estimation of the different spectral classes using vector-valued data into a Markov random field framework. The main structures are well delineated for both sets. From our point of view, the best results are achieved in Fig. 4.3(e) for South-Africa and Fig. 4.4(e) for Philippines. Moreover, we can notice that the MRF model prevent the formation of isolated pixels since 8-neighborhood is taken into account when proceeding to the MAP test, leading to a set of large homogenous regions.

Figures 4.5(c) to 4.5(f) shows the superiority of the multi-temporal segmentation compared to single image segmentation. Here, it is clear that even with Markov random fields and, pixels presents a sufficient level of variability inside homogenous due to the remaining noise to lower the MAP test discriminant power. In Figure 4.5(c), even if some particular details about the image are retrieved, we observe globally that detected regions no more corresponds homogenous regions. This effect is thus greatly alleviated by the use of vector-valued data which thus perform some regularization process with a better spectral classes parameter estimation.

We also notice that the number of classes selected for segmentation is of high importance. As seen in Figures 4.3(e) and 4.3(f), the statistical segmentation can vary quickly regarding to the total number of classes and introduce changes over the whole image. Therefore, a particular attention has to be taken from user on the choice of this parameter to get an accurate segmentation.

For the region growing segmentation with use of edge maps as side information, we observe also promising results in terms of visual accuracy. For instance in Figures 4.7(e) and 4.8(d), most of the regions have been detected, giving a good visual accuracy. The inherent properties of such a technique is highly suitable for the particular task of finding homogenous regions since it promotes the growth of large regions whose progressively “absorb” insignificant regions.
Figures 4.7(e) and 4.7(h) illustrate the importance of providing complementary edge information. Without such supplementary data, the regions are much more subject to the pixels variations and grows unbounded. Unlike with the statistical method where an iteration scheme heighten the regularization process, the multi-temporal information is here not sufficient. Therefore, the growing process slightly deviate from optimal solution to become much less accurate and provide large inaccurate regions in some cases.

In Figure 4.9(a), 4.9(c) and 4.9(e), we observe that the lack of multi-temporal information clearly lower the segmentation accuracy. The pixels variations here induce formation of large inaccurate regions. Thus, the use of multi-image consequently improve the final segmentation.

By opposition to GHMRF, the number of iteration in algorithm (which lead to the final number of segments) 4 is much less critical. At each iteration it only agglomerate two contiguous regions. Hence, from one step to another, the segmentations will be consistent between each other. It is then more easy for a user to keep track of changes and select an adequate number of segments.

As a conclusion, we can say that both GHMRF and region growing shows promising results in terms of quality, especially when compared to classical watershed segmentation (Fig. 4.10(b)). But, for the GHMRF algorithm a crucial point is to be resolved with the choice of the spectral classes number. For the agglomerative segmentation, an incorrect edge map can also lead to agglomeration of distinct regions (which is not the case with GHMRF) since Eq. 4.8 promote large regions with high number of edges on its boundary. Therefore, adding different “weights” to edges proportional to their magnitudes in Eq. 4.8 could improve the algorithm behavior in this case by preventing some merge operations even if it leads to a larger region with more edges on its boundary.
One of the main interesting property in SAR imagery is its ability to provide at any time and under any weather condition some data about the observed zone. Thus, a natural field of application for such data is the detection and quantification of changes occurring inside a given temporal batch. Two principal types of changes can be enlighten:

1. If this change is natural, for instance due to special weather conditions or to the growth of crops inside fields, images in the data set will be consistent modulo constant radiometric variation.

2. Changes can also be inherent to a human intervention (early harvest of cultures . . . ) or to natural disease (thus changing partial zones radiometric properties). In this case, the SAR time series will no more be consistent.

We have seen in chapters 2, 3 and 4 how multi-temporal data information could improve the speckle filtering process, edge detection and consequently enhanced the final region segmentation step. However, this global improvement is done at the expense of a lose of particular information concerning images. Especially, regarding to the region segmentation, the final output is a global region map common for the whole data set, thus in the first case, this segmentation will be accurate but in the second case, the potential variations appearing only in a minority of images in an image set will not be detected and only highly redundant structures will be retrieved. Hence, there is a real need to refine this global segmentation to retrieve all the possible fluctuations.

As a matter of fact, it has been demonstrated that region segmentation results accuracy drops down when using single image (see Fig. 4.5 and 4.9) with both statistical and region growing methods, thus preventing any reliable exploitation of the obtained regions. That is why, such methods clearly cannot be used for retrieving the regions changes. To this end, in order to fully exploit the promising potential of multi-image segmentation techniques presented before, we decided to take these global region maps obtained (Fig. 4.4(e) and 4.7(e)) as a starting point and process each segment obtained separately using the single image information.
As far as we know, no particular work has been done concerning this particular problem. In fact, change detection techniques [31] cannot be applied here as it only detect all radiometric variations between images, regardless to nature of the variation. Since in our case we are only interested in non-natural changes, one need to introduce different tools for performing such a task.

First, we will present the chosen method and comment about this particular choice. Then we will present the obtained results with this time-varying segmentation. Finally, we will discuss about their accuracy and comment about future work or improvements that could be taken over the described algorithm.

5.1 Methodology

What we essentially aim to do here is to take the multi-temporal segmentation obtained in Figures 4.4(e) and 4.7(e) as a reliable basis. From these homogenous regions obtained in the preceding chapter, we want to detect potential changes corresponding to the second case described before in each single image from a data set. Unlike with complete images where the histogram does not have a sufficient discriminant power and does not show clear differences between classes when using single image (see Fig. 5.1), the single image region histogram is very likely to be multimodal, each node corresponding to a specific subregion.

Provided that region segmentation is reliable, it is then possible to estimate accurately the final number of underlying spectral class composing single-region histograms.

Obviously, the estimation of the spectral class number is the most critical as it conditions the final accuracy for. Moreover, the complete procedure described needs to process a potentially large number of regions. Thus, one needs to estimate automatically and efficiently the number of classes needed for each region.

In [44], Figueiredo introduce a method to solve the problem of determining an optimal finite Gaussian mixture model (FGMM) with unknown number of components for univariate observations. This model will thus provide an optimal class number computation. The technique proposed in [44] perform the following actions. If $k$ is the number of classes, then:

1. Run a FGMM fitting algorithm.

2. Complexity of the model is estimated using the *Minimum Description Length* (MDL) criterion [45]

The whole process is repeated iteratively for different $k$ values, the optimal number of spectral classes corresponding to the model minimizing the MDL criterion.

Once estimated, this number can be used in a region growing region segmentation scheme using ANLD filtered single image information to refine the segmentation for a given region. Indeed, the use of small regions will ensure that no great perturbation in the execution of both algorithms will occur even
not using multi-images. For region growing, no great deviation in the algorithm leading to ineffective regions will be possible.

In section 5.1.1, we will present the fitting algorithm. Section 5.1.2 will give more details about the MDL criterion whereas section 5.1.3 will introduce an iterative distribution agglomeration described scheme in [44] in order to refine improve the fitting stability. Finally, section 5.1.4 will recall the different steps taken and present the global time-varying segmentation algorithm.

5.1.1 FGMM fitting

This process has been already described in section 4.2.2 in the Markov Random Field framework. In this step, regarding implementation constraints we will rather use the non-MRF classical problem. Thus, taking the same assumptions Let us index $n$ data points to be represented using $k$ classes. Let us furthermore denote each data feature vector (observed data) by $x_j \in [0, 255]$ with $j \in [1, n]$. The family of random variables $y_i, i \in [1, k]$ represents the different required spectral classes.

Now, let us suppose that all the random variables $x_j$ are identically and independently distributed. Then, the probability density function of the intensity
pixel can be defined by:

$$P(x_j) = \sum_{i=1}^{k} P(y_i)P(x_j|y_i). \quad (5.1)$$

A density function of this form is called finite mixture (FM) density. $P(y_i)$ is the prior probability of each class $y_i$, also called mixing parameter. $P(x_j|y_i)$ is the probability density function of $x_j$ given the class $y_i$, also named transition or conditional probability. A reasonably assumption for the probability density function of the observing data $x_j$ given the underlying class $y_i$ is the Gaussian function (see Eq. 4.16). If $\theta_{y_i} = \{\mu_{y_i}, \sigma_{y_i}\}$ represents the Gaussian PDF parameters for the $y_i$ spectral class, the fitting process aims to find the maximum likelihood (ML) estimation of $\Theta_k = \{\theta_{y_1}, \ldots, \theta_{y_k}, P(y_1), \ldots, P(y_k)\}$ based on a set of $N$ independent observations $x_{obs} = \{x_1, \ldots, x_n\}$ leads to:

$$\hat{\Theta}_k = \arg \max_{\Theta_k} L(\Theta_k, x_{obs}), \quad (5.2)$$

and $L(\Theta_k, x_{obs})$ is the log-likelihood function:

$$L(\Theta_k, x_{obs}) = \sum_{i=1}^{n} \log \sum_{m=1}^{k} P(y_m)P(x_i|\hat{\theta}_m). \quad (5.3)$$

In general, Equation 5.3 has no closed form solution but it can be approached with the Expectation Maximization (EM) algorithm. This is done by recursively estimating:

$$\omega_{m}^{(i,t)} = \frac{\hat{P}^{(t)}(y_m)P(x_i|\hat{\theta}_m^{(t)})}{\sum_{j=1}^{k} \hat{P}^{(t)}(y_j)P(x_i|\hat{\theta}_j^{(t)})}, \quad (5.4)$$

$$\hat{P}^{(t+1)}(y_m) = \frac{1}{n} \sum_{i=1}^{n} \omega_{m}^{(i,t)}, \quad (5.5)$$

$$\hat{P}_m^{(t+1)} = \frac{\sum_{i=1}^{n} \omega_{m}^{(i,t)} x_i}{\sum_{i=1}^{n} \omega_{m}^{(i,t)}}, \quad (5.6)$$

$$\hat{\sigma}_m^{(t+1)} = \frac{\sum_{i=1}^{n} (x_i - \hat{P}_m^{(t+1)})^2 \omega_{m}^{(i,t)}}{\sum_{i=1}^{n} \omega_{m}^{(i,t)}}, \quad (5.7)$$

where $t$ and $m$ denote the $t$-th iteration and a component of the mixture respectively. The algorithm stops when the relative difference between estimated means is lower than the stopping criterion $\epsilon$ (in this work we will fix it to 0.1%).

### 5.1.2 MDL criterion

It is well known that ML criterion cannot be used to estimate the number of mixture components since the maximized $L$ function is a non-decreasing function of the number of spectral classes.

In EM based approaches, the estimate number $\hat{k}$ of classes can be defined as minimizing some cost function:

$$\hat{k} = \arg \min_k \{C(\hat{\Theta}_k, k), k \in [1, k_{max}]\}, \quad (5.8)$$
where $k_{\text{max}}$ is the spectral classes upper bound. Under this general formulation, we find the MDL criterion [45] in which the cost function is

$$C_{\text{MDL}}(\hat{\Theta}_k; k) = -L(\hat{\Theta}_k, x_i) + \frac{N(k)}{2} \log n,$$  \hspace{1cm} (5.9)

where $N(k)$ is the number of parameters needed to specify a $k$-component mixture. In this case, it reduces to $N(k) = (k-1) + 3k/2$. Therefore, the algorithm for finding optimal number of classes becomes:

---

### Algorithm 6: Finding optimal spectral classes number

**INPUT:** ANLD filtered segmented region, $k_{\text{max}}$ the maximum number of classes  

**OUTPUT:** Estimated number of classes $\hat{k}$.

**PROCEDURE:**

1. for $k = 1$ to $k_{\text{max}}$ do

   1. Compute initial $\hat{\Theta}_1$ with K-means algorithm
   2. EM-algorithm to solve a Finite Gaussian Mixture Model, while $\epsilon > 0.1\%$ do
       3. Compute Eq. 5.4, Eq. 5.5, Eq. 5.6 and Eq. 5.7,
   end while
   4. Compute and store MDL value using Eq. 5.9

end for

5. Compute $\hat{k}$ using 5.8

---

### 5.1.3 Agglomerative Fitting

The main weakness in Algorithm 6 is that EM requires an initial parameter setting $\Theta_k$. As EM only converges to a local maximum of the likelihood function, this parameter is crucial. In this work we will use the well-known K-means algorithm to estimate $\hat{\Theta}_k$. Therefore, initializing with K-means for each value of $k$ when looking for $\hat{k}$ will lead to some uncertainty and can provide different results from an algorithm run to another. Moreover the convergence of the algorithm to the solution is in this case far from being guaranteed.

To overcome this situation, an Agglomerative EM (or AEM) can be used [44]. Here, instead of re-initializing EM with K-means, an iterative scheme described in Algorithm 7 is set up.
Algorithm 7: Finding optimal spectral classes number using AEM

**INPUT:** ANLD filtered segmented region. $k_{max}$ the maximum number of classes

**OUTPUT:** Estimated number of classes $\hat{k}$.

**PROCEDURE:**
1. Set $k = k_{max}$
2. Compute initial $\hat{\Theta}_k^1$ with K-means algorithm
   for $k = k_{max}$ downto 1 do
   3. EM-algorithm to solve a Finite Gaussian Mixture Model,
   while $\epsilon > 0.1\%$ do
   4. Compute Eq. 5.4, Eq. 5.5, Eq. 5.6 and Eq. 5.7,
   end while
   5. Compute and store MDL value using Eq. 5.9
   6. Obtain a $(k-1)$ component mixture “close” to the $k$-component one
   7. Let $\hat{\Theta}^1_{k-1}$ represent this model
   end for
8. Compute $\hat{k}$ using 5.8

The crucial aspect of algorithm 7 is to use a $(k-1)$ component mixture “close” to the $k$-component one. Actually, the first run of EM, due to the large number of component is somewhat insensitive to initialization, hence solving the difficulty of algorithm 6.

Obtaining the $(k-1)$ component mixture in algorithm 7 step 6 is done by merging two components of the $k$-component one that are close and simultaneously less probable. Let be $m_1$ and $m_2$ the selected components. We choose $m_1$ and $m_2$ as:

\[
(m_1, m_2) = \arg\min_{(i,j)} \{ (\hat{P}(y_i) + \hat{P}(y_j))D_s[P(x_{obs}|\hat{\theta}_i), P(x_{obs}|\hat{\theta}_j)] \}, \tag{5.10}
\]

where $D_s[P(x_{obs}|\hat{\theta}_i), P(x_{obs}|\hat{\theta}_j)]$ is the symmetric Kullback-Leibler (KL) divergence [46] which measure dissimilarities between PDFs. For two Gaussian PDFs, it can be expressed in our univariate case as:

\[
D_s[P(x_{obs}|\hat{\theta}_i), P(x_{obs}|\hat{\theta}_j)] = \frac{1}{2} \left( (\sigma_i - \sigma_j)(\sigma_i^{-1} - \sigma_j^{-1}) + \frac{(\mu_i - \mu_j)^2}{\sigma_i^{-1} - \sigma_j^{-1}} \right). \tag{5.11}
\]

Finally, $\hat{\Theta}^1_{k-1}$ is computed as [44]:

\[
P(y_m) = \begin{cases} 
\alpha_m, & m \neq m_1 \\
\alpha_m + \alpha_{m_2}, & m = m_1 
\end{cases}
\]

\[
\mu_m = \begin{cases} 
\frac{\mu_m}{\alpha_m + \alpha_{m_2}}, & m \neq m_1 \\
\frac{\alpha_m \mu_m + \alpha_{m_2} \mu_{m_2}}{\alpha_m + \alpha_{m_2}}, & m = m_1 
\end{cases}
\]

\[
\sigma_m = \begin{cases} 
\frac{\sigma_m}{\alpha_m (\sigma_m + \sigma_{m_2}^2) + \alpha_{m_2} (\sigma_{m_2} + \sigma_{m_2}^2)} - \mu_{m_1}^2, & m \neq m_1 \\
\frac{\alpha_m \sigma_m + \alpha_{m_2} \sigma_{m_2}}{\alpha_m + \alpha_{m_2}} - \mu_{m_1}^2, & m = m_1 
\end{cases}
\]
5.1.4 Time-varying segmentation

Now, we can set up the complete time-varying segmentation algorithm. If $\overline{I}$ is an ANLD filtered set of $k$ images $I_i$ and $i \in [1, k]$. Let each single image $I_i$ be composed of $N$ distinct regions $\{R_1, \ldots, R_N\}$ with $I_i = R_1 \cap R_2 \cap \ldots \cap R_N$, $\forall i$. Then, the time varying segmentation $\overrightarrow{TVS}$ for the whole data set if composed of $k$ different segmentations $TVS_i$, $i \in [1, k]$. Algorithm 8 summarize the proposed technique.

Algorithm 8: Time-varying segmentation

| INPUT: ANLD filtered complete data set. |
| INPUT: Region map set $M = \{R_1, \ldots, R_N\}$ |
| INPUT: $k_{max}$ |
| OUTPUT: Time varying segmentation $\overrightarrow{TVS}$ |

PROEDURE:

for $i=1$ to $k$ do

for $j=1$ to $N$ do

1: Get region $R$ as $I_i \cap R_j$

2: Compute $\hat{k}$ using algorithm 7

3: Segment $R$ using algorithm 4 with fixed scale stopping criterion $S = \text{size}(R) - \hat{k}$

5: $TVS_i = TVS_i \cup R$

end for

end for

Moreover in step 2, an additional criterion has been introduced stating that mixtures where some components probability is below a threshold $T$ cannot be taken into account when proceeding to Eq. 5.8. This criterion has been set up in order to avoid apparition of too small insignificant region. It can be seen as a constraint on the final region size and prevent the formation of small regions corresponding to radiometric variations to noise.

5.2 Results

This section presents the TV segmentations results obtained on three distinct regions of a 200x200 image sets extracted from the South-Africa sequence. For this purpose, the South-Africa ANLD filtered data sets obtained in Figure 2.7 (with the same parameters) have been used together with the multi-temporal segmentation shown in Figure 5.2(b).

For Figures 5.3, 5.4 and 5.5, original, filtered and corresponding TV segmentations have are presented for respectively region 1, 2 and 3 for different time sample from South-Africa data set. The dates taken are not necessarily the same for each region as they were chosen to illustrate the TV particular behavior. Each Figure is organized as follow:
CHAPTER 5. TIME VARYING REGION SEGMENTATION

For the TVS algorithm, we have taken $k_{max} = 8$, $\epsilon = 0.1$. Moreover, as additional parameter, mixtures containing a component $m$ with $p(y_m) < 0.1$ (see Eq. 5.5) have not been taken into account for MDL criterion selection. For the region growing step, a run of single image Canny edge detector have been used to provide an additional edge map as side information.

Figure 5.6 presents achieved results for histogram fitting with automatic model selection using AEM and MDL criterion for region 1, 29 November sample. Corresponding segmentation results are visible in Figure 5.3(i).

\begin{figure}
\centering
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{chosen_regions.png}
\caption{Chosen regions}
\end{subfigure}
\begin{subfigure}{0.4\textwidth}
\includegraphics[width=\textwidth]{segmentation.png}
\caption{15 regions segmentation}
\end{subfigure}
\caption{Studied sub-regions and multi-temporal region growing segmentation}
\end{figure}
CHAPTER 5. TIME VARYING REGION SEGMENTATION

Figure 5.3 — Region 1 TVS
CHAPTER 5. TIME VARYING REGION SEGMENTATION

Figure 5.4 — Region 2 TVS

Figure 5.5 — Region 3 TVS
CHAPTER 5. TIME VARYING REGION SEGMENTATION

5.3 Discussion

From Figures 5.3(l), 5.4(j) and 5.5(e) we can say that FGMM with automatic model selection together with region growing algorithm provide an accurate time varying change detector for the three regions. In such cases, histogram are well defined and provide high discriminant power for model selection (as for instance histogram in Figure 5.6).

For regions where no changes can be observed, the presented solution also give satisfactory results (Fig. 5.3(j), 5.4(k) and 5.5(j)). Again, in such cases, histogram provide an efficient mean to find an optimal clustering for investigated data points.

An interesting detail to notice is that no special process of filtering have been done regarding the specific TVS problem. The algorithm have been done using the multi-temporal ANLD filtering. Hence information have been “spread” across the different images of sequences, reducing the contrast for each image sample. Even under this assumption, promising results can be observed, confirming our choice of diffusion as an efficient filtering process. Such a constatation calls of course for straightforward modification in order to take into account the time consideration by using a single image ANLD filtering before proceeding to TVS. Such a process would be here suitable provided that all redundant structures have been extracted and we are only looking for single image singularities.

However, as seen in Figure 5.5(i), such an estimation of number of spectral classes is highly sensitive to remaining radiometric variations due to noise and to the edge enhancing effect inherent to diffusion. Such an effect, by enhancing some undesirable part of the regions will lead to an over-estimation of the class number. The region growing algorithm can alleviate this drawback during the merging process leading to big accurate regions and extremely small regions for supplementary spectral classes. But sometimes it will lead to an over-segmentation due to the diffusion artifacts. To overcome this situation and increase the accuracy of TVS, the hybrid diffusion algorithm evoked
in Algorithm 2.4 could certainly contribute to substantially heighten the obtained results with well-formed histograms and therefore much less class wrong estimation.

Unfortunately, no ground truth was available to quantify the accuracy of our results and not all the regions have been tested. But even under these assumptions, we can state our TVS algorithm applied to SAR images sequences shows promising results. Actually, most of the possible variations have been tracked and recognized in the considered time sequence for the three regions taken into account. Moreover, as we previously observed, time aspect plays a crucial role. In fact, we assumed that whole images contributed to the time varying segmentation. We also suggested to relax this constraint by using a single image ANLD before running TVS. But, in real world, such abrupt changes as we were looking for are very likely to occur during some time window. Hence, it could add some robustness to our TVS to define small time period during which several images could contribute to TVS. Combined with single image ANLD hybrid scheme, the results which already could be greatly improved. Therefore TVS seem to be an efficient and reliable way to track changes over SAR images with a lot of improvement perspectives.
Conclusion and future work

The main objectives of this project were to provide robust and efficient techniques for multi-temporal SAR image segmentation. To this end, a complete tool-chain have been designed and tested namely:

- Multi-temporal ANLD for speckle filtering
- Multi-temporal edge detector
- Multi-temporal Region segmentation
- Single image time-varying segmentation

It has been demonstrated in this work that for all processing stages, the use of multi-temporal information leads to much more higher performance than when using single image information.

For ANLD diffusion, a quantitative measure of ENL proved the accuracy of the filtering. In this case, the multi-temporal information led to more than 80% of improvement for ENL figure. Moreover, the anisotropy property of such filtering enabled the preservation and enhancement of edge features.

For Edge detection, an original multi-temporal extension of the Canny edge detector has been developed. This new extension clearly outperformed the traditional edge detectors such as single image Canny and Marr-Hildreth algorithms with much less of false edges detection and better behavior for retrieving global image sequences structures.

Again, for the region growing technique, we noticed a significant improvement in accuracy when exploiting image sequences information. As a result, we obtained sets of closed regions corresponding to image sequences main structures (fields, roads, ...) which was not the case when processing single images where segments did not necessarily correspond to image structure.

A direct application for the multi-temporal region segmentation has also been investigated. In order to take advantage of the obtained homogenous region sets, a time-varying segmentation scheme has been set up to detect the non-natural changes occurring in the time sequence. It enabled to determine a precise profile of radiometric changes within time series.
Hence, the obtained multi-temporal segmentation shows that this tool-chain seems to be very effective as satisfactory visual quality has been achieved for all the steps that have been taken. However, due to the fact that ground-truth images are difficult to obtain for multi-temporal data sets, the robustness of this process was not fully demonstrated. Some further work should then be done in this direction by for instance taking the cadastral maps to extract a reference ground truth.

The most interesting constatation surely lies in the fact that even if the proposed segmentation already provide some accurate results, several improvements can still be incorporated to the process. For instance, intra-region smoothing can be enhanced by using an hybrid ANLD algorithm combining NLD and ANLD steps. As said before, such an improvement is very likely to enhance results for all steps. Moreover, the number of user-defined parameters could be reduced to make the diffusion less user-dependant. To this end, the stopping time criterion and scale parameter for ANLD could be determined in an automatic way.

As a conclusion, we can say that the proposed methods are promising and even if some improvements can be taken to enhance performance it can already be used for multi-temporal SAR data processing.
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>SAR acquisition geometry</td>
<td>3</td>
</tr>
<tr>
<td>1.2</td>
<td>Example of SAR backscattering cases</td>
<td>3</td>
</tr>
<tr>
<td>1.3</td>
<td>South-Africa original multi-temporal sequence</td>
<td>4</td>
</tr>
<tr>
<td>1.4</td>
<td>Rice original multi-temporal sequence</td>
<td>5</td>
</tr>
<tr>
<td>1.5</td>
<td>SAR pre-processing chain</td>
<td>5</td>
</tr>
<tr>
<td>2.1</td>
<td>Local neighborhood</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Gamma-MAP estimation windows</td>
<td>10</td>
</tr>
<tr>
<td>2.3</td>
<td>Diffusivity (left) and flux (right) functions for fixed $\lambda = 2$</td>
<td>17</td>
</tr>
<tr>
<td>2.4</td>
<td>$3 \times 3$ discretization stencil</td>
<td>18</td>
</tr>
<tr>
<td>2.5</td>
<td>South-Africa NLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2, T = 100$</td>
<td>19</td>
</tr>
<tr>
<td>2.6</td>
<td>Philippines NLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2$</td>
<td>20</td>
</tr>
<tr>
<td>2.7</td>
<td>South-Africa ANLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2, T = 100, \phi_2 = 1$</td>
<td>20</td>
</tr>
<tr>
<td>2.8</td>
<td>Philippines ANLD with $\sigma = 2, \lambda = 0.01, \nu = 0.2, \phi_2 = 1$</td>
<td>21</td>
</tr>
<tr>
<td>2.9</td>
<td>South-Africa 22 Jan. ANLD with $\sigma = 2, \nu = 0.2, T = 100, \phi_2 = 1$</td>
<td>22</td>
</tr>
<tr>
<td>2.10</td>
<td>South-Africa 22 Jan. ANLD with $\sigma = 2, \nu = 0.2, T = 100$</td>
<td>22</td>
</tr>
<tr>
<td>2.11</td>
<td>South-Africa 22 Jan. Estimation of the number of Look</td>
<td>23</td>
</tr>
<tr>
<td>2.12</td>
<td>South-Africa 22 Jan. regularization process</td>
<td>24</td>
</tr>
<tr>
<td>2.13</td>
<td>South-Africa despeckling with InfoPACK using 100 iterations and cooling parameter = 0.25</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>directional filters for Ratio and Linear Feature edge detectors</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Pixel interpolation for nonmaximum suppression</td>
<td>31</td>
</tr>
<tr>
<td>3.3</td>
<td>Required data for nonmaximum suppression step</td>
<td>32</td>
</tr>
<tr>
<td>3.4</td>
<td>hysteresis thresholding conditions</td>
<td>33</td>
</tr>
<tr>
<td>3.5</td>
<td>Multi-temporal edge detection for South-Africa</td>
<td>34</td>
</tr>
<tr>
<td>3.6</td>
<td>Multi-temporal edge detection for Philippines</td>
<td>35</td>
</tr>
<tr>
<td>3.7</td>
<td>Canny versus Marr-Hildreth edge detection</td>
<td>35</td>
</tr>
<tr>
<td>3.8</td>
<td>Zoom on upper left corner edge detection for South-Africa 22 January edge map</td>
<td>36</td>
</tr>
<tr>
<td>3.9</td>
<td>Edge detection for South-Africa 22 Jan. with NLD filtering</td>
<td>36</td>
</tr>
<tr>
<td>4.1</td>
<td>Watershed segmentation (left) and thresholded watershed (right)</td>
<td>39</td>
</tr>
<tr>
<td>4.2</td>
<td>Encoder scanning scheme (left) and segment overview(right)</td>
<td>41</td>
</tr>
<tr>
<td>4.3</td>
<td>South-Africa multi-temporal GHMRF segmentation versus filtered images</td>
<td>48</td>
</tr>
<tr>
<td>4.4</td>
<td>Philippines GHMRF segmentation versus filtered images</td>
<td>49</td>
</tr>
</tbody>
</table>
List of Figures

4.5 South-Africa single image GHMRF segmentation versus filtered images ............................................. 50
4.6 Edge maps for South-Africa and Philippines .................. 51
4.7 South-Africa multi-temporal region growing segmentation versus filtered images .......................... 52
4.8 Philippines multi-temporal region growing segmentation with edge map versus filtered images ..................... 53
4.9 South-Africa single image region growing segmentation with edge map versus filtered images ..................... 54
4.10 Watershed segmentation over South-Africa ANLD 22 January ... 55

5.1 Whole image histogram versus single region histogram .......... 59
5.2 Studied sub-regions and multi-temporal region growing segmentation ................................................................. 64
5.3 Region 1 TVS .......................................................... 65
5.4 Region 2 TVS .......................................................... 66
5.5 Region 3 TVS .......................................................... 66
5.6 Histogram fitting ....................................................... 67

A.1 Interpolation grid ...................................................... 74
List of Algorithms

1. nonlinear diffusion process ........................................... 12
2. Anisotropic nonlinear diffusion process .......................... 14
3. Anisotropic Multi-temporal nonlinear diffusion process ...... 16
4. Boundary constraint agglomerative segmentation process ...... 43
5. The GHMRF segmentation ............................................. 46
6. Finding optimal spectral classes number .......................... 61
7. Finding optimal spectral classes number using AEM .......... 62
8. Time-varying segmentation ............................................. 63
Bilinear interpolation

Bilinear interpolation is an extension of linear interpolation for interpolating functions of two variables. The key idea is to perform linear interpolation first in one direction, and then in the other direction. In Figure A.1, the four red dots show the data points and the green dot is the point at which we want to interpolate. Enlarge The four red dots show the data points and the green dot is the point at which we want to interpolate.

Suppose that we want to find the value of the unknown function \( f \) at the point \( P = (x, y) \). It is assumed that we know the value of \( f \) at the four points \( Q_{11} = (x_1, y_1), Q_{12} = (x_1, y_2), Q_{21} = (x_2, y_1), \) and \( Q_{22} = (x_2, y_2) \).

We first do linear interpolation in the \( x \)-direction. This yields:

\[
\begin{align*}
f(R_1) & \approx \frac{x - x_2}{x_1 - x_2} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) & \text{where} & \quad R_1 = (x, y_1) \\
f(R_2) & \approx \frac{x - x_2}{x_1 - x_2} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) & \text{where} & \quad R_2 = (x, y_2)
\end{align*}
\]

We proceed by interpolating in the \( y \)-direction.

\[
f(P) \approx \frac{y - y_2}{y_1 - y_2} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2)
\]
This gives us the desired estimate of $f(x, y)$

$$f(x, y) \approx \frac{f(Q_{11})}{(x_1 - x_2)(y_1 - y_2)}(x - x_2)(y - y_2) - \frac{f(Q_{21})}{(x_1 - x_2)(y_1 - y_2)}(x - x_1)(y - y_2)$$

$$- \frac{f(Q_{12})}{(x_1 - x_2)(y_1 - y_2)}(x - x_2)(y - y_1) + \frac{f(Q_{22})}{(x_1 - x_2)(y_1 - y_2)}(x - x_1)(y - y_1).$$

If we choose a coordinate system in which the four points where $f$ is known are $(0, 0), (0, 1), (1, 0), (1, 1)$, then the interpolation formula simplifies to:

$$f(x, y) \approx f(0, 0)(x - 1)(y - 1) - f(1, 0) x(y - 1) - f(0, 1)(x - 1)y + f(1, 1)xy$$

Or equivalently, in matrix operations:

$$f(x, y) \approx \begin{bmatrix} f(0, 0) & f(0, 1) \\ f(1, 0) & f(1, 1) \end{bmatrix} \begin{bmatrix} 1 - x \\ x \end{bmatrix} \begin{bmatrix} 1 - y \\ y \end{bmatrix}$$

Contrary to what the name suggests, the interpolant is not linear. Instead, it is of the form:

$$(a_1x + a_2)(a_3y + a_4)$$

so it is a product of two linear functions. Alternatively, the interpolant can be written as

$$b_1 + b_2x + b_3y + b_4xy$$

In both cases, the number of constants (four) correspond to the number of data points where $f$ is given.

The result of bilinear interpolation is independent of the order of interpolation. If we had first performed the linear interpolation in the $y$-direction and then in the $x$-direction, the resulting approximation would be the same.
Bibliography


[28] "SAR Land Applications Tutorial" available on the web http://www.tiger.esa.int


