Universitat Politècnica de Catalunya
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
Signal Theory and Communications Department

**A DEGREE THESIS**

by Ferran Pérez Gamonal

# Speckle Noise Reduction in PolSAR Images with Binary Partition Tree

Academic Supervisors: Prof. Philippe Salembier Clairon
Prof. Carlos López-Martínez

In partial fulfilment of the requirements for the degree in
*Audiovisual Systems Engineering*

Barcelona January 2017

This page is intentionally left blank.

# Abstract

In some remote sensing applications such as PolSAR (Polarimetric Synthetic Aperture Radar), the use of Binary Partition Trees (BPTs) for Speckle Noise filtering schemes is currently gaining interest. In this thesis, a new approach using this representation is investigated: branch filtering. This approach consists in searching for each leaf its ancestors and selecting the one that best represents it, that is, the one that yields the lower error. A potentiality assessment is done to evaluate the margin of improvement that new techniques based on this approach may provide and describe the basic specifications of the algorithms based on it.

After that, different new techniques are developed, analysed and compared against the State-of-the-Art. We point out the main strengths and weaknesses of each technique. Our main goal is to understand the behaviour of the filtered data along the BPT branch and interpret how this information can be used in the future for speckle noise reduction in PolSAR images.

Finally some conclusions are drawn and some possible future lines of work are exposed and commented.

**Keywords:** PolSAR image processing, Speckle Noise Reduction, Binary Partition Tree, Support Vector Machine.

# Resum

En algunes aplicacions de teledetecció com Polarimetric SAR, l'ús d'Arbres de Partició Binària està guanyant interès. En aquesta tesi, s'investiga un nou mètode que utilitza aquesta representació: filtratge per branques. Aquest mètode consisteix en buscar per cada fulla els seus avantpassats i seleccionar el millor node, és a dir, el que dóni un error menor. Es duu a terme un anàlisi de potencialitat per evaluar el marge de millora que noves tècniques basades en aquest mètode podrien aportar i es descriuen els principis bàsics dels algorismes que s'hi basen.

Després, es desenvolupen diverses tècniques i es comparen amb les de l'estat de l'art. Destaquem les principals fortalesses i feblesses de cada tècnica. El nostre principal objectiu ès entendre el comportament de les dades filtrades al llarg de la branca del BPT i interpretar com podem utilitzar aquesta informació en un futur per la reducció del soroll especular (speckle) en imatges PolSAR.

Per últim s'exposen algunes conclusions i es proposen i comenten possibles noves línies de treball.

**Paraules clau:** Processat d'imatges PolSAR, Reducció de soroll granular (Speckle), Arbres de Partició Binària (BPT's), Màquines de Suport Vectorial (SVM's).

# Resumen

En algunas aplicaciones de teledetección como Polarimetric SAR, el uso de Árboles de Decisión Binarios está ganando interés. En esta tésis se incorpora un nuevo método que usa esta representación: filtraje por ramas. Este método consiste en buscar para cada hoja sus antepasados y seleccionar el mejor nodo como el que de el menor error. Se lleva a cabo un análisis de potencialidad para evaluar el márgen de mejora que nuevas técnicas basadas en este método podrían proporcionar y se describen los principios básicos de los algoritmos que se basan en él.

Tras esto, se desenvolupan distintas técnicas y se comparan con las del estado del arte. De cada técnica, destacamos sus principales fortalezas y debilidades. Nuestro objetivo principal es entender el comportamiento de los datos filtrados a lo largo de la rama del BPT e interpretar como podemos usar esta información en un futuro para la reducción de ruido especular (speckle) en imágenes PolSAR.

Por último, se exponen algunas conclusiones y se presentan y comentan algunas posibles líneas de trabajo futuras.

**Palabras clave:** Procesado de imágenes PolSAR, Reducción de ruido granular (Speckle), Árboles de Partición Binaria (BPT's), Máquinas de Soporte Vectorial (SVM's).

# Acknowledgement

I would like to show my gratitude to Prof. Philippe Salembier, my project supervisor, for his help and support, suggestions and the confidence he placed in me to fulfil this project. His deep knowledge in signal processing help me understand the details of the covered techniques.

I would also like to thank Prof. Carlos López, my project supervisor, for his help and advice all throughout the project. His deep knowledge about PolSAR technologies helped me to better understand this technology and thus the images I had to work with.

Furthermore, I want to give my thanks to Cristian Cuadrado and Ismael Vivó, colleagues working with PolSAR image processing related projects, for their help and feedback on my project. I hope I could help you with your projects as well.

I would also like to take a moment to appreciate the help given by the technical team in the D5 building, Albert Gil and Josep Pujal, for their constant backup in technical inquiries that arise throughout the development of this project.

Last but not least, I want to thank my family and my beloved partner, along with my friends for their encouragement all through this project.

### Revision history and approval record

| Revision | Date | Purpose |
|---|---|---|
| 0 | 30/09/2016 | Document creation |
| 1 | 24/12/2016 | Document revision |
| 2 | 06/01/2017 | Document revision |
| 3 | 11/01/2017 | Document revision |
| 4 | 13/01/2017 | Document revision |

### Document distribution list

| Name | E-mail |
|---|---|
| Ferran Pérez Gamonal | ferran.perez@alu-etsetb.upc.edu |
| Philippe Salembier Clairon | philippe.salembier@upc.edu |
| Carlos López-Martínez | carlos.lopez@tsc.upc.edu |

| Written by: | | Reviewed and approved by: | |
|---|---|---|---|
| Date | 13/01/2017 | Date | 17/01/2017 |
| Name | Ferran Pérez | Name | Philippe Salembier<br>Carlos López |
| Position | Project author | Position | Project supervisor |

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

This chapter provides a general overview of the project, the main goals to be achieved and a general workplan showing the project's organization.

## 1.1 General Overview

In some remote sensing applications such as PolSAR (Polarimetric Synthetic Aperture Radar), the use of Binary Partition Trees (BPTs) is currently gaining interest[1], [2]. One of its most powerful properties is its inherent multi-scale nature that gives additional information related to the image content at different detail levels, where each node represents a unique region in the image.

The purpose of this project is to study new techniques relaying on the use of Binary Partition Tree's structure for speckle noise reduction in PolSAR images. In particular, the potential of a new approach defined as the improvement of dB's of the relative error between the ground truth and the processed image is evaluated. This new approach is called branch filtering which exploits the multi-level structure of the BPT to select the node that bests approximates a leaf ground truth' value. A deep analysis is done to understand the potentiality of such an approach, the main issues involved and the important notions that could help in designing a robust and efficient speckle noise reduction techniques based on BPT branch filtering.

## 1.2 Project Goals

The project main goals are:

1.- Construction and analysis of BPT structures adapted to PolSAR images.

2.- Analysis of speckle noise reduction strategies based on a new approach for BPT speckle noise reduction in PolSAR images, branch filtering.

3.- Analyse and evaluate the developed techniques performance and compare them to the State-of-the-Art previously analysed.

4.- Draw some conclusions and expose some possible future lines of work.

## 1.3 Workplan

The purpose of this section, as stated above, is to describe the project's organization which is divided into seven chapters.

In this first chapter we have introduced the basic aspects of the project before getting into further details.

The second chapter is devoted to explaining the basics of PolSAR remote sensing technique and speckle noise filtering, the Binary Partition Tree representation and the State-of-the-Art speckle noise filtering techniques, stating their main strengths and weaknesses.

On the third chapter, the analysis made on the BPTs potentiality for speckle noise reduction is shown: focusing on the potential improvement that branch filtering may provide. In order to achieve this, different approaches are considered to search for the best procedure to obtain the largest margin of improvement.

The forth chapter is devoted to the proposed techniques based on the extraction of a partition from the BPT. In this chapter, we discuss its advantages and disadvantages and we draw some conclusions.

On the fifth chapter, Machine Learning techniques are considered as an alternative to the previous section. For our case and as a first approach, the Support Vector Machine (SVM) technique trained with a set of descriptors for the leaves and nodes, is used.

The sixth chapter proposes some improvements to the previous lines of work in order to improve the performance of such methods.

Finally, on the seventh chapter, the final project conclusions are drawn and some possible feature lines of work are considered.

Regarding the time organization, it must be stated that it has been successfully followed as it was planned in the *Project Critical Review* document delivered on the 24th of November, 2016[3].

# Chapter 2

# Introduction to PolSAR technology and Speckle Noise Filtering Techniques

This chapter gives an introduction to the concepts of Polarimetric Synthetic Aperture Radar, Binary Partition Tree and State-of-the-art Speckle Noise Filtering Techniques, including experimental tests.

## 2.1 Definition of PolSAR systems

SAR systems[4] are a type of active (they provide their own light source) remote sensing imaging sensor capable of obtaining high resolution information about the scene. It measures the complex reflectivity of the scene at the microwave region of the spectrum, being more sensitive to different target properties (geometrical and those related with electrical properties) than other methods while not being affected by most atmospheric effects.
PolSAR systems are multidimensional SAR systems that can obtain auxiliary information about the scene by exploiting the vectorial nature of the electromagnetic waves radiated by using different polarization states for transmission and reception in order to obtain various SAR images or channels.

## 2.2 PolSAR model

As we described above, PolSAR uses multiple wave polarizations in order to acquire additional information about the scene. In this section, the mathematical model behind it will be presented.

### Scattering matrix model

When a PolSAR wave is radiated towards the scene and encounters an object, part of its energy gets reflected back to the radar receiver and the rest is absorbed by the target. The polarization of the scattered wave will depend on the properties of the target hit.

#### Scattering Polarimetry

Assuming that the same antenna is used for both transmission and reception or that two are located at the same spot (*monostatic* case), a convention for the Cartesian coordinate system for transmitted and received waves may be defined. Normally, the transmitted wave coordinates are defined with respect to the transmitted antenna and the target line, whereas the received wave coordinates are specified by the line defined by the receiver antenna and the target, with the origin located over the target [1].

To simplify this problem, the electric field vector is described by two orthogonal polarization states that are assumed to be the lineal horizontal $h$ and vertical $v$, conforming the basis $\{h, v\}$. Both states are related through the scattering matrix:

$$\begin{bmatrix} E_h^s \\ E_v^s \end{bmatrix} = \frac{exp(-jkr)}{r} \begin{bmatrix} S_{hh} & S_{hv} \\ S_{vh} & S_{vv} \end{bmatrix} \begin{bmatrix} E_h^i \\ E_v^i \end{bmatrix} \tag{2.1}$$

where $E$ denotes the electric field, $r$ is the distance from the receiver antenna to the target. $\underline{\underline{S}}$ is the scattering matrix that relates incident and scattered waves. The convention between both waves (assuming the Backward Scattering Alignment, BSA) is: $h_s = -h_i$, $v_s = v_i$ and $\underline{k_s} = -\underline{k_i}$.

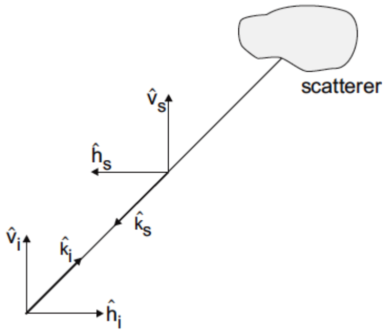


Figure 2.1: BSA convention for incident and scattered waves.

### Distributed Scattering

The former reasoning, though, can only be applied to point scatters which is not the case of PolSAR images. To solve this, *distributed scattering* assumes that a wide number of targets are present at each resolution cell or fixed so we can coherently combine their individual responses. Considering that, the scattering matrix $\underline{\underline{S}}$ is a random variable and the information is extracted from its statistics. $\underline{\underline{S}}$ is vectorized as the *scattering vector* $\underline{k}$ to simplify its characterization:

$$\underline{k_{3P}} = \frac{1}{\sqrt{2}} \left[S_{hh} + S_{vv}, S_{hh} - S_{vv}, 2S_{hv}\right]^T \tag{2.2}$$

where we are using the Pauli's matrix basis and consider $S_{hv} = S_{vh}$ (monostatic case), obtaining a more compact expression. The Pauli basis is used because its scattering vector has components that can be easily related with the physical scattering mechanisms.

Similarly, the lexicographic scattering vector can be defined as:

$$\underline{k_{3L}} = [S_{hh}, \sqrt{2}S_{hv}, S_{vv}]^T \tag{2.3}$$

Finally, a PolSAR system measures all the elements of the scattering matrix for each resolution cell, meaning that for transmission and reception, $\underline{\underline{S}}$ can be seen as 4 individual SAR images corresponding to a combination of polarization states. It can be proved that each of these images will follow a zero-mean complex Gaussian distribution and the scattering vector $\underline{k}$ will follow a multidimensional zero-mean complex Gaussian distribution, identified by its covariance matrix $\underline{\underline{C}}$:

$$p_k(k) = \frac{1}{\pi^p |C|} exp(-k^H C^{-1} k) \tag{2.4}$$

where $p$ denotes the dimension of the scattering vector and $^H$ states for hermitian transpose of complex arrays. This distribution may be referred to as $N(0, \underline{\underline{C}})$. For the monostatic case $\underline{\underline{C}}$ is defined as:

$$\underline{\underline{C}} = E\{\underline{k_{3L}k_{3L}}^H\} = \begin{bmatrix} E\{S_{hh}S_{hh}^*\} & \sqrt{2}E\{S_{hh}S_{hv}^*\} & E\{S_{hh}S_{vv}^*\} \\ \sqrt{2}E\{S_{hv}S_{hh}^*\} & E\{S_{hv}S_{hv}^*\} & \sqrt{2}E\{S_{hv}S_{vv}^*\} \\ E\{S_{vv}S_{hh}^*\} & \sqrt{2}E\{S_{vv}S_{hv}^*\} & E\{S_{vv}S_{vv}^*\} \end{bmatrix} \tag{2.5}$$

where $L$ denotes the Lexicographic matrix basis. Equivalently, a coherence matrix $\underline{\underline{T}}$ is defined as the covariance matrix with Pauli's basis and both are related as:

$$\underline{\underline{T}} = E\{\underline{k_{3p}k_{3p}}^H\} \text{ and } \underline{\underline{T}} = \frac{1}{2}\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 0 & \sqrt{2} & 0 \end{bmatrix} \underline{\underline{C}} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & \sqrt{2} \\ 1 & -1 & 0 \end{bmatrix} \tag{2.6}$$

The retrieved power at each polarization state is stored in the diagonal elements; information about the correlation among the different elements is kept in the off-diagonal members.

### Covariance matrix in practice

In practice, the ideal covariance $\underline{\underline{C}}$ or coherence matrix has to be estimated from the PolSAR data available by averaging the matrices of $n$ pixels, known as the *multilook*. The estimated covariance matrix (a.k.a: sample covariance matrix) is denoted as $\underline{\underline{Z}}$:

$$\underline{\underline{Z}} = \langle \underline{k}\underline{k}^H \rangle_n = \frac{1}{n}\sum_{i=1}^{n} \underline{k_i k_i}^H \tag{2.7}$$

A better estimation may be produced by incrementing the number of averaged pixels $n$. In this case, the spatial resolution is not reduced as much as it would if $n$ was smaller. This evidences that the spatial averaging acts as a low pass filter.
Moreover, sample averaging only makes sense with homogeneous regions, failing when we analyse bigger regions (strongly heterogeneous). Therefore a trade-off between the amount of speckle filtering achieved and the overall resolution loss will be present.

Lastly, assuming $\underline{k}$ as defined above, the PDF (Probability Density Function) of the covariance estimation matrix $\underline{\underline{Z}}$ can be expressed as a complex Wishart distribution, characterized by the covariance matrix $\underline{\underline{C}}$ and the number of averaged pixels $n$ and is noted as $W(\underline{\underline{C}}, n)$:

$$p_{\underline{\underline{Z}}}(\underline{\underline{Z}}) = \frac{n^{pn}|\underline{\underline{Z}}|^{n-p}}{|\underline{\underline{C}}|^n \tilde{\Gamma}_p} etr(-n\underline{\underline{C}}^{-1}\underline{\underline{Z}}) \text{ where: } \tilde{\Gamma}_p = \pi^{\frac{1}{2}p(p-1)} \prod_{i=1}^{p} \Gamma(n-i+1) \tag{2.8}$$

where $etr(X)$ is the exponential of the matrix trace, $p$ refers to the vector $\underline{k}$ dimension and $\Gamma$ is the gamma function.
The previous expression only applies to full rank matrices. As a consequence, this distribution is only valid for $n \geq p$, requiring a minimum of $p$ averaged and different samples.

## 2.3 Speckle Noise Characterization

The speckle noise can be described as a multiplicative noise over the single channel SAR intensity image, affecting each element of the covariance estimation matrix $\underline{\underline{Z}}$. Although the speckle noise is not a random process, as it is an electromagnetic measure of the interactions with all the individual targets, it is interpreted as a random process due to the complexity of the reflection process.

In fact, as stated in [1], after several derivations (for the case of PolSAR, using a pair of SAR images), we can find a more complete expression for each pair of elements on the multi-look covariance matrix $\underline{\underline{Z}}$:

$$S_l S_q^* = \underbrace{\phi N_c \bar{Z}_n n_m exp(j\varphi_x)}_{\text{Multiplicative Term}} + \underbrace{\phi(|\rho| - N_c\bar{Z}_n)exp(j\varphi_x) + \phi(n_{ar} + jn_{ai})}_{\text{Multiplicative Term}} \tag{2.9}$$

where $\phi = E\{|S_l|^2\}$, $N$ refers to the speckle noise, $l$ refers to the polarization state index of the scattering vector $k$, $\bar{Z}_n$ is the expected value of amplitude normalized by the hermitian product, obtained for the case of $\phi = 1$, $\rho$ is the complex correlation coefficient between images, $n_m$ is a multiplicative speckle term and the additive components $n_{ar}$ and $n_{ai}$, and $\varphi_x$ and $\varphi_y$ are phase terms.
Assuming $l = q$ and therefore $|\rho| = 1$ and $\phi_x = 0$ radians (a diagonal covariance matrix, often not true for SAR images), we obtain a expression consistent with the multiplicative model defined before:

$$S_l S_q^* = |S_l|^2 = \phi n_m \tag{2.10}$$

Once the PolSAR model has been defined, the BPT can be introduced and its adaptation to treat PolSAR data, exposed.

## 2.4 Binary Partition Tree

In this section, the binary partition tree representation is presented and described both for a generic case and for our specific case, where we want to represent and process PolSAR imagery.

### BPT representation: generic case

The BPT [5] is a region-based representation with a hierarchical structure: the leaves represent regions that belong to an initial partition and are merged to obtain their parent nodes up to the root node, that represents the whole image support. Smaller regions are found near the leaves, whereas bigger regions are closer to the root (e.g: multi-scale representation).
The merging process is based on a dissimilarity measure and is done following a divide and conquer algorithm strategy (recursively dividing the problem into smaller parts until they are tractable).

To process images in general, two main steps are usually followed:

1) *Tree Construction*: a bottom-up approach is followed, starting from the individual data elements, two adjacent regions are merged to generate a bigger one at each step, until the root node is generated.
   This step is the most computationally expensive, but it only has to be performed once per image as it is independent of the application at hand.

2) *Tree Pruning*: once constructed, the BPT is exploited to obtain meaningful regions for a particular purpose (application dependant). This process is a simplification of the tree that prunes some branches that may not hold meaningful regions, keeping the most useful ones.
   Its structure is not altered during this process as we are only interested in the leaves of the pruned tree, allowing the same BPT to be pruned several times for different applications.
   Finally, to construct a valid BPT representation, the following restrictions have to be applied: each element of the former BPT has to be assigned to a pruned region. In other words, the whole set of pruned regions must cover the whole image; each element of the pruned region can only belong to a single region, i.e: the pruned regions are disjoint.

A simple BPT representation is attached below. This tree has 5 leaves (A,B,C,D and E). The root node I has 2 sons, H and G. The merging sequence, from the initial set of regions A-E can be defined as: $children_1 \cup children_2 = parent_{12}$. For the BPT provided below, that would yield

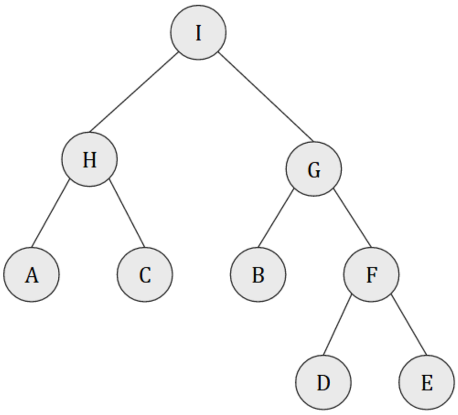


Figure 2.2: Example of a simple BPT representation

the following sequence: $[A \cup C = H,\ D \cup E = F,\ B \cup F = G,\ H \cup G = I]$.

### BPT for PolSAR image processing

In order to use the BPT to represent and process a PolSAR image[6], some key concepts have to be defined for this particular case.

#### PolSAR image connectivity

The region connectivity must be defined to generate the RAG (Region Adjacency Graph) needed to construct the BPT. There are 2 typical choices for pixel connectivity, *4-connectivity* or *8-connectivity*. The first is less computationally expensive but it does not consider the diagonal pixels as adjacent, being unable to represent thin diagonal structures that may belong to a single image region. For PolSAR images, the 8-connectivity scheme will be employed to escape this limitation, as physical structures may be present at any orientation.

#### Region model

The model assumed is the one characterized by the covariance matrix . Aside from those commented before, this model has another limitation: it results into singular matrices when it is directly computed for each pixel of the original image. In practice, this affects all regions with a number of pixels $n_A$ smaller than the sample covariance matrix size $p$. As a consequence, an initial filtering of the data is needed for some applications in order to work with full-rank matrices.

#### Dissimilarity measures

The dissimilarity measure defines the tree merging sequence. There are two groups of measurements[1]: those that use full covariance/coherence matrix and those that just use the diagonal elements. The former ones are more complex but keep some useful information regarding the off-diagonal elements of the matrix, being sensitive to the full polarimetric information.
For this thesis, and if not indicated otherwise, the full *Geodesic dissimilarity* is employed to construct each BPT.

## 2.5 State-of-the-art Speckle Noise reduction techniques

First, some of the most extensively used State-of-the-art (SOTA) techniques will be analysed from a theoretical point of view, pointing out its main strengths and weaknesses. Then, results obtained by each technique (including the main 2 SOTA BPT approaches) over a synthetic dataset will be shown and discussed at the end of this section.

### 2.5.1 Multilook/Boxcar Filter

The multilook/Boxcar filter makes the assumption of local stationarity around the given pixel and averages the $n$ pixels belonging to its $NxN$ neighbourhood defined by a rectangular window. It is the simplest and fastest speckle filter and, moreover, corresponds to the Maximum Likelihood Estimation (MLE) of the covariance matrix $\underline{\underline{C}}$, introducing no bias nor distortion[7].

Despite this, the assumption of local stationarity does not hold over some areas, specially with point scatters and regions near contours, where the Boxcar may yield bad results due to the mixture of heterogeneous samples. When this happens, the behaviour of the filter can be related with that of a low pass filter; reducing the spatial resolution and generating a blurred image. This non-desirable effect can be noticeable on strong point targets with high energy, as they appear enlarged in proportion to the processing window size.

### 2.5.2 Refined Lee Filter

The Refined Lee [8] rejects the local stationary hypothesis and is proposed as an adaptive filter that improves the precision of the estimated coherence matrix by considering 8 different edge-aligned processing windows. The edge-aligned window and filtering weights are determined based on the span image, a weighted average of $S_{hh}$, $S_{hv}$ and $S_{vv}$ intensities. Once this image is generated, the filter follows four basic steps:

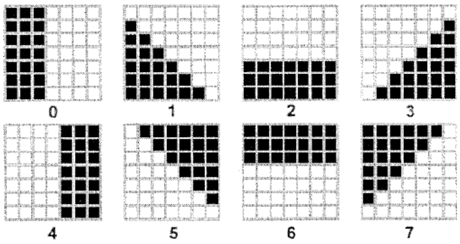


Figure 2.3: Eight edge-aligned windows

1) *Edge-aligned window selection:* to preserve edge sharpness, one of the 8 directional windows in fig 2.3 is selected based on the edge direction. Pixels in white are used in the filtering computation containing similar radiometric properties. This window is divided into subwindows and their mean is computed, reducing the effect of noise on the accuracy of edge detection and improving the weighting of the closest pixels.
   Edge direction is detected by a simple edge mask using the sub-means and selected as the one that is the maximum absolute.

$$\begin{bmatrix}-1 & 0 & 1\\-1 & 0 & 1\\-1 & 0 & 1\end{bmatrix}\begin{bmatrix}0 & 1 & 1\\-1 & 0 & 1\\-1 & -1 & 0\end{bmatrix}$$

$$\begin{bmatrix}1 & 1 & 1\\0 & 0 & 0\\-1 & -1 & -1\end{bmatrix}\begin{bmatrix}1 & 1 & 0\\1 & 0 & -1\\0 & -1 & -1\end{bmatrix}$$

2) *Apply the local statistics filter:* let $y$ be the center pixel value, $x$ the noise-free pixel value to be estimated and $n$ the noise with expectation equal to 1 and variance $\sigma_n^2$. $\sigma_n$ measures the speckle level and it is the standard-to-mean ratio in homogeneous areas. This value changes depending on the system (AIRSAR, ESAR,...) and the number of looks of the image. Assuming a multiplicative noise model, the linear minimum-square filter can be derived[8]:

$$y = xn \text{ then: } \hat{x} = \hat{y} + b(y - \bar{y}) \tag{2.11}$$

where $\hat{x}$ is the filtered pixel value, $\bar{y}$ is the local mean and $b$ is the weighting parameter, with value between 0 and 1. It can be computed by

$$b = \frac{var(x)}{var(y)} \text{ with } var(x) = \frac{var(y) - \bar{y}^2\sigma_n^2}{(1+\sigma_n^2)} \tag{2.12}$$

where $var(y)$, $var(x)$ are the local and zero speckle noise variance. For non-homogeneous areas, $var(x)$ is high, hence $b \approx 1$ and $\hat{y} \approx y$ whereas for homogeneous areas, $var(x) \approx 0$, $b \approx 0$ and $\hat{y} \approx y$. With: $var(x) \geq 0$ and $var(y) \geq 0$.

3) *Use span image to compute the weight in a selected edge-aligned window:* the window number obtained from 1) and the weighting parameter $b$ obtained from 2) are used to filter the covariance matrix by applying:

$$\hat{\underline{\underline{Z}}} = \bar{\underline{\underline{Z}}} + b(\underline{\underline{Z}} - \bar{\underline{\underline{Z}}}) \tag{2.13}$$

where each element of $\bar{\underline{\underline{Z}}}$ is the local mean, computed inside the same edge-aligned window. So as to preserve polarimetric properties, the same weighting should be applied to all channels while also selecting a proper value for $\sigma_n$: if the value is too high, over-filtering may blur the image; if it is too low, under-filtering may be very noticeable.
To achieve better results, $\sigma$ value for the span image should be experimentally determined by using a scatter plot of the standard deviation versus the mean for the polarizations $S_{hh}$, $S_{hv}$ and $S_{vv}$.

### 2.5.3 Sigma Lee Filter

The following deficiencies were discovered in the Refined Lee: it fails to maintain the mean value, introducing bias. This is particularly perceptible if the number of looks of the original SAR is small ($N < 4$); the highly reflective point targets are blurred, and their powers reduced; dark spotty pixels are not filtered.
The Sigma Lee filter[9] is based on the two-sigma probability of the Gaussian distribution and assumes the multiplicative model for the speckle noise. It stands on the concept that 95.5% of pixels are distributed within the two-sigma range from its mean.

The range for sigma is $(\tilde{x} - 2\sigma_n\tilde{x}, \tilde{x} + 2\sigma_n\tilde{x})$, where the *a priori* mean $\tilde{x}$ is not known and must be estimated. The Lee Sigma assumes $\tilde{x} = z$, the value of the center pixel. It reduces speckle noise by replacing the center pixel by an average of those pixels that fall inside the two-sigma range. Those that are left outside are considered outliers and are not included in the mean computation. Therefore, homogeneous areas of high-contrasting distributed scatters can be separated, reducing speckle noise.
The deficiencies commented above are addressed in the following manner:

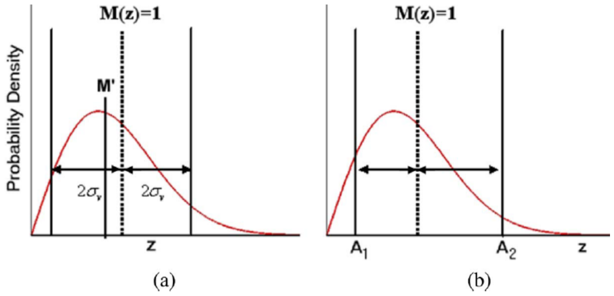


Figure 2.4: Bias problem with a Rayleigh pdf as an example. (a) The original two-sigma range shifts the mean to $M'$ which is less than one. (b) The revised sigma range $(A_1, A_2)$ will keep its mean value at one.

1) *Bias in the Filtered Data:* A bias is introduced due to the assumption of a symmetrical pdf with respect to its mean, underestimating the mean to $M'$. In order to fix this, a new sigma-range $(A_1, A_2)$ is computed based on the real pdf distribution to preserve the mean.
   The new sigma ranges have to be computed for a given sigma value $\xi$ and fulfil the following condition:

$$\xi = \int_{A_1}^{A_2} p_N(I)dI \ , \ \bar{I} = \frac{1}{\xi}\int_{A_1}^{A_2} Ip_N(I)dI \tag{2.14}$$

   Finally, the new sigma-range can be computed from the tables generated for different number of looks and $\sigma_n$ with any mean with $(A_1\tilde{x}, A_2\tilde{x})$. Different $\xi$ values are picked depending on the application.

2) *Selection of the a priori mean:* to determine the final sigma range, we need to know $\tilde{x}$ (a priori mean). In the non-revised Sigma Lee, $\tilde{x}$ was assumed to be equal to the central pixel $z$. However, the filtered results exhibited the effect of dark isolated spots and other deficiencies. Two solutions arise to address this problem:

   1) The 3x3 mean
   2) The estimate from MMSE (Minimum Mean Square Error) in a 3x3 window.

   The second is selected to estimate $\tilde{x}$ as it yields better results overall (retains fine details). MMSE computes a weighted average between the center pixel and its local mean, hence producing an unbiased estimate. Its mathematical expressions were introduced before 2.12,2.13.

   This process is used twice: first the MMSE is applied with a 3x3 window to estimate $\tilde{x}$ (original standard to mean ratio) and later, it is applied to pixels within the sigma range in a 9x9 or 11x11 window (use revised standard to mean ratio).

3) *Preservation of point scatters:* point targets are high energy pixels which are generally produced by the double-bounce scattering mechanism or by direct specular reflection. This expansion of the sigma lee filter only deals with high-return targets. Backscattering signatures of point targets are considerably different from those of distributed media and, as a consequence, have to be processed differently.
   The authors proposed an algorithm to detect and preserve the target signature of strong returns based on the assumption that point targets often form a cluster of several pixels unlike distributed media. The algorithm steps are:

   1) Compute the 98th percentile $Z_{98}$ of all pixels of the SAR image to be filtered.
   2) If the value of the processed pixel is higher than $Z_{98}$, check how many pixels in a 3x3 window around the selected pixel meet the former condition.
   3) If more than $T_k$ pixels meet this criterion, all the pixels within the window are assumed to be point targets and retain their values.

   Generally, $5 \geq T_k \leq 7$. The principle of this algorithm is that few pixels from distributed scatters have values above $Z_{98}$, so the probability of having $T_k$ in a 3x3 windows is even lower. Conversely, point targets cannot be totally preserved, since there will be small point targets that will not meet the previous criterion.

### 2.5.4 IDAN Filter

The Intensity-Driven Adaptive-Neighbourhood (AN) [10] builds a variable-size neighbourhood around the processed pixel (*seed*) with a region growing algorithm. This neighbourhood only contains connected pixels that belong to the same statistical population as the seed and are the ones used to compute the output value.
This filter adapts its response to the image's morphology and reduces the limitations of predefined windows (Refined Lee filter [8]) and instead produces an arbitrary set of adjacent pixels

per seed. The IDAN uses the 3 diagonal elements of the coherence matrix $\underline{\underline{T}}$ to drive the region growing process.

Step I

1) *Rough estimation of the seed value:* Inside a 3x3 centered neighbourhood, the median $\hat{[p]}(m,n)$ is computed for each of the six components in the matrix. This is specially useful when we are trying to smooth data, when blurring the signal edges is unacceptable, even less at this early stage in the process.

2) *Region growing:* all the 8 direct neighbours of the seed are accepted into the AN provided they fulfil the following aggregation condition that uses the tree distances for each of the 3 components of the vector $[p]_i$:

$$\sum_{i=1}^{3} \frac{||[p]_i(k,l) - \widehat{[p]}_i(m,n)||}{||\widehat{[p]}_i(m,n)||} \leq 2\frac{\sigma_n}{\mu_n} \tag{2.15}$$

where $\mu_n$ and $\sigma_n$ stand for the speckle mean and standard deviation, respectively. Then, this same procedure is applied for all the neighbours of the newly included pixels and so forth. The region growing iterates likewise until either the number of pixels included in the AN reach its upper boundary,$N_{max}$ or none of the neighbours verifies the above condition. The pixels which have been already tested and non-accepted into the AN (e.g.: *background pixels*) are stored separately.

Step II

1) *Refined estimation of the seed value:* a better estimate is obtained by averaging the pixels included in the "*strict* "neighbourhood produced in Step I.

2) *Reinspection of the background pixels:* background pixels $[p](o,p)$ of the listed created in Step I are tested again and aggregated to the AN if they meet the "enlarged" condition:

$$\sum_{i=1}^{3} \frac{||[p]_i(o,p) - \overline{[p]}_i(m,n)||}{||\overline{[p]}_i(m,n)||} \leq 6\frac{\sigma_n}{\mu_n} \tag{2.16}$$

This test is less restrictive since the inclusion threshold is larger.

The multilook induced by IDAN may affect the properties of the polarimetric scattering signatures of the targets by introducing a high bias. To compensate this, the IDAN estimation is combined with the LLMMSE (Linear Minimimum Mean Square Error).Instead of directional windows, the constructed AN provides the necessary spatial support for the final LLMMSE estimation of the coherency matrix as:

$$[\tilde{T}]_6 = \widehat{[T]_6^{IDAN}} + b([T]_6) - \widehat{[T]_6^{IDAN}} \tag{2.17}$$

where $b$ is the weighting parameter used for both Refined and Sigma Lee filters but computed using all the pixels of the six PolSAR images averaged instead of the average of the two interferometric images, reducing the bias.

### 2.5.5 Non-Local Means Filtering

In contrast with the former techniques, the non-local means filter (NL-means)[11] does not rely on supposing local stationarity for SAR data. It is based on a non-local weighted average of pixels but, in contrast with the Refined Lee or IDAN, they can be located outside the neighbourhood of the given pixel, at any position in the image.

In the case of a discrete image $v = v(i)|i \in I$, where $I$ represents the set of pixel positions, the NL-means filter for a pixel at position $i$ may be computed as the weighted average over the image pixels:

$$NL(i) = \sum_{j \in I} w(i,j)v(j) \tag{2.18}$$

where the set of weights $w(i,j)$ are chosen depending on the similarity between the pixel at position $i$ and $j$ and verify the following conditions:

$$0 \leq w(i,j) \leq 1 \;\; \forall i,j \in I \;\; , \;\; \sum_{j \in I} w(i,j) = 1 \;\; \forall i \in I \tag{2.19}$$

Furthermore, in order to compute the similarity between pixels $i$ and $j$, a set of values around a patch or neighbourhood is used instead of their values. As a consequence, this comparison requires a similar context around the analysed pixels. Mathematically, this can be expressed as:

$$w(i,j) = \frac{1}{Z(i)} exp(-\frac{||v(N_i) - v(N_j)||^2}{h^2}) \;\; , \;\; Z(i) = \sum_j exp(-\frac{||v(N_i) - v(N_j)||^2}{h^2}) \tag{2.20}$$

where $N_i$ and $N_j$ denote the set of neighbouring pixels around pixel $i$ and $j$, respectively; $Z(i)$ is the normalizing factor and h is a parameter that controls the amount of filtering. NL means has also been extended to process PolSAR data. In this case, the similarity comparison for each pixel in a path is computed using the scattering vector absolute value relations, which are proportional to:

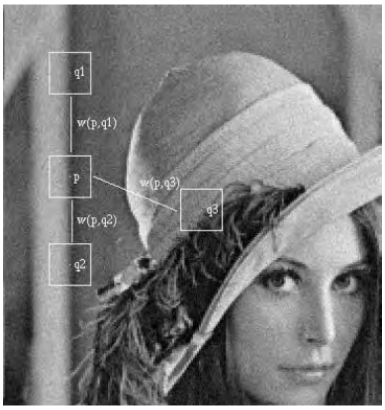


Figure 2.5: NL-means similarity comparison. $q1$ and $q2$ receive large weights $w$ whereas $q3$ takes a smaller weight since it is surrounded by a distinct neighbourhood

$$ln\left(\frac{|k_{1,1}|}{|k_{2,1}|} + \frac{|k_{2,1}|}{|k_{1,1}|}\right) + ln\left(\frac{|k_{1,1}|}{|k_{2,2}|} + \frac{|k_{2,2}|}{|k_{1,2}|}\right) + ln\left(\frac{|k_{1,3}|}{|k_{2,3}|} + \frac{|k_{2,3}|}{|k_{1,3}|}\right) \tag{2.21}$$

where $\underline{k_{1j}}$ and $\underline{k_{2j}}$ refer to the scattering vectors of each one of the two pixels compared on the patch, $k_{i,j}$ denotes the $j$-th component of the $i$ scattering vector and $ln$ is the natural logarithm.

Finally, an iterative refinement of the filter is proposed to take into account all the polarimetric information contained in $\underline{\underline{T}}$ (based on the Kullback Leibler divergence):

$$SD_{KL}(\hat{T_1}, \hat{T_2}) \propto tr(\widehat{T_1^{-1}}, \hat{T_2}) + tr(\widehat{T_2^{-1}}, \hat{T_1}) - 6 \tag{2.22}$$

However, two similar pixels are required to be located in a comparable context in order to be considered similar, resulting into sub-optimality results in terms of filtering.

### 2.5.6 Results with the SOTA techniques

In this subsection the analysed SOTA techniques alongside two techniques that use the BPT are assessed with the same test set and the results are shown in the following pages.
These approaches use two different pruning techniques, called *threshold max*[1] and *mincut*[5][12], respectively. Before pruning the tree, both techniques follow these steps:

1) *Read the input image and pre-process it with Sigma-Lee filter*: the image is pre-processed in order to work with a regular matrix. It also provides the algorithm with a less noisy image to work with and, as we commented in the previous section, Sigma-Lee introduces a relatively small resolution loss in comparison with most SOTA techniques.

2) *Tree creation and population*: given a set of configuration parameters, the BPT is created from the pre-processed image. The tree is then populated with some basic and specific features using the original noisy image. We select the denoised image for creation and the original for pruning, following the conclusions presented in [13], where marginally better results were obtained using this combination.

Once the BPT is constructed and populated, it is pruned and depending on the selected technique, the input parameters and consequently, the results, are different.
For the first technique, *threshold max*, introduced by Alberto Alonso-González in [1], the parameter $\delta_p$ is defined as the threshold for which any node with a 'feature' value below it, will be pruned. This means that a constant threshold is used for the whole BPT support.
With the second technique, *mincut*, introduced by Philippe Salembier, there is no pruning parameter as such. Instead, a regularization parameter $\lambda$ is defined when computing the feature value during the population of the BPT with specific parameters. Then, mincut computes the cut by minimizing the sum of the node feature, selecting the parent node or the children depending on which minimizes the sum.
In our case, the selected feature is *SAR_SE* which is a square-error-based measurement for SAR images. Concretely, this feature is defined in [13] as the sum of the norm of the difference between the pixel value and the mean of the region. This difference is then normalized by the norm of the region in order to take into account the presence of speckle noise. Mathematically, it can be expressed as follows:

$$\phi_R^{SAR\_SE} = \sum_{i,j \in R} ||Z_{ij} - Z_R||_F / ||Z_R||_F + \lambda \tag{2.23}$$

where $Z_{ij}$ denotes the pixels $i, j$ belonging to a given region $R$, $Z_R$ is its covariance/coherence matrix, $||\cdot||_F$ stands for the Frobenius norm and $\lambda$ is the regularization parameter. $\lambda$ is constant and penalizes the region presence (aiming to reduce the number of regions). This means that, for a large value, the pruning is more strict and is closer to the root node (obtaining, ideally, more homogeneous regions) whereas, for a small value, the pruning is less strict and we end up with regions closer to the leaves and hence, more noisy and heterogeneous.
In practice, some distinct regions may be mixed, in order to minimize the criterion for a given large $\lambda$. This is why this parameter should be optimized for each case of study.
An extension for both approaches refered to as "window", uses an averaging sliding window inside the region with smaller size and only using those pixels that belong to the processed region. This approach yields slightly different results to the standard mincut and threshold max approaches, seeing a "Boxcar effect" inside the regions introducing a blurring in non-homogeneous regions.

### Test Image Set

The image set is composed of 10 synthetic images generated with the algorithms developed by Samuel Foucher[14]. This set includes images with a random number of classes, from 3 to 5, with one of them corresponding to point scatters. The rest are selected randomly from 7 classes, each of them representing a unique response type (sea, forest, city, fields, among others). These images are square and with size equal to 256 pixels. Some examples are shown below, including the noisy samples with their corresponding Ground Truth (without noise) and Sigma-Lee filtered samples.

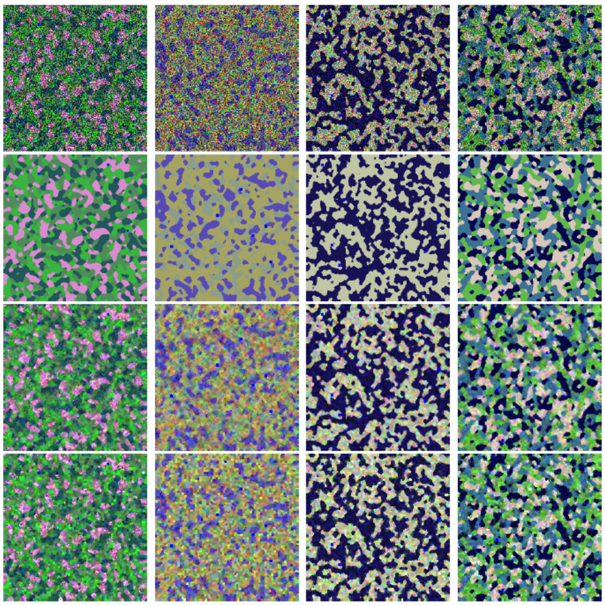


Figure 2.6: Examples of original images from the synthetic dataset (1st row), corresponding ground-truth (2nd row), processed images with sigma-lee (3rd row) and superpixel SLIC partitions (4th row). The colour coding used is: RGB Pauli. The polarimetric channels $|S_{hh} - S_{vv}|^2$, $\sqrt{2}|S_{hv}|^2$ and $|S_{hh} + S_{vv}|^2$ are assigned to R, G and B, respectively.

The analytic parameter selected to compare all the techniques is the *relative error ($E_R$)* measurement proposed by Alberto Alonso-González [1] that assesses the filtered image $X$ in comparison with the synthetic (or simulated) ground truth $Y$. The formula is the following:

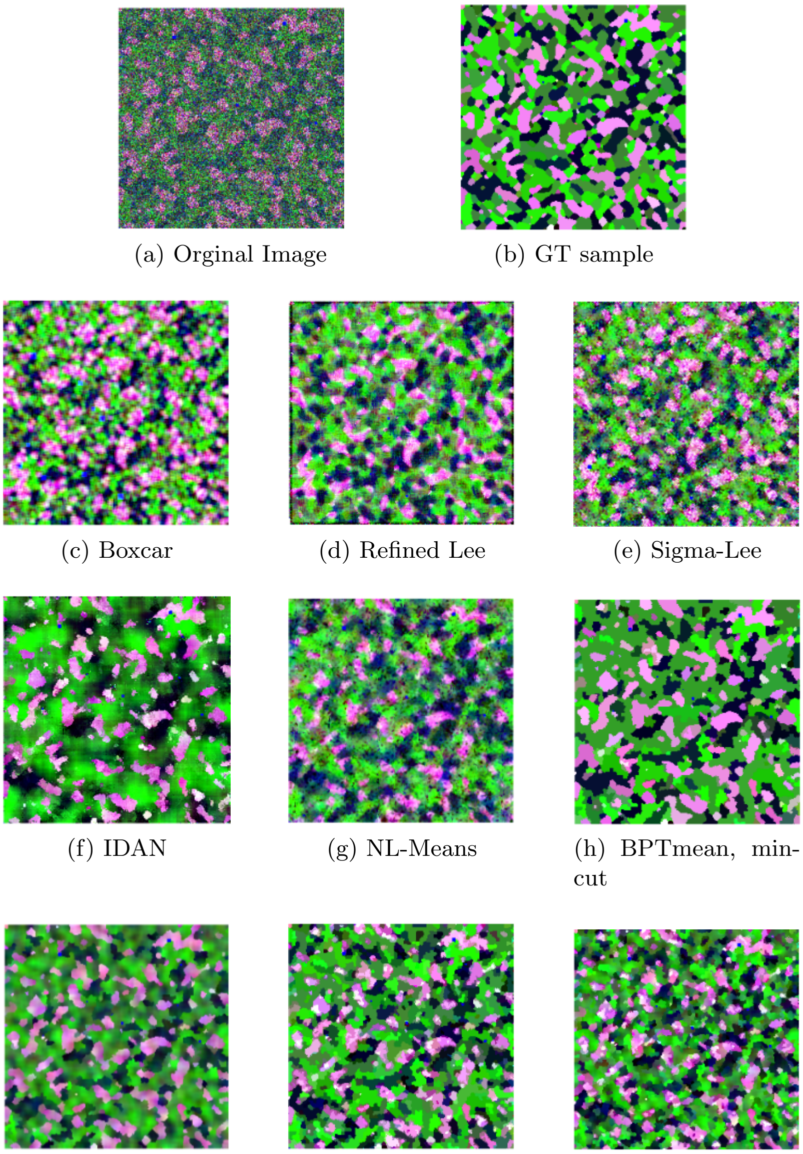
$$E_r(X, Y) = \frac{1}{n_h \cdot n_w} \sum_{i=1}^{n_h} \sum_{j=1}^{n_w} \frac{||X^{ij} - Y^{ij}||_F}{||Y^{ij}||_F} \tag{2.24}$$

where $i, j$ refer to the pixel $(i, j)$ of $X$ or $Y$ explained above, $n_h$ and $n_w$ detail the image height and width, respectively and $|| \cdot ||_F$ stands for the Frobenius matrix norm.

The closer the ground truth and filtered image are, the lower $E_R$ will be. The results shown are averaged over the whole set of 10 images to obtain more reliable results. A table showing the set of parameters that yield the best $E_R$ value for each technique, is added below alongside the different outputted images (for the first sample) compared with the original noisy image and the ground truth sample:

| **Criteria/ Filters** | **Pruning Param.** | **WinSize** | $\boldsymbol{E_R(dB)}$ |
|---|---|---|---|
| Boxcar | N/A | 5x5 | -8.7182 |
| Refined Lee | N/A | 7x7 | -11.4601 |
| Sigma-Lee | N/A | 9x9 | -10.6737 |
| IDAN | N/A | 80 | -10.1723 |
| NL-Means | N/A | 7x7 | -13.2333 |
| BPTmean(mincut) | $\lambda = 7$ | N/A | -14.6089 |
| BPTmean(thrmax) | $\delta_p = -2dB$ | N/A | -13.7800 |
| BPTwind(mincut) | $\lambda = 7$ | 9x9 | -14.8651 |
| BPTwind(thrmax) | $\delta_p = -2dB$ | 9x9 | -13.8300 |

Table 2.1: Results for the analysed state-of-the-art techniques.



(a) Orginal Image

(b) GT sample

(c) Boxcar

(d) Refined Lee

(e) Sigma-Lee

(f) IDAN

(g) NL-Means

(h) BPTmean, mincut

(i) BPTwindow, mincut

(j) BPTmean, thresholdMax

(k) BPTwindow, thresholdMax

Figure 2.7: SOTA outputted images compared with original and GT samples

We can clearly see that the best results are obtained by the BPT processing schemes by comparing them to the ideal ground truth sample and seeing how closely estimated are both mean and variance. The boxcar filter introduces a lot of spatial resolution loss, Sigma-lee and NL-means are the best approaches which do not use the BPT structure, obtaining a nice amount of filtering while not causing a big resolution loss. IDAN introduces a high bias in its estimation, underestimating the mean of the image. Finally the Refined Lee blurs out the point targets and also introduces a slight bias despite obtaining a reasonable amount of filtering.

# Chapter 3

# BPT potentiality assessment for Speckle Noise reduction

This chapter includes the potential assessment regarding a new approach for speckle noise reduction with Binary Partition Trees. This approach considers filtering the BPT along the branches of the tree at different scales, taking advantage of the hierarchical structure and the dependencies that can be extracted from it.
In the following sections, the main idea behind this approach will be exposed, including all the parameters that have been studied both in graphical and analytical forms.

## 3.1 Main Idea and Graphical Representations

We are trying to prove that BPT branch filtering may, indeed, improve the state-of-the-art results, both using or not the BPT, in terms of the relative error measurement introduced before. The potentiality is evaluated in terms of the gain in dB's between the best current result (BPT with mincut pruning) and the relative error value obtained by selecting the best value for each leaf of the BPT. This is done by comparing every ancestor's node value for the processed leaf (the whole branch) and the ideal leaf value available in the ground truth (GT) sample, that is, evaluating the error introduced by representing the leaf by each ancestor. The best value is then selected as the closest one to the GT mean value. A distance measure must be defined so we can select the closest node as the one that minimizes it. The selected distance is the one based on the relative error measurement introduced before as the global evaluating parameter in eq. 2.24.

Once the best value is selected, the output image is reconstructed by using the initial partition (obtained from the noisy image with the SLIC algorithm[15] which was used for the *mincut* and *threshold max* techniques commented above) and the optimum values for the BPT populated with the *original image*, found in the previous step.
Before analysing the obtained results, a brief introduction to a graphical representation employed, is needed. As we are exploring the BPT by branches, a suitable representation is the one given by a *branch representation*, introduced in [16]. This representation is useful to analyse the evolution of graph signals (e.g.: relative error).
This representation of the BPT is based on a gray level image: the horizontal axis denotes the individual tree branches and the vertical axis the distance of each node to the root. The magnitude of pixel $(i, j)$ denotes graph signal value of the node that belongs to branch $i$ and is at distance $j$ of the root. In this case, there are duplicated nodes in the representation that belong to multiple branches.
However, in our case, we try to represent the relative error of each leaf with respect to all its ancestors hence obtaining different values for each leaf. This is why the previous representation has been adapted to fulfil our case, adding the possibility of having different values for each

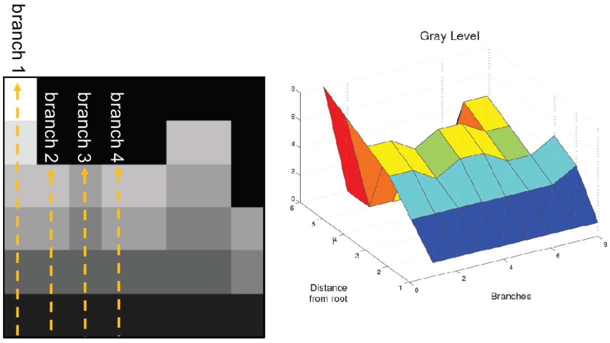


Figure 3.1: Example of branch representation in 2D (left) and 3D (right)

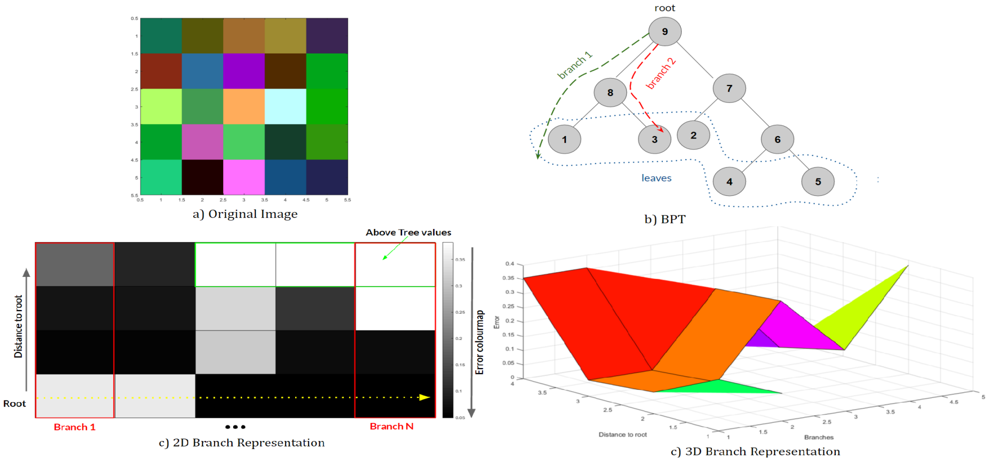
node in case nodes belong to multiple tree branches.
In order to understand the upcoming graphs, a simple example with a small PolSAR image is shown in Fig. 3.2. One can see the input image, the BPT standard representation, the branch representation of the relative error for each node, using a gray colourmap, and the corresponding 3D representation using a hsv (hue-saturation-value) colourmap.

Note that for the example provided, the root node (located at the bottom of the 2D graph) belongs to 5 different branches, having 5 different root values.This can also be seen by looking at the 3D class, where, at the coordinate of the y-axis (distance to root) equal to 0 we have different heights, representing the error.

When we planned to find the best node for each leaf, two initial approaches were proposed to answer the following questions:

1) Is the current partition too coarse to retrieve good candidates and a consequence, obtain a lower relative error (bigger margin of improvement).

2) Is it better to assign the best value (a single value) or estimate it as a weighted sum of a few candidates values?



a) Original Image

b) BPT

c) 2D Branch Representation

c) 3D Branch Representation

Figure 3.2: Example of BPT branch representations for a sample PolSAR synthetic image. Pauli RGB colour coding is used.

## 3.2 Influence of the initial partition

The first question that we want to answer is if the current initial partition of the image is good enough to obtain a meaningful BPT and hence provides a wide margin for branch filtering in

our potentiality assessment. To do so, we are going to represent each leaf of the tree by the value of the node that generates the minimum error (by comparing with the GT as we stated above) and we will compute the overall relative error for distinct initial partitions.
Currently, for both SOTA BPT pruning techniques (mincut and threshold max) the initial partition is generated by the SLIC algorithm[15]. *Simple Linear Iterative Clustering* is an adaption of $k$-means for superpixel generation with two main advantages with respect to similar algorithms: the complexity is linear with the number of pixels and independent of the number of superpixels; a weighted distance measure combines colour and spatial proximity while controlling the size and compactness of the superpixels. This algorithm uses the sigma-lee pre-processed image (3 power channels) and has two parameters, which are:

- *Step Size/regionSize*: is the starting size of the superpixels. A larger value enforces a coarser partition, that is, one with a smaller number of regions whereas a smaller value imposes a finer partition, closer to a pixel partition (1 initial region per pixel), i.e: a larger number of regions.
- *Regularizer*: controls the amount of spatial regularization introduced by the algorithm when clustering, meaning that the larger the value, the more spatial regularization is applied to the partition (less permissive with different shapes).

For the previous SOTA BPT methods, the selected values were 4 and 0.005, respectively. In our case, we analysed the influence of these values by adding new ones with different combinations of the parameters mentioned above.
The results obtained in terms of relative error, averaged over the 10 images test set, can be seen in table 3.1.

| ***Global $E_R$ in dB*** | |
|---|---|
| Relative Error | -17.7168 |
| Mincut Reference | -14.6090 |

Table 3.1: Results for different distances and starting initial partition (SLIC). Images at full resolution (256 x 256).

Observing the results for the starting partition, we can see that the largest margin (compared to the best SOTA filtering technique: BPT with mincut pruning called "Mincut Reference" in the table) is obtained with the distance measurement based on the relative error, as one may have expected. Nevertheless, the gain obtained, 3.1078 dB is not large enough to start developing techniques that do not rely on the ground truth. As a consequence, further analyses with finer initial partitions have to be made.

In the next table, the results for tests taken with finer SLIC partitions are shown. The initial partitions and their parameters are the following: SLIC with $step\_size = 2$ and $reg\_param = 0.005$: Ref.: SLIC1; SLIC with $step\_size = 2$ and $reg\_param = 0.010$: Ref.: SLIC2; SLIC with $step\_size = 1$ and $reg\_param = 0.005$: Ref.: SLIC3; SLIC with $step\_size = 1$ and $reg\_param = 0.010$: Ref.: SLIC4.

| ***Global $E_R$ in dB*** | | | | |
|---|---|---|---|---|
| Initial Partition | SLIC1 | SLIC2 | SLIC3 | SLIC 4 |
| Relative Error | -19.0033 | -18.9670 | -20.9245 | -20.7936 |
| Mincut Ref. | -14.6090 | — | — | — |

Table 3.2: Results for different distances with the optimum node.Finer SLIC partitions. Images at full resolution (256 x 256).

By looking at the table we can conclude that a finer partition helps when we are selecting a unique node as the best candidate. This is specially true for SLIC3, where we are using the smallest *step_size*, obtaining a finer partition with more regions, and *reg_param* = 0.005, a relatively small regularization parameter that is more permissive when it comes to the shape of the regions of the initial partition.

Once proven that a finer partition helps locating better candidates that generate a lower $E_R$, we would want to analyse the behaviour of the BPT. As we mentioned above, the *branch representation* is very useful to check the evolution of a parameter in the tree as a function of the branch location and the distance to the root.

Seeing the results, one wonders why can't we exploit the correlation found in a branch to estimate the best candidate as a weighted sum of some nodes instead of selecting a unique node value.

The following sections try to answer this question by running several analyses, testing different sets of weights and checking the impact that these have on the final $E_R$ value.
Before continuing, though, the next table shows the results obtained before for full-resolution images, but now achieved with the 64x64 set, so the results from now on are comparable. With a smaller image to process, we can run more tests with different parameters. In this regard, two new initial partitions based on SLIC are considered using a smaller regularization parameter. Note that we also report the result for an initial partition at the pixel level, meaning that one pixel belongs to a single region (e.g. N of regions = N of image pixels).

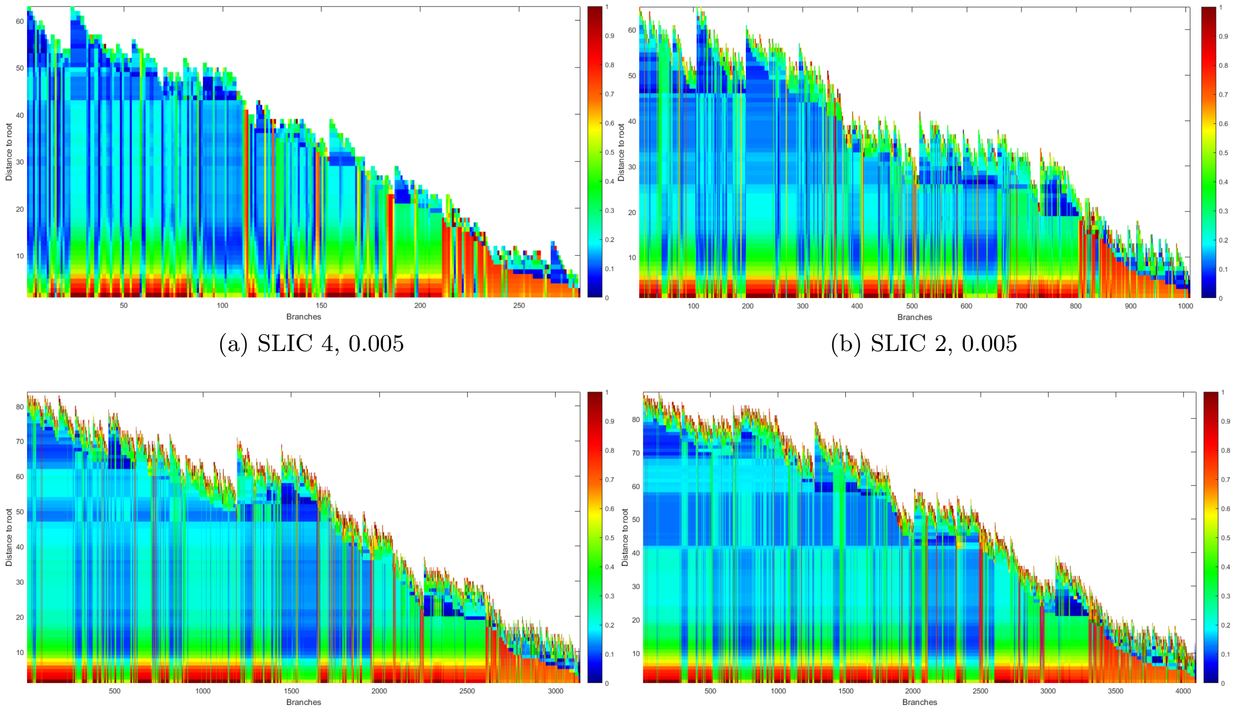
| ***Global $E_R$ in dB*** | | | | |
|---|---|---|---|---|
| | *Reg_param* | 0.01 | 0.005 | 0.0025 |
| *Step_size* | 4 | — | -17.1480 | — |
| | 2 | -18.2007 | -18.3840 | -18.6115 |
| | 1 | -20.4407 | -20.1747 | -20.3337 |
| | Pixel Partition | -21.3888 | — | — |
| | Mincut Ref. | -13.8538 | — | — |

Table 3.3: Results for different partitions with the optimum node. Images at reduced resolution (64x64).

Evaluating the results, we can see a similar behaviour to that presented for larger images: the smaller the step size, the finer the partition is, and the lower the $E_R$ value is. In addition, a smaller regularization parameter seems to help for a relatively small *step_size*, but is too small when dealing with the smallest SLIC partition with *step_size* = 1 which holds a higher number of regions.
Finally the pixel partition is the one that yields the best results, being the finest partition of all despite introducing a higher computational load (not appreciable with such small images but it will surely have an impact when dealing with larger images).

### A closer look: $E_R$ branch representation

Even though the global $E_R$ gives us a numeric value to assess the potentiality of branch filtering, we would want to see how the relative error evolves along the branches in more detail. To that effect, the $E_R$ branch representation for all the partitions considered above, has been computed.



(a) SLIC 4, 0.005

(b) SLIC 2, 0.005

(c) SLIC 1, 0.005

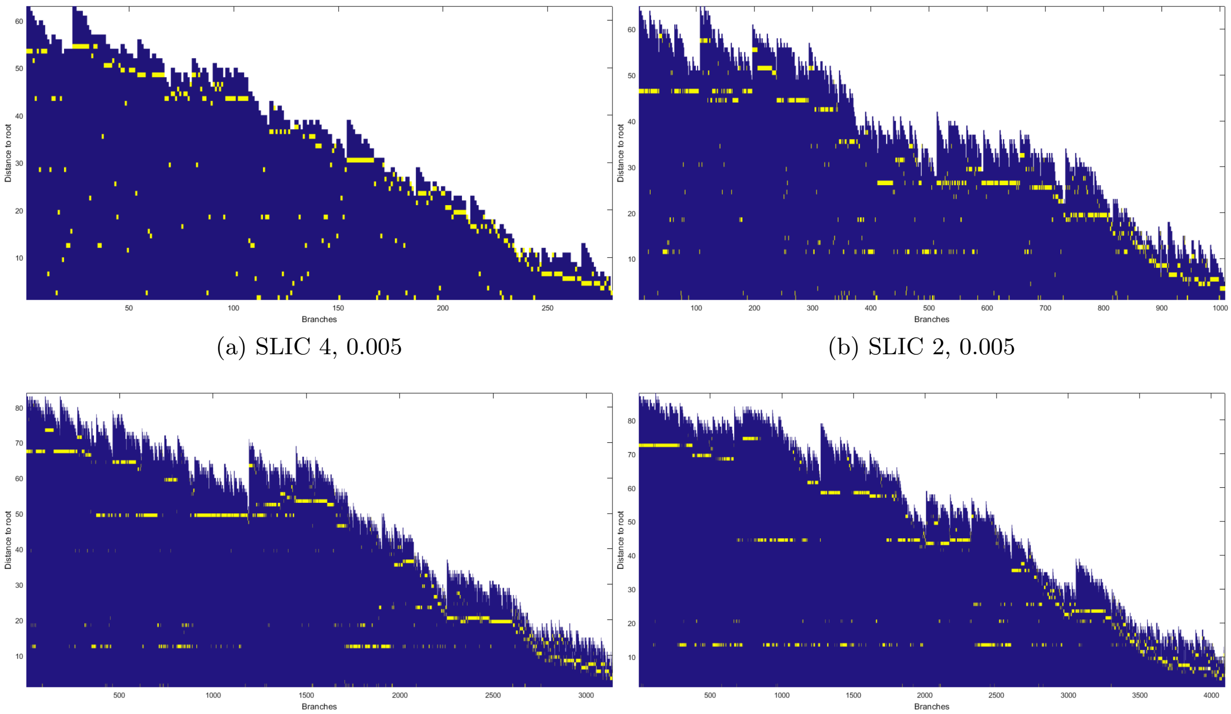
(d) Pixel Partition

Figure 3.3: Branch representation of relative error $E_R$ for the initial partitions detailed in table 3.3.

Although a similar distribution of levels is observed (smaller values are shown in dark blue, higher values in dark red), the finer the partition, the more concentrated the global branch minimums are. This may seem like a drawback for the pixel partition but is not so, as the global minimums are smaller than those found in coarser partitions were more similar values (around the global optimum) can be found.

Another important aspect to notice is that we can find "walls" for certain branches, that is, a high value for $E_R$ maintained throughout the whole branch (in dark red). This values may arise from the presence in the region of a point scatter carrying a lot of energy or a mixture between point scatters and distributed media, therefore generating a big error when compared to the ground truth sample.

For visualization purposes, a graph containing only the absolute branch minimums location is shown in Fig. 3.4.



(a) SLIC 4, 0.005

(b) SLIC 2, 0.005

(c) SLIC 1, 0.005

(d) Pixel Partition

Figure 3.4: Branch representation with the location of the $E_R$ branch minimums

There is not any repetitive pattern for all branches to mark the location of global minimums, although they are usually closer to the leaves than to the root because it is unlikely that the mean of a very heterogeneous region is the best representation for a given leaf (representing a very small region, down to a single pixel in the pixel partition), as it contains a mixture of many regions.

## 3.3 Influence of the number of candidates

This section aims to answer the second question proposed before: would an approximation based on more than a value (only in a perfect case that would be the branch optimum) improve that made with just one candidate? To answer this question, the following testing scenario is proposed: use the Ground Truth to combine the $L$ best candidates, that is the $L$ first minimums (in ascending order) to estimate the leaf value.

For all this section analyses, the starting initial SLIC partition has been considered ($step = 4$, $reg = 0.005$). Three different values for $L$ are used with different combinations of weights:

- For 3 candidates: $w1 = [0.85, 0.10, 0.05]$, $w2 = [0.88, 0.10, 0.02]$ and $w3 = [0.75, 0.15, 0.10]$.
- For 4 candidates: $w4 = [0.55, 0.25, 0.10, 0.10]$ and $w5 = [0.75, 0.15, 0.08, 0.02]$.
- For 5 candidates: $w6 = [0.80, 0.10, 0.06, 0.03, 0.01]$ and $w7 = [0.74, 0.15, 0.08, 0.06, 0.04]$.

The results show a slight improvement for the best result, $-17.2608$ dB, over the results for a single optimum node and the same initial partition shown in table 3.3, $-17.1480$ dB.

This improvement though, is not large enough to assure that weighting the best candidates will always help, as the other results in the table (for other weight combinations) show. In any case, finding the optimum combination of weights for each leaf (regression problem) may be a complex task for the improvements obtained.

| | ***Global $E_R$ in dB*** | | | |
|---|---|---|---|---|
| | Number of candidates $L$ | 3 | 4 | 5 |
| *Weight_set* | w1, w4, w6 | -17.2502 | -17.2072 | -16.7530 |
| | w2, w5, w7 | -17.2512 | -17.2608 | -17.2572 |
| | w3 | -17.2546 | — | — |
| | One candidate | -17.1480 | — | — |
| | Mincut Ref. | -13.8538 | — | — |

Table 3.4: Results for different number of first best minimums $L$

## 3.4 Chapter general conclusions

After the exhaustive analysis on BPTs potentiality, we can conclude that both approaches planned (finer partition and weighted estimation) help to enlarge the margin of improvement of the BPT for speckle noise reduction with respect to the SOTA. Despite the small improvement observed in table 3.5, this could be higher for techniques that do not rely on the ground truth as the candidates will be of much lower quality and more noisy. In this regard, combining candidates may act as a noise cancellation technique.

Nevertheless, once proven that a wide margin of improvement is available, if we are able to select the proper candidates, the next step is to try to develop techniques that should identify these candidates without using the ground truth. This is the purpose of the following chapters.

# Chapter 4

# Techniques based on partition extraction

In this chapter, new strategies that rely on the extraction of a partition from the BPT, are presented, analysed and compared against the current SOTA BPT techniques.
Seeing the results presented in previous chapter by employing a weighted estimation, three techniques based on a partition extraction along with an (optional) weighted estimation, are considered.

## 4.1 Mincut's cut as a reference plus a weighted estimation

Mincut's algorithm is based on the minimization of the dissimilarity measurement (e.g: $SAR\_SE$, in our case) for a given node with a regularization parameter, $\lambda$, that enforces the algorithm to find bigger yet homogeneous regions.
Based on that and following the weighting approach presented in the previous section, both techniques are combined into one.
Before proceeding, the parameter $\lambda$ has been optimized in terms of relative error for the case at hand (speckle noise filtering) and for different partitions.

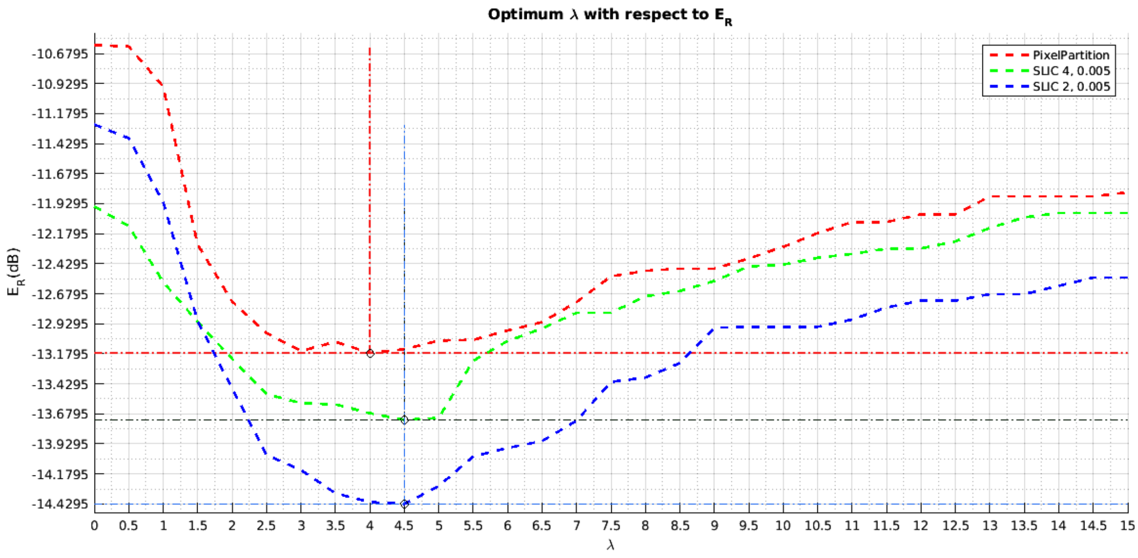
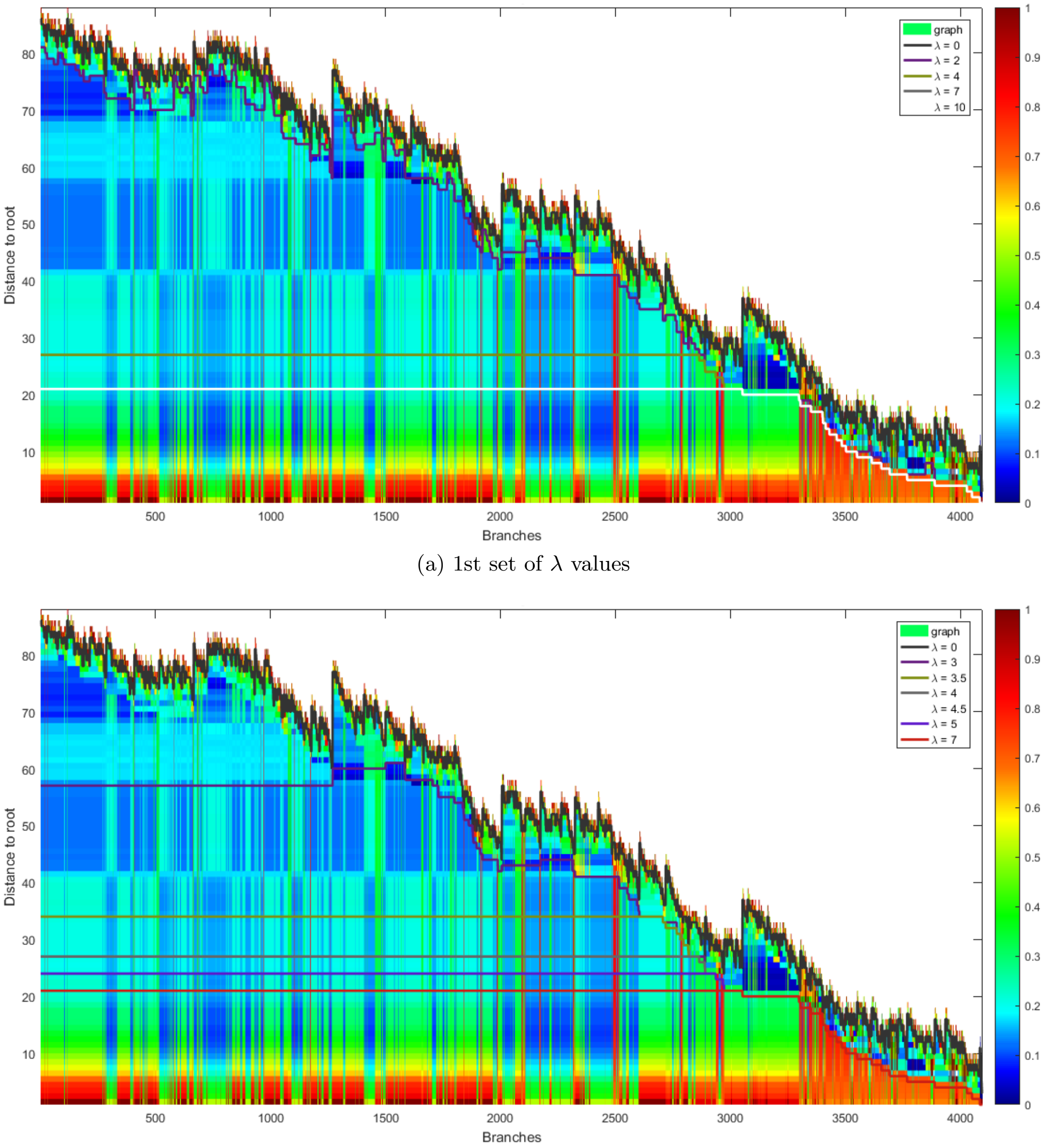


Figure 4.1: Mincut's optimization with respect to $E_R$

Note that the pixel partition is too fine to be useful when using the mincut as it is does not provide the best BPT representation.
Once $\lambda$ is optimized, different cuts can be visualized over the branch representation of $E_R$.



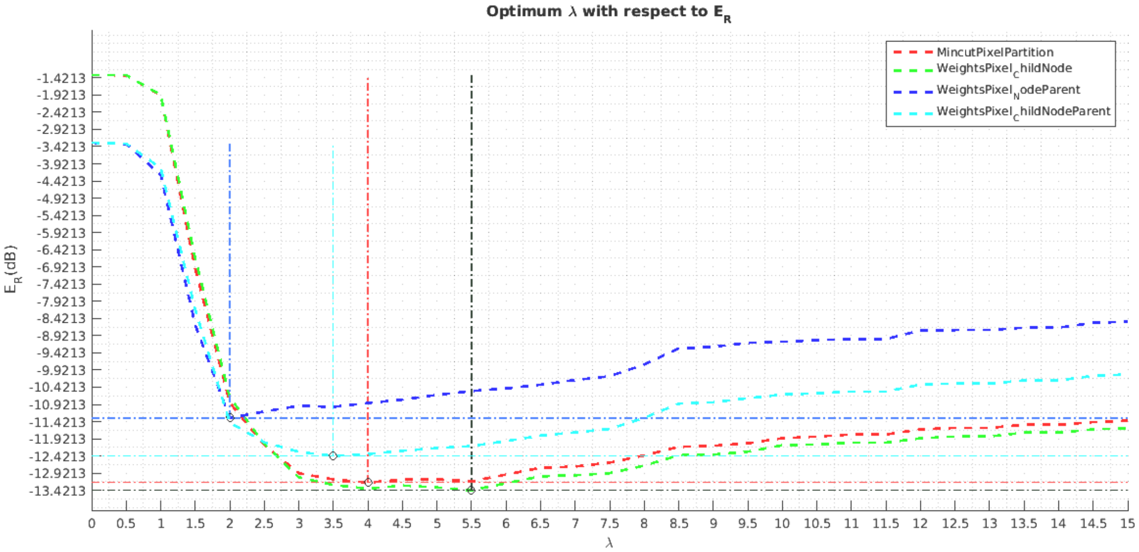
(a) 1st set of $\lambda$ values

(b) 2nd set of $\lambda$ values

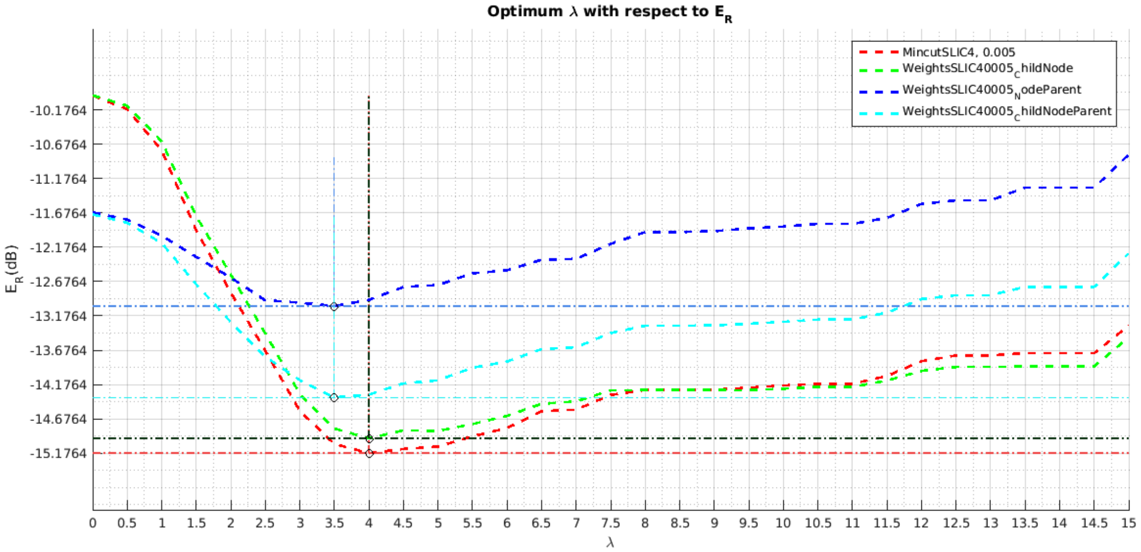
Figure 4.2: Mincut's cuts over $E_R$ branch representation

We would like to see whether these cuts fall close to nodes which could be considered as good candidates for speckle noise reduction. We can see that there is no mincut globally optimum to be used as a reference as one should have expected, instead it would seem like a different $\lambda$ value should be selected depending on the current's branch length: for smaller regions closer to the leaves, close values yield quite different cuts whereas for bigger regions closer to the root, different values yield the same cut.
Overall, the best cut is the one with $\lambda = 2$ (subfigure a) although it is too strict for the longest branches as it can be seen in the top left of the graph.

With the goal of improving the relative error obtained by the mincut algorithm, a weighted estimation is tested so as to assign nodes closer to the global minimum. To simplify the test, we have only considered weighting the closest ancestors/descendants with these approaches: weight the node selected by the mincut partition for the current leaf and its child (1/3 and 2/3, respectively); weight the node and its parent (1/3 and 2/3 in order); weight children, node and parent



(a) Mincut vs Weighted mincut. Results for the pixel partition.



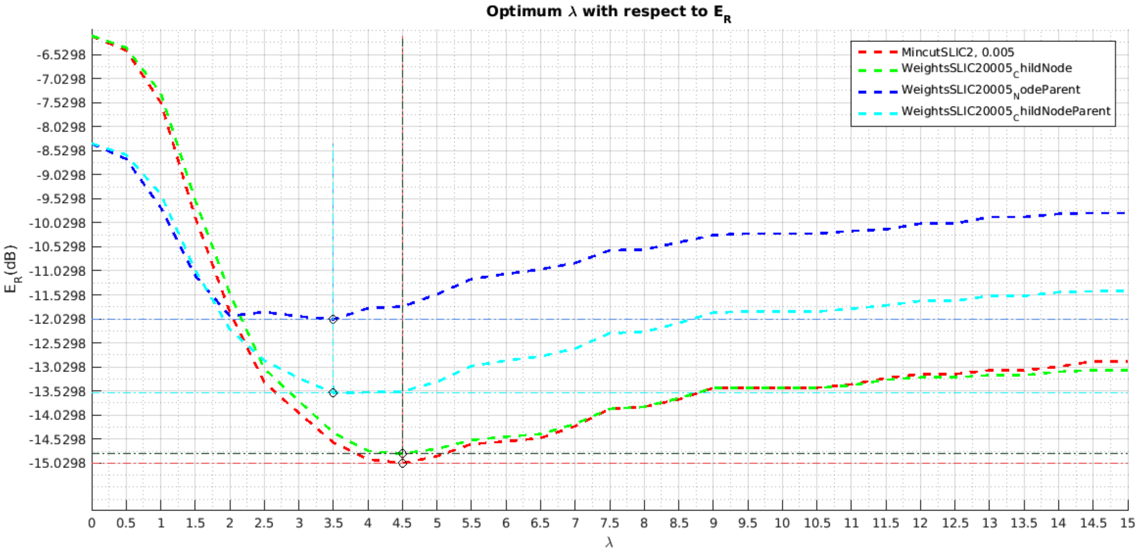
(b) Mincut vs Weighted mincut. Results for the SLIC partition ($step = 4$, $reg = 0.005$).

(3/8, 2/8 and 3/8). The performances have been evaluated as a function of the $\lambda$ value.

By looking at the graphs we can conclude that using the closest ancestors/descendants of the nodes chosen by the mincut algorithm does not improve the relative error value or when it does, being the case of the pixel partition (first subfigure), the improvement is too small to justify the effort.

Nevertheless, by taking a closer-look into the $E_R$ results for each image of the training set, one can see that the weighted estimation boosts the $E_R$ value for some but it also damages some others. More optimal results may be obtained by correctly combining both approaches by detecting where weighting helps (without ground truth) or by finding the optimal weights by regression. A table comparing the mincut's results and the weighted ones (child + node) at optimum $\lambda$, is shown below.

In our test case, the same weighting was applied to each branch of each tree and for all images, choosing only the closest ancestor, the closest descendant or the combination of both with the addition of the reference node.



(c) Mincut vs Weighted mincut. Results for the SLIC partition ($step = 2$, $reg = 0.005$).

Figure 4.3: Mincut's cuts plus weighted estimation for different partitions and ancestors/descendants weights.

| | | ***$E_R$ in dB*** | | | | | |
|---|---|---|---|---|---|---|---|
| | *Part.* | P1 Mincut | P1 Weigh. | P2 Minc. | P2 Weigh. | P3 Minc. | P3 Weigh. |
| Image number | 1 | -9.9030 | -10.6610 | -11.4586 | -11.4633 | -11.0196 | -11.1409 |
| | 2 | -16.7688 | -16.4863 | -16.8587 | -16.6168 | -17.0354 | -16.6697 |
| | 3 | -16.1576 | -16.2993 | -19.5374 | -19.3389 | -19.4763 | -19.4181 |
| | 4 | -10.8980 | -11.2292 | -12.3893 | -12.1000 | -12.6233 | -12.4086 |
| | 5 | -15.9969 | -15.7822 | -16.2120 | -16.1440 | -17.0017 | -16.5901 |
| | 6 | -13.0950 | -13.3553 | -14.9466 | -14.7162 | -14.7529 | -14.5738 |
| | 7 | -12.5875 | -12.9901 | -15.5373 | -15.1477 | -14.0979 | -13.9461 |
| | 8 | -15.1076 | -14.6237 | -16.2030 | -16.0249 | -16.4773 | -16.0931 |
| | 9 | -13.2038 | -13.1527 | -14.5611 | -14.2221 | -14.3017 | -13.9259 |
| | 10 | -10.9497 | -11.8697 | -16.8722 | -16.5893 | -16.7080 | -16.5732 |
| | Global | -13.1587 | -13.4213 | -15.1764 | -14.9618 | -15.0298 | -14.8338 |

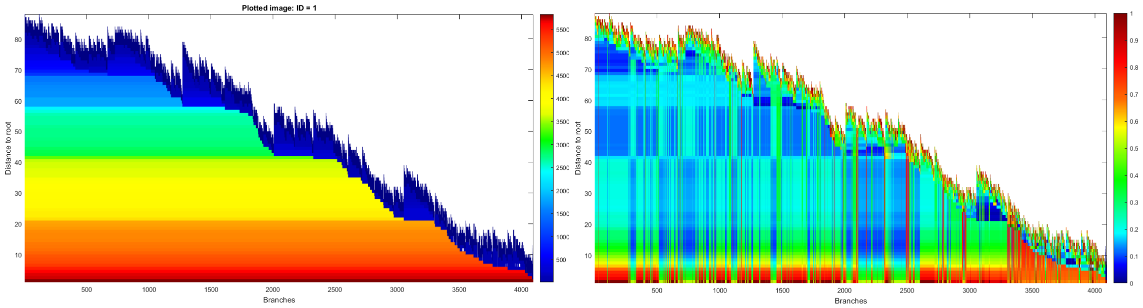
Table 4.1: Effect of the weighting for all images. Mincut results used as reference. Partitions: P1 = pixel, P2 = SLIC 4, 0.005 and P3 = SLIC 2, 0.005.

## 4.2 SAR_SE thresholded plus an optional weighted estimation

If we plot the dissimilarity measurement, $SAR\_SE$ in a branch representation, side by side versus the $E_R$, used as a reference, we can observe a brusque change in the $SAR\_SE$ value for each branch, going from dark blue to a very light blue. Seeing that, one may define a threshold to binarize the representation by keeping the nodes below the threshold and then select the node closer to the cut for each branch, generating a new partition.

The previous procedure has been tested with different thresholds, including small values, generating cuts closer to the leaves, and larger ones, closer to the root.

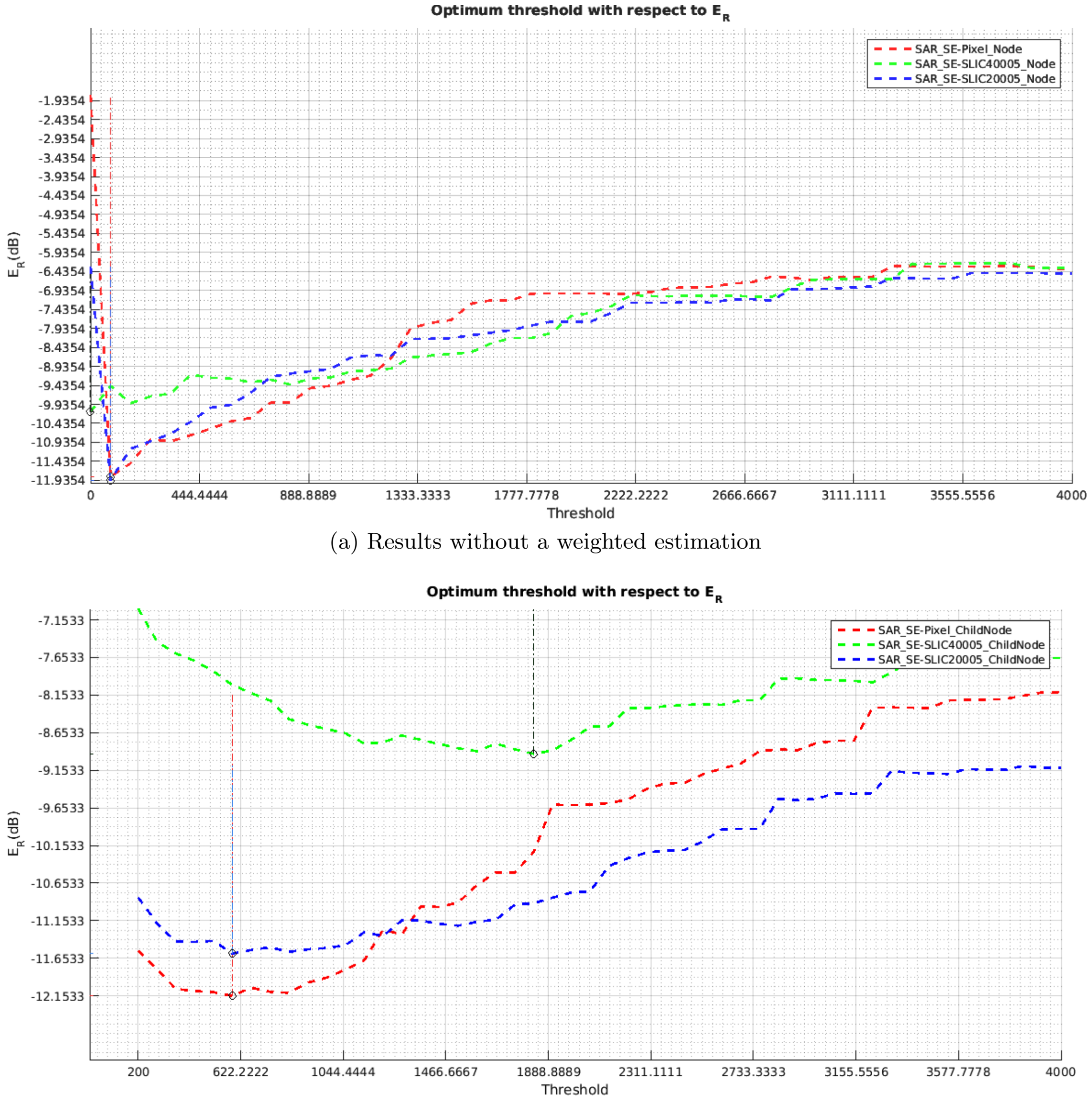
We have also considered an optional weighting with the selected node's child is used to evaluate if it improves the results in this scenario. We have 2 different graphs, depending if weighting is considered or not. Three different initial partitions are evaluated: pixel partition, starting SLIC with step = 4 and regularization 0.005 and a finer SLIC, with step = 2 and same regularization. The first graph a) shows the results for different thresholds without the weighted estimation,



(a) SAR_SE dissimilarity feature

(b) Relative error $E_R$

Figure 4.4: SAR_SE feature side-by-side with the relative error, both in branch representation.



(a) Results without a weighted estimation

(b) Results with a weighted estimation

Figure 4.5: Results obtained for the partition extracted from binarizing *SAR_SE* feature with and without weighting (only the partition' node/the node and its child)

populating the reconstructed regions with the mean of the selected nodes alone, extracted from the generated partition.

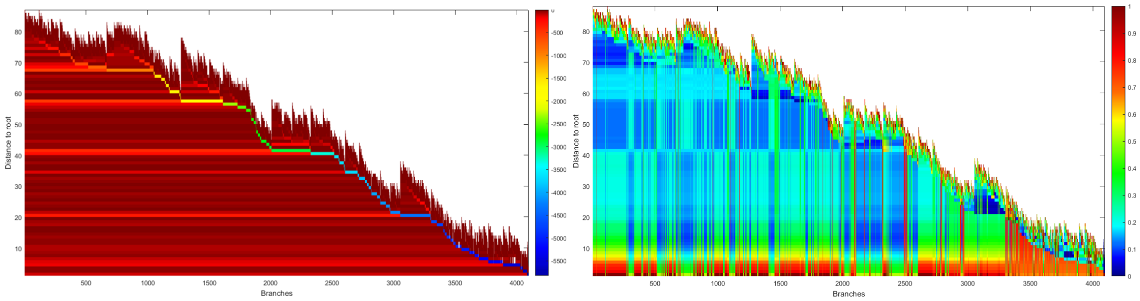
In this case, we can see that the relative error rapidly increases when we use relatively high values which are above the drastic change indicated in the *SAR_SE* graph before in Fig. 4.4. The lowest error obtained is above the current SOTA results obtained by mincut. One of the drawbacks of this test is that the selected threshold is fixed for all images instead of being adaptive and this limits its results.

However, seeing the current results, even if we selected an adaptive threshold each branch's structure, the results will unlikely outperform the mincut as the images from the test set are not that different from one another. Nevertheless, a more exhaustive analysis may be useful to highlight this technique's strengths and weaknesses which may lead to the development of finer techniques using a similar approach.

By looking at the second graph b), we can see that given the extracted partition, which by the location of the *SAR_SE* drastic increment in Fig. 4.4, may be too close to the leaves, choosing a weighted estimation with the selected node's child improves the previous results by 0.25 dB.
Once again though, the relative error is above that obtained by the mincut technique and a deeper analysis should be performed to obtain better results (e.g.: by adapting the threshold to the branch length of each processed BPT).

## 4.3 SAR_SE gradient thresholded plus an optional weighted estimation

Analogously, the procedure followed in the previous section can be used with the gradient of the *SAR_SE*. Again, this supposition arises from visualizing side-by side the 2 following branch representations, binarizing the branch representation and selecting the node that meets the criterion and is closer to the leaves:



(a) Gradient of *SAR_SE* dissimilarity feature

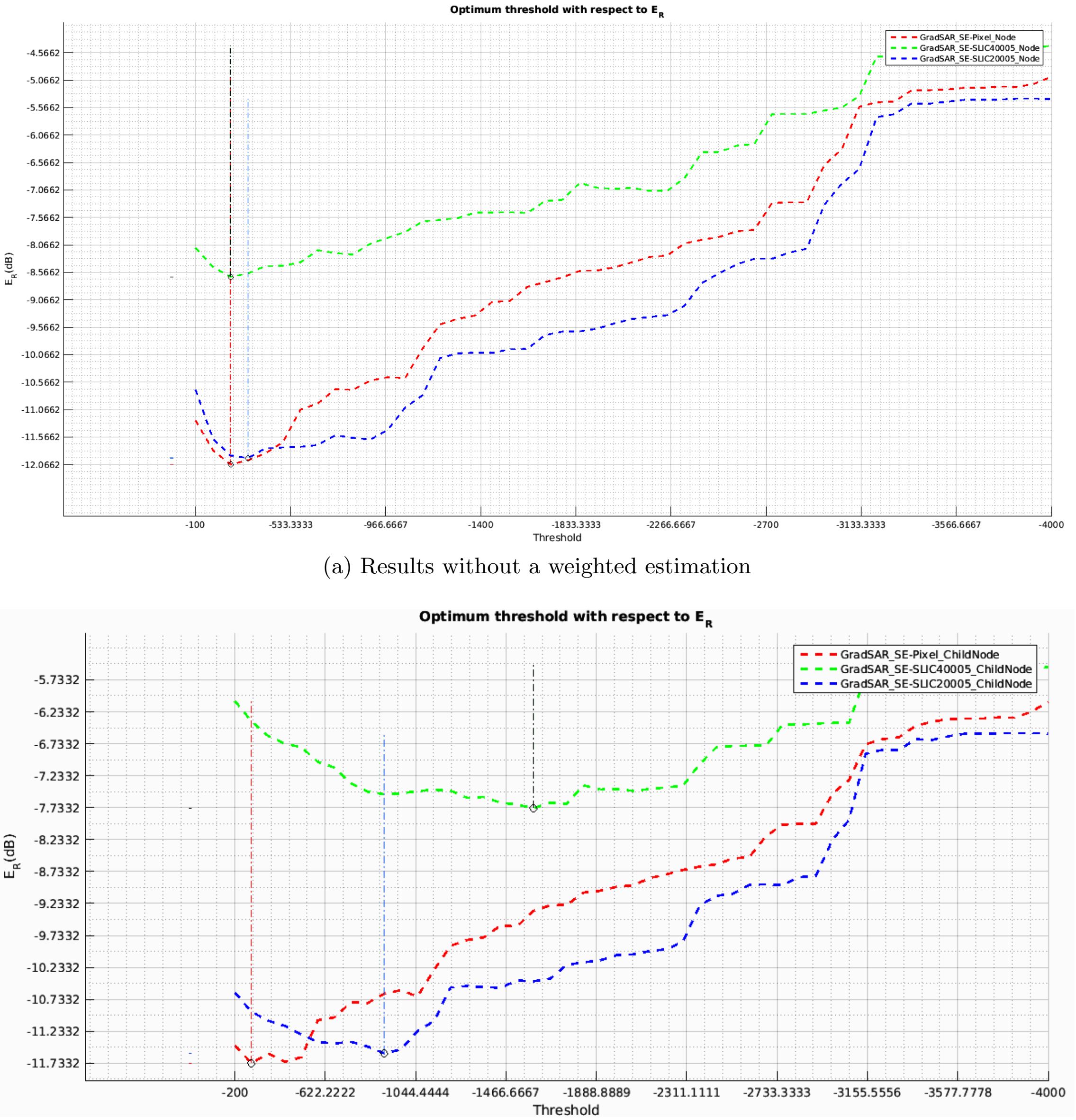
(b) Relative error $E_R$

Figure 4.6: Gradient of *SAR_SE* feature side-by-side with the relative error, both in branch representation.

The same tests performed in the previous section are repeated for this case, with the following results. Please take into account that the gradient for a child node is computed by subtracting its parent's *SAR_SE* value from its own, hence always being negative as *SAR_SE* increases the closer the root is.

In this case, the opposite behaviour with respect to the last section is observed, obtaining worse results for the weighted estimation. However, if we look at the lowest $E_R$ ($-12.15$ dB), both cases yield higher $E_R$ than the mincut and the results of the previous section, $-12.13$ dB and even though their parameters could be further optimized for each image, the results would unlikely outperform the SOTA.
On the whole, we can conclude that we need to develop a technique that is more adaptive to the BPT's structure for each image and also uses several parameters extracted from the BPT

to properly find more optimal nodes for a given leaf, studying in detail the branches structure. Machine Learning strategies may be a good approach to train a classifier with some useful descriptors extracted from the BPT with the goal of finding the best nodes for a given leaf by following an adaptive approach.



(a) Results without a weighted estimation

(b) Results with a weighted estimation

Figure 4.7: Results obtained for the partition extracted from binarizing the gradient of $SAR\_SE$ feature with and without weighting (only the partition' node/the node and its child).

# Chapter 5

# Techniques based on Machine Learning strategies

One possible approach to obtain a more adaptive and powerful technique that exploits several features extracted from the BPT representation is to use supervised Machine Learning strategies to train a classifier that, given a leaf, chooses the node (by looking at its ancestors) that bests represents it, that is, the one that yields the lower $E_R$ in the branch, generating a new partition that can outperform the current SOTA techniques.

To do so, we propose to use the Support Vector Machine (SVM) classifier to achieve our purpose given a set of descriptors for the nodes and leaves of the BPT. Given the limited amount of time corresponding to this TFG, it was not possible to fully develop a complete machine learning based strategy. As a result, we have contributed to the definition of the training and test datasets, a preliminary selection of possible features to be used by a classifier and an initial evaluation of the results. The idea is to use the outcome of this work in a forthcoming project and deepen the knowledge about the BPT-based strategy for speckle reduction.

## 5.1 Training scenario

We want our SVM classifier to detect for each leaf, the node in its branch that bests predicts it. To this end, each node should be characterized by features describing the node itself as well as the leaf it should predict.
An initial set of descriptors is proposed:

- *Region Area*: describes the number of pixels inside the image region enclosed by the processed node. The higher the node is in the tree, the bigger this parameter will be up to the root which has a region area equal to the image's size.
  This parameter is considered as a descriptor to take into account the size of the region to hopefully stop the SVM from selecting node's that are too small or big to correctly predict the leaf.

- *SAR_SE feature*: this feature is the sum of the norm of the difference between the pixel value and the norm of the region's mean. This feature enforces the homogeneity of regions and adds a regularization constant to penalize the region presence, avoiding the selection of the leaves as the nodes that minimize its value. This value is normalized by the norm of the region's mean to take into account the presence of speckle noise.
  The higher the node is in the tree, the higher this feature will be as the value of the pixel and the region's mean are very different as we are mixing very heterogeneous regions.

  This feature is used as a descriptor to describe the homogeneity of the candidate node and compare it with the other candidates.

- *Gradient of $SAR\_SE$ feature:* for each node, the difference between its value and its parent's is computed as shown in Following the definition of $SAR\_SE$, this feature will always be negative as the $SAR\_SE$ value for the parent will always be bigger that its children's.

  This feature complements the previous one as it tracks the evolution of the $SAR\_SE$ feature by computing the amplitude of the oscillations between a node and its parent.

- *Geodesic distance*: it is used while constructing the BPT in the merging sequence and it computes the distance between regions. Lower values mean that similar regions have been merged, whereas bigger values are obtained when two very different regions are merged.

  This parameter has been selected to stop the classifier from selecting nodes that have very big values (when possible) and which are, as a consequence, closer to the root and represent more heterogeneous and bigger regions.

- *Distance to mincut*: it denotes the number of nodes between the node selected by the mincut pruning and the one which is currently being analysed.

  This parameter is selected to help the classifier to select nodes closer to the mincut reference, as results evaluated with the global $E_R$ have shown that it makes very good choices.

- *Distance from node to leaf*: the number of nodes from the current node and the leaf that we want to predict.

  This descriptor may help to avoid selecting nodes which are too far (or too close) from the leaf to correctly predict it (too heterogeneous/too noisy).

- *Coherence matrix $T$*: it contains the full-polarimetric information of each node and leaf. It has been reshaped as a row 1x6 vector containing three real channels (power-related information) and three complex ones (correlation among the different elements).

  It may be a very useful descriptor as it contains the raw information for each image pixel and identifies the statistical distribution of the PolSAR image. *Leaf's parent descriptors*: in addition to those commented before, the descriptors for the parent of the leaf that we want to predict, are added as they describe a region close to the one defined by the leaf but reducing the amount of noise present in it.

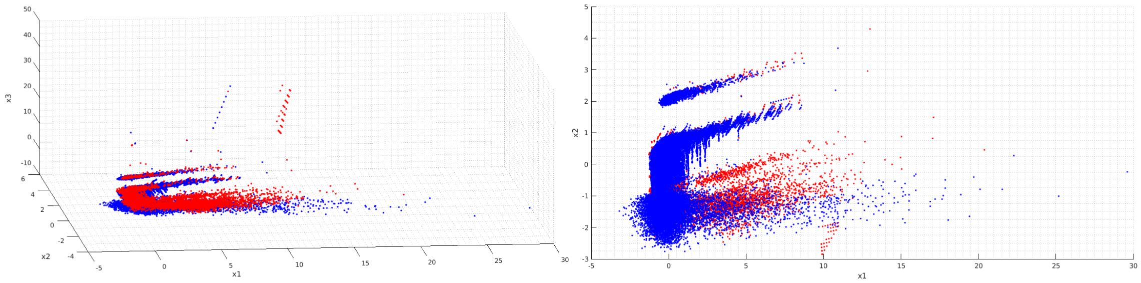
  This feature improves the accuracy of the leaf's descriptors by reducing the influence of the speckle noise.

Once the initial descriptors have been defined, one should define the training set, that is a set of nodes and their associated descriptors for each class. In our case, two classes are considered: good nodes and bad nodes, depending on whether they correctly predict the leaf or not.
For good nodes, we can simply select the examples as the first *numPosExamples* nodes for each leaf, that is, those which yield the lowest relative error values in the branch that is being processed.
For bad nodes, in order to have a comparable amount of examples as in the first class and avoid having unbalanced classes, one should impose a threshold on the $E_R$ of the remaining candidates to avoid labelling all of them as bad nodes.

One possible approach would be to only label as bad nodes those which are worse than what the mincut would give (our best reference). The former assumes that the five best nodes are much better than the mincut. The assumption is reasonable if we take a look at Fig. 4.2 where, for the pixel partition, the cut for $\lambda = 4$ yields the global best $E_R$ and we can see that there are several branches where it could have selected a better node as candidate.

For a pixel partition and one image of 64x64, the number of positive examples assuming a $numPosExamples$ for each leaf equal to 5 is 20,257 (25%). Similarly, setting the threshold defined by the mincut value to select the bad nodes as those below it yields a number of negative examples equal to 80,140 (75%). The rest of the initial dataset, 102,650 are better than the mincut reference but worse than the good nodes in class 1. To balance both classes, a more restrict threshold could be set to reduce the number of negative examples but that may be translated into not labelling as bad nodes (class -1) relatively bad candidates. Nevertheless, if the discarded nodes are not close to those examples found in either class, the quality of the training data example selection may not be deteriorated and will improve the system's performance.
The features values should be normalized by subtracting their mean and dividing by their standard deviation in order to give all of them the same importance.
For each training set image, the BPT must be created and populated with its parameters and then the descriptors should be computed. The optimal classifier should be found in cross-validation by evaluating the trained classifier with different parameters depending on the kernel used and selecting the one that gives the best results. F-score (harmonic mean of precision and recall) will be used as it evaluates at the same time the performance of both classes and takes into account the relative weight (the number of examples) of each class. Finally, the optimal classifier should be assessed with a test set not seen while training nor cross-validation to evaluate the system's performance.

The initial training set with one image of 64x64 for $numPosExamples = 5$ is plotted by selecting three and two principal components with a Principal Component Analysis (PCA), that selects the most important features in terms of energy.



(a) Feature Space 3D (b) Feature Space 2D

Figure 5.1: Training Feature Space with 2D and 3D. A Principal Component Analysis to select the features has been employed. Good nodes class in red (1), bad nodes class in blue (-1)

These graphs give an idea of the complexity of the classification problem at hand as we can see that the both classes are hardly separable *with this amount of features*. Please mind that the number of features used in the classification is way bigger (around 45 features if the leaf parent's descriptors are considered) and the problem may not be so complex as it appears in a 3D/2D projection plot. In addition, the use of other kernels aside from the linear may help to find discriminant functions that can improve the scatter between classes.

## 5.2 Results structure

Even though we did not have time to obtain optimal results and we could only do some initial tests, the procedure that should be followed to translate the classifier outputs into processed image and its related $E_R$, is explained here: in our case, we do not only need the label detailing the predicted class of that node with the current leaf, but also the probability of the decision so we can select the best node for each leaf even if the SVM returns more than a candidate. With the optimal node, an output image can be reconstructed using the mean of the optimal node so we can obtain the final $E_R$ and compare it with the SOTA techniques.

A first set of results that need to be further optimized with an extensive training and cross-validation with more than a single image, if possible, at full resolution (256x256) are shown. As said before, the number of positive examples per leaf was set to 5 and the procedure to assign the labels described in the previous section has been followed. A Gaussian kernel has been selected as, in the few tests we had time to do outperformed the results provided by the linear. Gamma is set to 2 and the corresponding F-score obtained by cross-validation using 5 folds is **0.846368**.

With the optimal classifier obtained in these initial tests, a test set 64x64 image obtained as the cut of a full-resolution image is used to do an initial evaluation on the system's performance. The results for the test are lower than expected at an f-score of **0.537127**. This may be due to the fact that in our test, the proportion of train and test data is 50/50 and more training data is needed in order to generalize well for the test set image.
To improve the f-score in test and reconstruct a more optimal image to compare with the SOTA techniques, more images are added to the training set, with a relative proportion of 60% train, 20% cross-validation and 20% test. As a consequence, the training time is augmented so we could not include those results in this thesis as the training is still in process.

Finally, to obtain the optimal results for a bigger dataset, this technique should be fully developed and optimized so we can test it with more PolSAR images, including non-synthetic ones so the system's overall performance can be determined.

# Chapter 6

# Conclusions, future improvements and feature lines of research

In this thesis a new approach using the BPT for speckle noise reduction has been presented and an analysis has been done to evaluate the potential of the BPT in such approach. In this regard, an exploratory work has been done before developing algorithms that rely on this approach. The main idea behind this new approach is to exploit the branch structure to find for each leaf the ancestor node that gives the best representation, that is, the lowest relative error with respect to the ground truth. This work has been divided into chapters, each of them dealing with an aspect of the potentiality assessment.

First, the influence of the initial partition has been explored and different types have been considered. These partitions have been analysed in terms of relative error, proving that a fine partition obtains lower relative error values despite introducing a higher computational load.

Secondly, a weighted estimation between more than a candidate has been proposed and analysed for different techniques, extracting different partitions. The results show that a weighted estimation can be helpful in some cases but it only introduces a small improvement. This indicates that this exploration technique must be refined in order to be more adaptive to the tree's structure so we can improve the obtained results by selecting the optimal node for each branch, combining useful nodes with optimal weights.

After that, new techniques that do not rely on the ground truth have been developed exploiting the branch exploration. The first set of techniques is based on the extraction of a partition from the BPT.
On the one hand, an extension of the *mincut* pruning algorithm including a weighted estimation has been conducted. The results show that weighting slightly improves the results in some cases but it also has no effect in others. This may be due to the fixed approach adopted, where the same weights were used for all branches combining the same number of nodes in the same fashion. On this subject, an adaptive weight and node selection may produce better results.
On the other hand, two new techniques that exploit the $SAR\_SE$ (eq.2.23) feature and its gradient with the addition of an optional weighted estimation, have been introduced. Both approaches select the nodes that are below a fixed threshold and closer to the leaves. The results show the same behaviour observed with the *weighted* mincut iteration and, as a consequence, similar conclusions can be drawn: a more adaptive technique is needed to smartly select good nodes and weights to produce a better estimation by choosing an adaptive threshold.

After that, from the study done in the previous two chapters, a more adaptive technique to select the optimal nodes for each leaf has been proposed. In this regard, Machine Learning strategies are considered and, as an initial approach, Support Vector Machines are chosen.

A set of descriptors has been selected for each node and leaf in order to create a matrix with training data. Similarly, the test and cross-validation set to optimize the parameters of the system have been created.
In addition, the feature space has been explored by reducing the number of features with the use of a Principal Component Analysis to evaluate the complexity of the classification problem at hand.
However, there has not been enough time to fine-tune the classifier and its parameters to obtain optimal results and reconstruct an image that could be compared with the State-of-the-art filtering techniques but all the steps previous to the training optimization have been completed and explained in chapter five.

Finally, some general conclusions are drawn concerning the whole project's scope:

- The potential of branch filtering is big as one can find really good candidates in the branch that improve the relative error obtained by the mincut's reference node.

- The intitial partition should be fine in order to have a large margin of improvement available.

- Classic techniques based on pruning (mincut, threshold on homogeneity) allow us to identify some good candidate nodes but not all.

- More sophisticated strategies are needed to find good nodes (e.g.: Machine Learning).

## Future lines of research

As future work, we may highlight the need of training and optimizing the SVM results to compare its results with all those commented during this thesis, including the State-of-the-art.
Another possible line of work would be to use Deep Neural Networks to search for useful features to select the optimal nodes. This technique would use the raw information contained in the pixels of the PolSAR image to train a Deep Neural Network that may obtain interesting results which may improve those obtained with the SOTA speckle noise filtering techniques.

All the code developed during this thesis will be available at https://github.com/imatge-upc/SAR-image-processing/tree/master/PolSAR-Denoising once is updated on January 30th, 2017 with its final version. As a private repository, its access is limited to those that are granted permissions by the repository master within the Imatge-upc research group.

# Bibliography

[1] Alberto Alonso-González. Multidimensional and temporal SAR data representation and processing based on Binary Partition Trees. *Thesis for the PhD in Signal Theory and Communications*, May 2014.

[2] Alberto Alonso-González, Carlos López-Martínez, and Philippe Salembier. Filtering and Segmentation of Polarimetric SAR Data Based on Binary Partition Trees. *IEEE transactions on Geoscience and Remote Sensing*, 50(2):593–605, February 2012.

[3] Ferran Pérez Gamonal. Speckle Noise Reduction in PolSAR images with Binary Partition Tree, Project Critical Review. *A Degree's thesis documentation*, November 2016.

[4] John C. Curlander and Robert N. McDonough. *Synthetic Aperture Radar : systems and signal processing*, volume XVII of *Wiley Series in Remote Sensing*. John Wiley & Sons, Inc., 1991.

[5] Philippe Salembier and Luis Garrido. Binary Partition Tree as an Efficient Representation for Image Processing, Segmentation, and Information Retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, April 2000.

[6] Chris Oliver and Shaun Quegan. *Understanding Synthetic Aperture Radar images*, volume XXVII of *Artech House Remote Sensing Library*. Artech House Publishers, 1998.

[7] Ridha Touzi, Armand Lopes, Jérôme Bruniquel, and Paris W. Vachon. Coherence Estimation for SAR Imagery. *IEEE transactions on Geoscience and Remote Sensing*, 37(1):135–149, 1999.

[8] Jong-Sen Lee, Mitchell R. Grunes, and Gianfranco de Grandi. Polarimetric SAR Speckle Filtering and Its Implication for Classification. *IEEE transactions on Geoscience and Remote Sensing*, 37(5):2363–2373, 1999.

[9] Jong-Sen Lee, Jem-Hung Wen, Thomas L. Ainsworth, Kun-Shan Chen, and Abel J. Chen. Improved Sigma Filter for Speckle Filtering of SAR Imagery. *IEEE transactions on Geoscience and Remote Sensing*, 47(1):202–213, 2009.

[10] Gabriel Vasile, Emmanuel Trouvé, Jong-Sen Lee, and Vasile Buzuloiu. Intensity-Driven Adaptive-Neighborhood Technique for Polarimetric and Interferometric SAR Parameters Estimation. *IEEE transactions on Geoscience and Remote Sensing*, 4(6):1609–1621, 2006.

[11] Charles-Alban Deledalle and Loïc Denis. Polarimetric SAR estimation based on non-local means. *IEEE Geoscience and Remote Sensing Symposium*, pages 2515–2518, 2010.

[12] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. *IEEE Transactions on PAMI*, 26(9):11241137, 2004.

[13] Philippe Salembier and Samuel Foucher. Optimum Graph-Cuts for Pruning Binary Partition Trees of Polarimetric SAR images. *IEEE transactions on Geoscience and Remote Sensing*, 54(9):5493–5502, May 2016.

[14] Samuel Foucher and Carlos López-Martínez. Analysis, Evaluation, and Comparison of Polarimetric SAR Speckle Filtering Techniques. *IEEE Transactions on Image Processing*, 23(4):1751–1764, April 2014.

[15] R.Achanta, A. Shaji, K.Smith, A.Lucchi, and P.Fua. SLIC Superpixels Compared to State-of-the-Art Superpixels Methods. *IEEE transactions on Pattern Analysis and Machine Learning*, 34(11):2274–2282, 2012.

[16] Philippe Salembier, Sergi Liesegang-Maria, and Carlos López-Martínez. Graph signal processing for object detection on tree-based representation of images: Application to ship detection in SAR images.

[17] Xinping Deng. Texture Analysis and Physical Interpretation of Polarimetric SAR Data. *Thesis for the PhD in Signal Theory and Communications*, May 2016.

[18] European Spatial Agency. PolSARpro software version 5.0.4. *https://earth.esa.int/web/polsarpro*, July 2015.

[19] Samuel Foucher and Carlos López-Martínez. An Evaluation of PolSAR Speckle Filters. *IEEE Geoscience and Remote Sensing Symposium*, pages 5089–5092, 2012.

[20] Ferran Pérez Gamonal. Speckle Noise Reduction in PolSAR images with Binary Partition Tree, Project Proposal and Work Plan. *A Degree's thesis documentation*, Oct 2016.

[21] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[22] Sergi Liesegang Maria. SAR Remote Sensing Image Analysis with Max-tree Representation for Ship Monitoring. *A Degree's Thesis*, July 2015.

# List of abbreviations

| | |
|---|---|
| SAR | Synthetic Aperture Radar |
| PolSAR | Polarimetric Synthetic Aperture Radar |
| BPT | Binary Partition Tree |
| BSA | Back Scattering Alignment |
| $S$,$k$ | Scattering Matrix/Vector |
| $C$ | Covariance Matrix (Lexicographic Basis) |
| $T$ | Coherence Matrix (Pauli Basis) |
| $Z$ | Estimated Covariance Matrix (Sample Covariance Matrix) |
| pdf | Probability Density Function |
| $E\{\cdot\}$ | Expected value of " $\cdot$ "(Statistics) |
| SOTA | State-of-the-art (techniques) |
| MLE | Maximum Likelihood Estimation |
| $\sigma_n$ | Standard deviation of the speckle noise $n$ (measures the Speckle level) |
| $N$ | Number of looks (also referred to as $L$) |
| $MMSE$ | Minimum Mean Square Error |
| $ENL$ | Equivalent Number of Looks |
| $Z_N$ | $N$-th percentile |
| NL | Non Local (means) |
| IDAN | Intensity-Driven Adaptive Neighbourhood |
| SLIC | Simple Linear Iterative Clustering (to generate superpixels) |
| GT | Ground Truth |
| SVM | Support Vector Machine |
| PCA | Principal Component Analysis |

# Colophon

THIS BACHELOR THESIS WAS TYPESET using LATEX, originally developed by Leslie Lamport and based on Donald Knuth's TEX. A template, which can be used to format a thesis with this look and feel has been released by the Faculty of Mathematics and Physics of the Charles University in Prague and whose creators are Martin Mares, Arnost Komarek and Michal Kulich. One can found it online at two of the most known LATEXtemplates repositories, Overleaf or ShareLatex. This version of the template is no longer available at the University's website. Despite this, a newer template from them can be found on their website. Finally, the edition of the former template has been performed thanks to the multi-platform LATEXeditor, TEXStudio.