# Morphological Hierarchies for Satellite Image Time Series

Çaglayan Tuna

DOCTORAT BRETAGNE LOIRE MATHSTIC

Université Bretagne Sud

# THÈSE DE DOCTORAT DE

L'UNIVERSITE BRÉTAGNE SUD

ÉCOLE DOCTORALE Nº 601
*Mathématiques et Sciences et Technologies de l'Information et de la Communication*
Spécialité : *Signal, Image, Vision*

Par

**Caglayan TUNA**

# Morphological Hierarchies for Analysing Satellite Image Time Series

**Thèse présentée et soutenue à Vannes, le 16/12/2020**
**Unité de recherche : IRISA**
**Thèse Nº : 567**

## Rapporteurs avant soutenance :

| | | |
|---|---|---|
| Philippe SALEMBIER | Prof. | Universitat Politecnica de Catalunya |
| Francesca BOVOLO | Prof. | Fondazione Bruno Kessler |

## Composition du Jury :

| | | | |
|---|---|---|---|
| Président : | Nicolas COURTY | Prof. | Université Bretagne Sud |
| Examinateurs : | Nicolas MEGER | Prof. | Université Savoie-Mont Blanc |
| | Dino IENCO | CR HDR | IRSTEA Institute |
| Dir. de thèse : | Sébastien LEFEVRE | Prof. | Université Bretagne Sud |
| Encadr. de thèse : | François MERCIOL | MCF | Université Bretagne Sud |
| Encadr. de thèse : | Charlotte PELLETIER | MCF | Université Bretagne Sud |

## Invité(s) :

| | | |
|---|---|---|
| Alain GIROS | ING | CNES |
| Guillaume HAJDUCH | Dr | CLS |

# Acknowledgement

I would like to thank my main supervisor, Sébastien Lefèvre, for his scientific, technical and personal support. His knowledge and expertise allowed me to improve myself and learn from my mistakes. It was a great opportunity to work with him during my PhD journey. I also would like to thank my co-supervisors François Merciol and Charlotte Pelletier for their support, guidance and constructive discussions. I have learned greatly and increased my capacity as a graduate student with their assistance.

I am deeply grateful to meet and work with Alain Giros from CNES during these last 3 years. His extensive knowledge and experience regarding the remote sensing area not only enabled me to improve myself but also greatly advanced this work. His suggestions and excellent scientific challenges cultivated my scientific perspective. It was an unforgettable chance to work with him.

I would also like to thank Guillaume Hajduch and Rodolphe Vadaine from the CLS company for their time and suggestions to improve my work. My visits to their company have broaden my view on the real-life applications of Earth Observation.

I would also like to thank all of my friends, Titouan, Behzad, Ahmed, Florent who have been with me in my most joyful and distressed times. The support of my friends has been vital in keeping me on the right track while I was trying to cope with the challenges of my PhD journey.

My special thank is for you, Begüm. I always feel very lucky to be your friend, confidant and partner in crime. As a professional perspective, I believe that I wouldn't succeed on anything without your helps that motivated me. Thank you for standing by my side.

Finally, my appreciations and gratitude go to my parents and my sister. Even though we were away from each other, their personal and mental supports had been always with me. Thank you so much for encouraging and leading the way for me to pursue a career in science and being there for me.

Lastly, I would like to acknowledge CNES and CLS for supporting me and this project.

# Resumé (Français)

## Motivation

L'observation de la Terre (OT) est l'un des moyens importants pour comprendre notre planète et sa dynamique. Les images satellites nous permettent de surveiller notre planète grâce à une technologie en constante amélioration offrant une résolution spatiale, spectrale et temporelle plus élevée et davantage de données disponibles. Un tel volume d'images satellites nécessite des algorithmes d'analyse d'image plus efficaces. L'analyse d'images par télédétection fournit des informations précieuses pour plusieurs applications telles que la cartographie de la couverture terrestre, la planification urbaine, la surveillance agricole, la gestion des catastrophes [76] *etc.*

En raison du changement climatique et de la crise sanitaire mondiale, la surveillance des ressources environnementales est beaucoup plus importante qu'auparavant. Bien que les données de télédétection soient utiles pour surveiller l'effet du changement climatique [189], peu de méthodes efficaces sont capables de tirer le meilleur parti de leur structure spatio-temporelle. Par ailleurs, un suivi des ressources pour lutter contre la pauvreté dans le monde est nécessaire. Enfin, des systèmes précis d'évaluation des catastrophes peuvent aider à trouver de meilleures solutions pour gérer les catastrophes naturelles [93]. Afin de faire face à ces situations difficiles, des données comparables sont nécessaires. En effet, de telles données comparables proviendraient d'une acquisition faite par la même scène mais à des dates différentes. Plus précisément, ces images multi-temporelles forment un empilement temporel autour de la même scène et sont appelées séries temporelles d'images satellites (STIS). Cette information temporelle apporte des

connaissances périodiques sur notre planète et donne l'occasion de comprendre son évolution dans le temps. Par conséquent, l'analyse STIS gagne en importance dans la communauté de la télédétection. STIS peut être vu comme une pile d'images (une image par date) ou comme une grille de séries temporelles où chaque pixel est associé à sa série temporelle. Les STIS ont été utilisés dans de nombreuses applications d'utilisation des terres / de couverture terrestre, par exemple, pour surveiller la dynamique de la végétation [161] ou les plans d'eau [163], détecter les changements de couverture terrestre [193], analyser les effets à long terme du changement climatique [64], suivi des dommages causés par une catastrophe [173]. Récemment, plusieurs méthodes ont été proposées pour analyser les STIS telles que la déformation temporelle dynamique [129], le cube de données spatio-temporelles [183], l'apprentissage en profondeur [121] ou les modèles séquentiels [98] *etc*. Cependant, il existe encore un vaste domaine de recherche pour surmonter les défis posés par STIS et trouver des méthodes efficaces pour les utiliser dans des applications du monde réel.

Le terme de temps de revisite fait référence au délai entre deux acquisitions sur le même point sur Terre et un temps de revisite plus court signifie plus d'informations disponibles sur l'évolution de la Terre. Le temps de revisite est également appelé résolution temporelle dans la littérature [28]. Certaines missions satellitaires comme Sentinel s'appuient sur plus d'un satellite, formant ainsi une constellation afin de réduire le temps de visite. Généralement, alors que le temps de visite d'un satellite diminue, la résolution spatiale de ses images augmente généralement. Une résolution spatiale plus élevée et un temps de visite plus court ouvrent la voie à l'utilisation simultanée des informations spatiales et temporelles. Par exemple, les satellites Sentinel-1 et Sentinel-2 capturent des images de toutes les surfaces émergées tous les cinq jours. Au contraire, certains fournisseurs d'images à très haute résolution tels que Pléiades peuvent acquérir des images en un temps de visite plus court (1 jour) mais uniquement pour des régions prédéfinies. La résolution spatiale des images satellites Sentinel va de 5 à 20 m pour les différents modes de fonctionnement de Sentinel-1 et de 10 à 60 m pour les différents canaux de Sentinel-2 [14]. Bien qu'il existe plusieurs satellites offrant une résolution temporelle élevée tels

que MODIS et Landsat, leur résolution spatiale n'est pas adéquate pour certaines applications. Par conséquent, les chercheurs du STIS accordent plus d'attention aux données Sentinel récemment en raison de leur résolution spatiale plus élevée.

Pour être efficace, l'analyse STIS doit utiliser des relations spatiales entre les pixels, qui donnent plus d'informations par rapport aux informations basées sur les pixels. Ces relations pourraient améliorer l'utilisation de STIS dans de nombreuses applications. Aujourd'hui, la représentation hiérarchique basée sur la morphologie mathématique (MM) est un cadre efficace pour l'analyse spatiale avec sa capacité de filtrage d'image et de segmentation [78]. Les représentations hiérarchiques ont prouvé depuis plusieurs années leur intérêt pour la modélisation et le traitement d'images satellitaires et aériennes à une seule image [43]. Dans de nombreuses œuvres de pointe en traitement d'image, ces représentations hiérarchiques sont implémentées à travers des structures arborescentes. Ils sont utilisés dans de nombreuses applications, telles que la détection d'objets [177], la simplification d'image [154], l'extraction de caractéristiques [115], la récupération d'images [17] et la classification d'images [56, 33] . Il existe également des travaux les utilisant pour des applications vidéo comme [102] qui prouvent également leur potentiel pour exploiter la dimension temporelle.

L'objectif de cette thèse est d'adapter les représentations arborescentes au STIS et d'extraire des informations de ces arbres afin de les utiliser pour des applications de cartographie de l'occupation du sol, de détection de changement ou de reconnaissance de formes spatio-temporelles. Seules quelques études [117, 177, 50] ont tenté d'étendre les représentations arborescentes à la dimension temporelle avec quelques applications limitées. Par rapport à ces travaux précédents, nous visons à étendre les représentations arborescentes avec de nouvelles stratégies, efficaces pour relever les défis STIS et plusieurs applications OT. Cette thèse propose également une étude comparative des stratégies possibles pour construire de tels arbres.

# Contributions

Comme mentionné dans la section précédente, l'objectif principal de cette thèse est de se concentrer sur l'analyse STIS et de contribuer à ce domaine de recherche avec des représentations arborescentes. Trois étapes principales sont suivies au cours de cette thèse :

1. Création d'arbres à partir de STIS
2. Création de descripteurs d'entités en calculant des attributs à partir de ces arbres
3. Utilisation de ces fonctionnalités pour l'analyse STIS dans des scénarios d'OT réalistes

Dans les paragraphes suivants, nous expliquerons brièvement nos contributions pour chacune de ces étapes.

L'ajout d'une dimension temporelle à une image numérique 2D offre de nombreuses alternatives pour construire une hiérarchie morphologique. Les dimensions spatiale et temporelle peuvent être considérées simultanément ou successivement. L'arbre peut contenir des nœuds définis sur le domaine spatial, temporel ou spatio-temporel. Pour chaque domaine, nous expliquerons comment il peut être construit à partir d'une séquence d'images, en s'appuyant sur des définitions et algorithmes existants précédemment introduits sur des images 2D avec des pixels définis comme des scalaires (pour les images panchromatiques) ou des vecteurs (pour des données multivariées). En bref, nous considérons trois stratégies différentes : (1) décrire chaque pixel par une série temporelle de pixels avant de construire l'arbre correspondant, (2) construire un arbre pour chaque image de la série séparément, et (3) construire un seul arbre pour le série temporelle entière. Cette étape peut être considérée comme une étude fondamentale pour des recherches ultérieures, qui comprendront une représentation hiérarchique dans le domaine temporel. Nous allons également introduire des algorithmes de streaming temporel pour chaque stratégie, pour faire face à la situation qui se produit lorsqu'une nouvelle image est ajoutée à la série temporelle. Cette contribution a été publiée dans Pattern Recognition Letters (numéro spécial sur les représentations hiérarchiques : nou-

veaux résultats et défis pour l'analyse d'image) [169]. Nous comparerons ensuite ces stratégies en fonction des propriétés structurales des arbres correspondants avec différents critères. Nous avons proposé cette comparaison dans ICPR [165].

Après avoir construit l'arborescence, différents types d'attributs peuvent être calculés. Chaque nœud d'arbre peut être équipé de ces attributs afin de caractériser le contenu de l'arbre. Dans cette thèse, les attributs spatiaux standard sont étendus dans la dimension temporelle et certains nouveaux attributs spatio-temporels sont introduits. Pour utiliser ces attributs, nous nous appuierons sur deux opérations courantes basées sur des représentations arborescentes : le filtrage et les spectres de motifs. Le filtrage d'un arbre avec certains attributs est utilisé pour faire face aux défis STIS tels que les valeurs aberrantes ou le bruit. Les spectres de motifs sont étendus à la dimension temporelle pour surveiller les objets dans le temps. Nous avons présenté une telle contribution dans ISPRS [168].

Enfin, nous proposerons de nouveaux algorithmes utilisant ces méthodes d'extraction de caractéristiques pour des applications d'OT spécifiques telles que la cartographie de la couverture terrestre, la détection des changements et la reconnaissance de formes spatio-temporelles. Pour la cartographie de la couverture terrestre, nous nous baserons sur des profils morphologiques tels que les profils d'attributs (PA) ou les profils d'entités (PE) avec chaque stratégie de construction d'arbres. La croissance urbaine sera observée à l'aide d'une opération de filtrage pour relever le défi de la couverture nuageuse. Le suivi des attributs dans l'arborescence des séries temporelles SAR conduira à l'introduction de deux algorithmes de détection des inondations. Enfin, les marées seront observées en exploitant les spectres de motifs des régions aquatiques. Chaque demande sera rapportée avec une analyse quantitative / qualitative et sera comparée aux méthodes de pointe. Ces contributions sont présentées dans JURSE [164], IGARSS [167], ISPRS [168] et Multitemp [166].

# Contents

# Symbols

| | |
|---|---|
| $G$: | Graph |
| $V$: | Set of nodes |
| $\mathcal{V}$: | Set of values |
| $E$: | Edges |
| $U$: | Arc |
| $\mathcal{T}$: | Tree |
| $I$: | Gray scale image |
| $I'$: | Reconstructed image |
| $x$: | Spatial coordinate |
| $\mathcal{N}$: | Neighbour list |
| $I_t(x)$: | Spatio-temporal pixel |
| $I_\tau$: | Image sequence |
| $I_\varsigma$: | Time-invariant image |
| $v$: | Intensity |
| $\lambda$: | Level, threshold |
| $L$: | Level set |
| $C$: | Connected component |
| $C_{\lambda,\tau}$: | Spatio-temporal connected component |
| $H$: | Hole filling operator |
| $S$: | Shape |
| $\mathcal{A}$: | Attribute |
| $\phi$: | Thickening operator |
| $\gamma$: | Thinning operator |
| $\zeta$: | Filtering operator |
| $h$: | Attribute threshold |
| $\sigma$: | Standard deviation |
| $n$: | Length of time series |
| $f$: | Feature |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## Contents



## 1.1 Motivation

### 1.1.1 Earth Observation

Earth observation (EO) is one of the important ways to understand our planet and its dynamics. Satellite images enable us to monitor our planet with continuously improving technology providing higher spatial, spectral and temporal resolution and more available data. Such a high volume of satellite images calls for more efficient image analysis algorithms. Remote sensing image analysis provides valuable information for several applications such as land cover mapping, urban planning, agricultural monitoring, disaster management [76] *etc.* Figure 1.1 illustrates an example of EO process.

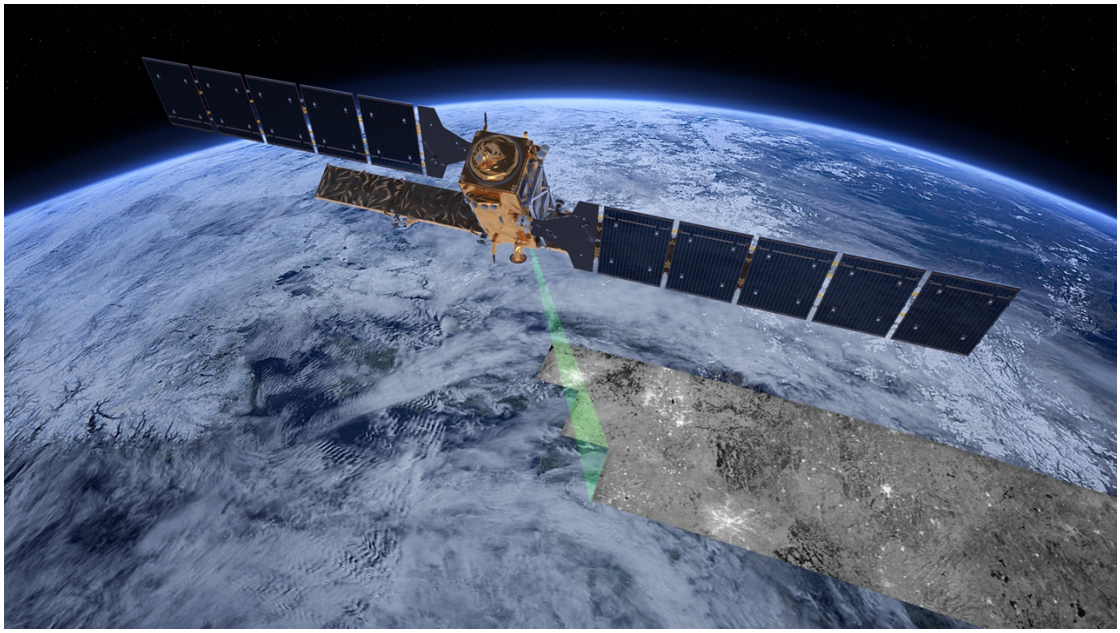


Figure 1.1 – EO from a satellite. Source: https://www.geospatialworld.net

### 1.1.2 Satellite image time series

As a consequence of climate change and global health crisis, monitoring environmental resources is much more important than before. Although remote sensing data are helpful to monitor effect of climate change [189], few efficient methods are able to make the most of their spatio-temporal structure. Besides, monitoring resources to fight against poverty in the world is necessary. Lastly, accurate disaster assessment systems may help for better solutions to manage natural disasters [93]. In order to deal with these challenging situations, comparable data is needed. Indeed, such comparable data would come from acquisition made by the same scene but at different dates. More precisely, these multi-temporal images form a temporal stack about the same scene and are called satellite image time series (SITS). This temporal information brings periodical knowledge on our planet and gives opportunity to understand its evolution through time. Therefore, SITS analysis is gaining importance in remote sensing community. SITS can be seen as a stack of images (one image per date) or as a grid of time series where each pixel is associated with its time series. SITS have been used in many land use/land cover applications, e.g., monitoring vegetation dynamics [161] or water bodies [163], de-

tecting land cover changes [193], analyzing long-term effects of climate change [64], tracking disaster damages [173]. Recently, several methods have been proposed to analyze SITS such as dynamic time warping [129], spatio-temporal data cube [183], deep learning [121], or sequential patterns [98] *etc.* However, there is still a broad research area to overcome challenges posed by SITS and to find efficient methods for using them in real world applications.

### 1.1.3 Current and future satellite missions

Table 1.1 provides a list of active EO satellites and some announced future missions. In this table, the revisit time term refers to the delay between two acquisitions over the same point on Earth and shorter revisit time means more available information on Earth evolution. The revisit time is also named as temporal resolution in the literature [28].

Some satellite missions such as Sentinel rely on more than one satellite, thus forming a constellation in order to decrease the revisit time. While the revisit time of a satellite decreases, spatial resolution of its images increases generally. Higher spatial resolution and shorter revisit time lead the way to use spatial and temporal information simultaneously. For instance, Sentinel-1 and Sentinel-2 satellites capture images of all emerged surfaces every five days. In contrary, some very high resolution image providers such as Pleiades can acquire images in shorter revisit time (1 day) but for only predefined regions. The spatial resolution of Sentinel satellite images ranges from 5 to 20m for the different operational modes of Sentinel-1 and from 10 to 60m for the different channels of Sentinel-2 [14]. Although there are several satellites providing high temporal resolution such as MODIS and Landsat, their spatial resolution is not adequate for some applications. Therefore, SITS researchers give more attention to Sentinel data recently due to their higher spatial resolution.

### 1.1.4 Spatial information with hierarchical representations

In order to be efficient, SITS analysis should use spatial relationships between the pixels, which give more information compared to pixel-based information. These relationships could improve the use of SITS in many applications. Today, hi-

| Satellite | Provider | Revisit time (day) | Spatial Resolution (m) | Launch year |
|---|---|---|---|---|
| Landsat 7 | NASA | 16 | 30 | 1999 |
| MODIS | NASA | 2 | 250-500 | 1999 |
| PlanetScope | Planet Labs | 1 | 3 | 2009 |
| TerraSAR-X | DLR | 2.5-11 | 1 | 2010 |
| Landsat 8 | NASA | 16 | 30 | 2013 |
| SPOT 7 | CNES | 1-3 | 1.5 to 6 | 2013 |
| Pleiades 1A-1B | CNES | 1 | 0.5 to 2 | 2013 |
| ALOS-2 | JAXA | 14 | 1 to 3 | 2014 |
| SkySAT | Planet Labs | 1 | 0.8 to 1 | 2014 |
| Sentinel 1-2 | ESA | 5 | 5-60 | 2015 |
| GeoEye 1 | Digital Globe | 3 | 0.5 to 2 | 2016 |
| PRISMA | ISA | 6 | 30 | 2019 |
| 2. COSMO-SkyMed | ISA | 5 | 3 to 5 | 2019 |
| EnMAP | DLR | 27 | 30 | 2020 |
| Landsat 9 | NASA | 16 | 30 | 2020 |
| Worldview Legion | Digital Globe | 1 | 0.3 | 2021 |
| NISAR | NASA/ISRO | 12 | 5 to10 | 2021 |
| Pleiades Neo | Airbus | 1 | 0.5 | 2022 |

Table 1.1 – Active EO Satellites and Some Future Missions

erarchical representation based on Mathematical Morphology (MM) is an efficient framework for spatial analysis with its image filtering and segmentation ability [78]. Hierarchical representations have proven from several years their interest in modelling and processing single frame satellite and aerial images [43]. In many state-of-the-art works in image processing, these hierarchical representations are implemented through tree based structures. They are used in many applications, such as object detection [177], image simplification [154], feature extraction [115], image retrieval [17] and image classification [56, 33]. There are also some works using them for video applications such as [102] which also proves their potential for exploiting the temporal dimension.

The aim of this thesis is to adapt the tree-based representations to SITS and to extract information from these trees in order to use them for land cover mapping, change detection or spatio-temporal pattern recognition applications. Only a few studies [117, 177, 50] have tried to extend tree-based representations to the temporal dimension with some limited applications. Compared to these previous

works, we aim at extending tree representations with novel strategies, which are efficient to tackle SITS challenges and several EO applications. This thesis also provides a comparative study of the possible strategies for building such trees.

## 1.2 Contributions

As mentioned in the previous section, the main goal of this thesis is to focus on SITS analysis and contribute to this research area with tree based representations. Three main steps are followed during this thesis:

1. Building trees from SITS
2. Creating feature descriptors by computing attributes from these trees
3. Using these features for SITS analysis in realistic EO scenarios

In the following paragraphs, we will briefly explain our contributions for each of these steps.

Adding a temporal dimension to a 2D digital image offers many alternatives to build a morphological hierarchy. The spatial and temporal dimensions may be considered simultaneously or successively. The tree may contain nodes defined on the spatial, temporal, or spatial-temporal domain. For each domain, we will explain how it can be built from an image sequence, relying on existing definitions and algorithms previously introduced on 2D images with pixels defined as scalars (for panchromatic images) or vectors (for multivariate data). In short, we consider three different strategies: (1) describing each pixel by a pixel time series before constructing the corresponding tree, (2) building a tree for each image of the series separately, and (3) building only one tree for the whole time series. This step can be considered as a seminal study for further research, which will include hierarchical representation in temporal domain. We will also introduce temporal streaming algorithms for each strategy, to deal with the situation occuring when a new image is appended to the time series. This contribution was published in Pattern Recognition Letters (special issue on Hierarchical Representations: New Results and Challenges for Image Analysis) [169]. We will then compare these strategies according to the structural properties of the corresponding trees with different criteria. We proposed this comparison in ICPR [165].

After building the tree, different kinds of attributes can be calculated. Every tree node can be equipped with these attributes in order to characterize the tree content. In this thesis, standard spatial attributes are extended in the temporal dimension and some novel spatio-temporal attributes are introduced. In order to use these attributes, we will rely on two common operations based on tree representations: filtering and pattern spectra. Filtering a tree with some attributes is used to deal with SITS challenges such as outliers or noise. Pattern spectra is extended to the temporal dimension to monitor objects through time. We presented such a contribution in ISPRS [168].

Lastly, we will propose novel algorithms using these feature extraction methods for specific EO applications such as land cover mapping, change detection and spatio-temporal pattern recognition. For land cover mapping, we will rely on morphological profiles such as Attribute Profiles (AP) or Feature Profiles (FP) with each tree building strategy. Urban growth will be observed using a filtering operation to tackle the cloud cover challenge. Tracking the attributes through the tree for SAR time series will lead to the introduction of two flood detection algorithms. Finally, tides will be observed by exploiting pattern spectra of water regions. Each application will be reported with quantitative/qualitative analysis and will be compared with the state-of-the-art methods. These contributions are presented in JURSE [164], IGARSS [167], ISPRS [168] and Multitemp [166].

## 1.3 Organization

The details of the aforementioned contributions will be explained in six chapters of this thesis. Figure 1.2 illustrates the workflow of this thesis with related illustrations.

— In chapter 2, the different tree models and their mathematical formulation will be discussed. More precisely, a tree building from a single frame image will be explained in detail. No temporal dimension will be included at this stage.

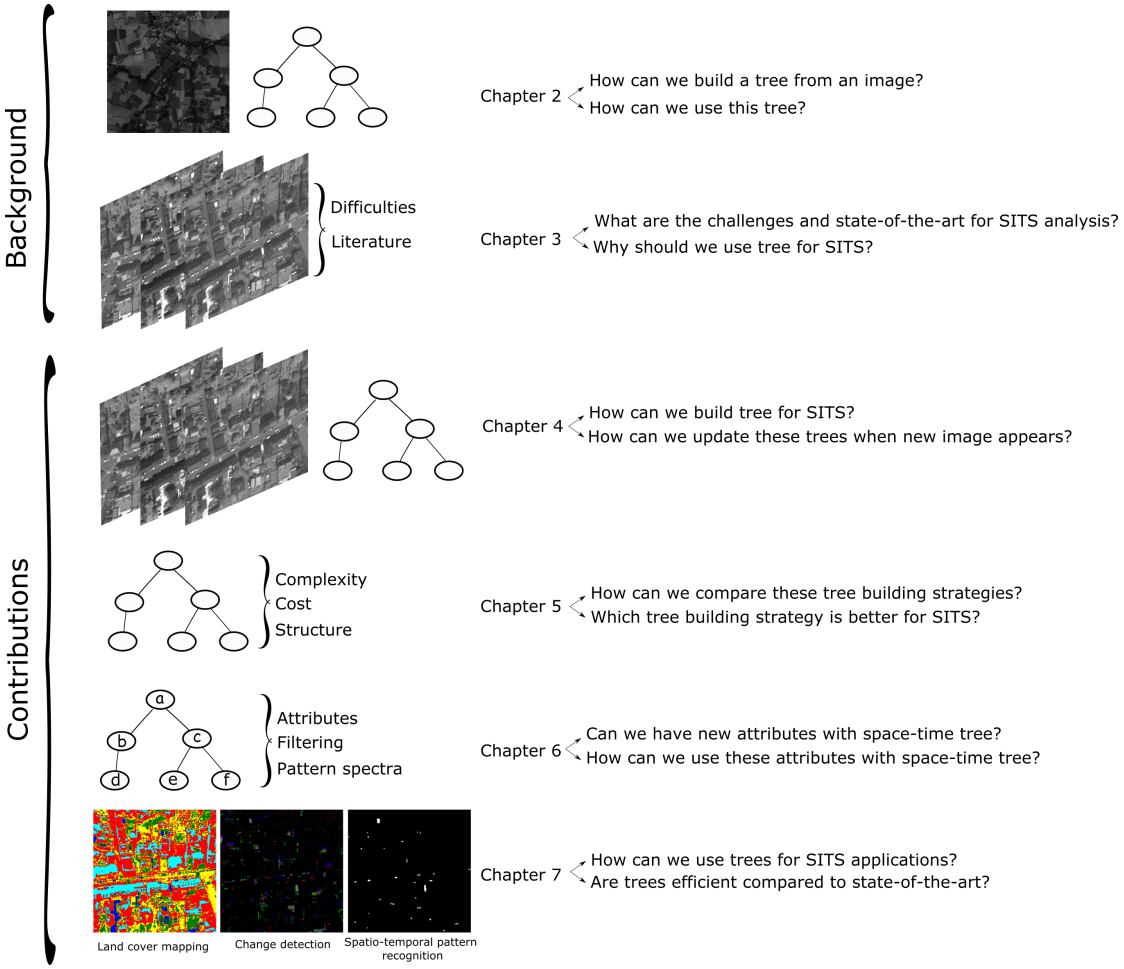



Figure 1.2 – Thesis workflow

— An overview of the challenges posed by SITS and works related to our thesis will be presented in chapter 3. Related works are divided into three categories: (1) time series analysis, (2) spatial information in time series, and (3) hierarchical representation for time series.

— Chapter 4 provides the first methodological contributions, where various strategies to analyze SITS through trees will be explored. This chapter also provides a streaming algorithm for each strategy.

— A structural comparison of tree building strategies for SITS will be examined in chapter 5.

— Temporal based attribute calculation from tree representations will be explained in chapter 6. Spatio-temporal attributes, pixel based feature extraction from SITS and Pattern spectra for SITS will be discussed.
— Datasets and experiments will be illustrated in chapter 7. Experimental results will be demonstrated for some time series application domains: land cover mapping, change detection and spatio-temporal pattern recognition.
— Finally, chapter 8 concludes the thesis and provides directions for future work.

# Chapter 2

# Fundamentals of Hierarchical Representations

## Contents

This chapter presents mathematical backgrounds of hierarchical data structures and morphological hierarchies. As we mentioned in the introductory chapter, we will use tree representation as a hierarchical data structure during this thesis. Finally, we will review existing attributes that can be computed from a tree and recall how to process trees with these attributes.

## 2.1 Hierarchical data structures

### 2.1.1 Graph

A graph $G$ consists of $V$ and $U$, where $V$ denotes a set of vertices, also called nodes, and $U$ is a set of arcs between the nodes [13]. Two vertices joined by an arc are said to be adjacent. If the arcs between nodes have a direction, the graph is known as a directed graph, otherwise it is an undirected one. A graph example can be seen in figure 2.1 where $A, B, C$ and $D$ are nodes.

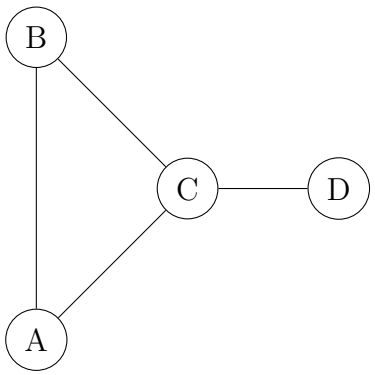



Figure 2.1 – A sample graph $G = (V, U)$ with $V = \{A, B, C, D\}$ and $U = \{(A, B), (A, C), (A, D), (B, D)\}$.

An edge is defined as a directed arc. If the graph is directed, it can be formulated as:

$$G = (V, E) \tag{2.1}$$

where $E$ represents the set of edges.

When representing an image $I$ as a graph, each vertex contains a set of image pixels, and each edge connects those vertices relying on same predefined adjacency relation or connectivity between nodes.

### 2.1.2 Tree

A tree is built according to the graph theory as an acyclic, directed graph from its unique root to the leaves [13]. More precisely, a tree is the special form of graph having only one path between any two vertices and there is no loop (a.k.a cycle) among the nodes. If two nodes are connected, one of them is called the parent node (i.e. the closest to the root) and the other one is its child node according to the direction of the edge. The only node which has no parent in a tree is called the root of the tree, while the nodes that have no children are called leaf nodes. A tree will be denoted in our manuscript by the symbol $\mathcal{T}$. Figure 2.2 illustrates a simple tree example.

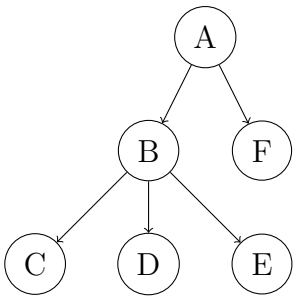



Figure 2.2 – A tree example having 4 leaves $\{C, D, E, F\}$. The root node $A$ has 2 children $\{B, F\}$ and the node $B$ has 3 children $\{C, D, E\}$.

## 2.2 Morphological hierarchies

Morphological hierarchies form one of the most significant development of Mathematical Morphology in the last decades. Mathematical Morphology (MM) is a well-established framework introduced in the 1960s [148] for analyzing the spatial domain of images given a predefined template (a.k.a. structuring element). These morphological hierarchies can be represented with a tree structure which is an efficient way to compute morphological (connected) operators. These trees are

able to combine different types of information in their nodes such as size, spectral signature, shape, *etc*. The structure of a tree depends on the underlying model used to construct the hierarchy. Morphological hierarchies were divided into hierarchies of partitions and partial partitions in [136]. Instead of these terms, in this thesis, we will use the categorisation from [21] in partitioning trees and inclusion trees.

Among the numerous models in both partitioning and inclusion trees, several tree based image models have been developed and applied until now such as max-tree [145], [29], Tree of Shapes (ToS) [106],[33], [55], Binary Partition Tree (BPT) [143], $\alpha$-tree [154], [100] and constrained connectivity hierarchies, such as $\omega$-trees [18]. Now, mathematical details will be given about these tree types.

### 2.2.1 Inclusion trees

In inclusion trees, the leaf nodes hold small regions or points, which are local maxima or minima. Nodes are merged and/or expanded by a region growing process which starts from the leaves and adds one or more pixels to the existing regions. The new node becomes a parent of the nodes by this operation and this continues until the root node covering the whole image domain. Inclusion trees are divided into two categories, min-max trees and ToS. For both tree definitions, neighbouring pixels are grouped together in connected components according to their location and intensities. Let $I$ be a gray scale image on domain $\Omega \subset \mathbb{N}^2$ taking its values in a finite set $\mathcal{V} \subset \mathbb{Z}$ with an order relation $\leq$, these values being between 0 and $2^{bit} - 1$ (255 for 8 bit images). $I$ can be seen as a function:

$$I : \Omega \rightarrow \mathcal{V} \tag{2.2}$$

For any $x \in \Omega$, $I(x)$ takes its values $v$ in a scalar space $\mathcal{V}$. For the sake of conciseness, we will write $x = (i, j)$ the coordinates of an image pixel.

A connected component ($C$) is a group of pixels that are connected according to their locations. The most well-known connectivity rules are 4- and 8-connectivity, where a pixel is considered as adjacent to four or eight of its neighbouring pixels, respectively. Those connected components are identified with levels ($\lambda$) at which they appear in inclusion trees. These levels can be seen as thresholds on inten-

sity. We consider the set of nodes $V$ where each node is one of these connected components:

$$V(I) = \bigcup_{\forall k,\lambda} C_\lambda^k(I) \tag{2.3}$$

where $C$ denotes a connected component, $k$ its index in the level $\lambda$ of the image $I$. For the sake of concision, we will omit $I$ from $C$ and $V$ notations in the rest of this manuscript. Any two nodes are either disjoint or nested. Nodes at the same level have no intersection:

$$\forall k \neq k', C_\lambda^k \cap C_\lambda^{k'} = \emptyset \tag{2.4}$$

We will omit $k$ when only one node is involved in some expression. An edge $E$ exists between two nodes $C_\lambda$ and $C_{\lambda'}$ if we have $C_{\lambda'} \subset C_\lambda$ and there is no other node in between, i.e. $\nexists\, C_{\lambda''} : C_{\lambda'} \subset C_{\lambda''} \subset C_\lambda$. The difference between the parent $C_\lambda$ and its child $C_{\lambda'}$ is the set of additional pixels which appear and become connected to the parent node. Then, following equation can be used when there is only one node at $\lambda$ level:

$$C_\lambda \setminus C_{\lambda'} = \{x \in \Omega \mid I(x) = \lambda\} \tag{2.5}$$

The additional pixels with value $I(x) = \lambda$ do not belong to any child node of $C_\lambda$. These pixels only appear in inclusion trees and are grouped in a single node called ghost node which is discussed in details in [20]. More precisely, each node except leaves may have an additional child node which only contains the pixels defined by equation 2.5. The goal is to avoid information loss and to consider these level set-based trees as possible results of a hierarchical clustering. We will denote these ghost nodes as $C'$. Let $C'_\lambda$ be a ghost node and $C_{\lambda'}$ be the only child of its parent. Then, the parent node of $C_{\lambda'}$ can be defined as:

$$C_\lambda = C_{\lambda'} \cup C'_\lambda \tag{2.6}$$

Ghost nodes can be considered as leaves because they do not have any child by definition. However, they will be discarded except when stated specifically.

While min, max and ToS have several common properties, they have also particular properties which will be explained in the following sections.

### Min and Max Trees

Component trees were introduced by Jones [79] as an efficient hierarchical image representation structure with respect to thresholding and inclusion relationship. If we assign a threshold $\lambda \in \mathbb{Z}$, the lower and upper threshold sets are defined as:

$$[I \leq \lambda] = \{x \in \Omega \mid I(x) \leq \lambda\} \tag{2.7}$$

$$[I \geq \lambda] = \{x \in \Omega \mid I(x) \geq \lambda\} \tag{2.8}$$

Moreover, we note that lower and upper level sets define the hierarchy of the min-tree and max-tree respectively:

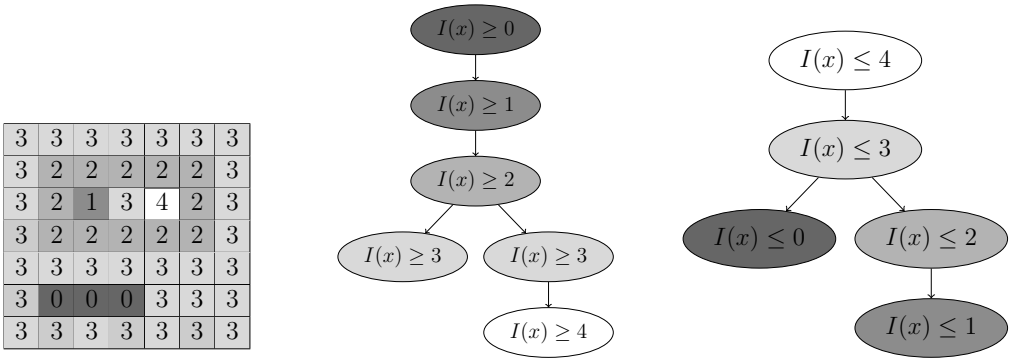
$$L^{\lambda}(I) = \{x \in \Omega \mid C(I(x)) \leq \lambda\} \tag{2.9}$$

$$L_{\lambda}(I) = \{x \in \Omega \mid C(I(x)) \geq \lambda\} \tag{2.10}$$

where $L$ represents the level set for $\lambda$ level.

The root node of a max-tree represents the lowest gray level of the image; whereas, the root node of min tree represents the highest gray level of the image. Thus, a path from root to leaves of the min and max-tree goes to progressively brighter and progressively darker objects respectively. Therefore, the min-tree is suitable for manipulating the dark image objects and max-tree allows for easy manipulation of bright objects.

Figure 2.3 shows a sample image matrix, its max-tree and min-tree respectively. Tree nodes are colored according to the level value that they include. In the max-tree, there are two nodes at level 3 because there is an isolated pixel with value 3 which is not a neighbour of any other group of pixels with intensity equal to 3. Besides, the node at level 4 is a child of this isolated pixel because its pixels are neighbours. Similarly, the node at level 2 and node the at level 0 are separated because relevant pixels are not connected to each other in the min-tree.




(a) Sample image (b) Max-tree (c) Min-tree

Figure 2.3 – Max and min-tree representation of a sample image

### Tree of shapes

The Tree of Shapes (ToS) is a topographic map of an image. Since it makes no assumption about the contrast of objects, it is also known as self-dual representation (i.e. the trees of an image and its dual w.r.t. complementation are similar). The aim of ToS is to build a node by using borders of the shapes in the image. This type of structure was first introduced by [106] and named as ToS in [55].

Let $X \subset \Omega$ be a given set of pixels and $\partial X$ be the border pixels of this set. The hole filling operator is defined by:

$$H(X) = \Omega \setminus C(X, \partial X) \tag{2.11}$$

In equation 2.11, $C(X, \partial X)$ is the union of connected component of $X$ and its border $\partial X$. The hole filling operator creates a shape $S$ after combining min-tree and max-tree with this operator:

$$S = H(L^{\lambda}(I)) \cup H(L_{\lambda}(I)) \tag{2.12}$$

Any two shapes of this tree representation are either disjoint or nested and ToS is created by ordering these shapes with the inclusion operator. Figure 2.4 shows a ToS example. The main difference of ToS w.r.t. min and max-trees is that leaves will contain minimum or maximum pixel values of the image. Therefore, ToS is a self-dual tree.

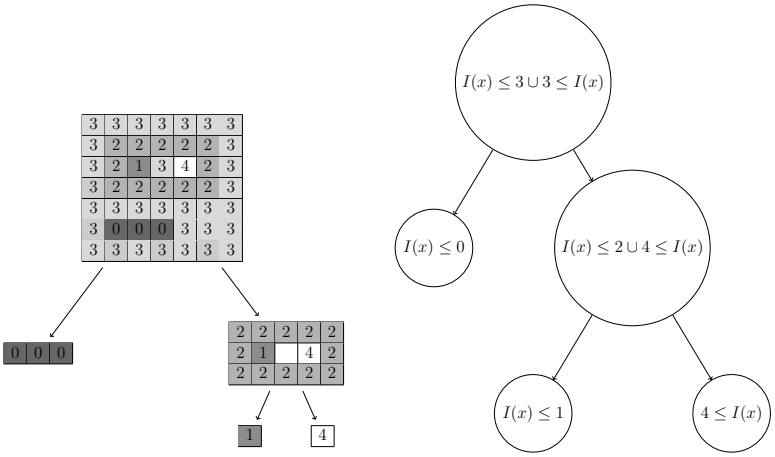



Figure 2.4 – Tree of shapes example

### 2.2.2 Partitioning trees

In contrast with inclusion trees, any leaf or any cut of a partitioning tree forms an image partition. New regions of the inner nodes are formed as unions of two or more existing adjacent regions. Partitioning trees have no ghost nodes and level of the nodes represent dissimilarities rather than pixel intensities. Beyond that, equations 2.3 and 2.4 are valid for partitioning trees as well. Among the possible partitioning trees, we will present $\alpha$-tree and BPT in this section because of their popularity in the literature.

#### $\alpha$-tree

The first partitioning tree is the $\alpha$-tree which relies on quasi-flat zones. Neighbouring pixels are connected if their gray level difference is less than or equal to $\alpha$. The $\alpha$-tree was used in [154] for the first time. The hierarchy relation can be denoted as:

$$C_\alpha \subseteq C_{\alpha'} \quad \forall \alpha \leq \alpha' \tag{2.13}$$

Figure 2.5 shows an $\alpha$-tree example for a sample image. In the right of the tree, we show the $\alpha$ level index line for each level. In the leaves of this kind of tree, pixels are alone if they do not share a similar value with any of their neighbours. However, the union of all the leaf nodes or any union of the nodes at the same level is equal to the root node of the tree.

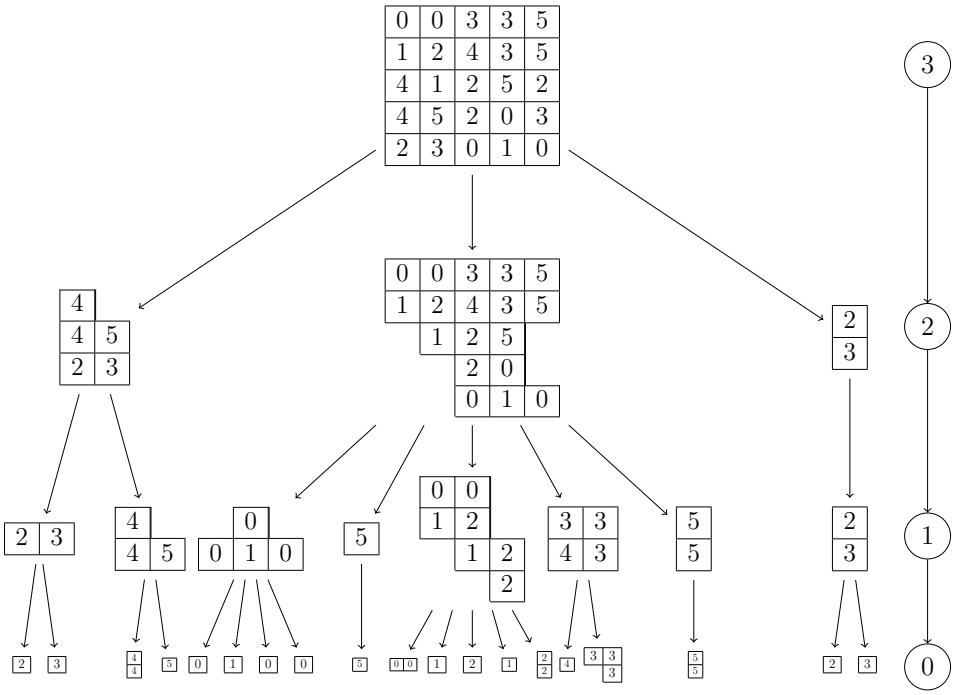



Figure 2.5 – $\alpha$-tree example

### Binary Partition Tree

Binary partition tree (BPT) is another partitioning tree and the hierarchy depends on the choice of the region model and the merging criterion [143]. Construction of this tree starts from the leaves and neighbour/adjacent regions are merged until a single region is obtained. The leaves represent the largest components with constant value and the other nodes represent the components obtained by merging the regions represented by their children. The merging order depends on the similarity measure used to compare adjacent regions.

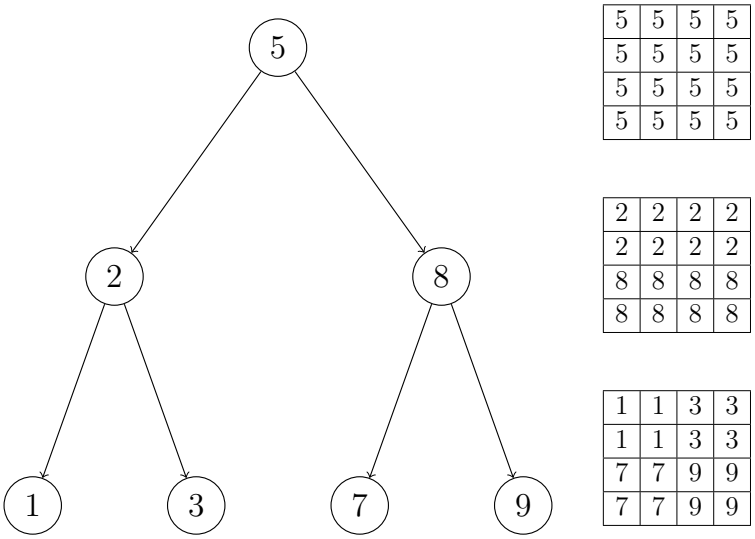



Figure 2.6 – A binary partition tree example

Let $R_1 \in \Omega$ and $R_2 \in \Omega$ be two distinct but adjacent regions belonging to different nodes. To combine these nodes in the same parent node, there are two steps to follow: the region model and the merging criterion. The region model $M_{R_i}$ can be simply characterized by the average intensity of a region. The merging criterion is used to find the most similar regions. One of the common merging criterion rule is the weighted squared distance which can be defined as:

$$O(R_1, R_2) = N_1 \, ||M_{R_1} - M_{R_1 \cup R_2}||_2 + N_2 \, ||M_{R_2} - M_{R_1 \cup R_2}||_2 \tag{2.14}$$

where $||.||_2$ denotes $\mathcal{L}_2$ norm, $N_1$ and $N_2$ are the number of pixels in the region $R_1$ and $R_2$ respectively. Elements of the merged region are calculated as:

$$\begin{aligned} &\text{If} \quad N_1 < N_2 \Rightarrow M_{R_1 \cup R_2} = M_{R_2} \\ &\text{If} \quad N_1 > N_2 \Rightarrow M_{R_1 \cup R_2} = M_{R_1} \\ &\text{If} \quad N_1 = N_2 \Rightarrow M_{R_1 \cup R_2} = (M_{R_1} + M_{R_2})/2 \end{aligned} \tag{2.15}$$

Figure 2.6 shows a BPT example. In the right of the figure, there are 3 images for each level cut. Pixel values are assigned according to their location in the tree. The input image is equal to the cut at leaves and it consists of 1, 3, 7 and 9 values. Each sibling pair has the same number of pixels. Therefore, they merge according to their average intensity values in their ancestor nodes.

## 2.3 Attributes of tree nodes

Nodes of a tree may include different kind of information. Therefore, after building a tree, the next step is the calculation of attributes for each node. Depending on the application, different kind of attributes can be calculated with the pixels in the nodes. Many attributes have been defined in the literature and the actual choice depends on the application.

These attributes can be related to the content, the shape, or the position of the nodes in the tree. We will use them in the experiments with SITS in chapter 7. We will denote an attribute as $\mathcal{A}_{attribute}$ with relevant subscript.

### 2.3.1 Content-based attributes

**Area:** is measured by the number of pixels of the node. This attribute is sometimes called "volume" in the context of multi-dimensional images such as those found in the biomedical domain.

$$\mathcal{A}_{area}(C_\lambda^k) = \mid x \mid, \forall x \in C_\lambda^k \tag{2.16}$$

**Raw Volume:** is the sum of the pixel intensities in the node.

$$\mathcal{A}_{rvol}(C_\lambda^k) = \sum_{x \in C_\lambda^k} I(x) \tag{2.17}$$

**Volume:** is the sum of differences between the pixel intensities in the node and the level of the node [112]. Volume is an important attribute used with 3D images.

$$\mathcal{A}_{vol}(C_\lambda^k) = \sum_{x \in C_\lambda^k} \mid I(x) - \lambda \mid \tag{2.18}$$

**Mean Gray Value:** provides the gray-level average of the tree nodes.

$$\mathcal{A}_{mean}(C_\lambda^k) = \frac{\mathcal{A}_{rvol}(C_\lambda^k)}{\mathcal{A}_{area}(C_\lambda^k)} \tag{2.19}$$

**Variance:** measures how far the values in the tree nodes are spread out from the average value in the node.

$$\mathcal{A}_{var}(C_\lambda^k) = \frac{\sum_{x \in C_\lambda^k} (I(x) - \mathcal{A}_{mean}(C_\lambda^k))^2}{\mathcal{A}_{area}(C_\lambda^k)} \tag{2.20}$$

**Standard deviation:** is a measure that is used to quantify the amount of variation or dispersion of values in the tree nodes.

$$\mathcal{A}_{\sigma}(C_\lambda^k) = \sqrt{\mathcal{A}_{var}(C_\lambda^k)} \tag{2.21}$$

**Histogram:** measures probability distribution of gray levels in the node.

$$\mathcal{A}_{hist}(C_\lambda^k) = hist(I(x), \forall x \in C_\lambda^k) \tag{2.22}$$

### 2.3.2 Location-based attributes

**Level:** represents the place of the node in the tree.

$$\mathcal{A}_{level}(C_\lambda^k) = \lambda \tag{2.23}$$

**Depth:** is the difference between the level of the root node and the level of the parent of the node (except for ToS).

$$\mathcal{A}_{depth}(C_\lambda^k) = |\lambda_r - \lambda| \tag{2.24}$$

In this equation, $\lambda_r$ represents the level of the root node. For ToS, the depth attribute is equal to the path length between the node and the root of the tree [30].

**Height:** is the difference between the maximum and the minimum of the node[40]. (also called range)

$$\mathcal{A}_{height}(C_\lambda^k) = \max_{x \in C_\lambda^k}(I(x)) - \min_{x \in C_\lambda^k}(I(x)) \tag{2.25}$$

In this equation, max and min notations represent maximum and minimum values of the node.

### 2.3.3 Shape-based attributes

**Moment of Inertia:** is a measure of compactness and is equal to the first moment of the shape in the node [122]. It is based on pixel coordinates and authors in [182] have extended it for 3D components.

$$\mathcal{A}_{moi}(C_\lambda^k) = \frac{\mathcal{A}_{area}}{4} + \sum_{x \in C_\lambda^k} (x - \bar{x})^2 \tag{2.26}$$

In this equation, $\bar{x}$ represents the location of the node centroid which corresponds to the average position of the pixels in the node.

**Elongation:** is a ratio between the moment of inertia and the square of the area [171].

$$\mathcal{A}_{elong}(C_\lambda^k) = \frac{\mathcal{A}_{moi}(C_\lambda^k)}{\mathcal{A}_{area}^2(C_\lambda^k)} \tag{2.27}$$

**Perimeter:** is the number of pixels on the border of the node [122]:

$$\mathcal{A}_{per}(C_\lambda^k) = \text{card}\left\{x \mid x \in C_\lambda^k,\ \exists x' \in \mathcal{N}(x) \text{ s.t. } x' \notin C_\lambda^k\right\} \tag{2.28}$$

where $\mathcal{N}(x)$ denotes the neighbourhood of $x$.

**Circularity:** is the ratio of the area of the node and the square of its perimeter. It is also called compactness in literature.

$$\mathcal{A}_{circ}(C_\lambda^k) = \frac{\mathcal{A}_{area}(C_\lambda^k)}{\mathcal{A}_{per}^2(C_\lambda^k)} \tag{2.29}$$

**Bounding Box** : is the smallest rectangle parallel to the axis which contains all the pixels of the node. Area of the bounding box coordinates can be found as:

$$\mathcal{A}_{bb}(C_\lambda^k) = \left(\max_{(i,j)\in C_\lambda^k}(i) - \min_{(i,j)\in C_\lambda^k}(i)\right) \times \left(\max_{(i,j)\in C_\lambda^k}(j) - \min_{(i,j)\in C_\lambda^k}(j)\right) \tag{2.30}$$

**Rectangularity:** is the ratio of the area of the node and of its bounding box.

$$\mathcal{A}_{rect}(C_\lambda^k) = \frac{\mathcal{A}_{area}(C_\lambda^k)}{\mathcal{A}_{bb}(C_\lambda^k)} \tag{2.31}$$

**Non-compactness:** relies on moment of inertia and volume attributes [180].

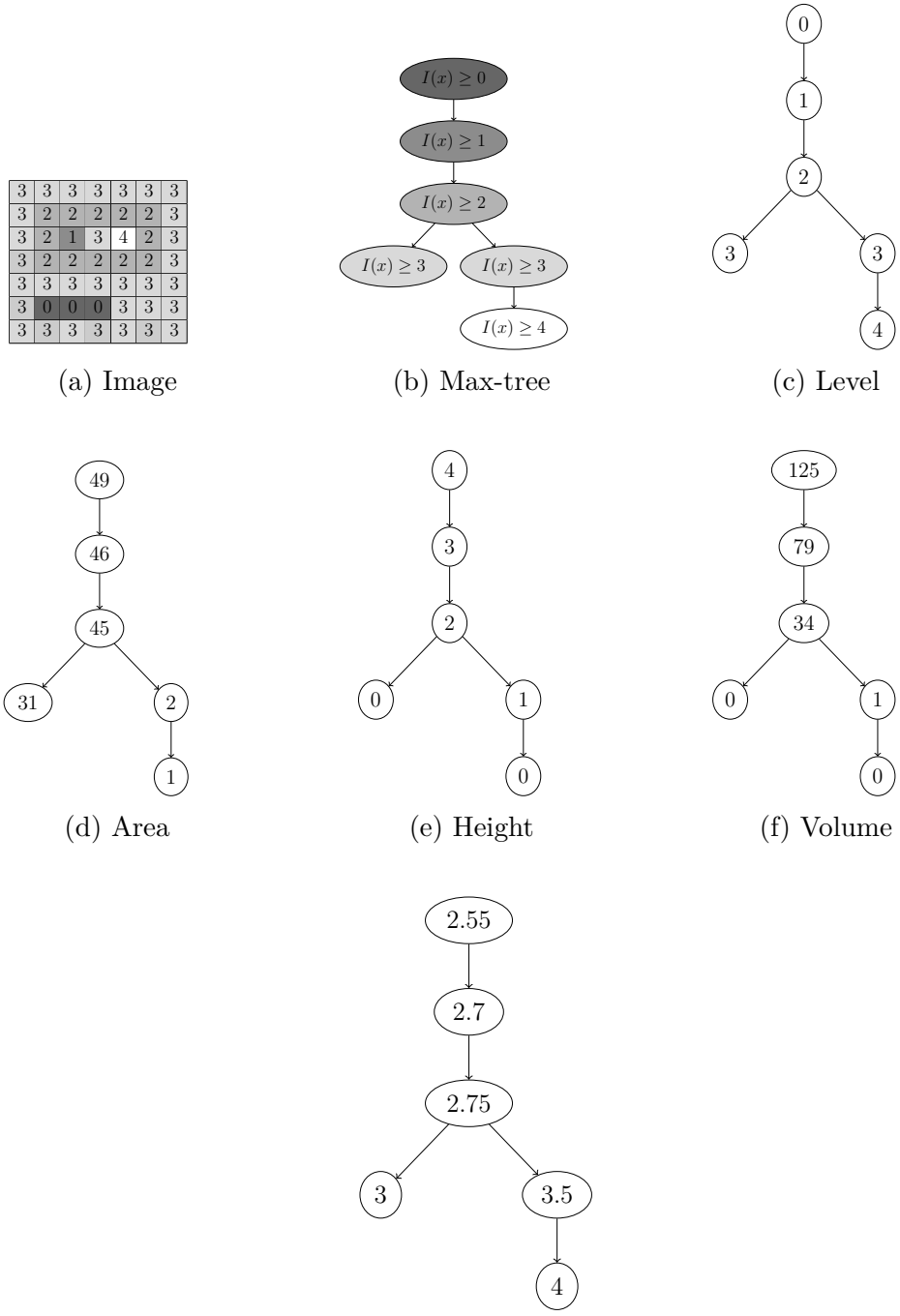
$$\mathcal{A}_{comp}(C_\lambda^k) = \frac{\mathcal{A}_{moi}(C_\lambda^k)}{\mathcal{A}_{vol}(C_\lambda^k)^{5/3}} \tag{2.32}$$

**Eccentricity:** is calculated from the ratio of the largest distance between two points of the the node and the node perimeter [144]. It behaves like the eccentricity parameter of an ellipse.

$$\mathcal{A}_{ecc}(C_\lambda^k) = \frac{\max\{||x-y||_2 \mid x,y \in C_\lambda^k\}}{\mathcal{A}_{per}(C_\lambda^k)} \tag{2.33}$$

**Sphericity:** can be calculated as [85]:

$$\mathcal{A}_{sph}(C_\lambda^k) = \frac{\pi^{\frac{1}{3}}(6\mathcal{A}_{vol}(C_\lambda^k))^{\frac{2}{3}}}{\mathcal{A}_{area}(C_\lambda^k)} \tag{2.34}$$



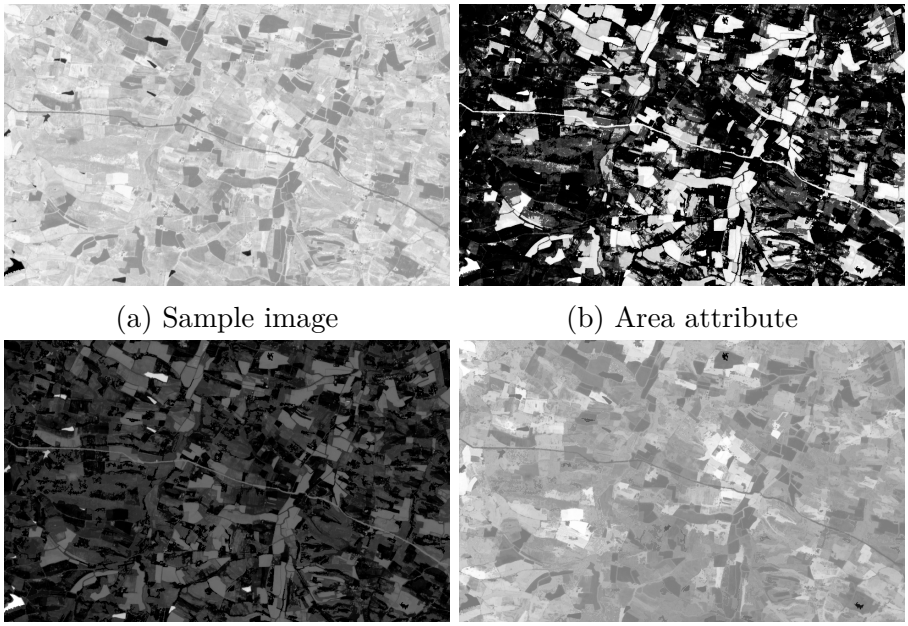

(a) Image (b) Max-tree (c) Level

(d) Area (e) Height (f) Volume

(g) Mean gray value (approximate values)

Figure 2.7 – Attributes extracted from the max-tree built in Figure 2.3

Figure 2.7 illustrates some attributes for the same example image with the max-tree. Figure 2.8 illustrates sample image and some feature images produced using corresponding attributes. First, attributes of the tree nodes are calculated. Then, these attribute values are assigned to each pixel location according to the first appearance of these pixels in the tree.



(a) Sample image (b) Area attribute

(c) Height attribute (d) Mean gray value attribute

Figure 2.8 – An original image and images of tree attributes

### 2.3.4 Some properties of nodes attributes

Let $\mathcal{A}(C)$ be an attribute value and $C_c$ a child node of $C$. If $\mathcal{A}(C)$ is always greater than $\mathcal{A}(C_c)$ for all nodes of the tree and is always increasing through one path from leaves to root, this attribute is called an increasing attribute. Otherwise, the attribute is non-increasing. Let us observe that, shape-based attributes are generally non-increasing [56].

All these attributes can also be used to calculate parent-child relationship according to the given attribute. For instance, area attribute difference between parent node and child node gives the growth rate of the connected component from the child to the parent node.

## 2.4 Tree processing

Trees give the opportunity to analyze images in a hierarchical way by the information contained in their nodes. In this section, pixel-wise and image-tile feature extraction methods with trees will be presented.

### 2.4.1 Pixel-wise feature extraction

#### Filtering

Tree filtering is a common operation used either as a final step for a method such as image simplification [186] or as an intermediate step for some applications such as feature extraction [188] or object detection [177]. The main advantage of tree filtering is its ability to produce a non-blurred image where edges of the connected components are preserved (a.k.a. connected filtering). Applying tree filtering on an image involves three main steps: (1) building the corresponding tree, (2) defining and applying a filtering strategy on it, and (3) reconstructing the image. The filtering operations can be thus expressed as follows:

$$I \mapsto \mathcal{T} \tag{2.35}$$

$$\mathcal{T} \mapsto \mathcal{T}' = \zeta^h(\mathcal{T}) \tag{2.36}$$

$$\mathcal{T}' \mapsto I' \tag{2.37}$$

where $I'$ is the filtered image obtained after filtering the tree with the $\zeta$ filtering operator and a threshold $h$. Figure 2.9 demonstrates the filtering workflow with a real image.

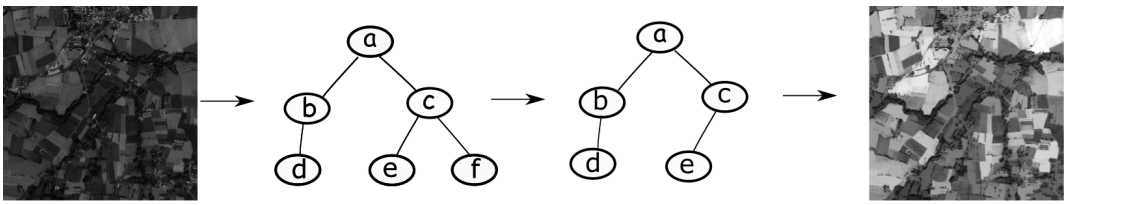



Figure 2.9 – Filtering workflow

A survey of filtering tools for hierarchical image representations was published in [146]. For inclusion trees, filtering removes small contrasted structures in the image without changing the larger structures. Filtering for partitioning trees was first proposed in [4]. Such a filtering corresponds to selecting a cut in the tree based on some criterion and assigning a model to each node in the selected cut to be used in image reconstruction. Since partitioning trees have no local extrema, cuts can represent regions of intermediate gray levels. Two general approaches, called pruning or non pruning, are used for filtering. Let $C_\lambda^k$ denotes a node of the tree with an attribute $\mathcal{A}(C_\lambda^k)$ and $h$ is a threshold considered to filter the tree. Then, the criterion for tree filtering is $\mathcal{A}(C_\lambda^k) < h$ for both approaches. It should be noted that if the attribute is increasing, both approaches give the same result. Values of the pixels in the removed nodes are replaced according to the level of their new node.

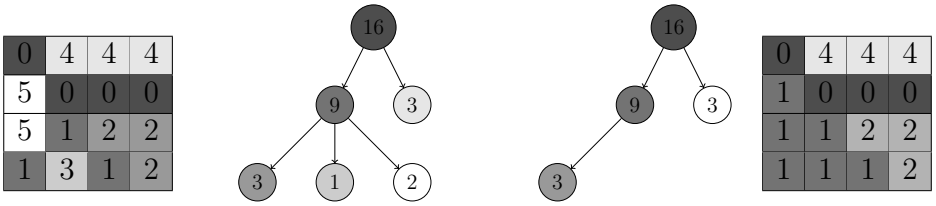
Pruning strategies aim at removing a set of branches from the tree. A single cut is made along each path from leaf to root and all nodes are collapsed onto the highest surviving ancestor. They are simple to apply when the attribute is increasing since all nodes on which the criterion is not verified are belonging to the same branches. Examples of pruning strategies are the Min and Max rules [172].

For the Min rule, if the criterion is satisfied, the node is removed with all its descendants. For the Max rule, a node that does not satisfy the criterion is suppressed, only if the criterion is not satisfied by all of its descendants. Figure 2.10 illustrates an area filtering example for a sample image. First, the max-tree is built from the image (Figure 2.10a). Figure 2.10b shows the filtered tree with $h = 3$ and the reconstructed image.

Selection of thresholds is a challenging part in the filtering operation. For instance, authors in [120], proposed a formula to select area thresholds:

$$A = \frac{1000}{s}\{1, 2, .., k\} \tag{2.38}$$

where $s$ represents the spatial resolution of the image in meters, and $k$ is the number of thresholds to be used. Their target is to determine an initial threshold with reasonable size.




(a) Original image and its max-tree with the area attributes

(b) Filtered tree ($h < 3$ and filtered image, changed values in bold)

Figure 2.10 – Example of filtering with area attribute.

Non pruning strategies provide a different solution when pruning strategies are not sufficient to obtain the required simplification. Many approaches have been proposed in the literature, such as the Viterbi algorithm [145], optimization methods [187], the direct rule [172] or the substractive rule [171].

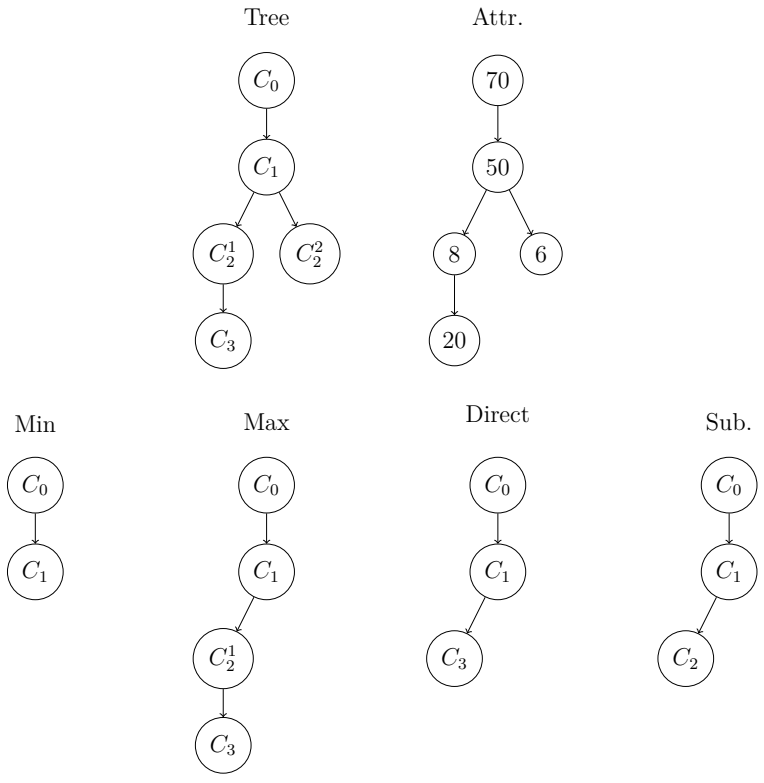



Figure 2.11 – Example of filtering rules using 10 value as a threshold.

In the direct and subtractive rules, the nodes are removed but their descendants remain unaffected. They differ on the way they replace the pixels from the removed node. The direct rule only replaces pixels in the removed nodes but the subtractive rule also replaces pixels in the descendants of the removed nodes. Figure 2.11 displays different filtering types for the same tree according to a non-increasing attribute. Threshold is fixed to "10" for each filtering.

Image segmentation can be achieved by filtering the partitioning trees because these trees include the whole image at each level as mentioned in section 2.2.2. Two important examples are segmentation with BPT [143] or with $\alpha$-tree [103].

As a basic approach, if we aim to select a specific number of regions, we can cut the tree from the level which contains the number of nodes equal to the desired amount. Besides, a pruning can be applied to trees relying on the attribute of the nodes. Filtering partitioning trees are used for attribute profile [18], image segmentation [142] or spectral unmixing [174]. In [111], authors proposed vector attribute filters (volume, height) to segment 3D biomedical images. In [84], optimal partitions from hierarchical representations are extracted by energy minimization on hierarchies. In [18], authors propose area filtering for $\alpha$-tree with min, max, average and level rules.

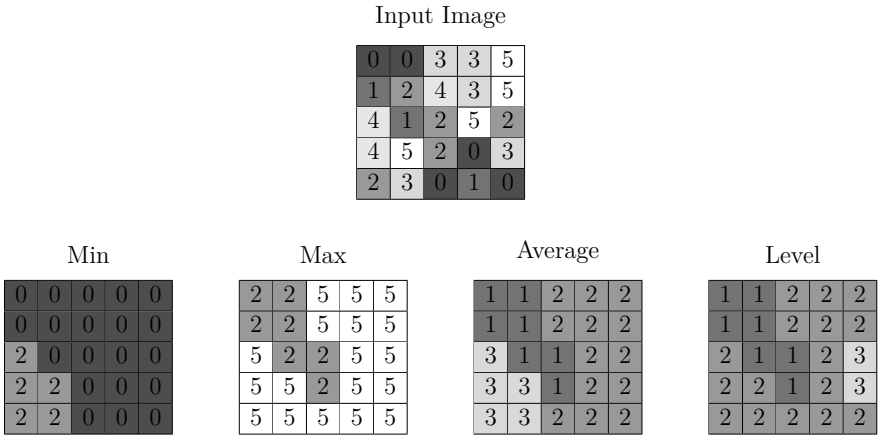



Figure 2.12 – Attribute filtering with the $\alpha$-tree creates a segmented image

Figure 2.12 illustrates an area filtering example using the $\alpha$-tree of figure 2.5. In this example, the threshold is equal to 5. Min and max filtering types convert pixel values of a node to their minimum or maximum. In average filtering, $\mathcal{A}_{mean}$ is calculated. Level type filtering assigns $\mathcal{A}_{level}$ to the regions.

Although using inclusion trees for segmentation application is not very common in literature, authors proposed interactive segmentation with component tree in [119], 3D medical image segmentation is implemented in [47] and directed connected operators are used in [124].

### Morphological Attribute Profiles

Morphological Profile (MP) has been introduced in [126]. A MP is produced by using thickening and thinning morphological operators. Many improvements have been done during last decade to extend MP to morphological hierarchies such as Morphological Attribute Profiles (AP) [43]. An AP is obtained by filtering the image $I$ using a predicate with increasing threshold values:

$$AP(I) = (I^{\phi^{h_m}}, I^{\phi^{h_{m-1}}}, ..., I, ..., I^{\gamma^{h_{m-1}}}, I^{\gamma^{h_m}}) \tag{2.39}$$

where $\phi$ and $\gamma$ are respectively thickening and thinning operators based on $m$ ordered $h$ thresholds. Since AP is applied with trees, these operators are applied by filtering the tree. Any attribute can be used to filter the tree. AP provide feature extraction from the images and give useful features especially for classification [33]. The attribute profile goes beyond a simple filtering operation and stacks images produced by filtering with different thresholds. One of the extension of the AP is the Differential Attribute Profiles (DAP) which provides residual information between the levels:

$$DAP(I) = \triangle(I^{\phi^h}) \cup \triangle(I^{\gamma^h}) \tag{2.40}$$

where $\triangle$ is the differential operator which calculates difference between the successive images.

Instead of building min and max-tree separately, one can use ToS, leading to the so-called Self Dual Attribute Profile (SDAP) [42]:

$$SDAP(I) = \{I, I^{\phi^{h_1}}, ..., I^{\phi^{h_{m-1}}}, I^{\phi^{h_m}}\} \tag{2.41}$$

Previous AP methods are used with panchromatic single frame images. In order to deal with multispectral or hyperspectral images, authors in [41] proposed the extended attribute profiles (EAP):

$$EAP(I) = \{AP(I_1), AP(I_2), ..., AP(I_b)\} \tag{2.42}$$

where $b$ represents amount of channels.

Instead of pixel values, using attribute values as a feature for each pixel was recently proposed in [130], leading to the so-called feature profiles (FP):

$$FP(I) = \{f[I^{\phi^{h_m}}], f[I^{\phi^{h_{m-1}}}], ..., I, ..., f[I^{\gamma^{h_{m-1}}}], f[I^{\gamma^{h_m}}]\} \tag{2.43}$$

In Equation 2.43, $f$ represents a feature or attribute value. Thus, objects are characterized with their attributes directly, instead of intensity values. In [104], authors provide a comparative study between FP and AP.

In [45], the histogram based attribute profile (HAP) was proposed in order to overcome complex spatial distributions. The HAP algorithm starts with creating standard APs, then for each filtered image, an histogram over a local patch is computed. Lastly, all local histograms obtained at the same pixel position are stacked:

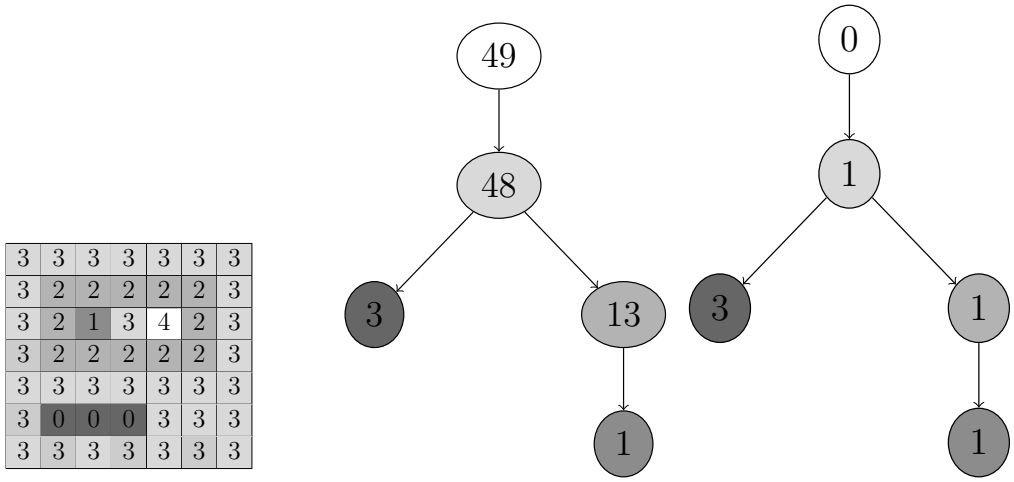
$$HAP(I(x)) = \{hist(I(x)^{\phi^{h_m}}), ..., hist(I(x)^{\gamma^{h_m}})\} \tag{2.44}$$

It has been shown recently that histograms can be efficiently replaced by only a few local statistics, leading to the Local Feature-Based Attribute Profiles (LFAP) [132].

Alternatively, Extinction Profile (EP) has been introduced in [57]. Extinction values are the extrema of some selected attributes in the leaves of the tree. Extinction filtering is removing branches which have extinction values in their leaves.

### 2.4.2 Image-tile feature extraction

In order to extract image-tile features, we will focus on Pattern Spectra (PS) which is a multiscale image descriptor which was initially introduced for describing the size distribution of objects within images through histogram-like representations [94]. Traditionally, it was produced by calculating the differences of consecutive morphological openings applied on the image. In the previous decade, tree representations have been used to produce the PS [170]. In order to increase efficiency of PS, two dimensional PS is used with max-tree for pattern-based analysis in [171]. Pattern spectra can be calculated for the whole image or for some small parts of it, leading to Local Pattern Spectra [17].




(a) Image (b) Area attributes (c) Level differences

| 9+1 | 13 | 0 | 0 | 48 |
|---|---|---|---|---|

(d) 1D Pattern spectra

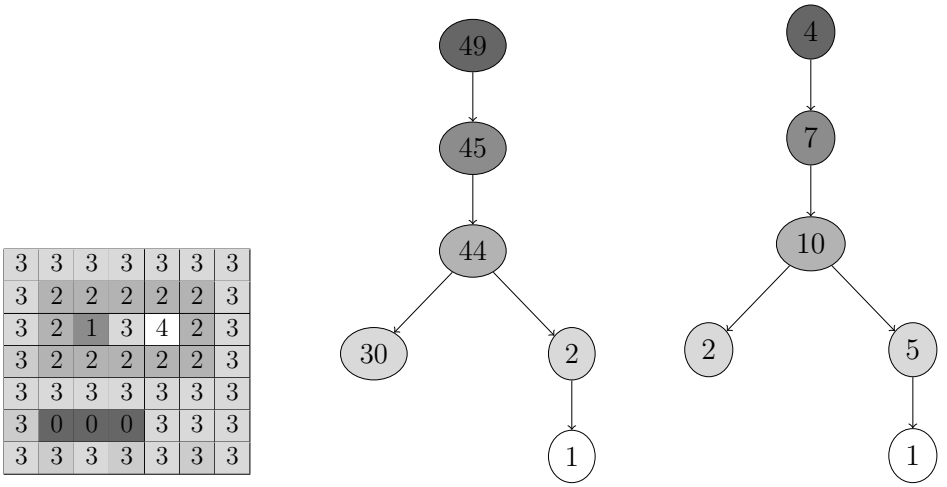
Figure 2.13 – Pattern spectra with only area attribute of the min-tree

To produce pattern spectra, first, an attribute is selected and calculated. Then, area weighted parent level difference is calculated for each node of the tree separately. To create a histogram-like distribution, a number of bins is chosen. The size of the bin depends on the distribution of the selected attribute. Finally, area weighted parent level difference values are assigned to the bins according to the interval of the bins. If the two weighted area values belong to the same bin, these

values are summed in this bin. Figure 2.13 illustrates a 1-dimensional pattern spectra example with the area attribute. The sample image is provided in Figure 2.13a. Attributes are calculated with the min-tree of the image in Figure 2.13b and level difference for each node is in figure 2.13c. Figure 2.13d shows a 1-dimensional pattern spectra with 5 bins. Since the maximum area is 49, the bin interval is 10 in this example. It should be noted that bins are created linearly but could be created with a logarithmic scale.

Figure 2.14 represents a 2D pattern spectra with the max-tree of the same image in Figure 2.13. It differs from a 1D pattern spectra by adding another attribute (the origin is set at the top-left position). The choice is made according to the considered application. Area weighted parent level difference values are assigned to the bins according to the attributes distributions. We note that Figure 2.14 uses a non-increasing attribute values.

In [171], authors compute area and elongation attribute for each node separately in order to create a 2D pattern spectra. They use size and shape attributes for each dimension and they call this 2D pattern spectra a size-shape pattern spectra.




(a) Image (b) Area attributes (c) An attribute

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 0 | 0 | 30 | 0 | 49 |
| 2 | 0 | 0 | 0 | 45 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 44 |

(d) 2D pattern spectra

Figure 2.14 – 2D Pattern spectra with the max-tree

Another important image descriptor with trees is Maximally Stable Extremal Regions (MSER). MSER relies on the growing rate of the ancestor-descendant nodes [95].

$$\mathcal{A}_{mser} = \frac{\mathcal{A}_{area}(C_\lambda) - \mathcal{A}_{area}(C_{\lambda+\triangle})}{\mathcal{A}_{area}(C_\lambda)} \tag{2.45}$$

Here, the nodes $C_\lambda$ and $C_{\lambda+\triangle}$ could be parent and child or there can be other nodes between them depending on the selected $\triangle$ value. Distance between these nodes (length of the path connecting them) should be selected before calculating MSER. Nodes with higher $\mathcal{A}_{mser}$ value includes more stable objects. Recently, MSER has been extended to ToS in [19]. As an object-based temporal analysis work, we can mention the use of MSER for video sequences in [46]. They build tree for each video frame and find MSER regions separately.

# Chapter 3

# Challenges and State-of-the-art of SITS analysis

## Contents



This chapter presents the challenges in SITS analysis and reviews the related works. Our goal here is to explain the necessity of extending morphological hierarchies to SITS and to identify in the literature the missing parts necessary for this extension.

## 3.1 Challenges of Satellite Image Time Series

Compared to still images, SITS have specific characteristics which make their processing more difficult. The most prominent one is their temporal dimension which leads to important challenges that must be tackled. In this section, we list the different effects of this temporal dimension which must be dealt with by the analysis tools.

### 3.1.1 Irregular sampling

Depending on the revisit period of the satellite, its acquisition plan, or the viewing conditions, time series may contain samples unevenly distributed in time. Moreover, applications which require data over large regions on the Earth may use a mosaick of images which can be built using different acquisition times over several parts of these regions. This issue is called irregular sampling in the literature [98], [8]. One of the drawbacks of irregular sampling is the fluctuation of the raw information: intensity at some location may change abruptly, and thus decreasing the temporal consistency for this location.

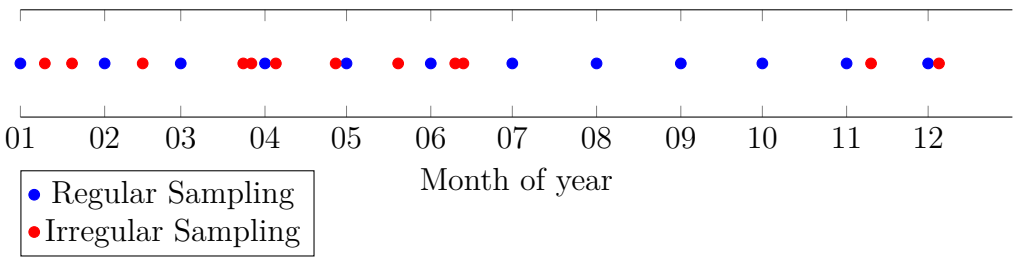



Figure 3.1 – Sampling examples according to acquisition times

Figure 3.1 gives an example for acquisition times of two different datasets during one year. Each coloured data point represents an acquisition time. Blue ones represent an ideal acquisition whereas red ones correspond to sampling as it occurs in usual SITS. For instance, there is a large gap between June and November for the irregularly sampled dataset.

In the literature, there are two common strategies in order to overcome this irregular sampling issue. Performing temporal interpolation in order to create regularly sampled time series using for instance moving average window [48] is a way to deal with irregular sampling. Using a distance metric such as Dynamic Time Warping (DTW) to calculate distance between pixel time series is another method [129]. DTW can exploit the temporal distortions by finding optimal alignment between two time series. Thus, DTW is not affected by irregularity of samples.

### 3.1.2 Instantaneous effects

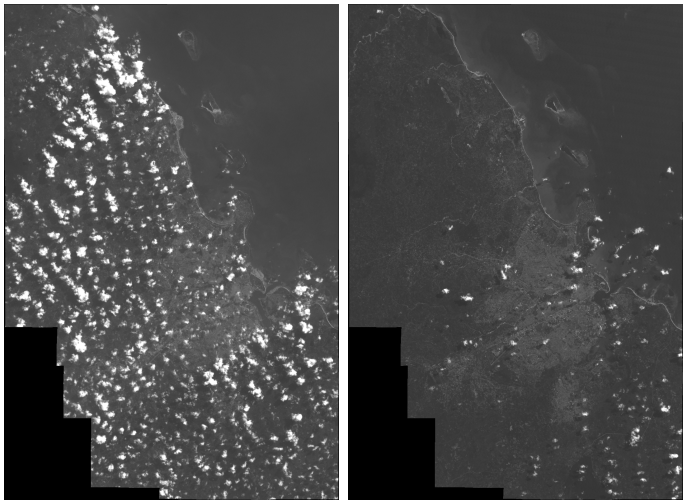
Due to the fact that time series images are acquired at different dates or possibly from different sensors, each image of the time series may be affected by different viewing conditions which are specific to each date. These effects may lead to missing or altered information. For instance, clouds prevent us to observe the ground and shadows alter the observation. Depending on the applications, these affected regions should be removed and then reconstructed, or just detected and masked.

Some attempts to deal with this issue can be found in the literature. For instance, there are several methods for cloud detection such as region-growing algorithm [36], mean shift algorithm [134] and for cloud removal such as temporal interpolation [73], unmixing of objects [35]. Recently, in [190], authors proposed deep learning method for missing data reconstruction. Besides, there are several methods for remote sensing image denoising such as [70].

Figure 3.2 demonstrates two examples with bi-temporal images. The first example includes cars in a very high resolution images. These cars are not stable through time and can be seen as a noise in some applications or as objects of interest in other ones. The second example shows clouds in different images which occlude different places.

It should be noted that radar data is not affected by the presence of clouds, and that hierarchical representations applied to SAR (Synthetic Aperture Radar) data are documented in several works such as [4].

(a) Cars as noise



(b) Cloud effect

Figure 3.2 – Examples of instantaneous effects.

### 3.1.3 Image registration and viewing angle

New space missions may provide data using several satellites or even a constellation of satellites such as Sentinel [14]. Therefore, time series samples may be acquired with different viewing angles. These differences affect the appearance of the objects especially for very high resolution imagery. For instance, while one image taken from nadir (directly below the satellite) will show only the roof of a building, another image taken from another angle will also show the facade of the same building. Different viewing angles can also alter the geolocalization of images

and cause misregistration between them. Since SITS processing has to be able to track or detect objects through time, misregistration or viewing angle challenges must be tackled.

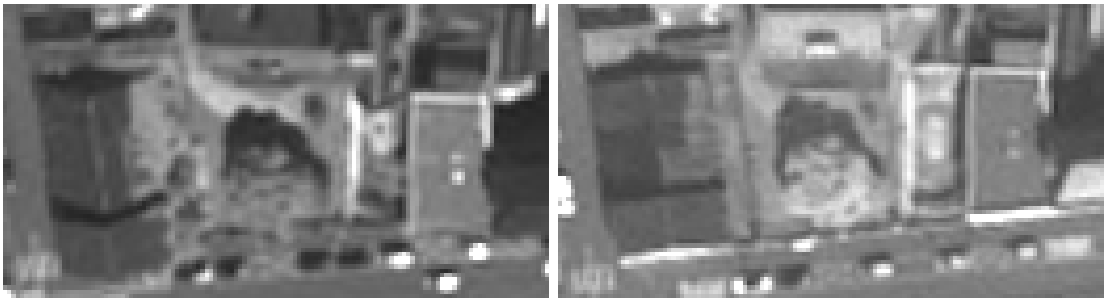
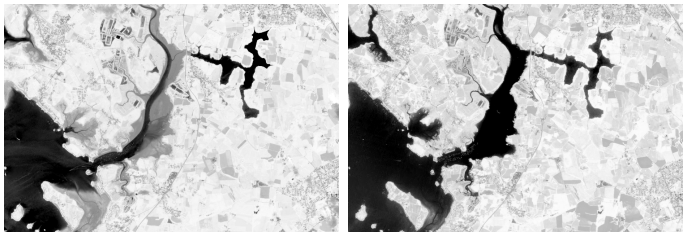


Figure 3.3 – Example of viewing angles difference.

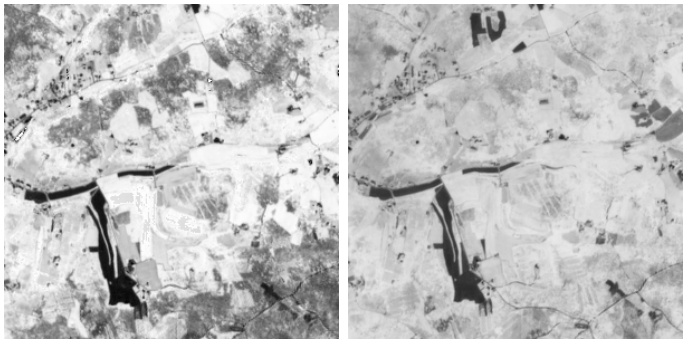
Figure 3.3 shows bi-temporal very high resolution images of the same scene and the difference between their viewing angles exhibits facades of the buildings which are neither new objects nor changes in time. In other words, same locations will have different appearance in successive images. This difference will be the cause of inefficiency of some methods, especially the pixel-based ones.

### 3.1.4 Dynamic classes

Appearance of some objects on Earth changes through time although their physical definition as objects remains the same. For instance, observed crops are in different status such as irrigation, seeding or harvesting during the year and this status also depends on the farmers' choices. As a second example, tide affects the observed bathymetry or sea coastline depending on the acquisition time. However, this region should belong to the same class since it has a consistent lifecycle. We call these classes as dynamic because they have different properties compared to steady classes. Such an issue was investigated with a graph based method in [69] to track dynamic clusters. However, this challenge is not a popular topic in the SITS community. Moreover, dynamic classes have not been strictly defined yet in the literature (and to the best of our knowledge). The reason could be the lack of multi-temporal reference data.



(a) Tide zone as dynamic class



(b) Crops as dynamic classes

Figure 3.4 – Examples of dynamic classes.

Figure 3.4 shows two different real examples of dynamic classes. For the intertidal zone example, the same location in the image may be classified as water, sand or mug depending on the acquisition time. The second example displaying Sentinel-2 Normalized Difference Vegetation Index (NDVI) images shows that intensity of fields is changing dramatically although they belong to the same crop class.

### 3.1.5 Temporal streaming

As mentioned in chapter 1, muti-temporal images become more and more available, and such data can be viewed as streams of images. Therefore, a SITS is continuously enriched/updated with some new data coming with a periodicity which depends on the satellite revisit time. Processing the stream of images is an important challenge for data analysis because we can not afford to reprocess the whole stream when new incoming data arrives. Moreover, shorter revisit time might make this challenge even more important.

For instance, once a tree has been built for a SITS, it should be updated with new incoming data. In the literature, updating a tree according to the new data distribution is known as the dynamic tree problem [152]. Although there are some works for graph based streaming such as [185, 157], there is no related work with tree based image representations.

### 3.1.6 Scalability

One of the challenges is the scalability of the available methods. A typical RS scene can be as large as $10,000 \times 10,000$ or even more which is much larger than images acquired with consumer devices. Moreover, when we cover larger areas with mosaicked images, the amount of pixels increases even further. Along the time dimension, SITS may include hundreds or even more images. Therefore, high computational resources and scalable algorithms are needed. While input information is getting larger, algorithms should preserve their efficiency.

Popular paradigms able to address this scalability issue includes the tile-based approach [127] and the Map-Reduce framework from the Hadoop stack [138]. Another strategy to design scalable algorithms is to rely on multiscale representations. Among them, morphological hierarchies [101] have shown their ability to lead to fast processing of big remote sensing data.

### 3.1.7 Lack of reference data

Another challenge is the lack of ground truth data. In the literature, ground truth is needed to train a model for a specific application, e.g. a supervised classification algorithm for land cover mapping [87] or change detection [25]. Obtaining reference data for an application of remote sensing is difficult or expensive. This is even worse when dealing with SITS because reference data for each time sample could be required especially for dynamic classes.

There are some publicly available ground truth datasets for some applications. For instance, Institut géographique national (IGN) [1] provides a reference for segmentation of Paris city. Cesbio [74] provides Land Cover Map of France for each

1. **http://data.ign.fr/benchmarks/UrbanAnalysis/**.

year or some shape files can be found to analyze disasters from Copernicus Emergency Management Service [2]. But these references do not include dynamic classes and they can only be used for specific applications.

There are several methods to overcome missing reference data issues such as active learning [149] or semi-supervised learning [192]. Recently, the focus has been put on unsupervised methods such as [26, 135, 133] which are very efficient to tackle the lack of reference data because they do not need any prior information about the scene.

## 3.2 State-of-the-art of SITS analysis

In this section, we review existing methods for SITS analysis and cover successively: methods which are using only pixel information, methods which are using spatial relationships between pixels, and, finally methods which are using hierarchical representations.

### 3.2.1 Time series analysis

Time series data can be one dimensional such as financial data, electrocardiogram (EGC), weather forecast, spectrogram, sound or a signal such as one pixel location in SITS. In this section, we will focus on works that use only the temporal dimension of SITS without considering the spatial information they contain. These approaches are also called pixel-based methods. Pixel-based approaches for SITS can be divided into three main strategies: using pixel-based features, using similarity metrics and using deep learning. We will review these strategies through two main applications of SITS analysis: land cover/land use mapping and change detection (CD). Land cover and land use mapping aims at providing information to understand the content of the land and it is the most studied application of SITS [60]. In order to create a land cover/land use map, pixel time series should be classified according to their properties. Change detection is another important challenging problem for SITS researchers and result of CD could be multi-temporal such as in change monitoring applications [27].

2. **https://emergency.copernicus.eu/**.

Each pixel of a SITS has a temporal profile based on the intensities which can be used as a feature. For instance, radiometric evaluation of pixel time series are used as a feature for classification in [88]. Moreover, in [64], authors use radiometry of long pixel time series for classification efficiently. Besides, new features can be extracted from each pixel time series. For instance, seasonality metrics (e.g. beginning of season) are extracted from SITS in [80]. These metrics are used to extract phenological features from crops [91] in order to classify them. In [184], regression analysis of pixel time series is used as a feature for classification. Some of the challenges listed in the previous section can hinder these features computation. For instance, nonlinear adaptation is considered in [9] in order to overcome noisy data.

Similarity metrics can be used to compare two 1-dimensional time series by calculating the distance between them [51]. In the literature, similarity metrics have been used to analyze pixel time series for several decades. For instance, a change detection algorithm was proposed with similarity measures in [2]. Although there are several similarity metrics to analyze SITS such as Euclidean Distance [31], DTW is gaining importance due its ability to overcome the irregular sampling challenge [179]. DTW is used for an efficient classification of satellite image time series in [129] and [96].

Deep learning has recently become the state-of-the-art method for almost all computer vision and image processing research areas. It is used to evaluate a model with a lot of parameters for a specific application. These parameters are used to extract features. However, these features have no physical meaning contrary to the features based on seasonal metrics mentioned previously in this section. In [72], researchers are using Recurrent Neural Networks (RNN) for pixel based classification of SITS. One dimensional convolutional neural networks (CNN) is proposed in [121, 191] to be able to process pixel time series. Besides, there are some works using attention mechanisms for pixel based time series classification such as [52],[140]. For change detection applications, deep learning based methods are also very popular. As an unsupervised change detection method, deep change vector analysis is proposed in [141].

Although they are efficient methods, pixel-based analysis of SITS has some drawbacks. Pixel-based approaches will not be robust against some challenges such as instantaneous effects and viewing angle. Moreover, processing each pixel separately is computationally expensive for large scale images. Therefore, recent works focus on spatial context based approaches.

### 3.2.2 Spatial information in time series

Analyzing evolution of objects through time is an important topic for SITS processing. Spatial information such as size and shape of these objects can be provided by object based analysis of a single image. Using jointly the time and space dimensions can provide information about evolution of objects. When spatial information is combined with temporal evolution, methods become robust according to the recent literature [155]. The spatio-temporal term is used when both spatial and temporal dimensions are considered jointly. We recall here important works that have been conducted with spatial or temporal information, but without hierarchical representation. We will review these approaches according to their strategies: tracking evolution of objects through time, modelling objects with spatio-temporal features and deep learning based methods.

A first approach relies on using predefined or identified objects in order to track their evolution through time. For instance, sequential patterns are explored in [81] by taking into account the connectivity of pixels in order to track their evolution through time. These patterns can also be used to summarize contents of SITS [98]. Spatio-temporal evolution of crops is explored in [156] using connected components. In [83], authors propose a tracking approach by observing a predefined object through time using a graph representation. However their method is limited compared to tree processing tools such as filtering and pattern spectra. Also, this method implies to select an object from the image in advance.

Researchers also aim at using specifically designed object based features [67] or extending their current methodology to object level features [118] in order to capture spatial or joint spatio-temporal information. These features can be calculated by first segmenting the images and then characterizing the segmented regions with their texture such as in [128]. They have been demonstrated as more efficient

compared to pixel based features for spatio-temporal data [183]. For instance, object based cropland mapping with Sentinel-2 time series outperforms pixel based mapping of the same SITS [10]. Change detection can also be implemented by comparing objects based features of two images such as in [178]. Recently in [183], spatio-temporal data cube is used in order to extract spatio-temporal features for both land cover mapping and change detection applications. In this thesis, we have made the choice to use these advanced spatio-temporal features with tree representations. Moreover, to the best of our knowledge, multiscale analysis of spatio-temporal features is missing in the literature.

Deep learning is also used to extract spatial features from SITS. Researchers use learning based methods to learn spatial and temporal representations in [11, 75]. A dedicated architecture called Convolutional Long Short Term Memory is used in [139] for land-cover mapping. Recurrent convolutional neural network is used in [108] in order to learn the temporal behaviour of objects. 3D convolutional neural network is used for classification in [77] in order to learn spatial and temporal features simultaneously. Recently, neural network autoencoder for change detection has been proposed in [82]. Authors combine their work with graph-based approaches in order to deal with multi-temporal images. One of the disadvantages of the deep learning methods is that they need large amount of labelled data for their training phase.

### 3.2.3 Hierarchical representation for time series

In this section, we discuss existing works using hierarchical representations taking into account time dimension which will be used as a basis of our work. These works are presented according to the input data they use: video or SITS.

Hierarchical representations dealing with time dimensions have been first used for video processing and have been introduced rather recently. There are few works related to tree representations [103, 102, 117]. In [103], a segmentation using tree-based representations is proposed to handle multidimensional sets of images and is applied on colour images or on videos. In order to deal with multivalued pixels, the authors propose to use Chebyshev Distance as a dissimilarity metric between neighbouring pixels in spatial domain only. These measures are then used to

build a partitioning tree, namely an $\alpha$-tree. In [102], in order to track moving objects, authors compute an $\alpha$-tree for each frame. Two consecutive $\alpha$-trees are then merged into a single $\alpha$-tree. Their goal is then to identify, among the nodes of this merged $\alpha$-tree, the ones corresponding to the selected object that has been marked in the initial frame. While this merged $\alpha$-tree links two consecutive frames, it does not span over the full video sequence. This approach allows for processing very long video sequences as well as video streams. Another important work [117] starts with a reliable identification of object trajectories along the frames. Then, the algorithm constructs a Trajectory Binary Partition Tree by iteratively merging neighbouring trajectories and builds a hierarchical representation of the entire sequence.

Extension of tree representations from videos to SITS was presented in [3] for the first time. The authors consider two different alternatives. The first approach involves a 3-D space-time processing using 10-connectivity. Each pixel is connected to the eight neighbouring pixels in the same acquisition (i.e. date) and to two pixels on the same position in the acquisitions just before and after. In this case, the temporal dimension is included within the signal model, and the goal is to combine pixels of the scene presenting the same temporal evolution among the acquisitions. Such an approach is leading to the space-time (ST) BPT. The second approach uses only spatial connectivity but embeds the temporal dimension. In this approach, the temporal information is included within the collocated pixels and 2D data is obtained. After that, the classical 8-connectivity can be applied. Such an approach produces a tree which is called temporal evolution (TE) BPT. However, both approaches rely only on standard spatial attributes and do not exploit possible temporal attributes. Moreover, their target application was segmentation which does not cover all application needs. Another work is a graph based approach [63] in which image objects computed at each individual time stamp are connected across the time series and generate a set of evolution graphs. Recently, Morphological Profiles (MP) have proven their efficiency to extract spatial features for land cover mapping from SITS [53]. Several other efforts have also been conducted to extend hierarchical representations to multivariate [6] and multimodal [162] images.

# Chapter 4

# Tree Representations for SITS and their streaming

## Contents



This chapter corresponds to our published paper *Component trees for image sequences and streams* [169] which is about tree building strategies for image sequences. Adding a temporal dimension to a 2D digital image offers several alternatives to build a morphological hierarchy. The spatial and temporal dimensions may be considered successively or simultaneously. The tree may contain nodes defined in the spatial, temporal, or spatio-temporal domains, thus leading to specific hierarchies. For each hierarchy, we explain how it can be built from an image

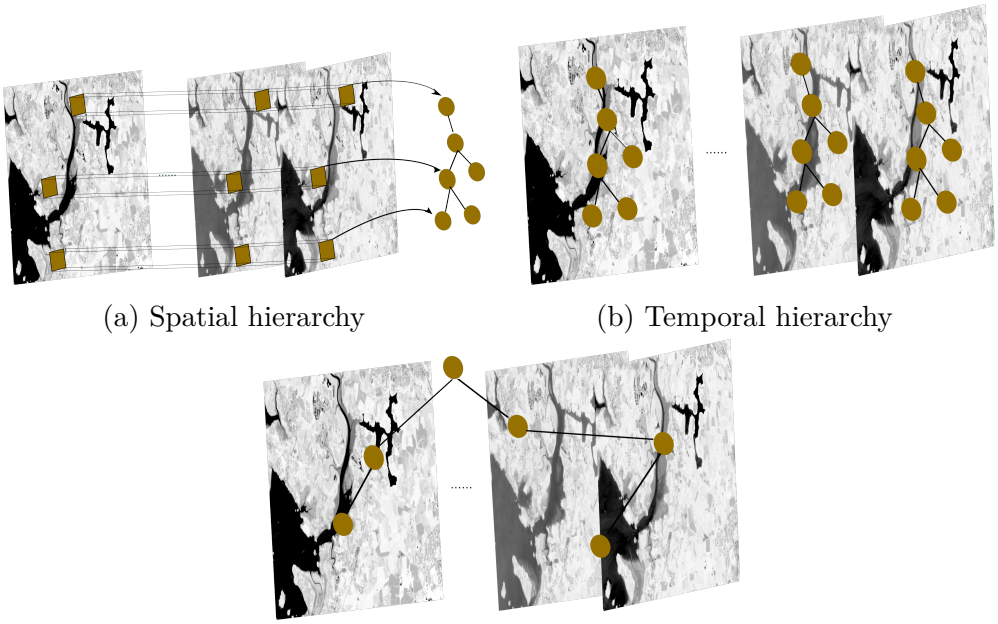
sequence, relying on existing definitions and algorithms introduced in chapter 2 on 2D images. In this chapter, we also cope with the streaming challenge which was explained in section 3.1.5. Instead of reconstructing a new tree from scratch when a new image is appended to the SITS, we seek for a scenario to update the existing tree with the new information brought by this additional image.

For the sake of simplicity, we will consider only grayscale image time series. Nevertheless, extension of the proposed methods to multivariate data (e.g. color or multispectral image time series) can be achieved by relying on existing solutions for multivariate morphology [6] and hierarchies [110, 30]. Similarly, while we focus here on the min and max-trees, most of the proposed solutions can be also easily adapted to other kinds of trees, either to inclusion trees (e.g. ToS) or to partitioning trees (e.g. $\alpha$-tree, BPT, *etc.*).

With image sequences, the coordinates combine both spatial and temporal information, noted $x = (i, j)$ and $t$, respectively. Here $t$ represents one time stamp within the entire sequence of length $n$, and we note $I_t$ the corresponding image. We also write $I_t(x)$ the value at spatio-temporal coordinates $x$ and $t$, and $I(x)$ the pixel-wise time series at spatial coordinate $x$. We will use the $\tau$ subscript to denote the presence of temporal information (i.e. temporal support), e.g. writing $I_\tau$ an image sequence instead of the single image $I$. Conversely, the $\varsigma$ subscript will be used to emphasize timeless information such as single frame image. Figure 4.1 provides the different kinds of trees that are introduced in the next sections to describe a SITS. In the following, we first describe the spatial hierarchy in section 4.1, then, the temporal hierarchy in section 4.2 and lastly, the spatio-temporal hierarchy in section 4.3. Section 4.4 discusses complexity of streaming algorithms.

## 4.1 Spatial hierarchy

Despite the fact that an image sequence combines both spatial and temporal information (i.e. every data instance is made of a spatial location and time stamp), it can still be modeled by a spatial-only hierarchy. In this case, the series of image frames $I_\tau = \{I_1, \ldots, I_n\}$ is first mapped into a single, representative image $I_\varsigma$ (or in other words, $f : I_\tau(x) \mapsto I_\varsigma(x)$). Then, a spatial hierarchy (SH) is



(a) Spatial hierarchy (b) Temporal hierarchy

(c) spatio-temporal hierarchy

Figure 4.1 – Tree representation strategies: (a) a tree structure is built based on pixel time series, (b) a tree structure is built for each image separately and these trees can be optionally merged, (c) the time series image is considered as a 3D image and a tree structure is built in three dimensions.

built from this representative image. By doing so, every node of the tree has a spatial support, similarly to still images but its content also takes into account temporal information. In order to ensure compatibility with the standard max-tree algorithm, we assume the representative image to be defined on grayscales, $I_\varsigma(x) : \Omega \to V \in \mathbb{Z}$ where $\Omega$ is a non empty finite set as in chapter 2.

In order to obtain the representative image, we use two strategies; relying either on the relationships of the pixel time series between each other, or on the distribution of each pixel location through time. Then, a tree is built from this single frame representative image such as recalled in chapter 2. We present here these two strategies, called time series ranking and time series projection.

### 4.1.1 Tree building

#### Time Series Ranking

In order to derive a single image representative of a SITS, and to use it subsequently to build a max-tree, a first approach consists in ranking all pixel time series $I_\tau(x)$. Such a strategy provides a behavior similar to a standard vector ordering used in multivariate morphology [175], with an ordering among pixel time series. We recall that a binary relation that is reflexive and transitive is called a pre-ordering (or quasi-ordering). It becomes an ordering if it is also anti-symmetric. Both pre-orderings and orderings are either total if the totality property is met or partial otherwise. Totality property is required here since comparison between all pixel time series is needed to produce the representative image by means of ranking.

We thus distinguish between total orderings and total pre-orderings. The former is the most satisfactory solution since it ensures a bijection between the set of pixel time series and their ranks. In other words, such a rank image can be modeled as a max-tree, processed with dedicated algorithms, and the resulting (updated) rank image reprojected into an image sequence. Indeed, thanks to the total ordering of all time series, it is possible to replace every rank by the corresponding original time series. This behavior is appealing but unfortunately the total orderings are often unbalanced in practice [7]. Nevertheless, we will experiment such a strategy with the most popular solution, namely the lexicographical ordering that compares the values in the first time stamp, and proceeds to further comparison on the second time stamp only if ties appear on the first one (and so on until reaching the last time stamp to assess a strict equality between vectors). While lexicographical ordering is common for color morphology such as in [125] using this ordering for pixel time series has never been explored to the best of our knowledge.

Relaxing the anti-symmetry constraint leads to the so-called pre-orderings. Ranking time series with a pre-ordering brings one major drawback. Due to the lack of anti-symmetry, it is possible for two different vectors (or time series) to be given the same rank. It does not prevent to obtain a representative image, to build a tree out of it, to process the tree and to return a resulting representative

| Operation | Description |
|---|---|
| create(n) | creates a node “n” |
| insert(n,v) | adds pixel “v” to node “n” |
| link(n,w) | creates a path between “n” and “w” nodes |
| merge(n,w) → m | merges nodes “n” and “w” into the node “m” |
| cut(n) | removes node “n” |
| addnode(n,t) | add node “n” to tree “t” |
| remove(v) | removes pixel “v” from its node |
| neighbours(v) | Returns the pixels adjacent to “v” |
| findchangedplaces(I,I’) | Finds the changed locations |
| findparent(n) | Finds the parent node of “n” |

Table 4.1 – List of operations used in the streaming algorithms.

image. Nevertheless, it will not be possible to reconstruct the image sequence from this updated rank image without an arbitrary choice among the time series having equal ranks. So this strategy is relevant only if the objective does not require reconstruction (thus tree filtering is out of reach). As a pre-ordering strategy, we focus here on reduced orderings which reduce vectors to scalar values before ranking them according to their natural order. Among reduced pre-orderings, distance-based approaches are very popular to order vectors according to their distance from a reference vector [34]. In [5], authors proposed a similar concept for color morphology. In the context of time series, one can rely on dedicated distance metrics such as Dynamic Time Warping (DTW) [109]. Since the choice of a reference time series might be a challenging problem, it is possible to rely instead on the average distance to a set of reference pixels.

The construction of the representative image is then done in two steps. First, all pixels in the image time series are ordered by applying a given vector ordering or pre-ordering on their respective time series. The pixels are then ranked, starting with rank 0 for the smallest one, and ending with rank $\mid \Omega - 1 \mid$ (in case of total ordering, possibly lower otherwise) for the largest one. Construction of the tree is then achieved based on the pixel ranks.

#### Time series projection

Time series projection aims at mapping the vectors into scalar values on which the comparison is performed such as in [176] for color images. Compared to time series ranking, this approach takes into account pixel time series as well but it also offers some specific advantages. First, applying the projection is straightforward since it only consists in relying on the natural ordering between scalars. Conversely to the previous strategy, there is no need to rank pixels to build the representative image. Furthermore, the computation of the representative image can be performed independently for each pixel allowing a parallel implementation.

The projection relies on a function $f : \Omega^n \to \mathbb{R}, x \mapsto f(I_1(x), \ldots, I_n(x))$. Let us note that this projection function $f$ can output real values, thus needing to rely on tree construction algorithms able to cope this kind of data. We still write $I_\varsigma(x)$ the output scalar value for this location. Depending on the application context, a wide range of functions are available e.g. mean, median, standard deviation, range, *etc*. In the most probable case of a non-injective function, this strategy also faces the difficulty of reconstructing the image sequence from the processed tree. It should be noted that time series ranking is sensitive to order of images whereas one can use a function for time series projection which may not affected by this order.

### 4.1.2 Streaming of Spatial Hierarchy

#### Time series ranking

The ranking approach requires the comparison of all time series from the image sequence. With the arrival of a new image, the ranks most probably need to be calculated again. The only exception is with a total ordering relying on the chronological order like the lexicographical ordering. For such an ordering, only pixels which were equally ranked need to be compared taking into account the updated time series.

Table 4.1 provides some operations used in all algorithms introduced in this manuscript. Figure 4.2 illustrates two updates of a max-tree after two changes of a single pixel. $I_r$ represents the image of ranks and it is successively updated as

**Algorithm 1** Update SH

**Require:** $\mathcal{T}(I_r), I_r, I'_r$
**Ensure:** Updated Tree: $\mathcal{T}(I'_r)$
1: $l = findchangedplaces(I'_r, I_r)$ *// changed locations*
2: $\mathcal{T}(I'_r) = \mathcal{T}(I_r)$ *// initial tree*
3: **for** *each changed location* $x$ *in* $l$ **do**
4: $a = I_r(x)$
5: **if** $\mathcal{C}(C_a) < I'_r(x) < \mathcal{P}(C_a)$ **then**
6: insert$(C_a, I'_r(x))$
7: **else**
8: create$(C'_a)$
9: insert$(C'_a, I'_r(x))$
10: findparent$(C'_a)$
11: link$(C'_a, \mathcal{P}(C'_a))$
12: **if** $neighbours(a) = \emptyset$ **then**
13: cut$(C_a)$
14: link$(\mathcal{C}(C_a), \mathcal{P}(C_a))$

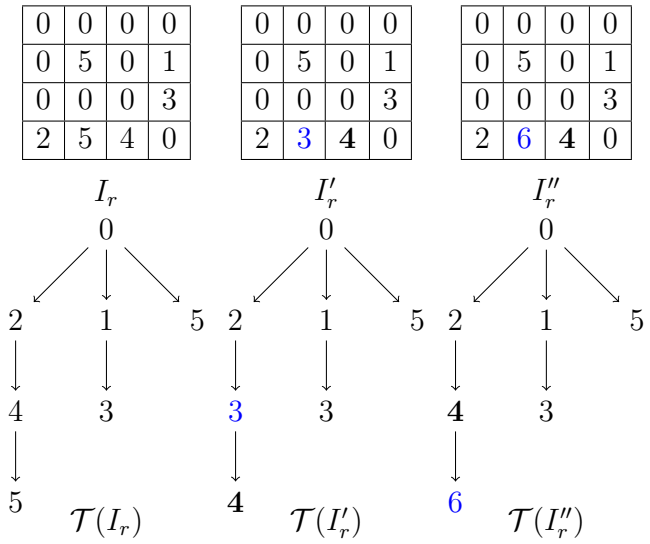



Figure 4.2 – Streaming for spatial hierarchy

$I'_r$ and $I''_r$. Tree nodes are represented by their level. The blue numbers represent changed pixels, and the bold ones represent pixels that do not change in the initial image but in the tree.

We provide in Algorithm 1 the pseudo-code to update the spatial max-tree built on ranking. We note $I_r(x)$ the ranked pixel time series in a location $x$ and $I'_r(x)$ its updated value. $\mathcal{P}$ and $\mathcal{C}$ are functions which provide the levels of the parent node and the child node respectively. When a new image appears, $I_r$ will be updated to $I'_r$. In that algorithm, first, locations of the changed pixels are determined, then changed values are replaced on a tree according to their value and connectivity. If a changed value is between the parent node level and the child node level, it remains in the same node. Otherwise, its location will change and a new node should be created for the new value. This algorithm is repeated for each changed location. More precisely, the time complexity increases if the amount of changed value increases. Lines 12, 13 and 14 remove the node if there is no remaining pixel in the node.

#### Time series projection

The streaming rules are the same as in Algorithm 1. To ease updating, one might prefer functions that do not require a full recomputation. Thus, the updated projected image can be created with a parallel computation.

## 4.2 Temporal hierarchy

The temporal hierarchy (TH) allows us process each image independently and to build the tree representation for each of them separately. This strategy relates either to the marginal approach in the literature [41] where trees are built and subsequently processed separately or to tree structures comparison [23]. We first explain the tree building process and then, show how to update this tree during the streaming step.

### 4.2.1 Tree building

The temporal hierarchy (TH) aims to provide a temporal set of trees instead of a single, timeless one as with SH. For a given image sequence $I_\tau$, every single image $I_t$ of the series leads to a given tree $\mathcal{T}_t$. The final set is defined as:

$$\mathcal{T}_\tau = \{\mathcal{T}_t\}_{t=[1,n]} \tag{4.1}$$

where $n$ represents the amount of images in the set. Every image $I_t$ of the series have its own tree $\mathcal{T}(I_t)$.

After building a tree for each image, we have two options to process them. We can process each tree independently (e.g. for extracting features) and stack the results together [6]. The second option is to merge the individual trees together and then process this merged tree. The critical merging step is a special case of the streaming process that we will now explain.

### 4.2.2 Streaming of Temporal Hierarchy

In the marginal case, streaming is straightforward and consists in sequentially adding a new tree to the set of initial trees. On the contrary, merging trees is not a straightforward process due to the memory and computational costs brought by the spatio-temporal connectivity. Existing methods to build a max-tree usually proceed by merging trees corresponding to some small parts in the image separately [61], while tree of shapes [30] or $\alpha$-tree [68] have also led to parallel algorithms for merging and [107] deals with extreme dynamic range data by merging sub-trees of sub-images.

Figure 4.3 illustrates the merging of two max-trees by increasing the connectivity from 4 to 6 neighbours. It should be noted that, the number of temporal neighbours to connect two trees should be selected before building the merged tree. The two images on the left are shown with their respective trees. The merged tree is given on the right. For the sake of conciseness, we have omitted intensity values from the tree representation of the two initial images.

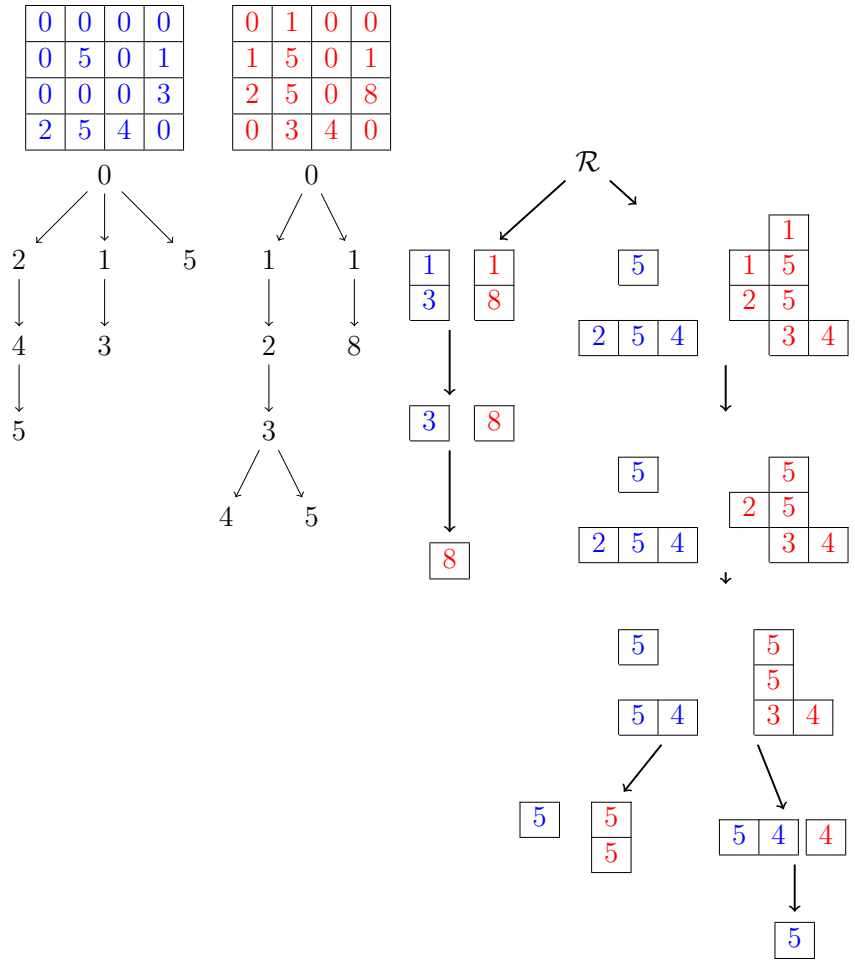



Figure 4.3 – Tree for each date and merged tree: two images and their max-trees (left), and the corresponding merged tree (right).

Algorithm 2 provides the pseudo-code for merging two max-trees. $\mathcal{R}$ represents the root of the tree and $\mathcal{T}_m$ is the merged tree. Besides, $C_\lambda^1$ and $C_\lambda^2$ represent the nodes for first and second image respectively. The output of Algorithm 2 is a multidimensional tree, i.e. a space-time tree which will be explained in the next section. More precisely, the connectivity of $\mathcal{T}(I_1), \mathcal{T}(I_2)$ is 4 but the connectivity of $\mathcal{T}_m$ is 6. Since $\mathcal{T}_m$ still includes pixels from previous images, the streaming preserves past information. In that algorithm, first, the initial root is created. Thus, two

**Algorithm 2** Update TH

**Require:** $\mathcal{T}(I_1), \mathcal{T}(I_2)$
**Ensure:** $\mathcal{T}_m$
1: $\mathcal{R}(\mathcal{T}(m)) = merge(\mathcal{R}(\mathcal{T}(I_1)), \mathcal{R}(\mathcal{T}(I_2)))$ *// initial root*
2: **for** $\forall\lambda$ **do** *// for all levels*
3: **for** $\forall C_\lambda^1 \in \mathcal{T}(I_1)$ **do**
4: **for** $\forall C_\lambda^2 \in \mathcal{T}(I_2)$ **do**
5: **if** $C_\lambda^1 \cap C_\lambda^2 = x$ and $I_1(x) = I_2(x)$ **then**
6: $a = I_2(x)$
7: $C_a = merge(C_a^1, C_a^2)$
8: $addnode(C_a, \mathcal{T}_m)$
9: **else**
10: $addnode(C_\lambda^1, \mathcal{T}_m)$
11: $addnode(C_\lambda^2, \mathcal{T}_m)$

trees create one tree. Then, the connectivity between nodes is investigated from root to leaves. If a node of the first tree has no connectivity to any nodes of the second tree, it is kept in the tree without merging.

Merging is applied to the trees chronologically. After building the merged tree $\mathcal{T}_m$ with $\mathcal{T}(I_1)$ and $\mathcal{T}(I_2)$, a new merging process can be implemented with $\mathcal{T}(I_3)$. Then, the algorithm should take into account the last image of $\mathcal{T}_m$ which is $I_2$ and the new image $I_3$. This process can continue with each incoming image.

## 4.3 Spatio-temporal hierarchy

The last strategy is the spatio-temporal hierarchy (STH). In this approach, one single tree is built for the whole image time series and nodes of this tree contain the full spatio-temporal information.

### 4.3.1 Tree building

The image sequence can be seen as a spatio-temporal cube, where each date corresponds to one layer, the two other dimensions being the spatial ones. From this cube it is possible to build a spatio-temporal tree, i.e. a single tree $\mathcal{T}(I_1, \ldots, I_n)$ that includes all the information from the whole time series. However, building

such a complex tree given a long time series is particularly challenging. While 3D data has received less attention than 2D data, few works have been proposed to compute trees on such data, e.g. [180] and [181] for 3D computed tomography (CT) images, [62] for 3D point clouds, or [3] for 3D SAR time series. While the trees built with the two previous strategies SH and TH have only a spatial support (shall it be timeless $\mathcal{T}_\varsigma$ or related to a single frame $\mathcal{T}_t$), here a spatio-temporal tree intrinsically combines both spatial and temporal dimensions. We call this representation a space-time tree. We write $V$ the set of nodes gathering all the spatio-temporal connected components:

$$V(I) = \bigcup_{\forall k,\lambda} C^k_{\lambda,\tau}(I) \tag{4.2}$$

where $C_{\lambda,\tau}$ is spatio-temporal connected component. Furthermore, we can split a spatio-temporal connected component $C$ into its spatial slices extracted at each time stamp $t$:

$$C^k_{\lambda,\tau} = C^k_{\lambda,1} \cup C^k_{\lambda,2} \cup \ldots \cup C^k_{\lambda,n} \tag{4.3}$$

As already indicated, the connected components are built by gathering neighbouring pixels following a given spatial adjacency relation, e.g. 4- or 8-connectivity. As we now deal with spatio-temporal connected components and not spatial-only connected components, connectivity rules should be revised accordingly. To illustrate, let us consider the 8-connectivity defined in the spatial domain only. The corresponding neighbour set $\mathcal{N}_8$ for a given location $x = (i, j)$ can be defined as:

$$\mathcal{N}_8(x) = \{(i', j') \neq (i, j) \mid \max(|i - i'|, |j - j'|) = 1\} \tag{4.4}$$

It can be extended to time using 26-connectivity, where a location $(x, t) = (i, j, t)$ has a neighbour set $\mathcal{N}_{26}$:

$$\mathcal{N}_{26}(i, j, t) = \{(i', j', t') \neq (i, j, t) \mid \max(|i - i'|, |j - j'|, |t - t'|) = 1\} \tag{4.5}$$

In this manuscript, we also consider another connectivity rule, called continuous connectivity where the neighbours along the time dimension are not limited to the previous and next images. It considers a set $\mathcal{N}_\tau$ of $3 \times 3 \times n - 1$ neighbours:

$$\mathcal{N}_\tau(i,j,t) = \{(i',j',t') \neq (i,j,t) \mid \max(|i-i'|, |j-j'|) = 1\} \tag{4.6}$$

Figure 4.4 shows spatio-temporal connectivity. In that figure, there are 3 squares representing a 3×3 matrix. Neighbour list can be defined according to the black circle in the middle. Other circles are colored according to the connectivity rule. If all pixels are selected, the connectivity rule becomes a 26-connectivity whereas if only yellow pixels are selected, the rule becomes a 6-connectivity. Alternatively, blue and yellow pixels together are creating a 10-connectivity.

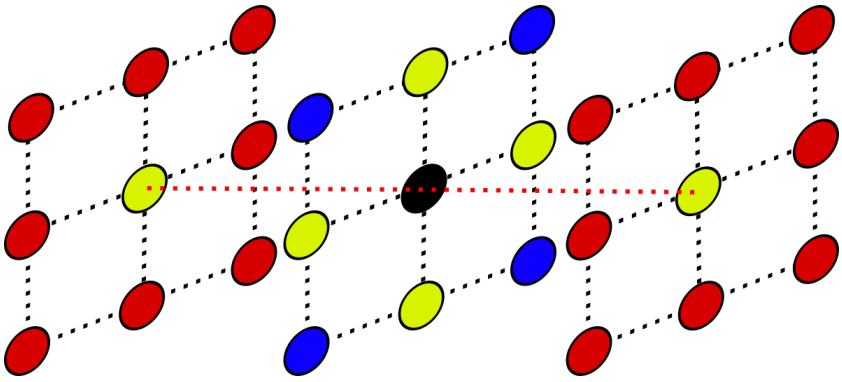


Figure 4.4 – Spatial hierarchy workflow

It should be noted that, space-time tree is not appropriate for building a ToS because the shape based hierarchy rule does not trivially transfer to the time domain. Although there are some works such as Multivariate-ToS [30] to extend this tree to multispectral images, it has not been used for multi-temporal images by the authors. This extension is not straightforward because multi-temporal images are not conventional multivariate data and shapes can change a lot through time.

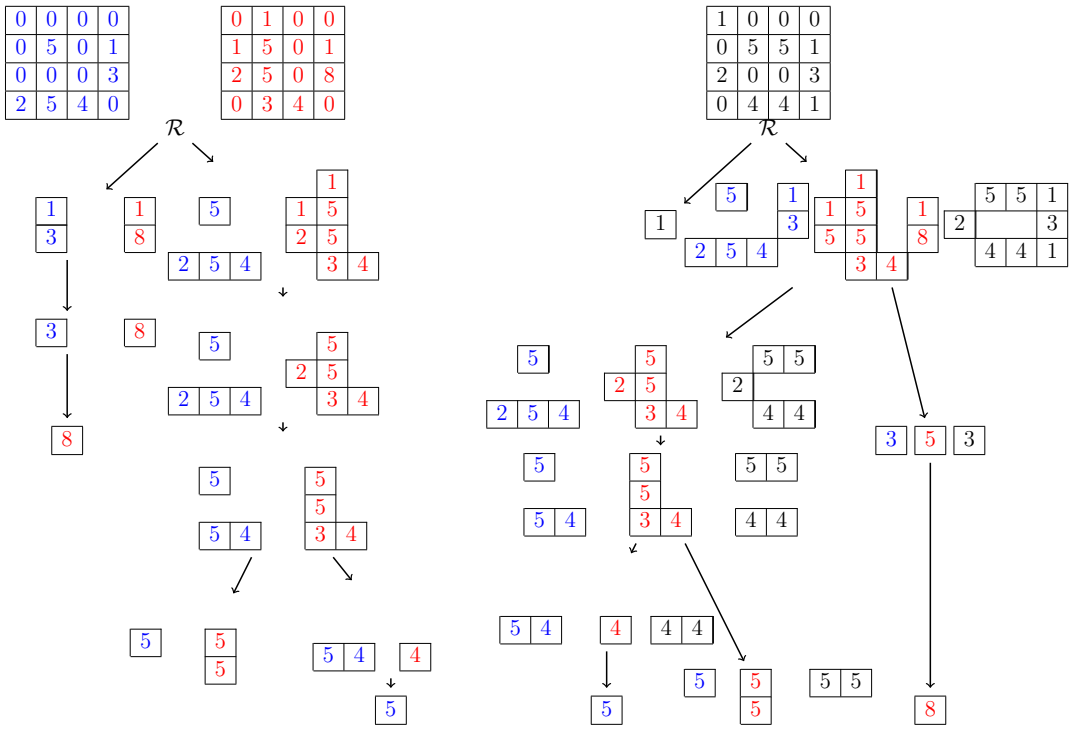


Figure 4.5 – Space-time tree and streaming (Two images and their space-time tree in the left, new image and updated space time-tree in the right)

### 4.3.2 Streaming of Spatio-Temporal Hierarchy

The streaming strategy for STH is similar to the one used for TH but here we use only a new image instead of a new tree. Figure 4.5 shows an updated space-time max-tree with the incoming third image. Different colors are used for different images in order to avoid confusion in the nodes. Two images and their space-time trees are shown on the left, the new image and the updated space-time tree being given on the right. Algorithm 3 provides the pseudo-code for updating a space-time max-tree. Here, space-time tree streaming uses the connectivity rule which was already chosen to build first the initial space-time tree. First, the root node is extended with all the pixels of the new image. Then new image pixels are placed on the tree according to their spatial location and their gray values.

**Algorithm 3** Update STH

**Require:** $\mathcal{T}(I_1, I_2), I_3$
**Ensure:** $\mathcal{T}(I_1, I_2, I_3)$
1: $insert(\mathcal{R}(\mathcal{T}(I_1, I_2)), I_3)$ // *initial root*
2: **for** $\forall x \in I_2$ **do**
3: **if** $I_2(x) = I_3(x)$ and $I_2(x) \in C$ **then**
4: $insert(C, I_3(x))$
5: $insert(C, neighbours(I_3(x)))$
6: **else**
7: $create(C_{I_3(x)})$
8: $findparent(C_{I_3(x)})$
9: $link(C_{I_3(x)}, \mathcal{P}(C_{I_3(x)}))$

Similarly to trees merging, streaming a space-time tree preserves the information from previous images. If merged tree and space-time tree are built with the same connectivity rule and the same image set, they are indeed identical.

## 4.4 Streaming time complexity

We applied our streaming algorithms to synthetic images which are shown in Figure 4.2, Figure 4.3 and Figure 4.5. We compared time complexity of the tree building strategies with streaming or from scratch in Table 4.2. The complexity of Algorithm 1 is $O(lc)$ where $c$ is the amount of changed pixels and $l$ is the number of tree levels used by this algorithm to find parent of the nodes. This depth $l$ value is $1 \leq l \leq 2^{bit} - 1$ where $bit$ is 8 or 16 depending on the image encoding. Algorithm 2 is $O(lk)$ where $k$ is the amount of nodes. According to [29], tree building for large integers requires at least a complexity of $O(p \log p)$ where $p$ is the amount of pixels of one image. Since satellite images are very large, the amount of pixels is usually huge, therefore the amount of tree nodes is lower than the amount of pixels in the image. For example, with 3 real Sentinel-2 images (10,980 px $\times$ 10,980 px), the computation times for tree building are 75s, 133s and 211s, and the required memory amounts are 3083MB, 5384MB and 7034MB for SH, TH and STH strategies, respectively. The space-time streaming requires only one image with $p$ pixels while building space time-tree from scratch requires

| Method | Streaming | Building from scratch |
|---|---|---|
| SH | $O(lc)$ | $O(p \log p)$ |
| TH | $O(lk)$ | $O(np \log p)$ |
| STH | $O(p + lk_{new})$ | $O(np \log np)$ |

Table 4.2 – Time complexity comparison ($p$: Number of pixels, c: number of changed pixels, $l$: search for a parent node, $k$: amount of existing nodes, $k_{new}$: amount of new nodes).

$n$ times that amount of pixels where $n$ is the number of images in a set. Besides, the complexity has an extra $hk_{new}$ caused by the new node creation part. For the SH strategy, we assume here that the ranking or projection has already been performed, so the default complexity is the same as for a still image. The SH streaming depends on the number of changed pixels. The proposed SH streaming algorithm is more efficient than building from scratch when the number of changed pixels satisfies the following relation:

$$c < \frac{p \log p}{l}. \tag{4.7}$$

For the TH strategy, we assume here that the building from scratch complexity considers no merging. The TH streaming complexity depends on the amount of nodes in the tree levels. For this case, better efficiency is ensured if $k$ denoting the amount of nodes satisfies:

$$k < \frac{np \log p}{l} \tag{4.8}$$

Finally, the STH streaming algorithm includes a process of creating new nodes. In order for the streaming to be relevant, the upper bound of the number of new created nodes should follow:

$$k_{new} < \frac{np \log np - p}{l} \tag{4.9}$$

Let us note that all our tree models were designed to deal with temporal information and are sensitive to the order of the images. Nevertheless, let us point two cases where shuffling the images will have no effect on the result: 1) with the SH strategy, it directly depends on the properties of the projection function

(e.g. using the average or standard deviation is unaffected by the shuffling, while using a DTW based ranking is). 2) with the TH strategy in the marginal case, it directly depends on the actual processing that is considered afterwards. To conclude, all tree building strategies enable us to tackle with the streaming challenge and streaming algorithms are more robust than building tree from scratch depending on the conditions in equations 4.7, 4.8 and 4.9. A detailed comparison of the proposed tree building strategies will be discussed in the next chapter.

# Chapter 5

# Comparison of SITS tree building strategies

## Contents



This chapter corresponds to our published paper *On morphological hierarchies for image sequences* [165]. In the previous chapter, morphological hierarchies have been extended to cope with image sequences, and different strategies have been proposed to allow their construction from spatio-temporal data. In this chapter, we compare these hierarchical representation strategies according to their structural

properties. Our goal is to determine the most effective hierarchical representation to analyze image sequences. We will show that the spatio-temporal hierarchy is more efficient compared to the other proposed strategies.

In section 5.1, we introduce a projection method to make these representations comparable. Section 5.2 presents comparison methods. Experiments were conducted on both synthetic and real datasets and are presented in section 5.3 and 5.4 respectively.

## 5.1 Tree projection

As explained in the previous chapter, tree building strategies for image sequences occur in different spaces which are spatial, temporal and spatio-temporal domains. However, to compare these hierarchical structures, they should lie in the same domain. Once a space-time tree has been built, it can be projected to the temporal or spatial domain. The temporal connectivity will provide structural difference between the projected tree and a tree built with SH or TH strategy.

### 5.1.1 Projection in the temporal domain

We define the projection of the space-time tree $\mathcal{T}_\tau = (V, E)$ in the temporal domain as the operator $\Gamma$ that returns a tree defined with a spatial support only, given a specific time stamp $t = [1, n]$:

$$\Gamma^t : \mathcal{T}_\tau \mapsto \mathcal{T}^t. \tag{5.1}$$

The new tree $\mathcal{T}^t$ will retain nodes and pixels from the given time stamp $t$ only. To do so, the projection operator first removes the nodes, which do not contain any pixels from $t$;

$$V \mapsto V' = V \setminus \{C^k_\lambda \mid C^k_{\lambda,t} = \emptyset\}, \tag{5.2}$$

thus leading to an intermediate tree $\mathcal{T}' = (V', E')$ where $E' \subseteq E$ only contains edges connecting nodes from $V'$.

Let us consider a leaf node containing pixels $(x, t)$ present in $I_t$. Since parent nodes are defined as supersets of their children (see equation 2.6), all the ancestors of the leaf node also have its pixels from $I_t$.

However, nodes of $\mathcal{T}'$ may have pixels which are not from the selected time stamp $t$. We should then shrink the nodes in $V'$ along the temporal dimension to limit their support to $I_t$:

$$V' \mapsto V'' = \{C^k_{\lambda,t} \mid C^k_\lambda \in V'\} \tag{5.3}$$

Again, we can define a new tree $\mathcal{T}'' = (V'', E')$. In this situation, we have not removed any nodes from $V'$ but have instead reduced their spatial support, there is no change in the set of edges $E'$.

The last operation consists in filtering the tree to remove the nodes that provide duplicate information. These nodes $C$ are those for which there is no associate ghost nodes $C'$ in the selected frame $I_t$, and whose added-value in the hierarchy was noticeable for some other time stamps. More formally:

$$\begin{aligned} V'' \mapsto V''' &= V'' \setminus \{C^k_\lambda \mid C'^k_{\lambda,t} = \emptyset\} \\ &= V'' \setminus \{C^k_\lambda \mid \nexists\, x \colon I_t(x) = \lambda\} \end{aligned} \tag{5.4}$$

i.e. we remove the nodes with a level which does not correspond to an existing intensity in the image $I_t$. The resulting tree is then defined as $\mathcal{T}^t = (V''', E'')$ where the edge set $E''$ has been updated according to the retained nodes in $V'''$. Let us observe that although $V''' \subset V''$, the inclusion relation between edge sets is not kept, i.e. $E'' \not\subset E'$. Indeed, since some internal nodes have been removed, there are now new edges to connect their parent and children nodes. Table 5.1 reports some information about space-time tree, internal trees and projected tree w.r.t $V$, $E$ and the amount of nodes in the trees.

| | Domain | Set of nodes | Set of edges | Amount of nodes |
|---|---|---|---|---|
| $\mathcal{T}$ | Spatio-temporal | $V$ | $E$ | $A_1$ |
| $\mathcal{T}'$ | Spatio-temporal | $V'$ | $E'$ | $A_2$ |
| $\mathcal{T}''$ | Spatial | $V''$ | $E'$ | $A_2$ |
| $\mathcal{T}^t$ | Spatial | $V'''$ | $E''$ | $A_3$ |

Table 5.1 – Comparison of space-time tree, internal trees and projected tree for temporal projection. ($A_3 < A_2 < A_1$)

### 5.1.2 Projection in the spatial domain

In order to allow comparison with SH, we consider the projection of the space-time tree in the spatial domain:

$$\Gamma^{\varsigma} : \mathcal{T} \mapsto \mathcal{T}^{\varsigma} \tag{5.5}$$

Similarly to the previous projection, we rely on successive steps to build $\Gamma^{\varsigma}$. But the main difference is that the resulting tree $\mathcal{T}^{\varsigma}$ is timeless and thus does not require any time stamp parameter $t$.

First, we filter out nodes for which timeless information cannot be extracted due to a limited temporal domain. These nodes are identified as those that do not appear in all time stamps of the series, leading to:

$$V \mapsto V' = V \setminus \{C_{\lambda}^{k} \mid \exists t \colon C_{\lambda,t}^{k} = \emptyset\}. \tag{5.6}$$

This leads to a new tree $\mathcal{T}' = (V', E')$ with $E'$ being the updated set of edges limited to those connecting the nodes from $V'$. Similarly to the temporal projection, this first step is not enough since here $\mathcal{T}'$ is not timeless yet. To do so, a second operation is required to remove the spatial locations for which the time series has missing values such as outliers described in section 3.1.2. In other words, we find intersected locations of spatial slices (see equation 4.3).

$$V' \mapsto V'' = V' \setminus \{C_{\lambda}^{k} \mid \exists x, \exists t, \exists t' \neq t \colon x \in C_{\lambda,t}^{k}, x \notin C_{\lambda,t'}^{k}\} \tag{5.7}$$

Since this step only reduces the spatial support of the nodes, it does not modify the edge set, thus we have $\mathcal{T}'' = (V'', E')$. While nodes are becoming smaller, amount of nodes is not changing. Pixels in the nodes of this tree are continuous time series without a gap in time.

Similarly to what has been proposed for SH, we need to finally apply a projection function on the pixel time series in order to obtain the timeless nodes. Since we consider here the max-tree, the proposed operation consists in finding the maximum value of each pixel time series, or equivalently removing the non-maximum pixels:

$$V'' \mapsto V''' = V'' \setminus \{(x,t) \mid (x,t) \in C^k_\lambda, \\ \exists t' \neq t, \exists \lambda' > \lambda \colon (x,t) \in C_{\lambda,t}, (x,t') \in C_{\lambda',t'}\} \quad (5.8)$$

Finally, the tree resulting from the projection $\Gamma^\varsigma$ can be defined as $\mathcal{T}^\varsigma = (V''', E')$. Table 5.2 gives differences between the trees involved in this projection.

| | Domain | Set of nodes | Set of edges | Amount of nodes |
|---|---|---|---|---|
| $\mathcal{T}$ | Spatio-temporal | $V$ | $E$ | $A_1$ |
| $\mathcal{T}'$ | Spatio-temporal | $V'$ | $E'$ | $A_2$ |
| $\mathcal{T}''$ | Spatio-temporal | $V''$ | $E'$ | $A_2$ |
| $\mathcal{T}^\varsigma$ | Spatial | $V'''$ | $E'$ | $A_2$ |

Table 5.2 – Comparison of space-time tree, internal trees and projected tree for spatial projection. ($A_2 < A_1$)

### 5.1.3 Properties of projected trees

Having defined the two projection methods $\Gamma^t$ and $\Gamma^\varsigma$, we now discuss some properties of the projected trees they return. Let us recall that there is only one projected tree $\mathcal{T}^\varsigma$ of the space-time tree in the spatial domain. Conversely, $n$ different $\mathcal{T}^t$ trees can be built from $\Gamma^t(\mathcal{T}^\tau)$. As shown previously, both projections provide only subsets of the initial node sets and do not create new nodes (but this property does not hold for the edge sets with the temporal projection).

In order to explain properties of the projected trees, we can study the trees in terms of isomorphism. Two trees are isomorphic one to each other if there is a one-to-one mapping from the vertices of one tree to the vertices of the other tree that preserves a vertice, edges and labels [37]. We consider here that the label of a node is simply its level ($\lambda$). This label is preserved when the tree is projected. Let $\mathcal{T} = (E, V)$ be a tree and $\mathcal{T}' = (E', V')$ one of its sub-trees. There are three types of sub-trees according to the literature [37];

1. *induced sub-trees* require $V' \subseteq V$ and $E' \subseteq E$;
2. *bottom-up sub-trees* are *induced sub-trees* which verify the additional condition: for any node $C \in V$, if $C \in V'$, then all the descendants of $C$ in $V$ are in $V'$;
3. and *embedded sub-trees* for which we have $V' \subseteq V$, and the ordering of nodes (ancestor and descendant relationship) is preserved.

According to the definition of sub-trees, $\mathcal{T}^{\varsigma}$ is an induced sub-tree of $\mathcal{T}_{\tau}$ and $\mathcal{T}^{t}$ is an embedded sub-tree of $\mathcal{T}_{\tau}$. The parent-child relationship changes for $\mathcal{T}^{t}$ because of equation 5.4.

## 5.2 Comparison of tree representations

Thanks to the two projections introduced in the previous chapter, we can now compare a space-time tree with the other hierarchies. We show here three different methods to ensure such a comparison, respectively based on node analysis, cost function and filtering capability.

### 5.2.1 Node analysis

Measuring the number of nodes contained in a tree provides a simple way to assess its structure. For instance, the number of parent nodes has been suggested in [123]. The number of nodes is an indicator of the complexity of a tree which is denoted as $\mid \mathcal{T} \mid$. If this value is high, the tree is considered having a complex structure [136]. Such a complexity can also be observed with attribute analysis from leaves to root such as in [16], where a node attribute along the path from

leaves to root is analysed through a so-called leaf attribute function (LAF). Every leaf has one path to root and the attribute value is changing from leaves to root. When the attribute value changes dramatically from one node to its parent, this node can be used as a significant feature. In order to find such elements, the LAF gradient is computed.

### 5.2.2 Dasgupta's Cost

Dasgupta's cost is an unsupervised measure of the quality of a hierarchical clustering [44]. In an inclusion tree, each internal node can have a leaf node with the definition of ghost nodes [20] as they are explained in section 2.2.1. Our trees act as a hierarchical clustering when ghost nodes are taken into account. Therefore, we will adapt this cost function to trees in order to compare their cost. Cost of building a tree is calculated by considering each pair of leaves:

$$Cost(\mathcal{T}) = \sum_{a,b \in \mathcal{T}} w_{a,b} \cdot \mid leaves(\mathcal{T}[a \wedge b]) \mid \tag{5.9}$$

$\mathcal{T}[a \wedge b]$ is the bottom up sub-tree rooted at the lowest common ancestor ($lca$) of $a$ and $b$ leaf nodes and $w_{a,b}$ is the edge weight between $a$, $b$ nodes. Here, each pair of leaves are processed separately. The interpretation of the cost function is an evaluation of the difficulty to split data into leaves by cutting the edges from the lowest common ancestor. The cost of a tree $\mathcal{T}$ is the sum of the splitting costs from the leaves. According to this cost, similar leaves (e.g. objects) should be penalized if they are merged in the higher nodes of the tree [58]. If an edge cut the tree close to the root, it will cause a high cost. Minimizing this cost provides an optimal tree for the associated graph. If the cost of a tree is lower than the cost of other possible trees for the same data, this tree is more preferable.

In order to adapt this cost to morphological hierarchies, edge between connected nodes should be found. In equation 5.9, edge weight information is calculated according to the graph of the tree. Gray-scale images can be represented as graphs which can be weighted either with vertices or with edges [124]. Then, component trees can be built from these weighted graphs [114]. Edge weights can be calculated as the absolute difference of intensities of the image pixels [113]. We

have used the level ($\lambda$) difference between leaf pairs as edge weight. Connected pairs of leaves have been determined according to the list of neighbors of each location as in equation 4.4. Although space-time trees are built with different neighbour sets (equations 4.5, 4.6), their projected versions lie in the spatial domain and they come with a neighbour set definition such as 4.4. When the trees are compared from the same image, their graph and edge weights are the same but the distance from their leaves to the *lca* node can be different.

Let $C_\lambda$ be a leaf node and $C'_{\lambda'}$ a ghost node. Their *lca* node can be $C_{\lambda'}$. In that case, the cost for this pair of leaves is calculated as $(\lambda' - \lambda) \cdot \mid C_{\lambda'} \mid$. This calculation is repeated for each pair of leaves. The amount of leaves is equal to the amount of nodes including the ghost nodes.

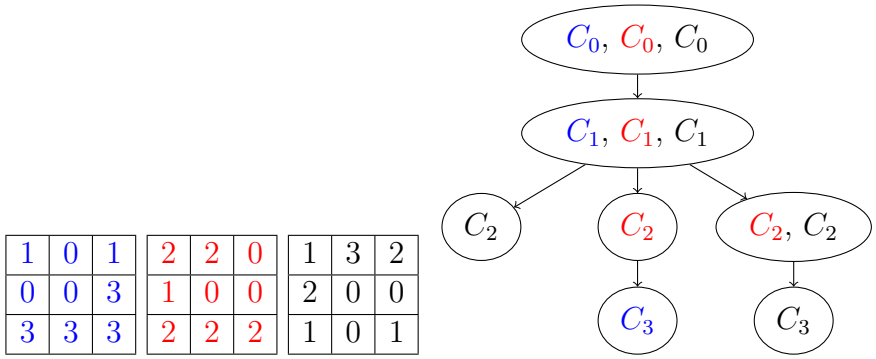
## 5.3 Synthetic dataset experiments

In order to compare tree representation strategies, we first consider synthetic images. Figure 5.1 illustrates our synthetic image experiments. We describe every image and its corresponding values in the tree with different colors. The structure of projected trees considering the blue and black time stamps are different from their single frame trees. For the third time stamp, there is only one $C_1$ node for $\mathcal{T}^3$ and it causes a structural difference between the image tree $\mathcal{T}_3$ and the temporal projection $\mathcal{T}^3$ of the space-time tree. For instance, leaves of $\mathcal{T}_3$ and $\mathcal{T}^3$ can be listed as:

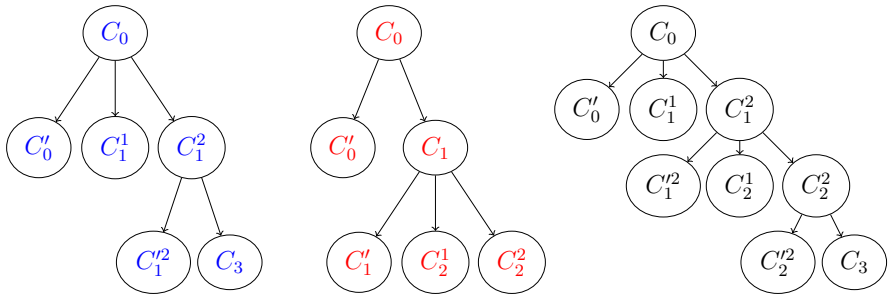
$$leaves(\mathcal{T}_3) = C'_0, C^1_1, C'^2_1, C^1_2, C'^2_2, C_3 \tag{5.10}$$

$$leaves(\mathcal{T}^3) = C'_0, C'_1, C^1_2, C'^2_2, C_3 \tag{5.11}$$

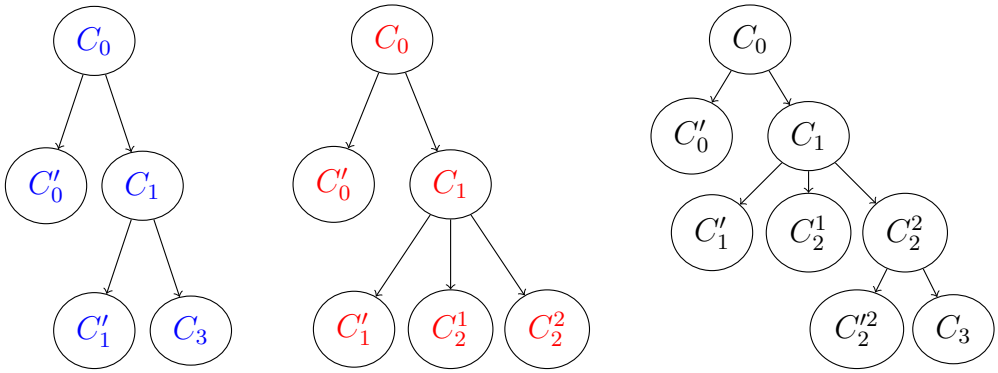
The cost between $C'_0$ and $C_3$ nodes can be calculated for both trees similarly. Their weight difference in both trees is $3 - 0 = 3$ and their *lca* node is same: $C_0$. The only difference between the trees is the number of leaves connected to their *lca* node. Therefore, the cost between $C'_0$ and $C_3$ is $4 \times 3 = 12$ in $\mathcal{T}_3$ and $3 \times 3 = 9$ in $\mathcal{T}^3$.



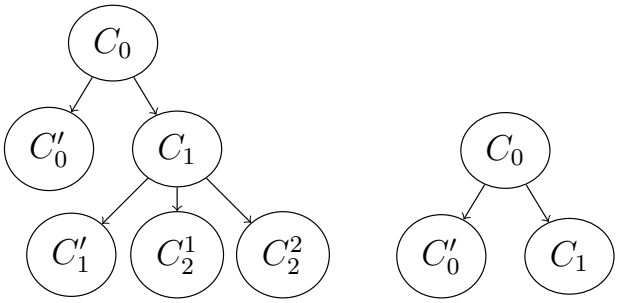

(a) Three synthetic images (b) Space-time tree ($\mathcal{T}$)




(c) Trees for each image ($\mathcal{T}_1$,$\mathcal{T}_2$,$\mathcal{T}_3$)




(d) Temporal projection of space-time tree ($\mathcal{T}^1$,$\mathcal{T}^2$,$\mathcal{T}^3$)




(e) Tree with spatial hierarchy and spatial projection of space-time tree ($\mathcal{T}_x$, $\mathcal{T}^x$)

Figure 5.1 – Illustration of trees built using the different strategies and the projected trees on a synthetic dataset.

We also show $\mathcal{T}_\varsigma$ and $\mathcal{T}^\varsigma$ trees. In order to obtain $\mathcal{T}_\varsigma$, we have considered the mean of each pixel time series. Projected trees for both temporal and spatial domains have a lower or equal amount of nodes. Less leaves pair provide less complexity for the corresponding tree. Since $\mathcal{T}_2$ and $\mathcal{T}^2$ are identical, their properties are the same. While the costs of $\mathcal{T}_1$ and $\mathcal{T}^1$ are 108 and 80 respectively, the costs of $\mathcal{T}_3$ and $\mathcal{T}^3$ are 142 and 122 respectively. As a result, according to the cost and amount of nodes comparisons, the projected trees behave better than the other trees for our synthetic image experiment.

## 5.4 Real dataset experiments

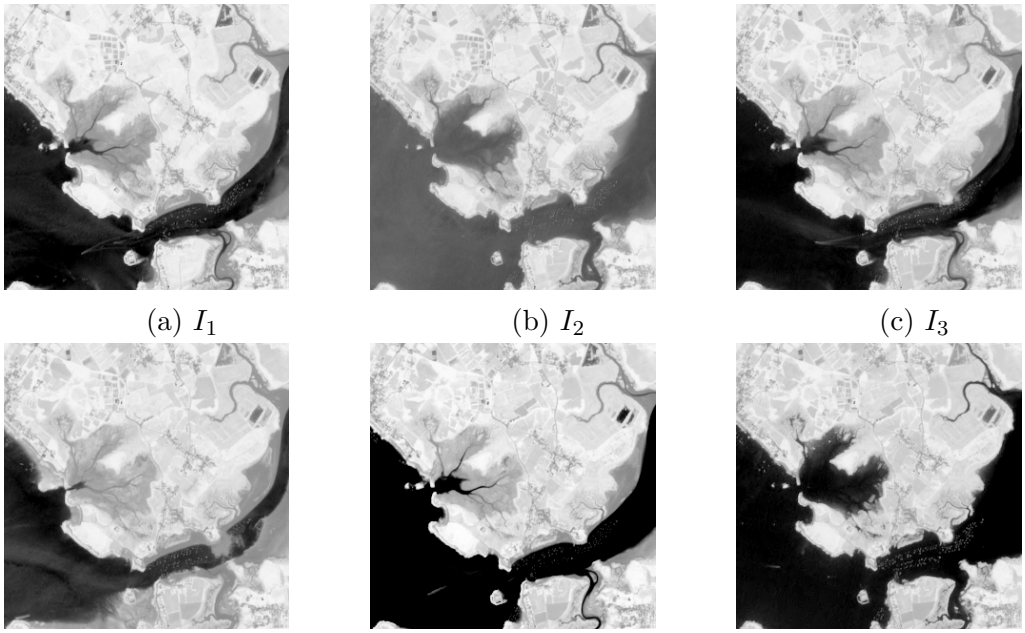
In addition to the experiments conducted with a synthetic dataset, we have also considered a real use case with the analysis of SITS from Brittany, France. The multispectral images were acquired with Sentinel-2 Copernicus mission, and were converted as gray-scale ones by computing the Normalized Difference Vegetation Index (NDVI). As illustrated in figure 5.2, we used a sample SITS made of small extracts ($300 \times 300$) from 6 observations which were acquired in 2018 from Morbihan with a spatial resolution of 10m. Alternatively, we used 50cm panchromatic Pleiades images extracted from Kalideos [1] ($300 \times 300$) located at Pontorson, also in Brittany, France.

The Pleiades time series is also composed of six images acquired between May 2018 and April 2019. Pleiades offers a very high spatial resolution and it is used to observe small details of Earth. Therefore, we use the Pontorson dataset for qualitative assessment of the filtering experiments because of their visual superiority, and rely on the Sentinel-2 data for LAF and cost experiments.

### 5.4.1 Number of node analysis

First, we measured the number of nodes for each $\mathcal{T}_t$ and their corresponding temporal projection of $\mathcal{T}_\tau, i.e. \mathcal{T}^t$. We reported the values in Table 5.3. We have considered both 26-connectivity and continuous connectivity, and denote the corresponding trees by $\mathcal{T}^t$ and $\mathcal{T}'^t$, respectively. We report the results for both max

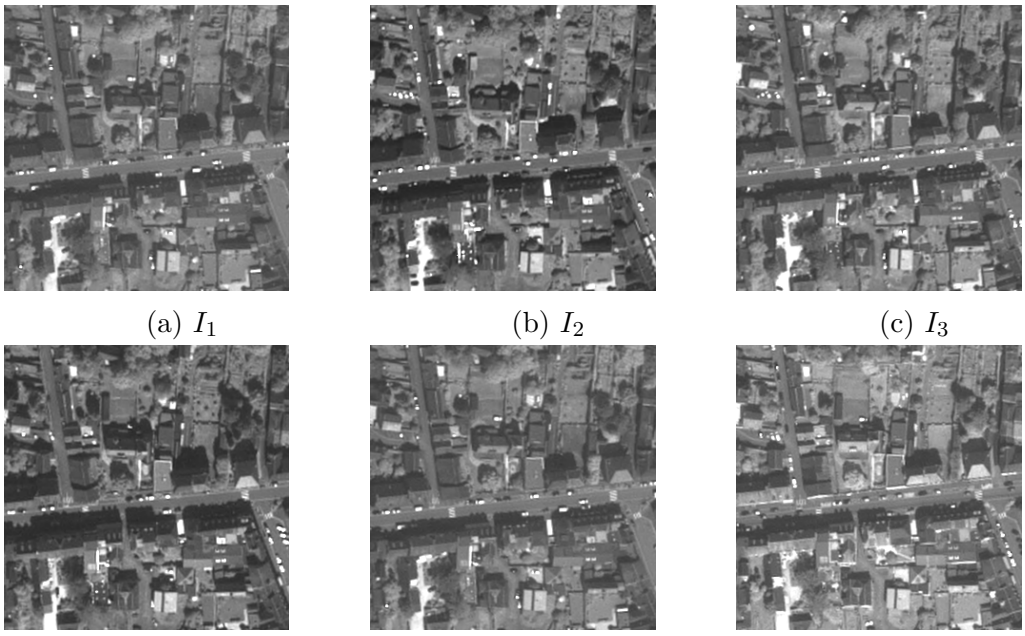
1. https://bretagne.kalideos.fr



(a) $I_1$ (b) $I_2$ (c) $I_3$

(d) $I_4$ (e) $I_5$ (f) $I_6$

Figure 5.2 – Sentinel-2 satellite image time series



(a) $I_1$ (b) $I_2$ (c) $I_3$

(d) $I_4$ (e) $I_5$ (f) $I_6$

Figure 5.3 – Pleiades satellite image time series

and min-tree separately. The last three rows provide the standard deviation, the average and the total of number of nodes. The numbers of nodes for TH strategy (column 1) are more stable in comparison to the projected trees (column 2 and 3). Indeed, the number of nodes varies for both projected trees. When the number of nodes increases for max-tree, it generally decreases with min-tree for both projected trees.

| | $\mathcal{T}_t$ | | $\mathcal{T}^t$ | | $\mathcal{T}'^t$ | |
|---|---|---|---|---|---|---|
| | Max | Min | Max | Min | Max | Min |
| $t=1$ | 13640 | 14274 | 6377 | 5874 | 4917 | 3481 |
| $t=2$ | 13577 | 14231 | 4471 | 4523 | 3688 | 4007 |
| $t=3$ | 14268 | 14002 | 2418 | 5726 | 2099 | 2924 |
| $t=4$ | 13883 | 14178 | 5111 | 3067 | 3469 | 2626 |
| $t=5$ | 12495 | 11592 | 6726 | 2966 | 6178 | 2862 |
| $t=6$ | 15176 | 13943 | 4106 | 6789 | 1614 | 5838 |
| avg. | 13839 | 13703 | 4818 | 4824 | 3631 | 3623 |
| std. | 804 | 951 | 1445 | 1438 | 1558 | 1090 |
| total | 83039 | 82220 | 28909 | 28945 | 21789 | 21738 |

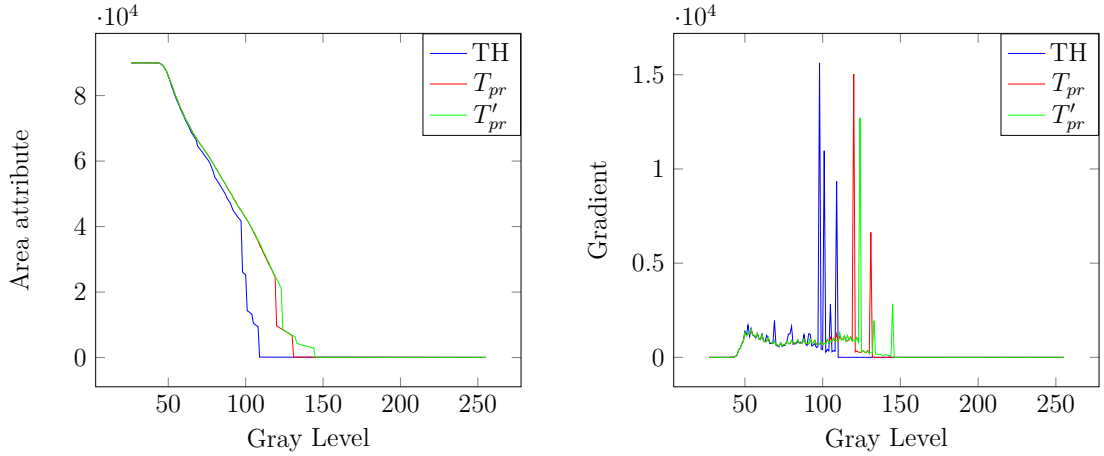
Table 5.3 – Amount of nodes with the S-2 SITS dataset.

All projected trees have less nodes than the corresponding single frame tree built from the same time stamp. As a result, the space-time tree provides a less complex structure. Besides, $\mathcal{T}'^t$ has less nodes than $\mathcal{T}^t$ consistently.

Figure 5.4 illustrates the evolution of the area attribute for leaves of $\mathcal{T}_4$, $\mathcal{T}^4$ and $\mathcal{T}'^4$. The maximum area is obtained for the root node and corresponds to the image size ($300 \times 300 = 90,000$). The gradient curve exhibits abrupt changes in the node attribute. These curves for $\mathcal{T}^4$ and $\mathcal{T}'^4$ generally overlap. The abrupt change for $\mathcal{T}_4$ appears at a low level which may cause information loss if a filtering or any other process is applied at that level.

### 5.4.2 Cost Analysis

In order to evaluate the usability of the cost function for morphological hierarchies, we added three levels of Gaussian noise to a single frame image. We then evaluated the cost function of the single frame image and of the three noisy image




(a) Node Signature (b) Gradient Curve

Figure 5.4 – Area attribute signatures of same location with TH, projected and their gradient curve on the Morbihan dataset.

versions. Then, we observed that the cost was increasing w.r.t the noise level. We have used $I_4$ image from Morbihan dataset for this experiment and report the normalized cost values in Table 5.4.

| Image | Cost |
|---|---|
| $I_4$ | **1.00** |
| $I_4$ + noise | 3.54 |
| $I_4$ + (2 × noise) | 5.06 |
| $I_4$ + (3 × noise) | 6.36 |

Table 5.4 – Normalized cost values for $T_4$ and trees with noisy images.

Table 5.5 reports the cost values for the temporal projection and temporal hierarchy. We normalized the cost values according to the lowest values which are $\mathcal{T}'^3$ for max-tree and $\mathcal{T}'^5$ for min-tree.

Since Morbihan is affected by tide, the amount and value of pixels on the sea is changing. While $I_1$, $I_3$, $I_5$ and $I_6$ have similar pixel values on the sea, $I_2$, $I_4$ are different from them because of the bathymetry. The cost value of $T'^3$ min-tree is much lower than for $T^3$, since the continuous connectivity is providing connection

| | $\mathcal{T}_t$ | | $\mathcal{T}^t$ | | $\mathcal{T}'^t$ | |
|---|---|---|---|---|---|---|
| | Max | Min | Max | Min | Max | Min |
| $t=1$ | 57.2 | 93.5 | 23.8 | 10.4 | 6.4 | 5.8 |
| $t=2$ | 41.7 | 82.2 | 4.4 | 12.8 | 3.2 | 9.9 |
| $t=3$ | 54.7 | 86.5 | 2.1 | 10.9 | **1.00** | 4.4 |
| $t=4$ | 58.6 | 86.0 | 12.8 | 4.5 | 5.2 | 2.42 |
| $t=5$ | 66.9 | 70.2 | 25.6 | 1.2 | 23.1 | **1.00** |
| $t=6$ | 70.6 | 98.4 | 7.2 | 20.6 | 1.3 | 14.1 |

Table 5.5 – Cost values for Temporal Hierarchy and Temporal Projection.

between $I_1$ and $I_5$. Similarly, the projected max-tree $T'^4$ has a much lower cost than $T^4$ because brighter pixels on the sea are connected between $I_2$ and $I_4$ in $T'^4$.

| | Max | Min |
|---|---|---|
| $\mathcal{T}_\varsigma$ | 500.00 | 1000.00 |
| $\mathcal{T}^\varsigma$ | 15.00 | 3.00 |
| $\mathcal{T}'^\varsigma$ | **1.00** | **1.00** |

Table 5.6 – Cost values for Spatial Hierarchy and Spatial Projection.

In order to compare spatial projection, Table 5.6 shows the cost values for $\mathcal{T}_\varsigma$, $\mathcal{T}^\varsigma$ and $\mathcal{T}'^\varsigma$ as in Table 5.5. We used the distance based ordering with Euclidean distance. According to the Dasgupta's cost, the space-time tree is preferable to both TH and SH strategies. In other words, one should prefer space-time tree in order to have a tree with less cost.

### 5.4.3 Filtering

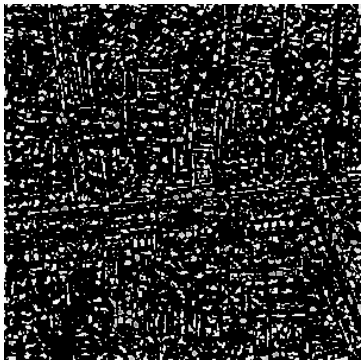
As we explained in section 2.4.1, the filtering operation is very common to process tree representations and we applied it to compare single frame tree and projected trees. Figure 5.5 illustrates our filtering experiments with $I_4$ of Pontorson dataset. We have used all 6 images but we illustrate only one of them that we selected for filtering. We have filtered trees using the area attribute and the same threshold ($h = 20$) for the sake of comparison. The second row of the figure shows the filtered images and its third row shows their difference with the original image.

(a) $I_4$

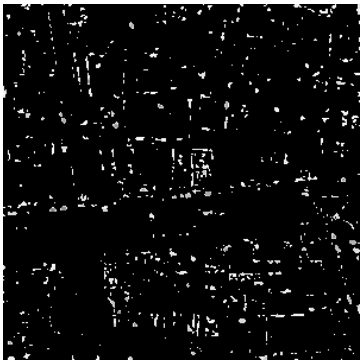
(b) $\zeta^{20}(\mathcal{T}_t)$

(c) $\zeta^{20}(\mathcal{T}^t)$

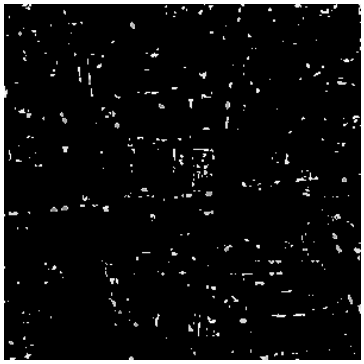
(d) $\zeta^{20}(\mathcal{T}'^t)$



(e) $I - (b)$



(f) $I - (c)$



(g) $I - (d)$

Figure 5.5 – Filtering results of one image of the Pleiades time series with $\mathcal{T}_t$ and projected trees, along with their residues that highlight changes by using $\mathcal{A}_{area}$ and $h = 20$.

Filtering removes the small objects such as cars or noises of the images. The amount of changed pixels for $T_t$, $T^t$ and $T'^t$ is 13,891, 3682 and 2243 respectively. Changes are less for $\mathcal{T}'^t$ compared to $\mathcal{T}^t$. Much changes are not desired with

such a small threshold. Indeed, filtering with the TH strategy changes many pixels compared to the projected tree. These results show that the compactness of space-time tree leads to an efficient filtering process.

# Chapter 6

# Feature extraction from Space-Time tree

## Contents



In the previous chapter, we have reported the superior capability of space-time tree over the other hierarchies. This chapter thus focuses on the use of space-time tree to extract spatio-temporal features. While we presented well-

established spatial based attributes in section 2.3, we propose here new spatio-temporal attributes. We will then report how to use them in SITS filtering and pattern spectra tools.

## 6.1 Time related attributes

Although numerous spatial based attributes have been introduced in the literature as discussed in chapter 2, no time based attribute has been proposed yet (to the best of our knowledge). In this section, we will use the spatio-temporal information to introduce the following novel attributes:

1. The beginning time of a node;
2. The ending time of a node;
3. Duration;
4. Time of the maximum value in a node;
5. Time of the minimum value in a node;
6. Difference between maximum and minimum value of a node (amplitude);
7. Node centroid;
8. Spatio-temporal stability;

While we give the definitions of these attributes, we will also precise whether they are increasing or non-increasing.

### 6.1.1 The beginning and the ending times of a node

Nodes of a space-time tree may not include pixels from all time stamps. A spatio-temporal connected component may occur in one or few consecutive time stamps. The beginning and the ending times of the node can then be measured as:

$$\mathcal{A}_{beg}(C_{\lambda,\tau}) = \min\{t \mid t \in \tau,\ C_{\lambda,t} \neq \emptyset\} \tag{6.1}$$

$$\mathcal{A}_{end}(C_{\lambda,\tau}) = \max\{t \mid t \in \tau,\ C_{\lambda,t} \neq \emptyset\} \tag{6.2}$$

These attributes take integer values between 1 and $n$. Besides, ancestors of a node can have different beginning or ending times. Since the spatio-temporal support of a node is included in the support of its ancestor, one ancestor node has an earlier or equal beginning time (resp. later or equal ending time) with its child node. Therefore, both attributes are increasing attributes. In the literature, the beginning and ending times of the crop season have been proposed for time series analysis in agricultural areas [48]. However, computing them on the space-time tree allows us to extend these features from a pixel-based to an object-based analysis.

### 6.1.2 Duration

The duration of a node can thus be expressed simply as the difference between its beginning and ending times:

$$\mathcal{A}_{dur}(C_{\lambda,\tau}) = \mathcal{A}_{end}(C_{\lambda,\tau}) - \mathcal{A}_{beg}(C_{\lambda,\tau}) \tag{6.3}$$

Correspondingly, it can be seen as the temporal length of a node. Since attributes involved in the equation are increasing, equivalently their difference is also an increasing attribute. The highest value of the duration attribute is equal to the length of the time series. While it can be possible to find objects with stable intensities such as buildings with high duration value, it can also be used to identify object with unstable intensities such as crops with a lower duration value. Outliers, noises and mobile objects such as cars are expected to have a low duration attribute because they are not stable through time.

### 6.1.3 Time of the maximum and minimum values in the object

Times of maximum and minimum values of the pixels in a node can be defined as follows:

$$\mathcal{A}_{maxt}(C_{\lambda,\tau}) = \arg_{t\in\tau} \max_{x\in C_{\lambda,t}} (I_t(x)) \tag{6.4}$$

$$\mathcal{A}_{mint}(C_{\lambda,\tau}) = \arg_{t\in\tau} \min_{x\in C_{\lambda,\tau}} (I_t(x)) \tag{6.5}$$

For instance, seasonal activities of crops vary and their local extrema points can serve as discriminative feature. These attributes are non-increasing.

### 6.1.4 Amplitude

The amplitude attribute is the difference between the maximum and the minimum values of the pixels in a node:

$$\mathcal{A}_{amp}(C_{\lambda,\tau}) = \max(C_{\lambda,\tau}) - \min(C_{\lambda,\tau}) \tag{6.6}$$

This difference is also called in the literature as height attribute of the tree nodes as explained in section 2.3. Here, we use this attribute with the space-time tree to bring some temporal based information. This attribute is increasing because the range of pixel intensities in a node is increasing from leaves to root for all kind of trees. The root node has a maximum amplitude which is equal to the difference between the maximum and minimum intensities of the entire image set. Minimum amplitude occurs in the leaves and is equal to 0.

### 6.1.5 Node centroid

As mentioned before, a spatio-temporal node consists of pixels from different time stamps which can be spread in time non-uniformly. The node centroid attribute aims at defining the center of gravity of a node in time. The node centroid can be found by weighting the time stamps by the number of pixels at relevant time:

$$\mathcal{A}_{cent}(C_{\lambda,\tau}) = \frac{\sum_{t=1}^{n} \mathcal{A}_{area}(C_{\lambda,t}) \times t}{\mathcal{A}_{area}(C_{\lambda,\tau})}, \tag{6.7}$$

where $n$ represents the length of the data. The node centroid of the root node is equal to $\frac{n+1}{2}$. If a node has pixels from only one time stamp, its centroid is equal to that time stamp. This attribute is non-increasing

### 6.1.6 Spatio-temporal stability

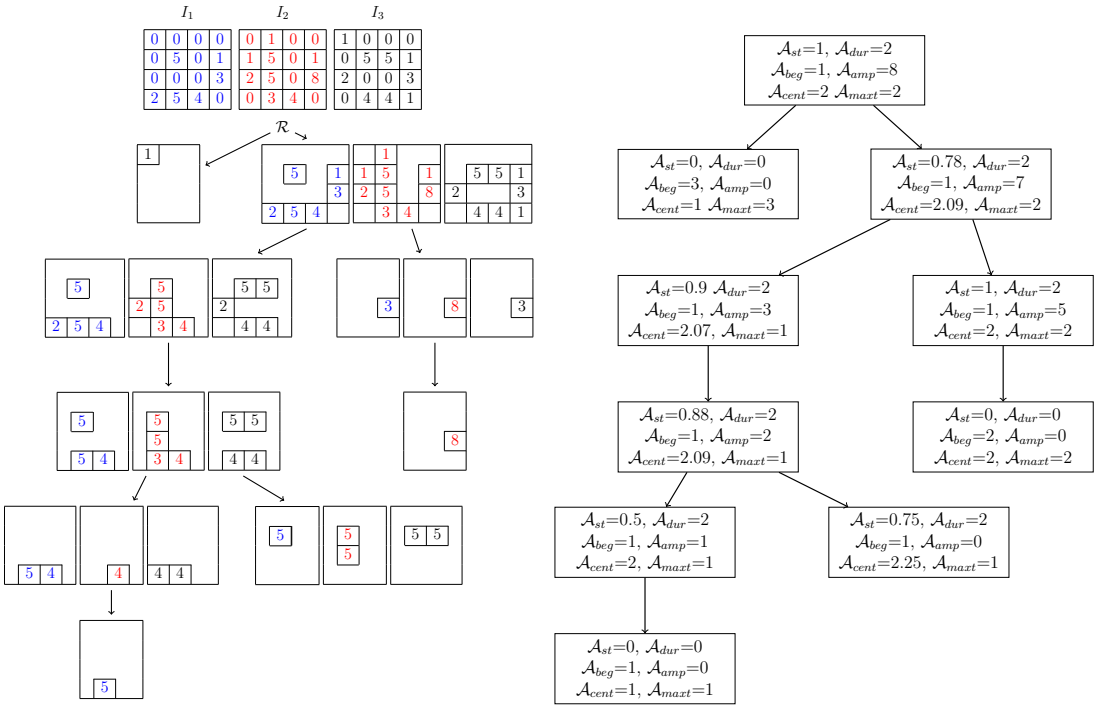
The spatio-temporal stability attribute relies on the number of pixels in the spatial slices as presented in equation 6.8. We define the spatio-temporal stability $\mathcal{A}_{st}$ of each node as the sum of the area ratio of the successive connected component slices through time:

$$\mathcal{A}_{st}(C_{\lambda,\tau}) = \frac{1}{n-1}\sum_{t=1}^{n-1}\frac{\min(\mathcal{A}_{area}(C_{\lambda,t}), \mathcal{A}_{area}(C_{\lambda,t+1}))}{\max(\mathcal{A}_{area}(C_{\lambda,t}), \mathcal{A}_{area}(C_{\lambda,t+1}))} \tag{6.8}$$

The root node of a space-time tree covers the whole spatial support all along the time series and thus has a spatio-temporal stability of 1 independently of the length $n$ of the series. The minimum value (0) is observed for the nodes whose pixels are in a single time stamp. The stability measure can be used to identify evolving or non-evolving objects in SITS. Nodes having low spatio-temporal stability attributes will include unstable objects such as sea when tide occurs. Although the area is an increasing attribute, spatio-temporal stability remains a non-increasing attribute.

### 6.1.7 Summary of proposed attributes

Some of the proposed attributes (beginning, ending, duration, amplitude) are similar to features in [48], where the authors have proposed definitions as pixel based features. One of the novelty of our approach is to extent these attributes into a tree structure; therefore, multiscale analysis such as filtering or pattern spectra becomes possible. Another advantage is that we calculate these attributes at the object scale instead of the pixel level. In addition to these spatio-temporal attributes, conventional ones presented in section 2.3 such as area, mean gray value or volume can also be used with the space-time tree. Some shape-based attributes such as eccentricity can not be easily extended along the time dimension. Duration, beginning and ending time attributes can also be computed directly from the date of acquisitions when analysing SITS. In that case, the highest duration value is not the length of the SITS, but the elapsed time between the first acquisition and the last one.




(a) Space-time tree example (b) Space-time attributes

Figure 6.1 – Space-time max-tree and space-time attributes, measured on its nodes.

Figure 6.1 shows an example for some of the proposed tree attributes. First, we show three small synthetic images and their space-time max-tree with one specific color per image. The space-time tree has 10 nodes. The tree depicted by Figure 6.1b shows the spatio-temporal attribute values for each node in the tree.

## 6.2 Pixel-wise feature extraction

Spatio-temporal attributes proposed in the previous section can be used to extract features from SITS. Among the possible feature extraction schemes recalled in section 2.4.1, we focus here on pixel-wise features which can be computed for each pixel location.

### 6.2.1 Filtering

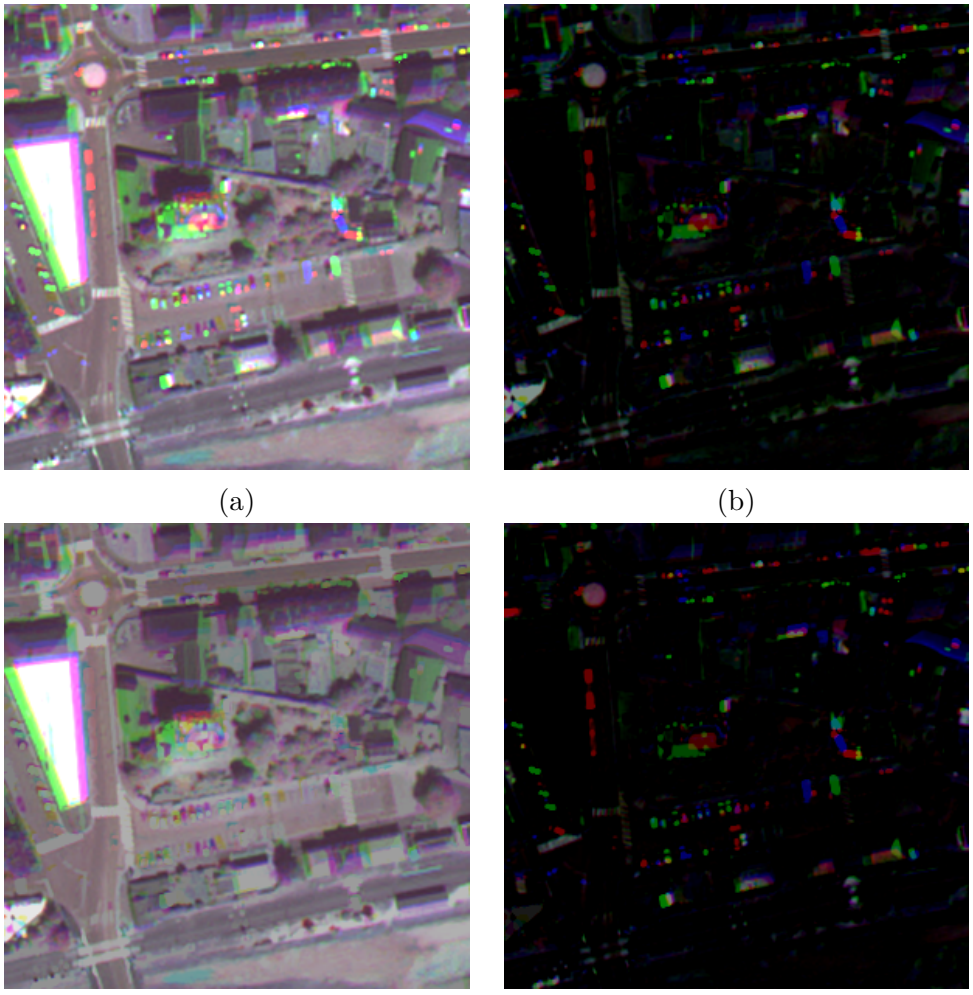
Filtering a tree built from a single frame image was recalled in section 2.4.1. Here, we will explain how spatio-temporal attributes can be used to filter a space-time tree.

We conducted a filtering experiment with three panchromatic Pleiades images acquired over the same scene at different dates. As a baseline, we built a tree for each date and filtered them, thus without considering the temporal information. We also stacked these images and built a space-time max-tree corresponding to the STH strategy. We used here the area attribute which is defined as the number of pixels in a node. We use a unique threshold (1000) for each filtering to ensure a fair comparison. As mentioned in section 2.2.1, max-tree leaves include regional extrema of the image. Therefore, max-tree filtering allows manipulating bright objects in the image. To emphasize the filtered objects, we show in Figure 6.2 the difference between original and filtered images (i.e. the residue provided by the connected top-hat operator). We can see that cars, which are located at different places for different dates have been removed by the filtering operation. However, filtering with a single frame cannot be used to filter out only cars since it removes other objects such as the building located at the bottom left of the scene. Let us recall that this example is provided only for the sake of illustration and proper detection of dynamic objects will be addressed in section 7.2.3.

### 6.2.2 Extension of morphological attribute profiles to SITS

Morphological attribute profiles can be easily adapted to SITS with the tree building strategies presented in chapter 4. Attributes proposed in the previous section can be used to produce these profiles, leading to new feature for SITS.

Attribute profiles (AP) require careful selection of a set of thresholds. Although there exist some solutions to select thresholds such as in equation 2.38, it is still an open topic for AP based methods. This of course also applies when deriving AP from SITS. For instance, the number of pixels in a spatial slice at any time stamp of the spatio-temporal node will be greater than the number of pixels in a node of the tree which is built for that time stamp. Therefore, when area filtering is applied with a space-time tree, higher thresholds should be preferred. However,



(a) (b)

(c) (d)

Figure 6.2 – SITS filtering example. Each triplet of dates is represented by a color image (one channel per date). (a): composite of original series (b): Difference between (a) and its respective spatial-only filtered image (c): Reconstructed STH filtering (d): Difference between (a) and (c).

these thresholds had an upper limit which is the root node of a tree. More precisely, the number of pixels in a root node of a space-time tree will be $n-1$ times greater than the number of pixels in a single frame image. AP for SITS can formalized as:

$$AP(I_t) = (I_t^{\phi^{h_m}}, I_t^{\phi^{h_{m-1}}}, ..., I_t, ..., I_t^{\gamma^{h_{m-1}}}, I_t^{\gamma^{h_m}}) \tag{6.9}$$

Figure 6.3 shows how AP is produced for a spatio-temporal dataset. Each threshold $h$ provides a different set of filtered images. Then, these filtered SITS sets are concatenated to produce AP from the spatio-temporal data. Length of the AP is equal to $n + m \times n$.

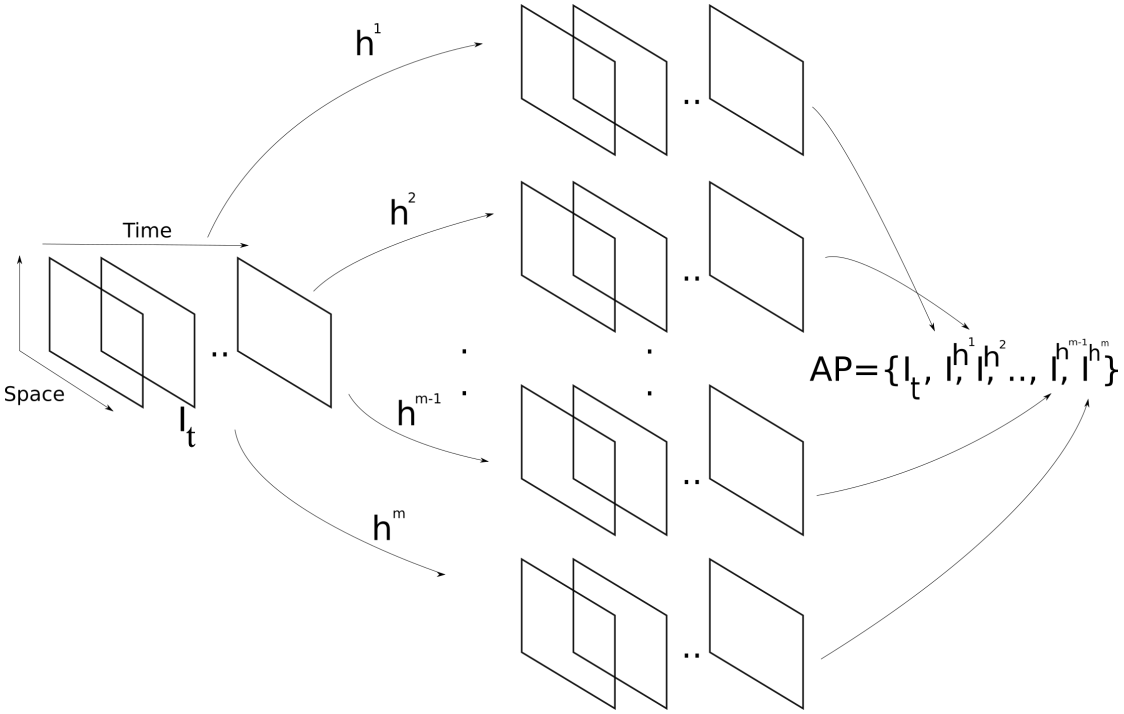



Figure 6.3 – AP with a spatio-temporal data

## 6.3 Image-tile feature extraction

Extension of PS to 2D PS was explained in section 2.4.2, providing 2D image-tile descriptors. This extension can be applied in more dimensions enabling us to take additional attributes into account. This multi-dimensional PS will be used for SITS analysis with 2 operations; interactive PS and PS filtering.

### 6.3.1 Interactive pattern spectra analysis

Interactive PS relies on selecting appropriate bin when analyzing images. Figure 6.4 shows the workflow of interactive pattern spectra analysis with spatio-temporal data. After producing 2D PS with selected attributes and bin size, the second step is to select a bin (or bins) from the PS. It allows us to find the nodes which contributed to this bin. The tree can then be pruned according to the selected nodes and their descendants. Finally, a new image set is reconstructed from this pruned tree. As a related work, interactive pattern spectra analysis was used in [116] in order to collect training samples with specific attributes for classification. However, our target is to detect patterns from the images without involving any machine learning technique.

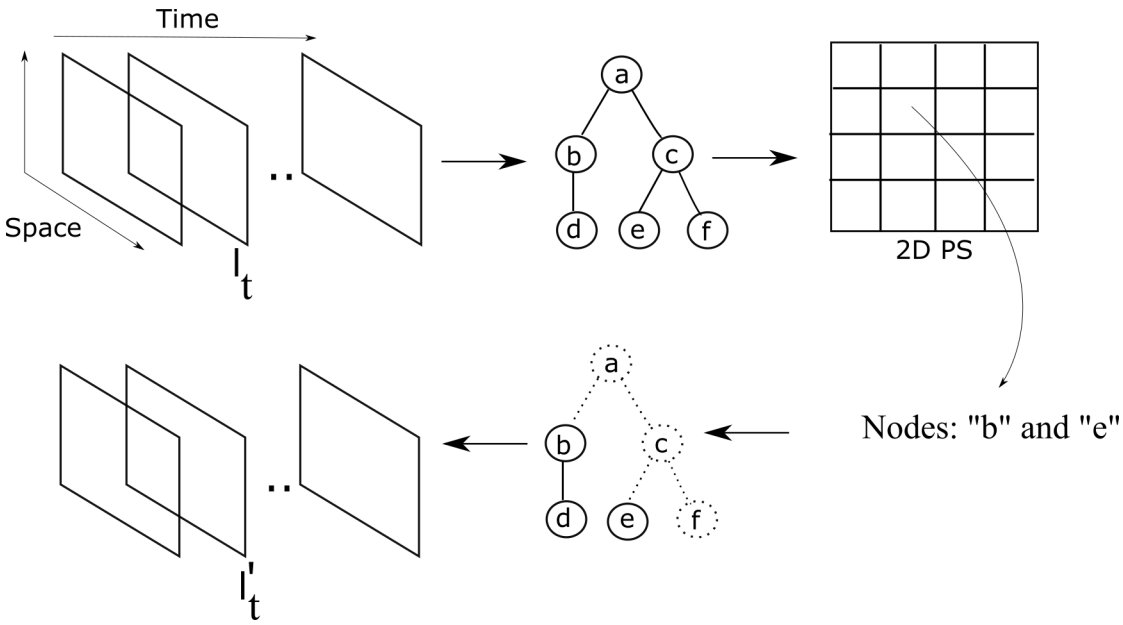
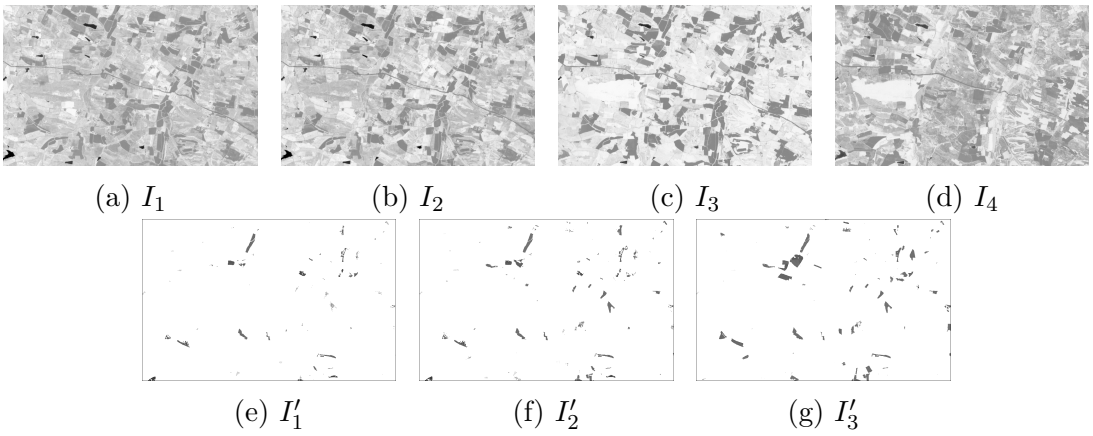



Figure 6.4 – PS with spatio-temporal data.

Here we illustrate the interactive PS analysis on space-time tree with an example constructing the PS with the ending time attribute. Experiments are conducted with 4 NDVI images and PS is produced with area, rectangularity and ending attribute. We produced 3D PS of size $100 \times 100 \times 4$ bins for these attributes. If we are interested in small rectangular objects connected until third time stamp, we can select a bin at the location $[1, 27, 3]$. Figure 6.5 shows the four input images

and the set of reconstructed images from the selected bin. This set contains only 3 images because we chose ending time equal to 3, and this selected bin has thus no information from $I_4$. Reconstructed images contain a crop with similar behaviour through time.



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) $I_4$

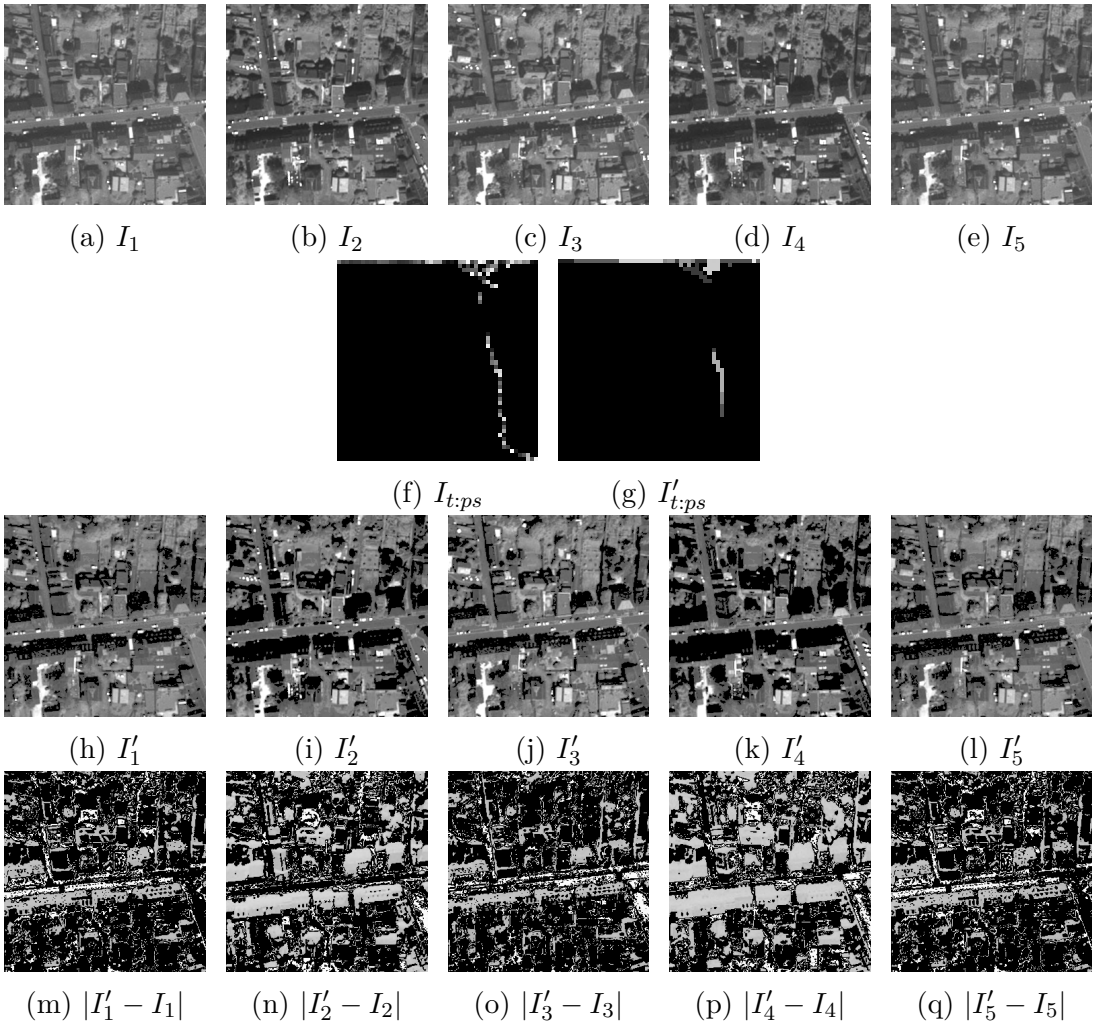
(e) $I'_1$ (f) $I'_2$ (g) $I'_3$

Figure 6.5 – Interactive pattern spectra example

### 6.3.2 PS filtering

Here we propose a filtering method by using PS and illustrate it through a proof of concept experiment. As we explained in section 2.4.1, tree filtering processes an attribute and it is difficult when the attribute is non-increasing. Moreover, it needs rebuilding the tree when we are interested in several attributes. Our aim is to filter an image set with several attributes (whether non-increasing or increasing) simultaneously by using PS. We will benefit from the connectivity of filled bins and discard alone or small group of bins which are not consistent with the other bins on the PS. To do so, we will build a tree on the PS which is considered as a tensor (2D or 3D) and then filter this tree with the area attribute. Finally, we reconstructed image or images from the filtered PS which has group of connected bins. In other words, we remove sole or connected small group of bins from the PS. As a similar concept, shape-space filtering was proposed in [186] and generalized in [22]. Their method relies on building a new tree from a traditional tree in order to filter the second tree to overcome the non-increasing attribute filtering issue.

Figure 6.6 shows a PS filtering example. The top line contains 5 Pleiades images. We produce PS with "area" and "spatio-temporal stability" attributes for these images. Then we built the max-tree from PS and filter the tree with the area attribute and $h = 10$. Figures 6.6f and 6.6g show 2D pattern spectra and its filtering result respectively. Last, we reconstructed the image set from filtered PS as they can be seen from Figures 6.6h to 6.6l. Figures 6.6h to 6.6l illustrate the difference between the input and the reconstructed images. In this example, the objects inside shadow regions become more visible in the filtered images. This method could serve as an efficient unsupervised feature extraction method from single frame or multi frame images.



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) $I_4$ (e) $I_5$

(f) $I_{t:ps}$ (g) $I'_{t:ps}$

(h) $I'_1$ (i) $I'_2$ (j) $I'_3$ (k) $I'_4$ (l) $I'_5$

(m) $|I'_1 - I_1|$ (n) $|I'_2 - I_2|$ (o) $|I'_3 - I_3|$ (p) $|I'_4 - I_4|$ (q) $|I'_5 - I_5|$

Figure 6.6 – Example for PS space filtering.

# Chapter 7

# Applications

## Contents



This chapter corresponds to our published conference papers *Attribute profiles for satellite image time series* [167], *Monitoring urban growth with spatial filtering of satellite image time series* [164] and *Analysis of min-trees over Sentinel-1 time series for flood detection* [166]. We provide qualitative and quantitative results of our experiments using spatio-temporal attributes and tools presented in chapter 6. We show how to use space-time tree for SITS based applications such as land cover mapping, change detection and spatio-temporal pattern recognition. We will propose novel methods for each application and compare with representative methods from the state-of-the-art.

## 7.1 Land cover mapping

Land cover indicates the physical type of earth surface and land cover mapping refers to classification of each pixel according to these physical types. The most successful approaches to automatically produce land cover maps from SITS are based on supervised learning. With these approaches, a model is trained based on a feature representation extracted from SITS data and prior knowledge (known as targets or labels). Therefore, to create a land cover map with a supervised classification approach, some ground truth data is needed. Here, our contribution is to extract some new feature representation from the image set in order to produce accurate land cover maps. Classification experiments have been conducted according to their methods: using attribute profiles, using feature profiles with a proper test/train split and attribute profiles with the spatio-temporal attributes.

### 7.1.1 A first experiment using attribute profiles

In our first experiment, we aim to compare tree building strategies (SH, TH, STH from chapter 4) for their classification application ability. We consider the French Land Cover Map provided by [74] [1] as a ground truth in our first land cover mapping experiments. Figure 7.1 displays at the below position the ground truth image that covers our sample SITS. The latter is made of 6 cloud-free images acquired by Sentinel-2 over the Pontorson city (west of France) during 2017 within one month interval approximately (shown chronologically from left to right in Figure 7.1). The image size is 300 $\times$ 300 pixels and the spatial resolution is 10m. As already stated, we focus on gray scale images in this thesis so only one band (band 8) was kept from the images. The region under study includes 5 land cover classes: summer crops, winter crops, urban areas forest and grasslands.

In order to extract features, we will produce AP and focus on two attributes: area and moment of inertia. While the former is increasing, the latter is not, and we consider here only the direct-rule for the sake of simplicity. The 10-connectivity was used to build the spatio-temporal tree. It is important to note that the length

1. https://www.theia-land.fr/en/product/land-cover-map/

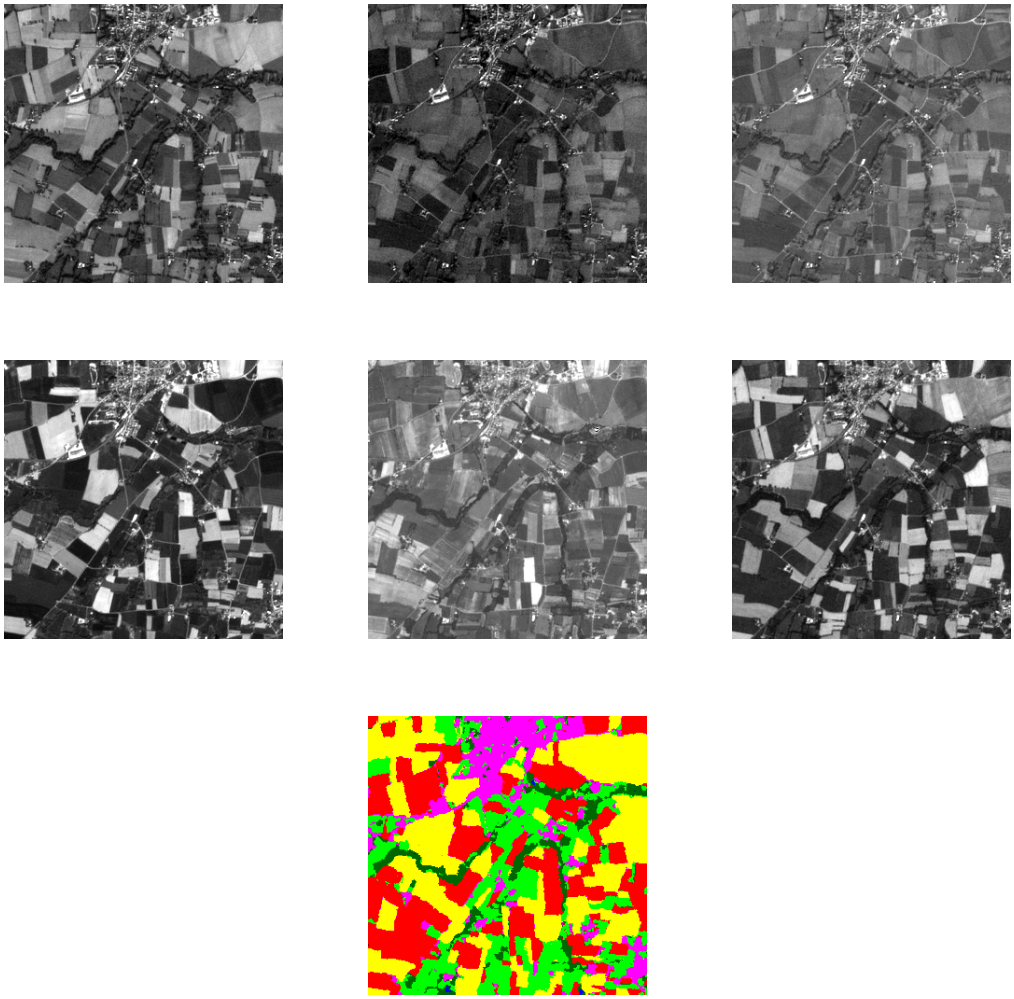


Figure 7.1 – SITS of the Pontorson dataset and colored ground truth. (yellow: summer crops, red: winter crops, pink: urban areas, dark green: forest and light green: grasslands)

of the feature vectors depends on the size of the input data. While the TH and STH strategies lead to 42 features per pixel, the SH provides only 13 features for each location.

For classification purpose, we randomly selected 100 pixels from each class for training. The remaining pixels were considered for testing. Let us note that this experimental setup, while being commonly used in the literature, does not ensure a proper split between training and testing features. The following experiments will address this issue. The thresholds related to the moment of inertia have been

defined according to the literature as $\lambda = (0.2, 0.3, 0.4)$. Three area thresholds were randomly selected for every approach. As far as classification methods are concerned, we consider two popular classifiers, namely Support Vector Machines (SVM) and Random Forests (RF), and set the number of trees in RF [24] to 100 and opt for the "linear kernel" in SVM.

We report in Table 7.1 the overall accuracy. RF performs generally better than SVM. The TH strategy gives a better result, but at a higher cost in terms of feature length. The preliminary results show that AP increases the classification accuracy except when using the lexicographical ordering approach, illustrating the fact that the theoretical correctness does not always come with a practical usefulness.

| Method | RF-Area | RF-Moment | SVM-Area | SVM-Moment |
|---|---|---|---|---|
| Without Tree | 67 | 67 | 68 | 68 |
| SH-Lex | 59 | 63 | 54 | 59 |
| SH-DTW | 68 | 68 | 59 | 61 |
| SH-Mean | 67 | 68 | 62 | 61 |
| TH | 70 | 68 | **68** | 54 |
| STH | **72** | 63 | 63 | 65 |

Table 7.1 – Comparison of overall classification accuracy (in %) for the Pontorson dataset.

### 7.1.2 Using feature profiles with a proper test/train split

In the first experiment, we have used training and testing samples from the same AP as it is used in the literature. Since we aim to have independent training and testing samples, we split the images for this second land cover mapping experiment. We used a sample SITS made of 17 Sentinel-2 images with size 1000 × 1000 pixels (at 10m/pixel resolution), which were acquired in 2016 over Dordogne, France. We used level 2A products provided by the THEIA land data center [2]. We selected only cloud-free images and calculated NDVI for each image. We used labelled data from the agricultural Land Parcel Information System known as Registre Parcellaire Graphique (RPG). The studied area includes 5 land cover classes: urban, crop, water, forest and moor. We inspired from the feature profiles [131]

2. https://www.theia-land.fr/en/satellite-data/

which is presented in section 2.4.1. We calculated the following node attributes: area, mean gray value, height and volume; then we used these attributes as a feature vector for each location.

Both min and max-tree are used to obtain feature images. These images are then stacked with the original image time series and the whole set is used for classification. In our experiments, we split vertically the image into two equal parts with a size of 1000 × 500 pixels. The first part of the image was used for training and validation, following a 5-fold cross validation procedure. As a classifier, we used the RF method and optimized its number of trees and maximum depth parameters among the sets {100, 200, 300} and {80, 90, 100} respectively. Then, we used the best parameters and evaluated the classification performance with the testing set.

As a baseline, we applied the same RF classifier with its own optimized parameters to the original SITS pixels described by NDVI values. We obtained an overall accuracy of 93.63. Figure 7.2 provides a qualitative result from the test area with the first image of the multitemporal Sentinel-2 image and its associated reference map. Each color represents a specific land cover class. Table 7.2 reports the classification results obtained with different spatial attributes for each tree representation strategy already applied to the Pontorson dataset. Results show that the spatial attributes offer an efficient way to increase the classification accuracy with TH and STH strategy compared to the baseline, i.e. without using spatial attributes. Table 7.3 provides $F_1$ score (i.e., the harmonic mean of precision and recall measures) for each class.

| Methods | Spatial attributes | | | |
|---|---|---|---|---|
| | Area | Mean gray value | Height | Volume |
| SH-Lex. | 56.31 | 57.52 | 57.14 | 57.64 |
| SH-DTW | 78.52 | 75.08 | 77.15 | 78.26 |
| SH-Mean | 60.97 | 70.19 | 61.25 | 75.25 |
| TH-Marginal | **94.52** | 94.44 | 94.45 | 94.26 |
| STH-Space-time | 94.20 | 94.11 | 94.05 | 94.14 |

Table 7.2 – Comparison of overall classification accuracy

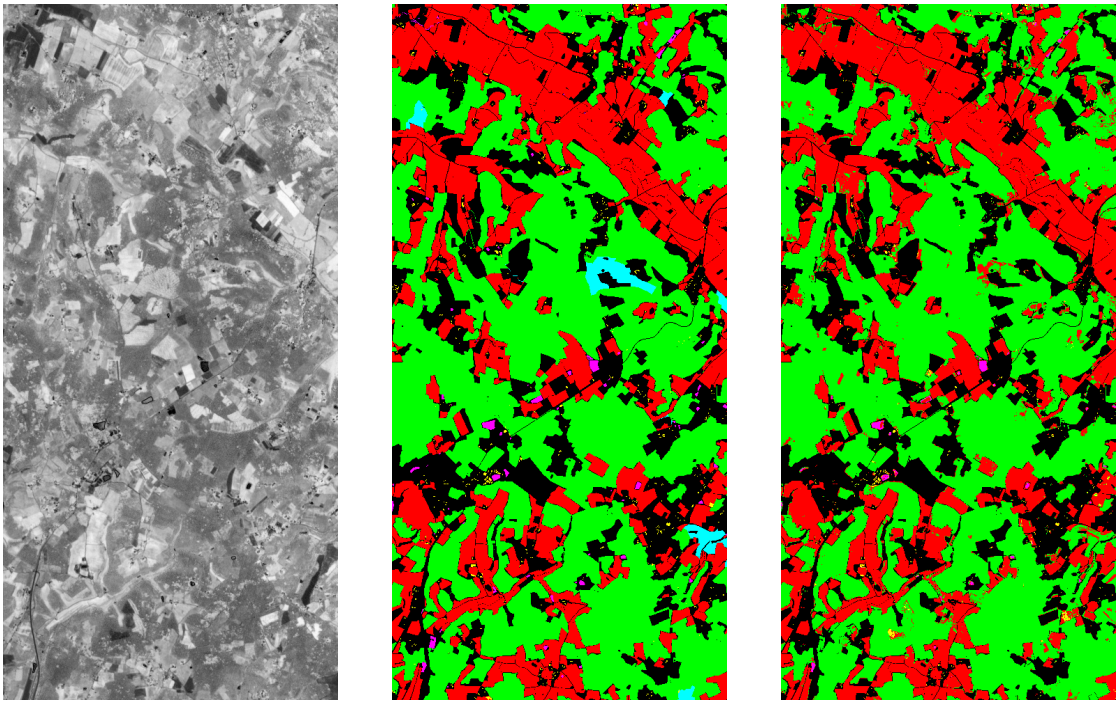


Figure 7.2 – SITS Classification for the Dordogne dataset (from left to right): Sentinel-2 extract (date 1), reference map and sample result. (yellow: urban, red: crop, cyan: water, green: forest and pink: moor (black for ignored pixels).

| | Urban | Crops | Water | Forest | Moor |
|---|---|---|---|---|---|
| Nb. of samples | 1381-1636 | 105214-133374 | 1754-2010 | 266649-226231 | 1901-5354 |
| Without Profile | 0.51 | 0.93 | 0.59 | 0.95 | 0.002 |
| SH-Lex. | 0.009 | 0.18 | 0.004 | 0.71 | 0.001 |
| SH-DTW | 0.17 | 0.68 | 0.009 | 0.83 | 0.003 |
| SH-Mean | 0.14 | 0.64 | 0.005 | 0.82 | 0.003 |
| TH-Marginal | 0.54 | 0.94 | 0.55 | 0.96 | 0.003 |
| STH-Space-time | 0.53 | 0.94 | 0.57 | 0.95 | 0.003 |

Table 7.3 – Number of samples for training and testing parts and $F_1$ score for each class.

According to these results, SH is not adequate to classify SITS, but on the other hand, it can be used for other applications such as change detection as in [164]. Results of TH and STH are very similar to each other and it is difficult to conclude on a better approach.

### 7.1.3 Constructing attribute profiles with the spatio-temporal attributes

The third experiment aims to demonstrate the benefits of spatio-temporal attributes for land cover applications. We investigate efficiency of beginning, ending, duration, node centroid and amplitude attributes for classification of SITS pixels. To do so, we used a dataset which covers a full Sentinel-2 tile of 10,980 ×10,980 pixels in the north-west of French Brittany. We collected top-of-canopy (level 2A) images from the Breizhcrops [3] covering the year 2017. We applied gapfilling in order to fill non-overlapping pixels and clouds by making a linear temporal interpolation. As with the Dordogne data, we used RPG to collect reference data representing the 11 following classes: wheat, barley, corn, rapeseed, protein crop, pastures, fodder, fallows, vegetable or flowers, permanent meadows and temporary meadows. We have calculated NDVI values of each image and used NDVI images as input for the classifier.

We split the images in half vertically and horizontally to use training and testing samples from different regions of the images. Furthermore, we introduce a gap of 100 pixels between training and testing areas. As we want to statistically evaluate the results, we have used training and testing parts reversely. More precisely, we applied classification four times and we report average of the four results.

As a classifier, we have used RF with 100 trees and without cross validation. We selected maximum depth as 25 and minimum split as 2. We compared our results with conventional Morphological Profiles as applied for SITS in [53] since this recent work is the most similar to ours. We have applied morphological filtering with a disk structuring element and we have selected three different sizes: 3, 5 and 7. Since we applied opening and closing separately, we have obtained 6 different results from one single image. In order to obtain profiles with the same amount of images, we have used 3 thresholds on each attribute with min and max-tree and 6 thresholds for ToS and $\alpha$-tree on each attribute also. In order to ensure comparable profiles, we set area attribute thresholds to 28, 81 and 149 which correspond to the amount of pixels in disks of 3, 5 and 7 radius respectively. In addition, we added three more empirical thresholds whose values are 1000, 10000, 100000 for

3. https://breizhcrops.org/

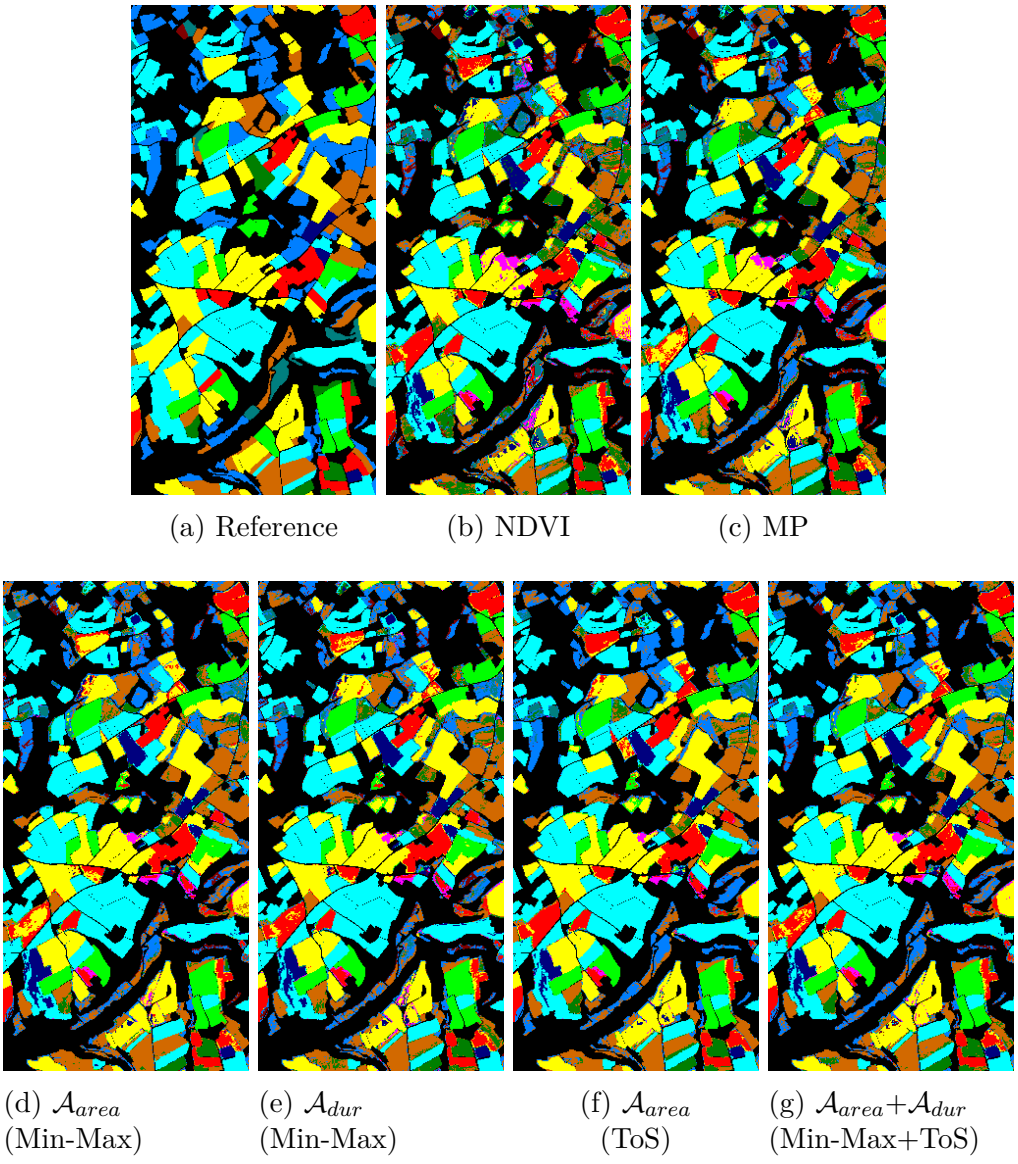
the $\alpha$-tree. On the other hand, we have used 100, 10000 and 1000000 thresholds for area filtering with other trees and 50, 100, 150 for height and amplitude attributes for all trees. Duration thresholds are set to 3, 6 and 9 for all kind of trees.

| Method | Hierarchy | Dimension | OA |
|---|---|---|---|
| NDVI | - | 1 | 56.79±1.0 |
| MP | - | 7 | 59.16±1.0 |
| Min-max tree | | | |
| $\mathcal{A}_{area}$ | TH | 7 | 59.33±1.1 |
| $\mathcal{A}_{height}$ | TH | 7 | 59.19±1.3 |
| $\mathcal{A}_{amp}$ | STH | 7 | 57.19±1.1 |
| $\mathcal{A}_{comp}$ | TH | 7 | 57.02±1.1 |
| $\mathcal{A}_{cent}$ | STH | 7 | 57.38±1.1 |
| $\mathcal{A}_{dur}$ | STH | 7 | 57.55±1.1 |
| $\mathcal{A}_{comp} + \mathcal{A}_{dur}$ | TH+STH | 13 | 57.62±1.1 |
| $\mathcal{A}_{beg} + \mathcal{A}_{end} + \mathcal{A}_{dur}$ | STH | 19 | 57.57±1.0 |
| $\mathcal{A}_{area} + \mathcal{A}_{dur}$ | TH+STH | 13 | 61.13±1.1 |
| Tree of Shape | | | |
| $\mathcal{A}_{area}$ | TH | 7 | 61.92±1.1 |
| $\mathcal{A}_{comp}$ | TH | 7 | 55.91±0.9 |
| $\mathcal{A}_{height}$ | TH | 7 | 59.86±1.3 |
| Min-max tree+Tree of Shape | | | |
| $\mathcal{A}_{area} + \mathcal{A}_{dur}$ | TH+STH | 13 | **62.07**±1.0 |
| $\alpha$-tree | | | |
| $\mathcal{A}_{area}$ | TH | 7 | 60.32±1.2 |
| $\mathcal{A}_{height}$ | TH | 7 | 56.63±1.1 |
| $\mathcal{A}_{comp}$ | TH | 7 | 56.60±0.8 |
| $\mathcal{A}_{amp}$ | STH | 7 | 56.24±1.3 |
| $\mathcal{A}_{dur}$ | STH | 7 | 57.08±1.3 |
| $\mathcal{A}_{area} + \mathcal{A}_{dur}$ | TH+STH | 13 | 59.34±1.5 |

Table 7.4 – Overall accuracy results with Brittany dataset

Table 7.4 reports the overall accuracy for each tree and attribute. As a baseline, we have calculated the overall accuracy by using only NDVI values. MP is the result provided by the method [53]. We have used three spatial attributes (area, height and compactness) with the TH strategy. Since compactness is not an increasing attribute, we have applied non-pruning strategies to filter the trees with this attribute. We have used ToS with only spatial attributes because ToS is not appropriate to calculate spatio-temporal attributes as it is justified in section 4.3.1. As expected, the proposed attributes give a higher accuracy compared to NDVI except amplitude with the $\alpha$-tree. It should be noted that, while spatial attributes give better results compared to MP, using only spatio-temporal attributes

do not surpass MP. However, we show that spatio-temporal attributes can be used to obtain better accuracy if they are used with spatial attributes such as area. Best result is obtained by using area attribute with ToS and duration attribute with min and max-tree together.



(a) Reference (b) NDVI (c) MP

(d) $\mathcal{A}_{area}$ (Min-Max) (e) $\mathcal{A}_{dur}$ (Min-Max) (f) $\mathcal{A}_{area}$ (ToS) (g) $\mathcal{A}_{area}$+$\mathcal{A}_{dur}$ (Min-Max+ToS)

Figure 7.3 – Reference Data and Results for Brittany dataset. (yellow: wheat, red: barley, cyan: corn, green: rapeseed, magenta: protein crops, brown: pastures, dark green: fodder, blue: vegetable).

Figure 7.3 visually illustrates some results of the different methods on a small area of the full Brittany dataset. Each colour represents a different class and black pixels are unclassified pixels. Figures 7.3d and 7.3f show results with area attribute for min-tree, max-tree and ToS respectively. The best result according to the Table 7.4 is displayed in Figure 7.3g. These results led to wrongly classified pixels in same regions. For instance, wheat (yellow) and barley (red) classes are interchanged occasionally. However, the result obtained with MP has more salt and pepper noise compared to the other results. The effect of the duration attribute can be seen by observing Figures 7.3f and Figure 7.3g: Wheat (yellow) and pastures (brown) are classified more efficiently.

## 7.2 Change detection

Change detection aims at finding changed pixels between successive images. In the literature, the proposed methods mostly rely on bi-temporal images. In this section, we will propose different methods for different kinds of change detection applications such as detecting changed urban regions, flooded water and mobile objects.

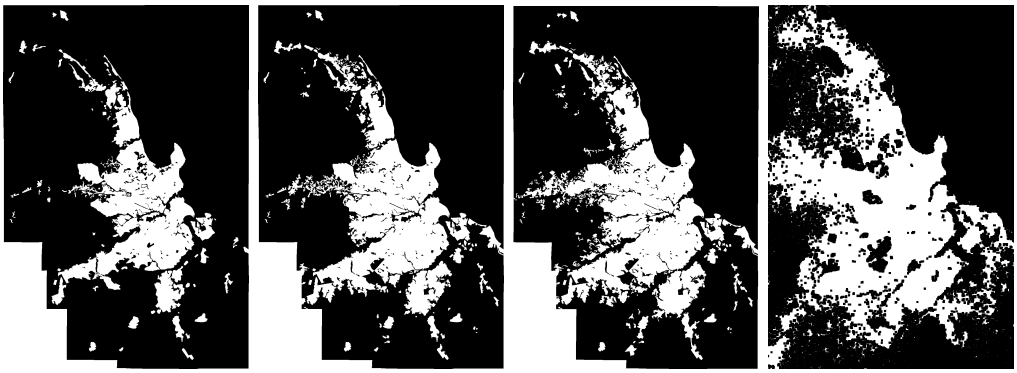
### 7.2.1 Monitoring urban growth

Monitoring urban growth and change is an important task for urban planning and disaster management. While several change detection approaches have been proposed to deal with growing urban areas, their performances are usually limited due to outliers in SITS. Our aim is to discriminate urban growth from the other changes by using tree representations.

#### Dataset

As a dataset, we consider here the Dar es Salaam region in Tanzania, that is known for careful assessment of its urban cover expansion (from 12,988 to 20,607 hectares, about 47% increase [105]). We illustrate such an expansion through

reference maps given in Figure 7.4 and taken from [153]. In these maps dating from 1992, 1998 and 2002 respectively, urban areas are identified in white areas. Furthermore, we also include a 2017 reference taken from OpenStreetMap [66].



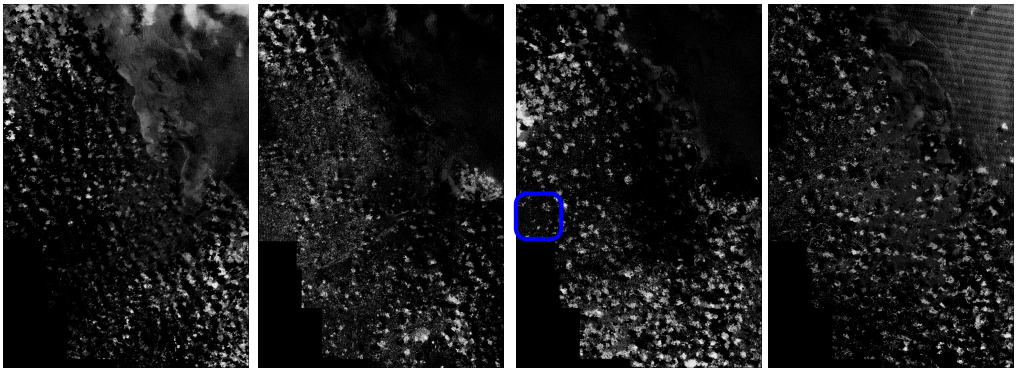
(a) 1992 (b) 1998 (c) 2002 (d) 2017

Figure 7.4 – Reference data for Dar es Salaam, Tanzania. White pixels indicate urban areas.

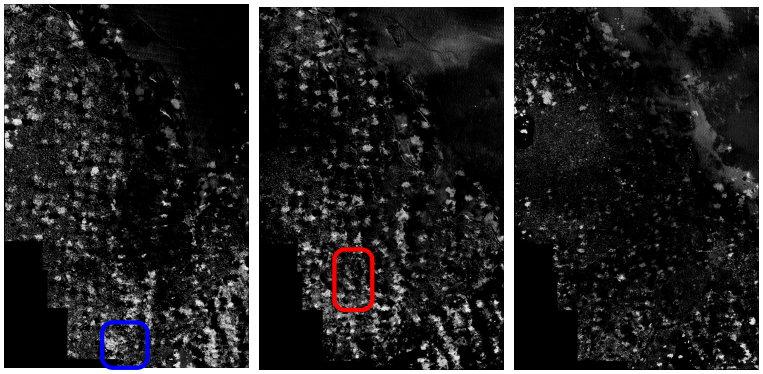
The SITS is made of Landsat-5 (1991, 1992, 1995, 1997, 1998, 2000, 2011) and Landsat-8 images (2017), with a 30m spatial resolution provided by the United States Geological Survey (USGS). We clipped the Landsat data according to the reference maps shown in Figure 7.4, leading to an image size of 1595 pixels $\times$ 1076 pixels. Quantitative evaluation is performed by comparing the obtained results with the difference map between 1992 and 2017 (see Figure 7.7a). We highlight changes between successive images in Figure 7.5. Among available spectral bands, we select only the blue band (first band) with the shortest wavelength (0.45-0.52$\mu$m) and 30m spatial resolution. This choice is motivated by the higher dynamic range and its proven ability to detect clouds [65] since we have cloudy images. The clouds were presented as an instantaneous effect challenge in section 3.1.2.

### Method

Our approach is based on two steps: first we conduct a temporal analysis to find the highly possible changed areas, before filtering the resulting image using spatial information. In order to find highly possible changed areas, we use the



(a) 1991-1992 (b) 1992-1995 (c) 1995-1997 (d) 1997-1998



(e) 1998-2000 (f) 2000-2011 (g) 2011-2017

Figure 7.5 – Difference between the Landsat images.

SH strategy by using a function to stack pixel time series to one scalar value. Since we aim to find changed areas, we measure the spread of the time series data. Statistical dispersion functions such as range, inter-quartile range and standard deviation are well-known to calculate such variability or spread [158]. Indeed, if spreading of the data is high, possibility of the change would be high as well. A higher pixel value in the new image commits a higher change for each location.

A simple spreading measure is range, i.e. the difference between the maximum and minimum values in each location:

$$I_{range}(x) = \max_{t\in[1,n]}(I_t(x)) - \min_{t\in[1,n]}(I_t(x)) \tag{7.1}$$

A second measurement is the inter-quartile range (IQR) which is here set as the difference between the $75_{th}$ (third quartile, $Q_3$) and $25_{th}$ (first quartile, $Q_1$) percentile of the data:

$$I_{IQR}(x) = Q_3(x) - Q_1(x). \tag{7.2}$$

Quartile coefficient of dispersion is also computed by using first and third quartiles:

$$I_{Q_{coef}}(x) = \frac{Q_3(x) - Q_1(x)}{Q_3(x) + Q_1(x)}. \tag{7.3}$$

Another measure of spread is standard deviation. It is calculated in terms of how far the data differs from the average:

$$I_{std.dev}(x) = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\Big(I_t(x) - \bar{I}(x)\Big)^2}, \tag{7.4}$$

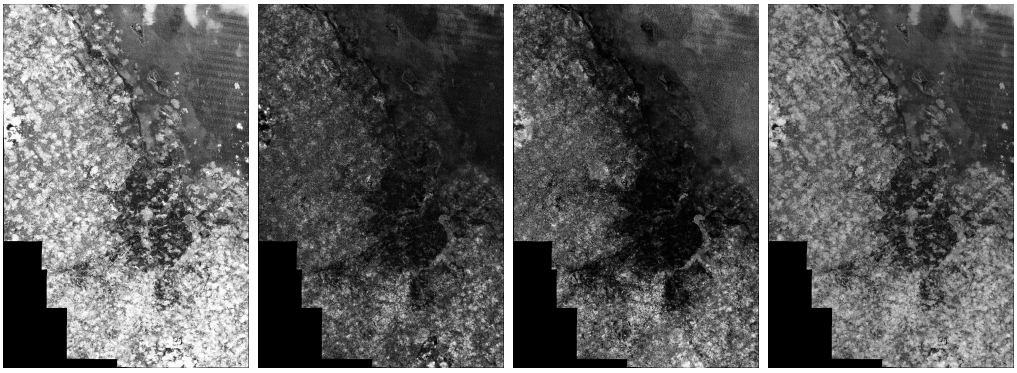
with

$$\bar{I}(x) = \frac{1}{n}\sum_{t=1}^{n} I_t(x) \tag{7.5}$$

Although other statistical dispersion measurements have been introduced (e.g. entropy [137]), we limit ourselves to these basic functions for our experiments. Dispersion measures were computed for every pixel location separately, and gathered into a new synthetic image as already explained.

After stacking pixel time series, the second step is to analyze the resulting image in spatial domain. To do so, we use the area attribute since we aim to distinguish between large connected components denoting the urban extent and smaller ones that can be seen as outliers or noise. We prefer here the max-tree over other alternatives since we focus on higher values corresponding to high possible changes. Area filtering was applied with several thresholds, that were here defined empirically. We assume that urban growth areas are connected sets of changed pixels whose area is larger than those of other parts of the satellite images such as clouds.

We show in Figure 7.6 the stacked images with the different functions from equations 7.1-7.5. These functions darken the stationary city area compared to changed areas that are given higher values. Nevertheless, we can see that clouds still bring some artifacts in these images. We then apply an area filter over each



(a) $I_{range}$ (b) $I_{IQR}$ (c) $I_{Qcoef}$ (d) $I_{std.dev}$

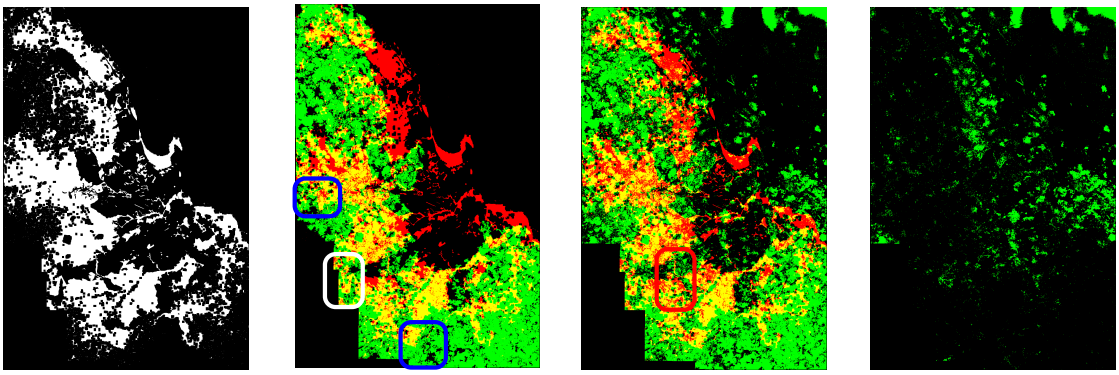
Figure 7.6 – Synthetic images computed from the SITS.

of these images, with an efficient implementation based on a max-tree structure. We have considered different thresholds for the area filtering and report here the observed results.

### Results

We provide in Figure 7.7 a visual comparison between the reference change map (a) made of difference between reference maps from 1992 and 2017, and the results provided by our method (b) considering range as the spread function. The color codes are as follows: red, green and yellow pixels represent false negative (i.e. wrongly classified non-urban areas), false positive (i.e. wrongly classified urban areas) and true positive (i.e. pixels correctly identified as new urban areas), respectively. The effect of spatial filtering is further assessed by comparing with the image before filtering (c), with corrected errors given in (d). As it can be observed, the dataset is challenging due to the presence of clouds and changes over a 25 years period.

We can see the effect of using an area filtering, that allows us to remove areas smaller than a given threshold such as clouds. One of them is highlighted with a red box in Figure 7.7c and corresponds to a cloudy area (see Figure 7.5f). As emphasized by differences between Figures 7.7b and 7.7c, spatially filtering of time series information allows us to reduce artifacts due to noise or clouds. White box in Figure 7.7b shows the correctly detected urban area after filtering compared



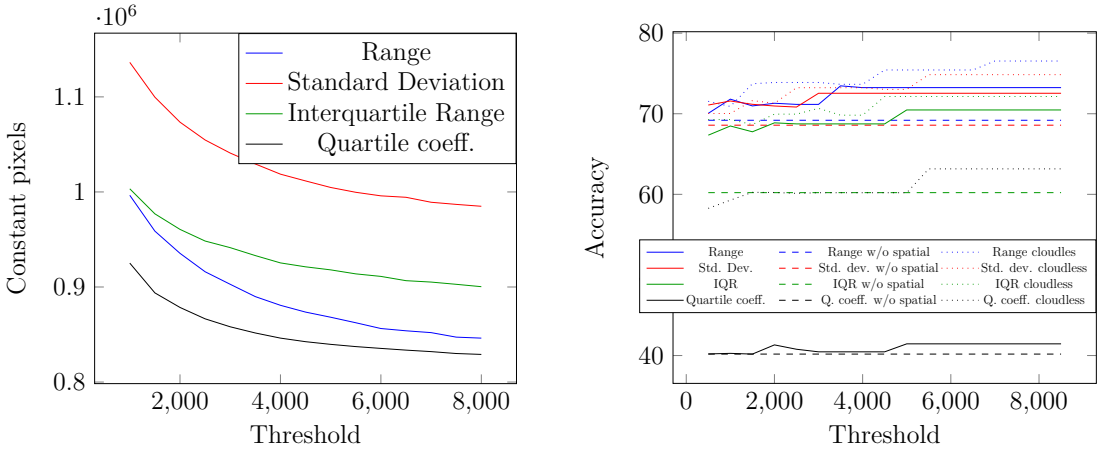
(a) Reference (b) h=10,000 (c) Unfiltered (d) Corrected

Figure 7.7 – Urban growth estimation: reference data, result with and without spatial filtering, and difference in between.

to the same area in Figure 7.7c. However, there are still wrongly classified parts caused by clouds connected to urban changes (see blue boxes in Figures 7.7b, 7.5c and 7.5e). Such an issue might be addressed with advanced connectivity [125].

Finally, we report quantitative evaluation in Figures 7.8a and 7.8b. We first show in Figure 7.8a the stability of the synthetic image w.r.t. the chosen threshold value. We can observe that after a severe decrease, increasing further the area has a limited filtering effect. Figure 7.8b provides a comparison analysis in terms of true positive rate w.r.t. the threshold value, that confirms the stable behavior. We can observe the relevance of the spatial filtering, with significant improvement (+5-10%) over the temporal (pixel) analysis only, and better results being obtained with higher area thresholds. Also, we can notice that the presence of clouds does not affect too strongly the accuracy with only a few points lost w.r.t. a cloud-free dataset (see discussion below). The whole process takes 24 seconds with a MacBook Pro 2.6 GHz Intel Core i5 processor and Python implementation. It can be further reduced with efficient implementations of tree-based processing [99].

For the sake of comparison, we compare the different images with and without the spatial filtering. Furthermore, we also compare with two variants inspired from [50], for which we set the thresholds similar to our settings. The first one computes a change map as the difference between respective AP from the first and last dates. While it can reaches a high accuracy (89%) when applied to cloudless images (1992 and 2017), it remains ineffective when applied to cloudy data (30%




(a) Threshold selection (b) True Positive rate

Figure 7.8 – Results with different dispersion functions, compared with cloudless data and without the spatial filtering.

if 2017 is compared with the cloudy 1991 image). The second variant computes the sum of AP differences of each successive frames. Again, we assess it on both the standard dataset made of all dates (i.e. containing clouds) and on a subset ignoring cloudy images (1991, 1997, 2000). It reaches an accuracy of 50% and 75% respectively, showing its higher sensitivity to clouds than the proposed approach. Let us note that using the spectral information would most probably lead to better results. Conversely to existing works that detect changes between only two dates, our approach accommodates with long time series data.

### 7.2.2 Flood detection

Monitoring flood is an important task for disaster management. It requires to distinguish between changes related to water from the other changes. We address such an issue by relying on both spatial and intensity information from SAR data. The SAR backscatter depends on the physical properties of the objects, with water reflecting less than other materials [160]. Therefore, radiometric thresholding is known as an efficient way to extract water areas [59]. However, such a method is very sensitive to noise, and spatial regularization thus appears as a relevant strategy to improve robustness. To do so, we propose here to rely on morpho-

logical representations and characterize dark connected components using various attributes such as level, area and stability. While bright object detection (ship) was performed with max-tree in [144], we focus here on dark objects and thus use rather the min-tree. We will propose two different approaches to detect flood with the space-time tree representation.

We illustrate our approach with Sentinel-1A images acquired over Montmirail, North of France and East of Paris. All images come with a 10m pixel spacing Ground Range Detected products with the Interferometric Wide Swath (IW) and VV polarization. The SITS is made of 3 images acquired on 17 July 2017, 10 August 2017 and 25 January 2018. We rely on the Copernicus Emergency Management Service [4] flood mapping shape files as a ground truth for assessing our method. The flood occurred on 22 January 2018, i.e. between second and third images of the series.

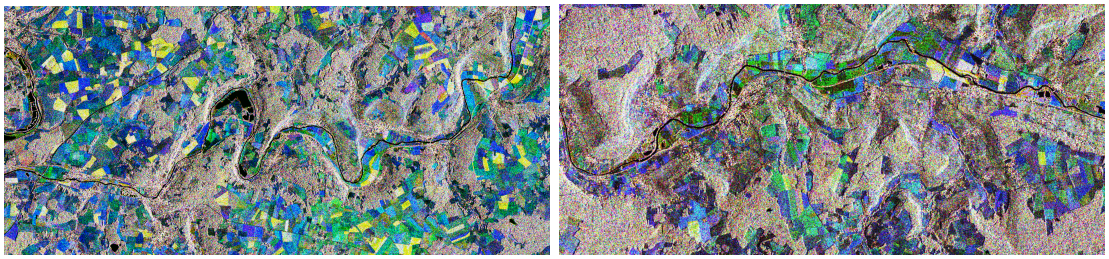
### First approach

Our first approach consists in two steps. We first perform a temporal analysis to identify images containing possible floods. Then a spatial analysis is achieved to detect flood areas on the selected images. Both steps rely on the analysis of component attributes extracted from the min-tree representation.

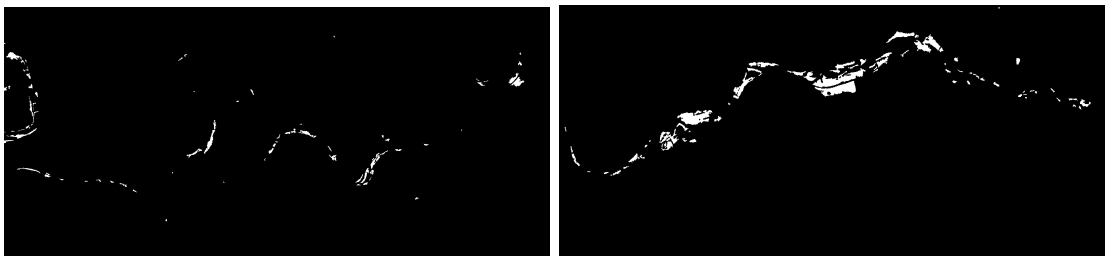
For the sake of illustration, we focus on two specific areas where floods occurred, as shown in Figure 7.9 (histogram equalization is included solely for visibility purpose) that also contains reference flood maps. Colorized images are created by merging the three images from different dates in chronological order to emphasize changed areas. The two SITS samples are made of $1276 \times 2803$ and $892 \times 1941$ pixels, respectively. A close-up view is provided in Figure 7.10 to ease visual assessment of the changes, and distinguishing changes due to floods from other changes (e.g. related to crops).

We first aim to identify which images of the SITS are most likely to contain floods. This process can be seen as a specific case of change detection, where only water-related changes are sought. In order to find which image is most likely to contain floods in the time series, we characterize each image $I_t$ by some measures

4. https://emergency.copernicus.eu/



(a) $I$ (b) $I'$

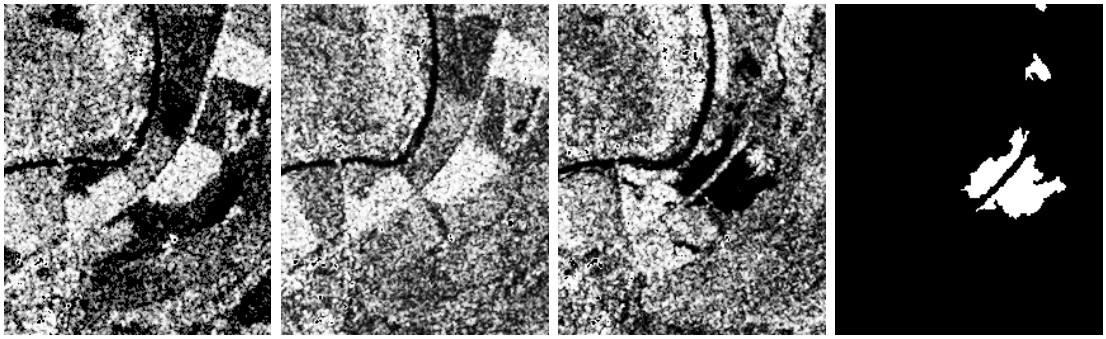


(c) GT of $I$ (d) GT of $I'$

Figure 7.9 – Colorized SITS of two study areas $I$ and $I'$, with corresponding reference flood maps.

computed from the nodes from its corresponding tree $\mathcal{T}(I_t)$. Indeed, we assume that the successive images of a scene share the same topology and contain similar spatial structures. Although intensity is changing (see Figure 7.10) over time, the shapes remain similar. The min-tree offers an efficient way to analyze such structural information through time. We consider here two different measures to characterize the structural complexity of an image. The first, straightforward measure is simply the amount of nodes contained in the tree (see section 5.4.1). Another, more advanced feature is based on the concept of stability used in the Maximally Stable Extremal Regions (MSER) and presented in equation 2.45.

Table 7.5 illustrates these two measures for the successive images of the two SITS. We can see that both are good indicators of the image complexity. More stable regions indicate that the corresponding image includes more shapes or structures at time $t = 3$, i.e. in case of a flood. Once the flood event date has been identified, we use related images $I_3$ and $I'_3$ as flooded images among their respective SITS. We then compare these images with other dates to achieve a spatial delineation of the floods, as described in the next subsection.



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) GT

(e) $I'_1$ (f) $I'_2$ (g) $I'_3$ (h) GT

Figure 7.10 – Close-up of the two study areas $I$ and $I'$ from Sentinel-1 imagery (subscript denotes the temporal index in the SITS) and corresponding references.

The second step is to analyze bitemporal images, which are flooded and non-flooded images. More precisely, we consider two successive images from the time series, respectively before and after the flood occurs, i.e. in our case $(I_2, I_3)$ and $(I'_2, I'_3)$. We now build a spatio-temporal min-tree where the connectivity is both in space and time domains (i.e. 6-connectivity). Within this space-time tree, we explore the relationship between parent and child nodes for each level of the tree. Since the nodes embed now some temporal information, the variability of intensity values helps to locate where the intensity values are dramatically changing. It gives us valuable insights for spatially delineating the flooded areas.

We thus measure for each node $C$ the difference in variances between the node and all its children $C_c^k$:

$$VD(C) = \sum_k \mid \mathcal{A}_{var}(C) - \mathcal{A}_{var}(C_c^k) \mid \tag{7.6}$$

| | # Nodes | # MSER |
|---|---|---|
| $I_1$ | 1,376,403 | 30,699 |
| $I_2$ | 1,486,696 | 34,400 |
| $I_3$ | 1,531,558 | 37,881 |

| | # Nodes | # MSER |
|---|---|---|
| $I'_1$ | 789,554 | 20,607 |
| $I'_2$ | 835,335 | 24,853 |
| $I'_3$ | 859,766 | 29,513 |

Table 7.5 – Evolution of structural complexity for $I$ and $I'$.

We then sum these values for each level $\lambda$ of the tree, leading to a total variance difference measure defined as:

$$TVD(\lambda) = \sum_{k} VD(C_\lambda^k) \tag{7.7}$$

Finally, we look for abrupt variance changes that relate to dark areas (i.e. water pixels) or min-tree leaves. We set a threshold value according to this variance change. To illustrate, we plot in Figure 7.11 the variance analysis for $\mathcal{T}(I_2, I_3)$ for each level $\lambda$. Here, our aim is to find the nodes including only water. Since there is a turning point close to the lowest intensity, we used this point as threshold and extracted all the nodes with a level lower than the threshold (higher level nodes would include not only water but also different objects). Indeed, variance difference between parent and child nodes is lower when the nodes correspond to similar objects.

Nevertheless, we observe that some artifacts [39] can still occur due to double bounce effect, backscatter similarity of dry soil, *etc*. In order to overcome these errors, we post-process the binary change detection map with a small area filtering (e.g. $\lambda_a = 20$). Figure 7.12 provides close-ups of colorized SITS (one date per channel) and our results compared with their respective ground truths. In both extracts, our method is able to correctly identify flood areas.

Furthermore, we report in Table 7.6 quantitative results obtained on the whole images from Figure 7.9. We consider here the $F_1$ score. For the sake of comparison, we also report results from two existing methods: AP-based change detection [50] that also relies on spatial attributes extracted from morphological hierarchies, and the Normalized Difference Flood Index (NDFI) [39] thresholding approach. Let

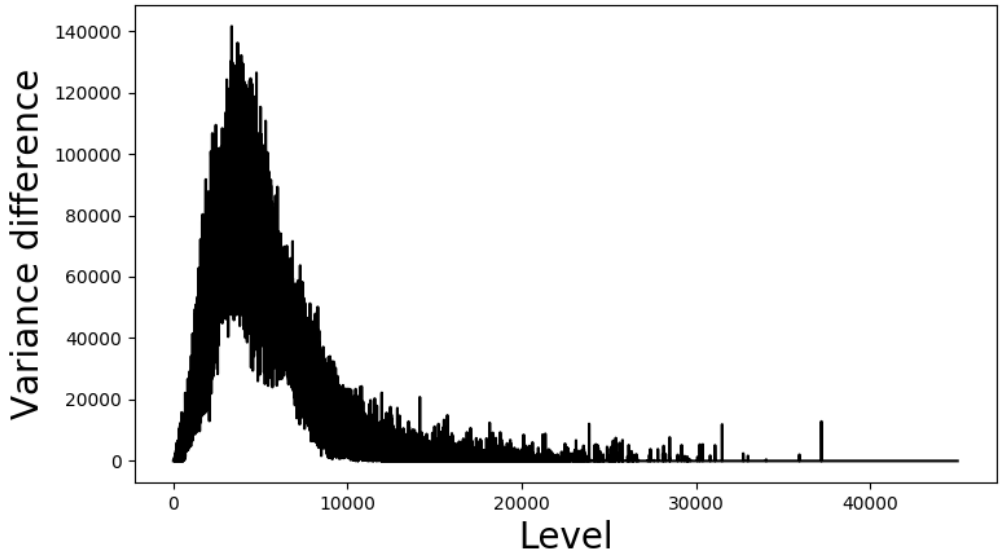



Figure 7.11 – Total variance difference for each level of the tree.

us note that authors used longer time series compared to our dataset in their experiments. We can observe that our method achieves better results and appears as an efficient solution for flood mapping from SAR imagery.

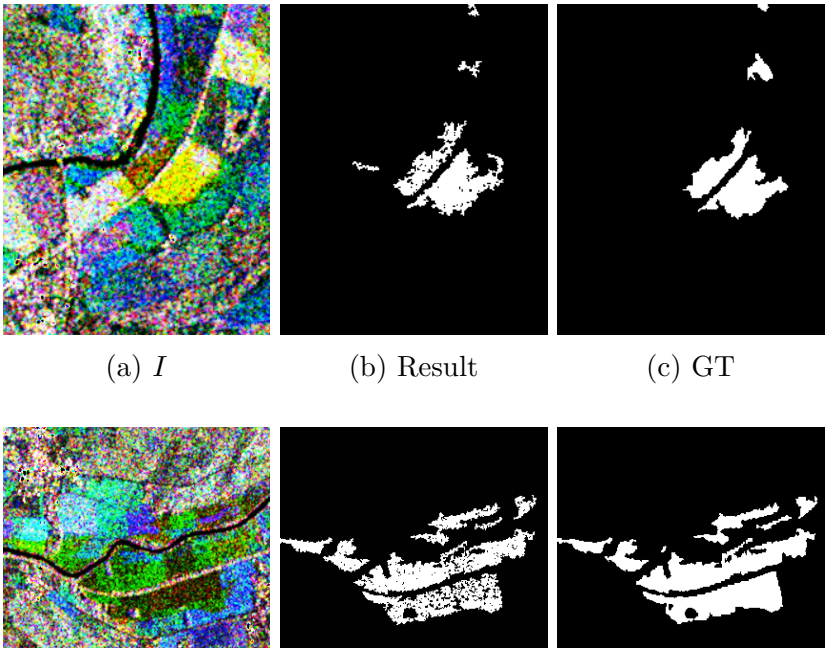
| SITS | Proposed | AP[50] | NDFI[39] |
|---|---|---|---|
| $I$ | **0.55** | 0.18 | 0.47 |
| $I'$ | **0.73** | 0.51 | 0.60 |

Table 7.6 – Quantitative evaluation of flood detection for different methods and SITS, using the $F_1$ measure. Bold values indicate highest accuracy.

### Second approach

Our second approach relies on the spatio-temporal stability attribute. Since water regions are not stable through time, lower stabilty values can be used to detect them. After finding the unstable nodes with a threshold empirically set to $h = 0.2$, we then end with the following difference image (where $i$ is the time when flooding is visible):

$$I_i^f = I'_i - I'_{i-1} \tag{7.8}$$



(a) $I$ (b) Result (c) GT

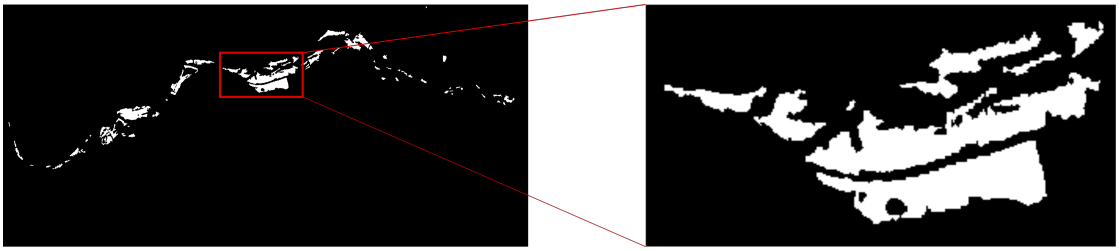
(d) I' (e) Result (f) GT

Figure 7.12 – Close-up of the results (from left to right): color composition of the SITS, comparison between our result and the ground truth (GT).

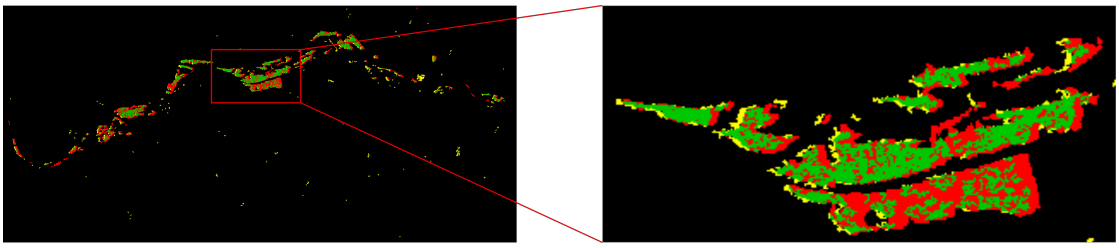
Since constant waters have the same location in successive images, they are removed in $I^f$. Finally, we build a binary map through a simple threshold that discards null values:

$$I^f = \begin{cases} 255 & I_i^f > 0 \\ 0 & \text{otherwise} \end{cases} \tag{7.9}$$

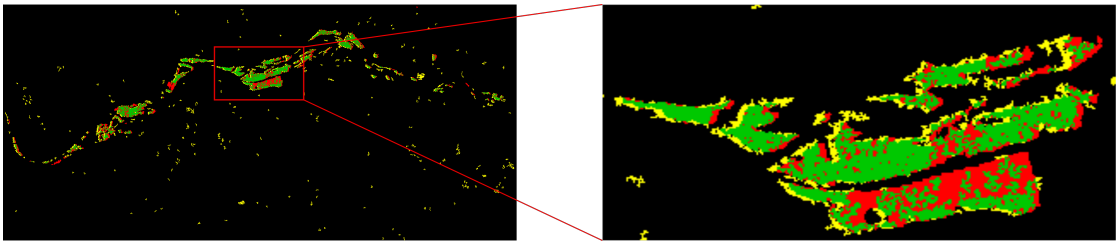
Figure 7.13 illustrates the reference data (a), as well as the results obtained by the different methods (b-d): NDFI ($I^{ndfi}$), min-tree based radiometric thresholding (first approach) ($I^{min}$) and this proposed method ($I^f$). To ease visual assessment, flooded areas are given using the following color codes: TPs in green, FNs in red, and FPs in yellow. We also provide some close-up illustrations. This approach reaches better $F_1$ score for the second dataset (0.79) compared to the first approach which is reported as 0.73 in Table 7.6.



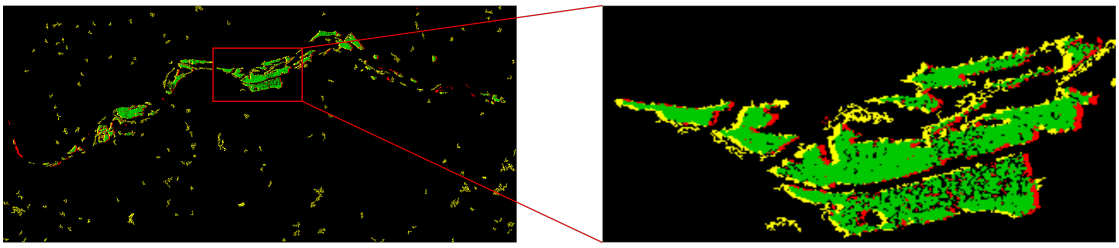
(a) $I^{ref}$



(b) $I^{ndfi}$



(c) $I^{min}$



(d) $I^{f}$

Figure 7.13 – Flood Observation (from top to bottom): Reference image for flood (a); Results with NDFI (b), the min-tree threshold analysis (c), and our method (d). Results are given in green, red, and yellow for TP, FN, and FP respectively.

### 7.2.3 Mobile objects

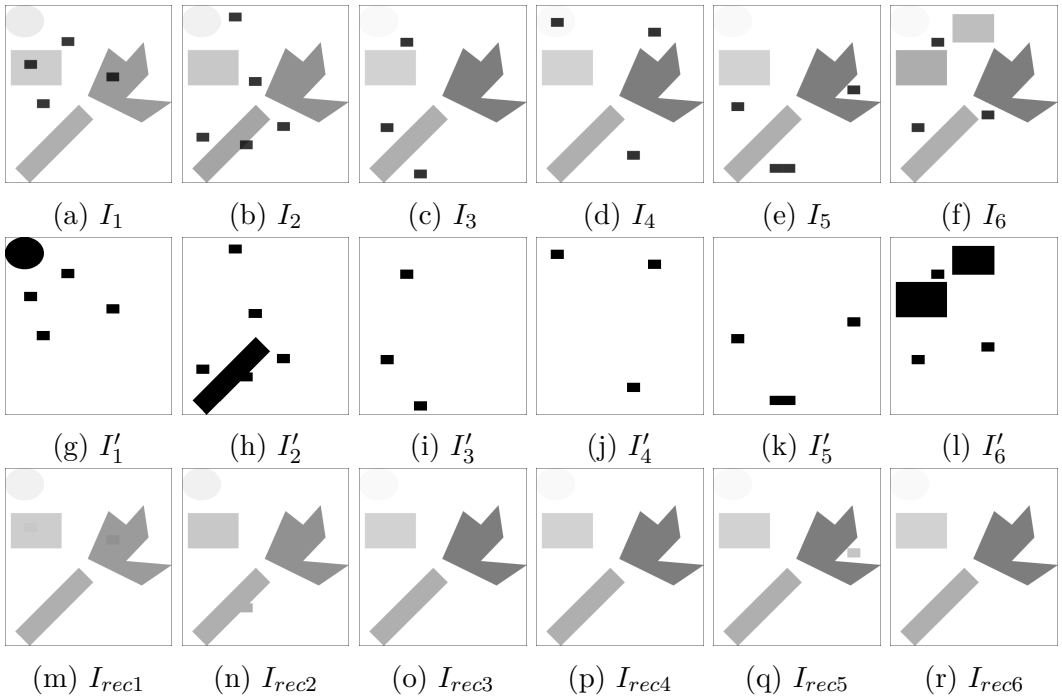
Very high resolution images contain very small details which could correspond to mobile objects. However, these mobile objects may be irrelevant for some applications such as vehicles on the road for road extraction as it is discussed in section 3.1.2. In this section, we propose a novel method to detect and remove mobile objects on very high spatial resolution SITS.

Although identifying mobile objects from the scene is a computer vision task [49], removing of moving objects and subsequent background inpainting is rarely addressed in the literature. There are few related works which utilize point cloud scenes [90, 1, 147]. In [15], the authors proposed a method including detection and removal of objects separately with a supervised approach (deep learning) from 3D cameras. In [54], they calculated the occurrence duration of the objects with laser scanning properties of LIght Detection And Ranging (LIDAR) systems. Background subtraction algorithms are used to remove objects from 3D camera images in [159]. To the best of our knowledge, removing mobile objects from optical SITS has not been addressed in the literature yet.

The closest works are probably object tracking in satellite videos [150], missing data reconstruction [151], change detection [92] and cloud removal [89]. Satellite videos are acquired within the same day and are thus less subject to intensity change or abrupt changes compared to SITS. Recently, unsupervised change detection with deep change vector analysis was proposed in [141] and unsupervised change detection with convolutional auto-encoder was proposed in [12]. They used bi-temporal very high resolution images in their experiments. Conversely to the aforementioned methods, we desire to use all spatiotemporal information at once and benefit from object based information since object based analysis has proven its success compared to pixel based analysis [71].

In order to include mobile objects in an image set, we first created an entirely synthetic image time series. The dataset is composed of six grayscale images, each containining different shapes (*e.g.*, circle, rectangles or polygons) with gray level values that evolve through time. Mobiles objects are simulated by adding 3 to

5 small dark rectangle shapes that appear randomly at different positions as in Figures 7.14g-7.14l. Figures 7.14a-7.14f display the six images of this toy dataset. Exceptionally, $I_6$ includes a large rectangle as an abrupt change.



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) $I_4$ (e) $I_5$ (f) $I_6$

(g) $I'_1$ (h) $I'_2$ (i) $I'_3$ (j) $I'_4$ (k) $I'_5$ (l) $I'_6$

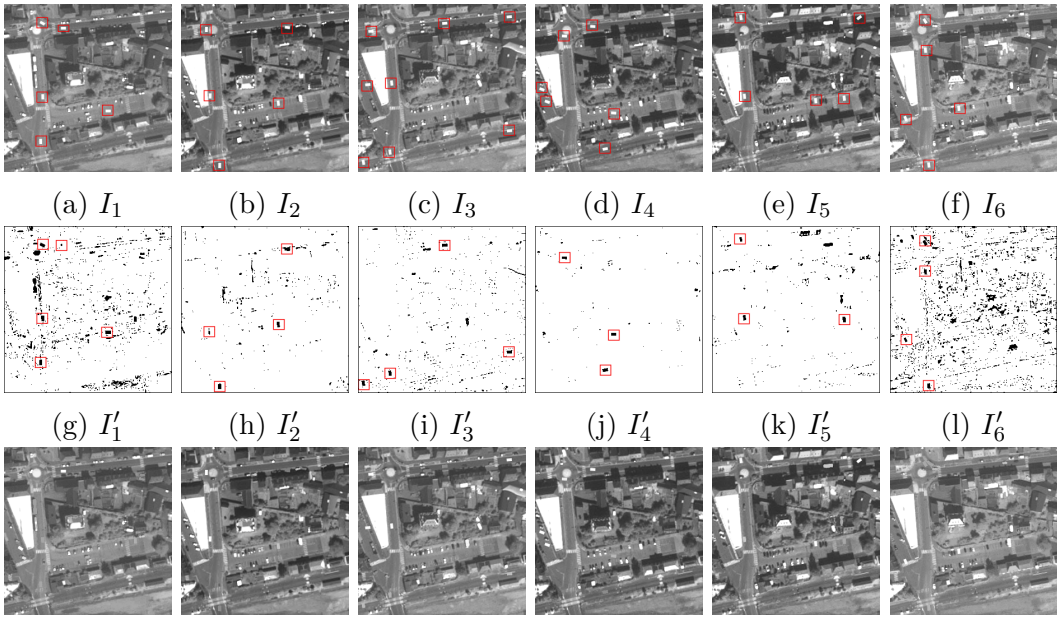
(m) $I_{rec1}$ (n) $I_{rec2}$ (o) $I_{rec3}$ (p) $I_{rec4}$ (q) $I_{rec5}$ (r) $I_{rec6}$

Figure 7.14 – Experiments with synthetic images.

For real image experiments, we used sample 0.5m panchromatic Pleiades images extracted from Kalideos [5] (300 pixels × 300 pixels). The studied area is located at Pontorson. The time series is composed of six images acquired from May 2018 to April 2019 (acquisition dates are given in Table 7.7). There are mobile objects (for example cars) but there is no available ground truth for this dataset. In order to evaluate quantitatively the proposed approach, we added artificial mobile objects to each Pléiades images and created an hybrid dataset from SITS images, which are displayed in the first row of Figure 7.15. Our synthetic mobile objects are inside of the red rectangles. Every image contains these synthetic objects and the number of these objects ranges from 4 to 8.

The quantitative evaluation is twofold:

5. https://bretagne.kalideos.fr



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) $I_4$ (e) $I_5$ (f) $I_6$

(g) $I'_1$ (h) $I'_2$ (i) $I'_3$ (j) $I'_4$ (k) $I'_5$ (l) $I'_6$

(m) $I_{rec1}$ (n) $I_{rec2}$ (o) $I_{rec3}$ (p) $I_{rec4}$ (q) $I_{rec5}$ (r) $I_{rec6}$

Figure 7.15 – Experiments with hybrid dataset from Real SITS

| Images | Acquisition dates |
|---|---|
| $I_1$ | 07/05/2018 |
| $I_2$ | 23/06/2018 |
| $I_3$ | 06/08/2018 |
| $I_4$ | 02/09/2018 |
| $I_5$ | 27/09/2018 |
| $I_6$ | 12/04/2019 |

Table 7.7 – Acquisition dates of Pléiades images.

1. Detection accuracy: evaluate if the chosen pruning operation (based on the duration attribute) is efficient to detect mobile objects.

2. Reconstruction error: evaluate if the reconstruction strategy outputs plausible intensity values.

For this purpose, a space-time min-tree for the synthetic dataset and a space-time max-tree for the hybrid dataset are built. The duration attribute is computed at each node. Then, the tree is filtered with $h = 6$, and finally images are recon-

structed. In order to find detection accuracy, we have calculated the difference between the filtered and the original images. We also do a qualitative analysis of the duration attribute at the end of this section.

### Detection accuracy

To evaluate quantitatively the detection accuracy, we use the standard precision (PPV = TP/(TP+FP)) and recall (TPR = TP/(TP+FN)) measures, where TPs are true positives (*i.e.*, mobile objects that were correctly detected by the method), FPs false positives (*i.e.*, other objects detected as mobile objects) and FNs false negatives (*i.e.* mobile objects that have not been detected). These measures are computed at the pixel level.

Second rows of Figures 7.14 and 7.15 show the detection results returned by our proposed method. Black pixels represent detection results. Table 7.8 displays precision and recall results for both datasets. As we do not have a full ground truth for the hybrid dataset (we only have a ground truth for mobile objects that we have added to the image), the recall cannot be computed. We compare our approach to background subtraction method, that has been used in the literature for vehicle detection in very high resolution satellite video data [86]. This method first estimates a background image that corresponds to the static objects present in the scene. Then it detects mobile objects by analysing the difference between the satellite and the background estimation images. For synthetic image experiments, our approach is better than the background subtraction in the case of finding mobile objects correctly but there are some FPs caused by intensity change such as in $I'_1$, $I'_2$ and $I'_6$. It should be noted that one of the detected rectangle is correctly detected while the other one is a wrong detection. For the hybrid dataset experiments, we have added red rectangles for each detected synthetic mobile object. When we compare $I_n$ and $I'_n$ images in Figure 7.15 visually, it is clear that most of the mobile objects are detected. Table 7.9 illustrates that our method has better precision values than the background subtraction technique. However, there are some missing objects caused by background similarity such as in upper left of $I'_3$ and in parking areas such as in $I'_2$ and $I'_4$.

| | Synthetic | | Hybrid |
|---|---|---|---|
| | PPV | TPR | PPV |
| Background Subtraction | 1.0 | 0.58 | 0.57 |
| Our approach | **0.85** | **1.0** | **0.75** |

Table 7.8 – Detection accuracy (precision (PPV) and recall (TPR)) for both synthetic and hybrid datasets. Bold values indicate better results.

### Reconstruction error

The second quantitative evaluation measures the Root Mean Square Error (RMSE) between the reconstructed image and the original one, only for true positives (TPs). We compare our approach to a simple linear temporal interpolation ("LTI") applied to true positives (pixel correctly detected as mobile by our approach).

Table 7.9 reports RMSE computed independently for each image and the averaged RMSE (last row). Bold values display lowest RMSE, and thus more accurate reconstruction.

| | Synthetic | | Hybrid | |
|---|---|---|---|---|
| | LTI | filtered ST | LTI | filtered ST |
| $I_1$ | 1.20 | **0.71** | 4.30 | **0.70** |
| $I_2$ | 2.47 | **0.41** | 4.19 | **0.58** |
| $I_3$ | **0.00** | **0.00** | 5.93 | **0.77** |
| $I_4$ | **0.00** | **0.00** | 5.43 | **0.65** |
| $I_5$ | **0.00** | 0.19 | 5.08 | **0.61** |
| $I_6$ | 9.05 | **0.00** | 2.63 | **0.61** |
| Average | 2.12 | **0.25** | 4.59 | **0.65** |

Table 7.9 – Reconstruction error computed at each time stamp for synthetic and hybrid datasets.

The proposed approach outperforms linear temporal interpolation for all cases with the hybrid dataset. This was expected as the linear interpolation will not be able to take into account changes in illumination (for example presence of shadows). One unsuccessful result of our method is for $I_5$ in synthetic dataset.

This is caused by the rectangle in the upper left. Although it is detected as a mobile object, reconstruction result is not preferable because of the its connectivity with the large object close to it.

#### Qualitative analysis

We visually analyse the precision of the detection for duration attribute threshold values ranging from 1 to 5. Figure 7.16 displays results for the real dataset composed of six Pléiades images. First row shows a zoom into the original images, and second to sixth rows display areas detected as mobile by our approach. Green (resp. red) pixels represent results obtained by max-tree (resp. min-tree).

As max-tree stores maximum values (white) in the leaves, filtering will allow to detect white cars. Similarly min-tree should be useful for detecting black cars. The first threshold result (second row) outputs cars that are parked for the current time stamp, but not for the previous and the next ones. There is also some noisy detection, especially at second time stamp where black shadows are detected as mobile (caused by the change in intensity). The third row detects cars that have been parked for at least two consecutive time stamps, see the white cars (green pixels) for time stamps 5 and 6. As there was no car parked for more than three consecutive days, rows 4 to 6 display only noise.

## 7.3 Spatio-temporal pattern recognition

Spatio-temporal recognition aims to recognize an object or pattern relying on both spatial and temporal information. As a spatio-temporal pattern, we select water movements when tide occurs which is a dynamic class as presented in section 3.1.4.

Coastal monitoring from space is an important environmental application [38]. To detect tide behaviours, we used Sentinel-2 images around Morbihan, France because Morbihan and the Brittany region are well-known for their high tide behaviours. In order to find tide regions, we have calculated spatio-temporal stability of each node, then we applied filtering according to a given threshold $h$ which represents the lower value of spatio-temporal stability. In order to build the

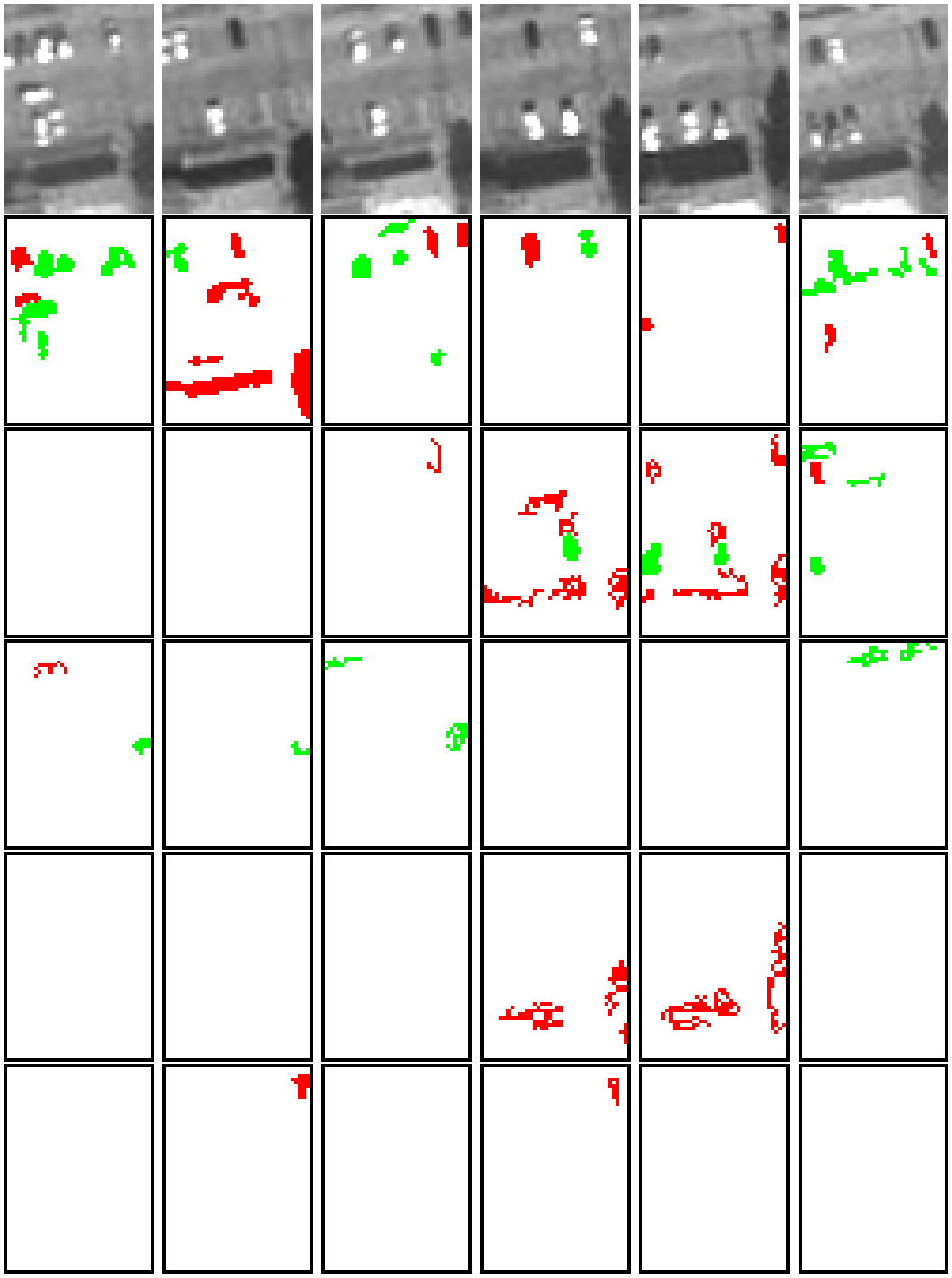


Figure 7.16 – Experiment on a real dataset. First row: original Pléiades images. 2th-6th rows: detection results for duration attribute values ranging from 1 to 5.

reconstructed image $I'$, we assigned to each pixel its node level:

$$I'_t(x) = \min(\lambda \mid x \in C_\lambda(I),\ \mathcal{A}_{st}(C_\lambda(I)) \leq h) \tag{7.10}$$

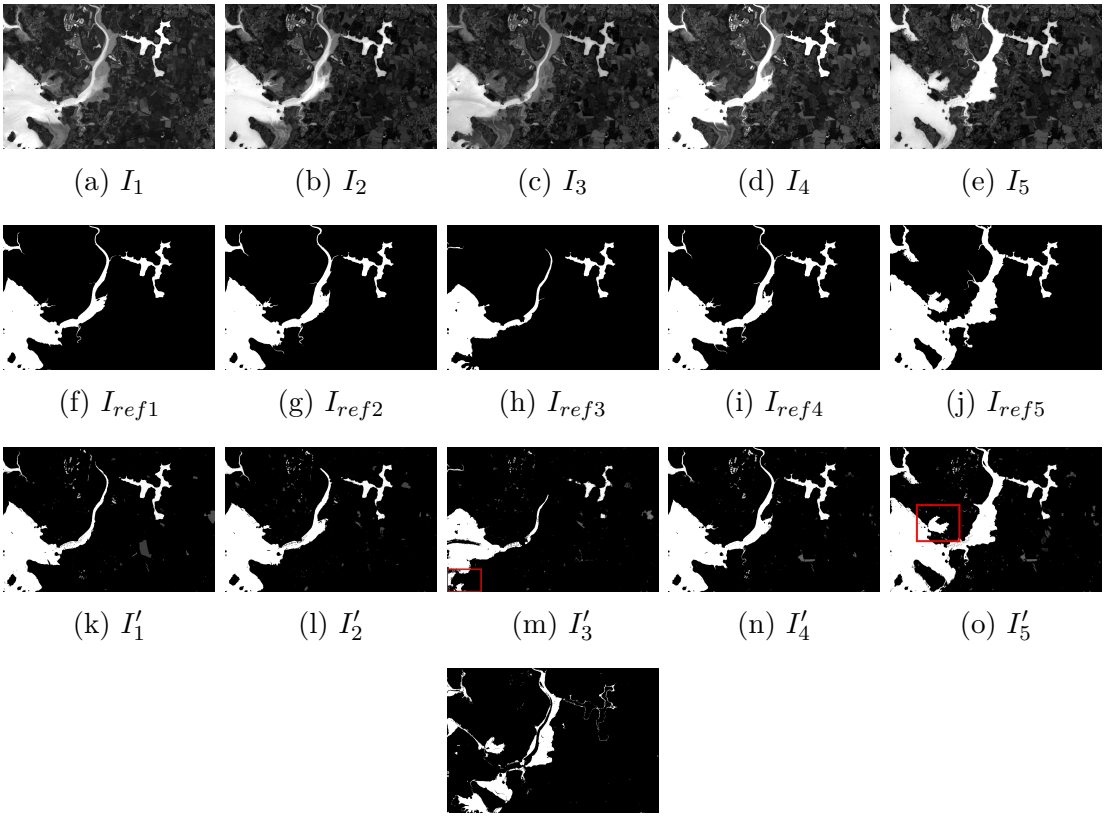
Remaining pixels are set to 0. Let us note that a first pruning step is systematically applied to remove all nodes with stability equal to 0, that correspond to noisy regions appearing at a single time stamp. We limit ourselves to a sample SITS made of small extracts (632 $\times$ 927 pixels) to ease visualization, considering 5 images that were acquired in 2018 with a spatial resolution of 10m. We selected only cloud-free images in this illustrative example. We used top-of-canopy images (level 2A products) provided by the Theia land data center. Since there is no ground truth data for this application, we have created it manually for each date separately. Acquisition dates of these images are given in Table 7.10.

| Image | Acquisition dates |
|---|---|
| $I_1$ | 01/06/2018 |
| $I_2$ | 01/07/2018 |
| $I_3$ | 11/07/2018 |
| $I_4$ | 31/07/2018 |
| $I_5$ | 05/08/2018 |

Table 7.10 – Acquisition dates of Sentinel-2 Images.

We simplify each multispectral image into a grayscale one by computing the Normalized Difference Water Index (NDWI) [97] for each pixel. We recall that NDWI can be calculated from green (G) and near-infrared (NIR) bands of an image as $NDWI = (G - NIR)/(G + NIR)$. Since water pixels appear brighter than other classes on the NDWI image, we used the max-tree in this experiment.

Figure 7.17 illustrates our experiments with Sentinel-2 images. Original images can be seen in the first row. The third row is the result obtained after filtering the space-time tree nodes with a spatio-temporal stability threshold empirically set to $h = 0.4$ and reconstructing the filtered SITS. We can notice some overlap between the detected regions and the water regions visible in the original images. As expected, land and sand regions are not detected as water by our method. For instance, $I_3$ shows some sandy areas on the water at the top left of the image



(a) $I_1$ (b) $I_2$ (c) $I_3$ (d) $I_4$ (e) $I_5$

(f) $I_{ref1}$ (g) $I_{ref2}$ (h) $I_{ref3}$ (i) $I_{ref4}$ (j) $I_{ref5}$

(k) $I'_1$ (l) $I'_2$ (m) $I'_3$ (n) $I'_4$ (o) $I'_5$

(p) $I^{var}$

Figure 7.17 – Intertidal Observation. (a)-(e): Input Sentinel-2 NDWI images; (f)-(j): Ground truth images for each date; (k)-(o): Water extent for each time sample as reported with our method; (p): Intertidal areas from [32].

and this is not detected as water in $I'_3$. We have also added a red rectangle in $I'_3$, to emphasize where the temporal connectivity has helped to ensure a correct detection. Although there is a gap in the spatial domain, this region still belongs to a node with high stability thanks to the successive images. Another red rectangle is displayed for $I'_5$. This part is dramatically growing at that time and correctly detected as an intertidal area with our method. Even if there is no temporal relationship, water pixels are connected in the spatial domain. For the sake of comparison, we provide in the last row the result of a pixel-based variation analysis $I^{var}$ obtained with the method from [32]. We can see in this last image that the tide areas are detected but the method does not provide a result per date. Besides, there are wrongly detected areas caused by intensity changes though time.

Alternatively, we produce pattern spectra with two attributes and select bins according to these attributes as another option. Figure 7.18 shows a 2D pattern spectra using spatio-temporal stability and mean gray value attributes of the Morbihan dataset. Size of the pattern spectra is $10 \times 10$ and bins which are not black correspond to non-zero values (the higher the brighter). Since we aim to find objects with high gray values and low stability values, we select an appropriate bin and extract the relevant nodes in that bin. We select the bin in the coordinates (8,9) as it corresponds to a local maximum among low stability / high mean gray value bins. Then, we reconstruct the set of images according to obtained nodes as in equation 7.10.

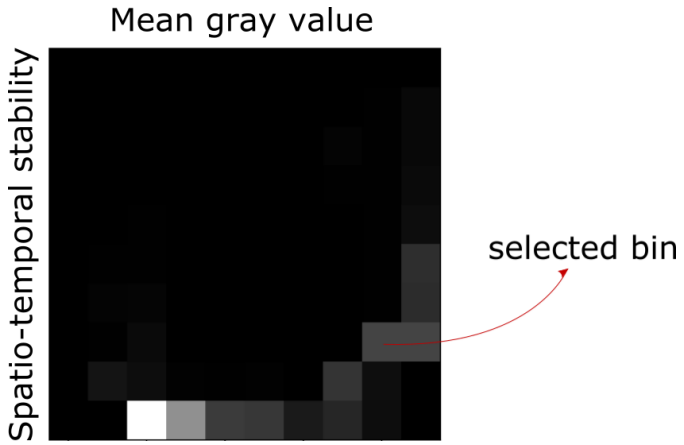
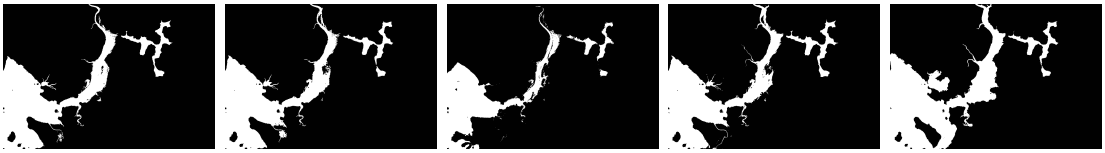



Figure 7.18 – 2D pattern spectra of Morbihan dataset.

Figure 7.19 shows the reconstructed images from the selected bin. Compared to the results in Figure 7.17, these results have no false negative but waters in the upper left could not be detected. It should be noted that we have extracted only one bin to propose a straightforward method. If we had used multiple bins, results would have been better.

In order to obtain quantitative results, we have used the ground truth images from Figure 7.17. We report $F_1$ scores in Table 7.11 for each date and both methods which are filtering and pattern spectra. In order to compare with the state-of-the-art reference which does not give temporal information, we have summarized our



(a) $I'_1$ (b) $I'_2$ (c) $I'_3$ (d) $I'_4$ (e) $I'_5$

Figure 7.19 – Reconstructed images from pattern spectra.

ground truth images according to the existence of the tide and we have calculated $F_1$ score for this summary. According to this quantitative analysis, pattern spectra method is much better than the others.

| Pattern spectra | Filtering | Standard Dev. |
|---|---|---|
| **0.97** | 0.83 | 0.69 |

Table 7.11 – Quantitative evaluation of tide observation using the $F_1$ measure.

# Chapter 8

# Conclusions and Open Challenges

In this chapter, we will conclude the thesis and give some future perspectives. Summary of this thesis will be first recalled in section 8.1. Then, section 8.2 will discuss open challenges about the tree representations for SITS which could include seminal directions for the researchers who are willing to explore further on this topic.

## 8.1 Conclusions

While morphological hierarchies have been largely explored in the literature (both from a theoretical and application point of view), their extension to time series remains barely addressed. As a first contribution of this thesis, we have discussed how to build such hierarchies for image sequences. More precisely, we have distinguished between spatial, temporal and spatial-temporal hierarchies, that can be built using various strategies. We also investigated the case of streaming, where new images or frames are appended to the series, thus needing to update the underlying tree structure in a scalable manner. We proposed streaming algorithms for each construction strategy.

As a second contribution, we have then compared morphological hierarchies to determine the better tree representation for image sequence analysis. Our goal was to select the most relevant and efficient hierarchical representation for subsequent studies. Comparison of trees requires them to lie in the same domain. To do so,

we have proposed some projection methods for space-time trees in order to make them comparable with the trees obtained with spatial and temporal hierarchy. We also proposed a continuous connectivity rule to build a space-time tree with low complexity and low cost which provides interesting filtering capabilities. In our experiments, we have used some gray-scale satellite images and shown that the temporal information provides a less complex hierarchical structure when projected in space or in time. This result encourages us to focus on space-time tree rather than the other strategies.

As a third contribution, we have introduced a space-time multi-scale analysis method by using the space-time tree for satellite image time series. This method has two main advantages. First, it is not necessary to have prior information about the scene. In other words, it is an unsupervised approach. Second, the method relies on object-based information which has several advantages compared to pixel based approaches. More precisely, we have reported how to calculate attributes from a space-time tree. We have proposed novel attributes related to time in order to use them with space-time tree such as spatio-temporal stability, beginning/ending time, duration, node centroid, amplitude and time of max/min value. In order to extract features from a space-time tree, we have applied filtering and produced pattern spectra with these attributes.

Finally, proposed methods enable us to process SITS for some important EO applications. Land cover mapping, change detection and spatio-temporal pattern recognition applications have been evaluated on different kinds of datasets and shown good results compared to the state-of-the-art.

## 8.2 Open Challenges

Although we have proposed several methods for building and using trees for SITS, there are still open challenges which could deserve future research. Firstly, it could be considered to extend the space-time tree applications for multispectral images rather than only gray-scale input. Thus, a 4-dimensional dataset will be obtained with some spectral channels to extract novel attributes. One option

is to apply ordering methods to the spectral channels as we did for the spatial hierarchy strategy. Another way could be processing each channel independently and merging their information.

Among future works, the capability of space-time tree with continuous connectivity for real remote sensing based applications such as land-cover mapping, pattern recognition and change detection might be investigated. This connectivity connects similar objects from distant time stamps. Therefore, our proposed attributes could be more efficient since they provide a way to overcome intensity fluctuations through time. In addition, it could be possible to consider different kind of attributes (such as crop cycle) with this connectivity.

Another open topic is the comparison of trees structures of images from different dates. Indeed, images of the same scene have similar topologies, in other words the similar shapes which are stored in the trees. Although pixel intensity changes, a shape based comparison of the trees could be promising for unsupervised change detection application. Moreover, space-time tree of the same scene but from different time period could be compared in order to analyze periodical dynamics. Comparison can be done by using filtering and pattern spectra tools.

# Bibliography


[1] A. Aijazi, P. Checchin, and L. Trassoudaine. "Automatic removal of imperfections and change detection for accurate 3D urban cartography by classification and incremental updating". In: *Remote Sensing* 5.8 (2013), pp. 3701–3728 (cit. on p. 118).

[2] V. Alberga. "Similarity measures of remotely sensed multi-sensor images for change detection applications". In: *Remote Sensing* 1.3 (2009), pp. 122–143 (cit. on p. 43).

[3] A. Alonso-González, C. López-Martınez, and P. Salembier. "PolSAR time series processing with binary partition trees". In: *Transactions on Geoscience and Remote Sensing* 52.6 (2014), pp. 3553–3567 (cit. on pp. 46, 58).

[4] A. Alonso-González, C. López-Martınez, and P. Salembier. "Filtering and segmentation of polarimetric SAR data based on binary partition trees". In: *IEEE Transactions on Geoscience and Remote Sensing* 50.2 (2012), pp. 593–605 (cit. on pp. 26, 37).

[5] J. Angulo. "Morphological colour operators in totally ordered lattices based on distances: Application to image filtering, enhancement and analysis". In: *Computer vision and image understanding* 107.1-2 (2007), pp. 56–73 (cit. on p. 51).

[6] E. Aptoula and S. Lefèvre. "A comparative study on multivariate mathematical morphology". In: *Pattern Recognition* 40.11 (2007), pp. 2914–2929 (cit. on pp. 46, 48, 55).

[7] E. Aptoula and S. Lefèvre. "On lexicographical ordering in multivariate mathematical morphology". In: *Pattern Recognition Letters* 29.2 (2008), pp. 109–118 (cit. on p. 50).

[8] T. Asplund, A. Serna, B. Marcotegui, R. Strand, and C. L. L. Hendriks. "Mathematical Morphology on Irregularly Sampled Data Applied to Segmentation of 3D Point Clouds of Urban Scenes". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2019, pp. 375–387 (cit. on p. 36).

[9] A. Bailly, L. Chapel, R. Tavenard, and G. Camps-Valls. "Nonlinear time-series adaptation for land cover classification". In: *IEEE Geoscience and Remote Sensing Letters* 14.6 (2017), pp. 896–900 (cit. on p. 43).

[10] M. Belgiu and O. Csillik. "Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis". In: *Remote sensing of environment* 204 (2018), pp. 509–523 (cit. on p. 45).

[11] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. G. Pensa, and S. Dupuy. "M3-Fusion: A Deep Learning Architecture for Multiscale Multimodal Multitemporal Satellite Data Fusion". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.12 (2018), pp. 4939–4949 (cit. on p. 45).

[12] L. Bergamasco, S. Saha, F. Bovolo, and L. Bruzzone. "Unsupervised change-detection based on convolutional-autoencoder feature extraction". In: *Image and Signal Processing for Remote Sensing XXV*. Vol. 11155. International Society for Optics and Photonics. 2019, p. 1115510 (cit. on p. 118).

[13] C. Berge and E. Minieka. *Graphs and hypergraphs*. North-Holland publishing company Amsterdam, 1973 (cit. on pp. 10, 11).

[14] M. Berger, J. Moreno, J. A. Johannessen, P. F. Levelt, and R. F. Hanssen. "ESA's sentinel missions in support of Earth system science". In: *Remote Sensing of Environment* 120 (2012), pp. 84–90 (cit. on pp. vi, 3, 38).

[15] B. Bescós, J. Facil, J. Civera, and J. Neira. "Removing dynamic objects from 3d maps using geometry and learning". In: *IEEE Conf. on Intelligent Robots and Systems (IROS) workshop of learning for localization and mapping*. 2017 (cit. on p. 118).

[16] K. Bhardwaj, S. Patra, and L. Bruzzone. "Threshold-Free Attribute Profile for Classification of Hyperspectral Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.10 (2019), pp. 7731–7742 (cit. on p. 70).

[17] P. Bosilj, E. Aptoula, S. Lefèvre, and E. Kijak. "Retrieval of Remote Sensing Images with Pattern Spectra Descriptors". In: *ISPRS International Journal of Geo-Information* 5.12 (2016), p. 228 (cit. on pp. vii, 4, 31).

[18] P. Bosilj, B. B. Damodaran, E. Aptoula, M. Dalla Mura, and S. Lefèvre. "Attribute Profiles from Partitioning Trees". In: *International Symposium on Mathematical Morphology (ISMM)*. Springer. 2017, pp. 381–392 (cit. on pp. 12, 28).

[19] P. Bosilj, E. Kijak, and S. Lefèvre. "Beyond MSER: Maximally stable regions using tree of shapes". In: *BMCV*. 2015 (cit. on p. 33).

[20] P. Bosilj, E. Kijak, and S. Lefèvre. "Partition and Inclusion Hierarchies of Images: A Comprehensive Survey". In: *Journal of Imaging* 4.2 (2018), p. 33 (cit. on pp. 13, 71).

[21] P. Bosilj, S. Lefèvre, and E. Kijak. "Hierarchical image representation simplification driven by region complexity". In: *International Conference on Image Analysis and Processing*. Springer. 2013, pp. 562–571 (cit. on p. 12).

[22] N. Boutry, T. Géraud, et al. "Connected filters on generalized shape-Spaces". In: *Pattern Recognition Letters* 128 (2019), pp. 348–354 (cit. on p. 91).

[23] F. Bracci, U. Hillenbrand, Z.-C. Marton, and M. H. Wilkinson. "On the Use of the Tree Structure of Depth Levels for Comparing 3D Object Views". In: *International Conference on Computer Analysis of Images and Patterns*. 2017, pp. 251–263 (cit. on p. 54).

[24] L. Breiman. "Random forests". In: *Machine learning* 45.1 (2001), pp. 5–32 (cit. on p. 98).

[25] I. T. Bueno, F. W. Acerbi Júnior, E. M. Silveira, J. M. Mello, L. M. Carvalho, L. R. Gomide, K. Withey, and J. R. S. Scolforo. "Object-based change detection in the Cerrado biome using landsat time series". In: *Remote Sensing* 11.5 (2019), p. 570 (cit. on p. 41).

[26] D. Bueso, M. Piles, and G. Camps-Valls. "Nonlinear PCA for Spatio-Temporal Analysis of Earth Observation Data". In: *IEEE Transactions on Geoscience and Remote Sensing* (2020), pp. 5752–5763 (cit. on p. 42).

[27] E. L. Bullock, C. E. Woodcock, and C. E. Holden. "Improved change monitoring using an ensemble of time series algorithms". In: *Remote Sensing of Environment* 238 (2020), p. 111165 (cit. on p. 42).

[28] J. B. Campbell and R. H. Wynne. *Introduction to remote sensing*. Guilford Press, 2011 (cit. on pp. vi, 3).

[29] E. Carlinet and T. Géraud. "A comparative review of component tree computation algorithms". In: *Transactions on Image Processing* 23.9 (2014), pp. 3885–3895 (cit. on pp. 12, 61).

[30] E. Carlinet and T. Géraud. "MToS: A tree of shapes for multivariate images". In: *Transactions on Image Processing* 24.12 (2015), pp. 5330–5342 (cit. on pp. 21, 48, 55, 59).

[31] O. A. Carvalho Júnior, R. F. Guimarães, A. R. Gillespie, N. C. Silva, and R. A. Gomes. "A new approach to change vector analysis using distance and similarity measures". In: *Remote Sensing* 3.11 (2011), pp. 2473–2493 (cit. on p. 43).

[32] J. Catalao and G. Nico. "Multitemporal backscattering logistic analysis for intertidal bathymetry". In: *IEEE Transactions on Geoscience and Remote Sensing* 55.2 (2016), pp. 1066–1073 (cit. on p. 126).

[33] G. Cavallaro, M. Dalla Mura, J. A. Benediktsson, and A. Plaza. "Remote sensing image classification using attribute filters defined over the tree of shapes". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.7 (2016), pp. 3899–3911 (cit. on pp. vii, 4, 12, 29).

[34] M. E. Celebi. "Distance measures for reduced ordering-based vector filters". In: *IET image processing* 3.5 (2009), pp. 249–260 (cit. on p. 51).

[35] D. Cerra, R. Müller, and P. Reinartz. "Cloud removal in image time series through unmixing". In: *International Workshop on the Analysis of Multi-temporal Remote Sensing Images (Multi-Temp)*. IEEE. 2015, pp. 1–4 (cit. on p. 37).

[36] N. Champion. "Automatic cloud detection from multi-temporal satellite images: Towards the use of pléiades time series". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 39 (2012), B3 (cit. on p. 37).

[37] Y. Chi, R. R. Muntz, S. Nijssen, and J. N. Kok. "Frequent subtree mining–an overview". In: *Fundamenta Informaticae* 66.1-2 (2005), pp. 161–198 (cit. on p. 70).

[38] S. Chu, L. Cheng, X. Ruan, Q. Zhuang, X. Zhou, M. Li, and Y. Shi. "Technical Framework for Shallow-Water Bathymetry With High Reliability and No Missing Data Based on Time-Series Sentinel-2 Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.11 (2019), pp. 8745–8763 (cit. on p. 123).

[39] F. Cian, M. Marconcini, and P. Ceccato. "Normalized Difference Flood Index for rapid flood mapping: Taking advantage of EO big data". In: *Remote sensing of environment* 209 (2018), pp. 712–730 (cit. on pp. 114, 115).

[40] M. Couprie and G. Bertrand. "Topological gray-scale watershed transformation". In: *Vision Geometry Vi*. Vol. 3168. International Society for Optics and Photonics. 1997, pp. 136–147 (cit. on p. 21).

[41] M. Dalla Mura, J. Atli Benediktsson, B. Waske, and L. Bruzzone. "Extended profiles with morphological attribute filters for the analysis of hyperspectral data". In: *International Journal of Remote Sensing* 31.22 (2010), pp. 5975–5991 (cit. on pp. 30, 54).

[42] M. Dalla Mura, J. A. Benediktsson, and L. Bruzzone. "Self-dual attribute profiles for the analysis of remote sensing images". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2011, pp. 320–330 (cit. on p. 30).

[43] M. Dalla Mura, J. A. Benediktsson, B. Waske, and L. Bruzzone. “Morphological attribute profiles for the analysis of very high resolution images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.10 (2010), pp. 3747–3762 (cit. on pp. vii, 4, 29).

[44] S. Dasgupta. “A cost function for similarity-based hierarchical clustering”. In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. 2016, pp. 118–127 (cit. on p. 71).

[45] B. Demir and L. Bruzzone. “Histogram-based attribute profiles for classification of very high resolution remote sensing images”. In: *IEEE Transactions on Geoscience and Remote Sensing* 54.4 (2016), pp. 2096–2107 (cit. on p. 30).

[46] M. Donoser, H. Riemenschneider, and H. Bischof. “Shape guided maximally stable extremal region (MSER) tracking”. In: *International Conference on Pattern Recognition (ICPR)*. IEEE. 2010, pp. 1800–1803 (cit. on p. 33).

[47] A. Dufour, O. Tankyevych, B. Naegel, H. Talbot, C. Ronse, J. Baruthio, P. Dokládal, and N. Passat. “Filtering and segmentation of 3D angiographic data: Advances based on mathematical morphology”. In: *Medical image analysis* 17.2 (2013), pp. 147–164 (cit. on p. 29).

[48] L. Eklundh and P. Jönsson. “TIMESAT for processing time-series data from satellite sensors for land surface monitoring”. In: *Multitemporal Remote Sensing*. Springer, 2016, pp. 177–194 (cit. on pp. 37, 83, 85).

[49] S. Y. Elhabian, K. M. El-Sayed, and S. H. Ahmed. “Moving object detection in spatial domain using background removal techniques-state-of-art”. In: *Recent patents on computer science* 1.1 (2008), pp. 32–54 (cit. on p. 118).

[50] N. Falco, M. Dalla Mura, F. Bovolo, J. A. Benediktsson, and L. Bruzzone. “Change detection in VHR images based on morphological attribute profiles”. In: *IEEE Geoscience and Remote Sensing Letters* 10.3 (2013), pp. 636–640 (cit. on pp. vii, 4, 109, 114, 115).

[51] T. Fu. “A review on time series data mining”. In: *Engineering Applications of Artificial Intelligence* (2011), pp. 164–181 (cit. on p. 43).

[52] V. S. F. Garnot, L. Landrieu, S. Giordano, and N. Chehata. "Satellite Image Time Series Classification with Pixel-Set Encoders and Temporal Self-Attention". In: *arXiv preprint arXiv:1911.07757* (2019) (cit. on p. 43).

[53] Y. J. E. Gbodjo, D. Ienco, and L. Leroux. "Toward Spatio–Spectral Analysis of Sentinel-2 Time Series Data for Land Cover Mapping". In: *IEEE Geoscience and Remote Sensing Letters* 17.2 (2019), pp. 307–311 (cit. on pp. 46, 101, 102).

[54] J. Gehrung, M. Hebel, M. Arens, and U. Stilla. "An approach to extract moving objects from MLS data using a volumetric background representation". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), p. 107 (cit. on p. 118).

[55] T. Géraud, E. Carlinet, S. Crozet, and L. Najman. "A quasi-linear algorithm to compute the tree of shapes of nD images". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2013, pp. 98–110 (cit. on pp. 12, 15).

[56] P. Ghamisi, M. Dalla Mura, and J. A. Benediktsson. "A survey on spectral–spatial classification techniques based on attribute profiles". In: *IEEE Transactions on Geoscience and Remote Sensing* 53.5 (2014), pp. 2335–2353 (cit. on pp. vii, 4, 24).

[57] P. Ghamisi, R. Souza, J. A. Benediktsson, X. X. Zhu, L. Rittner, and R. A. Lotufo. "Extinction profiles for the classification of remote sensing data". In: *Transactions on Geoscience and Remote Sensing* 54.10 (2016), pp. 5631–5645 (cit. on p. 30).

[58] D. Ghoshdastidar, M. Perrot, and U. von Luxburg. "Foundations of Comparison-Based Hierarchical Clustering". In: *Advances in Neural Information Processing Systems*. 2019, pp. 7454–7464 (cit. on p. 71).

[59] L. Giustarini, R. Hostache, P. Matgen, G. J.-P. Schumann, P. D. Bates, and D. C. Mason. "A change detection approach to flood mapping in urban areas using TerraSAR-X". In: *IEEE Transactions on Geoscience and Remote Sensing* 51.4 (2013), pp. 2417–2430 (cit. on p. 110).

[60] C. Gómez, J. C. White, and M. A. Wulder. "Optical remotely sensed time series data for land cover classification: A review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 116 (2016), pp. 55–72 (cit. on p. 42).

[61] M. Götz, G. Cavallaro, T. Géraud, M. Book, and M. Riedel. "Parallel computation of component trees on distributed memory machines". In: *Transactions on Parallel and Distributed Systems* 29.11 (2018), pp. 2582–2598 (cit. on p. 55).

[62] F. Guiotte, S. Lefèvre, and T. Corpetti. "Attribute Filtering of Urban Point Clouds Using Max-Tree on Voxel Data". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2019, pp. 391–402 (cit. on p. 58).

[63] F. Guttler, D. Ienco, J. Nin, M. Teisseire, and P. Poncelet. "A graph-based approach to detect spatiotemporal dynamics in satellite image time series". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 130 (2017), pp. 92–107 (cit. on p. 46).

[64] T. Guyet and H. Nicolas. "Long term analysis of time series of satellite images". In: *Pattern Recognition Letters* 70 (2016), pp. 17–23 (cit. on pp. vi, 3, 43).

[65] O. Hagolle, M. Huc, D. V. Pascual, and G. Dedieu. "A multi-temporal method for cloud detection, applied to FORMOSAT-2, VEN$\mu$S, LANDSAT and SENTINEL-2 images". In: *Remote Sensing of Environment* 114.8 (2010), pp. 1747–1755 (cit. on p. 105).

[66] M. Haklay and P. Weber. "Openstreetmap: User-generated street maps". In: *IEEE Pervas Comput* 7.4 (2008), pp. 12–18 (cit. on p. 105).

[67] E. Hamunyela, J. Reiche, J. Verbesselt, and M. Herold. "Using space-time features to improve detection of forest disturbances from Landsat time series". In: *Remote Sensing* 9.6 (2017), p. 515 (cit. on p. 44).

[68] J. Havel, F. Merciol, and S. Lefèvre. "Efficient tree construction for multiscale image representation and processing". In: *Journal of Real-Time Image Processing* (2016), pp. 1–18 (cit. on p. 55).

[69] P. Héas and M. Datcu. "Modeling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning". In: *IEEE Transactions on Geoscience and Remote Sensing* 43.7 (2005), pp. 1635–1647 (cit. on p. 39).

[70] Z. Huang, Y. Zhang, Q. Li, T. Zhang, N. Sang, and H. Hong. "Progressive dual-domain filter for enhancing and denoising optical remote-sensing images". In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 759–763 (cit. on p. 37).

[71] M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley. "Change detection from remotely sensed images: From pixel-based to object-based approaches". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 80 (2013), pp. 91–106 (cit. on p. 118).

[72] D. Ienco, R. Gaetano, C. Dupaquier, and P. Maurel. "Land cover classification via multitemporal spatial data by deep recurrent neural networks". In: *IEEE Geoscience and Remote Sensing Letters* 14.10 (2017), pp. 1685–1689 (cit. on p. 43).

[73] J. Inglada and S. Garrigues. "Land-cover maps from partially cloudy multitemporal image series: Optimal temporal sampling and cloud removal". In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2010, pp. 3070–3073 (cit. on p. 37).

[74] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes. "Operational high resolution land cover map production at the country scale using satellite image time series". In: *Remote Sensing* 9.1 (2017), p. 95 (cit. on pp. 41, 96).

[75] R. Interdonato, D. Ienco, R. Gaetano, and K. Ose. "DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 149 (2019), pp. 91–104 (cit. on p. 45).

[76] J. R. Jensen et al. *Introductory digital image processing: a remote sensing perspective*. Ed. 2. Prentice-Hall Inc., 1996 (cit. on pp. v, 1).

[77] S. Ji, C. Zhang, A. Xu, Y. Shi, and Y. Duan. "3D convolutional neural networks for crop classification with multi-temporal remote sensing images". In: *Remote Sensing* 10.1 (2018), p. 75 (cit. on p. 45).

[78] R. Jones. "Component trees for image filtering and segmentation". In: *Proceedings of the 1997 Workshop on Nonlinear Signal and Image Processing, Mackinac Island*. 1997 (cit. on pp. vii, 4).

[79] R. Jones. "Connected filtering and segmentation using component trees". In: *Computer Vision and Image Understanding* 75.3 (1999), pp. 215–228 (cit. on p. 14).

[80] P. Jonsson and L. Eklundh. "Seasonality extraction by function fitting to time-series of satellite sensor data". In: *IEEE transactions on Geoscience and Remote Sensing* 40.8 (2002), pp. 1824–1832 (cit. on p. 43).

[81] A. Julea, N. Méger, P. Bolon, C. Rigotti, M.-P. Doin, C. Lasserre, E. Trouvé, and V. N. Lazarescu. "Unsupervised spatiotemporal mining of satellite image time series using grouped frequent sequential patterns". In: *IEEE Transactions on Geoscience and Remote Sensing* 49.4 (2011), pp. 1417–1430 (cit. on p. 44).

[82] E. Kalinicheva, D. Ienco, J. Sublime, and M. Trocan. "Unsupervised Change Detection Analysis in Satellite Image Time Series using Deep Learning Combined with Graph-Based Approaches". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 1450–1466 (cit. on p. 45).

[83] L. Khiali, M. Ndiath, S. Alleaume, D. Ienco, K. Ose, and M. Teisseire. "Detection of spatio-temporal evolutions on multi-annual satellite image time series: A clustering based approach". In: *International Journal of Applied Earth Observation and Geoinformation* 74 (2019), pp. 103–119 (cit. on p. 44).

[84] B. R. Kiran and J. Serra. "Fusion of ground truths and hierarchies of segmentations". In: *Pattern Recognition Letters* 47 (2014), pp. 63–71 (cit. on p. 28).

[85] F. N. Kiwanuka, G. K. Ouzounis, and M. H. Wilkinson. "Surface-area-based attribute filtering in 3d". In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. Springer. 2009, pp. 70–81 (cit. on p. 22).

[86] G. Kopsiaftis and K. Karantzalos. "Vehicle detection and traffic density monitoring from very high resolution satellite video data". In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2015, pp. 1881–1884 (cit. on p. 121).

[87] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. "Deep learning classification of land cover and crop types using remote sensing data". In: *IEEE Geoscience and Remote Sensing Letters* 14.5 (2017), pp. 778–782 (cit. on p. 41).

[88] C. Le Men, A. Julea, N. Méger, M. Datcu, P. Bolon, and H. Maıtre. "Radiometric evolution classification in a high resolution satellite image time series (STIS)". In: *Proc. of the 5th Conference on Image Information Mining: pursuing automation of geospatial intelligence for environment and security (ESA-EUSC 2008), ESRIN-Frascati, Italy*. Vol. 23. 2008, pp. 35–56 (cit. on p. 43).

[89] X. Li, L. Wang, Q. Cheng, P. Wu, W. Gan, and L. Fang. "Cloud removal in remote sensing images using nonnegative matrix factorization and error correction". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 148 (2019), pp. 103–113 (cit. on p. 118).

[90] K. Litomisky and B. Bhanu. "Removing moving objects from point cloud scenes". In: *International Workshop on Depth Image Analysis and Applications*. Springer. 2012, pp. 50–58 (cit. on p. 118).

[91] J. Liu, W. Zhu, C. Atzberger, A. Zhao, Y. Pan, and X. Huang. "A phenology-based method to map cropping patterns under a wheat-maize rotation using remotely sensed time-series data". In: *Remote Sensing* 10.8 (2018), p. 1203 (cit. on p. 43).

[92] S. Liu, D. Marinelli, L. Bruzzone, and F. Bovolo. "A Review of Change Detection in Multitemporal Hyperspectral Images: Current Techniques, Applications, and Challenges". In: *IEEE Geoscience and Remote Sensing Magazine* 7.2 (2019), pp. 140–158 (cit. on p. 118).

[93] L. A. Manfré, E. Hirata, J. B. Silva, E. J. Shinohara, M. A. Giannotti, A. P. C. Larocca, and J. A. Quintanilha. "An analysis of geospatial technologies for risk and natural disaster management". In: *ISPRS International Journal of Geo-Information* 1.2 (2012), pp. 166–185 (cit. on pp. v, 2).

[94] P. Maragos. "Pattern spectrum and multiscale shape representation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11.7 (1989), pp. 701–716 (cit. on p. 31).

[95] J. Matas, O. Chum, M. Urban, and T. Pajdla. "Robust wide-baseline stereo from maximally stable extremal regions". In: *Image and Vision Computing* 22.10 (2004), pp. 761–767 (cit. on p. 33).

[96] V. Maus, G. Câmara, R. Cartaxo, A. Sanchez, F. M. Ramos, and G. R. De Queiroz. "A time-weighted dynamic time warping method for land-use and land-cover mapping". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 9.8 (2016), pp. 3729–3739 (cit. on p. 43).

[97] S. K. McFeeters. "The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features". In: *International Journal of Remote Sensing* 17.7 (1996), pp. 1425–1432 (cit. on p. 125).

[98] N. Méger, C. Rigotti, C. Pothier, T. Nguyen, F. Lodge, L. Gueguen, R. Andréoli, M.-P. Doin, and M. Datcu. "Ranking evolution maps for Satellite Image Time Series exploration: application to crustal deformation and environmental monitoring". In: *Data Mining and Knowledge Discovery* 33.1 (2019), pp. 131–167 (cit. on pp. vi, 3, 36, 44).

[99] F. Merciol, T. Balem, and S. Lefèvre. "Efficient and Large-Scale Land Cover Classification using Multiscale Image Analysis". In: *ESA Conference on Big Data from Space (BiDS)*. 2017 (cit. on p. 109).

[100] F. Merciol, L. Chapel, and S. Lefèvre. "Hyperspectral image representation through alpha-trees". In: *ESA-EUSC-JRC 9th Conference on Image Information Mining*. 2014, pp. 37–40 (cit. on p. 12).

[101] F. Merciol, L. Faucqueur, B. B. Damodaran, P.-Y. Rémy, B. Desclée, F. Dazin, S. Lefèvre, A. Masse, and C. Sannier. "Geobia at the terapixel scale: Toward efficient mapping of small woody features from heterogeneous vhr scenes". In: *ISPRS International Journal of Geo-Information* 8.1 (2019), p. 46 (cit. on p. 41).

[102] F. Merciol and S. Lefèvre. "Buffering hierarchical representation of color video streams for interactive object selection". In: *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer. 2015, pp. 864–875 (cit. on pp. vii, 4, 45, 46).

[103] F. Merciol and S. Lefèvre. "Fast image and video segmentation based on alpha-tree multiscale representation". In: *Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on*. IEEE. 2012, pp. 336–342 (cit. on pp. 28, 45).

[104] F. Merciol, M.-T. Pham, D. Santana, A. Masse, and C. Sannier. "BROCELIANDE: a comparative study of attribute profiles and feature profiles from different attributes". In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLIII-B3. 2020, pp. 1371–1377 (cit. on p. 30).

[105] C. C. Mkalawa. "Analyzing Dar es Salaam Urban Change and its Spatial Pattern". In: *Int. J. of Urban Planning and Transp.* 31.1 (2016), pp. 1138–50 (cit. on p. 104).

[106] P. Monasse and F. Guichard. "Fast computation of a contrast-invariant image representation". In: *IEEE Transactions on Image Processing* 9.5 (2000), pp. 860–872 (cit. on pp. 12, 15).

[107] U. Moschini, A. Meijster, and M. H. Wilkinson. "A hybrid shared-memory parallel max-tree algorithm for extreme dynamic-range images". In: *IEEE transactions on Pattern Analysis and Machine Intelligence* 40.3 (2018), pp. 513–526 (cit. on p. 55).

[108] L. Mou, L. Bruzzone, and X. X. Zhu. “Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.2 (2018), pp. 924–935 (cit. on p. 45).

[109] M. Müller. “Dynamic time warping”. In: *Information Retrieval for Music and Motion* (2007), pp. 69–84 (cit. on p. 51).

[110] B. Naegel and N. Passat. “Component-trees and multi-value images: A comparative study”. In: *International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing*. 2009, pp. 261–271 (cit. on p. 48).

[111] B. Naegel, N. Passat, N. Boch, and M. Kocher. “Segmentation using vector-attribute filters: methodology and application to dermatological imaging”. In: *International Symposium on Mathematical Morphology (ISMM)*. Vol. 1. INPE. 2006, pp. 239–250 (cit. on p. 28).

[112] L. Najman and M. Couprie. “Building the component tree in quasi-linear time”. In: *IEEE Transactions on Image Processing* 15.11 (2006), pp. 3531–3539 (cit. on p. 20).

[113] L. Najman and J. Cousty. “A graph-based mathematical morphology reader”. In: *Pattern Recognition Letters* 47 (2014), pp. 3–17 (cit. on p. 71).

[114] L. Najman, F. Meyer, et al. “A short tour of mathematical morphology on edge and vertex weighted graphs”. In: *Image Processing and Analysis with Graphs: Theory and Practice* (2012), pp. 141–174 (cit. on p. 71).

[115] D. Nistér and H. Stewénius. “Linear time maximally stable extremal regions”. In: *European Conference on Computer Vision* (2008), pp. 183–196 (cit. on pp. vii, 4).

[116] G. K. Ouzounis and L. Gueguen. “Interactive collection of training samples from the max-tree structure”. In: *IEEE International Conference on Image Processing (ICIP)*. IEEE. 2011, pp. 1449–1452 (cit. on p. 90).

[117] G. Palou and P. Salembier. “Hierarchical video representation with trajectory binary partition tree”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2013, pp. 2099–2106 (cit. on pp. vii, 4, 45, 46).

[118] C. Paris, L. Bruzzone, and D. Fernández-Prieto. "A novel approach to the unsupervised update of land-cover maps by classification of time series of multispectral images". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.7 (2019), pp. 4259–4277 (cit. on p. 44).

[119] N. Passat, B. Naegel, F. Rousseau, M. Koob, and J.-L. Dietemann. "Interactive segmentation based on component-trees". In: *Pattern Recognition* 44.10-11 (2011), pp. 2539–2554 (cit. on p. 29).

[120] M. Pedergnana, P. R. Marpu, M. Dalla Mura, J. A. Benediktsson, and L. Bruzzone. "A novel technique for optimal feature selection in attribute profiles based on genetic algorithms". In: *IEEE Transactions on Geoscience and Remote Sensing* 51.6 (2013), pp. 3514–3528 (cit. on p. 26).

[121] C. Pelletier, G. I. Webb, and F. Petitjean. "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series". In: *Remote Sensing* 11.5 (2019), p. 523 (cit. on pp. vi, 3, 43).

[122] B. Perret and C. Collet. "Connected image processing with multivariate attributes: An unsupervised Markovian classification approach". In: *Computer Vision and Image Understanding* 133 (2015), pp. 1–14 (cit. on p. 21).

[123] B. Perret, J. Cousty, S. J. F. Guimaraes, and D. S. Maia. "Evaluation of hierarchical watersheds". In: *IEEE Transactions on Image Processing* 27.4 (2017), pp. 1676–1688 (cit. on p. 70).

[124] B. Perret, J. Cousty, O. Tankyevych, H. Talbot, and N. Passat. "Directed connected operators: Asymmetric hierarchies for image filtering and segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.6 (2014), pp. 1162–1176 (cit. on pp. 29, 71).

[125] B. Perret, S. Lefèvre, C. Collet, and É. Slezak. "Hyperconnections and hierarchical representations for grayscale and multiband image processing". In: *IEEE Transactions on Image Processing* 21.1 (2011), pp. 14–27 (cit. on pp. 50, 109).

[126] M. Pesaresi and J. A. Benediktsson. "A new approach for the morphological segmentation of high-resolution satellite imagery". In: *IEEE Transactions on Geoscience and Remote Sensing* 39.2 (2001), pp. 309–320 (cit. on p. 29).

[127] M. Pesaresi, C. Corbane, A. Julea, A. J. Florczyk, V. Syrris, and P. Soille. "Assessment of the added-value of sentinel-2 for detecting built-up areas". In: *Remote Sensing* 8.4 (2016), p. 299 (cit. on p. 41).

[128] F. Petitjean, C. Kurtz, N. Passat, and P. Gançarski. "Spatio-temporal reasoning for the classification of satellite image time series". In: *Pattern Recognition Letters* 33.13 (2012), pp. 1805–1815 (cit. on p. 44).

[129] F. Petitjean and J. Weber. "Efficient satellite image time series analysis under time warping". In: *IEEE Geoscience and Remote Sensing Letters* 11.6 (2014), pp. 1143–1147 (cit. on pp. vi, 3, 37, 43).

[130] M.-T. Pham, E. Aptoula, and S. Lefèvre. "Feature Profiles from Attribute Filtering for Classification of Remote Sensing Images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.1 (2018), pp. 249–256 (cit. on p. 30).

[131] M.-T. Pham, E. Aptoula, and S. Lefèvre. "Feature profiles from attribute filtering for classification of remote sensing images". In: *Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 11.1 (2017), pp. 249–256 (cit. on p. 98).

[132] M.-T. Pham, S. Lefèvre, and E. Aptoula. "Local feature-based attribute profiles for optical remote sensing image classification". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.2 (2018), pp. 1199–1212 (cit. on p. 30).

[133] D. Pirrone, F. Bovolo, and L. Bruzzone. "A novel framework based on polarimetric change vectors for unsupervised multiclass change detection in dual-pol intensity SAR images". In: *IEEE Transactions on Geoscience and Remote Sensing* (2020), pp. 4780–4795 (cit. on p. 42).

[134] J. Qian, Y. Luo, Y. Wang, and D. Li. "Cloud Detection of optical remote sensing image time series using Mean Shift algorithm". In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2016, pp. 560–562 (cit. on p. 37).

[135] T. del Ríío-Mena, L. Willemen, A. Vrieling, and A. Nelson. "Understanding intra-annual dynamics of ecosystem services using satellite image time series". In: *Remote sensing* 12.4 (2020), p. 710 (cit. on p. 42).

[136] C. Ronse. "Ordering partial partitions for image segmentation and filtering: Merging, creating and inflating blocks". In: *Journal of Mathematical imaging and Vision* 49.1 (2014), pp. 202–233 (cit. on pp. 12, 70).

[137] M. Rostaghi and H. Azami. "Dispersion entropy: A measure for time-series analysis". In: *IEEE Signal Processing Letters* 23.5 (2016), pp. 610–614 (cit. on p. 107).

[138] S. Roy, S. Gupta, and S. Omkar. "Case study on: Scalability of preprocessing procedure of remote sensing in hadoop". In: *Procedia Computer Science* 108 (2017), pp. 1672–1681 (cit. on p. 41).

[139] M. Rußwurm and M. Körner. "Multi-temporal land cover classification with sequential recurrent encoders". In: *ISPRS International Journal of Geo-Information* 7.4 (2018), p. 129 (cit. on p. 45).

[140] M. Rußwurm and M. Körner. "Self-attention for raw optical satellite time series classification". In: *arXiv preprint arXiv:1910.10536* (2019) (cit. on p. 43).

[141] S. Saha, F. Bovolo, and L. Bruzzone. "Unsupervised Deep Change Vector Analysis for Multiple-Change Detection in VHR Images". In: *IEEE Transactions on Geoscience and Remote Sensing* 57.6 (2019), pp. 3677–3693 (cit. on pp. 43, 118).

[142] P. Salembier and S. Foucher. "Optimum graph cuts for pruning binary partition trees of polarimetric sar images". In: *IEEE Transactions on Geoscience and Remote Sensing* 54.9 (2016), pp. 5493–5502 (cit. on p. 28).

[143] P. Salembier and L. Garrido. "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval". In: *IEEE transactions on Image Processing* 9.4 (2000), pp. 561–576 (cit. on pp. 12, 17, 28).

[144] P. Salembier, S. Liesegang, and C. López-Martıínez. “Ship detection in SAR images based on maxtree representation and graph signal processing”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.5 (2018), pp. 2709–2724 (cit. on pp. 22, 111).

[145] P. Salembier, A. Oliveras, and L. Garrido. “Antiextensive connected operators for image and sequence processing”. In: *IEEE Transactions on Image Processing* 7.4 (1998), pp. 555–570 (cit. on pp. 12, 27).

[146] P. Salembier and M. H. Wilkinson. “Connected operators”. In: *IEEE Signal Processing Magazine* 26.6 (2009) (cit. on p. 26).

[147] J. Schauer and A. Nüchter. “The Peopleremover—Removing Dynamic Objects From 3-D Point Cloud Data by Traversing a Voxel Occupancy Grid”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1679–1686 (cit. on p. 118).

[148] J. Serra. “Introduction to mathematical morphology”. In: *Computer Vision, Graphics, and Image Processing* 35.3 (1986), pp. 283–305 (cit. on p. 11).

[149] B. Settles. *Active learning literature survey*. Tech. rep. University of Wisconsin-Madison Department of Computer Sciences, 2009 (cit. on p. 42).

[150] J. Shao, B. Du, C. Wu, and L. Zhang. “Tracking Objects From Satellite Videos: A Velocity Feature Based Correlation Filter”. In: *IEEE Transactions on Geoscience and Remote Sensing* 57.10 (2019), pp. 7860–7871 (cit. on p. 118).

[151] H. Shen, X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, and L. Zhang. “Missing information reconstruction of remote sensing data: A technical review”. In: *IEEE Geoscience and Remote Sensing Magazine* 3.3 (2015), pp. 61–85 (cit. on p. 118).

[152] D. D. Sleator and R. E. Tarjan. “A data structure for dynamic trees”. In: *Journal of Computer and System Sciences* 26.3 (1983), pp. 362–391 (cit. on p. 41).

[153] R. Sliuzas, A. Hill, C. Lindner, and S. Greiving. “Dar es Salaam Land use and informal settlement data set”. In: *NASA SEDAC* (2016) (cit. on p. 105).

[154] P. Soille. "Constrained connectivity for hierarchical image partitioning and simplification". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.7 (2008), pp. 1132–1145 (cit. on pp. vii, 4, 12, 16).

[155] Y. T. Solano-Correa, F. Bovolo, L. Bruzzone, and D. Fernández-Prieto. "A Method for the Analysis of Small Crop Fields in Sentinel-2 Dense Time Series". In: *IEEE Transactions on Geoscience and Remote Sensing* 58.3 (2019), pp. 2150–2164 (cit. on p. 44).

[156] Y. T. Solano-Correa, F. Bovolo, L. Bruzzone, and D. Fernández-Prieto. "Spatio-temporal evolution of crop fields in Sentinel-2 Satellite Image Time Series". In: *International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*. IEEE. 2017, pp. 1–4 (cit. on p. 44).

[157] K. J. de Souza, A. d. A. Araújo, S. J. Guimarães, Z. K. do Patrocínio, and M. Cord. "Streaming graph-based hierarchical video segmentation by a simple label propagation". In: *Conference on Graphics, Patterns and Images*. IEEE. 2015, pp. 119–125 (cit. on p. 41).

[158] P. Sprent and N. C. Smeeton. *Applied nonparametric statistical methods*. CRC press, 2016 (cit. on p. 106).

[159] Y. Sun, M. Liu, and M. Q.-H. Meng. "Invisibility: A moving-object removal approach for dynamic scene modelling using RGB-D camera". In: *International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2017, pp. 50–55 (cit. on p. 118).

[160] D. Tang, F. Wang, Y. Xiang, H. You, and W. Kang. "Automatic Water Detection Method in Flooding Area for GF-3 Single-Polarization Data". In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. 2018 (cit. on p. 110).

[161] F. Tian, M. Brandt, Y. Y. Liu, A. Verger, T. Tagesson, A. A. Diouf, K. Rasmussen, C. Mbow, Y. Wang, and R. Fensholt. "Remote sensing of vegetation dynamics in drylands: Evaluating vegetation optical depth (VOD) using AVHRR NDVI and in situ green biomass data over West African Sahel". In: *Remote Sensing of Environment* 177 (2016), pp. 265–276 (cit. on pp. vi, 2).

[162] G. Tochon, M. Dalla Mura, M. A. Veganzones, T. Géraud, and J. Chanussot. “Braids of partitions for the hierarchical representation and segmentation of multimodal images”. In: *Pattern Recognition* 95 (2019), pp. 162–172 (cit. on p. 46).

[163] M. G. Tulbure and M. Broich. “Spatiotemporal dynamic of surface water bodies using Landsat time-series data from 1999 to 2011”. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 79 (2013), pp. 44–52 (cit. on pp. vi, 2).

[164] C. Tuna, F. Merciol, and S. Lefèvre. “Monitoring Urban Growth with Spatial Filtering of Satellite Image Time Series”. In: *IEEE Joint Urban Remote Sensing Event (JURSE)*. 2019, pp. 1–4 (cit. on pp. ix, 6, 95, 100).

[165] C. Tuna, A. Giros, F. Merciol, and S. Lefèvre. “On Morphological Hierarchies for Image Sequences”. In: *International Conference on Pattern Recognition (ICPR)*. IEEE. 2020 (cit. on pp. ix, 5, 65).

[166] C. Tuna, F. Merciol, and S. Lefèvre. “Analysis of Min-Trees over Sentinel-1 Time Series for Flood Detection”. In: *International Workshop on the Analysis of Multitemporal Remote Sensing Images (MultiTemp)*. IEEE. 2019, pp. 1–4 (cit. on pp. ix, 6, 95).

[167] C. Tuna, F. Merciol, and S. Lefèvre. “Attribute profiles for satellite image time series”. In: *International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE. 2019, pp. 126–129 (cit. on pp. ix, 6, 95).

[168] C. Tuna, F. Merciol, and S. Lefèvre. “Spatio-Temporal Object Stability for Monitoring Evolving Areas in Satellite Image Time Series”. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XLIII-B2. 2020, pp. 1273–1280 (cit. on pp. ix, 6).

[169] C. Tuna, B. Mirmahboub, F. Merciol, and S. Lefèvre. “Component trees for image sequences and streams”. In: *Pattern Recognition Letters* 129 (2020), pp. 255–262 (cit. on pp. ix, 5, 47).

[170] F. Tushabe and M. H. Wilkinson. "Content-based image retrieval using combined 2D attribute pattern spectra". In: *Workshop of the Cross-Language Evaluation Forum for European Languages*. Springer. 2007, pp. 554–561 (cit. on p. 31).

[171] E. R. Urbach, J. B. Roerdink, and M. H. Wilkinson. "Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.2 (2007), pp. 272–285 (cit. on pp. 21, 27, 31, 32).

[172] E. R. Urbach and M. H. Wilkinson. "Shape-only granulometries and grey-scale shape filters". In: *International Symposium on Mathematical Morphology (ISMM)*. 2002, pp. 305–314 (cit. on pp. 26, 27).

[173] C. J. Van Westen. "Remote sensing and GIS for natural hazards assessment and disaster risk management". In: *Treatise on Geomorphology* 3 (2013), pp. 259–298 (cit. on pp. vi, 3).

[174] M. A. Veganzones, G. Tochon, M. Dalla-Mura, A. J. Plaza, and J. Chanussot. "Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation". In: *IEEE Transactions on Image Processing* 23.8 (2014), pp. 3574–3589 (cit. on p. 28).

[175] S. Velasco-Forero and J. Angulo. "Vector ordering and multispectral morphological image processing". In: *Advances in Low-Level Color Image Processing*. Springer, 2014, pp. 223–239 (cit. on p. 50).

[176] S. Velasco-Forero and J. Angulo. "Random projection depth for multivariate mathematical morphology". In: *Journal of Selected Topics in Signal Processing* 6.7 (2012), pp. 753–763 (cit. on p. 52).

[177] V. Vilaplana, F. Marques, and P. Salembier. "Binary partition trees for object detection". In: *IEEE Transactions on Image Processing* 17.11 (2008), pp. 2201–2216 (cit. on pp. vii, 4, 25).

[178] X. Wang, S. Liu, P. Du, H. Liang, J. Xia, and Y. Li. "Object-based change detection in urban areas from high spatial resolution images based on multiple features and ensemble learning". In: *Remote Sensing* 10.2 (2018), p. 276 (cit. on p. 45).

[179] V. Wegner Maus, G. Câmara, M. Appel, and E. Pebesma. "dtwsat: Time-weighted dynamic time warping for satellite image time series analysis in R". In: *Journal of Statistical Software* 88.5 (2019), pp. 1–31 (cit. on p. 43).

[180] M. A. Westenberg, J. B. Roerdink, and M. H. Wilkinson. "Volumetric attribute filtering and interactive visualization using the max-tree representation". In: *IEEE Transactions on Image Processing* 16.12 (2007), pp. 2943–2952 (cit. on pp. 22, 58).

[181] M. Wilkinson, H. Gao, W. H. Hesselink, J. Jonker, and A. Meijster. "Concurrent computation of attribute filters on shared memory parallel machines". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (2008), pp. 1800–1813 (cit. on p. 58).

[182] M. H. Wilkinson and M. A. Westenberg. "Shape preserving filament enhancement filtering". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2001, pp. 770–777 (cit. on p. 21).

[183] W. Xi, S. Du, Y.-C. Wang, and X. Zhang. "A spatiotemporal cube model for analyzing satellite image time series: Application to land-cover mapping and change detection". In: *Remote Sensing of Environment* 231 (2019), p. 111212 (cit. on pp. vi, 3, 45).

[184] J. Xiao, H. Wu, C. Wang, and H. Xia. "Land Cover Classification Using Features Generated From Annual Time-Series Landsat Data". In: *IEEE Geoscience and Remote Sensing Letters* 15.5 (2018), pp. 739–743 (cit. on p. 43).

[185] C. Xu, C. Xiong, and J. J. Corso. "Streaming hierarchical video segmentation". In: *European Conference on Computer Vision*. Springer. 2012, pp. 626–639 (cit. on p. 41).

[186] Y. Xu, T. Géraud, and L. Najman. "Connected filtering on tree-based shape-spaces". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.6 (2015), pp. 1126–1140 (cit. on pp. 25, 91).

[187] Y. Xu, T. Géraud, and L. Najman. "Context-based energy estimator: Application to object segmentation on the tree of shapes". In: *IEEE International Conference on Image Processing (ICIP)*. IEEE. 2012, pp. 1577–1580 (cit. on p. 27).

[188] Y. Xu, P. Monasse, T. Géraud, and L. Najman. "Tree-based morse regions: A topological approach to local feature detection". In: *IEEE Transactions on Image Processing* 23.12 (2014), pp. 5612–5625 (cit. on p. 25).

[189] J. Yang, P. Gong, R. Fu, M. Zhang, J. Chen, S. Liang, B. Xu, J. Shi, and R. Dickinson. "The role of satellite remote sensing in climate change studies". In: *Nature climate change* 3.10 (2013), pp. 875–883 (cit. on pp. v, 2).

[190] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei. "Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network". In: *IEEE Transactions on Geoscience and Remote Sensing* 56.8 (2018), pp. 4274–4288 (cit. on p. 37).

[191] L. Zhong, L. Hu, and H. Zhou. "Deep learning based multi-temporal crop classification". In: *Remote Sensing of Environment* 221 (2019), pp. 430–443 (cit. on p. 43).

[192] X. Zhu and A. B. Goldberg. "Introduction to semi-supervised learning". In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 3.1 (2009), pp. 1–130 (cit. on p. 42).

[193] Z. Zhu and C. E. Woodcock. "Continuous change detection and classification of land cover using all available Landsat data". In: *Remote sensing of Environment* 144 (2014), pp. 152–171 (cit. on pp. vi, 3).

DOCTORAT BRETAGNE LOIRE / MATHSTIC

Université Bretagne Sud

**Titre :** Hiérarchies Morphologiques pour l'Analyse de Séries Temporelles d'Images Satellites

**Mots-clés :** Hiérarchies morphologique, Filtrage d'Attributs, Spectres de Motifs

**Résumé :** Bien que les hiérarchies morphologiques représentent aujourd'hui un cadre méthodologique bien établi en traitement d'image, leur extension aux données temporelles reste largement inexplorée. Cette thèse aborde la problématique de l'analyse de séries temporelles d'image satellites en utilisant les hiérarchies morphologiques ou représentations arborescentes. Pour ce faire, nous distinguons trois types de modèles, à savoir les hiérarchies spatiales, temporelles et spatio-temporelles. Pour chaque modèle, nous proposons un algorithme de calcul en flux pour le mettre à jour lorsque de nouvelles images sont ajoutées à la série temporelle. Nous analysons les propriétés structurelles des différentes stratégies de construction d'arbres, ce qui nécessite des méthodes de projection de l'arbre spatio-temporel afin de disposer de structures comparables. Nous comparons également les arbres en fonction de leur distribution de nœuds, de leur capacité de filtrage et de leur coût, et concluons à la supériorité de l'arbre spatio-temporel aussi appelé arbre espace-temps. Nous passons ensuite en revue les différents attributs spatio-temporels, dont certains originaux, qui peuvent être extraits de l'arbre spatio-temporel afin de produire des caractéristiques multi-échelle aux niveaux pixel ou image. Ces attributs sont finalement exploités avec des outils de type filtrage ou spectre de forme pour différentes applications en télédétection.

**Title:** Morphological Hierarchies for Satellite Image Time Series

**Keywords:** Hierarchical representation, Attribute Filtering, Pattern Spectra

**Abstract:** Although morphological hierarchies are today a well-established framework for single frame image processing, their extension to time-related data remains largely unexplored. This thesis aims to tackle the analysis of satellite image time series with tree-based representations. To do so, we distinguish between three kinds of models, namely spatial, temporal and spatial-temporal hierarchies. For each model, we propose a streaming algorithm to update the tree when new images are appended to the series. Besides, we analyze the structural properties of the different tree building strategies, thus requiring some projection methods for the spatio-temporal tree in order to obtain comparable structures. Then, trees are compared according to their node distribution, filtering capability and cost, leading to a superiority of the spatio-temporal tree (a.k.a. space-time tree). Hence, we review spatio-temporal attributes, including some new ones, that can been extracted from the space-time tree in order to compute some multiscale features at the pixel or image level. These attributes are finally involved in tools such as filtering and pattern spectrum for various remote sensing based applications.