Approches de topologie algébrique pour l'analyse d'images

Rabih Assaf

UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

ÉCOLE DOCTORALE SCIENCES TECHNOLOGIE SANTE (547)

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE

Discipline : ***MATHEMATIQUES APPLIQUEES ET SCIENCES SOCIALES***

Spécialité : Mathématiques appliquées au traitement d'images

Présentée et soutenue publiquement par

RABIH ASSAF

Le 19 janvier 2018

APPROCHES DE TOPOLOGIE ALGEBRIQUE POUR L'ANALYSE D'IMAGES

Thèse dirigée par **VALERIU VRABIE, MOHAMMAD KACIM ET ALBAN GOUPIL**

JURY

Mme Stéphanie SALMON,	Professeur des Universités,	Université de Reims Champagne Ardenne,	**Président**
M. Valeriu VRABIE,	Maître de conférences,	Université de Reims Champagne Ardenne,	**Directeur de thèse**
M. Mohammad KACIM,	Maître de conférences,	Université Saint Esprit de Kaslik,	**Co-directeur de thèse**
M. Alban GOUPIL,	Maître de conférences,	Université de Reims Champagne Ardenne,	**Encadrant de thèse**
Mme Michèle ROMBAUT,	Professeur des Universités,	Université Grenoble Alpes,	**Rapporteur**
M. Ali WEHBE ,	Professeur des Universités,	Université Libanaise,	**Rapporteur**

Aknowledgment

This thesis marks the end of three years of doctoral studies at the CReSTIC laboratory of the University of Reims Champagne-Ardenne and at the department of mathematics of the Holy Spirit University of Kaslik.

I would like to begin by thanking my thesis supervisors, Valeriu VRABIE, Mohammad KACIM and Alban GOUPIL for supervising my works during these three years. I can not thank them enough for their availability, their unwavering support, their scientific enthusiasm and their encouragement that allowed me to carry out my work.

I'd like to give special thanks to Valeriu VRABIE for his patience even in the most difficult moments, research guidance, and help with administrative formalities. Thank you very much for pushing me beyond what I thought it was my limits. I thank also Alban GOUPIL who has invested a lot of time and effort in this work during these three years of thesis, I learned a lot and progressed, and it's partly thanks to you. I had the chance and the pleasure of working under your supervision. Thanks to Mohammad KACIM for his scientific guidance, advices and for supporting and believing in me from the days of my master studies to reach this level.

My thanks to Mrs Michèle ROMBAUT and Mr Ali WEHBE for having accepted to be examiners of my work and members of the jury, I thank also Mrs Stéphanie SALMON who agreed to evaluate this work as jury president.

Thanks to the joint program of the National Counsel for Scientific Research in Lebanon and the Holy Spirit University of Kaslik for financing this work.

I would also like to thank all the awesome people that I knew during my stay in Reims either from the laboratory or outside it and that are not directly related to this thesis, but that surely helped by sharing good moments.

I have no words to express how grateful I am to my family, my father Nemer, my mother Guitta and my brothers Kamil and Wassim, thanks for supporting me. I wouldn't get here if it wasn't for you.

Contents

1 Introduction **1**
- 1.1 Context and problematic . . . 1
- 1.2 Contributions of the thesis . . . 2
- 1.3 Structure of this document . . . 3

2 Topology, algebraic topology and applications **5**
- 2.1 Image processing context . . . 6
- 2.2 Topology . . . 10
  - 2.2.1 Notions of topology . . . 11
  - 2.2.2 Applications of topology . . . 13
- 2.3 Algebraic topology . . . 14
  - 2.3.1 Panorama on algebraic topology . . . 15
  - 2.3.2 Cell complexes and their types . . . 17
  - 2.3.3 Combinatorial representation . . . 21
  - 2.3.4 Homology groups . . . 24
    - 2.3.4.1 Definitions of homology groups . . . 24
    - 2.3.4.2 The relative homology group . . . 28
    - 2.3.4.3 Technique of computing homology groups . . . 29
    - 2.3.4.4 Example of homology groups computing . . . 31
    - 2.3.4.5 Cycles optimization . . . 38
  - 2.3.5 Applications of algebraic topology in engineering . . . 39
- 2.4 Application objectives of the thesis . . . 41
- 2.5 Conclusion . . . 44

3 Persistent homology and applications to images **45**
- 3.1 Persistent homology . . . 46
  - 3.1.1 Spatialization . . . 47
  - 3.1.2 Filtration and persistence . . . 49
  - 3.1.3 Computing persistent homology . . . 53
- 3.2 Applications of persistent homology . . . 58
- 3.3 Filtration defined on images . . . 59
  - 3.3.1 Combinatorial representation of pixels' cell complex . . . 60
  - 3.3.2 Construction of the filtration of pixels' cell complex . . . 60
  - 3.3.3 Combinatorial representation of superpixels' cell complex . . . 63

3.3.4 Construction of the filtration of superpixels' cell complex 64
3.3.5 How to deal with nested homology classes in 2D images 64
3.3.6 How to deal with nested homology classes in 3D images 65
3.4 Image segmentation using lifespans of homology classes 66
3.4.1 Description of the method 68
3.4.2 Real applications 69
3.5 2D and 3D object segmentation using homology classes 72
3.5.1 Synthetic and real images 74
3.5.2 Otsu thresholding 76
3.5.3 MSER technique 77
3.5.4 Proposed method 79
3.5.4.1 Synthetic image 79
3.5.4.2 Real applications 83
3.5.5 Object segmentation based on superpixels 85
3.5.5.1 SLIC method 85
3.5.5.2 Synthetic image 86
3.5.5.3 Real 2D images 87
3.5.5.4 Real 3D images 89
3.6 Object tracking using relative persistent homology 91
3.6.1 Description of the method 91
3.6.2 Synthetic image 93
3.6.3 Real applications and comparisons 94
3.7 Conclusion 98

4 Sheaf theory and its application in image processing 101
4.1 Introduction to sheaf theory 102
4.2 Sheaves of topological spaces 103
4.3 Cellular sheaves 104
4.3.1 Sheaf of vector spaces 105
4.3.2 Local, global and pseudo sections 106
4.3.3 Categorification 108
4.3.3.1 Introduction 109
4.3.3.2 Changing types using functors 110
4.3.4 Cellular sheaf cohomology and interpretation 111
4.4 Sheaves on partial orders 113
4.4.1 Definitions of sheaves over posets 113
4.4.2 Cohomological analaysis of sheaves over posets 116
4.5 Applications of sheaf theory 118
4.6 Proposed applications of sheaf theory on images 119
4.6.1 Application of cellular sheaves on images 119
4.6.2 Sections on RGB images 124
4.6.3 Interpretation of sheaves of models 128
4.6.3.1 Scale analysis using sheaves on posets 129
4.6.3.1.1 Example 1 129

4.6.3.1.2 Example 2 . . . 135
4.6.3.1.3 Example 3 . . . 138
4.6.3.2 Localization using sheaves on posets . . . 139
4.7 Conclusion . . . 143

5 Conclusion and perspectives **145**
5.1 General conclusion . . . 145
5.2 Perspectives . . . 147

Appendix A Résumé étendu en français **151**
A.1 Chapitre 1: Introduction . . . 152
A.1.1 Contexte et problématique . . . 152
A.1.2 Contributions de la thèse . . . 153
A.1.3 Structure de ce document . . . 154
A.2 Chapitre 2: Topologie, topologie algébrique et applications . . . 154
A.2.1 Le traitement d'images . . . 155
A.2.2 Panorama des méthodes utilisant la topologie algébrique . . . 156
A.2.3 Représentation combinatoire . . . 156
A.2.4 Groupes d'homologie . . . 157
A.2.5 Homologie relative . . . 159
A.2.6 Conclusion . . . 159
A.3 Chapitre 3: L'homologie persistante et ses applications . . . 160
A.3.1 Filtration . . . 160
A.3.2 Persistance . . . 161
A.3.3 Segmentation d'images utilisant les durées de vie des classes d'homologie 161
A.3.4 Segmentation d'objets en utilisant les classes d'homologie . . . 162
A.3.4.1 Segmentation basée sur un complexe cubique de pixels . . . 162
A.3.4.2 Segmentation basée sur complexe cellulaire de superpixels 164
A.3.5 Suivi d'objets en utilisant l'homologie persistante relative . . . 168
A.3.6 Conclusion . . . 170
A.4 Chapitre 4: Théorie des faisceaux et applications sur les images . . . 170
A.4.1 Introduction à la théorie des faisceaux . . . 171
A.4.2 Faisceaux cellulaires . . . 172
A.4.3 Faisceaux sur les posets . . . 174
A.4.4 Applications sur les images . . . 175
A.4.4.1 Applications sur les complexes de Čech . . . 175
A.4.4.2 Sections sur les images RGB . . . 177
A.4.4.3 Interprétation des faisceaux des modèles . . . 178
A.4.4.4 Conclusion . . . 180
A.5 Chapitre 5: Conclusion et perspectives . . . 180
A.5.1 Conclusion générale . . . 180
A.5.2 Perspectives . . . 182

Bibliography **185**

Table des matières

1 Introduction **1**
1.1 Contexte et problématique . . . 1
1.2 Contributions de la thèse . . . 2
1.3 Structure de ce document . . . 3

2 Topologie, topologie algébrique et applications **5**
2.1 Contexte de traitement d'images . . . 6
2.2 Topologie . . . 10
2.2.1 Notions de topologie . . . 11
2.2.2 Applications de topologie . . . 13
2.3.0 Topologie algébrique . . . 14
2.3.1 Panorama sur la topologie algébrique . . . 15
2.3.2 Complexes cellulaires et leurs types . . . 17
2.3.3 Représentation combinatoire . . . 21
2.3.4 Groupes d'homologie . . . 24
2.3.4.1 Définitions des groupes d'homologie . . . 24
2.3.4.2 Le groupe d'homologie relative . . . 28
2.3.4.3 Technique de calcul des groupes d'homologie . . . 29
2.3.4.4 Exemple de calcul des groupes d'homologie . . . 31
2.3.4.5 Optimisation des cycles . . . 38
2.3.5 Applications de la topologie algébrique en ingénierie . . . 39
2.4 Objectifs applicatifs de la thèse . . . 41
2.5 Conclusion . . . 44

3 Homologie persistante et applications sur les images **45**
3.1 Homologie persistante . . . 46
3.1 Spatialisation . . . 47
3.1.2 Filtration et persistance . . . 49
3.1.3 Calcul d'homologie persistante . . . 53
3.2 Applications d'homologie persistante . . . 58
3.3 Filtration définée sur les images . . . 59
3.3.1 Représentation combinatoire de complexe cellulaire des pixels . . . 60
3.3.2 Construction de filtration de complexe cellulaire des pixels . . . 60
3.3.3 Représentation combinatoire de complexe cellulaire des superpixels 63

3.3.4 Construction de la filtration de complexe cellulaire des superpixels . 64
3.3.5 Comment gérer les classes d'homologie imbriquées dans les images 2D . . . 64
3.3.6 Comment gérer les classes d'homologie imbriquées dans les images 3D . . . 65
3.4 Segmentation d'images utilisant les durées de vie des classes d'homologie . 66
3.4.1 Description de la méthode . . . 68
3.4.2 Applications réelles . . . 69
3.5 Segmentation d'objets 2D et 3D utilisant les classes d'homologie . . . 72
3.5 Image synthétique et images réelles . . . 74
3.5.2 Seuillage d'Otsu . . . 76
3.5.3 Technique MSER . . . 77
3.5.4 Méthode proposée . . . 79
3.5.4.1 Image synthétique . . . 79
3.5.4.2 Applications réelles . . . 83
3.5.5 Segmentation d'objets basée sur les superpixels . . . 85
3.5.5.1 Méthode de SLIC . . . 85
3.5.5.2 Image synthétique . . . 86
3.5.5.3 Images réelles 2D . . . 87
3.5.5.4 Images réelles 3D . . . 89
3.6 Tracking d'objets utilisant l'homologie persistante relative . . . 91
3.6.1 Description de la méthode . . . 91
3.6.2 Image synthétique . . . 93
3.6.3 Applications réelles et comparaisons . . . 94
3.7 Conclusion . . . 98

4 Théorie des faisceaux et ses applications en traitement d'images 101
4.1 Introduction à la théorie des faisceaux . . . 102
4.2 Faisceaux des espaces topologiques . . . 103
4.3 Faisceaux cellulaires . . . 104
4.3.1 Faisceaux des espaces vectorielles . . . 105
4.3.2 Sections locales, globales et pseudosections . . . 106
4.3.3 Catégorification . . . 108
4.3.3.1 Introduction . . . 109
4.3.3.2 Changement des types utilisant des foncteurs . . . 110
4.3.4 Cohomologie des faisceaux cellulaires et interpretation . . . 111
4.4 Faisceaux sur des ordres partiels . . . 113
4.4.1 Définitions des faisceaux sur des posets . . . 113
4.4.2 Analyse cohomologique des faisceaux sur des posets . . . 116
4.5 Applications de la théorie des faisceaux . . . 118
4.6 Applications proposées de la théorie des faisceaux sur les images . . . 119
4.6.1 Applications des faisceaux sur les images . . . 119
4.6.2 Sections sur les images RGB . . . 124
4.6.3 Interpretation des faisceaux de modèles . . . 128

4.6.3.1 Analyse d'échelle utilisant les faisceaux sur des posets . . . 129
4.6.3.1.1 Exemple 1 . . . 129
4.6.3.1.2 Exemple 2 . . . 135
4.6.3.1.3 Exemple 3 . . . 138
4.6.3.2 Localisation utilisant les faisceaux sur des posets . . . 139
4.7 Conclusion . . . 143

5 Conclusion et perspectives **145**
5.1 Conclusion générale . . . 145
5.2 Perspectives . . . 147

A Résumé étendu en français **151**

List of Figures

2.1 Synthetic image and its Otsu segmentation. 10
2.2 Cubical complex of dimension 1. 18
2.3 Cubical complex of dimension 2 19
2.4 Simplices of different dimensions. 20
2.5 A simplicial complex comparison. 21
2.6 Čech complexes with two different ball radii. 22
2.7 Example of a combinatorial space containing 0-cells, 1-cells and 2-cells. . . 23
2.8 Example of chains of one dimension. 25
2.9 Example of a cell complex containing cycles and boundaries. 26
2.10 Chains, cycles and boundaries between dimensions. 27
2.11 Examples of relative cycles where A is the subcomplex of vertical lines. . . 28
2.12 Example of a surface with two holes. 32
2.13 An image of prostate gland. 42
2.14 Sample of real images for object segmentation. 42
2.15 Example of an instant of a time lapse image. 43

3.1 Workflow of computing persistent homology. 47
3.2 A Morse function with four critical points. 48
3.3 A non noisy hole versus a noisy one. 49
3.4 A Morse function and its associated persistent diagram and barcode. . . . 53
3.5 Simple filtration example. 54
3.6 Persistence diagrams of 0 and 1 dimensions. 57
3.7 Barcode of homology classes of 0 and 1 dimensions. 58
3.8 Set of pixels and their cubical complex. 59
3.9 Set of voxels and their cubical complex. 60
3.10 Value assigning in 2D case. 61
3.11 Value assigning in 3D case. 62
3.12 Pixels' complex subfiltration for the 2D case presented in figure 3.10. . . . 62
3.13 Simple example of superpixels combinatorial representation. 63
3.14 Superpixels complex value assigning. 65
3.15 Image segmentation methodology plan. 67
3.16 Satellite image segmentation. 70
3.17 Biomedical image segmentation. 71
3.18 First persistent entropy distribution with respect to classes. 73

3.19 A synthetic image. . . . 74
3.20 An image containing 24 imperceptible dots. . . . 75
3.21 10 coins image. . . . 75
3.22 Results of Otsu segmentation on coins image. . . . 76
3.23 Otsu's thresholding result on the synthetic image . . . 77
3.24 An image of dots and its otsu thresholding result. . . . 78
3.25 Results of MSER on coins image. . . . 78
3.26 Results of MSER on dots image. . . . 79
3.27 Results of MSER method on the synthetic image. . . . 80
3.28 Results of the proposed method on the synthetic image. . . . 81
3.29 Results of the proposed mehod after scaling and rotating the synthetic image. 82
3.30 Results of object segmentation on coins' image. . . . 83
3.31 Results of our method on a biomedical image. . . . 84
3.32 Results of the proposed method on a biomedical image that shows components segmentation. . . . 85
3.33 Results of the proposed object segmentation method on the synthetic image using a SLIC space reduction. . . . 86
3.34 Results of the proposed segmentation method on the image of dots and its inverse using a SLIC space reduction. . . . 88
3.35 Segmentation results of the proposed method on a 2D real biomedical image using SLIC space reduction. . . . 89
3.36 Results of the proposed segmentation method on a 3D image. . . . 90
3.37 The "worm hole" and sphere detection by relative homology. . . . 92
3.38 Results of the proposed relative persistent homology method on a synthetic 2D+t image of 15 frames. . . . 94
3.39 Persistence diagram of H_2 elements of the 2D+t sequence shown in figure 3.40. 95
3.40 Results of relative persistent homology proposed method on a real grayscaled 2D+t biomedical image. . . . 96
3.41 Results obtained on a real grayscaled 2D+t biomedical image using the relative persistent homology showing two moving cells. . . . 96
3.42 Results of the state of the art "TrackMate" tool on the 2D+t sequence shown in figure 3.40. . . . 97

4.1 Representation of sheaves of continuous functions. . . . 104
4.2 Topological base space for sheaf construction. . . . 105
4.3 Stalks assigning to the cell complex. . . . 105
4.4 Restrictions assigned to each cell inclusion. . . . 106
4.5 Sections consistent with restrictions. . . . 107
4.6 A composition of morphisms and its associativity scheme. . . . 110
4.7 A simple base space for sheaf construction. . . . 111
4.8 The coboundary that represents the coboundary map d^p . . . 112
4.9 A poset and its dual. . . . 114
4.10 Alexandroff topology on E where arrows are inclusions. . . . 115
4.11 Stalks on a poset and its dual with restrictions. . . . 116

4.12 An image containing two glued objects. . . . 120
4.13 The keypoints of the image in figure 4.12 with the open balls. . . . 121
4.14 The base space for the construction of sheaves. . . . 121
4.15 The sheaf base space with the restrictions. . . . 123
4.16 6 pixels with their opens with radius $1/2 + \epsilon$. . . . 124
4.17 The Čech complex and the stalks over it. . . . 125
4.18 Sections over two vertices and one edge of the complex. . . . 126
4.19 Sections consistent with restrictions over the 6 pixels. . . . 126
4.20 Local sections over 2 pixels with identity maps as restrictions. . . . 127
4.21 Consistent components according to the sheaves giving a partition of the image. . . . 127
4.22 Local sections over 2 pixels with Luma maps as restrictions. . . . 127
4.23 Three opens A, B and C. . . . 128
4.24 The poset $\mathcal{E}$ with arrows meaning inclusions. . . . 129
4.25 The opens with the cycle a in $A \cup B$. . . . 130
4.26 The sheaf over the poset $\mathcal{E}$. . . . 130
4.27 Global sections of the sheaf $\mathscr{S}$ over poset $\mathcal{E}$. . . . 131
4.28 A local section on the sheaf $\mathscr{S}$ over the poset $\mathcal{E}$ which can not be extend to a global section. . . . 132
4.29 The opens with the cycle $a \in A \cup B \cup C$. . . . 135
4.30 Restrictions over the sheaf $\mathscr{S}'$ of poset $\mathcal{E}$. . . . 136
4.31 A local section on the sheaf $\mathscr{S}'$ over the poset $\mathcal{E}$ which can not be extend to a global section. . . . 136
4.32 The opens with the cycles a, b and c for the case of example 3 in scale analysis. . . . 137
4.33 The sheaf $\mathscr{S}''$ over the poset $\mathcal{E}$. . . . 137
4.34 Global sections of the sheaf $\mathscr{S}''$ over poset $\mathcal{E}$. . . . 138
4.35 The poset $\mathcal{E}'$ with arrows in decreasing direction. . . . 139
4.36 The opens with the cycles a, b and c for the localization example. . . . 140
4.37 The sheaf $\mathscr{S}$ over the poset $\mathcal{E}'$. . . . 140
4.38 [Global sections of sheaf $\mathscr{S}$ over poset $\mathcal{E}'$. . . . 142

A.1 Une image synthétique et sa segmentation à l'aide de la méthode OTSU . . 156
A.2 Exemple d'un complexe cellulaire. . . . 157
A.3 Les groupes des cycles et des bords. . . . 158
A.4 Segmentation d'une image biomédicale. . . . 163
A.5 Une image synthétique. . . . 164
A.6 Résultats de méthode proposée sur l'image synthétique. . . . 165
A.7 L'image des points et les résultats de la segmentation. . . . 166
A.8 Résultats de segmentation pour une image 3D. . . . 167
A.9 La détection de "trou de ver" et de la sphère par homologie relative. . . . 168
A.10 Résultats de l'homologie relative persistante sur une 2D+t image réelle biomédicale. . . . 169
A.11 Affectation des fibres au complexe cellulaire. . . . 172

A.12 Restrictions affectées pour chaque inclusion de cellule. 172
A.13 Sections consistantes avec les restrictions. 173
A.14 Un poset et son dual. 174
A.15 Les points clés avec les boules ouvertes. 176
A.16 L'espace de base des faisceaux sur les restrictions. 177
A.17 6 pixels avec leurs ouverts avec rayons $1/2 + \epsilon$. 178
A.18 Sections consistantes avec leurs restrictions sur les 6 pixels. 179
A.19 L'espace entier avec le cycle a. 179

Chapter 1

Introduction

1.1 Context and problematic 1
1.2 Contributions of the thesis 2
1.3 Structure of this document 3

1.1 Context and problematic

In the last decade, there have been concerted efforts to transform algebraic topology from an abstract mathematical field into a more concrete domain. These efforts have motivated mathematicians and scientists to apply the concepts of this branch of mathematics and develop them. Applications to engineering problems led to the resolution of a lot of challenges and helped to enlarge the research bridge between mathematics and engineering.

Algebraic topology is often considered a difficult and abstract theory but it's starting to find a significant number of applications in a large number of scientific domains, and in particular in data analysis. Therefore, the choice of this theory as a tool for the treatment of data provided by images is natural. However, it is necessary that the topological space of the data could be represented combinatorially to be usable by algorithms and implemented by a machine.

The algebraic topology allows to answer these requests because it uses the well developed tools of linear algebra. One of the main profits of algebraic topology is its ability to construct spaces on points with notion of neighborhood that represent the data, which make it very useful on the level of image processing. Indeed, algebraic topological tools provide features about spaces, which are insensitive to continuous deformations. Applied to images, the topological analysis could reveal important characteristics: how many connected components are present, which ones have holes and how many, how are they related one to another, how to infer from local coherent information to a global vision, etc.

Some tasks of image processing like segmentation, object tracking and data fusion are complex and limited by many considerations. Choosing the size of an object or a specific intensity for a threshold can enormously change the result of an image processing method. The diversity in the background can make results mixing the salient objects with the background. The specificity of the applications make them in a lot of cases not able to detect the overlaid objects. However the choice of prior parameters, the suppression of background influence and the processing of superposition of objects, that generally occur in image processing tasks, don't affect algebraic topology approaches and techniques.

1.2 Contributions of the thesis

We propose in this work methods based on algebraic topology in order to solve some of the main challenges in image processing. Noting that algebraic topology comes not to eliminate the use of other techniques in image processing but to complete or to be associated to them when needed.

More precisely, we propose methodologies and approaches using persistent homology, which is one of the most powerful tools in algebraic topology, and sheaves theory which is a complex but promising part of algebraic topology in applications.

First, a known technique to segment images is to compute features inside windows and classify then in order to get the image segmentation. It would be interesting to see if the algebraic topology can add features that are more pertinent and eventually can improve the quality of segmentation.

Second, classical methods used in object segmentation fail in many cases to identify only the interesting objects because of the presence of the noise. They depend also on the background level, which make them weak in detecting overlaid objects. Several classical methods depend highly on some choices of parameters like volumes or sizes of objects. Since algebraic topology studies the presence of holes and voids using one of its powerful invariant, which is homology groups, it would be interesting to see if it is able to segment the objects. The advantage of computing cycles and homology classes is their insensitivity to background changes and independence from prior parameters.

Next, object detection and tracking are usually regarded as one of the major and challenging tasks in the pipeline of image processing and pattern analysis. Most of the existing techniques are non generic methods based mainly on complex algorithms controlled by many parameters and metrics. Since construction of topological complexes is possible on pixels of 3D images, and as well as on 2D+t images, algebraic topology can be a simple solution of these challenges. The relative version of homology can detect movement of objects without the use of prior parameters.

Finally, a totally novel concept in association of algebraic topology to image processing is the use of sheaf theory. Utilization and applications of this theory are still not elaborated on the level of image processing. It could be worthful to initiate some of these applications

trying to interpret them using the cohomological analysis.

1.3 Structure of this document

In the next chapter "Topology, algebraic topology and applications", we make a brief description of some of image processing techniques and their limits. Then we detail basic notions of topology and mostly the algebraic topology and explaining the purpose of their use in image processing. Computation of homology groups is next explained and the chapter ends with the tasks of image processing that we are interested in this work.

The following chapter "Persistent homology and applications to images" represents the main contributions in proposing several methodologies of algebraic topology constructions on images. Beginning by descriptions of persistent homology and its computation, we explain how to transform images to combinatorial representations. Then we propose new methods in image segmentation, multidimensional object segmentation using pixels and superpixels and we end this chapter by presenting a new method for tracking of objects in motion using the relative homology.

We initiate a framework of image analysis based on sheaf theory in the chapter "Sheaf theory and its image applications". First, the concept of the sheaf theory is described then we explain a methodology of data fusion using sheaves. We show how we can associate these concepts to image tasks. And we finish by its association to image processing and analysis like construction of sections on colored images, scale analysis and localization.

The last chapter concludes this work and presents a several number of perspectives on short and long term. These are partly linked with developments of our work like continuation of the application of sheaves on real images but concern also the use of other aspects of the algebraic topology, in particular the Morse theory in order to associate critical points of different dimensions or the multidimensional persistence, to integrate many factors in the filtration scheme construction.

Chapter 2

Topology, algebraic topology and applications

2.1 Image processing context 6
2.2 Topology 10
2.2.1 Notions of topology 11
2.2.2 Applications of topology 13
2.3 Algebraic topology 14
2.3.1 Panorama on algebraic topology 15
2.3.2 Cell complexes and their types 17
2.3.3 Combinatorial representation 21
2.3.4 Homology groups 24
2.3.5 Applications of algebraic topology in engineering 39
2.4 Application objectives of the thesis 41
2.5 Conclusion 44

Our works are interested mainly in developing tools issued from algebraic topology to achieve image processing tasks and approaches. This is why this chapter attempts to make a quick state of art about some of the image processing techniques and extend some of the tools employed by topology and algebraic topology in scientific problems. Detailed explanations on homology theory and computation of homology groups are presented giving some explications of their contributions in applications.

For this purpose, the section 2.1 presents some of the methods applied in image processing tasks such as image segmentation, object segmentation and track detection. We talk in this section about problems faced in image processing and why the algebraic topology methods is useful. Then, the section 2.2 focuses on introducing of some notions of topology and presents a state of the art of the applications of topology in engineering problems

and image processing tasks. The section 2.3 concentrates on introducing some theories of algebraic topology like homology, Morse theory and sheaf theory. Then it extends the combinatorial representation of topological spaces and explains in details notions of homology theory and finishes by developing the state of the art applications on engineering problems. Afterward, the section 2.4 explains the image processing problems that our methods try to solve. Finally, the section 2.5 concludes the chapter.

2.1 Image processing context

In this section, we aim to explain some of the methods and techniques used in image processing specially in image segmentation, object segmentation and object tracking because our main applications will be concentrated on these fields.

Image segmentation is a challenging task that has been considered as a key step in image processing and remaining as a long-standing problem in the field with a massive literature, see [MC15, ZMC16] for thorough surveys on this topic. The objective of image segmentation is to partition an image into non-overlapping homogeneous regions or to locate objects of interest in the image. This tool permits to simplify image representation into other form which makes it easier for further processing of higher level tasks. Object segmentation on the other hand is the process of extracting an object in an image from its other aspects. These techniques carry a variety of applications that include many sciences topics like computer vision [LST16, YSA16], image analysis [MFC13, Mah14], medical image processing [LWD14, PCO16] and remote sensing [TGGP15, SZ17].

A lot of reasons make the segmentation a hard and challenging task. Mainly the complexity of the used algorithms that depend on many parameters and metrics that control the segmentation procedure and that cause the lack of generic "off-the-shelf" solutions. The proposed techniques and solutions depend hugely on types of applications and images in question.

A large variety of segmentation techniques and methods have been discussed and developed in the literature. The classical ones are mostly based on mathematical or statistical methods. In other class, we find the clustering and soft computing techniques that involve image segmentation.

Mathematical and statistical methods

- Thresholding: the values of pixels belonging to image objects are different from the values of the pixels belonging to the background or other background in many applications of image processing. This make thresholding a simple but effective tool to separate those foreground objects from the background. The intensity value of each pixel is compared to a suitable threshold value and thus can be assigned to a class of the image [XCG17, CYC+14, GMACA14]. Thresholding techniques are grouped into local thresholding techniques that depend on the local properties of

the pixels and their neighborhoods, and global techniques that segment an image basing on information obtained globally by using image histogram or global texture properties. Otsu thresholding remains one of the classical thresholding techniques that optimize the best threshold to segment the image [GBY+18]. Noting that the choice of the the threshold is affected highly by noise in the image, and the nature of thresholding doesn't allow to detect overlaid objects. Moreover, maximally stable regions [NS08] represent also one of the most used thresholding techniques, but it depends on th choice of previous parameters. We will detail this technique in the next chapter.

- Mathematical morphology: it is dedicated to extract information from images that concentrates on the geometrical structure and forms [SL16, PTS14]. Most of its operations are based on dilation of the topological and geometrical continuous space concepts of images such as shape, size, connectivity and geodesic distance.

- Wavelet transformations: it is a mathematical tool that is widely applied in extracting information from signals [LLY17, SZS+13]. Its framework provides accurate tools for multi-scale image analysis and representations since it works in both time domain and frequency domain comparing to Fourier Transform that works with only time domain. For image segmentation tasks, wavelet transformation performs mainly features extraction to find edges in images or input to clustering algorithms for example. But the wavelet transformation is limited by its complexity, it's hard to choose the proper wavelets for a specific purpose in application, knowing that it's computationaly intensive.

- Graph partitioning methods: the pixels of an image can be manipulated as a graph which is defined as a group of objects and their relations designed by vertices and edges. Graph partitioning methods aim to model the image as a weighted graph using the impact of pixel neighborhoods on a given cluster of pixels. Many approaches are used in this method like Markov random fields [LWD17, KZ12], optimization algorithms [AP13], etc.

- Partial differential equations (PDE): since an image is seen as a function defined in a continuous space, segmentation techniques can be derived from solving partial differential equations extracted from this function [KKF+13, LWE+15]. Parametric method specialized in parameterizing the contours such as snakes introduced in [KWT88] and level set methods that address the problem of curve and surface propagation and initiated in [OS88] are typical of PDE based techniques in image and object segmentation.

- Watershed transformations: the watershed transform consider a gray-scale image as a topological surface, where the values of $f(x, y)$ are interpreted as heights. The role of watershed transform is to find the catchment basins and peak lines in signals such as grayscale image. Regarding the problem of image segmentation, the key concept is to change the input image into another one whose catchment basins are the objects or regions we want to identify [GMP+15, LZW10].

Clustering and soft Computing techniques

- Basic clustering techniques: clustering is the division of data into groups of similar elements so that the elements in different classes are as different as possible and elements that belong to the same cluster are as analogue as possible [WCV12] There are many types of clustering techniques like partitioning algorithms that include K-mean, probabilistic algorithms, hierarchical algorithms, grid based algorithms. All these techniques can be used in image segmentation in order to group the pixel in the image into different classes.

- Neural networks: a neural network is a way to process information inspired by the mechanism of biological nervous systems that is learned by example. Its diverse forms are widely used in image segmentation [MVM+16, PPA16]. Convolution neural networks for example aim to compute convolved features from the image to achieve image segmentation [SLD17]. These computed features, for example on windows, will segment the image depending on their similarities.

- Fuzzy logic approaches: fuzzy logic is a technique of computing based on degrees of truth that belong to the $[0, 1]$ interval rather than the usual true or false (1 or 0) boolean logic on which the modern computer is based. These degrees of truth are suitable to formalize reasoning when dealing with vague terms in image processing techniques specially in image segmentation [JRG16, STJ+16] since they can divide data points into clusters or homogeneous classes.

- Genetic algorithms: they represent an intelligent use of a random search handled to solve optimization problems and hence reduce the complexity of the studied problem. They rely deeply on probabilistic tools. Recently the genetic algorithms have been very effective on the level of image segmentation due to its robustness to image noise [WLL+14, MST+16]. They can be used for the modification of the parameters in existing segmentation algorithms and are viewed as function optimizers.

Moreover, object detection and tracking are usually regarded as another major and challenging tasks in the pipeline of image processing and pattern analysis. The objective of tracking is to pursue the movement of an object through the time changing. Most of the existing techniques are non generic methods based mainly on complex algorithms controlled by many parameters and metrics. The "one-size-fits-all" universally appropriate tracking method doesn't exist at this time according to the study made in [CSdC+14]. Many techniques have been proposed and developed in the literature, see [LWZ08, PF08] for thorough explanations. These techniques differ in aim, motivation and used algorithms such as:

- Frame differencing: tracking objects is widely performed by frame difference method. The purpose of the technique is to detect the moving objects from the difference between the existing frame and the reference frame. It uses the pixel-based difference to find the moving object [RYK14, AC17].

- Point detectors: for each new frame in the image sequence, an interest point detector

is used to detect candidate points for tracking. Then a feature descriptor is computed for each of the candidate points. The goal then is to seek for this descriptor to match with those of previously computed features [GKK+15, ZMAT16].

- Background subtraction: it is a method that aim to localize the connected pixels moving on the foreground despite the prior information of the sequence in order to get the initial estimate of motion [YQF+14, THV16].
- Supervised learning: it is a technique that search to link input variables to an output variable and that uses algorithms to learn the mapping function from the input to the output [TT12, BDA16]. Recently, supervised learning techniques like random forests, convolutional neural networks, including deep learning approaches have been widely used in the tracking process of objects.

It exists other image processing techniques such as image preprocessing, image enhancement, feature extraction, image classification. We will not provide deep explanations of these tasks since our applications didn't contribute in.

But we must note that in summary these methods and techniques face a lot of problems that control its capabilities, taking for example, at the technical level:

- Depending on prior parameters: the use of prior parameters is essential in many image processing techniques and specially in segmentation. The size of objects, the intensity for thresholding, the compactness, circularity and many other parameters must be taken into account as prior parameters to the suggested methods.
- The problem of background: variations in background and foreground still a trouble in image processing. Discriminating the objects from noisy backgrounds for example is a very hard task, and not to forget the cases of dynamic backgrounds and illumination changes.
- Overlaid objects: objects that are inside each others are very hard to discriminate automatically without specific parameters for the specific applications. A lot of methods succeed in segmenting the big objects or the smaller ones but so rarely to find a one that can do both.

And at the general level:

- Accuracy: the results obtained are never satisfying, this is the common problem between all the researchers in image processing. The efficiency of a certain technique can be questionable from a specialist to another.
- Absence of generic solutions: it doesn't exist one method that fit all the applications. Each application needs its appropriate technique that can fit with it and doesn't be accurate with another.
- Diversification: the presence of many possible solutions for a given problem. The choice of the better solution that corresponds for the desired application is a hard task because of the diversity of the techniques in the literature.

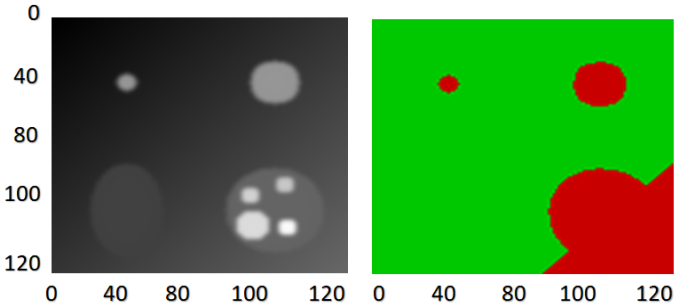


Figure 2.1: Synthetic image and its Otsu segmentation.

To take a feeling of these problems we illustrate a simple example that show the problems of Otsu thresholding which is the best known global thresholding method. We have built a 2D grayscale synthetic image of size 120×120 pixels shown in figure 2.1 on which Otsu thresholding was applied. We remark that such method is not able to segment some objects in the image as the one in bottom left nor small objects that are overlaid in bigger one as in the bottom right. Moreover, this method depends on the background level, the objects not being extracted correctly. A detailed explanations and comparisons will be discussed in the next chapter.

We propose to develop methods from algebraic topology to overcome the issues of image processing such as the one explained above. In fact, topology and algebraic topology represent an interesting field that produce powerful techniques in analyzing images [Car14, Ghr08a]. It gives the advantage of the dependency from prior parameters like volume or intensity values, and does not demand preprocessing steps. Also its strengths are revealed in their invariance to continuous deformations. The same results would be obtained if the image was rotated, stretched, rescaled, etc. All these criteria makes it suitable to achieve more general tasks than the existing methods.

For this purpose we will present in the next sections, the concepts issued from topology and algebraic topology and specially the homology theory to explain their constructions on spaces of points like pixels. In the next chapters, we will see how algebraic topology succeeded in developing many appropriate techniques for image processing.

2.2 Topology

After developing some of the techniques used in image processing tasks, we aim in this section to explain some notions of general topology and to extend some of the applications of topology in engineering applications and in image processing field. Noting that topology is divided to many subfields such as general topology or point set topology, combinatorial topology, algebraic topology and differential topology which have relatively many concepts in common.

2.2.1 Notions of topology

Topology is a more invariant approach of geometry that is concerned in the study of all different sorts of spaces [Mun75]. The most important thing that differs topology from other kinds of geometry is the type of transformations allowed before considering something changed. It is called "rubber-sheet geometry" because objects in topology are allowed to be stretched and contracted without regard to size and absolute position like rubber but must not broken. A famous quote says that the topologist is a man who does not know the difference between a coffee cup and a donuts.

In this context, topology aims to study qualitative properties of objects called topological spaces that are invariant under continuous transformations and deformations [Cro05]. In an explicit way, topology analysis the shapes and their properties like connectedness and compactness, deformations of these shapes and mappings between them. So it looks for holes and their properties, the boundaries of mathematical objects and to extract general information from spaces.

Concerning the relation of topology and the study of images, topology aims to interpret spaces with the notion of neighborhood between points [AF08]. This notion of neighborhood must be taken in the large sense; it does not necessarily depend on a given metric because it will produce quantitative results. This notion of proximity is encoded due to a set of subsets, where the entire space is stable by union, by finite or infinite intersections and by complementary which define the notion of topology. These subsets are called the opens of the space and describe the topology.

The continuous functions are functions between spaces that respect the topology, in other words, the notion of neighborhood of the domain is transported to the the codomain. If these notions are equivalent for the domain and codomain spaces, the spaces are said homeomorphic and the function as homeomorphism. Topology provides very flexible invariants on the spaces like the number of connected components, the compactness, the type of surfaces. This flexibility comes from the generality of continuous functions.

We introduce now some basic and formal notions useful for constructions and tools in analyzing the images. The book [Bou66] is one of the first in understanding general topology, also in [Mor89] the author provides a yearly updated modern reference of general topology. There is also a formal definition for topology defined in terms of set operations.

Topology: A topology on a non empty set X is a collection T of subsets of X if the subsets in T are conform to the following properties:

1. The empty set $\emptyset$ and the set X are in T.
2. The intersection of finite number of sets in T belongs to T.
3. The union of finite or infinite sets in T is belongs of T.

The sets that belong to the topology T are called **open** sets and the pair (X, T) of a set X and a topology T is a **topological space**. Referring to this pair, the T will usually be

omitted, and X will be used to refer to the space and the set when there is no ambiguity. These opens carry several topological characteristics and the topological spaces can be related by functions that have several properties:

- **Neighborhood:** a set of X is called a "neighborhood" of an element a of X if it contains an open set of T containing a.

- **Open cover:** An open cover of X is a family of opens sets O_α such that each O_α is an element of T and $X \subseteq \bigcup O_\alpha$.

Relations between topological spaces are provided by functions that may transport topological properties from a space to another. We can distinguish:

- **Continuous functions:** a function $f : X \to Y$ is continuous if for every open set O in Y, $f^{-1}(A)$ is open in X.

- **Homeomorphisms:** a function $f : X \to Y$ between two spaces X and Y is called a homeomorphism if it is bijective, and f and its inverse are both continuous. In this case, we say that X and Y are homeomorphic.

- **Topological invariants:** it's any property of topological spaces that is invariant under homeomorphism. Connectedness is an example of topological invariants. Two points a and b are said to be connected in a space X if there exists a path from a to b consisting entirely of points included in X. Connectedness in X is an equivalence relation and the classes defined by this relation are called the connected components of X.

In addition, types of topological constructions, can be seen from metric or distance point of views that are analogue to the notions of opens and neighborhood:

- **Metric** A metric or distance function $d : X \times X \to \mathbb{R}$ is a function satisfying the following axioms:

  1. Positivity: For all $x, y \in X$, $d(x, y) \geq 0$.

  2. Non-degeneracy: If $d(x, y) = 0$, then $x = y$.

  3. Symmetry: For all $x, y \in X$, $d(x, y) = d(y, x)$.

  4. The triangle inequality: For all $x, y, z \in X$, $d(x, y) + d(y, z) \leq d(x, z)$.

- **Open ball:** the open ball $B(a, r)$ with center a and radius $r > 0$ with respect to metric d is defined to be $B(x, r) = \{y | d(x, y) < r\}$.

2.2.2 Applications of topology

Recently, topology has become an interesting and valuable component of applied mathematics, with many mathematicians and scientists employing and engaging concepts of topology to model and understand real world structures and phenomena. Topological methods have shown in the past decade to be a promising new paradigm for analyzing and manipulating many scientific and engineering problems.

Engineering problems

- Image processing: topological methods have been broadly used in image processing because of the structures given by pixels, objects shapes, and forms of the image specially in biomedical applications. As a strong advantage, they permit to build a space formalism that allows a construction of methods and algorithms inherently invariant to transformations such as stretching, rescaling and other continuous transformations [XZC+16].

  Hence, involving topology in image segmentation can considerably improve the segmentation accuracy and reliability. For example, in [WLW+15] the authors propose an automatic segmentation method for spinal canals by extracting their topology represented by a medial line that is computed using the connectedness properties of curves. An object segmentation tool is discussed in [DSM12] and integrates the topology preserving of the active membrane with its architecture ans shape.

  Moreover, digital topology deals with the topological properties of the numeric representation of images. Considering the images as discrete arrays in two or more dimensions, it presents the theoretical foundations for many image processing operations such as connected component labeling and computation, contour filling, boundaries following etc. The authors in [SSB15] provide a survey on many digital topology techniques used in the literature such as connectivity and tracking, distance transformations, local topology, minimal path computation and object characterization etc.

- Signal processing: topology has contributed in many signal processing aspects such as wireless sensor networks, communications systems and network protocols etc. The authors in [LZL15] propose a topology control algorithm for signal irregularity to solve the control problem of underwater wireless sensor networks.

  The problem of moving entities with models that exhibit static topology is solved by proposing time-varying topologies that use the notions of neighborhood and metric and overcome the classical work flows proposed in this problem [Bar16]. A topology discovery protocol for an independent path that reduces not only the topology updating time but also traffic to the controller is proposed in [CKL17].

- Artificial intelligence: topology have helped recently in modeling and predicting the work of intelligent machines. In [KM16], the authors uses artificial neural network,

distances between agents and topologies of structure networks to predict opponent movements in fighting games. A study of the shortest path internet protocol routing using artificial intelligence and topology is evaluated in [SNK17].

- Electrical engineering: a robot must often have a path planning to its environment that contains many obstacles and candidate partners and neighbors. Capturing and representing the topological properties of the space in which the robot may move or studying the neighbors with whom it may communicate are of major interest. The work in [NHZ+17] is concerned with the collective behaviors of robots beyond the nearest neighbor rules when robots interact with others by applying angle tests. In [AAR16], the authors propose a path planning algorithm for robots using topological notions to avoid collisions with other robots.
- Pattern analysis: topology have played an interesting role in the automatic detection of patterns in data. In this sense, the authors in [PK15] study the pattern detection of geological faults like seismic movements using topology and shape optimization. A method for investigating designs of interlocking geometric shapes as inspiration for structural topologies that can be used in lightweight temporary sheltering systems is discussed in [TWS+16].

Sciences problems: topological tools were significantly employed in scientific fields such as biology, chemistry and physics. For example, the authors in [ABB+15] elaborate rigorously the relation of topological methods on chemical concepts. Also modeling protein chains using topology is a well developed concept in the biological field [BTM16].

Social sciences: topology had contributed in resolution in many social and economic problems. In [ZPR+16], the authors analyze the topology of the network of credit card transactions data and their confidentiality and sensitivity aspects. Also topology had been involved in connectivity studies of social networks like in [BQC+17].

While topology is developed basically to deal with intuitions about spaces, connectivity, continuity and notions of distances and proximity, algebraic topology, which is a subfield of topology, aims to add to the topological concepts an algebraic flavor using algebraic structures such as groups and vector spaces in order to remove the ambiguity in understanding spaces.

2.3 Algebraic topology

This section is dedicated to the developing of some notions and concepts of algebraic topology. More precisely, we will extend in details some types of cell complexes and concepts of homology theory. For this purpose, we begin by a brief introduction to some of the theories of algebraic topology and how topological spaces can be manipulated to different types of cell complexes. Then we explain how we can transform topological spaces

issued from input data to a combinatorial representation into cell complexes and chain complexes. A detailed explanations of homology groups of different dimensions in their absolute and relative forms are discussed afterward. Next, we describe a technique for computing these homology groups and we illustrate it by a complete example. We finish by presenting some applications of algebraic topology in the state of the art applications.

2.3.1 Panorama on algebraic topology

From the several topological methods in the literature, some methods used in studying topological spaces are given by algebraic topology [Mas91, Hat01]. Algebraic topology studies the global properties of spaces, but uses algebraic objects such as groups and rings to answer topological questions. While the general topological methods are concentrated on connectivity and connections between spaces, the algebraic topology methods are more concrete. It tries to transform a topological problem into an algebraic problem that is easier to solve or to compute. For example, each space can be associated to a group called a homology group. We can distinguish the torus and the Klein bottle from each other because they have different homology groups. The combinatorial structure of spaces are often used by algebraic topology to calculate the various groups associated to that space.

Algebraic topology was introduced by Poincaré towards the end of the XIXth century and the beginning of XXth century for quantifying the topological spaces with more general objects than only numbers like a) homotopy, fundamental groups and b) homology groups. It exists many relations between the homotopy and homology groups as the abelinazation of fundamental group and homology groups of first dimensions.

Recently, algebraic topology have emerged and contributed in many applications of real world and specially in understanding efficiently engineering problems [Ghr14]. Concerning the applications of methods and tools issued from algebraic topology, they consist of associating some discrete algebraic structures like homology classes or sheaves to topological spaces such as cell complexes built on set of pixels or any point cloud in order to understand their connectivity issues in any dimension such as the number of holes, voids, tunnels etc.

Homotopy and fundamental groups: Homotopy is a continuous deformation between two continuous functions f and g from a topological space X to another Y. It's an approach to analyze topological spaces by examination of different paths, whether they're loops or not, that exist in the spaces and their behavior within its holes. The map that relates X to Y can define a homotopy class and we say that X and Y are homotopy equivalent. Homotopy is a topological invariant since homotopy equivalence between spaces X and Y is preserved under homeormophism. Fundamental groups study the structure of homotopic loops or paths that start and end at same point, thus it can be more powerful in detection of holes.

Even though homotopy and fundamental groups carries important information concerning the studied topological spaces, they are hard to compute. Homology as topological invariant is more useful in applications even if it's less precise because it's easier to compute.

Homology theory It is one of the fields that give well understood tools that can be functional on the applications level. It represents a principal part of algebraic topology which accomplishes the connection between topological and algebraic concepts. Homology theory takes advantage of the properties of groups and their homomorphisms to analyze the properties of spaces and functions by relating to each space a certain sequence of groups, and to each continuous functions of spaces, homomorphisms of the respective groups. For example, such properties include the study of holes and voids of different dimensions of the spaces and the connections between dimensionalities insured by boundaries. This is a brief introduction to homology theory and we will talk in details about concepts of homology theory and homology groups in the rest of the section.

The appropriate reduction of a specific space may not produce changes in homology groups. This reduction can be realized using another theory issued from algebraic topology which is discrete Morse theory.

Discrete Morse theory The homology groups aren't modified if the base space makes retractions. More precisely, a space X and a retraction A have isomorphic homology groups. A retraction A is here a subspace of X obtained by a continuous function $F : W \times [0,1] \to X$ such that $F(0,.)$ is the identity on X and $F(1,X) \in A$. Since F is continuous, no "holes" can appear or disappear during the deformation of retraction. If the deformation is well chosen, the computation of homology groups of A becomes more simple than the initial space X.

The most simple method to describe this deformation consists of using a vector field on the surface X. This vector field is created by a potential function f sufficiently regular; the field is given by the opposite of the gradient of f. This differential aspect of Morse theory corresponds to a discrete version that facilitates the use of algorithms. The basic idea of the discrete Morse theory is to construct a discrete vector field that indicates how to reduce the initial space. This equivalence is done by transforming the vectors to pairs of incident cells, which ensure the transformation of the differential aspect of Morse theory to a discrete version. These concepts were introduced by Forman in [For98, For02] and made possible the use of discrete Morse theory in algorithmic way. Moreover, the simplification of functions using the connection between persistent homology and discrete Morse theory permitted to eliminate the noise from functions [BLW12] which made the input functions more simple and easy to manipulate.

Recently the mathematics of sheaves focused on many aspects of engineering and data understanding and many results have make this field a promising force at this level.

Sheaf theory Sheaf theory which is an abstract field of pure mathematics is derived from category theory. It is known by its ability to codify rather complicated concepts in topology and analysis. Recently, Ghrist has been a driving force behind applying the sophisticated sheaf theory to practical problems [Ghr08a]. Using the cellular sheaves, the global inference of data is concluded by transforming local information to more higher aspects [Rob14b]. We will not extend sheaves concepts in this paragraph since the chapter 4 will present the transformation of sheaf theory from pure topological concepts to applied ones.

2.3.2 Cell complexes and their types

The computational approach of topology and these theories remains hard on general spaces. It becomes necessary to make the spaces manipulable and thus defined combinatorially. The classic procedure is to approach a given continuous space in the form of a variety, differential or not, by a cell complex.

Given a finite set of points M, the general procedure used in computational topology implicates two steps. We first must approximate the topological space X of the set of points M associated with a notion of neighborhood with a combinatorial structure K. This can be done for example by a cubical or simplicial complex. We will cover the approaches of this step in the next paragraphs. At the second step we compute the topological invariants of K, the homology groups, which will provide us a close information about the topological properties of the space X. In the next chapter, we will concentrate on the most essential topological invariant in our methodologies for the applications on images: the persistent homology.

We present now the important combinatorial structures that we used in construction of the topological spaces.

Cell complex We start by the generalization of the combinatorial form of topological spaces were we construct n-blocks called n-cells using attaching maps to build the cell complex. We begin by defining a n-cell to be a space which is homeomorphic to B_n, the ball in dimension n, which will give the approximation of blocks to the term of topological spaces.

A cell complex or a CW complex is constructed by induction from X^0 that is a discrete, finite set of points of M regarded as 0-cells or vertices. Then we form the n-skeleton X^n from X^{n-1} by attaching n-cells who are regarded as open balls because of the homeomorphism via attaching maps from the boundary of the n-cell to the $n-1$ skeleton. This process ends and $X^n = X$, for some finite $n \in \mathbb{N}$ for the cell complexes and for any $n \in \mathbb{N}$ for CW complexes. In our case, we will stick to cell complexes since the spaces that we build on images are finite.

Figure 2.2: Cubical complex of dimension 1.

We will present now some types of cell complexes such as cubical and simplicial complexes that are used in our applications:

Cubical complex A cubical complex is a type of cell complexes that can be manipulated naturally in Euclidean space by n-dimensional blocks, the n-cells. The approach to decomposition of the cubical complex, in any dimension, is deduced from the cell complex principles to the following: the n-dimensional space is composed of cells in such a way that p-cells are attached to each other along $(p-1)$-cells, for $p = 1, 2, ..., n$.

Starting with the dimension 1, in figure 2.2 a 0-cell or a vertex is n with $n = \ldots, -2, -1, 0, 1, 2, 3, \ldots$; a 1-cell or an edge is $(n, n+1)$ with $n = \ldots - 2, -1, 0, 1, 2, 3, \ldots$ the 1-cells are attached to each other along 0-cells.

For the dimension 2, cubical cells are defined for all integers n, m as:

- a vertex, or a 0-cell, is $n \times m$;
- an edge, or a 1-cell, is $n \times (m, m+1)$ or $(n, n+1) \times (m)$;
- a square, or a 2-cell, is $(n, n+1) \times (m, m+1)$.

The 2-cells are attached to each other along 1-cells, and 1-cells are still attached to each other along 0-cells.

For example, as shown in figure 2.3, the 1×2 is a 0-cell ; $(2, 3) \times 1$ and $2 \times (2, 3)$ are 1-cells and $(1, 2) \times (1, 2)$ is a 2-cell.

Concerning the 3 dimensional cubical complex, for all integers n, m, k we have:

- a vertex or a 0-cell, is $n \times m \times k$;
- an edge or a 1-cell is $n \times (m, m+1) \times k$, $(n, n+1) \times m \times k$, $n \times m \times (k + k + 1)$;
- a square, or a 2-cell, is $(n, n+1) \times (m, m+1) \times k$, $(n, n+1) \times m \times (k + k + 1)$ or $n \times (m, m+1) \times (k + k + 1)$;
- a cube, or a 3-cell, is $(n, n+1) \times (m, m+1) \times (k, k+1)$.

Simplicial complex As a type of cell complexes, simplicial complex provides a good way for representing combinatorial aspects of topological structures.

A given set of points $\{a_0, a_1, \ldots, a_d\} \in \mathbb{R}^n$ is geometrically independent, or affinely independent, if the equations $\sum_{i=0}^{d} \alpha_i a_i = 0$, and $\sum_{i=0}^{d} \alpha_i = 0$, where α_i are constants, hold

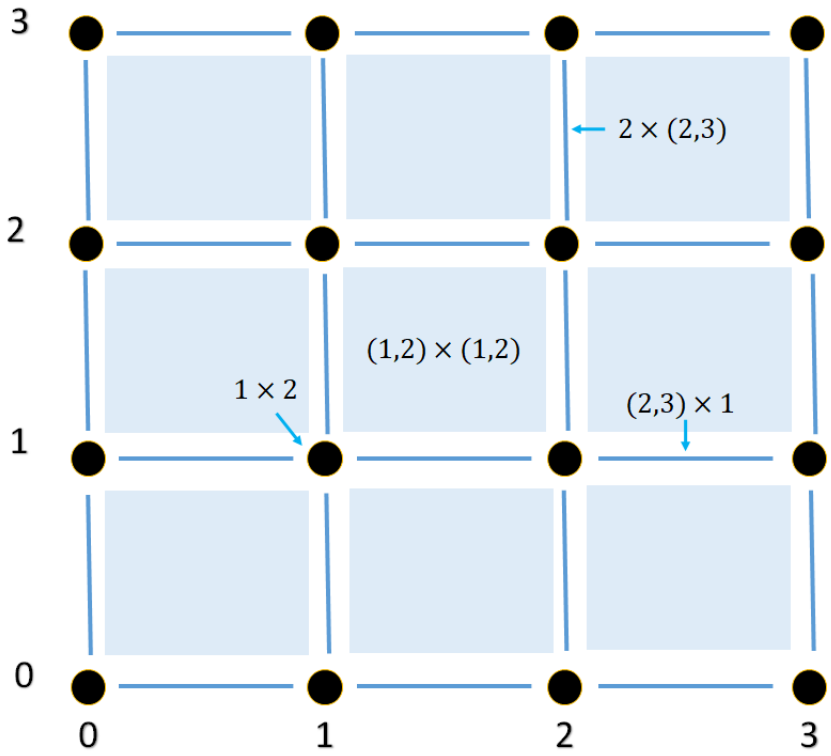


Figure 2.3: Cubical complex of dimension 2

only if each $\alpha_i = 0$. In $\mathbb{R}^d$, we can have at most $d-1$ geometrically independent points. A combination $x = \sum_{i=0}^{d} \alpha_i a_i$ is a convex combination if $\sum_{i=0}^{d} \alpha_i = 1$ and all α_i are non negative. The convex hull of a given point set $\{a_0, a_1, \ldots, a_d\}$ is the set of all convex combinations, denoted as $Conv(\{a_0, a_1, \ldots, a_d\}) = \{\sum_{i=0}^{d} \alpha_i a_i | \sum_{i=0}^{d} \alpha_i = 1 \text{ and } \alpha_i \geq 0\}$.

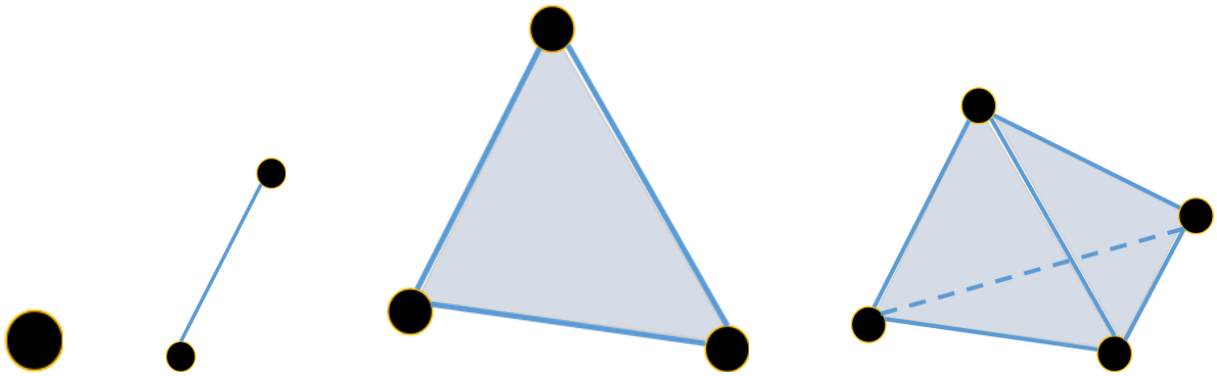
A d-simplex τ is the convex hull of $d+1$ geometrically independent points $\{a_0, a_1, \ldots, a_d\}$, i.e., $\tau = Conv(\{a_0, a_1, \ldots, a_d\})$. We can also say that the point set $\{a_0, a_1, \ldots, a_d\}$ spans τ. The d is called dimension of σ, denoted as $\dim \sigma = d$. The first dimensional simplices hold their own names: 0-simplex, 1-simplex, 2-simplex, and 3-simplex are also called vertex, edge, triangle and tetrahedron, respectively, shown in figure 2.4.

Any non-empty subset S of a point set $\{a_0, a_1, \ldots, a_d\}$ spans a simplex $\sigma' \subset \sigma$ called a face of σ.

After defining the notion of simplex and its faces, we can describe a simplicial complex K as a collection of simplices such that:

1. If $\sigma \in K$, then for any face σ_i of σ we have $\sigma_i \in K$,
2. For two simplices σ_i , $\sigma_i' \in K$, $\sigma_i \cap \sigma_i'$ is either $\emptyset$ or a face of both σ_i and σ_i'.

The set of simplices shown in figure 2.5 at left is a simplicial complex, whereas the one at right is not a simplicial complex because it doesn't satisfy the second condition cited above. According to definition, we can see K as a combinatorial representation of a



(a) Vertex. (b) Edge. (c) Triangle. (d) Tetrahedron.

Figure 2.4: Simplices of different dimensions.

topological space. The dimension of K is the highest dimension of any of its simplices.

Simplicial complexes can be hard to deal with. The requirement presented is that for anything other than small complexes, there are a big number of simplices which have to intersect in specific ways, as required by definition of simplicial complexes. Abstract simplicial complexes allow us to avoid this complication.

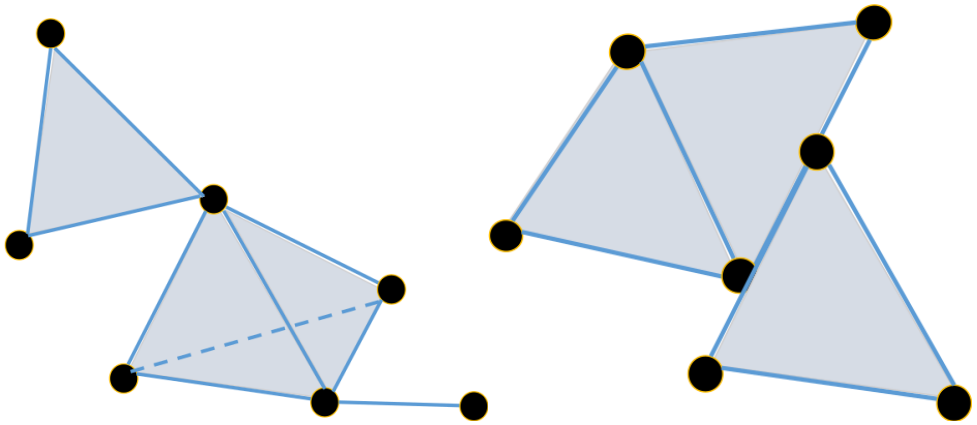
Abstract simplicial complex An abstract simplicial complex K is a pair (V, S), where V is a finite set, whose elements are vertices, and S is a set of nonempty finite subsets of V. These subsets are simplices, such that all singleton subsets of V are in S and if $\alpha \in K$ and $\beta \subseteq \alpha$, then $\beta \in K$. Noting that the abstract simplicial complex is a purely combinatorial description of the simplicial complex and does not need the property of intersection of simplices.

One of the most well known abstract simplicial complexes is the Čech complex that relies on the notion of balls and distances.

Čech complex Given an open cover of a set of points V, $O = \{O_i\}_{\{i \in I\}}$, where I is some indexing set, the nerve of O is the nonempty set, denoted by $Nerve$, given by:

$$Nerve(O) = \begin{cases} \emptyset \in Nerve \\ \text{if } \bigcap_{j \in J} O_j \neq \emptyset \text{ for } J \subset I \text{ then } J \in Nerve. \end{cases} \tag{2.1}$$

The Čech complex of S, given a strictly positive number ϵ, is isomorphic to the nerve of the collection of balls $B(a_i, \epsilon)$. $\check{C}ech(\epsilon) = \{\sigma \text{ s.t. } \bigcap_{a_i \in \sigma} B(a_i, \epsilon) \neq \emptyset \text{ for } a_i \in S\}$.



(a) A simplicial complex. (b) Not a simplicial complex.

Figure 2.5: A simplicial complex comparison.

In this manner, two intersecting balls of the same radius will define an edge and three of them define a triangle and so on. Hence $\check{C}ech(\epsilon)$ is simply the abstract simplicial complex whose p-simplices correspond to non-empty intersection of $(p+1)$ balls of radius ϵ and centered at the $(p+1)$ distinct points of V. The figure 2.6 illustrates the Čech complexes $\check{C}ech(\epsilon_1)$ and $\check{C}ech(\epsilon_2)$ with $\epsilon_1 \leq \epsilon_2$ built on a set of points. It's obvious that $\check{C}ech(\epsilon_1) \subset \check{C}ech(\epsilon_2)$. The choice of a specific ϵ will reduce the information issued from the complex. This issue is studied by the notion of persistence as we will see in section 3.1.

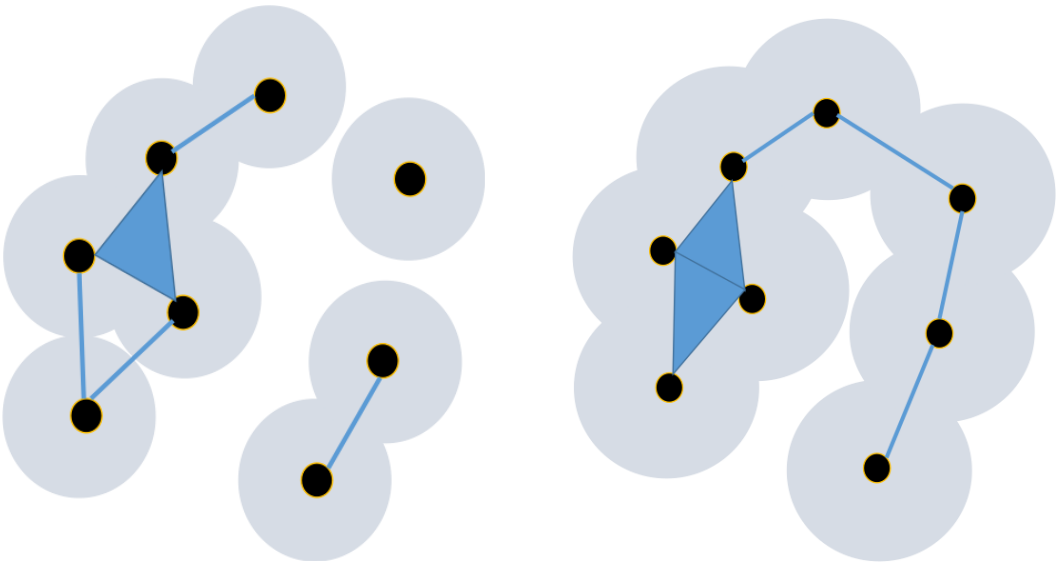
It exists many other examples of simplicial complexes like Vietoris-Rips complex [Zom10b], alpha complexes [Koz07], Delaunay complexes etc [EH10].

2.3.3 Combinatorial representation

The spaces in our works must be represented by combinatorially using simplicial complexes or cubical complexes in order to be manipulated by algorithms. These spaces are directly defined by a set of points, then a set of edges, then a set of 2-cells, triangles or squares followed by cubes or tetrahedron.

A space X of dimension n is decomposed into cells of dimension $0, 1$ to n. The gluing between these cells is done via the notion of boundary: the boundary of a cell of dimension k is a set of cells of dimension $k-1$. A cell τ on the border of the σ is called a face of σ while σ is a coface of τ.

Spaces given by point clouds with a notion of neighborhood like images can be manipulated as combinatorial spaces or topological complexes that permit the computation of homology classes. These spaces defined combinatorially permit the use of algorithms. Algorithms that are gathered within a new branch of topology known as combinatorial algebraic topology [Koz07]. Noting that since topological spaces aren't always a cubical or simplicial complex, we can use the more generalized term, the cell complex.



(a) Čech complex corresponding to radius ϵ_1

(b) Čech complex corresponding to radius ϵ_2

Figure 2.6: Čech complexes with two different ball radii.

In a cell complex, disks, squares, triangles are 2-cells; circles, edges, are 1-cells; while vertices are 0-cells. The relation between these patches are given by their boundary. For example, the boundary of a triangle is given by its edges, which are patches of lower dimension. The set of all patches and the gluing information provided by the boundary will form the cell complex [Hat01].

The figure 2.7 shows a decomposition of an annulus with two holes to a cell complex. This annulus will serve as an example for our explanations on homology groups. This decomposition is composed from:

- 14 cells of dimension 0, its vertices: (a), (b), (c), (d), (e), (f), (g), (h), (i), (j), (k), (l), (m), (n).
- 21 cells of dimension 1, its edges: (ab), (bc), (cd), (de), (ef), (af), (ag), (bk), (cm), (dn), (ej), (fh), (gh), (gi), (hj), (ij), (jk), (kl), (lm), (mn), (nl).
- 6 cells of dimension 2: (aghf), (abkjig), (bcmlk), (cdnm), (dnlkje), (efhij).

The boundary of a 2-cell is composed from its edges. For example the boundary of $(cdnm)$ is the sum of (cd), (dn), (nm) and (cm). The vertices don't belong to the boundary of $(cdnm)$.

Data structures that allow manipulation of combinatorial spaces are especially relying on the border relationship. Indeed, it suffices to retain the cells and the relation of incidence between faces and between cofaces [BM12].

The topological invariants proposed by the general topology are often qualitative and therefore difficult to calculate for a computer. The algebraic topology [Mas91, Hat01,

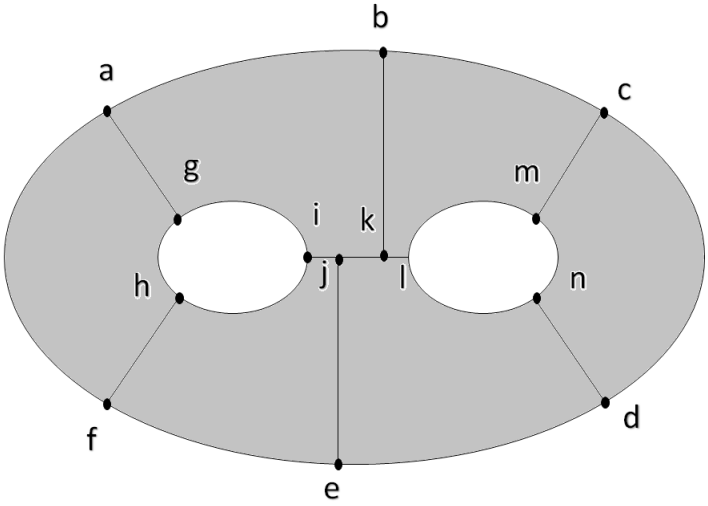


Figure 2.7: Example of a combinatorial space containing 0-cells, 1-cells and 2-cells.

Koz07] responds to this by calculating more flexible invariants. These invariants are not necessarily scalar values but rather groups, vector spaces or other algebraic structures.

We are interested mostly in homology groups as topological invariants for our combinatorial construction on images. The topological structure of a complex is encoded by vector spaces used by homology theory. We can build a linear structure on top of the complex by defining p-chains c_p as formal sums of p-cells. We also choose a coefficient group. The coefficient of chains is often taken as integer or belong to a field. Moreover, the use of the binary field $\mathbb{Z}_2$ permits to omit the orientation of the cells because an element is the equal of its opposite. Thereafter, when the loss of orientation information is not primordial in our case we can use the binary field.

A chain is therefore represented as a collection of cells. For example, on figure 2.7, combinations a, $a+d$ and $e+f$ are 0-chains. In a more general manner, each p-chain may be expressed uniquely as $c_p = \sum_{i=1}^{n_p} \alpha_i \sigma_i$ where σ_i represents p-cells, n_p is the number of p-cells in the space and α_i belong to the chosen field, taking into account the corresponding sign with the orientation in the space.

The set of all p-chains together with the operation of addition form a vector space C_p. The collection of the $(p-1)$-dimensional faces of a p-cell σ, which is a $(p-1)$-chain, is the boundary $\partial_p \sigma$ of σ. The boundary of the p-chain c_p is the sum of the boundaries of the cells σ_i in the chain, i.e. $\partial_p c_p = \sum_{i=1}^{n_p} \alpha_i \partial_p(\sigma_i)$ where n_p is the number of cells in the chain c_p.

Relations between chains is insured by its boundaries. The boundary operator ∂_p is a linear map between chains of different dimensions, $\partial_p : C_p \to C_{p-1}$. The set of vector

spaces C_p and the boundary operator ∂_p between them are called a chain complex and is noted:

$$0 \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \xrightarrow{\partial_{p-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0. \tag{2.2}$$

Since the complex is of dimension n, it doesn't exist a cell of dimension n+1. Consequently, the vector space C_{n+1} is trivial which explain the null vector space at left of 2.2. As well as the dimension -1 is here non-existent hence a trivial space for C_{-1} and the presence of the second zero of equation 2.2.

The fundamental relation of a chain complex is that the boundary of a boundary is void, that is, $\partial_p \partial_{p+1} = 0$ for all p. Intuitively, the boundary of a disk is a circle that has no boundary. This relation will play a striking role in definition of homology groups.

The linearization of the space of interest into a series of vector spaces and the linearization from the notion of boundary to a linear operator makes it possible to use linear algebra to compute the homology invariant.

2.3.4 Homology groups

Homology is the topological invariant that is often employed in practice because of its ability to be computed by linear algebraic methods in all dimensions, and thus by matrices manipulated by algorithms. This algebraic group describes the connectivity of a space X through the structure of its holes. This mechanism is executed by using equivalence classes of cycles called homology classes. However, the calculation of these classes is difficult and computationally complex. The linearized version system that relies on the complexes presented in subsection 2.3.2 is much more accessible because it depends only on linear algebra, but it is less powerful because it discriminates less the topological spaces.

The complexes make it possible to represent a sub-space by a vector. However, all the sub-spaces are not necessary interesting in algebraic topology; homology groups keep the essential aspects of the interesting subspaces while removing the subspaces without interest.

After describing the combinatorial representation of the studied data into a chain complex, we introduce the intuitive concept of homology groups giving some examples to illustrate it.

2.3.4.1 Definitions of homology groups

First, we start by homology groups of zero dimension to get a feeling of the intuitive purpose of homology. The homology group $H_0(X)$ tries to capture the notion of connectivity of the complex X. For example, on the figure 2.8 we see that the red chain $\boldsymbol{c} = (ab) + (bk) + (kj)$ connects the vertex (a) and vertex (j). Therefore, (a) and (j) belong to the same connected component.

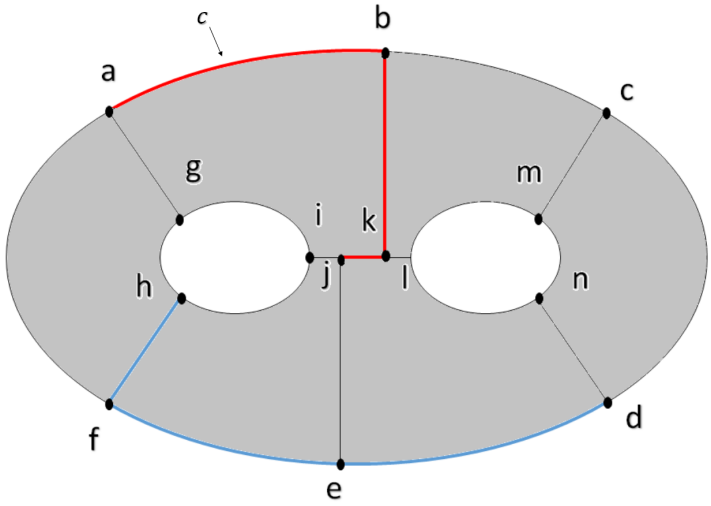


Figure 2.8: Example of chains of one dimension.

It remains to transport what we see towards linear algebra. The boundary of the chain $\boldsymbol{c}$ is $\partial\boldsymbol{c} = \partial(ab) + \partial(bk) + \partial(kj) = (b) - (a) + (k) - (b) + (j) - (k) = (j) - (a)$ since we have remained general and have taken into account the orientation of the cells. This last equation is written also $(j) = (a) + \partial\boldsymbol{c}$ or $(j) - (a) = \partial\boldsymbol{c}$ which expresses the fact that the vertex (j) is reached from the vertex (a) by following the chain of edges $\boldsymbol{c}$. We find therefore the notion of connection between (a) and (j); in other words, (a) and (j) are equivalent. Similarly, (d) and (h) are equivalent because the second chain, in blue, the figure 2.8 connects them.

An arbitrary chain of edges $\boldsymbol{c}$ connects its vertices $\partial\boldsymbol{c}$. Consequently, the subspace of C_0 given by the image of ∂C_1 provides the set of vertices connected to each other. The boundary of a vertex is always zero because $\partial_0 = 0$. So all the vertices are in $\ker \partial_0$, hence $\operatorname{Im} \partial_1 \subset \ker \partial_0$. The homology group $H_0(X)$ is written

$$H_0(X) = \ker \partial_0 / \operatorname{Im} \partial_1, \tag{2.3}$$

which is a vector space whose elements are equivalent vertices as (a) and (j) are.

As all the vertices of the figure 2.8 are reachable from (a) by following a chain, the group H_0 is a vector space of dimension 1 which means that there is only one connected component.

For other higher dimensions, naturally homology is a way to uncover p-dimensional holes in a cell complex. The idea is to find chains that surround holes without being able to be reduced continuously to zero. Noting, for the need of example, B, R and G as the three chains of dimensions 1 in the figure 2.9, which are in strong blue, red and green lines respectively. First of all a chain that surround a subspace or a hole is necessarily without boundaries. Boundaryless p-chains are interesting and form a subgroup of C_p that we call the p-th cycle group Z_p. The set Z_p of all p-cycles is defined as the subspace of C_p of

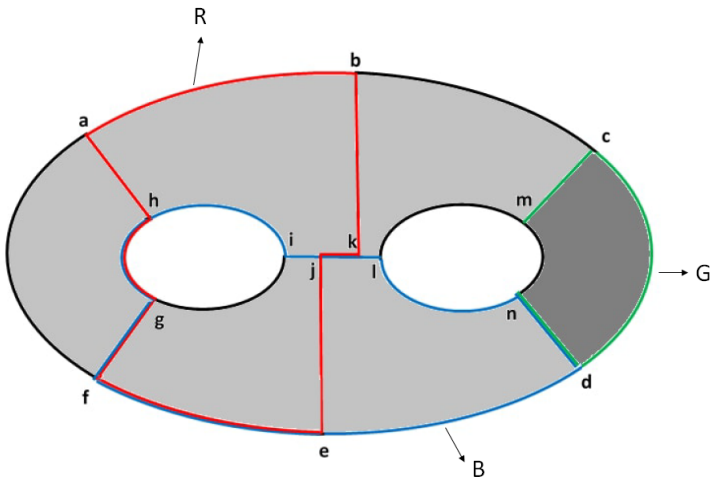


Figure 2.9: Example of a cell complex containing cycles and boundaries.

chains without boundary:

$$Z_p = \{x \in C_p \mid \partial_p x = 0\} = \ker \partial_p. \tag{2.4}$$

Among these cycles, we consider the ones that are boundaries of other chain of higher dimension while staying on the surface. It means intuitively that they can be collapsed into a point. They form a subgroup called the p-boundary group B_p:

$$B_p = \{x \in C_p \mid \exists y \in C_{p+1}, \partial_{p+1} y = x\} = \operatorname{Im} \partial_{p+1}. \tag{2.5}$$

For example, the chain in green $G = (cd) + (dn) + (nm) + (mc)$ displayed on figure 2.9, is a 1-cycle, because $\partial G = 0$. Chains B and R in figure 2.9 are also 1-cycles. Thus, they all belong to Z_1. Cycle G is a boundary of the 2-chain $(cdnm)$, the dark surface in figure 2.9, and it belongs to B_1.

Since the boundary of a boundary is void, B_p is a subgroup of Z_p, thus a quotient group can be created. Figure 2.10 shows schematically the relationship between the different vector spaces concerned by homology. The goal of homology is to discard cycles that are also boundaries because it can not contain voids and thus reduced to zero.

To this purpose, we build an equivalence relation on Z_p. The equivalence relation $\sim$ defined above partitions the p-cycles Z_p into a union of disjoint subsets, called homology classes. A chain that belongs to Z_p and B_p at the same time, meaning that it is a cycle that doesn't contain a void, will be reduced to zero and is not interesting, contrary to a chain that belongs to Z_p but not to B_p because it is a cycle that contains a void and isn't a boundary of a $p+1$-cell, thus it is interesting.

The homology group H_p keeps the count of cycles that are interesting by distributing them into equivalent classes. Thus, an element of H_p gathers together these equivalent cycles which can be deformed continuously one onto the other and a class of H_p will be represented by generators cycles.

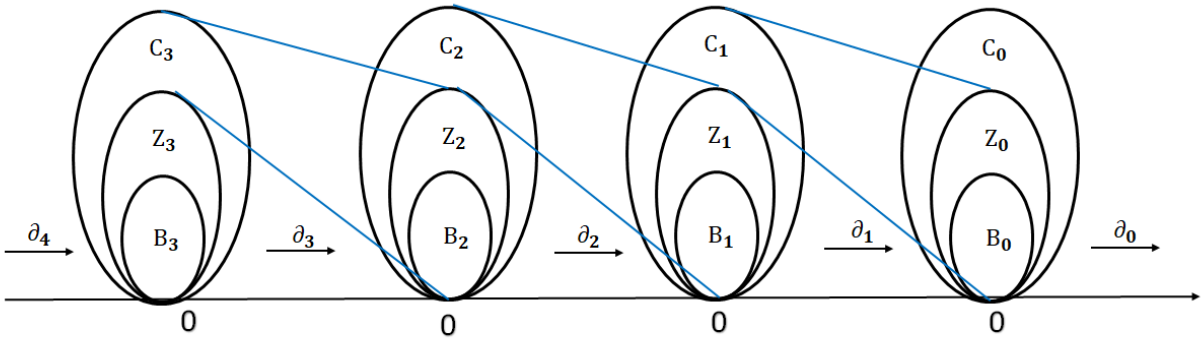


Figure 2.10: Chains, cycles and boundaries between dimensions.

Algebraically speaking, two cycles z_1 and $z_2 \in Z_p$ are said homologous or equivalent, written $z_1 \sim z_2$, if they differ by a boundary, i.e., $z_1 - z_2 \in B_p$. We say that z_1 and z_2 belong to the same class $[z]$. We let $[z]$ denote the homology class of $z \in Z_p$ and define the p-th homology of a space as the quotient of the vector space Z_p by the subspace B_p, which is a set of homology classes:

$$H_p = Z_p/B_p = cycles/boundaries = \{[z], z \in Z_p\}. \tag{2.6}$$

For example, the chain G on figure 2.9 does not surround a hole. Indeed, it can be shrunk into a point without being blocked by a hole. It belongs to Z_1 and B_1 and hence it is reduced to zero. Thus, this cycle is not of primary interest. On the other hand, chains R and B encircle a hole. They can not be shrunk while remaining on the space and they are not boundary of any 2-chain because of the hole. Hence, they belong to Z_1 but not to B_1. Also, the interesting 1-cycles R and B are inherently the same, they're said homologous as they surround the same hole. The difference between both cycles is a boundary of a 2-chain. Thus, the red cycle can be deformed continuously into the blue one. In fact $R = B + \Gamma$ where $\Gamma \in B_1$ and $\Gamma = (ab)+(bk)+(kl)+(ln)+(nd)+(de)+(ej)+(ji)+(ih)+(ha)$. Unfortunately, there are no canonical representatives, which means R and B are both valid for representing the hole in the annulus.

The dimension of H_p is called the p-th Betti number. For a 3D space, the zeroth Betti number counts the number of connected components. The first and the second Betti numbers counts the number of "holes" and enclosed "voids" respectively.

As a functorial property [Pie91, Mac71], every continuous map between topological spaces $f : X \to Y$ induces an homomorphism between homology groups $H_p(f) : H_p(X) \to H_p(Y)$ of these topological spaces at a dimension p. This functoriality means that we can associate algebraic data to each space and then attempt to transfer this information from space to space in order to make a comparison in homology classes. In this sense, homology vector spaces H_p transform continuous maps into linear ones. In addition, this functoriality will play a major role in persistent homology as we will see in section 3.1.

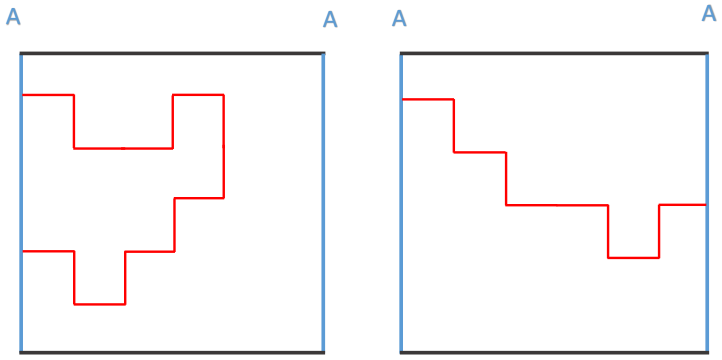


Figure 2.11: Examples of relative cycles where A is the subcomplex of vertical lines.

All computations involving homology groups can be easily handled using standard matrix manipulations given a combinatorial representation of the chain complex.

2.3.4.2 The relative homology group

Sometimes it's useful not to take into consideration a subcomplex A of the space X to compute the homology groups. We talk about another version of homology involving the dump of a subcomplex A that yield to the relative homology $H_p(X, A)$.

It is necessary for this purpose, to completely reduce the subspace A so that it "disappears" during the computing. The relationship of sub-part $A \subset X$ is represented by a subcomplex relation of $C_p(A)$, which is the complex that represents the space A at dimension p, and the complex $C_p(X)$, which is the linearization of X. We will refer to the classical homology described before as the absolute homology or simply the homology if we want to distinguish it from relative homology.

For that, we will note $C_p(X, A) = C_p(X)/C_p(A)$ as the quotient complex and $\overline{\partial}_p$ as the boundary operator, taking into consideration that A is always a subcomplex of X. Reducing $C_p(A)$ in $C_p(X)$ amounts to considering that the chains of $C_p(A)$ are null. The quotients vector spaces $C_p(X)/C_p(A)$ play this role. The elements of these vector spaces are chains of $C_p(X)$ where the cells of $C_p(A)$ are not taken into account. Thus, certain cycles may appear in $C_p(X)/C_p(A)$ whereas they did not exist in $C_p(X)$ and all cycles completely included in $C_p(A)$ disappear.

Therefore, taking chains on X modulo chains on A shrinks the requirement of a chain to be called a cycle, namely whenever its boundary is contained in A. This is illustrated in figure 2.11, where A represents the subspace containing the blue vertical lines at left and right. At left, the red chain is a cycle detected by relative homology since its boundaries lie in A. This also includes the case when the boundary is empty, which can be detected also by absolute homology. In the same way as the p-boundaries, relative p-boundaries

are defined as $B_p(X, A) = \operatorname{Im} \overline{\partial}_{p+1}$. Similarly, relative p-cycles are $Z_p(X, A) = \ker \overline{\partial}_p$ and correspond to p-chains c_p that satisfies $\partial_p c_p \in C_{p-1}(A)$ or $\partial_p c_p = 0$.

The relative homology groups $H_p(X, A)$ are computed as the homology groups that use these new vector spaces $C_p(X)/C_p(A)$. As in the absolute case of homology, we have $B_p(X, A) \subset Z_p(X, A)$ and the p-th relative homology groups is defined by $H_p(X, A) = Z_p(X, A)/B_p(X, A)$. However, for the previous definition of relative homology and its computation, it is necessary that $C_p(A)$ be either a subcomplex and in particular that the boundary of a chain of $C_p(A)$ belongs to $C_p(A)$.

2.3.4.3 Technique of computing homology groups

The calculation of the homology groups is done by matrix reduction [EH10]. If the coefficients of homology are in an algebraic field then the reduction is similar to a Gauss elimination and if the coefficients are integers then a reduction in Smith's normal form is possible but more demanding in cost of calculation. We now restrict ourselves to the first case since it's more simple.

If the boundary operator is expressed in a suitable base, the generators of the homology groups are read directly in the matrix representation of ∂. Since the matrix of this operator is given in a non-necessarily interesting base, we must make an intelligent base change that is provided by a matrix reduction.

This reduction can be computed by an algorithm equivalent to a Gauss elimination, and uses only elementary operations on the columns of the matrix $\boldsymbol{\partial}$. The pseudo-code is given on the algorithm 2.1 where $\boldsymbol{M}_{:i}$ means the i-th column in the matrix $\boldsymbol{M}$ and the rank of ∂ is computed to help in finding the reduced matrix Q.

These elementary operations can be written in the form of matrix multiplication to the right and can be inverted in the same form. Noting that the rank helps here in finding the rank of ∂ and thus the rank of Q. These operations are:

- adding a multiple of the column j to the column i;
- exchanging the columns i and j;
- multiplying the column i by a scalar α.

Thanks to the reduction of $\boldsymbol{\partial}$ we can find two matrices $\boldsymbol{Q}$ and $\boldsymbol{V}$ such that $[\boldsymbol{Q} \mid \boldsymbol{0}] = \boldsymbol{\partial}_p \boldsymbol{V}$ where the columns of $\boldsymbol{Q}$ are independent between them and V is an identity matrix in origin that registers the operations made on ∂ during the reduction operation. More

Algorithm 2.1: Reduction by column reduced echelon form

Input : Boundary matrix $\boldsymbol{\partial}$ of size $m \times n$
Output: Matrices $\boldsymbol{Q}$ and $\boldsymbol{V}$ such that $\boldsymbol{Q} = \boldsymbol{\partial V}$ and $\boldsymbol{Q}$ under reduced form and the rank of $\boldsymbol{\partial}$

Data: rank
Data: first: table of n indices

1 $\boldsymbol{Q} \leftarrow \boldsymbol{\partial}$
2 $\boldsymbol{V} \leftarrow \boldsymbol{I}_n$
3 rank $\leftarrow 0$
4 **for** $i \leftarrow 1$ **to** m **do**
5 $\quad \boldsymbol{c} \leftarrow i$-th column of $\boldsymbol{Q}$
$\quad$ /* Reduction of vector $\boldsymbol{c}$ */
6 $\quad$ **for** $j \leftarrow 1$ **to** rank **do**
7 $\quad\quad$ **if** $c_{\mathsf{first}[j]}$ *is non zero* **then**
$\quad\quad\quad$ /* Set to zero $c_{\mathsf{first}[j]}$ */
8 $\quad\quad\quad \boldsymbol{c} \leftarrow \boldsymbol{c} - c_{\mathsf{first}[i]}\, \boldsymbol{Q}_{:j}$
9 $\quad\quad\quad \boldsymbol{V}_{:i} \leftarrow \boldsymbol{V}_{:i} - c_{\mathsf{first}[i]}\, \boldsymbol{Q}_{:j}$
10 $\quad\quad$ **end**
11 $\quad$ **end**
12 $\quad \boldsymbol{Q}_{:i} \leftarrow \boldsymbol{c}$
$\quad$ /* update rank if necessary */
13 $\quad$ **if** $\boldsymbol{c} \neq \boldsymbol{0}$ **then**
$\quad\quad$ /* $\boldsymbol{c}$ is an independent vector of previous ones */
14 $\quad\quad$ rank $\leftarrow$ rank $+ 1$
15 $\quad\quad$ first[rank] $\leftarrow$ index of first non zero coefficient $\boldsymbol{c}$
16 $\quad\quad$ Sorted by ascending order first
17 $\quad\quad$ Perform the same permutations on the columns of $\boldsymbol{V}$
18 $\quad$ **end**
19 **end**

precisely, the former decomposition can be written in the form

$$\underbrace{\left[\underbrace{\begin{array}{cccccc} | & & | & | & & | \\ b_1 & \cdots & b_n & 0 & \cdots & 0 \\ | & & | & | & & | \end{array}}_{B_{p-1} \qquad \partial Z_p}\right]}_{\dim C_{p-1}} = \partial_p \underbrace{\left[\underbrace{\begin{array}{cccccc} | & & | & | & & | \\ v_1 & \cdots & v_n & z_1 & \cdots & z_m \\ | & & | & | & & | \end{array}}_{\partial^{-1} B_{p-1} \qquad Z_p}\right]}_{\dim C_p}. \tag{2.7}$$

By a direct reading, the vectors $z_1, \ldots, z_m$ form a basis of cycles Z_p. Similarly, the vectors b_1 to b_n are independent by construction and generate all boundaries of dimension $p-1$, which gives the relation $\dim C_p = \dim B_{p-1} + \dim Z_p$.

The previous reduction applied to the boundary operators permits to recover a base for all B_p and Z_p. The calculation of $H_p = Z_p/B_p$ consists about reducing the base of Z_p to that of B_k. This reduction is obtained for example by putting in reduced column form the matrix $[\boldsymbol{B} \mid \boldsymbol{Z}]$, where the columns of $\boldsymbol{B}$ and $\boldsymbol{Z}$ are the vectors of bases of B_p and Z_p respectively, to retrieve a matrix of form $[\boldsymbol{B} \mid \boldsymbol{H} \mid \boldsymbol{0}]$ where the columns of $\boldsymbol{H}$ are representative independent generators of H_p.

All previous reductions can be made in a single operation on a large matrix containing all the boundary matrices. The only reduction of this large matrix permits to directly recover the homology groups. It should be noted that this approach can be usable because the boundary matrices are very hollow matrices. Using this specificity and the adapted algorithms of reduction, the result can be manipulated in the memory of a computer.

2.3.4.4 Example of homology groups computing

In order to make the explanations above more concrete, we propose to compute the groups H_0, H_1 and H_2 of the cell complex of the figure 2.12. Noting that this space is in one piece with two holes and has no cavities, thus the number of connected component or β_0 is one. The number of one dimensional holes β_1 is two, while the voids of two dimensions do not exist, i.e. $\beta_2 = 0$.

The representing cell complex of the space in figure 2.12 is composed of three vector spaces. We will not make the distinction by the notation between a cell c and its associated vector (c).

- The vector space C_0 is of dimension 14 and is generated by the vertices (a) to (n). A chain is therefore of the form $c_a\,(a) + c_b\,(b) + c_c\,(c) + \cdots + c_n\,(n)$, where c represent coefficient of the cell in the chosen algebraic field. The vector $(d)+3\,(i)$, for example, represents the union of the vertex d and the vertex i with a coefficient of 3 if the field is $\mathbb{Z}$ for example;
- C_1 of dimension 21 is generated by the edges $(ab), (bc), \ldots$. For example, the chain $(ab) + (bc) + (cd) + (de) + (ef) + (fa)$ represents the turn of the outer circle;

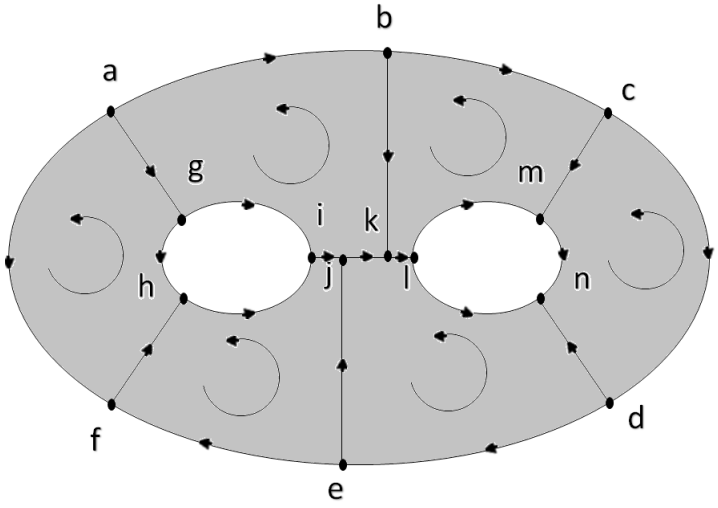


Figure 2.12: Example of a surface with two holes.

- C_2 of dimension 6 has as base the chains $(abkjig)$, $(bcmlk)$, $(cdnm)$, $(dejkln)$, $(efhij)$ and $(fhga)$ corresponding to the different cells $abkjig$, $bcmlk$, $cdnm$, $dejkln$, $efhij$ and $aegd$.

In the natural base of the cells, the boundary matrices are the incidence matrices of vertex-edge or edge-(2-cell). However, the coefficients will be taken as integers in our example to take into account the orientation of the cells which has been fixed and represented by the arrows on the figure 2.12.

The boundary of the edge (ab) is formed by its vertices a and b but by taking into account the orientation, we have $\partial_1(ab) = (b) - (a)$. This same process makes it possible to obtain the boundary matrix $\boldsymbol{\partial}_1$.

$$\boldsymbol{\partial}_1 = \begin{array}{c} \\ (a) \\ (b) \\ (c) \\ (d) \\ (e) \\ (f) \\ (g) \\ (h) \\ (i) \\ (j) \\ (k) \\ (l) \\ (m) \\ (n) \end{array} \begin{array}{c} \begin{bmatrix} \overbrace{(ab)} & \overbrace{(af)} & \overbrace{(ag)} & \overbrace{(bc)} & \overbrace{(bk)} & \overbrace{(cd)} & \overbrace{(cm)} & \overbrace{(de)} & \overbrace{(dn)} & \overbrace{(ef)} & \overbrace{(ej)} & \overbrace{(fh)} & \overbrace{(gh)} & \overbrace{(gi)} & \overbrace{(hi)} & \overbrace{(ij)} & \overbrace{(jk)} & \overbrace{(kl)} & \overbrace{(lm)} & \overbrace{(ln)} & \overbrace{(mn)} \\ - & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ + & \cdot & \cdot & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + & \cdot & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & + & \cdot & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & + & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & + & - & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & - & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & - & - & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & - \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & + \end{bmatrix} \end{array} \tag{2.8}$$

where $-$ indicates the entry -1, $+$ corresponds to the coefficient $+1$ and a point to 0. The orientation is chosen so that, from the notations, the arrow goes from vertex of the smallest identifier to the vertex of the largest identifier within the alphabetical order. Taking for example the first column represented by the 1-cell (ab) its boundary is $(b) - (a)$ then the entry corresponding to the row b is $+$ and that corresponds to row a is $-$. This convention makes it possible to avoid difficulties related to the memorization of the orientation and our implementations on machine follow it.

Similarly, following the orientation defined in the figure 2.12, the boundary operator is represented by the matrix ∂_2:

$$\boldsymbol{\partial}_2 = \begin{array}{c} \\ (ab) \\ (af) \\ (ag) \\ (bc) \\ (bk) \\ (cd) \\ (cm) \\ (de) \\ (dn) \\ (ef) \\ (ej) \\ (fh) \\ (gh) \\ (gi) \\ (hi) \\ (ij) \\ (jk) \\ (kl) \\ (lm) \\ (ln) \\ (mn) \end{array} \begin{array}{c} \begin{array}{cccccc} (abkjig) & (bcmlk) & (cdnm) & (dejkln) & (efhij) & (fhga) \end{array} \\ \left[\begin{array}{cccccc} - & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & + \\ + & \cdot & \cdot & \cdot & \cdot & - \\ \cdot & - & \cdot & \cdot & \cdot & \cdot \\ - & + & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & - & \cdot & \cdot & \cdot \\ \cdot & - & + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & + & \cdot & \cdot \\ \cdot & \cdot & - & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & - & \cdot \\ \cdot & \cdot & \cdot & - & + & \cdot \\ \cdot & \cdot & \cdot & \cdot & - & + \\ \cdot & \cdot & \cdot & \cdot & \cdot & - \\ + & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & - & \cdot \\ + & \cdot & \cdot & \cdot & - & \cdot \\ + & \cdot & \cdot & - & \cdot & \cdot \\ \cdot & + & \cdot & - & \cdot & \cdot \\ \cdot & + & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & - & \cdot & \cdot \\ \cdot & \cdot & + & \cdot & \cdot & \cdot \end{array}\right] \end{array}. \tag{2.9}$$

The orientation is chosen according to the trigonometric direction, all the edges of the boundary of a cell that follow this direction are counted positively and the others negatively.

From the boundary matrix $\boldsymbol{\partial}_1$, thanks to a reduction of columns in echelon form, we have a representation of the given form by the equation (2.7). By applying this reduction to (2.8), noting that V_1 is a square matrix, we find the decomposition $\boldsymbol{Q}_1 = \boldsymbol{\partial}_1 \boldsymbol{V}_1$ given by:

$$
\begin{array}{c}
\\
(a)\\(b)\\(c)\\(d)\\(e)\\(f)\\(g)\\(h)\\(i)\\(j)\\(k)\\(l)\\(m)\\(n)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccccccccc}
\overbrace{(ab)} & \overbrace{(af)} & \overbrace{(ag)} & \overbrace{(bc)} & \overbrace{(bk)} & \overbrace{(cd)} & \overbrace{(cm)} & \overbrace{(de)} & \overbrace{(dn)} & \overbrace{(ef)} & \overbrace{(ej)} & \overbrace{(fh)} & \overbrace{(gh)} & \overbrace{(gi)} & \overbrace{(hi)} & \overbrace{(ij)} & \overbrace{(jk)} & \overbrace{(kl)} & \overbrace{(lm)} & \overbrace{(ln)} & \overbrace{(mn)}
\end{array}\\
\left[\begin{array}{cccccccccccccccccccccc}
+ & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
- & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & - & - & - & - & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & - & - & - & - & - & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}\right]\\
\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{B_0 = \operatorname{Im}\partial_1}\ \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\partial Z_1}
\end{array}
=
$$

$$
\boldsymbol{\partial}_1
\begin{array}{c}
\\
(ab)\\(af)\\(ag)\\(bc)\\(bk)\\(cd)\\(cm)\\(de)\\(dn)\\(ef)\\(ej)\\(fh)\\(gh)\\(gi)\\(hi)\\(ij)\\(jk)\\(kl)\\(lm)\\(ln)\\(mn)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccccccccccccc}
\overbrace{(a)} & \overbrace{(b)} & \overbrace{(c)} & \overbrace{(d)} & \overbrace{(e)} & \overbrace{(f)} & \overbrace{(g)} & \overbrace{(h)} & \overbrace{(i)} & \overbrace{(j)} & \overbrace{(k)} & \overbrace{(l)} & \overbrace{(m)} & \overbrace{(n)} & \overbrace{(o)} & \overbrace{(p)} & \overbrace{(q)} & \overbrace{(r)} & \overbrace{(s)} & \overbrace{(t)} & \overbrace{(u)} & \overbrace{(v)}
\end{array}\\
\left[\begin{array}{cccccccccccccccccccccc}
- & + & + & + & + & \cdot & - & - & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & + & \cdot & \cdot & \cdot\\
\cdot & - & - & - & - & + & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & \cdot & - & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & + & \cdot & \cdot & \cdot & \cdot & + & - & + & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & + & + & + & \cdot & - & - & - & \cdot & - & - & \cdot & \cdot & \cdot & - & + & + & - & - & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & + & \cdot & \cdot & \cdot & \cdot & - & \cdot & + & + & \cdot\\
\cdot & \cdot & \cdot & + & + & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & - & \cdot & \cdot & - & + & + & \cdot & - & -\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & - & - & - & - & - & + & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & +\\
\cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & - & + & + & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & -\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & + & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & + & + & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & +
\end{array}\right]\\
\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\partial_1^{-1}B_0}\ \underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{Z_1}
\end{array}
\tag{2.10}
$$

Firstly, we remark that the first thirteen columns of $\boldsymbol{Q}_1$ are non zero and independent, because they're in echelon form. These columns represent then a base of $B_0 = \operatorname{Im}\partial_1$. On the other hand, the eight last columns of $\boldsymbol{Q}_1$ are null, thus the eight last columns of $\boldsymbol{V}_1$ that are associated with them have empty boundary. It follows that these last eight columns form a base of $Z_1 = \ker\partial_1$.

The same procedure of column reduction applied on the boundary matrix of $\boldsymbol{\partial}_2$ gives the following decomposition $\boldsymbol{Q}_2 = \boldsymbol{\partial}_2\,\boldsymbol{V}_2$

$$
\begin{array}{c}
\\ (ab)\\(af)\\(ag)\\(bc)\\(bk)\\(cd)\\(cm)\\(de)\\(dn)\\(ef)\\(ej)\\(fh)\\(gh)\\(gi)\\(hi)\\(ij)\\(jk)\\(kl)\\(lm)\\(ln)\\(mn)
\end{array}
\begin{array}{c}
\begin{array}{cccccc}(abkjig)&(bcmlk)&(cdnm)&(dejkln)&(efhij)&(fhga)\end{array}\\
\left[\begin{array}{cccccc}
+ & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & + & \cdot & \cdot & \cdot & \cdot\\
- & - & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & + & \cdot & \cdot & \cdot\\
+ & \cdot & - & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & + & \cdot & \cdot\\
\cdot & \cdot & + & - & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & + & \cdot\\
\cdot & \cdot & \cdot & + & - & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & +\\
\cdot & \cdot & \cdot & \cdot & + & -\\
\cdot & + & \cdot & \cdot & \cdot & +\\
\cdot & - & \cdot & \cdot & \cdot & \cdot\\
- & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & +\\
- & \cdot & \cdot & \cdot & + & +\\
- & \cdot & \cdot & \cdot & + & \cdot\\
\cdot & \cdot & - & \cdot & \cdot & \cdot\\
\cdot & \cdot & - & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & + & \cdot\\
\cdot & \cdot & \cdot & - & \cdot & \cdot
\end{array}\right]
\end{array}
= \boldsymbol{\partial}_2
\begin{array}{c}
\\ (abkjig)\\(bcmlk)\\(cdnm)\\(dejkln)\\(efhij)\\(fhga)
\end{array}
\begin{array}{c}
\\
\left[\begin{array}{cccccc}
- & \cdot & \cdot & \cdot & \cdot & \cdot\\
\cdot & \cdot & - & \cdot & \cdot & \cdot\\
\cdot & \cdot & \cdot & - & \cdot & \cdot\\
\cdot & \cdot & \cdot & \cdot & - & \cdot\\
\cdot & \cdot & \cdot & \cdot & \cdot & -\\
\cdot & + & \cdot & \cdot & \cdot & \cdot
\end{array}\right]
\end{array}
\tag{2.11}
$$

We remark that ∂_2 is of full rank. Therefore the columns $\boldsymbol{Q}_2$ are vectors of the base of $B_1 = \operatorname{Im}\partial_2$ and $Z_2 = \ker\partial_2 = 0$.

Once the basis of B_0, B_1, Z_1 et Z_2 are known, it remains the calculation of H_0 and H_1. Knowing that $H_0 = Z_0/B_0$ and $H_1 = Z_1/B_1$, therefore they represent the basis of Z_0 and Z_1 that cannot be written in form of the basis of B_0 and B_1 respectively. For this, it is enough to project the vectors of $Z_0 = C_0$ on the base B_0 and the vectors of Z_1 on the basis of B_1. To do this, the reduction of columns can be reused. In fact, the matrix $\boldsymbol{B}_0$ which contains the columns of a base of $\operatorname{Im}\partial_1$,under non-zero matrix of $\boldsymbol{Q}_1$, is already in echelon form. So if we want to reduce the matrix $[\boldsymbol{B}_1 \mid \boldsymbol{Z}_1]$, we obtain a matrix of form $[\boldsymbol{B}_1 \mid \boldsymbol{H}_1 \mid \boldsymbol{0}]$, where $\boldsymbol{H}_1$ provides the representatives of H_1 generators since they cannot be expressed in terms of basis of B_1.

Using this method on our example, we find

$$\boldsymbol{H}_1 = \begin{matrix} (ab) \\ (af) \\ (ag) \\ (bc) \\ (bk) \\ (cd) \\ (cm) \\ (de) \\ (dn) \\ (ef) \\ (ej) \\ (fh) \\ (gh) \\ (gi) \\ (hi) \\ (ij) \\ (jk) \\ (kl) \\ (lm) \\ (ln) \\ (mn) \end{matrix} \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ + & \cdot \\ - & \cdot \\ + & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & + \\ \cdot & - \\ \cdot & + \end{bmatrix}. \tag{2.12}$$

Also, the calculation of representatives of H_0 using the same approach will provide,

$$\boldsymbol{H}_0^T = \begin{matrix} (a) & (b) & (c) & (d) & (e) & (f) & (g) & (h) & (i) & (j) & (k) & (l) & (m) & (n) \end{matrix} \\ \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \end{bmatrix}. \tag{2.13}$$

In other words, H_0 is a vector space of dimension 1 generated by the equivalence class $(n) + B_0$. That is, the complex is composed of a single connected component.

The group H_1 is, for its part, of dimension 2 and generated by the two vectors $(gh)+(hi)-(ig) + B_1$ and $(lm) + (mn) - (ln)$. The two representatives cycles $u = (gh) + (hi) - (ig)$ and $v = (lm)+(mn)-(ln)$ that surround well the two holes in the space of the figure 2.12.

The group H_2 is zero because there are no cycles of dimension 2; $Z_2 = 0$. In other words, there is no cavity in the space of the example.

It should be noted that the order of the columns of the boundary matrices $\boldsymbol{\partial}_1$ and $\boldsymbol{\partial}_2$ greatly influences the representatives obtained by our algorithm. Indeed, it is possible that the algorithm returns, while remaining correct, that H_0 is generated by $(a) + B_0$.

Similarly, a change in order of the cells could provide as representatives of H_1 the vectors $(ab)+(bc)+(cd)+(de)+(ef)-(af)$ and $(gh)+(hi)-(ig)$. Here, the first cycle « surrounds »

the two holes and the second only the hole to the left of the Figure 2.12. There is no set of canonical representatives, that is to say, naturally defined. Similarly, no basis is canonical, for example $\{u, v\}$ forms a base just as valid as $\{u + v, v\}$ or $\{u, u + v\}$.

The dimensions of H_0 and H_1 give respectively the number of Betti $\beta_0 = 1$ and $\beta_1 = 2$.

2.3.4.5 Cycles optimization

Since homology groups are quotient groups $H_p = H_p = \ker \partial_p / \operatorname{Im} \partial_{p+1}$, the computed generators are not the unique representatives. It means that all the cycles surrounding the holes of dimension p and equivalent to a generator $\boldsymbol{c}$ are of the form $\boldsymbol{x} = \boldsymbol{c} + \partial \boldsymbol{y}$ where $\boldsymbol{x}$ and $\boldsymbol{c}$ belong to $\ker \partial_p$ and $\partial \boldsymbol{y}$ to $\operatorname{Im} \partial_{p+1}$.

It is sometimes interesting to find the cycle equivalent to a generator $\boldsymbol{c}$ which minimizes its size. This permits, for example, to locate a hole by knowing the vertices that are on its boundary. This optimization is written in the form

$$\begin{aligned} &\min_{\boldsymbol{x}} \operatorname{size}(\boldsymbol{x}) \\ &\text{such that } \boldsymbol{x} = \boldsymbol{c} + \partial \boldsymbol{y}. \end{aligned} \tag{2.14}$$

The size of a cycle $\boldsymbol{x}$ is not necessarily well defined because it can be the number of cells that form its base or the sum of its values according to the application. This type of approach is studied in [DHK11] and [Eri11] which propose to transform these optimization problems in linear programming.

These cycle optimization problems were discussed for applications to coverage holes for example [TSJ08]. Their approach is not combinatorial as here because the authors use the isomorphism between the kernel of the operator of Laplace-Beltrami Δ_1 and the group $H_1(X)$ by the discrete Hodge theory [ME06]. Thus, it becomes possible to distribute the computation of the homology group generator and to optimize its length by methods of decreasing gradient.

The article [DSW10] pursues the same objective but considering a complete basis of the homology group and the notion of size derives directly from the notion of distance underlying to the space for which the cell complex is an approximation.

Finally, the article [CF10] defines the notion of size by a very general and typically topological approach. No geometric notion is required. The idea is to have a set of sub-complexes that serve as stallions for size measurement. An algorithm for calculating a minimal basis is then provided.

All these algorithms are interesting but depend strongly on the application. Moreover, the choice of the cells that represent the optimal basis seems very complex because they require a global knowledge of the groups Z_1, Z_2, B_2 and B_3.

In our application, we intended to use algorithms that ensure a legible basis for the object segmentation in images. These algorithms consist on separating the homological basis in

2D and 3D cases as described extensively afterwards in the chapter 3 in subsections 3.3.5 and 3.3.6.

2.3.5 Applications of algebraic topology in engineering

After explaining in the previous section some notions of algebraic topology and after expanding in details the purpose of homology groups and how to compute them, we aim in this section to develop some hottest directions of applications of algebraic topology in different topics.

Relations between discrete and continuous worlds Recently, many developments that bridge the discrete and continuous worlds have appear in scientific researches for understanding the topological and geometrical aspects of networks and discrete metric spaces. Topological and geometrical tools help in treating discrete metric spaces as described in Gromov's work [Gro06].

The interface of algebraic topology and computer science gives birth to schemes that excessively progressed the comprehension of functions between discrete samples from metric spaces within structures such as algebraic sets and Riemannian manifolds. Studying the structure of large networks is done using these two approaches. Networks that represent a very important topic in engineering, computer science and social sciences are manipulated as simplicial complexes.

Considering these networks as topological spaces after enrichment of their structures permits to use tools from algebraic topology in biological, engineered and social networks. For example, the authors in [WMS13] propose a method to reduce the number of simplices that represent network structures. This reduction will simplify homology and hole location computations using isomorhphisms between homological groups that contribute to a collapse from bigger groups to smaller ones.

Sensing and communication Algebraic topology concepts have been involved in sensing and communication. The computation of optimal processing elements with limited local connectivity in systems that include multi agent robotics, sensor networks and cell phones is innovated in the transformation from local to global ensured by robust algorithms from algebraic topology principles. These principles solve the problem of performing whole system analysis by collecting local information to deduce or infer global results. This include detecting homology based coverage holes in wireless sensor networks, as the work initiated by de Sliva and Ghrist in [SG07] where homology groups of first dimension are computed to detect coverage holes, and those of second dimension for sensor selection, or as in [YMD14], where a measure of the accuracy of Rips complex in addressing coverage holes problems is proposed. Also the study of network data optimization and aggregation is ensured by sheaf theory [Rob13b] that extends locally defined structures to globally accurate interpretations.

Dynamics and differential equations Algebraic topology is also explored and analyzed by dynamics and differential equations. Invariants of dynamical systems that are well adapted to numerical methods can be computed thanks to homological quantities and characterizing the qualitative behavior of dynamical systems is ensured by algebraic topology invariants. Examples cover recording patterns of nodal domains [MW07], modeling vector field topology using Morse decomposition on a manifold surface transformed to simplicial complex [SZ12], homology based methods that measure dynamical finite size effects in spatiotemporally chaotic convective flows [HKS11] and techniques of time series analysis that solve the non stationarity problem using computational topology [Rob00].

Molecular biology Many fundamental questions in molecular biology are based on phenomena raised from topological features of metric geometry of proteins. Many challenges in biology need the qualitative outcome that topology ensures. Persistent homology have been used to analyze the protein structure, flexibility and folding, the authors in [XW14] introduce a technique for obtaining molecular topological fingerprints depending on the persistence of molecular topological invariants computed from persistent homology. A way for protein-ligand binding affinity prediction that integrate persistent homology with machine learning techniques is proposed in [CW17]. Studying behavior of genes using persistent homology helped in detecting cancer forms in [LK15].

Homology and cohomology The dual version of homology called cohomology handles many applications where the usual homology doesn't find the expected results. Algebraic structures on the cochain complex that consist of transforming chains by a coboundary maps from low dimensions to higher dimensional structures. Recent applications of cohomology in analyzing the families of forms in electrodynamics and fluid dynamics focuses on creating new ways of computing cohomology generators [DS13]. Identificatin of interesting circle structures in data is addressed in [dSMVJ11a]. Persistence of cohomology classes are used combined with integration to get the circle valued functions in order to solve a problem of non linear dimensionality reduction.

Notable advances have been marked in connecting cochain complex to issues of stability in finite element methods [AFW06]. The relation of cohomology with Hodge-De Rahm theory is the core of many topics of cohomology computation using the hodge laplacians on graphs [Lim15]. In our work, we relied on the notion of sheaves cohomology in chapter 4 to deduce inferences in scale analysis and localization.

Homotopy theory Homotopy theory was largely applied in engineering applications. Predictions of paths for robots and machines have been considered also by homotopy theory [HCR15, DAVLHM+17]. Homotopy classes aids in automatic path planning methods that pass obstacles neighborhood and ensure the optimal path.

For example, in [OK13] a method that uses homotopy and algebraic topology concepts is used to assist people in maintaining very complex systems by ignoring unnecessary

details. This approach called incrementally modular abstraction hierarchy (IMAH) and that relies on the fact of hiding information and managing complexity at the same time of making individuals work independently at each level of the hierarchy. It is widely used in computer science and information technology. The authors use a combination of homotopy theory, category theory and set theory to derive general concepts derived from algebraic topology and accessible by non mathematician community. They involve pullback and pushout morphisms instead of homotopy and employ them when dividing a system or integrating two subsystems and when descending and ascending an abstraction hierarchy which are very important operations in IMAH.

Persistent homology Latterly, many techniques have appeared to try to deduce algebraic properties of a space from finite metric subspaces of sampled points. Computing approximations of homology of manifolds is a basic problem in large data like images. An important issue in this topic is the dependence of the topological invariants computed from parameters or fixed threshold levels. This is the role of persistent homology.

Like homology, persistent homology concepts try to understand the relationship between the underlying geometry of the space in question and sampling procedure. This type of difficulty is classical in methods of data analysis based on certain geometry where a scale parameter is often required to know the best representative space of the data. The topological data analysis [Car09, Car14, Ghr17] provides a very elegant solution thanks to the use of homological persistence. The idea is to build the complex as previously while labeling the cells in the complex by certain values. The concept is to build subcomplexes using these values in a way to have sub-complexes of simplexes labeled by a value below the threshold. Therefore, by varying the threshold, we can construct a scheme called filtration on the complete simplicial complex.

This filtration will be very relevant for our application as we will see uin the next chapter. The complete complex and the filtration makes it possible to calculate the persistence of the homology groups. This topological invariant has led to a large number of analysis explanations of data like in molecular biology as we have seen, neuroscience [Baa17], fluid dynamics [KLJ+16], etc.

2.4 Application objectives of the thesis

In this work, we aim to apply algebraic topology concepts to analyze images specifically and to solve some of the problems faced by the majority of image processing techniques at the level of prior parameters, background and overlaid objects etc.

We aim in this section to explain some of the application objectives of this thesis how topological invariants computation can help in image processing tasks specially in biomedical aspects.

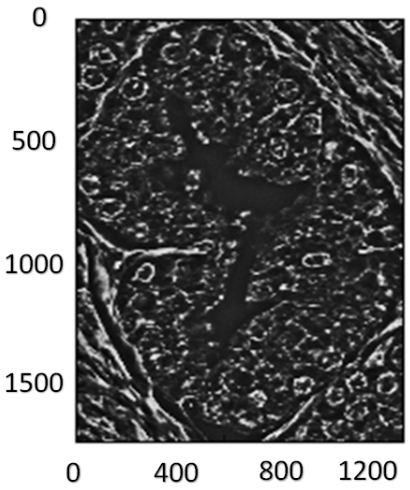
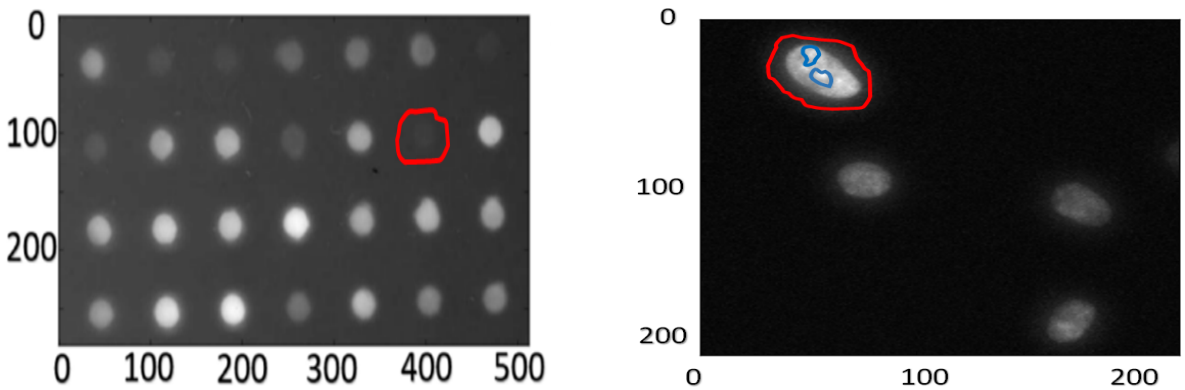


Figure 2.13: An image of prostate gland.



(a) Dotsimage. (b) Cells image.

Figure 2.14: Sample of real images for object segmentation.

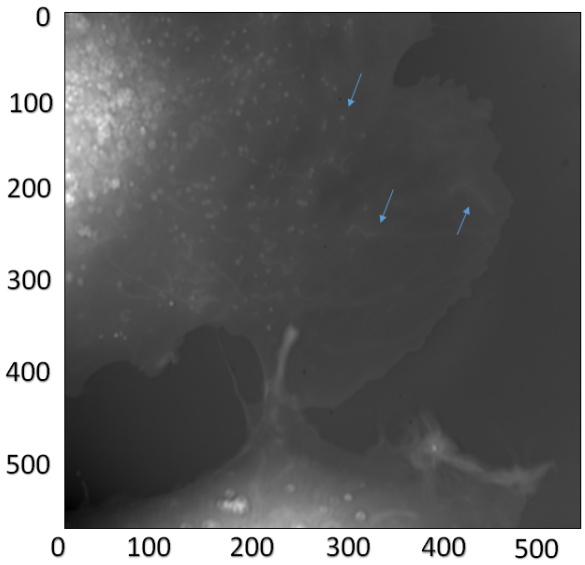


Figure 2.15: Example of an instant of a time lapse image.

As a first objective we may use the topological features that we compute on biomedical images as the image that represents a prostate gland in figure 2.13. We will segment this gland into it different components using these topological features computed on squared windows in order to distinguish the cells, the stroma, etc.

Secondly, we have images that contain objects as shown in figure 2.14(a) or cells in the biomedical case as in figure 2.14(b). We will prove that our methods relying on algebraic topology are not affected with the backgound/foreground discrimination as in the case of image of figure 2.14(a) where the intensity of dots is very close to the non homogeneous background and they are very different in size and form. Also our method is able to segment the cells in figure 2.14(b) , as marked in red color in the figure, and/or find its components, as with the blue color, without the use of prior parameters, while the most existing methods can't execute these two tasks together, or need the existence of these parameters. We will prove the efficacy of this method on 2D as well as on 3D images like the biomedical image of nucleolus in figure 3.36 on page 90 using topological constructions on pixels and superpixels which will give the method a generic flavor.

Moreover, detection of moving object from a sequence of frames captured from a static camera is one of the missions of applying the algebraic topology tools in this thesis. The goal in this case is to track the movement of cells and vesicles through time from the first to the last frame of the image sequence.

The figure 2.15 is a biomedical image taken by a time-lapse technique using the SID4Bio quantitative phase imaging system introduced in [BMW09]. This image of this sequence, that we represent by one instant, is of size 500×500 and contains plenty of biomedical objects like vesicles, trains of vesicles and cells. We will track the movement of these objects during the image sequence like the image in 3.40 on page 96 that shows the

movement of a vesicle and a train of vesicles.

2.5 Conclusion

In this chapter, we intended to make an overview on some of the image processing techniques. Then we discussed some general topological notions and how topology is applied in scientific world and image processing. In the next section we plunged in the world of algebraic topology, we discussed some of its theories, then we explained in details the homology theory in its absolute and relative form and how to compute homology groups. This was followed by a state of art applications of algebraic topology in scientific problems. Finally we addressed the requirements of this thesis on the type of images that we have and how the topological methodologies can be applied in order to perform image processing tasks.

On the other hand, we have seen that homology is an effective tool in capturing topological characteristics of static spaces. But what if these characteristics are not of major importance for deducing the inferences in images? And what if we wanted to increase the topological space where the homology is computed? This is why we present in the next chapter, the persistent homology, a way to detect resistance of homology classes in face of variations of topological spaces. And we see how this tool is suitable for achieving the tasks and requirements needed in section 2.4.

Chapter 3

Persistent homology and applications to images

3.1 Persistent homology 46
3.1.1 Spatialization 47
3.1.2 Filtration and persistence 49
3.1.3 Computing persistent homology 53
3.2 Applications of persistent homology 58
3.3 Filtration defined on images 59
3.3.1 Combinatorial representation of pixels' cell complex 60
3.3.2 Construction of the filtration of pixels' cell complex 60
3.3.3 Combinatorial representation of superpixels' cell complex . . . 63
3.3.4 Construction of the filtration of superpixels' cell complex . . . 64
3.3.5 How to deal with nested homology classes in 2D images 64
3.3.6 How to deal with nested homology classes in 3D images 65
3.4 Image segmentation using lifespans of homology classes . . . 66
3.4.1 Description of the method 68
3.4.2 Real applications 69
3.5 2D and 3D object segmentation using homology classes . . . 72
3.5.1 Synthetic and real images 74
3.5.2 Otsu thresholding 76
3.5.3 MSER technique 77
3.5.4 Proposed method 79
3.5.5 Object segmentation based on superpixels 85
3.6 Object tracking using relative persistent homology 91

3.6.1 Description of the method 91
3.6.2 Synthetic image 93
3.6.3 Real applications and comparisons 94
3.7 Conclusion 98

In chapter 2 we developed some notions used in topology and we introduced some theories issued from algebraic topology, then we focused on homology theory giving some state of the art applications of this theory in engineering problems and specially in image processing domain.

In this chapter, we extend the homology theory to a more suitable phase for data understanding and inference conclusions in image analysis. This phase consists about computing variations of homology during modifications of topological spaces by a procedure called filtration. We talk about persistent homology where the goal is to detect homology classes that persist during variations within the topological spaces.

The importance of this procedure relies on the concept that topological features detected over a range of varying scales are more suitable to represent correct features of the studied data instead of detecting noise or using a particular choice of parameters. Applications of persistent homology depend highly on the construction of the cell complexes. We show in this chapter different applications of homology theory in image processing like image segmentation, object segmentation and detection and object tracking.

We develop the idea and concepts of homological persistence in section 3.1 by extending and explaining in details the workflow of computing persistent homology. Then we describe in 3.2 some state of the art applications of persistent homology in engineering and image applications. In section 3.3, we propose a combinatorial representation of pixels and superpixels into cell complexes suggesting a filtration scheme associated to these complexes. We first execute the use of this combinatorial representation in section 3.4 where we rely on the workflow of the computation of lifespans of homology classes on pixels in order to achieve image segmentation using a combination of topological and statistical features. In section 3.5, we explain how to apply persistence concepts on pixels to perform object segmentation then we extend this application to superpixels for 2D and 3D object segmentation. In section 3.6, we propose a method relying on relative homology in its persistence form to develop an object tracking and detection technique during variation of time. The relative homology adds to the absolute homology the capability to detect cycles in more desired spaces used as quotient spaces for the cell complex. We finish this chapter by a conclusion that draws a summary for this chapter.

3.1 Persistent homology

Among the algebraic topology tools, homology represents a way for talking in an unambiguous manner about how a space is connected. It associates algebraic objects such as

Figure 3.1: Workflow of computing persistent homology.

abelian groups to topological spaces like cell complexes built on the set of pixels. Pixels' values of an image arise like a natural function to develop this tool on. Alternatively, it can detect the connected components, tunnels, voids etc., which represent image clusters or features of any dimension.

Furthermore, persistence is the state of occurring or existing for a long time despite the changes in the studied space. Thus, it measures the endurance of an image feature and its importance through the variation of a scale parameter built on pixel intensity for example. Persistent homology represents one of the most powerful algebraic yet computationally feasible tools in measuring topological features of functions. It is an algebraic invariant that captures topological features at varying spatial resolutions. More precisely, persistent homology probes topological properties of a space from a set of sampled points like pixels. It can track the birth or appearance and death or disappearance of a topological feature despite the changes in the nested space constructed by an operation called filtration in which a parameter scale, the pixels intensity in our case, is increased to detect the changes in the studied space.

The procedure of persistent homology computation follows a workflow illustrated in figure 3.1 that consists in two big steps. The combinatorial representation permits to convert the image to a topological space, the cell complex. Then the filtration scheme construct a nested sequence of cell complexes where persistent homology is computed at the end to provide to persistence its features.

We have explained in section 2.3.3 how to represent a set of points into a combinatorial form in order to be able to be manipulated by algorithms. After this transformation, the procedure of computing persistent homology begins by a spatialization of the given space in order to construct a filtration scheme of the induced topological spaces that will contribute to compute the persistent homology by algorithms available by linear algebra.

3.1.1 Spatialization

The input data is manipulated as a measure function. The spatialization of the data and its combinatorial representation permit to construct a topological space called cell complex as we have seen in chapter 2. Boundary maps between the cell complexes allow the construction of chain complexes that ensure the computation of homology groups.

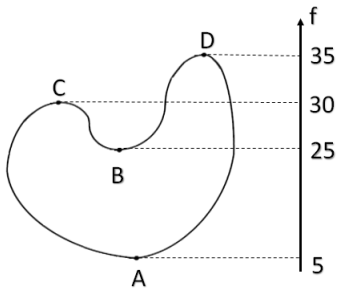


Figure 3.2: A Morse function with four critical points.

Each of these steps is described in extent below and the reader can refer to [Hat01, EH10] for more explanations on the following descriptions.

The input data is viewed as a continuous function from a domain $D \subset \mathbb{R}^n$ into the space of real numbers $\mathbb{R}$ i.e. $f : D \to \mathbb{R}$. This point of view is correct for 2D and 3D grayscale images since the domain D is a subset of $\mathbb{R}^2$ or $\mathbb{R}^3$. The spatialization allows to use the notions of surface and neighborhood using the associated function.

The topology of a space is related to the sublevel sets that consist of all points of D whose value does not exceed a level a: $U_a = \{x \in D / f(x) \leq a\}$.

Taking into account the sublevel sets for increasing threshold values, Morse theory highlights that changes in topological features can only appear at so-called critical points for well behaved functions [Mil63]. For example, in figure 3.2, the function f has four critical points from A to D. These critical points correspond to changes in the structure of homology groups. Persistent homology ensures this connection between Morse theory and homology.

The sublevel sets could be ordered by their level a under inclusion, that is, $U_a \subset U_b$ when $a < b$. Under mild hypothesis, the topology of the sublevel sets change in the neighborhood of a critical point. In other words, the topology of the sub-level sets evolves only when crossing a critical point. We can define the filtration as the nested sequence of spaces

$$\emptyset \subset U_a \subset U_b \subset \ldots \subset U_z \subset D. \tag{3.1}$$

The topological changes observed between the levels of the filtration, such as holes arrival/extinction or components arrival/extinction, can be computed thanks to the topological tools described below.

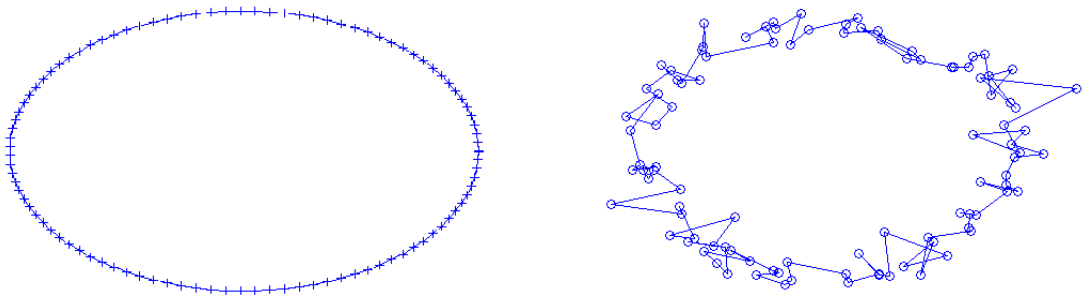


Figure 3.3: A non noisy hole versus a noisy one.

3.1.2 Filtration and persistence

Before diving into persistent homology, we will try to briefly motivate the concept. In subsections 2.3.1 and 2.3.4.1, we introduced some methods for constructing cell complexes from point data and computing its corresponding homology groups. We can compute the homology $H_p(U_x)$ for all sublevel sets U_x of (3.1) in order to depict the evolution of the number of topological features of an input data. However, we lose the information concerning the evolution of each particular cycle. Detecting the evolution of these topological features along the procedure of filtration described in (3.1) can give more important details like lifespans of homology classes and, the most significant, the independence from intensity values. Indeed, the topological features of a complex may be due to noise and computing the homology of a single sub-level set is not rich enough to describe the significant topological feature that we look for. For example, in figure 3.3, homology can detect the big hole alone and the hole with its noisy holes on its borders. We will now see how persistent homology solves this problem starting with the spatialization idea described in 3.1.1.

Given a cell complex K, let $f : K \to \mathbb{R}$ be a non decreasing function, which means that if σ is a face of τ, then $f(\sigma) \leq f(\tau)$. The level subcomplexes are then defined to be $K(a) = f^{-1}(-\infty, a]$. Denoting by a_i the values of f on cells of K in increasing order, the corresponding level subcomplexes define a filtration of K as in (3.1). Let $K^i = K(a_i)$, then we have an analogue filtration of (3.1):

$$\emptyset \subset K^0 \subset K^1 \subset \ldots \subset K^i \subset \ldots \subset K. \tag{3.2}$$

Persistence As the scale parameter increases, new cells are added until we obtain the entire complex K. For every $i \leq j$, we get an inclusion map from K^i to K^j. This continuous map is transformed to a linear one which will lead to an induced homomorphism of homology noted $f_p^{i,j} : H_p(K^i) \to H_p(K^j)$ for every dimension p of the whole complex K.

Hence, by tracking the topological evolution of the filtration using these homomorphisms,

we get a sequence of homology groups that are connected by linear maps $H_p(.)$ induced by inclusions

$$H_p(K^0) \hookrightarrow H_p(K^1) \hookrightarrow \ldots \hookrightarrow H_p(K^i) \hookrightarrow \ldots \hookrightarrow H_p(K). \tag{3.3}$$

Going from K^0 to the whole complex K through these maps will cancel the need to a specific intensity value as a parameter. These maps in (3.3) will perturb the life of homology classes through the filtration. Considering the step from $H_p(K^i)$ to $H_p(K^{i+1})$, changes can occur: new homology classes that represent holes can be created or already existed homology classes can merge or vanish signifying the fill of the hole.

Persistent homology provides tools to track the appearance or disappearance of classes in this sequence. It detects homology classes that are present through many steps in the filtration procedure and considers those non bounding cycles which remain non-bounding along these steps. Thus, it can separate noisy classes from important ones with no need to fix an intensity value.

Algebraically talking, persistence can be defined by two ways. Given a filtered complex, each complex K^i is associated with a boundary operator ∂_p^i and groups C_p^i that represents the chains of the complex, Z_p^i, B_p^i, and $H_p^i = H_p(K^i)$. Then the k-persistent $p-th$ homology group of K^i, meaning the homology groups of dimension p and that exist in $i+k$, is:

$$H_p^{i,k} = Z_p^i / B_p^{i+k} \bigcap Z_p^i. \tag{3.4}$$

This definition considers non-bounding cycles which remain non-bounding for k steps in the filtration. Moreover, this equation is well defined since both groups in the denominator are subgroups of C_p^{i+k}. The intersection of these two groups is also a group, which is a subgroup of the numerator. The rank of the subgroup $H_p^{i,k}$ is called the k-persistent p-th betti number. Intuitively, $H_p^{i,k}$ illustrates the p-dimensional holes in K^{i+k} created by the subcomplex K^i. These holes exist for all complexes K^j in the filtration with index $i \leq j < i+k$.

Also, persistence can be explained following a more concrete procedure that consists in taking $f_p^{i,j} : H_p(K^i) \to H_p(K^j)$. The k-persistent $p-th$ homology groups are the images of the homomorphisms induced by inclusion, $H_p^{i,k} = \text{Im}\, f_p^{i,k}$ for $0 \leq i \leq k$. This definition is equivalent to (3.4) since $\text{Im}\, f_p^{i,k} = Z_p^i / B_p^{i+k} \bigcap Z_p^i$. We say that a class $\alpha \in H_p(K^i)$ is born in $H_p(K^i)$ if it does not belong to $H_p^{i-1,i} = \text{Im}\ f_p^{i-1,i}$. We say that α dies at $H_p(K^j)$ if $f_p^{i,j-1}(\alpha) \notin H_p^{i-1,j-1}$ but $f_p^{i,j}(\alpha) \in H_p^{i-1,j}$. If there is a class α born in $H_p(K^i)$ that dies in $H_p(K^k)$, we record this as a persistence pair (i,j).

Algebraic structure of persistence and its visualization We have seen in the previous paragraph that persistent homology tries to relate topological features between two different complexes in a filtration. In this paragraph, we make a different view of homological persistence in order to understand its formation and its relation with persistence diagrams.

We begin to explain the persistence complex by combining the homology of all the complexes that exist in the filtration into one algebraic structure. A persistence complex K is a family of chain complexes ${K^i}_{i\geq 0}$ over a field F, together with chain maps that associate them: $f^i : K^i \to K^{i+1}$. We show below a part of the persistent complex with the expansion of chain complexes. The index that designs the filtration increases horizontally to the right under the chain maps f^i. And under the boundary operators ∂_p, the dimension decreases vertically to the bottom.

$$\begin{array}{ccccccc}
\vdots & & \vdots & & \vdots & & \\
\downarrow{\scriptstyle \partial_3} & & \downarrow{\scriptstyle \partial_3} & & \downarrow{\scriptstyle \partial_3} & & \\
K_2^0 & \xrightarrow{f^0} & K_2^1 & \xrightarrow{f^1} & K_2^2 & \xrightarrow{f^2} & \dots \\
\downarrow{\scriptstyle \partial_2} & & \downarrow{\scriptstyle \partial_2} & & \downarrow{\scriptstyle \partial_2} & & \\
K_1^0 & \xrightarrow{f^0} & K_1^1 & \xrightarrow{f^1} & K_1^2 & \xrightarrow{f^2} & \dots \\
\downarrow{\scriptstyle \partial_1} & & \downarrow{\scriptstyle \partial_1} & & \downarrow{\scriptstyle \partial_1} & & \\
K_0^0 & \xrightarrow{f^0} & K_0^1 & \xrightarrow{f^1} & K_0^2 & \xrightarrow{f^2} & \dots
\end{array} \tag{3.5}$$

Now, we form a relation that reveals a simple characterization over fields. Most important, we highlight that the standard homology of a graded module over a polynomial ring represents the persistent homology of a filtered complex.

The p-th **persistence module** $\mathscr{H}_p$, where p designs the dimension, is the family of p-th homology F-modules $H_p^i = H_p(K^i)$ together with module homomorphisms $f_p^{i,i+1} : H_p(K^i) \to H_p(K^{i+1})$.

For example, the homology of the persistent complex defined above $\{K^i, f^i\}$ can also be seen as a persistent module where f^i maps a homology class to the one that contains it.

This connection between the persistent complex and persistent module gives birth to the p-th persistence module. As a **correspondence** this persistence module can be accorded the structure of a graded module over the polynomial ring $F[x]$:

$$\mathscr{H}_p = \bigoplus_{i=0}^{\infty} H_p^i, \tag{3.6}$$

meaning that $\mathscr{H}_p$ is the direct sum of H_p^i, i.e. $\mathscr{H}_p$ is generated by groups of H_p^i. The structure theorem over the principle ideal domain [Hun80] (PID) $F[x]$ reflects the decomposition of the p-th persistence module into homology generators:

$$\mathscr{H}_p \cong \bigoplus_{i}^{n} x^{t_i}.F[x] \bigoplus (\bigoplus_{j}^{m} x^{r_j}.(F[x]/x^{s_j}.F[x])). \tag{3.7}$$

The first part of the direct sum represents the homology classes that appear at t_i and persist forever. And the second part represent homology generators that appear at r_j and persist until $r_j + s_j$.

This decomposition represents a complete discrete invariant [CZC04]. It gives n half-infinite intervals $[t_i, \infty)$ and m finite intervals $[r_j, r_j + s_j)$. The visualization of the algebraic structure of the persistence will be done via a barcode [CZC04] or a persistent diagram [Zom10a]. Thus, the representation of persistent homology of a filtered cell complex will be done by these two invariants.

The barcode represents the persistent homology classes with an horizontal line that begins at the first filtration level when the class appears (birth time) and ends at the filtration level of its disappearance (death time). While a persistent diagram marks a point for each homology class with its abscissa, that represents the birth time and its ordinate for the death time.

We can summarize the visualization by a persistent diagram or a barcode by the following:

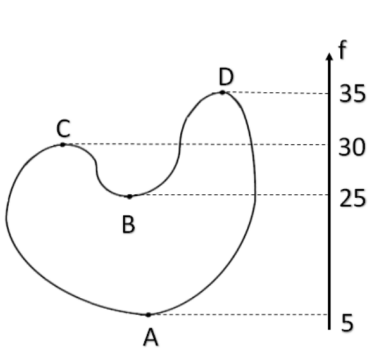
- Each occurrence of a component, hole, and void which represent 0, 1 and 2 - dimensional homology classes respectively is represented by a bar in the barcode or a point in the persistent diagram.

- The starting point of the bar and the abscissa of the point in a persistent diagram correspond to the value of the level of filtration at which the homology class appear, the birth time. The ending point of the bar and the ordinate of the point in the persistent diagram corresponds to the value of the level line at which the homology class disappear, the death time.

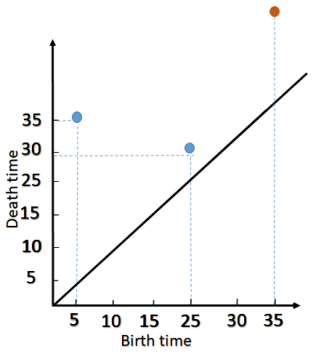
- The position and length of the bar and the difference between the abscissa and the ordinate of a point in persistent diagram represent the lifespan of the corresponding component, hole and void.

For example, the life duration of homology classes of dimension zero derived from the Morse function in figure 3.4 (a) that is born at level 5 on the critical point A will die at level 35 on D. The cycle born at level 25 on B will die at C at level 30. Also a cycle of first dimension is born at D at level 35 and never dies, for visualization purposes we refereed to ∞ as its death level. The cycles of dimension zero that may be homology classes have 30 and 5 as lifespans respectively. All these cycles are represented on the diagram and the barcode on the figures 3.4(b) and (c).

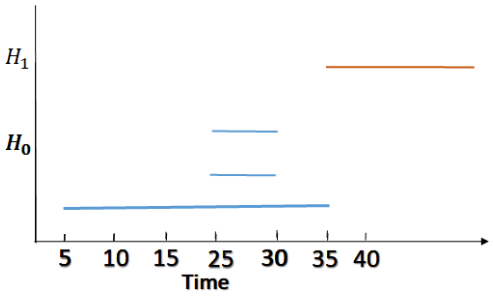
Following this manner, a class having a long service life will be topologically more informative whereas a class of short life will be due to noise. This motto is supported by the stability of the lifespans of classes under continuous deformations [CSEH07] like stretching, translation etc.



(a) A morse function.



(b) Its persistent diagram for zero and one dimension.



(c) Its barcode for zero and one dimension.

Figure 3.4: A Morse function and its associated persistent diagram and barcode.

3.1.3 Computing persistent homology

At each step of the filtration scheme, it turns out from the algebraic information above that only two kinds of changes could happen. Adding a new cell σ' at the level j and dimension p will disrupt the topology of the complex. A new homology class can thus be created and we call σ' a creator. An existing class σ of level i and dimension $p-1$ can also be destroyed, that is, it can be made homologous to zero, and in this case we call σ' a destroyer. Each negative cell σ' is associated to a unique positive cell σ with levels $i < j$, which corresponds to a persistence pair.

This allows us to define the persistence of a topological feature as the difference (e.g. in function value or ordering index) between its death and its birth in the construction, and also to organize the critical points in pairs of creators and destroyers of topological features. The lifespan of the homology class will be equal to the level difference between destroyer and creator, $j - i$. The long life homology class, that is associated with a creator/destroyer pair, is a major point of interest for us as we will see its abilities in image processing applications specially in identification of interesting objects in the image, while the short ones will represent the noise in the image.

On the level of computation, persistent homology can be computed very efficiently. Sim-

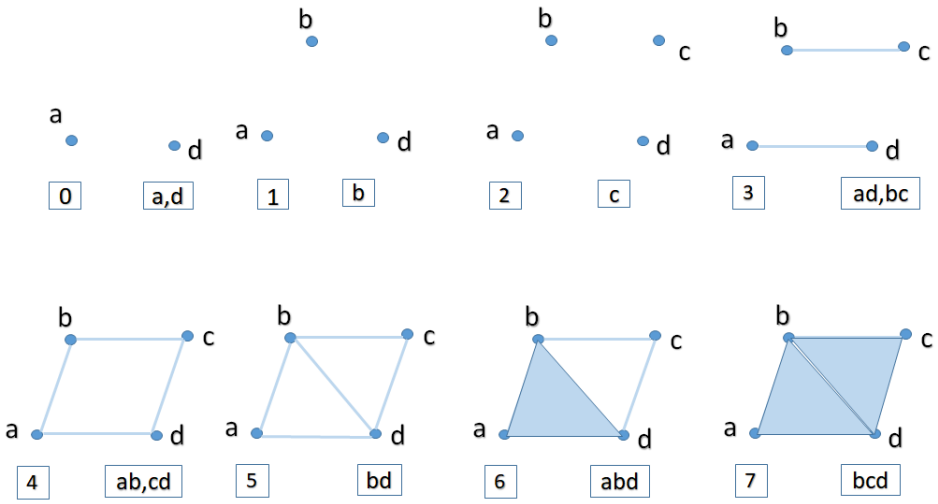


Figure 3.5: Simple filtration example.

ple matrix elimination procedure is able to compute homology classes with chains in field $\mathbb{Z}_2$ and their lifespans by a given cell ordering of the cell complex. We propose a combination between the algorithms described in [Zom10a, CC11, dSMVJ11a], to compute the persistent homology operation and to get the basis of homology classes and their lifespans.

To detail the methodology that we have proposed, we use a compatible ordering of the cells, in a manner that the sequence of cells $\sigma_1, \sigma_2, \ldots$ is ordered by dimension of cells then by their function value. If σ_i is a face of σ_j then it's necessary having a value less or equal to σ_j. Such an ordering exists because the function that asserts the values to f is monotonic. In this way, every cell complex of the filtration will be a subcomplex of another complex with higher level in the filtration.

This sequence is used for setting up the boundary matrix ∂ of the filtration that stores all cells of all dimensions in one matrix and so an entry (i, j) is 1 whenever σ_i is a face of σ_j, i.e. $\sigma_i \in \partial\sigma_j$ and 1 otherwise.

We clarify the notions with a simple example. Let K consists of the final complex in the figure 3.5 at level 7. In this figure we build a filtration showing in left boxes the level or the time of the filtration, and in right box the cells that are added at this level. Noting that this level of filtration is decided by a function that assign values to the cells. In this way an edge will not appear before its vertices, and a 2-cell before its edges and thus we're respecting the subcomplexes inclusion condition in filtration. For the boundary matrix that represents the filtration we order the cells, first by their dimensions, then by their values or level in the filtration then by alphabetic order as shown in equation (3.8). Noting that the boundary of a point is null, so we put a 1 in the entry in the column of the vertex and in row 0.

The algorithm that manipulates the boundary matrix uses column operations to reduce it to another matrix D. Let $lowest(j)$ be the row index of the lowest one in column j. We don't define $lowest(j)$ if the entire column is zero. D is considered as reduced if $lowest(j) \neq lowest(i)$ whenever $j \neq i$ specify two non-zero columns. And so the algorithm reduces ∂ by adding columns from left to right. We store in a matrix $D^{'}$ the column additions that we made to ∂ to be reduced to D.

After applying the algorithm, we remark the case in which column j ends up to have all zero numbers from the other case in which it has a lowest one.

- In the first case when the column j of D is zero, we call σ_j the cell that represents a positive cell because its addition during the filtration scheme has caused the birth to a new homology class.

- In the second case, where the column j of D is a non zero it definitely keeps the boundary of the chain that grew and accumulated in column j of matrix $D^{'}$ and thus is a cycle. In this case, σ_j is called a negative cell because its penetration in the filtration indicates the death of a homology class. This accumulated cycle is born at i, the time of the cell of the lowest one in column j, with $i = lowest(j)$.

$$\boldsymbol{\partial} = \begin{array}{c} \\ (0) \\ (a) \\ (d) \\ (b) \\ (c) \\ (ad) \\ (bc) \\ (ab) \\ (cd) \\ (bd) \\ (abd) \\ (bcd) \end{array} \begin{array}{c} \begin{array}{ccccccccccc} (a) & (d) & (b) & (c) & (ad) & (bc) & (ab) & (cd) & (bd) & (abd) & (bcd) \end{array} \\ \left[\begin{array}{ccccccccccc} 1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot & 1 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array}\right] \end{array} \tag{3.8}$$

From this construction it is understood that the lowest ones in the matrix D corresponds to points in the diagram that represents the negative-positive pairs and called the persistent diagram. This persistent diagram registers the level of births and deaths of homology classes. Literally, if we have $i = lowest(j)$, then (x, y) is a point in the persistent diagram, where x and y designs the level of the filtration when σ_i and σ_j are added respectively.

Not to forget that naturally homology classes of long lifespans will indicate the presence of interesting topological features in the data, since they resisted to the topological changes

in the space, while those with small lifespans are considered as topological noise.

$$
\begin{array}{l}
D = \partial D^{'} \Leftrightarrow \\
\end{array}
\begin{array}{c}
\\
(0) \\
(a) \\
(d) \\
(b) \\
(c) \\
(ad) \\
(bc) \\
(ab) \\
(cd) \\
(bd) \\
(abd) \\
(bcd)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
(a) & (d) & (b) & (c) & (ad) & (bc) & (ab) & (cd) & (bd) & (abd) & (bcd)
\end{array} \\
\left[\begin{array}{cccccccccccc}
1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot
\end{array}\right]
\end{array}
\tag{3.9}
$$

$$
= \boldsymbol{\partial}
\begin{array}{c}
\\
(0) \\
(a) \\
(d) \\
(b) \\
(c) \\
(ad) \\
(bc) \\
(ab) \\
(cd) \\
(bd) \\
(abd) \\
(bcd)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
(a) & (d) & (b) & (c) & (ad) & (bc) & (ab) & (cd) & (bd) & (abd) & (bcd)
\end{array} \\
\left[\begin{array}{cccccccccccc}
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
1 & 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & 1 & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1
\end{array}\right]
\end{array}
$$

Back to the example in the figure 3.5 and the boundary matrix shown in (3.8). We reduce this matrix to D and get the decomposition as in 3.9. In matrix notation, the algorithm computes the reduced matrix as $D = \partial D^{'}$, where $D^{'}$ is an invertible upper-triangular matrix with $\mathbb{Z}_2$ coefficients that stores the operations made on ∂ to get reduced. For example, at column (d) of $D^{'}$ we look for operations made on column (d) of ∂. These operations are adding column (a) to column (d) and thus we register 1 in the rows (a) and (d) of column (d) of matrix $D^{'}$ and so on for other columns of $D^{'}$.

As we can see in D, the first lowest one is in row 0 and column (a). This row corresponds

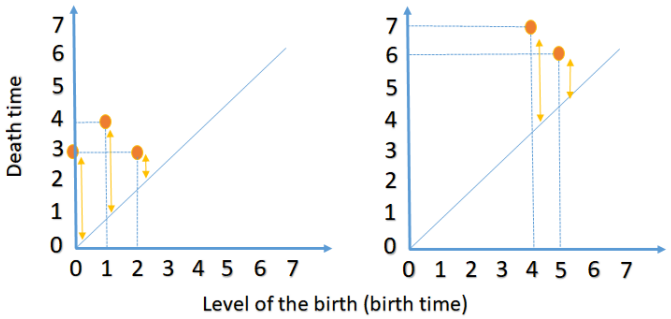


Figure 3.6: Persistence diagrams of 0 and 1 dimensions.

to a (-1)-dimensional homology class that dies when the vertex (a) is added. The second lowest one is in row (d) and column (ad). It means that the vertex (d) gives birth to the 0-cycle that the edge (ad) kills. The column (bc) has a lowest one in row (c) and thus the vertex (c) gives birth to the 0-cycle that the edge (bc) kills. Also the column (ab) has a lowest one in row (b) and thus the vertex (b) gives birth to the 0-cycle that the edge (ab) kills.

Adding the edges (cd) and (bd) don't kill anything, which is obvious in the matrix D since their columns are zeros. In fact, adding the edge (cd) corresponds to a 1-cycle obtained by adding the columns (cd), (bc), (ab) and (ad) as showed in column (cd) of D'. Thus, the edge (cd) gives birth to a 1-cycle formed by edges (cd), (bc), (ab) and (ad) of added columns which is killed after by the triangle (bcd). Similarly for the 1-cycle is created by the edge (bd) and killed by the triangle (abd).

With this example, we showed that we can recover homology classes and their lifespans from the reduced boundary of the filtration. In addition, considering the function value of cells as their indices, the figure 3.6 illustrates the persistence diagrams of dimensions 0 and 1 with arrows indicating their lifespans, and in analogue manner the lifespans are represented in a barcode as in figure 3.7. For example, the 0-cycle that is born by vertex (d) at level 0 is killed at level 3 by the edge (ad) when it is added to the filtration. Similarly for the 0-cycle that is born by vertex (b) at level 1 is killed at level 4 by edge (ab), and the 0-cycle that is born by vertex (c) at level 2 is killed by edge (bc) at level 3. For this we have the points $(0, 3)$, $(1, 4)$ and $(2, 3)$ in the persistent diagram of 0 dimension that represent these cycles in figure 3.6 (a) and the bars in the barcode that begin with birth time of the cycle and ends with its death time in figure 3.7 .

In addition, the 1-cycle born by the edge (cd) at level 4 is killed when the triangle (bcd) is added at level 7 and the 1-cycle born by edge (bd) at level 5 is killed when the triangle (abd) is added at level 6. For this we have the points $(4, 7)$ and $(5, 6)$ in the persistent diagram of first dimension in figure 3.6 (a) and the associated barcode in figure 3.7.

Other algorithms that compute persistent homology profit from the duality between homology groups and cohomology [dSMVJ11a], or reduce the initial complex before the

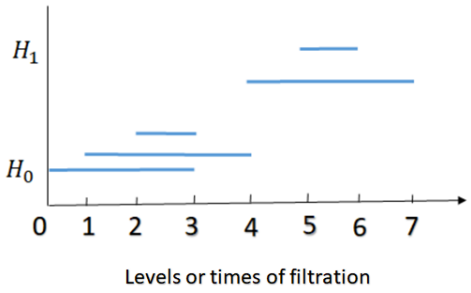


Figure 3.7: Barcode of homology classes of 0 and 1 dimensions.

effective calculation [MW10, WCV12]. Also other algorithms rely on discrete Morse theory like [MN13] or on the possibility to distribute the algorithm for more speed [BKR14]. The illustrated algorithm above remains the most basic and the simplest one and can be associated with many other variations in the algorithm like the twist in [CC11] or with combinations of algorithms.

3.2 Applications of persistent homology

Variety of applications include the need of topological persistence for deriving inferences from the studied data. For example in [VRT16], the authors propose a novel structure for dynamical analysis of human actions from 3D motion capture data employing topological persistence. An algorithm for topological clustering based on relative persistent homology is introduced in [PGK16] to cluster trajectories with varying end-points based on the same simplicial complexes used for the classification of trajectories with fixed start and end points. In [ESM15], the authors propose a new framework for extracting the characteristic points of the peripheral pressure wave based on persistent homology.

The idea of topological persistence has been also used in recent years in computer vision to perform image processing tasks. Persistent homology returns and numerical results may be used on images to accomplish classification of studied data. In [ARC14], the authors represent a methodology of classifying hepatic lesions using persistent homology, the bottleneck distance, and a support vector machine. Also, a combination of image processing, geometry topology and machine learning are used in [DEL+16] to classify patterns of stomach images. In this work, the pixels are binarized depending on a threshold in order to compute persistent homology. A novel tumor detecting tool is presented in [QSN+16], using the novel idea of persistent homology profiles computed on binarized pixels after selection of patches manipulated by convolutional neural networks. In these works, the lifespans of homology classes form the tool of the classification step.

On the level of image segmentation, a split-and-merge algorithm is studied in [LF07]. Edge detection is first performed using a wavelet-based detector. Then the image is split into regions using persistent homology. Finally, regions with similar topological features

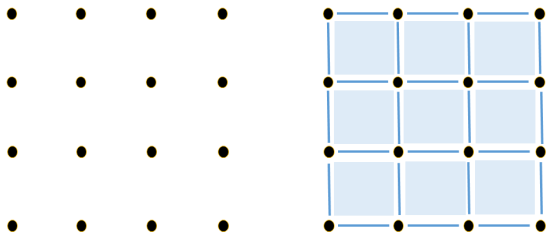


Figure 3.8: Set of pixels and their cubical complex.

are selected and merged in order of topological persistence. Noting that the accurate split and merge operations are controlled by two parameters that must be right chosen to perform the segmentation aimed for.

An algorithm that perform hierarchical segmentation of images using the mean shift method and topological persistence is discussed in [PD07]. After grouping pixels whose feature points are in the same mode of a density function, the authors merge pairs of clusters depending on the persistence of the boundary between the two modes. This is equal to the simplification of a Morse Smale complex and the hierarchy of merging is built depending on a positive threshold. The creation of this hierarchy must be controlled by modification of density function and persistence. As in [LF07], this work doesn't intend to build a topological complex on the pixels and they consider the density function as a height function of the Morse Smale complex.

3D segmentation of point clouds is discussed in [BP16]. First, the authors rely their segmentation method on a filter representation of 2D images removing all outlying points and those that belong to planar models. Then they downsample the remaining point cloud using a voxel grid. Using a Vietoris Rips complex, the zeroth homology groups are computed to find clusters of points and eventually perform segmentation of connected components.

After briefly analyzing the state of art of the use of persistent homology in image applications, we present now the way that we developed in construction of the filtration scheme and the cell complexes that we built on pixels and superpixels.

3.3 Filtration defined on images

In this section, we will show that the methodology issued from homology theory and persistence and described extensively in the section 3.1 can be successfully applied on grayscale images of different dimensions in order to perform image processing tasks.

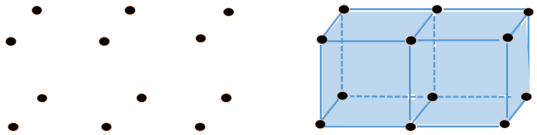


Figure 3.9: Set of voxels and their cubical complex.

3.3.1 Combinatorial representation of pixels' cell complex

In order to find the long lifespan homology classes in images, we must follow the pipeline of the computation of persistent homology by building the desired cell complex and associating the filtration to obtain the nested sequence of complexes. Before explaining the way we assign cells in the cubical complex, we define the neighborhood relation N_4. If $x = (x_1, x_2) \in \mathbb{Z}^2$, $N_4(x) = \{y = (y_1, y_2) \in \mathbb{Z}^2, |y_1 - x_1| + |y_2 - x_2| \leq 1\}$.

An image is a set of points named pixels arranged on a rectangular grid of 2D or 3D matrix. Thus, the cubical complex [KMM03] built from the image depends on its pixels and it is constructed as follow:

1. Beginning the construction by vertices or 0-cells. Each pixel in the image will be considered as a vertex,
2. Then, we build the 1-cells or edges from each two adjacent pixels taking into consideration the four neighbors of a pixel that satisfy the N_4 neighborhood relation.
3. We then adjoin squares as 2-cells where we have 4 adjacent pixels,
4. In 3D images, we form the 3-cells or cubes from the corresponding 6 adjacent squares.

At the end, the cubical complex K will be formed by squares and their components, edges and vertices, in the 2D case as illustrated in figure 3.8, or by cubes and their parts, squares, edges and vertices in 3D images as in figure 3.9.

For large images, a method consisting in reducing the initial cell complex can be applied. It consists in computing superpixels by a technique such as SLIC technique introduced in [ASL+12], as explained later the cell complex being constructed on the resulting superpixels which are not this time a rectangular grid.

3.3.2 Construction of the filtration of pixels' cell complex

In order to compute persistent homology of this complex, we must determine a filtration on the cell complex K built on the image. We begin by assigning a value to each vertex in K. The pixels' intensities arise like a natural function to develop this strategy on. In this manner, vertices of K will carry the pixels' intensities as values.

Naturally, this can be represented by a function $f : D \to \mathbb{R}$ where $D \subset \mathbb{R}^2$ and f

represents typically the pixels' intensities as described in 3.1. This function will order the cells of K by their increasing values.

For our filtration structure, the value of a p-cell will be the maximum of values of its $(p-1)$-cells that represent its boundary. Indeed, if τ is a k-cell then $f(\tau) = \max_{\sigma \in \partial_p(\tau)} f(\sigma)$. Following this procedure, the vertices hold the value of its grayscale intensity, the value of an edge is the maximum of surrounding pixels, a square's value is the maximum of its enclosing edges, and the cubes will have the highest value of its 6 squares as shown in figures 3.10 and 3.11. For example in 3.10, the first 4 pixels in top left of the images and encircled by a dashed circle have the grayscale values 4, 1, 5, 9. The edge, for example between pixel 4 and pixel 1 has the value 4 because it holds the maximum of its vertices and the square or the 2-cell holds the value 9 as it's the maximum of its edges. Of course,

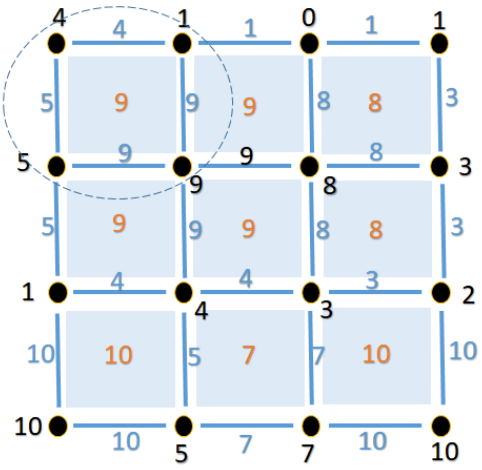


Figure 3.10: Value assigning in 2D case.

other possibilities for the function that control the filtration scheme can be taken into consideration, like means of vertices values for the value of edges or medians of edges values for squares but they must take into account the subcomplex relation in filtration scheme. Taking the maximum values of vertices for an edge or of edges for a 2-d cell will ensure that an edge won't appear before its vertex nor a 2-d cell before one of its edges.

Inspired by the filtration in equation (3.2), we represent the grayscale values of an image by i where $i_{\min} \leq i \leq i_{\max}$, considering that $i_{\min}$ and $i_{\max}$ are the minimum and maximum values of pixels' intensities of the image.

A cell complex K_i include all the cells σ with values less or equal to i,

$$K_i = \{\sigma \mid f(\sigma) \leq i\}. \tag{3.10}$$

Noting that all kinds of filtration must take into account the fact that K_i is a subcomplex of K_j, i.e. $K_i \subset K_j$ whenever $i < j$. In our application, we will have an intensity filtration

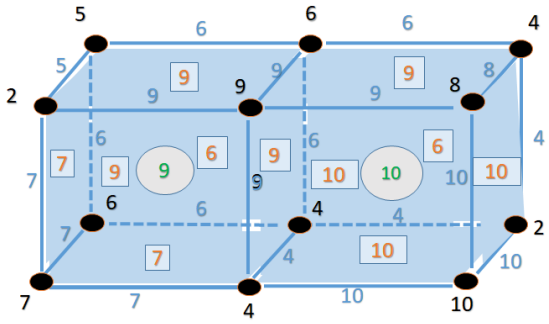


Figure 3.11: Value assigning in 3D case.

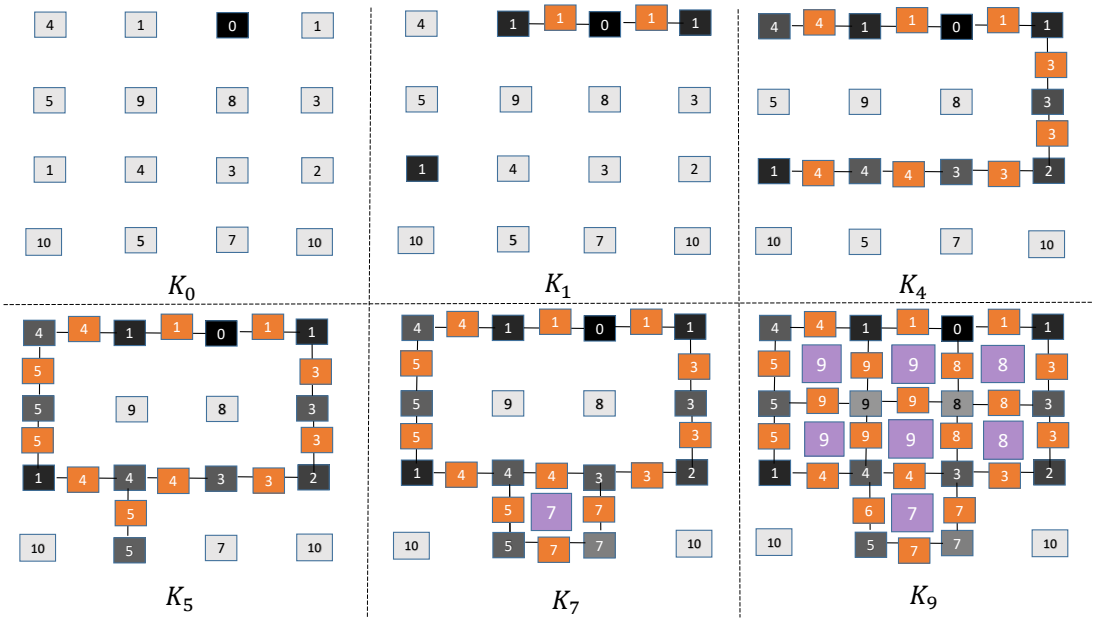


Figure 3.12: Pixels' complex subfiltration for the 2D case presented in figure 3.10.

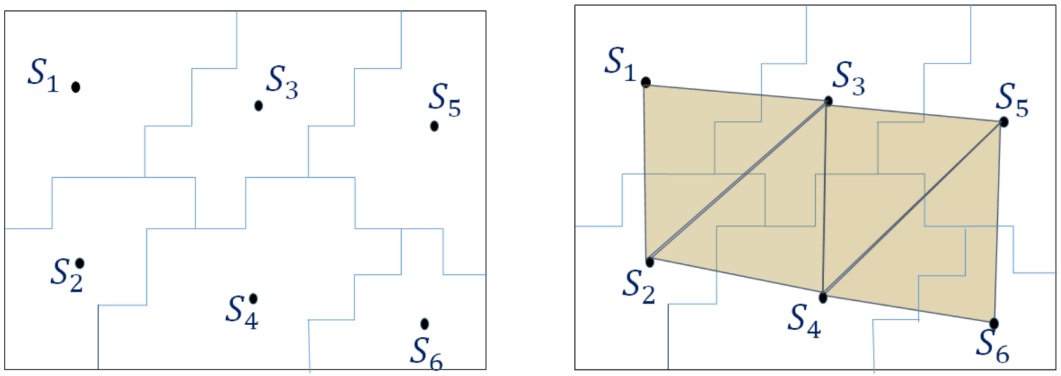
built on the studied image and we will call it intensity subfiltration since it depends on sublevels of pixels' intensity:

$$\emptyset \subset K_{i_{\min}} \subset K_{i_{\min}+1} \subset \ldots \subset K_i \subset \ldots \subset K_{i_{\max}} = K. \tag{3.11}$$

Reversing the inequality in equation (3.10) will allow us to use another filtration in our complex and we will refer to it as intensity superfiltration:

$$\emptyset \subset K_{i_{\max}} \subset K_{i_{\max}-1} \subset \ldots \subset K_i \subset \ldots K_{i_{\min}} = K. \tag{3.12}$$

An example of the intensity subfiltration the image shown in 3.10 on which we explained the cell complex construction is displayed in figure 3.12. The dark and colored cells represent the cells that enter the pixels' complex subfiltration at its corresponding level while the transparent one are not yet in the complex. Note that superfiltration can be



(a) A set of superpixels with their centers S_i.

(b) The cell complex created from these superpixels.

Figure 3.13: Simple example of superpixels combinatorial representation.

constructed in a similar way as in figure 3.12 but beginning from the greatest value in a decreasing way.

We notice that in this filtration, in K_5 a cycle is born, indicating the presence of a whole. This cycle will die at K_9 when it's fully covered by 2-cells. Thus, the lifespan of this cycle is $9 - 5 = 4$.

During the computation of persistent homology of an image using the methods and algorithms described in section 3.1, we will obtain as result p-cycles that represent the p-persistent homology classes. These p-cycles that are p-classes and their corresponding lifespans will be of major interest for our image processing tools in segmentation and tracking.

3.3.3 Combinatorial representation of superpixels' cell complex

As we said before, beside its usual pixel-grid form, a large image can be reduced to a set of superpixels. Superpixels are regions or clusters that group "similar" pixels into meaningful atomic regions that can be used to replace the rigid structure of the pixel grid. They capture similarities mainly by jointly considering color and spatial proximity and thus try to provide a concise and semantic image representation [JCH16] by grouping pixels into meaningful segments. Thanks to its various advantages, the superpixels segmentation is an increasingly popular image preprocessing technique used in many computer vision applications.

There are two primary reasons for the performance gains of superpixels representation. The first is that superpixels reduce the computational primitives significantly without obvious information loss. The second is that they can appropriately respects the boundary of different objects in the image, preserving the region based consistency of pixels.

Superpixels partition the image into segments that group the "similar" pixels. First, we

assign a vertex to each superpixel, we will refer to it as (S_i). We will have a set of points that we begin our construction with. In order to add higher dimension cells to the cell complex K that we want to construct to compute persistent homology, we follow this procedure:

We pivot over the 8-connected pixels of each pixel, in each superpixel (S_i). The choice of the 8-connected pixels increase the number of one cells in the complex and thus the number of candidate homology classes. If at least one of these neighbors belong to other superpixel (S_j), we add an edge (S_iS_j) to the complex. Similarly, if two of the neighbors belong to two other superpixels (S_j) and (S_k), we add a 2-cell $(S_iS_jS_k)$, that represents a triangle as shown in figure 3.13.

In the three dimension case, the superpixels will swing over the depth of the image as each superpixel will belong to different depths. We pivot over the 26-connected pixels of each pixel, we add the 1-cells and the 2-cells in the same way as in the 2D dimensions case. Additionally, a 3-cell is added to the complex when three of the 26 neighbors belong to other superpixels to three other superpixels (S_j), (S_k) and (S_l), we add a 3-cell $(S_iS_jS_kS_l)$, that represents a tetrahedron.

3.3.4 Construction of the filtration of superpixels' cell complex

In order to construct a nested sequence of subcomplexes, we begin by assigning a value to each vertex S_i in the complex K. The values of vertices can be the mean, the variance of pixels' intensity in each superpixel or other criteria. Many measures can be affected to p-cells in the procedure of filtration, as we explained before, like the mean of their $p-1$ cells, their variance or their median etc. For our application, the values of vertices will be the mean of pixels' intensities in each superpixel and the p-cells can hold the maximum values of its boundary as described previously in section 3.1 The figure 3.14 shows the 6 superpixels shown in figure 3.13 with their values. The centers of neighbors superpixels are connected with edges holding the maximum values of the centers and triangles with maximum values of edges.

We will obtain a nested sequence of subcomplex in an analogue way to equation (3.11) or equation (3.12). After the computation of persistent homology on this nested sequence of topological complexes, we can gather the interesting homology classes, elements of H_p, the main tool of our segmentation process.

3.3.5 How to deal with nested homology classes in 2D images

We have talked in section 2.3, that for the computation of homology groups there is no set of canonical representative. For that we can have an homology class encircle two holes No basis is canonical, and many basis can be valid. For this purpose, we proposed an algorithm that deals with the homology classes of first dimension in 2D images and that are inside each others in order to separate them. Noting that homology classes of first

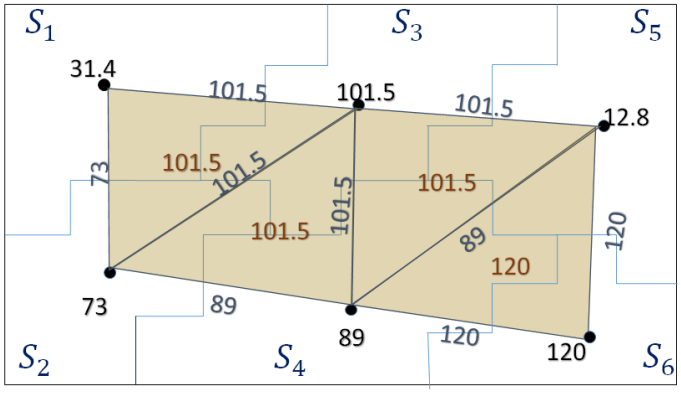


Figure 3.14: Superpixels complex value assigning.

dimension are one chains, thus they are edges that connect vertices or the pixels or centers of superpixels in our case. The steps of this algorithm are described as follows:

1. Recognize which classes have common edges, i.e. those that are intersecting,
2. From the intersecting classes, we recognize those that are inside each others using the coordinates of the vertices of edges to see if they are inside other classes. It is sufficient to see if one vertex of one of the non common edges is inside the class or not since two intersecting classes are whether completely inside or outside each others.
3. We separate the classes that are inside each others and we get a new contour $\mathcal{C}$ using $\mathcal{C} = (A \bigcup B) - (A \bigcap B)$ where A and B design two homology classes with $B \subset A$. Noting that when we say $A \bigcup B$ it means the edges that belong to homology classes A and B and not the zone that is inside it.
4. We repeat his procedure until we get contours that don't contain other classes inside them.

3.3.6 How to deal with nested homology classes in 3D images

In the 3D case, the computed basis for homology classes is not canonical so we developed a specific algorithm for the separation of second dimension homology classes. This algorithm that depends on algebraic topology techniques and tests if a point is inside a polyhedron. We are thus able to detect the including criteria of the homology classes and separate them in 3D images. These classes are 2D chains thus they are triangles since we work with superpixels that form simplical complex in the 3D case. Thus, the homology class is a polyhedron or an $\mathbb{R}^3$ surface. The steps of separating the classes are quite similar to 2D case but the way that we know here if two homology classes are inside each others is more complicated.

It's essential to know if two polyhedron, or second homology classes, are inside each others. For this purpose, we must know how to recognize if a point is inside a polyhedron or not. We use an algorithm that compute the "winding number" of a closed oriented surface S around a point I not on S. The winding number of the closed oriented surface S around a point I not on S is the number of times that the surface encloses that point.

More precisely for a polyhedron, like in our case, the winding number will be 1 for points inside the surface and 0 for points outside. According to [SZ05], the mapping of the vertices to $\mathbb{R}^3$ gives a continuous map from the simplicial complex C of S to $\mathbb{R}^3 - I$. And this in turn gives a map on homology groups of second dimensions: this winding number can be defined as $H_2(C, \mathbb{Z}) \rightarrow H_2(\mathbb{R}^3 - O, \mathbb{Z})$. The image of S by this map will give the winding number.

In summary, the steps of the proposed separation algorithm are the following:

1. Recognize which classes have common triangles, i.e. those that are intersecting,
2. From the intersecting classes, identify those that are inside each others. Noting that all homology classes of second dimension that are intersecting are completely inside or outside each others only. This step is composed from several steps in its turn:
   - Make the orientation of the triangles in the classes to prepare them for the winding number computation.
   - For two intersecting classes A and B, take one point I inside of one of them.
   - Compute its winding number with respect to the other class in order to know if it's inside or not.
   - If it's inside, then the whole polyhedron where it belongs is inside the other and outside otherwise.
3. We separate the classes that are inside each others and we get a new contour $\mathcal{C}$ using $\mathcal{C} = (A \bigcup B) - (A \bigcap B)$, where A and B design two homology classes with $B \subset A$. Noting that when we say $A \bigcup B$ it means the 2-cells that belong to homology classes A and B and not the zone that is inside it.
4. We repeat his procedure until we get contours that don't contain other classes inside them.

3.4 Image segmentation using lifespans of homology classes

In this section we propose our first application of the persistent homology on grayscale images. We show that the combination of topological features with classical statistical ones leads to new ways of grayscale image segmentation. We will develop in 3.4.1 the tools and techniques that we used in our methodology, then in 3.4.2 we show some results that

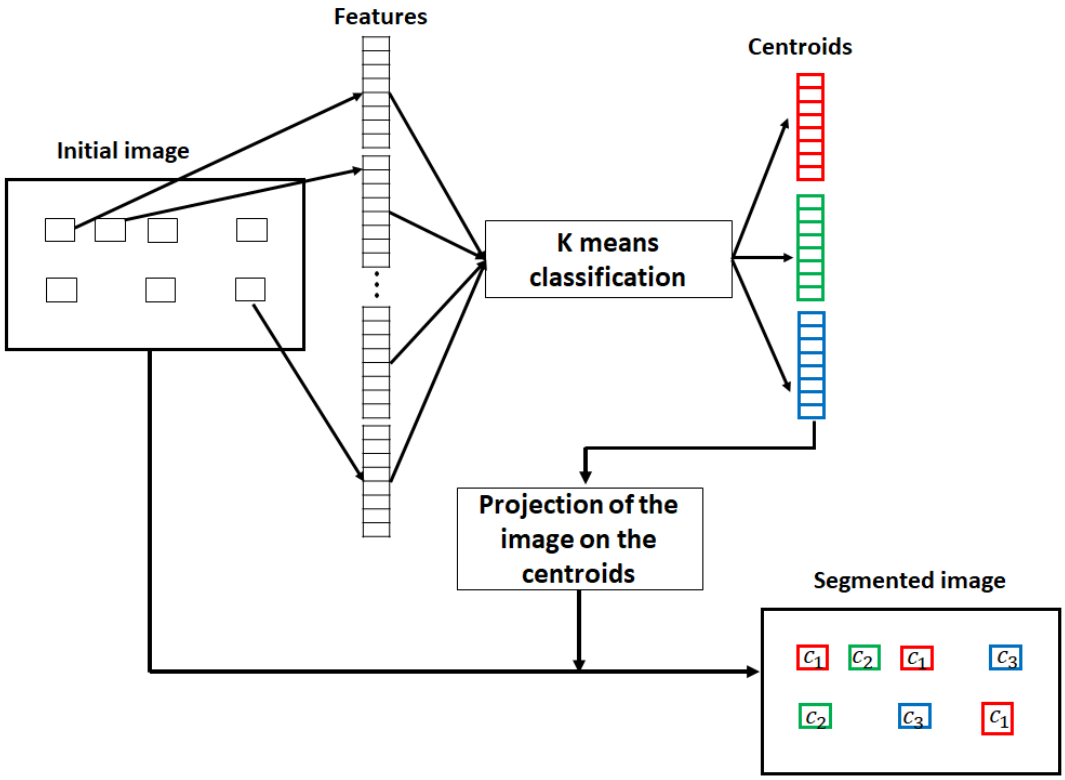


Figure 3.15: Image segmentation methodology plan.

illustrate two applications of this methodology in order to perform image segmentation on a satellite grayscale image and on unstained tissue section imaged by a quantitative phase imaging system.

3.4.1 Description of the method

Since homology is originally investigated to detect holes and voids in images, it can be very efficient in detecting cells or biological aspects in images during the scheme of filtration. For our image segmentation method, we analyze the images with superposed sliding windows. For each window we build a cubic complex whose vertices are the pixels, the edges connect the neighbor pixels, while the squares complete 4 neighboring edges.

The vertices weights for filtration are given by grayscale values while the edges and squares hold the maximum values of their vertices and edges respectively as described in 3.3.1 as shown in the figure 3.10. This filtration then builds a series of nested sub-complexes on which persistent homology is calculated. This allows to find homology classes and their lifespans.

In each window, we compute then several classical characteristics, the average and the variance of the gray levels to which are added topological characteristics, the means and variances of the lifespans of the homology classes of dimensions 0 and 1 and their persistent entropies.

Entropy in general is the average amount of information produced by a probabilistic source. The measure of entropy associated with each possible data value is the negative logarithm of the probability mass function for the value. Thus, the data source with a lower-probability value carries more information than the source with high probability value. The latter is defined in [MRS15] for every dimensions by $E = -\sum_{i \in I} p_i \log p_i$, where I represents the range of lifespans, $p_i = l_i/L$, l_i= date of disappearance - date of appearance, and $L = \sum_{i \in I} l_i$. Thus E can be formulated as $E = log(L) - \frac{1}{L} \sum_{i \in I} L_i log(l_i)$.

Noting that we will consider intervals that extend all the way to the end of the filter are denoted by $[a, \infty]$, and we will replace them by $[a, m]$ where m is the maximum value of the function that controls filtration.

Therefore, we will have 8 characteristics in each window, the mean and variance of zero dimension homology classes, the mean and variance of one dimension homology classes, the persistent entropies of zero and one dimension, and the mean and variance of pixel values inside each window.

These 8 characteristics: form vectors associated with each window. The figure 3.15 shows in its first step this assigning.

Then, all these characteristic vectors are classified in K classes. For simplicity, we considered an unsupervised classification by the K-means method, other supervised or non-supervised methods being feasible.

K-means is one of the simplest unsupervised classification algorithms aiming to partitioning a dataset $X = \{x_j / x_j \in \mathbb{R}^d\}_{j=1}^{n}$, with n objects composed of d features, into K disjoint clusters, $P(X,K) = \{P_k\}_{k=1}^{K}$ (also called partition), represented by their centroids, $C = \{c_k / c_k \in \mathbb{R}^d\}_{k=1}^{K}$. These clusters are estimated by minimizing the total within-cluster variation defined as:

$$D(C) = \sum_{j=1}^{n} \sum_{k=1}^{K} I(x_j \in P_k) \left\| x_j - c_k \right\|^2$$

where I is the indicator function defined as: $I(x_j \in P_k) = 1$ if $x_j \in P_k$ and 0 otherwise; and $\left\| x_j - c_k \right\|^2$ is the squared Euclidean distance between the j-th object x_j and the k-th centroid c_k.

The K-means objective function $D(C)$ is locally optimized by the following iterative algorithm:
Step 1: Initialize the K cluster centers C by choosing randomly K different objects of X.
Step 2: For each $j = \{1, \ldots, n\}$, assign the j-th object x_j to the m-th cluster p_m such as $m = \arg\min_{k=1,\ldots,K}(\{\left\| x_j - c_k \right\|^2\})$.

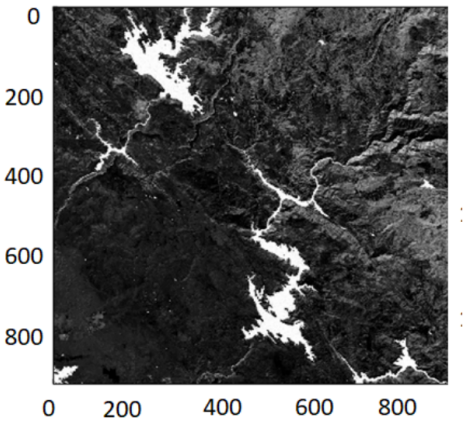
Step 3: For each $k = \{1, \ldots, K\}$, update the k-th cluster center c_k using $c_k = \frac{\sum_{x_j \in P_k} x_j}{N_k}$ where N_k is the number of objects belonging to the k-th cluster P_k.

Step 4: Repeat steps 1 to 3 until convergence of the algorithm.

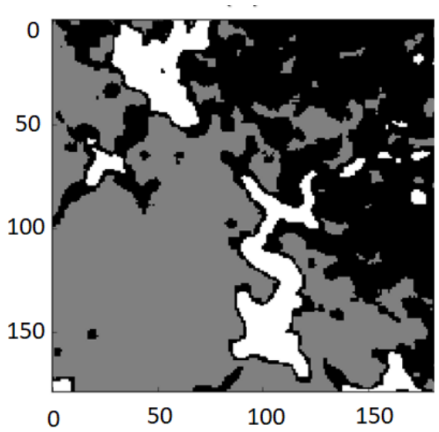
After the application of K-means, each window is then classified into a class as shown in the second step of the figure 3.15, which permits to achieve the segmentation of the image because we can affect each window in the image to a class.

3.4.2 Real applications

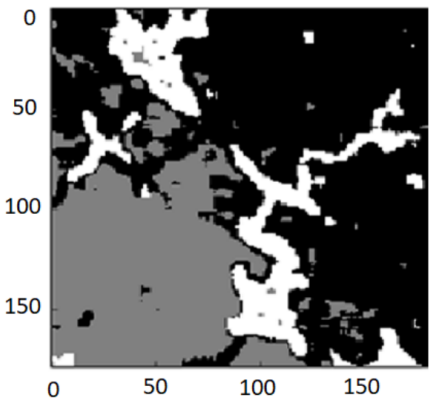
First we applied this methodology on a real grayscale image taken by a satellite shown in 3.16 (a) of size 929×960 pixels. This image is distributed into the snow at the middle, forest at left of snow and mountains at right. We processed the image with an overlapping square sliding window of size 30 × 30 pixels and the overlapping to 10 pixels. In 3.16 (b) we show the segmented image with the use of statistical characteristics only, the mean and the variance of the pixels, while 3.16 (c) exhibits the result obtained with the 8 characteristics listed in 3.4.1, i.e. combining the topological characteristics with the statistical ones. It's obvious that the former result doesn't success in discriminating the three classes of the image since the gray color that represents the forest at the left passes the snow and emerges in the mountain. The association of topological features permits to discriminates the 3 classes of the image and don't allow this immersion. This fact proves



(a) Original satellite image.

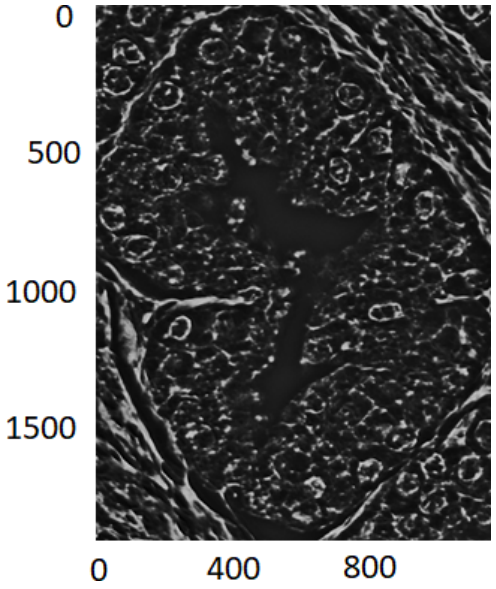


(b) Segmentation of the image using statistical characteristics.

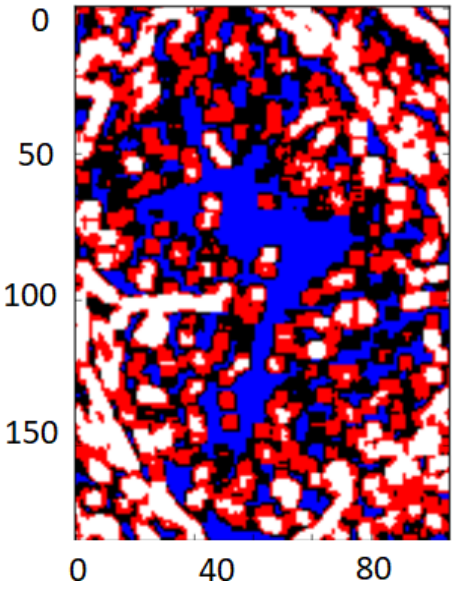


(c) Segmentation of the image using statistical and topological characteristics.

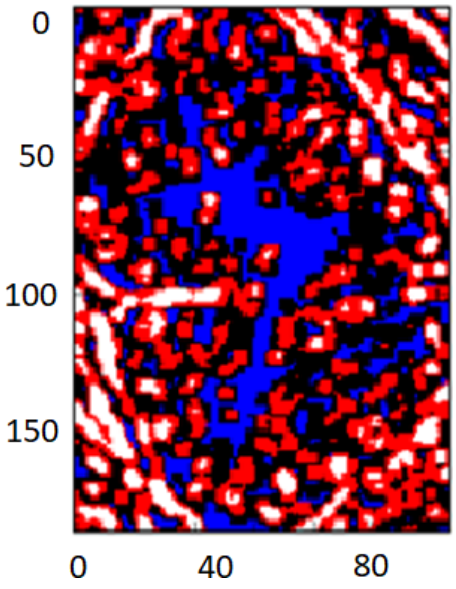
Figure 3.16: Satellite image segmentation.



(a) Original biomedical image.



(b) Segmentation of the image using statistical characteristics.



(c) Segmentation of the image using statistical and topological characteristics.

Figure 3.17: Biomedical image segmentation.

that these characteristics give more refined measures than only using statistical ones in sake of detecting texture features of images.

This methodology was also applied on a quantitative phase image of a prostate gland shown in the figure 3.17 (a), the grayscale levels reflect the refractive index map of the unstained histopathology slide. The same 8 charachteristics were computed, the six topological ones and the two statistical. We processed the image with an overlapping square sliding window of size of the window was chosen to 50×50 pixels and the overlapping to 10 pixels. Then the windows are classified using K-means to 4 classes. Noting that these measures can vary depending on the construction and the nature of application. The eight features were calculated for each window after filtration of non-interesting cycles. The segmentation of the gland shows four classes corresponding to the main types of tissue areas.

We show in figure 3.17 (b) the segmentation results of the gland using only the statistical charachteristics while the figure 3.17 (c) shows the results of segmentation using the 6 topological characteristics associated with the 2 statistical ones. For example, we see that the patches of the class represented in white do not discriminate between border of the tissue and its cells using the statistical characteristics while this distinction is respected depending on the topological characteristics associated with the statistical ones.

Moreover, to see the impact of persistent entropy for example on the segmentation procedure, we show in 3.18 the histogram of persistent entropy of first dimension for each class. We see clearly that the values of persistent entropy define four distinct distributions according to the classes, which shows the contribution of the topological characteristics in image segmentation. For example, patches of the colored class in blue have an entropy close to 0, those colored in black, red and white are close to 4, 4.5 and 5 respectively. This permits to isolate the tissue and to identify the cells and stroma when this characteristic is accompanied by the others presented for example.

The results discussed in this section were presented in an international conference [AGV16a] and published in a journal [AGV16b].

3.5 2D and 3D object segmentation using homology classes

In contrast to works that focus on partitioning the image to non-overlapping homogenous regions that segment the entire image, we are looking in this section to object segmentation by finding interesting parts in the studied image. The purpose is to define an object segmentation methodology based on the persistent homology. We will show the potentiality of the homology classes that form the significant topological features persisting for a long time in the filtration procedure in order to perform object segmentation in any dimension.

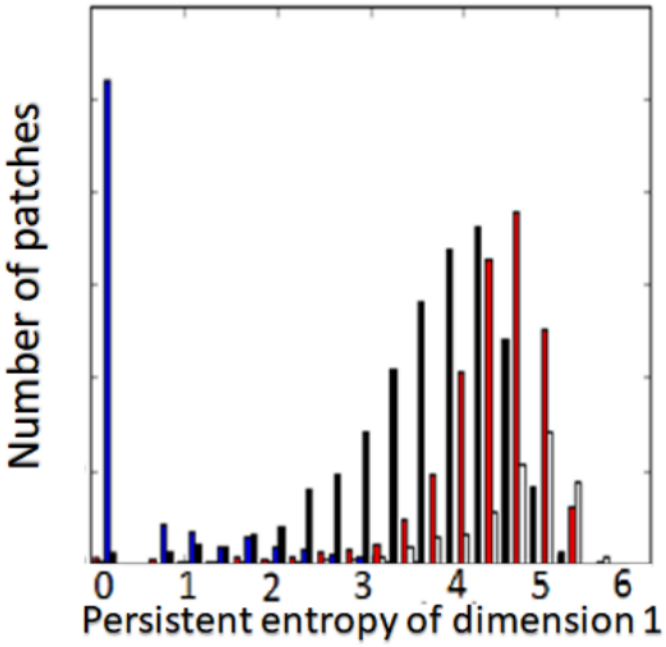


Figure 3.18: First persistent entropy distribution with respect to classes.

In some of the works, the binarization of the pixel intensity during of the process of filtration reduces the functionality of raw pixels' intensity all along the construction of the topological complex. As well, other methods are based on a topological study that depends solely on the values of the input function without the benefit of the construction of topological complex on the pixels' image [LF07, PD07].

Unlike the methods that depend their construction on many parameters like objects sizes or intensities, our tool doesn't need the use of a prior parameter. Moreover, while the existed methods perform segmentation on images of specific dimension, we benefit from homological persistence in the methodology described in this section for processing image segmentation to support multidimensional tasks.

On the other hand, other works benefit from the statistical outcome of the lifespans of the persistent topological features and use its results to perform classification on image patterns [ARC14, DEL+16], or like the methodology we proposed in previous section 3.4, instead of looking for its algebraic effects in order to perform image segmentation.

This section demonstrates also that the methodology that we based our approach on has a multidimensional efficiency. It can be successfully applied on 2D grayscale images as same as on 3D images. A combination between the methodological topological construction and image superpixels can also be executed successfully to achieve the segmentation aimed for.

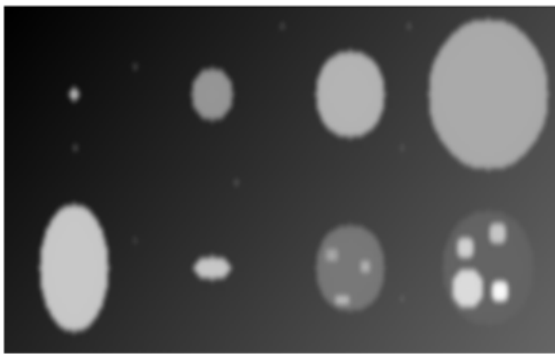


Figure 3.19: A synthetic image.

Moreover, this technique had shown its various advantages over existing methods. It is independent from prior parameters and metrics and doesn't demand preprocessing steps. Also its strength is revealed in its invariance to continuous deformations. The same results would be obtained if the image was rotated, stretched, rescaled, etc. All these criteria makes it suitable to achieve more general tasks than the existing methods.

We start by representing some images, a synthetic and real images, each one represents difficulties in object segmentation on different levels. Then we explain classical methods of object segmentation like Otsu thresholding and Maximally Stable Extremal Regions (MSER). We show the results on the synthetic image and the other images then we pass to our method to prove its efficiency and its advantages.

3.5.1 Synthetic and real images

Returning to the application, we want to make a comparison between our method and other existing methods. For this purpose, we made a synthetic image of size 240×120 pixels shown in figure 3.19, that contains objects of different sizes, shapes and intensities as well as some noise. Some of these objects are overlaid in other ones and all object are overlaid on a slightly variable background. The objectives here is to find all the objects, with the overlaid ones and discriminating them from the background without the use of prior parameters. Other images having the same problems will be the center of the application of our method in the rest of this section.

As a sample of real images, we have an image of dots of size 513×282 shown in figure 3.20. The dots don't have same sizes, forms or intensities and some of them are hard to distinguish by naked eye. Their intensity is very close to the non homogeneous background. Moreover a simple image of size 300 × 246 containing 10 coins shown in figure 3.21 will be also considered. This image is simple having some features on its coins with an homogeneous background and can play the role of a witness to object segmentation methods.

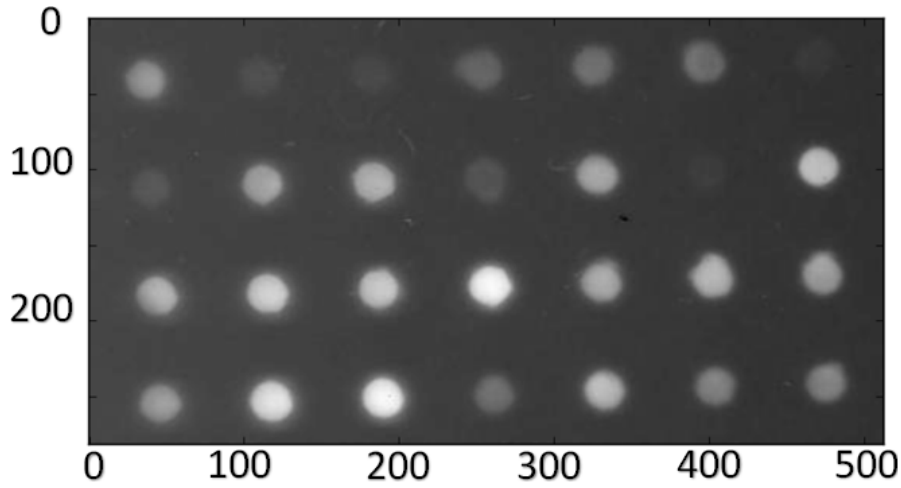


Figure 3.20: An image containing 24 imperceptible dots.

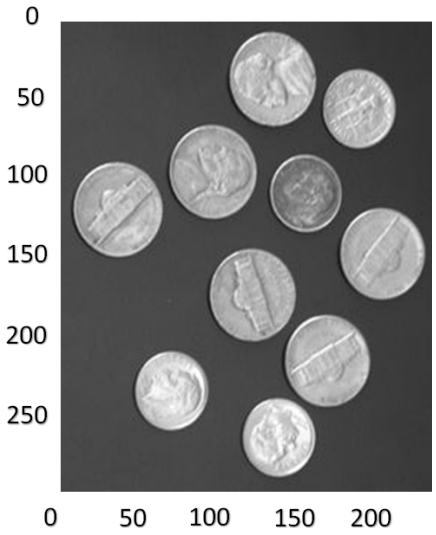


Figure 3.21: 10 coins image.

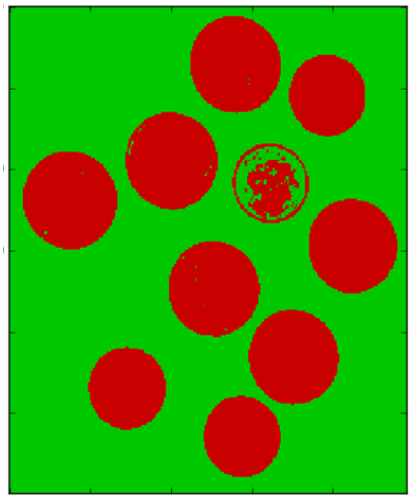


Figure 3.22: Results of Otsu segmentation on coins image.

3.5.2 Otsu thresholding

Thresholding is the simplest segmentation method. The pixels in the image are partitioned depending on their intensity value. Global thresholding consists on binarizing an image using an appropriate threshold T:

$$g(x,y) = \begin{cases} 1 \text{ if } f(x,y) > T, \\ 0 \text{ if } f(x,y) \leq T \end{cases} \tag{3.13}$$

Otsu thresholding [GBY+18] is one of the very well known and very effective methods of global thresholding. The goal in Otsu threshold is to find the threshold that minimizes the spreads of two clusters of images by optimization of the best threshold. We start by defining the within-class variance as the weighted sum of the variances of each cluster:

$$\sigma^2_{Within}(T) = n_B(T)\sigma^2_B(T) + n_O(T)\sigma^2_O(T) \tag{3.14}$$

where $n_B(T) = \sum_{i=0}^{T-1} p(i)$ such that $p(i)$ = nb of pixels of intensity i/total number of pixels; $n_O(T) = \sum_{i=T}^{N-1} p(i)$; $\sigma^2_B(T)$ is the variance of the pixels below the threshold; $\sigma^2_O(T)$ is the variance of the pixels above the threshold and $[0, N-1]$ is the range of intensity levels. Subtracting the within-class variance from the total variance of the combined distribution, will allow to get the between-class variance:

$$\sigma^2_{Between}(T) = \sigma^2 - \sigma^2_{Within}(T) = n_B(T)n_O(T)[\mu_B(T) - \mu_O(T)]^2. \tag{3.15}$$

where σ^2 is the combined variance and μ is the combined mean.

So, for each potential threshold T we do the following steps:

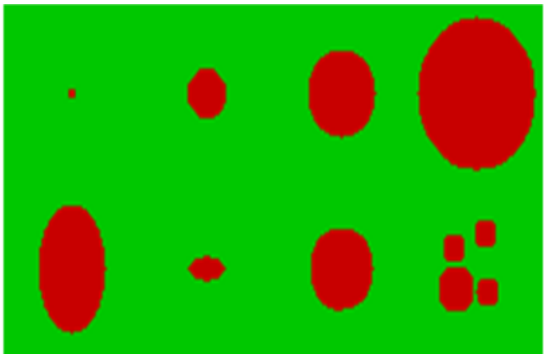


Figure 3.23: Otsu's thresholding result on the synthetic image

1. According to the threshold, we separate the pixels into two clusters
2. Then we compute the mean of each cluster.
3. We square the difference between the means.
4. Multiply by the number of pixels in one cluster times the number in the other.

The value that maximizes the between-class variance or, conversely, minimizes the within-class variance will be the optimal threshold.

First we apply the Otsu's method on the image of 10 coins, we get the results in figure 3.22. We remark that the results of segmentation are pretty good but a single coin represents a weak points since it contains pixels with intensities below the threshold, which will make them seen in green color.

On another hand, the figure 3.23 shows the results using the Otsu thresholding on the synthetic image of figure 3.19. We remark that this method is not able to segment the small objects that are overlaid in the bottom right but not this big object because of the use of an unique threshold.

Another application of Otsu's method that show its disadvantages is the image of dots in figure 3.24. The dots don't have same sizes, forms or intensities and some of them are hard to distinguish by naked eye. Also, the spaces between these dots contain plenty of noise. We remark that Otsu segmentation didn't succeed in detecting all the dots in this image because of its depending on intensities in the specified image. We will see after how our method based on the persistent homology successfully detected and segmented all the dots in this image.

3.5.3 MSER technique

To overcome the use of an unique threshold, the Maximally Stable Extremal Regions (MSER) method described in [GMACA14] uses different thresholds to iteratively detect objects and store them into a hierarchy or tree-like structure. Hence, for the lowest thresh-

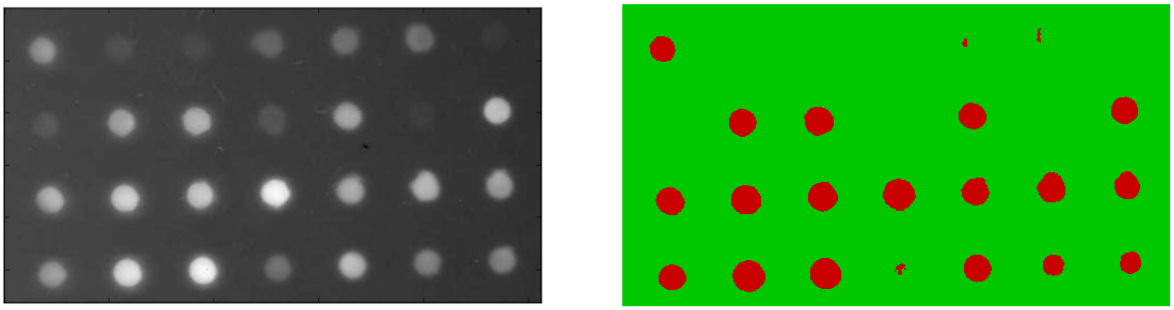


Figure 3.24: An image of dots and its otsu thresholding result.

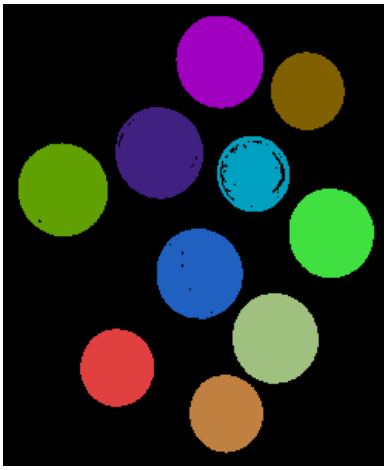


Figure 3.25: Results of MSER on coins image.

old only background is detected. With increasing threshold, some objects are detected and if their volumes fall into a specific range, these objects are extracted and stored into the tree-like structure. If the same object is detected with different thresholds, this object is stored into a branch of the three then the best object from this branch is extracted. For MSER it is the object having minimal volume variation between two thresholds. In case an object will be divided into smaller objects at higher threshold, a division will be created into the tree-like structure. Best detected objects for each branch are computed and displayed in different channels in the final results image.

Applying the MSER on the coins and dots images in figures 3.22 and 3.24 we get the results in figures 3.25 and 3.26. These images correspond to the superposition of the objects detected in the three branches of the tree-like hierarchy that we obtain for the initial images. The obtained results show the potential of this method.

We now turn to the complex image that contains objects of different sizes and intensities and with a non homogeneous background, the synthetic image. Using the MSER, and considering the entire tree structure, we get the result in figure 3.27(a). In the first channel shown in figure 3.27(b), this method identifies some of the objects with different

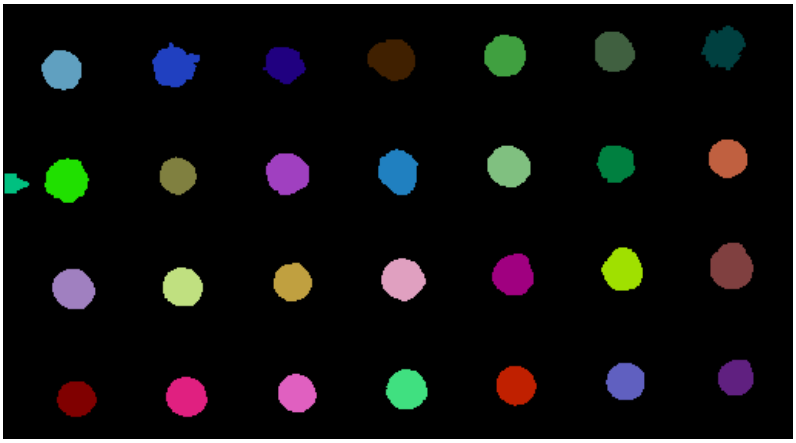


Figure 3.26: Results of MSER on dots image.

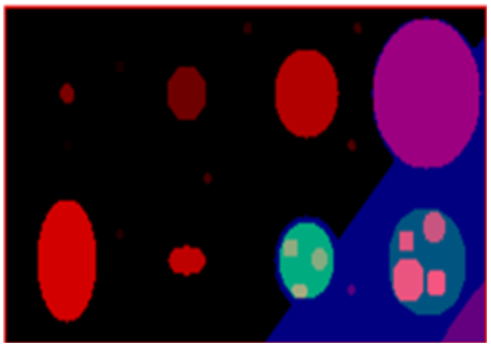
sizes. However, this method identifies also the small dots corresponding to the noise as well as a part of the background on the bottom-right. In the second channel shown in figure 3.27(c), that contains bigger object emerging from the merging of smaller objects, two other objects that are the base of the smallest ones are highlighted. In the third channel shown in figure 3.27(d), these two objects as well as the object with the greatest size from the first image are embedded into another part of the background.

This method falls to identify only the interesting objects since the noise is detected to. Moreover, this method depends on the background level, the objects not being extracted correctly. One way to segment overlaid objects with their bases and to avoid the noise is to put a specific range of volumes as parameter. In this manner, the MSER will achieve segmentation depending on the given corresponding information. Only objects that accord to the corresponding range volume will be detected. For example, figure 3.27(e) shows the objects that corresponds to the range of volume $100-1000$. A part of the background isn't eliminated, and not all objects can be identified. Moreover, if the volume parameter is lowered in order to extract the small objects, the noise will appear too. If the volume parameter is large enough, we cannot extract the small object which are overlaid over other objects.

3.5.4 Proposed method

3.5.4.1 Synthetic image

Considering our method, a topological complex is first built directly on the pixels of the image in figure 3.19 as we explained before. In an analogue way to the tree, the filtration scheme is constructed. Points, edges and 2-dimensional cells are added to the complex as the intensity increases until we get the whole complex. Computing the persistent homology on this complex gives the persistence diagram of the homology classes. This persistence diagram, shown in figure 3.28(a) is the main output of our method since it



(a) All MSER detected objects.

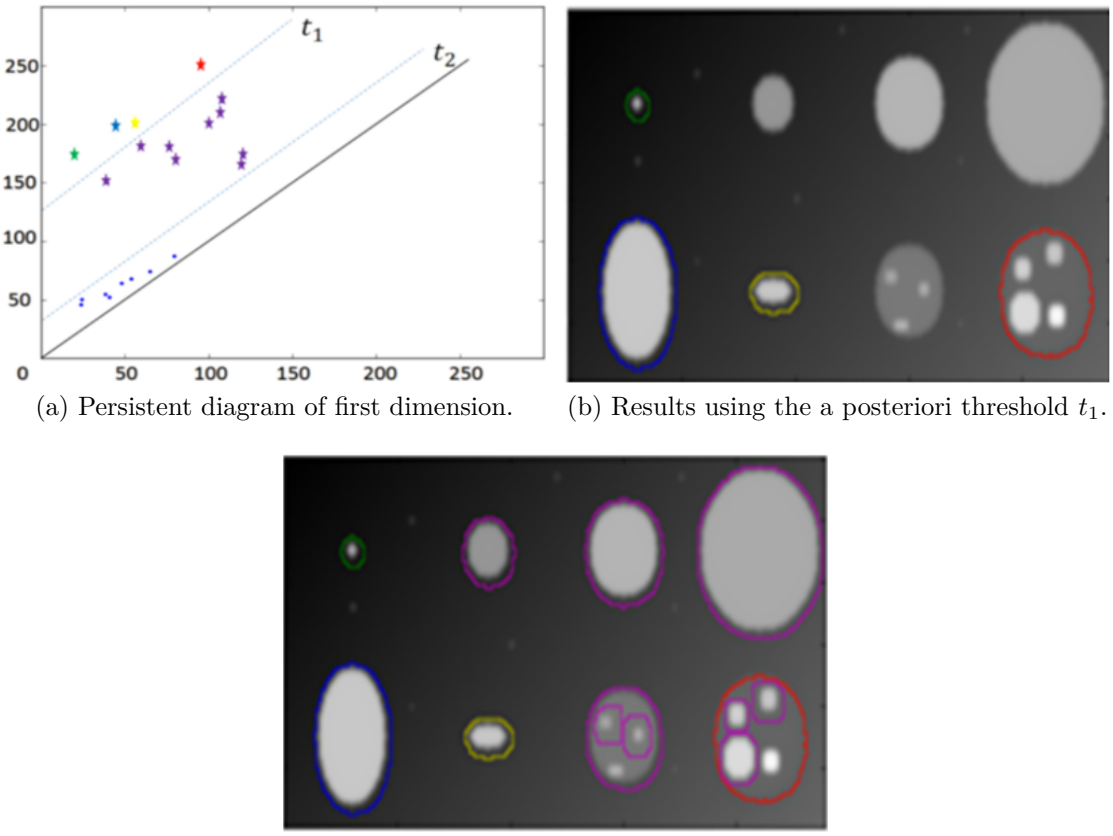
(b) First channel of the image in (a).

(c) Second channel of the image in (a).

(d) Third channel of the image in (a).

(e) Objects in volume range 100-1000.

Figure 3.27: Results of MSER method on the synthetic image.



(a) Persistent diagram of first dimension.

(b) Results using the a posteriori threshold t_1.

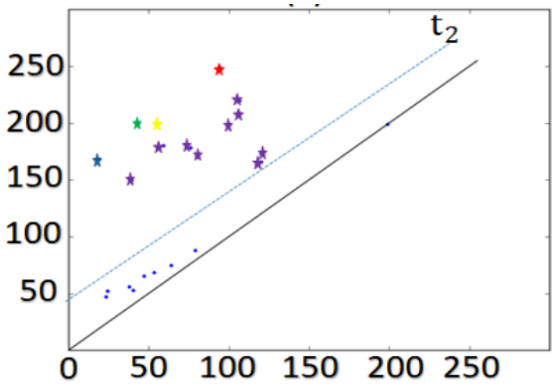
(c) Results using the a posteriori threshold t_2.

Figure 3.28: Results of the proposed method on the synthetic image.

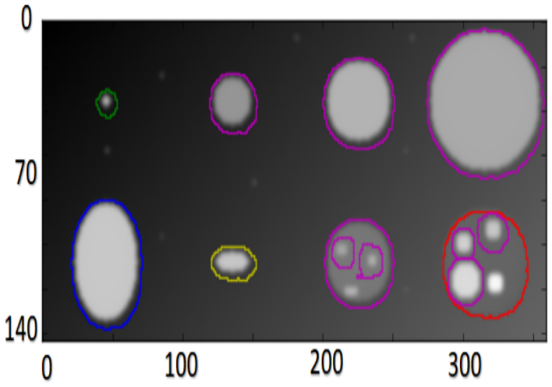
indicates the births and deaths of homology classes of first dimension. Each point carries all the information necessary to homology classes. The importance of the points and what they represent is proportional to their distance from the diagonal. Thus, points that are close to the diagonal have a small lifespan meaning that they correspond to the noise, while those that are far are interesting.

It must be noted that having the construction of the complex, we can find for each point in the persistence diagram the x-y coordinates of the corresponding class. For example, the homology class represented with a * in figure 3.28(a) corresponds to the class that identifies the object shown in blue in figure 3.28(b), the homology class represented with * to the one corresponding to the object in red, etc. Highlighting the homology classes in the persistence diagram allows then to select interesting objects by imposing a parallel threshold to the first diagonal. For example, the homology classes that represent points above the parallel (t_1) are highlighted in figure 3.28(b) by their corresponding colors. Pulling down the parallel to (t_2) will allow to highlight other homology classes that are represented by all points above (t_2) as in figure 3.28(c).

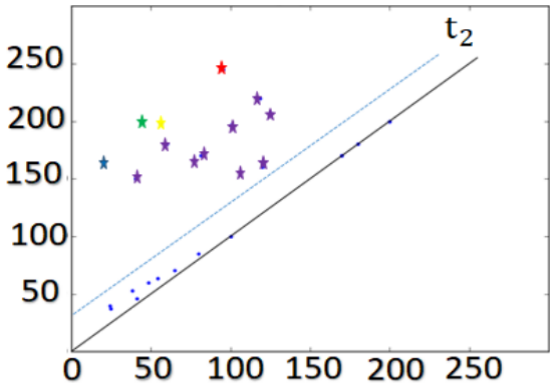
Our method allows to compute all the homology classes in a single pass and to highlight them by simply changing a posteriori threshold. It's insensitive to continuous transforma-



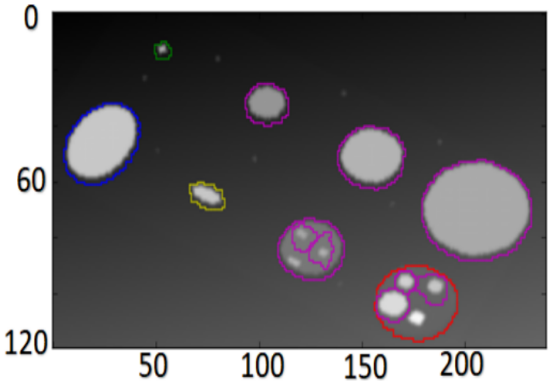
(a) Persistent diagram of first dimension of the scaled image.



(b) Results on the scaled image using threshold t_2.



(c) Persistent diagram of first dimension of the rotated image.



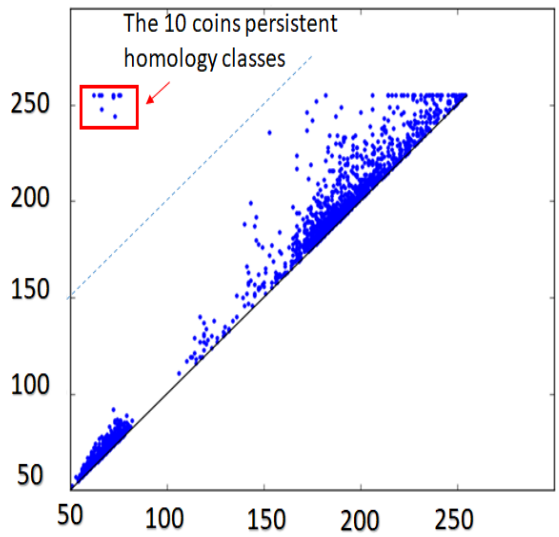
(d) Results on the rotated image using threshold t_2.

Figure 3.29: Results of the proposed mehod after scaling and rotating the synthetic image.

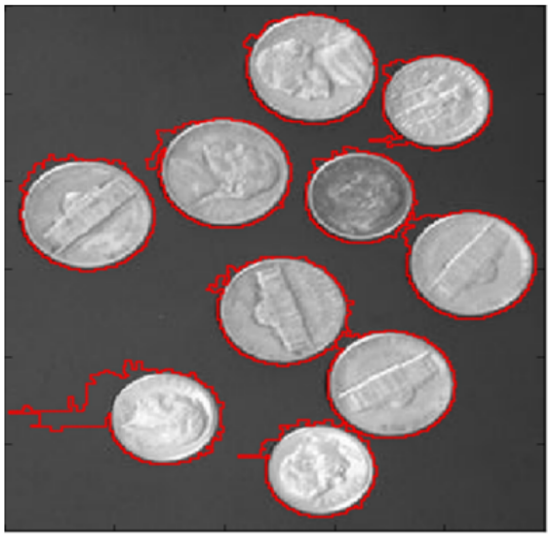
tions of the images as rotation, stretching or scaling etc. For demonstration purposes, the figure 3.29(a) shows the persistence diagram of first dimension after scaling the synthetic image in figure 3.19 to 150% and 120% of the initial width and height respectively. Also, we show in 3.29(c) the persistence diagram of first dimension after rotating the synthetic image by 30 degrees.

Using the same threshold (t_2), the same objects as previously are segmented. Indeed, the corresponding homology classes are colored stars in the scaled and rotated image in figures 3.29(b) and (d). The same important homology classes are obtained after scaling or rotating the image. Some slight changes have occurred in lifespans of classes and the presence of some noisy classes of null lifespans, due to the differences in the intensity values caused by scaling and rotation. However, these slight changes don't affect the result, the same threshold (t_2) highlighting the same classes into original, rotated and scaled images.

Moreover, the homology classes are not optimized as MSER does by choosing objects with specific volume range. This can be done as a post processing by classical methods



(a) Persistent diagram of first dimension.



(b) Results on the coins' image using the a posteriori threshold.

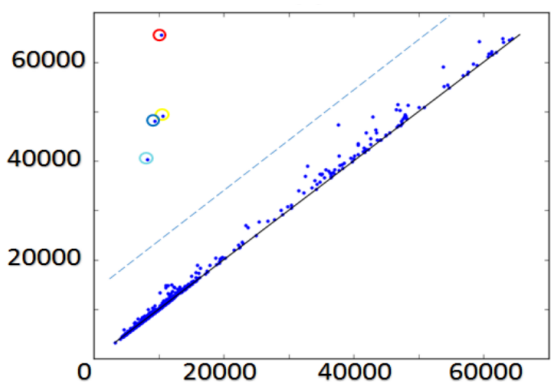
Figure 3.30: Results of object segmentation on coins' image.

like active contour or thresholding techniques in order to obtain a pertinent segmentation for a specific application. Active contours start with an initial guess for the contour represented by a closed curve, which is moved after to the boundaries of the desired objects. This can be executed by operations of shrinking or expansions that depend on the constraints of the image. Noting that these operations are done by simulation of partial differential equations [OS88] or by the minimization of an energy function. While thresholding techniques, can start with the boundary of the created homology class to classify their interior into object or non object using the histogram of pixel values [PG87].

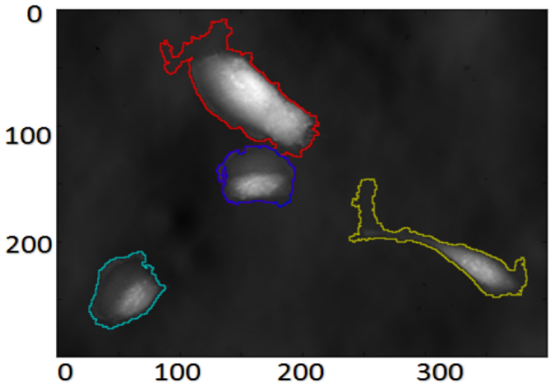
3.5.4.2 Real applications

This part is dedicated to the presentation of the results obtained by the proposed method on the real images. We apply first our algorithm on the image of ten coins shown in figure 3.21. Computing the persistent homology on this complex, will provide as a result the persistence diagram shown in figure 3.30 (b).

It seems clear that the coins will be holes of the subcomplexes in the nested sequence for a long time. These holes are detected by the persistence tool as can be seen on the persistence diagram. The advantage of the persistence diagram is his capability to present the surviving duration of an homology class. It can present its times of birth and death on the horizontal and vertical axes respectively. We can find all the information necessary to homology classes behind each point in the persistence diagram. The homology classes that are born to form the cycles that encircle the holes at a level i will die after, when they are filled by 2 dimensional cells at another higher level j going through the filtration



(a) Persistent diagram of first dimension.



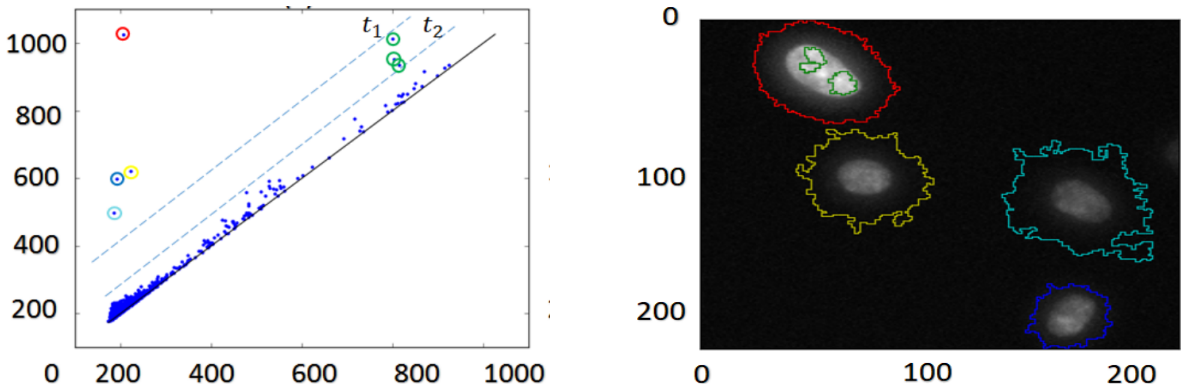
(b) Results on the image using the a posteriro threshold.

Figure 3.31: Results of our method on a biomedical image.

scheme. The basis of these classes are composed by 1D chains of edges. Highlighting the basis of the most persistent homology classes of first dimension as shown in figure 3.30 (a) will achieve the segmentation aimed for. The lifespans of the homology classes is proportional to their distance from the first diagonal. As an advantage, we can control the significance of a class by sliding a parallel to the diagonal of the diagram as threshold. In this manner, the ten points on upper left corner above the threshold are far away from the diagonal and we highlight their homology classes by a red color figure 3.30 (b) while other points are considered as noise.

As another real application, we consider a biomedical image of size 400×300 pixels shown in figure 3.31(b), taken by a technique that uses the SID4Bio quantitative phase imaging system introduced in [BMW09]. The result of our method is a persistence diagram shown in figure 3.31 (a) that represents the homology classes of first dimension highlighted on the image and executed in 13.4 seconds. For visualization purpose, we highlight the most persistent classes that are represented by points above the dashed parallel with their corresponding colors to segment the salient objects in figure 3.31 (b).

Another application on a classical grayscale image of cells also obtained by the SID4Bio quantitave phase imaging of size 210×220 is shown in figure 3.32. As described before, the most persistent homology classes are highlighted by their corresponding color in persistent diagram to segment the salient objects in the image. Points above the threshold (t_1) represent classes that segment entire cells, each point represented by its corresponding color. Moving down the threshold to (t_2), the method is able to detect cells components as we can see the classes in green. This technique permits to find all the homology classes in the same time and to visualize more or less important details by simply varying a threshold on the persistence diagram, which permit to segment biological objects and their constituents. These results were represented in an IEEE International Conference on Advances In Biomedical Engineering in Beirut [AGK17b]



(a) Persistent diagram of first dimension.

(b) Results on the image using the two thresholds t_1 and t_2.

Figure 3.32: Results of the proposed method on a biomedical image that shows components segmentation.

3.5.5 Object segmentation based on superpixels

Now, we benefit from the advantages given by superpixels, specially in the reduction of the constructed spaces, to perform object segmentation and we apply our methodology on images presegmented to superpixels using the SLIC technique [ASL+12].

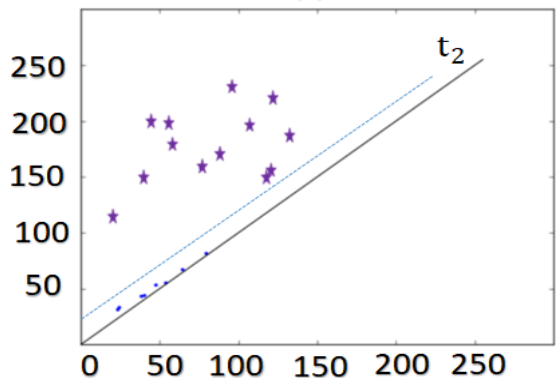
3.5.5.1 SLIC method

Before merging into our application, we will provide a brief introduction into SLIC superpixels method. The authors in [ASL+12] claim that the algorithm "simple linear iterative clustering" (SLIC) is fast, easy to use and more robust than state of the art existing superpixels method. The algorithm SLIC uses the 5-D space composed by x, y pixel coordinates and L, a, b values of the CIELAB color space to execute a local clustering. To reinforce compactness and uniformity in superpixel architecture that smoothly fits grayscale and color images, the authors propose a new distance measure in their algorithm.

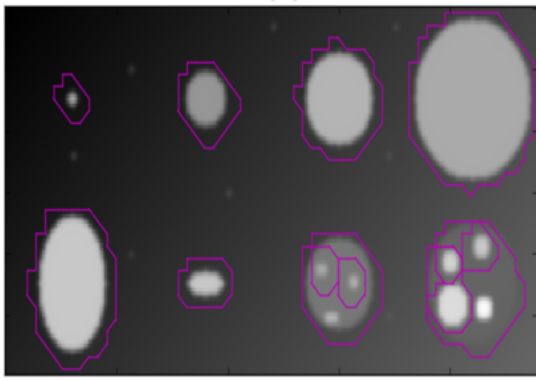
Assuming that n is the number of pixels in the image and k the number of desired superpixels, then the desired interval grid $S = \sqrt{n/k}$ and pixels that are linked with each initiated cluster center $C_k = [l_k, a_k, b_k, x_k, y_k]$ lies within a $2S \times 2S$ area around the superpixel center. In their algorithm, first they begin by sampling K regular spaced cluster centers, then they move them to the locations that correspond to the lowest gradient position in a 3×3 neighborhood. This gradient is computed as:

$$G(x, y) = \|I(x + 1, y) - I(x - 1, y)|^2 + \|I(x, y + 1) - I(x, y - 1)|^2. \tag{3.16}$$

where $I(x, y)$ is the Lab vector color space corresponding to the (x, y) position of pixel and $\|.|^2$ is the L^2 norm. Then for each cluster center they assign the best matching pixels



(a) Persistent diagram of first dimension.



(b) Results on the image using the threshold t_2.

Figure 3.33: Results of the proposed object segmentation method on the synthetic image using a SLIC space reduction.

from the $2S \times 2S$ square neighborhood around the cluster center according to the distance measure defined as:

$$\begin{aligned} d_{lab} &= \sqrt{(l_k - l_i)^2 + (a_k - a_i)^2 + (b_k - b_i)^2} \\ d_{xy} &= \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \\ D_s &= d_{lab} + \frac{m}{S} d_{xy}. \end{aligned} \tag{3.17}$$

A variable m is introduced in D_s to allow the control of compactness of superpixels and it's equal to 10 in default. After associating the pixels with their nearest cluster centers, a new cluster center is computed as the average of labxy vector of all the pixels in the cluster.

3.5.5.2 Synthetic image

We propose to apply this approach first on the synthetic image in figure 3.19. As described in section 3.3.3, we build our topological complex on the set of superpixels, then we apply the steps of computation of persistent homology on this complex. Each superpixel is represented by a 0-cell (S_i) in our topological construction. These 0-cells are represented by centers of their associated superpixels and they will hold the mean value of their corresponding pixels. Each 1-cell $(S_i S_j)$ in the homology class will be an edge connecting (S_i) and (S_j), the centers of two neighbor superpixels, and will take the maximum value of its corresponding 0-cells. The higher dimensional cells follow the same construction.

After presegmentation of the image 2000 superpixels instead of 240×120 pixels, the births and deaths of these homology classes of first dimension extracted from the set of superpixels are shown in figure 3.33(a). Some slight differences are observable as comparing

them with the homology classes shown in figure 3.28(a). These differences are due to the fact that the values of 0-cells are not the same since a superpixel realize an averaging of several raw pixels. However, we found the same number of homology classes of first dimension, represented by *, as well as the same distribution of the topological noise.

Imposing the same threshold as previously, the same objects as in figure 3.28(j) are segmented, as shown the results in figure 3.33 (b) that represents the object segmentation by keeping the classes represented by *. We notice that the outlines of the segmented objects are roughest, which is explained by the fact that the 0-cells are the centers of superpixels that are more spaced out than the raw pixels. However, the computation time is substantially shorter (2.876 seconds instead of 7.134 for raw pixels) because we compute persistent homology over 17278 cells instead of 114481 and the amount of the required memory for the computation is lower.

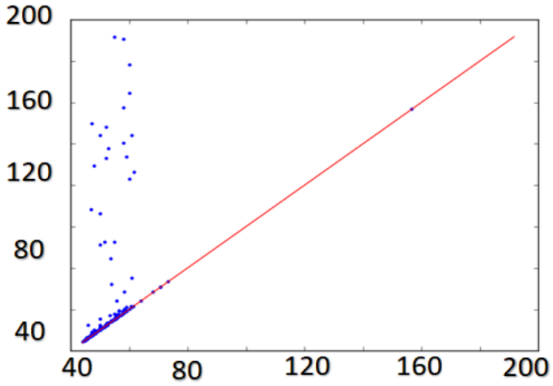
3.5.5.3 Real 2D images

Here we represent some results after topological constructions obtained on superpixels of 2D real and biomedical images to demonstrate the potential of our method.

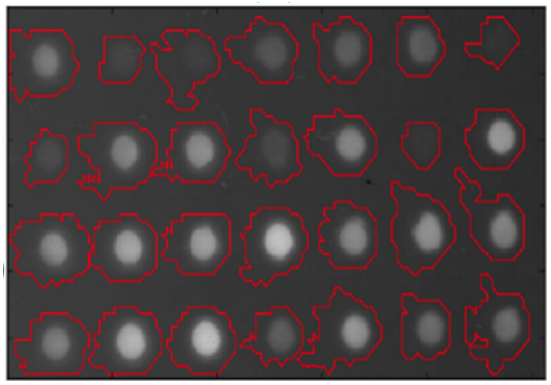
As a real image application, we consider employing the method on superpixels of the image of dots of size 513×282 shown in figure 3.20. As a result of computing persistent homology on the set of the 5000 superpixels, we have the lifespans of homology classes of first dimension illustrated by the persistent diagram in figure 3.34(a). The persistent diagram demonstrates the 28 homology classes far from the diagonal and that represent the dots. We highlight in figure 3.34(b) these homology classes that represent the dots of the image. Depending on the nature of the input image, this methodology can also be applied using the superfiltration procedure by starting from the maximal intensity in a decreasing way and in an analogue way to subfiltration as described in section 3.3. Applying the method on the superpixels of the image after reversing the intensities values, we will get the same homology classes of first dimension as shown in figure 3.34(a) and we highlight the most persistent between them to get the image in figure 3.34(c).

As a biomedical application, we consider a 500×512 classical image of cells of 16 bits as shown in figure 3.35(a). We build our topological complex on the superpixels set and we compute the persistent homology. As same as explained before, we highlight the most persistent homology classes of first dimension. The persistent homology gives us an extra information about the including criteria of homology classes that segment objects in biomedical images.

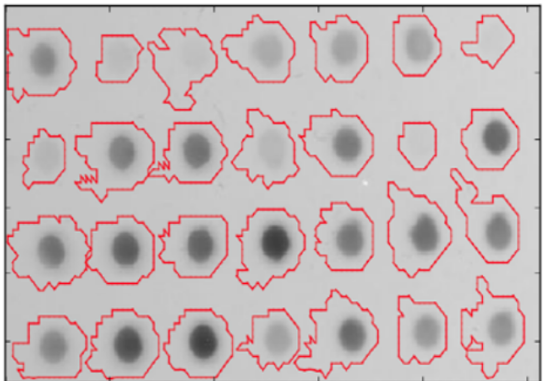
As explained in chapter 2, there's no canonical representatives for the homology classes, thus we can face homology classes that are nested in each other and thus that contain many interesting objects. We proposed in subsection 3.3.5, a method to deal with those that are nested in 2d images. In figure 3.35(b), we remark that the classes of bigger lengths, as the class (7) are born before the smallest, as (8) and (9) and dying after them, thus they have bigger lifespans. Using the information derived from homology classes, and



(a) Persistence diagram of first dimension of the image of dots in 3.20.

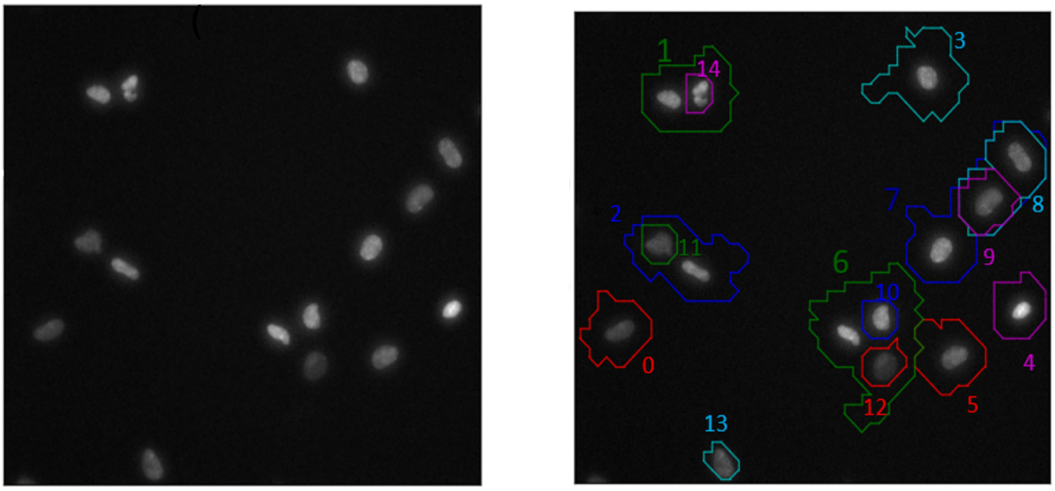


(b) Results of subfiltration.



(c) Results of superfiltration.

Figure 3.34: Results of the proposed segmentation method on the image of dots and its inverse using a SLIC space reduction.



(a) The initial biomedical image

(b) Results of segmentation on the biomedical image

Figure 3.35: Segmentation results of the proposed method on a 2D real biomedical image using SLIC space reduction.

the algorithm of separation of homology classes of 2D case described in subsection 3.3.5, we can know the following including scheme of classes, for example: (9) $\subset$ (8) $\subset$ (7) and (11) $\subset$ (2) and eventually separate them to have one contour to each object.

3.5.5.4 Real 3D images

In order to demonstrate the multidimensional aspect of our method, we present in figure 3.36 a segmentation of a 3D biomedical grayscale image of nucleolus using the superpixels and persistent homology combination construction. This figure represents the results of segmentation of 61×249×308 grayscale biomedical image of nucleolus at z = 0, 15, 30, 40, 50 and 60 respectively. The homology classes of second dimension are chains of 2-cells or triangles ($S_iS_jS_k$).

We highlighted the seven most persistent homology classes formed by 2-chains or sums of triangles. Using the algorithm of separation of homology classes of 3D case that we proposed in 3.3.6, we were able to detect the including criteria of the homology classes and separate them.

As a result, we obtain in figure 3.36 four essential objects. The homology classes that represent them and mentioned in increasing order of lifespans are highlighted in red (the bottom right), green, blue, red (the top left). Our tool is capable to detect the interesting objects inside them and that have smaller lifespans using the proposed algorithm. As shown in figure 3.36, the old blue homology class that detect the essential objects contains two other younger classes highlighted in red and green, while the green one contains a younger blue class. The separation algorithm is capable to know the inclusion criteria for

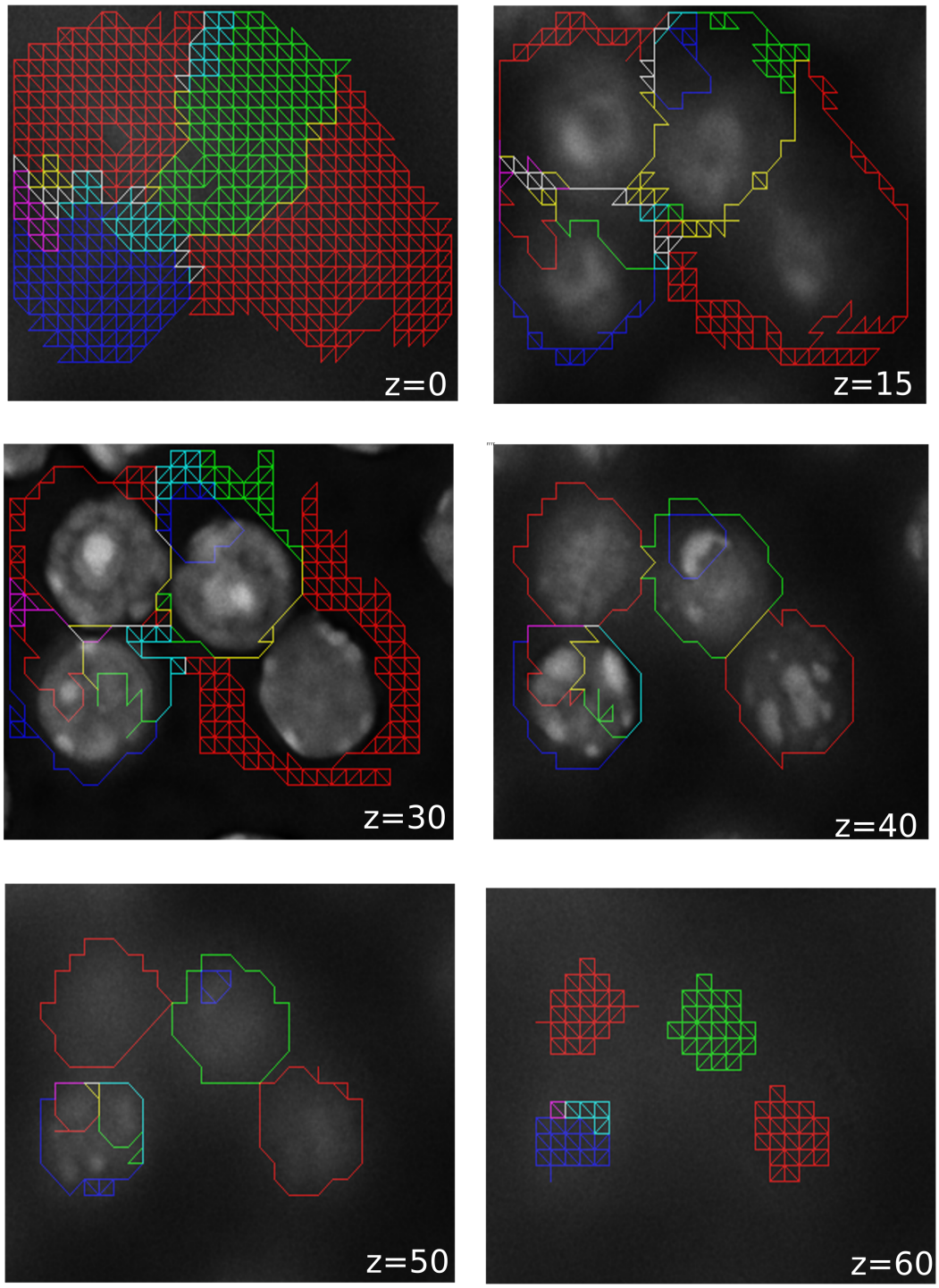


Figure 3.36: Results of the proposed segmentation method on a 3D image.

all these classes as shown in the results.

The methodology described in this section and a part of the results motivated the submission of the article [AGV+17] to the Pattern recognition letters journal .

3.6 Object tracking using relative persistent homology

In this section, we introduce a new way of 2D+t object detection for grayscale images based on relative persistent homology. The power of this method is revealed also in its independence to prior parameters, the generability that characterizes its construction and its insensitivity to continuous deformations of the image. Detecting interesting objects is done by highlighting the 2D squares of long life second relative homology classes as explained in 3.6.1. This technique can also be restricted to 2D images as well as extended to greater dimensions.

The potential of this method is shown on a synthetic example 3.6.2. The two spheres that don't appear on the first and last frames can be identified by a classical absolute homology, while the two moving rectangles in all frames are identified only by the relative homology. In 3.6.3 we apply our method on two real grayscale image sequences. This method allows identifying interesting parts and detecting the temporal evolution of vesicles along the sequence. A comparison made with other tracking tools demonstrates the capability of our approach over some of the state of art methods.

3.6.1 Description of the method

Object detection and tracking are usually regarded as one of the major and challenging tasks in the pipeline of image processing and pattern analysis. There are many techniques that have been proposed and developed in the literature, such as frame differencing [AC17, RYK14], point detectors [ZMAT16, GKK+15], background subtraction [THV16, YQF+14], supervised learning [BDA16, TT12] and deep learning techniques [CHTH15, ZPL16] see section 2.1 for thorough explanations.

The quick advancements in image processing tools and approaches ensure the development of new methods for their analysis. In this context, algebraic topology can be a very interesting field that produce alternative techniques in object tracking [Ghr08b, Car14].

The purpose in this section is to propose a novel, generic and parameter free way of performing 2D+t track detection in grayscale images based on the relative persistent homology. This method identifies the interesting parts in an image and detects temporal evolution of objects in time sequences of images using the most relative persistent homology classes of second dimension. A nested sequence of subcomplexes is built from the set of pixels of the image. Then relative persistent homology is computed through

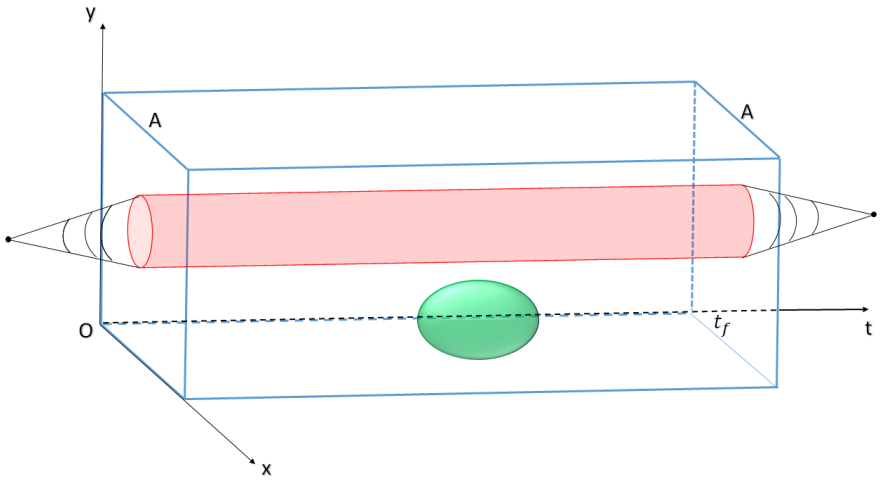


Figure 3.37: The "worm hole" and sphere detection by relative homology.

this sequence. This augmented version of absolute homology allows to detect objects in images' sequences from the first to the last frame by highlighting the persistent classes.

As far as we know, relative persistent homology has not been applied until now to the processing of images. However, it has been probed in other engineering applications. In [PGK16], the authors present an algorithm for topological trajectory clustering of robots and vehicles based on relative persistent homology. A study of the susceptibility to jamming of wireless communication networks by building of simplicial complex models is studied using the relative homology in [Rob14a].

Our proposed method follows the worlflow of computing persistent homology classes described in 3.1. This requires imposing an algebraic structure on a constructed topological complex called cubical complex and described in 2.3.1 on page 15.

As we work with 2D+t grayscale images, we need to model our construction on a 3D based concept since we need the temporal neighbors of each voxel. We will work on a cubical complex that we will abusively call it the *voxel*' complex, where cells are attributed according to the combinatorial representation described in 3.3.1.

The input data of the image sequence is viewed as a function from a domain $D \subset \mathbb{N}^3$ into the space of real numbers $\mathbb{R}$ i.e. $f : D \to \mathbb{R}$ where $D = \{(x, y, t)|0 \leq x < \text{width}, 0 \leq y < \text{height}, 0 \leq z < t_f\}$, where t_f represents the number of 2D frames in the sequence.

For 2D+t case, the chain complex is:

$$\emptyset \to C_3 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \emptyset. \tag{3.18}$$

Highlighting the basis elements of H_2 allows to detect interesting objects in the image.

On another side, sometimes it's useful not to take into consideration a part A of the space X to compute the homology groups. We talk about another version of homology

involving the dump of a subcomplex A that yields to the relative homology $H_p(X, A)$. For this purpose, we will note $C_p(X, A) = C_p(X)/C_p(A)$ as the quotient chain complex and $\overline{\partial}_p$ as the boundary operator. More explanations on relative homology groups, cycles and boundaries were presented in 2.3.4.2 on page 28.

Taking chains on X modulo chains on A shrinks the requirement of a chain to be called a cycle, namely whenever its boundary is contained in A. This is illustrated in figure 3.37, where A represents the subspace containing all the cells that belong to $t = 0$ and t_f. As we see, the border of red cylinder that belong to A are collapsed to a point since we work on the quotient groups $C_p(X)/C_p(A)$. This concept allows to detect objects in all frames of the image like the "worm hole" illustrated by the red cylinder. This also includes the case when the boundary is empty, as the object in green, which can be detected also by absolute homology.

Relative persistent homology comes from the ideas of filtration and the functionality of relative homology described above. Following this manner, we will note K_i as the voxels' complex that contains cells whose values don't exceed the integer i. Hence we get a filtration of K in an analogue manner to equation (3.11) as follows:

$$\emptyset \subset K_0 \subset K_i \ldots \subset K. \tag{3.19}$$

By tracking the topological evolution of this filtration using relative homology, we get a sequence of homology groups connected by linear maps induced by inclusions for any dimension p:

$$\begin{aligned} H_p(K_0, A) \to H_p(K_1, A) \to \ldots \to \\ H_p(K_i, A) \to \ldots \to H_p(K, A). \end{aligned} \tag{3.20}$$

Note that these relations can be restricted to absolute homology by considering A as an empty set. Highlighting the most persistent basis elements of H_2 allows to detect the interesting objects in the image sequence since we work on 2D+t case.

3.6.2 Synthetic image

First, we have built a 2D+t grayscale synthetic image of size $15 \times 32 \times 32$ to prove the efficacy of our method. This image carries two "worms' holes" constructed by moving rectangles that pivot over all the sequence of the image and change in size. Two spheres that don't appear on the first and last frames are also present.

As mentioned in previously, after engaging the filtration procedure and the computation of relative persistent homology procedure in the process, we are able to compute homology classes and their basis in dimensions 0, 1 and 2. We are interested in H_2 elements specifically. These elements consist of equivalence classes of 2-cycles of chains that are sums of 2-squares and that are not boundaries of any 3-cell or cube. We color the pixels that form these squares with different colors for relative and absolute persistent homology.

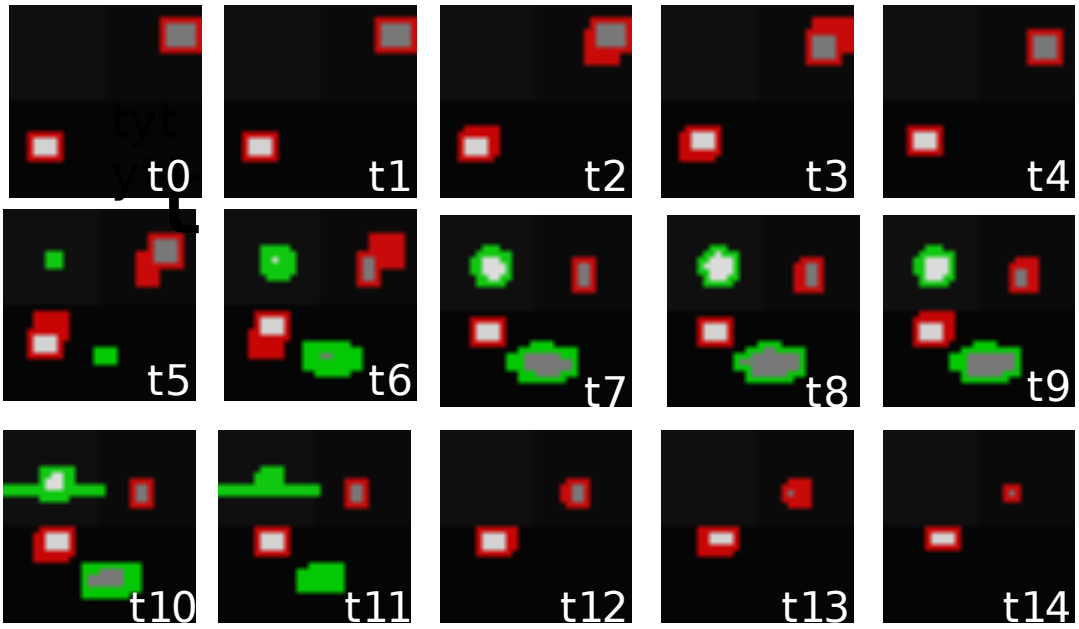


Figure 3.38: Results of the proposed relative persistent homology method on a synthetic 2D+t image of 15 frames.

The figure 3.38 illustrates how persistent homology in its relative version is capable without the need of prior parameters to capture interesting objects in first and last frames of the sequence and to follow their temporal evolution. In this figure, we show track detection results on a grayscaled toy image sequence. The red color represents the persistent classes computed by relative persistent homology while the green by both absolute and relative homology. While methods like frame differencing and background subtraction rely on parameters like pixels' intensity or objects sizes, this method don't depend on a prior parameter to be executed. It can be applied directly on the input image.

Taking A as the subcomplex containing the cells that belong to these frames and computing the corresponding relative homology will ensure the detection of boundaryless 2-chains and those with boundaries lying in A. This allows to spot objects movements in image sequences and to identify them from its start to its end. Pixels that belong to the squares that form the relative homology classes are colored in red, while green ones represent those able to be detected by the 2 kinds of homology.

3.6.3 Real applications and comparisons

As a real application, we consider two biomedical images taken by a time-lapse technique using the SID4Bio quantitative phase imaging system introduced in [BMW09]. The objective here is to detect the vesicles that move from first to last frames.

The rate of importance of computed homology classes is proportional to its surviving time.

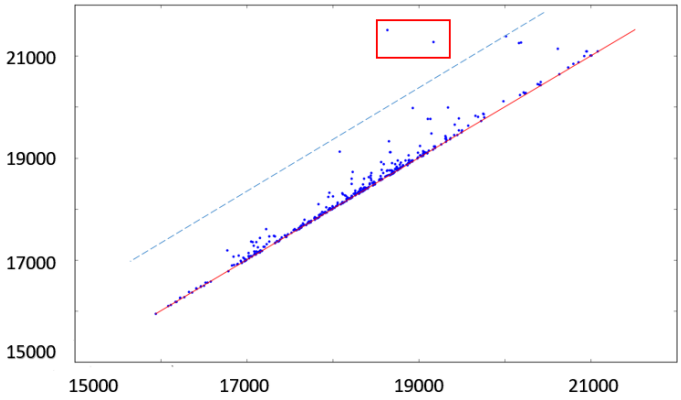


Figure 3.39: Persistence diagram of H_2 elements of the 2D+t sequence shown in figure 3.40.

As an advantage, the persistence diagram can be very efficient on the choice of interesting persistent classes. Points far from the diagonal are interesting while those lying along the diagonal are considered as a topological noise.

Following this manner, we chose the two classes above the graded line in figure 3.39 that represents the persistent diagram of the image in figure 3.40 of size $12 \times 77 \times 88$ encoding their times of births and deaths. Highlighting these two elements will detect the objects of the image shown in Fig. 3.40. Using the relative persistent homology object detection method, we are able to identify a moving vesicle (top) and the small train of vesicles (bottom) that increases in size and moves from the first frame in the sequence to the last one, as shown on figure 3.40. Also, other results shown on figure 3.41 on a 2D+t image of size $12 \times 50 \times 70$ prove the potential of this method. The two moving vesicles are detected by our tool from the first frame until the last one.

Contrary to the most existing methods, this technique doesn't depend on a specific parameter or attribute like the size of the vesicles, their form or intensity values, etc.

Concerning the other tracking techniques in the state of the art, the method described in [SK05] for example depends on a feature point tracking algorithm for automated detection of particles. This algorithm is based on the choice of three parameters that change in each frame: the approximate radius of the particles, the score cut-off for the non-particle discrimination, and the percentile that determines which bright pixels are accepted as particles.

The success of this method that relies clearly on the appropriate choice of the parameters is very limited on the case of 2D+t sequence shown in figure 3.40 since the sizes of the two objects are very different and the difference of luminosity with their background is very slight. To overcome the problem of the choice of the appropriate algorithms and parameters, the authors in [TPS+17] propose a tool named "TrackMate" that may offer a versatile solution for the large variety of the characteristics and needs of the image sequence. The benefit of "TrackMate" is strengthened through its ability to be customized

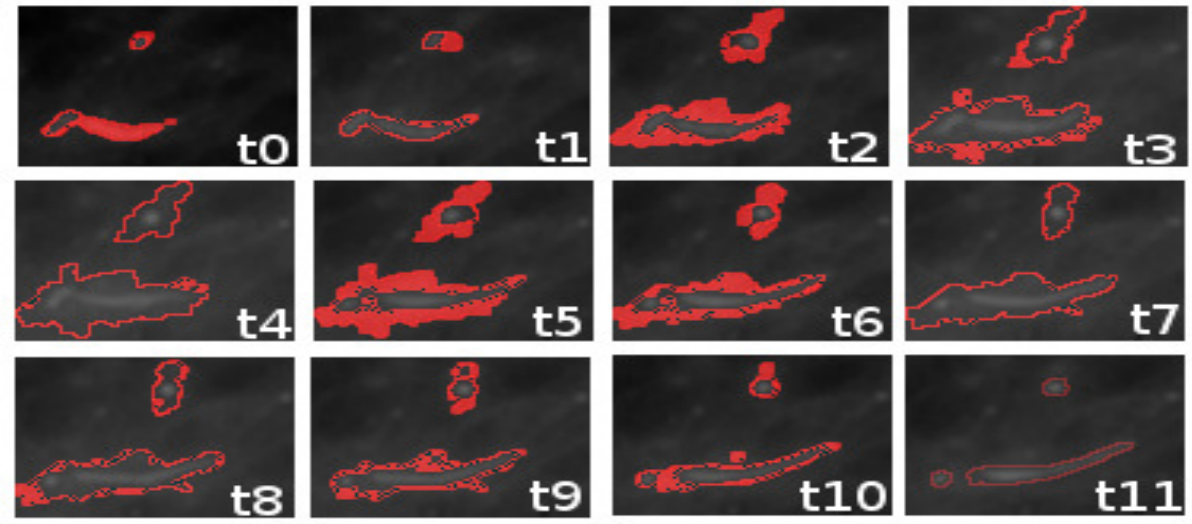


Figure 3.40: Results of relative persistent homology proposed method on a real grayscaled 2D+t biomedical image.

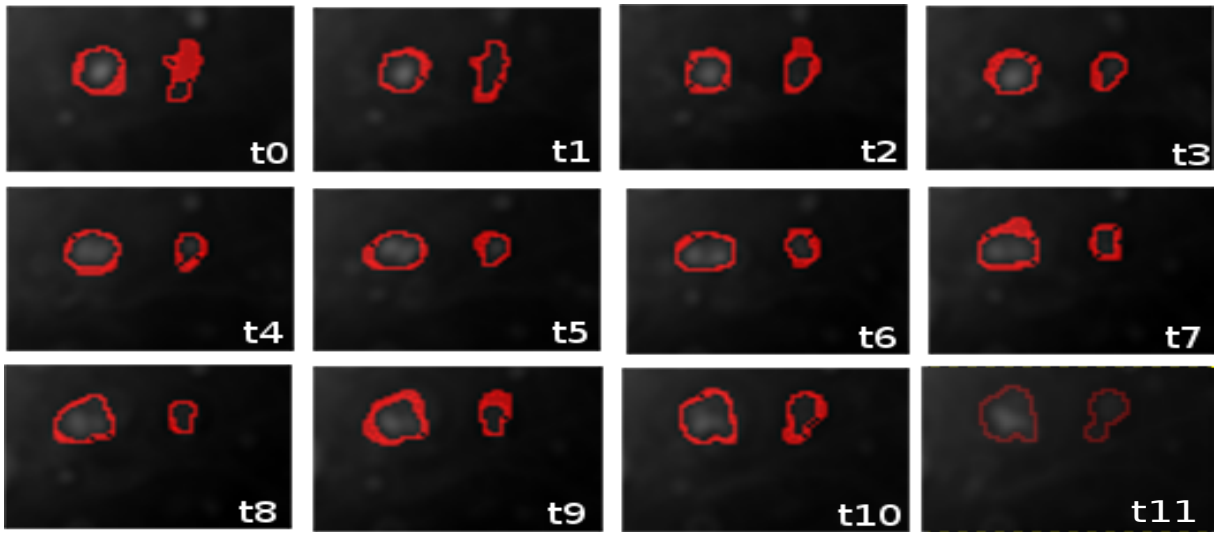


Figure 3.41: Results obtained on a real grayscaled 2D+t biomedical image using the relative persistent homology showing two moving cells.

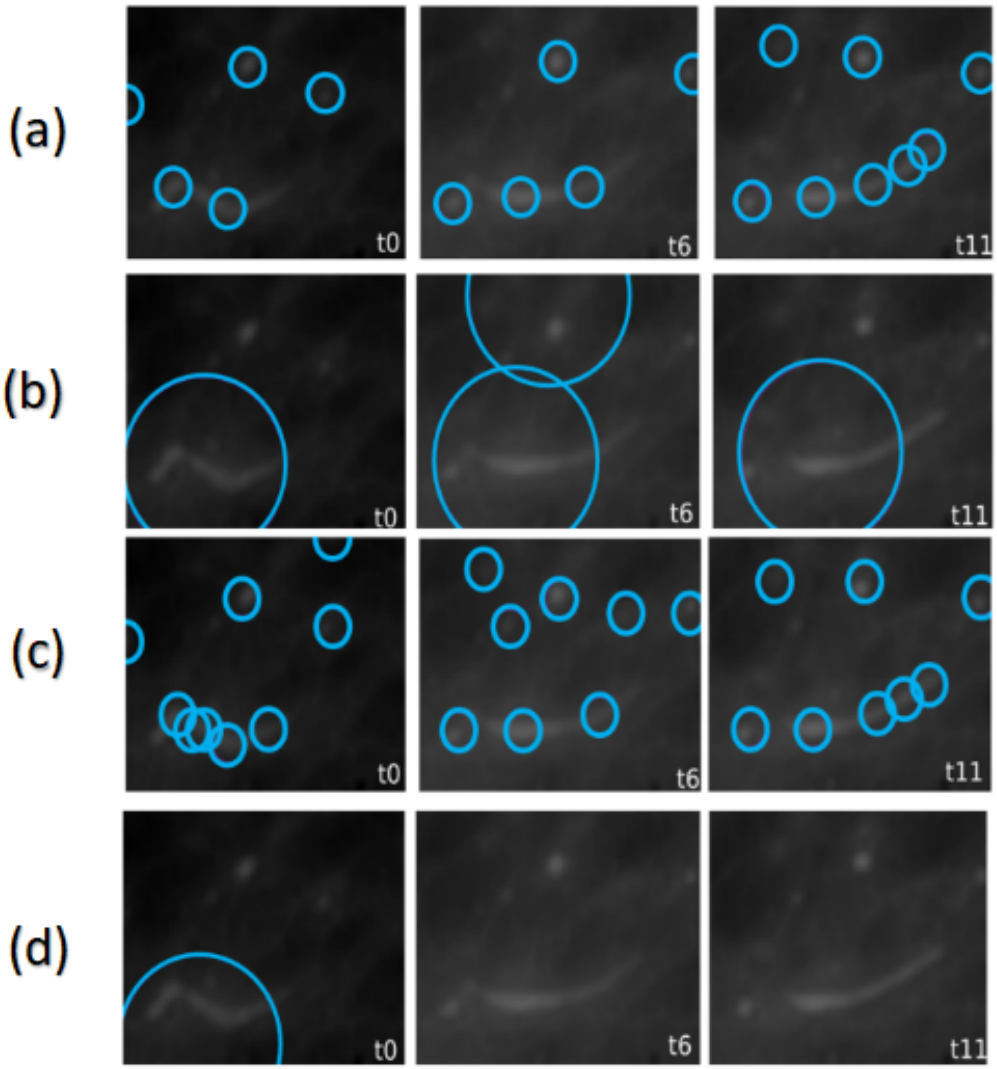


Figure 3.42: Results of the state of the art "TrackMate" tool on the 2D+t sequence shown in figure 3.40.

to meet specific tracking problems. The user may change the algorithms and parameters of the technique steps to find the suitable ones for its applications.

The results obtained by using this tool on the image in figure 3.40 are shown in figure 3.42. This results was obtained using four bundles (a), (b), (c), and (d) of parameters and algorithms on the image sequence. In the first step of the use of the tool, we choose for all the four bundles the standard parameters for the spatial and temporal calibration of the data that design the pixels' width, height and depths. For the second step we must pick the detection algorithm.

The Laplacian of Gaussian segmentation (LoG) detector applies a plain LoG segmentation on the image. All calculations are made in the Fourier space which makes it optimal for intermediate spot sizes. The Downsample LoG detector (DoG) uses the LoG detector, but decreases the size of the image by an integer factor before filtering. This makes it optimal for large spot sizes. We used the LoG detector for (a) and (b) and the DoG detector for (c) and (d). For the third step, we picked 10 pixels for (a) and (c) and 50 pixels for (b) and (d) for the radius of the detected particles. For the last step a quality feature is calculated to the candidate features of the image. We selected an automatic threshold as a value for this quality feature for the four bundles.

As shown in the figure 3.42 (a), using the LoG algorithm, the tool is capable to detect the small vesicle at the top while many inaccurate objects are detected instead of detecting the whole train of vesicles. By increasing the radius of the spots from 10 to 50, the tool detects the vesicle of train as shown in figure 3.42 (b) while it cannot detect the small vesicle in all frames of the image and specially in the first and the last ones.

Changing the detection algorithm to DoG instead of LoG in figure 3.42 (c) didn't allow to detect the train of vesicles. If we increase the radius from 10 to 50 in figure 3.42 (d), the tool detected the train of vesicles in the first frame and failed to track its movement afterward, as well as it didn't detect the small vesicle at the top.

The dynamic processes of moving objects need their own specialized tracking techniques that make use of specific aspects of the objects such as their shapes, sizes and luminosity etc. These facts make the issue of expanding a general and non-complex tracking tool an exigent and challenging task. Consequently, one approach that works well for a given problem is expected to fail for another one. In this context, our approach proposes a new generic and free parameter tool for object detection and tracking as shown the comparison of results in figure 3.40(our method) and figure 3.42 (state of the art method).

3.7 Conclusion

This chapter started by explaining the concept of persistent homology and how it differs from homology of a static topological space, starting from transforming the set of points into cell complexes until the persistent homology classes computation. Then, a state of

the art study of the persistent homology applications in engineering and image processing was made in section 3.2.

In section 3.3 we proposed a method to build the cell complex and its associated filtration scheme on the set of the pixels and superpixels of the image. Then, we initiated the applications of persistent homology on grayscale images in section 3.4 by introducing a novel method to image segmentation using the lifespans of homology classes. In section 3.5, we profited from the algebraic outcome of homology classes to segment salient objects in 2D and 3D images using topological construction on pixels and superpixels. Unlike the methods that rely their construction on the presence of prior parameters like objects sizes or intensities, our methods don't need this requirement. Moreover, while the existed methods are applied on images of specific dimension, we profit from homology to accomplish segmentation to support multidimensional tasks. Then an augmented version of homology in its absolute form, which is the relative homology is used in our applications in section 3.6. We have taken advantage from relative persistent homology to detect tracks of moving cells in time lapse images from the first until the last frame of the image. The power of this method is present also in its independence to prior parameters and the generability that characterizes its construction. Noting that our methods have proven their capability to solve many well known issues in image processing like the background/foreground discrimination, the overlaid objects, the multidimensional efficacy, the prior parameters etc. In chapter 4, we will see some techniques in deriving inferences from images using the sheaves theory, trying to detect global aspects of the image from local information and we see how to use interpretation of sheaves cohomology whether to scale analysis or localization.

Chapter 4

Sheaf theory and its application in image processing

4.1 Introduction to sheaf theory . . . 102
4.2 Sheaves of topological spaces . . . 103
4.3 Cellular sheaves . . . 104
4.3.1 Sheaf of vector spaces . . . 105
4.3.2 Local, global and pseudo sections . . . 106
4.3.3 Categorification . . . 108
4.3.4 Cellular sheaf cohomology and interpretation . . . 111
4.4 Sheaves on partial orders . . . 113
4.4.1 Definitions of sheaves over posets . . . 113
4.4.2 Cohomological analaysis of sheaves over posets . . . 116
4.5 Applications of sheaf theory . . . 118
4.6 Proposed applications of sheaf theory on images . . . 119
4.6.1 Application of cellular sheaves on images . . . 119
4.6.2 Sections on RGB images . . . 124
4.6.3 Interpretation of sheaves of models . . . 128
4.7 Conclusion . . . 143

In the chapter 2, we had focused on general notions of topology and algebraic topology and their applications explaining types of cell complexes and homology groups computation. In the chapter 3, we shifted to another kind of homology, more appropriate to image processing, which includes the concept of persistence. We have proposed methodologies that realize image processing tasks like image segmentation, object segmentation and object tracking. In this chapter, we extend another theory issued from the algebraic

topology. This concept that has contributed in the last five years in many engineering applications that depend on data fusion and translating the local information to global aspects. The sheaf theory founded by J. Leray [Mil00] is an abstract field of algebraic topology theory that mainly concern topologists in its basic aspects [Ser55, Swa64] or in its more modern forms [Bre97] because of its relations with the study of topological spaces and open sets. However, it can be a driving force in the world of data analysis and engineering.

In section 4.1, we represent how the sheaf theory can be used to analyze and extract general inferences from spaces of data. Then we start by defining sheaves over topological spaces in section 4.2. In section 4.3 we develop the notions of cellular sheaves explaining the concepts of sections. This section ends with computation of sheaves cohomology and its interpretation. In section 4.4, we explain the construction of sheaves over partially ordered sets, and the explanation of cohomological analysis and its computation. Then we make a look on the state of art applications of sheaves in section 4.5. Our initiation to the applications of sheaves on images is elaborated in the section 4.6, we propose firstly an application using sheaves on Čech complexes, then we explain how the sections of sheaves can serve in analyzing color images. This section ends with proposing techniques of scale analysis and localization using cohomology of sheaves. The section 4.7 concludes this chapter.

4.1 Introduction to sheaf theory

This section introduces the purpose and the goals of using sheaf theory in analyzing data and their inferences. The sheaf theory was introduced in the mid 1940s as a part of algebraic topology to arrange the collation of local data on topological spaces, note this old theory is built on general topological spaces. This theory is now essential in modern mathematics because of its richness through the relation between topology and algebraic geometry. The generalization in manipulating local to global conversions permitted applications to the science and engineering fields. For example, in the field of scientific data analysis, integration of heterogeneous systems [Hen14] is a popular problem that needs strong and robust theories to deal with. Whence the necessity of a mathematical framework for heterogeneous integration that should be general enough to correctly illustrate the data in all its richness and be efficient in summarizing data into its essential and meaningful features.

In this context, the use of a mathematical construction, that takes into consideration local properties and transfer them to global inferences but, at the same time, not theoretically complex, is essential.

The suitable structure to do that is that of a sheaf. Robert Ghrist and its students have taken sheaves and other sophisticated constructs from pure mathematics and applied them to practical problems such as network coding [GH11], electronics [Rob12], signal processing [CGR12, Rob14a], sampling [Rob15], etc.

Despite the fact that it's regarded as an abstract concept, a sheaf can be simply regarded as a technique to appoint several kinds of data that may be categorified into sets to each part of a topological space and to inspect the consistency of this data between neighbors in this space. Basically, sheaves, in its applied version represents, according to one of its initiators Michael Robinson [Rob17b], the correct way of data construction that stocks the local data and the appropriate summarization of a topological model by cohomological summaries. Moreover, sheaves cover the fact that the consistent mutuality of the information over two overlapping regions results in the validity of the information over the union of those two regions, and giving it the capability to globalize the studied data.

In addition, sheaf theory ensures computational methods that follow the general structure and has been occasionally emerged in applications [GH11, JHR14, Rob16]. The combinatorial progress [She85, Cur13] have make possible the manipulation of data structures into sheaves point of view.

Thus, we think that the sheaves theory can integrate information in images from a local perspective to a global version. For that, we pass through the process of encoding existing data into sheaves called **Sheafification** in order to obtain capabilities enabled by sheaves which are sections and sheaves cohomology. The proposed steps of the use of sheaf theory and its invariants are:

1. Designing the base topological space and the multiway interactions between data sources.
2. Sheafify: building the model of relations between data sources.
3. Categorify: placing the data streams in vector spaces to aid in computation and analysis.
4. Compute cohomology: globalizing the data to find robust invariants.

4.2 Sheaves of topological spaces

We define here the sheaves over topological spaces. Considering that X represents a topological space and R a commutative ring.

Preasheaf A Preasheaf $\mathcal{P}$ on X is built according to the following construction:

- For each open subset $U \subset X$, we assign an R-module $\mathcal{P}(U)$
- For each pair $V \subset U \subset X$, we assign an R-module linear map $f_{VU} : \mathcal{P}(U) \to \mathcal{P}(V)$
- $f_{UU} = id_U$ and $f_{WV} \circ f_{VU} = f_{WU}$ for all $W \subset V \subset U$, with id_U as the identity map: $\mathcal{P}(U) \to \mathcal{P}(V)$ and $f_{WV} \circ f_{VU} = \mathcal{P}(U) \to \mathcal{P}(V) \to \mathcal{P}(W)$.

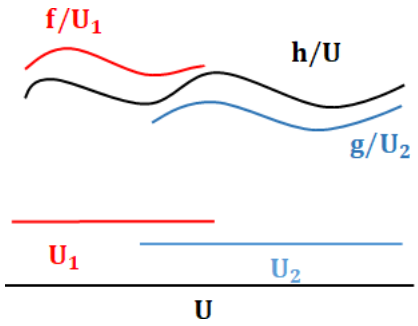


Figure 4.1: Representation of sheaves of continuous functions.

Elements s of $\mathcal{P}(U)$ are called sections on $\mathcal{P}(U)$ and $f_{VU}(s)$ is called the restriction of s from U to V. We will explain the concept of sections in their local and global aspects in detail in the next section.

Sheaf A presheaf $\mathcal{P}$ on X is a sheaf when it satisfies the following:

- For all open sets $U \subset X$ and any open covering $U = \bigcup_{i \in I} U_i$, and any section $s \in \mathcal{P}(U)$, if $f_{U_i U}(s) = 0$ then $s = 0$ with I an index set.
- For all open sets $U \subset X$, and any open covering $U = \bigcup_{i \in I} U_i$, for all families $s_i \in \mathcal{P}(U_i)$ that satisfy $f_{(U_i \cap U_j)U}(s_i) = f_{(U_i \cap U_j)U}(s_j)$ for all pairs (i, j), it exists $s \in \mathcal{P}(U)$ such that $f_{U_i U}(s) = s_i$ for all $i \in I$.

A well known example of sheaves is the sheaves of continuous functions, where restriction maps are function restrictions. In figure 4.1, the second condition expresses that having an open set $U \subset X$ and the open covering $U = U_1 \cup U_2$, and the function f and g defined over U_1 and U_2 that satisfy $f|_{U_1 \cap U_2} = g|_{U_1 \cap U_2}$ then it exists a common function h over their union $U = U_1 \cup U_2$. This is equivalent to say that whenever two continuous functions with overlapping domains are equal on the overlap, then they extend to a common continuous function over the union.

The construction of sheaves over topological spaces is a generalization of their structure on cell complexes or partially ordered sets since these spaces can be topologized to fit the definitions of sheaves over topological spaces as we will see in extend below.

4.3 Cellular sheaves

The topological version of sheaves described in section 4.2 is the source of a more combinatorial version that uses vector spaces in sections [She85, Cur13] that allows the transformation of the topological invariant problems to linear algebra.

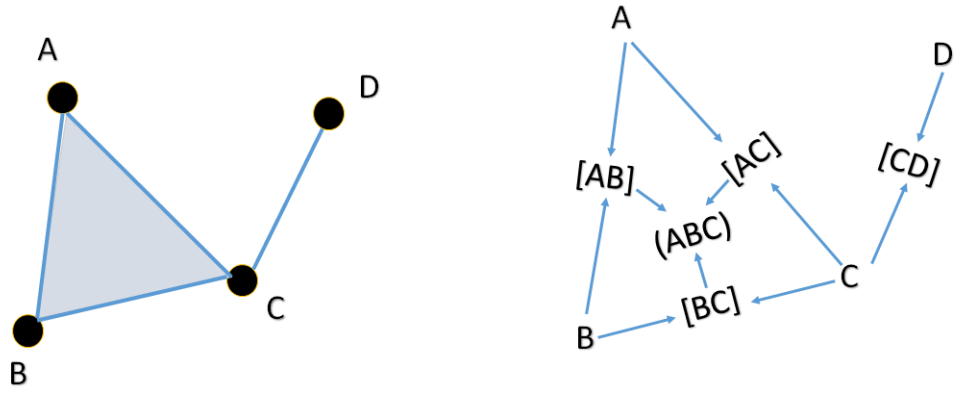


Figure 4.2: Topological base space for sheaf construction.

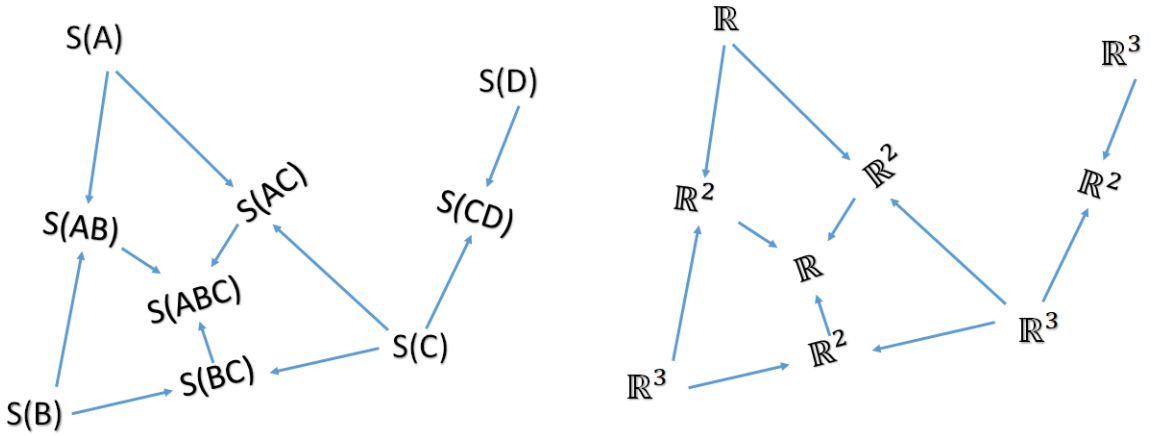


Figure 4.3: Stalks assigning to the cell complex.

4.3.1 Sheaf of vector spaces

A sheaf $\mathcal{S}$ of vector spaces or simply a sheaf on a cell complex X as the one shown in 4.2 is considered by [She85] as the base space that corresponds with the assignment of

1. A vector space $\mathcal{S}(\sigma)$ to each cell of X called the **stalk**. For example, in figure 4.3 the stalks assigned to A and B are $\mathbb{R}$ and $\mathbb{R}^3$ respectively.

2. A linear map $\mathcal{S}(\sigma \rightsquigarrow \tau) : \mathcal{S}(\sigma) \to \mathcal{S}(\tau)$ that is called **restriction** along $\sigma \rightsquigarrow \tau$ whenever σ is a face of higher dimensional cell τ, $\sigma \subset \tau$ and such that the restriction of σ to itself is the identity map. If σ is a face of τ $(\sigma \rightsquigarrow \tau)$ and τ is a face of ω $(\tau \rightsquigarrow \omega)$ then, $\mathcal{S}(\tau \rightsquigarrow \omega) \circ \mathcal{S}(\sigma \rightsquigarrow \tau) = \mathcal{S}(\sigma \rightsquigarrow \omega)$. For example, in figure 4.4 $\mathcal{S}(B \rightsquigarrow [AB]) = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ which is 2×3 matrix since it's a restriction from $\mathbb{R}^3$ to $\mathbb{R}^2$. Note that in figure 4.4 we assign restrictions that respect these conditions and that serve our continuous example in this section.

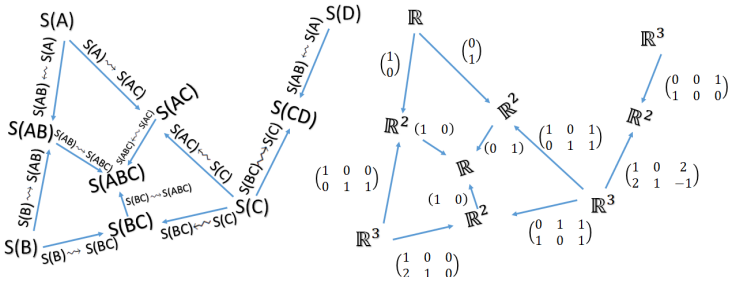


Figure 4.4: Restrictions assigned to each cell inclusion.

Then restrictions go from low dimensional cells to higher dimensional ones. When maps go from higher dimensional cells to lower ones, we call them **extensions** and we have a **cosheaf**. We call the built specification a sheaf, but many others call it a presheaf however every presheaf can be uniquely realized as a sheaf.

Each edge in this example is consistent so this is globalized. If no global consistency check, there can be conflicting information.

4.3.2 Local, global and pseudo sections

A **local section** of a sheaf is an element s in $\bigoplus_{\sigma \text{ is a cell}} \mathcal{S}(\sigma)$ which is the direct sum of the stalks in the base space and σ belongs to a subspace Y of X, i.e. $\sigma \in Y \subset X$. The term **local section** is defined only locally over a subspace of the space X. This element must satisfy the consistency relation $\mathcal{S}(\sigma \rightsquigarrow \tau)(s(\sigma)) = s(\tau)$ for all $\sigma \subset \tau$ for all $\sigma \in \tau$ in the subspace Y of X, where $\mathcal{S}(\sigma \rightsquigarrow \tau)$ is a linear restriction map, $s(\sigma)$ is an element of $\mathcal{S}(\sigma)$, and $s(\tau)$ is an element of $\mathcal{S}(\tau)$.

A **global section** is specifically an assignment of values from each of the stalks that is consistent with the restrictions as shown in figure 4.5. When we say consistent we mean that information coming from vertices to their common edge must be the same. Formally, the global section on an edge e satisfies this relation:

$$\mathcal{S}(v_1 \rightsquigarrow e)s(v_1) = \mathcal{S}(v_2 \rightsquigarrow e)s(v_2), \tag{4.1}$$

where v_1 and v_2 are the vertices of the edge e. For example, taking the section $\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ at vertex C in figure 4.5 who is transformed by the restriction $\mathcal{S}(C \rightsquigarrow [CD]) = \begin{bmatrix} 1 & 0 & 2 \\ 2 & 1 & -1 \end{bmatrix}$ to $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$ and the section $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ at vertex D who is transformed by the restriction $\mathcal{S}(D \rightsquigarrow$

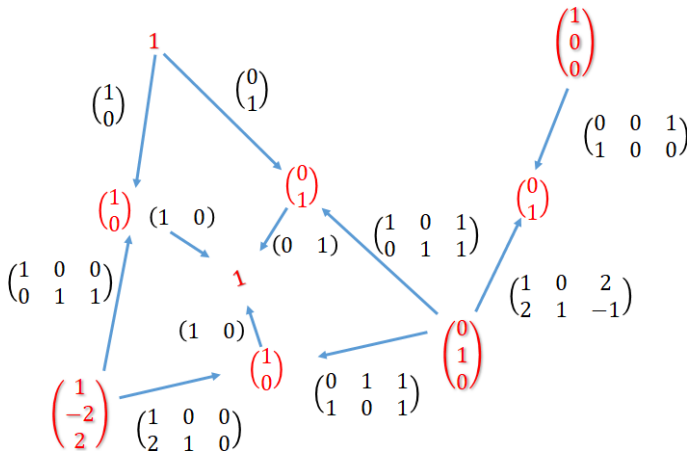


Figure 4.5: Sections consistent with restrictions.

$[CD]) = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ to $\begin{bmatrix} 0 \\ 1 \end{bmatrix}$, we remark that $\mathcal{S}(C \rightsquigarrow [CD])s(C) = \mathcal{S}(D \rightsquigarrow [CD])s(D)$, then we have a global section that is extended everywhere this relation is satisfied.

Noting that the process of sheaf construction is done after assigning the stalks and restrictions going up in cells after the respect of the local sections criteria.

Sections of sheaves are fundamental element in our work because they permit to globalize consistently information across data sources, but they seem to require exact matches between data sources, which is undesirable especially in noisy data. For this, what if instead we want consistency checks that are approximate, to a certain tolerance?

For example, consider that we are measuring the temperature of a particular object and we have two independent measurements, we can build a sheaf in this way:

$$Sheaf: \quad \mathbb{R} \xrightarrow{id} \mathbb{R} \xleftarrow{id} \mathbb{R}. \tag{4.2}$$

If these two measurements say that the temperature of the object is $100°\mathbf{C}$ and the object itself is $100°\mathbf{C}$ then the associated data is:

$$Data: \quad 100°\mathbf{C} \rightarrow 100°\mathbf{C} \leftarrow 100°\mathbf{C}. \tag{4.3}$$

So we do not have a section, since the consistency is available and we have identity maps.

But what if the first measurement is 99°C and the second is 101°C and the actual object is 100°C in this way:

$$Data: \quad 99°C \rightarrow 100°C \leftarrow 101°C. \tag{4.4}$$

So we do not a section, since the consistency check is false. But if we take the variance of the pieces of data coming together to the edge, which is $variance\{99°C, 100°C, 101°C\}$ and check if it's less than 10% for example, we will consider that we have a **pseudosection**. Like this we're accepting an error tolerance and consider that these measurements are self consistent. Every function that takes these three values and returns a truth

value, true when they are consistent and false when they are not will express the presences of a **pseudosection** like the fact that $variance\{99°C, 100°C, 101°C\} < 10\%$ or $\{99, 100, 101\} \in [90, 110]$ for example.

Relaxing matching requirements for sections gives birth to the notion of **pseudosections**. The basic modeling idea is to identify for a non vertex cell which data is needed to check for consistency at that cell. For an edge, we need to check for consistency of data coming from the two vertices and already we have data on that edge. So at every cell, we can have a function that checks for consistency and gives true and false values. The true value means the presence of a pseudosection.

Thus, existence of pseudosections require consistency structures. A consistency structure reposes on a triple $(X; \mathcal{S}; \mathbf{C})$ where X is a cell complex and $\mathcal{S}$ is a sheaf over X. $\mathbf{C}$ is the assignment to the non vertex cell $a \in X$ of dimension k of a function:

$$\mathbf{C}_a = (\mathcal{S}(a))^{2+\dim a} \to \{0, 1\}. \tag{4.5}$$

So the consistency structure C_a returns 1 whenever the data at non vertex cell a are consistent.

A $(X; \mathcal{S}; \mathbf{C})$-**pseudosection** p on a non vertex cell a and that belongs to $\bigoplus \mathcal{S}(a)$ satisfies everywhere it's defined:

$$\mathbf{C}_a(p(a), (\mathcal{S}(v_0 \rightsquigarrow a), \ldots, \mathcal{S}(v_k \rightsquigarrow a)) = 1, \tag{4.6}$$

where v_i are faces of a.

By [Rob15], sheaves can be built by pseudosections with preserving the same purpose of its constructions, which is the transformation from the local to the global information. A sheaf built by pseudosection can be extended in a way that its pseudosections are sections in the classical concept of sections of sheaves. And also, like local sections, we must have consistent pseudosections all over the complex in order to have global section.

4.3.3 Categorification

Sheaves can support faithful models information integration problems. Indeed they are canonical, but sometimes they can become too complicated to be useful and to be represented by vector spaces. In the general case stalks are sets so the manipulation of these sets and assigning restrictions will cause problems contrary to the specific case where stalks are vectors so they are easy to use linear algebra tools.

If we want to derive actionable, relevant summaries of sheaves, they must be sheaf invariants and computationally tractable. The problem here is how to encode data for convenient computation. This is where category theory can interfere to transfer sets to vector spaces able to be manipulable algorithmically.

4.3.3.1 Introduction

We will develop this idea refereeing to the tutorials of Michael Robinson [Rob18] on the sheaves theory applications. Consider any function between sets $f : A \to B$ and let $\mathbb{R}(A)$ be the vector space generated by A. The basis of $\mathbb{R}(A)$ is the set of elements of A so the $\dim(\mathbb{R}(A)) = cardinality(A)$ and every element of $\mathbb{R}(A)$ is a linear combination of the elements of A (similarly for B) and f **lifts** uniquely to a linear map $\mathbb{R}f$:

$$\begin{array}{ccc} \mathbb{R}(A) & \xrightarrow{\mathbb{R}f} & \mathbb{R}(B) \\ {\scriptstyle (1\times)}\uparrow & & {\scriptstyle (1\times)}\uparrow \\ A & \xrightarrow{f} & B \end{array} \tag{4.7}$$

where maps $(1\times)$ interpret elements of A and B as elements of vector spaces. Taking an element of A and applying $(1\times)$, we will have a vector whose components are zero except for that element.

For example, consider that $A = \{cat, dog, bird\}$ and $B = \{mammal, not\ mammal\}$, then $f(cat) = mammal$, $f(dog) = mammal$ and $f(bird) = not\ mammal$. $\mathbb{R}(A) = \mathbb{R}^3$, $\mathbb{R}(B) = \mathbb{R}^2$ and $\mathbb{R}f : \mathbb{R}^3 \to \mathbb{R}^2$ is a 2×3 matrix:

$$\begin{array}{c} \\ mammal \\ not\ mammal \end{array} \begin{array}{c} \begin{array}{ccc} cat & dog & bird \end{array} \\ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}. \tag{4.8}$$

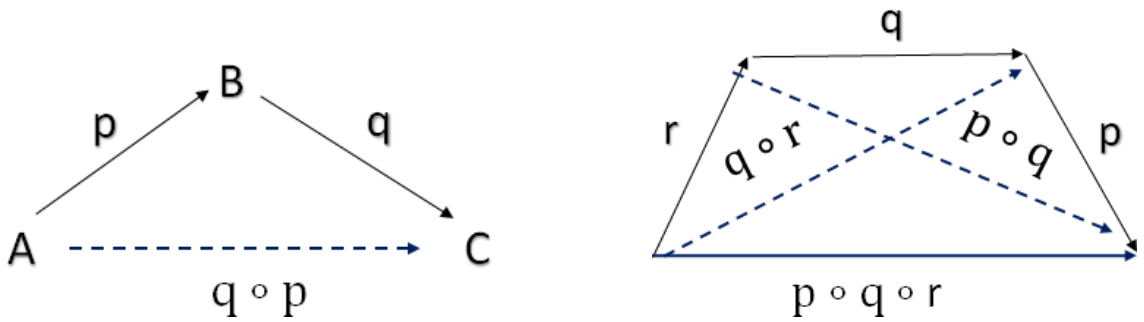
So taking the vector cat and applying the Rf we have:

$$Rf \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{4.9}$$

with $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$ a basis element for $\mathbb{R}(B)$. This kind of construction though allows to turn a set of valued function to a linear map.

Adding a quantitative data as the number of cats, dogs and birds this will give the number of mammal and not mammals. For example, for 3 cats, 2 dogs, and 1 bird, we will have:

$$Rf \begin{bmatrix} 3 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 7 \\ 2 \end{bmatrix}. \tag{4.10}$$



(a) Composition of two morphisms.

(b) Associativity of composition.

Figure 4.6: A composition of morphism and its associativity scheme.

In order to find $\mathbb{R}(A)$, $\mathbb{R}(B)$ and $\mathbb{R}f$, functions on sets are transformed into **functors** on categories. We will explain this notion extensively below. Before starting in with precision, we must note that there may be several possible categorifications for a given set. Choosing the best one is still an art and this process allows us to normalize a sheaf with many different data types into a sheaf with just vector data.

4.3.3.2 Changing types using functors

A category C consists of a class $Ob(C)$ of objects and a class $Mor(C)$ of morphisms for which:

- Each morphism m has a source and target object, usually written $m : A \to B$.
- Morphisms can be composed: if $p : A \to B$ and $q : B \to C$, then there is an unique morphism $q \circ p : A \to C$ called their composition, as shown in figure 4.6 (a).
- Composition is associative: $(p \circ q) \circ r = p \circ (q \circ r)$, as shown in figure 4.6 (b).
- There is an identity morphism 1_A for every object A for which $p \circ 1_A = p$ and $1_A \circ q = q$.

For example, taking a category of sets, we will have an object of **set** noted $Ob(set)$, which is a collection of sets and a morphisms of sets noted $Mor(set)$, which are functions that respect the previous axioms. Similiraly for category of vector spaces **vec**, where the $Ob(\mathbf{vec})$ are vector spaces and $Mor(\mathbf{vec})$ are linear maps between space of the same field $\mathbb{F}$.

Now, we can manipulate categories using functors. If C and D are categories, a covariant functor $F : C \to D$ assigns:

- An object $F(A)$ in D for each object A in C.
- A morphism $F(m) : F(A) \to F(B)$ in D for each morphism $m : A \to B$ in C so that composition is preserved $F(m \circ n) = F(m) \circ F(n)$.

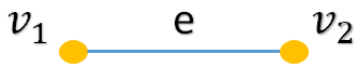


Figure 4.7: A simple base space for sheaf construction.

The lifting in equation (4.7) is a functor where the category C is the category of sets and category D is the category of vector spaces. Indeed, we can take a set of {cat, dog, bird} and turn it into an object of the other category D, which is the vector space $\mathbb{R}^3$. The morphism from {cat, dog, bird} to {mammal, not mammal} got turned into a linear map between $\mathbb{R}^3$ and $\mathbb{R}^2$. So functors helps in changing types of data in order to translate types to vector spaces or other algebraic structures depending on methods from representation theory.

4.3.4 Cellular sheaf cohomology and interpretation

Hubbard states in his appendix on sheaf theory [Hub06] "It is fairly easy to understand what a sheaf is, especially after looking at a few examples. Understanding what they are good for is rather harder; indeed, without cohomology theory, they are not good for much." If we want to derive actionable, relevant summaries of sheaves, these summaries are called sheaf invariants and should be computationally tractable. This is the case of sheaves cohomology as we will see in extend below.

Some faults or redundancy cannot be detected by the space of global sections. Another topological invariant may help in this issue. This invariant is born from the idea of rewriting the basic condition for a global section s of a sheaf $\mathcal{S}$.

In figure 4.7, we have a cell complex, formed by the two vertices v_1 and v_2 and the edge connecting them is e. The sheaf constructed on this complex is as follows:

$$\mathcal{S}(v_1) \xrightarrow{\mathcal{S}(v_1 \rightsquigarrow e)} \mathcal{S}(e) \xleftarrow{\mathcal{S}(v_2 \rightsquigarrow e)} \mathcal{S}(v_2). \tag{4.11}$$

The global section on this sheaf must satisfy: $\mathcal{S}(v_1 \rightsquigarrow e)s(v_1) = \mathcal{S}(v_2 \rightsquigarrow e)s(v_2)$ then $+\mathcal{S}(v_1 \rightsquigarrow e)s(v_1) - \mathcal{S}(v_2 \rightsquigarrow e)s(v_2) = 0$. This proposes that $(v_1 \rightsquigarrow e)$ and $(v_2 \rightsquigarrow e)$ should be assigned opposite signs during constructions and the concept of orientation makes this assignment precise. Putting the last equality in a vector spaces form we will have:

$$\begin{bmatrix} +\mathcal{S}(v_1 \rightsquigarrow e) & -\mathcal{S}(v_2 \rightsquigarrow e) \end{bmatrix} \begin{bmatrix} s(v_1) \\ s(v_2) \end{bmatrix} = 0. \tag{4.12}$$

This will help to reduce the question of finding the global section to finding the kernel of a matrix. This brings us back to the idea of homology that is a kernel of a

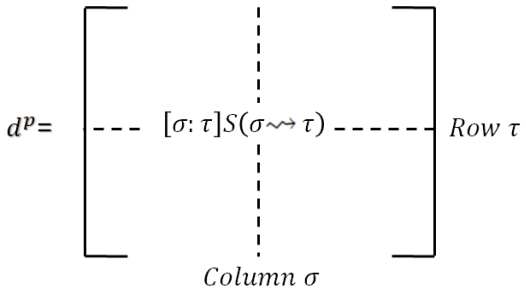


Figure 4.8: The coboundary that represents the coboundary map d^p

chain boundary modulo the image of the higher dimensional chain boundary, i.e. $H_p = \ker \partial_p(C_p)/\operatorname{Im} \partial_{p+1}(C_{P+1})$. We have seen that homology is dealing with transforming data from higher dimensional aspects to lower ones and this transformation is based on a chain complex as seen in (2.2) on page 24. For sheaves, we are dealing with the inverse sense transformation with data going down from higher dimensional cells to lower ones. This will give birth to the notion of **cochain complex** for a sheaf $\mathcal{S}$.

Supposing that $\mathcal{S}$ is a sheaf of abelian groups on a cell complex X. The p-th cochain group $C^p(X, \mathcal{S})$ of $\mathcal{S}$ is the direct sum of the stalks over the p-cells of X. They are represented by

$$C^p(X, \mathcal{S}) = \bigoplus_{\sigma \in X^p} \mathcal{S}(\sigma). \tag{4.13}$$

Keeping in mind that elements of $C^p(X, \mathcal{S})$ represent functions from the p-cells to the stalks over those cells. Similarly to chains, relations between cochains are given by coboundary maps. Coboundary maps work like discrete derivatives and compute differences between functions on higher dimensional cells. The p-th coboundary map d^p is the homomorphism $d^p : C^p(X, \mathcal{S}) \to C^{P+1}(X, \mathcal{S})$ given by:

$$(d^p f)(\tau) = \sum_{\sigma \in X^p} [\sigma : \tau]\mathcal{S}(\sigma \rightsquigarrow \tau) f(\sigma), \tag{4.14}$$

where $\tau \in X^{p+1}$, $f \in C^p(X, \mathcal{S})$, and $[\sigma : \tau]$ represents the orientation between σ and τ expressed by $+1$, -1 or 0. Here d^p acts on a particular element of C^p, knowing that a cochain of $C^p(X, \mathcal{S})$ is a vector and also a function. We want to ask what is the stalk associated to $C^{p+1}(X, \mathcal{S})$. Hence we're looking for the value of $d^p(f)$ on an element τ of $C^{P+1}(X, \mathcal{S})$.

We show in figure 4.8 the matrix that represents the coboundary map described in equation (4.14). The columns represent cells with dimension p while the rows are indexed by $(p+1)$-cells. Elements of the matrix that corresponds to column σ and row τ under the form $[\sigma : \tau]\mathcal{S}(\sigma \rightsquigarrow \tau)$ are filled by restrictions maps between σ and τ.

The set of vector spaces C^p and the boundary operator d^p between them are called a chain complex and is noted:

$$C^0(X,\mathcal{S}) \xrightarrow{d^0} C^1(X,\mathcal{S}) \xrightarrow{d^1} C^2(X,\mathcal{S}) \dots C^{p-1}(X,\mathcal{S}) \xrightarrow{d^{p-1}} C^p(X,\mathcal{S}). \tag{4.15}$$

Since in chain complexes the emptyness boundary of a boundary gave birth to homology groups, the coboundary of coboundary of cochains is also a void by [She85], hence $\operatorname{Im} d^{k-1} \subset \ker d^k$.

Cellular sheaf cohomology The facts cited above gives birth to the cohomology of cellular sheaves defined by:

$$H^p(X,\mathcal{S}) = \ker d^p / \operatorname{Im} d^{p-1}. \tag{4.16}$$

Hence, the sheaf cohomology group of dimension p represents cochains that exist in dimension p but were not already present in $p-1$. Noting that p here deals with the dimension of cells and not the stalks.

As interpretations of sheaf cohomology:

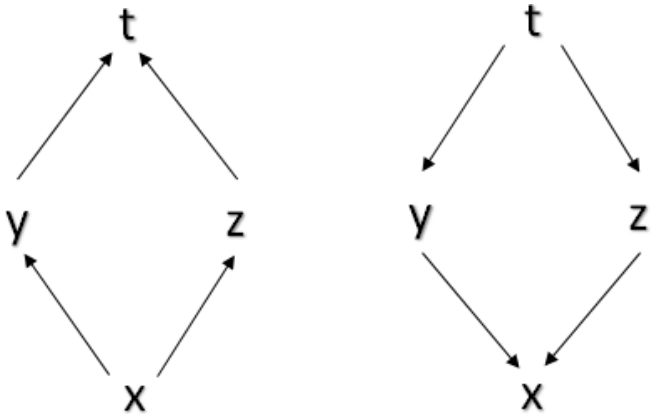
- The space of global sections of a sheaf $\mathcal{S}$ on a cell complex X is isomorphic to $H^0(X;\mathcal{S}) \simeq \ker d^0$.
- The $H^1(X;\mathcal{S})$ may represent the new sections that are not present as global sections when using only edges. Some references call it data loops or misinformation gaps. So it's a power invariant since it describes what happens when we don't have the full story.
- $H^p(X;\mathcal{S})$ represents information gaps of higher dimensions depending on the nature of application and problem.

4.4 Sheaves on partial orders

Other than constructions on cellular spaces, like simplicial complexes or others, sheaves can also be constructed on partial orders [Rob17a]. These partial orders are represented by a general relation that may include many aspects if they satisfy the conditions of the relation.

4.4.1 Definitions of sheaves over posets

A partial order on a set E is a relation $\leq$ built on that set and that respect the following characteristics:



(a) A poset E where arrows are pointed in increasing direction.

(b) A poset E where arrows are pointed in decreasing direction.

Figure 4.9: A poset and its dual.

1. Reflexivity: $x \leq x$ for all $x \in E$.
2. Antisymmetry: if $x \leq y$ and $y \leq x$ then $x = y$.
3. Transitivity: if $x \leq y$ and $y \leq z$ then $x \leq z$.

The pair $(E, \leq)$ is called a partially ordered set or a poset.

We will refer to this pair as $E = (E, \leq)$ when it's clear from the context. This partially ordered set has a dual partial order defined by the relation $\leq^{op}$ on E, where $x \leq^{op} y$ if and only if $y \leq x$. The partially order set defined by this duality $E^{op} = (E, \leq^{op})$ is called the dual poset to E. The figure 4.9 shows an example of a poset E and its dual.

Every topological space (X, T) can define a poset noted $Open(X, T) = (T, \subseteq)$ on its open sets that are partially ordered by the subset relation.

In a poset $(E, \leq)$, the collection of upper level sets of the form $O_x = \{y \in E \text{ such that } x \leq y\}$ for each $x \in E$ forms a base for a topology that is called Alexandroff topology [Ale37] as shown in the figure 4.10. Noting that every intersection of opens sets in the Alexandroff topology on a poset E is open.

The definition of the sheaf comes from the diagram of a poset, shown in 4.9 (a), where the vertices represent elements and arrows are pointing from lesser elements to greater ones. We will replace each vertex by a set or a space and each arrow by a function.

Sheaves of sets: A sheaf $\mathscr{S}$ of sets on the poset E with the Alexandroff topology consists of the following requirements:

1. Assigning a set $\mathscr{S}(x)$ for each $x \in E$ called the stalk at x as shown in figure 4.11 (a).

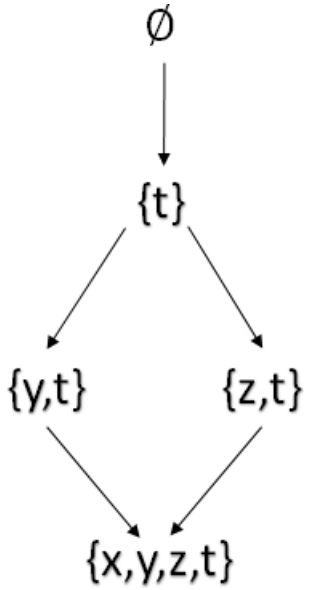


Figure 4.10: Alexandroff topology on E where arrows are inclusions.

2. Assigning a function $\mathscr{S}(x \leq y) : \mathscr{S}(x) \to \mathscr{S}(y)$ for each pair $x \leq y \in E$. This function is called a restriction as shown in figure 4.11 (a),

3. Verifying that $\mathscr{S}(x \leq z) = \mathscr{S}(y \leq z) \circ \mathscr{S}(x \leq y)$, for each triple $x \leq y \leq z \in E$.

Noting that when the stalks are vector spaces, we will obtain a sheaf of the type that preserve this structure which are linear functions in this case.

Similarly, the sheaf of sets $\mathscr{C}$ over the dual poset E^{op} with the Alexandroff topology shown in figure 4.9 (b) is expressed in an analogue way to the sheaf of poset E just reversely:

1. Assigning a set $\mathscr{C}(x)$ for each $x \in E^{op}$ called the stalk at x as shown in figure 4.11 (b),

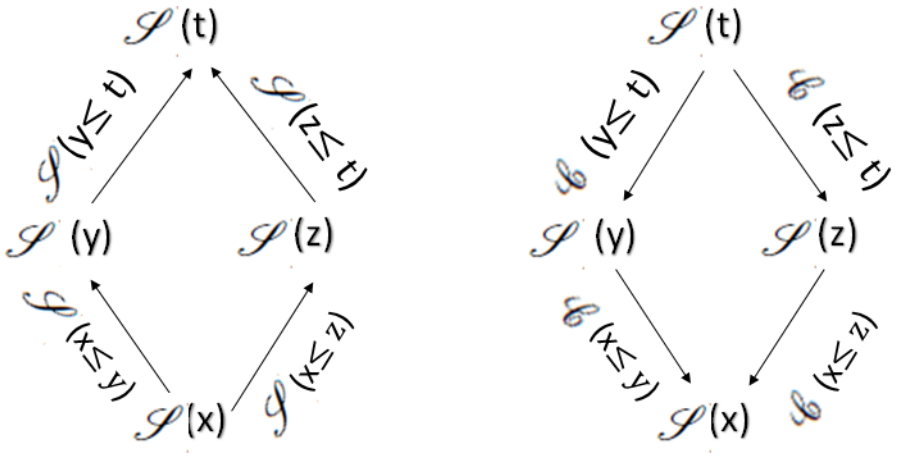
2. Assigning a function $\mathscr{C}(x \leq y) : \mathscr{C}(y) \to \mathscr{C}(x)$ for each pair $x \leq y \in E^{op}$ as shown in figure 4.11 (b). This function is called an extension,

3. Verifying that $\mathscr{C}(x \leq z) = \mathscr{C}(x \leq y) \circ \mathscr{C}(y \leq z)$ for each triple $x \leq y \leq z \in E^{op}$.

A global section of a sheaf $\mathscr{S}$ on a poset E consists in an element s of the direct product $\prod_{x \in E} \mathscr{S}(x)$ such that for all $x \leq y \in E$ we have $\mathscr{S}(x \leq y)(s(x)) = s(y)$. Here, the direct product is not the direct sum since E may be infinite. A local section is defined in the same way but on a subset $F \subset E$.

In a dual way, we may define a global section of a sheaf $\mathscr{C}$ on the dual poset E^{op} by an element t of the direct product $\prod_{x \in E} \mathscr{C}(x)$ such that for all $x \leq y \in E$ then $t(x) = \mathscr{C}(x \leq y)(t(y))$. A local section is defined in the same way but on a subset $F \subset E$.

For the categorified point of view, a poset E defines a category on which the objects are



(a) Stalks on the poset E and restrictions.

(b) Stalks on the dual poset of E and restrictions.

Figure 4.11: Stalks on a poset and its dual with restrictions.

elements of E and a unique morphism between two elements x and y of E exists if $x \leq y$. The proprieties of the comparison $\leq$ correspond to those of morphisms of categories. A sheaf on a poset E can be seen as a functor between the category E and the category *set* or *vec* depending on the type of data on stalks.

4.4.2 Cohomological analaysis of sheaves over posets

If a specific system is encoded as a sheaf, its analysis can be done using cohomological concepts. Sheaves built over posets with stalks represented by vector spaces and whose restrictions are linear, or extensions in the dual poset, have topological invariants that can be computed automatically. If the constructed sheaf is not linearized, then it is necessary to use the categorification to transform the data to linear spaces in order to be manipulated in matrices as described above. Noting that the computation of cohomology of sheaves on posets is not natural as much as cohomology of cellular sheaves, some complexity is behind the scene.

If $\mathscr{S}$ is a sheaf of vector spaces having linear restriction maps over a poset E, then the p-cochain space $C^p(\mathscr{S})$ of $\mathscr{S}$ is the direct product of stalks at the end of chains of length p:

$$C^p(E, \mathscr{S}) = \prod_{x_0 < \ldots < x_p} \mathscr{S}(x_p). \tag{4.17}$$

So $C^p(E, \mathscr{S}) = [\mathscr{S}(x_p)]^{n_p}$ where n_p is the number of chains of length p when the poset E is finite. Elements of C^p are represented by chains in E of length p, and can therefore be considered as a function s from the collection of chains in E. Noting that a chain of

length 1 is expressed by every two comparable elements like $x < y$ or $x < t$ and chains of length 2 are under the form $x < y < z$ or $x < z < t$.

The p-coboundary map $d^p : C^p(E, \mathscr{S}) \to C^{P+1}(E, \mathscr{S})$ is described by the formula:

$$\begin{aligned}(d^p s)(x_0 < \ldots < x_{p+1}) = \sum_{i=0}^{p} (-1)^i s(x_0 < \ldots \widehat{x_i} < \ldots < x_{p+1}) \\ +(-1)^{p+1} \mathscr{S}(x_p < x_{p+1})(s(x_0 < \ldots < x_p)).\end{aligned} \tag{4.18}$$

where $(x_0 < \ldots < x_{p+1}) \in C^{p+1}$, $s \in C^p(E, \mathscr{S})$ and $\widehat{x_i}$ means the omit of an element x_i. Noting that $\sum_{i=0}^{p} s(x_0 < \ldots < \widehat{x_i} < \ldots < x_{p+1})$ eliminates x_i from the cochain $x_0 < \ldots < x_i < \ldots < x_{p+1}$ from 0 to p and keeping x_{p+1} always. Also, d^p acts on a particular element of C^p, knowing that a cochain of $C^p(E, \mathscr{S})$ is a vector and also a function. Noting that we want to ask what is the stalk associated to $C^{p+1}(E, \mathscr{S})$. Hence we're looking for the value of $d^p(s)$ on an element $(x_0 < \ldots < x_{p+1})$ of $C^{p+1}(E, \mathscr{S})$.

By equation (4.18), $(d^0 s)(x_0 < x_1) = s(x_1) - \mathscr{S}(x_0 < x_1)s(x_0)$, d^0 compares $s(x_1)$ with $s(x_0)$ brought back on x_1 by $\mathscr{S}(x_0 < x_1)$. $(d^1 s)(x_0 < x_1 < x_2) = s(x_1 < x_2) - s(x_0 < x_2) + \mathscr{S}(x_1 < x_2)s(x_0 < x_1)$ and d^1 compares $s(x_0 < x_2)$ on this scheme $x_0 \longrightarrow x_1 \longrightarrow x_2$ with $s(x_1 < x_2) + \mathscr{S}(x_1 < x_2)s(x_0 < x_1)$.

We represent in the equation (4.19) a simple coboundary matrix for $(d^0 s)(x_0 < x_1)$ where cochains of length zero are on columns and those of length one on rows. The equation (4.20) represents $(d^1 s)(x_0 < x_1 < x_2)$ where chains of length one are on columns and those of length two are on rows.

$$(d^0 s)(x_0 < x_1) = \begin{array}{c} \\ x_0 < x_1 \end{array} \begin{array}{c} \begin{array}{cc} x_0 & x_1 \end{array} \\ \begin{bmatrix} -\mathscr{S}(x_0 < x_1) & I \end{bmatrix} \end{array}. \tag{4.19}$$

$$(d^1 s)(x_0 < x_1 < x_2) = \begin{array}{c} \\ x_0 < x_1 < x_2 \end{array} \begin{array}{c} \begin{array}{ccc} x_0 < x_1 & x_0 < x_2 & x_1 < x_2 \end{array} \\ \begin{bmatrix} \mathscr{S}(x_1 < x_2) & -I & I \end{bmatrix} \end{array}. \tag{4.20}$$

The cohomology of the sheaf $\mathscr{S}$ over the posets is defined in the same way as cellular sheaves as the kernel of d^p modulo image of d^{p-1}:

$$H^p(E, \mathscr{S}) = \ker d^p / \operatorname{Im} d^{p-1}. \tag{4.21}$$

In a similar way to the interpretation of cohomological cellular sheaves, the cohomology of sheaves over poset of dimension zero $H^0(E, \mathscr{S})$ is isomorphic to the space of global sections of $\mathscr{S}$. This means that the space of global sections returns to compute the kernel of the matrix of the coboundary map of dimension zero since the image of dimension -1

doesn't exist, which is natural since the kernel checks the local consistency of sections. The higher degree sheaf cohomology spaces can hold useful information. By [Rob17b], a nontrivial element of $H^p(E;\mathscr{S})$ describes observations on the p-way intersections of source domains that are consistent on further restriction to $(p+1)$- way intersections, i.e. kernels of d^p, but do not arise from any observations from $(p-1)$-way intersections, i.e. image of d^{p-1}. Therefore, it exists many classes of limited consistency that cannot be included in global sections. Hence H^1 designs sections that are not present as global sections and reveals the values of data or variables that are consistent not across all models.

After presenting the notions of cellular sheaves, the concepts of sections, the construction of sheaves and computation of sheaves cohomology, we briefly present some applications of sheaves in engineering problems.

4.5 Applications of sheaf theory

On the level of signal processing, [Rob15] addressed a new overview and perspective to the sampling theorem that builds a connection between continuous-time signals and discrete-time signals. The author admits that the appropriate algebraic way to model sampling from classes of non-bandlimited functions is the sheaf because of its sensitivity to topology and its ability to transfer from local constructions to global conclusions. He used sheaf morphisms to generalize the process of sampling by transformations of samples to vector values of different dimensions.

In [Rob13a], the classical Nyquist- Shannon sampling theorem was recovered and extended by sheaf cohomology. Moreover, the author proves the interference of sheaves cohomology for sampling theorem of higher dimensions. The null cohomology of an associated sheaf that models the sample makes possible the reconstruction from this sample.

Another example of the application of the sheaf theory is in optimization domain: the classical max-flow-min-cut theorem is discussed from an algebraic topology point of view using sheaves in [GK13]. This theorem is modeled via sheaf theoretical concepts, that formulate the flow and cut values via sheaves cohomology and cosheaves homology.

Sheaves has also been used for studying networks. A new way of sheaf theory application on network coding problems is proposed in [GH11]. The authors represent a general multi source coding scheme by network sheaves. In this representation, they compute several designs of sheaf cohomologies. After demonstrating that the sheaf cohomology is equivalent to the information flows on the network, and basing on concepts from homology theory and exact sequences, this assumption is applied to popular problems in network coding like data fusion and global prolongation.

Data and its aggregation was largely studied by the applied view of sheaves. In [JHR14], the authors employ topology, category theory and combinatorics to build techniques of so called topological data modeling. They used these techniques to study problems in information fusion to get well understood data sources. The pairwise properties of analytics

and their multiple communication are represented by set systems and different kind of cell complexes that are suitable to check the consistency of information across the constructed topological complex. This construction will help in discovering reveal recurrent dependencies amongst data sources where misleading information might not be able to be diagnosed.

Instead of the cohomology of sheaves, an alternative reduction technique using the relation with Euler integrals is proposed in [CGR12]. The study of data aggregation over a domain can be realized using the proposed alternative of the cohomology of an associated constructed sheaf over the same domain.

Now we will proceed to the proposed methods of sheaves applications in images.

4.6 Proposed applications of sheaf theory on images

4.6.1 Application of cellular sheaves on images

Here we describe the strategy of building sheaves on a data source, which is the images in our case, recapitulating the concepts and the notions detailed in section 4.3. An application on an image will demonstrate how we are applying the concept of cellular sheaves on images.

When we build a sheaf over a complex, the process of assigning stalks and restriction maps is fundamentally a task of modeling. As such, it is a balance between being able to represent the data sufficiently, and being tractable for analysis, but we must keep in mind that:

1. If we build our base space in a systematic way, so that pixels or group of pixels lie on vertices, usually we can figure the stalks over the vertices without too much hassle.

2. We need to figure out how to assemble stalks over the "rest" of the complex. If we are working with a cell or CW complex, we now need to assign stalks over each 1-dimensional cell, then over each 2-dimensional cell, and then up the dimensional chain. For a given cell, we look at the stalks of its faces to try to discern the "best fit" common space for both. Often, there is an obvious choice, like common columns from a database, that works. But sometimes, we need to poke around a bit, to find what is common.

3. One of the big payoffs of using sheaves over purely global approaches is that they work locally, and this is often much easier to model. It may be hard to figure out "what is common" between many sources of information like pixels or group of pixels, but usually we can figure out what's common between two! That's all we need in order to go up by one dimension in the cell complex.

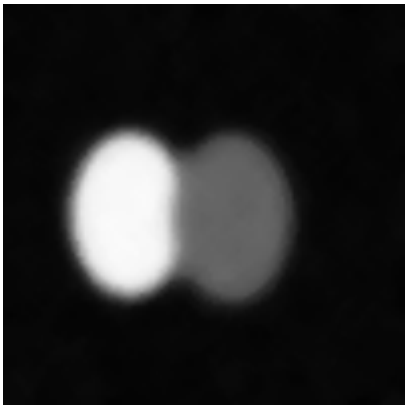


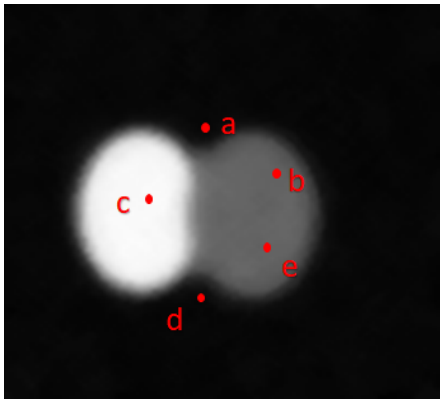
Figure 4.12: An image containing two glued objects.

4. If it is not entirely obvious what to pick for stalks on the intermediate faces, the thing to do is taking a step back and looking at the way we built the base space in the first place. If it's a Čech complex, and most are, then the vertices can also be thought of as open sets on which each data source "lies". Moving up in dimension means that we move to a smaller open subset, retaining only what is common to all larger sets containing that subset. This viewpoint is particularly useful for images, since the "open set" really is a set of pixels or locations in the scene.

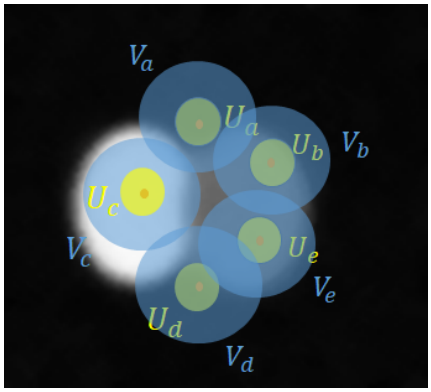
5. Finally, since we only need to work locally, the restriction maps need only act on pairs of stalks: one input and one output stalk. Sometimes, the restriction maps are obvious: forming a mosaic of several images uses fairly constrained image warps and crops as the restriction maps and these are often linear maps, since they transform the pixel values linearly, or they work like extracting columns from a table. But sometimes, they're a bit more involved.

Now, once we have got a sheaf, we can ask about global sections. Any given sheaf may or may not have global sections, and it can be hard to figure out what they are with no further constraints on the problem. But if all the stalks are vector spaces and each restriction map is linear, then we can compute cohomology. The nice consequence is that computing cohomology will automatically compute the space of global sections, regardless of whether this space is trivial or not. But, cohomology gives us potentially more information. Interpreting this information requires some reference back to the original problem: to see the purpose of building this sheaf and how we assigned restrictions on this sheaf and also to relate to the application task as we will see in this example below where we are trying to detect the presence of two different glued objects.

We explain thereafter how to apply this methodology on images. Here we present an application of our method on an image represented in 4.12 of size 100×100 pixels. In this image, we try to find a way to detect the two glued objects using cellular sheaves cohomology. Classical methods like thresholding and our proposed method using homology classes are not able to detect the presence of these two glued objects. The base space, is based here on the natural complex for sheaves over images, which is the Čech complex.



(a) Salient keypoints given by the SIFT method.



(b) The image with the open balls centered at its key points.

Figure 4.13: The keypoints of the image in figure 4.12 with the open balls.

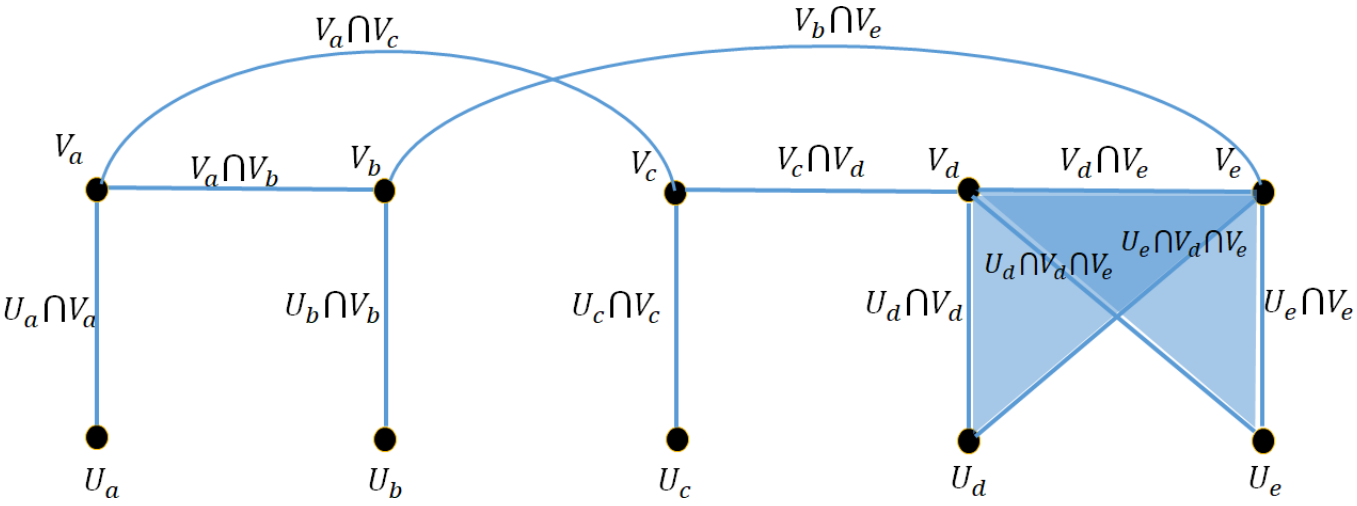


Figure 4.14: The base space for the construction of sheaves.

To build the Čech complex, we must choose some points in the image. We based this choice on the concept of keypoints. In image processing, keypoints represent interesting locations in the image. They're invariant with respect to images changes such as rotation, scaling, translation, etc. A technique to find these salient key points is the Scale Invariant Feature Transform (SIFT) method [Low04].

The steps of SIFT algorithm are the following. Determine approximate location and scale of salient feature points (also called keypoints), refine their location and scale, determine orientation(s) for each keypoint, and determine descriptors for each keypoint. We see in figure 4.13 (a) the keypoints of the image computed using the SIFT method. These keypoints were noted from a to e.

Now we must construct the Čech complex. We start from the idea that sheaves translate the common information between two spaces to the union of these two spaces. So we

thought in constructing two opens at each key point. The first open is a ball of radius 6 and the second is of radius 17 as shown in figure 4.13(b). The balls in yellow will be noted as $U_x = B(x, 6) = \{y | d(x, y) < 6\}$, and those in blue $V_x = B(x, 17) = \{y | d(x, y) < 17\}$ where $x, y \in \mathbb{R}^2$. Of course, the choice of the opens and its radii can change depending on the desired application. The purpose here is to present an example of how to construct cellular sheaves on images.

Once we assigned the complex, we can have the base space for the Čech complex constructed shown in figure 4.14. Vertices U_x and V_x represent the opens at a specific point x. Each 2 vertices are connected by an edge if their intersection is non empty. For example each U_x will be connected to V_x since their intersection is non empty. V_a is thus connected to V_b, U_d to V_e etc. For 2 dimensional faces, when 3 opens are intersected, we will add a triangle like the case of U_d, V_d and V_e.

We choose the stalks over the vertices. The chosen stalks are the most natural characteristics over the opens on these vertices. For that, we have chosen the means of intensity values and of gradient values in this open, so the vector space associated is $\mathbb{R}^2$. On the level of the edges, we will look on the intersections of the 2 opens and compute the same values, like the values inside $V_a \bigcap V_b$ for example. For the triangles, we look for the values inside the 3 intersected opens like in $U_d \bigcap V_d \bigcap V_e$. In summary, all the cells in the complex will hold $\mathbb{R}^2$ as a vector space over their stalks since we look only for mean of pixels values and gradients. Of course, other characteristics can be studied like the variance, the maximum value in each open and even homology classes for example.

Now we want to find the restrictions from lower dimensional cells to higher ones. These restrictions have $\mathbb{R}^2$ as input and output so they will be matrices of the form $\mathbb{R}^{2\times 2}$. Since we're looking for coherence over the sheaf space, we will compare the data over the low dimensional cells and their higher dimensional ones. But this time it's accompanied by a pseudosection definition introduced before in 4.3.2.

Taking for example an edge e with values $\{e_0, e_1\}$ associated with its vertices a and b with values $\{a_0, a_1\}$ and $\{b_0, b_1\}$ respectively

$$Data : \quad a : \{a_0, a_1\} \rightarrow e : \{e_0, e_1\} \leftarrow b : \{b_0, b_1\}. \tag{4.22}$$

Of course, if a_0 and b_0 are the mean pixel values inside V_a and V_b for example and e_0 is the mean pixels value inside their intersection, then e_0 is definitely different from a_0 and b_0. So it's necessary to use the tolerance introduced by the pseudosection concept.

For this, we will check if the value e_0 is between a_0 and b_0. The same for e_1, a_1 and b_1. If so, we will have a truth value and eventually a **pseudosection** and so we are accepting to have a tolerance and consider that these measures are consistent.

Like in the example of subsection 4.3.3 of cat/dog/bird and mammal/not mammal, we choose $A = \{x, y\}$ where x and y represent the mean pixel values and the mean gradient values inside an open represented by a cell. Also $B = \{coherent, not\ coherent\}$, we need a functoriality to transform these sets into vector spaces like described before. Noting that the cell represented by A is of lower dimension than B.

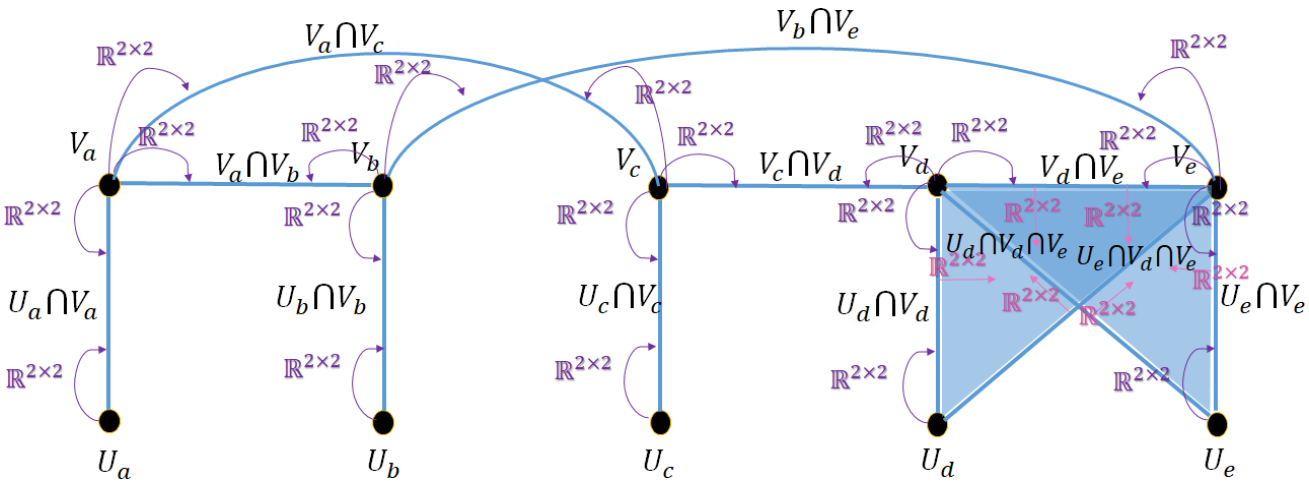


Figure 4.15: The sheaf base space with the restrictions.

We will try to lift the function between these two sets to a linear one like in equation (4.7):

$$\begin{array}{ccc} \mathbb{R}(A) & \xrightarrow{\mathbb{R}f} & \mathbb{R}(B) \\ {\scriptstyle(1\times)}\uparrow & & \uparrow{\scriptstyle(1\times)} \\ A & \xrightarrow{\quad f \quad} & B \end{array} \tag{4.23}$$

Here $\mathbb{R}(A) = \mathbb{R}^2$, $\mathbb{R}(B) = \mathbb{R}^2$ and $\mathbb{R}f : \mathbb{R}^2 \to \mathbb{R}^2$ is a 2×2 matrix.

If the mean pixel values on a specific cell is between the mean pixel value along its boundaries then $f(x) = 1$ and its 0 otherwise. And so the same for $f(y)$ but for mean gradient values.

Suppose that we have the following values on the edge $V_d \cap V_e$ and its vertices V_d and V_e represented in equation (4.24):

$$\begin{aligned} \text{Data :} \quad & V_d : \{148.3, 235.4\} \to V_d \cap V_e : \{98.6, 228.8\} \leftarrow V_e : \{69.7, 247.4\}. \\ \text{Sheaf over data :} \quad & V_d : \{x, y\} \xrightarrow{r_0} V_d \cap V_e : \{coherent, notcoherent\} \xleftarrow{r_1} V_e : \{x', y'\}. \end{aligned} \tag{4.24}$$

where r_0 and r_1 are restrictions from vertices V_d and V_e to edge $V_d \cap V_e$.

Then in this case $R_f = r_0 = r_1 \in \mathbb{R}^2$, its matrix is represented by:

$$\begin{array}{c} \\ \text{coherent} \\ \text{not coherent} \end{array} \begin{array}{c} \begin{array}{cc} x & y \end{array} \\ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \end{array}. \tag{4.25}$$

We represent in figure 4.15 the restrictions from low dimensional cells to their higher dimensional attached ones. Of course the restrictions will belong to $\mathbb{R}^{2\times2}$ since we're transforming the data from $\mathbb{R}^2$ to $\mathbb{R}^2$ from low dimensional cells to higher ones.

Now we will get coboundary maps d^0 and d^1 using

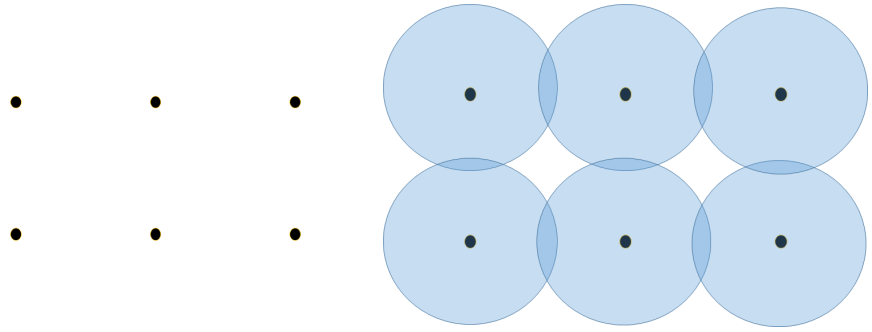
$$(d^p f)(\tau) = \sum_{\sigma \in X^p} [\sigma : \tau] \mathcal{S}(\sigma \rightsquigarrow \tau) f(\sigma) \tag{4.26}$$

and we're able to compute sheaves cohomology of zero and one dimensions over this complex using this equality $H^p(X, \mathcal{S}) = \ker d^p / \operatorname{Im} d^{p-1}$. In this case we have $H^0 =< U_a, U_b, U_c, U_e >$ and $H^1 =< U_d V_d + U_e V_e + V_d V_e, U_e V_d + U_e V_e + U_e V_e >$, where $< \ldots >$ includes the vectors generators of the cohomology group. We notice that H^1 detects some kind of incoherence between opens U_d, V_d, U_e, V_e which indicate an inconsistency between them and the absence of U_d in H^0 and so in the space of global section, that represent consistent sections, may justify this interpretation.

Even though H^0 and H^1 do not bring a lot of information in this case, but some kind of inconsistency is detected at the level of H^1. Other ways of modeling of the sheaf may bring an important information like the detection of the two glued objects by interpretation of H^1. Noting that the choice of the vertices, like chosing them manually, the radii of opens, the stalks over the vertices and restrictions is a task of modeling and affect the value of the cohomology. The right choice will lead to the detection of these glued objects.

4.6.2 Sections on RGB images

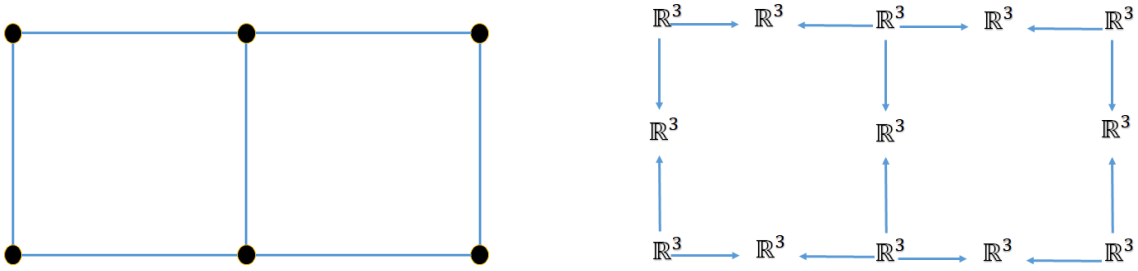
We have seen that sheaves also can be seen as sections of sheaves without their cohomological computations. These sections are useful on colored images. In fact, colored images have pixels with values belonging to $\mathbb{R}^3$ which correspond to the RGB channel. Each pixel will represent a vertex on the base space. First we construct a Čech complex to have a base space for our sheaf. The centers of the open balls will be naturally the pixels and the radius is $1/2 + \epsilon$ supposing that the distance between vertices is 1 and $0 < \epsilon < (\frac{\sqrt{2}}{2} - \frac{1}{2})$, in a way that two diagonal opens don't intersect. In this way we have only intersections between two opens and not 3, and thus only edges and not 2 dimensional faces as shown in figure 4.16.



(a) 6 pixels/vertices. (b) Opens over the 6 pixels.

Figure 4.16: 6 pixels with their opens with radius $1/2 + \epsilon$.

The stalks over the vertices and edges will be $\mathbb{R}^3$ as shown in figure 4.17. Now we must look for sections. As data over the vertices, we have chosen the RGB channel (r_i, g_i, b_i) for each vertex in the base space. For edges, we assign the difference of the values coming from vertices $(r_j - r_i, g_j - g_i, b_j - b_i)$. The restrictions from vertices to edges will be matrices of the form $\mathbb{R}^{3\times 3}$ as shown in the example in figure 4.18 that contains two vertices with restrictions to their common edge.



(a) The Čech complex of figure 4.16.

(b) Stalks assigning to the base space.

Figure 4.17: The Čech complex and the stalks over it.

It's clear that these restrictions respect the property of local sections described in subsection 4.3.2 and showed in these equations:

$$\begin{bmatrix} \frac{r_j}{r_i} - 1 & 0 & 0 \\ 0 & \frac{g_j}{g_i} - 1 & 0 \\ 0 & 0 & \frac{g_j}{g_i} - 1 \end{bmatrix} \begin{bmatrix} r_i \\ g_i \\ b_i \end{bmatrix} = \begin{bmatrix} r_j - r_i \\ g_j - g_i \\ b_j - b_i \end{bmatrix} \tag{4.27}$$

$$\begin{bmatrix} 1 - \frac{r_i}{r_j} & 0 & 0 \\ 0 & 1 - \frac{g_i}{g_j} & 0 \\ 0 & 0 & 1 - \frac{g_i}{g_j} \end{bmatrix} \begin{bmatrix} r_j \\ g_j \\ b_j \end{bmatrix} = \begin{bmatrix} r_j - r_i \\ g_j - g_i \\ b_j - b_i \end{bmatrix} \tag{4.28}$$

With these restrictions we get for the stalks assignment in figure 4.17 the sections that are consistent with the restrictions as shown in figure 4.19.

After the construction of these sections over the one dimensional space, we can compute the persistent homology on the image. Values over the edges can be $\|\mathcal{X}\|_2$, where $\mathcal{X} = (r_j - r_i, g_j - g_i, b_j - b_i)$ are the stalks value over the edges. The value of the 2 dimensional face can be the maximum of the values over the edges. Of course, any other criteria and assigning can be taken into consideration to construct the filtration scheme in order to compute persistent homology.

Noting that all the data resources can be different than pixels, like superpixels or anything that gives measures and values can be different than RGB values, or any ontology. All

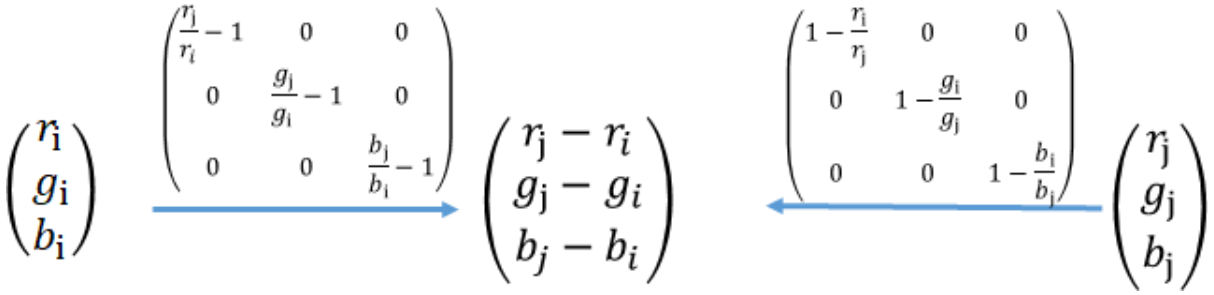


Figure 4.18: Sections over two vertices and one edge of the complex.

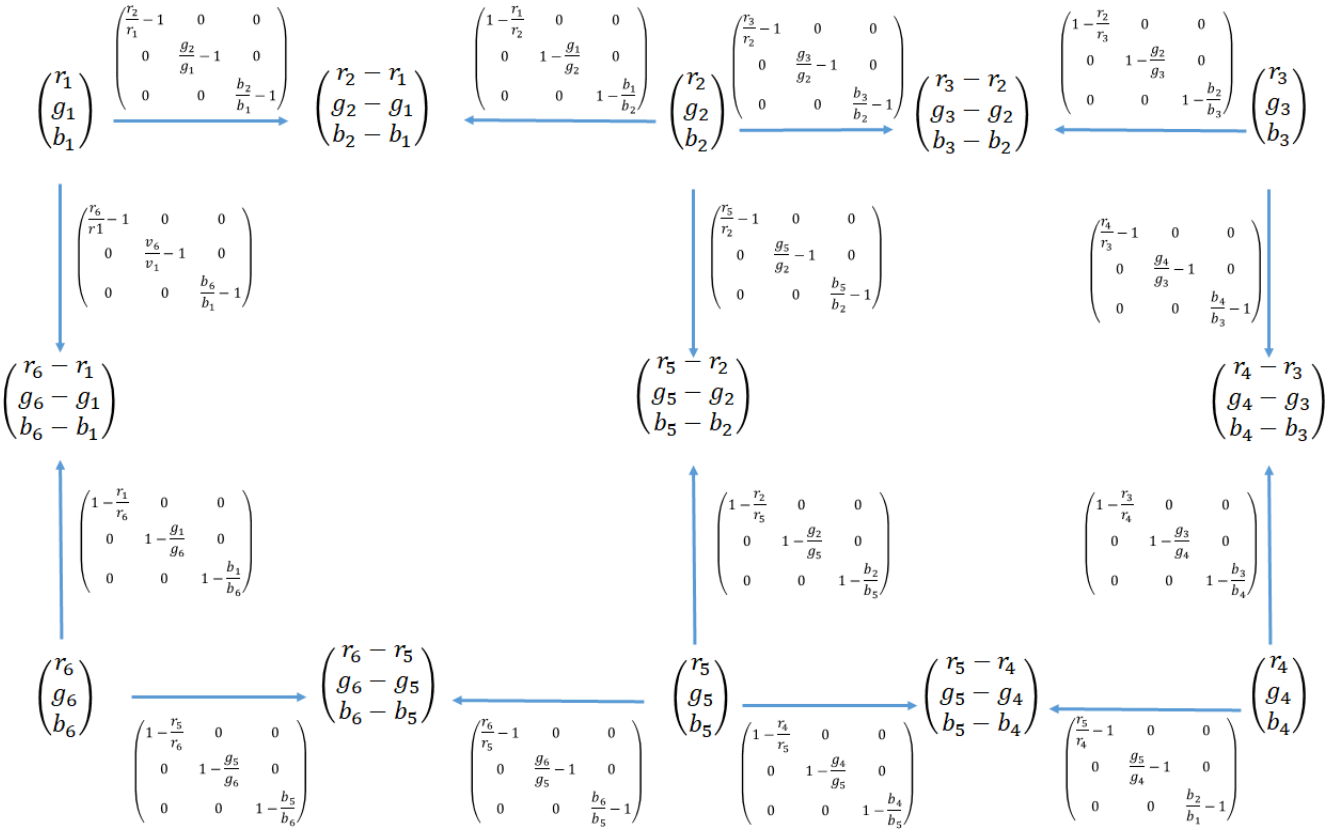


Figure 4.19: Sections consistent with restrictions over the 6 pixels.

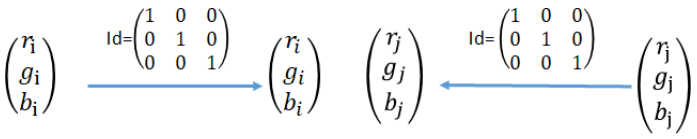


Figure 4.20: Local sections over 2 pixels with identity maps as restrictions.

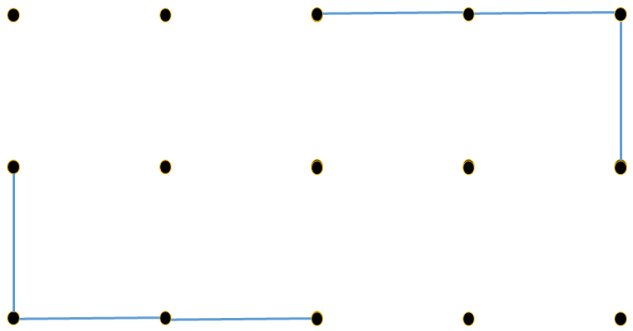


Figure 4.21: Consistent components according to the sheaves giving a partition of the image.

these criteria depends highly on the nature of the desired application and can change depending on their specific utilization.

Another way to benefit from sections on images is using identity matrices as restrictions on RGB values of pixels. This gives local sections all over the image. In figure 4.20, we build a section on two pixels/vertices and restrictions are identity maps of the form $\mathbb{R}^{3\times 3}$ since we're transforming stalks from $\mathbb{R}^3$ to $\mathbb{R}^3$. It's a local section since it satisfies the relation $\mathcal{S}(\sigma \rightsquigarrow \tau)(s(\sigma)) = s(\tau)$ for $\sigma \subset \tau$. This local section may globalized if it respects consistency, i.e. when it satisfies $\mathcal{S}(v_1 \rightsquigarrow e)s(v_1) = \mathcal{S}(v_2 \rightsquigarrow e)s(v_2)$, where v_1 and v_2 are vertices of the edge e. It means when the information coming from vertices on an edge is the same, i.e. if $r_i = r_j$, $g_i = g_j$ and $b_i = b_j$ and may rest local otherwise.

We will associate an edge between two pixels/vertices whenever we have the consistency respected. In this way the data coming directly from image are encoded as a local section. Other section "globalize" this data as edges until non consistency as shown in figure 4.21 where vertices are connected by edges when the sections are global.

Another example is to associate the Luma component, which designs the brightness in an image, to edges between pixels/vertices. The Luma component value of a pixel is $Y' = 0.299R + 0.587G + 0.114B$ where RGB designs the three channels of a pixel in a colored image. So restrictions are matrices of the form $\mathbb{R}^{1\times 3}$ and equal to $(0.299, 0.587, 0.114)$

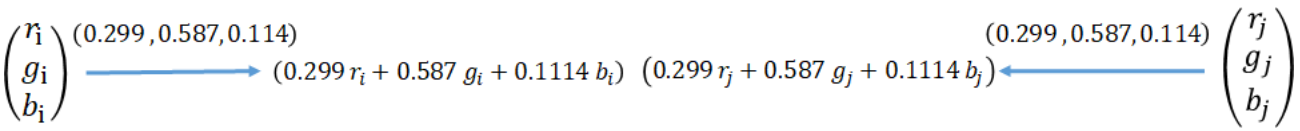


Figure 4.22: Local sections over 2 pixels with Luma maps as restrictions.

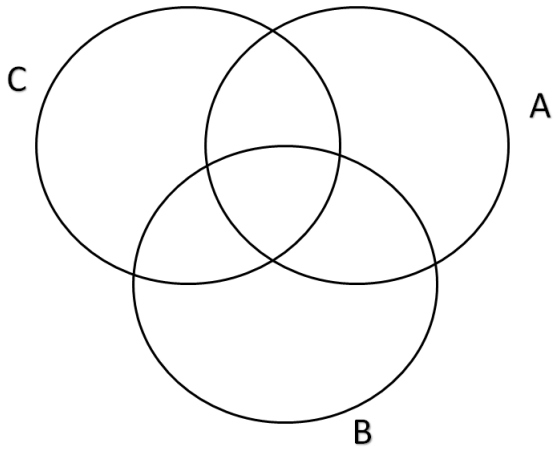


Figure 4.23: Three opens A, B and C.

since we're transforming stalks from $\mathbb{R}^3$ to $\mathbb{R}$ as shows the example on 2 pixels in figure 4.22. These sections respect the condition of a local section. We have a global section whenever the information coming from vertices is equal. We can associate an edge between two pixels/vertices in the same as previous example whenever we have the consistency respected in a local section.

Of course other constructions can be done on superpixels or any other aspect of images or data sources. Pseudosections can be taken into account to relax the consistency check in order to globalize the sections and also categorification may be used to transform the stalks that are sets to vector spaces.

4.6.3 Interpretation of sheaves of models

Models and their relations have many interactions. The most immediate characteristic of a multi model system is its topology. Modeling the topology of these models will allow to specify spaces and maps in accordance with the topology. This construction is based on sheaves. Sheaf theory is able to ensure techniques to construct and predict inferences described by equations. This will allow to manage local information into a consistent whole. For this purpose models can be represented by diagrams where sheaves are assigned to make inferences of models. These diagrams are manipulated by partially ordered sets.

We try in this subsection to create a tool that helps in understanding constitutes and characteristics of spaces in increasing nested order or decreasing one. This tool will permit to more understand and analyze the scaling of these spaces and the localization of objects or features inside them. Thanks to the cohomological analysis, or the homological in cosheaf case, we are able to create a prospective tool that permit an automatic analysis of data.

Constructing the sheaf models over the nested spaces will allow the analysis of the topo-

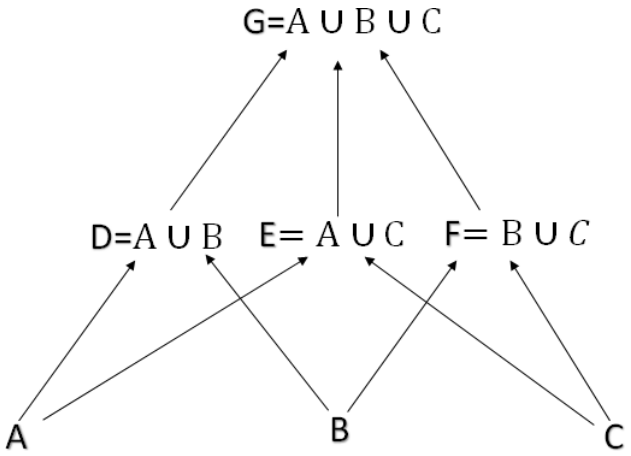


Figure 4.24: The poset $\mathcal{E}$ with arrows meaning inclusions.

logical invariants depending on the following steps:

1. Using the sheaf over a poset to encode diagrams of models as described in section 4.4
2. Linearising the data if necessary,
3. Computing the coboundary maps of this sheaf,
4. Computing cohomology to summarize the sheaf models and interpret them.

4.6.3.1 Scale analysis using sheaves on posets

Sheaves on posets can be used to perform inferences for scale analysis. Suppose that we have 3 opens A, B and C, as shown in figure 4.23.

Here we will go up in opens in an increasing way. We will note $D = A \cup B$, $E = A \cup C$, $F = B \cup C$ and $G = D \cup E \cup F$. Of course D, E, F and G define a topology on the set $\mathcal{E} = \{A, B, C, D, E, F, G\}$. As we have seen, that every topological space can define a poset on its open sets that are partially ordered by the subset relation. This poset is presented in 4.24 where arrows are pointed in increasing direction.

4.6.3.1.1 Example 1 Suppose we have the following model:

- A three opens like A, B and C.
- A cycle or an homology class that exist in $D = A \cup B$ as shown in 4.25 but not in $E = A \cup C$ nor $F = B \cup C$.

Now we will construct a sheaf $\mathscr{S}$ of sets on the poset $\mathcal{E}$ with the Alexandroff topology. For this, we must respect the three conditions mentioned in section 4.4. The figure 4.26 (a) shows the general form of the stalks and restrictions over this sheaf.

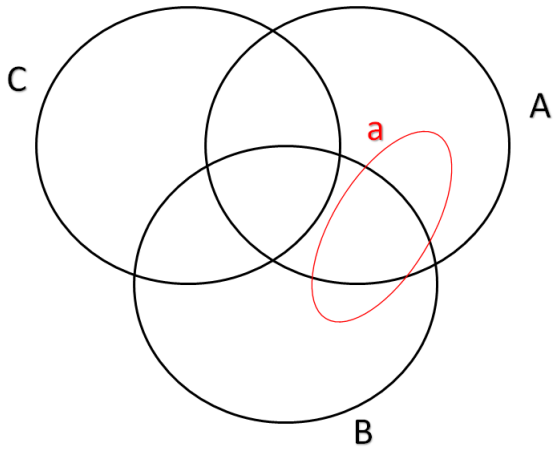
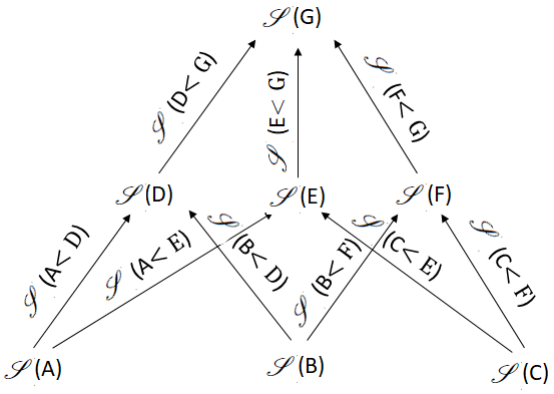
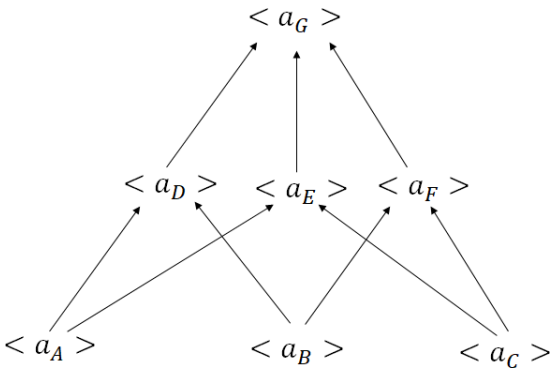


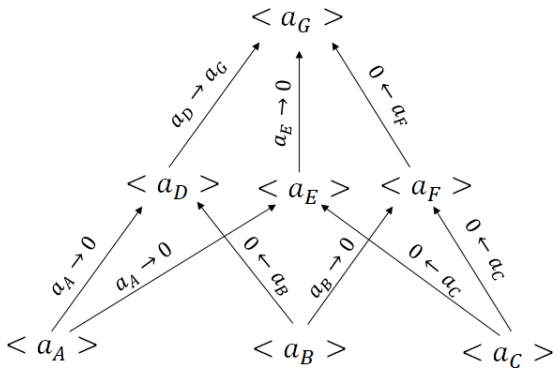
Figure 4.25: The opens with the cycle a in $A \cup B$.



(a) General form of stlaks and restrictions over the poset $\mathcal{E}$.

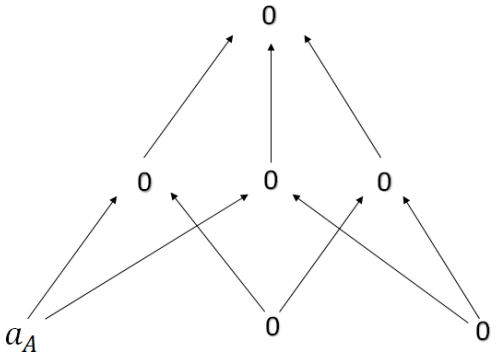


(b) Stalks over the sheaf $\mathscr{S}$ of the poset $\mathcal{E}$.

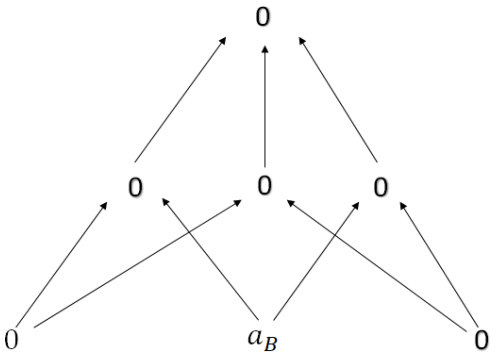


(c) Restrictions over the sheaf $\mathscr{S}$ of poset $\mathcal{E}$.

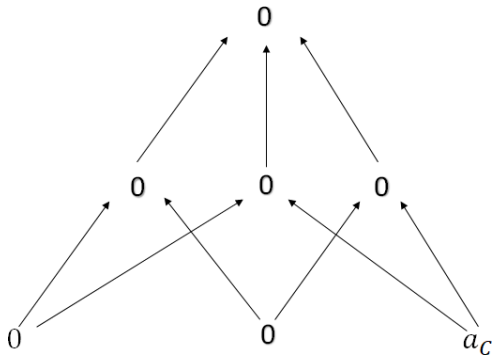
Figure 4.26: The sheaf over the poset $\mathcal{E}$.



(a) Global section generated by a_A.



(b) Global section generated by a_B.



(c) Global section generated by a_C.

Figure 4.27: Global sections of the sheaf $\mathscr{S}$ over poset $\mathcal{E}$.

Stalks over the cochains of length 0 are isomorphic to $\mathbb{R}$ and generated by a. The vector space generated by a is noted $< a >$ and we will refer to $< a_A >, < a_B >, < a_C >, < a_D >, < a_E >, < a_F >, < a_G >$ to stalks over A, B, C, D, E, F, G respectively as shown in figure 4.26 (b).

There is a restriction $\mathscr{S}(x \leq y)$, for each pair $x \leq y$ in the poset $\mathcal{E}$. It will transform a stalk a to 0 when it does not exit in the open so $\mathscr{S}(A \leq D) : a_A \to 0$, $\mathscr{S}(B \leq E) : a_B \to 0$, etc. because a doesn't exist in A nor B. If an element exists in two cochains it will be under the form $a_D \to a_G$ since the cycle a exists in D and G. The figure 4.26 (c) illustrates these restrictions all over the sheaf $\mathscr{S}$ over the poset $\mathcal{E}$.

Now we study the sections. The sections over the opens that do not contain the cycle a are 0, hence the sections $\mathscr{S}$ over A, B, C, E, F are 0. While the sections over the opens that contain the cycle a are a so the sections over D and G are a_D and a_G. It's clear that a_A, a_B and a_c are generators of the global section since their assigning is compatible with the consistency check of global section all over the sheaf $\mathscr{S}$, i.e. for all $x \leq y \in \mathcal{E}$ we have $\mathscr{S}(x \leq y)(s(x)) = s(y)$. We show in figure 4.27 the global sections generated by a_A, a_B and a_c respectively and that are prolonged all over the sheaf $\mathscr{S}$.

As a local section we have the section shown in 4.28. This section is defined in the same way as the global section but only on a subset $\mathcal{F} \subset \mathcal{E}$, thus this local section cannot be

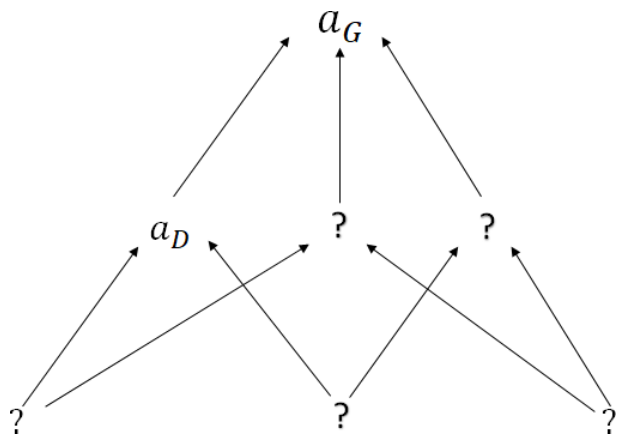


Figure 4.28: A local section on the sheaf $\mathscr{S}$ over the poset $\mathcal{E}$ which can not be extend to a global section.

extended all over the sheaf $\mathscr{S}$ and remains limited in a subspace of $\mathcal{E}$.

This interpretation of global and local section can be done automatically using sheaves cohomology. The cohomological analysis of this sheaf will allow us to interpret and understand the studied model. The cochain space is $C^p(\mathcal{E}, \mathscr{S}) = \prod_{x_0 < \ldots < x_p} \mathscr{S}(x_p)$, hence:

- $C^0(\mathcal{E}, \mathscr{S}) = (A) \otimes (B) \otimes (C) \otimes (D) \otimes (E) \otimes (F) \otimes (G)$.

- $C^1(\mathcal{E}, \mathscr{S}) = (A < D) \otimes (A < E) \otimes (B < D) \otimes (B < F) \otimes (C < E) \otimes (C < F) \otimes (A < G) \otimes (B < G) \otimes (C < G) \otimes (D < G) \otimes (E < G) \otimes (F < G)$.

- $C^2(\mathcal{E}, \mathscr{S}) = (A < D < G) \otimes (A < E < G) \otimes (B < D < G) \otimes (B < F < G) \otimes (C < E < G) \otimes (C < F < G)$.

Using the equation (4.19) on page 117, the coboundary matrix $d^0_{12\times 7}$ of this sheaf is:

$$d^0(\mathcal{E},\mathscr{S}) = \begin{array}{c} \\ (A<D) \\ (A<E) \\ (B<D) \\ (B<F) \\ (C<E) \\ (C<F) \\ (A<G) \\ (B<G) \\ (C<G) \\ (D<G) \\ (E<G) \\ (F<G) \end{array} \begin{array}{c} \begin{array}{ccccccc} (A) & (B) & (C) & (D) & (E) & (F) & (G) \end{array} \\ \begin{bmatrix} -\mathscr{S}(A<D) & \cdot & \cdot & I & \cdot & \cdot & \cdot \\ -\mathscr{S}(A<E) & \cdot & \cdot & \cdot & I & \cdot & \cdot \\ \cdot & -\mathscr{S}(B<D) & \cdot & I & \cdot & \cdot & \cdot \\ \cdot & -\mathscr{S}(B<F) & \cdot & \cdot & \cdot & I & \cdot \\ \cdot & \cdot & -\mathscr{S}(C<E) & \cdot & I & \cdot & \cdot \\ \cdot & \cdot & -\mathscr{S}(C<F) & \cdot & \cdot & I & \cdot \\ -\mathscr{S}(A<G) & \cdot & \cdot & \cdot & \cdot & \cdot & I \\ \cdot & -\mathscr{S}(B<G) & \cdot & \cdot & \cdot & \cdot & I \\ \cdot & \cdot & -\mathscr{S}(C<G) & \cdot & \cdot & \cdot & I \\ \cdot & \cdot & \cdot & -\mathscr{S}(D<G) & \cdot & \cdot & I \\ \cdot & \cdot & \cdot & \cdot & -\mathscr{S}(E<G) & \cdot & I \\ \cdot & \cdot & \cdot & \cdot & \cdot & -\mathscr{S}(F<G) & I \end{bmatrix} \end{array} =$$

$$\begin{array}{c} \\ (A<D) \\ (A<E) \\ (B<D) \\ (B<F) \\ (C<E) \\ (C<F) \\ (A<G) \\ (B<G) \\ (C<G) \\ (D<G) \\ (E<G) \\ (F<G) \end{array} \begin{array}{c} \begin{array}{ccccccc} (A) & (B) & (C) & (D) & (E) & (F) & (G) \end{array} \\ \begin{bmatrix} \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot \\ \cdot & \cdot & \cdot & + & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot \\ \cdot & \cdot & \cdot & \cdot & + & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & + & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & - & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & + \end{bmatrix} \end{array} \tag{4.29}$$

And by equation (4.20) on page 117 $d^1_{6\times 12}$ is:

$$
d^1(\mathcal{E},\mathscr{S}) =
\begin{array}{c}
\\ (A<D<G)\\ (A<E<G)\\ (B<D<G)\\ (B<F<G)\\ (C<E<G)\\ (C<F<G)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
(A<D) & (A<E) & (B<D) & (B<F) & (C<E) & (C<F) & (A<G) & (B<G) & (C<G) & (D<G) & (E<G) & (F<G)
\end{array}\\
\begin{bmatrix}
\mathscr{S}(D<G) & \cdot & \cdot & \cdot & \cdot & \cdot & -I & \cdot & \cdot & I & \cdot & \cdot \\
\cdot & \mathscr{S}(E<G) & \cdot & \cdot & \cdot & \cdot & -I & \cdot & \cdot & \cdot & I & \cdot \\
\cdot & \cdot & \mathscr{S}(D<G) & \cdot & \cdot & \cdot & \cdot & -I & \cdot & I & \cdot & \cdot \\
\cdot & \cdot & \cdot & \mathscr{S}(F<G) & \cdot & \cdot & \cdot & -I & \cdot & \cdot & \cdot & I \\
\cdot & \cdot & \cdot & \cdot & \mathscr{S}(E<G) & \cdot & \cdot & \cdot & -I & \cdot & I & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \mathscr{S}(F<G) & \cdot & \cdot & -I & \cdot & \cdot & I
\end{bmatrix}
\end{array}
$$

$$
=
\begin{array}{c}
\\ (A<D<G)\\ (A<E<G)\\ (B<D<G)\\ (B<F<G)\\ (C<E<G)\\ (C<F<G)
\end{array}
\begin{array}{c}
\begin{array}{cccccccccccc}
(A<D) & (A<E) & (B<D) & (B<F) & (C<E) & (C<F) & (A<G) & (B<G) & (C<G) & (D<G) & (E<G) & (F<G)
\end{array}\\
\begin{bmatrix}
+ & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & \cdot & + & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & \cdot & \cdot & + & \cdot \\
\cdot & \cdot & + & \cdot & \cdot & \cdot & \cdot & - & \cdot & + & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & \cdot & \cdot & + \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & + & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & - & \cdot & \cdot & +
\end{bmatrix}
\end{array}
\tag{4.30}
$$

Knowing that the value of $\mathscr{S}(A < D)$ is zero because it's translating a_A to 0 since $\mathscr{S}(A \leq D) : a_A \to 0$ and so on for the others who are similar. The value of $\mathscr{S}(D < G)$ is 1 because it's translating a_D to a_G since $\mathscr{S}(A \leq D) : a_D \to a_G$.

Once we have the coboundary matrices $d^0(\mathcal{E},\mathscr{S})$ and $d^1(\mathcal{E},\mathscr{S})$, we can compute the cohomology of zero and one dimension. Recall that $H^p(E,\mathscr{S}) = \ker d^p / \operatorname{Im} d^{p-1}$, then $H^0(E,\mathscr{S}) = \ker d^0 / \operatorname{Im} d^{-1} = \ker d^0$.

Thus $H^0(\mathcal{E},\mathscr{S})$ is generated by $< a_A, a_B, a_C >$, which is the space of global sections of this sheaf model. This reveals the consistency of these elements across the sheaf which is normal since a_A, a_B and a_C are transformed via null functions.

On another side $H^1(\mathcal{E},\mathscr{S}) =< a_{C<E}, a_{C<F} >$, which are sections of cochains of length 1 that are not present as global sections and may give extra information and inference of the model at the level of cochains of length 1. The presence of $a_{C<E}$ and $a_{C<F}$ in H^1

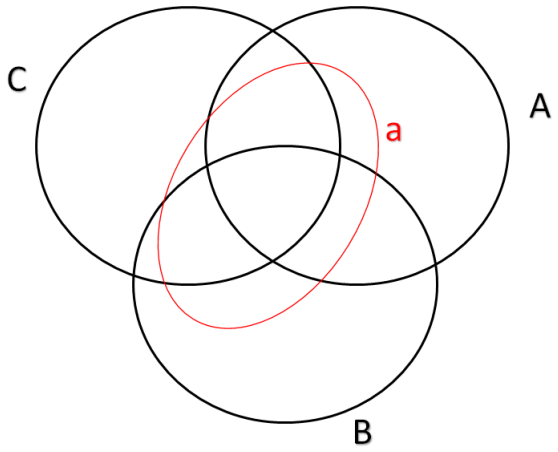


Figure 4.29: The opens with the cycle $a \in A \cup B \cup C$.

means that there are obstructions to the extension of the global sections at the level of the cochains $C < E$ and $C < F$.

Interpreting the results of H^0, we conclude that the cycle a doesn't exist in any of the big opens A, B and C. For H^1, a is not included in $E = A \cup C$ nor $F = B \cup C$, which let us deduce that this element is in $D = A \cup B$.

Now we will try to change the study a little bit. Let the cycle a be in the union of all the opens and not included in any other open as shown in figure 4.29.

4.6.3.1.2 Example 2 Supposing we have the following model:

- A three opens like A, B and C.
- A cycle or an homology class that doesn't exist in $D = A \cup B$, $E = A \cup C$ nor $F = B \cup C$ as shown in 4.29 but rather in $G = A \cup B \cup C$.

Now we will construct a sheaf $\mathscr{S}'$ of sets on the same poset $\mathcal{E}$ with the Alexandroff topology. For this, we must respect the three conditions mentioned in section 4.4. The general form of this sheaf $\mathscr{S}'$ is as same as the one described in described in example 1 in figure 4.26 (a).

As in example 1, stalks over the cochains of length 0 are isomorphic to $\mathbb{R}$ and generated by a. The stalks over A, B, C, D, E, F, G respectively are the same as shown in figure 4.26 (b).

There is a restriction $\mathscr{S}'(x \leq y)$, for each pair $x \leq y$ in the poset $\mathcal{E}$. It will transform a stalk a to 0 when it doesn't exit in the open so $\mathscr{S}'(A \leq D) : a_A \to 0$, $a_B \to 0$, $a_D \to 0$ etc. because a doesn't exist in A, B nor D. The figure 4.30 illustrates these restrictions all over the sheaf $\mathscr{S}'$ over the poset $\mathcal{E}$.

Now for the sections, The sections over the opens that do not contain the cycle a are 0. Hence, the sections $\mathscr{S}$ over A, B, C, ,D, E and F are 0. While the sections over the opens

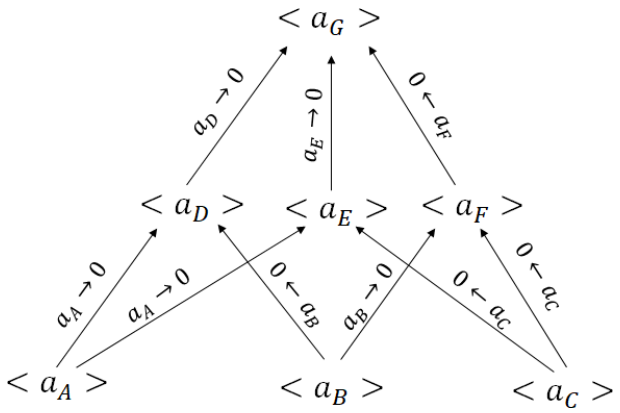


Figure 4.30: Restrictions over the sheaf $\mathscr{S}'$ of poset $\mathcal{E}$.

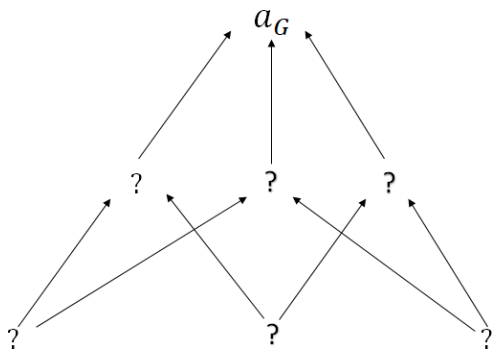


Figure 4.31: A local section on the sheaf $\mathscr{S}'$ over the poset $\mathcal{E}$ which can not be extend to a global section.

that contain the cycle a are a so the section over G is a_G. Like in example 1, it's clear that a_A, a_B and a_c are generators of the global section since their assigning is compatible with the consistency check of global section all over the sheaf $\mathscr{S}$, i.e. for all $x \leq y \in \mathcal{E}$ we have $\mathscr{S}(x \leq y)(s(x)) = s(y)$. The figure 4.27 of example 1 shows the global sections generated by a_A, a_B and a_c respectively and that are prolonged all over the sheaf $\mathscr{S}$.

As a local section for this example 2 we have the section showed in 4.31. This section is defined in the same way as the global section but on a subset $\mathcal{F} \subset \mathcal{E}$, thus this local section cannot be extended all over the sheaf $\mathscr{S}$ and remains limited in a subspace of $\mathcal{E}'$.

We will see these interpretation automatically using sheaves cohomology. Concerning d^0 and d^1, the only difference in d^0 and d^1 is $\mathscr{S}'(D < G)$ which is equal to 0 in this case. Computing the sheaves cohomology over this poset will lead to $H^0(\mathcal{E}, \mathscr{S}') =< a_A,\ a_B,\ a_C >= H^0(\mathcal{E}, \mathscr{S})$ and $H^1(\mathcal{E}, \mathscr{S}) =< a_{B<D},\ a_{C<E},\ a_{C<F} >$. The presence of $a_{B<D}$, $a_{C<E}$ and $a_{C<F}$ in H^1 means that there are obstructions to the extension of the global sections at the level of the cochains $B < D$, $C < E$ and $C < F$. As same as example 1, H^0 let us conclude that the class a doesn't exist in any of the big opens. The interpretation of cohomology of one dimension over poset $\mathcal{E}$ at the level of cochains of length 1 leads to verify that a is not in any of $D = A \cup B$, $E = A \cup C$ and $F = B \cup C$

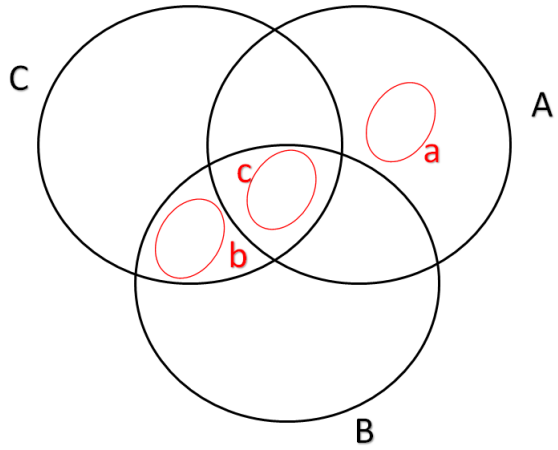
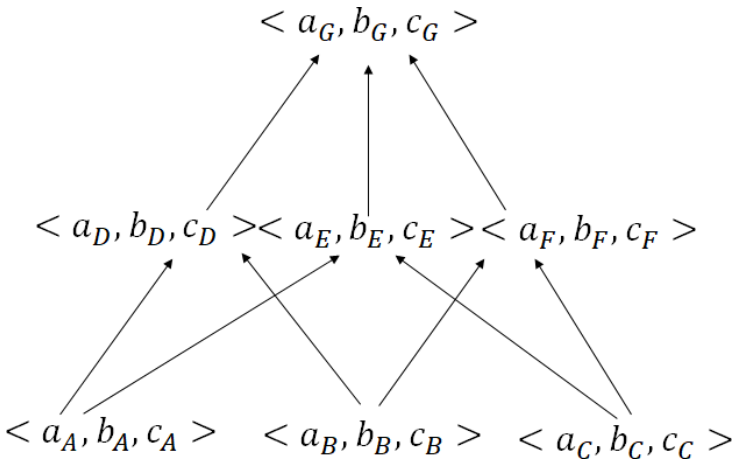
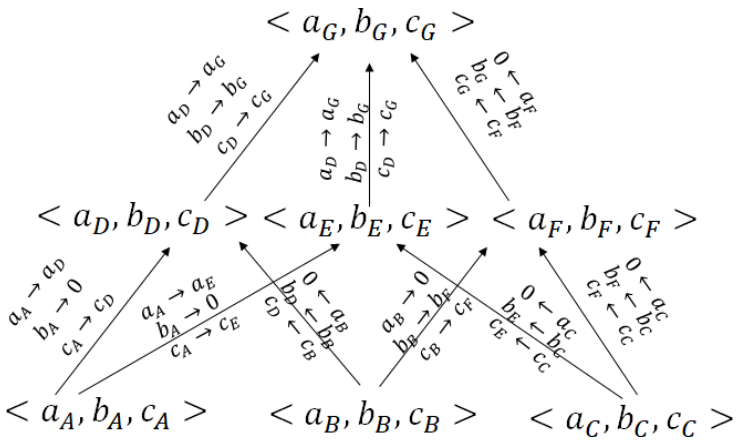


Figure 4.32: The opens with the cycles a, b and c for the case of example 3 in scale analysis.

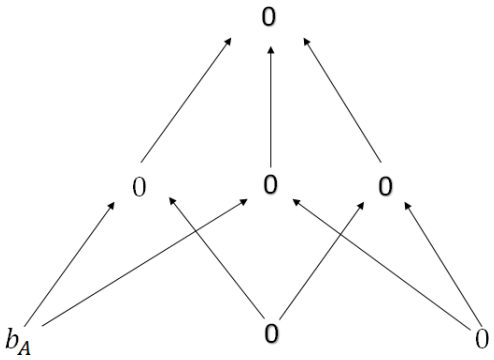


(a) Stalks over the sheaf $\mathscr{S}''$ of poset $\mathcal{E}$.

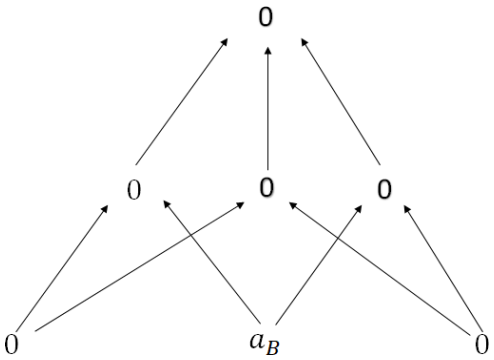


(b) Restrictions over the sheaf $\mathscr{S}''$ of poset $\mathcal{E}$.

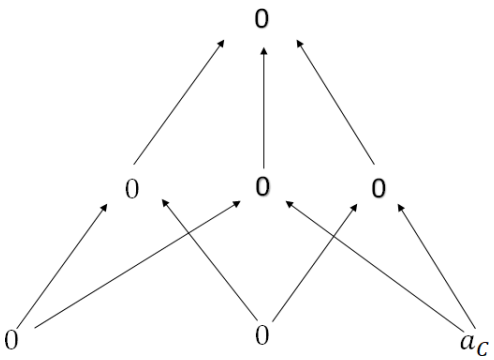
Figure 4.33: The sheaf $\mathscr{S}''$ over the poset $\mathcal{E}$.



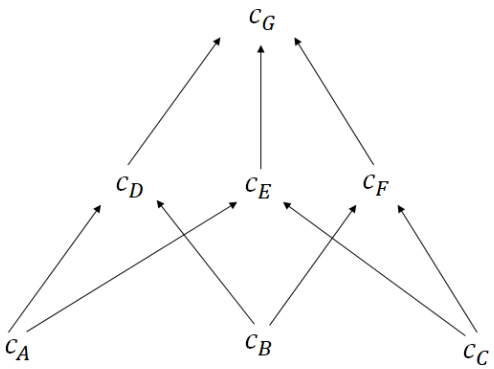
(a) Global section generated by b_A.



(b) Global section generated by a_B.



(c) Global section generated by a_C.



(d) Global section generated by $c_A + c_B + c_C + c_D + c_E + c_F + c_G$.

Figure 4.34: Global sections of the sheaf $\mathscr{S}''$ over poset $\mathcal{E}$.

and so a in in $G = A \cup B \cup C$.

Now, we will move to a more sophisticated model described in the next example.

4.6.3.1.3 Example 3

Suppose we have the following model:

- A three opens like A, B and C.
- Cycles $a, c \in A$, $b, c \in B$ and $b, c \in C$ as shown in figure 4.32.

Now we will construct a sheaf $\mathscr{S}''$ of sets on the poset $\mathcal{E}$ with the Alexandroff topology. Again the figure 4.26 (a) shows the general form of the stalks and restrictions over this sheaf.

Stalks over the cochains of length 0 are isomorphic to $\mathbb{R}^3$ now and generated by a, b and c. The vector space generated by a, b and c are noted $< a >$, $< b >$ and $< c >$ respectively and we will refer to $< a_A, b_A, c_A >, < a_B, b_B, c_B >, < a_C, b_C, c_C >, < a_D, b_D, c_D >, < a_E, b_E, c_F >, < a_F, b_F, c_F >, < a_G, b_G, c_G >$ to stalks over A, B, C, D, E, F, G respectively as shown in figure 4.33 (a).

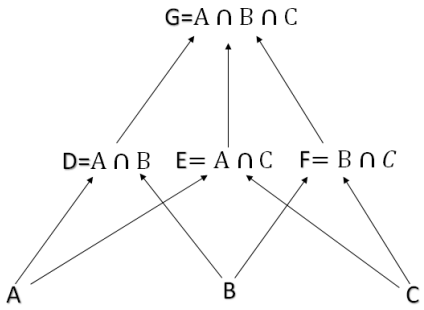


Figure 4.35: The poset $\mathcal{E}'$ with arrows in decreasing direction.

Restrictions of this sheaf $\mathscr{S}''$ look also for the existence of an element between two cohains as same as examples 1 and 2. The figure 4.33 (b) illustrates these restrictions all over the sheaf $\mathscr{S}''$ over the poset $\mathcal{E}$.

Now we study the sections. The sections over the opens that do not contain the cycle a are 0, and it's a otherwise, for example the section over A is $(a_A, 0, c_A)$ since it contains the cycles a and c and does not contain the cycle b. The same assigning is done for the other opens in the poset.

Generators of the global section are sections consistent all over the sheaf that respect the condition of consistency, i.e. for all $x \leq y \in \mathcal{E}$ we have $\mathscr{S}''(x \leq y)(s(x)) = s(y)$. The figure 4.34 illustrates these global sections generated by b_A, a_B, a_C and $c_A + c_B + c_C + c_D + c_E + c_F + c_G$ since their assigning is compatible with the consistency check of global section all over the sheaf $\mathscr{S}''$ and thus extended on all the sheaf.

Computing the sheaves cohomology on this sheaf $\mathscr{S}''$, we get $H^0(\mathcal{E}, \mathscr{S}'') =< b_A,\ a_B,\ a_C,\ (c_A + c_B + c_C + c_D + c_E + c_F + c_G) >$. H^0 indicates the absence of b in A and a in B and C, it also indicates the presence of c all over the scale increasing which produce a cocycle. H^0 is the space of global sections so it resumes consistent stalks across the sheaf. In the case of b_A, a_A, a_C, they are transformed via null functions and the case of elements of $(c_A + c_B + c_C + c_D + c_E + c_F + c_G)$ that are transformed via identity maps all the way so they respect the global section conditions at the level of cochain of length 0.

For the one dimension, $H^1 =< c_{A<G} + c_{B<G} + c_{C<G} + c_{D<G} + c_{E<G} + c_{F<G} >$, which confirm the existence of the cocycle even in length 1 cochains, and the presence of c at these cochains.

4.6.3.2 Localization using sheaves on posets

Sheaves on posets can be applied to localize objects or other characteristics over the posets in a decreasing way. Suppose the sheaf model is built on a poset $\mathcal{E}'$ in the way described in figure 4.35.

Here we will go down in opens in a decreasing way. We will note $D = A \cap B$, $E = A \cap C$, $F = B \cap C$ and $G = D \cap E \cap F$. Of course G, F, E, D, C, B and A define a topology

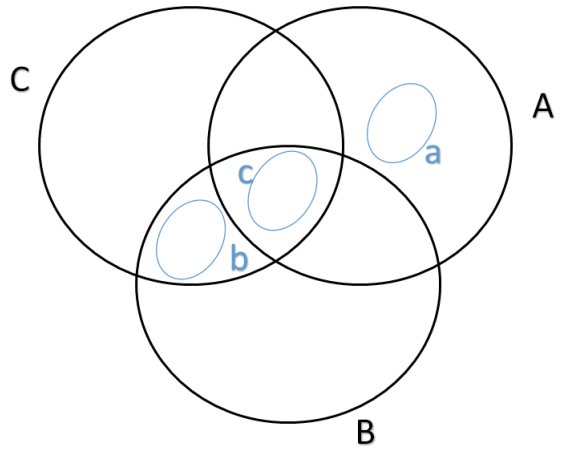
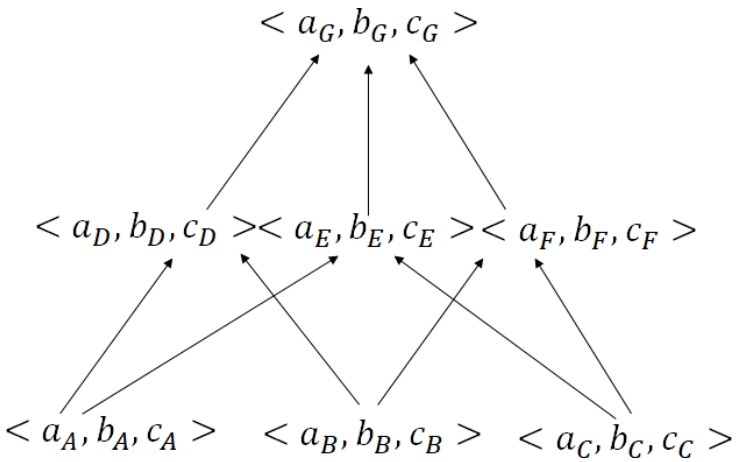
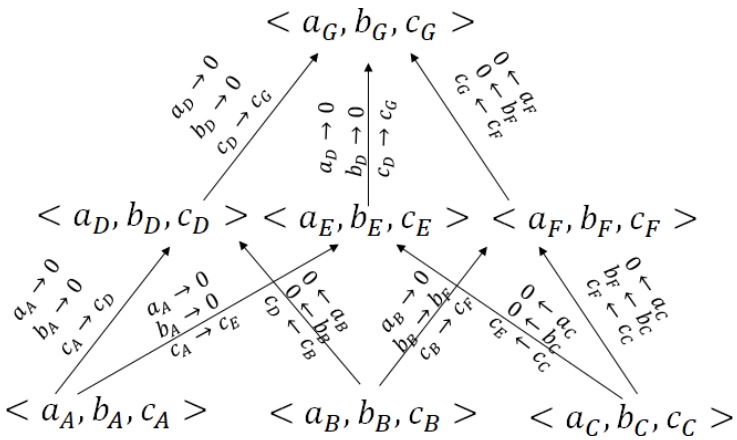


Figure 4.36: The opens with the cycles a, b and c for the localization example.



(a) Stalks over the sheaf $\mathscr{S}$ of poset $\mathcal{E}'$.



(b) Restrictions over the sheaf $\mathscr{S}$ of poset $\mathcal{E}'$.

Figure 4.37: The sheaf $\mathscr{S}$ over the poset $\mathcal{E}'$.

on the set $\mathcal{E} = \{G, F, E, D, C, B, A\}$. The superset relation on these defines a relation of partial order on $\mathcal{E}'$ which makes it a poset. This poset is presented in figure 4.35 where arrows are pointed in decreasing direction.

Suppose we have the following model:

- A three opens like A, B and C.
- Cycles $a, c \in A$, $b, c \in B$ and $b, c \in C$ as shown in figure 4.36.

Now we will construct a sheaf $\mathscr{S}$ of sets on the poset $\mathcal{E}'$ with the Alexandroff topology. Again the figure 4.26 (a) serves as a general form of the stalks and restrictions over this sheaf.

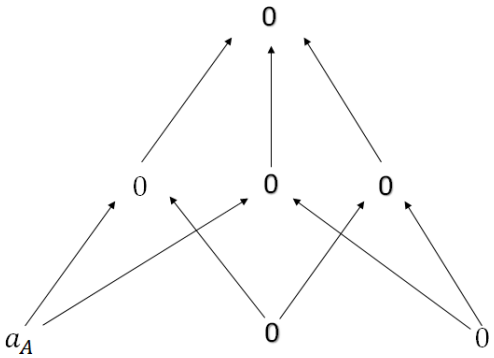
Stalks over the cochains of length 0 are isomorphic to $\mathbb{R}^3$ now and generated in the same way as shown in figure 4.37 (a).

Restrictions of this sheaf $\mathscr{S}$ look also for the existence of an element between two cohains as same as examples3. The figure 4.37 (b) illustrates these restrictions all over the sheaf$\mathscr{S}$ over the poset $\mathcal{E}'$.

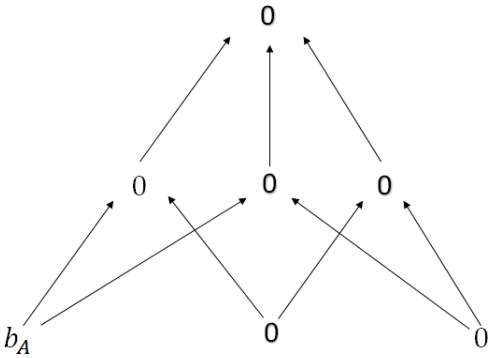
Now we study sections. Generators of the global section are sections consistent all over the sheaf that respect the condition of consistency, i.e. for all $x \leq y \in \mathcal{E}$ we have $\mathscr{S}''(x \leq y)(s(x)) = s(y)$. The figure 4.38 illustrates these global sections generated by a_A, b_A, a_B, $b_B + b_C + b_F$ and $c_A + c_B + c_C + c_D + c_E + c_F + c_G$ since their assigning is compatible with the consistency check of global section all over the sheaf $\mathscr{S}''$ and thus extended on all the sheaf.

Constructing the sheaf in the same way as the previous descriptions, the values over the opens follow their existence and the restrictions between them respect this criteria. After computation of d^0 and d^1, we are able to compute the space of global sections. $H^0 =< a_A,\ b_A,\ a_B,\ a_c,\ b_B + b_C,\ c_A + c_B + c_C + c_D + c_E + c_F + c_G >$. In the decreasing way, vectors of H^0 may indicate the absence of this element like b_A, a_B and a_c or its presence followed by an absence like a_A because they are transformed by null functions which make them consistent and thus appearing in global sections. While vectors of many elements indicate the presence of these elements across the cochains of length 0 like $(b_B + b_C)$ and $(c_A + c_B + c_C + c_D + c_E + c_F + c_G)$ because they are transformed via identity maps which make them also consistent sections that respect the global section condition. $(b_B + b_C)$ makes b in B and C while $(c_A + c_B + c_C + c_D + c_E + c_F + c_G)$ makes c present in all opens, cochains of length 0, including $A \cap B \cap C$ and thus we make sure that b_A indicates the absence of b in A.

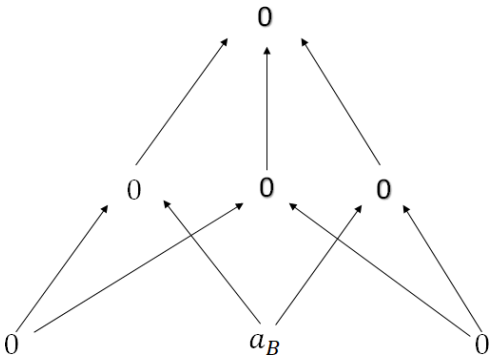
$H^1 = \{a_{B<D}, b_{B<D}, a_{C<E}, b_{C<E}, a_{C<F}\}$. A lot of information can be deduced from H^1 since vectors with one element indicate the absence of this element in cochains of length one. We will see also if it can discriminate the case when a vector with one element in H^0 indicates the presence or the absence of the element like the case of a_A, a_B and a_C. The vector represented by $a_{B<D}$ indicates the absence of a in $D = A \cap B$, $a_{C<E}$ and $a_{C<F}$ means also that a is not in $E = A \cap C$ nor $E = B \cap C$.



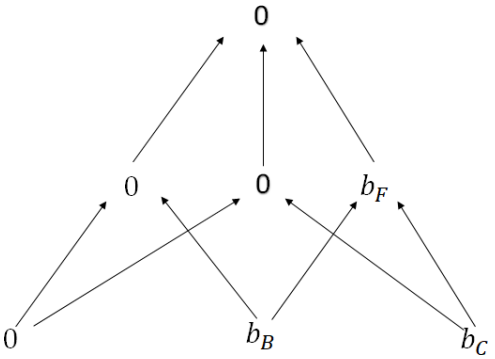
(a) Global section generated by a_A.



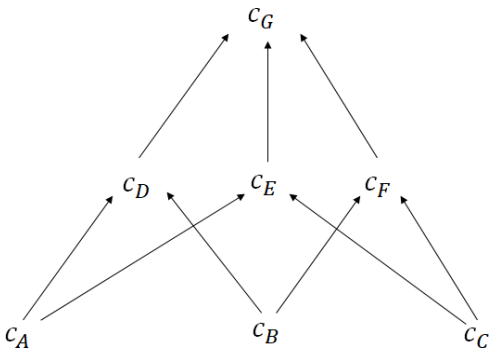
(b) Global section generated by b_A.



(c) Global section generated by a_B.



(d) Global section generated by $b_B + b_C + b_F$.



(e) Global section generated by $c_A + c_B + c_C + c_D + c_E + c_F + c_G$.

Figure 4.38: Global sections of sheaf $\mathscr{S}$ over poset $\mathcal{E}'$.

In turn, $b_{B<D}$ and $b_{C<E}$ indicate that b is not in $D = A \cap B$ nor in $E = A \cap C$ then we're sure that it is in $F = B \cap C$. c isn't any element of H^1 so it's in $D = A \cap B$, $E = A \cap C$, and $F = B \cap C$ thus it's in $G = A \cap B \cap C$ which verify the interpretation of H^0. As we have seen cohomological analysis verify the data and detect the propagation of characteristics across it which permit to localize them.

Remarking that these opens can be any open structures in the image that respect the Alexandroff topology like windows of pixels, superpixels, adjacent spaces etc. and characteristics can be any object or feature inside these opens.

4.7 Conclusion

In this chapter, we tried to dive into a new aspect of the applied algebraic topology that relies on sheaves theory. We started by an introduction into sheaves in section 4.1 followed by the basic aspect of sheaves on topological spaces 4.2. This fundamental aspect is translated to cellular spaces. We provide in section 4.3 the concept of cellular sheaves describing how to construct a sheaf on a cell complex and getting therefore cochain complexes associated with coboundary maps. These coboundary maps allow to compute sheaves cohomology in order to get a topological invariant that transform the local information into general inference. The section 4.4 introduced the construction of sheaves over partially ordered sets and its cohomological analysis. Next the section 4.5, developp the state of art of sheaves applications.

The final part 4.6, extends our contribution in this chapter. We initiated applications of cellular sheaves on image examples and we sheafify a Čech complex constructed on some keypoints of this image. The coboundary maps over this complex allowed to compute cohomology and deducing inferences. Changing the base space and restrictions may allow the detection of glued objects. Moreover sections of sheaves permitted to compute persistent homology on color images with RGB channels because of benefit from the vector spaces genereated by these channels. We set up the basics of scale analysis and localization over spaces using cohomological analysis over partially ordered sets. We tried to better understand and predict contents of the sheaf model and then verifyin it using sheaves cohomology. Not to forget, that constructing sheaves is a task of modelling, so it depends highly on the application and many criteria and way of constructions can be taken into consideration. Applications of these concepts on real images is one of the major perspectives on the court term.

Chapter 5

Conclusion and perspectives

5.1 General conclusion 145
5.2 Perspectives 147

5.1 General conclusion

Topology and algebraic topology feed the development of several methodologies to answer engineering problems. This mathematical theory is particularly interesting, and its applicability is mainly due to the possibility to return a computable part and represented on a real system. For example, for image processing tasks, simplicial homology in its persistence aspect and its relative form, and sheaf theory have been particularly effective. Lifespans of persistent homology classes give rise to a new way of image segmentation, while highlighting the most persistent homology classes that resisted the variations of spaces permitted to segment the objects in 2D and 3D dimensions. Not forgetting the relative version that allows to dump a subspace in the complex and thus to enlarge the notion of cycle for tracking of moving objects. Sheaf theory allowed to infer from local perspectives global invariants that may serve the solution of some issues in image processing.

The success of these topological approaches is due to the flexibility of the manipulated spaces. Indeed, a simple notion of neighborhood is required. This is transformed into a list of cells and boundaries relationship between them through a boundary operator. Once these data are provided, it is possible to compute several invariants measuring various quantities that are not necessarily scalar as the homology groups. The quantities resulting from the topology describe the shape of the space and give a qualitative representation. These topological invariants associated with their qualitative representation where profitable on the level of image processing. Lifespans of persistent homology classes for example permitted to segment images while visualization of most persistent homology

classes allowed to segment objects in images without the need of prior parameters like sizes or intensities of the objects.

The difficulty of using algebraic topology wasn't focused on the tools and the techniques but on the spatialization of the problem and the interpretation of the invariants. For example in chapter 3, the way of constructions of combinatorial presentation and the filtration depend highly on the studied space and data. The way of forming the simplicial complex in not immediate and linking the computed homology groups to the origin of the problem requires some work. But the gain is important, no previous hypothesis are required if we're starting from raw pixels for example or superpixels as in chapter 3, and no assumptions about the size of the objects or intensities values are required.

The topology is often called "geometry of rubber", but its subfield, the algebraic topology, is providing mathematical tools allowing the passage from the local to the global, that is, to say the passage from a simple local notion of proximity into complex and more general notion of global shape of space. The tools that allow this integration of the local to the global are well performing and can give rise to several algorithms of image understanding and analysis as we have seen in this work. For example, the sheaf theory allowed to transform information from local to global aspects using cohomology of sheaves. The interpretation of the scale analysis and localization permitted to predict and understand the studied data on the global aspect.

Not to forget to mention that even if the results are not perfect for scientists working in image processing field, specially in the biomedical aspect, our work that translates the algebraic topology to the image processing presents a completely novel change of paradigm that still needs more room to discuss and move forward.

Beginning with image segmentation, we proposed a technique that associates the computation of topological features using lifespans of homology classes with statistical features in overlapping windows. These features were classified using the K-means method giving an image segmentation methodology. The results of this methods were presented in an international conference [AGV16a] and published in a journal [AGV16b].

Additionally, the algebraic returns of the most persistent homology classes were used to achieve object segmentation in images. A construction of the filtration scheme on raw pixels allows to segment interesting objects such cells and their components in biomedical images.

Another idea was to use the combinatorial presentation of superpixels in order to segment objects in large images of 2D and 3D without the use of prior parameters. This technique was presented in national conference [AGK17a] and illustrated in an article under review [AGV+17]. Moreover an article on the applied aspect of this technique that permits to find components of cells in biomedical images was accepted for presentation to an international conference [AGK17b].

The movement of cells along sequences of images was at the origin of the development

of a technique that detected and tracked objects from the first to the last frame. This technique uses the relative homology that relaxed the principle of homology to more spaces. The object of this technique was resumed in an article under review [AGK+17c].

Also we have initiated the use of the sheaf theory on images in order to transform information from local to global aspects. Sheaves on cellular spaces and on posets allow to transform algebraic aspect of sheaf to a linear one. This construction permit the cohomological analysis of the sheaf, which allows to better understand and predict the characteristics of spaces. This interpretation is profitable in the scale analysis of increasing spaces and localization in decreasing ones.

5.2 Perspectives

The perspectives that our work open are numerous, since the methods issued from algebraic topology are just beginning to appear. We can quickly turning them into short-term and long-term perspectives.

In the first place, it would be interesting to continue to work on the case of glued objects to try to detect the presence of these two objects using sheaf theory and variations of vertices coordinates, radii of opens and restrictions. Moreover, we can associate our approach on the scale analysis using sheaves theory to separate homology classes that are inside each other in 2D and 3D case. Sheaves cohomology will allow to detect the precise place of cycles inside the opens.

Moreover, the persistent homology that uses only the intensity values have faced problems in detecting glued objects. We intend to use the multidimensional persistence to solve this problem. Even if there isn't any complete topological invariant for multidimensional persistence [CZ09], the authors in [ML15] succeeded to create a tool for visualization of their lifespans. This tool that depends highly on the combinatorial construction must be adapted to the given problem. We have however attempted this approach without real immediate success but a more constructed approach will surely be beneficial. Also there still a big work on ways of computing the homology classes in multidimensional persistence.

In another aspect, it would be important to improve the cycles that segment the objects. The optimization of these cycles may be done by applying another theory from algebraic topology which is discrete Morse theory [Mil63, Koz07]. Following the flow that helped in construction of Morse complex using the discrete vector fields, we can associate critical points to get a contour that may reduce radically the contour that segment the objects.

Not to forget the use of zigzag persistence [CdSM09] that depends on level sets complexes and may studies several problems that may be not covered by usual persistence.

Algebraic topology is not restricted to homology. In the longer term, it would be interesting to develop other parts. For example cohomology, while it's a dual of homology,

studies a space by studying the local coherence of defined functions on the latter. The inconsistencies are therefore very informative. For example, having the information on higher dimensional aspects it's useful to use cohomology to translate information to lower dimensions [dSMVJ11b].

Appendices

Appendix A

Résumé étendu en français

A.1 Chapitre 1: Introduction 152

A.1.1 Contexte et problématique 152

A.1.2 Contributions de la thèse 153

A.1.3 Structure de ce document 154

A.2 Chapitre 2: Topologie, topologie algébrique et applications . 154

A.2.1 Le traitement d'images 155

A.2.2 Panorama des méthodes utilisant la topologie algébrique 156

A.2.3 Représentation combinatoire 156

A.2.4 Groupes d'homologie 157

A.2.5 Homologie relative 159

A.2.6 Conclusion 159

A.3 Chapitre 3: L'homologie persistante et ses applications . . . 160

A.3.1 Filtration 160

A.3.2 Persistance 161

A.3.3 Segmentation d'images utilisant les durées de vie des classes d'homologie 161

A.3.4 Segmentation d'objets en utilisant les classes d'homologie . . . 162

A.3.5 Suivi d'objets en utilisant l'homologie persistante relative . . . 168

A.3.6 Conclusion 170

A.4 Chapitre 4: Théorie des faisceaux et applications sur les images 170

A.4.1 Introduction à la théorie des faisceaux 171

A.4.2 Faisceaux cellulaires 172

A.4.3 Faisceaux sur les posets 174

A.4.4 Applications sur les images 175
A.5 Chapitre 5: Conclusion et perspectives 180
A.5.1 Conclusion générale 180
A.5.2 Perspectives 182

A.1 Chapitre 1: Introduction

A.1.1 Contexte et problématique

Au cours de la dernière décennie, des efforts concertés ont été déployés pour appliquer la topologie algébrique, qui est un champ mathématique abstrait, à des domaines plus concrets. Ces efforts ont motivé les mathématiciens et les scientifiques à appliquer et à développer des concepts s'appuyant sur cette branche des mathématiques. Les applications aux problèmes d'ingénierie ont permis de résoudre de nombreux défis et ont aidé à élargir les liens entre les mathématiques et l'ingénierie. La topologie est souvent considérée comme une théorie difficile et abstraite, mais elle commence à avoir un nombre important d'applications réelles dans un grand nombre de domaines scientifiques, en particulier dans l'analyse des données. Par conséquent, le choix de cette théorie comme un outil pour le traitement des données, notamment des images, est tout à fait justifié. Cependant, il est nécessaire que l'espace topologique des données puisse être représenté de manière combinatoire pour être utilisable par des algorithmes et programmé sur un ordinateur.

L'un des points forts de la topologie algébrique est sa capacité à construire des espaces autour des points qui représentent les données, ce qui la rend très utile pour le traitement des images. En effet, les outils topologiques algébriques fournissent des caractéristiques sur les espaces, qui sont insensibles aux déformations continues. Appliquée aux images, l'analyse topologique pourrait révéler des caractéristiques importantes : combien de composants connectés sont présents, lesquels ont des trous et combien, comment sont-ils liés les uns aux autres, comment transformer information cohérente locale à une information globale, etc.

Certaines tâches de traitement d'images comme la segmentation, le suivi d'objet et la fusion de données sont complexes et limitées par de nombreuses considérations. Choisir la taille d'un objet ou spécifier le niveau pour un seuil peut changer facilement le résultat d'une méthode de traitement d'images. Les variations de l'arrière-plan peuvent conduire à une extraction incomplète des objets saillants. La plupart des méthodes sont incapables de segmenter des objets superposés. En revanche, les approches et les techniques de topologie algébrique ne sont pas affectées par des paramètres a priori, par la variabilité de l'arrière-plan ou par la superposition d'objets, comme c'est généralement le cas des algorithmes existants de traitement d'images.

A.1.2 Contributions de la thèse

Nous proposons dans ce travail des méthodes basées sur la topologie algébrique pour résoudre certains des principaux défis dans le traitement d'images. La topologie algébrique ne va pas remplacer l'utilisation d'autres techniques éprouvées de traitement d'images, mais elle va compléter ou va être associée à ces dernières si nécessaire.

Nos travaux se concentrent sur l'utilisation des concepts de topologie algébrique pour des tâches de traitement d'images. Nous proposons des méthodologies et des approches utilisant d'une part l'homologie persistante, qui est l'un des outils les plus puissants en topologie algébrique, et d'autre part la théorie des faisceaux, qui est une notion complexe mais prometteuse pour traiter certaines applications.

Une approche classique pour segmenter les images consiste à calculer des caractéristiques à l'intérieur de fenêtres glissantes puis à les classer pour obtenir une segmentation de l'image. Nous allons voir que la topologie algébrique peut apporter d'autres caractéristiques qui sont plus pertinentes et qui permettent d'améliorer la qualité de la segmentation d'images naturelles.

D'autre part, les algorithmes à seuillage ou les algorithmes adaptatifs utilisant des concepts d'arbres sont souvent utilisés pour la segmentation d'objets. Ces méthodes n'arrivent pas, et cela dans de nombreux cas, à identifier uniquement les objets intéressants. Le bruit est souvent détecté et le résultat dépend du niveau de l'arrière-plan qui fausse la détection des objets superposés. En outre, elles dépendent fortement des paramètres a priori comme le volume ou la taille d'objets à identifier. Puisque la topologie algébrique étudie la présence de trous et des vides en utilisant des invariants puissants, qui sont les groupes d'homologie, nous allons voir qu'elle est capable de construire des cycles (ou des classes d'homologie) qui ne sont pas sensibles aux variations de l'arrière-plan et qui ne dépendent pas des paramètres a priori. Ces méthodes permettent de segmenter les objets intéressants, y compris des objets superposés.

La détection et le suivi d'objets sont généralement considérés comme l'un des principaux défis dans le traitement et l'analyse d'images. La plupart des techniques existantes sont des méthodes spécifiques basées principalement sur des algorithmes complexes contrôlés par de nombreux paramètres et métriques. Puisque les complexes topologiques peuvent être construits directement sur les pixels d'une image, aussi bien 2D que 3D, ainsi que sur des séquences d'images 2D+t, la topologie algébrique est une des solutions aux défis actuels en traitement d'images. La version relative de l'homologie permet de détecter le mouvement des objets sans l'utilisation de paramètres a priori.

Par ailleurs, un concept totalement nouveau dans l'association de la topologie algébrique au traitement d'images est l'utilisation de la théorie des faisceaux. L'utilisation et les applications de cette théorie ne sont qu'à leur début et le domaine du traitement d'images pourra en tirer profit. Il est donc utile d'initier la mise en œuvre de ces notions dans le contexte du traitement d'images notamment à travers l'analyse cohomologique.

A.1.3 Structure de ce document

Ce mémoire est organisé de la façon suivante.

Dans le chapitre suivant "Topologie, topologie algébrique et applications", nous faisons une brève description de certaines techniques de traitement d'images et de leurs limites. Ensuite, nous détaillons des notions basiques de topologie et surtout la topologie algébrique en expliquant brièvement comment on peut les utiliser en traitement d'images. Le calcul des groupes d'homologie est ensuite expliqué puis on termine par les tâches de traitement d'images auxquelles nous sommes intéressés par ce travail.

Le chapitre "Homologie persistante et applications aux images" présente les principales contributions. Plusieurs méthodologies de constructions de la topologie algébrique sur des images sont proposées. Nous commençons par la présentation de l'homologie persistante et de son calcul, puis nous expliquons comment transformer des images en représentations combinatoires. Ensuite, nous proposons de nouvelles méthodes de segmentation d'images et de segmentation d'objets multidimensionnels. Nous montrons que les complexes topologiques peuvent être construits directement sur les pixels d'une image, aussi bien 2D que 3D, mais également sur des superpixels, ce qui permet de réduire le temps de calcul et les ressources nécessaires. Nous terminons ce chapitre en présentant une nouvelle méthode qui permet de suivre des objets en mouvement et qui utilise l'homologie relative.

Nous introduisons des concepts issus de la théorie des faisceaux dans le chapitre "La théorie des faisceaux et ses applications d'images". D'abord, nous expliquons le concept de la théorie des faisceaux, puis nous expliquons une méthodologie de fusion de données à l'aide de faisceaux. Nous montrons comment nous pouvons associer ces concepts à des tâches de traitement d'images. Nous finissons ce chapitre en montrant comment on peut utiliser ces notions dans le traitement et l'analyse d'images : la construction des sections sur les images couleur, l'analyse d'échelle et la localisation.

Le dernier chapitre représente la conclusion de ce travail et présente un plusieurs perspectives à court et à long terme. Celles-ci sont liées à la continuité de ce travail, mais concernent également l'utilisation d'autres aspects de la topologie algébrique, en particulier la théorie de Morse. Nous pouvons en effet tirer profit de la persistance multidimensionnelle pour intégrer de nombreux facteurs dans la construction de la filtration.

A.2 Chapitre 2: Topologie, topologie algébrique et applications

Nos travaux s'intéressent principalement au développement d'outils issus de topologie algébrique pour réaliser des tâches de traitement d'images. C'est pourquoi ce chapitre commence par une étude rapide de l'état de l'art des techniques de traitement d'images. Nous présentons ensuite les outils employés par la topologie et la topologie algébrique dans les problèmes scientifiques. Des explications détaillées sur la théorie de l'homologie

et le calcul des groupes d'homologie sont présentées en donnant quelques pistes en ce qui concerne leurs applications.

A.2.1 Le traitement d'images

Le domaine du traitement d'images comporte beaucoup de tâches. La segmentation d'images est une tâche difficile qui a été considérée comme une étape clé dans le traitement de l'image. Elle reste un problème de longue date dans le domaine avec de nombreux travaux. Le lecteur pourra consulter [MC15, ZMC16] pour des analyses approfondies sur ce sujet. L'objectif de la segmentation d'images est de partitionner une image en régions homogènes non superposées de manière à localiser des objets d'intérêt dans l'image. De nombreuses raisons font que la segmentation d'images est une tâche difficile. La complexité des algorithmes utilisés dépend de nombreux paramètres qui contrôlent les algorithmes de segmentation et qui causent le manque de solutions génériques " sur étagère". Les techniques et les solutions proposées dépendent énormément de l'application visée.

Une grande variété de techniques et de méthodes de segmentation a été discutée et développée dans la littérature. Les approches classiques sont principalement basées sur des méthodes mathématiques ou statistiques. Selon une autre classification, nous trouvons des approches de classification (le clustering) et des techniques de soft computing.

La détection et le suivi d'objets sont considérés comme une tâche spécifique du traitement d'images. La plupart des techniques existantes ne sont pas génériques et les méthodes existantes sont basées principalement sur des algorithmes complexes contrôlés par de nombreux paramètres et métriques. La "taille unique" universellement appropriée aux méthodes de suivi n'existe pas selon l'étude faite dans [CSdC+14].

Nous notons que ces méthodes sont dépendantes des paramètres a priori, de la variabilité de l'arrière-plan d'une image, de la superposition des objets, etc. Par ailleurs, ces méthodes sont en général complexes et relativement imprécises. Prenons l'exemple de la méthode OTSU qui est une méthode de seuillage global robuste. Nous appliquons cette méthode sur une image synthétique. Nous remarquons que cette méthode n'est pas en mesure de segmenter certains objets dans l'image, comme celui en bas à gauche ou les petits objets qui sont superposés sur l'objet le plus grand comme en bas à droite. De plus, cette méthode dépend du niveau de l'arrière-plan, les objets n'étant pas extraits correctement comme on le voit en bas à gauche.

Pour contourner ces problèmes, nous proposons dans cette thèse des méthodes basées les concepts de topologie algébrique. Cette théorie est indépendante des paramètres tels que le volume ou les valeurs d'intensité, et ne demande pas d'étapes de prétraitement. Cette théorie permet également de construire des outils invariants à des déformations continues. Tous ces critères rendent les méthodes utilisant les concepts de topologie algébrique aptes à réaliser des tâches plus génériques que les méthodes existantes.

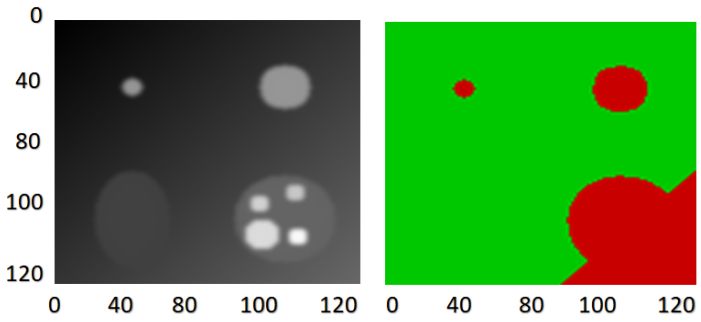


Figure A.1: Une image synthétique et sa segmentation à l'aide de la méthode OTSU

A.2.2 Panorama des méthodes utilisant la topologie algébrique

Plusieurs méthodes topologiques ont été proposées dans la littérature. Certaines méthodes utilisées dans l'étude des espaces topologiques sont données par la topologie algébrique [Mas91, Hat01]. La topologie algébrique, en tant que topologie, étudie les propriétés globales des espaces, mais utilise des objets algébriques tels que des groupes et des anneaux pour répondre à des questions topologiques. Alors que les méthodes topologiques générales sont concentrées sur la connectivité et les connexions entre les espaces, les méthodes de topologie algébrique sont plus concrètes. La topologie algébrique essaie de transformer un problème topologique en un problème algébrique plus facile à résoudre et à calculer.

Chaque espace peut être associé à un groupe appelé un groupe d'homologie. Nous pouvons distinguer par exemple le tore et la bouteille de Klein les uns des autres parce qu'ils ont différents groupes d'homologie. La structure combinatoire des espaces est souvent utilisée par la topologie algébrique pour calculer les différents groupes associés à ces espaces.

La topologie algébrique récente a émergé et contribué dans de nombreuses applications du monde réel [Ghr14]. Concernant les applications des méthodes et des outils issus de la topologie algébrique, celles-ci consistent à associer quelques structures algébriques discrètes comme les classes d'homologie ou des faisceaux à des espaces topologiques tels que les complexes cellulaires. Ces complexes peuvent être construits sur un ensemble de pixels ou sur un nuage de points afin de comprendre les problèmes de connectivité dans n'importe quelle dimension telle que le nombre de trous, de vides, de tunnels, etc.

A.2.3 Représentation combinatoire

L'homologie est un moyen de mesurer la connectivité d'un espace pour différentes dimensions notamment pour la découverte et l'analyse des "trous". Cela implique d'imposer une structure algébrique sur un espace topologique construit.

Il existe de nombreux types de complexes cellulaires : les complexes cubiques, dans le cas d'utilisation de pixels d'une image, les complexes simpliciaux (qui représentent des triangulations de l'espace) dans le cas des superpixels, etc. Nous n'allons pas fournir des

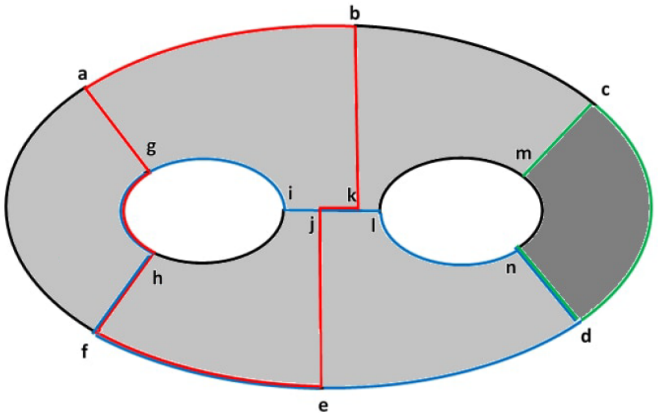


Figure A.2: Exemple d'un complexe cellulaire.

explications détaillées sur les types de complexes cellulaires et leurs constructions préférant rediriger le lecteur vers [Hat01]. Dans la suite de ce mémoire, les complexes considérés sont soit cubiques soit simpliciaux. Ils sont alors composés de : points qui peuvent être vus comme des cellules de dimension zéro, d'arrêts ou de côtés qui peuvent être vus comme des cellules de dimension un, ainsi que des carrées ou des triangles qui peuvent être vus comme des cellules de dimension deux. La structure topologique d'un espace combinatoire X est algébriquement encodée par des espaces vectoriels et par des applications linéaires qui sont associés sous la forme d'un complexe. En effet, nous pouvons définir l'espace vectoriel $C_p(X)$ des p-chaines comme étant celui des séries formelles des cellules de dimension p, avec des coefficients dans $\mathbb{Z}_2$, et munis d'une opération d'addition naturelle. Les relations entre les chaînes de différentes dimensions sont assurées par les opérateurs de bords. L'opérateur de bord $\partial_p : C_p(X) \to C_{p-1}(X)$ entre $C_p(X)$ et $C_{p-1}(X)$ est une application linéaire qui associe à une k-chaîne la somme des chaînes aux bords de chaque cellule de dimension inférieure. Par exemple, à un triangle est associée la somme de ses trois arêtes. Par construction, les opérateurs de bords satisfont $\partial_p \partial_{p+1} = 0$. Le complexe de chaîne est la séquence $C_p(X)$ avec les applications ∂_k:

$$0 \xrightarrow{\partial_{p+1}} C_p \xrightarrow{\partial_p} C_{p-1} \xrightarrow{\partial_{p-1}} \dots \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} 0. \tag{A.1}$$

A.2.4 Groupes d'homologie

Pour d'autres dimensions supérieures, l'homologie est un moyen naturel pour découvrir des trous p-dimensionnels dans un complexe cellulaire. L'idée est de trouver des chaînes qui entourent les trous sans être capables d'être réduites continuellement à zéro. Notant, pour exemplifier, B, R et G comme trois chaînes de dimensions 1 sur la figure A.2 qui sont illustrées par des lignes bleues, rouges et vertes, respectivement. Tout d'abord, une chaîne qui entoure un sous-espace ou un trou est nécessairement sans bords. Les p-chaînes sans bords sont intéressantes et forment un sous-groupe de C_p que nous appelons le groupe

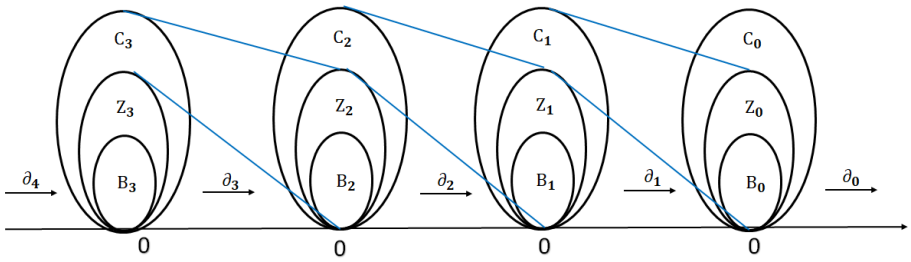


Figure A.3: Les groupes des cycles et des bords.

de p-cycles Z_p. L'ensemble Z_p de tous les p-cycles est défini comme le sous-espace de C_p de chaînes sans bords :

$$Z_p = \{x \in C_p \mid \partial_p x = 0\} = \ker \partial_p. \tag{A.2}$$

Parmi ces cycles, nous considérons ceux qui sont les bords d'autres chaînes de dimensions supérieures tout en restant à la surface. Cela signifie intuitivement qu'ils peuvent être réduits en un point. Ils forment un sous-groupe appelé le groupe p-bords B_p.

$$B_p = \{x \in C_p \mid \exists y \in C_{p+1}, \partial_{p+1} y = x\} = \operatorname{Im} \partial_{p+1}. \tag{A.3}$$

Par exemple, la chaîne en vert $G = (cd) + (dn) + (nm) + (mc)$ affichée sur la figure A.2, est un 1 -cycle parce que $\partial G = 0$. Les chaînes B et R dans la figure A.2 sont également des 1-cycles. Ainsi, elles appartiennent toutes à Z_1. Le cycle G est une limite de la chaîne 2 $(cdnm)$, la surface sombre dans la figure A.2, et il appartient à B_1.

Puisque le bord d'un bord est nul, B_p est un sous-groupe de Z_p, donc un groupe quotient peut être créé. La figure A.3 montre schématiquement la relation entre les différents espaces vectoriels concernés par l'homologie. Le but de l'homologie est de distinguer les cycles qui sont aussi des bords parce qu'ils ne contiennent pas des trous et ne peuvent donc pas être réduits à zéro.

Les groupes d'homologie H_p conservent le nombre de cycles essentiels qui sont intéressants en distribuant tous les cycles en classes équivalentes. Ainsi, un élément de H_p regroupe ces cycles équivalents qui peuvent être déformés continuellement l'un sur l'autre et une classe de H_p sera représentée par un seul cycle.

Algébriquement, deux cycles z_1, $z_2 \in Z_p$ sont dites homologues ou équivalentes, écrites $z_1 \sim z_2$, s'ils diffèrent par un bord, c'est-à-dire $z_1 - z_2 \in B_p$. Nous disons que z_1 et z_2 appartiennent à la même classe $[z]$. Nous laissons $[z]$ dénoter la classe d'homologie de $z \in Z_p$ et définissons l'homologie de dimension p d'un espace par le quotient de l'espace vectoriel Z_p par le sous-espace B_p, ce qui est un ensemble des classes d'homologie:

$$H_p = Z_p / B_p = cycles/bords = \{[z], z \in Z_p\}. \tag{A.4}$$

Tous les calculs impliquant des groupes d'homologie peuvent être facilement réalisés en utilisant des manipulations matricielles standard à travers une représentation combinatoire du complexe de chaînes.

A.2.5 Homologie relative

Il est parfois utile de ne pas prendre en compte un sous-complexe A de l'espace X pour calculer les groupes d'homologie. Nous parlons d'une autre version de l'homologie impliquant la réduction d'un sous-complexe A qui aboutit à l'homologie relative $H_p(X, A)$.

Il est nécessaire alors de réduire complètement le sous-espace A pour qu'il "disparaisse" lors du calcul. La relation de sous-partie $A \subset X$ est représentée par une relation de sous-complexe de $C_p(A)$, qui est le complexe représentant l'espace A à la dimension p, et le complexe $C_p(X)$ qui est la linéarisation de X.

Pour cela, nous notons $C_p(X, A) = C_p(X)/C_p(A)$ comme le complexe quotient et $\overline{\partial}_p$ comme l'opérateur de bord, en prenant en compte que A est toujours un sous-complexe de X. Réduire $C_p(A)$ en $C_p(X)$ revient à considérer que les chaînes de $C_p(A)$ sont nulles. Les espaces vectoriels quotients $C_p(X)/C_p(A)$ jouent ce rôle. Les éléments de ces espaces vectoriels sont des chaînes de $C_p(X)$ où les cellules de $C_p(A)$ ne sont pas prises en compte. Ainsi, certains cycles peuvent apparaître dans $C_p(X)/C_p(A)$ alors qu'ils n'existaient pas dans $C_p(X)$ et tous les cycles complètement inclus dans $C_p(A)$ disparaissent. Par conséquent, prendre des chaînes sur X modulo chaînes sur A réduit l'exigence d'une chaîne pour être appelé un cycle, à chaque fois que son bord est contenu dans A.

Cela inclut également le cas où le bord est vide, ce qui peut être détecté aussi par l'homologie absolue. De la même manière que le p-bords, p-bords relatifs sont définis comme $B_p(X, A) = \operatorname{Im} \overline{\partial}_{p+1}$. De même, les p-cycles relatifs sont $Z_p(X, A) = \ker \overline{\partial}_p$ et correspondent à p-chaînes c_p qui satisfait $\partial_p c_p \in C_{p-1}(A)$ ou $\partial_p c_p = 0$.

Les groupes d'homologies relatives $H_p(X, A)$ sont calculés comme les groupes d'homologie qui utilisent ces nouveaux espaces vectoriels $C_p(X)/C_p(A$. Comme dans le cas absolu de l'homologie, nous avons $B_p(X, A) \subset Z_p(X, A)$ et l'homologie relative de dimension p est définie par $H_p(X, A) = Z_p(X, A)/B_p(X, A)$.

A.2.6 Conclusion

Dans ce chapitre, nous avons fait un tour d'horizon des techniques de traitement d'images qui existent dans la littérature. Ensuite nous nous sommes intéressés à la topologie algébrique. Avant de développer les concepts d'homologie, nous avons expliqué comment représenter de manière combinatoire des ensembles de points. Puis, nous avons expliqué cette théorie dans sa version absolue et relative et la façon de calculer les groupes d'homologie.

Nous avons vu que l'homologie est un outil efficace pour extraire des caractéristiques topologiques des espaces statiques. Mais que faire si ces caractéristiques ne sont pas d'une importance majeure pour déduire les inférences dans les images ? Et si nous voulions augmenter l'espace topologique où l'homologie, comment fait-on ? C'est pourquoi nous présentons dans le chapitre suivant l'homologie persistante qui est un moyen de détecter la persistance des classes d'homologie face aux variations d'espaces topologiques. Nous allons voir comment cet outil est adapté pour effectuer des tâches de traitement d'images.

A.3 Chapitre 3: L'homologie persistante et ses applications

Dans ce chapitre, nous étendons la théorie de l'homologie à une phase appropriée pour la compréhension et l'inférence des données dans l'analyse d'images. Cette phase consiste à calculer les variations d'homologie au cours des modifications d'espaces topologiques par une procédure appelée filtration. Nous parlons d'homologie persistante où l'objectif est de détecter les classes d'homologie persistantes pendant les variations dans les espaces topologiques.

L'importance de cette procédure repose sur le concept que les caractéristiques topologiques détectées sur un intervalle d'échelle variable sont plus appropriées pour représenter les caractéristiques des données étudiées. Elles sont insensibles au bruit et ne nécessitent pas un choix particulier de paramètres. Les applications de l'homologie persistante dépendent fortement de la construction des complexes cellulaires. Nous montrons dans ce chapitre différentes applications de la théorie de l'homologie dans le traitement d'images, notamment en ce qui concerne la segmentation d'images, la détection d'objets et le suivi des objets.

Grâce à la notion de filtration et à la notion de l'homologie, il est possible de suivre l'évolution des "trous" dans une séquence d'espace à travers l'homologie persistante.

A.3.1 Filtration

Une filtration est simplement une suite finie d'espaces topologiques imbriqués :

$$X_0 \subset \ldots \subset X_i \subset \ldots \subset X_j \subset \ldots \subset X. \tag{A.5}$$

La construction d'une filtration est souvent réalisée par une fonction numérique $f : X \to \mathbb{R}$. Dans ce cas la droite réelle $\mathbb{R}$ est partitionnée en intervalles $]t_{i-1}, t_i]$ et les espaces X_i de la filtration sont les ensembles de sous-niveaux, les hypographes, $X_i = f^{-1}\big(]-\infty, t_i]\big)$. Ainsi, la condition $X_i \subset X_{i+1}$ est naturellement vérifiée.

Dans le cas d'un espace X combinatoire comme un complexe simplicial ou cubique, la fonction f est représentée comme un poids associé à chaque cellule, par exemple à la valeur

de pixel. Les hypographes X_i sont alors l'ensemble des cellules de poids inférieurs à un seuil t_i. Toutefois, il faut que les X_i soient aussi des sous-complexes de X pour pouvoir utiliser les groupes d'homologie. Cette contrainte est vérifiée si les poids des cellules de dimension k sont supérieurs à ceux de leur bord. Cette condition sera mise en œuvre dans la suite en fixant le poids d'une cellule de dimension k comme étant le maximum des poids de ses bords. Ce faisant, la fonction f est complètement définie sur X à partir des valeurs associées aux sommets.

A.3.2 Persistance

En suivant l'évolution topologique d'une filtration via l'homologie, on obtient une séquence de groupes d'homologie qui sont reliés par des applications linéaires induites par les inclusions :

$$H_k(X_0) \to \ldots \to H_k(X_i) \to \ldots \to H_k(X). \tag{A.6}$$

On peut calculer l'homologie $H_k(X_i)$ pour tous les niveaux i afin de connaitre les classes d'homologie. Mais nous perdons alors l'information concernant l'évolution de chaque classe particulière. Or, la mesure de la "durée de vie" des classes d'homologie lors de la filtration est plus riche. En effet, au cours de la filtration, l'addition d'une cellule de dimension k peut modifier la topologie selon deux scénarios. Soit il "remplit" un trou de dimension $k-1$ soit il en "crée" un de dimension k. Ainsi, les "trous" apparaissent et disparaissent pendant la filtration. Des dates d'apparition et de disparition peuvent alors leur être associées. Celles-ci sont calculables grâce à la suite de groupe d'homologies (A.6) par des algorithmes dédiés [EH10, Zom10a].

Les classes d'homologie ayant une grande durée de vie indiquent la présence de phénomènes topologiques intéressants, tandis que celles de courtes durées de vie sont vues comme du bruit topologique. L'évolution des classes d'homologie peut être visualisée à l'aide d'un diagramme de persistance. Une classe d'homologie y est représentée par un point dont l'abscisse fournit la date d'apparition et l'ordonnée celle de sa disparition. Par conséquent, sa distance à la diagonale indique la durée de vie. Les points éloignés de la diagonale représentent des objets topologiquement significatifs tandis que ceux qui sont proches sont considérés comme du bruit. Ce mécanisme est renforcé par la stabilité de diagramme de persistance sous les transformations continues de l'image [CSEH07].

A.3.3 Segmentation d'images utilisant les durées de vie des classes d'homologie

Pour la segmentation, nous analysons les images avec des fenêtres glissantes superposées. Pour chaque fenêtre nous construisons un complexe cubique dont les sommets sont les pixels, les côtés relient les pixels voisins, tandis que les carrées complètent 4 côtés voisins. Les poids des sommets pour la filtration sont donnés par le niveau de gris, tandis que les

côtés et les carrées portent la valeur maximale de leurs sommets et côtés respectivement. Dans chaque fenêtre on calcule ensuite plusieurs caractéristiques classiques qui sont la moyenne et la variance des niveaux de gris, auxquelles s'ajoutent des caractéristiques topologiques comme la moyenne et la variance des durées de vie des trous de dimensions 0 et 1 ainsi que leurs entropies persistantes. Cette dernière est définie dans [MRS15] pour chaque dimension par $-\sum_{i\in I} p_i \log p_i$, où I représente les intervalles des durées de vie, $p_i = l_i/L$, l_i= date de disparition - date d'apparition et $L = \sum_{i\in I} l_i$. Ces 8 caractéristiques forment des vecteurs associés à chaque fenêtre.

Ensuite, l'ensemble de ces vecteurs caractéristiques est classé en N classes. Pour simplicité, nous avons considéré une classification non supervisée par la méthode k-moyennes, d'autres méthodes supervisées ou non étant envisageables. Chaque fenêtre est alors classée dans une classe, ce qui permet de réaliser une segmentation de l'image.

La figure A.4 montre le résultat obtenu sur une image biomédicale. Nous avons traité l'image avec des fenêtres carrées glissantes qui se chevauchent. La taille de la fenêtre a été choisie à 50×50 pixels et le chevauchement à 10 pixels. Ensuite, les fenêtres sont classées en utilisant l'algorithme k-moyennes à 4 classes. Il faut noter que ces mesures peuvent varier en fonction de la construction et de la nature de l'application. La segmentation de la glande montre quatre classes correspondant aux principaux types de zones de tissus.

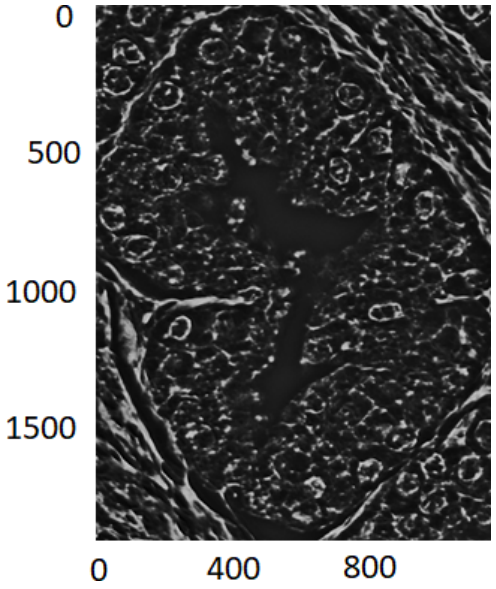
Nous montrons dans la figure A.4 (b) les résultats de la segmentation de la glande en utilisant seulement les caractéristiques statistiques tandis que la figure A.4 (c) montre le résultat de la segmentation en utilisant les 6 caractéristiques topologiques associées avec les 2 statistiques. Par exemple, on voit que les patches de la classe représentée en blanc ne font pas de distinction entre la bordure du tissu et ses cellules en utilisant les caractéristiques statistiques alors que cette distinction est respectée en fonction des caractéristiques topologiques associées aux caractéristiques statistiques.

A.3.4 Segmentation d'objets en utilisant les classes d'homologie

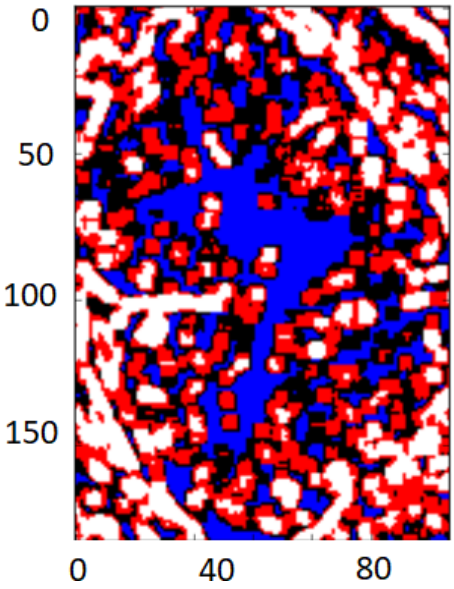
Cette sous-section démontre que la méthodologie surlaquelle nous avons basé notre approche est efficace en construisant des complexes topologiques sur des pixels. Cette méthodologie a montré son efficacité dans le cas d'images pré-segmentées en superpixels. Une combinaison entre la construction topologique méthodologique et l'image des superpixels peut être exécutée avec succès pour atteindre la segmentation visée.

A.3.4.1 Segmentation basée sur un complexe cubique de pixels

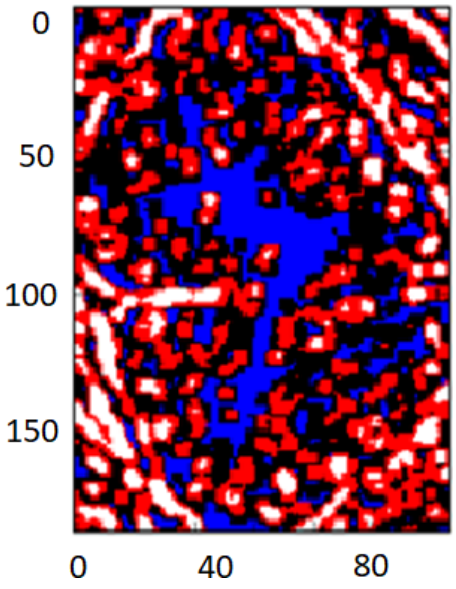
Considérant notre méthode, un complexe topologique est d'abord construit directement sur les pixels de l'image affichée dans la figure A.5, comme expliqué précédemment. Ensuite le schéma de la filtration est construit. Points, côtés et cellules de dimensions 2 sont ajoutés au complexe quand l'intensité augmente jusqu'à ce que nous obtenons le complexe



(a) Une image biomédicale.



(b) Segmentation de l'image utilisant des caractéristiques statistiques.



(c) Segmentation de l'image utilisant des caractéristiques statistiques et topologiques.

Figure A.4: Segmentation d'une image biomédicale.

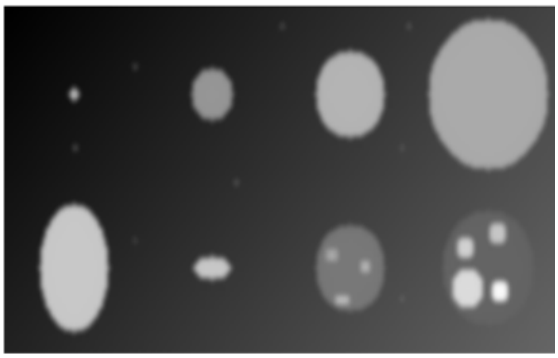


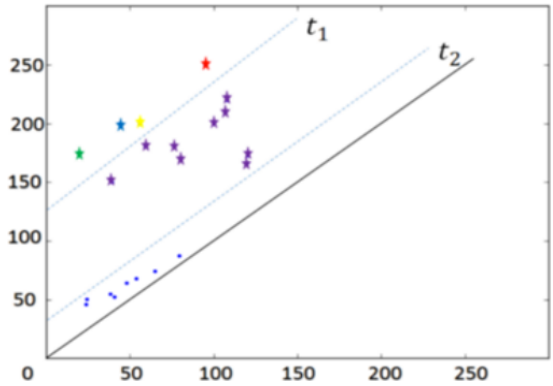
Figure A.5: Une image synthétique.

entier. Le calcul de l'homologie persistante sur ce complexe donne le diagramme de persistance des classes d'homologie. Ce diagramme de persistance, montré sur la figure A.6 (a) est le résultat principal de notre méthode, car il indique les naissances et la disparition de classes d'homologie de première dimension. Chaque point porte toutes les informations nécessaires aux classes d'homologie. L'importance des points et ce qu'ils représentent est proportionnelle à leur distance de la diagonale. Ainsi, les points proches de la diagonale ont une petite durée de vie, ce qui signifie qu'ils correspondent au bruit, tandis que ceux qui sont loin sont intéressants.

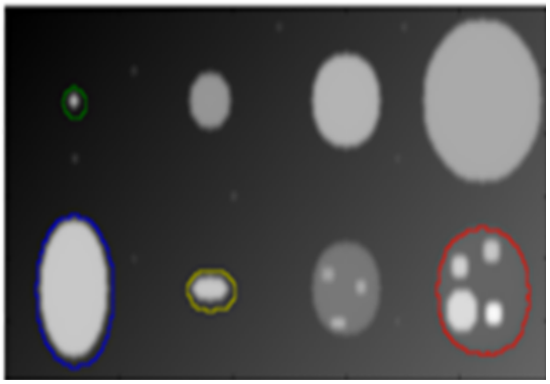
Il faut noter qu'ayant la construction du complexe, nous pouvons trouver pour chaque point dans le diagramme de persistance les coordonnées x-y de la classe correspondante. Par exemple, la classe d'homologie représentée par un * dans la figure A.6 (a) correspond à la classe qui identifie l'objet représenté en bleu dans la figure A.6 (b), la classe d'homologie représentée avec * correspond à l'objet en rouge, etc. Les classes d'homologie dans le diagramme de persistance permettent ensuite de sélectionner les objets intéressants en imposant un seuil parallèle à la première diagonale. Par exemple, les classes d'homologie qui représentent des points au-dessus de la parallèle (t_1) sont mises en évidence dans la figure A.6 (b) par leurs couleurs correspondantes. Imposer une droite parallèle à la diagonale au niveau (t_2) permettra alors de mettre en évidence d'autres classes d'homologie qui sont représentées par tous les points au-dessus de (t_2) comme dans la figure A.6 (c).

A.3.4.2 Segmentation basée sur complexe cellulaire de superpixels

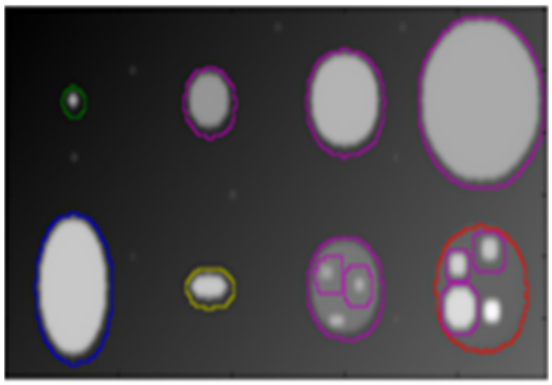
Autre que sa forme de grille de pixels habituelle, une image peut être simplifiée par un ensemble de superpixels. En effet, nous pouvons regrouper des pixels voisins s'ils partagent un même critère (par exemple de luminance) et ainsi réduire la dimension de l'espace de départ. Chaque superpixel est représenté par un sommet qui est son centre. Nous construisons notre complexe topologique sur ces superpixels. Ainsi, nous considérons deux superpixels comme voisins si un pixel de l'un est adjacent à un pixel de l'autre et cela en analysant les 8 voisins de chaque pixel. Nous relions les deux sommets des superpixels par un côté tandis que 3 côtés adjacents forment un triangle. Ensuite, nous attribuons la



(a) Diagramme de persistance de dimension 1.

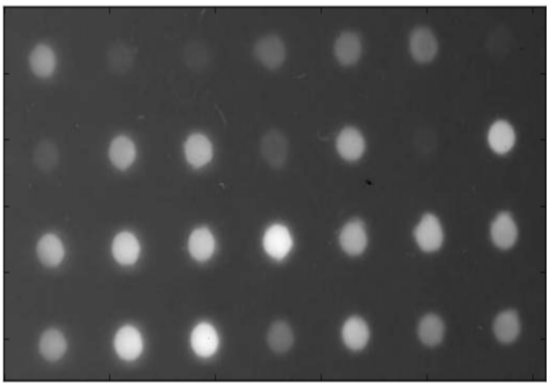


(b) Résultats utilisant le seuil a posteriori t_1.

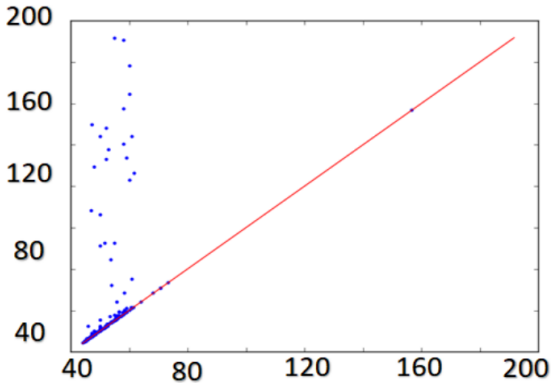


(c) Résultats utilisant le seuil a posteriori t_2.

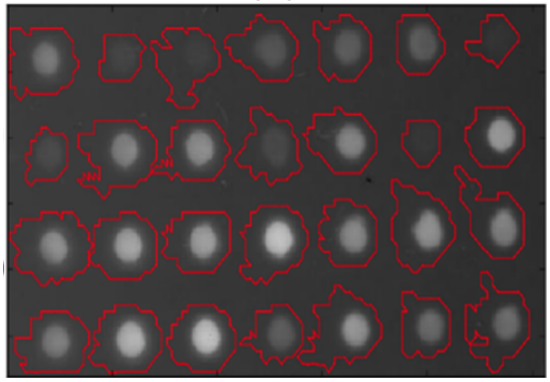
Figure A.6: Résultats de méthode proposée sur l'image synthétique.



(a) L'image de points.



(b) Diagramme de persistance de dimension 1.



(c) Résultats de la segmentation.

Figure A.7: L'image des points et les résultats de la segmentation.

moyenne des valeurs de pixels contenants dans chaque superpixel S_i pour les sommets et nous construisons le complexe simplicial. En suivant les étapes de calcul de l'homologie persistante, nous obtenons les classes d'homologie de dimension 1 qui apparaissent et disparaissent tout au long de filtration. La mise en évidence des classes qui ont la plus grande durée de vie permet de détecter des objets dans l'image.

Nous illustrons cette méthode sur une image test de 513×282 pixels comportant 24 pièces de taille, texture et niveaux de gris moyen différents comme le montre la figure A.7. On calcule l'homologie persistante après la pré-segmentation en 5000 superpixels en utilisant la technique appelée SLIC décrite dans [ASL+12]. Nous réduisons donc l'espace de départ de $513 \times 282 = 144666$ pixels à 5000 superpixels. La technique SLIC génère des superpixels d'une manière plus rapide que les autres méthodes existantes, est plus efficace en termes de mémoire nécessaire et dépasse les autres en ce qui concerne l'adhérence des bords. Les points éloignés de la diagonale désignent les 24 pièces de monnaie. Les chaînes qui représentent ces classes sont colorées en rouge, ce qui permet de détecter les pièces dans la figure A.7. Le résultat obtenu est

Afin de démontrer l'aspect multidimensionnel de notre méthode, nous présentons sur la figure A.8 une segmentation d'une image biomédicale 3D en niveaux de gris en utilisant

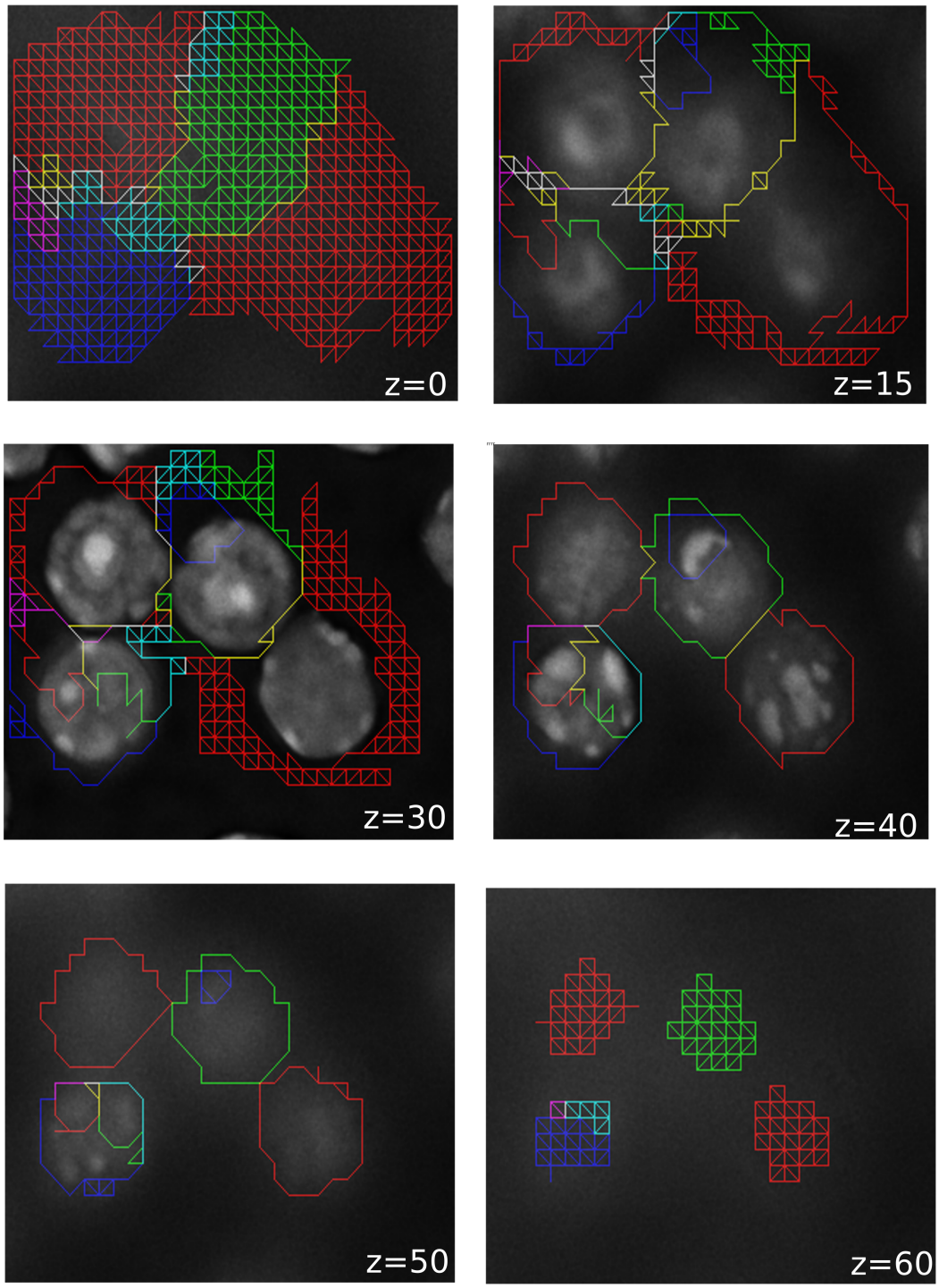


Figure A.8: Résultats de segmentation pour une image 3D.

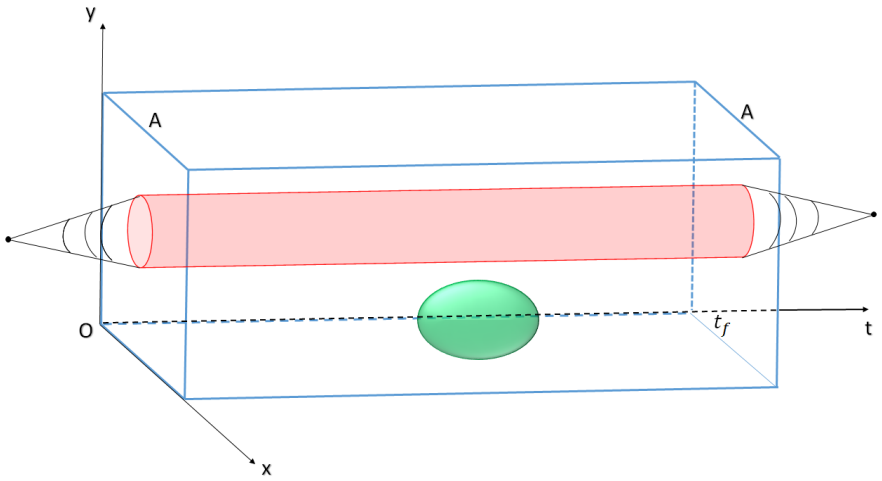


Figure A.9: La détection de “trou de ver” et de la sphère par homologie relative.

la combination de la construction sur des superpixels l'homologie persistante. Cette figure représente les résultats de la segmentation d'une image biomédicale de dimensions 61 × 249 × 308 acquise à différents instants : z = 0, 15, 30, 40, 50 et 60 respectivement. Les classes d'homologie de deuxième dimension sont des chaînes de 2 cellules ou de triangles ($S_iS_jS_k$). Nous avons mis en évidence sept classes d'homologie les plus persistantes formées par 2 chaînes ou sommes de triangles.

A.3.5 Suivi d'objets en utilisant l'homologie persistante relative

La détection et le suivi d'objets sont généralement considérés comme une des tâches majeures et difficiles dans le domaine du traitement et de l'analyse d'images.

Comme nous travaillons avec des images 2D + t en niveaux de gris, nous devons modéliser nos constructions sur un concept basé sur 3D puisque nous avons besoin des voisins temporels de chaque voxel. Nous allons travailler sur un complexe cubique que nous appelons abusivement le complexe *voxel*.

Les données de la séquence d'images sont considérées comme une fonction d'un domaine $D \subset \mathbb{N}^3$ dans l'espace des nombres réels $\mathbb{R}$ i.e. $f : D \to \mathbb{R}$ où $D = \{(x, y, t)|0 \leq x <$ largeur$, 0 \leq y <$ hauteur$, 0 \leq z < t_f\}$, où t_f représente le nombre de cadres 2D dans la séquence.

Pour le cas de 2D+t, le complexe de chaines est:

$$\emptyset \to C_3 \xrightarrow{\partial_3} C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \xrightarrow{\partial_0} \emptyset. \tag{A.7}$$

Prenant des chaînes sur X modulo chaînes sur A réduit l'exigence d'une chaîne à être appelée un cycle, à chaque fois que son bord est contenu dans A. Ceci est illustré dans la

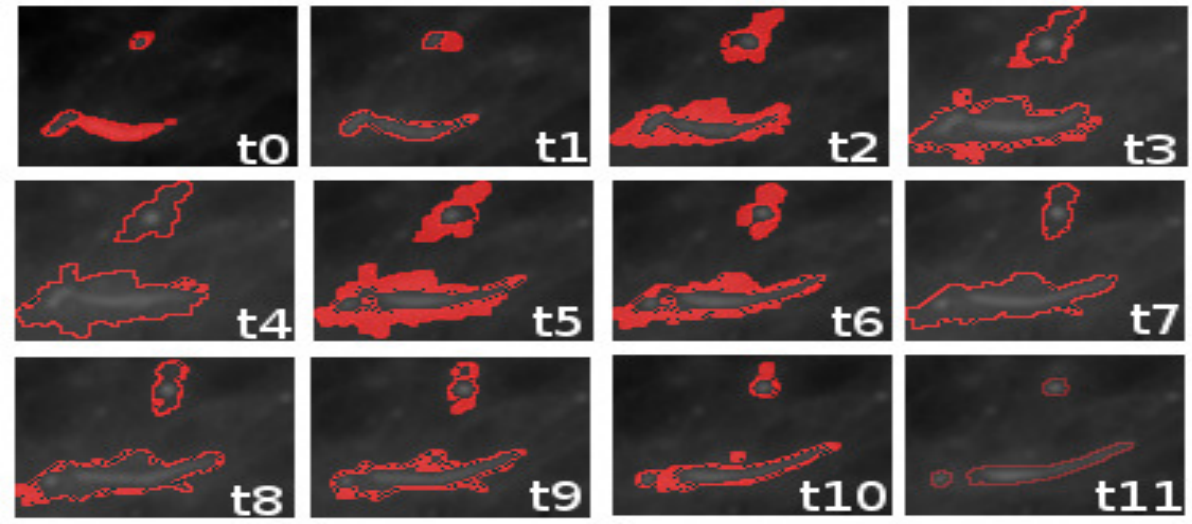


Figure A.10: Résultats de l'homologie relative persistante sur une 2D+t image réelle biomédicale.

figure A.9, où A représente le sous-espace contenant toutes les cellules qui appartiennent à $t = 0$ et t_f. Comme nous le voyons, la frontière du cylindre rouge qui appartiennent à A s'effondre à un point depuis que nous travaillons sur les groupes de quotients $C_p(X)/C_p(A)$. Ce concept permet de détecter des objets dans tous les cadres de l'image comme le "trou de ver" illustré par le cylindre rouge. Cela inclut également le cas où le bord est vide, comme l'objet en vert, qui peut être détecté aussi par l'homologie absolue.

La mise en évidence de la base des éléments de H_2 permet de détecter des objets intéressants dans la séquence d'images. En suivant l'évolution topologique de cette filtration en utilisant l'homologie relative, nous obtenons une séquence de groupes d'homologie reliés par des cartes linéaires induites par inclusions pour toute dimension p:

$$\begin{aligned} H_p(K_0, A) \to H_p(K_1, A) \to \ldots \to \\ H_p(K_i, A) \to \ldots \to H_p(K, A). \end{aligned} \tag{A.8}$$

Comme application réelle, nous considérons une image biomédicale prise par une technique dite de suivi temporel utilisant le système d'imagerie quantitative de phase SID4Bio décrite dans [BMW09]. L'objectif ici est de détecter les vésicules qui se déplacent de premier au dernier cadre.

En utilisant la méthode de détection d'objet d'homologie relative persistante, nous sommes capables d'identifier une vésicule en mouvement (en haut) et le petit train de vésicules (en bas) qui augmente en taille et se déplace du premier cadre de la séquence au dernier, comme indiqué sur A.10.

A.3.6 Conclusion

Dans ce chapitre, nous avons commencé par expliquer le concept de l'homologie persistante à partir de la transformation d'un ensemble des points dans les complexes cellulaire jusqu'au calcul de classes d'homologie persistante.

Ensuite, nous avons appliqué l'homologie persistante sur les images en niveaux de gris en introduisant une nouvelle méthode de segmentation des images qui utilise les durées de vie des classes d'homologie. Puis, nous avons utilisés les notions de classes d'homologie pour segmenter des objets saillants dans des images 2D et 3D. Pour cela, nous avons réalisé la construction topologique sur les pixels et les superpixels, le dernier cas permettant de simplifier l'espace d'origine et par conséquent de réduire le temps de calcul et les ressources nécessaires. Contrairement aux méthodes de segmentation existantes qui nécessitent des paramètres a priori, comme la taille ou l'intensité des objets, nos méthodes n'ont pas besoin de paramètres a priori. Les classes de homologies sont calculées et les cycles intéressants sont sélectionnés a posteriori, sans refaire le calcul, en imposant un seuil. De plus, alors que les méthodes existantes sont appliquées sur des images de dimension spécifique, l'homologie persistante peut être utilisée pour segmenter des images multidimensionnelles.

En outre, une version augmentée d'homologie dans sa forme absolue, qui est l'homologie relative, a été utilisée dans nos applications. L'homologie relative peut détecter les trajets de cellules en mouvement dans les séquences d'images et cela du premier au dernier cadre. Cette méthode ne nécessite non plus des paramètres à priori et est générique dans sa construction. Nous avons montré que nos méthodes permettent de résoudre de nombreux problèmes bien connus dans le traitement d'images liés à la variabilité de l'arrière-plan, à la superposition des objets, à l'efficacité multidimensionnelle, aux paramètres a priori, etc. Dans le chapitre suivant, nous verrons quelques techniques qui permettent d'en déduire des inférences à partir d'images en utilisant la théorie des faisceaux, que ce soit par analyse d'échelle ou par localisation.

A.4 Chapitre 4: Théorie des faisceaux et applications sur les images

Dans le chapitre 2, nous nous sommes concentrés sur les notions de topologie et topologie algébrique et leurs applications et nous avons passé par les complexes cellulaires et le calcul des groupes d'homologie. Dans le chapitre 3, nous avons détaillé les notions d'homologie qui comprennent le concept de persistance et nous avons développé différentes méthodologies utilisant cette notion de topologie algébrique. Nous avons montré que la topologie algébrique peut être utilisée pour des tâches de traitement d'images comme la segmentation d'images, la segmentation des objets ou le suivi des objets. Dans ce chapitre, nous présentons une autre théorie issue de la topologie algébrique qui a été utilisée au cours des cinq dernières années dans des applications d'ingénierie. Celle-ci permet la fusion des

données et la transmission de l'information locale aux aspects globaux. Plus précisément, il s'agit de la théorie des faisceaux fondée par J. Leray [Mil00] qui est un champ abstrait de la théorie de la topologie algébrique qui concerne principalement la topologie dans ses aspects fondamentaux [Ser55, Swa64] ou sous ses formes plus modernes [Bre97] à cause de ses relations avec l'étude des espaces topologiques et des ensembles ouverts. La théorie des faisceaux est une notion de plus en plus utilisée dans le monde de l'analyse de données et en ingénierie.

A.4.1 Introduction à la théorie des faisceaux

Malgré le fait qu'il est considéré comme un domaine abstrait, un faisceau peut être simplement considéré comme une technique qui associe plusieurs types de données pouvant être catégorisées en ensembles à chaque partie d'un espace topologique et d'inspecter la cohérence de ces données entre les voisins dans cet espace. Fondamentalement, les faisceaux dans leurs versions appliquées représentent, selon un de ses initiateurs Michael Robinson [Rob17b], la bonne façon de construire une représentation qui stocke les données locales et récapitule un modèle topologique par des résumés cohomologiques. Les faisceaux couvrent le fait que la réciprocité de l'information sur deux régions qui se chevauchent entraîne la validité de l'information sur l'union de ces deux régions, et cela lui donne la capacité à globaliser les données étudiées.

En outre, la théorie des faisceaux assure le calcul des méthodes qui suivent la structure générale et commence à émerger dans les applications [GH11, JHR14, Rob16]. De plus, le progrès combinatoire [She85, Cur13] a rendu possible la manipulation des structures de données de point de vue des faisceaux.

Nous pensons donc que la théorie des faisceaux peut intégrer des informations d'une perspective locale dans les images à une version globale exemptée des informations inutiles. Pour cela, nous passons par un processus de codage des données existantes en faisceaux appelés **Sheafification** afin d'obtenir des éléments des faisceaux qui sont les sections et la cohomologie des faisceaux.

Les étapes proposées de l'utilisation de la théorie des faisceaux et de ses invariants sont :

1. Conception de l'espace topologique de base et d'interactions multivoies entre les sources de données.
2. Sheafifier : construire le modèle de relations entre les sources de données.
3. Catégoriser : placer les flux de données dans les espaces vectoriels pour aider au calcul et à l'analyse.
4. Calcul de cohomologie : globaliser les données pour trouver des invariants robustes.

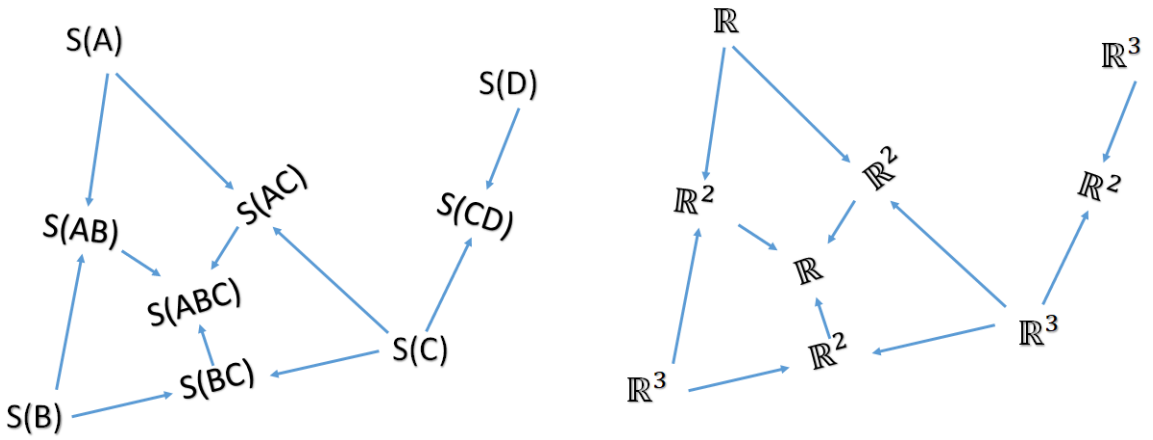


Figure A.11: Affectation des fibres au complexe cellulaire.

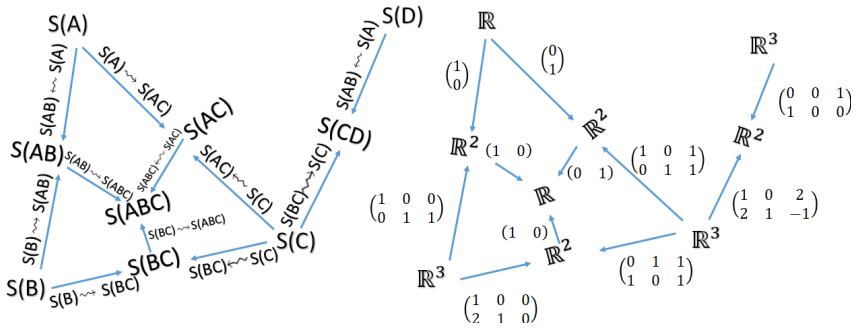


Figure A.12: Restrictions affectées pour chaque inclusion de cellule.

A.4.2 Faisceaux cellulaires

Par [She85], un faisceau $\mathcal{S}$ d'espaces vectoriels ou simplement sur un complexe cellulaire X qui est considéré comme l'espace de base, correspond à l'affectation de :

1. Un espace vectoriel $\mathcal{S}(\sigma)$ pour chaque cellule de X et appelé **fibre**, comme montré dans la figure A.11.
2. Une application linéaire $\mathcal{S}(\sigma \rightsquigarrow \tau) : \mathcal{S}(\sigma) \to \mathcal{S}(\tau)$ qui est appelée **restriction** le long de $\sigma \rightsquigarrow \tau$ à chaque fois que σ est un bord d'une cellule de plus grande dimension τ, $\sigma \subset \tau$, comme montré dans la figure A.12 et tels que
3. La restriction de σ à lui même est l'application identique, et si σ est un bord de τ $(\sigma \rightsquigarrow \tau)$ et τ est un bord de ω $(\tau \rightsquigarrow \omega)$ donc $\mathcal{S}(\tau \rightsquigarrow \omega) \circ \mathcal{S}(\sigma \rightsquigarrow \tau) = \mathcal{S}(\sigma \rightsquigarrow \omega)$.

Une **section locale** d'un faisceau est un élément s dans $\bigoplus_{\sigma \text{est ne cellule}} \mathcal{S}(\sigma)$ qui est la somme directe des fibres dans l'espace base. Cet élément doit satisfaire la relation $\mathcal{S}(\sigma \rightsquigarrow \tau)(s(\sigma)) = s(\tau)$ pour chaque $\sigma \subset \tau$, où $\mathcal{S}(\sigma \rightsquigarrow \tau)$ est une restriction linéaire, $s(\sigma)$ est un élément de $\mathcal{S}(\sigma)$ et $s(\tau)$ est un élément de $\mathcal{S}(\tau)$. Donc une section locale est spécifiquement une attribution de valeurs de chacune des fibres qui est cohérente avec les

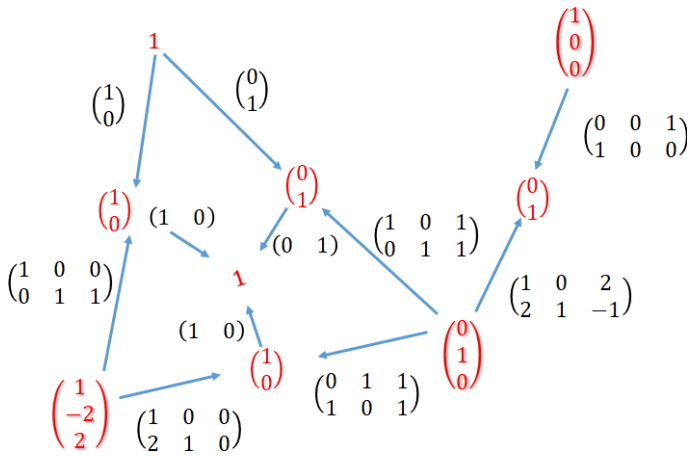


Figure A.13: Sections consistantes avec les restrictions.

restrictions, comme montré dans A.13. Une **section globale** est une section qui est tenue sur tout l'espace X.

Si nous voulons obtenir des résumés pertinents et exploitables des faisceaux, ces résumés sont appelés invariants de faisceaux et devraient être traitables par calcul. C'est le cas de la cohomologie de faisceaux comme nous le verrons ci-dessous.

Supposons que $\mathcal{S}$ est un faisceau des groupes abéliens sur un complexe cellulaire X. Le groupe cochaîne de dimension p, $C^p(X, \mathcal{S})$ de $\mathcal{S}$ est la somme directe des fibres sur les p-cellules de X. Ils sont représentés par :

$$C^p(X, \mathcal{S}) = \bigoplus_{\sigma \in X^p} \mathcal{S}(\sigma) \tag{A.9}$$

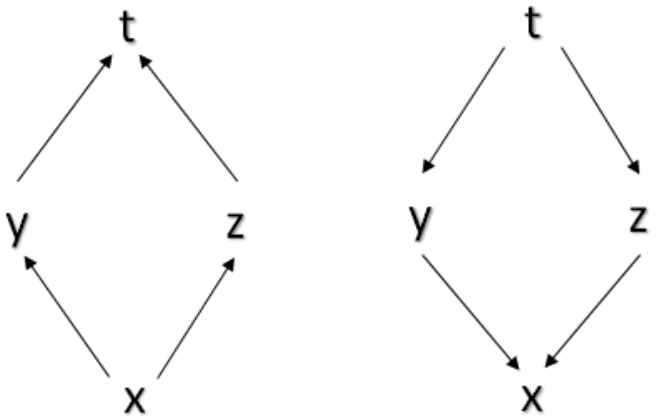
De même pour les chaînes, les relations entre cochaînes sont données par les fonctions de cobord. Les fonctions de cobord fonctionnent comme des dérivées discrètes et calculent les différences entre les fonctions sur les cellules de dimensions plus élevées. La fonction de cobord de dimension p est l'homorphosime $d^p : C^p(X, \mathcal{S}) \to C^{P+1}(X, \mathcal{S})$ donnée par:

$$(d^p f)(\tau) = \sum_{\sigma \in X^p} [\sigma : \tau] \mathcal{S}(\sigma \rightsquigarrow \tau) f(\sigma) \tag{A.10}$$

Les notions citées ci-dessus donnent naissance à la cohomologie des faisceaux cellulaires qui est définie par:

$$H^p(X, \mathcal{S}) = \ker d^p / \operatorname{Im} d^{p-1}. \tag{A.11}$$

D'où le groupe de cohomologie de faisceaux de dimension p représente les cochaines qui existent dans la dimension p, mais n'étaient pas déjà présents dans $p-1$. De plus, nous pouvons voir des cycles comme des cochaînes compatibles qui respectent les conditions des sections.



(a) A poset E où les flèches sont pointés en direction croissante.

(b) A poset E où les flèches sont pointés en direction décroissante.

Figure A.14: Un poset et son dual.

Comme interprétations de la cohomologie des faisceaux, nous avons:

- L'espace des sections globales d'un faisceau $\mathcal{S}$ sur un complexe cellulaire X est isomorphe à $H^0(X;\mathcal{S}) \simeq \ker d^0$.
- Le $H^1(X;\mathcal{S})$ peut représenter les nouvelles sections qui ne sont pas présentes en tant que sections globales lorsque vous utilisez uniquement des arêtes. Certaines références appellent cela des boucles de données ou des lacunes de désinformation. Donc c'est un pouvoir invariant, car il décrit ce qui se passe quand nous n'avons pas l'histoire complète.

A.4.3 Faisceaux sur les posets

Autres que les constructions sur les espaces cellulaires, comme des complexes simpliciaux ou autres, les faisceaux peuvent être construits sur des ordres partiels [Rob17a].

La définition du faisceau sur les posets provient du diagramme d'un poset, montré dans A.14 (a), où les sommets représentent les éléments et les flèches pointent des éléments les plus petits aux plus grands. Nous remplacerons chaque sommet par un ensemble ou un espace et chaque flèche par une fonction.

Faisceaux d'ensembles: Un faisceau $\mathscr{S}$ d'ensembles sur le poset E avec la topologie d'Alexandroff comprend les exigences suivantes:

1. Affectation d'un ensemble $\mathscr{S}(x)$ pour chaque $x \in E$ appelé fibre sur x.

2. Affectation d'une fonction $\mathscr{S}(x \leq y) : \mathscr{S}(x) \to \mathscr{S}(y)$ pour chaque paire $x \leq y \in E$, cette fonction est appelée une restriction,
3. $\mathscr{S}(x \leq z) = \mathscr{S}(y \leq z) \circ \mathscr{S}(x \leq y)$, pour chaque triplet $x \leq y \leq z \in E$.

Si un système spécifique est codé en tant que faisceau, son analyse peut être faite en utilisant des concepts cohomologiques. Les faisceaux construits sur des posets avec des fibres représentées par des espaces vectoriels et dont les restrictions sont linéaires ont des invariants topologiques qui peuvent être calculés. Si le faisceau construit n'est pas linéaire, alors il est nécessaire d'utiliser la catégorification pour transformer les données en aspects linéaires afin d'être manipulées dans des matrices.

Espace des cochaines des posets: Si $\mathscr{S}$ est un faisceau d'espaces vectoriels et ayant des fonctions de restriction linéaire sur un poset E alors l'espace de p-cochaîne $C^p(\mathscr{S})$ de $\mathscr{S}$ est le produit direct des fibres à la fin des chaînes de longueur p:

La fonction des p-cobords: $d^p : C^p(E, \mathscr{S}) \to C^{P+1}(E, \mathscr{S})$ est décrit dans la formule:

$$(d^p s)(x_0 < \ldots < x_{p+1}) = \sum_{i=0}^{p} s(x_0 < \ldots \widehat{x_i} < \ldots x_{p+1}) + (-1)^{p+1} \mathscr{S}(x_p < x_{p+1})(s(x_0 < \ldots < x_p)). \tag{A.12}$$

De plus, la cohomologie du faisceau $\mathscr{S}$ sur les posets est définie de la même manière que les faisceaux cellulaires:

$$H^p(E, \mathscr{S}) = \ker d^p / \operatorname{Im} d^{p-1}. \tag{A.13}$$

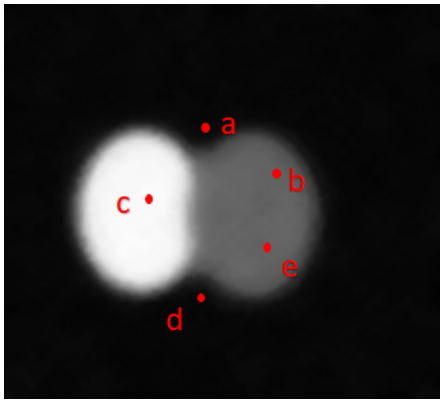
D'une manière similaire à l'interprétation de cohomologie des faisceaux cellulaires, la cohomologie des faisceaux des posets [Rob17b] exprime qu'un élément non trivial de $H^0(E; \mathscr{S})$ décrit des observations sur les intersections de p dimension des domaines sources qui sont compatibles avec d'autres restrictions aux intersections de dimension $p+1$, c'est-à-dire les noyaux de d^p, mais ne proviennent pas d'observations à partir des intersections $(p-1)$, c'est-à-dire l'image de d^{p-1}.

A.4.4 Applications sur les images

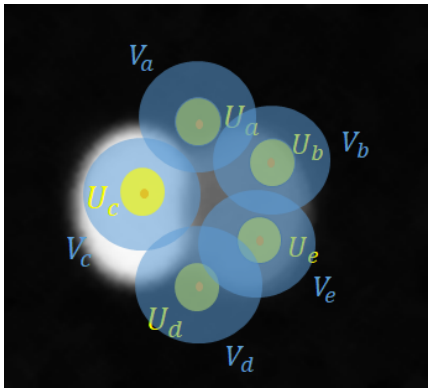
A.4.4.1 Applications sur les complexes de Čech

Nous présentons ici une application de notre méthode sur une image de taille 100×100 pixels montrée dans A.15.

Pour construire un complexe de Čech sur cette image, il faut choisir quelques points dans l'image. Nous avons basé ce choix sur le concept des points clés dans l'image en traitement. Les points clés représentent des emplacements intéressants dans l'image. Ils sont invariants par rapport aux changements d'images tels que la rotation, mise à l'échelle, la traduction, etc. Une technique pour trouver ces points clés saillants est la méthode SIFT (Scale Invariant Feature Transform) [Low04].



(a) L'image des objets collés avec les points clés.



(b) L'image avec les boules ouvertes centrées sur les points clés.

Figure A.15: Les points clés avec les boules ouvertes.

Maintenant, nous devons construire le complexe de Čech. Il faut noter que les faisceaux traduisent l'information commune entre deux espaces à l'union de ces deux espaces. Nous avons donc pensé à construire deux ouverts à chaque point clé. Le premier est une boule de rayon 6 et le second est de rayon 17 comme le montre la figure A.15 (b). Les boules en bleu seront notées comme $U_x = B(x, 6) = \{y | d(x, y) < 6\}$, et ceux en vert $V_x = B(x, 17) = \{y | d(x, y) < 6\}$ où $x, y \in \mathbb{R}^2$. Bien sûr, le choix des ouvertures et de ses rayons peut changer en fonction sur l'application souhaitée. Le but ici est de présenter un exemple de la façon de construire des faisceaux cellulaires sur les images.

Une fois que nous avons attribué le complexe, nous pouvons avoir l'espace de base pour le complexe de Čech construit. Il est montré dans la figure A.16. Maintenant, nous choisissons les fibres sur les sommets. Les fibres choisies sont les plus naturelles caractéristiques sur les ouvertures sur ces sommets. Pour cela, nous avons choisi la moyenne des valeurs d'intensité et des valeurs de gradient dans cet ouvert, de sorte que l'espace vectoriel associé est $\mathbb{R}^2$. Au niveau des bords, nous allons regarder sur les intersections des 2 ouverts et calculer les mêmes valeurs. Pour les triangles, nous recherchons les valeurs dans les 3 ouvertures intersectées. En résumé, toutes les cellules du complexe contiendront $\mathbb{R}^2$ comme espace vectoriel sur leurs tiges puisque nous regardons seulement la moyenne des valeurs de pixels et des gradients. Bien sûr, d'autres caractéristiques peuvent être étudiées comme la variance, la valeur maximale dans chaque ouverte et même les classes d'homologie par exemple.

Maintenant, nous voulons trouver les restrictions allant des cellules de dimension inférieures aux plus élevés. Ces restrictions ont $\mathbb{R}^2$ comme entrée et sortie donc ils seront des matrices de la forme $\mathbb{R}^{2\times2}$. Puisque nous recherchons la cohérence sur l'espace des faisceaux, nous allons comparer les données sur les cellules de faibles dimensions et sur leurs dimensions supérieures.

Nous représentons dans la figure A.16 ces restrictions. Bien sûr, les restrictions appartiendront à $\mathbb{R}^{2\times2}$.

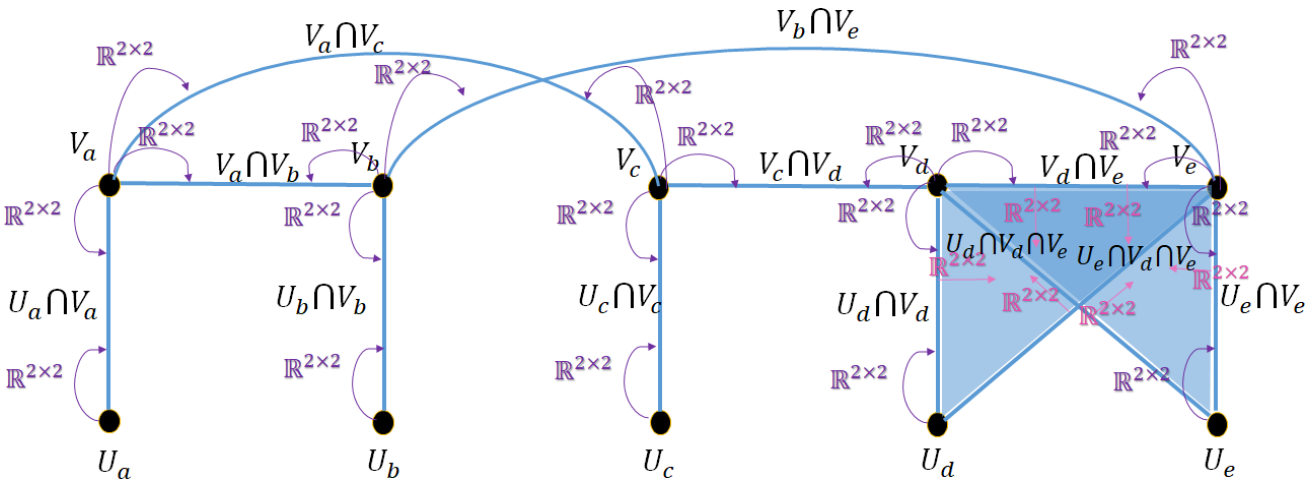


Figure A.16: L'espace de base des faisceaux sur les restrictions.

Maintenant, nous allons obtenir des fonctions de cobord d^0 et d^1 en utilisant

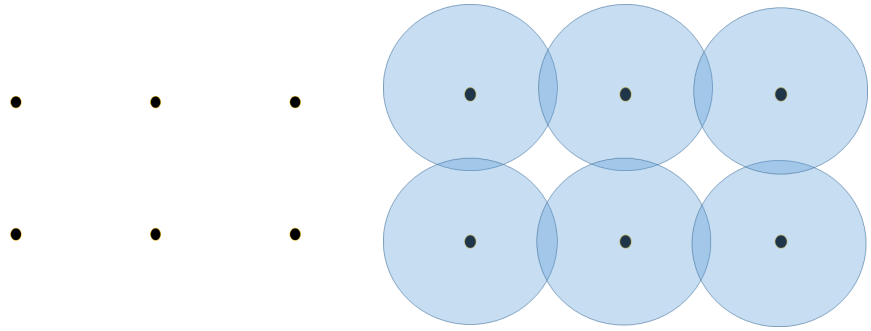
$$(d^p f)(\tau) = \sum_{\sigma \in X^p} [\sigma : \tau] \mathcal{S}(\sigma \rightsquigarrow \tau) f(\sigma) \tag{A.14}$$

Nous sommes ainsi capables de calculer la cohomologie des faisceaux de dimensions zéro et dimension 1, sur ce complexe en utilisant cette égalité $H^p(X, \mathcal{S}) = \ker d^p / \operatorname{Im} d^{p-1}$. Même si H^0 et H^1 n'apportent pas beaucoup d'informations dans ce cas, autres façons de modéliser le faisceau peuvent apporter une information importante comme la détection des deux objets collés par l'interprétation de H^1. Sachant que le choix des sommets et des ouvertures, les fibres sur les sommets et les restrictions sont une tâche de modélisation et représentent la valeur de la cohomologie.

A.4.4.2 Sections sur les images RGB

Nous avons vu que les faisceaux peuvent également être considérés comme des sections des faisceaux sans leurs calculs cohomologiques. Ces sections sont utiles sur les images en couleur. En réalité, les images colorées ont des pixels avec les valeurs appartenant à $\mathbb{R}^3$ correspondant au canal RGB. Chaque pixel représentera un sommet sur l'espace de base. Nous construisons d'abord un complexe de Čech pour avoir un espace de base pour notre faisceau. Les centres des boulles ouvertes seront naturellement les pixels et le rayon est $1/2 + \epsilon$ en supposant que la distance entre les vertices est 1 et $0 < \epsilon < (\frac{\sqrt{2}}{2} - \frac{1}{2})$. De cette façon, nous ne ferons que des intersections entre deux ouvertures et non trois, et donc seulement des arêtes et non des faces bidimensionnelles comme le montre la figure A.17.

Les fibres sur les sommets et les arêtes seront $\mathbb{R}^3$ comme le montre la figure A.18. Maintenant, nous devons chercher les sections. Comme données sur les sommets, nous avons choisi le canal RGB (r_i, g_i, b_i) pour chaque sommet dans l'espace de base. Pour les bords, nous assignons la différence des valeurs provenant des sommets $(r_j - r_i, g_j - g_i, b_j - b_i)$. Les



(a) 6 pixels/sommets. (b) Opens over the 6 pixels.

Figure A.17: 6 pixels avec leur ouverts avec rayons $1/2 + \epsilon$.

restrictions des sommets aux arêtes seront des matrices de la forme $\mathbb{R}^{3\times 3}$ comme indiqué dans la figure A.18.

A.4.4.3 Interprétation des faisceaux des modèles

Construire les modèles de faisceaux sur les espaces imbriqués permettra l'analyse des invariants topologiques en fonction des étapes suivantes:

1. Utilisation du faisceau sur un poset pour encoder des diagrammes de modèles.
2. Linéarisation des données si nécessaire,
3. Calculer les fonctions de cobord de ce faisceau,
4. Calcul de cohomologie pour résumer les modèles et les interpréter.

Supposons que nous ayons le modèle suivant :

- Trois ouverts comme A, B et C.
- Un cycle ou une classe d'homologie existe en $D = A \cup B$ comme indiqué dans A.19, mais pas dans $E = A \cup C$ ni $F = B \cup C$.

L'analyse cohomologique de ce faisceau nous permettra d'interpréter et comprendre le modèle étudié. Une fois que nous avons les matrices de cobords $d^0(\mathcal{E}, \mathscr{S})$ et $d^1(\mathcal{E}, \mathscr{S})$, on peut calculer les cohomologie de dimensions zéro et un. Rappelons que $H^p(E, \mathscr{S}) = \ker d^p / \operatorname{Im} d^{p-1}$, alors $H^0(E, \mathscr{S}) = \ker d^0 / \operatorname{Im} d^{-1} = \ker d^0$.

Ainsi $H^0(\mathcal{E}, \mathscr{S})$ est généré par $< a_A, a_B, a_C >$, qui est le l'espace des sections globales de ce modèle de faisceau. Cela révèle la cohérence de ces éléments à travers le faisceau qui est normal puisque $0.a_A$, $0.a_B$ et $0.a_C$ sont transformés via des fonctions nulles.

D'un autre côté, $H^0(\mathcal{E}, \mathscr{S}) =< a_{C<E}, a_{C<F} >$ sont des sections qui ne sont pas présentes en tant que sections globales et peuvent donner des informations supplémentaires et inférence du modèle. Pour interpréter les résultats de H^0, nous concluons que la classe a

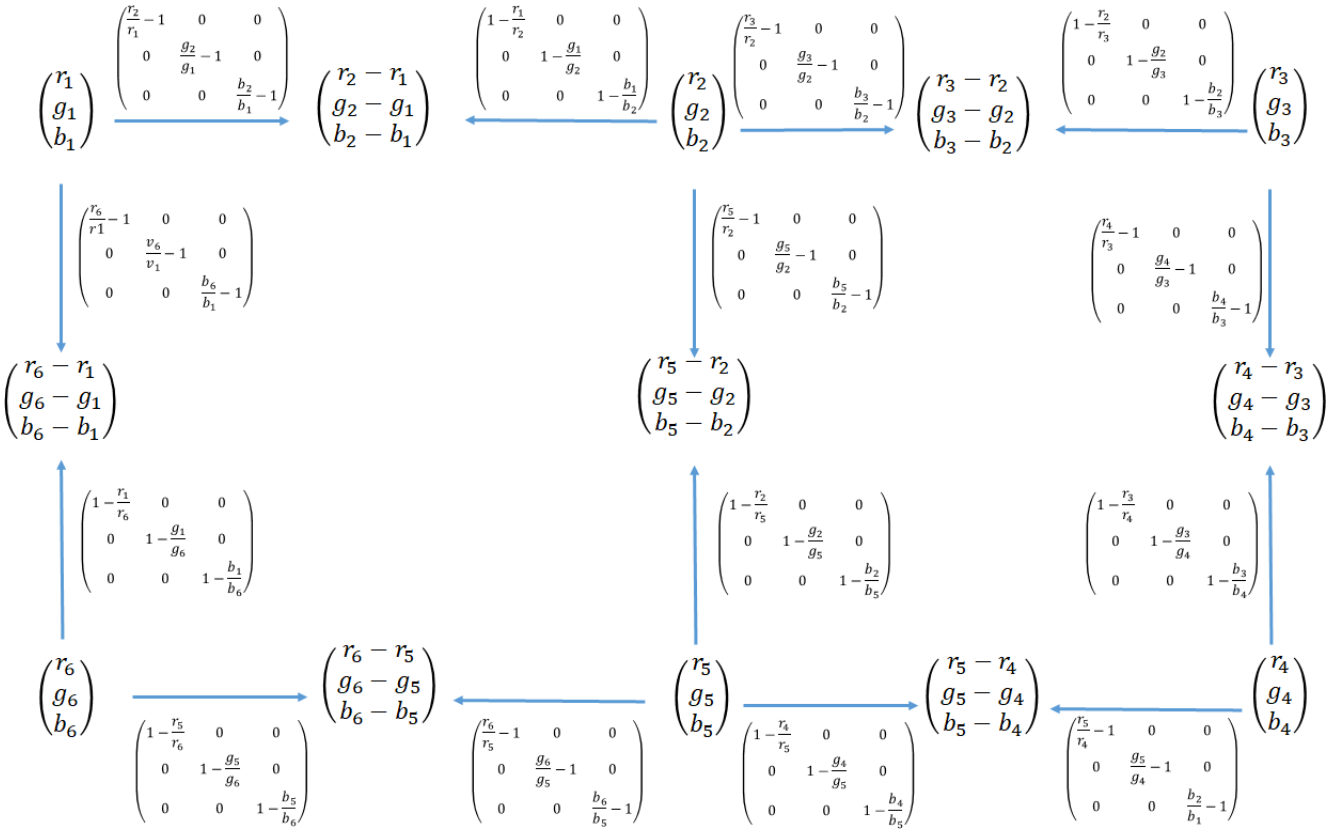


Figure A.18: Sections consistantes avec leurs restrictions sur les 6 pixels.

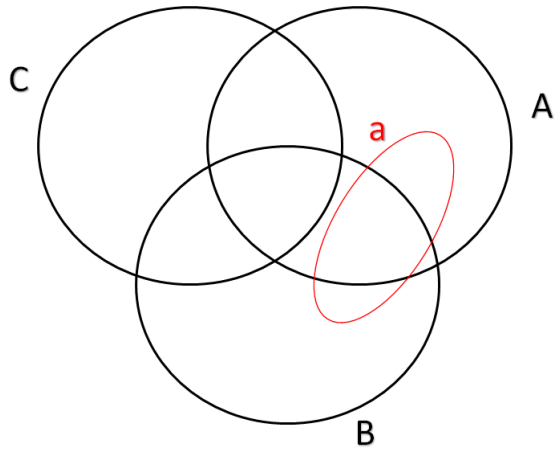


Figure A.19: L'espace entier avec le cycle a.

n'existe pas dans l'un des grands ouverts. Pour H^1, a n'est pas inclus dans E ni dans F, ce qui laisse en déduire que cet élément est dans $D = A \cup B$.

Bien sûr, autres modèles peuvent être construits et nous pouvons les analyser par des interprétations cohomologiques.

A.4.4.4 Conclusion

Dans ce chapitre, nous nous sommes intéressés à un nouvel aspect des approches de la topologie algébrique qui repose sur la théorie des faisceaux. Nous avons commencé par une introduction suivie de la présentation de certains aspects fondamentaux des faisceaux sur des espaces topologiques. Ces aspects fondamentaux ont été traduits aux espaces cellulaires. Le concept des faisceaux de cellules décrit comment on peut transformer un faisceau sur un complexe cellulaire et comment on peut obtenir des complexes de cochaine associés à des fonctions de cobord. Ces fonctions de cobords permettent de calculer la cohomologie des faisceaux afin d'obtenir un invariant topologique qui transforme l'information locale en une information globale. De plus, nous introduisons la construction de faisceaux sur des ensembles partiellement ordonnés pour aboutir à son analyse cohomologique.

A.5 Chapitre 5: Conclusion et perspectives

A.5.1 Conclusion générale

La topologie et la topologie algébrique peuvent être utilisées pour développer plusieurs méthodologies utiles aux problèmes d'ingénierie. Cette théorie mathématique est particulièrement intéressante et son applicabilité est principalement due à la possibilité calculer et de représenter des notions sur un système informatique. Pour les tâches de traitement d'images, l'homologie persistante et sa forme relative, ainsi que la théorie des faisceaux, sont particulièrement efficaces. Les durées de vie des classes d'homologie persistantes donnent lieu à une nouvelle façon de segmenter les images. Mettre en évidence les classes d'homologie les plus persistantes qui ont résisté aux variations des espaces permet de segmenter des objets dans les dimensions 2D et 3D. Sans oublier la version relative qui permet de vider un sous-espace dans le complexe et donc d'élargir la notion de cycle pour le suivi d'objets en mouvement.

Le succès de ces approches topologiques est dû à la flexibilité de la manipulation des espaces. En effet, une simple notion de voisinage est requise. Celle-ci est transformée en une liste des cellules et des relations entre eux à travers un opérateur de bord. Une fois ces données extraites, il est possible de calculer plusieurs invariants mesurant diverses quantités qui ne sont pas nécessairement scalaires comme les groupes d'homologie. Les quantités résultant de la topologie décrivent la forme de l'espace et donnent une représentation qualitative. Bien que cet aspect qualitatif semble très faible par rapport

aux séries quantitatives de valeurs, nous retenons la flexibilité de l'approche. Grâce à cela, l'homologie persistante permet par exemple de segmenter les objets dans les images sans avoir besoin de paramètres préalables comme les tailles ou les intensités des objets.

La difficulté d'utiliser la topologie algébrique réside principalement dans la spatialisation du problème et l'interprétation des invariants. Par exemple, au chapitre 3, le moyen des constructions de la présentation combinatoire et de la filtration dépend fortement de l'espace et des données étudiés. La façon de former le complexe simplicial n'est pas immédiate et relier les groupes d'homologie calculés à l'origine de problème nécessite un peu de travail. Mais le gain est important, aucune hypothèse précédente requise si nous commençons à partir de pixels bruts par exemple ou superpixels comme au chapitre 3, et aucune hypothèse sur la taille des objets ou des valeurs d'intensités n'est requise.

La topologie est souvent appelée "géométrie du caoutchouc", mais son sous-champ, la topologie algébrique, fournit des outils mathématiques permettant le passage du local au global, c'est-à-dire le passage d'une simple notion locale de proximité à notion plus générale de la forme globale de l'espace. Les outils qui permettent cette intégration du local au global sont performants et peuvent donner lieu à plusieurs algorithmes d'analyse d'images comme nous avons vu dans ce travail. Par exemple, la théorie des faisceaux a permis de transformer l'information des aspects locaux aux aspects globaux en utilisant la cohomologie des faisceaux. L'interprétation de l'analyse d'échelle et la localisation a permis de prédire et comprendre les données étudiées sur l'aspect global. Le travail réalisé dans cette thèse, qui a consisté à translater les notions de topologie algébrique au traitement d'images présente un paradigme de changement complètement nouveau et qu'il faut approfondir.

Commençant par la segmentation d'images, nous avons proposé une technique qui associe le calcul des caractéristiques topologiques utilisant des durées de vie de classes d'homologie avec des caractéristiques statistiques dans les fenêtres glissantes. Ces caractéristiques ont été classées en utilisant la méthode K-moyennes donnant une méthodologie de segmentation d'image. Les résultats de ces méthodes ont été présentés dans une conférence internationale [AGV16a] et publiés dans un journal [AGV16b].

Les notions algébriques des classes d'homologie les plus persistantes ont été utilisées pour réaliser la segmentation d'objet dans les images. Une construction du schéma de filtration sur pixels bruts permet de segmenter des objets intéressants tels que les cellules et leurs composants en images biomédicales.

Une autre idée a été d'utiliser la présentation combinatoire des superpixels pour segmenter les objets dans de grandes images de 2D et 3D sans l'utilisation de paramètres a priori. Cette technique a été présentée dans une conférence nationale [AGK17a] et illustrée dans un article actuellement en cours de revision [AGV+17]. En outre, un article sur l'aspect appliqué de cette technique qui permet de trouver des composants de cellules dans les images biomédicales a été accepté pour la présentation dans une conférence internationale [AGK17b].

Le mouvement des cellules le long des séquences d'images a été à l'origine du développe-

ment d'une technique qui détecte et suit les objets de la première à la dernière image. Cette technique utilise l'homologie relative qui détend le principe de l'homologie à plus d'espaces. L'objet de cette technique a été repris dans un article actuellement en cours de révision [AGK+17c].

Nous avons également initié l'utilisation de la théorie des faisceaux sur les images afin de transformer l'information du local au global. Les faisceaux sur les espaces cellulaires et sur les posets permettent de transformer l'aspect algébrique du faisceau en un aspect linéaire. Cette construction permet l'analyse cohomologique du faisceau, qui permet de mieux comprendre et prédire des caractéristiques des espaces. Cette interprétation est utile dans l'analyse d'échelle de l'augmentation des espaces et localisation en espaces décroissants.

A.5.2 Perspectives

Les perspectives que notre travail ouvre sont nombreuses, puisque les méthodes issues de la topologie algébrique commencent tout juste à apparaître. Nous pouvons rapidement les classer perspectives à court terme et perspectives à long terme.

En premier lieu, il serait intéressant d'associer notre approche à l'analyse d'échelle en utilisant la théorie des faisceaux pour séparer les classes d'homologie qui sont à l'intérieur l'une de l'autre en 2D et en 3D. De plus, il serait important d'améliorer les cycles qui segmentent les objets. L'optimisation de ces cycles peut être faite en appliquant une autre théorie de la topologie algébrique qui est la théorie de Morse discrète. En fait, la théorie de Morse discrète [Mil63, Koz07] permet de simplifier un espace combinatoire X sans dénaturer sa structure topologique. Cette simplification est complétée par l'utilisation d'un champ vectoriel discret V. Puisque les groupes d'homologie $H_p(X)$ sont isomorphes au complexe de Morse, leur calcul est beaucoup plus simple. Après le flux qui a aidé à la construction du complexe Morse en utilisant les champs de vecteurs discrets, nous pouvons associer des points critiques pour obtenir un contour qui peut réduire radicalement le contour qui segmente les objets.

Dans un autre aspect, l'homologie persistante qui utilise uniquement les valeurs d'intensité a été confrontée à des problèmes dans la détection des objets collés. Nous avons l'intention d'utiliser la persistance multidimensionnelle pour résoudre ce problème. Même s'il n'y a pas d'invariant topologique complet pour persistance multidimensionnelle [CZ09], les auteurs de [ML15] ont réussi à créer un outil de visualisation de durées de vie des classes d'homologie. Cet outil qui dépend fortement de la construction combinatoire doit être adapté au problème donné. Nous avons cependant tenté cette approche sans réel succès immédiat, mais une approche plus construite sera sûrement bénéfique. Aussi, il y a encore un gros travail sur les moyens de calculer les classes d'homologie persistante multidimensionnelle.

La topologie algébrique n'est pas limitée à l'homologie. À plus long terme, ce serait intéressant de développer d'autres parties. Par exemple la cohomologie, alors que c'est un dual

d'homologie, étudie un espace en étudiant la cohérence locale des fonctions définies sur ce dernier. Les incohérences sont donc très informatives. Par exemple, avoir l'information sur aspects dimensionnels plus élevés, il est utile d'utiliser la cohomologie pour traduire l'information aux dimensions plus basses [dSMVJ11b]. Ne pas oublier l'utilisation de la persistance en zigzag [CdSM09] qui dépend des complexes des ensembles de niveaux et peut étudier plusieurs problèmes qui peuvent ne pas être couverts par la persistance habituelle.

Bibliography

[AAR16] E. T. S. Alotaibi and H. Al-Rawi. Push and spin: A complete multi-robot path planning algorithm. In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1–8, Nov 2016.

[ABB+15] P. Alotto L. Ayers, R. J. Boyd, P. Bultinck, M. Caffarel, R. Carbó-Dorca, M. Causá, J. Cioslowski, J. Contreras-Garcia, D. L. Cooper, P. Coppens, C. Gatti, Simon Grabowsky, P. Lazzeretti, P. Macchi, Á.Martín Pendás, P. L.A. Popelier, K. Ruedenberg, H. Rzepa, A. Savin, A. Sax, W.H. Eugen Schwarz, S. Shahbazian, B. Silvi, M. Solà, and V. Tsirelson. Six questions on topology in theoretical chemistry. *Computational and Theoretical Chemistry*, 1053:2 – 16, 2015. Special Issue: Understanding structure and reactivity from topology and beyond.

[AC17] A. Abdelli and H.-J. Choi. A four-frames differencing technique for moving objects detection in wide area surveillance. In *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 210–214, Feb 2017.

[AF08] C.C. Adams and R.D. Franzosa. *Introduction to Topology: Pure and Applied*. Featured Titles for Topology Series. Pearson Prentice Hall, 2008.

[AFW06] D. Arnold, R. Falk, and R. Winther. Differential complexes and stability of finite element methods I: The de Rham complex. In D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *The IMA Volumes in Mathematics and its Applications*, pages 23–46. Springer, Berlin, 2006.

[AGK17a] R. Assaf, A. Goupil, M. Kacim, and V. Vrabie. Topologie algébrique pour l'analyse d'images en niveaux de gris. In *Colloque Gretsi*, 2017.

[AGK17b] R. Assaf, A. Goupil, M. Kacim, and V. Vrabie. Topological persistence based on pixels for object segmentation in biomedical images. In *IEEE International conference on Advances in Biomedical Engineering*, October 2017.

[AGK+17c] R. Assaf, A. Goupil, M. Kacim, B. Wattellier, T. Boudier, and V. Vrabie. 2d+t track detection via relative persistent homology. *Manuscript submitted for publication to Signal Processing letters*, 2017.

[AGV16a] R. Assaf, A. Goupil, V. Vrabie, and M. Kacim. Homology functionality for grayscale image segmentation. In *Current Research In Information Technology, Mathematical Sciences, Science And Technology International Conference*, 2016.

[AGV16b] R. Assaf, A. Goupil, V. Vrabie, and M. Kacim. Homology functionality for grayscale image segmentation. *Journal of Informatics and Mathematical Sciences*, v. 8(n. 4):p. 281–286,, 2016.

[AGV+17] R. Assaf, A. Goupil, V. Vrabie, T. Boudier, and M. Kacim. Persistent homology for object segmentation in multidimensional grayscale images. *Manuscript submitted for publication to Pattern Recognition letters*, 2017.

[Ale37] P. Alexandroff. Diskrete raume. *Mat. Sb.*, 2:501–519, 1937.

[AP13] N. Y. An and C. M. Pun. Iterated graph cut integrating texture characterization for interactive image segmentation. In *2013 10th International Conference Computer Graphics, Imaging and Visualization*, pages 79–83, Aug 2013.

[ARC14] A. Adcock, D. Rubin, and G. Carlsson. Classification of hepatic lesions using the matching metric. *Computer Vision and Image Understanding*, 121:36 – 42, 2014.

[ASL+12] R. Achanta, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods, 2012.

[Baa17] N. Baas. On the concept of space in neuroscience. *Current Opinion in Systems Biology*, 1(Supplement C):32 – 37, 2017. Future of Systems Biology • Genomics and epigenomics.

[Bar16] F. Barros. Modeling mobility through dynamic topologies. *Simulation Modelling Practice and Theory*, 69:113 – 135, 2016.

[BDA16] F. Baumann, E. Dayangac, J. Aulinas, and M. Zobel. Medianstruck for long-term tracking applications. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, Dec 2016.

[BKR14] U. Bauer, M. Kerber, and J. Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Topological Methods in Data Analysis and Visualization III: Theory, Algorithms, and Applications*, pages 103–117. Springer International Publishing, Cham, 2014.

[BLW12] U. Bauer, C. Lange, and M. Wardetzky. Optimal topological simplification of discrete functions on surfaces. *Discrete & Computational Geometry*, 47(2):347–377, Mar 2012.

[BM12] J.-D. Boissonnat and C. Maria. The simplex tree: An efficient data structure for general simplicial complexes. In *Algorithms - ESA 2012: 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, pages 731–742. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[BMW09] P. Bon, G. Maucort, B. Wattellier, and S. Monneret. Quadriwave lateral shearing interferometry for quantitative phase microscopy of living cells. *Opt. Express*, 17(15):13080–13094, Jul 2009.

[Bou66] N. Bourbaki. *Elements of Mathematics: General Topology.* Addison–Wesley, 1966.

[BP16] W. J. Beksi and N. Papanikolopoulos. 3D region segmentation using topological persistence. In *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, pages 1079–1084, October 2016.

[BQC+17] W. Bickel, A. Quisenberry, P. Chandrasekar, M. Koffarnus, E. Fox, and C. Franck. The social interactome of recovery: Network topology influences social media engagement. *Drug and Alcohol Dependence*, 171:e20, 2017.

[Bre97] G. E. Bredon. Sheaves and presheaves. In *Sheaf Theory.* Springer, January 1997.

[BTM16] A. Broom, K. Trainor, D. WS MacKenzie, and E. M Meiering. Using natural sequences and modularity to design common and novel protein topologies. *Current Opinion in Structural Biology*, 38:26 – 36, 2016. New constructs and expression of proteins • Sequences and topology.

[Car09] G. Carlsson. Topology and data. *Bull. Amer. Math. Soc.*, 46:255–308, 2009.

[Car14] G. Carlsson. Topological pattern recognition for point cloud data. *Acta Numerica*, 2014.

[CC11] M. Kerber C. Chen. Persistent homology computation with a twist, 2011.

[CdSM09] G. Carlsson, V. de Silva, and D. Morozov. Zigzag persistent homology and real-valued functions. In *Proceedings of the Twenty-fifth Annual Symposium on Computational Geometry*, SCG '09, pages 247–256, New York, NY, USA, 2009. ACM.

[CF10] C. Chen and D. Freedman. Measuring and computing natural generators for homology groups. *Computational Geometry*, 43(2):169 – 181,

2010. Special Issue on the 24th European Workshop on Computational Geometry (EuroCG'08).

[CGR12] J. Curry, R. Ghrist, and M. Robinson. Euler calculus and its applications to signals and sensing. In *Proceedings of Symposia in Applied Mathematics: Advances in Applied and Computational Topology*, 2012.

[CHTH15] A. Chan-Hon-Tong and S. Herbin. Tracking based sparse box proposal for time constraint detection in video stream. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 81–86, Nov 2015.

[CKL17] J. S. Choi, S. Kang, and Y. Lee. Design and evaluation of a pcep-based topology discovery protocol for stateful pce. *Optical Switching and Networking*, 26:39 – 47, 2017. Advances on Path Computation Element.

[Cro05] M. D. Crossley. *Essential Topology*. Springer, 2005.

[CSdC+14] N. Chenouard, I. Smal, F. de Chaumont, M. Maska, I. Sbalzarini, Y. Gong, J. Cardinale, C. Carthel, S. Coraluppi, M. Winter, A. R. Cohen, W. Godinez, K. Rohr, Y. Kalaidzidis, L. Liang, J. Duncan, H. Shen, Y. Xu, K. Magnusson, J. Jalden, H. Blau, P. Paul-Gilloteaux, P. Roudot, C. Kervrann, F. Waharte, J.Y. Tinevez, S. Shorte, J. Willemse, K. Celler, G. van Wezel, H. W. Dan, Y.-S. Tsai, C. O de Solorzano, J.-C. Olivo-Marin, and E. Meijering. Objective comparison of particle tracking methods. *Nat Meth*, 11(3):281–289, March 2014.

[CSEH07] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete Comput. Geom. 37 (2007), no. 1, 103 120. MR 2279866 (2008i:68130)*, 2007.

[Cur13] J. Curry. Sheaves, cosheaves and applications. *arxiv:1303.3255.*, 2013.

[CW17] Z. Cang and G. Wei. Topologynet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Computational Biology*, 13(7):e1005690, July 2017.

[CYC+14] H. Cai, Z. Yang, X. Cao, W. Xia, and X. Xu. A new iterative triclass thresholding technique in image segmentation. *IEEE Transactions on Image Processing*, 23(3):1038–1046, March 2014.

[CZ09] G. Carlsson and A. Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, Jul 2009.

[CZC04] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, pages 124–135, New York, NY, USA, 2004. ACM.

[DAVLHM+17] G. Diaz-Arango, H. Vázquez-Leal, L. Hernandez-Martinez, M. T. Sanz Pascual, and M. Sandoval-Hernandez. Homotopy path planning for terrestrial robots using spherical algorithm. *IEEE Transactions on Automation Science and Engineering*, PP(99):1–19, 2017.

[DEL+16] Olga Dunaeva, Herbert Edelsbrunner, Anton Lukyanov, Michael Machin, Daria Malkova, Roman Kuvaev, and Sergey Kashin. The classification of endoscopy images with persistent homology. *Pattern Recognition Letters*, 83, Part 1:13 – 22, 2016. Geometric, topological and harmonic trends to image processing.

[DHK11] T. K. Dey, A. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *SIAM J. Comput.*, 40(4):1026–1044, July 2011.

[DS13] P. Dłotko and R. Specogna. A novel technique for cohomology computations in engineering practice. *Computer Methods in Applied Mechanics and Engineering*, 253:530 – 542, 2013.

[DSM12] S. Kumar Das, S. Kumar Saha, and D. Prasad Mukherjee. Segmentation of multiple objects evolving conditional random field based topology adaptive active membrane. *Signal Processing*, 92(10):2341 – 2355, 2012.

[dSMVJ11a] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems 27 (2011) 124003 (17pp)*, 2011.

[dSMVJ11b] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete & Computational Geometry*, 45(4):737–759, Jun 2011.

[DSW10] T. K. Dey, J. Sun, and Y. Wang. Approximating loops in a shortest homology basis from point data. In *Proceedings of the Twenty-sixth Annual Symposium on Computational Geometry*, SoCG '10, pages 166–175, New York, NY, USA, 2010. ACM.

[EH10] H. Edelsbrunner and J. L. Harer. *Computational topology:An introduction*. American Mathematical Society, 2010.

[Eri11] J. Erickson. Combinatorial optimization of cycles and bases, 2011.

[ESM15] S. Emrani, T. S. Saponas, D. Morris, and H. Krim. A novel framework for pulse pressure wave analysis using persistent homology. *IEEE Signal Processing Letters*, 22(11):1879–1883, November 2015.

[For98] R. Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90 – 145, 1998.

[For02] R. Forman. A user's guide to discrete morse theory. *Séminaire Lotharingien de Combinatoire [electronic only]*, 48:B48c, 35 p., electronic only–B48c, 35 p., electronic only, 2002.

[GBY+18] T. Y. Goh, S. Nisha Basah, H. Yazid, M. Juhairi Aziz Safar, and F. Syahir Ahmad Saad. Performance analysis of image thresholding: Otsu technique. *Measurement*, 114(Supplement C):298 – 307, 2018.

[GH11] R. Ghrist and Y. Hiraoka. Applications of sheaf cohomology and exact sequences on network codings. *preprint*, 2011.

[Ghr08a] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:61–75, 2008.

[Ghr08b] R. Ghrist. Barcodes: the persistent topology of data. *Bull. Am. Math. Soc. New Ser.*, pages 61–75, 2008.

[Ghr14] R. Ghrist. *Elementary Applied Topology*. Createspace Independent Pub, 2014.

[Ghr17] R. Ghrist. Homological algebra and data. preprint, 2017.

[GK13] R. Ghrist and S. Krishnan. A topological max-flow-min-cut theorem. In *In Proc. Global Sig. Inf. Proc*, page 815–818, 2013.

[GKK+15] M. Gupta, S. Kumar, N. Kejriwal, L. Behera, and K. S. Venkatesh. Surf-based human tracking algorithm for a human-following mobile robot. In *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 111–116, Nov 2015.

[GMACA14] J. Gul-Mohammed, I. Arganda-Carreras, V. Andrey, P.and Galy, and T. Boudier. A generic classification-based method for segmentation of nuclei in 3d images of early embryos. *BMC Bioinformatics*, 15(1):9, 2014.

[GMP+15] R. Gaetano, G. Masi, G. Poggi, L. Verdoliva, and G. Scarpa. Marker-controlled watershed-based segmentation of multiresolution remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(6):2987–3004, June 2015.

[Gro06] M. Gromov. *Metric Structures for Riemannian and Non-Riemannian Spaces*, volume 152. Springer, 12 2006.

[Hat01] A. Hatcher. Algebraic topology. *Cambridge Univ. Press*, November 2001.

[HCR15] E. Hernandez, M. Carreras, and P. Ridao. A comparison of homotopic path planning algorithms for robotic applications. *Robotics and Autonomous Systems*, 64:44 – 58, 2015.

[Hen14] James Hendler. Data integration for heterogenous datasets. *Big Data*, 2(4):205–215, December 2014.

[HKS11] K. Mischaikow H. Kurtuldu and M. F. Schatz. Extensive scaling from computational homology and karhunen-loe‘ve decomposition analysis of rayleigh-be´nard convection experiments. *PHYSICAL REVIEW LETTERS*, 2011.

[Hub06] J.H. Hubbard. *Teichmüller Theory*, volume 1. Matrix Editions, 2006.

[Hun80] T. Hungerford. *Algebra*. Springer, 1980.

[JCH16] Z. Li J. Chen and B. Huang. Linear spectral clustering superpixel. *IEEE TRANS. ON IMAGE PROCESSING , DECEMBER 2016*, 2016.

[JHR14] C. A. Joslyn, E. A. Hogan, and M. Robinson. Towards a topological framework for integrating semantic information sources. In *Semantic Technologies for Intelligence, Defense, and Security (STIDS)*, United States, 2014. KB Laskey, I Emmons and PCG Costa; George Mason University, Fairfax, VA, United States(US).

[JRG16] U. Javed, M. M. Riaz, A. Ghafoor, and T. A. Cheema. SAR image segmentation based on active contours with fuzzy logic. *IEEE Transactions on Aerospace and Electronic Systems*, 52(1):181–188, February 2016.

[KKF+13] P. Karasev, I. Kolesov, K. Fritscher, P. Vela, P. Mitchell, and A. Tannenbaum. Interactive medical image segmentation using pde control of active contours. *IEEE Transactions on Medical Imaging*, 32(11):2127–2139, November 2013.

[KLJ+16] M. Kramár, R. Levanger, J.Tithof, B. Suri, M. Xu, M. Paul, M. F. Schatz, and K. Mischaikow. Analysis of kolmogorov flow and rayleigh–bénard convection using persistent homology. *Physica D: Nonlinear Phenomena*, 334:82 – 98, 2016. Topology in Dynamics, Differential Equations, and Data.

[KM16] T. Kristo and N. U. Maulidevi. Deduction of fighting game countermeasures using neuroevolution of augmenting topologies. In *2016 International Conference on Data and Software Engineering (ICoDSE)*, pages 1–6, Oct 2016.

[KMM03] T. Kaczynski, K. Mischaikow, and M. Mrozek. *Computational Homology*. Springer, 2003.

[Koz07] D. Kozlov. *Combinatorial Algebraic Topology*. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2007.

[KWT88] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, Jan 1988.

[KZ12] Zoltan K. and Josiane Z. Markov random fields in image segmentation. *Foundations and Trends® in Signal Processing*, 5(1–2):1–155, 2012.

[LF07] D. Letscher and J. Fritts. Image segmentation using topological persistence. *Computer Analysis of Images and Patterns*, 2007.

[Lim15] L.-H. Lim. Hodge laplacians on graphs, 2015.

[LK15] S. Lockwood and B. Krishnamoorthy. Topological features in cancer gene expression data>. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 108–19, 2015.

[LLY17] W. Y. Lee, C. Y. Li, and J. Y. Yen. Integrating wavelet transformation with markov random field analysis for the depth estimation of light-field images. *IET Computer Vision*, 11(5):358–367, 2017.

[Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov 2004.

[LST16] T. Liu, M. Seyedhosseini, and T. Tasdizen. Image segmentation using hierarchical merge tree. *IEEE Transactions on Image Processing*, 25(10):4596–4607, October 2016.

[LWD14] X. Li, X. Wang, and Y. Dai. Robust global minimization of active contour model for multi-object medical image segmentation. In *2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 1443–1448, May 2014.

[LWD17] W. Li, Y. Wang, J. Du, and J. Lai. Synergistic integration of graph-cut and cloud model strategies for image segmentation. *Neurocomputing*, 257:37 – 46, 2017. Machine Learning and Signal Processing for Big Multimedia Analysis.

[LWE+15] C. Li, X. Wang, S. Eberl, M. Fulham, Y. Yin, and D. Dagan Feng. Supervised variational model with statistical inference and its application in medical image segmentation. *IEEE Transactions on Biomedical Engineering*, 62(1):196–207, January 2015.

[LWZ08] M. Liu, C. Wu, and Y. Zhang. A review of traffic visual tracking technology. In *Proc. Language and Image Processing 2008 Int. Conf. Audio*, pages 1016–1020, July 2008.

[LZL15] L. Liu, N. Zhang, and Y. Liu. Topology control models and solutions for signal irregularity in mobile underwater wireless sensor networks. *Journal of Network and Computer Applications*, 51:68 – 90, 2015.

[LZW10] D. Li, G. Zhang, Z. Wu, and L. Yi. An edge embedded marker-based watershed algorithm for high spatial resolution remote sensing image segmentation. *IEEE Transactions on Image Processing*, 19(10):2781–2787, Oct 2010.

[Mac71] S. MacLane. *Categories for the Working Mathematician*. Springer-Verlag, New York, 1971. Graduate Texts in Mathematics, Vol. 5.

[Mah14] D. Mahapatra. Analyzing training information from random forests for improved image segmentation. *IEEE Transactions on Image Processing*, 23(4):1504–1512, April 2014.

[Mas91] W. S. Massey. *A basic course in algebraic topology*. Springer, 1991.

[MC15] M.S.Sonawane and C.A.Dhawale. Article: A brief survey on image segmentation methods. *IJCA Proceedings on National conference on Digital Image and Signal Processing*, DISP 2015(1):1–5, April 2015. Full text available.

[ME06] A. Muhammad and M. Egerstedt. Control using higher order laplacians in network topologies. In *In Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, page 1024–1038, Kyoto, Japan, 2006.

[MFC13] M. Musci, R. Q. Feitosa, and G. A. O. P. Costa. An object-based image analysis approach based on independent segmentations. In *Joint Urban Remote Sensing Event 2013*, pages 275–278, April 2013.

[Mil63] J. W. Milnor. *Morse theory*, volume Vol. 51. of *Ann. of Math. Stud.* Princeton University Press, 1963.

[Mil00] J. Milnor. Leray in oflag xviia: the origins of sheaf theory, sheaf cohomology, and spectral sequences. In *Gazette des mathématiciens*, volume no. 84 suppl., page 17–34. American Mathematical Society, 2000.

[ML15] M. Wright M.l Lesnick. Interactive visualization of 2-d persistence modules. preprint, 2015.

[MN13] K. Mischaikow and V. Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, Sep 2013.

[Mor89] S.A. Morris. *Topology Without Tears*. University of New England, 1989.

[MRS15] E. Merelli, M. Rucco, P. Sloot, and L. Tesei. Topological characterization of complex systems: Using persistent entropy. *Entropy*, 17(10):6872–6892, 2015.

[MST+16] S. K. Mylonas, D. G. Stavrakoudis, J. B. Theocharis, G. C. Zalidis, and I. Z. Gitas. A local search-based genesis algorithm for the segmentation and classification of remote-sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(4):1470–1492, April 2016.

[Mun75] J. R. Munkres. *Topology: A First Course*. Prentice Hall, 1975.

[MVM+16] P. Moeskops, M. A. Viergever, A. M. Mendrik, L. S. de Vries, M. J. N. L. Benders, and I. Išgum. Automatic segmentation of mr brain images with a convolutional neural network. *IEEE Transactions on Medical Imaging*, 35(5):1252–1261, May 2016.

[MW07] K. Mischaikow and T. Wanner. Probabilistic validation of homology computations for nodal domains. *Ann. Appl. Probab.*, 17(3):980–1018, 06 2007.

[MW10] M. Mrozek and T. Wanner. Coreduction homology algorithm for inclusions and persistent homology. *Comput. Math. Appl.*, 60(10):2812–2833, November 2010.

[NHZ+17] B. Ning, Q. L. Han, Z. Zuo, J. Jin, and J. Zheng. Collective behaviors of mobile robots beyond the nearest neighbor rules with switching topology. *IEEE Transactions on Cybernetics*, PP(99):1–14, 2017.

[NS08] David Nistér and Henrik Stewénius. Linear time maximally stable extremal regions. *Computer Vision – ECCV 2008: 10th European Conference on Computer Vision, Marseille, France, October 12-18, 2008, Proceedings, Part II*, pages 183–196, 2008.

[OK13] K. Ohmori and T. L. Kunii. A general design method based on algebraic topology – a divide and conquer method. In *2013 International Conference on Cyberworlds*, pages 267–273, Oct 2013.

[OS88] S. Osher and J. A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[PCO16] A. Pratondo, C. K. Chui, and S. H. Ong. Robust edge-stop functions for edge-based active contour models in medical image segmentation. *IEEE Signal Processing Letters*, 23(2):222–226, February 2016.

[PD07] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[PF08] D. H. Parks and S. S. Fels. Evaluation of background subtraction algorithms with post-processing. In *Proc. IEEE Fifth Int. Conf. Advanced Video and Signal Based Surveillance*, pages 192–199, September 2008.

[PG87] A. Pérez and R. C. Gonzalez. An iterative thresholding algorithm for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(6):742–751, June 1987.

[PGK16] F. T. Pokorny, K. Goldberg, and D. Kragic. Topological trajectory clustering with relative persistent homology. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 16–23, May 2016.

[Pie91] B. C. Pierce. *Basic Category Theory for Computer Scientists*. MIT Press, Cambridge, MA, USA, 1991.

[PK15] C. Panagiotakis and E. Kokinou. Linear pattern detection of geological faults via a topology and shape optimization method. *IEEE Journal*

of Selected Topics in Applied Earth Observations and Remote Sensing, 8(1):3–11, Jan 2015.

[PPA16] S. Pereira, A. Pinto, V. Alves, and C. A. Silva. Brain tumor segmentation using convolutional neural networks in MRI images. *IEEE Transactions on Medical Imaging*, 35(5):1240–1251, May 2016.

[PTS14] R. E. Putra, H. Tjandrasa, and N. Suciati. Hemorrhage segmentation using mathematical morphology and digital image processing. In *Proceedings of International Conference on Information, Communication Technology and System (ICTS) 2014*, pages 141–146, Sept 2014.

[QSN+16] T. Qaiser, K. Sirinukunwattana, K. Nakane, Y.-W. Tsang, D. Epstein, and N. Rajpoot. Persistent homology for fast tumor segmentation in whole slide histology images. *Procedia Computer Science*, 90:119 – 124, 2016. 20th Conference on Medical Image Understanding and Analysis (MIUA 2016).

[Rob00] V. Robins. *Computational Topology at Multiple Resolutions: Foundations and Applications to Fractals and Dynamics*. PhD thesis, University of Colorado at Boulder, Boulder, CO, USA, 2000. AAI9979393.

[Rob12] M.l Robinson. Asynchronous logic circuits and sheaf obstructions. *Electronic Notes in Theoretical Computer Science*, 283(Supplement C):159 – 177, 2012. Proceedings of the workshop on Geometric and Topological Methods in Computer Science (GETCO).

[Rob13a] M. Robinson. The nyquist theorem for cellular sheaves. *Proc. Sampling Theory and Applications*, page pp 293–296, 2013.

[Rob13b] M. Robinson. Understanding networks and their behaviors using sheaf theory. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 911–914, Dec 2013.

[Rob14a] M. Robinson. Analyzing wireless communication network vulnerability with homological invariants. In *Proc. IEEE Global Conf. Signal and Information Processing (GlobalSIP)*, pages 900–904, December 2014.

[Rob14b] M. Robinson. *Topological Signal Processing*. Mathematical Engineering. Springer Berlin Heidelberg, 2014.

[Rob15] M. Robinson. A sheaf-theoretic perspective on sampling. *Sampling Theory, a Renaissance*, January 2015.

[Rob16] M. Robinson. Imaging geometric graphs using internal measurements. *Journal of Differential Equations*, 260(1):872 – 896, 2016.

[Rob17a] M. Robinson. Sheaf and duality methods for analyzing multi-model systems. In I. Pesenson, Q.T. Le Gia, A. Mayeli, H. Mhaskar, and D. Zhou, editors, *Recent Applications of Harmonic Analysis to Function Spaces,*

Differential Equations, and Data Science: Novel Methods in Harmonic Analysis, Volume 2, pages 653–703, Cham, 2017. Springer International Publishing.

[Rob17b] M. Robinson. Sheaves are the canonical data structure for sensor integration. *Information Fusion*, 36:208 – 224, 2017.

[Rob18] M. Robinson. Michael robinson webpage, 2018.

[RYK14] S. S. M. Radzi, S. N. Yaakob, Z. Kadim, and H. H. Woon. Extraction of moving objects using frame differencing, ghost and shadow removal. In *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, pages 229–234, Jan 2014.

[Ser55] J.P. Serre. Faisceaux algebriques coherents. *Annals of Mathematics*, 61(2):197–278, 1955.

[SG07] V. De Silva and R. Ghrist. Homological sensor networks. *Notices Amer. Math. Soc*, pages 10–17, 2007.

[She85] A. Shepard. *A cellular description of the derived category of a stratified space*. PhD thesis, Brown University, 1985.

[SK05] I.F. Sbalzarini and P. Koumoutsakos. Feature point tracking and trajectory analysis for video imaging in cell biology. *Journal of Structural Biology*, 151(2):182 – 195, 2005.

[SL16] H. C. Shih and E. R. Liu. Automatic reference color selection for adaptive mathematical morphology and application in image segmentation. *IEEE Transactions on Image Processing*, 25(10):4665–4676, Oct 2016.

[SLD17] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, 2017.

[SNK17] Y. Smirnov, A. Nikolaidis, and J. Kuri. Designing a resilient virtual topology in a multi-layer datacenter interconnection network. In *DRCN 2017 - Design of Reliable Communication Networks; 13th International Conference*, pages 1–6, March 2017.

[SSB15] P. K. Saha, R. Strand, and G. Borgefors. Digital topology and geometry in medical imaging: A survey. *IEEE Transactions on Medical Imaging*, 34(9):1940–1964, September 2015.

[STJ+16] R. Shang, P. Tian, L. Jiao, R. Stolkin, J. Feng, B. Hou, and X. Zhang. A spatial fuzzy clustering algorithm with kernel metric based on immune clone for SAR image segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(4):1640–1652, April 2016.

[Swa64] R. Swan. *The theory of sheaves*. University of Chicago Press, Chicago, 1964.

[SZ05] T. Schöneborn and G. M. Ziegler. The topological tverberg theorem and winding numbers. *Journal of Combinatorial Theory, Series A*, 112(1):82 – 104, 2005.

[SZ12] A. Szymczak and E. Zhang. Robust morse decompositions of piecewise constant vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 18(6):938–951, June 2012.

[SZ17] T. Su and S. Zhang. Local and global evaluation for remote sensing image segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:256 – 276, 2017.

[SZS+13] A. R. Sadri, M. Zekri, S. Sadri, N. Gheissari, M. Mokhtari, and F. Kolahdouzan. Segmentation of dermoscopy images using wavelet networks. *IEEE Transactions on Biomedical Engineering*, 60(4):1134–1141, April 2013.

[TGGP15] A. Troya-Galvis, P. Gançarski, N. Passat, and L. Berti-Équille. Unsupervised quantification of under- and over-segmentation for object-based remote sensing image analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):1936–1945, May 2015.

[THV16] R. T'Jampens, F. Hernandez, F. Vandecasteele, and S. Verstockt. Automatic detection, tracking and counting of birds in marine video content. In *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6, Dec 2016.

[TPS+17] J.Y. Tinevez, N. Perry, J. Schindelin, G. M. Hoopes, G. D. Reynolds, E. Laplantine, S. Y. Bednarek, S. L. Shorte, and K. W. Eliceiri. Trackmate: An open and extensible platform for single-particle tracking. *Methods*, 115:80 – 90, 2017. Image Processing for Biologists.

[TSJ08] A. Tahbaz-Salehi and A. Jadbabaie. Distributed coverage verification in sensor networks without location information. In *2008 47th IEEE Conference on Decision and Control*, pages 4170–4176, Dec 2008.

[TT12] A. Teichman and S. Thrun. Tracking-based semi-supervised learning. *Int. J. Rob. Res.*, 31(7):804–818, June 2012.

[TWS+16] M.D. Tumbeva, Y. Wang, M.M. Sowar, A.J. Dascanio, and A.P. Thrall. Quilt pattern inspired engineering: Efficient manufacturing of shelter topologies. *Automation in Construction*, 63:57 – 65, 2016.

[VRT16] V. Venkataraman, K. N. Ramamurthy, and P. Turaga. Persistent homology of attractors for action recognition. In *Proc. IEEE Int. Conf. Image Processing (ICIP)*, pages 4150–4154, September 2016.

[WCV12] H. Wagner, C. Chen, and E. Vuçini. *Efficient Computation of Persistent Homology for Cubical Data*, pages 91–106. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[WLL+14] F. Wang, J. Li, S. Liu, X. Zhao, D. Zhang, and Y. Tian. An improved adaptive genetic algorithm for image segmentation and vision alignment used in microelectronic bonding. *IEEE/ASME Transactions on Mechatronics*, 19(3):916–923, June 2014.

[WLW+15] Q. Wang, L. Lu, D. Wu, N. El-Zehiry, Y. Zheng, D. Shen, and K. S. Zhou. Automatic segmentation of spinal canals in ct images via iterative topology refinement. *IEEE Transactions on Medical Imaging*, 34(8):1694–1704, Aug 2015.

[WMS13] A. C. Wilkerson, T. J. Moore, A. Swami, and H. Krim. Simplifying the homology of networks via strong collapses. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5258–5262, May 2013.

[XCG17] Y. Xu, E. Carlinet, T. Géraud, and L. Najman. Hierarchical segmentation using tree-based shape spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(3):457–469, March 2017.

[XW14] K. Xia and G.W. Wei. Persistent homology analysis of protein structure, flexibility and folding. *International journal for numerical methods in biomedical engineering*, 30(8):814–844, June 2014.

[XZC+16] M. Xian, Y. Zhang, H. D. Cheng, F. Xu, and J. Ding. Neutro-connectedness cut. *IEEE Transactions on Image Processing*, 25(10):4691–4703, October 2016.

[YMD14] F. Yan, P. Martins, and L. Decreusefond. Accuracy of homology based coverage hole detection for wireless sensor networks on sphere. *IEEE Transactions on Wireless Communications*, 13(7):3583–3595, July 2014.

[YQF+14] L. Yang, Y. Qi, J. Fang, X. Ding, T. Liu, and M. Li. Frogeye: Perception of the slightest tag motion. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2670–2678, April 2014.

[YSA16] T. Yao, J. Song, P. An, and L. Liu. Multi-class object segmentation based on jointly integrating segment-level and image-level object priors. In *2016 IEEE International Conference on Information and Automation (ICIA)*, pages 1770–1775, Aug 2016.

[ZMAT16] M. Zandi, A. Mahmoudi-Aznaveh, and A. Talebpour. Iterative copy-move forgery detection based on a new interest point detector. *IEEE Transactions on Information Forensics and Security*, 11(11):2499–2512, Nov 2016.

[ZMC16] H. Zhu, F. Meng, J. Cai, and S. Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12 – 27, 2016.

[Zom10a] A Zomorodian. *Computational topology.* Atallah, M., Blanton, M. (eds.) Algorithms and Theory of Computation Handbook, 2nd edn. CRC, Boca Raton, 2010.

[Zom10b] A. Zomorodian. Fast construction of the vietoris-rips complex. *Computers & Graphics*, 34(3):263 – 271, 2010. Shape Modelling International (SMI) Conference 2010.

[ZPL16] G. Zhu, F. Porikli, and H. Li. Robust visual tracking with deep convolutional neural network based object proposals on pets. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1265–1272, June 2016.

[ZPR+16] M. Zanin, D. Papo, M. Romance, R. Criado, and S. Moral. The topology of card transaction money flows. *Physica A: Statistical Mechanics and its Applications*, 462:134 – 140, 2016.

Approches de topologie algébrique pour l'analyse d'images

La topologie algébrique, bien que domaine abstrait des mathématiques, apporte de nouveaux concepts pour le traitement d'images. En effet, ces tâches sont complexes et restent limitées par différents facteurs tels que la nécessité d'utiliser un paramétrage, l'influence de l'arrière-plan ou la superposition d'objets. Nous proposons ici des méthodes dérivées de la topologie algébrique qui diffèrent des méthodes classiques de traitement d'images par l'intégration d'informations locales vers des échelles globales grâce à des invariants topologiques. Une première méthode de segmentation d'images a été développée en ajoutant aux caractéristiques statistiques classiques d'autres de nature topologique calculées par homologie persistante. Une autre méthode basée sur des complexes topologiques a été développée dans le but de segmenter les objets dans des images 2D et 3D. Cette méthode segmente des objets dans des images multidimensionnelles et fournit une réponse à certains problèmes habituels en restant robuste vis à vis du bruit et de la variabilité de l'arrière-plan. Son application aux images de grande taille peut se faire en utilisant des superpixels. Nous avons également montré que l'homologie relative détecte le mouvement d'objets dans une séquence d'images qui apparaissent et disparaissent du début à la fin. Enfin, nous posons les bases d'un ensemble de méthodes d'analyse d'images basé sur la théorie des faisceaux qui permet de fusionner des données locales en un ensemble cohérent. De plus, nous proposons une seconde approche qui permet de comprendre et d'interpréter la structure d'une image en utilisant les invariants fournis par la cohomologie des faisceaux.

Topologie algébrique, homologie persistante, théorie des faisceaux, segmentation d'images, segmentation d'objets.

Algebraic topology approaches for image analysis

Algebraic topology, which is as an abstract domain of mathematics, can bring new concepts in the execution of the image processing tasks. Indeed, these tasks might be complex and limited by different factors such as the need of prior parameters, the influence of the background, the superposition of objects. In this thesis, we propose methods derived from algebraic topology that differ from classical image processing methods by integrating local information at global scales through topological invariants. A first method of image segmentation was developed by adding topological characteristics calculated through persistent homology to classical statistical characteristics. Another method based on topological complexes built from pixels was developed with the purpose to segment objects in 2D and 3D images. This method allows to segment objects in multidimensional images but also to provide an answer to known issues in object segmentation remaining robust regarding the noise and the variability of the background. Our method can be extended to large scale images by using the superpixels concept. We also showed that the relative version of homology can be used effectively to detect the movement of objects in image sequences. This method can detect and follow objects that appear and disappear in a video sequence from the beginning to the end of the sequence. Finally, we lay the foundations of a set of methods of image analysis based on sheaf theory that allows the merging of local data into a coherent whole. Moreover, we propose a second approach that allows to understand and interpret scale analysis and localization by using the sheaves cohomology.

Algebraic topology, persistent homology, sheaves theory, image segmentation, object segmentation.

Discipline : MATHEMATIQUES APPLIQUEES ET SCIENCES SOCIALES

Spécialité : Mathématiques appliquées au traitement d'images

Université de Reims Champagne-Ardenne

CRESTIC - EA 3804

Moulin de la Housse – 51867 REIMS