



UNIVERSIDAD DE SEVILLA
Escuela Técnica Superior de Ingeniería Informática
Departamento de Matemática Aplicada I

Homological Spanning Forests for Discrete Objects

Memoria presentada para optar al título de Doctor

Doctorando: Helena Molina Abril

Director: Pedro Real Jurado

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my Supervisor, Dr. Pedro Real Jurado, Head of the Group of Computational Topology and Applied Mathematics of Seville University. His wide knowledge, extraordinary ideas and personal guidance have been crucial throughout this period.

I am very grateful to Professor Walter G. Kropatsch for his detailed and constructive comments, and his important support.

Throughout this work I have collaborated with many colleagues for whom I have great regard. I wish to extend my thanks to all my colleagues from the Applied Mathematics Department of Seville University, and from the Pattern Recognition and Image Processing group at the Technical University of Vienna, for their good advices and encourage.

I warmly thank my family and friends for their unconditional support.

Resumen

El cálculo y representación de información topológica constituye parte fundamental en numerosas aplicaciones, tales como representación y compresión de imágenes, clasificación de imágenes, reconocimiento de patrones, modelado geométrico, etc. La homología en el contexto de objetos digitales es una noción algebraica que proporciona una descripción concisa de la topología de los mismos, en términos de sus componentes conexas, túneles y cavidades.

El propósito de este trabajo es desarrollar un marco teórico y práctico para extraer y explotar, de manera eficiente, información homológica en el marco de la imagen digital nD . Para ello combinamos técnicas clásicas de topología algebraica y de procesamiento de imágenes.

La herramienta principal creada para tal propósito consiste en una representación combinatorial, que llamamos Bosque Recubridor Homológico (ó HSF) de un objeto o imagen digital. Este nuevo modelo está compuesto por un conjunto de bosques dirigidos, construidos sobre un complejo celular subyacente de la imagen. La representación HSF se basa en el concepto algebraico de homotopía de cadenas y puede ser considerada como una generalización a complejos celulares de mayor dimensión del significado topológico de árbol recubridor de un grafo geométrico.

Restringiendo la definición HSF a $2D$, presentamos en este trabajo un marco de procesamiento secuencial y paralelo de imágenes y objetos digitales basado en homología.

Abstract

Computing and representing topological information form an important part in many applications such as image representation and compression, classification, pattern recognition, geometric modelling, etc. The homology of digital objects is an algebraic notion that provides a concise description of their topology in terms of connected components, tunnels and cavities.

The purpose of this work is to develop a theoretical and practical framework for efficiently extracting and exploiting useful homological information in the context of nD digital images. To achieve this goal, we intend to combine known techniques in algebraic topology, and image processing.

The main notion created for this purpose consists of a combinatorial representation called Homological Spanning Forest (or HSF, for short) of a digital object or a digital image. This new model is composed of a set of directed forests, which can be constructed under an underlying cell complex format of the image. HSF's are based on the algebraic concept of chain homotopies and they can be considered as a suitable generalization to higher dimensional cell complexes of the topological meaning of a spanning tree of a geometric graph.

Based on the HSF representation, we present here a $2D$ homology-based framework for sequential and parallel digital image processing.

Contents

1	Introduction	1
2	Preliminaries	5
2.1	Digital Image Processing	5
2.2	Combinatorial Topology	6
2.2.1	Neighbourhood and adjacency	7
2.2.2	Cell complexes	8
2.2.3	Simplicial complexes	9
2.2.4	Cubical complexes	11
2.3	Algebraic notions	12
2.3.1	Chain complexes	13
2.3.2	Simple-homotopy type	14
2.4	Homological notions	15
2.4.1	Homology of chain complexes	16
2.4.2	Cohomology of chain complexes	17
2.4.3	Chain contraction	18
2.5	Algebraic Discrete Morse Theory	19
3	Homological Spanning Forest representation	21
3.1	State of the art: AT-models	22
3.2	Intuitive idea	25
3.3	HSF for finite cell complexes	26
3.4	Integral-chain complexes	28
3.5	HSF and Discrete Morse Theory	35
3.6	HSF algorithm: A first HSF formal definition and computation	40
4	HSF Framework for 2D Image Analysis	51
4.1	HSF and ASDR cell complexes	53

4.2	HSF and 2D digital image processing.	57
4.2.1	Cellular Level.	58
4.2.2	Conceptual Level: tree-based homology information.	60
4.3	Homology and Cohomology of objects of interest.	65
4.4	Parallel homology-based processing	70
5	Implementation and Software	73
5.1	Homology computation programs	73
5.2	The HSF software	74
5.2.1	Class diagram	75
5.2.2	Interface	76
5.2.3	Algorithms	84
6	Conclusions and future work.	91

List of Figures

2.1	Voxel 3D image	6
2.2	2D neighbours of a point	7
2.3	3D neighbours of a point	8
2.4	Simplicial and cell complexes	10
2.5	Boundary operator	11
2.6	Simplicial complex	12
2.7	Cubical complex	13
2.8	Boundary relations	14
2.9	Triangulation of a torus	17
2.10	Cell homology collapsing	20
2.11	Chain homotopy	20
3.1	AT-model	25
3.2	HSF simple example	26
3.3	Integral chain complex and integral operator	28
3.4	Homology integral chain complex	29
3.5	Integral chain equivalence	31
3.6	Cells pairing	37
3.7	discrete vector field	39
3.8	HSF algorithm: Bing's house	44
3.9	HSF representation: Bing's house	45
3.10	HSF algorithm: torus	45
3.11	HSF representation: torus	46
3.12	HSF algorithm: sphere	46
3.13	HSF representation: sphere	47
3.14	HSF algorithm: double torus	48
3.15	HSF representation: double torus	49

4.1	2D HSF representation	55
4.2	Upper integral path	56
4.3	2D Optimal gradient vector field	56
4.4	HSF example	57
4.5	ASDR complex	59
4.6	Cell complex with pockets	60
4.7	Arrow reversing	62
4.8	Edge rotation	63
4.9	Face rotation	64
4.10	HSF of objects of interest	66
4.11	HSF of ROIs	67
4.12	Homology generators	68
4.13	HSF and cohomology	69
4.14	Acyclic scenario	69
4.15	Processing Elements	71
5.1	Class diagram	75
5.2	Interface design	76
5.3	File menu	77
5.4	Points displayed in the drawing area	78
5.5	Sequence of CT trabecular bone images	78
5.6	CT trabecular bone points	79
5.7	Simplicial complex	81
5.8	Cell complex	82
5.9	Different views of a complex	84
5.10	HSF of dimension $0 - 1$ and $1 - 2$	84
5.11	Overlapped cubes	86
5.12	Different distributions of black points in a unit cube	86
5.13	Eight points tetrahedralization	88
5.14	Different configurations of 5 black points inside a cube	88
5.15	3D binary digital image and simplicial complex	89
5.16	AT-model output	90

Chapter 1

Introduction

In many Image Processing and Computer Vision applications, the representation of an object in terms of essential features concerning its structure, shape and geometry is crucial. Topology concerns for those spatial properties that are preserved under continuous deformations of objects. For example deformations which involve stretching, bending, squeezing, or compressing, but no tearing or breaking.

A typical problem in topology is to classify homeomorphic spaces. A homeomorphism is a continuous function between two topological spaces that has a continuous inverse function. Roughly speaking, considering a geometric object as a topological space, a homeomorphism can be seen as a continuous stretching and bending of the object into a new shape.

Topological invariants are properties of topological spaces that are preserved under homeomorphism. These “numbers” can help us to distinguish between non-homeomorphic spaces. Given a topological invariant α , if $\alpha(X) \neq \alpha(Y)$ for two objects X e Y , we can conclude that X e Y are non-homeomorphic.

The homology groups are an easily computable topological invariant. Homological tools have already proved their usefulness in applications, e.g. [González-Díaz 05b, Niethammer 02, Żelawski 05], and their potential in multi-dimensional digital image analysis is undeniable. The homology of an n -dimensional digital image provides information on the number of connected components and holes of various dimensions. Informally speaking, for each dimension p , the homology of an object is characterized by its p -holes. In this way 0-holes can be seen as connected components, 1-holes can be seen as tunnels and 2-holes as cavities. The notion of p -holes is defined

for any dimension.

Given a digital image, computing homology generators makes possible to localize its connected components, to enclose the various holes geometrically, and to distinguish the corresponding dimensions of the holes. The way in which these holes are related to each other is also one important issue that would contribute to a better understanding of the degree of topological complexity of the analyzed digital object, and would shed light on its geometric features.

Most existing methods in the context of discrete imagery use a narrow set of topological properties, like connectedness, boundary components, Euler characteristic, simple points [Rosenfeld 70], local characterization of surfaces [Bertrand 97], etc. Those properties are not sufficient in order to characterize the topological complexity of images. Therefore new topological invariants would be useful to deal with a more exhaustive classification.

The adaptation of more powerful topological invariants to discrete data is an active field of research, where efficient methods from combinatorial or algebraic topology to compute more complicated topological features are needed. Working with n -dimensional digital objects, topological features are not limited to Betti numbers, Euler characteristic, connectivity, number of holes or cavities, but also include other advanced characteristics such as cohomology algebra, cohomology operations or homotopy groups, etc. which can help to, for instance, discriminate non-topologically equivalent objects.

The main aim of this work is to design a combinatorial representation of a binary digital image, from which advanced geometrical and topological information can be directly and efficiently extracted. Given a digital object, this representation can be constructed over an underlying cell complex format of the image. The geometrical and topological information will be codified in terms of trees “spanning” all the cells of the complex (considering them as vertices of these trees).

Topological information is meant here as homological information at two levels: (a) the (co)cycle level, in which the analysis is understood in terms of construction of computer calculus systems (cycle’s calculus) dealing with geometric (co)cycles as inputs and outputs (b) at algebraic topological invariant level, in a geometric aseptic ambiance, whose perspective is the design of fast algorithms for computing advanced topological invariants more complex than homology groups (cohomology algebra, homology

$A(\infty)$ -coalgebra, homotopy, (co)homology operations, etc).

We present in this thesis a first approach in digital context to progress in both directions, that is, the topology-based digital image representation and the processing framework called Homological Spanning Forest (HSF, for short). This framework allows a complete algebraic topological analysis and provides a representation that can be used for the development of efficient algorithms to compute analytical, algebraic, geometrical and topological features of discrete objects. We employ here a strategy of exploiting homology in the discrete context by setting up a combinatorial scaffolding (a set of trees) which allows us to build a strong algebraic relationship between the object and its homology groups: that is, a chain homotopy equivalence. This idea is underlying in the work of Eilenberg and MacLane [Eilenberg 54], Sergeraert [Sergeraert 94], Forman [Forman 95] and that of theory of discrete differential forms [Desbrun 05].

The trees of this HSF representation have as vertices the different cells of the complex but, in general, its edges can not be easily described in the connectivity graph of the complex (that is, having as vertices the cells of the complex and the edges being determined by the relation “to be boundary of”).

We present here an HSF representation for $2D$ images that is described in the connectivity graph of the complex, and our future aim is to obtain such a representation for higher dimensional images. This “homological representation” has some properties that we will study later under a more general, rigorous and formal mathematical context:

- (a) *Non-uniqueness*: it is not unique and strongly depends on how the cells are managed in order to obtain a minimal homological expression.
- (b) *Local Transformability*: a global HSF representation can be transformed into a different one by using local combinatorial operations.
- (c) *Geometric Acuity*: the use of coordinate-based trees allows to capture the geometry of the original object (each node is specified by an ordered pair of integer coordinates with respect to the square grid of the initial image).
- (d) *Topological Acuity*: it suitably encodes advanced topological features (Euler characteristic, Betti numbers, classification of cycles, determining the contractibility and transformability of cycles inside the object,

numerical invariants related to cohomology algebra, cohomology operations, ...), due to the fact that the HSF representation can be automatically rewritten in algebraic terms as a chain homotopy operator determining a strong relationship at chain level (formal sums of cells) between the geometric object and its minimal homological expression.

- (e) *Reusability*: an HSF representation of part of the image can be derived automatically from a previously computed HSF representation of the whole image.
- (f) *Parallelism*: The processing of HSF models is susceptible to be parallelized.

The following work is structured in chapters whose contents are described here:

- **Chapter 2** recalls the main notions and properties required in the following chapters. These notions are related to different fields like image processing, digital topology, algebraic topology and discrete Morse theory.
- **Chapter 3** defines the HSF representation and theoretical results. The chapter concludes with an HSF computation algorithm and examples of this computation in several objects.
- **Chapter 4** focuses in the 2D HSF framework that has been developed for topological image analysis.
- **Chapter 5** presents the software that has been created in order to test the HSF representation.
- **Chapter 6** concludes the thesis and lists important open questions.

Chapter 2

Preliminaries

The main ideas presented in this work involve the combining of different academic fields like image processing and algebraic topology. We introduce here some basic notions from each one of the different areas that have been united throughout this work. Due to this interdisciplinary, the results that we finally achieve may be useful for any of the involved fields.

In the Section 2.1 some concepts belonging to the digital image processing world are introduced. In Section 2.2 and 2.3 some combinatorial and algebraic topology terminology is presented. This terminology follows Munkres's book ([Munkres 84]). In Section 2.4 we briefly explain some homological notions, and in Section 2.5 concepts of Discrete Morse Theory are introduced.

2.1 Digital Image Processing

A *digital image* I is a function $I : D \rightarrow V$ defined on a discrete set D in \mathbb{R}^n (carrier of the image) onto a discrete set V in \mathbb{R} . The values in V can be for example grey values or colours. Depending on V images can be binary (usually $V = \{0, 1\}$), greyscale (V is a one dimensional domain representing grey values, usually $V = \{0, 1, \dots, 255\}$) or colour (V is a representation of the colour). Digital images are usually the result of an imaging process, which maps the continuous real world domain to a discrete one.

The elements of a two dimensional image ($n = 2$) are called pixels; the elements of a three dimensional image ($n = 3$) are called voxels (see Figure 2.1), and the elements of a four dimensional image ($n = 4$) are called doxels. Each resel (pixel, voxel or doxel) is associated with a lattice point in the

space.

The carrier of I is usually a subset of a uniform regular grid defining \mathbb{Z}^2 . An object of interest $O \subset D$ in a digital image $I : D \rightarrow V$, has as image support function $w : D \rightarrow \{0, 1\}$, which assigns the value 1 to any pixel of O , whereas each pixel of $O^c = D \setminus O$ has the value 0 (O^c is named the background).



Figure 2.1: A three dimensional image and its representation using voxels.

Digital image processing is the use of computer algorithms to perform image processing on digital images. In particular, digital image processing allows classification, feature extraction, pattern recognition, etc.

Within the field of digital image processing, digital topology is the study of the topological properties of digital images (see [Kong 89] for an introduction in digital topology). Results in digital topology provide a sound mathematical basis for image processing operations such as image thinning, border following, contour filling, etc.

2.2 Combinatorial Topology

In opposition to the classic algebraic topology mainly considering sets with infinitely many elements, computer imagery manipulates sets with finite numbers of points.

A first attempt to formulate consistent topology notions equivalent to the continuum notions of neighbourhoods is due to Azriel Rosenfeld, whose publications on the subject played a major role in establishing and developing the field of Digital Topology (see for instance [Rosenfeld 66, Rosenfeld 70]).

In digital topology a *topological space* X is a set of points with a definition for the open subsets of X , usually called neighbourhoods. Therefore, in order to define a topological space in the context of digital imaging, the topological characterization of *adjacency* and *neighbourhood* relations needs to be defined.

2.2.1 Neighbourhood and adjacency

In two dimensional images, the adjacencies that are used most commonly to relate points to their neighbours are the 4-adjacency and the 8-adjacency. Two lattice points in the plane are said to be 8-adjacent if they are distinct and each coordinate of one differs from the corresponding coordinate of the other by at most 1. Two lattice points are 4-adjacent if they are 8-adjacent and differ in at most one of their coordinates. In terms of point coordinates, every point that has the coordinates $(x \pm 1, y)$ or $(x, y \pm 1)$ is 4-adjacent to the point (x, y) . In addition to 4-adjacent points, each pixel with coordinates $(x \pm 1, y \pm 1)$ or $(x \pm 1, y \mp 1)$ is 8-adjacent to the point (x, y) (see Figure 2.2).

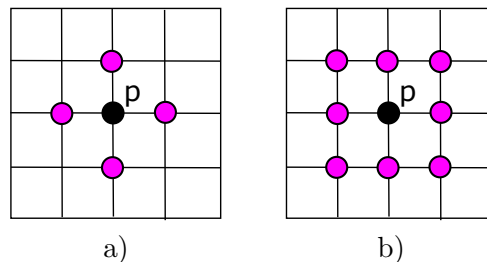


Figure 2.2: a) 4-neighbours and b) 8-neighbours of a point p .

In 3-dimensions two lattice points are said to be 26-adjacent if they are distinct and each coordinate of one differs from the corresponding coordinate of the other by at most 1. Two points are 18-adjacent if they are 26-adjacent and differ in at most two of their coordinates; and two lattice points are 6-adjacent if they are 26-adjacent and differ in at most one coordinate (see Figure 2.3).

For $n = 4, 8, 6, 18,$ or 26 an n -neighbour of a lattice point p is a point that is n -adjacent to p .

We say a set S of lattice points is n -connected if S cannot be partitioned into two subsets that are not n -adjacent to each other.

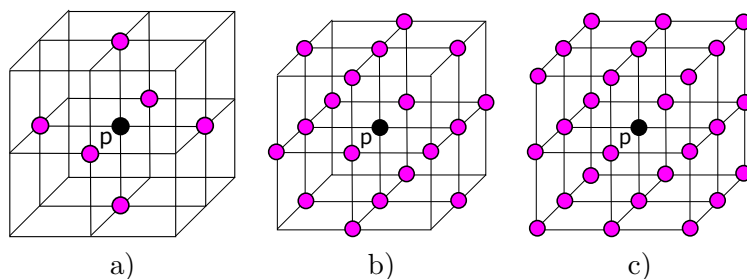


Figure 2.3: a) 6-neighbours, b) 18-neighbours and c) 26-neighbours of a point p .

A n -connected component of S is a non empty subset of S such that it is not n -adjacent to any other point in S .

In order to avoid some paradoxes such as those pointed out in [Rosenfeld 66], different adjacency relations are used for black and white points.

Following this idea, the concept of *digital image* can be redefined as a quadruple (V, m, n, B) , where $V = \mathbb{Z}^2$ or $V = \mathbb{Z}^3$, $B \subseteq V$, and where $(m, n) = (4, 8)$ or $(8, 4)$ if $V = \mathbb{Z}^2$, and $(m, n) = (6, 26), (26, 6), (6, 18)$, or $(18, 6)$ if $V = \mathbb{Z}^3$.

From now on we will consider this definition of digital image that takes connectivity into consideration (introduced in [Kong 89]).

We confine our attention to two dimensional and three dimensional binary digital images, although extensions to four dimensional (image sequence that represents a three dimensional space together with another dimension such as time) and greyscale data are planned.

2.2.2 Cell complexes

The goal of a topological map is to partition a topological space up into regions that are homeomorphic to open balls (see [Whitehead 49, Cardoze 06]). More formally, for $q \geq 1$ define $B^q = \{x \in \mathbb{R}^q : |x| < 1\}$, $\overline{B}^q = \{x \in \mathbb{R}^q : |x| \leq 1\}$, $S^q = \{x \in \mathbb{R}^{q+1} : |x| = 1\}$.

A space homeomorphic to B^q is called an *open q -cell*, a space homeomorphic to \overline{B}^q is called a *closed q -cell*, and a space homeomorphic to S^q is called a *q -sphere*. By convention we say that single points are both open and closed 0-cells. A partition of a space into open cells is called a *cell complex*. Recall that a q -dimensional (finite, normal, homogeneous) cell complex K is a pair $(X, \{K_i\}_{i=0}^q)$ where X is a Hausdorff space and $\{K_i\}_{i=0}^q$ is a finite

partition of X into open cells such that:

- (i) the set K_i is the set of all open i -cells with $0 \leq i \leq q$.
- (ii) for every open q -cell, σ , there is a continuous map $h_\sigma : \overline{B}^q \rightarrow X$ whose restriction to B^q is a homeomorphism onto σ and whose restriction to S^{q-1} , called *boundary of σ* and denoted $\partial\sigma$, is the union of open cells in K of dimension less than q . The space $h_\sigma(\overline{B}^q) = \overline{\sigma}$ is called *closure of the open cell σ* and $\overline{\sigma} = \sigma \cup \partial\sigma$. In addition it is required that every open cell σ is either a q -cell or is in the boundary of a q -cell.

In order to indicate relationships between cells, we write $\tau > \sigma$ (or $\sigma < \tau$) and we say that σ is a face of τ if $\sigma \neq \tau$ and $\sigma \subset \overline{\tau}$. We write $\tau \geq \sigma$ if either $\tau \geq \sigma$ or $\tau > \sigma$. We say that a cell σ is a *facet* of a cell τ when σ is a proper face of τ of maximal dimension.

The q -skeleton $K(q)$ of K is the set of all k -cell $\bigcup_{r=0}^q K_r$, with $0 \leq k \leq q$.

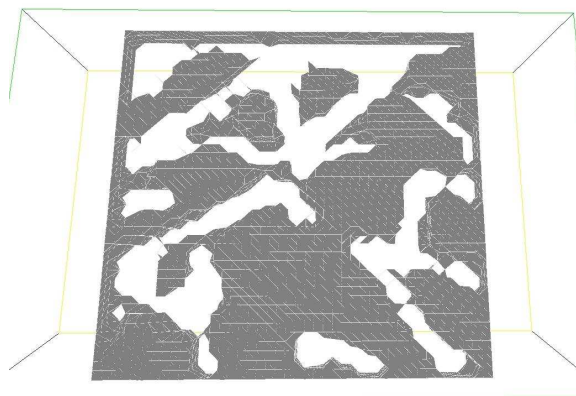
If all the cells of K are convex sets of the Euclidean q -dimensional space, then each cell can be represented by a point interior to it (commonly, its barycentre) and K is called *convex cell complex*.

A cell complex generalizes the notion of simplicial complex and cubical complexes. In simplicial complexes, polygons become triangles and polyhedra become tetrahedra (see Figure 2.4). In cubical complexes polygons become squares and polyhedra become cubes. General cell complexes are particularly important for spaces with non-uniform fractal structure, offering a dramatically more compact representation than simplicial or cubical complexes.

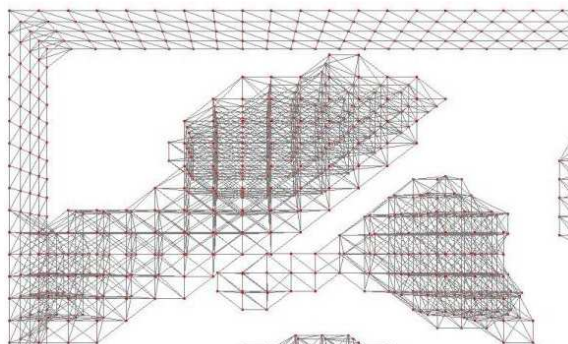
2.2.3 Simplicial complexes

Considering an ordering on a vertex set, a q -simplex with $q + 1$ affinely independent vertices $v_0 < \dots < v_q$ is the convex hull of these points, denoted by $\langle v_0, \dots, v_q \rangle$. In \mathbb{R}^3 : a 0-simplex is a vertex, a 1-simplex is an edge joining two vertices, a 2-simplex is a triangle limited by three edges, and a 3-simplex is a tetrahedron limited by four triangles.

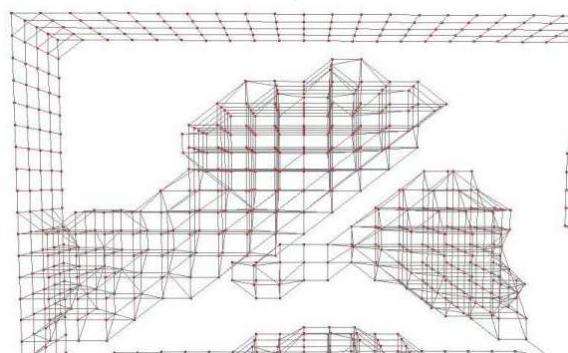
Given a q -simplex $\sigma = \langle v_0, \dots, v_q \rangle$ and $i < q$, an i -face of σ is an i -simplex whose vertices are in the set $\{v_0, \dots, v_q\}$. A *facet* of σ is a $(q - 1)$ -face of it. A simplex is *shared* if it is a face of more than one simplex. Otherwise, the simplex is *free* if it belongs to one higher dimensional simplex,



a)



b)



c)

Figure 2.4: a) A digital volume corresponding to a trabecular bone CT-image series b) a zoom to its associated simplicial complex and c) a zoom to its associated cell complex

and *maximal* if it does not belong to any. In Figure 2.5 boundary relations between simplexes are shown.

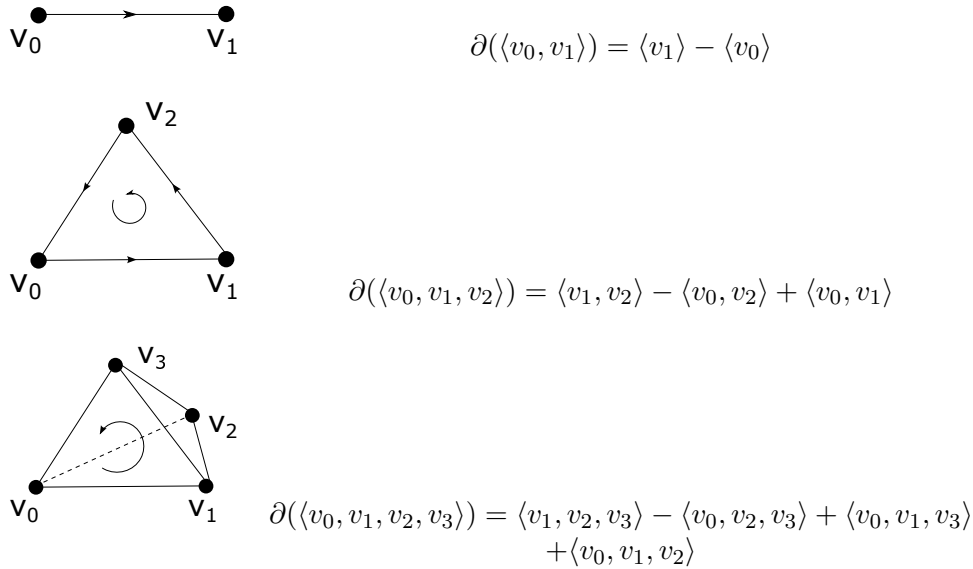


Figure 2.5: Boundary operator of an edge, a triangle and a tetrahedron (considering the orientation of the simplexes).

A *simplicial complex* K is a collection of simplexes such that ([Munkres 84]): Every face of a simplex of K is in K , and the intersection of any two simplexes of K is either a face of each of them, or empty (see Figure 2.6).

Given a digital image I , there exist different processes to associate a simplicial complex $K(I)$ to it. This simplicial complex is constructed on the triangulation of the Euclidean space determined by the neighbourhood relation. The 0-simplexes of $K(I)$ are the points of I . The i -simplexes of $K(I)$ are constituted by the different sorted sets of i -neighbour black points of I . Analogously a simplicial complex considering white points of I could be constructed.

A subset $\mathcal{L} \subseteq K$ is a *subcomplex* of K if it is a simplicial complex itself.

2.2.4 Cubical complexes

A cubical complex is in essence exactly the same as the simplicial complex described before, except one deals with cubes instead of simplexes [Kaczynski 04].

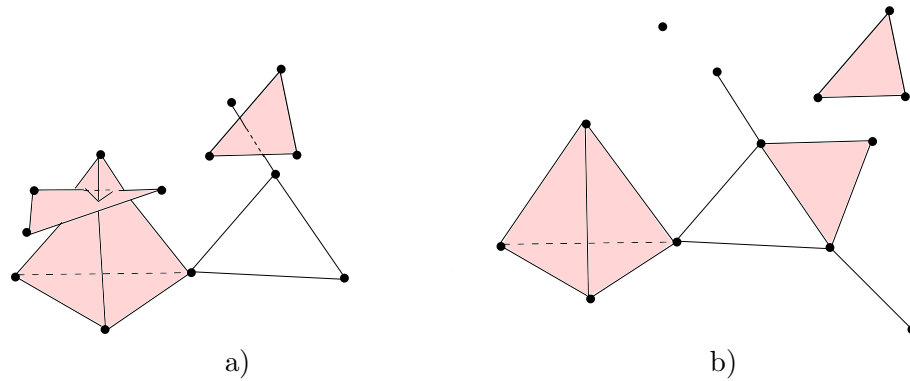


Figure 2.6: a) A set of elements that does not satisfy the conditions of a simplicial complex and b) a simplicial complex.

An *elementary interval* is a closed interval $J \subset \mathbb{R}$ of the form $J = [l, l+1]$ or $J = [l, l]$ for some $l \in \mathbb{Z}$.

An *elementary cube* Q is a finite product of elementary intervals, that is, $Q = J_1 \times J_2 \times \cdots \times J_q \subset \mathbb{R}^n$ where J_i is an elementary interval. The number $q \in \{0, 1, \dots, n\}$ of nondegenerate intervals is by definition the dimension of Q , and Q will be called a q -cube.

The faces and the boundary of a q -cube are defined in the same manner as for a simplex. In \mathbb{R}^3 : a 0-cube is a vertex, a 1-cube is an edge, a 2-cube is a filled square and a 3-cube is a filled cube.

A *cubical complex* K in \mathbb{R}^n , is a collection of q -cubes where $0 \leq q \leq n$ such that every face of a cube in K is also in K , and the intersection of any two cubes of K is either empty or a face of each of them [Allili 01]. See Figure 2.7 for some examples of cubical complexes.

For generality, in the following chapters we have considered the broader concept of cell complex embedded into the Euclidean n -dimensional space.

2.3 Algebraic notions

In most of the literature, a digital image has been endowed with a graph structure; the vertices being the points of the image, and the edges giving the connectivity between the points. This has enabled the use of combinatorial methods to provide theorems and proofs for basic topological results, and

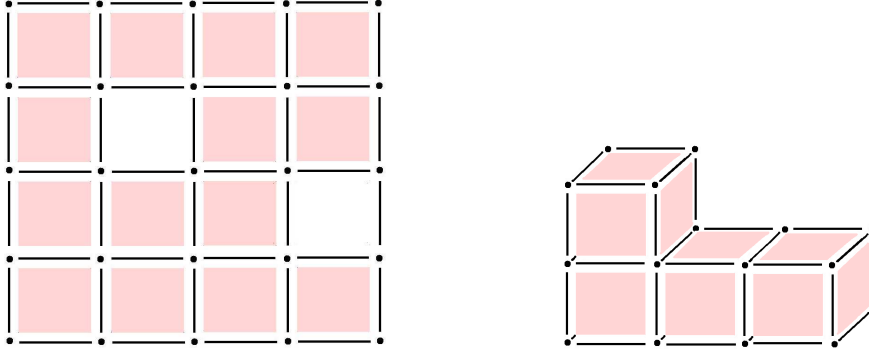


Figure 2.7: Examples of cubical complexes

many of the supporting topological issues have already been successfully resolved. However, these methods have been shown to be inadequate for the development of a topological theory in dimensions higher than three. This has led to the investigation of algebraic topology as an useful tool for providing results in digital topology [McAndrew 96].

2.3.1 Chain complexes

A *chain complex* \mathcal{C} is defined as a sequence

$$\dots \xrightarrow{d_{q+1}} \mathcal{C}_q \xrightarrow{d_q} \mathcal{C}_{q-1} \xrightarrow{d_{q-1}} \dots \xrightarrow{d_1} \mathcal{C}_0 \xrightarrow{d_0} 0$$

of abelian groups \mathcal{C}_q (called groups of q -chains) and homomorphisms $d_q : \mathcal{C}_q \rightarrow \mathcal{C}_{q-1}$ (called differential of \mathcal{C} in dimension q), indexed with the integers such that $d_q d_{q+1} = 0$, for every $q \geq 0$. Each group \mathcal{C}_q is an abelian group of finite rank.

A chain complex \mathcal{C} can be encoded as a couple (C, d) where:

- (i) $C = \{\mathcal{C}_q\}$ and for each q , \mathcal{C}_q is a base of \mathcal{C}_q ;
- (ii) $d = \{d_q\}$ and for each q , d_q is the differential of \mathcal{C} in dimension q with respect to the bases \mathcal{C}_q and \mathcal{C}_{q-1} .

Let $\{x_1, x_2, \dots, x_n\}$ be a finite set of symbols. The finite vector space of

formal linear combinations $\lambda_1 x_1 + \lambda_2 x_2 + \dots + \lambda_n x_n$, with $\lambda_i \in \Lambda$, is denoted by $\Lambda[x_1, \dots, x_n]$.

A q -chain $a \in C_q$ is a formal sum of elements of C_q , that is, $a = \sum \lambda_i a_i$, $\lambda_i \in \Lambda$ (the ground ring) and $a_i \in C_q$. The dimension of the q -chain a is q . The dimension of a chain complex \mathcal{C} is the maximal dimension of all the chains of \mathcal{C} .

Given a cell complex K , a chain complex can be associated to K in the following way: a q -chain a is a finite sum of q -cells of K . The groups of q -chains of K is denoted by $\mathcal{C}_q(K)$. The boundary of a q -cell is a $(q - 1)$ -chain defined by ∂_q . By linearity, ∂_q can be extended to q -chains. Then, the chain complex $\mathcal{C}(K)$ is the collection of chain groups $\mathcal{C}_q(K)$ connected by the boundary operators ∂_q (see Figure 2.8).

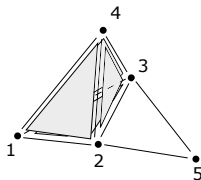


Figure 2.8: A cell complex showing the boundary relations

2.3.2 Simple-homotopy type

One of the basic problems of topology is to determine whether two given topological spaces are homeomorphic or not. There is no method for solving this problem in general, but techniques do exist that apply in particular cases.

Showing that two spaces are homeomorphic is a matter of constructing a continuous mapping from one to the other having a continuous inverse. To show that two spaces are not homeomorphic, the non-existence of a map satisfying those conditions has to be proved. Due to the difficulty of solving this problem, the usual way of proceeding is to find some topological properties that are satisfied by one space, but not the other [Munkres 00].

One of these properties that has been widely studied is the *homotopy type* of a space. Let us consider two spaces X and Y and its identity functions 1_X and 1_Y respectively. We denote I as the unit interval $[0, 1]$, If f and g are maps (i.e., continuous functions) from X to Y then f is *homotopic* to g , written $f \simeq g$, if there is a map $F : X \times I \rightarrow Y$ such that $F(x, 0) = f(x)$

and $F(x, 1) = g(x)$, for all $x \in X$.

$f : X \rightarrow Y$ is a *homotopy equivalence* if there exists $g : Y \rightarrow X$ such that $gf \simeq 1_X$ and $fg \simeq 1_Y$. We write $X \simeq Y$ if X and Y are homotopy equivalent.

Unfortunately, when given two spaces it is very hard to decide whether they are homotopy equivalent.

A particularly nice kind of homotopy equivalence is a *strong deformation retraction* [Cohen 73]. If $X \subset Y$ then $D : Y \rightarrow X$ is a strong deformation retraction if there is a map $F : Y \times I \rightarrow Y$ such that:

- (i) $F_0 = 1_Y$.
- (ii) $F_t(x) = x$ for all $(x, t) \in X \times I$.
- (iii) $F_1(y) = D(y)$ for all $y \in Y$.

In trying to understand homotopy equivalence, from the 1930's on, this problem has been approached in a combinatorial manner. Following this idea, the simple-homotopy type was defined in [Whitehead 50].

If K and L are finite simplicial complexes we say that there is an elementary simplicial collapse from K to L if L is a subcomplex of K and $K = L \cup aA$ where a is a vertex of K , A and a are simplexes of K , and $aA \cap L = aA$.

We say that K *collapses simplicially* to L , written $K \searrow_s L$, if there is a finite sequence of elementary simplicial collapses $K = K_0 \rightarrow K_1 \rightarrow \cdots \rightarrow K_q = L$. If $K \searrow_s L$ we also write $L \nearrow_s K$ and say that L expands simplicially to K . K and L have the same *simple-homotopy type* if there is a finite sequence $K = K_0 \rightarrow K_1 \rightarrow \cdots \rightarrow K_q = L$. where each arrow represents a simplicial expansion or a simplicial collapse.

Since an elementary simplicial collapse easily determines a strong deformation retraction (unique up to homotopy) it follows that, if K and L have the same simple-homotopy type, they must have the same homotopy type.

2.4 Homological notions

The homology of a topological space was introduced by Poincaré at the end of the nineteenth century and is one of the best understood topological invariants.

Homology is a very powerful tool in that it allows one to draw conclusions about global properties of spaces and maps from local computations. It also involves a wonderful mixture of algebra, combinatorics, computation and topology [Kaczynski 04].

Roughly speaking, the idea of homology of a cell complex consists of analyzing its degree of connectivity by using formal sums of cells.

2.4.1 Homology of chain complexes

Let $\mathcal{C} = (C, d)$ be a chain complex. A q -chain $a \in \mathcal{C}_q$ is a q -cycle if $d_q(a) = 0$. If $a = d_{q+1}(a')$ for some $(q+1)$ -chain $a' \in \mathcal{C}_{q+1}$, then a is a q -boundary. Denote the groups of q -cycles and q -boundaries by Z_q and B_q , respectively. Define the q^{th} homology group with coefficients in Λ to be the quotient group Z_q/B_q , denoted by $H_q(\mathcal{C}; \Lambda)$.

Two q -cycles a and b are homologous if there exists a $(q+1)$ -chain $c \in \mathcal{C}_{q+1}$ such that $a = b + d_{q+1}(c)$.

For each dimension q , there exists a finite number of elements in $H_q(\mathcal{C}; \Lambda)$, from which we can extract every element in $H_q(\mathcal{C}; \Lambda)$. These elements are the so called *homology generators* of dimension q . We say that $a \in Z_q$ is a *representative q -cycle* of a homology generator α of dimension q if $\alpha = a + B_q$. We denote $\alpha = [a]$.

A set of representative q -cycles c_1, \dots, c_n is a base of representative cycles in dimension q if the set $[c_1], \dots, [c_n]$ is a set of homology generators of the q^{th} homology group.

Like all finite generated abelian groups, each homology group is isomorphic to a product of cyclic groups $H_q(\mathcal{C}; \Lambda) \cong \mathbb{Z}^{\beta_q(\mathcal{C})} \oplus \bigoplus_i (\mathbb{Z}/d_i\mathbb{Z})$ for some integers β_q and $1 \leq d_1 \leq d_2 \leq \dots \leq d_i$ where each integer d_i is a divisor of its successor d_{i+1} (see [Munkres 84]). The rank β_q of the free component of $H_q(\mathcal{C})$ is called the q^{th} Betti number of \mathcal{C} . The components $(\mathbb{Z}, d_i\mathbb{Z})$ are called torsion subgroups. Intuitively, β_0 is the number of connected components, β_1 is the number of independent “holes” and β_2 is the number of “cavities”.

For instance, let T be the simplicial complex obtained from a triangulation of the torus (see figure 2.9). The homology groups with coefficients in \mathbb{Z} are:

$$H_0(T; \mathbb{Z}) = \mathbb{Z}, H_1(T; \mathbb{Z}) = \mathbb{Z} \oplus \mathbb{Z}, H_2(T; \mathbb{Z}) = \mathbb{Z}$$

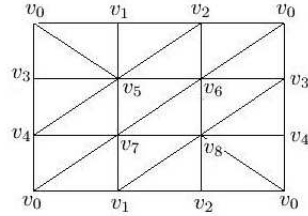


Figure 2.9: Triangulation of a torus

The representative cycles are $\langle v_0 \rangle$ in dimension 0, $\langle v_0, v_1 \rangle + \langle v_1, v_2 \rangle - \langle v_0, v_2 \rangle$ and $\langle v_0, v_3 \rangle + \langle v_3, v_4 \rangle - \langle v_0, v_4 \rangle$ in dimension 1 and the sum of all the triangles of T , in dimension 2. Therefore, the Betti numbers are: $\beta_0 = 1$, $\beta_1 = 2$ and $\beta_2 = 1$. Then, the torus has one connected components, two independent “holes” and one cavity.

2.4.2 Cohomology of chain complexes

Let $\mathcal{C} = (C, d)$ be a chain complex. A *cochain complex* \mathcal{C}^* is a sequence

$$\mathcal{C}^0 \xrightarrow{\delta_0} \mathcal{C}^1 \xrightarrow{\delta_1} \mathcal{C}^2 \xrightarrow{\delta_2} \dots$$

where $\mathcal{C}^q := \text{Hom}(\mathcal{C}_q; \Lambda)$ (called *q-cochain groups*) and $\delta_q : \mathcal{C}^q \rightarrow \mathcal{C}^{q+1}$ is a homomorphism given by $\delta_q(c) = cd_{q+1}$, for $c \in \mathcal{C}^q$ (called *codifferential of the cochain complex* in dimension q).

A cochain complex \mathcal{C}^* can be encoded as a couple (\mathcal{C}^*, δ) where:

- (i) $\mathcal{C}^* = \{\mathcal{C}^q\}$ and for each q , \mathcal{C}^q is a base of \mathcal{C}^q ;
- (ii) $\delta = \{\delta_q\}$ and for each q , δ_q is the codifferential of \mathcal{C}^* in dimension q with respect to the bases \mathcal{C}^q and \mathcal{C}^{q+1} .

Let $\mathcal{C} = (C, d)$ be a chain complex and $\mathcal{C}^* = (\mathcal{C}^*, \delta)$ the associated cochain complex. If $C_q = \{a_1, \dots, a_n\}$ is a base of the q -chain group C_q , then a base of the q -cochain group \mathcal{C}^q is given by $\mathcal{C}^q = \{a_1^*, \dots, a_n^*\}$, where $a_i^* : \mathcal{C}^q \rightarrow \Lambda$ is defined as: $a_i^*(a_i) = 1$ and $a_i^*(a_j) = 0$ for $1 \leq i, j \leq n$ and $j \neq i$.

A q -cochain $a^* \in \mathcal{C}^q$ is a q -cocycle if $\delta_q(a^*)=0$. If $a^* = \delta_{q-1}(b^*)$, then a^* is a q -coboundary. Denote the groups of q -cocycles and q -coboundaries by Z^q and B^q , respectively. Define the q^{th} cohomology group with coefficients in the ring Λ to be the quotient group Z^q/B^q , denoted by $H^q(\mathcal{C}; \Lambda)$.

For each q , there exists a finite number of elements of $H^q(\mathcal{C}; \Lambda)$ from which we can extract every element in $H^q(\mathcal{C}; \Lambda)$. These elements are called *cohomology generators* of dimension q . We say that $a^* \in Z^q$ is a representative q -cocycle of a cohomology generator α^* of dimension q if $\alpha^* = a^* + B^q$. We denote $\alpha^* = [a^*]$. A set of representative q -cocycles $\{c_1^*, \dots, c_n^*\}$ is a base of *representative cocycles* in dimension q if the set $\{[c_1^*], \dots, [c_n^*]\}$ is a base of the q^{th} cohomology group.

For each dimension q , the q^{th} cohomology group $H^q(\mathcal{C}; \mathbb{Z})$ is a finitely generated abelian group isomorphic to $F^q \oplus T^q$, where F^q and T^q are the free subgroup and the torsion subgroup of $H^q(\mathcal{C}; \mathbb{Z})$ respectively.

2.4.3 Chain contraction

Let $\mathcal{C} = (C, d)$ and $\mathcal{C}' = (C', d')$ be two chain complexes. A *chain map* from \mathcal{C} to \mathcal{C}' is a family of maps, $f = \{f_q : \mathcal{C}_q \rightarrow \mathcal{C}'_q\}$, such that for each dimension q ,

$$d'_q f_q = f_{q-1} d_q$$

is satisfied.

Let $f : \mathcal{C} \rightarrow \mathcal{C}'$ and $g : \mathcal{C}' \rightarrow \mathcal{C}$ be chain maps. A *chain homotopy* from f to g is a family of maps $\phi = \{\phi_q : \mathcal{C}_q \rightarrow \mathcal{C}'_{q+1}\}$, such that for each dimension q , the following equality is satisfied:

$$d_{q+1} \phi_q + \phi_{q-1} d_q = f_q - g_q$$

A *chain contraction* from $\mathcal{C} = (C, d)$ to $\mathcal{C}' = (C', d')$ is a set of three maps (f, g, ϕ) , $f = \{f_q : \mathcal{C}_q \rightarrow \mathcal{C}'_q\}$, $g = \{g_q : \mathcal{C}'_q \rightarrow \mathcal{C}_q\}$ and $\phi = \{\phi_q : \mathcal{C}_q \rightarrow \mathcal{C}_{q+1}\}$ such that for every q ([Eilenberg 54]):

- (i) f and g are chain maps, that is, $f_{q-1} d_q = d'_q f_q$ and $d_q g_q = g_{q-1} d'_q$
- (ii) fg is the identity map of \mathcal{C}' , that is, $f_q g_q = id_{\mathcal{C}'_q}$
- (iii) ϕ is a chain homotopy of the identity map of \mathcal{C} to gf , that is, $id_{\mathcal{C}_q} - g_q f_q = \phi_{q-1} d_q + d_{q+1} \phi_q$

2.5 Algebraic Discrete Morse Theory

Morse theory is a fundamental tool for analyzing the geometry and topology of smooth manifolds. Forman translated this theory to discrete structures such as cell complexes, by using discrete Morse functions or equivalently gradient vector fields (see [Forman 95, Forman 98]).

The aim of Forman's Discrete Morse Theory is to find simplicial collapses (cell pairings) that transform the initial complex to a simpler complex.

We interpret now some elementary notions of Discrete Morse Theory in terms of chain homotopy equivalences. From now on we will consider that the ring of coefficients is the finite field $\mathbb{Z}/2\mathbb{Z} = \{0, 1\}$, but most of the results are valid for any commutative field (another finite field, the rational numbers, the real numbers, etc).

Let K be a finite cell complex and σ a cell of K of dimension t ($t \in \{1, 2\}$). If u_1, \dots, u_r ($r \in \mathbb{N}$) are the $(t-1)$ -cells which form the boundary $\partial(\sigma)$ of σ , let us take $\phi_{u_i, \sigma} : \mathbb{Z}/2\mathbb{Z}[K] \rightarrow \mathbb{Z}/2\mathbb{Z}[K]$ defined by $\phi(u_i) = \sigma$ and zero elsewhere. The map $\phi_{u_i, \sigma}$ is called *cell homology collapsing* and it is a *chain homotopy operator* satisfying the properties: (a) $\phi_{u_i, \sigma} \circ \phi_{u_i, \sigma} = 0$, (2-nilpotency condition) and (b) $\phi_{u_i, \sigma} \circ \partial \circ \phi_{u_i, \sigma} = \phi_{u_i, \sigma}$ (chain contraction condition). In fact, this map generates the following chain homotopy equivalence between differential graded vector spaces (also called a chain contraction in [Eilenberg 54]) :

$$(f_{u_i, \sigma}, \text{incl}, \phi_{u_i, \sigma}) : (\mathcal{C}(K) = \mathbb{Z}/2\mathbb{Z}[K], \partial) \rightarrow f(\mathcal{C}(K)) = (f(\mathcal{C}(K)), \partial')$$
(2.1)

where $f_{u_i, \sigma} : \mathcal{C}(K) \rightarrow f(\mathcal{C}(K))$ is the map $f_{u_i, \sigma} = 1_{\mathcal{C}(K)} + \partial \circ \phi_{u_i, \sigma} + \phi_{u_i, \sigma} \circ \partial$ (being $1_{\mathcal{C}(K)} : \mathcal{C}(K) \rightarrow \mathcal{C}(K)$ the identity function on $\mathcal{C}(K)$), $f(\mathcal{C}(K)) = \{f(c) / c \in \mathcal{C}(K)\} \subset \mathcal{C}(K)$, $\text{incl} : f(\mathcal{C}(K)) \rightarrow \mathcal{C}(K)$ is the linear operator defined by $\text{incl}(z) = z$, $\forall z \in f(\mathcal{C}(K))$ and $\partial' : f(\mathcal{C}(K))_q \rightarrow f(\mathcal{C}(K))_{q-1}$ is the differential defined by $\partial'(f(c)) = f(\partial(c)) = (\partial + \partial \circ \phi_{u_i, \sigma} \circ \partial)(c)$. Concretely, $f(u_i) = \sum_{k \neq i} u_k$, $f(\sigma) = 0$ and if $\sigma' \in \mathcal{C}_t(K)$ and $\partial(\sigma') = u_i + \dots$, then $f(\sigma') = \sigma + \sigma'$ and $\partial'(f(\sigma')) = \partial(\sigma + \sigma')$. Finally, f is the identity map and $\partial' = \partial$ for the rest of cells. In Figure 2.10, a cell homology collapsing operator (represented with a red arrow) and the result after its application are shown.

It is straightforward to prove the following properties (guaranteeing in

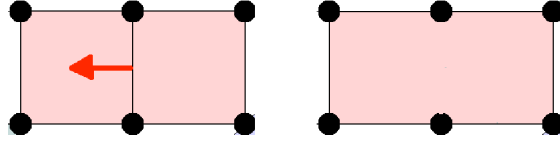


Figure 2.10: A cell homology collapsing and the resulting cell complex.

this way that $c_{u_i, \sigma} = (f_{u_i, \sigma}, \text{incl}, \phi_{u_i, \sigma})$ is a chain contraction):

$$(i) \ 1_{C(K)} + \text{incl} \circ f_{u_i, \sigma} = \partial \circ \phi_{u_i, \sigma} + \phi_{u_i, \sigma} \circ \partial$$

$$(ii) \ f_{u_i, \sigma} \circ \text{incl} = 1_{f(C(K))}$$

$$(iii) \ \phi_{u_i, \sigma} \circ \text{incl} = 0 = f_{u_i, \sigma} \circ \phi_{u_i, \sigma}$$

In particular, these properties mean that the homology groups of the differential graded vector spaces $C(K)$ and $f(C(K))$ are isomorphic. If u_i belongs to the boundary of only one cell σ ($\{u_i, \sigma\}$ is also called a *free pair*), then $f(C(K)) = (\mathbb{Z}/2\mathbb{Z}[K \setminus \{u_i, \sigma\}], \partial)$ and this chain contraction is a classical cell collapse (see Figure 2.11 for an example).



Figure 2.11: A cell complex and the resulting cell complex after applying the chain homotopy $\phi(\langle 1 \rangle) = \langle 1, 2 \rangle$ (on the left) and a cell complex and the result after applying $\phi(\langle 2, 3 \rangle) = \langle 2, 3, 4 \rangle$ (on the right).

These elementary chain homotopy operators acting on a cell complex K can be used to describe the process of homology computation for K at chain complex level. In the case of coefficients on a field, it has been proved (see [Kaczynski 98, González-Díaz 08, Molina-Abril 09b]) that the whole homology computation process can be exclusively specified by the exhaustive use of these operations.

Chapter 3

Homological Spanning Forest representation

In this chapter, we intend to exploit homology as a representation model of subdivided objects. The main aim of this idea is to obtain a combinatorial representation of a finite cell complex, from which we can deduce advanced geometrical and topological information in a efficient manner.

In order to progress in this direction, we redefine homology in such a way that the homology class of every linear combination (with coefficient in $\mathbb{Z}/2\mathbb{Z}$) of cells can be automatically determined. This is done here in terms of linear chain homotopy operators acting on cells and defining an explicit strong algebraic relationships between the chain complex and its homology groups (classically named “chain homotopy equivalences”). There are infinite homotopy operators accomplishing these conditions. We closely follow here the Effective Homology theory of Sergeraert [Sergeraert 94] that guarantees the computation of advanced topological invariants (Betti number, cohomology algebra, cohomology operations, homotopy groups,...), starting from such a chain homotopy operator.

The next step consists of codifying a chain homotopy operator in terms of trees “spanning” all the cells of the complex (considering them as vertices of these trees). The reason for employing this strategy is to try to generalize the well-known algorithm in digital imagery of connected component labelling (homology information at level 0), using the combinatorial tool of the spanning forest in the connectivity graph of cell complex.

In this section, we develop an algorithm in the context of Discrete Morse

theory in which a hierarchy of forest \mathcal{F} of this kind can be codified as chain homotopy operators and allows us to quickly compute the homology groups. This forest could a first candidate to solve the “homology codification” problem and we called it “Homological Spanning Forest” or HSF for short. The trees of this HSF representation have as vertices the different cells of the complex but, in general, its edges can not be easily described in the connectivity graph of the complex (that is, having as vertices the cells of the complex and the edges being determined by the relation “to be boundary of”).

In Chapter 4, we present an HSF representation for $2D$ images that is described in the connectivity graph of the complex. Our future aim will be to obtain such a representation for higher dimensional images.

3.1 State of the art: AT-models

The HSF approach follows the philosophy of the algebraic topological model (AT-model for short) for digital volumes presented in [González-Díaz 05b, González-Díaz 05a, González-Díaz 09]. The AT-model is based in the Effective Homology Theory developed by F. Sergeraert in [Sergeraert 94]. The main problem solved by effective homology is to make available algorithms capable of computing finite algebraic objects associated to finite topological spaces. Carefully combining functional programming methods and perturbation lemma (see [Gugenheim 89, Gugenheim 91]) gives the effective homology theory.

Roughly speaking, an AT-model is an extra algebraic-topological information of the image, that can be used to solve some image analysis problems of topological nature. This idea is mainly based on two facts: (1) to consider a simplicial model $K(I)$ for a digital volumes I using a $(14, 14)$ -adjacency relation between voxels; and (2) to apply an “algebraic homological process” in which a special type of chain homotopy equivalence (see [Mac Lane 95]) c connecting the chain complex canonically associated to the simplicial version of the digital image with its homology is constructed.

Being \mathcal{C} a chain complex, an AT-model of \mathcal{C} is defined as the set $\{C, H, f, g, \phi\}$, where:

- (i) $C = \{C_q\}$ is a base of \mathcal{C} ,
- (ii) $H = \{H_q\}$ where H_q is a subset of elements of C_q , and

- (iii) $c = (f, g, \phi)$ is a chain contraction from \mathcal{C} to the chain complex \mathcal{H} , generated by H with null differential.

The existing algorithm for computing AT-models is based on the technique presented in [Delfinado 95], where the authors propose an incremental technique that computes the Betti numbers of a topological space that is represented in terms of a simplicial complex.

For simplicial complexes in \mathbb{R}^3 , the Delfinado and Edelsbrunner's method leads to an algorithm which runs in time $O(n\alpha(n))$, where n is the number of simplexes, and $\alpha(n)$ is the extremely slowly growing inverse of the Ackermann function (see [Cormen 01]). By using an efficient representation of the simplicial complex, the time complexity can be improved to $O(n)$. The algorithm relies on a sequential ordering of simplexes in K , termed a filtration (a nested sequence of subcomplexes $0 = K_0 \subset K_1 \subset \dots \subset K_n = K$). The Betti numbers are computed incrementally as each simplex is added to the complex.

Based on that, the algorithm for computing AT-models works as follows (see [González-Díaz 05b]). Given a sorted set $(\sigma_1, \dots, \sigma_m)$ of all the simplexes in K , where any subset $\{\sigma_1, \dots, \sigma_i\}$, $i \leq m$, is a subcomplex of K , the algorithm computes a chain complex \mathcal{H} with a set of generators h and a chain contraction (f, g, ϕ) from $C(K)$ to \mathcal{H} . Initially h is an empty set, and at every step a simplex σ_i is added to the subcomplex $\{\sigma_1, \dots, \sigma_{i-1}\}$. A new homology class is created in case that $f\partial(\sigma_i) = 0$. Otherwise, a homology class “involved” in the expression $f\partial(\sigma_i)$ is removed. The pseudocode of the algorithm is Algorithm 1. An example is shown in Figure 3.1.

The algorithm allows to determine both, a representative cycle for each homology class and the homology class for each cycle. Moreover, for any q -boundary a on K , a $(q+1)$ -chain $a' = \phi(a)$ in K such that $a = \partial(a')$ can be obtained. Concerning the time complexity, the algorithm runs in time at most $O(m^3)$.

Besides the computational time complexity, the main drawback of AT-models is that they are not easy to handle. Its complicated structure makes them hard to understand, and not intuitive to work with. In the following, it will be shown that the HSF model overcomes these drawbacks.

Algorithm 1 AT-model($\sigma_1, \dots, \sigma_m$)

 $f_{top}(\sigma) = \sigma, \phi_{top}(\sigma) = 0$ for each $\sigma \in K, h = \{\}$ **for** $i = 1$ to $i = m$ **do** **if** $f_{alg}\partial(\sigma_i) = 0$ **then** $h = h \cup \{\sigma_i\}$ $f_{alg}(\sigma_i) = \sigma_i$ $\phi_{alg}(\sigma_i) = 0$ **else** take any σ_j of $f_{alg}\partial(\sigma_i)$ $h = h - \{\sigma_j\}$ $f_{alg}(\sigma_i) = 0$ $\phi_{alg}(\sigma_i) = 0$ **for** $k = 1$ to $k = m$ **do** **if** σ_j appears in $f_{alg}(\sigma_k)$ **then** $f_{alg}(\sigma_k) = f_{alg}(\sigma_k) + f_{alg}\partial(\sigma_i)$ $\phi_{alg}(\sigma_k) = \phi_{alg}(\sigma_k) + \sigma_i + \phi_{alg}\partial(\sigma_i)$ **end if** **end for** **end if****end for****for each** $\sigma \in h$ **do** $g_{alg} = \sigma + \phi_{alg}\partial(\sigma)$ **end for****return** The chain contraction $(f_{alg}, g_{alg}, \phi_{alg})$ from \mathcal{C} to \mathcal{H} .

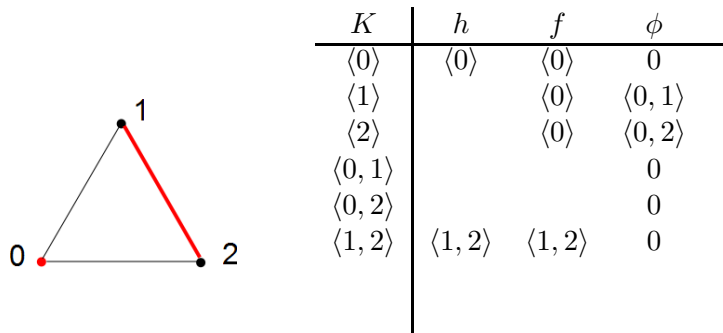


Figure 3.1: a) A simplicial complex K and b) one possible AT-model for K (where g is the inclusion map). The elements in h are coloured in red.

3.2 Intuitive idea

The HSF representation is a graph-based structure that in some sense codifies the information provided by the AT-model. Due to the richness (in a topological sense) of this information, and the clarity of its graph structure, the HSF model can be used for the development of efficient algorithms to compute and analyze geometrical and topological features of discrete objects.

In order to facilitate the understanding of this idea, we start with an elementary example of a subdivided object. Given a geometric graph G , the homology information in which we are interested can be directly captured by means of a spanning tree T of G . In fact, we transform T into a directed tree T^d by adding arrows to every edge in T , in such a way that at most one arrow comes out from each vertex. Therefore, there will be only one vertex s of G , called sink, from which no arrow comes out.

Let us take now the example of Figure 3.2, drawn on \mathbb{R}^2 . We interpret an arrow (f, e) in T^d from the vertex f to the vertex e as an elementary “deformation” operation “contracting” in a continuous way the vertex f onto e through the edge (e, f) inside the object. The result of applying (no matter the order we choose) the set of homology-preserving operations $\mathcal{V} = \{(a, f), (b, c), (c, d), (d, e), (f, e)\}$ on G is a reduced (in terms of bricks) subdivided structure consisting of only three bricks: the vertex e , and two loops or “edges” starting and ending at the same common vertex e (in fact, they represent the cycles $\{(c, f), (e, f), (d, e), (c, d)\}$ and $\{(a, b), (a, f), (e, f), (d, e), (c, d), (b, c)\}$ coming from (c, f) and (a, b) , respectively).

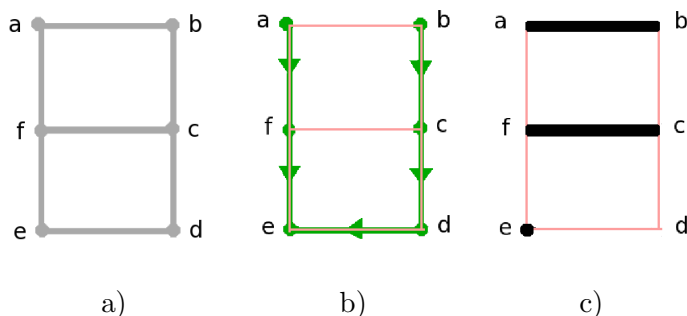


Figure 3.2: a) A geometric graph G drawn on \mathbb{R}^2 , b) a directed spanning tree (in green) showing a homological "deformation" process and c) the minimal homological object (in black).

The directed spanning tree T^d can be interpreted in dynamical terms, as the way in which the set of vertices of the graph is "collapsed" to a representative vertex of the connected component (in this case, the vertex e in black).

The three representative cycles of homology generators (in this case, no matter of the ground ring we use but heavily dependent on the spanning tree T) are determined by the following bricks of G (called *critical*): the edges (c, f) , (a, b) belonging to $G \setminus T$, and the sink vertex e belonging to T .

The integer homology groups in this example are one copy of \mathbb{Z} in dimension 0 and two copies in dimension 1.

One of the multiple possible Homological Spanning Forest or HSF representation $\mathcal{F}(G)$ for the subdivided geometric structure G is the set of coordinate-based trees $\mathcal{F}(G) = \{T^d, T_1, T_2\}$, where T_1 and T_2 are trees composed by only one "vertex": the original edges (a, b) and (c, f) respectively.

3.3 HSF for finite cell complexes

In this section, we extract the homological nature of an HSF structure for a finite cell complex, which is half-way between the combinatorial notion of an optimal gradient vector field (Discrete Morse Theory) and the classical algebraic concept of chain contraction (Effective Homology Theory) or even the less elaborate abstraction of a chain homotopy operator (AT-model theory).

In Figure 3.2, it is shown that the spanning tree together with the two

critical bricks can be seen as a non-redundant structure $M(G)$ which represents the geometric graph G in a combinatorial manner. There is a strong algebraic relationship between $M(G)$ and G that can be described by means of a chain contraction between them with a chain homotopy operator specified by the HSF structure. Denoting by $\mathbb{Z}/2\mathbb{Z}[G]$ the graded vector space formed by the finite linear combinations of 0-cells and 1-cells of G , we can formalize this chain homotopy operator:

$$\phi_M : \mathbb{Z}/2\mathbb{Z}[G] \rightarrow \mathbb{Z}/2\mathbb{Z}[G] \quad (3.1)$$

with $\phi_M(a) = (a, f) + (f, e)$, $\phi_M(b) = (b, c) + (c, d) + (d, e)$ and, for the rest of vertices, $\phi_M(x)$ is the sum of arrows of the directed path from x to e following the “maximal paths” $\phi_M(a)$ or $\phi_M(b)$. Concerning the vertex e , $\phi_M(e) = 0$. The evaluation of ϕ_M on each of the edges of G is zero.

Using ϕ_M , the following operators can be defined:

1. *The flow, defined as $f_M = \mathbf{1} + \phi_M \circ \partial + \partial \circ \phi_M : \mathbb{Z}/2\mathbb{Z}[G] \rightarrow \mathbb{Z}/2\mathbb{Z}[G]$, where $\mathbf{1}$ and ∂ are functions from $\mathbb{Z}/2\mathbb{Z}[G]$ to $\mathbb{Z}/2\mathbb{Z}[G]$ denoting the identity and the boundary operator of the graph cell complex G , respectively.*
2. *The inclusion operator $incl : \mathbb{Z}/2\mathbb{Z}[C_0, C_1, C_2] \rightarrow \mathbb{Z}/2\mathbb{Z}[G]$, where C_i ($i \in \{0, 1, 2\}$) are the homology representative cycles.*

In Figure 3.2, $f_M(x) = \{C_0 = e\}$ for each vertex x of G and $f_M((x, y)) = 0$ for each edge (x, y) in the spanning tree T . The flow of the critical edges (a, b) and (f, c) are, respectively the cycles $C_1 = (a, b) + (b, c) + (c, d) + (d, e) + (e, f) + (f, a)$ and $C_2 = (c, f) + (c, d) + (d, e) + (e, f)$.

Therefore, under these conditions, any cell c of G can be obtained as a sum of a homology representative cycle C_i ($i \in \{0, 1, 2\}$) and the homologically inessential linear combination $(\partial \circ \phi_M + \phi_M \circ \partial)(c)$. The cycles C_i ($i \in \{0, 1, 2\}$) are invariant linear combinations of cells through the flow.

In other words, the triple $(f_M, incl, \phi_M)$ is a chain contraction from the chain complex $C(G)$ to $M(G) = \mathbb{Z}/2\mathbb{Z}[C_0, C_1, C_2]$, which is entirely determined by the chain homotopy operator ϕ_M .

In any HSF representation of K , if exists, its 0-dimensional homological trees specify (not considering the arrows) a barycentre subdivision of a spanning forest of the 1-dimensional skeleton $K^{(1)}$ of K . In some way, this

decomposition method can be seen as a “natural” extension to higher dimension of the graph-based techniques for computing the spanning forest of a graph complex and its corresponding zero and one dimensional homology groups. For that reason, we have named this decomposition “Homological Spanning Forest”.

Informally speaking, an HSF representation of K is a set of trees constructed on the skeleton of K , from which we can directly extract a chain homotopy operator connecting K with its homology. A first formal definition of HSF will be given in Section 3.6.

3.4 Integral-chain complexes

In order to completely understand the construction and usefulness of the HSF representation, some concepts need to be defined. As mentioned before, some of these notions are underlying in the work of Sergeraert [Sergeraert 94], Forman [Forman 95, Forman 98] and that of Theory of Discrete Differential Forms [Desbrun 05].

Throughout this Section, we consider a field as the coefficient ring.

Definition 3.4.1 *An integral chain complex (C, d, ϕ) is a graded module $C = \{C_p\}_{p=0}^n$ endowed with two linear maps: a differential operator $d : C_* \rightarrow C_{*-1}$, and an integral operator $\phi : C_* \rightarrow C_{*+1}$, satisfying the global nilpotency properties $d \circ d = 0$ and $\phi \circ \phi = 0$.*

This integral operator, can also be called *chain homotopy operator* (see [Eilenberg 54]). We will represent an integral operator by an arrow from the cell of lower dimension to the cell of higher dimension (see Figure 3.3 and Table 3.1).

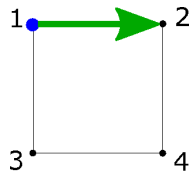


Figure 3.3: An integral chain complex and an integral operator (represented by an arrow)

σ	d	ϕ
$\langle 1 \rangle$	0	$\langle 1, 2 \rangle$
$\langle 2 \rangle$	0	0
$\langle 3 \rangle$	0	0
$\langle 4 \rangle$	0	0
$\langle 1, 2 \rangle$	$\langle 1 \rangle + \langle 2 \rangle$	0
$\langle 1, 3 \rangle$	$\langle 1 \rangle + \langle 3 \rangle$	0
$\langle 2, 4 \rangle$	$\langle 2 \rangle + \langle 4 \rangle$	0
$\langle 3, 4 \rangle$	$\langle 3 \rangle + \langle 4 \rangle$	0

Table 3.1: The d and ϕ values of the complex shown in Figure 3.3

Definition 3.4.2 An integral chain complex (C, d, ϕ) is d -pure if the condition $d = d \circ \phi \circ d$ (called homology condition) is satisfied. An integral chain complex (C, d, ϕ) is ϕ -pure if the condition $\phi = \phi \circ d \circ \phi$ (called Strong Deformation Retract condition) is satisfied. An integral chain complex that is both, d -pure and ϕ -pure, is called homology integral chain complex. In this case, d (resp. ϕ) is a homology differential (resp. integral) operator.

For instance, the integral chain complex in Figure 3.4, is a homology integral chain complex (the conditions $d = d \circ \phi \circ d$ and $\phi = \phi \circ d \circ \phi$ are satisfied for every cell of the complex). The d and ϕ values of the complex are shown in Table 3.2

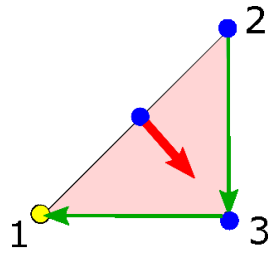


Figure 3.4: A homology integral chain complex. The homology integral operator is represented by arrows.

Given two integral chain complexes (C, d, ϕ) and (C', d', ϕ') , a map of integral chain complexes $(f, g) : (C, d, \phi) \Rightarrow (C', d', \phi')$ is a couple of linear maps $f : C \rightarrow C'$ and $g : C' \rightarrow C$ such that $f \circ d = d' \circ f$, $g \circ d' = d \circ g$, $f \circ \phi = \phi' \circ f$ and $g \circ \phi' = \phi \circ g$.

σ	d	ϕ
$\langle 1 \rangle$	0	0
$\langle 2 \rangle$	0	$\langle 2, 3 \rangle + \langle 1, 3 \rangle$
$\langle 3 \rangle$	0	$\langle 1, 3 \rangle$
$\langle 1, 2 \rangle$	$\langle 1 \rangle + \langle 2 \rangle$	$\langle 1, 2, 3 \rangle$
$\langle 1, 3 \rangle$	$\langle 1 \rangle + \langle 3 \rangle$	0
$\langle 2, 3 \rangle$	$\langle 2 \rangle + \langle 3 \rangle$	0
$\langle 1, 2, 3 \rangle$	$\langle 1, 2 \rangle + \langle 1, 3 \rangle + \langle 2, 3 \rangle$	0

Table 3.2: The d and ϕ values of the complex shown in Figure 3.4

Definition 3.4.3 Given two integral chain complexes (C, d, ϕ) and (C', d', ϕ') , we say that they are integral chain equivalent if there exists a map of integral chain complexes (f, g) , such that $f \circ g = id_{C'} - d' \circ \phi' - \phi' \circ d'$ and $g \circ f = id_C - d \circ \phi - \phi \circ d$.

Two integral chain equivalent complexes are shown in Figure 3.5.

The homology $H_*(C, d, \phi)$ of an integral chain complex (C, d, ϕ) is the graded abelian group $H_*(C)$, such that $(H_*(C), 0, 0)$ is integral chain equivalent to (C, d, ϕ) . The differential (resp. integral) homology of an integral chain complex (C, d, ϕ) is the homology of $(C, d, 0)$ (resp. the homology of $(C, 0, \phi)$). If (C, d, ϕ) is a homology integral-chain complex, then $H_*(C, d, \phi) \sim H_*(C, d, 0) \sim H_*(C, 0, \phi)$.

The integral chain equivalence relation can be seen as the natural extension of the classical chain homotopy equivalence between chain complexes to the integral case (see, for example, [Eilenberg 54]).

The computation of the homology of a chain complex (C, d) can be directly obtained from an integral operator $\phi : C_* \rightarrow C_{*+1}$, satisfying the Strong Deformation Retract (SDR, for short) and homology conditions with regards to the differential operator d ([Gugenheim 89, Gugenheim 91]).

Proposition 1 Let (C, d, ϕ) be an integral chain complex. Let $\pi : C_* \rightarrow C_*$ be the linear map (called the flow of (C, d, ϕ)) defined by $\pi = id_C - d \circ \phi - \phi \circ d$ and let $\Delta : C_* \rightarrow C_*$ be the linear map (called Laplacian of (C, d, ϕ)) defined by $\Delta = d \circ \phi + \phi \circ d$. Then, the following properties hold:

- (i) $d \circ \pi = d - d \circ \phi \circ d = \pi \circ d$ and $\phi \circ \pi = \phi - \phi \circ d \circ \phi = \pi \circ \phi$. In the case of a homology integral chain complex, $d \circ \pi = 0 = \pi \circ d$ and $\phi \circ \pi = 0 = \pi \circ \phi$.

(ii) $d \circ \Delta = d \circ \phi \circ d = \Delta \circ d$ and $\phi \circ \Delta = \phi \circ d \circ \phi = \Delta \circ \phi$. In the case of a homology integral chain complex, $d \circ \Delta = d = \Delta \circ d$ and $\phi \circ \Delta = \phi = \Delta \circ \phi$.

(iii) Given a q -chain a , we have the following equality $a = \pi(a) + \Delta(a)$.

(iv) $\pi^2 = \pi - \phi \circ (d - d \circ \phi \circ d) - (d - d \circ \phi \circ d) \circ \phi = \pi - d \circ (\phi - \phi \circ d \circ \phi) - (\phi - \phi \circ d \circ \phi) \circ d$.

(v) $\Delta^2 = (d + \phi)\Delta(d + \phi)$.

(vi) $\pi \circ \Delta = (d - d \circ \phi \circ d) \circ \phi + \phi \circ (d - d \circ \phi \circ d) = d \circ (\phi - \phi \circ d \circ \phi) + (\phi - \phi \circ d \circ \phi) \circ d = \Delta \circ \pi$.

Definition 3.4.4 The integral chain complex $\pi(C, d, \phi) = (\pi(C), d|_{\pi(C)}, \phi|_{\pi(C)})$ is the harmonic complex associated to (C, d, ϕ) . If (C, d, ϕ) is a d -pure or a ϕ -pure integral chain complex, then $\pi^2 = \pi \circ \pi = \pi$ and $\pi(C) = \{x \in C | x = \pi(x)\}$.

In other words, the harmonic complex $(\pi(C), d|_{\pi(C)}, 0)$ associated to a pure integral chain complex (C, d, ϕ) is formed by the π -equivariant chains of C . If (C, d, ϕ) is a homology integral chain complex, its harmonic complex is of the kind $(\pi(C), 0, 0)$ and given any q -chain the chain map π describes a representative cycle of the homology class associated to this q -chain.

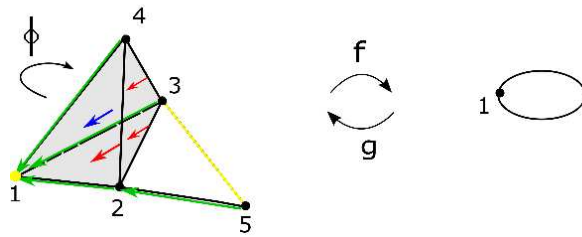


Figure 3.5: Two integral chain equivalent complexes, (C, d, ϕ) on the left and (C', d', ϕ') on the right

In Figure 3.5 two integral chain equivalent complexes (C, d, ϕ) and (C', d', ϕ') are shown. The complex (C, d, ϕ) on the left is a ϕ -pure integral chain complex. The complex (C', d', ϕ') on the right is the harmonic complex

σ	$d(\sigma)$	$\phi(\sigma)$
$\langle 2 \rangle$		$\langle 1, 2 \rangle$
$\langle 3 \rangle$		$\langle 1, 3 \rangle$
$\langle 4 \rangle$		$\langle 1, 4 \rangle$
$\langle 5 \rangle$		$\langle 1, 2 \rangle + \langle 2, 5 \rangle$
$\langle 2, 3 \rangle$	$\langle 2 \rangle + \langle 3 \rangle$	$\langle 1, 2, 3 \rangle$
$\langle 2, 4 \rangle$	$\langle 2 \rangle + \langle 4 \rangle$	$\langle 1, 2, 4 \rangle$
$\langle 3, 4 \rangle$	$\langle 3 \rangle + \langle 4 \rangle$	$\langle 1, 3, 4 \rangle$
$\langle 2, 3, 4 \rangle$	$\langle 2, 3 \rangle + \langle 2, 4 \rangle$ $+ \langle 3, 4 \rangle$	$\langle 1, 2, 3, 4 \rangle$

Table 3.3: The d and ϕ values of the complex (C, d, ϕ) shown in Figure 3.5

of the complex (C, d, ϕ) . (C', d', ϕ') is a homology integral chain complex ($d'(\sigma) = 0, \phi'(\sigma) = 0 \forall \sigma \in C'$). The d and ϕ values of the complex (C, d, ϕ) are shown in Table 3.3

Definition 3.4.5 *The integral chain complex $\Delta(C, d, \phi) = (\Delta(C), d|_{\Delta(C)}, \phi|_{\Delta(C)})$ is the Laplacian complex associated to (C, d, ϕ) . If (C, d, ϕ) is a d -pure or ϕ -pure integral chain complex, then $\Delta \circ \Delta = \Delta$ and $\Delta(C) = \{x \in C | x = \Delta(x)\}$.*

In other words, the Laplacian complex $\Delta(C, d, \phi)$ associated to a pure integral chain complex (C, d, ϕ) is formed by all the Δ -equivariant chains.

Proposition 2 *If (C, d, ϕ) is a (differential or integral) pure integral-chain complex, we can derive the following properties:*

$$(i) \pi \circ \Delta = 0 = \Delta \circ \pi.$$

(ii) $(C, d, \phi) = \pi(C, d, \phi) \oplus \Delta(C, d, \phi)$ as integral-chain complexes. In particular, $\text{Ker } \Delta = \pi(C)$ and $\Delta(C) = \text{Ker } \pi$.

(iii) $\Delta(C) = \phi(C) \oplus (d \circ \phi)(C)$ as graded modules.

In order to emphasize the dependency of π and Δ with regards to d and ϕ , we will denote these maps by $\pi_{(d, \phi)}$ and $\Delta_{(d, \phi)}$, respectively.

The following proposition will be fundamental in developing an HSF framework. In fact, it shows that to use pure integral operators as chain homotopies decomposing finitely generated chain complexes is a key point:

Proposition 3 *If (C, d, ϕ) is a (differential or integral) pure integral-chain complex, we have that*

$$\text{Ker } \phi \cong \pi(C) \oplus \phi(C) \cong \text{Ker } \Delta(C) \oplus \phi(C)$$

as graded modules.

In particular, a map of integral-chain complexes (f, g) satisfies that $f \circ \pi_{(d, \phi)} = \pi_{(d', \phi')} \circ f$, $g \circ \pi_{(d', \phi')} = \pi_{(d, \phi)} \circ g$, $f \circ \Delta_{(d, \phi)} = \Delta_{(d', \phi')} \circ f$ and $g \circ \Delta_{(d', \phi')} = \Delta_{(d, \phi)} \circ g$. That is, f and g are compatible with regards to the respective flows and Laplacians.

In spite of its simplicity, the following result is essential for developing the homological theory of integral-chain complexes:

Lemma 1 [*Integral-Chain Lemma*] *An integral chain complex (C, d, ϕ) is integral-chain equivalent to its harmonic complex $\pi(C, d, \phi)$. This last harmonic complex $\pi(C, d, \phi)$ is of the form $(\pi(C), d_\pi, \phi_\pi)$ where $d_\pi(\pi(x)) = (d - d \circ \phi \circ d)(x)$ and $\phi_\pi(\pi(x)) = (\phi - \phi \circ d \circ \phi)(x)$.*

Proof. Let $f : C \rightarrow \pi(C)$ be the linear map defined by $f(x) = \pi(x)$, $\forall x \in C_*$. Let $g : \pi(C) \rightarrow C$ be the linear map defined by $g(x) = x$, $\forall x \in \pi(C)$. Then, it is a simple exercise to show that (f, g) is a couple of maps of integral chain complexes which induces the integral-chain equivalence. The rest of assertions can be directly deduced from Proposition 1 (i).

◇

Corollary 1 *The harmonic complex $\pi(C, d, \phi)$ associated to a d -pure (resp. ϕ -pure) integral-chain complex (C, d, ϕ) is of the form $(\pi(C), 0, \phi_\pi)$ (resp. $(\pi(C), d_\pi, 0)$), $\phi_\pi(\pi(x)) = (\phi - \phi \circ d \circ \phi)(x)$ (resp. $d_\pi(\pi(x)) = (d - d \circ \phi \circ d)(x)$).*

An example of Lemma 1 can be seen in Figure 3.5, where (C, d, ϕ) is a ϕ -pure integral-chain complex, and (C', d', ϕ') is its harmonic complex.

Now, we give some definitions related to integral-chain perturbation of complexes.

Definition 3.4.6 *An integral chain complex (C, d, ϕ) is called d -pointwise nilpotent (resp. ϕ -pointwise nilpotent) if for any $a \in C$ there is some $n(a) \in \mathbb{N}$ with $d \circ (id_C - d \circ \phi - \phi \circ d)^{n(a)} = 0$ (resp. with $\phi \circ (id_C - d \circ \phi - \phi \circ d)^{n(a)} = 0$). The smallest value for $n(a)$ is called the degree of differential (resp. integral) nilpotency of a .*

Proposition 4 *Given an ϕ -pointwise (resp. d -pointwise) nilpotent chain complex (C, d, ϕ) , it is integral chain equivalent to a ϕ -pure (resp. d -pure) integral chain complex $(C, d, \tilde{\phi})$ (resp. (C, \tilde{d}, ϕ)).*

Proof. We only prove the existence of the ϕ -pure integral chain complex $(C, d, \tilde{\phi})$. The other result can be derived directly from the fact that (C, ϕ, d) is also a ϕ -pointwise nilpotent integral-chain complex. Define $\tilde{\phi} : C_* \rightarrow C_{*+1}$ by $\tilde{\phi} = \sum_{k \geq 0} \phi \circ (id_C - d \circ \phi)^k$. This is well defined due to the pointwise nilpotency of (C, d, ϕ) , since all but finitely many terms vanish on the right hand side. The map $\tilde{\phi}$ satisfies $\tilde{\phi} \circ \tilde{\phi} = 0$ and the SDR condition $\tilde{\phi} \circ d \circ \tilde{\phi} = \tilde{\phi}$. The couple of maps $(\pi_{(d, \tilde{\phi})}, id_C)$ establishes the integral chain equivalence between (C, d, ϕ) and $(C, d, \tilde{\phi})$. \diamond

From now on, all the integral chain complexes that we consider will be differential or integral pointwise nilpotent. Analogous results can be determined for differential pointwise nilpotent.

Now, let us determine some algebra constructors for integral chain complexes: the composition and the differential/integral perturbation.

Homological Perturbation theory [Gugenheim 89, Gugenheim 91] is an efficient technique for transferring structures from one object to another up to homotopy. First, let us define the notion of perturbation of a pure integral chain complex:

Definition 3.4.7 *Let (C, d, ϕ) be a pointwise nilpotent integral chain complex. Let $\delta : C_* \rightarrow C_{*-1}$ (respectively, $\psi : C_* \rightarrow C_{*+1}$) be a map such that:*

- (i) δ (resp. ψ) is a linear map of chains.
- (ii) (Global 2-nilpotency) $(d + \delta)^2 = 0$ (resp. $(\phi + \psi)^2 = 0$).
- (iii) (Pointwise δ -nilpotency) For any $a \in C$ there is some $m(a) \in \mathbb{N}$ with $(\phi \circ \delta)^{m(a)}(a) = 0$ (resp. there is some $n(a) \in \mathbb{N}$ with $(\phi \circ \psi)^{n(a)} = 0$). The smallest value for $m(a)$ (resp. for $n(a)$) is referred to as the degree of δ -nilpotency of a (resp. the degree of ψ -nilpotency of a).

We call δ (resp. ψ) a differential perturbation (resp. an integral perturbation) of the integral chain complex (C, d, ϕ) .

Theorem 1 (Basic (differential or integral) Perturbation Lemma) *Given a d -pure (resp. a ϕ -pure) integral chain complex (C, d, ϕ) and an integral perturbation ψ (resp. a differential perturbation δ), $(C, d_\psi, \phi + \psi)$ (resp.*

$(C, d + \delta, \phi_\delta)$ is a new d -pure integral-chain complex (resp. a ϕ -pure integral chain complex). The new map $d_\psi : C_* \rightarrow C_{*-1}$ (resp. $\phi_\delta : C_* \rightarrow C_{*+1}$) is given by the following formula:

$$d_\psi = \Sigma_d^\psi \circ d, \quad (\text{resp. } \phi_\delta = \Sigma_\phi^\delta \circ \phi),$$

where $\Sigma_f^g = \sum_{i \geq 0} (-1)^i (f \circ g)^i$. Let us note that $\Sigma_d^\psi(a)$ and $\Sigma_\phi^\delta(a)$ are respectively finite sums for each $a \in C$, due to the pointwise ψ and δ -nilpotency properties.

Proposition 5 *The set of d -pure (resp. ϕ -pure) integral chain complexes is closed by basic integral (resp. differential) perturbation.*

Finally, we define the composition of an integral-chain complex that can be derived as a simple consequence of the Integral-Chain Lemma 1.

Definition 3.4.8 *Given a d -pure (resp. a ϕ -pure) integral-chain complex (C, d, ϕ) and a differential operator d' satisfying the homology condition (resp. an integral operator ϕ' satisfying the strong deformation retract condition) for $\pi_{(d, \phi)}(C)$, a new d -pure (resp. ϕ -pure) integral chain complex $(C, d + d' \circ \pi_{(d, \phi)}, \phi)$ (resp. $(C, d, \phi + \phi' \circ \pi_{(d, \phi)})$) can be constructed. This new integral chain complex is called composition of (C, d, ϕ) by d' (resp. by ϕ'). A d -pure integral chain complex (C, d, ϕ) can suffer composition with regards to the differential operator d (resp. with regard to the own integral operator ϕ) restricted to $\pi_{(d, \phi)}(C)$.*

3.5 HSF and Discrete Morse Theory

As mentioned in Section 2.5, the aim of Discrete Morse Theory (DMT for short) is to find simplicial collapses that transform a complex K to a simpler one. One of the main instruments of Discrete Morse theory, that are gradient vector fields, can be interpreted in terms of chain homotopy equivalences. Once a discrete gradient vector field has been defined on a finite cell complex, information about its homology can be directly deduced from it. The idea and construction of the HSF representation are strongly related with Discrete Morse Theory and its gradient vector fields. In order to clarify this relation and proceed with the HSF construction algorithm, some concepts need to be defined.

Now, we give some basic notions of DMT with some slight modifications and without using, in principle, discrete Morse functions.

Definition 3.5.1 A discrete vector field \mathcal{V} defined on a connected cell complex K is a pairwise disjoint collection of sets of two incident cells $\{\alpha^{(q)} < \beta^{(q+1)}\}$.

Definition 3.5.2 A \mathcal{V} -path or gradient path γ is an alternating sequence of cells $a_0^{(q)}, b_0^{(q\pm 1)}, a_1^{(q)}, b_1^{(q\pm 1)}, a_2^{(q)}, \dots$, such that for each pair of consecutive cells, one is a facet of the other, and the following condition is satisfied: either $\{a_i^{(q)} < b_i^{(q\pm 1)}\}$ or $\{b_i^{(q\pm 1)} < a_{i+1}^{(q)}\}$ belongs to γ , $\forall i \geq 0$.

If the final cell in the gradient path γ above is $\alpha_r^{(q)}$, then we say that γ has length r ([Forman 02]). If it ends at $\beta_r^{(q\pm 1)}$ then we say that γ has length $r\frac{1}{2}$. If the cells b_i of the gradient path γ are of dimension $q + 1$ and it has length $r\frac{1}{2}$, the gradient path γ is called *upper integral path*. For any cells a and b , let $\Gamma(a, b)$ denote the set of gradient paths from a to b (of any length), i.e., such that the first cell in the sequence is a and the last cell in the sequence is b . A \mathcal{V} -path is non trivial and closed if $r \geq 1$ and the first and last cells in the sequence are the same.

Definition 3.5.3 A discrete gradient vector field \mathcal{V} is a discrete vector field with non trivial closed \mathcal{V} -paths. In this way, it can be seen as an acyclic cells pairing. A cell α is a critical cell of \mathcal{V} if it is not paired with any other cell in \mathcal{V} .

Definition 3.5.4 A combinatorial integral operator defined on a cell complex K is a pairwise disjoint collection of sets of two (not necessary incident) cells $\{\alpha^{(q)}, \beta^{(q+1)}\}$ of the same connected component.

Therefore, a discrete gradient vector field is a special kind of combinatorial integral operator.

Forman proved [Forman 95, Forman 98] that the topology of a discrete manifold is related to the critical cells of a discrete function defined on it, mimicking the results of Morse in the smooth case. The number of critical cell depends on the discrete gradient vector field considered (see Figure 3.6). In [Lewiner 03], the problem of the optimality (that is minimizing the number of critical cells for discrete vector fields) on a 2-manifold is analyzed

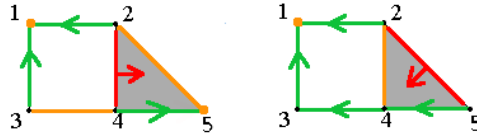


Figure 3.6: A cell pairing on the left ($\langle 1, \langle 5 \rangle$, $\langle 3, 4 \rangle$ and $\langle 2, 5 \rangle$ are critical), and an optimal one on the right ($\langle 1 \rangle$ and $\langle 2, 4 \rangle$ are critical). The pairing is represented with an arrow from the cell of lower dimension to its paired cell of higher dimension

using Hasse diagram and hypergraph tools. However, this problem has not been solved for the general case.

If we restrict ourselves to use DMT techniques, it is not always possible to obtain a number of critical cells that coincides with the Betti numbers of the complex. This is the case of the Bing’s house and the Dunce hat complexes, that are contractible but not collapsible (see [Ayala 10]).

The next results will show that by using integral operators for chain complexes, we can solve this problem, and always reduce the initial complex to the minimum number of critical cells, that corresponds with the Betti numbers. This means that we are able to guarantee homological optimality (what is called perfection in the DMT context, see [Ayala 10]).

Definition 3.5.5 *Given an upper integral path $\gamma = \Gamma(a, b)$ formed by the alternating sequence $\gamma : a_0^{(q)}, b_0^{(q+1)}, a_1^{(q)}, b_1^{(q+1)}, \dots, a_t^{(q)}, b_t^{(q+1)}$ its corresponding chain homotopy path is defined by the sequence $b_0^{(q+1)}, b_1^{(q+1)}, \dots, b_t^{(q+1)}$.*

First, the combinatorial integral operators derived from discrete vector fields are ϕ -pointwise nilpotent.

Proposition 6 *A discrete gradient vector field \mathcal{V} gives rise to a ϕ -pointwise nilpotent integral-chain complex $(C(K), d, V)$.*

Proof. Let us emphasize that two pairs $\{a, b\}$ and $\{a', b'\}$ of \mathcal{V} have no elements in common. The discrete vector field \mathcal{V} gives rise to a linear map $V : C_*(K) \rightarrow C_{*+1}(K)$, defined by $V(a) = b$ if $\{a, b\} \in \mathcal{V}$ and $V(a) = 0$ for the rest of the cells. It is clear that $V \circ V = 0$. The map V is an integral operator for $C_*(K)$. It is straightforward to prove that $(C(K), d, V)$ is a ϕ -pointwise integral-chain complex. \diamond

Combining Prop. 1, Prop. 2 and Prop. 3, we assert the following result which is the key for reinterpreting DMT in terms of an integral-chain complex:

Proposition 7 *If (C, d, V) is a ϕ -pointwise nilpotent integral-chain complex being V a discrete gradient vector field, then there is an integral-chain equivalent ϕ -pure complex (C, d, \tilde{V}) , such that its harmonic complex $\pi(C, d, \tilde{V}) = (\text{Ker } \tilde{V} \setminus \tilde{V}(C), d_\pi, 0)$.*

This last integral-chain complex is constituted by finite linear combinations of the different critical cells of V and d_π can be seen as the boundary operator of the corresponding cell complex determined by the critical cells, also called harmonic Morse cell complex $\mathcal{M}(C, d, V)$ associated to (C, d, V) . Analogously, the Laplacian complex $\Delta(C, d, \tilde{V})$ can be seen as the acyclic chain complex of the cell complex $\mathcal{M}(C, d, V)$, also called *Laplacian Morse complex* associated to (C, d, V) . Moreover, its boundary operator $\partial_{\mathcal{M}}$ is determined by $\partial_{\mathcal{M}}(\Delta(\sigma^{(q)})) = d \circ \tilde{V} \circ d(\sigma^{(q)})$, $\forall \sigma^{(q)} \in C$.

Proof. Due to Prop. 3 and Prop. 1 and defining $\tilde{V} : C_* \rightarrow C_{*+1}$ by $\tilde{V} = \sum_{k=0}^t V \circ (id_C - d \circ V)^k = \sum_{k=0}^t (id_C - V \circ d)^k V$, we have that

$$(f, g) : \text{Ker } \tilde{V} \setminus \tilde{V}(C) \cong \pi(C, d, \tilde{V})$$

is an isomorphism of chain complexes, with $f : \text{Ker } \tilde{V} \setminus \tilde{V}(C) \rightarrow \pi(C, d, \tilde{V})$ and $g : \pi(C, d, \tilde{V}) \rightarrow \text{Ker } \tilde{V} \setminus \tilde{V}(C)$ respectively defined by $f(\sigma) = \pi_{(d, \tilde{V})}(\sigma) = \sigma - \tilde{V} \circ d(\sigma)$, $\forall \sigma \in \text{Ker } \tilde{V} \setminus \tilde{V}(C)$ and $g(\pi_{(d, \tilde{V})}(\beta)) = \beta - d \circ \tilde{V}(\beta)$, $\forall \beta \in C$.

Now, let us prove that $\text{Ker } \tilde{V} = \text{Ker } V$ and $\tilde{V}(C) = V(C)$.

It is clear that $\text{Ker } V \subset \text{Ker } \tilde{V}$. Let $x \in \text{Ker } \tilde{V}$ be an element such that $x \notin \text{Ker } V$. That means that $\sum_{k=0}^t \pi(d, V)^k(x) \in \text{Ker } V$. That implies that $V(id_C - \pi^{t+1})(x) = V(d \circ V + V \circ d) \circ (\sum_{k=0}^t \pi(d, V)^k)(x) = 0$ and, then $V(x) = V\pi^{t+1}(x) = 0$. In a similar manner, it is possible to deduce that $\tilde{V}(C) = V(C)$ and that it also admits a combinatorial basis.

Let us now prove that the chain complex $(\tilde{V}(C) \oplus (d \circ \tilde{V})(C), d)$ is acyclic. We have that $d(\tilde{V}(C)) \subset (d \circ \tilde{V})(C)$ and $d((d \circ \tilde{V})(C)) = 0$. Now, let us suppose that there is a chain $x = x' + x'' \in \tilde{V}(C) \oplus (d \circ \tilde{V})(C)$ such that $d(x) = 0$. This means that $d(x') = 0$. Since $x' = \tilde{V}(z)$, then $(d \circ \tilde{V})(z) = 0$. Due to the fact that the integral operator satisfy the SDR condition $\tilde{V} \circ d \circ \tilde{V} = \tilde{V}$, we conclude that $x' = 0$.

Finally, the boundary operator of $\mathcal{M}(C, d, V)$ is the differential operator d restricted to the complex and its acyclicity can be proved using Prop. 3.

◇

Let us note that $H_*(\mathcal{M}(C, d, V)) \cong H_*(K, \Lambda)$. Moreover, the boundary operator d_π of the Morse cell complex $\mathcal{M}(C, d, V)$ has a clear interpretation in terms of gradient paths of \tilde{V} .

Proposition 8 *In the conditions of Prop. 7, and given a q -cell α , $\tilde{V}(\alpha)$ is a chain homotopy path.*

In [Molina-Abril 09a], an integral operator ϕ giving rise to a homology integral chain complex is determined from a filtered cell complex by using an incremental technique. Given a q -cell σ , $\phi(\sigma)$ is a sum of $(q + 1)$ -cells in which at least one cell τ satisfies that $\sigma \in \partial(\tau)$. This operator ϕ gives rise in a natural way to a combinatorial integral operator on K .

Due to the fact that $\tilde{V}(C)$ admits a combinatorial basis, and the chain complex $\tilde{V}(C) \oplus (d \circ \tilde{V}(C), d)$ is acyclic, we can assume that the sum ω of the elements in the combinatorial basis of $\tilde{V}(C)$ satisfies that $d(\omega) = 0$. That means that ω can be represented in terms of graphs using trees. In these trees, the nodes are q -cells and $(q + 1)$ -cells $\forall q \geq 0$ of the complex. The neighbours of a q -cell are $(q + 1)$ -cells and vice versa (see Figure 3.7). This forest, that is a representation in homological terms of the cell complex K , is an homological spanning forest representation for K (see [Molina-Abril 09b, Molina-Abril 10, Molina-Abril 12a]).

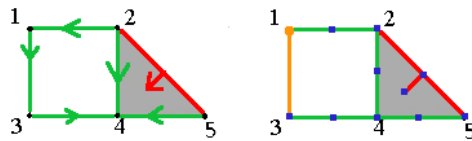


Figure 3.7: A discrete vector field (on the left). On the right a gradient set of trees where cells $\langle 1 \rangle$ and $\langle 1, 3 \rangle$ do not belong to the forest, $\langle 2, 5 \rangle$ and $\langle 2, 4, 5 \rangle$ belong to the tree of dimension $1 - 2$ and the rest of cells belong to the tree of dimension $0 - 1$

Given an HSF, it is possible to distinguish two kind of trees: homologically essential and inessential trees. In a homologically inessential tree the

number of q -cells is the same as the number of $(q + 1)$ -cells. In a homologically c -essential tree, the difference between the number of q -cells and $(q + 1)$ -cells is a positive integer c . In this last case there exist c q -cells within this tree that represent a critical cell, that is, a homology generator. Therefore, given an HSF, a combinatorial integral operator can be directly deduced by maximally pairing each q -cell with a $(q + 1)$ -cell using some specific strategy (eventually, allowing the pairing of non-incident cells) for each homologically essential or inessential tree. In this process, only c q -cells (critical cells) of a homologically c -essential tree will remain unpaired.

Let us emphasize that the notion of optimality here is guaranteed in terms of finding a homology integral operator. Therefore, the minimum number of critical cells will always coincide with the Betti numbers. In the pairing process, we might find some pairs of non-incident cells $\{\alpha, \beta\}$. In order to obtain optimality in the sense of Forman (pairing of incident cells), classical cancellation results (see [Forman 98]) involving the single path joining α and β can be applied.

3.6 HSF algorithm: A first HSF formal definition and computation

Let us remind that the main goal of this work, and therefore of the HSF, is twofold: (a) on one hand, to easily deduce from the HSF model a chain homotopy operator algebraically connecting K with its homology groups $H(K)$; and (b) on the other hand to combinatorially represent a cell complex K . Roughly speaking, our main aim consists of obtaining a combinatorial representation which condenses in an algebraic way the topology of the complex.

In order to finally achieve these goals, we now introduce a first definition of an admissible Homological Spanning tree for a general finite cell complex K .

Definition 3.6.1 *An admissible homological i -tree T_i ($i = 0, 1, \dots, n - 1$) of a finite cell complex K of dimension n is a tree satisfying the following conditions:*

- (i) *The vertices of an admissible homological i -tree T_i are either i -cells (called primary vertices of T_i) or $(i + 1)$ -cells (called secondary vertices*

of T_i).

(ii) An admissible homological i -tree T_i is a directed tree where the arrows describe paths to a single i -cell called sink.

(iii) Every leave of an admissible homological i -tree T_i is an i -cell.

The reason why these trees are called homological is the following: recalling the results presented in Section 3.5, it is clear that a discrete gradient vector field can be extracted from the directed edges of T_i (edges going from an i -cell to an $(i + 1)$ -cell). Then by Proposition 6, we have:

$$\phi(T_i) \circ \partial \circ \phi(T_i) = \phi(T_i)$$

where ∂ is the differential of K and $\phi(T_i) : \mathcal{C}_i(K) \rightarrow \mathcal{C}_{i+1}(K)$, called chain homotopy operator associated to T_i , is defined following certain rules within the “flow” of directed edges of T_i :

- $\phi(T_i)(\sigma^{(i+1)})$ is 0 if $\sigma^{(i+1)}$ is a $i + 1$ -cell belonging to a (i) -tree.
- If $\sigma^{(i)}$ is a vertex of a i -tree T_i , then $\phi(T_i)(\sigma^{(i)}) = \sum_i (a_i \text{ mod } 2) \beta_i^{(i+1)}$, where $\beta_i^{(i+1)}$ runs over all the $(i + 1)$ -cells of T_i and a_i is the number of upper integral paths from $\sigma^{(i)}$ to $\beta_i^{(i+1)}$.

An admissible homological tree is *trivial* when it is composed by a unique i -cell.

In the following we will explain how the associated chain homotopy operator of the original cell complex K can be composed by using admissible homological trees and DMT. This composition is done with the composition construction introduced in Definition 3.4.8.

In this way we achieve the first of the HSF goals that were mentioned before: that is, to obtain the chain homotopy operator algebraically connecting K with its homology groups $H(K)$.

Focussing in this first goal, the procedure to obtain this global chain homotopy operator is described here: given an initial cell complex K of dimension n with an associated differential operator ∂ , the algorithm consists of an iterative process, where at each step i , K is reduced to a smaller Morse complex K^i . Then a set of forests $\{F_0 \dots F_p\}$ formed by admissible trees is computed over the cell complex K^i . In fact this information can be described in terms of integral chain complexes $(K^i, \partial^i, \phi^i)$, where ϕ^i is directly deduced

from the admissible trees, and $\partial^i = \partial^{(i-1)} + \partial^{(i-1)}\phi^i\partial^{(i-1)}$ (see Proposition 7).

The pseudocode of the process is shown in Algorithm 2.

Algorithm 2 HSF(K, ∂)

```

 $i = 0$ 
 $\partial^0 = \partial$ 
 $\phi^0 = 0$ 
 $K^0 = K$ 
 $\mathcal{K}^0 = (K^0, \partial^0, \phi^0)$ 
while ! (Trivial ( $\mathcal{F}^i$ )) do
   $F_0 = ST\ Cell_{0,1}(K^i)$ 
   $F_1 = ST\ Cell_{1,2}(K^i \setminus F_0)$ 
  ...
   $F_p = ST\ Cell_{n-1,n}(K^i \setminus F_{p-1})$ 
   $\mathcal{F}^i = F_0 \cup F_1 \cup \dots \cup F_p$ 
   $\phi^i = CH(\mathcal{F}^i)$ 
   $\partial^i = \partial^{(i-1)} + \partial^{(i-1)}\phi^i\partial^{(i-1)}$ 
   $\mathcal{G}^i = \mathcal{G}^{i-1} \cup \mathcal{F}^i$ 
   $\mathcal{K}^{(i+1)} = \mathcal{M}(K^i, \partial^i, \phi^i)$ 
   $i = i + 1$ 
end while
return ( $\mathcal{G}^i$ )

```

Where:

- $Cell_{m,m+1}(K^i)$ are the m -cells and the $(m+1)$ -cells of the cell complex K^i .
- $Cell_{m,m+1}(K^i) \setminus F_q$ means that we considered every m -cell and $(m+1)$ -cells of the complex K^i , except the ones belonging to F_q .
- $CH(\mathcal{F}^i)$ is the sum of the chain homotopy operators of the different admissible homological trees in \mathcal{F}^i .
- $ST\ Cell_{m,m+1}(K^i)$ constructs admissible homological m -trees of the m -cells and the $(m+1)$ -cells of the complex. These trees are constructed using a basic spanning tree algorithm. They are in some way “spanning” trees, in the sense that they try to cover the maximal number of cells, but do not always reach a complete covering.
- \mathcal{K}^i is the integral chain complex created at each step.

- The function $\mathcal{M}(K^i, \partial^i, \phi^i)$ returns a Morse complex, that is its harmonic integral chain complex (see Proposition 7). This new complex is $(K^{(i+1)}, \partial^{(i+1)}, \phi^{(i+1)})$ where $K^{(i+1)}$ are the set of critical cells of K^i .

The process stops when every admissible homological tree is trivial. By using the composition defined in Definition 3.4.8, an homology integral chain complex (K, ∂, ϕ) can be constructed, where this “global” ϕ is composed by the ϕ^i of each step of the algorithm. In this way, the first of the HSF aims is reached.

In [González-Díaz 05b] the authors present an algorithm to reduce a initial chain complex to its minimal homological expression, building up an AT-model. The advantage of this method is that the obtained integral operator encodes the homological information of the initial complex (homology groups, cohomology, homology generators, relations between them, etc.). The complexity in time of this method is $O(n^3)$.

The output of Algorithm 2 encodes exactly the same information that the previous mentioned method. The advantages of this new algorithm is that by using graph techniques, it reduces the cubical complexity of the AT-model method. The heart of the proposed algorithm runs in linear time, and the question of how many times the loop should be executed, crucially depends on the particular complex. At the end, what we do here could be seen as a “union” of several cell collapses in terms of trees.

Concerning the second goal of the HSF technique, we could create a combinatorial HSF representation by considering the forest formed by the admissible homological trees of each step of the algorithm. A first definition of an HSF representation is the following:

Definition 3.6.2 *A HSF structure $\mathcal{F} = \{F_0, F_1, F_2, \dots\}$ on a finite cell complex K is a hierarchy of forests such that any of the trees of each forest F_i is an admissible homological tree T_i of the resulting cell complex K^j of the stage j of Algorithm 2.*

The main drawback here, is that these trees are not defined in the connectivity graph of the original cell complex K , and may be difficult to handle and process. For instance to see whether two HSF representations belong to the same object, may not be an easy task. In Chapter 4, this drawback is overcome: we obtain an HSF representation on the connectivity graph of the original cell complex K by restricting ourselves to $2D$. In a near future,

we intend to give a purely combinatorial definition of an HSF as a subgraph of the connectivity graph for a general cell complex. This will allow us to think an HSF structure as a true combinatorial representation codifying the topology and geometry of K .

Algorithm 2 has been implemented and tested. The implementation is written in C++, and it works either with simplicial or cell complexes. More details about this software will be given in Section 5.

Several experiments have been performed (see Figures 3.8, 3.10, 3.12, 3.14) using well known examples. The software has provided valid HSF representations and the minimum number of critical cells for each example. The corresponding HSF representations are shown in Figures 3.9, 3.11, 3.13 and 3.15.

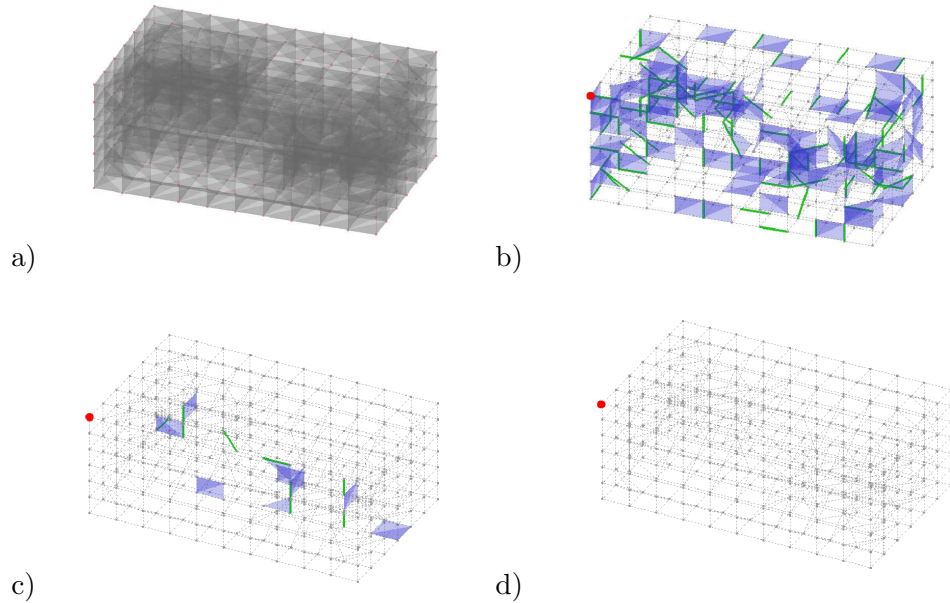


Figure 3.8: A Bing's house and its corresponding Morse complexes after some reductions. The number of critical cells in the final complex (that is one critical 0-cell, shown in Figure d)) coincides with the Betti numbers of the Bing's house $\beta_0 = 1, \beta_1 = 0, \beta_2 = 0$.

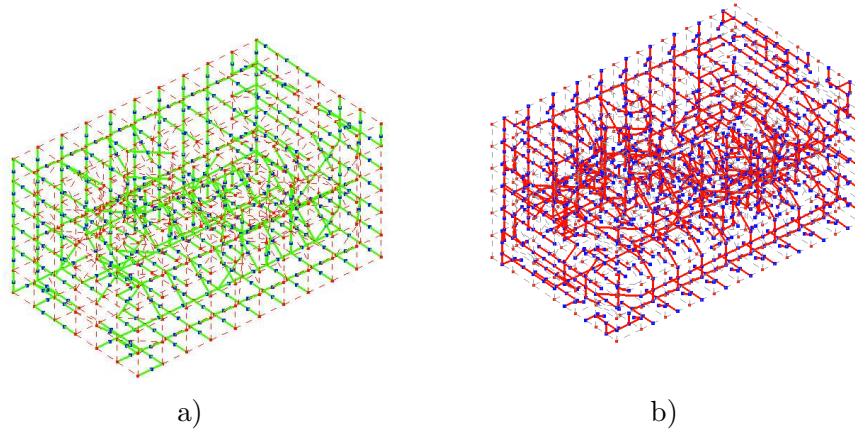


Figure 3.9: The corresponding HSF representation of the Bing's house of Figure 3.8. The HSF of dimension 0 – 1 on the left and the HSF of dimension 1 – 2 on the right.

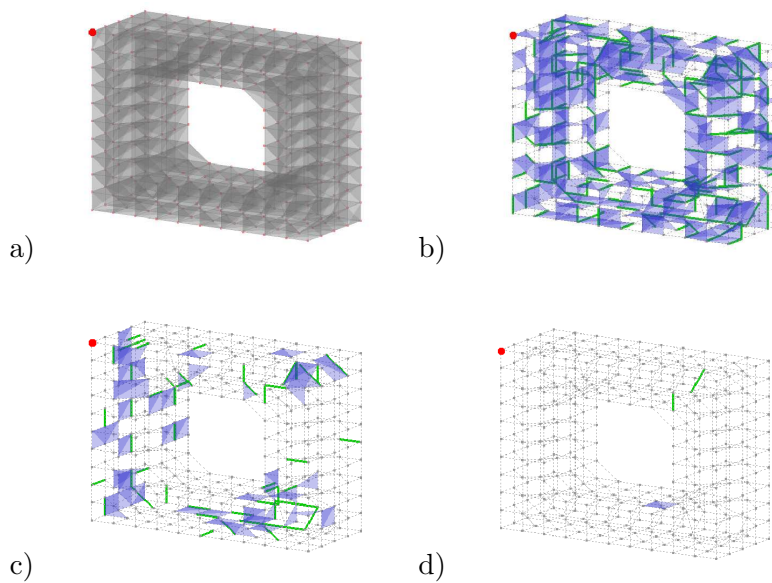


Figure 3.10: A torus and its corresponding Morse complexes after some reductions. The number of critical cells in the final complex (that is one critical 0-cell, two 1-cells and one 2-cell, shown in Figure d)) coincides with the Betti numbers of the torus $\beta_0 = 1, \beta_1 = 2, \beta_2 = 1$.

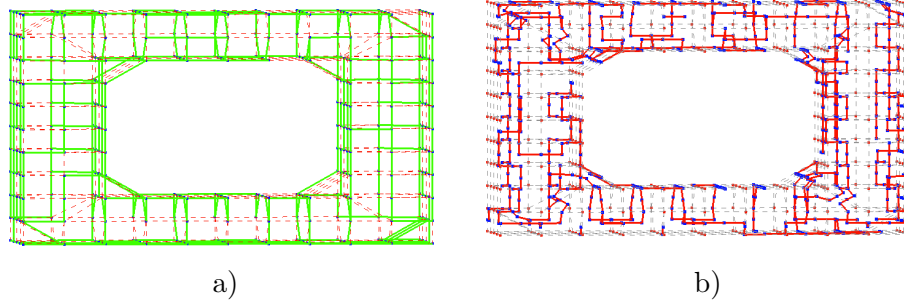


Figure 3.11: The corresponding HSF representation of the torus of Figure 3.10. The HSF of dimension 0 – 1 on the left and the HSF of dimension 1 – 2 on the right.

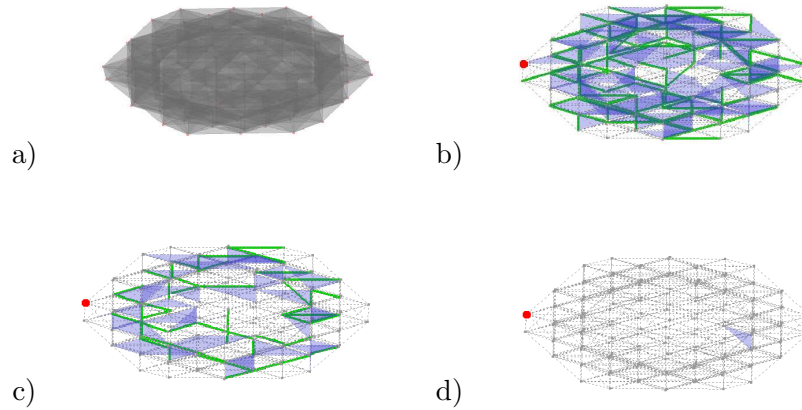


Figure 3.12: A sphere and its corresponding Morse complexes after some reductions. The number of critical cells in the final complex (that is one critical 0-cell and one critical 2-cell, shown in Figure d)) coincides with the Betti numbers of the sphere $\beta_0 = 1, \beta_1 = 0, \beta_2 = 1$.

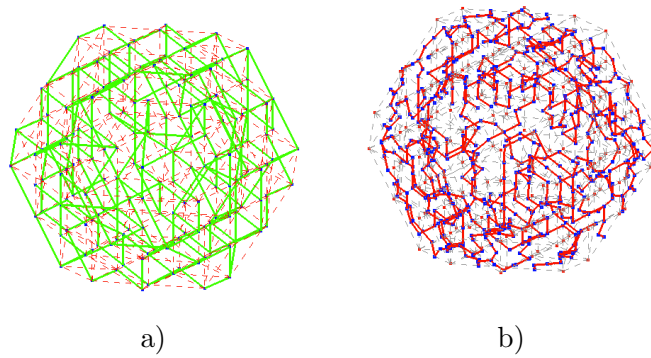


Figure 3.13: The corresponding HSF representation of the sphere of Figure 3.12. The HSF of dimension 0 – 1 on the left and the HSF of dimension 1 – 2 on the right.

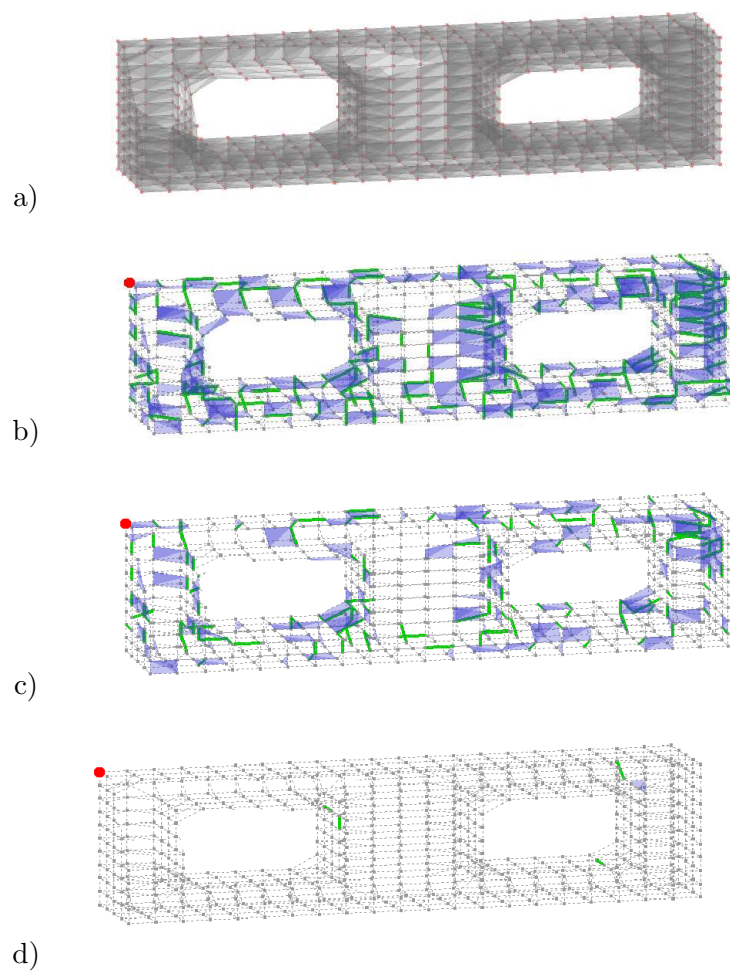
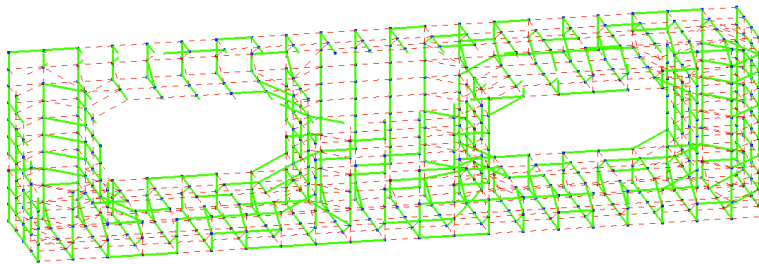
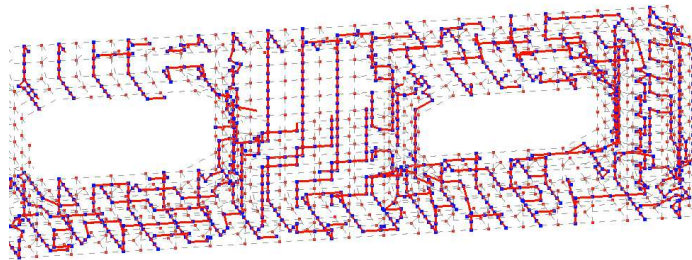


Figure 3.14: A double torus and its corresponding Morse complexes after some reductions. The number of critical cells in the final complex (that is one critical 0-cell, four 1-cells and one 2-cell, shown in Figure d)) coincides with the Betti Numbers of the double torus $\beta_0 = 1, \beta_1 = 4, \beta_2 = 1$.



a)



b)

Figure 3.15: The corresponding HSF representation of the double torus of Figure 3.14. a) The HSF of dimension 0 – 1 and b) the HSF of dimension 1 – 2.

Chapter 4

Homological Spanning Forest Framework for 2D Image Analysis

Once we have defined the HSF structure for general and higher dimensional cell complexes, we now focus in the two-dimensional case and in cell complexes embedded in a regular cartesian grid (considering this last as a cell complex). In these complexes, the construction of a purely combinatorial definition of an HSF as a subgraph of the connectivity simplicial graph can be easily done, and can be calculated for instance by means of “deformations” of a previously defined HSF structure on the ambient space. Therefore, under these conditions, the HSF structure can be seen as a true combinatorial representation codifying the topology and geometry of the complex (see [Molina-Abril 12b]).

The new definition for an HSF representation in this context is the following:

Definition 4.0.3 *Let K be a 2D finite cell complex. A HSF representation \mathcal{F} of K is a set of coordinate-based directed trees $\mathcal{F} = \{T_0^1, \dots, T_0^q, T_1^0, \dots, T_1^r\}$ (for some positive integers r and q) satisfying the following conditions:*

- (i) *Every cell of K is a vertex of only one of the trees of \mathcal{F} and it is represented by its barycentre.*
- (ii) *The vertices of a tree $T_i^j \in \mathcal{F}$ are either i -cells (called primary vertices of T_i^j) or $(i + 1)$ -cells (called secondary vertices of T_i^j). The directed*

edges of T_i^j are either upwards \mathcal{F} -arrows from an i -cell $\alpha^{(i)}$ to a $(i+1)$ -cell $\beta^{(i+1)}$ (from which $\alpha^{(i)}$ is boundary), or downwards \mathcal{F} -arrows from an $(i+1)$ -cell $\beta^{(i+1)}$ to one of its boundary i -cell $\alpha^{(i)}$. T_i^j is called an i -tree of \mathcal{F} . T_i^j can have no directed edges and, in this case, it is a trivial tree with only one vertex. Each i -tree T_i^j has at least one leaf that is a i -cell. The upwards \mathcal{F} -arrows determine a discrete gradient vector field $\mathcal{V}(\mathcal{F})$ on K .

(iii) The number of primary vertices must be greater or equal than the number of secondary vertices.

(iv) The following equalities hold:

$$\phi(\mathcal{F}) \circ \partial \circ \phi(\mathcal{F}) = \phi(\mathcal{F}) \quad \text{and} \quad \partial \circ \phi(\mathcal{F}) \circ \partial = \partial.$$

where ∂ is the differential of K and $\phi(\mathcal{F}) : C_*(K) \rightarrow C_{*+1}(K)$, called chain homotopy operator associated to \mathcal{F} , is defined by: $\phi(\mathcal{F})(\sigma^{(i)})$ is 0 if $\sigma^{(i)}$ is a i -cell belonging to a $(i-1)$ -tree of \mathcal{F} ; if $\sigma^{(i)}$ is a vertex of a i -tree T_i^j of \mathcal{F} , then $\phi(\mathcal{F})(\sigma^{(i)}) = \sum_j (a_j \bmod 2) \beta_j^{(i+1)}$, where $\beta_j^{(i+1)}$ runs over all the $(i+1)$ -cells of T_i^j and a_j is the number of upper integral $\mathcal{V}(\mathcal{F})$ -paths from $\sigma^{(i)}$ to $\beta_j^{(i+1)}$.

Based on this HSF representation, we present here a 2D topology-based digital image processing framework that allows an efficient and complete topological region-of-interest (ROI, for short) analysis (that is to process a single subregion of an image, leaving other regions unchanged). The proposed framework provides a representation that can be used for the development of efficient algorithms to compute analytical, geometrical and topological features of discrete objects.

There are in the literature an enormous number of papers dealing with techniques that reduce the structural shape of a 2D region to a graph (skeletons, shock graph, cut-graph methods, combinatorial maps, pyramids, ...) (see for example [El-Kwae 00, Lienhardt 91, Kropatsch 07]). A substantial minor number of papers handle a cell complex based representation of a 2D digital image (see for instance [Ankeney 83, Kovalevsky 06, Kovalevsky 05, Klette 00]).

Our combinatorial HSF framework for 2D digital image processing can be classified as a hybrid method that is based on the description of homolo-

gical information about cell complexes in terms of directed graphs. Extensions to higher dimensions will be carried out in the future.

4.1 HSF and ASDR cell complexes

As mentioned before, in this section we focus in the context of discrete $2D$ images. In this highly structured setting, preference will be given to an ambiance-based digital image processing rather than to an object-based one. Assuming that the cell complex of a digital image is topologically trivial (i.e., it has the homology of a point), we extend here to dimension two the homological meaning (in terms of cell collapse-like operations) of the spanning tree notion over a graph. In this way, such a combinatorial HSF scaffolding, in which the values of the pixels of a digital image I become the “weights” for the respective 0-cells, fully represent I from a topological point of view.

Referring to a digital image I of size $n \times m$, its associated HSF set of coordinated-based directed trees can be initially independent of the pixel content of the image I . A HSF representation $\mathcal{F}(I)$ leans on a common underlying continuous analogous for all the images. In $2D$, it is a finite cell complex $K_{n,m}$ that is collapsible to a cell complex version of the Euclidean plane, whose 0-cells are the pixels of I , the 1-cells describe the relationships between 4-adjacent pixels in terms of straight lines, and the closure of 2-cells are squares formed by four pixels that are mutually 8-adjacent. Although this framework works for other adjacencies between pixels (8, 6 or 4 connectedness), we focus here on 8-connectedness. In this case, $K_{n,m}$ is a cell complex with elementary “pockets” (see Figure 4.6).

Starting from an HSF representation of a digital image, there is an efficient way to “deform” it to a new one, and to isolate ROIs in order to, for example, analyze further topological features on them. Let us emphasize that these operations can be done in a parallel setting.

Let K be a finite cell complex such that a cell complex version of a finite regular cartesian grid is a strong deformation retract of K . From now on, we name such type of space as ASDR (Acyclic Strong Deformation Retract) cell complex. Let \mathcal{V} be an optimal discrete gradient vector field on K . Therefore, K is “homologically null” and all the cells of K are paired by \mathcal{V} , excepting a 0-cell s called *sink* of \mathcal{V} . In fact, s is a representative cycle of the zero

dimensional homology group of K and the chain contraction generated by $\phi_{\mathcal{V}}$ connects the chain complex canonically associated to K with $\mathbb{Z}/2\mathbb{Z}[s]$. In these conditions, K can be “decomposed” into an HSF structure. Moreover, in this case, due to the fact that the context of ASDR cell complexes is highly structured and the vertices of the HSF structure are convex cells determined by their barycentre, we can talk about HSF representation of digital objects.

In order to prove this result, we take advantage of the algebraic technique of Homological Perturbation Theory [Gugenheim 89, Gugenheim 91] applied to the discrete gradient vector field $\phi_{\mathcal{V}}$. This idea has already been exploited in a more general setting in the paper of Romero-Sergeraert [Romero 10] for establishing a strong interplay between Effective Homology and Discrete Morse Theories. In that paper, starting from an optimal gradient vector field on a general finite cell complex, a chain contraction is determined using homological perturbation. Focusing on the chain homotopy operator of this last chain homotopy equivalence, we here give a proof of its graph-based nature.

Lemma 2 [Romero 10] *Let K be an ASDR cell complex, $\mathcal{C}(K)$ be its corresponding chain complex and $\phi_{\mathcal{V}}$ be the chain homotopy operator associated to an optimal discrete gradient vector field \mathcal{V} . Let s be the sink of \mathcal{V} . Then, it is possible to construct the following chain contraction $(f_{\mathcal{V}}, g_{\mathcal{V}}, \phi_{\mathcal{V}})$ from $(\mathcal{C}(K), \partial')$ to $(\mathbb{Z}/2\mathbb{Z}[s], \mathbf{0})$, such that the differential $\partial_{\mathcal{V}} : C_*(K) \rightarrow C_{*-1}(K)$ is defined by $\partial_{\mathcal{V}}(\sigma) = v_{\sigma}$ where $(v_{\sigma}, \sigma) \in \mathcal{V}$ and the differential $\mathbf{0} : \mathbb{Z}/2\mathbb{Z}[s] \rightarrow \mathbb{Z}/2\mathbb{Z}[s]$ is defined by $\mathbf{0}(s) = 0$. The formulae for $f_{\mathcal{V}}$ and $g_{\mathcal{V}}$ are the following ones:*

$$f_{\mathcal{V}} = \mathbf{1} + \partial_{\mathcal{V}} \circ \phi_{\mathcal{V}} + \phi_{\mathcal{V}} \circ \partial_{\mathcal{V}}$$

$$g_{\mathcal{V}}(s) = s$$

Romero and Sergeraert apply the differential perturbation technique to the chain contraction $(f_{\mathcal{V}}, g_{\mathcal{V}}, \phi_{\mathcal{V}})$ from $(\mathcal{C}(K), \partial_{\mathcal{V}})$ to $\mathbb{Z}/2\mathbb{Z}[s]$, using as differential perturbation $\delta = \partial - \partial_{\mathcal{V}}$, in order to deduce a true chain contraction $(f_{\delta}, g_{\delta}, \phi_{\delta})$ connecting $\mathcal{C}(K)$ (with the original boundary operator) with its homology.

We focus our interest here in the new chain homotopy operator $\phi_{\delta} = \phi_{\mathcal{V}} + \phi_{\mathcal{V}} \circ \delta \circ \phi_{\mathcal{V}} + \phi_{\mathcal{V}} \circ \delta \circ \phi_{\mathcal{V}} \circ \delta \circ \phi_{\mathcal{V}} + \dots$ to derive an HSF graph structure from it.

For a q -cell a ($q = 1, 2$), $\delta(a) = u_1 + \dots + u_t$, such that each u_i is not paired with a by means of \mathcal{V} .

For a q -cell a_0 , the value $\phi_\delta(a_0)$ is a sum of $(r_1 + r_2 + \dots + r_t)$ $(q+1)$ -cells $\phi_\mathcal{V}(a_0) = b_{0,1}$, $\phi_\mathcal{V} \circ \delta \circ \phi_\mathcal{V}(a_0) = b_{1,2} + \dots + b_{r_2,2}$, \dots , $(\phi_\mathcal{V} \circ \delta)^{t-1} \circ \phi_\mathcal{V}(a_0) = b_{1,t} + \dots + b_{r_t,t}$. In fact, $\delta \circ \phi_\mathcal{V}(a_0) = a_{1,2} + \dots + a_{r_2,2}$, with $\phi_\mathcal{V}(a_{1,2}) = b_{1,2}$, \dots , $\phi_\mathcal{V}(a_{r_2,2}) = b_{r_2,2}$. Analogously, $(\delta \circ \phi_\mathcal{V})^{t-1}(a_0) = a_{1,t} + \dots + a_{r_t,t}$, with $\phi_\mathcal{V}(a_{1,t}) = b_{1,t}$, \dots , $\phi_\mathcal{V}(a_{r_t,t}) = b_{r_t,t}$. On the other hand, $\phi_\delta(C(K))$ is an acyclic graded vector space. It is combinatorial in the sense that it admits a basis formed by cells of K . If $\phi_\mathcal{V}(a) = b$, then b also belongs to $\phi_\delta(C(K))$. Let us note that $\phi_\delta(a) = \phi_\mathcal{V}(a) + \phi_\delta(\delta(b))$ and therefore $b = \phi_\delta(a) - \phi_\delta(\delta(b))$.

With all these results at hand, $\phi_\delta(a_0) = b_{0,1} + b_{1,2} + \dots + b_{r_2,2} + \dots + b_{1,t} + \dots + b_{r_t,t}$ can also be expressed as a directed tree $T_{\mathcal{V},a_0}$, having as vertices $V(T_{\mathcal{V},a_0})$ all the p and $(p+1)$ -cells

$$\{a_0 = a_{0,1}, b_{0,1}, a_{1,2}, \dots, a_{r_2,2}, b_{1,2}, \dots, b_{r_2,2}, \dots, a_{1,t}, \dots, a_{r_t,t}, b_{1,t}, \dots, b_{r_t,t}\},$$

previously described. The set of edges $E(T_{\mathcal{V},a_0})$ is formed by arrows from $a_{i,j}$ to $b_{i,j}$, $\forall i, j$, and from a $(q+1)$ -cell $b_{i,j}$ with a q -cell $a_{k,\ell}$ belonging to its boundary. Moreover, any path starting from a_0 and finishing in a $(p+1)$ -cell is an upper integral path for \mathcal{V} . In fact, $\{a_0\} \cup \phi_\delta(a_0)$ is homotopy equivalent to $T_{\mathcal{V},a_0}$ (see Figure 4.1).

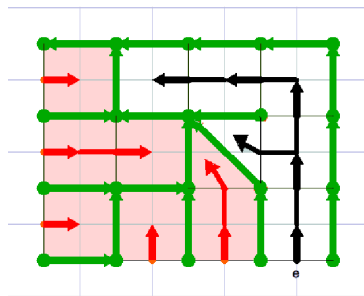


Figure 4.1: HSF representation in which the sum of 2-cells $\phi(e)$ is coloured in white. Its associated homological tree $T_{\mathcal{V},e}$ is coloured in black.

It can occur that there are two cells a, a' , with $a \neq a'$, for which $V(T_{\mathcal{V},a}) \cap V(T_{\mathcal{V},a'}) \neq \emptyset$. In this case, the union of the corresponding associated trees is again a new tree containing the previous ones. Finally, from this data it is immediate to establish an HSF structure accomplishing all the conditions of Definition 4.0.3.

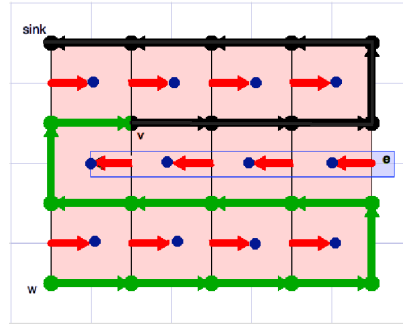


Figure 4.2: An optimal gradient vector field for a cubical complex and the upper integral path starting from the edge e . The homological tree $T_{V,v}$, or in this case $\phi(v)$, is coloured in black.

In Figure 4.2, an optimal gradient vector field for a cubical complex K and the upper integral path starting from the edge e are shown. In fact, this path can also be seen as sums of cells $\phi_\delta(e)$. The 2-cells forming the sum $\phi_\delta(e)$ are represented by blue thick points. Then, it is possible to construct an HSF structure on the ASDR cell complex K from $\phi_\delta(C(K))$.

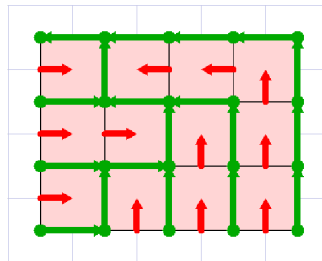


Figure 4.3: An optimal gradient vector field for a cubical complex.

In Figure 4.3, an optimal discrete gradient vector field is described for the acyclic cubical complex K . In Figure 4.4, an HSF representation of K is shown.

Theorem 2 *Let K be a finite 2-dimensional ASDR cell complex and suppose that there is an optimal discrete gradient vector field \mathcal{V} on it. Then, there is an HSF structure $\mathcal{F}^\mathcal{V}$ uniquely associated to \mathcal{V} . Reciprocally, an HSF graph structure on K produces in a natural way an optimal discrete gradient vector field. Moreover, one of the discrete gradient vector fields associated to an HSF structure coming from an initial optimal discrete gradient vector field \mathcal{V} is \mathcal{V} itself.*

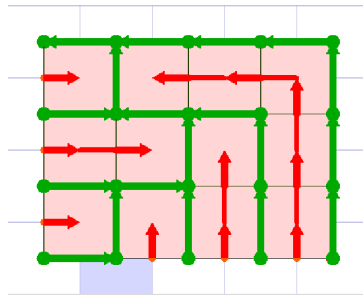


Figure 4.4: HSF representation where the 0-dimensional tree is coloured in green and the 1-dimensional trees in red.

The first part of the previous theorem has already been proved. Now, an HSF configuration on a general cell complex generates discrete gradient vector fields, which can be not optimal. The acyclicity of an ASDR cell complex allow to guarantee the optimality. The arrows from a i -cell to an $(i + 1)$ -cell ($i = 0, 1$) of the HSF forest produce the corresponding vectors in the gradient vector field. This strong relationship between these two important notions for ASDR cell complexes supports the existence of a homology-based digital image processing framework in this setting.

In the next section, we also confirm a good behaviour of the HSF configurations under local transformations within an ASDR cell complex. We also highlight the power of this representation for advanced topological sequential or parallel computation.

4.2 HSF and 2D digital image processing.

Throughout this section, we describe the functional architecture of our 2D digital image processing framework. This schema has four levels: Device, Logical or Cellular, Conceptual and Continuous Level. In the Device Level we represent the objects in a computer screen, that is, as digital images. In this Chapter, this representation is exclusively restricted to that of a digital image based on square pixels. The carrier of all the digital images is defined on the regular cartesian grid. The pixels of the digital image are the vertices of the grid. We mainly use the raster representation of images at Device Level. The Logical or Cellular Level models the connectivity relationships among pixels using a cell complex structure.

The only restriction that needs to be imposed is that the ambiance cell

complex must be an ASDR complex. We have demonstrated in the previous Section that for this type of subdivided spaces, optimal gradient vector fields can be identified with HSF representations (see Theorem 2). The Conceptual Level deals with the homological information of the previous cell complex codified in terms of coordinated-based direct acyclic graphs or “homological spanning trees”. Finally, the Continuous Level is used to find a continuous solution. We follow the general organization of the digital framework of [Ayala 96], but integrating new proposals for the logical and conceptual levels.

The models for the Device and Continuous Levels are well known and do not need more explanation. We focus our interest on the Cellular and Conceptual Levels of the framework. It is in these levels of digital content where the HSF graph-based structure becomes a true geo-topological (geometric and topological) representation of digital objects. The representation of the vertices of the HSF in terms of the coordinates (in \mathbb{R}^2) of the barycentre of the cells, allows us to fully reconstruct the cell complex from the HSF structure.

4.2.1 Cellular Level.

In order to propose a general topological framework for digital images in which most of their properties or features correspond to topological properties in \mathbb{R}^n , two main different types of methods have been developed in the literature: those based on the adjacency graph (see for instance [Rosenfeld 70, Chassery 79, Klette 04, Parker 97]) and those based on 2-dimensional cell complexes (see for example [Khalimsky 86, Kong 91, Webster 01, Kovalevsky 06]). The method proposed here is a hybrid model in which the image is processed using and modifying a set of directed trees in the whole image. Any ASDR complex could be valid, and we can choose the most suitable one, depending on the application and the processing we want to execute.

First, we model in a semi-continuous way a topology of the Euclidean plane. To do this, and supposing that the 2D image is defined on a discrete set $D \subset \mathbb{R}^2$, the usual idea is that D is identified with the set of 2-cells of the complex and the lower-dimensional cells have to be generated additionally. In [Kovalevsky 89, Kovalevsky 08], $D = \mathbb{Z}^2$ (standard case) is identified with the set of 2-cells of a uniform planar square cell complex \mathcal{K} , called

Kovalevsky's cell complex.

We present here an extension of this technique, taking into account that the discrete carrier $D \subset \mathbb{R}^2$ is also the uniform square planar grid \mathbb{Z}^2 . The Cellular Level consists in principle of a cell complex \mathcal{L} simple homotopically equivalent to \mathbb{R}^2 . Its construction starts with a cell complex $\tilde{\mathcal{K}}$ (equivalent to the Kovalevsky's cell complex \mathcal{K}) such that the points of D are the 0-cells of $\tilde{\mathcal{K}}$, its 1-cells are the segments connecting 4-adjacent points and its 2-cells are the squares having as corners to four mutually 8-adjacent points (see Figure 4.5).

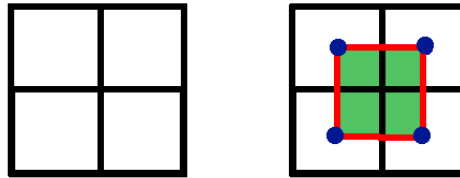


Figure 4.5: A 2×2 digital image based on square pixel and its corresponding initial ASDR complex. 0-cells are coloured in blue, 1-cells in red and 2-cell (square) in green.

After that, triangular 2-cells connecting three mutually 8-adjacent points are added to $\tilde{\mathcal{K}}$. The intersection of each 2-cell c_t of this type with $\tilde{\mathcal{K}}$ is formed by two perpendicular 1-cells having in common a point of D . The third 1-cell belonging to the boundary of c_t is that joining two 8-adjacent diagonal points in D and it is a free edge (it belongs to only one 2-cell, that is, c_t). In this way, the triangular 2-cells add elementary “pockets” to $\tilde{\mathcal{K}}$ and specify the morphology of \mathcal{L} . The possible cell configuration for an object of interest in the subset $N_8 \subset D$ of any four points of D mutually 8-adjacent are (up to isometry) shown in Figure 4.6. The underlying idea behind the integration of a planar structure with elementary “pocket” defects at the cellular level is twofold: to automatically obtain optimal gradient vector fields for the ambiance space and to express all the previous point configurations suitably in terms of vectors at conceptual level.

Summing up, the Logical level has been determined in terms of a cell subcomplex of \mathcal{L} . In this schema, some geometric information (for example, diagonal edges) appears as elementary cellular perturbation of the topology of the Euclidean plane. This is the way in which Geometry is integrated in

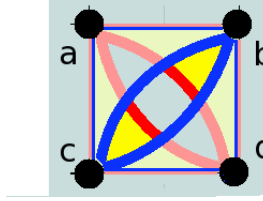


Figure 4.6: Cell complex showing the four “pockets” corresponding to four mutually 8-adjacent points. The edges forming the triangular cells are: $((a, d), (d, c), (c, a))$, $((a, b), (b, d), (d, a))$ (in red) and $((a, b), (b, c), (c, a))$, $((c, b), (b, d), (d, c))$ (in blue)

this topological schema. Notice that in practice, the cell complex \mathcal{L} , having four triangular micro-pockets for each square 2-cell in the plane, is the maximal (in terms of cells) ambiance cell complex in our framework. The next conceptual level for a digital image determine the concrete ambiance space, with each triangular pocket at cellular level specified by a corresponding diagonal arrow at conceptual level.

4.2.2 Conceptual Level: tree-based homology information.

The Conceptual Level briefly consists of installing homology information on terms of an HSF representation on an ASDR cell subcomplex \mathcal{L}' of \mathcal{L} and handling this information in a combinatorial way. More precisely, we manage and modify at a cell level, the vector space produced by the image of a chain homotopy operator describing the acyclicity of the cell complex \mathcal{L}' . In other words, we apply graph-based techniques for transforming an HSF representation of an image into another via local operations at conceptual level.

In order to develop a consistent and reusable framework for homology-based 2D digital image processing, we take advantage of an ambiance-based image processing and the one-to-one mapping between HSF representations and optimal discrete gradient vector fields in ASDR cell complexes.

We subdivide this section into two parts: (a) Possible HSF models for the square subdivided topological plane \mathcal{L} with triangular micro-pockets; and (b) Local interchanging operations in the HSF model of a digital image $I : D \rightarrow V$. From now on, taking into account that we work here with ASDR cell sub-complexes of \mathcal{L} , we identify optimal discrete gradient vector fields and its corresponding combinatorial chain homotopy operators (see

Theorem 2) with HSF representations. These notions are suitably mixed in the following statements and results.

Homological initial state for the ambiance space.

Let us install on the ASDR cell complex $\tilde{\mathcal{K}}$ defined in Section 4.1, an initial optimal gradient vector field \mathcal{V} . It is an easy task to create such vector field. In fact, it describes the acyclicity of the cell complex and can be identified with an HSF representation \mathcal{F} of $\tilde{\mathcal{K}}$ (see Theorem 2). The construction of the HSF structure from the optimal gradient vector field has been detailed in the previous section.

Local operations involving combinatorial chain homotopies.

Using an ambiance-based approach, problems related to the “measurement” of topological phenomena (like holes or tunnels of a 3D digital image), can be solved in a satisfactory way. On the other hand, topological invariants (in particular, homology) are global characteristics of the object, and a consistent framework for topology-based image processing must give a quick and correct answer for extracting this topological information when an elementary local “deformation” is applied.

We demonstrate here that some elementary local changes on the corresponding HSF representation, have an automatic translation to the global setting. These local changes are seen in terms of chain homotopy operators (or, equivalently, in terms of discrete vector fields) involving a reduced subset S of neighbour cells. In fact, the only constraint for S is that it must be closed by the concrete discrete vector field installed on \mathcal{L} . We put the emphasis here on three types of HSF operations: (a) Arrow reversing (b) edge rotation and (c) face rotation.

Algorithm 1 (Arrow Reversing) *Let $\mathcal{F} = \{T_0^1, \dots, T_0^q, T_1^1, \dots, T_1^r\}$ be an HSF representation of an ASDR cell subcomplex of \mathcal{L} . Let c_0, c'_0 be two 0-cells, such that c'_0 is the sink vertex in \mathcal{F} and there is a directed path p of 1-cells $c_0 : e_1, e_2, \dots, e_n : c'_0$ in \mathcal{F} from c_0 to c'_0 . Then, we can construct a new HSF representation \mathcal{F}' that is identical to \mathcal{F} , except for the 0-cells belonging to the path p . In fact, the new pairs in the resulting HSF representation are $\{c'_0, e_n\}$ and those pairs from 0-cells to 1-cells in the directed path $c'_0 : e_n, e_{n-1}, \dots, e_1 : c_0$. In \mathcal{F}' , c_0 is the new sink vertex.*

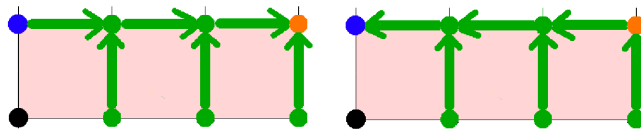


Figure 4.7: Arrow reversing example where the blue point represents c_0 and the orange one represents c'_0

In Figure 4.7, an arrow-reversing operation is shown (see the path drawn in green). Arrow-reversing operations for 1-cells are not allowed due to the fact that each 1-cell that does not belong to a 0-tree of the HSF \mathcal{F} is paired with a 2-cell.

Algorithm 2 (Edge Rotation) *Let c_0 be a 0-cell, c_1 and c'_1 be two 1-cells and c_2 a 2-cell of an ASDR cell subcomplex \mathcal{L}' of \mathcal{L} . Let $\mathcal{F} = \{T_0^0, \dots, T_0^q, T_1^0, \dots, T_1^r\}$ be an HSF representation of \mathcal{L}' with associated combinatorial chain homotopy operator ϕ . Working at Conceptual Level, if $\{c_0, c_1\}$ is an arrow in a 0-tree T_0^j , $\{c'_1, c_2\}$ is an arrow in the 1-tree T_1^k and c_0, c_1 and c'_1 belongs to the boundary of c_2 , then we can generate an HSF representation $\tilde{\mathcal{F}}$ with associated combinatorial chain homotopy operator $\tilde{\phi}$ defined by $\tilde{\phi}(c) = \phi(c)$ for any cell c different from c_0 and c_1 , $\tilde{\phi}(c_0) = c'_1$ and $\tilde{\phi}(c_1) = c_2$.*

In the Edge Rotation transformation, the underlying cellular structure can be modified. We can not guarantee that the final HSF representation belongs to the original cell complex \mathcal{L}' . In Figure 4.8, we show some elementary examples of edge rotations.

Algorithm 3 (Face Rotation) *Let c_2^1 and c_2^2 two (both square or both triangular) 2-cells of an ASDR cell subcomplex \mathcal{L}' of \mathcal{L} , sharing a common edge $e_1^1 = e_1^2$. Let \mathcal{F} be an HSF representation of \mathcal{L}' and ϕ an associated combinatorial chain homotopy operator. Working at Conceptual Level, if all the cells of the subcomplex $C(c_2^1, c_2^2)$ generated by the closures of c_2^1 and c_2^2 can be grouped by pairs of the gradient vector field \mathcal{F} , then we can generate a new acyclic combinatorial chain homotopy operator $\tilde{\phi}$. Using the labelling indicated in Figure 4.9, its HSF representation $\tilde{\mathcal{F}}$ agrees with \mathcal{F} excepting for the pairs of $C(c_2^1, c_2^2)$. For such pairs and for $i, j \in \{1, 2\}$ with $i \neq j$, $\{v_k^i, e_\ell^i\}$ belongs to a 0-tree of $\tilde{\mathcal{F}}$ if $\{v_k^j, e_\ell^j\}$ belongs to a 0-tree of \mathcal{F} and $\{e_k^i, c_2^i\}$ belongs to a 1-tree of $\tilde{\mathcal{F}}$ if $\{e_k^j, c_2^j\}$ belongs to a 1-tree of \mathcal{F} (see Figure 4.9).*

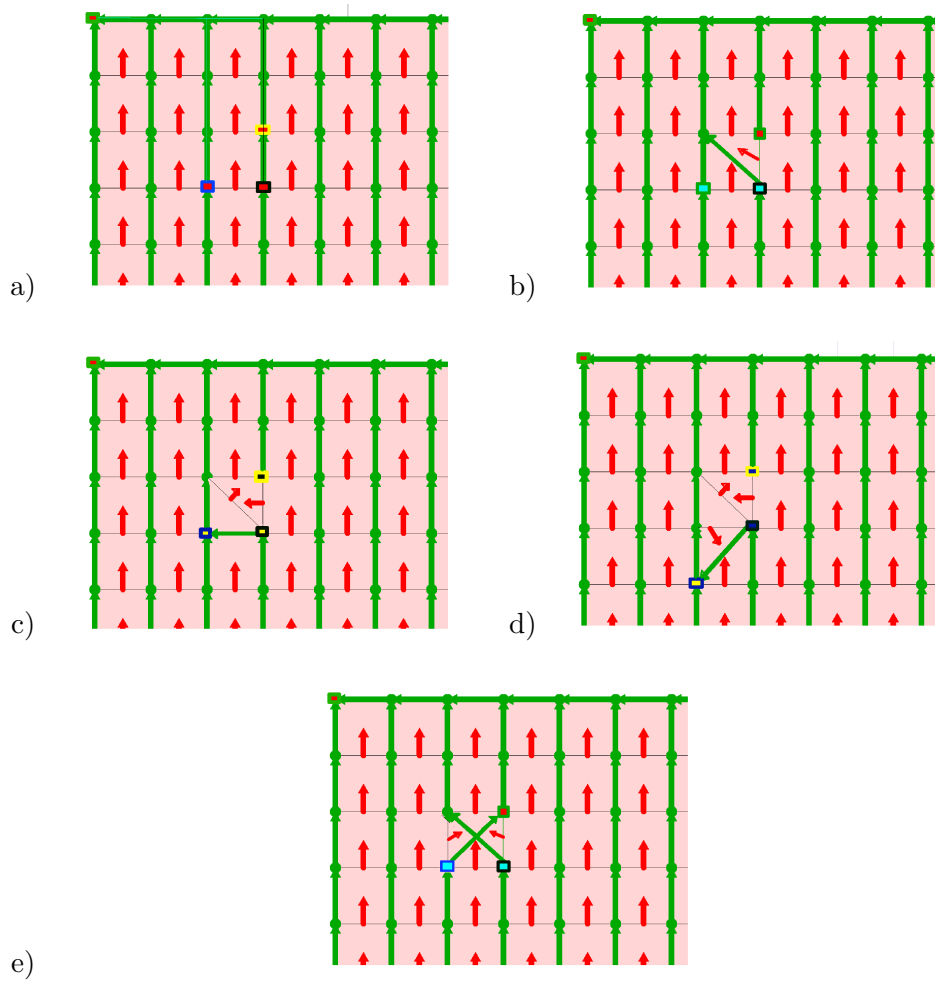


Figure 4.8: Edge rotation examples

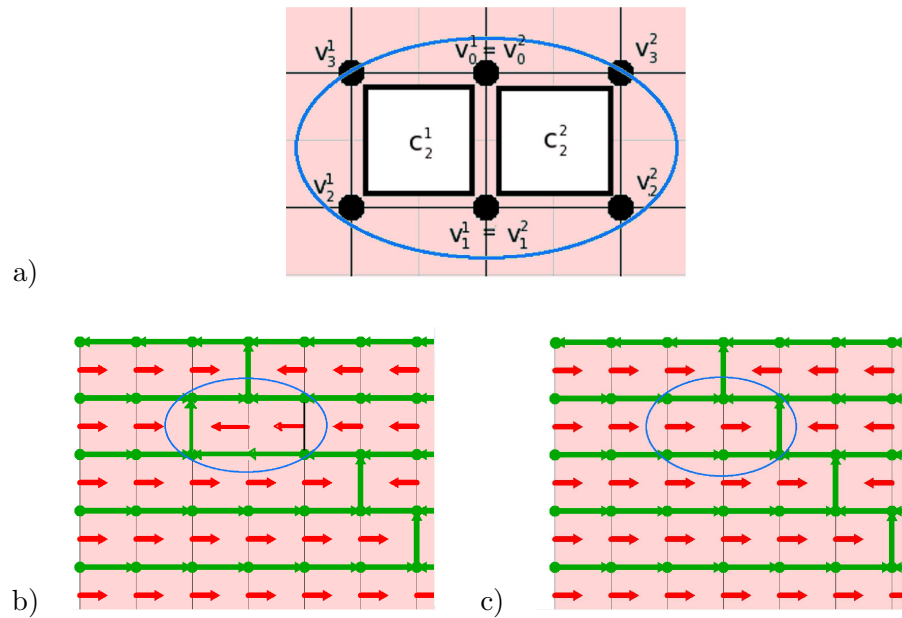


Figure 4.9: Face rotation example. a) Zoom of the involved cells, b) initial configuration, c) final configuration

Let $I : D \rightarrow R$ be a digital image with an 8-connected object of interest $O \in D$. Then, it is possible to modify an initial HSF representation (and its corresponding combinatorial chain homotopy operator) by means of the previous local operations, in such a way that the result is a new HSF representation, called HSF representation of I based on O , in which a finite set of edges (called *bridge edges*) link the object of interest with the background.

This homology-based transformation is processed in two steps:

- (i) For each pixel in the inner boundary of O (that is, belonging to O and having a pixel not in O as 8-neighbour) we select, if possible, an arrow connecting this pixel with other neighbour pixel of the inner boundary of O (by using an elementary edge rotation).
- (ii) For each pixel in the outer boundary of O (that is, belonging to the complementary of O and having a pixel in O as 8-neighbour) we select, if possible, an arrow connecting this pixel with other neighbour pixel of the outer boundary of O (by using an elementary edge rotation).

At the end of this process, we have almost “isolated” (along its crack) the object O from the background, excepting the existence of a set of edges in the new HSF representation, called bridge edges, that connect O with its complementary.

In Figure 4.10, some HSF representations based on an object of interest are shown.

4.3 Homology and Cohomology of objects of interest.

Let $I : D \rightarrow R$ be a 2D digital image and $O \in D$ be a (non-necessarily 8-connected) object of interest. Let $\mathcal{F} = \{T_0^1 \dots, T_0^q, T_1^1 \dots T_1^r\}$ be an HSF representation based on O and $\phi : \mathcal{L} \rightarrow \mathbb{Z}/2\mathbb{Z}[\mathcal{L}']$ be its corresponding acyclic combinatorial chain homotopy operator over an ASDR cell subcomplex \mathcal{L}' of \mathcal{L} . It is possible to deduce homology groups and generators of the cell subcomplex $\mathcal{L}(O) \subset \mathcal{L}$ generated by the pixels of O (it is called the homology of O with coefficients in $\mathbb{Z}/2\mathbb{Z}$) from the HSF representation \mathcal{F} . In other words, the 8-connected components and holes of the object of interest can be easily determined from this HSF representation. In other words, it is possible to deduce an HSF representation for a digital object from one of

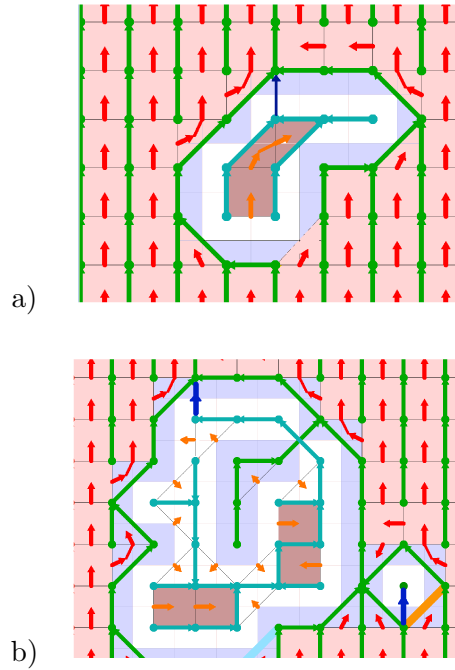


Figure 4.10: Examples of HSF representation based on objects of interest

the ambiance space, by doing some minor local modifications (see Figure 4.11).

The idea is to consider the sub-forest \mathcal{F}' of the forest \mathcal{F} , corresponding to the subcomplex $\mathcal{L}(O)$ (that is, the HSF representation of the object O).

Let us denote $\mathcal{F}' = \{T_0^1 \dots, T_0^q, T_1^1 \dots T_1^r\}$. Let us suppose that the sink s is not in O . Then, the following results hold:

- (i) The (unpaired) vertices in O (also called critical vertices) are representative cycle generators of the corresponding connected components of O . Therefore c is the number of connected components in O .
- (ii) Any unpaired 1-cell (u, v) (also called critical edge) gives rise to a 1-homology generator fitting geometrically with the outer or some inner boundary of O . In fact, this homology cycle is obtained by the formula $\{u, v\} + \phi(u) + \phi(v)$. If the complementary of O has h connected components (including background), the number of critical edges is $h - 1$. Associated to each critical edge, there is a 1-tree of \mathcal{F}' . The number of 1-cells corresponding to critical edges of a 1-tree of \mathcal{F}' can be greater than one.

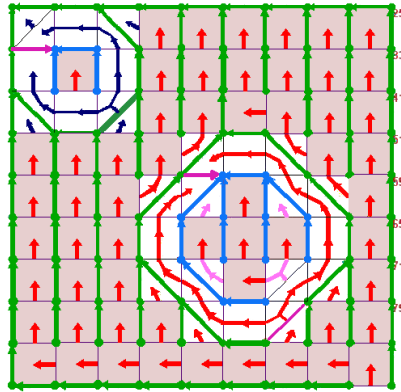


Figure 4.11: Example of HSF representation based on two ROIs (set of black pixels in the image). The 0–tree is represented by a spanning tree and the 1–tree of the HSF structure is determined by a vector field.

Each critical edge can be “moved” along its associated tree T_1^j , in order to get different representative cycles of the corresponding 1–homology generator. This translation of the critical edge can be done using reversing-arrow operations. It is clear that an HSF representation based on an object of interest O is also suitable for obtaining homology information about the complementary of O .

In Figure 4.12 two different HSF representations of an object of interest are shown. The black edge represents the 1–homology generator of the hole in the object. The homology cycle is coloured in blue.

Moreover, we can easily deal with cohomology information starting from an HSF representation of an object. For example, the trees T_1^j associated to a critical edge (u, v) determines in a straightforward manner a representative cocycle c_T of a cohomology generator of dimension 1. This cochain $c_T : \{1 - \text{cells} \in \mathcal{L}(O)\} \rightarrow \mathbb{Z}/2\mathbb{Z}$ is not null only for the vertices (that is, the 1–cells of $\mathcal{L}(O)$) belonging to the tree T_1^j . In Figure 4.13 we show how cohomology can be equivalent to a type of paths in the HSF representation such that if we “cut” the object through this path, the resulting object has one hole less than before.

By using the HSF framework, we can produce algorithmic answers to certain problems related to the homological classification of cycles (i.e., a sum of cells having zero boundary). The main important ones in the area of discrete image processing are the shortest cycle, contractibility and trans-

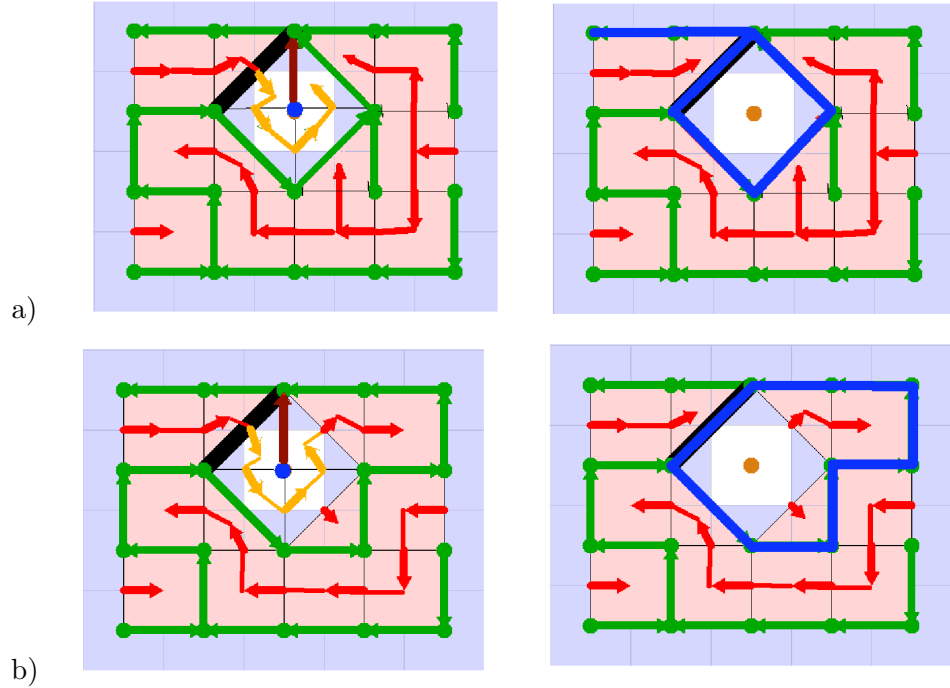


Figure 4.12: Two different examples of HSF representations and the resulting homology generator (in blue) computed for the same object of interest.

formability problems. The shortest cycle problem is a generalization of the well-known shortest path problem [Dijkstra 59] and can be stated as follows: Given a cycle c on a cell complex version K of a ROI, what is the shortest cycle on K homologically equivalent to it? The contractibility problem consists of checking whether a cycle can be contracted to a point and the transformability problem analyses whether two cycles can be transformed into each other. These problems have significant connections with another in computational topology: to determine the fundamental group of K or, equivalently, to construct a polygonal schema (cut a closed genus g surface to a canonical polygon with $4g$ edges). The work of Gouillard [Gouillard 05] gives a very complete account of the state of the art about these questions, treating them under a homotopical perspective.

We limit ourselves to demonstrate that contractibility and transformability problems can be automatically solved using the chain contraction (f_K, g_K, ϕ_K) canonically associated to an HSF structure F_K of a ROI K .

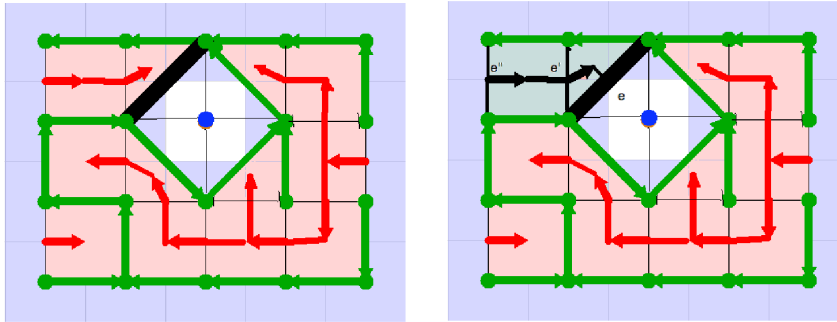


Figure 4.13: A HSF representation (on the left) and the path (on the right picture, in black) representing cohomology.

Given a contractible q -cycle c on K , then we have:

$$c + g_K f_K(c) = \partial_K \phi_K(c) + \phi_K \partial_K(c)$$

Since $\partial_K(c) = 0$ (c is a cycle) and $f_K(c) = 0$ (because c is contractible and its associated homology generator is zero), we reduce the previous equality into the following one:

$$c = \partial_K \phi_K(c)$$

That means, that $\phi_K(c)$ is the $(q + 1)$ -chain whose boundary is c (see Figure 4.14).

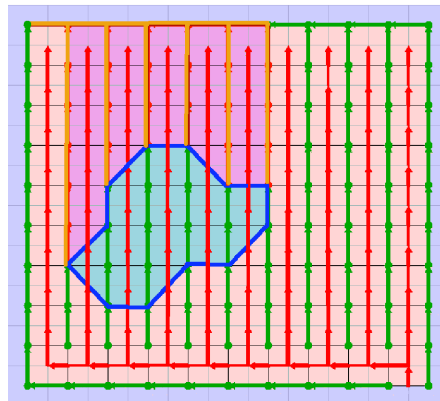


Figure 4.14: An acyclic scenario, one cycle c coloured in dark blue, and the corresponding sum of 2-cells $\phi_K(c)$ coloured in light blue

Given now two q -cycles homologically equivalent c and c' , then:

$$c + g_K f_K(c) = \partial_K \phi_K(c)$$

$$c' + g_K f_K(c') = \partial_K \phi_K(c')$$

If we subtract one equation from another, we have:

$$c + c' = \partial_K \phi_K(c - c')$$

due to the fact that $g_K f_K(c) = g_K f_K(c')$. That means that $\phi_K(c + c')$ is the $(q + 1)$ -chain whose boundary is the difference between the original cycles.

4.4 Parallel homology-based processing

Within this context of homology-based processing, it is possible to exploit data parallelism. Although the methods designed here open the possibility to a parallel processing, we are still far from devising a realistic and practical parallel approach. We limit ourselves to give some hints about our intentions with respect to the parallelization of the proposed framework.

Being n the number of pixels or 0-cells in \mathcal{L} , the idea is to decompose the cell complex \mathcal{L} into n subsets of cells, each of them included into the neighbourhood of a concrete vertex.

We consider as many Processing Elements (PE) as pixels the image has. A PE(v) in this architecture consists of a subset E of cells having a pixel v with integer coordinates (x, y) as element of its boundary. More concretely: $E = \{\{(x, y)\}, \{(x, y), (x - 1, y + 1)\}^+, \{(x, y), (x - 1, y + 1)\}^-, \{(x, y), (x +$

$1, y)\}, \{(x, y), (x - 1, y), (x - 1, y + 1)\}, \{(x, y), (x - 1, y + 1), (x, y + 1)\}, \{(x, y), (x, y + 1)\}, \{(x, y), (x + 1, y + 1)\}^+, \{(x, y), (x + 1, y + 1)\}^-, \{(x, y), (x, y + 1), (x + 1, y + 1)\}, \{(x, y), (x + 1, y), (x + 1, y + 1)\}, \{(x, y), (x + 1, y), (x + 1, y + 1)\}, \{(x, y), (x + 1, y), (x + 1, y + 1)\}, \{(x, y), (x + 1, y), (x + 1, y + 1)\}\}$ where $(x, y)^\pm$ indicates the respective upper (+) and

lower (-) diagonal edges.

In order to clarify this idea, a PE is shown on the left image of Figure 4.15. The pixel P is represented in green, the six 1-cells in blue, the four triangular-like 2-cells in yellow, and the one square 2-cell in orange.

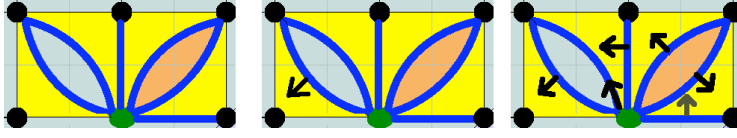


Figure 4.15: A PE on the left and a pairing example on the right

As it is shown in the middle image of Figure 4.15, we consider that the cells $\{(x, y), (x - 1, y + 1)\}^-$ and $\{(x, y), (x - 1, y), (x - 1, y + 1)\}$ are always paired.

Given a pixel which is not the sink, the task consists of pairing the vertex P with an edge of E , being paired the rest of cells in E in a straightforward manner. For example, if we pair P with the edge $\{(x, y), (x - 1, y + 1)\}^+$, then the pairing of the rest of cells is (see right image of Figure 4.15):

$$\begin{aligned} & \{(\{(x, y), (x, y + 1)\}, \{(x, y), (x - 1, y + 1), (x, y + 1)\}), \\ & (\{(x, y), (x + 1, y + 1)\}^+, \{(x, y), (x, y + 1), (x + 1, y + 1)\}), \\ & (\{(x, y), (x + 1, y + 1)\}^-, \{(x, y), (x + 1, y), (x + 1, y + 1)\}), \\ & (\{(x, y), (x + 1, y)\}, \{(x, y), (x + 1, y), (x + 1, y + 1), (x, y + 1)\}) \} \end{aligned}$$

This procedure is specified in the following algorithm:

Algorithm 4 (Parallel optimal discrete gradient vector field) $PE(v)$
is a unit processing element “centred” at pixel v .

for each $PE(v)$ parallel do
 Choose one edge e having as source the vertex v .
 Establish the pair $\{v, e\}$.
 Pair the rest of incident edges and 2-cells in $PE(v)$.
end

Taking n processors (as many as pixels the image has), the speedup S_n , measuring how much the parallel algorithm of establishing a random acyclic combinatorial chain homotopy operator ϕ is faster than the corresponding sequential algorithm, is ideal. That means that $S_n = \frac{n}{1} = n$. We suppose here that the sink of ϕ is always a priori known pixel.

Chapter 5

Implementation and Software

We present here the computational application that has been developed in order to test and experiment with the Homological Spanning Forest representation. Our purpose is to have at hand algorithms and software allowing to perform experiments and to obtain explicit results that are useful in the context of digital imagery.

This software implements the results and algorithms introduced in Section 3, and although some of the results of Section 4 have also been implemented, the complete development of the $2D$ HSF Framework is planned to be included in the future.

5.1 Homology computation programs

Some of the computer programs implementing homological and cohomological procedures that have been lately developed are the following:

- **Linbox**: a C++ library with GAP and Maple interfaces (see [Lin]).
- **HAP**: Homological Algebra Programming, a GAP package (see [HAP]).
- **Kenzo**: a Lisp program for computing homology, cohomology, and homotopy groups. It implements several spectral sequences, can build the first stages of the Whitehead and Postnikov towers, and has a particular emphasis on iterated loop spaces (see [Dousson 99]).

- **CHomP**: the Computational Homology Project, has a set of tools for computing the homology of a collection of n -dimensional cubes, with a view towards applied applications in dynamical systems, chaos theory, and pattern characterization (see [Cho]).
- **Plex**: A package developed as a research tool for building and studying persistent homology of simplicial complexes, generated from real or synthetic point-cloud data (see [Ple]).
- **Voxelo**: Voxelo is a 3D modeller in which we can simulate different algebraic topological processes (computation of homology groups, cup products, etc) in voxel-based digital images.

Some of these tools work with symbolic computations. Some others dealing with images and numerical computation have restrictions like computing exclusively homology groups and Betti numbers, or working with a concrete type of complexes.

In addition to that, as we work with images, we consider essential to have at our disposal a tool that allows a complete visualization of our computations and experimental results. Within the previously mentioned programs, the only one that provides a complete visualization interface is Voxelo. Besides that, Voxelo presents several drawbacks with regards to our purpose. Some of them are that it is educational-oriented, it is not multiplatform (works only under Windows), it deals exclusively with simplicial complexes and the 14-adjacency between voxels, it is not easy to extend due to its lack of modularity, etc. Therefore, we have developed our software based on this tool, but trying to overcome these drawbacks.

5.2 The HSF software

This software has been developed in C++, using Qt and OpenGL libraries. The C++ language, also called object-oriented C allows to use data structures in a natural way. The Qt library has been used to implement the user interface. The OpenGL library has been used to show the objects and the homology computations. We have followed these paradigms throughout the software development:

- Independence of the modelling problem. The algorithms operate on different complexes. This will allow to use them in any kind of problem.

- Integration with previously developed software, so that the interfaces are able to communicate. In this sense, the output files of Voxelo are readable by the developed software and viceversa.
- Portability between systems, which is largely guaranteed by the choice of C++ as programming language since it is available in almost all systems. We have tested the software on different machines under Linux, Windows and Mac OS.
- Scalability for adding features, process data or routines without altering the original development.

5.2.1 Class diagram

In C++ complex data types can be defined using classes. A class defines a data type with its attributes and operations, by providing an abstraction of a modelled entity. An object is an instance of a class, that has the attributes defined by the class, and on which you can run the operations defined for the class.

The different classes of the HSF software and the relations between them are shown in Figure 5.1

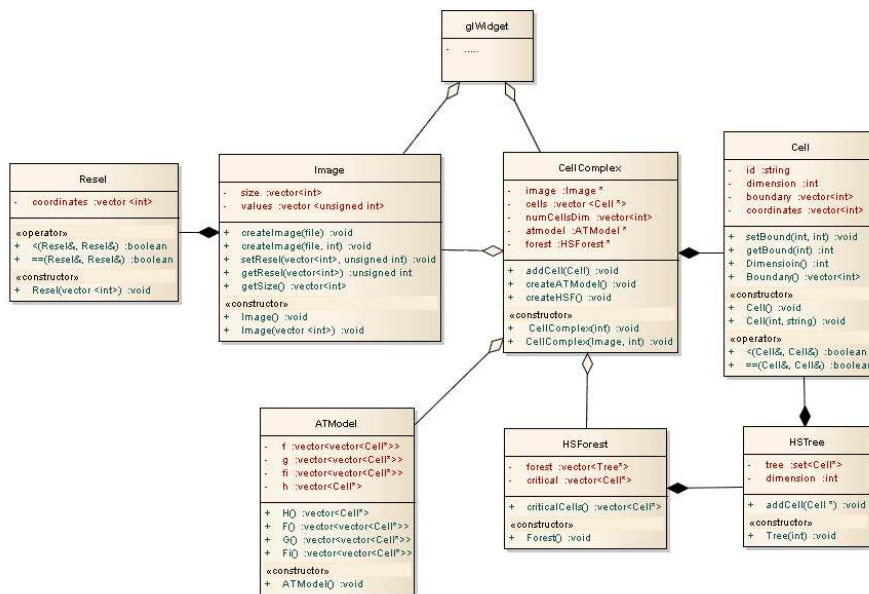


Figure 5.1: Class diagram

5.2.2 Interface

The main interface consists of the following parts, as illustrated in Figure 5.2:

- **The drawing area:** Main part of the interface, where the complex and homology computations are shown.
- **Menu bar:** Displays menu items and menu options. This bar gives access to Open, Clear and Quit commands.
- **Tool bar:** Displays small icons, each one corresponding to a process tool or a command.
- **Status bar:** The Status bar displays helpful information after an option is executed.

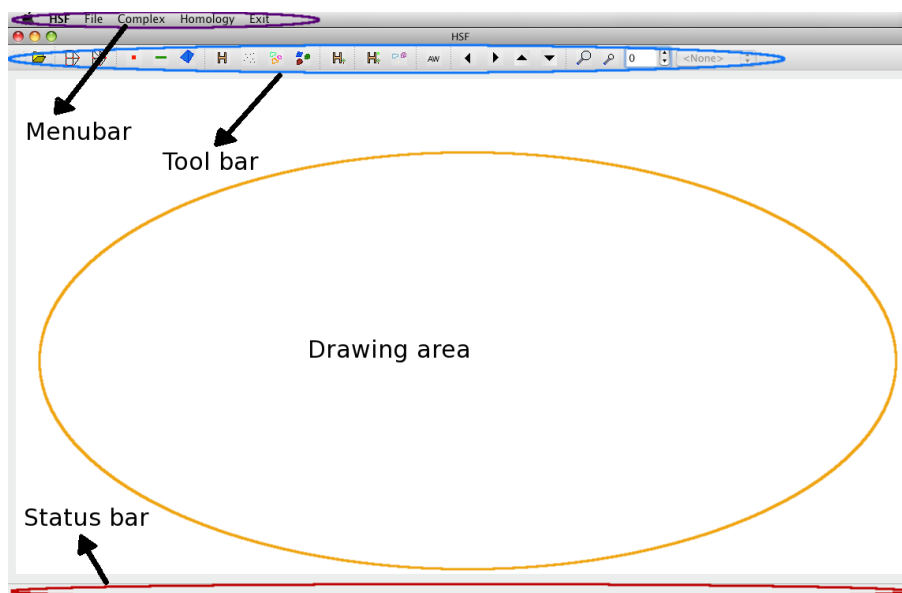


Figure 5.2: Interface design

Menu bar

File

The options available in the File menu, are *Open* and *Clear*. When selecting *Open*, a new menu is displayed (see Figure 5.3). This menu allows to open different types of files with the options *Open File*, *Open Image* and *Open Chain Complex* (see Figure 5.3).

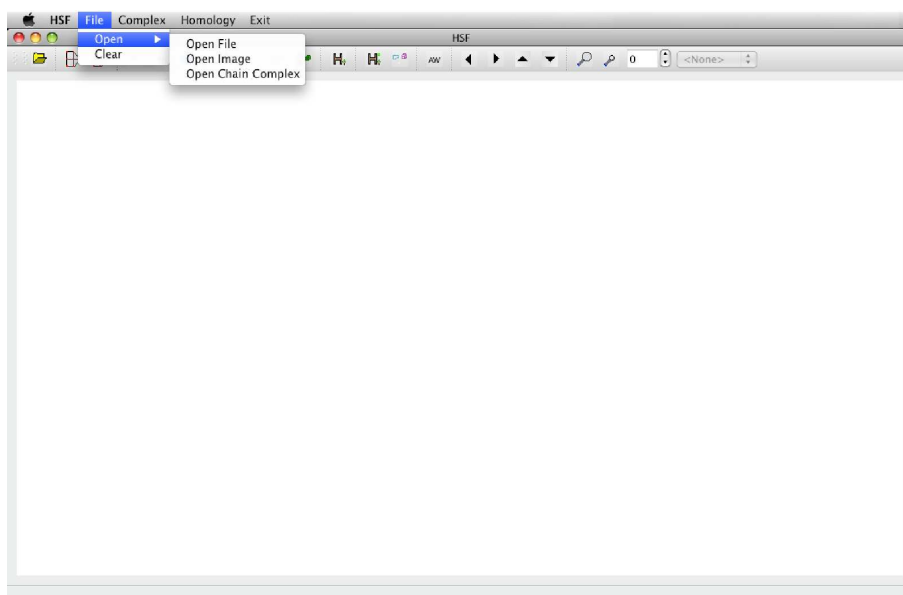


Figure 5.3: File menu

Open File The *Open File* option lets you open an existing file in which a set of points and their coordinates are stored. This file has a *.txt* extension, and the coordinates are inserted in the following way:

Size $x y z$
Points n
 $1 (x_1, y_1, z_1)$
 $2 (x_2, y_2, z_2)$
 \vdots
 $n (x_n, y_n, z_n)$

Where x, y, z are the dimensions of each one of the axes, n is the number of points, and x_i, y_i, z_i are the x, y, z coordinates of each point.

Once the file is selected, the set of points are shown in the drawing area (see Figure 5.4).

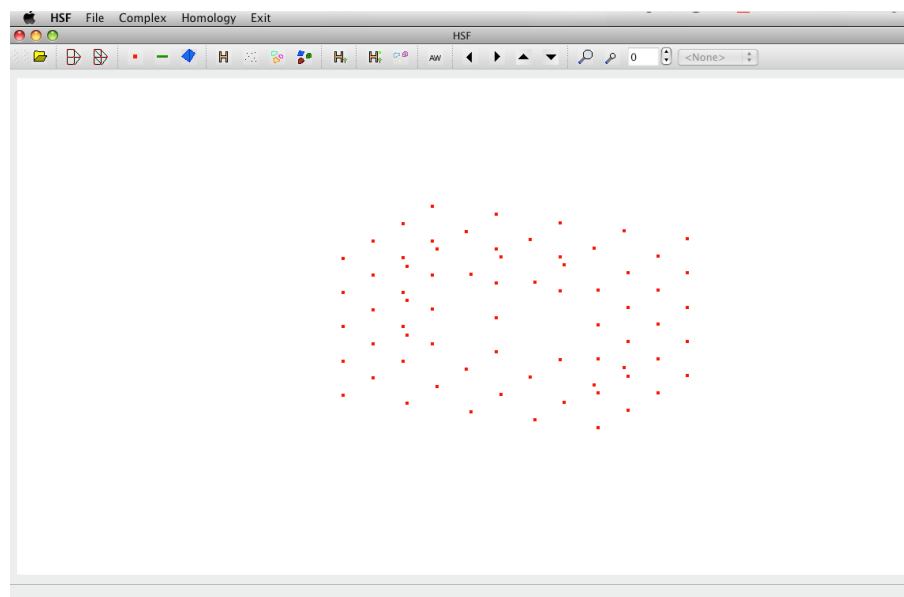


Figure 5.4: Points displayed in the drawing area

Open Images The *Open Images* option lets you open a set of *.jpg* 2D binary images, in order to create the 3D volume out of them. The menu displayed allows the selection of several images. The file name of the images needs to end in a consecutive number, like for instance *image_1.jpg*, *image_2.jpg*, *image_3.jpg*, etc. In Figures 5.5 and 5.6 a sequence of CT trabecular bone images and its corresponding display in the drawing area are shown.



Figure 5.5: Sequence of CT trabecular bone images

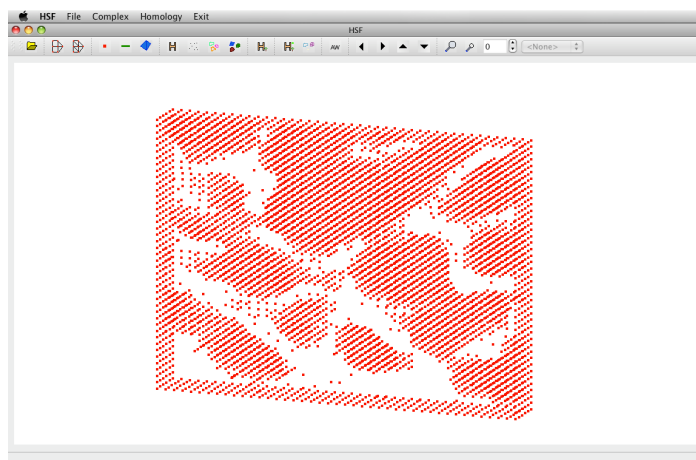


Figure 5.6: Points corresponding to the images shown in Figure 5.5, displayed in the drawing area

Open Chain Complex The *Open Chain Complex* option lets you open an existing file in which a chain complex is stored. This file has a *.chn* extension, and the chains are inserted in the following way:

```

maximal dimension k
dimension 0: n0
                id00
                id01
                ⋮
                id0n0
dimension 1: n1
                boundary id10 = id0i + id0j + ...
                boundary id11 = id0k + id0l + ...
                ⋮
                boundary id1n1 = id0s + id0t + ...
⋮
dimension k: nk
                boundary idk0 = idn-1r + idn-1m + ...
                boundary idk1 = idn-1u + idn-1v + ...
                ⋮
                boundary idknk = idn-1d + idn-1w + ...
  
```

Where k indicates the maximal dimension of the complex, and then for each dimension, the cells identifiers and its boundaries are included. Let us notice that the 0-dimensional cells are considered to have null boundary, and the boundary of each n -cell is represented as a sum of cells of dimension $n - 1$.

This option is designed for chain complexes whose cells do not satisfy the boundary relations of a simplicial or cell complex. In this case, we could perform homology computations on this complex, but the drawing area will not show neither the complex or the computations.

Clear In case we want to start working with a new object, we press *Clear*. This option will clean the drawing area and all the stored variables.

Complex

The options available in the Complex menu, are *Simplicial Complex*, *Cellular Complex* and *Save*. These options will be available only in case we have chosen before the *Open File* or *Open Image* options.

Simplicial Complex The *Simplicial Complex* option creates a simplicial complex from the previously stored set of points. The algorithm implemented to create the complex will be explained in section 5.2.3. Once this option has been selected, the simplicial complex will be immediately shown in the drawing area (see Figure 5.7).

Cell Complex The *Cell Complex* option creates a cell complex from the previously stored set of points. The algorithm implemented to create the complex will be explained in section 5.2.3. Once this option has been selected, the Cell Complex will be immediately shown in the drawing area (see Figure 5.8).

Save The *Save* option saves the complex in a selected file. This file has *.chn* extension, and the complex is stored following the *.chn* format explained before.

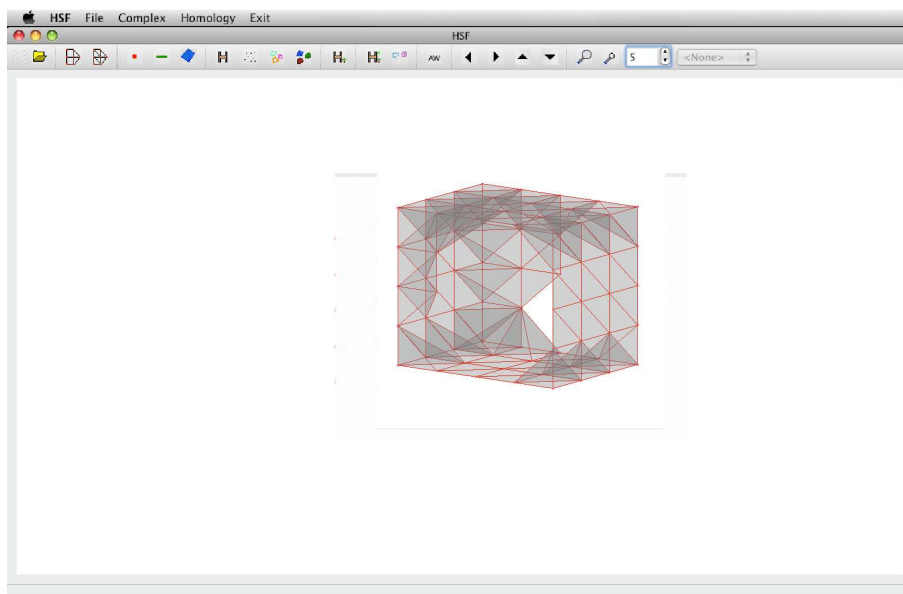


Figure 5.7: Simplicial complex displayed in the drawing area

Homology

The options available in the Homology menu, are *Compute HSF*, *Compute AT-model* and *Save*. These options will be available only in case we have chosen before the *Open File*, *Open Image* or *Open Chain Complex* options.

Compute HSF The *Compute HSF* option computes the HSF representation of the previously computed/stored cell/simplicial/chain complex. The algorithm implemented to create the HSF representation is explained in Section 3.6.

Compute AT-model The *Compute AT-model* option computes the AT-model of the previously computed/stored cell/simplicial/chain complex. The algorithm implemented to create the AT-model is explained in Section 3.1.

Exit

The option available in the Exit menu, is *Quit*. This option exists the application.

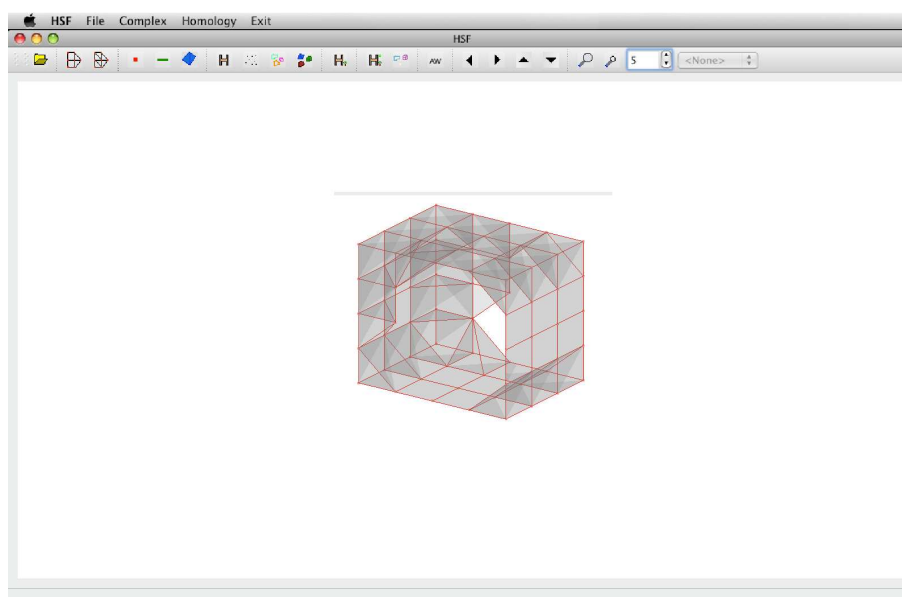


Figure 5.8: Cell complex displayed in the drawing area

Tool bar



Displays a dialog where a file can be selected to be open. This tool is equivalent to the *Open File* option of the *File* menu.



Creates a cell complex. This tool is equivalent to the *Cell Complex* option of the *Complex* menu.
















Creates a simplicial complex. This tool is equivalent to the *Simplicial Complex* option of the *Complex* menu.



When a complex has been created, if this tool is selected, only the 0-cells will be shown in the drawing area (see Figure 5.9 a))



When a complex has been created, if this tool is selected, only the 0-cells and 1-cells will be shown in the drawing area (see Figure 5.9 b))

-  When a complex has been created, if this tool is selected, the 0-cells, 1-cells and 2-cells will be shown in the drawing area (see Figure 5.9 c))
-  Computes the AT-Model. This tool is equivalent to the *Compute AT-model* option of the *Homology* menu.
-  When an AT-model has been computed and this tool is selected, cells belonging to the same connected component are colored in the same colour.
-  When an AT-model has been computed and this tool is selected, the 1-cycles of the complex are shown.
-  When an AT-model has been computed and this tool is selected, the 2-cycles of the complex are shown.
-  Computes the HSF representation. This tool is equivalent to the *Compute HSF* option of the *Homology* menu.
-  Computes one single step of the HSF algorithm.
-  Computes the next step of the HSF algorithm.
-  When the HSF algorithm is running step by step, if we select this option, the remaining cells at each step are shown (see for instance the examples shown in Section 3.6).
-  Shifts the drawing to the left.
-  Shifts the drawing to the right.
-  Shifts the drawing to the top.
-  Shifts the drawing to the bottom.



Zoom in



Zoom out



Allows different layers of the image to be shown



This menu allows to select between *None*, *0-1*, *1-2* and *2-3*. That is, each options shows the HSF trees of dimension $0 - 1$, $1 - 2$ and $2 - 3$ respectively (see Figure 5.10 for an example).

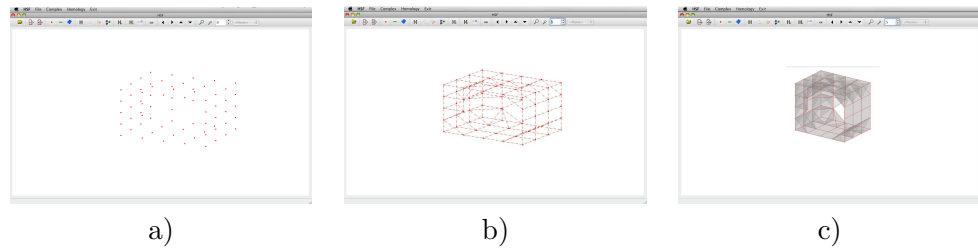


Figure 5.9: Different views of a complex

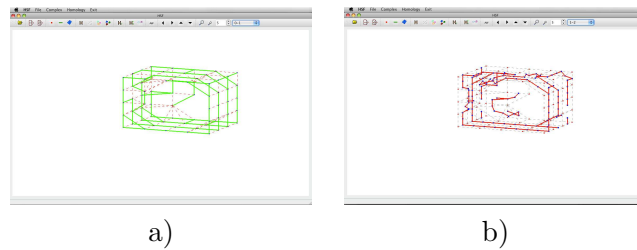


Figure 5.10: HSF of dimension $0 - 1$ on the left and $1 - 2$ on the right.

5.2.3 Algorithms

In this section we will briefly explain the algorithms that have been used in the software implementation. We can distinguish two main algorithm categories, one related with the object storage and complex generation, and another related with the homology computations.

Storing the object

The first step that needs to be done when using the HSF Software, consists of storing the object on which we will compute the homological information.

As mentioned in Section 5.2.2 three different types of input are considered in the HSF software. These inputs are a text file containing a set of points, a sequence of images and a text file containing a chain complex. In this last case, no further computations need to be done before computing homology. In the two first cases, either a Cell or a Simplicial complex needs to be associated to the input. The algorithms used for that purpose are the ones presented in [Molina-Abril 08].

Cell complex algorithm. A digital object V formed by a set of voxels is extracted from the input of the software (either a sequence of images, or the file containing the point's coordinates). This object V is stored as a matrix where the 0 value corresponds to the absence of points in such coordinates, and 1 corresponds to the existence of a point in such coordinates.

The cell complex $K(V)$ and the cubical complex canonically associated to V consisting in the set of geometric realizations of the voxels forming V are homotopically equivalent, and consequently, present the same homological information.

To obtain the cell complex $K(V)$ we do as follows (see [Molina-Abril 09a]). Each black voxel can be seen as a point (0-cell) of our complex. The algorithm consist on dividing the volume into overlapped (its intersection being a "square" of four voxels mutually 8-adjacent) unit cubes formed by eight voxels mutually 26-adjacent, and to associate each unit cube configuration with its corresponding cell. We scan the complete volume, always taking as elementary step a unit cube (see Figure 5.11).

The cell associated to a unit cube configuration is a 0-cell if there is a single point. If there are two points, the complex is a 1-cell which is the edge connecting both of them. With three or four coplanar points on the set, the 2-cell associated is a polygon. If there are four non coplanar points or more, the 3-cell is a polyhedra. In other words, the cell associated to a unit cube configuration is just the convex hull of the black points and all its lower dimension faces (see Figure 5.12). Note that for 3-cells, their 2-dimension faces are either triangles or squares.

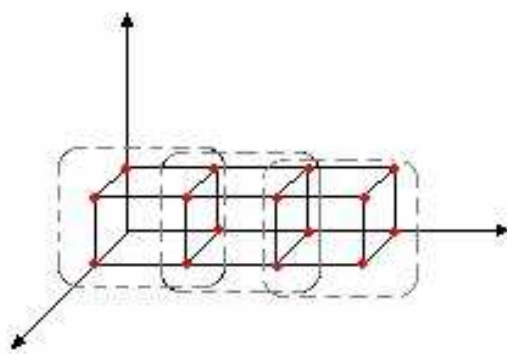


Figure 5.11: Overlapped cubes

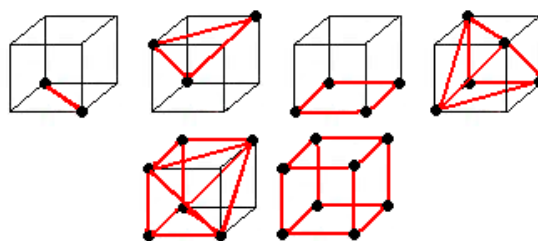


Figure 5.12: Different cells for different distributions of black points inside a unit cube

Once we have covered all the volume and joined all the cells, we can build the complete cell complex without incoherencies. This method has been implemented, and its efficiency is performed using a look-up table including all the possible configurations of black points. The pseudocode of the algorithm is 3.

Algorithm 3 CellComplex(V)

```

K = EmptyCellComplex()
s = EmptySetofCells()
for every unit cube  $C \in V$  do
  s = LookUpTable( $C$ )
  for every cell  $\sigma \in s$  do
    if  $\sigma \ni K$  then
       $K = K + \sigma$ 
    end if
  end for
end for
Return  $K$ 

```

Simplicial complex algorithm. The method offers the possibility of canonically associate a simplicial complex equivalent to V and this fact allows to define in a straightforward manner the boundary operator of each cell in the complex (see [Molina-Abril 08]). The main idea is a slight modification of the algorithm proposed in [Kenmochi 98] to create a new triangulation method on the surface of a digital volume.

Due to the fact that a $2 \times 2 \times 2$ volume includes eight points, each of which can be black or white, there are 256 different patterns considering the distribution of these points. If we ignore the congruent patterns differing only by the rotations of the centre of the unit cube, the 256 patterns can be reduced to 23.

The aim is to divide each cell into points, edges, triangles or tetrahedra to build the simplicial complex. Using 26-adjacency to divide each cell, we will find crossing simplex when there are four coplanar points into the cube. In all these cases we choose one diagonal to avoid simplex crossing (see Figure 5.13).

The problematic situations arise only when there are four or more black points into a $2 \times 2 \times 2$ set. In all these cases we have to study each possible configuration separately.



Figure 5.13: Example showing one possible selection of the diagonals for each face of the cube. This selection gives rise to a concrete tetrahedralization of the cube.

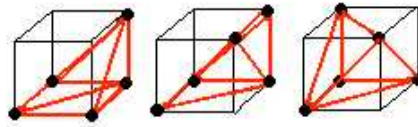


Figure 5.14: Different configurations containing 5 black points

As is shown in Figure 5.14, given a cube with five black points, there are three possible positions of this points without taking into account cube's rotations. In figure 5.14(a) a diagonal in the lower face has been chosen for the tetrahedralization. The same diagonal must be chosen in every case in order to avoid incoherencies and crossing edges between neighbour cubes. Due to this fact, all possible rotations of this point configuration must be considered separately, because depending on which face we find the four coplanar points, we have to choose a different tetrahedralization, according to the chosen diagonal. For this particular case, there exist six different rotations that must be taken into account. The other two situations ((b), (c)) are easier to carry out, because we can apply the tetrahedralization showed in the picture, to all of the possible rotations of the cube without incoherencies.

With six black points, there are also three possible configurations. In two of them we have to consider every rotation separately, and there are twelve possibilities for each one.

Considering seven black points in the cube, eight different configurations are possible, and we have to build a different tetrahedralization for each one of them.

Finally, when all the points are black into the cube, there is only one possible tetrahedralization (see Figure 5.13) according to the diagonal we

chose for each cube's face.

Once we have covered the entire volume and associate the simplicial complex to each cube, we proceed building a general simplicial complex for the total image. If we join the different complexes obtained in the previous step, and avoid taking several times the same simplex (those who are in the intersection faces between cubes) a simplicial complex without incoherencies is finally obtained.

This method has been implemented, and its efficiency is performed using a look-up table including all the possible configurations of black points.

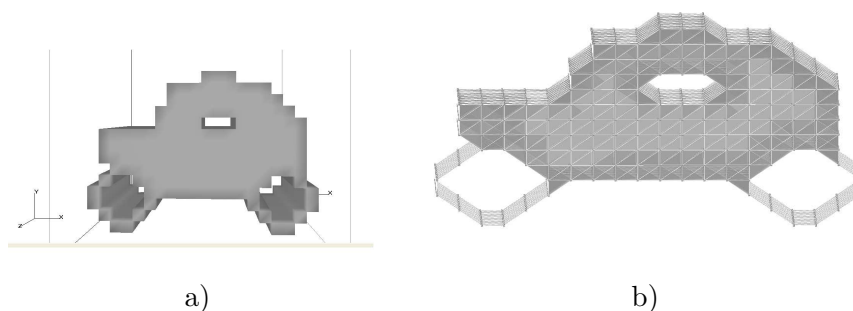


Figure 5.15: a) A 3D binary digital image and b) the corresponding simplicial complex.

The pseudocode of the algorithm is 4.

Algorithm 4 $\text{SimplicialComplex}(V)$

```

S = EmptySimplicialComplex()
s = EmptySetofSimplexes()
for every unit cube  $C \in V$  do
  s = LookUpTable(C)
  for every simplex  $\sigma \in s$  do
    if  $\sigma \ni S$  then
       $S = S + \sigma$ 
    end if
  end for
end for
Return S

```

Homology algorithms

Computing the HSF representation Given a simplicial/cell complex, its HSF representation is computed using Algorithm 2 of Section 3.6. Several examples of the output of this Algorithm can also be found in that Section.

Computing the AT-model The AT-model for the previously obtained simplicial/cell complex is computed using the algorithm described in Section 3.1. An example of the output of this algorithm is shown in Figure 5.16.

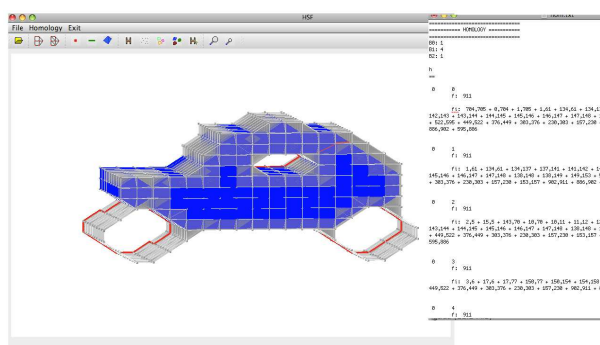


Figure 5.16: The output of the homology computation process: representative homology generators are colored in red (1-cycles) and blue (2-cycles), and an output file containing the Betti numbers and the value of the morphisms f , g and ϕ for every cell.

Chapter 6

Conclusions and future work.

Several applications in digital imagery are in need of a flexible and topologically-consistent framework for nD object analysis and recognition. We have presented here a new digital image representation and processing framework based on chain homotopies allowing an advanced topological analysis of cell complexes. The main notion in this representation is called Homological Spanning Forest (or HSF, for short) for a digital object due to the fact that it can be considered as a suitable generalization to higher dimensional cell complexes of the topological meaning of a spanning tree of a geometric graph. This new model for a digital object O is a set of directed forests, which can be constructed under an underlying cell complex format $K(I)$ of the image.

In the immediate future, we have the intention of progressing in the following directions:

- (a) A short-term objective is to extend this method to higher dimensions. This would allow a fast topological-controlled processing of big images like for example medical 4D images. Whereas the study of 2D and 3D digital images has been very fruitful, in the study of 4D-phenomena many research questions related to the topology are still fully open. 4D-images analysis is an important next step, because it adheres to the dimensionality of what is the physical reality. We also intend to achieve parallelism in this context.
- (b) The application of the HSF schema brings many advantages for fast geometrical transformations of images, and for implementing flexible and reliable methods for structural image analysis (see for instance

[Feichtinger 10]). Depending of the application we want to deal with, it seems possible to modulate the potential full flexibility of the proposed framework, in such a way that its conceptual description would be suitable for a concrete topological task (skeletons, thinning, Reeb graphs, mathematical morphology, etc).

- (c) To integrate in the HSF framework persistence homological techniques [Edelsbrunner 00] and discrete differential forms methods [Desbrun 05].
- (d) A carefully study about cycle transformability questions and others related to relative homology and advanced algebraic topological information (cohomology algebra, homology $A(\infty)$ -coalgebra, cohomology operations,...) in 3D or 4D HSF context will also be done in a future work. First steps in this direction have been done in [Berciano 09] and [Berciano 12].

Bibliography

- [Allili 01] M. Allili, K. Mischaikow & A. Tannenbaum. *Cubical homology and the topological classification of 2d and 3d imagery*. In Proceedings of International Conference Image Processing, volume 2, pages 173–176, 2001.
- [Ankeney 83] Lawrence A. Ankeney & Gerhard X. Ritter. *Cellular topology and its applications in image processing*. International Journal of Parallel Programming, vol. 12, pages 433–456, 1983.
- [Ayala 96] Rafael Ayala, Eladio Domínguez, Angel R. Francés & Antonio Quintero. *Determining the components of the complement of a digital $(n-1)$ -manifold in \mathbb{Z}^n* . In Proceedings of the 6th International Workshop on Discrete Geometry for Computer Imagery, pages 163–176, London, UK, 1996. Springer-Verlag.
- [Ayala 10] R. Ayala, D. Fernandez-Tertero & J.A. Vilches. *Perfect Discrete Morse Functions on 2-Complexes*. Proceedings of the 3rd International Workshop on Computational Topology in Image Context (CTIC 2010), pages 19–25, 2010.
- [Berciano 09] A. Berciano, H. Molina-Abril, A. Pacheco, P. Pilarczyk & P. Real. *Decomposing Cavities in Digital Volumes into Products of Cycles*. In Srečko Brlek, Christophe Reutenauer & Xavier Provençal, editors, DGCI, volume 5810 of *Lecture Notes in Computer Science*, pages 263–274. Springer, 2009.

- [Berciano 12] A. Berciano, H. Molina-Abril & P. Real. *Searching high order invariants in computer imagery*. *Applicable Algebra In Engineering, Communication and Computing*, doi: 10.1007/s00200-012-0169-5, 2012.
- [Bertrand 97] G. Bertrand & M. Couprie. *Some Structural Properties of Discrete Surfaces*. In *DGCI '97: Proceedings of the 7th International Workshop on Discrete Geometry for Computer Imagery*, pages 113–124, London, UK, 1997. Springer-Verlag.
- [Cardoze 06] D.E. Cardoze, G.L. Miller & T. Phillips. *Representing Topological Structures Using Cell-Chains*. In *Geometric Modeling and Processing*, pages 248–266, Pittsburgh, PA, July 2006.
- [Chassery 79] Jean-Marc Chassery. *Connectivity and consecutivity in digital pictures*. *Computer Graphics and Image Processing*, vol. 9, no. 3, pages 294–300, 1979.
- [Cho] *Computational Homology Project*. <http://chomp.rutgers.edu/>.
- [Cohen 73] M.M. Cohen. *A course in simple-homotopy theory*. *Graduate Texts in Mathematics*, 1973.
- [Cormen 01] T.H. Cormen, Ch. E. Leiserson & Rivest R.L. *Introduction to algorithms*. MIT, 2001.
- [Delfinado 95] C.J.A. Delfinado & H. Edelsbrunner. *An Incremental Algorithm for Betti Numbers of Simplicial Complexes on the 3-Sphere*. *Comput. Aided Geom. Design*, vol. 12, pages 771–784, 1995.
- [Desbrun 05] Mathieu Desbrun, Eva Kanso & Yiyong Tong. *Discrete differential forms for computational modeling*. In *ACM SIGGRAPH 2005 Courses, SIGGRAPH '05*, New York, NY, USA, 2005. ACM.

- [Dijkstra 59] E. W. Dijkstra. *A note on two problems in connexion with graphs*. Numerische Mathematik, vol. 1, pages 269–271, 1959.
- [Dousson 99] X. Dousson, J. Rubio, F. Sergeraert & Y. Siret. *The Kenzo program* <http://www.fourier.ujfgrenoble.fr/sergeraert/>, 1999.
- [Edelsbrunner 00] H. Edelsbrunner, D. Letscher & A. Zomorodian. *Topological persistence and simplification*. In Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pages 454–463, Washington, DC, USA, 2000. IEEE Computer Society.
- [Eilenberg 54] S. Eilenberg & S. Mac Lane. *On the groups $H(\pi, n)$, I, II, III*. Annals of Math, vol. 58, 60, 60, pages 55–106, 48–139, 513–557, 1953, 1954.
- [El-Kwae 00] Essam A. El-Kwae & Mansur R. Kabuka. *Binary object representation and recognition using the Hilbert morphological skeleton transform*. Pattern Recognition, vol. 33, no. 10, pages 1621–1636, 2000.
- [Feichtinger 10] Hans G. Feichtinger & Darian M. Onchis. *Constructive realization of dual systems for generators of multi-window spline-type spaces*. J. Comput. Appl. Math., vol. 234, pages 3467–3479, October 2010.
- [Forman 95] Robin Forman. *A discrete Morse theory for cell complexes*. In S. T. Yau, editeur, Geometry, Topology and Physics for Raoul Bott. International Press, 1995.
- [Forman 98] Robin Forman. *Morse theory for cell complexes*. Advances in Mathematics, vol. 134, pages 90–145, 1998.
- [Forman 02] R. Forman. *A Users Guide to Discrete Morse*. Sèminare Lotharinen de Combinatore 48, 2002.
- [González-Díaz 05a] R. González-Díaz, B. Medrano, P. Real & J. Sanchez-Pelaez. *Algebraic Topological Analysis of Time-*

- sequence of Digital Images*. Lecture Notes in Computer Science, vol. 3718, pages 208–219, 2005.
- [González-Díaz 05b] Rocío González-Díaz & Pedro Real. *On the cohomology of 3D digital images*. Discrete Appl. Math., vol. 147, no. 2-3, pages 245–263, 2005.
- [González-Díaz 08] R. González-Díaz, M.J. Jiménez, B. Medrano, H. Molina-Abril & P. Real. *Integral Operators for Computing Homology Generators at Any Dimension*. In Jos Ruiz-Shulcloper & Walter Kropatsch, editors, Progress in Pattern Recognition, Image Analysis and Applications, volume 5197 of *Lecture Notes in Computer Science*, pages 356–363. Springer Berlin / Heidelberg, 2008.
- [González-Díaz 09] R. González-Díaz, M. J. Jiménez, B. Medrano & P. Real. *Chain homotopies for object topological representations*. Discrete Appl. Math., vol. 157, no. 3, pages 490–499, 2009.
- [Gouaillard 05] A. Gouaillard. *Contexte générique bi-multirésolution basé ondelettes pour l'optimisation d'algorithmes de surfaces actives*. PhD thesis, Inst. National des Sciences Appliquées de Lyons, 2005.
- [Gugenheim 89] V. K. A. M. Gugenheim, L. A. Lambe & J. D. Stasheff. *Perturbation theory in differential homological algebra*. Illinois J. Math, vol. 33, pages 357–373, 1989.
- [Gugenheim 91] V. K. A. M. Gugenheim, L. A. Lambe & J. D. Stasheff. *Perturbation theory in differential homological algebra. II*. Illinois J. Math., vol. 35, no. 3, pages 357–373, 1991.
- [HAP] *Homological Algebra Programming*
<http://hamilton.nuigalway.ie/Hap/www/index.html>.
- [Kaczynski 98] T. Kaczynski, M. Mrozek & M. Slusarek. *Homology computation by reduction of chain complexes*. Computers & Mathematics with Applications, vol. 35, no. 4, pages 59–70, 1998.

- [Kaczynski 04] T. Kaczynski, K. Mischaikow & M. Mrozek. Computational homology. Applied Mathematical Sciences, 2004.
- [Kenmochi 98] Y. Kenmochi & Ichikawa A. Imiya A. *Boundary extraction of discrete objects*. Computer vision and image understanding, vol. 71, pages 281–293, 1998.
- [Khalimsky 86] E. Khalimsky. *Pattern analysis of N-dimensional digital images*. Proc. IEEE Int. Conf. Systems, Man and Cybernetics, pages 1559–1562, 1986.
- [Klette 00] Reinhard Klette. *Cell complexes through time*. In Longin J. Latecki, David M. Mount & Angela Y. Wu, editeurs, Proc. Vision Geometry IX, volume 4117, pages 134–145, 2000.
- [Klette 04] Reinhard Klette & Azriel Rosenfeld. Digital geometry: Geometric methods for digital picture analysis. Morgan Kaufmann Series in Computer Graphics, 2004.
- [Kong 89] T. Y. Kong & A. Rosenfeld. *Digital topology: introduction and survey*. Comput. Vision Graph. Image Process., vol. 48, no. 3, pages 357–393, 1989.
- [Kong 91] T. Yung Kong, Ralph Kopperman & Paul R. Meyer. *A topological approach to digital topology*. Am. Math. Monthly, vol. 98, pages 901–917, 1991.
- [Kovalevsky 89] V. Kovalevsky. *Finite topology as applied to image analysis*. Computer Vision, Graphics and Image Processing, vol. 46, pages 141–161, 1989.
- [Kovalevsky 05] V. Kovalevsky. *Algorithms in Digital Geometry based on Cellular Topology*. Lecture Notes in Computer Science, vol. 3322, pages 366–393, 2005.
- [Kovalevsky 06] V. Kovalevsky. *Axiomatic Digital Topology*. J. Math. Imaging Vision, vol. 26, pages 41–58, 2006.
- [Kovalevsky 08] V. Kovalevsky. *Geometry of locally finite spaces*. House Dr. Baerbel Kovalevski, 2008.

- [Kropatsch 07] Walter G. Kropatsch, Yll Haxhimusa & Adrian Ion. *Multiresolution Image Segmentations in Graph Pyramids*. In Abraham Kandel, Horst Bunke & Mark Last, editeurs, *Applied Graph Theory in Computer Vision and Pattern Recognition*, volume 52 of *Studies in Computational Intelligence*, pages 3–41. Springer, 2007.
- [Lewiner 03] T. Lewiner, H. Lopes, G. Tavares & L. Matmídia. *Towards optimality in Discrete Morse Theory*. *Experimental Mathematics*, vol. 12, page 2003, 2003.
- [Lienhardt 91] P. Lienhardt. *Topological models for Boundary Representation: a comparison with n-dimensional generalized maps*. *Computer-Aided Design*, vol. 23, no. 1, pages 59–82, 1991.
- [Lin] *Exact computational linear algebra <http://linalg.org/>*.
- [Mac Lane 95] S Mac Lane. *Homology*. Springer, Verlag, 1995.
- [McAndrew 96] A McAndrew & C. Osborne. *A survey of algebraic methods in digital topology*. *Journal of Mathematical Imaging and Vision*, vol. 6, pages 139–159, 1996.
- [Molina-Abril 08] H. Molina-Abril & P. Real. *Advanced Homological information on 3D Digital volumes*. SSPR 2008, LNCS, vol. 5342, pages 361–371, 2008.
- [Molina-Abril 09a] H. Molina-Abril & P. Real. *Cell AT-models for digital volumes*. GbR 2009, LNCS, vol. 5534, pages 314–323, 2009.
- [Molina-Abril 09b] H. Molina-Abril & P. Real. *Homological Computation Using Spanning Trees*. CIARP 2009, Lecture Notes in Computer Science, vol. 5856, pages 272–278, 2009.
- [Molina-Abril 10] H. Molina-Abril & P. Real. *Towards Optimality in Discrete Morse Theory through Chain Homotopies*. CTIC 2010, Imagen-a, vol. 1, pages 33,40, 2010.

- [Molina-Abril 12a] H. Molina-Abril & P. Real. *Homological Optimality in Discrete Morse Theory through Chain Homotopies*. Pattern Recognition Letters, doi: 10.1016/j.patrec.2012.01.014, 2012.
- [Molina-Abril 12b] H. Molina-Abril & P. Real. *Homological Spanning Forest Framework for 2D Image Analysis*. Annals of Mathematics and Artificial Intelligence, doi: 10.1007/s10472-012-9297-7, 2012.
- [Munkres 84] J.R. Munkres. Elements of algebraic topology. Addison Wesley, 1984.
- [Munkres 00] J.R. Munkres. Topology (second edition). Prentice Hall, 2000.
- [Niethammer 02] M. Niethammer, A.N. Stein, W.D. Kalies, P. Pilarczyk, K. Mischaikow & A. Tannenbaum. *Analysis of blood vessel topology by cubical homology*. Proc. of the International Conference on Image Processing, vol. 2, pages 969–972, 2002.
- [Parker 97] J.R. Parker. Algorithms for image processing and computer vision. John Wiley and Sons., 1997.
- [Ple] *Plex: Simplicial complexes in Matlab*. <http://comptop.stanford.edu/programs/plex/>.
- [Romero 10] Ana Romero & Francis Sergeraert. *Discrete Vector Fields and Fundamental Algebraic Topology*. CoRR, vol. abs/1005.5685, 2010.
- [Rosenfeld 66] A. Rosenfeld & J.L. Pfaltz. *Sequential Operations in Digital Picture Processing*. J. ACM, vol. 13, no. 4, pages 471–494, 1966.
- [Rosenfeld 70] A. Rosenfeld. *Connectivity in Digital pictures*. Journal for Association of Computing Machinery, vol. 17, no. 1, pages 146–160, 1970.

- [Sergeraert 94] F. Sergeraert. *The computability problem in algebraic topology*. Advances in Mathematics, vol. 104, pages 1–29, 1994.
- [Webster 01] J. Webster. Cell complexes and digital convexity, pages 272–282. Springer-Verlag New York, Inc., 2001.
- [Whitehead 49] J.H.C. Whitehead. *Combinatorial homotopy I*. Bull. Amer. Math. Soc, vol. 55, pages 213–245, 1949.
- [Whitehead 50] J.H.C. Whitehead. *Simple homotopy types*. Amer. J. Math., vol. 72, pages 1–57, 1950.
- [Żelawski 05] M. Żelawski. *Pattern Recognition based on homology theory*. Machine Graphics and Vision, vol. 14, pages 309–324, 2005.