Morse complexes for shape segmentation and homological analysis: discrete models and algorithms

Leila De Floriani[1] and Ulderico Fugacci[1] and Federico Iuricich[2] and Paola Magillo[1]

[1]Department of Computer Science, Bioengineering, Robotics, and Systems Engineering, University of Genova, Genova, Italy.
[2]Department of Computer Science and UMIACS, University of Maryland, College Park (MD), USA.

Abstract

Morse theory offers a natural and mathematically-sound tool for shape analysis and understanding. It allows studying the behavior of a scalar function defined on a manifold. Starting from a Morse function, we can decompose the domain of the function into meaningful regions associated with the critical points of the function. Such decompositions, called Morse complexes, provide a segmentation of a shape and are extensively used in terrain modeling and in scientific visualization. Discrete Morse theory, a combinatorial counterpart of smooth Morse theory defined over cell complexes, provides an excellent basis for computing Morse complexes in a robust and efficient way. Moreover, since a discrete Morse complex computed over a given complex has the same homology as the original one, but fewer cells, discrete Morse theory is a fundamental tool for efficiently detecting holes in shapes through homology and persistent homology. In this survey, we review, classify and analyze algorithms for computing and simplifying Morse complexes in the context of such applications with an emphasis on discrete Morse theory and on algorithms based on it.

1. Introduction

Morse theory [Mil63, Mat02] studies the relationships between the topology of a shape and the critical points of a real-valued smooth function defined on it. It has been recognized as an important tool for shape analysis and understanding in many applications, including physics, chemistry, medicine and geography.

Based on Morse theory, we can define a segmentation of a geometric shape endowed with a scalar function into collections of cells, called *Morse* and *Morse-Smale complexes*, based on the critical points of such function. Skeletal representations rooted in Morse theory, like Reeb graphs and contour trees, are compact 1-dimensional topological descriptors of the evolution of the level sets of a Morse function [BGSF08].

Morse theory is defined for smooth functions, but recently a discrete counterpart has been proposed in an entirely combinatorial setting by Robin Forman for cell complexes [For98b]. Discrete Morse theory is the basis for computing Morse decompositions of discretized geometric shapes, and it provides a tool for retrieving topological invariants, like homology groups. This survey focuses on a class of techniques, rooted in various approximations of smooth Morse theory and on discrete Morse theory, for shape analysis through segmentation, provided by Morse and Morse-Smale complexes, and through detection of holes in shapes provided by homological information.

In Section 2, we present Morse theory in the smooth case and some notions on homology and persistent homology. In Section 3, we review theoretical tools at the basis of several algorithms in the literature for computing Morse decompositions of shapes, namely piecewise linear Morse theory, the watershed transform, and the flow complex. We then review and classify algorithms based on such tools, which have been applied in terrain modeling, shape retrieval and scientific data visualization.

In Section 4, we present discrete Morse theory and, in Section 5, we review and classify approaches for computing a discrete Morse gradient. We distinguish between methods which conform to a given scalar function, useful for shape segmentation, and methods which do not have any constraint, useful for homology computation. In Section 6, we describe how to extract Morse and Morse-Smale com-

plexes from a discrete Morse gradient, as well as a combinatorial representation of such complexes.

Section 7 is devoted to simplification techniques, used to eliminate less meaningful or noisy morphological features in a dataset, and to adapt the level of abstraction to the current application task. In Section 8, we discuss the use of morphology in various applications, including multi-resolution morphological models for analysis and visualization of scalar fields, which encompass a hierarchy of simplifications and allow extracting a specific level of detail on the fly. Finally, Section 9 draws some concluding remarks.

2. Background notions

In this section, we present some background notions on cell complexes, on homology and persistent homology, and on Morse theory, that we will use in the rest of the survey.

2.1. Cell complexes

Cell complexes are used as a discretization and modeling tool in a large variety of application domains. Intuitively, a cell complex defines a decomposition of a geometric shape into simple subsets, called *cells*, glued together along their boundaries.

A *k-cell* (or cell of dimension k) is a homeomorphic image of the k-disk $D^k = \{x \in \mathbb{R}^k : |x| \leq 1\}$. A space $\Gamma \subseteq \mathbb{R}^n$ is called a *(regular) cell complex* if Γ is a finite union of cells with disjoint interiors such that, for each k-cell σ of Γ, the boundary of σ is in Γ^{k-1}, where, Γ^i, called the *i-skeleton* of Γ, is the union of the cells of Γ with dimension less than or equal to *i*. The union of the cells of Γ considered as a point set forms the *domain* of Γ.

If Γ is a cell complex, and τ and σ are two cells of Γ, τ is called a *(proper) face* of σ if τ is contained in the boundary of cell σ; σ is called a *(proper) coface* of τ. Any cell of Γ, which is not a face of any other cell of Γ, is called a *top cell*.

The *star* of a cell $\tau \in \Gamma$ is the set of cells $\sigma \in \Gamma$ which are cofaces of τ and is denoted as $St(\tau)$. A k-cell τ is said to be *adjacent* to a k-cell τ' if τ and τ' share a $(k-1)$-face. The *link* of a cell $\tau \in \Gamma$, denoted as $Lk(\tau)$, is the set of cells $\gamma \in \Gamma$ such that γ is a face of a coface of τ, and is not incident in τ.

2.1.1. Simplicial and cubical complexes

Simplicial and cubical complexes are special subclasses of cell complexes widely used for shape discretization. Simplicial complexes are built upon the concept of simplex. A *simplex* of dimension k, (or a *k-simplex*) is the convex hull of $k+1$ affinely independent points in $\mathbb{R}^n$. Given a k-simplex σ, any simplex τ, which is the convex hull of a non-empty subset of the points generating σ, is called a *face* of σ. A *simplicial complex* Σ is a finite set of simplices, such that each face of a simplex in Σ belongs to Σ, and the non-empty

intersection of any two simplices in Σ is a face of both. A simplicial complex is called a complex of order d if d is the maximum of the dimensions of its simplices. Simplicial d-complexes with a manifold domain are usually called *simplicial meshes*, which are *triangle meshes* for $d = 2$ and *tetrahedral meshes* for $d = 3$. Recall that a d-manifold $\mathbb{M}$ is a Hausdorff space that is locally homeomorphic to the d-dimensional Euclidean space.

An *elementary (hyper-)cube* of dimension k is the Cartesian product $\eta = I_1 \times \cdots \times I_k$ of k elementary intervals. The dimension of η is the number of non-degenerate intervals in the product decomposition, and a face of η is the product of a non-empty subset of the intervals $I_1, \cdots, I_k$. A *(hyper-)cubical complex* $\mathcal{Q}$ in $\mathbb{R}^k$ is a finite collection of elementary cubes in $\mathbb{R}^k$ such that, if an elementary cube belongs to $\mathcal{Q}$, then all of its faces belong to $\mathcal{Q}$, and each non-empty intersection of any two elementary cubes in $\mathcal{Q}$ is a face of both.

Cubical complexes, also called *regular grids*, have been extensively used for discretization of the domain of a scalar field, such as an image or a terrain. They admit very compact data structures. By simply storing a matrix of field values, we can retrieve all cells and their mutual adjacencies and incidences. On the other hand, simplicial and cell complexes require data structures which explicitly maintain connectivity information (see [DH05] for a survey).

2.2. Homology and persistent homology

Homology and persistent homology are powerful tools in shape analysis, since they provide invariants for shape description and characterization. These notions can be defined for any discretization of a geometric shape such as a cell, simplicial or hyper-cubical complex. For the sake of generality, we define homology and persistent homology for cell complexes.

Given a cell complex Γ, it is possible to define a *chain complex* associated with Γ. This latter is a pair $C_*(\Gamma) = (C_k(\Gamma), \partial_k)_{k \in \mathbb{Z}}$, where $C_k(\Gamma)$ is the $\mathbb{Z}_2$-vector space generated by the k-cells of cell complex Γ, and $\partial_k : C_k(\Gamma) \to C_{k-1}(\Gamma)$ is a homomorphism, called *boundary map*, which encodes the boundary relations between the k-cells and the $(k-1)$-cells of Γ such that $\partial^2 = 0$. We denote as $Z_k(\Gamma) = \ker \partial_k$ the $\mathbb{Z}_2$-vector space of the k-cycles of Γ and as $B_k(\Gamma) = \mathrm{Im} \partial_{k+1}$ the $\mathbb{Z}_2$-vector space of the k-boundaries of Γ. Then, the kth *homology group of Γ (with coefficients in $\mathbb{Z}_2$)* is defined as $H_k(\Gamma) = H_k(C_*(\Gamma)) = Z_k(\Gamma)/B_k(\Gamma)$.

Roughly speaking, homology groups reveal the presence of "holes" in a shape. The non-null elements of each homology group are cycles, which do not represent the boundary of any collection of cells of the cell complex Γ.

Given a cell complex Γ, we call the dimension β_k of its kth homology group the kth *Betti number* of Γ. For each k, the kth Betti number β_k measures the number of independent,

non-bounding, k-cycles in Γ. Thus, β_0 counts the number of connected components of the complex, β_1 counts the number of tunnels and holes, β_2 counts the number of voids or cavities, and so on.

Homology can be defined with coefficients in any Abelian group. The most complete homological information use $\mathbb{Z}$ as coefficient group, which is capable of dealing with the presence of torsion in a shape. It can be proven (see [AH35], Chapter X) that simplicial complexes embeddable in $\mathbb{R}^3$ have no torsion and thus, their $\mathbb{Z}$-homology groups reduce to $\mathbb{Z}_2$-homology.

Independently of the coefficient group, computation of homology is a difficult task. The classical way to compute homology describes the boundary maps ∂_k through matrices and reduces these boundary matrices by using an algorithm similar to Gauss elimination, called a *Smith Normal Form reduction* [Ago05, Mun84]. Unfortunately, the time complexity of such reduction algorithm is super-cubical in the number of cells of the cell complex.

Persistent homology [EH08, Zom05] aims at overcoming some intrinsic limitations of classical homology. Let Γ be a cell complex. A *filtration F of Γ* is a finite sequence of subcomplexes $\{\Gamma_m \mid 1 \leq m \leq M\}$ of Γ such that $\Gamma_1 \subseteq \Gamma_2 \subseteq \cdots \subseteq \Gamma_M = \Gamma$. The *$p$-persistent k^{th} homology group $H_k^p(\Gamma_m)$ of Γ_m* consists of k-cycles included from $C_k(\Gamma_m)$ into $C_k(\Gamma_{m+p})$ modulo boundaries. Persistent homology provides more information about a shape than classical homology: while homology captures cycles in a shape by factoring out the boundary cycles, persistent homology allows the retrieval of cycles that are non-boundary cycles at a certain step of the filtration and that turn into boundaries in some subsequent step. The persistence of a cycle during the filtration gives quantitative information about its relevance for the analysis of the shape.

Persistent homology can be computed using a Smith Normal Form reduction algorithm, and, by restricting to persistent homology with coefficients in a field (such as $\mathbb{Z}_2$), it can be retrieved more efficiently [Zom05]. For this reason, in spite of the fact that persistent homology admits arbitrary coefficients, in the literature it is always computed with coefficients in $\mathbb{Z}_2$.

2.3. Morse theory and Morse complexes

Morse theory [Mat02, Mil63] studies the relationships between the topology of a d-manifold $\mathbb{M} \subseteq \mathbb{R}^n$ and the critical points of a C^2-differentiable real-valued function defined on it. A point $p \in \mathbb{R}^n$ is a *critical point* of f if and only if the gradient $\triangledown f$ of f vanishes on p, i.e., $\triangledown f(p) = 0$. If the determinant of the Hessian matrix $Hess_p(f)$ of the second-order partial derivatives of f, evaluated in p, is not null (i.e. $det(Hess_p(f)) \neq 0$), then p is a *non-degenerate critical point*. The number of negative eigenvalues of $Hess_p(f)$

is called the *index k* of a critical point p and p is called a *k-saddle*, with $0 \leq k \leq n$. A 0-saddle is called a *minimum* and a d-saddle a *maximum*. The corresponding eigenvectors define the directions in which function f is decreasing. A function f is a *Morse function* when all its critical points are not degenerate. As a consequence of the *Morse Lemma* [Mil63], each non-degenerate critical point is isolated and, therefore, it has a neighborhood which does not contain other critical points.

An integral line of a function f is a maximal path everywhere tangent to the gradient of f. An integral line follows the direction in which the function has the maximum growth. An integral line, which connects a critical point p of index k to a critical point q of index $k+1$, is called a *separatrix line*. The integral lines cover the entire domain of f and they form cells, each corresponding to a critical point. Integral lines that converge to a critical point p of index k form a *k-cell*, called the *descending manifold* of p. Integral lines that originate from a critical point p of index k form a $(d-k)$-cell, called the *ascending manifold* of p. The ascending/descending manifolds are pairwise disjoint and partition the domain of $\mathbb{M}$ into cells, which form a cell complex since the boundary of each cell is the union of lower-dimensional cells. The collection of all the descending [ascending] manifolds form the *descending Morse complex Γ_D* [*ascending Morse complex Γ_A*]. The two Morse complexes are mutually dual.

In the 2D case (see Figure 1), the integral lines converging to a maximum/saddle/minimum form a 2-cell/1-cell/0-cell of the descending Morse complex, respectively (see Figures 1 (a) and (c)). Dually, the integral lines originating from a minimum/saddle/maximum form a 2-cell/1-cell/0-cell of the ascending Morse complex, respectively (see Figures 1 (b) and (d)). In 3D, the set of integral lines converging to a maximum/2-saddle/1-saddle/minimum form a 3-cell/ 2-cell/1-cell/0-cell of the descending Morse complex, respectively. The dual situation happens in the ascending Morse complex.

A Morse function f is called a *Morse-Smale function* if and only if the descending and ascending Morse complexes intersects transversally. The connected components of the intersection of the descending and ascending cells decompose $\mathbb{M}$ into a *Morse-Smale (MS) complex*, denoted Γ_{MS}. If f is a Morse-Smale function, then there is no integral line connecting two different critical points of f of the same index. Note that, in a d-dimensional *MS* complex, each 1-saddle is connected to exactly two minima and each $(d-1)$-saddle is connected to exactly two maxima, not necessarily distinct. The *1-skeleton* of the Morse-Smale complex, i.e., is the subcomplex composed of its 0-cells and 1-cells, is often called the *critical net*.
The intersection of the descending and ascending Morse complexes of Figures 1 (c) and (d) forms the Morse-Smale complex illustrated in Figure 1 (e). For each critical point
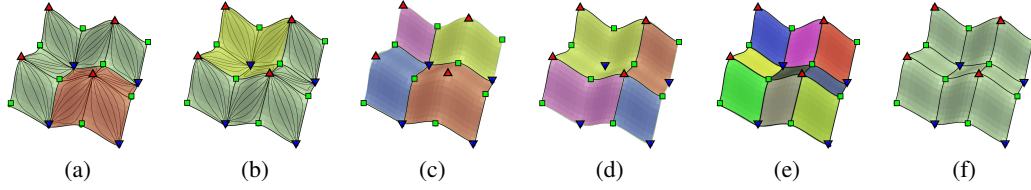
Figure 1: *Red and blue triangles indicate maxima and minima. Green squares indicate saddles. (a) The set of integral lines converging to a maximum and forming the (red) descending cell. (b) The set of integral lines originating from a minimum and forming the (yellow) ascending cell. The set of all the descending and ascending cells forming (c) the descending Morse complex Γ_D and (d) the ascending Morse complex Γ_A. (e) Resulting Morse-Smale complex and (f)1-skeleton of the Morse-Smale complex.*

p, the descending and ascending cells of p intersect only at p. Thus, the 0-cells of the Morse-Smale complex are the critical points. The 1-cells are the separatrix lines connecting pairs of critical points. The distinctive feature of Morse-Smale cells is the uniform gradient flow inside each of them.

A combinatorial representation for both ascending and descending Morse complexes is provided by an incidence graph. A *Morse Incidence Graph (MIG)* is a labeled graph $G = (N, A, \varphi)$, in which the set of nodes N is partitioned into $d + 1$ subsets $N_0, N_1, ..., N_d$, such that there is a one-to-one correspondence between the nodes in N_i (*i-nodes*) and the *i*-cells of the descending complex Γ_D, (and thus the $(n-i)$-cells of the ascending complex Γ_A), and there is an arc joining an *i*-node p with an $(i+1)$-node q if and only if the corresponding cells p and q differ in dimension by one, and p is on the boundary of q in Γ_D, (and thus q is on the boundary of p in Γ_A). Each arc a connecting an *i*-node p to an $(i+1)$-node q is labeled with the number $\varphi(a)$ of times *i*-cell p (corresponding to *i*-node p) in Γ_D is incident to $(i+1)$-cell q (corresponding to $(i+1)$-node q) in Γ_D. Thus, the *MIG* is a representation of the Hasse diagram [Ede87] of the ascending and descending Morse complexes and is the combinatorial representation of the 1-skeleton of the Morse-Smale complex.

Simplification of Morse functions, often necessary to remove non-relevant critical points, is achieved by applying an operator called *cancellation* [Mat02]. A *k*-cancellation transforms a Morse function f into a new Morse function g by removing two critical points, $(k+1)$-saddle p and a *k*-saddle q (with $0 \leq k < d$), and by locally modifying the gradient field of f around the integral lines of p and q. Critical points p and q can be canceled if and only if there is a unique integral line connecting them. After a *k-cancellation*(p,q), the two critical points p and q are removed and the integral lines originated/converging into them are modified as follows:

- the set of integral lines converging to p or q before the *cancellation* are transformed into a set of integral lines converging to those critical points of index $j > k$, that were the destination of integral lines starting at p,
- the set of integral lines that originated at q or p before the
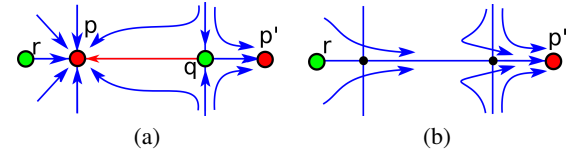
Figure 2: *Example of 1-cancellation(p,q) performed on a 2-saddle p and a 1-saddle q.*

cancellation are transformed into a set of integral lines originating at those critical points of index $i < k+1$ that were the origin of integral lines ending at q.

Figure 2 shows an example of a 1-*cancellation* performed on a 2-saddle p and a 1-saddle q. The integral lines that converged before to p and q (Figure 2 (a)) are redirected into integral lines converging to p' (Figure 2 (b)). Changes in the integral lines result in a transformation of the Morse and Morse-Smale complexes. The effects of a *cancellation* on the Morse complexes will be discussed in Section 7.

3. Morse complexes for shape segmentation

In this section, we discuss the most common approaches to the computation of Morse and Morse-Smale complexes for segmenting shapes discretized as simplicial or cubical complexes and endowed with a scalar field. Examples are intensity images, terrains, volume data, etc. The approaches discussed here are based on piecewise linear Morse theory, on the watershed transform, or on the flow complex. Algorithms based on discrete Morse theory will be reviewed in Sections 5.2 and 5.3. It has been shown in [VCY12] that the Morse-Smale complex built on the discretization of a smooth function has not, in general, the same structure as the 'true' Morse-Smale complex of the original function.

3.1. Piecewise linear Morse theory

Piecewise linear Morse theory has been introduced by Banchoff [Ban67, Ban70] to extend the results of smooth Morse
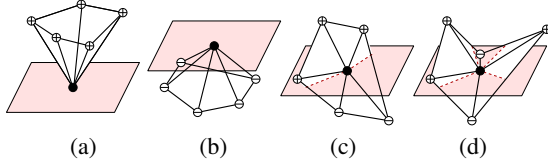
Figure 3: *Classification of a vertex according to Banchoff [Ban67]: (a) minimum, (b) maximum, (c) regular, (d) simple saddle.*

theory to polyhedral surfaces, and, thus, triangulated ones (see [BDF*08] for more details).

A triangulated surface is the graph of a real-valued function f defined at the vertices of a triangle mesh Σ on the plane, interpolated through linear interpolation. In order to ensure that critical points are isolated, function f is assumed to have different values at every pair of vertices connected by an edge in Σ, i.e., no *flat edges* are allowed [TIKU95].

Let us consider a vertex v, its star $St(v)$, and the horizontal plane passing through v (i.e., tangent to the graph of f at v), as illustrated in Figure 3. Vertex v is a local *maximum* or a local *minimum* if the plane does not intersect $St(v)$; it is *regular* if the plane cuts $St(v)$ into two parts; it is a *saddle* if the plane cuts $St(v)$ into k parts, with $k \geq 4$. The saddle is *simple* if $k = 4$, and *multiple* otherwise. The *multiplicity* of a saddle is equal to $k/2 - 1$. Note the piecewise linear case includes isolated degenerate critical points, such as multiple saddles, that do not occur in smooth Morse theory.

The characterization provided by Banchoff [Ban67] correctly detects critical points in dimensions two and three, while a more complete characterization for higher-dimensional spaces is based on Betti numbers (see [EHNP03]).

In [EHZ01, EHNP03], the notion of *Quasi Morse-Smale* (*QMS*) complex is introduced as a piecewise linear counterpart of the Morse-Smale complex for triangle and tetrahedral meshes. The idea behind a *QMS*, called *simulation of differentiability*, is that of extending the smooth notions to the piecewise linear case so as to guarantee that the complex has the same structure of its smooth counterpart. Numerical accuracy is achieved via local transformations that preserve the structure of the complex. In the 2D case, a *QMS* complex is a quadrangulation of the triangle mesh $\mathbb{M}$. In the *QMS* complex 0-cells are the critical points of f, the 1-cells connect minima to saddles and maxima to saddles [EHNP03]. In 3D, the 3-cells of the *QMS* complex are *crystals*, the faces of such crystals are quadrangles with vertices at a minimum, two 1- saddles and a 2-saddle, or at a maximum, two 2-saddles and a 1-saddle. Note that 1-cells and 2-cells of a *QMS* complex are not necessarily those of maximal ascent / descent, as they are in a Morse-Smale complex.

3.2. Watershed in the smooth and discrete cases

The watershed transform is an alternative framework to Morse theory. It has been first defined for grey-scale images, and several definitions exist in the discrete case [Mey94, VS91]. The watershed transform has also been defined for C^2-differentiable functions f over a connected domain D, having the property that the gradient ∇f is non-null everywhere except possibly at some isolated points. This includes Morse functions. Basic notions in the watershed transform are *catchment basins* and *watershed lines*, both defined in terms of topographic distance [Mey94, RM00].

The *topographic distance* $T_D(p,q)$ between two points p and q belonging to the domain D of f is

$$T_D(p,q) = \inf_P \int_P ||\nabla f(P(s))|| ds$$

where P is a smooth path inside D such that $P(0) = p$, $P(1) = q$. This definition ensures that the path which minimizes the topographic distance between p and q is the path of steepest slope, if it exists.

Given a minimum m_i of function f, the *catchment basin* $CB(m_i)$ of m_i is defined as the set of points which are closer in terms of topographic distance to m_i than to any other minimum. The *watershed* (or *watershed lines*) $WS(f)$ of f is defined as the set of points in D which do not belong to any catchment basin. When f is a C^2-differentiable Morse function, then the closure of the catchment basins of the minima of f are the closure of the 2-cells of the ascending Morse complex of f, and watershed lines form a subset of separatrix lines that connect saddles to maxima. Symmetrically, the closure of the catchment basins of function $-f$ provides the closure of the 2-cells of the descending Morse complex of f.

The watershed transform in the discrete case uses a discrete version of the topographic distance [Mey94] defined for an undirected labeled graph, $H = (N_H, A_H, f)$, where the nodes in N_H are labeled through function f. The nodes of graph H can be the pixels or the voxels in an image, or the vertices of a cell complex. The arcs can represent the adjacencies of the top cells in a grid or of the 1-cells of a cell complex. The *lower slope* at a node p is the maximal slope linking p to any of its neighbors of lower function value. A cost is associated with the arcs of H defined in terms of the lower slope. Given a path π between two nodes p and q in H, the π-*topographic distance* is given by the sum of costs for traversing all directed arcs composing π. The *topographic distance* $T(p,q)$ between p and q is the minimum of the π-topographic distances along all paths π between p and q.

The topographic distance is actually not a true distance function because it is equal to zero on distinct nodes of H, if they belong to the same plateau. A *plateau* is a connected set of nodes in H having the same function values. The definition of a catchment basin in graph H is similar to the smooth case. Given a minimum m_i of function f, where

m_i can be a single node or a plateau, the *catchment basin* of m_i is defined as $CB(m_i) = \{p \in D : f(m_i) + T(p, m_i) < f(m_j) + T(p, m_j), j \in I - \{i\}\}$.

3.3. Flow complex

The notion of *flow complex* [GJ08, DGG03] has been introduced in order to segment a shape defined through a finite set of points P in $\mathbb{R}^d$ belonging to its boundary. No scalar function is given in this case. Thus, the idea is to consider the distance function h which associates with each point in $\mathbb{R}^d$ its distance value from the nearest point of P. Even if, in general, h is not a smooth function, flow theory allows defining a vector field and a flow on $\mathbb{R}^d$ simulating the gradient and the integral lines of h, respectively. Critical points of the distance function h are the intersection points of the Voronoi diagram and its dual Delaunay diagram of the finite set of points P. The relation between the segmentation computed and Morse theory has been studied in [Ede03]. Analogously to the notion of descending manifold of a critical point in Morse theory, the *stable manifold $S(p)$* of a critical point p of h is defined as the set of all points of $\mathbb{R}^d$ that flow into p.

3.4. Classification of the algorithms

Algorithms for computing Morse decompositions can be classified based on different criteria, such as the *input* they accept, the *output* they generate, and the *algorithmic approach* they apply (for a more detailed discussion see [ČDMI14]).

Algorithms may differ in: (i) the dimension d of the scalar field (they may be specific for $d = 2$ or $d = 3$, or be dimension-independent); (ii) the underlying shape discretization (simplicial complex, cubical complex); and (iii) the required properties (e.g., no flat edges). Algorithms may differ in the information computed (ascending/descending Morse complex, or Morse-Smale complex), and in the output format (e.g., the cells of such complexes are described as collections of vertices or d-cells of the original complex).

Based on the algorithmic technique they apply, we have the following classification:

- *Boundary-based* approaches, which are based on piecewise linear Morse theory, and compute the Morse-Smale complex by constructing the separatrix lines (in 2D), or the separatrix lines and surfaces (in 3D).
- *Region-growing* approaches, which are generally based on piecewise linear Morse theory, and compute the ascending (or descending) Morse complex by growing the top cells, here called *regions*, from seeds located at the minima (or maxima). Note that the construction of the stable flow complex can be obtained through a region-growing approach.
- *Watershed* approaches, which are based on the discrete watershed transform, and compute the ascending Morse complex.

- *Forman-based* approaches, which are based on Forman's discrete Morse theory (see Sections 4, 5, 6).

3.5. Boundary-based algorithms

Boundary-based algorithms have been developed for triangle and tetrahedral meshes and for 2D and 3D regular grids. Such algorithms compute an approximation of the Morse-Smale complex through its 1-skeleton in 2D and through its 2-skeleton in 3D.

In the 2D case, a *boundary-based* algorithm generally performs two major steps:

1. *Vertex classification* to extract maxima, minima, and saddles.
2. *Path computation*: starting from each saddle, ascending separatrix lines are traced until they reach a maximum, and descending separatrix lines are traced until they reach a minimum.

In the smooth case, two ascending and two descending paths are incident in each saddle. In the discrete case, the number of ascending/descending paths in a saddle can be larger. A saddle of multiplicity equal to k has $k + 1$ ascending and $k + 1$ descending paths ($k + 1 = 2$ for simple saddles). Thus, the boundary-based algorithms unfold multiple saddles into simple saddles.

Path tracing should follow the steepest ascent/descent of the function. In the discrete case, this latter may be approximated in different ways. Takashahi et al. [TIKU95] move from the current vertex to its highest/lowest adjacent vertex. Edelsbrunner et al. [EHZ01] consider the edge slopes, and move from the current vertex v to its higher/lower adjacent vertex connected to v through the steepest edge. Such algorithms, which compute separatrix lines by following the edges of the triangle mesh, do not guarantee that traced lines have the steepest ascent/descent. Bremer et al. [BEHP04] and Pascucci et al. [Pas04] consider the slope of both edges and triangles, and allow separatrix lines to cross triangles. Because tracing separatrix lines across triangles is computationally intensive, de Berg et al. [dBT11] present a hybrid approach which tries to balance computation time and precision.

The approach in [EHZ01] has been extended to 3D [EHNP03]. Given a tetrahedral mesh, the algorithm computes the 1- and 2-cells which bound the 3-cells in the Morse-Smale complex. The extracted complex has the correct combinatorial structure described by a quasi-Morse-Smale complex. Two sweeps are performed over the input mesh. The first sweep (by decreasing function value) computes the descending 1- and 2-cells, and the second sweep (by increasing function value) computes the ascending 1- and 2-cells.

Boundary-based algorithms on regular grids are based on the construction of interpolating functions within each top

cell. Two conflicting issues arise: the need of avoiding, or limiting, the creation of new critical points, and that of guaranteeing a certain degree of continuity on the whole domain.

Bajaj et al. [BPS98] use a globally C^1-differentiable Bernstein-Bezier bi-cubic function for two-dimensional regular grids, and a tri-cubic function for 3D regular models. They define an algorithm to reduce the number of newly introduced critical points.

Schneider and Wood [SW04, Sch05] propose two methods for 2D regular grids. The first one uses a bilinear C^0 interpolating function on each top cell, which, thus, cannot introduce additional minima or maxima, but may introduce saddles. The second method uses a bi-quadratic approximation on each top cell. No new critical point is introduced, but the resulting approximation is globally discontinuous, formed by local surface patches.

All algorithms trace four separatrix lines from each saddle point, which can follow grid edges, or go through 2-cells. Both function derivatives and separatrix lines are computed numerically in [BPS98]. The two algorithms in [SW04, Sch05] compute derivatives analytically. The algorithm in [Sch05] uses this information to trace the separatrix lines, while the algorithm in [SW04] uses a step-by-step numerical procedure.

3.6. Region-growing algorithms

Region-growing algorithms operate on triangle or tetrahedral meshes, and compute the top cells of the ascending (descending) Morse complex, called *regions*, expanding an initial seed, i.e., the minimum (or maximum) the top cell is associated with. They are mostly based on piecewise-linear Morse theory, but the class of region-growing algorithms includes also techniques to build a flow complex. Regions are collections of triangles (for 2D scalar fields) or tetrahedra (for 3D scalar fields) from the underlying mesh Σ: each triangle or tetrahedron $t \in \Sigma$ is labeled with the minimum (or maximum) associated with the region containing t. The only exception is the 3D algorithm by Gyulassy et al. [GNPH07]. We focus here on the computation of the descending Morse complex. The computation of the ascending Morse complex is completely dual.

In the two-dimensional case, a region-growing algorithm conceptually contains two major steps:

1. *Extraction of seed vertices*.
2. *Region-growing*: from each seed, the corresponding region is built by iteratively adding triangles which are edge-adjacent to it.

In the two algorithms by Danovaro et al. [DDM03b, DDM*03a], the two steps are interlaced: a seed is extracted and its region is immediately grown. Each new seed p is the vertex with maximum function value among the ones with unclassified incident triangles. The region γ of p is initialized with unclassified triangles incident in p and grown by adding edge-adjacent triangles. The algorithm in [DDM03b] decides on adding a triangle t based just on function values, whereas the one in [DDM*03a] uses a discrete gradient defined over triangles. As extracted seeds include other vertices besides the maxima of f, a final merging of regions is required. The algorithm in [DDM03b] works for tetrahedral meshes as well, and can be extended to higher-dimensional scalar fields.

The algorithm in [MDD*09] first extracts all seeds, which are the maxima of function f. Then, their regions are computed one at a time in any order. Similarly to [DDM03b], the criterion for adding a triangle t to γ is based just on function values, but it is less restrictive and permits to build the entire region of a maximum. This algorithm accepts triangulated surfaces with flat edges, by using ad-hoc solutions for handling plateaus.

The algorithm in [GNPH07] computes all the i-cells of the 3D Morse-Smale complex (for all dimensions $i = 0, 1, 2, 3$) as collections of vertices, i.e., through vertex labeling. First, minima are found and an ascending 3-cell is grown from each of them. Then, ascending 2-cells are built starting from boundary vertices of 3-cells. Finally, ascending 1-cells are built starting from boundary vertices of 2-cells. Descending 3-, 2-, and 1-cells (in this order) are computed inside the ascending 3-cells using the same approach in a symmetric way. This algorithm has been presented for 3D scalar fields, but it can be extended to higher dimensions.

The computation of the flow complex is described in [GJ08] as a region-growing process. In [DGG03] an approximation of the flow complex is computed with lower computational costs. This latter is based on the construction of the Voronoi diagram V_P, and of its dual Delaunay diagram D_P, of the given set of points P. The intersection of V_P and D_P identifies the set of critical points corresponding to the cells of the stable flow complex. For a maximum, corresponding to a simplex σ_D in the Delaunay diagram (triangle in 2D and tetrahedron in 3D), the corresponding region is initialized with σ_D and grown by adding simplices which are face-adjacent to the boundary of the region. A simplex σ outside the region F is added into F if it is adjacent to F and σ flows into an internal simplex σ_1 (i.e. $\sigma < \sigma_1$). In [DGG03], two *flow relations* are defined specifically for the 2D and 3D cases. Given two edge-adjacent triangles σ_1 and σ_2, $\sigma_1 < \sigma_2$ if σ_2 and its dual Voronoi vertex lie on the opposite sides of the supporting line of the common edge. The same relation is provided in 3D between two face-adjacent tetrahedra. A tetrahedron σ_1 flows into an adjacent tetrahedron σ_2 if σ_1 and its dual Voronoi vertex lie on the opposite sides of the plane of the common face. Since in $\mathbb{R}^3$ flows can bifurcate creating ambiguity, in [DGG03] a *strengthened* flow relation is defined to determine where a Delaunay tetrahedron flows.

3.7. Watershed algorithms

As described in Section 3.2, given a labeled graph $H = (N_H, A_H, f)$, watershed algorithms produce the ascending Morse complex as a classification of the nodes of H: each node p is labeled with the minimum corresponding to the ascending top cell containing p. Some methods produce so-called *watershed* nodes, which lie on the boundary between two or more top cells. Watershed algorithms work on cubical complexes of any dimension or on simplicial meshes. In the first case, they label the top cells of the complex. In the latter case, the nodes in N_H are the vertices of the underlying mesh Σ, the arcs in A_H are the edges of Σ, and the output is a vertex classification.

Watershed algorithms are usually based on *topographic distance* [Mey94], on *simulated immersion* [VS91, Soi04], or on *rain falling simulation* [MW99, SS00].

Watershed algorithms based on topographic distance directly apply the definition of catchment basin in terms of topographic distance and shortest paths (see Section 3.2). The *image integration* algorithm by Meyer [MB90, Mey94] is a variation of the Dijkstra-Moore algorithm [Dij59] for computing the shortest path from a source node to every other node in a graph. In this case, the topographic distance is used. The *hill climbing* algorithm [Mey94] is a simplified version of image integration, which applies to regular grids, since the distance between two adjacent top cells p and q in domain space is constant.

The intuitive idea behind the *simulated immersion* approach [VS91, Soi04] is that of letting water raise from local minima, and label as watershed those nodes of graph H where water coming from different minima merge. The algorithm expands catchment basins by processing the nodes of H by increasing function value. In the stage in which a certain function value h is processed, all catchment basins of minima with value $h' < h$ have been started and, up to now, contain just nodes with function values lower than h. Processing function value h will add new nodes to existing basins, and will start new basins from minima having a function value equal to h.

Watershed approaches presented so far have in common the idea of growing catchment basins upwards from the minima of f. The *rain falling* paradigm [MW99, SS00] uses the opposite idea of letting water fall down from each vertex until it reaches at a minimum. The advantage of the rain-falling approach is that it does not require a preliminary sorting of the vertices, nor a priority queue.

The main steps of rain falling are:

1. Find the minima and label each minimum as belonging to the basin of itself.
2. For each node v, which is still unlabeled, start descending from v to its lowest neighbor u, continue until an already labeled vertex u is found, and give the label of u to all traversed nodes.

The algorithm by Mangan and Whitaker [MW99] is for triangle meshes, and the one by Stoev and Strasser [SS00] for regular grids. An implementation of the rain-falling simulation for triangle meshes has been used in [MDI13], where triangles are then classified based on the labels of their vertices. A similar approach can also be applied for tetrahedral meshes [Iur14].

3.8. Analysis and comparison

Some algorithms applied to meshes consider both the field difference between two vertices, and their distance in domain space [EHZ01, BEHP04, Mey94]. Other algorithms simply consider field difference. The two boundary-based algorithms in [TIKU95, EHZ01], and the two region-growing ones in [DDM03b, DDM*03a] just differ in this aspect. Note that the two approaches are equivalent on a regular grid.

Region-growing and watershed methods compute the top cells of the Morse complex through a classification of the top cells of the input simplicial or cubical complex, and, thus, their accuracy cannot go beyond the granularity of the input scalar field model. Boundary based algorithms exist which do the same, or trace separatrix lines also inside top cells, thus splitting an input triangle or pixel across several output top cells [BEHP03, Pas04]. These latter algorithms are less efficient, but avoid a number of violations of the properties of a *QMS* complex, such as a path from a saddle reaching another saddle before arriving at its final maximum or minimum, two overlapping paths in opposite directions, and regions bounded by extracted lines having a disconnected interior.

Boundary-based algorithms guarantee that saddles lie on the 1-cells of the output complex. As two paths going in the same direction join before reaching their final maximum or minimum, 2-cells bounded by such lines may be non-manifold because of so-called *dangling edges*. Region-growing methods suffer of the opposite problem. They build 2-cells that are 2-manifold by construction, but saddle vertices may not lie on their boundary.

Region-growing and watershed approaches do not extract the Morse-Smale complex, which can be obtained by intersecting the top cells of the Morse complexes. On the contrary, boundary-based approaches can easily produce the ascending or descending Morse complex by simply tracing just separatrix lines in one direction.

All algorithms based on locally steepest descent, including algorithms for computing the Forman gradient (see Section 5), do not converge to the ground truth smooth function when the underlying discrete domain is refined. Intuitively, the gradient of the input data is affected by a sampling error (due to the data points) and by a quantization error (due to the limited directions of the edges incident in each point). Although the sampling error can be decreased using a denser sampling, this is not the case for the quantization error. For

watershed based algorithms, this problem has been studied in [SCP08] where an improved technique is proposed based on a probability propagation scheme. New techniques has also been developed for algorithms computing a Forman gradient [GBP12, RGH*12].

4. Discrete Morse theory

Forman theory [For98b] is a discrete counterpart of Morse theory, with the main purpose of transposing the results of Morse theory from a smooth to a combinatorial setting. This goal is achieved by considering a function (also called a *Forman function*) defined over all the cells of a cell complex Γ. A discrete function F, defined on all the cells of Γ, is called a *Forman function* if, for any k-cell σ, all the $(k-1)$-faces of σ have a lower F value than σ, and all the $(k+1)$-faces have a higher F value than σ, with at most one exception. A cell is *critical* if there is no exception. If σ is a critical cell, then the index of σ is the same as its dimension. In particular, minima arise at vertices and maxima arise at d-dimensional cells [For98b]. We observe that, unlike the smooth case, if F is a Forman function on Γ, then $-F$ is not necessarily a Forman function on the same complex.

Figure 4 (a) shows a Forman function F defined on a simplicial complex. Each simplex is labeled by the value of function F. Vertex 1 is critical (minimum), since F has a higher value on all edges incident to it. Triangle 9 is critical (maximum), since F has a lower value on all edges incident to it. Edge 8 is critical (saddle), since F has a higher value on the incident triangle 9, and lower values on its extreme vertices.

The unique exception to the above rule, which holds for a non-critical cell, permits to pair such cell with either one of its faces, or one of its cofaces. Thus, a Forman function F induces a collection of pairs (σ, τ), where σ is a k-cell and τ is a $(k+1)$-cell, coface of σ. Such pair can be depicted as an arrow going from σ (tail) to τ (head). Each cell is a head or a tail of at most one arrow, and critical cells are those cells that are neither the head nor the tail of any arrow. A *discrete vector field V* on a cell complex Γ is a collection of pairs (σ, τ), each cell of Γ is in at most one pair of V.

A *V-path* is a sequence $\sigma_1, \tau_1, \sigma_2, \tau_2, ..., \sigma_r, \tau_r$ of k-cells σ_i and $(k+1)$-cells τ_i, $i=1,..,r$ with $r \geq 1$, such that $(\sigma_i, \tau_i) \in V$, σ_{i+1} is a face of τ_i, and $\sigma_i \neq \sigma_{i+1}$. A V-path with $r > 1$ is *closed* if σ_1 is a face of τ_r different from σ_{r-1}. A discrete vector field V is called a *discrete (Forman) gradient vector field* (or *Forman gradient*) if and only if there are no closed V-paths in V. A *critical cell* of V of index k is a k-cell γ which does not appear in any pair of V.

There is a correspondence between Forman functions and Forman gradient vector fields [For98a]. Namely, for each Forman function F, a Forman gradient vector field V_F can be constructed. Figure 4 (b) shows the Forman gradient vector field V_F corresponding to the Forman function F in Figure
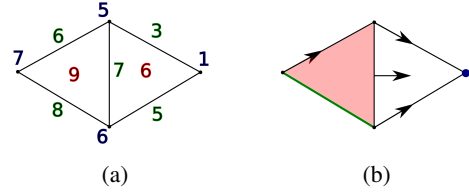
Figure 4: *(a) A Forman function on a simplicial complex and (b) the corresponding Forman gradient vector field.*

4 (a). Conversely, for each Forman gradient vector field V there exists a (non-unique) Forman function F such that the gradient field of F is V [For98b].

In the combinatorial setup of Forman theory, maximal V-paths correspond to the separatrix lines of a Morse function f in the smooth case, directed downwards. We call a *separatrix V_k-path*, connecting critical $(k+1)$-cell τ to critical k-cell σ, any V-path $\sigma_1, \tau_1, \sigma_2, \tau_2, ..., \sigma_r, \tau_r$ such that σ_1 is a face of τ and σ is a face of τ_r. In Figure 4, the separatrix V_1-paths are (*8,7,6,5,3,1*) and (*8,6,5,1*).

Discrete Morse theory has been used as a tool in shape data analysis and understanding, mainly shape segmentation, and homology and persistence homology computation. In the first application, discrete Morse theory becomes the basis for an efficient and a derivative-free computation of a segmentation of a discretized shape endowed with a scalar field through Morse or Morse-Smale complexes, and, thus, it is a efficient alternative to previous approaches based on piecewise linear Morse theory, or on the discrete watershed transform.

Considering homology computation, in [For98b] it has been proven that a cell complex Γ and any discrete Morse complex $\mathcal{M}_*$ defined by a Forman gradient on Γ, provide the same homological information, i.e., $H_k(\Gamma) \cong H_k(\mathcal{M}_*)$. The above equivalence allows for the computation of the homology groups and of the homology generators of a cell complex Γ through the Smith Normal Form reduction of the boundary maps of the discrete Morse complex. In many practical situations, the discrete Morse complex $\mathcal{M}_*$ is much smaller than the original complex Γ, and so, this approach significantly reduces the time to compute the homology groups of a cell complex. Also in persistent homology computation, it is possible to obtain the persistence of the input cell complex by studying the persistence of a considerably smaller discrete Morse complex.

4.1. Algorithms rooted in discrete Morse theory

Algorithms based on discrete Morse theory do not compute Morse, or Morse-Smale complexes explicitly, but compute a Forman gradient V from which all the cells of the Morse and Morse-Smale complexes can be extracted if needed. Such algorithms are purely combinatorial, dimension-independent

and independent of the type of underlying cell complex. Note that algorithms based on piecewise linear Morse theory are dimension-specific, and specific for simplicial meshes or regular grids. Furthermore, algorithms based on discrete Morse theory can provide all the cells of both the Morse and the Morse-Smale complexes, whereas watershed algorithms compute only the Morse cells.

When working with a Forman gradient, all the Morse and Morse-Smale cells are obtained by traversing the V-paths of the gradient in a dimension-independent way. Some algorithms [RWS11, SMN12, SN12, WIFD13] have been defined for computing a Forman gradient on real world datasets and then are easily parallelizable or have been specifically developed for distributed computation. We can classify algorithms for Forman gradient computation into two categories: *constrained algorithms* [CCL03, KKM05, GBHP11, RWS11, GBP12, GRWH12, SMN12, SN12]; and *unconstrained algorithms* [LLT03, MB09, MW10, DKMW11, HMMN14, BL14, FID14].

Constrained algorithms start from a discrete scalar function f defined over the vertices of a cell complex Γ, and aim at constructing a Forman gradient that best fits function f. They focus on extracting a minimum number of critical simplices in order to avoid spurious cells in the Morse complexes [RWS11, GBP12], or they perform a-posteriori simplifications to reduce them [KKM05, GBHP11]. The typical applications for constrained algorithms are data analysis and visualization, since a Forman gradient provides a computationally efficient way for extracting the Morse and Morse-Smale cells representing the regions of influence of the critical points.

Unconstrained algorithms, on the other hand, compute a Forman gradient on a cell complex when no information about the Forman function is provided. Here, a Forman gradient is extracted with the aim of computing homology and persistent homology of the original complex. The focus of these algorithms is computing a pairing for all the cells of the input cell complex while leaving as few unpaired cells as possible. In [LLT04], Lewiner et al. presented the first algorithm of this type, and showed applications to topology visualization and mesh compression. We provide a survey of constrained and unconstrained algorithms in Sections 5.2 and 5.3, respectively.

A challenging problem, when working with a Forman gradient V, is traversing the V-paths in order to extract information about the Morse or Morse-Smale complexes. In the case of constrained algorithms, traversing the V-paths corresponds to building a segmentation of the underlying shape into Morse or Morse-Smale complexes. In an unconstrained algorithm, V-paths are used for computing homology or homology generators of the underlying cell complex. In 2D, all V-paths can be visited in linear time by traversing all cell pairs at most once. However, for extracting only separatrix V-paths, the gradient paths between sad-

dles and maxima can be visited in reverse order, thus reducing the numbers of pairs to be visited. In higher dimensions the situation is more involved. We will describe in details the algorithms proposed for the efficient extraction of Morse and Morse-Smale complexes from a Forman gradient [GBP12, GRWH12, SN12, WIFD13] in Section 6.

Both constrained and unconstrained algorithms are involved, with a different flavor, in persistent homology computation. In constrained approaches, the values of function f at the vertices of cell complex Γ naturally induce a filtration of Γ. In [RWS11], for example, the generic element Γ_m of the filtration induced by the input function f on Γ is the cell complex containing all the cells of Γ that have no vertex with a function value greater than m. This choice guarantees that each cell of the discrete Morse complex corresponds to a change in topology between successive cell complexes of the filtration. In unconstrained approaches, the filtration is set as input, and the construction of the Forman gradient has to comply with the given filtration [MN13, DW14].

5. Computing the discrete Morse gradient

In this section, we present first an encoding for the discrete Morse gradient field for a simplicial complex and then we describe algorithms for computing it.

5.1. Forman gradient encoding on simplicial complexes

Since a Forman gradient field is a collection of pairs of cells on a cell complex Γ, we need a representation for Γ in which all cells are explicitly encoded as well as their mutual incidence relations, as in the *Incidence Graph (IG)* [Ede87]. This latter is the most common and general data structure for cell complexes, being an implementation of the Hasse diagram of the complex, where the nodes correspond to the cells of the complex and the arcs to the immediate boundary and co-boundary relations.The Forman gradient can be easily implemented on the IG as a Boolean function associated with its arcs. For a cubical complex, all such relations are encoded implicitly by indexing the cells of the grid. Moreover, since a Forman gradient V defines a pairing between incident cells, V can be defined as a bit vector based on the same indexing [GRWH12].

For simplicial complexes, a data structure encoding all simplices and their incidence relations is too verbose. Other data structures exist for simplicial complexes, which encode only vertices and top simplices [DH05, CDW11], thus being much more compact, and scalable with the dimension. An encoding for the Forman gradient on such data structures associates the gradient pairs to the top simplices, and is called a *compact gradient*. The use of such data structures with the compact gradient makes the computation of the Forman gradient and of the Morse and Morse-Smale complexes feasible on simplicial complexes on large size [WIFD13, FID14, FIDW14].

Let us consider a d-dimensional simplicial complex Σ and a d-simplex σ in Σ. The compact encoding associates with σ a subset of the pairs involving its faces, namely, all pairs in the discrete vector involving σ or two of its boundary simplices. In 2D, a triangle has 12 possible pairs for a total of $2^{12} = 4,096$ cases. However, for a Forman gradient, there are only 97 valid cases for a triangle. Thus, all possible configurations can be encoded by using only one byte per triangle. Similarly, in 3D, there are 32 pairs for a total of $2^{32} = 4,294,967,296$ possible configurations, and the valid ones are only 51,030, thus they can be represented with 2 bytes per tetrahedron. We refer to [WIFD13] for details.

5.2. Constrained algorithms

Several algorithms have recently been developed in the literature to compute the Forman gradient when a scalar value is given at the vertices of a cell complex. Such algorithms are general, but their implementation is usually restricted to the 2D and 3D cases, and mainly to cubical complexes.

The algorithm proposed in [CCL03] builds a Forman gradient from a triangle mesh Σ endowed with a scalar function, which in this case is the discrete Connolly's function f [Con86]. Function values are extended to edges and triangles by considering the mean value of their incident vertices. They consider the *primal graph* G_P of Σ, which is the graph having as nodes the vertices of Σ and as arcs its edges, and the *dual graph* G_D of Σ defined as the graph having as nodes the triangles of Σ and the arcs in one-to-one correspondence with edge-adjacent triangles. A spanning forest T_D is created by building a spanning tree on G_D for each local maximum of f on Σ, processing the edges by increasing function value. In a dual fashion, a spanning forest T is built on G_P by creating a spanning tree for each local minimum. The Forman gradient V is computed from T and T_D. The minima (vertices) are the roots of T, the maxima (triangles) are the roots of T_D, and the saddles are the edges belonging neither to T nor to T_D. The V-paths of V formed by vertices and edges correspond to paths from a root of T to one of its leaves, while V-paths formed by edges and triangles correspond to the paths from a root of T_D to one of its leaves. The algorithm can be extended to d-dimensional complexes but only by restricting to the computation of the pairings between 0-simplices and 1-simplices (forming V-paths connecting minima to 1-saddles) and between $(d-1)$-simplices and d-simplices (forming V-paths connecting maxima to $(d-1)$-saddles).

The algorithm proposed in [KKM05] takes as input a scalar function f defined over the vertices of a three-dimensional simplicial complex Σ. The algorithm builds the gradient vector field in the lower link $Lk^-(v)$ of each vertex v in Σ, where the *lower link $Lk^-(\gamma)$* of a simplex γ is the subset of the link of γ (see Sect. 2.1) containing only simplices with a lower function value than γ. Then, it extends this function to the cone $(v; Lk^-(v))$, which is the simplex

generated by the union of the vertices of v and $Lk^-(v)$. The Forman gradient computed in this way may have an arbitrary large number of critical simplices compared with the number of the critical points of the original scalar function f. Thus, the algorithm performs a simplification step for reducing the number of critical cells [For98b].

The algorithm proposed by Gyulassy et al. in [GBHP08] is one of the first algorithms defined in a dimension-independent way. It computes a Forman gradient starting from a d-dimensional regular grid Γ with scalar function f defined at the vertices of Γ. Function f is extended to a Forman function F, defined on all cells of Γ, such that $F(\tau)$ is slightly larger than $F(\sigma)$ for each cell τ and each face σ of τ. For such function F, all cells of Γ are critical. A gradient vector field is computed by assigning gradient arrows in a greedy manner during sweeps over the cells of Γ according to increasing values of dimension and of F. Each current non-paired and non-critical cell in the sweep is paired with its coface having only one non-marked face (as critical or as already paired). If there are several of such cofaces, the lowest one is taken. If there is no such coface, the cell is critical. Pairs built in this way define a gradient vector field. The order in which the cells in Γ are processed is not deterministic, since different k-cells in Γ may have the same value of function F. As a consequence, some unnecessary critical cells may be produced. In [SMN12] and [SN12], a similar approach based on a *weighted discrete function* has been defined for computing a Forman gradient on 2D and 3D regular grids. The pairs found by the algorithm are unique and independent of the order in which the cells are considered, thus providing a basis for a parallelization. In [GBP12], a similar algorithm is proposed which focuses on improving the poor geometric approximation of the gradient caused by the local assignment of the gradient arrows. This is especially useful in scalar field analysis, but not for homology computation.

In [RWS11], a dimension-independent algorithm is proposed for a regular grid with scalar function values given at its vertices. In [WIFD13, FIDW14], this algorithm has been extended to simplicial complexes in 2D and in 3D by developing very compact representations for the underlying complex, leading to the first efficient algorithm for Forman gradient computation on simplicial complexes. The algorithm processes the lower star of each vertex v in Σ independently, where the *lower star* of a cell γ is the subset of the star of γ (see Sect. 2.1) containing only cells with a lower function value than γ. Each cell σ is considered in ascending order of function values and of dimension in such a way that each cell σ is considered after its faces. All the k-simplices incident in the *lower star* are paired via homotopy expansion. Two cells, k-cell σ and $(k+1)$-cell τ, are paired via homotopy expansion when σ has no unpaired boundary cells and τ has only one unpaired boundary cell (i.e. σ). In [RWS11] it has been proved that in the 3D case, similarly to the smooth Morse setting, the critical cells identified are in one-to-one correspondence with the topological changes in the sub-level sets

of the function defined over the regular grid. This aspect, in addition to the easy implementation and parallelization, candidates the algorithm to be one of the best, topologically correct algorithms for computing a Forman gradient.

5.3. Unconstrained algorithms

Several algorithms have been proposed for computing a Forman gradient on a cell complex without any constraint, such as the one given by a scalar value at the vertices.

5.3.1. The algorithm by Lewiner et al. [LLT03]

The algorithm in [LLT03] has been the first algorithm of this kind presented in the literature, with the purpose of providing a combinatorial characterization of a shape. The algorithm has been defined for manifold 2-complexes first and then extended for general cell 2-complexes. The algorithm works independently on each connected component of a cell 2-complex Γ, performing three steps. In the first step, a spanning tree T is built on the dual graph of Γ. Recall that the dual graph of Γ is the graph in which the nodes correspond to the top d-cells of Γ and the arcs connect $(d-1)$-adjacent top cells. In the second step, a discrete Morse function is defined on the nodes and edges of T. Every node of T gets as function value equal to its distance from the root plus $(v+1)$ (with v number of 0-cells in Γ), and every edge of T gets the minimum value of its two end nodes. In the last step, the primal graph defined by the 1-skeleton of the complex is considered. A spanning tree U is built on such a graph. Every node on U gets as value its distance from the root and, every edge, the maximum value of its incident nodes. Recall that 0-cells of Γ correspond to nodes of U, 1-cells correspond to arcs in both U and T, and 2-cells correspond to nodes in T. Finally, discrete Morse function values are assigned to cells of Γ from the corresponding values in the two graphs.

5.3.2. Algorithms based on reduction and coreduction

Several algorithms have been proposed based on two simplification operators, called *reduction* and *coreduction*, deleting a pair of cells from a cell complex Γ while preserving the homology groups of Γ. Even if these two operators have been defined as simplification operators, they can be seen as pairing operators for building a Forman gradient V on Γ, i.e., once Γ has been fully simplified through *reduction* and *coreduction* operators obtaining the simplified cell complex Γ', arrows of V correspond to the cell pairs eliminated and the cells in Γ' correspond to the critical cells of V. Here, we will describe two of these algorithms presenting a dual strategy for computing the Forman gradient [HMM*10, BL14].

Algorithm by Benedetti et al. This algorithm [BL14] uses the reduction operator in the construction of a Forman gradient. A reduction corresponds to a deformation retraction of a cell which is a face of only one other cell in the complex. For this reason, it enables reducing the size of a cell complex

without affecting its homology. Let Γ be a cell complex and let σ be a k-cell of Γ. We call *(immediate) coboundary* of σ with respect to Γ the set $\mathrm{cbd}_\Gamma \sigma$ of the $(k+1)$-dimensional cofaces of σ. A pair (τ, σ) of cells of Γ, such that σ is incident only once in τ, is called a *reduction pair* if $\mathrm{cbd}_\Gamma \tau = \{\sigma\}$. The algorithm builds a Forman gradient vector field V on a cell complex Γ by using reduction pairs and removals of top cells. The algorithm first initializes V as the null set. Let us denote as Γ' the set of non-excised cells of Γ, initially equal to Γ. While Γ' admits a reduction pair (τ, σ), the algorithm excises cells τ and σ from Γ' and adds the pair (τ, σ) to set V. When no more reduction is feasible, a top cell is excised from Γ', which becomes a critical cell, and the main loop of the algorithm is iterated until Γ' is empty.

Algorithm by Harker et al. This algorithm [HMM*10, HMMN14] uses coreduction as a homology-preserving operator to construct a Forman gradient on a cell complex. The *coreduction operator* (see [MB09]) can be viewed as dual with respect to a reduction and the two operators combined represent a powerful preprocessing tool to efficiently compute homology of a cell complex, as described in [MB09, MW10, DKMW11]. Let Γ be a cell complex and let σ be a k-cell of Γ. We call *(immediate) boundary* of σ with respect to Γ the set $\mathrm{bd}_\Gamma \sigma$ of the $(k-1)$-dimensional faces of σ. Let Γ be a cell complex and, given σ, τ two cells of Γ, a pair (τ, σ) of elements of Γ, such that σ is incident only once in τ, is called a *coreduction pair* if $\mathrm{bd}_\Gamma \sigma = \{\tau\}$.

In [HMMN14] a Forman gradient is built by removing coreduction pairs and free cells, where a *free cell* is a cell with an empty boundary. In this approach, a Forman gradient vector field V on Γ is built as follow. The algorithm looks for available coreduction pairs in the complex. While a coreduction pair is feasible, the pair is excised from the cell complex and it is added to V. When no more reduction pair is available, a free cell is excised from the complex and considered as a critical cell. The algorithm iterates this process until no more cells can be considered.

Comparison The two approaches described above are dual to each other. The equivalence in the use of reductions or coreductions has been proven in [FID14] in the context of simplicial complexes. Any Forman gradient obtained through a sequence of reductions and removals of top cells can be obtained through a suitable sequence of coreductions and removals of free cells, and vice versa. In [FID14] an algorithm has been proposed to build a Forman gradient vector field by executing reduction and coreduction pairs in an interleaved way. It has been shown that any interleaved approach still produces a Forman gradient and that such a gradient can be obtained through reduction or coreduction pairings.

Both from an application and a theoretical point of view, it is interesting to find a method to build a Forman gradient vector field which minimizes the number of critical cells.
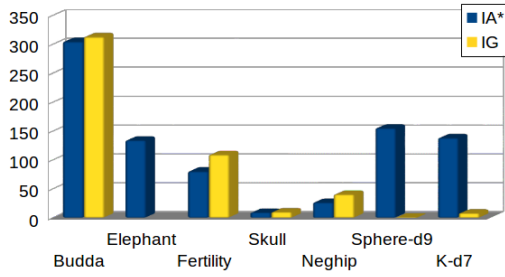
Figure 5: *Timings (in seconds) for homology computation using the IA* and the IG data structures. For dataset Elephant, the IG implementation runs out of memory.*

Computation of a Forman gradient is known to be NP-hard. In [BLPS13], problems related to discrete Morse theory are analyzed in terms of parametrized complexity determining their hardness using the mathematically rigid framework of the W-hierarchy. The results in [FID14] show that the use of different simplification operators or the combination of several operators does not affect the number of resulting critical cells. As discussed in [FID14], when using a verbose data structure for encoding a cell complex, storing all the boundary/coboundary relations like in [MN13], an approach involving both reductions and coreductions is more efficient from a computational point of view. When only a subset of these relations is explicitly stored in the data structure encoding the cell complex, the storage cost decreases considerably. In this case, an approach involving only reductions or coreductions has to be selected, depending on whether boundary or coboundary relations can be retrieved faster from the specific data structure used to encode the complex. In [FID14], the trade-off between a verbose data structure (the incidence graph *IG*) and a compact data structure encoding only the vertices and top simplices of a simplicial complex (the *IA** data structure [CDW11]) has been analyzed. The *IG*-based implementation is about 3 times more costly for 2-complexes. For-complexes, the ratio between the storage costs of the *IG* and the *IA** implementations is equal to 4 and the same ratio becomes 17 and 24 for complexes of dimension 7 and 9, respectively. On the other hand, the timings of the two data structures are comparable for 2- and 3-complexes, while an implementation based on the *IA** data structure becomes slower when working in higher dimensions (see Figure 5).

6. Morse complexes from the Forman gradient

In this section, we discuss algorithms for retrieving Morse and Morse-Smale complexes, which are implicitly represented by the Forman gradient. We will discuss the algorithms for simplicial complexes, even if they are valid also for cell complexes.

Note that when we use Morse decompositions as a segmentation of a shape endowed with a scalar field, we are interested in a geometrical representation for the cells of the Morse or Morse-Smale complexes. In this case, the domain of the complex is always a manifold. In 3D, the 1-skeleton of the Morse-Smale complex has been shown to be very useful for understanding the structure of the dataset [GKK*12]. For homology and persistent homology computation, instead, we need to compute the boundary maps (incidence relations) between the cells of the descending Morse complex. Computing the Morse Incidence Graph (*MIG*) (see its definition in Section 2.3) is crucial for retrieving such information efficiently.

6.1. Computing Morse decompositions

A descending [ascending] k-cell associated with a critical simplex σ is extracted by starting from σ and traversing the simplicial complex Σ along the pairs of the gradient field. We present here the general strategy and afterwards improvements proposed by several authors.

Descending Morse complex The k-cells of the descending Morse complex Γ_D are naturally defined as a collection of k-simplices of Σ. The computation of a descending k-cell always starts from a critical k-simplex σ. All the $(k-1)$-simplices in the immediate boundary of σ are selected and, among them, only the $(k-1)$-simplices paired with a k-simplex different from σ are considered. Such k-simplices are inserted into a queue, and the traversal of the simplicial complex Σ continues in a breadth-first fashion until all the V-paths starting from σ have been visited. In 2D, we start from a critical triangle (maximum) σ and, by following gradient pairs, we continue adding edge-adjacent triangles in a breadth-first fashion until all V-paths from σ have been traversed.

Ascending Morse complex The k-cells of the ascending Morse complex Γ_A are defined as a collection of $(d-k)$-simplices of Σ. The computation of an ascending k-cell starts from a critical $(d-k)$-simplex σ of Σ. All the $(d-k+1)$-simplices in its immediate coboundary are selected and, among them, only the $(d-k+1)$-simplices paired with a $(d-k)$-simplex different from σ are considered. From such $(d-k)$-simplices the breadth-first traversal of the complex continues until all the V-paths ending in σ have been visited in reverse order. In 2D, the computation starts from a critical vertex (minimum) and navigates the edges in a dual manner with respect to what is done for the descending case. We can observe that the duality described in the algorithm by Cazals et al. [CCL03] when computing a Forman function is also revived when following the Forman gradient. This underlines the relation between a Forman function and the corresponding Forman gradient (see Section 4).

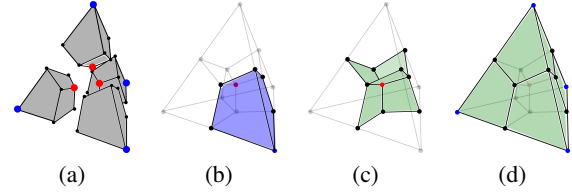The computation of the ascending/descending Morse
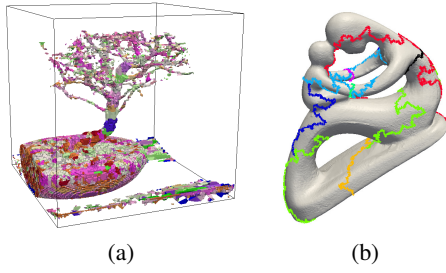
Figure 6: *(a) Descending 3-cells computed on the Bonsai dataset and (b) descending 1-cells computed on the fertility mesh.*

Figure 7: *(a) In the dually subdivided mesh, a tetrahedron (represented by the red dot) is split into four hexahedra. (b) Each hexahedron is uniquely identified by the red dot (tetrahedron) and a blue dot (vertex). (c) The six quads created by adjacent hexahedra in the interior of a tetrahedron. (b) The twelve quads, created by adjacent hexahedra, on the boundary of two face-adjacent tetrahedra.*

complex is performed through constant-time operations at each cell on the V-paths visited. In 2D, the extraction of a descending/ascending k-cell takes time linear in the number of simplices forming the k-cell, since each simplex is visited at most once. In higher dimensions, the situation is more involved. For instance, in 3D, gradient paths may branch and merge, potentially resulting in many-to-many adjacency relationships between critical 1-cells and critical 2-cells. For example, the discrete Morse function of a tetrahedral mesh Σ with N vertices, contains $\mathcal{O}(N)$ critical 1-cells, each of which is connected to $\mathcal{O}(N)$ critical 2-cells. This produces a discrete Morse complex containing $\mathcal{O}(N^2)$ gradient paths between critical 1 and 2-simplices. Since the number of critical 1 and 2-simplices is bounded by N, the number of traversals for any cell during the breadth first search is also bounded by N, and thus the complexity of the whole extraction process is $\mathcal{O}(N^3)$.

In Figure 6, two examples of Morse cells are shown. In Figure 6(a), the descending 3-cells are computed and, based on the number of tetrahedra inside each cell, larger regions are excluded from the visualization, thus highlighting the inner structure of a volumetric dataset. In Figure 6(b), the 1-cells of the descending Morse complexes are computed for retrieving the homology generators of degree one, where each descending 1-cell corresponds to a 1-cycle which is independent of the others.

Morse-Smale complex The Morse-Smale (*MS*) complex is obtained by intersecting the ascending and descending Morse complexes. The *MS k*-cells are obtained by intersecting (implicitly) the k-simplices of complex Σ and the k-cells of the *dual complex* Σ_D. Σ_D is a d-dimensional cell complex, in which the 0-cells correspond to the d-simplices of Σ, the 1-cells correspond to the $(d-1)$-simplices of Σ, and, in general, the k-cells correspond to the $(d-k)$-simplices of Σ. The maximal cells of Γ_D correspond to the vertices of Σ.

A top cell of the Morse-Smale complex corresponds to a pair of critical points (a maximum and a minimum) and is encoded, as described in [WIFD13], as a collection of top cells in the dually subdivided mesh Σ_S. The *dually subdi-*

vided mesh is obtained by intersecting the simplices of the primal mesh Σ (i.e., vertices, edges, triangles, etc.) with the cells of the dual complex Σ_D (i.e., 0-cells, 1-cells, 2-cells, etc.).

Let us consider the 3D case. A top cell of the Morse-Smale complex corresponds to a pair of critical points (a maximum and a minimum) and is encoded as a collection of hexahedra in the dually subdivided mesh Σ_S obtained by intersecting the descending 3-cell (corresponding to the maximum) and the ascending 3-cell (corresponding to the minimum), which are collections of tetrahedra and vertices, respectively (see Figure 7(b)). Moreover, in the 3D case the 3-cells of the Morse-Smale complex are bounded by a set of 2-cells corresponding to pairs of saddles (1-saddle and 2-saddle) and composed of a collection of quads. Each quad is created by face-adjacent hexahedra which could be inside the same tetrahedron (Figure 7(c)) or on the boundary of two face-adjacent tetrahedra (Figure 7(d)). Each quad belongs to a Morse-Smale 2-cell if the two hexahedra belong to different Morse-Smale 3-cells.

The 1-skeleton of the Morse-Smale complex is composed of different sets of 1-cells. In the 2D case, the 1-cells corresponding to a maximum-saddle or a minimum-saddle are the 1-manifolds of the ascending and descending Morse complex, respectively. In the 3D case, the same sets of 1-cells is combined with a set of 1-cells, called *saddle-connectors*, which connect 1-saddles with 2-saddles. A saddle connector between a 1-saddle p and a 2-saddle q is computed by extracting the descending and ascending 2-manifolds associated with p and q. The descending 2-manifold extraction is performed first, and all the traversed triangles are marked as visited. Then, starting from the critical primal edge e corresponding to p and its adjacent edges, the same process as for extracting ascending 2-manifolds is performed, but only the triangles previously marked as visited are considered. In Figure 12(b) the 1-skeleton of the Morse-Smale complex, computed on the Buckyball dataset, is shown.

6.2. Computing the Morse Incidence Graph (*MIG*)

The incidence relations encompassed in the Morse incidence graph are computed as follows: the *V*-paths of the *compact gradient V* defined on Σ are traversed, all the Morse cells in one of the two Morse complexes are computed, one node for each critical simplex is saved and two nodes in the graph are connected with an arc if there is a separatrix *V*-path in *V* connecting the two corresponding critical simplices. Since only separatrix *V*-paths are needed, we can use ad-hoc strategies for reducing the number of simplices visited in practice.

Let us consider the 2D case. The set of separatrix *V*-paths between saddles and minima are visited starting from each critical edge and following the gradient paths until a minimum is reached. Such paths never branch and, thus, a limited number of edges are visited in practice during their traversal. Dually, the set of separatrix *V*-paths between saddles and maxima are visited starting from each edge following the gradient path in reverse order until a critical triangle is reached. Also in this case, such paths never branch.

In 3D, a new step is introduced to compute the saddle connectors, i.e., the arcs of the *MIG* between 1-saddles and 2-saddles. Saddle connectors are extracted, as discussed in Section 6.1, for extracting the 1-skeleton. All the descending 2-manifolds are extracted first, and all the traversed triangles are marked as visited. Then, starting from each critical edge *e* corresponding to a 1-saddle *p*, the same process as for extracting ascending 2-manifolds is performed, but only the triangles previously marked as visited are considered. In this way, only the separatrix V_2-paths are traversed and, for each of them, an arc connecting the two saddle nodes is created.

When computing homology, the *MIG* is fundamental for representing the boundary maps. Formally, a discrete Morse complex is a chain complex $\mathcal{M}_* := (\mathcal{M}_k, \tilde{\partial}_k)_{k \in \mathbb{Z}}$, where $\mathcal{M}_k$ is the $\mathbb{Z}_2$-vector space generated by the critical cells of dimension *k* of the cell complex Γ, and $\tilde{\partial}_k : \mathcal{M}_k \rightarrow \mathcal{M}_{k-1}$ is the boundary map which encodes the multiplicity of incidence *k*-cells and the $(k-1)$-cells of $\mathcal{M}_*$ such that $\tilde{\partial}^2 = 0$. The Forman gradient computation provides the critical cells of the discrete Morse complex, and thus groups $\mathcal{M}_k$. The computation of the boundary maps is strictly related to the extraction of the *MIG*. If we consider boundary maps with $\mathbb{Z}_2$ coefficients, the multiplicity of the incidences of two critical cells of consecutive dimension is just the number of arcs that connect the corresponding nodes in the *MIG*, considered modulo 2. For a more accurate information about incidence relations between critical cells, we have to consider the boundary maps with integer coefficients.

6.3. Improvements in three and higher dimensions

As described in Section 6.1, during the computation of Morse and Morse-Smale cells between 1- and 2-saddles in 3D, a single simplex may be visited multiple times, and this

results in a $\mathcal{O}(N^3)$ worst-case time complexity. In the literature, some solutions have been proposed to handle this problem [GRWH12, SN12, WIFD13].

In [WIFD13, GRWH12], the space complexity of the algorithm is slightly increased. During a gradient path traversal between a *k*-simplex and a $(k-1)$-simplex, a Boolean function is defined, for each *k*-simplex, indicating whether the simplex has already been visited. In this way, all the simplices are never visited twice and the worst-case time complexity drops to $\mathcal{O}(N^2)$. The same approach is adopted for computing the boundary maps in $\mathbb{Z}_2$ of the Morse complex in time $\mathcal{O}(cN^2)$, where *c* denotes the number of critical points. This method works well when the computation of the saddle connectors is sequential. In a parallel implementation, each thread would require a buffer for representing the Boolean function, whose size is equal to the number of *k*-simplices. Thus, this latter approach does not scale with the growth of the size and dimension of the data sets.

In [SN12], a method is proposed which is specifical to be used in parallel on a cell complex Γ, and an implementation restricted to cubical complexes in 3D is described. A priority queue *PQ* is defined based on the discrete Morse function associated with the cells of Γ. A cell σ is always inserted into *PQ* as a pair (σ, n_p), where n_p indicates the number of *V*-paths passing by cell σ. We consider the gradient *V*-paths starting at a critical cell σ. Then, the $(k-1)$-cells τ on the boundary of σ are inserted into *PQ* as pairs $(\tau, 1)$. The cells are then processed from *PQ* in decreasing order of function value. When a cell is dequeued, the new cells τ are visited following the *V*-paths, and new pairs $(\tau, 1)$ are enqueued in *PQ*. Due to the branching and merging of the *V*-paths, a cell τ may be inserted in *PQ* more than once. In such cases, however, since these cells have the same function value, they will occupy consecutive positions in *PQ*, and we can remove them from the queue all at once, updating n_p.

Newly inserted cells have a lower function value than τ. So, τ never re-enters *PQ*. A cell is inserted into *PQ* when processing one of its neighboring cells. So, the number of copies of the cell in *PQ* is bounded from above by the number of its neighbors. Each cell is inserted in *PQ* only a constant number of times. So, the complexity of the algorithm is $\mathcal{O}(n_{k,k-1} \log n_{k,k-1})$, where $n_{k,k-1}$ is the number of *k*-cell-$(k-1)$-cell pairs. The latter approach works in any dimension and is fundamental for extracting saddle connectors between a *k*-saddle and a $(k+1)$-saddle, with $k \neq \{0, d-1\}$.

7. Simplification of Morse and Morse-Smale complexes

Due to the huge size of available data sets and to the presence of noise, morphological descriptions of scalar fields may contain many uninteresting features. Thus, a fundamental issue is the simplification of such descriptions. *Cancellation* operator, defined in Morse theory for simplifying a Morse function (see Section 2.3), has been defined in the discrete case as well.
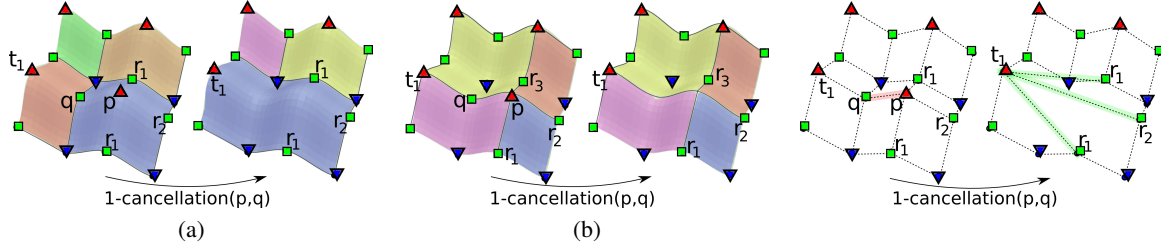
Figure 8: *Effect of a* 1*-cancellation*(p, q) *on the descending (a) and ascending (b) Morse complex and on the corresponding MIG (c).*

We can classify the simplification operators proposed in the literature based on whether they act on Morse or Morse-Smale complexes, represented explicitly through the Morse Incidence Graph (*MIG*) [ELZ02, BEHP04, GNP*06, ČomićD11] (reviewed in Section 7.1), or represented implicitly through a Forman gradient V [GSW12, GRSW13, FIDW14] (reviewed in Section 7.2).

In Section 7.3, we review and compare simplification algorithms defined for Morse and Morse-Smale complexes, which are based on the iterative application of simplification operators.

7.1. Simplifying the Morse Incidence Graph

A *k-cancellation* transforms the ascending and descending Morse complexes into complexes with fewer cells. In the descending Morse complex Γ_D, let p be a $(k+1)$-cell and q a k-cell of Γ_D, with $k = 0, \cdots, d-1$. A k-cancellation is performed on Γ_D provided that q is incident in p only once. The *k-cancellation*(p, q) removes cells p and q and changes the connectivity of the remaining cells of Γ_D. The effect of a k-cancellation on the ascending Morse complex Γ_A is entirely dual. Figure 8 shows an example of 1-*cancellation*(p, q). On Γ_D (see Figure 8(a)): the 2-cell p and the 1-cell q are removed, the 1-cells on the boundary of p (green squares) are moved to the boundary of the cell corresponding to the maximum and adjacent to p, and the 0-cells (blue triangles) in $b(q)$ are removed from the boundary of q. The effect on Γ_A (see Figure 8(b)) is dual: 0-cell p is collapsed into a 0-cell adjacent to p and 1-cell q is deleted as well. All the 1-cells in the coboundary of p are extended to p' and q is removed from the boundary of all 2-cells in its coboundary (blue triangles).

We consider the Morse Incidence Graph (*MIG*) G describing both Γ_D and Γ_A. A k-cancellation deletes two nodes from N, and locally modifies the connection between the nodes in their neighborhood. A k-node p and a $(k+1)$-node q can be cancelled if p is connected to q through a single arc in G. After *cancellation*, each k-node t, which was connected to $(k+1)$-node q, becomes connected to each $(k+1)$-node r which was connected to k-node p.

In 2D, a *cancellation* removes an extremum and a saddle. Let us consider the 1-*cancellation* of a maximum (2-node) p and a saddle (1-node) q, shown in Figure 8. After cancellation, q and p are deleted. In the descending Morse complex, shown in Figure 8(b), 1-cell q is deleted and 2-cell p is merged with the other 2-cell t_1 incident in q. In the *MIG* (see Figure 8(c)), nodes p and q are deleted as well as all their incident arcs. Nodes r_1, r_2 and r_3 are connected to t_1 as the corresponding 1-cells become part of the boundary of t_1. The 0-*cancellation* of a minimum and a saddle is dual.

In 3D, a 2-cancellation (maximum and 2-saddle) and a 0-cancellation (minimum and 1-saddle) are exactly the same as in the 2D case. A 1-*cancellation*, instead, may increase the number of cells in the Morse-Smale complex as a consequence of the increase in the incidence relations in the Morse complexes. Let p be a 1-node and q be a 2-node in the *MIG*. Let $R = \{r_i, j = 1, .., i_{max}\}$ be the set of 2-nodes connected to p and different from q, and let $T = \{t_j, k = j, .., j_{max}\}$ be the set of 1-nodes connected to q and different from p. 1-*cancellation*(p, q) on the *MIG* $G = (N, A, \varphi)$ removes both p and q from N and removes $|R| + |T| + 1$ arcs from A, while the number of arcs added to A is $|R| \cdot |T|$. Thus, the number of arcs in G (and the number of cells in the *MS* complex) may increase. In the *MIG* shown in Figure 9, nodes p and q are removed and the nodes r_1, r_2 and r_3, previously connected to p, become connected to nodes t_1, t_2 and t_3, previously connected to q. In the corresponding descending Morse complex 1-cell p and 2-cell q are removed and the 2-cells r_1, r_2 and r_3 are stretched to fill the space of q, with t_1, t_2 and t_3 becoming part of their boundary.
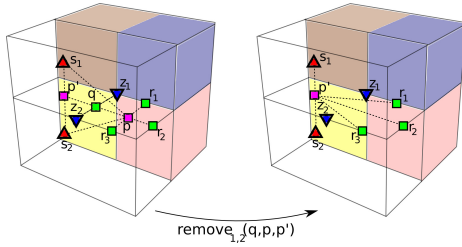
In [ČDI12], a dimension-independent simplification operator, called *k-remove*, has been defined for simplifying Morse and Morse-Smale complexes. On the *MIG*, *k-remove*(q, p, p') deletes a k-node q and a $(k+1)$-node p if and only if q is connected to p through a single arc, and to at most one $(k+1)$-node p' different from p. We denote as R the set of k-nodes r_i connected with p, Z the set of $(k-1)$-nodes z_j connected with q and S the set of $(k+1)$-nodes s_h connected with p. Operator *k-remove*(q, p, p') deletes all the arcs incident in either p or q. Each node r_i connected to p with an arc of the form (p, r_i), is redirected to p' with an arc
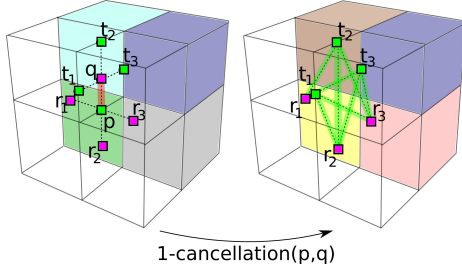
Figure 9: *1-cancellation of a 1-saddle p and a 2-saddle q, in 3D, on a MIG. Colored cells are the top cells in the descending Morse complex. Critical points are marked in green (1-saddles) and purple (2-saddles). Dotted lines are the arcs of the MIG.*

Figure 11: *1-cancellation(q, p, p') on the Forman gradient V of a triangle mesh. The original V has two critical triangles p and p' (in red) and one critical edge e (in green). V-paths outside the triangulation illustrated, are depicted with bold lines if critical and with dotted lines otherwise. Red arrows indicate the V-path involved in the simplification.*

(p', r_i). Thus, operator *remove* deletes $|R| + |S| + |Z| + 1$ arcs and insert at most $|R|$ arcs.
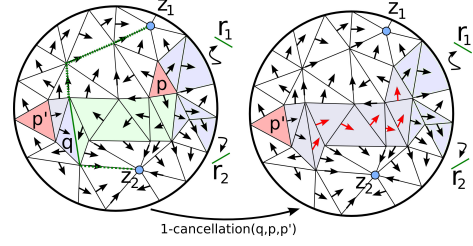

Figure 10: *1-remove(q, p, p') of a 2-saddle p and a 1-saddle q in 3D. Colored regions are the 3-cells of the descending Morse complex. Critical points are marked in blue (minima), green (1-saddles), purple (2-saddles) and red (maxima). Dotted lines are arcs of the Incidence Graph.*

When $k = \{0, d-1\}$, k-remove(q, p, p') is equivalent to a k-cancellation(q, p). When $1 < k < d-1$, k-remove(q, p, p') can be considered as a *cancellation* with stronger feasibility conditions. On the 3D *MIG*, for example, 1-remove(q, p, p') is feasible if 1-node q is shared by exactly two distinct 2-nodes p and p', and if q shares one arc with p. The operator removes 1-node q and merges 2-node p into p'. All the 1-nodes connected to p (different from q) are redirected to p'. Figure 10 shows an example of a 1-remove(q, p, p') in 3D. In the descending Morse complex, the 2-cells corresponding to p and p' are merged by deleting q. In the *MIG*, nodes p and q are removed along with all the arcs connecting them with the nodes in Z and S. Nodes r_1, r_2 and r_3 are then connected with p'.

7.2. Simplifying the Forman gradient

In [For98b], the *cancellation* operator is described in terms of updates on the Forman gradient V. In the following, we report the definition of such operator on a simplicial complex Σ endowed with a Forman gradient V, and for simplicity, we show examples for a triangle mesh (see [Gün12] for more details). It can be easily seen that the *remove* operator, discussed in Section 7.1, can be described in a similar fashion. Recall that a *separatrix V_k-path* is a V-path starting at a critical k-cell p and ending at a critical $(k-1)$-cell q. Operator *k-cancellation(q, p)* deletes a critical k-simplex q and a critical $(k+1)$-simplex p if and only if p and q are connected through exactly one separatrix V-path. The effect of a *k-cancellation(q, p)* on V is to reverse the gradient arrows on the separatrix V-path between p and q pairing, as a consequence, both p and q. Figure 11 shows the effect of *cancellation(q, p)* on a triangle mesh: q is a critical edge and p and p' are two critical triangles. Starting from q, the separatrix V_2-path, connecting q to p, is reversed. As a consequence, p and q are not critical anymore. The two separatrix V_2-paths, connecting p to r_1 and r_2, are extended with the reversed V-path, and now connect r_1 and r_2 to p'. The two separatrix V_1-paths starting from q and reaching z_1 and z_2 (in the left side), become non-separatrix V-paths (depicted with dotted lines in the right side).

7.3. Simplification algorithms

Several algorithms have been proposed for simplifying the morphology of 2D [EHZ01, BEHP04] and 3D [GNP*05, GNP*06, GSW12, ČomićDI13, FIDW14] scalar fields. Within a simplification algorithm, a simplification operator is iteratively applied to a Morse or a Morse-Smale complex, and this requires a priority-based scheme for ordering all possible simplifications. Priority measures the importance of the critical pairs to be deleted, and is defined in such a way to cause the removal of less important critical pairs (representing noise, for instance) at first. Algorithms have been proposed based on different functions used for estimating the priority value, namely *persistence*, *separatrix persistence*, or *topological saliency*.

Persistence has been widely used in the literature to estimate the importance of a pair of critical points [ELZ02]. Critical points pairs are computed sweeping the dataset in the direction of increasing function value adding vertices progressively. The topology of the partial dataset changes only when a critical point is added and each change can be described in terms of *creation* of a new component (or cycle) or *destruction* of a component/cycle. Two critical points *p* and *q* are paired if and only if *q* destroys what *p* has created. The importance of a critical point pair is then computed as the absolute difference of function values between the two critical points. However, the number of critical points pairs provided by the algorithm in [ELZ02] is not sufficient for effectively reducing the size of the segmentation. Thus, all the adjacent critical points (critical points connected by a separatrix path) are more likely to be considered as critical points pairs [BEHP04, GNP*05, GNP*06, ČomićDI13].

More recently, other methods have been proposed with the purpose of measuring the importance of critical points pairs in relation with the associated geometry of the Morse-Smale cells. In [WG09, GSW12] the notion of *separatrix persistence* is introduced for measuring the importance of the separatrices, i.e. , the extremal lines and surfaces, in an *MS* complex instead of only the critical points pairs. In [DSNW13] the notion of *topological saliency* is introduced for estimating the importance of a topological feature based on the presence (or absence) of other features in the neighborhood. Intuitively, the topological saliency of a feature normalizes the persistence of the features present in its neighborhood.

We have two types of algorithms for simplifying a Morse complex: algorithms based on simplification operators on the *MIG* [BEHP04, GNP*05, GNP*06, ČomićDI13] and algorithms based on operators on the Forman gradient [WG09, GSW12].

In [BEHP04], the geometry of the Morse cells is computed from scratch after each cancellation by re-triangulating the interior of each Morse 2-cell. Also the original function is modified accordingly in order to agree with the simplified topology. The storage cost for the geometry thus decreases progressively with the undergoing simplifications, but the computation time is highly affected by the reconstruction and smoothing step. In [GNP*06], the geometry of all the Morse cells is stored in a dedicated data structure organized as a Directed Acyclic Graph (*DAG*). When two cells of a Morse complex, *p* and *p'*, are merged during a simplification step, a new node is inserted in the *DAG* connected with the *DAG* nodes storing the geometry of *p* and *p'*. This approach provides an efficient handling for the geometry at the expense of a considerable increase in storage costs. In [ČomićDI13], the top cells and the 0-cells of the ascending and descending Morse complexes are stored in the nodes corresponding to maxima and minima. This reduces the storage cost of the geometry but affects the performances of the algorithm, since the top simplices (forming the ascending

and descending top cells) have to be merged after each simplification involving extrema.

Algorithms developed for a Forman gradient [WG09, GSW12] are typically slower since they have to update, for each simplification, all the gradient pairs along the separatrix *V*-path. On the other hand, they take advantage of the implicit representation that the Forman gradient provides, for both the Morse and Morse-Smale complexes, resulting in low storage costs. In these algorithms, simplifications are always represented as pairs of critical simplices. Simplification updates are performed on the Forman gradient and, at each step, the incidence relations among the remaining critical simplices are recomputed.

There have been some proposals in the literature to modify not only the Morse and Morse-Smale complexes using *cancellation*, but also the scalar function *f*, thus constructing a new function *g* that corresponds to the simplified field. The work for 2D scalar fields, presented in [BEHP04] and improved in [WGS10], modifies function *f* numerically, using Laplacian smoothing. In [BEHP04], function *f* is locally modified after each *cancellation* in order to agree with the new topology, by minimizing the error and obtaining a smooth approximation. In [WGS10], the bottleneck of the smoothing step performed after each *cancellation* in [BEHP04] is solved by constructing a topologically valid function after all *cancellation* steps. The two resulting C^0 functions are comparable but the algorithm in [WGS10] is faster. Moreover, in [WGS10], a novel scheme is devised to provide C^1-continuity. Another approach, presented in [EMP06] for the 2D case, modifies the scalar field *f* combinatorially, by changing the order in which the vertices appear in their sorted list according to function values.

7.3.1. Discussion

All algorithms for 2D scalar fields are equivalent in the sense that the resulting simplification process is monotonic, i.e., after each simplification all the new simplification operators have higher persistence value. Differences arise when working in three or higher dimensions.

The first problem arises with the *cancellation* operator on a Morse-Smale complex in 3D, because it may increase the number of arcs in the *MIG* (and thus the number of cells in the *MS* complex), as discussed in Section 7.1.

Several strategies have been proposed in [GBHP11], which aim at postponing a *cancellation* that would introduce a number of arcs greater than a predefined threshold, or nodes having more than a certain number of incident arcs. For overcoming the same problem in [ČomićDI13] a simplification algorithm based on the *remove* operator has been defined, which constantly reduce the number of arcs in *G* (see Section 7.1). In [Iur14], two simplification algorithms have been developed and compared, based on *remove* and *cancellation*, respectively. In Figure 12, an example of the simplified *MS* 1-skeleton obtained is shown.
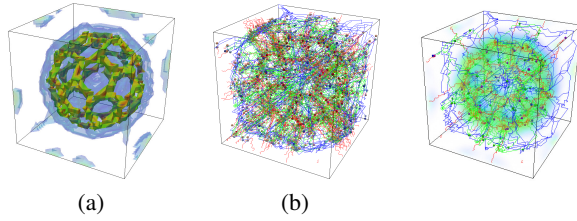
Figure 12: *The original scalar field (a) the original 1-skeleton of the MS complex (b) and its simplified version (c) computed with a persistence threshold of 0.2 with respect to the maximum persistence.*
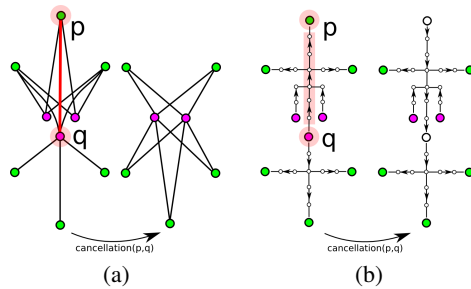
Figure 13: *1-cancellation(p,q) on the Incidence Graph (a) and on the Forman gradient (b). In (a), green dots denote 1-saddles and purple dots denote 2-saddles. In (b), cells forming the V-paths are depicted with white dots. Arrows between two dots indicate a gradient pair formed by the corresponding cells while straight lines between two dots indicate the incidence relation between the corresponding cells.*

It has been observed experimentally in [ĆomićDI13] that saddle-saddle operators are likely to be performed early in the simplification process (such simplifications can be interpreted as noise removal). When the data set is small and the number of simplifications is high compared to the total number of nodes, the two algorithms behave quite similarly. With the growth in the size of the data set, they start to differ: by using *remove* instead of *cancellation*, a 20% more compressed *MIG* can be obtained in about half the time. The number of arcs remains approximately the same by using *cancellation*, while immediately decreases by using *remove*.

Issues arise when performing 1-*cancellation* in the context of a Forman gradient defined on 3D Morse complexes. As described in [GRSW13], the connectivity among the critical cells involved in a 1-*cancellation* between saddles may change without control.

We use Figure 13 to explain the problem. In Figure 13(a), 1-*cancellation*(p,q) deletes a 1-saddle q and a 2-saddle q on a *MIG*. Thus, as described in Section 2.3, all the arcs connected to p and q are deleted, and the remaining nodes (previously connected with p and q) become connected by new

arcs. Figure 13(b) shows the same configuration with the separatrix V-paths between the critical simplices connected to p and q. When performing 1-*cancellation*(p,q), the arrows in the separatrix V_2-path between p and q are swapped. As a consequence, following the gradient arrows outgoing from the remaining 2-saddles (purple dots), the new separatrix V-paths will end at the three 1-saddles (green dots below them) only.

As pointed out in [GRSW13], this is not only a problem for the number of *MS* cells introduced by the simplification, but represents also a basic difference between the graph-based simplification and the gradient-based one.

8. Applications

Morse and Morse-Smale complexes have been used in a variety of applications including biology, medicine, terrain analysis, visualization of static and time-varying volume data. The computation of Morse and Morse-Smale complexes (see Sections 3 and 6) is at the basis of the feature-based analysis. The other essential step is the simplification of Morse complexes, discussed in Section 7, with the aim of removing noise or uninteresting features.

In [FHJB13], the Morse-Smale complex, computed on the auto diffusion function [GBAL09], and the corresponding *MIG* is used for finding non-rigid correspondences between feature points on triangle meshes. The topological graph obtained is invariant under isometry. In [DGG03], the simplified stable flow complex is applied for matching of shapes based on topological features instead of local geometry.

Simplified Morse-Smale complexes computed on real data have been used in [GBC*14]. The *MS* complex is used as a topological representation of molecular interactions (i.e., covalent and noncovalent bonds) for studying chemical systems through visualization. In [GDN*07], the same complex is used for extracting interesting features from a porous solid. In [NKWH08], a hierarchy of simplifications performed on the *MS* complex is computed and stored for studying biomolecular surfaces. In [SCP08], a watershed-based algorithm is adopted for extracting a skeletal model representing the main features of the cosmic evolution of the web. The skeleton computed is actually the 1-skeleton of the ascending Morse complex.

A lot of effort has been recently devoted to the analysis of time-varying scalar fields. In [LBM*06,BWT*11], two similar approaches are used for analyzing a mixing flow and the burning cells originated by a laboratory low-swirl nozzle, respectively. The *MS* complex is computed for each time step and, by keeping track of the critical points at different time steps, a graph [LBM*06] and a merging tree [BWT*11] is built for modeling the merge and split of the *MS* regions.

In [RKWH12], the combinatorial feature flow field is obtained as a combination of Forman gradients extracted from

each time step of a 2D scalar field. Using the combinatorial feature flow, critical points are tracked among the time-steps obtaining a tool for analyzing flow simulations. A similar framework is used in [KRHH11] for extracting and tracking minima into two-dimensional time-dependent vortex datasets.

In [GGL*14], a combination of the region-growing algorithm described in [GNPH07] and of the Forman gradient computation described in [RWS11] is defined. The *MS* complex is here used for the automated segmentation in histopathology. The Forman gradient (and the related *MS* complex) is computed using, as a constraint, the *MS* segmentation obtained through a semi-automated approach, i.e., the original segmentation is computed with the algorithm defined in [GNPH07] and then false negatives are removed from a domain expert.

Recently, homology and persistent homology computation have been applied to a plethora of fields including multivariate data analysis [RML12, RL14], sensor network analysis [dSG07], chemistry [MTCW10], astrophysics [VdWVE*11], etc., with a growing attention in the analysis of data in high dimensions.

As described in Section 2.2, persistent homology requires a filtration, defined on the input complex, to be computed. The algorithm described in [MN13] allows, given a filtration on a cell complex, to obtain a filtration on the discrete Morse complex computed with the coreduction-based algorithm described in Section 5.3.2 and, consequently, to compute persistent homology on such complex.

In [RWS11] and [GRWH12], persistence homology is applied to the study of 3D images. As described in Section 5.2, the critical cells obtained with the algorithm defined in [RWS11] (in the 3D case) are in one-to-one correspondence with the topological changes in the sub-level complexes and, thus, the persistent homology of the input complex corresponds to the persistent homology of the much smaller discrete Morse complex computed.

8.1. Multiresolution models for data analysis

One of the most prominent applications of Morse theory is the interactive visualization of scalar fields based on topological features, for analysis and exploration of data. Here, the degree of simplification of a morphological structure should adapt to current needs: less important areas of the dataset should be simplified while important areas should be represented at a certain user-defined quality. Because simplification processes are time consuming and not suited for real-time interaction, multi-resolution models have been designed to provide a hierarchy on which on-line queries can extract morphological representations at different resolution levels. A *multi-resolution model* is built based on a sequence of simplification operators applied to an initial representation at full resolution and producing a final coarse represen-

tation. The simplification operators are organized into a hierarchy (usually, a directed acyclic graph), representing the relations of dependency existing among them.

The first hierarchical representation of scalar fields, called *hierarchical watershed representation* [Beu94], can be found in image analysis to reduce over-segmentation. Most common hierarchical models for Morse complexes have been developed for terrains, and can be applied to 2-manifolds embedded in 3D space and endowed with a scalar field [EHZ01, BEHP04, DDPV06, DDV07, ČDI12]. In [BEHP04, BPH05], a multi-resolution morphological model for 2D scalar fields has been described, which is based on the *cancellation* operator and on the Morse-Smale complex. The basic component is a *diamond*, which consists of a saddle *s*, its adjacent minima and maxima, and the integral lines connecting them.

In [ČDI12], a similar model has been defined for scalar fields in any dimensions, called the *Multi-resolution Morse Incidence Graph* (*MMIG*). The *MMIG* is based on the *remove* operator [ČomićD11] and on the Morse Incidence Graph (*MIG*). Results are presented from the extraction of representations at various persistence levels at both uniform and variable resolution from several 2D and 3D data sets.

All multi-resolution models reviewed so far are only concerned with the morphology of the scalar, while they do not modify the underlying geometrical model of the scalar field. In [BEHP04], the Morse cells are re-triangulated after each simplification/refinement for describing the given morphological resolution.

In [DDM*03a], a triangle mesh Σ with associated Morse-Smale complex is simplified through iterative vertex removal, while considering Σ as a constrained triangulation with separatrix lines acting as constraints. Only the geometric structure of separatrix lines is simplified, while the combinatorial structure of the Morse-Smale complex (described by the Morse incidence graph) remains the same across all levels of resolution. Similar ideas, based on recursive simplex bisection and edge-collapse, have been applied for the extraction of iso-surfaces, at different resolution levels, from volume datasets in [GP00] and [CL03] respectively.

The *Multi-resolution Morse Triangulation (MMT)* [DDMV10] is a first attempt to simplify both geometry and morphology of a scalar field in a combined and consistent way. However, it is verbose, and the simplification of the triangle mesh through half-edge collapse may generate new critical points.

Recently, a new multi-resolution model for triangle meshes endowed with a scalar field has been proposed [Iur14], rooted in discrete Morse theory. The model, called a *Hierarchical Forman Triangulation (HFT)*, is based on a discrete Morse gradient computed on a triangle mesh Σ, a topological simplification operator on the gradient, and

a so-called *gradient-aware edge-contraction* operator on Σ, which avoids deleting or creating critical simplices.

9. Concluding remarks

We have reviewed algorithms for computing Morse and Morse-Smale decompositions. Algorithms based on piecewise linear Morse theory, or on the watershed transform, have been used for segmenting a shape endowed with a scalar field, thus obtaining morphological compact representations of the original dataset. Most of these methods are rooted in geographic data processing and have been used for terrain segmentation and analysis. More recently, similar algorithms have been developed for analysis and visualization of volume data.

With the increasing availability of large multidimensional data sets, arising from scientific simulations, much more attention has been devoted to methods rooted in discrete Morse theory. In this survey, we have addressed two distinct application areas. The first one is the analysis of scalar fields in two and three dimensions as an efficient alternative to more traditional approaches. Forman-based algorithms are preferable compared to algorithms rooted in piecewise linear Morse theory since the former are dimension independent by definition and particularly well suited for dealing with high dimensional data. Moreover, the range of features which can be extracted efficiently is not limited to the top cells of the Morse complexes or the 1-skeleton of the *MS* complex. The second application area is homology and persistent homology computation. A Forman gradient implicitly describes a Morse complex which has the same homology of the original complex. Then, by using a filtration on the input dataset, topological information about the dataset are obtained by computing persistent homology. Table 1 shows a summary of the algorithms discussed in this paper.

Because of the increasing precision of machinery involved in medical and scientific analysis, datasets are characterized by a constantly growing number of sample points. This poses a serious issue to Morse and Morse-Smale computation. Thus, future investigations will require developing parallel algorithms and data structures able to take fully advantage of multi-core architectures for improving the computation time without running low in memory. Compact topological data structures based on a spatial index, like the PR-star tree [WDFV11], could be an excellent tool for handling huge datasets. Such data structures offer a natural subdivision of the input geometry defining the independent sets used in parallel computation. Moreover, these structures offer compact representations for a dataset as well as for the relations among its geometrical entities. This is especially interesting for homology, which needs to be computed effectively and efficiently on data sets of high dimensions and large size, where the ratio between the number of simplices and sample points of the datasets increases exponentially. Thus, ad-hoc data structures have to be used for limiting the

ALGORITHM	INPUT	OUTPUT
MORSE DECOMPOSITIONS		
———— Boundary-based ————		
Takashahi et al. [TIKU95]	2D Σ	*MS* 1-cells
Edelsbrunner et al. [EHZ01]	2D Σ	*MS* 1-cells
Bremer et al. [BEHP04]	2D Σ	*MS* 1-cells
Pascucci et al. [Pas04]	2D Σ	*MS* 1-cells
De Berg et al. [dBT11]	2D Σ	*MS* 1-cells
Edelsbrunner et al. [EHNP03]	3D Σ	*MS* 1-/2-cells
Bajaj et al. [BPS98]	2D/3D Γ	*MS* 1-/2-cells
Schneider and Wood [SW04, Sch05]	2D Γ	*MS* 1-cells
———— Region-growing ————		
Danovaro et al. [DDM*03a]	2D Σ	*Morse* top cells
Danovaro et al. [DDM03b]	2D/3D Σ	*Morse* top cells
Dey et al. [DGG03]	2D/3D Σ	stable flow complex
Magillo et al. [MDD*09]	2D Σ	*Morse* top cells
Gyulassy et al. [GNPH07]	2D/3D Σ	*MS* all cells
———— Watershed ————		
Topographic distance [MB90, Mey94]	2D Γ	*Morse* 2-cells
Simulated Immersion [VS91, Soi04]	nD Γ	*Morse* top cells
Rain Falling [MW99, SS00]	2D/3D Σ	*Morse* top cells
———— Forman gradient ————		
Gunther et al. [GRWH12]	Forman	all *MS* cells
Shivashankarar et al. [SN12]	Forman	all *MS* cells
Weiss et al. [WIFD13]	Forman	all *MS* cells
FORMAN GRADIENT		
———— Constrained ————		
Cazals et al. [CCL03]	2D Σ	*Forman*
King et al. [KKM05]	3D Σ	*Forman*
Gyulassy et al. [GBHP08]	nD Γ	*Forman*
Robins et al. [RWS11]	nD Γ	*Forman*
Gyulassy et al. [GBP12]	nD Γ	*Forman*
———— Unconstrained ————		
Lewiner et al. [LLT03]	2D Γ	*Forman*
Benedetti et al. [BL14]	nD Γ	*Forman*
Harker et al. [HMMN14]	nD Γ	*Forman*
Fugacci et al. [FID14]	nD Σ	*Forman*

Table 1: *Summary of the reviewed algorithms. For each of them the expected input, simplicial complex (Σ) or cell complex (Γ), and the cells of the Morse or Morse-Smale (MS) complex computed are indicated.*

number of stored simplices. Data structures storing only the vertices and the top simplices have given already good preliminary results [FID14]. Since the PR-star tree is even more compact, we believe that its generalization could improve existing approaches to homology and persistent homology computation.

References

[Ago05] AGOSTON M. K.: *Computer Graphics and Geometric Modeling: Mathematics*. Springer-Verlag London Ltd., 2005. 3

[AH35] ALEXANDROFF P., HOPF H.: *Topologie I*, vol. 1035. Berlin, 1935. 3

[Ban67] BANCHOFF T.: Critical points and curvature for embedded polyhedra. *J. of Differential Geometry 1* (1967), 245–256. 4, 5

[Ban70] BANCHOFF T.: Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly 77*, 5 (1970), 475–485. 4

[BDF*08] BIASOTTI S., DE FLORIANI L., FALCIDIENO B., FROSINI P., GIORGI D., LANDI C., PAPALEO L., SPAGNUOLO M.: Describing shapes by geometrical-topological properties of real functions. *ACM Computing Surveys 40*, 4 (2008), Article 12. 5

[BEHP03] BREMER P.-T., EDELSBRUNNER H., HAMANN B., PASCUCCI V.: A multi-resolution data structure for two-dimensional Morse functions. In *Proc. IEEE Visualization 2003* (2003), IEEE Computer Society, pp. 139–146. 8

[BEHP04] BREMER P.-T., EDELSBRUNNER H., HAMANN B., PASCUCCI V.: A topological hierarchy for functions on triangulated surfaces. *Trans. on Visualization and Computer Graphics 10*, 4 (July/August 2004), 385–396. 6, 8, 16, 17, 18, 20, 21

[Beu94] BEUCHER S.: Watershed, hierarchical segmentation and waterfall algorithm. In *Mathematical Morphology and its Applications to Image Processing* (1994), Serra J., Soille P., (Eds.), vol. 2 of *Computational Imaging and Vision*, Springer, pp. 69–76. 20

[BGSF08] BIASOTTI S., GIORGI D., SPAGNUOLO M., FALCIDIENO B.: Reeb graphs for shape analysis and applications. *Theoretical Computer Science 392*, 1 (2008), 5–22. 1

[BL14] BENEDETTI B., LUTZ F. H.: Random discrete Morse theory and a new library of triangulations. *Experimental Mathematics 23*, 1 (2014), 66–94. 10, 12, 21

[BLPS13] BURTON B. A., LEWINER T., PAIXÃO J., SPREER J.: Parameterized complexity of discrete Morse theory. In *Symposuim on Computational Geometry 2013, SoCG '13, Rio de Janeiro, Brazil, June 17-20, 2013* (2013), da Fonseca G. D., Lewiner T., Peñaranda L. M., Chan T. M., Klein R., (Eds.), ACM, pp. 127–136. 13

[BPH05] BREMER P.-T., PASCUCCI V., HAMANN B.: Maximizing adaptivity in hierarchical topological models. In *Proc. Int. Conf. on Shape Modeling and Applications 2005 (SMI '05)* (Los Alamitos, California, 2005), Belyaev A., Pasko A., Spagnuolo M., (Eds.), IEEE Computer Society Press, pp. 300–309. 20

[BPS98] BAJAJ C. L., PASCUCCI V., SHIKORE D. R.: Visualization of scalar topology for structural enhancement. In *Proc. IEEE Visualization'98* (1998), IEEE Computer Society, pp. 51–58. 7, 21

[BWT*11] BREMER P., WEBER G. H., TIERNY J., PASCUCCI V., DAY M. S., BELL J. B.: Interactive exploration and analysis of large-scale simulations using topology-based data segmentation. *IEEE Trans. Vis. Comput. Graph. 17*, 9 (2011), 1307–1324. 19

[CCL03] CAZALS F., CHAZAL F., LEWINER T.: Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *Proc. 9th Annual Symposium on Computational Geometry* (New York, USA, 2003), ACM Press, pp. 351–360. 10, 11, 13, 21

[ČDI12] ČOMIĆ L., DE FLORIANI L., IURICICH F.: Dimension-independent multi-resolution Morse complexes. *Computers & Graphics 36*, 5 (2012), 541–547. 16, 20

[ČDMI14] ČOMIĆ L., DE FLORIANI L., MAGILLO P., IURICICH F.: *Morphological Modeling of Terrains and Volume Data*. Springer Briefs in Computer Science. Springer, 2014. 6

[CDW11] CANINO D., DE FLORIANI L., WEISS K.: IA*: An adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions. *Computers & Graphics 35*, 3 (2011), 747–753. 10, 13

[CL03] CHIANG Y.-J., LU X.: Progressive simplification of tetrahedral meshes preserving all isosurface topologies. *Computer Graphics Forum 22* (2003), 493–504. 20

[ČomićD11] ČOMIĆ L., DE FLORIANI L.: Dimension-independent simplification and refinement of Morse complexes. *Graphical Models 73*, 5 (Sep 2011), 261–285. 16, 20

[ČomićDI13] ČOMIĆ L., DE FLORIANI L., IURICICH F.: Simplification operators on a dimension-independent graph-based representation of Morse complexes. In *ISMM* (2013), Hendriks C. L. L., Borgefors G., Strand R., (Eds.), vol. 7883 of *Lecture Notes in Computer Science*, Springer, pp. 13–24. 17, 18, 19

[Con86] CONNOLLY M. L.: Measurement of protein surface shape by solid angles. *J. of Molecular Graphics 4*, 1 (1986), 3 – 6. 11

[dBT11] DE BERG M., TSIROGIANNIS C.: Exact and approximate computations of watersheds on triangulated terrains. In *Proc. 19th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems* (2011), pp. 74–83. 6, 21

[DDM*03a] DANOVARO E., DE FLORIANI L., MAGILLO P., MESMOUDI M. M., PUPPO E.: Morphology-driven simplification and multiresolution modeling of terrains. In *Proc. ACM GIS 2003 - The 11th Int. Symposium on Advances in Geographic Information Systems* (2003), E.Hoel, P.Rigaux, (Eds.), ACM Press, pp. 63–70. 7, 8, 20, 21

[DDM03b] DANOVARO E., DE FLORIANI L., MESMOUDI M. M.: Topological analysis and characterization of discrete scalar fields. In *Geometry, Morphology, and Computational Imaging*, T.Asano, R.Klette, C.Ronse, (Eds.), vol. 2616 of *Lecture Notes in Computer Science*. Springer Verlag, 2003, pp. 386–402. 7, 8, 21

[DDMV10] DANOVARO E., DE FLORIANI L., MAGILLO P., VITALI M.: Multiresolution Morse triangulations. In *Symposium on Solid and Physical Modeling* (2010), Elber G., Fischer A., Keyser J., Kim M.-S., (Eds.), ACM, pp. 183–188. 20

[DDPV06] DANOVARO E., DE FLORIANI L., PAPALEO L., VITALI M.: A multi-resolution representation for terrain morphology. In *Proc. fourth Int. Conf. on Geographic Information Science - GIScience 2006* (2006). 20

[DDV07] DANOVARO E., DE FLORIANI L., VITALI M.: Multiresolution Morse-Smale complexes for terrain modeling. In *14th Int. Conf. on Image Analysis and Processing* (2007). 20

[DGG03] DEY T. K., GIESEN J., GOSWAMI S.: Shape segmentation and matching with flow discretization. In *Algorithms and Data Structures, 8th International Workshop WADS* (2003), Dehne F. K. H. A., Sack J., Smid M. H. M., (Eds.), pp. 25–36. 6, 7, 19, 21

[DH05] DE FLORIANI L., HUI A.: Data structures for simplicial complexes: An analysis and a comparison. In *Proc. 3rd Eurographics Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, 2005), Desbrun M., Pottmann H., (Eds.), vol. 255 of *ACM Int. Conf. Proceeding Series*, Eurographics Association, pp. 119–128. 2, 10

[Dij59] DIJKSTRA E. W.: A note on two problems in connexion with graphs. *Numerische Mathematik 1* (1959), 269–271. 8

[DKMW11] DŁOTKO P., KACZYNSKI T., MROZEK M., WANNER T.: Coreduction homology algorithm for regular cw-complexes. *Discrete & Computational Geometry 46*, 2 (2011), 361–388. 10, 12

[dSG07] DE SILVA V., GHRIST R.: Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology 7* (2007), 339–358. 20

[DSNW13] DORAISWAMY H., SHIVASHANKAR N., NATARAJAN V., WANG Y.: Topological saliency. *Computers & Graphics 37*, 7 (2013), 787–799. 18

[DW14] DŁOTKO P., WAGNER H.: Simplification of complexes for persistent homology computations. *Homology, Homotopy and Applications 16*, 1 (2014), 49–63. 10

[Ede87] EDELSBRUNNER H.: *Algorithms in Combinatorial Geometry*. Springer Verlag, Berlin, 1987. 4, 10

[Ede03] EDELSBRUNNER H.: Surface reconstruction by wrapping finite sets in space. In *Discrete and Computational Geometry*, Aronov B., Basu S., Pach J., Sharir M., (Eds.), vol. 25 of *Algorithms and Combinatorics*. Springer Berlin Heidelberg, 2003, pp. 379–404. 6

[EH08] EDELSBRUNNER H., HARER J.: Persistent homology-a survey. *Contemporary mathematics 453* (2008), 257–282. 3

[EHNP03] EDELSBRUNNER H., HARER J., NATARAJAN V., PASCUCCI V.: Morse-Smale complexes for piecewise linear 3-manifolds. In *Proc. 19th ACM Symposium on Computational Geometry* (2003), pp. 361–370. 5, 6, 21

[EHZ01] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical Morse complexes for piecewise linear 2-manifolds. In *Proc. 17th ACM Symposium on Computational Geometry* (2001), pp. 70–79. 5, 6, 8, 17, 20, 21

[ELZ02] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. *Discrete & Computational Geometry 28*, 4 (2002), 511–533. 16, 18

[EMP06] EDELSBRUNNER H., MOROZOV D., PASCUCCI V.: Persistence-Sensitive Simplification of Functions on 2-Manifolds. In *SCG'06: Proc. of the 22nd Annual Symposium on Computational Geometry 2006* (2006). 18

[FHJB13] FENG W., HUANG J., JU T., BAO H.: Feature correspondences using Morse Smale complex. *The Visual Computer 29*, 1 (2013), 53–67. 19

[FID14] FUGACCI U., IURICICH F., DE FLORIANI L.: Efficient computation of simplicial homology through acyclic matching. In *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2014 16th International Symposium on* (Sept 2014), pp. 587–593. 10, 12, 13, 21

[FIDW14] FELLEGARA R., IURICICH F., DE FLORIANI L., WEISS K.: Efficient computation and simplification of discrete Morse decompositions on triangulated terrains. In *22th ACM SIGSPATIAL Int. Symposium on Advances in Geographic Information Systems, ACM-GIS 2014, November 4-7, 2014, Dallas, Texas, USA, to appear* (2014). 10, 11, 16, 17

[For98a] FORMAN R.: Combinatorial vector fields and dynamical systems. *Mathematische Zeitschrift 228* (1998), 629–681. 9

[For98b] FORMAN R.: Morse Theory for Cell Complexes. *Advances in Mathematics 134* (1998), 90–145. 1, 9, 11, 17

[GBAL09] GĘBAL K., BÆRENTZEN J. A., AANÆS H., LARSEN R.: Shape analysis using the auto diffusion function. In *Proceedings of the Symposium on Geometry Processing* (Aire-la-Ville, Switzerland, Switzerland, 2009), SGP '09, Eurographics Association, pp. 1405–1413. 19

[GBC*14] GÜNTHER D., BOTO R. A., CONTRERAS-GARCIA J., PIQUEMAL J., TIERNY J.: Characterizing molecular interactions in chemical systems. *IEEE Trans. Vis. Comput. Graph. 20*, 12 (2014), 2476–2485. 19

[GBHP08] GYULASSY A., BREMER P.-T., HAMANN B., PASCUCCI V.: A practical approach to Morse-Smale complex computation: Scalability and generality. *IEEE Trans. on Visualization and Computer Graphics 14*, 6 (2008), 1619–1626. 11, 21

[GBHP11] GYULASSY A., BREMER P.-T., HAMANN B., PASCUCCI V.: Practical considerations in Morse-Smale complex computation. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, Pascucci V., Tricoche X., Hagen H., Tierny J., (Eds.), Mathematics and Visualization. Springer Verlag, Heidelberg, 2011, pp. 67–78. 10, 18

[GBP12] GYULASSY A., BREMER P., PASCUCCI V.: Computing Morse-Smale complexes with accurate geometry. *IEEE Trans. Vis. Comput. Graph. 18*, 12 (2012), 2014–2022. 9, 10, 11, 21

[GDN*07] GYULASSY A., DUCHAINEAU M. A., NATARAJAN V., PASCUCCI V., BRINGA E., HIGGINBOTHAM A., HAMANN B.: Topologically clean distance fields. *IEEE Trans. Vis. Comput. Graph. 13*, 6 (2007), 1432–1439. 19

[GGL*14] GYULASSY A., GÜNTHER D., LEVINE J. A., TIERNY J., PASCUCCI V.: Conforming Morse-Smale complexes. *IEEE Trans. Vis. Comput. Graph. 20*, 12 (2014), 2595–2603. 20

[GJ08] GIESEN J., JOHN M.: The flow complex: A data structure for geometric modeling. *Comput. Geom. 39*, 3 (2008), 178–190. 6, 7

[GKK*12] GYULASSY A., KOTAVA N., KIM M., HANSEN C. D., HAGEN H., PASCUCCI V.: Direct feature visualization using Morse-Smale complexes. *IEEE Trans. Vis. Comput. Graph. 18*, 9 (2012), 1549–1562. 13

[GNP*05] GYULASSY A., NATARAJAN V., PASCUCCI V., BREMER P., HAMANN B.: Topology-based simplification for feature extraction from 3d scalar fields. In *16th IEEE Visualization Conference (VIS 2005), 23-28 October 2005, Minneapolis, MN, USA* (2005), p. 68. 17, 18

[GNP*06] GYULASSY A., NATARAJAN V., PASCUCCI V., BREMER P.-T., HAMANN B.: A topological approach to simplification of Three-dimensional scalar functions. *IEEE Trans. on Visualization and Computer Graphics 12*, 4 (2006), 474–484. 16, 17, 18

[GNPH07] GYULASSY A., NATARAJAN V., PASCUCCI V., HAMANN B.: Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. *IEEE Trans. on Visualization and Computer Graphics 13*, 6 (2007), 1440–1447. 7, 20, 21

[GP00] GERSTNER T., PAJAROLA R.: Topology preserving and controlled topology simplifying multi-resolution isosurface extraction. In *Proc. IEEE Visualization'00* (2000), pp. 259–266. 20

[GRSW13] GÜNTHER D., REININGHAUS J., SEIDEL H.-P., WEINKAUF T.: Notes on the simplification of the Morse-Smale complex. In *Proc. TopoInVis* (Davis, U.S.A., March 2013). 16, 19

[GRWH12] GÜNTHER D., REININGHAUS J., WAGNER H., HOTZ I.: Efficient computation of 3D Morse-Smale complexes and persistent homology using discrete Morse theory. *The Visual Computer 28*, 10 (2012), 959–969. 10, 15, 20, 21

[GSW12] GÜNTHER D., SEIDEL H., WEINKAUF T.: Extraction

of dominant extremal structures in volumetric data using separatrix persistence. *Comput. Graph. Forum 31*, 8 (2012), 2554–2566. 16, 17, 18

[Gün12] GÜNTHER D.: *Topological analysis of discrete scalar data*. PhD thesis, 2012. Saarbrücken, University, Dissertation, 2012. 17

[HMM*10] HARKER S., MISCHAIKOW K., MROZEK M., NANDA V., WAGNER H., JUDA M., DŁOTKO P.: The efficiency of a homology algorithm based on discrete Morse theory and coreductions. In *Proceedings 3rd International Workshop on Computational Topology in Image Context (CTIC 2010). Image A* (2010), vol. 1, pp. 41–47. 12

[HMMN14] HARKER S., MISCHAIKOW K., MROZEK M., NANDA V.: Discrete Morse theoretic algorithms for computing homology of complexes and maps. *Foundations of Computational Mathematics 14*, 1 (2014), 151–184. 10, 12, 21

[Iur14] IURICICH F.: *Multi-resolution shape analysis based on discrete Morse decompositions*. PhD thesis, University of Genova – DIBRIS, Italy, 2014. 8, 18, 20

[KKM05] KING H., KNUDSON K., MRAMOR N.: Generating discrete Morse functions from point data. *Experimental Mathematics 14*, 4 (2005), 435–444. 10, 11, 21

[KRHH11] KASTEN J., REININGHAUS J., HOTZ I., HEGE H.: Two-dimensional time-dependent vortex regions based on the acceleration magnitude. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2080–2087. 20

[LBM*06] LANEY D., BREMER P. T., MASCARENHAS A., MILLER P., PASCUCCI V.: Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (Sept. 2006), 1053–1060. 19

[LLT03] LEWINER T., LOPES H., TAVARES G.: Optimal discrete Morse functions for 2-manifolds. *Computational Geometry 26*, 3 (2003), 221 – 233. 10, 12, 21

[LLT04] LEWINER T., LOPES H., TAVARES G.: Applications of Forman's discrete Morse theory to topology visualization and mesh compression. *IEEE Trans. Vis. Comput. Graph. 10*, 5 (2004), 499–508. 10

[Mat02] MATSUMOTO Y.: *An Introduction to Morse Theory*, vol. 208 of *Translations of Mathematical Monographs*. American Mathematical Society, 2002. 1, 3, 4

[MB90] MEYER F., BEUCHER S.: Morphological segmentation. *J. of Visual Communication and Image Representation 1* (1990), 21–45. 8, 21

[MB09] MROZEK M., BATKO B.: Coreduction homology algorithm. *Discrete & Computational Geometry 41*, 1 (2009), 96–118. 10, 12

[MDD*09] MAGILLO P., DANOVARO E., DE FLORIANI L., PAPALEO L., VITALI M.: A discrete approach to compute terrain morphology. *Computer Vision and Computer Graphics Theory and Applications 21* (2009), 13–26. 7, 21

[MDI13] MAGILLO P., DE FLORIANI L., IURICICH F.: Morphologically-aware elimination of flat edges from a TIN. In *Proc. 21th ACM SIGSPATIAL Int. Conf. on Advances in Geographic Information Systems* (2013), pp. 244–253. 8

[Mey94] MEYER F.: Topographic distance and watershed lines. *Signal Processing 38* (1994), 113–125. 5, 8, 21

[Mil63] MILNOR J.: *Morse Theory*. Princeton University Press, New Jersey, 1963. 1, 3

[MN13] MISCHAIKOW K., NANDA V.: Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry 50*, 2 (2013), 330–353. 10, 13, 20

[MTCW10] MARTIN S., THOMPSON A., COUTSIAS E. A., WATSON J.-P.: Topology of cyclo-octane energy landscape. *The Journal of Chemical Physics 132*, 23 (2010), –. 20

[Mun84] MUNKRES J.: *Elements of Algebraic Topology*. Advanced book classics. Perseus Books, 1984. 3

[MW99] MANGAN A., WHITAKER R.: Partitioning 3D surface meshes using watershed segmentation. *Trans. on Visualization and Computer Graphics 5*, 4 (1999), 308–321. 8, 21

[MW10] MROZEK M., WANNER T.: Coreduction homology algorithm for inclusions and persistent homology. *Comput. Math. Appl. 60*, 10 (Nov. 2010), 2812–2833. 10, 12

[NKWH08] NATARAJAN V., KOEHL P., WANG Y., HAMANN B.: Visual analysis of biomolecular surfaces. In *Visualization in Medicine and Life Sciences*, Linsen L., Hagen H., Hamann B., (Eds.), Mathematics and Visualization. Springer Berlin Heidelberg, 2008, pp. 237–255. 19

[Pas04] PASCUCCI V.: Topology diagrams of scalar fields in scientific visualization. In *Topological Data Structures for Surfaces*, Rana S., (Ed.). John Wiley & Sons Ltd, 2004, pp. 121–129. 6, 8, 21

[RGH*12] REININGHAUS J., GÜNTHER D., HOTZ I., WEINKAUF T., SEIDEL H.: Combinatorial gradient fields for 2d images with empirically convergent separatrices. *CoRR abs/1208.6523* (2012). 9

[RKWH12] REININGHAUS J., KASTEN J., WEINKAUF T., HOTZ I.: Efficient computation of combinatorial feature flow fields. *IEEE Trans. Vis. Comput. Graph. 18*, 9 (2012), 1563–1573. 19

[RL14] RIECK B., LEITTE H.: Structural analysis of multivariate point clouds using simplicial chains. *Computer Graphics Forum 33*, 8 (2014), 28–37. 20

[RM00] ROERDINK J., MEIJSTER A.: The watershed transform: Definitions, algorithms, and parallelization strategies. *Fundamenta Informaticae 41* (2000), 187–228. 5

[RML12] RIECK B., MARA H., LEITTE H.: Multivariate data analysis using persistence-based filtering and topological signatures. *Visualization and Computer Graphics, IEEE Transactions on 18*, 12 (Dec 2012), 2382–2391. 20

[RWS11] ROBINS V., WOOD P. J., SHEPPARD A. P.: Theory and algorithms for constructing discrete Morse complexes from grayscale digital images. *IEEE Trans. on Pattern Analysis and Machine Intelligence 33*, 8 (2011), 1646–1658. 10, 11, 20, 21

[Sch05] SCHNEIDER B.: Extraction of hierarchical surface networks from bilinear surface patches. *Geographical Analysis 37*, 2 (2005), 244–263. 7, 21

[SCP08] SOUSBIE T., COLOMBI S., PICHON C.: The fully connected n-dimensional skeleton: probing the evolution of the cosmic web. *CoRR abs/0809.2423* (2008). 9, 19

[SMN12] SHIVASHANKAR N., MAADASAMY S., NATARAJAN V.: Parallel computation of 2D Morse-Smale complexes. *IEEE Trans. on Visualization and Computer Graphics 18*, 10 (2012), 1757–1770. 10, 11

[SN12] SHIVASHANKAR N., NATARAJAN V.: Parallel computation of 3D Morse-Smale complexes. *Computer Graphics Forum 31*, 3 (2012), 965–974. 10, 11, 15, 21

[Soi04] SOILLE P.: *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, Berlin and New York, 2004. 8, 21

[SS00] STOEV S. L., STRASSER W.: Extracting regions of interest applying a local watershed transformation. In *Proc. IEEE Visualization'00* (2000), ACM Press, pp. 21–28. 8, 21

[SW04] SCHNEIDER B., WOOD J.: Construction of metric surface networks from raster-based DEMs. In *Topological Data Structures for Surfaces*, Rana S., (Ed.). John Wiley & Sons Ltd, 2004, pp. 53–70. 7, 21

[TIKU95] TAKAHASHI S., IKEDA T., KUNII T. L., UEDA M.: Algorithms for extracting correct critical points and constructing topological graphs from discrete geographic elevation data. In *Computer Graphics Forum* (1995), vol. 14, pp. 181–192. 5, 6, 8, 21

[VCY12] VEGTER G., CHATTOPADHYAY A., YAP C. K.: Certified computation of planar Morse-Smale complexes. In *Proceedings of the Twenty-eighth Annual Symposium on Computational Geometry* (New York, NY, USA, 2012), SoCG '12, ACM, pp. 259–268. 4

[VdWVE*11] VAN DE WEYGAERT R., VEGTER G., EDELSBRUNNER H., JONES B. J., PRANAV P., PARK C., HELLWIG W. A., ELDERING B., KRUITHOF N., BOS E., ET AL.: Alpha, Betti and the megaparsec universe: On the topology of the cosmic web. In *Transactions on Computational Science XIV* (2011), Springer-Verlag, pp. 60–101. 20

[VS91] VINCENT L., SOILLE P.: Watershed in digital spaces: An efficient algorithm based on immersion simulation. *IEEE Trans. on Pattern Analysis and Machine Intelligence 13*, 6 (1991), 583–598. 5, 8, 21

[WDFV11] WEISS K., DE FLORIANI L., FELLEGARA R., VELLOSO M.: The PR-star octree: a spatio-topological data structure for tetrahedral meshes. In *GIS* (2011), Cruz I. F., Agrawal D., Jensen C. S., Ofek E., Tanin E., (Eds.), ACM, pp. 92–101. 21

[WG09] WEINKAUF T., GÜNTHER D.: Separatrix persistence: Extraction of salient edges on surfaces using topological methods. *Comput. Graph. Forum 28*, 5 (2009), 1519–1528. 18

[WGS10] WEINKAUF T., GINGOLD Y. I., SORKINE O.: Topology-based smoothing of 2D scalar fields with C^1-continuity. *Computer Graphics Forum 29*, 3 (2010), 1221–1230. 18

[WIFD13] WEISS K., IURICICH F., FELLEGARA R., DE FLORIANI L.: A primal/dual representation for discrete Morse complexes on tetrahedral meshes. *Computer Graphics Forum 32*, 3 (2013), 361–370. 10, 11, 14, 15, 21

[Zom05] ZOMORODIAN A. J.: *Topology for computing*. No. 16. Cambridge University Press, 2005. 3

Biographical information

Leila De Floriani is Professor of Computer Science at the University of Genova, where she founded the Geometric Modeling and Computer Graphics group. She has written over 280 publications in the fields of geometric modeling, computer graphics, visualization and geographic data processing. Her current research interests include multiresolution shape modeling, computational topology, shape analysis and understanding, scientific visualization, and terrain modeling. She is the Editor-in-Chief of the IEEE Visualization and Computer Graphics. She is and has been on the editorial board of major journals and on the program committee of major conferences in the field. She has held positions at the Italian National Research Council, the University of Nebraska, Rensselaer Polytechnic Institute, and the University of Maryland. She is a member of ACM, a senior member of IEEE, and a fellow of the International Association for Pattern Recognition (IAPR).

Ulderico Fugacci received the MSc in Mathematics from the University of Genova, Genova (Italy), in 2012. Since 2013, he is a PhD student in Computer Science at the University of Genova under the supervision of Leila De Floriani and Maria Evelina Rossi. His research interests are in computational algebraic topology, simplicial and cellular homology, topological and multiresolution analysis.

Federico Iuricich received the MSc in Computer Science at the University of Genova, Genova (Italy), in 2010, and a PhD in Computer Science, at the same university in 2014. He is now a postdoctoral fellow at the University of Maryland, College Park (MD), USA. His research interests include geometric modeling, topological analysis, multiresolution analysis and visualization.

Paola Magillo received an advanced degree in Computer Science in 1992, and a PhD in Computer Science in 1999, both from the University of Genova, Genova (Italy). In 1996, she became a researcher, and, since 2004, she is an associate professor at University of Genova. Her research interests include computational geometry, geometric modeling, geographic information systems and computer graphics.