

Describing shapes by geometrical-topological properties of real functions

S. BIASOTTI

National Research Council, Institute of Applied Mathematics and Information Technologies, Genova, Italy

L. DE FLORIANI

Department of Computer and Information Science, University of Genova, Italy

B. FALCIDIENO

National Research Council, Institute of Applied Mathematics and Information Technologies, Genova, Italy

P. FROSINI

Department of Mathematics and ARCES, University of Bologna, Italy

D. GIORGI

National Research Council, Institute of Applied Mathematics and Information Technologies, Genova, Italy

C. LANDI

Department of Engineering Sciences and Methods, University of Modena and Reggio Emilia, Italy

L. PAPALETTO

Department of Computer and Information Science, University of Genova, Italy

and

M. SPAGNUOLO

National Research Council, Institute of Applied Mathematics and Information Technologies, Genova, Italy

Differential topology, and specifically Morse theory, provides a suitable setting for formalizing and solving several problems related to shape analysis. The fundamental idea behind Morse theory is that of combining the topological exploration of a shape with quantitative measurement of geometrical properties provided by a real function defined on the shape. The added value of approaches based on Morse theory is in the possibility of adopting different functions as shape descriptors according to the properties and invariants that one wishes to analyze. In this sense, Morse theory allows one to construct a *general framework* for shape characterization, parametrized with respect to the mapping function used, and possibly the space associated with the shape. The mapping function plays the role of a *lens* through which we look at the properties of the shape, and different functions provide different insights.

In the last decade, an increasing number of methods that are rooted in Morse theory and make use of properties of real-valued functions for describing shapes have been proposed in the literature. The methods proposed range from approaches which use the configuration of contours for encoding topographic surfaces to more recent work on size theory and persistent homology. All these have been developed over the years with a specific target domain and it is not trivial to systematize this work and understand the links, similarities and differences among the different methods. Moreover, different terms have been used to denote the same mathematical constructs, which often overwhelms the understanding of the underlying common framework.

The aim of this survey is to provide a clear vision of what has been developed so far, focusing on methods that make use of theoretical frameworks that are developed for *classes* of real functions rather than for a single function, even if they are applied in a restricted manner. The term geometrical-topological used in the title is meant to underline that both levels of information content are relevant for the applications of shape descriptions: geometrical, or metrical, properties and attributes are crucial for characterizing specific instances of features, while topological properties are necessary to abstract and classify shapes according to invariant aspects of their geometry. The approaches surveyed will be discussed in detail, with respect to theory, computation and application. Several properties of the shape descriptors will be analyzed and compared. We believe this is a crucial step to exploit fully the potential of such approaches in many applications, as well as to identify important areas of future research.

Categories and Subject Descriptors: I.3.5 [**Computational Geometry and Object Modeling**]: Curve, surface, solid, and object representations

General Terms: Theory, Algorithms

Additional Key Words and Phrases: Computational topology, contour tree, shape analysis, Morse complexes, Morse theory, persistent homology, Reeb graph, size theory

Contents

1	Introduction	5
2	Overview	7
2.1	Organization of the survey	10
3	Background notions	11
3.1	Mathematical background	12
3.1.1	Topological spaces	12
3.1.2	Simplicial complexes	12
3.1.3	Cell complexes	13
3.1.4	Homology groups	14
3.1.5	Morse theory	16
3.1.6	Homology of manifolds	19
3.2	Computational background	20
3.2.1	Representation of shape models	20
3.2.2	Discretization of differential concepts in a computational setting	21
4	Morse and Morse-Smale complexes	25
4.1	Theoretical aspects	26
4.2	Computational aspects	30
4.2.1	Computing ascending and descending Morse complexes . . .	30
4.2.2	Computing a Morse-Smale complex	34
4.3	Applications	39
5	Contour trees	40
5.1	Theoretical aspects	41
5.2	Computational aspects	44
5.3	Applications	50
6	Reeb graphs	51
6.1	Theoretical aspects	51
6.2	Computational aspects	53
6.3	Applications	60
7	Size theory	63
7.1	Theoretical aspects	63
7.2	Computational aspects	68
7.2.1	Choosing the size pair	68
7.2.2	Computing $\ell_{(S,f)}$	68
7.3	Applications	70
8	Persistent homology	71
8.1	Theoretical aspects	72
8.2	Computational aspects	74
8.3	Applications	76

9 Morse shape descriptor	78
9.1 Theoretical aspects	78
9.2 Computational aspects	79
9.3 Applications	79
10 Discussion	79
10.1 Overall comparison and general remarks	80
10.2 Expressiveness of shape descriptors	82
10.3 Suitability for applications	83
11 Conclusions and future emerging methods	86

1. INTRODUCTION

Digital shapes are becoming an important media for exchanging information in many scientific and applied scenarios. By *digital shape* we mean any multi-dimensional media that is primarily characterized by form or spatial extent in a space of two, three or higher dimension. Digital shapes are focal resources in many scientific domains, e.g. molecular surfaces, CT data, force fields, mechanical parts, and they populate virtual environments in advanced scientific simulations, as well as in emerging edutainment applications.

Computer Graphics and Computer Vision are the two most representative disciplines dealing with digital shapes. Their major role is in the study of basic models and methods for representing, generating and analyzing shapes. At the beginning, Computer Graphics mostly focused on solving basic problems related to representation issues [Requicha 1980; Mäntylä 1988]. In Computer Vision, less emphasis has been put on representation issues, as in this field digital representations are generally limited to the pixel-, or voxel-based encoding of objects acquired from the real world. On the other hand, researchers in Computer Vision introduced the fundamental idea of using compact representations of shapes, namely *shape descriptors*, and addressed issues related to analyzing, understanding and recognizing objects.

More recently, we have seen a gradual shift of research interests from methods to represent shapes towards methods to *describe* shapes in Computer Graphics as well. While a digital model, either pixel- or vector-based, is a digital representation that is quantitatively similar to an object, its description is only qualitatively similar. The distinction between *representation* and *description* can be expressed as follows [Nackman 1984]:

an object representation contains enough information to reconstruct (an approximation to) the object, while a description only contains enough information to identify an object as a member of some class.

The representation of an object is thus more detailed and accurate than a description, but it does not necessarily contain any high-level information on the shape of the object explicitly. The description is more concise and conveys an elaborate and composite view of the object class.

Shape analysis and understanding are basic tools for constructing object descriptions, as the processes aiming at detecting the main features of a given shape and their configuration. Shape understanding has been successfully approached in specific application domains with a well-established formalization of features, such as, for example, the manufacturing context. The problem of describing shapes in general is highly complex, since the definition of features is intrinsically vague in free-form generic shapes. Shape understanding has been approached using a variety of methods, in both pixel-based and vector-based domains. Several examples have been proposed in the literature, ranging from segmentation methods to skeletonization techniques, to the computation of global shape descriptors. Among the many references in the field of image processing, we may cite some survey papers and books that can serve as an entry point in the literature [Besl and Jain 1985; Pavlidis 1995; Dryden and Mardia 1998; Loncaric 1998]. In the last decade, we have seen a considerable growth of shape analysis methods also in Computer Graphics,

where shape segmentation is a key ingredient in many shape manipulation processes [Shamir 2006], and shape descriptors are crucial for 3D search and retrieval [Veltkamp and Hagendoorn 2001; Tangelder and Veltkamp 2004; Bustos et al. 2005; Iyer et al. 2005; Del Bimbo and Pala 2006].

In this scenario, this survey focuses on a class of methods that are grounded in Morse theory and that we believe deserve specific attention. Intuitively, methods based on Morse theory analyze a given shape by studying either the properties, the configuration, or the evolution of the critical points of a real-valued function f defined on the shape itself. Critical points are associated to the *features* of interest that one wishes to extract, and the configuration, or evolution, of these critical points captures a global description of the shape. In this context, we focus on methods studying the configuration of critical points on the shape boundary (Morse and Morse-Smale complexes), methods studying the evolution of the level sets of f (contour trees and Reeb graphs), and methods studying the evolution, or growth, of the lower level sets of f (size theory, persistent homology and Morse shape descriptors).

There are two motivations for this choice. The first is that methods based on Morse theory provide general results with respect to the specific selection of the function f : *different* functions can be used to analyze different aspects of the same shape, according to the features that one wishes to characterize or to the invariants that the shape description should be able to capture. Therefore, Morse theory defines a general framework for shape characterization, because different properties can be studied using different functions. It is very important to provide a generalized and unifying view of this class of methods, showing how and why they all adhere to the same conceptual framework, and therefore give the same theoretical guarantees and results.

The second motivation, which induces our selection of the survey focus, is related to the interesting combination of theoretical aspects, discretization strategies and computational issues that these methods imply. They have been developed over the years with a specific target domain, and it is not trivial to systematize and understand the links, similarities and differences between them. Moreover, different terms have been used to denote the same mathematical constructs, which often overwhelm the understanding of the underlying common framework. Bearing in mind that the Computer Graphics community is becoming highly inter-disciplinary, we think it is timely to propose a discussion on this class of methods, attempting to provide a clear vision of what has been developed so far, from a theoretical and computational perspective. We believe this is a crucial step to fully exploit the potential of such approaches in several application domains, as well as to identify important areas of future research.

We are aware that a survey of the methods based on Morse theory does not cover the whole spectrum of shape analysis methods. Also, there are a number of shape descriptors that are not directly defined using Morse theory, but closely related to it, at least from the point of view of the extraction procedure. A notable example is the class of methods that use the *distance function* for shape analysis [Jones et al. 2006]. In this case, the function is actually not applied to the shape itself, but used to define a scalar field of distances between points in space and the shape itself.

Similarly, the critical points of the distance field [Buchin et al. 2007] are related to the main shape features and define the so-called *medial axis*, probably among the best-known shape descriptors. A recent survey of techniques based on the distance field appeared in [Jones et al. 2006], and the theoretical properties of the medial axis have been the topic of several papers in the literature [Aurenhammer 1991; Wolter 1992; Giblin and Kimia 2004]. It should be noted that approaches based on distance functions do not properly belong to the class of methods that the survey addresses: the unifying paradigm of the techniques surveyed is indeed their parametric nature with respect to the mapping function f . Methods that use the distance functions, indeed, provide one single type of description, where the features are those characterized by the distance field itself. Conversely, in any of the methods surveyed, f can vary, and changing f actually means changing the properties, or features, by which the shape will be described.

While an exhaustive review of the literature on distance transform is beyond the scope of our work, we discuss in this survey some aspects of methods for shape analysis based on the use of distance fields, but limited to the discussion of recent works that present a formulation of the distance transform in terms of the conceptual elements of Morse theory.

Summarizing, this survey attempts to classify and compare methods based on Morse theory, characterized by a common theoretical framework that is developed for a whole *class* of real functions, rather than for a single function. The survey aims to be viable for a varied audience, from experienced researchers in computational topology to less experienced readers. The methods are reviewed at a theoretical and implementation level, and we also describe the geometric intuition behind the various techniques. In Section 2, we present a detailed overview of the survey and explain the strategy of the presentation, which helps the reader to find the reading path appropriate to his/her level of expertise.

2. OVERVIEW

Differential topology, and specifically Morse theory, provides a suitable setting for formalizing and solving several problems related to shape analysis [Milnor 1963]. In the late 1990s, the potential of this approach was fully recognized by a group of researchers in Computer Graphics who started a new research area called *computational topology*. This term was first introduced in [Vegter and Yap 1990; Vegter 1997; Dey et al. 1999], and denotes research activities involving both mathematics and computer science in order to formalize and solve topological problems in computer applications, and to study the computational aspects of problems with a topological flavour.

The intuition behind Morse theory is that of combining the topological exploration of a shape S with quantitative measurements of its geometrical properties provided by a mapping function f , defined on S [Milnor 1963]. Integrating the classifying power of topology with the differentiating power of geometry enables us to extract information about shapes at different levels, taking into account global as well as local shape properties. Therefore, we focus on methods for shape understanding that find their roots in Morse theory, ranging from simple to complex shapes, from single to arbitrary mapping functions, and from unstructured to alge-

braically structured sets of descriptors.

Broadly speaking, the methods discussed in the survey can be divided into three groups: methods studying the configuration of critical points on the shape boundary (Morse and Morse-Smale complexes), methods studying the evolution of the level sets of f (contour trees and Reeb graphs), and methods studying the evolution, or growth, of the lower level sets of f (size theory, persistent homology and Morse shape descriptors). The groups of methods discussed reflect, to different extents, the modularity of the approaches based on Morse theory, especially from the point of view of the applications they have been traditionally designed for.

Morse and Morse-Smale complexes were introduced in Computer Graphics for the analysis of two-dimensional scalar fields, but recently their use has also been extended to handle generic 3D shapes. These structures provide a view of shape properties from the perspective of the *gradient* of the mapping function. Intuitively, the aim is to describe the shape by decomposing it into cells of uniform behaviour of the gradient flow. The decomposition can be interpreted as having been obtained by a network on the surface that joins the critical points of the mapping function f through lines of steepest ascent, or descent, of the gradient.

The theory behind Morse and Morse-Smale complexes holds in general for n -manifolds, and is also related to the theory of dynamical systems. These two views are clearly reflected in the literature and a considerable number of techniques have been developed to extract critical points and lines, especially for terrain surface modelling and analysis.

Contour trees have been used mainly to study the shape of scalar fields, and no distinction is made between the shape and the function used to analyze it: both coincide with the scalar field itself. Contour trees describe the shape of a scalar field f by analyzing the evolution of its level sets, as f spans the range of its possible values: components of level sets may appear, disappear, join, split, touch the boundary or change genus. The contour tree stores this evolution and provides a compact description of the properties and structure of the scalar field. Contour trees, however, could, in principle, be defined for any shape with any mapping function, and the theory behind them is general. Contour trees, in all their variants, are discussed with emphasis on the methods developed in Computer Graphics and with pointers to similar structures defined in Computer Vision.

The generalization of a contour tree is given by Reeb graphs, even if their definition is slightly different, as presented in the literature. While the definition and use of contour trees developed mainly as an answer to computational issues, Reeb graphs have a more theoretical nature. Their definition and theoretical study date back to 1946, thanks to the research work of a French mathematician, George Reeb. With respect to the modularity of Morse theory, Reeb graphs are the first example of a fully modular framework for studying the shape of a manifold: here the shape exists by itself and the function used to study it can be arbitrarily chosen. In recent years, Reeb graphs have become popular in Computer Graphics as a tool for studying shapes through the evolution and arrangement of the level sets of a real function defined over the shape. Reeb graphs effectively code the shape, both from a topological and geometrical perspective. While the topology is described by the connectivity of the graph, the geometry can be coded in a variety of different ways,

according to the type of applications the Reeb graph is devised for.

Contour trees and Reeb graphs are frequently associated, in the literature, with the concept of skeletal graphs or centerline skeletons. By coding the barycentres of each level set, it is indeed very easy to trace a kind of centerline spanning the volume enclosed by the shape. Centerline skeletons are very popular in Computer Graphics and Vision, and are, in principle, related to the medial axis transformation, in the sense that they represent an effective way of reducing a complex 3D shape to a simple one-dimensional geometric abstraction [Cornea et al. 2005; Lam et al. 1992]. While the notion of centerline skeleton, per se, does not fall within the ambit of this survey, we discuss some recent results in that specific field in the section devoted to Reeb graphs, in order to point out their similarities in theory and applications.

Besides the possibility of adopting different functions for describing shapes, at a higher level of abstraction, the modularity of the approach based on Morse theory can be extended to the choice of the space used to represent the shape, or phenomenon, under study. The third group of methods reflects this higher degree of modularity, and it is concerned with methods allowing one or more real functions to be defined on spaces associated with the shapes under study. Size theory and persistent homology theory fall into this last group and are characterized by the possibility of varying the space underlying the shape and the real functions defined on it. Furthermore, an extensive use of algebraic structure characterizes these techniques.

Size theory has been developed since the beginning of the 1990s with the idea of defining a suitable mathematical setting for the problem of shape comparison, and, as such, it relies on the adoption of the natural pseudo-distance between size pairs as the key for shape comparison, and the size functions, the size homotopy groups and the size functor for shape description and discrimination. A common property is that shapes are studied by varying the underlying space and the real functions defined on it.

Persistent homology follows a similar approach, but introduces another paradigm, *persistence*, which is based on growing a space (i.e., the shape) incrementally, and analyzing the topological changes that occur during this growth. The occurrence and placement of topological events (e.g., creation, merging, cancellation of the connected components of the lower level sets) within the history of this growth characterize the shape. Persistent homology aims to define a scale of the *relevance* of these topological events, characterizing the features of the shape. The main assumption is that longevity is equivalent to significance. In other words, a significant topological attribute must have a long life-time in a growing complex.

Also, another contribution pertaining to this class is discussed, the Morse shape descriptor, which differs from the other two, as it makes use of the theory of relative homology groups to define a shape description.

In the first works related to Morse and Morse-Smale complexes, and to contour trees, the shape was entirely encoded by the scalar field f , defined over a fixed domain. For this reason, changing f would mean studying a completely different object. Reeb graphs enable a shape S to be analyzed according to the behaviour of a chosen function f , defined on S . Different choices of the function yield insights on S from different perspectives. Finally, size theory and persistent homology theory

allow a user to model the phenomenon by adopting suitable choices for both the space to be associated to S and the function f .

To summarize, this survey attempts to provide an up-to-date description and analysis of the major methods developed within the framework of Morse theory, and to clarify the mathematics pertaining to each group of methods. For each set of techniques, we discuss different algorithms and compare them, based on the type of analysis, computational complexity and application context. The relationship between the different types of descriptors is discussed as well, in an overall comparison section.

Since a detailed discussion of these methods calls for a comparison of theoretical aspects, discretization strategies and computational issues, there is a substantial amount of mathematical background necessary in order to discuss the methods, ranging from basic notions related to cell complexes to Morse theory, and even to homology theory. To provide a self-contained study, we preferred to include a summary of background mathematical notions, which the expert reader can skip completely. Also, since discretization strategies play a fundamental role in the way the results stated in a smooth context can be achieved in discrete ones, we have included a short discussion of the main discretization options for basic concepts, such as the definition of critical points and Morse functions for piece-wise linear shapes.

2.1 Organization of the survey

The survey is organized as follows.

Section 3 discusses the mathematical and computational background necessary to understand the theories we are using. The mathematical overview introduces the most important definitions and results that are used throughout the survey. The discussion of computational aspects concerns the most relevant discretization issues we are faced with when transferring theoretical frameworks to discrete digital contexts. Specifically, the definition of critical points for discrete models is discussed, and theories are presented which rely on this basic notion to extract information about shapes.

Since not all the notions described are equally necessary for the comprehension of the various sections of the survey, and the reader may already be familiar with some of them, in Table I and II we provide a kind of *guide* to the background notions reviewed in Section 3.

After this, each group of methods finalized to the computation of a specific descriptor is discussed in a separate section, organized into the theoretical, computational and application aspects: Morse and Morse-Smale decompositions in Section 4, contour trees in Section 5, Reeb graphs in Section 6, size theory in Section 7, persistent homology in Section 8, and, finally, Morse shape descriptors are reviewed in Section 9. The methods surveyed within each section are compared, based on their properties, mainly from a computational point of view.

The presentation of the different techniques is followed by an overall comparison, provided in Section 10. The methods are discussed, pointing out the similarities and differences at the level of their combinatorial, structural or algebraic properties. This comparison also highlights the differences in terms of loss of information of the description with respect to the representation, context of applicability (e.g., type

Subsection	Concepts reviewed
3.1.1 Topological spaces	topological space, open set, continuous function, homeomorphism, homotopy type
3.1.2 Simplicial complexes	simplex, face, oriented simplex, simplicial complex, subcomplex, star, link, r -skeleton, abstract simplicial complex
3.1.3 Cell complexes	cell, attaching map, cell complex
3.1.4 Homology groups	chain, boundary operator, cycle, boundary, homology class, Betti numbers, Euler characteristic
3.1.5 Morse theory	manifold, critical point, critical value, Hessian, non-degenerate critical point, index, Morse function, monkey saddle, level set, lower level set
3.2 Computational background	regular grids, critical points on polyhedra, discrete Morse theory, general functions, lower link stratified Morse theory, discrete exterior calculus

Table I. Summary of the concepts reviewed Section 3.

Topics	Mathematical requisites
Morse Smale complexes	simplicial complexes, cell complexes, Morse theory
Contour trees	Morse theory
Reeb graph	simplicial complexes, Morse theory
Size theory	topological spaces, homology groups, Morse theory
Persistent homology	simplicial complexes, homology groups, Morse theory
Morse shape descriptor	homology groups, Morse theory

Table II. Summary of the mathematical concepts pertaining to each group of methods surveyed.

of discrete model or dimensionality of the shape) and properties of the descriptors.

To the extent of our knowledge, this is the first attempt to systematize the work done in this area over the last decade.

Finally, the concluding remarks discuss open problems and forthcoming areas of development in the field are drawn in Section 11.

3. BACKGROUND NOTIONS

This section contains an overview of the main background notions to which the methods surveyed refer. We do not claim to be exhaustive, but we attempt to list concepts and theories that are frequently encountered in the portion of the literature related to the topics of this paper. References are provided to direct the interested reader to more detailed descriptions of the subjects.

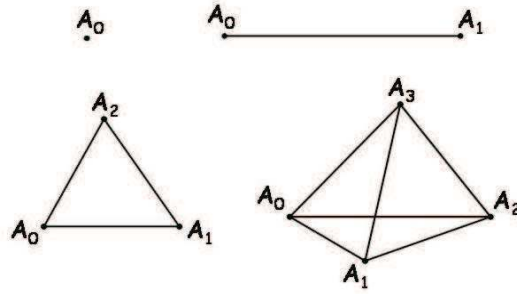


Fig. 1. Examples of 0-, 1-, 2- and 3-simplices.

3.1 Mathematical background

This first section provides a review of the theoretical concepts underlying the methods presented in this survey. See Tables I and II for a summary of the content.

3.1.1 Topological spaces. Topological spaces are mathematical structures that allow the generalization of concepts such as closeness, limits, connectedness, or continuity, from the Euclidean space \mathbb{R}^n to arbitrary sets of points. This is achieved using relationships between sets, rather than distances between points. A detailed treatment of this subject can be found in [Willard 1970].

A *topological space* is a set X on which a *topology* is defined, i.e., a collection of subsets of X which are called the *open* subsets of X and which satisfy the following axioms: both X itself and the empty subset must be among the open sets, all unions of open sets are open, and the intersection of two open sets is open.

A function between topological spaces is said to be *continuous* if the inverse image of every open set is open. A *homeomorphism* is a bijection that is continuous and whose inverse is also continuous. Two spaces are said to be *homeomorphic* if there exists a homeomorphism between them. From the viewpoint of topology, homeomorphic spaces are essentially identical.

A weaker relation type between topological spaces is captured by the notion of homotopy type. Two spaces X and Y are of the same *homotopy type* if one can find two continuous maps $f : X \rightarrow Y$ and $g : Y \rightarrow X$ such that the map compositions $g \circ f$ and $f \circ g$ are not necessarily equal to the identity maps on X and Y respectively, but are homotopic to them, i.e. they can be reduced to identity maps by continuous deformations.

3.1.2 Simplicial complexes. In order to construct topological spaces, one can take a collection of simple elements and glue them together in a structured way. Probably the most relevant example of this construction is given by simplicial complexes, whose building-blocks are called simplices.

A *k-simplex* Δ^k in \mathbb{R}^n , $0 \leq k \leq n$, is the convex hull of $k+1$ affinely independent points A_0, A_1, \dots, A_k , called *vertices*. Figure 1 shows the simplest examples of simplices: Δ^0 is a point, Δ^1 an interval, Δ^2 a triangle (including its interior), Δ^3 a tetrahedron (including its interior).

A *k-simplex* can be oriented by assigning an ordering to its vertices: two or-

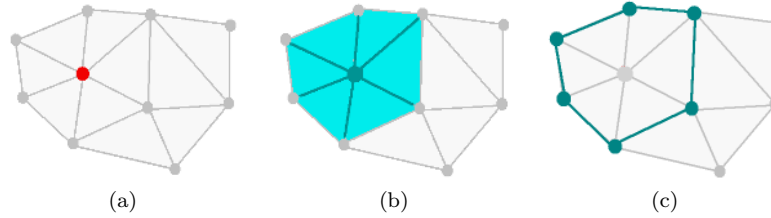


Fig. 2. (a) A 0-simplex, (b) its star and (c) its link.

derings of the vertices that differ by an even permutation determine one and the same orientation of the k -simplex. In this way, each k -simplex with $k > 0$ can be given a positive or a negative orientation. The *oriented k -simplex* with ordered vertices (A_0, A_1, \dots, A_k) is denoted by $[A_0, A_1, \dots, A_k]$, whereas the k -simplex with opposite orientation is denoted by $-[A_0, A_1, \dots, A_k]$.

A *face* of a k -simplex Δ^k is a simplex whose set of vertices is a non-empty subset of the set of vertices of Δ^k . A *finite simplicial complex* can now be defined as a finite collection of simplices, together with their faces of any dimension, and simplices can meet only along a common face. A concrete example of a simplicial complex is given by triangulated surfaces, where the vertices, edges and faces of the triangulation are 0-, 1- and 2-simplices, respectively. The *dimension* of a simplicial complex is the maximum dimension of its simplices.

A *subcomplex* of a complex K is a simplicial complex whose set of simplices is a subset of the set of simplices of K . Particular instances of subcomplexes are given by the star and the link of a simplex. Given a simplex Δ , the *star* of Δ is the union of all the simplices containing Δ . The *link* of Δ consists of all the faces of simplices in the star of Δ that do not intersect Δ . The concepts of star and link are illustrated in Figure 2 for the case of a 0-simplex. Other useful subcomplexes of a complex K are its skeletons: for $0 \leq r \leq \dim(K)$, the r -*skeleton* of K is the complex of all the simplices of K whose dimension is not greater than r .

Note that it is also possible to define an *abstract simplicial complex* without using any geometry, as a collection \mathcal{A} of finite non-empty sets such that if A is any element of \mathcal{A} , so is every non-empty subset of A . Simplicial complexes can be seen as the geometric realization of abstract simplicial complexes.

For more details on simplicial complexes please refer to [Munkres 2000].

3.1.3 Cell complexes. Compared to simplicial complexes, cell complexes allow the construction of a more general class of spaces, while still having a combinatorial nature. The name suggests the idea of a topological space constructed from basic building-blocks, called *cells*, that generalize the concept of simplices; these cells are glued together via *attaching maps*. A λ -*cell* e^λ corresponds to the closed unit ball $B^\lambda = \{x \in \mathbb{R}^\lambda \mid \|x\| \leq 1\}$ of dimension λ . Attaching the cell e^λ to a space Y by the continuous map $\varphi: S^{\lambda-1} \rightarrow Y$, with $S^{\lambda-1} = \{x \in \mathbb{R}^\lambda \mid \|x\| = 1\}$ the boundary of B^λ , requires taking $Y \cup B^\lambda$, where each point $x \in S^{\lambda-1}$ is identified with the point $\varphi(x) \in Y$. The space so obtained is denoted by $Y \cup_\varphi e^\lambda$. It is important to note that different attaching maps φ can lead to different spaces.

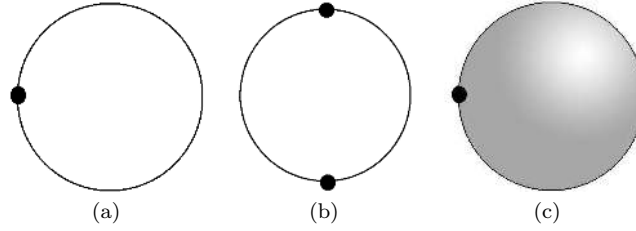


Fig. 3. Examples of cell complexes. (a) Circle built starting with a point and attaching a 1-cell. (b) Circle built starting with two points and attaching two 1-cells. (c) Sphere built starting with a point and attaching a 2-cell.

The space X obtained by subsequently attaching finitely many cells is a finite *cell complex*. This means that there exists a finite nested sequence $\emptyset \subset X_0 \subset X_1 \subset \dots \subset X_k = X$ such that, for each $i = 1, 2, \dots, k$, X_i is the result of attaching a cell to X_{i-1} . Further details can be found in [Greenberg and Harper 1981].

Examples of cell complexes are given in Figure 3. In Figure 3(a) and (b) the same circle is constructed through cell adjunction in two different ways. In (a), we start with a 0-cell, i.e. a point, and we attach a 1-cell via the attaching map that identifies the boundary of B^1 with the starting point. In (b), two 1-cells are attached to two 0-cells. In Figure 3(c) the sphere is obtained by attaching a 2-cell directly to a 0-cell, that is, identifying the boundary of B^2 with a point.

3.1.4 Homology groups. The approach adopted by algebraic topology is the translation of topological problems into an algebraic language, in order to solve them more easily. A typical case is the construction of algebraic structures to describe topological properties, which is the core of homology theory, one of the main tools of algebraic topology.

The homology of a space is an algebraic object which reflects the topology of the space, in some sense counting the number of holes. The *homology* of a space X is denoted by $H_*(X)$, and is defined as a sequence of groups $\{H_q(X) : q = 0, 1, 2, \dots\}$, where $H_q(X)$ is called the *q-th homology group* of X . In the literature there are various types of homologies [Spanier 1966]; the one we are addressing here is (integer) *simplicial homology*, which is strictly related to the concept of simplicial complex.

Let K be a simplicial complex in \mathbb{R}^n . For each $q \geq 0$, a *q-chain* of K is a formal linear combination $\sum_i a_i \Delta_i$, of oriented q -simplices Δ_i , with integer coefficients a_i . Two q -chains $a = \sum_i a_i \Delta_i$ and $b = \sum_i b_i \Delta_i$ are added componentwise, that is to say $a + b = \sum_i (a_i + b_i) \Delta_i$. We denote by $C_q(K)$ the group of q -chains of K with respect to the addition; for q larger than n or for q smaller than 0, we set $C_q(K)$ equal to the trivial group. On the group $C_q(K)$, we can define the *boundary operator* $\partial_q : C_q(K) \rightarrow C_{q-1}(K)$. This is defined as the trivial homomorphism if $q \leq 0$, while for $q > 0$ it acts on each q -simplex via:

$$\partial_q[A_0, A_1, \dots, A_q] = \sum_{i=0}^q (-1)^i [A_0, A_1, \dots, A_{i-1}, \hat{A}_i, A_{i+1}, \dots, A_q]$$

where $[A_0, A_1, \dots, A_{i-1}, \hat{A}_i, A_{i+1}, \dots, A_q]$ is the $(q-1)$ -simplex obtained by elim-

inating the vertex A_i ; the boundary map ∂_q extends by linearity to arbitrary q -chains. In Figure 4, the boundary operator is evaluated on some elementary simplices. The arrows represent the orientation of the simplices. Notice that changing the orientation of the simplices implies a different result for the boundary operator.

A chain $z \in C_q(K)$ is called a *cycle* if the boundary operator sends z to zero, i.e. $\partial_q z = 0$; a chain $b \in C_q(K)$ is called a *boundary* if it is the image, through the boundary operator, of a chain of dimension greater by one, i.e. there exists $c \in C_{q+1}(K)$ such that $b = \partial_{q+1} c$. The sets of cycles and boundaries form two subgroups of $C_q(K)$:

$$Z_q(K) = \{z \in C_q(K) \mid \partial_q z = 0\} = \ker \partial_q$$

$$B_q(K) = \{b \in C_q(K) \mid b = \partial_{q+1} c, \text{ for some } c \in C_{q+1}(K)\} = \text{Im} \partial_{q+1},$$

where \ker and Im denote the kernel and the image of the map, respectively. It holds that $B_q(K) \subseteq Z_q(K)$, since $\partial_q \partial_{q+1} = 0$. The q -th *simplicial homology group* of K is then the quotient group:

$$H_q(K) = Z_q(K) / B_q(K).$$

Specifically, an element of $H_q(K)$ is an equivalence class, called *homology class*, of homologous q -cycles, i.e. cycles whose difference is a boundary. The homology $H_*(K)$ is a topological invariant of K ; it is indeed an invariant of homotopy type.

The rank of $H_q(K)$ is called the q -th *Betti number* of K , and it is a measurement of the number of different holes in the space K . As an example, for three-dimensional data the three Betti numbers β_0 , β_1 and β_2 count the number of connected components, tunnels and voids, respectively. The Betti numbers can be used to define a well-known topological invariant, the *Euler characteristic*: $\chi(K) = \sum_{i=0}^n (-1)^i \beta_i(K)$.

A *simplicial map* f between two simplicial complexes K and L (i.e., a function from the set of vertices of K to the set of vertices of L such that the image of any

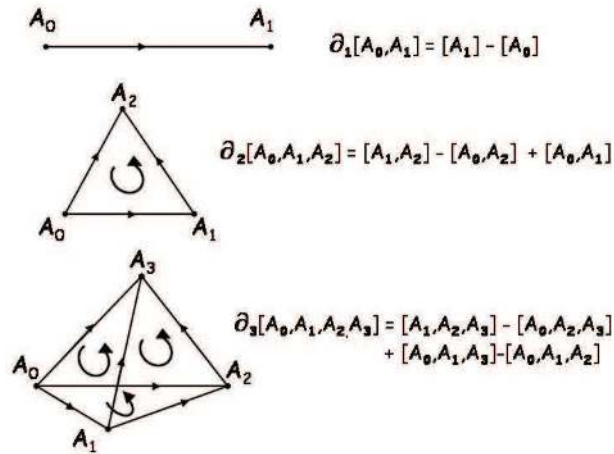


Fig. 4. The boundary operator on elementary q -simplices.

simplex in K is a simplex in L) induces uniquely a set $H_*(f)$ of homomorphisms between the homology groups:

$$H_q(f) : H_q(K) \rightarrow H_q(L)$$

for each degree q . The maps $H_q(f)$ satisfy two elementary properties: (i) the identity map $id_K : K \rightarrow K$ induces the identity map on homology; and (ii) the composition $g \circ f$ of two maps corresponds to the composition $H_q(g \circ f) = H_q(g) \circ H_q(f)$ of the induced homomorphisms.

3.1.5 Morse theory. Morse theory can be seen as the investigation of the relation between functions defined on a manifold and the shape of the manifold itself. Intuitively, a *manifold* is a topological space that is locally Euclidean, meaning that around every point there is a neighborhood that is topologically the same as the open unit ball in \mathbb{R}^n ; the number n is the *dimension* of the manifold. More formally, the following definition [Hirsch 1997] introduces the definition of manifold:

Definition 3.1. A topological Hausdorff space M is called an *n -dimensional manifold* (*n -manifold*) if there is an open cover $\{U_i\}_{i \in A}$ of M such that for each $i \in A$ there is a map $\varphi_i : U_i \rightarrow \mathbb{R}^n$ which maps U_i homeomorphically onto the open n -dimensional disk $D^n = \{x \in \mathbb{R}^n \mid \|x\| < 1\}$. An *n -manifold with boundary* is a Hausdorff space in which each point has an open neighborhood homeomorphic either to the open disk D^n or the open half-space $\mathbb{R}^{n-1} \times \{x_n \in \mathbb{R} \mid x_n \geq 0\}$.

Each pair (U_i, φ_i) is called a *coordinate map*, or a *chart*. The union of charts $\{(U_i, \varphi_i)\}_{i \in A}$ is called an *atlas* on the manifold M . Let (U_i, φ_i) and (U_j, φ_j) be two arbitrary charts and consider the intersection $U_i \cap U_j$. On this intersection we can consider the restriction $\bar{\varphi}_i = \varphi_i|_{U_i \cap U_j}$ and $\bar{\varphi}_j = \varphi_j|_{U_i \cap U_j}$. Since the composition of homeomorphisms is a homeomorphism, the maps $\varphi_{i,j} = \bar{\varphi}_j \circ \bar{\varphi}_i^{-1} : \varphi_i(U_i \cap U_j) \rightarrow \varphi_j(U_i \cap U_j)$ are homeomorphisms, and are called *transition functions*, or *gluing functions*, on a given atlas. A manifold is said to be *smooth* if all transition functions $\varphi_{i,j}$ are smooth.

The key feature in Morse theory is that information on the topology of the manifold is derived from the information about the critical points of real functions defined on the manifold. Let us first introduce the definition of Morse function, and then state the main results provided by Morse theory for the topological analysis of smooth manifolds, such as surfaces. A basic reference for Morse theory is [Milnor 1963], while details about notions of geometry and topology can be found, for example, in [Hirsch 1997].

Let M be a smooth compact n -dimensional manifold without boundary, and $f : M \rightarrow \mathbb{R}$ a smooth function defined on it. Then, a point p of M is a *critical point* of f if we have

$$\frac{\partial f}{\partial x_1}(p) = 0, \frac{\partial f}{\partial x_2}(p) = 0, \dots, \frac{\partial f}{\partial x_n}(p) = 0,$$

with respect to a local coordinate system (x_1, \dots, x_n) about p . A real number is a *critical value* of f if it is the image of a critical point. Points (values) which are not critical are said to be *regular*. A critical point p is *non-degenerate* if the

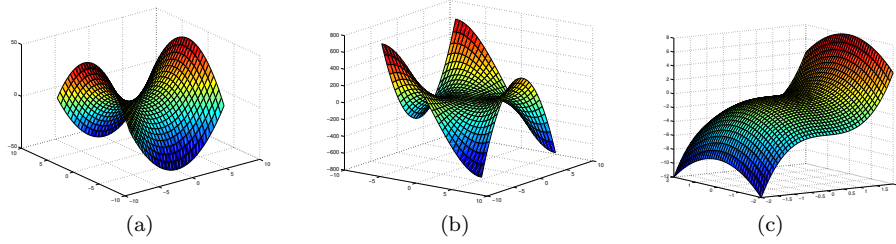


Fig. 5. (a) The graph of $f(x, y) = x^2 - y^2$. The point $(0, 0)$ is a non-degenerate critical point. (b) and (c) The graphs of $f(x, y) = x^3 - 3xy^2$ (a “monkey saddle”) and $f(x, y) = x^3 - y^2$. In both cases the point $(0, 0)$ is a degenerate critical point.

determinant of the *Hessian* matrix of f at p

$$H_f(p) = \left(\frac{\partial^2 f}{\partial x_i \partial x_j} (p) \right)$$

is not zero; otherwise the critical point is *degenerate*. Figure 5 shows some examples of non-degenerate and degenerate critical points. For a non-degenerate critical point p , the number of negative eigenvalues of the Hessian $H_f(p)$ of f at p determines the *index* of p , denoted by $\lambda(p)$.

We say that $f : M \rightarrow \mathbb{R}$ is a *Morse function* if all its critical points are non-degenerate. The Morse Lemma states that the function f looks extremely simple near each non-degenerate critical point p . Indeed, we can choose appropriate local coordinates (x_1, \dots, x_n) around p , in such a way that f has a quadratic form representation: $f(x_1, \dots, x_n) = f(p) - \sum_{i=1}^{\lambda(p)} x_i^2 + \sum_{i=\lambda(p)+1}^n x_i^2$. Therefore, intuitively, the index of a critical point is the number of independent directions around the point in which the function decreases. For example, on a 2-manifold, the indices of minima, saddles, and maxima are 0, 1, and 2, respectively.

An important property is that a Morse function defined on a compact manifold admits only finitely many critical points, each of which is isolated. This means that, for each critical point p , it is always possible to find a neighborhood of p not containing other critical points.

Topological information about M is captured by the changes of the level sets and the lower level sets of M relative to the function f . The *level set* of f corresponding to the real value t is the set of points $V_t = \{p \in M \mid f(p) = t\} = f^{-1}(t)$; t is called an *isovalue* of f . The *lower level set* is given by $M_t = \{p \in M \mid f(p) \leq t\} = f^{-1}((-\infty, t])$.

We begin by studying how the lower level set M_t changes as the parameter t changes (Figure 6(a)). Morse theory states that the topology of M_t stays unchanged (formally, the homotopy type is preserved) as the parameter t goes through regular values of f , while changes occur when t passes through a critical value. More precisely, the following theorem holds:

THEOREM 3.2. *Let a, b be real numbers such that $a < b$ and the set $f^{-1}([a, b])$ contains no critical points for f . Then M_a and M_b have the same homotopy type.*

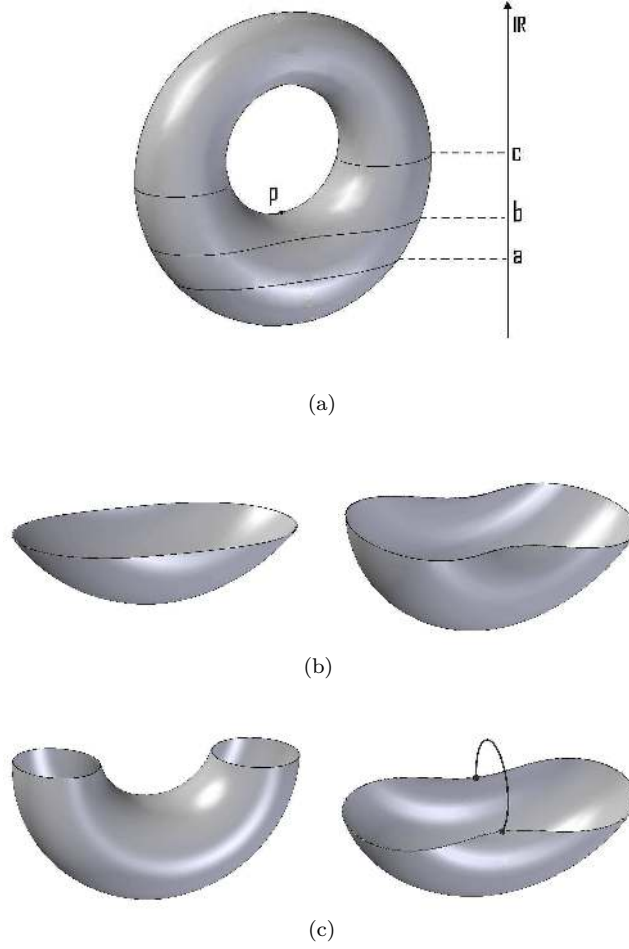


Fig. 6. (a) A manifold M and three lower level sets M_a , M_b , M_c , with respect to the height function. (b) There are no critical points in $f^{-1}([a, b])$: M_a (left) and M_b (right) are diffeomorphic. (c) The passage through the critical point p of index 1 causes a topological change: M_c (left) has the same homotopy type as M_b with a 1-cell attached (right).

Actually, a stronger result holds in this case stating that M_a and M_b are diffeomorphic, namely, there is a differentiable and invertible function between M_a and M_b , whose inverse is also differentiable (see Figure 6(b)).

THEOREM 3.3. *Let p be a critical point of f with index λ and let $f(p) = c$. Then, for each ε such that $f^{-1}([c - \varepsilon, c + \varepsilon])$ contains no critical points other than p , the set $M_{c+\varepsilon}$ has the same homotopy type of the set $M_{c-\varepsilon}$ with a λ -cell attached:*

$$M_{c+\varepsilon} \cong M_{c-\varepsilon} \cup_{\varphi_p} e^\lambda.$$

According to the definitions in Section 3.1.3, the *attaching map* φ_p identifies each point $x \in S^{\lambda-1}$ with the point $\varphi_p(x) \in M_{c-\varepsilon}$ (see Figure 6(c)).

In order to study the changes in the level sets $V_t = f^{-1}(t)$, an approach to Morse theory based on the attaching of handles [Milnor 1965], rather than cells, can be used, as in [Gramain 1971] for the case of surfaces. When f is defined on a surface, if t is a regular value for f then V_t , if not empty, is the union of finitely many smooth circles. Moreover, if a, b are real numbers such that $a < b$, then

- (1) if the set $f^{-1}([a, b])$ contains no critical points for f , then V_a and V_b are diffeomorphic;
- (2) if the set $f^{-1}([a, b])$ contains only one critical point of index 0 for f , then V_b is the union of V_a with a circle;
- (3) if the set $f^{-1}([a, b])$ contains only one critical point of index 2 for f , then V_b is diffeomorphic to V_a without one of its circles;
- (4) if the set $f^{-1}([a, b])$ contains only one critical point of index 1 for f , then the number of connected components of V_b differs from that of V_a by -1 , 0 or 1 depending on the attaching map.

In case (4), the difference in the number of connected components is non-zero if the handle (in this case, the strip $[0, 1] \times [0, 1]$) is attached without twists (or with an even number of twists), while it is 0 if there is an odd number of twists. The presence of an odd number of twists implies that the surface is non-orientable. Therefore, when the surface is embedded in \mathbb{R}^3 , V_a and V_b necessarily have a different number of connected components.

3.1.6 Homology of manifolds. Morse theory asserts that changes in the topology of a manifold endowed with a Morse function occur in the presence of critical points; since smooth manifolds can be triangulated as simplicial complexes [Cairns 1934] and a Morse function can be discretized on simplices, those changes in the topology can be interpreted in terms of homology. Thus, we have the following description of the homology of a manifold M [Greenberg and Harper 1981]:

THEOREM 3.4. *Let a, b be real numbers such that $a < b$ and $f^{-1}([a, b])$ contains only one critical point p of f , of index λ , and let φ_p be the attaching map of the λ -cell corresponding to p . Then:*

$$(a) \text{ if } k \neq \lambda \text{ and } k \neq \lambda - 1 \text{ then } H_k(M_b) \cong H_k(M_a)$$

$$(b) H_{\lambda-1}(M_b) \cong H_{\lambda-1}(M_a) / \text{Im} H_{\lambda-1}(\varphi_p)$$

$$(c) H_{\lambda}(M_b) \cong H_{\lambda}(M_a) \oplus \text{Ker} H_{\lambda-1}(\varphi_p)$$

This means that, depending on the attaching map φ_p , only the homology degrees $\lambda - 1$ and λ can be affected by the adjunction of a λ -cell. In particular, the Betti number $\beta_{\lambda-1}$ can decrease and β_{λ} can increase. For example, in the case shown in Figure 6, when we pass through the critical point p according to the increasing value of f , a tunnel is created, and β_1 increases from the value 0 to the value 1.

This characterization gives a hint of the ideas underlying Reeb graphs, size theory, persistent homology theory and Morse shape descriptors.

3.2 Computational background

In this section, we present an overview of some of the issues arising when applying concepts and theories defined in the continuum to a computational setting. Specifically, we discuss approaches to the *representation* of shapes in a digital context and to the definition of *differential concepts* and related properties in the discrete, with emphasis on the definition of critical points. Finally, we survey approaches for handling *differential operators*.

3.2.1 Representation of shape models. Generally speaking, by shape we mean any phenomenon in the real or virtual world which exhibits a geometric nature, that is, is characterized by spatial extension (e.g. positions in some space, dimensionality, size). Thus, we have shapes acquired from existing objects (e.g. images, 3D scans), shapes that are defined by sampling mathematical surfaces (e.g. implicit or algebraic surfaces), or shapes that are defined by the behavior of a physical quantity (e.g. temperature, or viscosity of a fluid).

From a mathematical point of view, a shape can be modeled as a topological space in \mathbb{R}^n , and usually we consider shapes in \mathbb{R}^3 when dealing with applications in Computer Graphics. Often these shapes are abstracted as manifolds embedded in \mathbb{R}^n , usually orientable and smooth.

This first stage in the modeling pipeline is known as *mathematical modeling* [Requicha 1980] and consists of formulating the basic properties that an abstract computational model should have. In most of the methods discussed (e.g. contour trees, Reeb graphs, Morse decompositions), shapes are abstracted as manifolds, but we also discuss methods that assume the shape to be a more general topological space (e.g. in the context of size theory).

The next step is the selection of a computational representation scheme consistent with the mathematical model. Cell decompositions are the most common geometric model used in computer graphics and CAD/CAM [Mäntylä 1988; Mortenson 1986]. From a historical perspective, the first type of model used was the *wireframe* model, which consists of the representation of edge curves and points on the object boundary. This has been developed further into *surface* models, that provide the full representation of the geometry of the boundary of a 3D shape, and by *solid* models, that encode a shape as a composition of volumes.

Shapes defined as graphs of scalar fields are the simplest kind of shapes studied in the methods surveyed. Formally, a scalar field is defined by any real-valued function $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Generally, a d -dimensional scalar field Γ is denoted by the pair (D, f) and the values of f describe a physical phenomenon measured at a discrete set of points in D . In all methods surveyed, the domain D is assumed to be simply-connected, that is, f is considered only on a domain $D \subset \mathbb{R}^n$ corresponding to a d -dimensional interval in \mathbb{R}^n , where $d \leq n$. As a consequence, the shapes defined by f have genus $g = 0$.

Representing a scalar field Γ in a computational setting implies the discretization of the domain D as well as the discretization of the range of f .

The domain is generally partitioned through a d -dimensional simplicial or cell complex. Thus, a model for a scalar field is called *simplicial* if the domain decomposition is a simplicial complex, while it is called *regular* if the domain is discretized through a *regular grid*, i.e., a cell complex in which all the cells are hypercubes.

These cells are known as pixels in 2D and voxels in 3D, when the scalar field describes a two-dimensional or three-dimensional image [Klette and Rosenfeld 2004]). In a regular model the value of the field can be associated either with vertices or with d -cells, and it is interpolated at other locations. If the field value is associated with the d -cells, then a step function is used as field approximation. If the field values are associated with the grid vertices, usually an at least C^0 -interpolant is used on the d -cells of the complex.

In general, most of the methods discussed in this survey deal with shape models represented by simplicial meshes (e.g. 3D shapes discretized as triangle or tetrahedral meshes, scalar fields defined on simplicial decompositions of the domain) or regular grids (e.g. scalar fields defined on rectangular 2D or 3D grids). Simplicial meshes are usually based on a piece-wise linear interpolation of the shape geometry. Regular grids define a step-wise or analytical approximation of the shape geometry, according to the type of interpolation associated with the hypercubes.

Regular grids can be encoded in very compact data structures. On the other hand, simplicial meshes require data structures which maintain connectivity information, the relation between the d -simplices and the vertices of the complex, plus adjacency relations among d -simplices. Simplicial models, however, are better adapted to variation of the shape, since they can adaptively be built from irregularly distributed data points.

3.2.2 Discretization of differential concepts in a computational setting. The main differential concept we discuss concerns the definition of *critical points*. The definition of a critical point for a function defined over a cell complex, or a simplicial complex, has received a lot of attention in the literature. As most of the concepts are related to differential geometry, the properties characterizing critical points can be expressed in either geometric, analytic, or topological terms. The equivalence among these *views* gives rise to alternative discretization approaches.

Banchoff [Banchoff 1970] introduced critical points for height functions defined over polyhedral surfaces, that is piece-wise linear surfaces defined over a cell decomposition, by using a geometric characterization of critical points. This latter takes into account the position of the tangent plane with respect to the surface. A simplicial model in which linear interpolation is used on the triangles of the underlying mesh is the most common example of a polyhedral surface. A small neighborhood around a local maximum or minimum, never intersects the tangent plane, as shown in Figure 7(a), while a similar small neighborhood is split into four pieces at non-

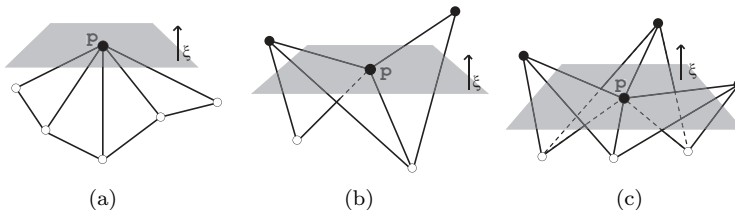


Fig. 7. Configuration of vertices around a maximum point (a), around a non-degenerate saddle (b), and around a monkey-saddle point (c).

degenerate saddles, as shown in Figure 7(b). The number of intersections is used to associate an *index* with each discrete critical point.

Consider a two-dimensional simplicial complex Σ in R^3 with a manifold domain, and the height function $\xi : \mathbb{R}^3 \rightarrow \mathbb{R}$ with respect to the direction ξ in \mathbb{R}^3 ; ξ is called *general for Σ* if $\xi(v) \neq \xi(w)$ whenever v and w are distinct vertices of Σ . Under these assumptions, critical points may occur only at the vertices of the simplices and the number of times that the plane through vertex p and perpendicular to ξ cuts the link of p is equal to the number of 1-simplices in the link of p with one vertex above the plane and one below (see Figure 7). Point p is called *middle* for ξ for these 1-simplices. Then, an indexing scheme is defined for each vertex of Σ as follows [Banchoff 1967]:

$$i(v, \xi) = 1 - \frac{1}{2}(\text{number of } 1\text{-simplices with } v \text{ middle for } \xi). \quad (1)$$

Discrete critical points are at the vertices of the simplicial model and are defined as points with index different from 0. In particular, the index is equal to 1 for maxima and minima, while it can assume an arbitrary negative integer value for saddles. For example, a so-called *monkey-saddle* will have index equal to -2 , as shown in Figure 7(c). The discrete index is different from the classical index (discussed in Section 3.1.5), but Banchoff proved the following *Critical Point Theorem*, that holds for general height functions defined on polyhedral surfaces:

$$\sum_{v \in \Sigma} i(v, \xi) = \chi(\Sigma). \quad (2)$$

Note that for the discrete case, the theorem holds under the assumption that f is general, and it also includes the case of isolated degenerate critical points, such as monkey-saddles, that are not considered by Morse theory. Banchoff also proved the validity of the previous results for general functions defined over d -dimensional complexes, where the *indicator* function generalizes the index defined in Equation (1) [Banchoff 1967]. The characterization provided by Banchoff correctly distinguishes critical points in dimension 2 and 3, while for higher-dimensional spaces the Betti numbers of the *lower link*, that is the set of connected components of the link of a vertex which join points with a height less than that the vertex, provide a more complete characterization of discrete critical points, as suggested in [Edelsbrunner and Harer 2002].

Banchoff's work has been used by most of the authors dealing with computational topology. In many applications, however, the shapes to be analyzed are likely to have degenerate critical points. Degenerate critical points can be handled either by replacing the notion of critical point with that of critical area [Biasotti et al. 2002; Cox et al. 2003], or by perturbing and unfolding the simplicial complex [Edelsbrunner and Mücke 1990; Axen 1999; Edelsbrunner et al. 2003b]. The first approach preserves the characteristics of the embedded manifold, while the second approach requires a slight modification of the shape and the unfolding is ambiguous since there are several splitting choices possible. Depending on the technique in question, the unfolding ambiguity may or may not be able to be removed in a post-processing phase.

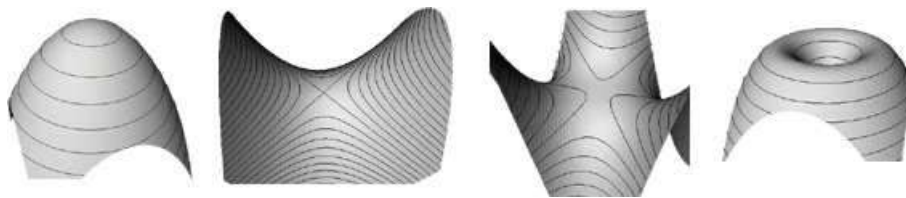


Fig. 8. Transition in the topology of contours around a critical point.

The geometric characterization of critical points can also be used to define critical points on regular models. The difference is that the concept of neighborhood has to be adapted to the connectivity of the underlying regular grid. Moreover, in this case the height function is often viewed as a step function defined at the d -cells of a grid (the pixels or voxels in a digital image), while a piece-wise linear interpolant is always assumed for simplicial models when detecting critical points. The field value at a vertex p of a regular grid is compared to the field values of a suitably-defined set of neighbors of p on the grid [Peucker and Douglas 1975; Toriwaki and Fukumura 1978; Bajaj et al. 1998; Gerstner and Pajarola 2000; Weber et al. 2002; Papaleo 2004]. These approaches are rooted in digital geometry and have been extensively used in image processing [Klette and Rosenfeld 2004].

Another approach commonly used for regular grids is what we call an analytic approach. In this case, there is no attempt at simulating the concept of critical point in the discrete case, but the approach relies on the general idea of fitting an approximating function, sometimes globally discontinuous, on the vertices of the grid (at which the field values are known) [Watson et al. 1985; Bajaj et al. 1998; Schneider and Wood 2004; Schneider 2005; Weber and Scheuermann 2004; Weber et al. 2003]. Critical points are then usually detected through analytical techniques.

Another approach to the definition of critical points in the discrete case is based on a study of the evolution of the level sets (see Section 3.1.5). The level sets may have several connected components, each of which is called a *contour* or *isocontour* of f . In the case of three-dimensional data, each contour is an *isosurface*. A critical point is defined by each change of the topology of the contours, see Figure 8. Thus, the knowledge of the existence of a critical point of Morse index λ between two level sets allows us to topologically reconstruct the shape from the level sets, by joining them appropriately with a λ -cell [Gramain 1971]. This approach is more oriented towards homology than to differential properties, and as pointed out in [Allili et al. 2004], it is more stable with respect to the presence of noise compared with analytical approaches. Above all, it can handle non-isolated critical points without ad-hoc solutions.

A discretization approach related to Morse theory is provided by the so-called *discrete Morse theory* introduced by Forman in [Forman 1998; 2002]. While Morse proved that the topology of a manifold is related to the critical points of a smooth function defined over it, Forman gave an analogous result based on a function defined on the discretization of the manifold as a simplicial or cell complex. As discussed in Section 4, Forman's theory has been applied as a basis for computing discrete approximations of the Morse and Morse-Smale complexes through an entirely combinatorial approach.

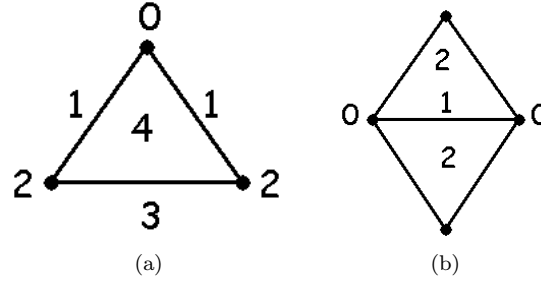


Fig. 9. (a) A discrete Morse function. (b) The edge $f^{-1}(1)$ is critical of index 1.

Forman defines a *discrete Morse function* on a simplicial complex K as a function which assigns higher numbers to higher-dimensional simplices, with at most one exception, locally, at each simplex. This fact implies that the function flows from 2-simplices to 1-simplices and from 1-simplices to 0-simplices, simulating a discrete gradient on the complex. More formally, a function $f : K \rightarrow \mathbb{R}$ is a discrete Morse function if, for every p -simplex $\alpha^p \in K$, the following two conditions hold:

- (1) $\text{Card}(\{\beta^{p+1} \supset \alpha^p \mid f(\beta^{p+1}) \leq f(\alpha^p)\}) \leq 1$,
- (2) $\text{Card}(\{\gamma^{p-1} \subset \alpha^p \mid f(\gamma^{p-1}) \geq f(\alpha^p)\}) \leq 1$,

where β^{p+1} and γ^{p-1} are, respectively, a $(p+1)$ - and a $(p-1)$ -simplex of K , $\beta \supset \alpha$ means that α is a face of β , and $\text{Card}(\cdot)$ denotes the cardinality of the set. A trivial example of a discrete Morse function is given by a function which associates with each simplex its dimension. Figure 9(a) shows another example of a discrete Morse function.

Based on the above definition, Forman has introduced the definition of a *critical simplex*. A simplex α^p is said to be a critical simplex of index p for a discrete Morse function f if it is true that

- (1) $\text{Card}(\{\beta^{p+1} \supset \alpha^p \mid f(\beta^{p+1}) \leq f(\alpha^p)\}) = 0$,
- (2) $\text{Card}(\{\gamma^{p-1} \subset \alpha^p \mid f(\gamma^{p-1}) \geq f(\alpha^p)\}) = 0$.

With reference to Figure 9(a), vertex $f^{-1}(0)$, edge $f^{-1}(3)$ and triangle $f^{-1}(4)$ are critical, and they are the only critical simplices. The above definition provides a discrete analogue of the smooth notion of a critical point of index p , as illustrated by Figure 9(b). In this example edge $e = f^{-1}(1)$ is critical of index 1, since the value $f(e)$ is greater than the value of f at either boundary vertex, and less than the value of f at the 2-simplices bounded by edge e . This means that f decreases as one moves from the edge to its boundary (formed by its two extreme vertices), and increases in every transversal direction, corresponding to the two 2-simplices bounding it. This is a discrete analogue of what happens for smooth functions, where a saddle is connected by an integral line to its two adjacent minima (which are critical points of index 0) and to its two adjacent maxima (which are critical points of index 2).

By the use of the combinatorial approach proposed in Forman's work [Forman 1998], different theorems, analogues of the main theorems of classical Morse theory,

have been proved. Forman defined a Morse complex as a differential complex which consists of the critical cells of a discrete Morse function, and proved that this complex has the same homology as the underlying manifold [Forman 1998].

Forman theory is not the only attempt at the development of a discrete theory which maintains the results of the smooth setting. Different approaches have been proposed, such as *stratified Morse theory* [Goresky and MacPherson 1988] and, more recently, *chainlet geometry* [Harrison 2005]. Another very interesting piece of work has also been proposed recently: the theory of *discrete exterior calculus* by [Hirani 2003] which develops an entire calculus framework which is based only on geometric and combinatorial properties, and extends the concept of differential forms, vector fields and operations on them to discrete objects.

4. MORSE AND MORSE-SMALE COMPLEXES

The intuition behind Morse and Morse-Smale complexes is nicely described by Maxwell [Maxwell 1870]:

Hence each point of the earth's surface has a line of slope, which begins at a certain summit and ends in a certain bottom. Districts whose lines of slope run to the same bottom are called basins or dales. Those whose lines of slope come from the same summit may be called, for want a better name, hills. Hence, the whole earth may be naturally divided into basins or dales, and also, by an independent division, into hills, each point of the surface belonging to a certain dale and also to a certain hill.

If we consider the height function on a terrain, the partition of the surface into its hills corresponds to the decomposition defined by the unstable, or ascending, Morse complex. Similarly, the decomposition of the surface into its dales corresponds to the partition defined by the stable, or descending, Morse complex. If we overlap the decompositions based on the hills and on the dales, we obtain what is called a Morse-Smale decomposition.

This intuitive notion generalizes to any smooth surface and any mapping function f . The distinctive characteristics of Morse and Morse-Smale complexes are that they provide the study of shape properties from the perspective of the *gradient* of the mapping function. Morse and Morse-Smale complexes describe the shape by decomposing it into cells of uniform behavior of the gradient flow and by encoding the adjacencies among these cells in a complex which describes both the topology and the geometry of the gradient of f .

The use of Morse and Morse-Smale complexes was originally introduced in Computer Graphics for the analysis of two-dimensional scalar fields, but, recently, their use has been extended to handle three-dimensional scalar fields as well as generic 3D shapes. The theory behind Morse and Morse-Smale complexes, however, is of general application and has its roots in the theory of dynamical systems [Palis and Melo 1982]. Moreover, Morse and Morse-Smale complexes are strongly related to visualization of vector field topology [Helman and Hesselink 1989; Theisel et al. 2007]. Morse complexes correspond to a decomposition of the shape into either the stable or unstable manifolds associated with the mapping function, and the Morse-Smale complex is defined as the intersection of the stable and unstable manifolds,

under some hypotheses that will be discussed below. Alternatively, this decomposition can be interpreted as been obtained by joining the critical points of the mapping function f by lines of steepest ascent or descent of the gradient, in the case of a two-dimensional scalar field, or surfaces in the case of a three-dimensional scalar field.

These two views are clearly reflected in the literature. A considerable number of algorithms have been developed for extracting critical points and lines, with a specific focus on terrain modeling and analysis. Morse and Morse-Smale complexes have been extensively studied, mainly for the understanding and visualization of scalar fields, but also for more general applications in shape analysis, by using as a mapping function the curvature [Mangan and Whitaker 1999; Page 2003], or the Connolly function [Cazals et al. 2003].

In this section, we review relevant works reported in the literature that cover both interpretations of Morse and Morse-Smale complexes, including also methods that are oriented towards a segmentation of the shape into catchment basins of its minima, that again can be seen as a geometric interpretation of the same mathematical concept, namely regions of uniform flow of the gradient vector field of the mapping function. In other words, we have collected all methods that are rooted, explicitly or not, in the same mathematical framework. We have interpreted different classes of methods as different computational approaches for detecting the same geometrical and topological characterization of a shape.

In the remainder of this section, we discuss first some theoretical aspects, and then review and compare existing algorithms for the extraction of Morse and Morse-Smale complexes, or related structures.

4.1 Theoretical aspects

As pointed out in [Bremer et al. 2004], the first works related to the analysis of the behavior of the gradient over a surface date back to the 19th century [Cayley 1859; Maxwell 1870], even before the publication of the results on Morse theory developed by Thom [Thom 1949] and Smale [Smale 1960]. The definition of Morse complexes relies on the concepts of critical point and of integral line, as discussed below.

Let M be a smooth compact n -manifold without boundary, and let $f : M \rightarrow \mathbb{R}$ be a smooth Morse function. Let us also assume that M is embedded in \mathbb{R}^n or that a Riemannian metric is defined on M . An *integral line* $\gamma : \mathbb{R} \rightarrow M$ of f is defined as a maximal path on M whose velocity vectors, or tangent vectors, agree with the gradient of f , meaning that $\frac{\partial \gamma}{\partial s} = \nabla f(\gamma(s))$ for all s in \mathbb{R} . Each integral line is open at both ends, having its origin (i.e., $\lim_{s \rightarrow -\infty} \gamma(s)$) and its destination (i.e., $\lim_{s \rightarrow +\infty} \gamma(s)$) at critical points of f [Palis and Melo 1982]. Note that the critical points are images of constant integral lines by themselves.

It can be shown that integral lines are pair-wise disjoint, that is, if their images share a point, then they are the same line. The images of integral lines cover the whole M , but if we consider the integral lines associated with the critical points of f , their images define a partition of M .

This partition is used to decompose M into regions of uniform flow, thus capturing the characteristics of the gradient field. More precisely, the *descending manifold* of a critical point p is the set $D(p)$ of points that flow towards p , and the *ascending*

manifold of p is the set $A(p)$ of points that originate from p . In formulae:

$$A(p) = \{q \in M : \lim_{t \rightarrow +\infty} \gamma_q(t) = p\}$$

$$D(p) = \{q \in M : \lim_{t \rightarrow -\infty} \gamma_q(t) = p\},$$

where γ_q is the integral line at the point q . Note that the descending manifold of f is the ascending manifold of $-f$.

In the mathematical literature, the term *unstable* is used instead of ascending, and the term *stable* is used instead of descending [Palis and Melo 1982]. Note also that the partition into ascending manifolds is similar to watershed decomposition [Meyer 1994; Vincent and Soille 1991; Mangan and Whitaker 1999], as we will discuss below.

Here, we follow the terminology and notations adopted in the description of the majority of the methods we discuss. Note that there are slight differences in the definitions of ascending or descending manifolds, depending on whether the critical points are considered as belonging to the manifolds or not. For instance, in [Edelsbrunner et al. 2003b; Bremer et al. 2004], the critical points are added to the related descending and ascending manifolds.

The descending manifold of a critical point p of index i is an open i -cell. Similarly, the ascending manifold of a critical point of index i is a $n - i$ open cell. For example, if M is a 2-manifold the descending manifold of a maximum is an open disk, that of a saddle is an open interval, and that of a minimum is the minimum itself.

The collection of all descending manifolds form a complex, called the *descending Morse complex*, and the collection of all ascending manifolds also form a complex, called the *ascending Morse complex*, which is dual with respect to the descending complex. For instance, when M is a 2-manifold, the 2-cells of the descending Morse 2-complex correspond to the maxima of f , the 1-cells to the saddle points, and the 0-cells to the minima. Symmetrically, the 2-cells of the ascending Morse 2-complex correspond to the minima of f , the 1-cells again to the saddle points, and the 0-cells to the maxima. An example of a decomposition of the domain of a two-dimensional scalar field into an ascending Morse complex is shown in Figure 10 (a). When M is a 3-manifold, the 3-cells of a descending Morse 3-complex correspond to the maxima, the 2-cells to the 2-saddles, the 1-cells to the 1-saddles, and the 0-cells to the minima. Symmetrically, the 3-cells of the ascending Morse 3-complex correspond to the minima, the 2-cells to the 1-saddles, the 1-cells to the 2-saddles, and the 0-cells to the maxima.

Morse-Smale complexes are defined for functions belonging to the important class of dynamical systems, called *Morse-Smale* systems. They are structurally stable on compact manifolds meaning that their structure is preserved under topological equivalencies of the manifold. Intuitively, this means that the topological behavior of the images of the integral lines does not change under small perturbations of the vector field [Palis and Melo 1982]. This property is guaranteed when function f is a *Morse-Smale* function, that is, the descending and ascending Morse complexes intersect only transversally¹. In 2D this means that, if an ascending 1-manifold

¹By definition, two submanifolds A and B of a manifold M intersect transversally in p if $T_p A + T_p B = T_p M$ where T_p is the tangent space at p .

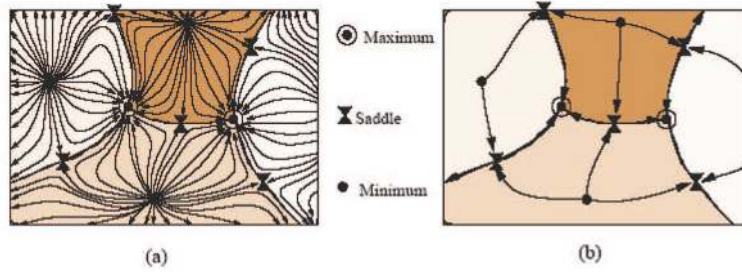


Fig. 10. (a) The ascending Morse complex of a two-dimensional scalar field (the 2-cells correspond to the minima). (b) The Morse-Smale complex. Its 1-skeleton (the set of simplices of dimension 0 and 1) is the critical net.

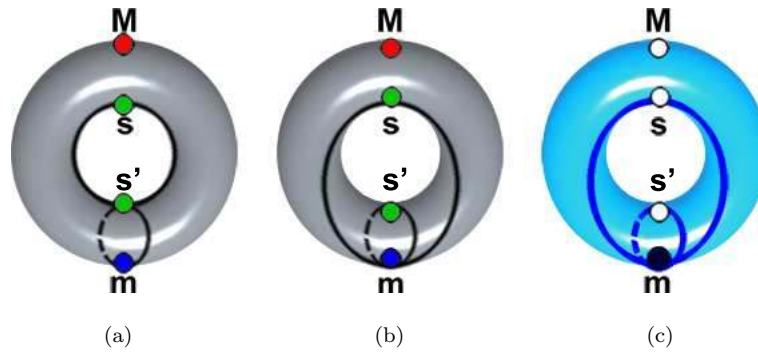


Fig. 11. (a) The integral lines emanating from the higher saddle s reach s' , but a slight perturbation of the height direction causes the integral lines to reach m instead (b). The Morse complex is shown in (c). In (c) the black vertex correspond to the 0-cell, the blue lines represent the 1-cells while the light blue region is the 2-cell.

intersects a descending 1-manifold transversally, they cross at exactly one point. An example can be found in Figure 10 (b), where the two lines in the 1-skeleton intersect at most in one point (a saddle). The importance of the above condition is that it is a stable and generic condition, which is independent of small perturbations of function f and of manifold M . In Figure 11, an example of function whose gradient is not Morse-Smale is shown. The height function on the torus defines four critical points, one maximum, one minimum and two saddles. The integral lines emanating from the higher saddle have the lower saddle as destination in (a), but a slight perturbation of the height directions causes the gradient to flow towards the minimum m instead, as illustrated in (b,c).

In the case of Morse-Smale functions, it is possible to define a complex, called the *Morse-Smale complex*, as the intersection of the ascending and descending manifolds. The cells of the *Morse-Smale complex* are the components of sets $D(p) \cap A(q)$, for all critical points p and q of function f [Edelsbrunner et al. 2001; Edelsbrunner et al. 2003a]. Each cell of the Morse-Smale complex is the union of the integral lines sharing the same origin p of index i and the same destination q of index j .

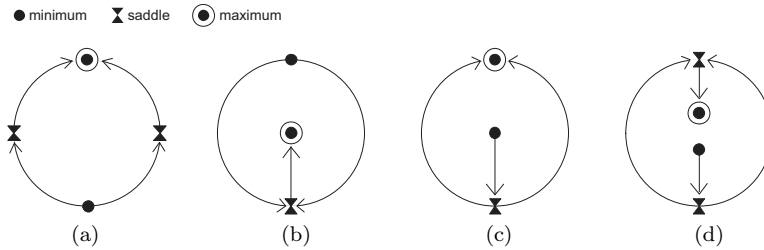


Fig. 12. The four possible configurations for the slope districts in a CPCG.

The dimension of the cell is given by the difference of the indices. Figure 10 (b) shows an example of a Morse-Smale complex. Notice that, in general, the closure of the cells of a Morse-Smale complex may not be homeomorphic to a closed ball.

The Morse-Smale complex is characterized by cells with a regular connectivity. In the 2D case, each saddle point p has four incident 1-cells, two joining p to maxima, and two joining p to minima. Such 1-cells alternate in a cyclic order around p . Also, the 2-cells are quadrangles whose vertices are critical points of f of index 1, 0, 1, 2 (i.e., saddle, minimum, saddle, maximum) in this order. In the 3D case, all 2-cells are quadrangles whose vertices are a minimum, 1-saddle, 2-saddle, 1-saddle in this order (*quadrangles of type 1*), or a 1-saddle, a 2-saddle, a maximum, a 1-saddle in this order (*quadrangles of type 2*). A 1-cell connecting a 1-saddle and a 2-saddle is on the boundary of four quadrangles that alternate between quadrangles of type 1 and type 2. The 3-cells are called *crystals* and are bounded by quadrangles [Edelsbrunner et al. 2001; Edelsbrunner et al. 2003a].

It is interesting to note the similarity between the Morse-Smale complex and the configuration of slope districts defined in [Nackman 1984], where the function is not necessarily a Morse-Smale one, but simply a Morse function. In particular, Nackman defined a graph, called the *Critical Point Configuration Graph (CPCG)*, in which the nodes represent critical points and the arcs represent the integral lines connecting them. The *CPCG* is a planar graph and its embedding on the domain M of f induces a partition of M into two-dimensional regions, called *slope districts*, characterized by the uniformity of the gradient flow. Since f is not necessarily a Morse-Smale function, there are configurations of the critical points of f which do not occur for Morse-Smale functions. Nackman shows [Nackman 1984] that there are only four basic possible configurations for a slope district and they can be obtained, up to equivalence, by inserting saddle points in the arcs. These configurations are illustrated in Figure 12. The configurations in Figures 12(a), (b) and (c) are formed by saddle, minimum, saddle, and maximum. All three illustrate the possible types of 2-cells in a Morse-Smale complex. The ones in Figures 12 (b) and (c) correspond to degenerate situations, usually called *strangulations* [Gyulassy et al. 2005]. The configuration in Figure 12(d) cannot happen for a Morse-Smale function, since ascending and descending 1-manifolds do not intersect transversally, but coincide.

The 1-skeleton of a Morse-Smale complex is a 1-complex formed by integral lines joining critical points. Similar structures among critical points have been widely studied in the literature under the name of *critical nets*. A graph representation

of the critical net in a two-dimensional Morse-Smale complex is the so-called *surface network* [Pfaltz 1976; Schneider and Wood 2004], widely used in spatial data processing for morphological terrain modeling and analysis (see [Rana 2004] for an interesting collection of contributions on this specific topic).

4.2 Computational aspects

In this section, we describe and analyze different techniques proposed in the literature to compute the Morse or the Morse-Smale complexes. The majority of the algorithms have been developed for two-dimensional scalar fields, or for scalar functions defined over a 2-manifold without boundary.

Most of these algorithms use what we call a *boundary-based* approach, since they extract the Morse-Smale, or the Morse complexes, by computing the critical points and then tracing the integral lines joining them, or their approximations, starting from saddle points and converging to minima and maxima. In this sense, they compute the Morse-Smale complex by extracting the boundaries of its 2-cells. Other algorithms use a *region-based* approach in the sense that they compute an approximation of the ascending and/or descending Morse complex by growing the 2-cells corresponding to the minima, or to the maxima, of the Morse function f defined on a manifold M . If f is a Morse-Smale function, then the Morse-Smale complex can be obtained as the intersection of the ascending and descending Morse complexes. For this reason, we organize the presentation of the various methods by grouping them as methods for extracting the Morse, or the Morse-Smale, complexes. Note that, in general, region-based methods aim to extract a Morse complex, while boundary-based approaches typically focus on the extraction of a Morse-Smale complex. Exceptions are the algorithms by [Cazals et al. 2003] and [Ni et al. 2004]. We can also classify the algorithms on the basis of their input, namely a regular model [Bajaj et al. 1998; Schneider and Wood 2004; Schneider 2005], or a simplicial model [Takahashi et al. 1995; Edelsbrunner et al. 2001; Bajaj and Schikore 1998; Bremer et al. 2003; Cazals et al. 2003; Danovaro et al. 2003a; Danovaro et al. 2003b; Edelsbrunner et al. 2003a; Pascucci 2004; Magillo et al. 2007; Ni et al. 2004]. Finally, watershed algorithms, developed for image segmentation, can be viewed as region-based methods for computing the ascending and descending Morse complexes [Meyer 1994; Vincent and Soille 1991; Mangan and Whitaker 1999; Stoev and Strasser 2000].

4.2.1 Computing ascending and descending Morse complexes. In [Danovaro et al. 2003b; Magillo et al. 2007], algorithms have been presented for computing the descending and ascending Morse complexes for a 2D simplicial model. These algorithms have been applied to the segmentation and morphological analysis of terrain models, and the algorithm in [Magillo et al. 2007] has also been applied to 3D shape segmentation. In both algorithms, the ascending and descending complexes are computed independently by applying a region-growing technique on the triangles of the simplicial model. Basically, the algorithms perform a breadth-first traversal of the dual graph of the triangle mesh, in which the nodes correspond to triangles of the mesh, and the arcs to the edges shared by edge-adjacent triangles. When extracting the descending Morse complex, maxima are extracted (as discussed in Section 3.2.2). A descending 2-cell C , associated with a current maximum m , is

initialized with all the triangles in the star of m , which have not yet been assigned to any 2-cell. Then, the cell C is grown in a breadth-first fashion by adding one triangle at a time according to a criterion which is specific for each algorithm.

In [Danovaro et al. 2003b], the 2-cell C of the ascending complex is extended to include a triangle $t = pqr$ which is adjacent to C along a common edge pq if the remaining vertex r of t has a height lower than those of p and q . At the end of the process, any 2-cell containing a maximum on its boundary is merged with the 2-cell adjacent along that boundary. In [Magillo et al. 2007], the growing technique in [Danovaro et al. 2003b] is improved to avoid over-segmentation. A triangle t is attached to the descending 2-cell C when the previous condition is verified, but also when vertex r has a height values which is between the height values of p and q . In this case, also some of the triangles in the star of r are added to C .

The ascending complex is computed in a completely symmetric way for both algorithms. Their worst-case time complexity is linear in the number of vertices of the simplicial model. Note that both algorithms are entirely combinatorial, since they do not involve floating-point computations. In [Danovaro et al. 2003a], a variant of this approach is presented which is based on the gradient associated with each triangle, similarly to [Yu et al. 1996].

The algorithm described in [Danovaro et al. 2003b] has been defined in a dimension-independent way and implemented for 3D simplicial models as well. In [Mesmoudi and De Floriani 2007] the authors have defined the discrete gradient vector field associated with the decomposition produced by the algorithm in [Danovaro et al. 2003b] and have shown that it is a subfield of the gradient field of a Forman function F whose restriction over the vertices of the simplicial model coincides with the given scalar field function f .

A region-based combinatorial algorithm for computing ascending manifolds has been proposed in [Dey et al. 2003] to segment a 3D shape M into meaningful features. In this case, the shape is defined by a set of points belonging to its boundary and is discretized as a Delaunay tetrahedral mesh Σ with vertices at the data points. In this case, a distance function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is defined over the space \mathbb{R}^n where the shape M is embedded by the relation:

$$f(x) = \min_{p \in M} \|p - x\|^2, \forall x \in \mathbb{R}^n$$

and, most importantly, the results are discussed for that specific function selection. Note that the distance function is not a Morse function, so we cannot properly talk about a Morse complex, but the authors are interested in the computation of the ascending manifolds of maxima. Maxima of the distance function are detected at a subset of the vertices of the dual Voronoi diagram of the mesh Σ . Approximations of the three-dimensional ascending manifolds are computed by a region-growing approach starting from the tetrahedra which contain the maxima. A discrete gradient field is computed by defining a flow relation between face-adjacent tetrahedra, whose transitive closure is acyclic [Edelsbrunner et al. 1998; Giesen and John 2003].

In [Ni et al. 2004], a boundary-based approach to the computation of the descending Morse complex is proposed. The algorithm is applied to a triangle mesh Σ discretizing a 2-manifold without boundary endowed with a Morse function f which minimizes the number of critical points. The algorithm extracts first the

critical points and successively traces the descending 1-manifolds starting from the saddles, in no specific order. If a point p is a multiple saddle of multiplicity m , then m manifolds start at p , each corresponding to one connected component of the lower link of p , see Section 3.2.2. Descending 1-manifolds are constructed by moving along the edges of the triangle mesh by choosing each time a point with lowest height in the lower link of the current point (as in [Baja] and Schikore 1998; Takahashi et al. 1995]). These manifolds can merge, but they cannot cross, and they do not separate after merging. Descending 1-manifolds are not allowed to pass through saddles other than their starting saddle. In other words, if a descending 1-manifold has reached a point p in the link of a saddle s , then the edge ps cannot be used to extend the manifold, i.e., saddle s is not taken into account when searching the neighbor of p with lowest height. Further improvements are proposed, which include handling flat regions, or plateaus, or surfaces with boundary. In the case of plateaus, each simply-connected flat region is considered as a single vertex. This is equivalent to collapsing edges connecting vertices in the flat region. Surfaces with boundary are treated by adding a vertex p for each boundary component, together with edges and triangles connecting p to vertices on the boundary loop, producing a model without boundary.

The ascending and descending Morse complexes can also be computed by applying the *discrete watershed transform*. The watershed transform was first introduced in image analysis for the segmentation of gray-scale images and several definitions exist in the discrete case [Beucher and Lantuejoul 1979; Bieniek and Moga 1998; Meijster and Roerdink 1996; Meyer 1994; Vincent and Soille 1991]. It provides a decomposition of a the domain of a C^2 function f into regions of influence of the minima, called *catchment basins*. The boundary of the catchment basins of the minima form the *watershed lines*. Catchment basins and watershed lines are described in terms of topographic distance, using the formalization proposed in [Meyer 1994]. In the 2D case, if f is a Morse function, it can be seen that the catchment basins of the minima of f are the 2-cells in the ascending Morse complex of f and the watershed lines are 1-cells in such complex. Through a change in the sign of function f , the descending manifolds of the maxima can be extracted, and thus, we can obtain the descending Morse complex for the original function.

Several algorithms have been developed for the computation of the watershed transform (see [Roerdink and Meijster 2000; Najman and Couprie 2003] for a survey). Here, we briefly review the two major approaches proposed in the literature, namely those based on the discretization of the topographic distance [Meyer 1994], and those based on simulating the immersion of a catchment basin in water [Vincent and Soille 1991]. Both types of watershed algorithms have been developed for images and, thus, can be applied to regular models of terrain without any change, but their extension to simplicial models is straightforward. Watershed algorithms only label the vertices of the mesh, but such labeling can be easily propagated to the triangles. Note that the watershed algorithms discussed here are entirely discrete and differ from the approach developed in the computational geometry literature for piece-wise linear triangulated terrains, in which the terrains are considered as continuous, and several steepest descent paths can cross a single triangle [Yu et al. 1996; McAllister 1999; de Berg et al. 2007]. In this case, the drainage network

computed on the basis of the watersheds of the points on the terrain model can have a worst-case complexity of $\Omega(n^3)$, even if in practice the size of the network has been observed to be linear in the number of vertices of the model [Yu et al. 1996].

The definition of a catchment basin, watershed, and watershed transform in the discrete case is similar to the definition in the continuous case as proposed in [Meyer 1994]. The only difference is that the continuous topographic distance is replaced with a (cost-based) discrete topographic distance. The *discrete topographic distance* between two vertices p and q of a simplicial model is defined as the minimum-cost path joining them. A path between two vertices p and q of a simplicial model is defined as a sequence of edges in the model joining p and q . The cost of an edge uv , having u and v as extreme vertices, can be defined in different ways (e.g. the Euclidean distance between u and v , the Euclidean distance composed with the function values $f(u)$ and $f(v)$ and so on) [Meyer 1994]. The cost of a path is just the sum of the costs of the edges forming it. By selecting as topographic distance a minimum-cost path, the catchment basin of a minimum p becomes the set of vertices in the simplicial model which are closer, in terms of the discrete topographic distance, to p than to any other minimum in the model. The watershed lines are the complement of the collection of the catchment basins of the minima, when the complement is taken on the set of vertices of the simplicial model. There are different implementations of the approach proposed in [Meyer 1994]. All of them implement a modification of a classical shortest path algorithm in order to grow the ascending manifold, or equivalently, the catchment basin associated with each minimum. The worst-case time complexity of the algorithm presented in [Meyer 1994] is $O(n \log n)$ where n is the number of vertices in the simplicial model, due to the initial sorting of the vertices according to their height values.

The discrete topographic distance is not the only way to apply a watershed algorithm to a simplicial model. In [Vincent and Soille 1991; Soille 2004], a different paradigm is presented, based on the idea of *simulated immersion* (see also [Serra 1983] for the binary case). This idea can be described in a very intuitive way. Let us consider the graph of a two-dimensional scalar field f , and assume that holes are drilled in place of local minima. We immerse this surface in a pool of water, building dams to prevent water coming from different minima from merging. Then, the watershed of f is described by these dams, and the catchment basins of the minima are delineated by the dams. The definition of watershed by simulated immersion is given in a recursive manner in [Beucher and Lantuejoul 1979; Vincent and Soille 1991], where the immersion procedure is simulated by the use of the concept of *skeleton by influence zones* which associates with a family of components C_1, C_2, \dots, C_n (dams), called *markers*, the connected set of points at equal Euclidean distance from two different components (see [Soille 2004] for definitions and details). The skeleton basically maintains the information regarding how and when the different dams merge together. The algorithm first sorts the vertices of the model in increasing function value and extracts the set of minima and then it performs the *flooding* step level by level, starting from the minima. This technique is similar to the plane sweep approach used in the computation of the contour tree, e.g. [de Berg and van Kreveld 1997] in Section 5.2. As shown in [Stoev and

Strasser 2000], there exists an implementation of the algorithm in [Vincent and Soille 1991] which has a linear worst-case time complexity in the number of vertices of the initial regular model since it does not require sorting. But, for simplicial models, the algorithm requires a sorting of vertices according to their height value and this results in an $O(n \log n)$ worst-case complexity.

Both approaches described above start from the minima and let the catchment basins grow until all the points in the model are labeled as belonging to catchment basins, or to watershed lines. A dual strategy is used in the watershed algorithms presented in [Mangan and Whitaker 1999; Stoev and Strasser 2000], which are based on the *rain falling* paradigm. Both algorithms label the vertices of the mesh and construct steepest descending paths from each vertex p until a minimum m , or a labeled vertex q , is reached. The membership label of m (or q) is propagated backwards along the steepest path reaching p . As result of this process, the simplicial model is segmented into catchment basins associated with the minima of f (but no watershed lines are identified), where each catchment basin is defined by the set of vertices sharing a membership-label. Note the correspondence between these labels and the components of the union-find data structure used in the contour tree computation [van Kreveld et al. 1997; Pascucci and Cole-McLaughlin 2002; Carr 2004], Section 5.2. The two algorithms differ basically in the type of input they use since [Mangan and Whitaker 1999] was proposed for triangle meshes, while [Stoev and Strasser 2000] was presented for 2D, or 3D, regular grids, even if [Stoev and Strasser 2000] could be easily extended to grid structures with irregular connectivity. [Mangan and Whitaker 1999; Stoev and Strasser 2000] produce a labeling of the vertices of the input simplicial model as output. A drawback of the rain falling paradigm is in the non-uniqueness of the lowest neighbor q of a point p , and the occurrence of plateaus, or flat zones (see [Stoev and Strasser 2000]). In the case of plateaus, both algorithms distinguish among plateau types. For plateaus corresponding to a minimum, both algorithms mark all the points of each plateau with a unique label. In [Mangan and Whitaker 1999] each non-minimal plateaus Q is assigned entirely to a single catchment basin, while in [Stoev and Strasser 2000] Q is split among different catchment basins. No strategy is presented in [Mangan and Whitaker 1999] to solve the ambiguity arising when a point p has more than one lowest neighbor, while a disambiguation procedure is given in [Stoev and Strasser 2000]. The worst-case time complexity of both watershed algorithms in [Mangan and Whitaker 1999; Stoev and Strasser 2000] is $O(n \log n)$, where n is the number of vertices in the simplicial model. For regular models, the complexity becomes linear in the number of vertices, as shown in [Stoev and Strasser 2000].

4.2.2 Computing a Morse-Smale complex. Most boundary-based methods for computing a Morse-Smale complex [Takahashi et al. 1995; Bajaj and Schikore 1998; Edelsbrunner et al. 2001; Bremer et al. 2003; Pascucci 2004; Bremer and Pascucci 2007] from two-dimensional complexes follow the same algorithmic approach, consisting of two basic steps:

- extract the critical points and unfold multiple saddles;
- compute the 1-cells of the Morse-Smale complex, or their approximations, by starting from the saddle points, and tracing two paths on the underlying shape

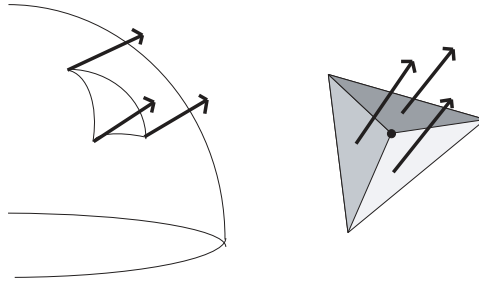


Fig. 13. The normal space for a cluster of triangles incident on a vertex.

model which stop at minima and maxima, respectively.

The extraction of the critical points is usually performed based on techniques implementing the classification by Banchoff [Banchoff 1967] (see Section 3.2.2). In [Bajaj and Schikore 1998], a different approach is used even if, as in the previous algorithms, the technique performs the classification of vertices with respect to the local neighborhood. In this case, at a given vertex of the simplicial model, the authors estimate the gradient as it can take a range of values based on the normals of the triangles incident in that vertex. As a result, critical points are defined as vertices at which the normal space of the incident triangles includes the vector $(0, 0, 1)$ (see Figure 13).

The main difference among the different methods relies in the technique used to trace the integral lines that define the 1-cells: either the steepest ascent or descent is traced, or approximated integral lines are used instead, provided that they respect the connectivity of the Morse-Smale complex.

The algorithms in [Takahashi et al. 1995; Bajaj and Schikore 1998; Bremer and Pascucci 2007] extract the integral lines forming the Morse-Smale complex by computing paths only along the edges of the triangle mesh, selecting the vertex of highest, or lowest, height at each step. As observed by [Chiang et al. 2005], the time complexity of [Takahashi et al. 1995] is $O(nc)$, where c denotes the number of critical points. As a pre-processing step, to simulate that the critical points are non degenerate and there exist no boundary saddles, [Bremer and Pascucci 2007] adopted a symbolic perturbation of the mesh before extracting the Morse-Smale complex. The algorithms in [Bremer et al. 2003; Pascucci 2004] estimate the gradient along 1-simplices and 2-simplices, and compute the ascending and descending paths not only along the edges, but possibly crossing triangles in order to follow the actual paths of steepest ascent, or descent.

In [Edelsbrunner et al. 2001; Edelsbrunner et al. 2003a] the notion of *Quasi Morse-Smale (QMS)* complex is introduced as an intermediate step towards the computation of the Morse-Smale complex. The QMS is defined for 2D and 3D simplicial complexes, which triangulate a 2-manifold or a 3-manifold without boundary, respectively. The QMS has the same combinatorial structure of a Morse-Smale complex, but it differs from it in that the 1-cells in 2D, and the 1-cells and 2-cells in 3D are not necessarily those of maximal ascent, or descent. The idea behind a QMS, called *simulation of differentiability* is that of extending the smooth notions to the piece-wise linear case so as to guarantee that the complex has the same structural

form of the smooth counterpart, and to achieve numerical accuracy via local transformations that preserve the structure of the complex [Edelsbrunner et al. 2001]. The QMS is a splittable quadrangulation of M whose vertices are the critical points of f and whose arcs are strictly monotonic in f . The 0-cells of a QMS complex are the critical points of f , the 1-cells connect minima to saddles (1-saddles in 3D), maxima to saddles (2-saddles in 3D) and, in the 3D case, 1-saddles to 2-saddles [Edelsbrunner et al. 2003a]. Since the computed lines are an approximation of the integral lines in the smooth case, the algorithm resolves problems arising when merging and forking of paths occur. Once the QMS complex is computed, a series of operations, called *handle slides*, are applied to turn the QMS into a Morse-Smale complex. For 2-manifolds, it is possible to find such a sequence of handle slides, while for 3-manifolds this is still an open question [Edelsbrunner et al. 2003a].

Note that the algorithm in [Ni et al. 2004], discussed in Section 4.2.1, is an extension of the boundary-based algorithm in [Edelsbrunner et al. 2001] with the difference that it extracts the descending Morse complex, it does not split multiple saddles, and it can deal with flat regions.

The boundary-based algorithm in [Edelsbrunner et al. 2003a] extracts the Quasi Morse-Smale complex for a simplicial model of a three-dimensional scalar field by computing first the critical points through the reduced Betti numbers of the lower link of the vertices. Second, the descending Morse complex is computed and, finally, the algorithm extracts the ascending manifolds in pieces inside the cells formed by the descending manifolds. In other words, the structure of the descending manifolds is used while computing the ascending ones in order to maintain the structural integrity of the whole complex. Note that it is not guaranteed that the same complex would be obtained if first the ascending, and then the descending manifolds were computed. An implementation of this algorithm is described in [Natarajan and Pascucci 2005]. The running time of the algorithm for computing the Morse-Smale complex is bounded from above by the time for sorting the vertices, plus the input size, for constructing and analyzing the vertex links, plus the output size for describing the resulting Morse-Smale complex, as pointed out in [Edelsbrunner et al. 2003a]. Sorting the n vertices of the input model takes $O(n \log n)$ time, the input size is $O(n^2)$, while the worst case for the size of the output can be arbitrary large.

In [Cazals et al. 2003], an approach, rooted in the discrete Morse theory proposed by Forman (see Section 3.2), is presented for the computation of the Morse-Smale complex for a two-dimensional simplicial complex. In order to apply Forman theory to a scalar field f which is given only at the vertices of a mesh, f is suitably extended by defining it on all 1- and 2-simplices (i.e. edges and triangles), in such a way that minima, saddles and maxima of f occur at vertices, edges, and triangles, respectively. A discrete gradient vector field is induced on the complex by the discrete Morse function [Forman 1998; 2002]. The algorithm proposed in [Cazals et al. 2003] is based on the analysis of the 1-skeleton of the simplicial complex (i.e. the graph formed by its vertices and edges) and the dual graph of the underlying triangle mesh (in which the nodes correspond to the triangles and the arcs correspond to the edges). The ascending manifold of a minimum p consists of vertices and edges, while the descending manifold of a maximum q is made of triangles and edges. In [Lewiner

et al. 2003; 2004], it has been shown that a gradient vector field is equivalent to two spanning forests, one on the 1-skeleton and the other on the dual graph of the model, such that an edge cannot belong to both forests. The roots of the two forests are the minima and the maxima, respectively. The connected components of the two forests define the descending and ascending 2-manifolds. The Morse-Smale complex is obtained as the intersections of these regions. This algorithm can be classified both as a boundary-based and as a region-based technique, since the ascending 2-manifolds are computed through a sort of region-growing approach around the maxima, while the boundaries of the descending 2-manifolds are computed from the forests in the 1-skeleton. The worst-case time complexity of the preprocessing edge sorting step is $O(n \log n)$, and that of the forest creation step is $O(n\alpha(n))$, where n denotes the number of vertices of the model, and α is the inverse of the Ackermann function [Ackermann 1928; Cazals et al. 2003].

Algorithms exist in the literature that compute the Morse-Smale complex directly from two-dimensional regular grids [Bajaj et al. 1998; Schneider and Wood 2004; Schneider 2005]. They are boundary-based in nature, since they compute the 1-skeletons of the Morse-Smale complex (i.e. the critical net), through a technique conceptually very similar to the one used for two-dimensional simplicial models. First, the critical points are extracted, and then the integral lines joining them are computed as lines of steepest ascent, or descent. All three algorithms fit a surface with a certain degree of continuity to the input data set in order to extract the critical points.

The algorithm presented in [Bajaj et al. 1998] uses a globally C^1 Bernstein-Bézier bi-cubic interpolant, locally defined on each two-dimensional cell. This interpolating function does not remove any critical point of the initial input data, but it may add a small number of additional critical points that are defined as vertices at which the normal space of the adjacent triangles includes the vector $(0, 0, 1)$, see Figure 13. The integral lines are computed through a Runge-Kutta integration technique [Press et al. 1992].

The algorithm proposed in [Schneider 2005] uses a bilinear C^0 interpolating function on each 2-cell of the grid. Minima and maxima can occur only at grid vertices, but additional saddles may be introduced by the interpolation inside a 2-cell. A grid point p is classified by considering only the height of its 4-adjacent neighbors, while a saddle inside a 2-cell can be detected by considering the height of the four vertices of the 2-cell. Integral lines can follow grid edges, or go through 2-cells. When an integral line crosses a grid cell, it is approximated with small linear steps, or computed exactly, by solving a linear system of differential equations.

The algorithm presented in [Schneider and Wood 2004] fits a bi-quadratic polynomial by considering the 8-adjacent neighbors of each grid vertex p , four along grid edges and four diagonally. The classification of the critical points is done by looking at the neighbors. The algorithm produces a globally discontinuous approximation, formed by local surface patches and computes the first and second derivatives analytically, as in [Schneider 2005]. This information is used to trace the integral lines starting from the saddles.

As in the case of simplicial models, the problem of computing a Morse-Smale complex from 3D regular models has not been studied extensively. There are few

algorithms that extract critical points [Bajaj et al. 1998; Weber et al. 2002; Weber et al. 2003; Weber and Scheuermann 2004; Papaleo 2004]. [Weber et al. 2002; Papaleo 2004] use Banchoff’s definition of critical points in the discrete case presented in Section 3.2.2, while [Bajaj et al. 1998] use a different approach, as discussed above in this Section for the 2D case. Moreover, in [Bajaj et al. 1998] integral lines connecting the critical points are computed, thus providing an approximation of the 1-skeleton of the Morse-Smale complex. In [Weber et al. 2003; Weber and Scheuermann 2004], a trilinear interpolant is used on the 3-cells. Also, the authors assume that the values of scalar field f at any pair of edge-adjacent vertices of the grid are different. Thus, critical points can occur only at vertices, and saddle points only inside the 3- and the 2-cells. In [Weber et al. 2003], the above approach is further extended to detect critical regions associated with minima, maxima and saddles, and not just isolated critical points. Critical regions arise in practice since data sets often contain regions of constant values.

Table III summarizes the algorithms reviewed. The running time complexity is not specified by several of the authors and is therefore not reported in the table. However, the time complexity varies from $O(n)$ to $O(n \log n)$, where n is the number of vertices in the initial model. An $O(n)$ implementation is possible for several algorithms which do not require an initial sorting step.

Algorithm	Input	Output
[Vincent and Soille 1991]	Regular	Morse complex
[Meyer 1994]	Regular	Morse complex
[Takahashi et al. 1995]	Simplicial (2D)	Morse-Smale complex
[Bajaj and Schikore 1998]	Simplicial (2D)	Morse-Smale complex
[Bajaj et al. 1998]	Regular (2D)	Morse-Smale complex
[Bajaj et al. 1998]	Regular (3D)	1-skeleton of the Morse-Smale complex
[Mangan and Whitaker 1999]	Simplicial (2D)	Morse complexes
[Stoev and Strasser 2000]	Simplicial (2D/3D)	Morse complexes
[Edelsbrunner et al. 2001]	Simplicial (2D)	Morse-Smale complex
[Dey et al. 2003]	Simplicial (3D)	Morse complexes (stable manifolds)
[Danovaro et al. 2003a]	Simplicial (2D)	Morse complexes
[Danovaro et al. 2003b]	Simplicial (2D/3D)	Morse complexes
[Bremer et al. 2003]	Simplicial (2D)	Morse-Smale complex
[Weber et al. 2003]	Regular (3D)	Critical regions for maxima and minima
[Cazals et al. 2003]	Simplicial (2D)	Morse-Smale complex
[Edelsbrunner et al. 2003a]	Simplicial (3D)	Morse-Smale complex
[Ni et al. 2004]	Simplicial (2D)	Morse complexes
[Pascucci 2004]	Simplicial (2D)	Morse-Smale complex
[Schneider and Wood 2004]	Regular (2D)	Morse-Smale complex
[Schneider 2005]	Regular (2D)	Morse-Smale complex
[Magillo et al. 2007]	Simplicial (2D)	Morse complex

Table III. Algorithms for the extraction of the descending/ascending Morse complex or of the Morse-Smale complex. For each algorithm the input model is outlined (regular or simplicial) as well as the output it produces (Morse or Morse-Smale complex).

4.3 Applications

Applications of Morse and Morse-Smale complexes can be found in scientific visualization [Bremer et al. 2003; Pascucci 2004; Edelsbrunner et al. 2001; Edelsbrunner et al. 2003a; Gyulassy et al. 2005; Gyulassy et al. 2006], where scientific data consist of measurements over a geometric domain or space. Applications in physics simulation of the turbulent mixing between two fluids are discussed in [Bremer and Pascucci 2007]. The segmentation provided by these complexes allows modeling the *bubbles* formed during the mixing process. Some methods have been applied for segmenting and analyzing molecular 3D shapes [Cazals et al. 2003; Natarajan et al. 2006; Bremer and Pascucci 2007] to study the role of cavities and protrusions in protein-protein interactions.

The computation of ascending manifolds of the distance function has been used also in shape matching and retrieval [Dey et al. 2003]. Since the method is tuned on the use of the distance function, the resulting segmentation extracts the protrusions of the shape. Based on such segmentation, a shape signature is defined which associates with each Morse complex of the segmented 2-manifold a number of properties, e.g. the weighted volume or the bounding box, that are used for similarity assessment. The results are mainly geared towards shapes that exhibit a structure that is well-described by protrusions.

Interesting results have also been reported for the analysis of terrains in Geographic Information System applications [Takahashi et al. 1995; Danovaro et al. 2003a; Bremer et al. 2003]. For example, in [Danovaro et al. 2003a], the extraction of the Morse-Smale complex was applied to generate a multi-resolution model for terrains which encodes the terrain morphology at a continuous range of different resolutions.

In the field of image analysis, watershed algorithms have been used for image segmentation [Vincent and Soille 1991; Meyer 1994; Najman and Schmitt 1996; Bieniek and Moga 2000]. Watershed approaches have been applied to 3D shape segmentation, as for example in [Page 2003; Mangan and Whitaker 1999], where the curvature is used as the height function in order to obtain *natural* shape segmentations from a human perception point of view. The algorithm in [Magillo et al. 2007] has been applied to curvature-based shape segmentation, based on the discrete curvature estimation technique in [Mesmoudi et al. 2007]. These approaches have important applications in form feature extraction, mesh reduction, and texture mapping. Scalar topology detection has been proven to be useful for image co-registration, iso-contouring, and mesh compression [Bajaj and Schikore 1998; Bajaj et al. 1998].

Recent research activities have moved towards *hierarchical representations* of scalar fields. For example, the Morse-Smale complex has been used to perform controlled simplification of topological features in functions defined on two-dimensional domains [Bremer et al. 2004; Edelsbrunner et al. 2003b; Danovaro et al. 2006]. These works are motivated by two major issues.

The first issue is a common problem in both image and mesh segmentation algorithms and is the over-segmentation due to the presence of noise in the data sets. For this purpose, *generalization algorithms* have been developed by several authors that locally simplify the structure of a Morse-Smale complex [Wolf 2004; Edels-

brunner et al. 2001; Bremer et al. 2004; Takahashi et al. 1995; Takahashi 2004; Gyulassy et al. 2005; Gyulassy et al. 2006; Natarajan et al. 2006]. In [Laney et al. 2006], for example, the authors build a hierarchical Morse-Smale complex for the envelope surfaces describing the boundary between undisturbed and mixed fluids, and they use topological persistence (see Section 8) to automatically cleanup the noise.

The generalization of a Morse-Smale complex for a 2D scalar field consists of collapsing a maximum-saddle pair into a maximum, or a minimum-saddle pair into a minimum, so as to maintain the consistency of the underlying complex. This operation is called *cancellation* (see also Section 8.3). A generalization can be described in terms of the combinatorial representation of the critical net, defined by the surface network [Danovaro et al. 2006]. The problem of generalizing 3D Morse-Smale complexes has been recently investigated in [Edelsbrunner et al. 2003a; Gyulassy et al. 2005; Gyulassy et al. 2006], by defining three cancellation operators and extending the 2D technique described in [Edelsbrunner et al. 2001; Bremer et al. 2004].

The second issue is related to the large size and complexity of available scientific data sets. Thus, a multi-resolution representation is crucial for an interactive exploration of such data sets. There exist just a few proposals in the literature for multi-resolution representations for 2D scalar fields based on Morse-Smale complexes [Bremer et al. 2004; Bremer et al. 2005; Danovaro et al. 2006; Danovaro et al. 2007]. All such proposals are based on the general multi-resolution framework for cell complexes introduced in [De Floriani et al. 1999].

Starting from the eigenfunctions of the Laplacian matrix of a closed triangle mesh, the Morse-Smale complexes are used as quadrangular complexes and applied to mesh parameterization and remeshing in [Dong et al. 2006]. The Morse-Smale complexes are simplified through cancellation operations of the critical points until a *denoised* complex is obtained. Then, the simplified base complex is used to build a global parameterization over the complex. Once possibly degeneracies of the complex were eliminated, the quadrangular complex is used to produce a semi-regular mesh.

Note that generalization operators may be defined on Morse-Smale complexes that use a global view over the spatial distribution of the function critical points for detecting, ordering, and removing features. The Morse-Smale complex may also be generalized locally, by restricting the generalization to a local neighborhood of the non-significant features [Gyulassy et al. 2006]. Simplification of topological features has also been studied in the context of vector fields [de Leeuw and van Liere 1999; Tricoche et al. 2001]. These approaches use numerical techniques and are therefore prone to instability. In contrast, approaches based on Morse-Smale complexes are usually more stable since they are combinatorial in nature.

5. CONTOUR TREES

Isocontours are very useful to visualize the variation of a scalar field in its domain. Pioneering work in this field can be dated back to the beginning of the 18th century, when Edmond Halley published the results of his extensive measurements of the difference between compass readings and determinations of north by astronomical

observation. The results were published as a sea chart depicting also the isogons of the earth's magnetic field, that is, lines connecting those points at which the magnetic declination is the same [Halley 1702]. More than one century after, Cayley [Cayley 1859] highlighted the relevance of contour lines for topographic analysis. Cayley defined contour lines to be connected sets of points at which the elevation field has some specific constant value. Similar concepts were extended by Maxwell [Maxwell 1870]:

The results of the survey of the surface of a country are most conveniently exhibited by means of a map on which are traced contour-lines, each contour-line representing the intersection of a level surface with the surface of the earth, and being distinguished by a numeral which indicates the level surface to which it belongs.

Starting from the consideration that the evolution of contour lines on a surface explicitly represents hills and dales with their elevation-based adjacency relationships, contour trees were originally introduced as an efficient data structure to store containment relationships among contours in contour maps, typically representing terrain elevations or any other continuous real function of two variables [Boyell and Ruston 1963].

The main target of contour trees was the fast evaluation of elevations at locations other than the points on the contours [Freeman and Morse 1967; Merrill 1973]. In the last decade, contour trees have been extensively used as a more general tool to analyze and understand shapes defined by n -dimensional scalar fields, as a support to scientific visualization of complex phenomena. In their general form, contour trees explore the shape of a scalar field $\Gamma = (D, f)$ by analyzing the evolution of its level sets as f spans the range of its possible values over D : isocontours may appear, disappear, join, split, touch the boundary or change genus. The contour tree stores this evolution and provides a compact description of the properties and structure of the scalar field.

From the perspective of the survey, the focus here is on the study of the function that defines the shape, the scalar field itself, rather than on the variation of the description with respect to changes of the mapping function. Contour trees and their properties are grounded in the Morse theory of critical points and are closely related to the more general Reeb graphs. The two structures, however, are surveyed separately as the literature on the topics also evolved separately. A discussion on their relationships will be given in Section 10.

5.1 Theoretical aspects

In the literature, several slightly different definitions of contour trees have been introduced. All of them reflect the intuition that each connected component of the level sets of the scalar field is contracted to a point and the contour tree represents the *events* in their evolution, as the isovalue varies in the range of possible values. These events, which correspond for example to the creation, union, or disappearance of isocontours, are closely related to the presence of critical points of the scalar field. In a more general case it has been demonstrated that, given a smooth 2-manifold M and a Morse function f on M , when the isovalue spans a range of values containing a critical value of f then the isocontours change (see Section 3.1.5 and [Gramain

1971]). In contrast, a change in the topology of the level sets locates a critical value of f . The same results can be derived for piecewise linear functions [Chiang et al. 2005].

The differences in the surveyed definitions of contour trees mainly depend on the type of evolution, that is on the type of critical point, stored in the structure. The contour tree typically keeps track of the critical points in which *only* the number of components of the level set varies, but not the genus of isocontours. For two-dimensional scalar fields this situation does not occur, but for three-dimensional scalar fields there are critical values at which the topological genus of the isosurface changes without modifying the number of connected components nor the adjacency of the isocontours, see Figure 14.

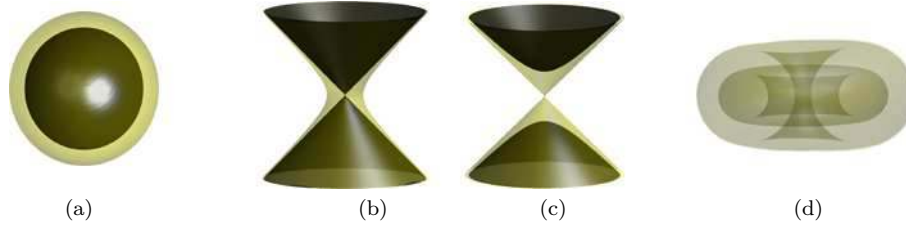


Fig. 14. Isosurfaces around a minimum (a) and a saddle (b-c). At the saddle point in (d), a torus evolves into a sphere changing the genus of the isosurface without altering the number of components (1) of the level set.

In this survey, we have decided to adopt the term *component-critical points* to denote critical points at which only the number of connected components of the level set varies, as used for example in [Chiang et al. 2005]. In the literature, the terminology used to distinguish different types of critical points with respect to changes in the level sets is quite inhomogeneous: for example, in [Carr 2004] critical points are named *Morse-critical* points while the component-critical points are simply called critical points.

In the following, we use a Morse-theoretic definition of contour tree inspired by the one proposed in [Carr 2004], see also theorem 3.2. Given a scalar field $\Gamma = (D, f)$, with f Morse, two isocontours \mathcal{C} and \mathcal{C}' are said to be *equivalent* if there exists some f -monotone path α in D that connects some point in \mathcal{C} with another in \mathcal{C}' such that no point $x \in \alpha$ belongs to a contour of any component-critical points of f [Carr 2004]. The classes induced by this equivalence are called *contour classes*. Contours that include critical points are the sole members of their (finite) contour classes. In contrast, infinite contour classes correspond to open intervals and represent a set of contours of essentially identical connectivity. Then, the *contour tree* is a graph (V, E) such that:

- (1) $V = \{v_i | v_i \text{ is a component-critical point of } f\}$;
- (2) for each infinite contour class created at a component-critical point v_i and destroyed in another component-critical point v_j , $(v_i, v_j) \in E$.

Finally, it is assumed that an arc (v_i, v_j) is directed from the higher to the lower value of f on it. Figure 15 shows the contour tree of a two-dimensional scalar field.

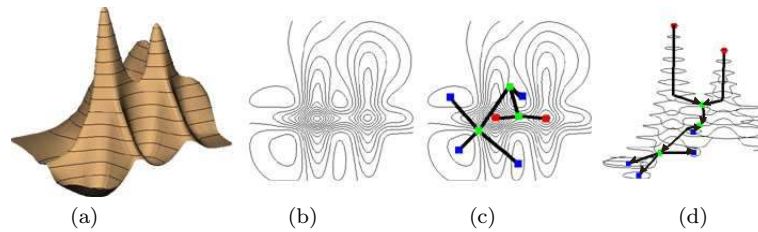


Fig. 15. A two-dimensional scalar field (a,b) and its contour tree (c,d). The edge orientation and the spatial embedding of the contour tree are shown in (d).

It should be noted that, while the domain D of the scalar function f is supposed to be any compact subset of \mathbb{R}^n , in practice, only simply connected domains are considered, like rectangles or parallelepipeds [Pascucci and Cole-McLaughlin 2003; Chiang et al. 2005]. This directly influences the type of connectivity that the contour tree may assume, since cycles cannot appear and therefore the structure is that of a tree.

The contour tree may be also viewed as the dual graph of the regions bounded by the level sets and by the boundary of D [Carr 2004]. Contours that intersect the boundary of the domain D may be thought of as manifolds with boundary. Two main approaches to the analysis of the boundary have been proposed: a local classification of the boundary vertices, as in [Chiang et al. 2005], or the setting of the function f at $-\infty$ (or $+\infty$) outside D [Cox et al. 2003]. The latter case is equivalent to closing the boundary with a virtual root of the graph [Takahashi et al. 1995; Biasotti et al. 2000a], and making the image of the scalar field homeomorphic to a (hyper)sphere [Griffiths 1976]. Since critical points (and therefore the nodes of the contour tree) depend on the interpretation chosen, these two approaches generate contour trees that differ along the contours that intersect the boundary [Mizuta and Matsuda 2005]. In particular, the virtual closure forces an interpretation of the scalar field that could be a limit when both positive and negative values of the scalar field are significant.

Several variations of the contour tree have been proposed in the literature. For example, when resolving all multiple critical points into simple ones, Takahashi et al. [Takahashi et al. 2004a] named the contour tree of a 3D scalar field as the *volume skeleton tree*. Moreover, the contour tree may be enriched with further information on all topological changes of the level sets by adding nodes that correspond to critical values where not the number but the topology of the contours changes. This tree was first introduced in [Pascucci and Cole-McLaughlin 2002] with the name of *augmented contour tree*, and renamed *contour topology tree* by [Chiang et al. 2005] to avoid confusion with the augmented contour tree described in [Carr et al. 2000], which denotes the refinement of the contour tree with the inclusion of all isocontours traced at the vertices of the input mesh. Other variations of the contour tree are the *criticality tree* [Cox et al. 2003], which corresponds to the subpart of the contour tree that codes only the contour joins, and the *topographic change tree* [Giertsens et al. 1990]. Finally, in the context of image processing, analogous structures are the *region-based contour tree* [Mizuta and Matsuda 2005], the *component tree* [Coupric and Bertrand 1997; Jones 1999], the *fast level lines*

transform [Monasse and Guichard 2000], the *max tree* [Salembier et al. 1998] and the *scale-tree* [Bangham et al. 1998a].

5.2 Computational aspects

Algorithms for the computation of contour trees first appeared in the field of spatial data handling for coding the evolution of contour lines in topographic data. One of the first papers on this topic considers the nesting relationships of a set of polygonal contours manually extracted from a topographic map [Boyell and Ruston 1963]. The whole surface is enclosed by an outside region, so that each contour has an inside and outside region. Nodes representing contours are added to the tree one at time.

A more systematic approach to encode geographical data organized in a triangulation was also proposed in [de Berg and van Kreveld 1997]. After ordering the function values, the authors extract contours and keep track of their evolution by sweeping the data set twice: first, from the highest to the lowest isovalue, and then sweeping again in the reverse direction from the lowest to the highest. This method is specialized for two-dimensional scalar fields and runs in $O(n \log n)$ operations, where n is the number of vertices of the mesh. A simplification of the algorithm and its extension to higher dimensions was proposed in [van Kreveld et al. 1997]. The complexity remains $O(n \log n)$ for two-dimensional scalar fields and becomes $O(N^2)$ for higher dimensions, where N is the number of cell higher-dimensional simplices of the mesh. In this class of methods, the vertices and the edges of the tree are called *super-nodes* and *super-arcs*, while nodes are introduced along super-arcs to represent regular points. The function f is supposed to be general in order to guarantee that all super-nodes are simple, i.e., the function f is injective on them. If f is not general, the original data are perturbed. The algorithm in [van Kreveld et al. 1997] has been further improved for 3D scalar fields by Tarasov and Vyalı [Tarasov and Vyalı 1998] by showing that the re-labeling process can also be done efficiently in three dimensions (it requires $O(N \log N)$ operations), and by extending the pre-processing of complex saddles.

Carr et al. [Carr et al. 2000; 2003; Carr 2004; Carr and Snoeyink 2007] propose a generalization of the contour tree extraction for higher dimensions, which simplifies and extends the method in [Tarasov and Vyalı 1998]. In this case, contours are not explicitly maintained and the pre-processing of the multi-saddles and of the surface boundary is not required. In this approach, the two sweeps are used explicitly to code the connectivity of the upper and lower level sets, $\{x : f(x) \geq h\}$ and $\{x : f(x) \leq h\}$ according to the definitions in Section 3.1.5, in a join tree and a split tree, which represent the connectivity of the two sets respectively, as shown in Figure 16. Finally, the contour tree is assembled by picking local extrema from the join and split trees and transferring them into a so-called augmented contour tree that contains all mesh vertices. As an optional step, regular points may be removed so that the contour tree is explicitly coded. The most efficient implementation of this method requires $O(C \log C + N\alpha(N))$ operations [Carr 2004], where C is the number of nodes of the tree, N is the number of higher-dimensional simplices and α is the inverse of the Ackermann function [Ackermann 1928].

In [Pascucci et al. 2004], the method in [Carr et al. 2000; 2003] is modified in order to store the contour tree in a multi-resolution fashion. The contour tree is

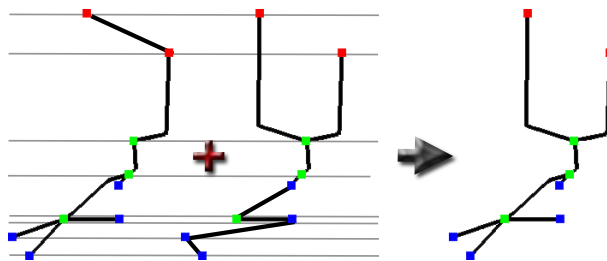


Fig. 16. The join tree and the split tree (left) of the contour tree (right) of the scalar field in Figure 15.

hierarchically decomposed into a set of branches that may identify more nodes and arcs. The decomposition is not unique but, once a hierarchy has been extracted, it is possible to generate different approximations of the original tree by adding branches to a root. To obtain this representation, the merge of the join and split trees is modified considering as atoms of the operations the branches of the trees instead of the leaves. In particular, branches to be inserted in the hierarchical contour tree are selected according to a priority queue that measures the length (i.e. the difference in function values of its end-points) of each branch. Even if the algorithm requires a multi-resolution data structure, the procedure for computing the hierarchical contour tree has the same complexity of [Carr et al. 2000], that is $O(N \log N)$. However, this cost may be improved to $O(N)$ by using a FIFO queue instead of a priority queue, eventually generating a unbalanced tree, as discussed in [Pascucci et al. 2004].

An algorithm that computes the contour tree for 2D and 3D scalar fields optimally, both in space and time, has been proposed in [Chiang et al. 2005]. Here the sweep algorithm described in [Carr et al. 2003] is slightly modified, combined with an analytic approach and changed in the way the join and the split tree are obtained. Instead of ordering all mesh vertices, Chiang et al. first characterize the critical points, and then sort and connect them through monotone paths. The component-critical points are identified through an initial unordered scan of the vertices that analyses the neighborhood of each vertex. Only the component-critical points are ordered, and this improves the computational complexity to $O(N + c \log c)$, where N is the number of higher-dimensional simplices and c the number of component-critical points of the mesh. Therefore the time complexity depends on the output size. In particular, [Berglund and Szymczak 2004] proposed an encoding technique that, beside the contour tree of the whole domain, is able to return also a representation of the contour tree of its sub-domains.

Another extension of the algorithm in [van Kreveld et al. 1997] to 3D domains has been proposed in [Pascucci and Cole-McLaughlin 2002] and further extended in [Pascucci and Cole-McLaughlin 2003]. In this case, the domain D is split into subparts and the isosurfaces are separately processed using a parallel approach, based on a divide-and-conquer paradigm. This paradigm is used to compute both the join and the split trees (it is worth noticing that in these papers the notions of join and split tree are used inversely from [Carr et al. 2003]). In addition, a more detailed characterization of the contours is achieved coding the Betti numbers

associated to each arc of the tree. Finally, the different trees are merged according to adjacency of the subparts of D , eventually revising the Betti numbers of each component. The resulting tree is called *augmented contour tree* because all the critical points of the scalar field correspond to nodes of the tree and the contours associated with an arc contain no critical points. The complexity of the algorithm for a 3D regular mesh, that is a triangle mesh obtained from a regular grid, is $O(n + c \log n)$ where n is number of vertices of the mesh and c is the number of critical points. Even if the method works on meshes whose vertices do not lie on a grid, the split of the mesh into subparts for the parallel computation does not improve the complexity of this method, which still is $O(n \log n)$, see discussion in [Chiang et al. 2005].

The method proposed in [Takahashi et al. 1995; Takahashi 2004] is devised for building contour trees of two-dimensional scalar fields represented by a regular grid, that is converted to a triangulation by splitting each cell with its diagonal; more details on how to triangulate a regular grid are in Section 3.2.1. Note that the authors refer to their contour structure as a Reeb graph. A surface network, see Section 4.1, is extracted from the grid and used as intermediate structure for constructing the contour tree. In fact, the authors prove that the contour tree of a two-dimensional scalar field may be deduced by its surface network. This method is mainly analytic because it identifies the critical points on the vertices of the triangulation by analyzing the star of each vertex and simulates the paths of steepest descent on the model. A global virtual minimum acts as the root of the tree giving a unique interpretation of the surface behavior along its boundary. The computational complexity of this algorithm was not stated by the authors, but its analysis is given in [Chiang et al. 2005], where it is claimed that the method requires $O(N)$ operations for reading the data, $O(nc)$ operation for finding the paths and $O(c^2)$ time for constructing the tree, where N , n and c are respectively the number of triangles, vertices and critical points of the mesh. This approach has been extended to volumetric gridded data [Takahashi et al. 2004b] represented by tetrahedral cells, using a voxel flow network as support during the contour tree extraction instead of a surface network. Moreover, the method admitted the removal of critical points whose relevance (in the sense of length of the adjacent arcs) is irrelevant to describe the global topological structure of the volume. Even if the authors do not provide the computational complexity of their algorithm, it seems reasonable that the computational cost may be expressed with the same expression as in the two-dimensional case, considering that the number N of triangles is related also to tetrahedra. Finally, the method in [Takahashi et al. 2004a] combines three approaches: the extraction of the contour tree in [Carr et al. 2003] is joined together with the analysis of the genus of each isosurface component proposed in [Pascucci and Cole-McLaughlin 2002] and simplified according to the procedure described in [Takahashi et al. 2004b].

The method proposed in [Itoh and Koyamada 1995] automatically detects a graph from a three-dimensional mesh. The authors remove simplices (i.e. vertices, edges, faces and tetrahedra) while preserving the connection between the mesh critical points. In this case the nodes of the tree might differ from the critical points and the edges are compound of a generic monotonic path between two nodes. However,

this method is computationally efficient because it is linear and performs in $O(n)$ operations, where n is the number of mesh vertices.

The approach proposed in [Biasotti et al. 2000a] extracts a contour tree, called an *Extended Reeb Graph (ERG)*, from two-dimensional scalar fields described by a set of isocontours. Starting from these contours, a Delaunay constrained triangulation is built, forcing the isocontours (in the sense of both edges and vertices) to belong to the mesh. The function f of the scalar field (in this case the height of the vertices) is not required to be either Morse or simple and, therefore, the method can deal with multi-saddles, plateaus and volcano rims. Similarly to [Takahashi et al. 1995], the surface is virtually closed introducing a global virtual minimum. Since it is supposed that all vertices in the mesh lie on contours, the authors use a characterization of the triangles which is based on the number of vertices with the same elevation. Based on this characterization, *critical areas* are defined as areas containing critical points. Then, the tree is built using a region growing process that starts from the critical areas, spreads them in all directions and completely covers the surface keeping track of the contours crossed. Unlike the approach in [van Kreveld et al. 1997], the region growing process is based on a visit of mesh triangles which is linear in the number of triangles, $O(N)$. Therefore, the computational complexity of the method depends on the construction of the Delaunay triangulation constrained to contours, that requires $O(n \log n)$ operations, where n is the number of vertices of the isocontours in input. An extension of this approach has been proposed in [Biasotti et al. 2004]: here the classification of the critical areas and the graph construction do not require that the triangle mesh is originally constrained to the level sets. However, the successive insertion of the contours in the mesh increases the computational complexity of the method to $O(\max(m + n, n \log n))$, where n is the number of vertices of the original mesh and m is the number of vertices added during the insertion of the level sets (in the worst case m may be $O(n^2)$). The extension of the method to closed surfaces [Attene et al. 2001; 2003] and to generic two-dimensional surfaces [Biasotti 2004b] is discussed in Section 6.2, which is devoted to Reeb graphs.

Cox et al. [Cox et al. 2003] propose a variant of the contour tree called the *criticality tree* which is based on the analysis of the isosurfaces without relying on the classical Morse theory. In particular, the authors develop a discrete theory, called *digital Morse theory*, in order to disambiguate the characterization of the isosurface evolution on cubic grids and to include non-general functions. In practice, the criticality tree proposed in this work is a join tree augmented by the component-critical points rather than a contour tree. In fact the criticality tree stores the evolution of the volumes that, starting from maxima, are bounded by the isosurfaces. These volumes are denoted topological zones and are locally nested. Due to the need of extracting the topological zones, the computational complexity of this method is $O(kN \log(kN))$, where N is the number of 3-cells and k is the length of the longest path traversed in the tree.

In the field of image processing, several variations of the contour tree have been adopted. The *region-based* contour tree proposed in [Mizuta and Matsuda 2005] transposes the classical contour tree definition to gray-level images. The key idea behind this method is that the changes in the isosurfaces (i.e. the variations in

the number of connected components of pixels at a given gray-level) are directly encoded in a tree. Therefore, the nodes of the tree correspond to regions of pixels. The algorithm adopts an extraction procedure analogous to that proposed in [Carr et al. 2003] which runs in $O(p \log p)$ operations, where p is the number of pixels of the image. However, since the pixels assume only integer values, the authors claim that, if the gray-level values vary between 8 and 16, a bucket-sort algorithm to order the pixels would decrease the computational cost to be $O(p)$.

The *component tree*, also known as *confinement tree* or *dendrone* [Jones 1999; Couprie and Bertrand 1997; Mattes and Demongeot 2000], displays all image components in a hierarchical sequence. Gray-levels are therefore piled one on the other in a graph, which is a useful structure for further filtering. This hierarchy implies that the component tree is a variation of the join tree rather than a contour tree. This description has been successfully used for image segmentation and has been proved to achieve better results in comparison with standard connected filters [Breen and Jones 1996]. An efficient algorithm for the extraction of component trees is proposed in [Mattes and Demongeot 2000] where the global computational complexity performs in $O(p \log p)$ operations, where p is the number of pixels of the image. Closely to the component tree, the *Fast Level Lines Transform (FLLT)* codes the evolution of the connected components of the gray levels of an image [Monasse and Guichard 2000]. Notice that the sorting of the components is obtained checking the geometric inclusion of the level sets, thus discarding the arc orientation induced by the increase/decrease of the gray-level intensity.

Another popular structure in image processing is the *max-tree* introduced by Salembier et al. [Salembier et al. 1998], and its dual, the *min-tree*. Similarly to component tree, the max-tree encodes in a tree the hierarchy of the connected level set of pixels. The main difference between a component tree and a max-tree does not involve the topology (i.e., the connection among the nodes), but the construction process and the kind of information stored in each node. An efficient method for storing and building the max tree has been proposed in [Huang et al. 2003]. This method performs in linear time in the number of pixels $O(p)$ and combines the dynamic allocation of the memory in a linked list with the node tree storage provided by a hash table. The *scale-tree* in [Bangham et al. 1998b; Bangham et al. 1998a] is also based on the level-set image decomposition. In particular, the scale-tree automatically selects a number of scales and codes the regions with slightly different attributes, like their amplitude, shape and position so that the coding is invertible.

More recently, the interest for time-dependent data sets has increased leading to the introduction of methods for extracting the graph for time-varying models [Szymczak 2005; Sohn and Bajaaj 2006]. The main innovation of these methods is not in the extraction algorithm itself (they refer to [Pascucci and Cole-McLaughlin 2003] and [Carr et al. 2003], respectively) but in the way contours of 2D and 3D data sets join and split during a time interval. These methods also investigate the relationships between the contour tree over the entire domain D and those restricted to its sub-domains. Contour trees are computed in a pre-processing phase and the focus is the definition of efficient structures and algorithms for the query execution. For example, the computational complexity of the query algorithm in [Szymczak

2005] is $O(s(1 + \log(t_1 - t_0)))$, where s denotes the maximum size of a contour tree and t_0, t_1 are two time values. In [Sohn and Bajaj 2006] the correspondence between the nodes of the contour tree is stored in a graph called a *topology change graph* that supports visualization of contour evolution. The computational complexity for extracting the topology change graph is $O(n \log n + N + (c_t)^2 c_{t+1})$, where n and N are respectively the number vertices and tetrahedra of the mesh and c_t is the number of critical points of f at time t .

A summary of methods for contour tree extraction is proposed in Table IV.

Contour tree					
Approach	Method	Domain			Costs
		2D	3D	nD	
[Boyll and Ruston 1963]	Manual	X			-
[Itoh and Koyamada 1995]	Erosion	X	X		$O(n)$
[de Berg and van Kreveld 1997]	Sweep	X			$O(N \log N)$
[van Kreveld et al. 1997; 2004]	Sweep ¹	X	X		$O(N \log N) / O(N^2)$
[Tarasov and Vyalyi 1998]	Sweep	X	X		$O(N + n \log n)$
[Carr et al. 2000; 2003]	Sweep ²	X	X	X	$O(N \log N)$
[Carr 2004]					$O(C \log C + N\alpha N)$
[Pascucci et al. 2004]	Sweep ³				$O(N \log N)$ $O(N)$
[Chiang et al. 2005]	Sweep	X	X		$O(N + c \log c)$
[Pascucci and Cole-McLaughlin 2002]	Sweep ⁴	X	X		$O(n + c \log n) /$ $O(n \log n)$
[Pascucci and Cole-McLaughlin 2003]					
[Takahashi et al. 1995]	Analytic	X			$O(N) + O(nc) + O(c^2)$
[Takahashi 2004]		X			
[Takahashi et al. 2004b]	Analytic		X		$O(N) + O(nc) + O(c^2)$
[Biasotti et al. 2000a]	Contours	X			$O(n \log n)$
[Biasotti et al. 2004]	Contours	X			$O(\max(m + n, n \log n))$
[Cox et al. 2003]	Isosurfaces		X		$O(kn \log(kn))$
[Mizuta and Matsuda 2005]	Level sets ⁵	2D Images			$O(p \log p)$
[Mattes and Demongeot 2000]	Level sets	2D Images			$O(p \log p)$
[Bangham et al. 1998b]	Level sets	2D Images			$O(p \log p)$
[Bangham et al. 1998a]					
[Huang et al. 2003]	Level sets	2D Images			$O(p)$

Table IV. Classification of the methods for extracting the contour tree. Symbols: N is the number of higher-dimensional simplices or cells; n is the number of vertices or points; m is the number of vertices inserted in the mesh during contouring phases; c is the number of critical points; C is the number of tree nodes; α is the inverse of the Ackermann function; k is the length of the longest path traversed in the tree and p is the number of pixels. Note that, in the 2D case, $N = O(n)$.

¹The complexity is, respectively, $O(N \log N)$ for 2D and $O(N^2)$ for 3D domains.

²The method is essentially the same in both papers. The cost improvement is due to a more efficient implementation of the data structures.

³The complexity of this method depends on the use of a FIFO or a priority queue.

⁴The complexity of this method varies if considering regular or simplicial meshes.

⁵The complexity of the method may decrease to $O(p)$ if the range of pixel values is between 8 and 16.

5.3 Applications

The original application of contour trees was in topography [Boyll and Ruston 1963], Geographic Application System (GIS) applications [Takahashi et al. 1995; Takahashi 2004] and surface analysis and understanding [Biasotti et al. 2000a; 2004]. Another possible application is shape matching and retrieval [Zhang et al. 2004].

As far as three-dimensional scalar fields are concerned, the main applications of contour trees are image processing and analysis of volume data sets [Cox et al. 2003; Takahashi et al. 2005], volume simplification [Chiang and Lu 2003; Carr et al. 2004; Weber et al. 2007], automatic isosurface propagation [Itoh and Koyamada 1994; 1995; Itoh et al. 2001] and scientific visualization [Pascucci and Cole-McLaughlin 2002; Pascucci 2004; Cox et al. 2003; Takahashi et al. 2004b]. Recently, Bajaj et al [Bajaj et al. 2007] used the contour tree for molecular data analysis. The scalar field for the contour tree extraction is defined by the distance function from the surface of the molecule, that is able to detect the secondary structural motifs of a protein molecule [Bajaj and Goswami 2006]. This allows the identification of tunnels and pockets of the molecule. Finally, contour trees in higher dimensions apply to X-ray analysis and visualization [Carr et al. 2003] and to scientific visualization [Pascucci 2004].

The contour tree is an effective tool for representing the abstract structure of the scalar field with an explicit description of the topological changes of the level sets. Beside the *contour spectrum* [Bajaj et al. 1997], that is a set of geometric measures (like length, area, gradient integral, etc.) that are computed over the level sets of the scalar field, the contour tree has been proposed as an element of an user interface able to drive interactive data exploration sessions [Kettner et al. 2003; Carr and Snoeyink 2003; Carr 2004; Takahashi et al. 2005].

In practical applications, the fact that most of the algorithms extract a representation that may be overwhelming in size and that a planar embedding of the tree may generate many self-intersections has limited the use of the contour tree. To overcome the problem of the size of the graph, the work in [Pascucci et al. 2004] proposes a method for extracting a multi-resolution contour tree and visualizing the tree subparts, that are called branches. Since a branch of the tree may be drawn as a chain of connected arcs, branches are sorted according to their importance into the contour tree and progressively rendered. The representation of the tree is embedded into the 3D space by moving each node to a z value that corresponds to the value of the scalar field in that node. In addition, 2D self-intersections are avoided using the algorithm for rooted trees proposed in [di Battista et al. 1999]. More recently, these branches have been used to topologically simplify the rendering of complex volumetric data sets [Carr et al. 2004; Weber et al. 2007].

In the field of image processing, contour trees [Mizuta and Matsuda 2005; Turner 1998] and their variations [Jones 1999; Monasse and Guichard 2000; Salembier et al. 1998; Bangham et al. 1998a] have been mainly used for optimizing the coding and the manipulation of images and their meaningful components. While the topology of image surfaces is simple (an image may be easily represented as a scalar field whose scalar function is provided by the gray-level intensity), the configuration of contour trees provides an interesting support for many filtering and segmentation

operations. In particular, the max tree has been adopted in a number of Computer Vision problems including stereo matching, image filtering, segmentation and information retrieval.

6. REEB GRAPHS

Similar to contour trees, the main idea behind Reeb graphs is to encode the evolution and the arrangement of the level sets of a real function defined on a shape. While the definition and use of contour trees developed mainly as an answer to computational issues, Reeb graphs have a more theoretical nature. They originated in 1946 in the work of a French mathematician, George Reeb. In recent years, Reeb graphs have become popular in Computer Graphics as tools for shape description, analysis and comparison.

Reeb graphs present a framework for studying the shape of a manifold: here the shape exists by itself and the function used to study its shape can be arbitrarily chosen. Different functions can be used for extracting a structure that effectively codes the shape from both a topological and geometrical perspective. Topology here means that the shape can be described as a configuration of parts that are *attached* together respecting the topology of the shape, while geometry means that the different parts correspond to features of the shape, as embedded into the Euclidean space, that have specific properties and descriptive power (e.g., protrusions, elongated parts, wells).

6.1 Theoretical aspects

Reeb graphs were first defined by Georges Reeb in 1946 [Reeb 1946] as topological constructs. Given a manifold M and a real-valued function f defined on M , the simplicial complex defined by Reeb, conventionally called the Reeb graph, is the quotient space defined by the equivalence relation that identifies the points belonging to the same connected component of level sets of f . Under some hypotheses on M and f , Reeb stated the following theorem, which actually defines the Reeb graph.

THEOREM 6.1. *Let M be a compact n -dimensional manifold and f a simple² Morse function defined on M , and let us define the equivalence relation “ \sim ” as $(P, f(P)) \sim (Q, f(Q))$ iff $f(P) = f(Q)$ and P, Q are in the same connected component of $f^{-1}(f(P))$.*

The quotient space on $M \times \mathbb{R}$ induced by “ \sim ” is a finite and connected simplicial complex K of dimension 1, such that the counter-image of each vertex Δ_i^0 of K is a singular connected component of the level sets of f , and the counter-image of the interior of each simplex Δ_j^1 is homeomorphic to the topological product of one connected component of the level sets by \mathbb{R} [Reeb 1946; Ehresmann and Reeb 1944].

Reeb also demonstrated the following theorems, which clarify the relations between the degree, or order, of the vertices of the simplicial complex K associated with the quotient space and the index of the corresponding critical point.

THEOREM 6.2. *The degree of a vertex Δ_i^0 of index 0 (or n) is 1 and the index of a vertex Δ_i^0 of degree 1 is 0 or n .*

²A function is called simple if its critical points have different values

THEOREM 6.3. *If $n \geq 3$ the degree of vertices Δ_i^0 of index 1 (or $n - 1$) is 2 or 3. If $n = 2$ the degree of vertices Δ_i^0 of index 1 is 2, 3, or 4. The degree of vertices Δ_i^0 of index different from 0, 1, $n - 1$, or n is 2.*

In other words, the first theorem states that leaf nodes of K can be either maxima or minima of f , while from the second theorem we can deduce that, for 2-manifolds that can be embedded in \mathbb{R}^3 , the degree of vertices representing saddles is always 3, see Section 3.1.5.

To the extent of our knowledge, Reeb graphs were first introduced in Computer Graphics by Shinagawa et al. [Shinagawa et al. 1991] and the term Reeb graph is used to identify the simplicial complex associated with the quotient space. As a consequence of their ability to extract high-level features from shapes, since their introduction in Computer Graphics Reeb graphs have been gaining popularity as an effective tool for shape analysis and description tasks, especially in case of 2-manifolds.

The Reeb graph has been also used for the analysis of 3-manifolds with boundary. In this case the structure of the 3-manifold is studied either by introducing a virtual closure of the manifold [Edelsbrunner et al. 2004], or by associating a Reeb graph to each 2-manifold boundary component of the 3-manifold and keeping track with a supplementary graph of the changes between interior and void [Shinagawa et al. 1991; Shattuck and Leahy 2001].

For orientable, closed 2-manifolds, the number of cycles in the Reeb graph corresponds to the genus of the manifold, and this result has been generalized in [Cole-McLaughlin et al. 2003], where the authors demonstrate that the number $\beta_1(M)$ of non-homologous loops of the surface is an upper bound of the number of loops $\beta_1(K)$ of the Reeb graph. The equality holds in case of orientable surfaces without boundary [Cole-McLaughlin et al. 2003], while, in general, the following relation holds:

$$g \leq \beta_1(K) \leq 2g + b_M - 1,$$

where b_M denotes the number of boundary components of the 2-manifold M having genus g . Theoretical results are available for non-orientable 2-manifolds. In this case, the number of loops of the Reeb graph verify the following relations: $0 \leq \beta_1(K) \leq \frac{g}{2}$ when M is closed, and $0 \leq \beta_1(K) \leq g + b_M - 1$ for manifolds with boundary.

As for 3-manifolds, it is not true that the number of the loops of an orientable, closed 3-manifold is independent of the mapping function f . In addition, it has been proven that for every 3-manifold M there exists at least one Morse function f such that the Reeb graph of M with respect to f is a tree [Cole-McLaughlin et al. 2003].

The extension of Reeb graphs to shapes defined by piecewise linear approximations has been studied by several authors. For example, in [Biasotti 2004a] the definition of Reeb graph was extended to triangle meshes representing 2-manifolds embedded in \mathbb{R}^3 , with or without boundary, and which admit degenerate critical points. The relationship between the genus of the mesh and the cycles in the extended Reeb graph is maintained, as discussed in [Biasotti 2004a; 2004b]. On the basis of this representation, a further extension of the domain of the Reeb graph to point clouds was proposed in [Werghi et al. 2006], defining a so-called discrete

Reeb graph.

Figure 17 shows two examples of Reeb graphs of a closed surface. In Figure 17(a) some level sets of the height function are drawn with the corresponding Reeb graph; in Figure 17(b) the Reeb graph of the same object is shown using the Euclidean distance from a point.

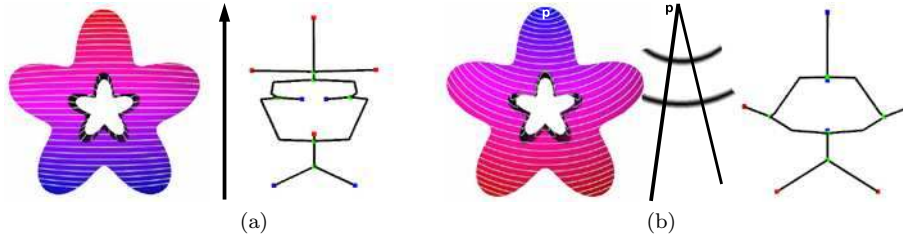


Fig. 17. Reeb graph with respect to the height function (a) and to the distance from a point (b).

6.2 Computational aspects

Several algorithms have been proposed for the computation of the Reeb graph of closed surfaces, while only a few algorithms deal with 3-manifolds, higher-dimensional or time-dependent data.

The first algorithm, proposed by Shinagawa et al. [Shinagawa and Kunii 1991], automatically constructs the graph from surface contours generated by the height function. Since the contour ordering proposed in [Boyell and Ruston 1963] is not suitable for manifolds without boundary, a weight function, which depends on the average distance between the vertices of two different contours, is defined for each pair of contours lying on adjacent (consecutive) level sets. First, the algorithm automatically generates most of the arcs of the Reeb graph where the number of contours of two consecutive cross sections is one. Then the rest of the graph is determined by using the weight function and *a priori* knowledge of the surface genus. Specifically, the graph is completed by adding edges in decreasing order of the weight between contour pairs, so that the genus of the graph preserves that of the original surface. The main drawbacks of this algorithm are the need for *a priori* knowledge of the genus of the surface and the fact that this procedure is limited to contour levels of the height function. In addition, since this algorithm loses the shape information between two consecutive cross sections, the frequency of the contours of the surface is critical; therefore, a reasonable computation of the graph requires a high number of surface slices and it is time and space consuming ($O(n^2)$, where n represents the total number of vertices of the scattered contours). To deal with models with cavities, Shinagawa et al. encoded the Reeb graphs of both the shape and its complement. Similar considerations on the evolution of the model and its complement have been proposed in [Shattuck and Leahy 2001] to define a method for the extraction of a topological graph for cortical volume data.

The method proposed in [Hilaga et al. 2001] provides a *multi-resolution Reeb graph (MRG)* representation of triangle meshes which is independent of the object topology. The construction of the MRG begins with the extraction of the graph at

the finest resolution desired, then adjacency rules are used to complete the multi-resolution representation in a fine-to-coarse order. First of all, the domain of the mapping function is divided into a number of intervals and triangles whose image under f lies in two intervals are subdivided so that the image of every triangle belongs to only one interval. Second, the triangle sets, that is the sets of connected components of triangles whose images belong to the same interval, are calculated. A node of the graph is associated with each triangle set and arcs are detected by checking the region adjacency of triangle sets. It is worth noticing that this contouring approach induces a quantization of the interval of f that, although it may locally amend the local topological noise, does not guarantee that the number of loops of the MRG equals the number of holes of the surface. Once the function f has been evaluated on the vertices of the mesh and triangles have been split, the Reeb graph extraction requires $O(n+m)$ operations, where n represents the number of triangles of the original mesh and m represents the number of triangles inserted during the subdivision phase. Notice that the evaluation of the function f may be a computationally expensive step. For example, the exact computation of the geodesic function proposed in [Hilaga et al. 2001] requires $O(n^2 \log n)$ operations, while its approximation runs in $O(kn \log n)$, where k is a user-defined constant (usually greater than 150). In Figure 18 an example of the Reeb graph construction method proposed in [Hilaga et al. 2001] is shown; in this case the domain of f is subdivided in 4 intervals. The contour insertion in Figure 18(b) determines a set of mesh regions that correspond to the graph nodes in Figure 18(c), while their adjacency originates the arcs of the graph (see Figure 18(d)).

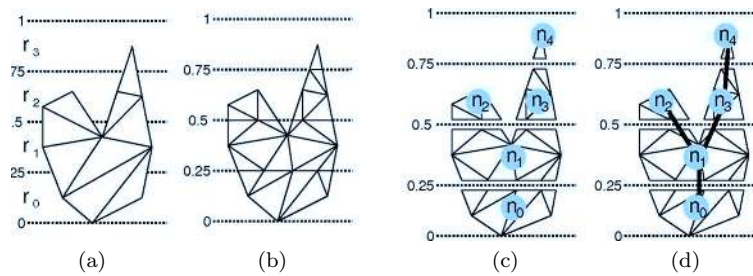


Fig. 18. Pipeline of the multi-resolution Reeb graph extraction in [Hilaga et al. 2001].

A general algorithm for the Reeb graph extraction of 2-manifolds with or without boundary represented by a simplicial complex was proposed in [Cole-McLaughlin et al. 2003]. This approach also works for non-orientable models, like the Klein bottle. The basic assumption here is that the mapping function is Morse and simple, so that critical points have pairwise different function values. Then, the Reeb graph is constructed by storing the level sets while sweeping the domain of the function. To identify the critical points, the star of each vertex is classified according to the approach in [Edelsbrunner et al. 2003b]. Once critical points have been detected, all vertices of the model are processed according to the increasing value of the function f and the evolution of level sets is tracked. Since operations are done on the edges,

the complexity of the algorithm is $O(n \log n)$, where n is the number of edges of the complex.

Recently, Pascucci et al. [Pascucci et al. 2007] proposed an algorithm able to compute efficiently the Reeb graph of simplicial complexes of arbitrary dimension. This algorithm is based on the assumption that the Reeb graph of the simplicial complex is equivalent to the Reeb graph of its 2-skeleton. This assumption implies that the structure obtained from this algorithm slightly differs from the original definition given by Reeb [Reeb 1946] because it is not able to distinguish changes in the topology of the isosurfaces. For instance, the 2-skeleton of a simplicial complex of dimension 3 is not sensitive to the inner cavities of the model. The usage of a stream approach eliminates the need of an initial ordering of the mesh vertices, allows a progressive computation of the graph able to encode very large models and provides a way to compute a run-time simplification of Reeb graph loops based on their relevance. To achieve these tasks, two structures are used: one for the input 2-skeleton and the other for the Reeb graph. These two structures are related to each other and, when a new element is inserted in the 2-skeleton, the Reeb graph is consequently updated by creating new nodes and arcs and merging of two paths. In particular, two paths are merged if, during the progressive visit of the 2-skeleton, a hole is filled or a loop is removed from the Reeb graph. The computational complexity of the algorithm is still $O(n \log n)$, where n represent the number of vertices, which is the theoretical lower bound complexity for the Reeb graph extraction. However, the usage of the stream approach makes this method efficient in practice, allowing a fast computation of the Reeb graph also on large meshes.

The *Extended Reeb Graph (ERG)* representation proposed in [Attene et al. 2001; 2003; Biasotti 2004a] is able to represent a surface with or without boundary through a finite set of contour levels of a given mapping function f . Besides the combinatorial representation of the graph, the ERG is embedded in \mathbb{R}^3 by associating with each node the position of the barycenter of the corresponding region and visualized as a centerline skeleton of M . By considering a partition of $\text{Im}(f)$ based on a finite number of values of f (f -values) provided in input, an extended form of Reeb equivalence relation is defined as the equivalence that identifies all points that are in the pre-image of an interval, or an f -value, and belong to the same connected component of the model. The ERG extraction is based on a generalized surface characterization aimed at a region-oriented rather than a point-oriented classification of the behavior of the surface. Since the level sets decompose M into a set of regions, the behavior of their boundaries is used for detecting regular or critical areas and for classifying them as maximum, minimum and saddle areas. Critical areas correspond to nodes of the graph. Then arcs between nodes are detected through an expansion process of the critical areas, which tracks the evolution of the contour lines. Since the ERG considers critical areas instead of critical points, it is also able to deal with degenerate configurations, such as volcano rims [Biasotti et al. 2004]. In [Biasotti 2004b], the method was extended to surfaces having an arbitrary number of boundary components. To obtain a minimal (in the sense of graph loops) representation of the ERG, this algorithm virtually closes all boundary components, as detailed in [Biasotti 2005]. In Figure 19 the main steps of the

ERG extraction process are depicted: first, the arcs between minima and saddles and between maxima and saddles are inserted, expanding all maxima and minima to their nearest (in terms of region expansion) critical area; then the other arcs are detected. The computational cost of the whole algorithm for the *ERG* extraction mainly depends on the insertion of the contours in the triangle mesh, which requires $O(n \log n)$ operations, and the region growing process, which is linear. Therefore, the computational cost of the overall graph construction is $O(\max(m + n, n \log n))$ where m is the number of vertices inserted in the mesh during the slicing phase (in the worst case m is $O(n^2)$).

An extension of the ERG definition to unorganized point clouds of 3D scan data that represent a human body has been proposed in [Werghi et al. 2006]. Since a polygonal mesh is not available, a surface is implicitly defined by assuming that the Euclidean distance among a point p and its closest point q is smaller than a given threshold ϵ . Point sets whose sampling is sufficiently fine are connected in a discrete sense. Therefore, level sets are defined as points that share the same value of a mapping function and are connected in the discrete sense. The resulting graph is called the *Discrete Reeb Graph (DRG)*. Once a set of level sets has been detected, the graph is progressively constructed visiting all points that are linked with respect to the threshold ϵ . The declared complexity of the graph computation mainly depends on the extraction of the level sets, which requires $O(nk^2)$ operations, where n is the number of points and k is the average of the points contained in the neighborhood of each point.

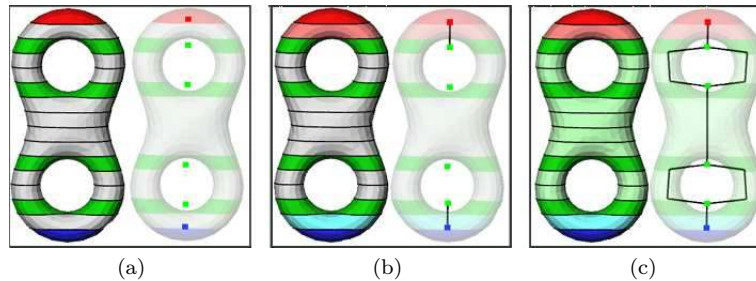


Fig. 19. Pipeline of the ERG extraction: the recognition of the critical areas (a), the expansion of maxima and minima (b) and the complete graph (c).

The approach proposed in [Wood et al. 2004] works on volume data. In this case, the data are swept with a plane that generates a sequence of *slices*, which are formed by the sets of grid elements bounded by two adjacent isosurfaces. Each connected component of a slice is called a *ribbon* while the *contours* are given by the intersection of the isosurfaces with a set of slicing planes. The graph described in this approach is called an *augmented Reeb graph* because it also encodes geometric information for each contour and each ribbon. The traversal is analyzed at discrete z intervals of the volumetric grid along the boundary of a distance function and may be done out-of-core on the dataset. Isosurfaces are analyzed one slice at time and contours are constructed by searching from an arbitrary edge in the plane until the contour is closed. Similarly, ribbons are constructed through a breadth-first

traversal of the slice elements, starting from the polygons adjacent to a contour until the connected component is completely detected. Both ribbons and contours correspond to nodes of the Reeb graph while their adjacency is coded in the edges. To avoid an object handle being completely contained within a ribbon, the Euler characteristic of each isosurface component is computed and, possibly, the sweep is locally refined. In this way the topology of the volume is completely coded and, in each interval, the Reeb graph structure corresponds to the Euler characteristic of the object ribbons.

In the analogous context of 3D binary images represented by voxel models, a graph similar to the Reeb graph has been proposed in [Shattuck and Leahy 2001]. Since in a volume model configurations with internal cavities are not fully described by the shape boundary, the 3D image is embedded in its bounding box and the graphs are extracted for both the object and its complement. A point is associated with each section. For each direction x , y and z , a foreground connectivity graph G is extracted. Since cycles are not admitted in the final graph (because cycles denote inner cavities and handles of the model to be removed as noise), a maximum spanning tree is computed to select the set of arcs that should be eventually removed from G . Then, graph tests are performed to simplify the graph until all cycles are removed. Although the computational cost of this algorithm was not provided by the authors, the computation of the maximum spanning tree seems to be the most time consuming operation. Therefore, the complexity of this method is $O(v \log v)$, where v represents the number of voxels of the cortical volume.

The *hyper Reeb graph* proposed in [Fujishiro et al. 1999; Fujishiro et al. 2000] deals with 3D volume fields. The main concept behind this description is the representation of the isosurfaces of a volume in terms of their Reeb graphs, and the detection of the topology changes of these isosurfaces through the modifications of their Reeb graphs. Once the volume data set is swept in the height direction and a Reeb graph has been extracted for each isosurface, this collection of graphs is encoded in a hyper graph, whose nodes correspond to the Reeb graphs. The Reeb graph of a single isosurface is extracted from the surface network of the isosurface extending the algorithm proposed in [Takahashi et al. 1995] for two-dimensional scalar fields. This computation costs $O(N) + O(nc) + O(c^2)$ operations where N and n represent the number of faces and vertices of the mesh and c is the number of its critical points (see Table IV in Section 5.2). Therefore, the computational complexity of the hyper Reeb graph is quite significant and depends on the number of isosurfaces taken into account.

Centerline skeletons play a relevant role in Computer Graphics and Vision, because they represent a manner to reduce a complex 3D shape to a simple one-dimensional geometric abstraction [Lam et al. 1992; Cornea et al. 2005]. In general, centerline skeletons are related to the medial axis in the sense that they yield a shape description that always falls inside the shape and aims to be in the *middle* of the volume enclosed by a surface [Biasotti et al. 2007a; Siddiqi and Pizer 2007].

A formal definition of curve-skeletons, which is valid for surfaces without boundary, was proposed in [Dey and Sun 2006] on the basis of the distance function from a volume boundary. Here the curve-skeleton is defined as the set of points of the medial axis that are singular with respect to a medial geodesic function. More

formally, given a space $O \subset \mathbb{R}^3$ bounded by a connected manifold surface M , let MA , $MA \subset O$, be the medial axis of O . Two subsets of MA are defined, namely MA_2 and MA_3 . MA_2 is the set of points whose maximal inscribed balls touch the surface M at exactly two distinct points. MA_3 is constituted by curves lying at the intersection of the closure of three sheets in MA_2 . The maximum balls lying on points of MA_3 touch the surface M at three points. A *Medial Geodesic Function (MGF)* is defined on MA_2 and MA_3 . At a point $x \in MA_2$, MGF is the length of the geodesic path between two points at the intersection of the surface M with the maximum ball centered in x . The definition of MGF is extended to each point of MA_3 on the basis of the values that MGF assumes in the three half-disks of MA_2 around the point. The medial geodesic function is used to define the curve-skeletons of the two subsets MA_2 and MA_3 . The curve skeleton Sk_2 of MA_2 is the set of singular points of MGF on MA_2 . The curve skeleton Sk_3 of MA_3 is the set of points where the gradient flow of MGF sink into from the three local neighbors. Finally, the curve-skeleton of the space O is defined as the closure of $Sk_2 \cup Sk_3$. Since the exact computation of the curve-skeleton is extremely hard, it is approximated by adopting a rough evaluation of the function MGF on the centers of the medial axis facets and a polygonal approximation of the medial axis.

The curve-skeleton and, in general, centerlines related to the concept of medial axis provide a shape description which is unique and independent of other functions defined on the shape. Therefore, these descriptions yield a geometric centerline of the shape rather than a topological one. On the other hand, the spatial embedding of the Reeb graph of a surface is often regarded as a topological centerline skeleton. In this case, each contour may be visualized through its centroid.

In addition, various methods adopt a centerline encoding that can be related to Reeb graphs, even if they do not explicitly originate from the Reeb graph definition. These approaches generally do not require the mapping function to be general nor simple and, differently from the centerlines based on the distance function, act as topological centerlines. The construction of the *Level Set Diagrams (LSD)* from triangulated polyhedra proposed in [Lazarus and Verroust 1999] uses Euclidean distances for wave propagation from a seed point. An heuristic detects a point at the top of a protrusion on the basis of the geodesic distance. This source point automatically determines a privileged “slicing direction”. In this approach, a skeleton-like structure, available for input objects of genus zero, is proposed, which is essentially a tree made of the *average* points (in the sense of centroids) associated with the connected components of the level sets of the geodesic distance from the source. The resulting skeleton is invariant under rotation, translation and uniform scaling (see Figure 20). The method in [Axen 1999] follows the same approach replacing Euclidean distance with topological distance.

An extension of the approaches in [Axen 1999] and [Lazarus and Verroust 1999] to non-zero genus surfaces was presented in [Hetroy and Attali 2003]. In this case, the evaluation of the measuring function, the mesh characterization (based on local criteria) and the construction of the graph are performed at the same time using Dijkstra’s algorithm. A similar approach was introduced in [Wood et al. 2000] for implicit storage of meshes obtained from distance volumes. In this case the graph extraction is driven by the evolution of the isosurfaces of the geodesic distance

from a single point. In fact, the object topology is reconstructed by considering a wavefront-like propagation from a seed point by applying Dijkstra's algorithm. The cost complexity of these methods, all of which require the vertices to be sorted, is $O(n \log n)$, where n is the number of vertices of the mesh.

Finally, the application of the approach in [Lazarus and Verroust 1999] to point clouds was proposed in [Verroust and Lazarus 2000]. The algorithm runs in $O(e + n \log n)$, where e is the number of edges in the neighborhood graph and n is the number of points.

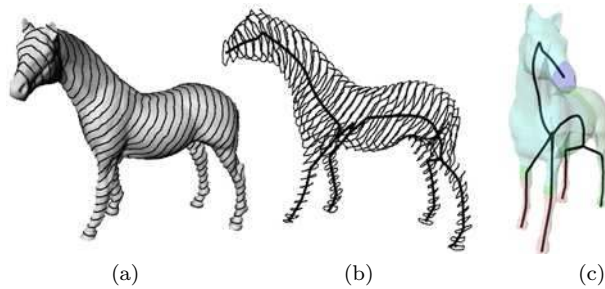


Fig. 20. Level sets (a) and the centerline (b,c) of an horse using the geodesic distance from a source point as proposed in [Lazarus and Verroust 1999].

The method in [Mortara and Patané 2002] uses the multi-resolution curvature evaluation proposed in [Mortara et al. 2004] to locate seed points that are subsequently used during the geodesic expansion. Seed points, also called *representative vertices*, are sequentially linked by using a wavefront traversal defined on the simplicial complex (see Figure 21(a,b)). Once a set of representative vertices is selected, rings made of vertices of increasing neighborhoods are computed in parallel until the whole surface is covered (see Figure 21(c)), in a way similar to the wave-traversal technique [Axen and Edelsbrunner 1998]. Rings growing from different seed points will collide and join where two distinct protrusions depart, thus identifying a branching zone; self-intersecting rings can appear when expanding near handles and through holes. A skeleton is drawn according to the ring expansion: *terminal nodes* are identified by the *representative vertices*, while union or split of topological rings give *branching nodes*. It is worth noticing that the terminal nodes of the graph have degree 1. Arcs are drawn joining the center of mass of all rings (see Figure 21(d)). Therefore, the complexity of the proposed graph, in terms of number of nodes and branches, depends on the shape of the input object and on the number of seed points which have been selected using the curvature estimation criterion. The number of operations needed for extracting the skeleton is $O(n)$ in the number of mesh vertices (each vertex is visited once) even if an accurate evaluation of the high curvature points may require $O(n^2)$ operations. In addition, this curve-line representation has at least as many cycles as the number of holes of the surface; however, some unforeseen cycles may appear as a result of colliding wavefronts.

Time-varying data are not always best dealt with by means of a function defined over four equivalent dimensions. In practice, most four-dimensional data consist

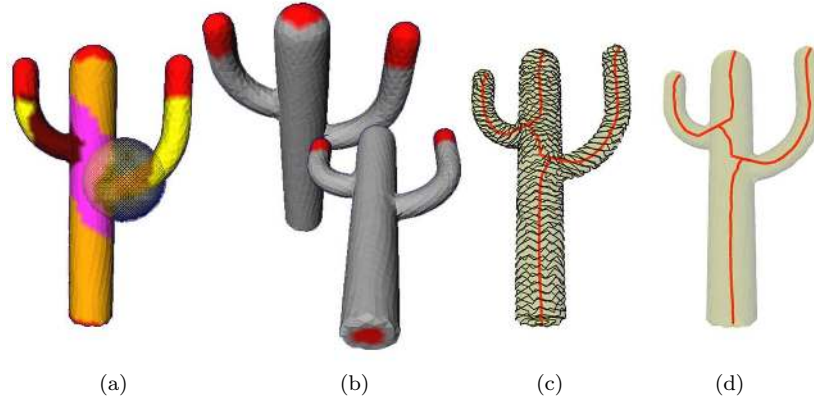


Fig. 21. (a) Vertex classification based on Gaussian curvature; (b) high curvature regions are depicted in red; (c) topological rings expanded from centers of high curvature regions; (d) the graph obtained as proposed in [Mortara and Patané 2002].

of time-slices of three-dimensional data that can be treated with a refinement of 3D algorithms. An algorithm for studying the evolution of the Reeb graph when the mapping function varies with time is proposed in [Edelsbrunner et al. 2004]. To simplify the topological complexity of the space, a point at infinity is added. In this way the space becomes topologically equivalent to the 3-sphere and each Reeb graph will be equivalent to a tree. In this framework, the time is represented by a conventional priority queue that stores birth-death and interchange events, that are prioritized by the moments in time they occur. Once a Reeb graph is computed, the evolution of the graph over time is detected using a Jacobi curve that collects the birth-death points and maintains the occurrence of a new event. A data structure is used to store the entire evolution. The computational cost of this algorithm, $O(N + En)$, depends on the number N of simplices of the triangulation of the space-time data, the upper number n of simplices at a time t and the amount E of birth-death and interchange events.

Table V summarizes the algorithms for Reeb graph computation surveyed here.

6.3 Applications

First introduced in Computer Graphics by Shinagawa et al. [Shinagawa et al. 1991], Reeb graphs were initially used only for Morse mapping functions and their extraction required *a priori* knowledge of the object genus [Shinagawa and Kunii 1991].

Application fields related to the use of Reeb graphs include surface analysis and understanding [Shinagawa et al. 1991; Attene et al. 2003]; identification of topological quadrangulations [Hetroy and Attali 2003]; data simplification [Biasotti et al. 2002]; animation [Kanongchaiyos and Shinagawa 2000; Xiao et al. 2003]; surface parameterization [Steiner and Fischer 2001; Patané et al. 2004; Zhang et al. 2005]

¹Once a set of seed points has been recognized, the complexity of the skeleton extraction is linear in the number of mesh vertices but an accurate evaluation of the high curvature points requires $O(n^2)$ operations.

Reeb graph					
Approach	Method	Domain			Costs
		2D	3D	nD	
[Shinagawa and Kunii 1991] [Shinagawa et al. 1991]	Contours	X	X		$O(n^2)$
[Hilaga et al. 2001] [Bespalov et al. 2003] [Tung and Schmitt 2004]	Contours	X			$O((n + m))$
[Attene et al. 2001; 2003] [Biasotti 2004b; 2005]	Contours	X			$O(\max(m + n, n \log n))$
[Cole-McLaughlin et al. 2003]	Analytic	X			$O(n \log n)$
[Pascucci et al. 2007]	Analytic	X	X	X	$O(n \log n)$
[Axen 1999] [Lazarus and Verroust 1999] [Hetroy and Attali 2003]	Contours	X			$O(n \log n)$
[Mortara and Patané 2002] ¹	Contours	X			$O(n)$
[Wood et al. 2000]			X		$O(n \log n)$
[Wood et al. 2004]	Contours		X		$O(n \log n)$
[Shattuck and Leahy 2001]	Contours		X		$O(v \log v)$
[Verroust and Lazarus 2000]	Contours	Point clouds			$O(e + n \log n)$
[Werghi et al. 2006]	Contours	Point clouds			$O(n)$

Table V. Classification of the methods for Reeb graph extraction. Symbols: n represent the number of vertices or points; m the number of vertices inserted in the mesh during a possible contouring phase; e the number of edges in the neighborhood tree; v the number of voxels.

and remeshing [Wood et al. 2000]; shape [Berretti et al. 2006] and human body [Werghi et al. 2006] segmentation; approximation and modification of the input geometry [Shinagawa and Kunii 1991]; object reconstruction [Biasotti et al. 2000b] and editing where the stored information is exploited for shape recovering. Moreover, the knowledge of the shape topology given by the graph structure improves the surface reconstruction from contour lines [Biasotti et al. 2000b], thus solving the correspondence and the branching problems. Details on this topic may be found in [Fuchs et al. 1977; Meyers et al. 1992; Oliva et al. 1996].

The compactness of the one-dimensional structure, the natural link between the function and the shape, and the possibility of adopting different functions for describing different aspect of shapes have led to a massive use of Reeb graphs for similarity evaluation, shape matching and retrieval [Hilaga et al. 2001; Biasotti et al. 2003]. In [Hilaga et al. 2001] the Reeb graph is used in a multi-resolution fashion for shape matching. The ratio of the area and the length of the model sub-part in the whole model are associated with each node, i.e. shape slice, and are used as attributes during the graph-matching phase. The set of the geometric attributes is further enriched in [Tung and Schmitt 2005], where, for each slice, the authors consider the volume, a statistic measure of the extent and the orientation of the triangles, an histogram of the Koenderink shape index, which provides a representation of the local shape curvatures [Koenderink 1990], and a statistic of the texture. In particular, [Tung and Schmitt 2004] shows how the performance for shape retrieval improves when these geometric attributes are added to nodes of the multi-resolution graph structure. Finally, the method in [Hilaga et al. 2001] has

been successfully applied also to database retrieval of CAD models as proposed in [Bespalov et al. 2003].

The application of the extended Reeb graph to database retrieval has been addressed in [Biasotti et al. 2003] and discussed in the context of CAD models in [Biasotti and Marini 2005]. The decomposition into significant regions induced by the ERG defines a structural description of the shape, which is coupled with an error-correcting subgraph isomorphism to build up a shape retrieval system. In particular, the proposed graph matching framework makes it possible to plug in heuristics for tuning the algorithm to the specific application and for achieving different approximations to the optimal solution. In [Biasotti et al. 2006] the authors discuss a method for measuring the similarity and recognizing sub-part correspondences of 3D shapes, based on the coupling of a structural descriptor, like the ERG, with a geometric descriptor, like spherical harmonics. In the same paper it is discussed how a structure-based matching can improve the retrieval performance in terms of both functionalities supported (i.e., partial and sub-part correspondences) and the variety of shape descriptions that can be used to tune the retrieval with respect to the context of application. Recently, the ERG has been adopted to compare sets of 3D objects and results have been shown in two databases of 80 3D scenes, respectively made of 2 and 3 objects [Paraboschi et al. 2007]. In this case, the ERGs of the objects in a scene are grouped in a scene graph obtained by connecting each ERG to a virtual node and the distance between two scene graphs is defined as a measure on the graph spectrum.

In the field of regularly sampled 3D grids of scalar values (that is a volume model in which each grid cube has 8 neighbor grid points), the method proposed in [Wood et al. 2004] topologically simplifies and repairs the topological noise that affects large scattered datasets. Similarly, the method in [Shattuck and Leahy 2001] analyzes, and eventually corrects, the topology of cortical volumes because it is assumed that cortical volumes are homeomorphic to a sphere and no cycles are expected in the graph representation.

Similarly, the method proposed in [Pascucci et al. 2007] has been used to find and highlight small defects, such as small manifold handles and tunnels, in models of arbitrary dimension. Since these features correspond to loops of the Reeb graph, the model is simplified by removing parts that correspond to irrelevant loops, where the relevance of a loop is defined as its persistence (cf. Section 8), in the sense of a percentage of the function image.

The hyper Reeb graph defined in [Fujishiro et al. 1999; Fujishiro et al. 2000] has been proposed to enhance traditional volume visualization techniques with a double-layered topological structure. In [Fujishiro et al. 2000], the method is tested on a large-scale, time-varying volume data set. There, the hyper Reeb graph is used to simulate an ion-atom collision problem between a proton and a hydrogen atom and to investigate the variation of the electron density distribution during the collision. In particular, the identification of the Reeb graphs that correspond to the simplest structure of the isosurfaces permits to approximate the collision time of the atomic structures.

7. SIZE THEORY

Size theory has been developed since the beginning of the 1990s in order to provide a geometrical-topological approach to the comparison of shapes (cf. [Frosini 1990; 1991; Verri et al. 1993]). Introductory presentations can be found in [Frosini and Landi 1999], [Kaczynski et al. 2004].

The basic notion behind size theory is the abstraction of the similarity between shapes in terms of the *natural pseudo-distance* between the topological spaces that represent the shapes. Intuitively, two shapes are similar when they exhibit similar properties: this fact can be conceptualized by considering a shape as a pair defined by a topological space *and* a function that measures some properties, which are relevant in a specific context. Then, the similarity between shapes can be expressed by a small variation in the measure of these properties when we move from one shape to the other. In this setting, shapes are similar if there exists a bi-continuous transformation, or, more precisely, a homeomorphism, that preserves the properties conveyed by the functions.

The idea comes from the observation that many groups of transformations, such as isometries or affinities, can be expressed in terms of the preservation of the values of some real functions on topological spaces. For instance, the concept of isometry can be traced back to the preservation of the distance function, and the concept of affinity to the preservation of the affine area. Therefore, it seems natural to consider shapes as spaces equipped with real functions and study the changes of such functions, that is, study how the function values are modified or preserved under the action of homeomorphisms between the spaces. The evaluation of such changes yields a measure to compare shapes.

The formalization of this approach lead to the definition of the *natural pseudo-distance*, intuitively defined as the infimum of the variation of the values of the functions when we move from one space to the other through homeomorphisms.

Notice that the most common transformation groups, such as those mentioned above, can be expressed using the language of the natural pseudo-distance [Frosini 1991]. The natural pseudo-distance actually defines a concept of similarity between shapes. Indeed, in this theoretical setting, two objects have the same shape if they share the same shape properties, expressed by the functions' values, i.e. their natural pseudo-distance vanishes.

In order to effectively estimate the natural pseudo-distance and compare shapes, *size functions* were introduced. They are shape descriptors that analyze the variation of the number of connected components of lower level sets with respect to the real function, which describes the shape properties we are interested in. This theoretical approach is quite general and flexible, and has been recently extended to multi-variate functions.

7.1 Theoretical aspects

The main idea in size theory is to compare shape properties that are described by real functions defined on topological spaces associated with the “objects” to be studied. This leads to considering *size pairs* (S, f) , where S is a topological space and $f : S \rightarrow \mathbb{R}$ is a continuous *measuring function*.

When two objects X and Y must be compared, the first step is to find the “right”

set of corresponding properties, i.e. of size pairs $(S(X), f_X), (S(Y), f_Y)$. Depending on the problem, one can decide to work directly on X , so that $S(X) = X$, or to compute a derived space $S(X)$ which is different from X . For example, $S(X)$ can be chosen (whenever applicable) to be the Cartesian product $X^{(n)} = X \times X \times \cdots \times X$, or the tangent space of X , or a projection of X onto a line, or the boundary of X , or the skeleton of X , and so on. The measuring functions are meant to give a quantitative description of $S(X), S(Y)$. Their choice is driven by the set of properties that one wishes to capture.

The next step in the comparison process is to consider the *natural pseudo-distance* d . The main idea in the definition of natural pseudo-distance between size pairs is the minimization of the change in measuring functions due to the application of homeomorphisms between topological spaces. Formally, d is defined by setting

$$d((S(X), f_X), (S(Y), f_Y)) = \inf_{h \in H_{X,Y}} \sup_{P \in S(X)} |f_X(P) - f_Y(h(P))|,$$

where h varies in a subset $H_{X,Y}$ of the set H of all homeomorphisms between $S(X)$ and $S(Y)$. The subset $H_{X,Y}$ must satisfy the following axioms: the identity map $id_X \in H_{X,X}$; if $h \in H_{X,Y}$ then the inverse $h^{-1} \in H_{Y,X}$; if $h_1 \in H_{X,Y}$ and $h_2 \in H_{Y,Z}$ then the composition $h_2 \circ h_1 \in H_{X,Z}$ [Frosini 1991]. Often, $H_{X,Y}$ coincides with H (cf. [Donatini and Frosini 2004b; 2007]). If $S(X)$ and $S(Y)$ are not homeomorphic the pseudo-distance is set equal to ∞ . It should be noted that the existence of a homeomorphism is not required for X and Y but for the associated spaces $S(X)$ and $S(Y)$. In this way, two objects are considered as having the same shape if and only if they share the same shape properties, i.e. the natural pseudo-distance between the associated size pairs vanishes.

Since the set of homeomorphisms between two topological spaces is rarely tractable, simpler mathematical tools are required to estimate the natural pseudo-distance. To this end, the main mathematical tool introduced in size theory is given by size functions, which provide a lower bound for the natural pseudo-distance.

Size functions are shape descriptors that analyze the variations of the number of connected components of the lower level sets of the studied space with respect to the chosen measuring function. Given a size pair (S, f) , the (reduced) *size function*

$$\ell_{(S,f)} : \{(x, y) \in \mathbb{R}^2 : x < y\} \rightarrow \mathbb{N}$$

can easily be defined when S is a compact and locally connected Hausdorff space: $\ell_{(S,f)}(x, y)$ is equal to the number of connected components of the lower level set $S_y = \{P \in S : f(P) \leq y\}$, containing at least one point of the lower level set S_x (see [d'Amico et al. 2006]).

An example of size function is illustrated in Figure 22. In this example we consider the size pair (S, f) , where S is the curve represented by a continuous line in Figure 22(a), and f is the function “distance from the point P ”. The size function associated with (S, f) is shown in Figure 22(b). Here, the domain of the size function is divided by solid lines, representing the discontinuity points of the size function. These discontinuity points divide the set $\{(x, y) \in \mathbb{R}^2 : x < y\}$ into regions on which the size function is constant. The value displayed in each region is the value taken by the size function in that region. For instance, for $a \leq x < b$, the set $\{P \in S : f(P) \leq x\}$ has two connected components which are contained in

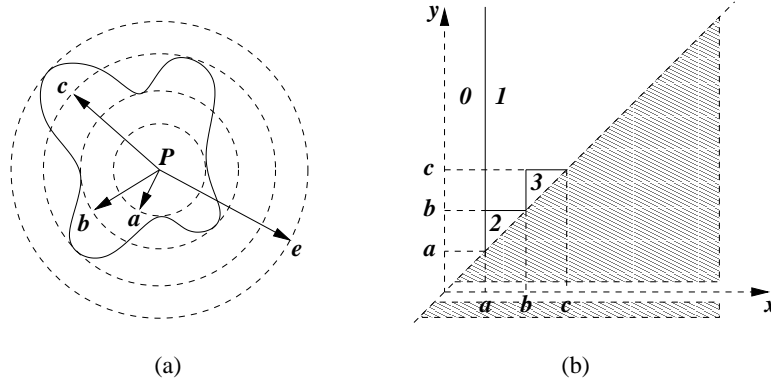


Fig. 22. (a) The size pair (S, f) , where S is the curve represented by a continuous line and f is the function “distance from the point P ”. (b) The size function of (S, f) .

different connected components of $\{P \in S : f(P) \leq y\}$ when $x < y < b$. Therefore, $\ell_{(S,f)}(x, y) = 2$ for $a \leq x < b$ and $x < y < b$. When $a \leq x < b$ and $y \geq b$, all the connected components of $\{P \in S : f(P) \leq x\}$ are contained in the same connected component of $\{P \in S : f(P) \leq y\}$. Therefore, $\ell_{(S,f)}(x, y) = 1$ for $a \leq x < b$ and $y \geq b$. When $b \leq x < c$ and $y \geq c$, all of the three connected components of $\{P \in S : f(P) \leq x\}$ belong to the same connected component of $\{P \in S : f(P) \leq y\}$, implying that in this case $\ell_{(S,f)}(x, y) = 1$.

In [Frosini and Landi 1997] a new kind of representation of size functions was introduced, based on the fact that they can always be seen as linear combinations of characteristic functions of triangles (possibly unbounded triangles with vertices at infinity), with a side lying on the diagonal $\{x = y\}$ and the other sides parallel to the coordinate axes. For example, the size function of Figure 23 (left) is the sum of the characteristic functions of the triangles with right angles at vertices a , b , c plus the characteristic function of the infinite triangle on the right of line r . This suggests that the size function is completely determined by a, b, c, r . In fact, the property that size functions can be represented as collections of vertices (called *cornerpoints*) and lines (called *cornerlines*) always holds [Frosini and Landi 2001]. This provides a simple and concise representation for size functions in terms of points and lines in \mathbb{R}^2 , drastically reducing the required descriptive dimensionality.

As suggested in Figure 23, this representation also allows for the comparison of size functions using distances between sets of points and lines [Donatini et al. 1999], e.g. the Hausdorff metric or the matching distance. In particular, the matching distance between two size functions l_1 and l_2 , respectively represented by the sequences (a_i) and (b_i) of cornerpoints and cornerlines, can be defined as

$$d_{match}(l_1, l_2) := \min_{\sigma} \max_i d(a_i, b_{\sigma(i)})$$

where i varies in the set of natural numbers \mathbb{N} , σ varies among all the bijections from \mathbb{N} to \mathbb{N} , and the pseudo-distance d between two points p and p' is equal to the smaller between the cost of moving p to p' and the cost of moving p and p' onto the diagonal $\{x = y\}$, with costs induced by the max-norm [d’Amico et al. 2006]. An example is shown in Figure 23 (right), where an optimal matching between

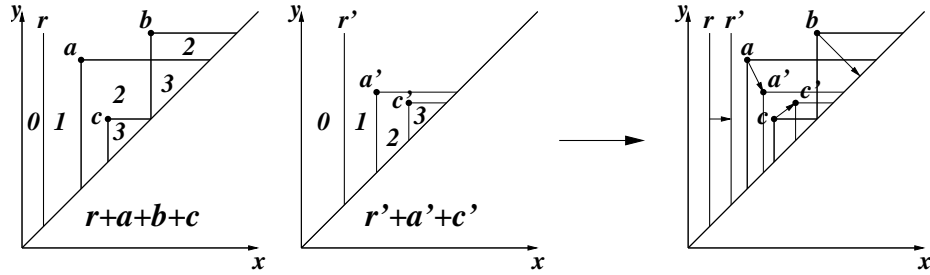


Fig. 23. Two size functions can be described by cornerpoints and cornerlines and compared by the matching distance.

cornerpoints and cornerlines of two size functions is shown. Here the matching distance is given by the cost of moving the cornerpoint denoted by b onto the diagonal.

The stability of this representation has been studied in [d'Amico et al. 2003; 2005]. In particular, it has been proven that the matching distance between size functions is continuous with respect to the measuring functions, guaranteeing a property of perturbation robustness. Moreover, it can be shown that the matching distance between size functions produces a sharp lower bound for the natural pseudo-distance between size pairs [Donatini and Frosini 2004a; d'Amico et al. 2005], thus guaranteeing a link between the comparison of size functions and the comparison of shapes [d'Amico et al. 2006].

As pointed out in [Edelsbrunner and Harer 2007], an important problem is given by the possibility of working in a k -dimensional setting, that is, using measuring functions with values in \mathbb{R}^k . Multi-dimensional measuring functions and, consequently, multi-dimensional size theory and natural pseudo-distance were introduced in [Frosini and Mulazzani 1999]. However, a direct approach to the multi-dimensional case implies working in subsets of $\mathbb{R}^k \times \mathbb{R}^k$, and it is unclear how to combinatorially represent multi-dimensional size functions.

A solution is proposed in [Cerri et al. 2007; Biasotti et al. 2007b], where it is proven that the computation and comparison of multi-dimensional size functions can be reduced to the one-dimensional case by a suitable change of variables. The idea is that the domain of k -dimensional size functions can be suitably partitioned into half-planes, such that the restriction of a k -dimensional size function to each half-plane is a classical one-dimensional size function. This implies that, on each half-plane of the domain partition, the size functions can be represented by cornerpoints and cornerlines. A multi-dimensional matching distance can also be defined, based on the one-dimensional matching distance on each half-plane, which is stable for small changes in the measuring functions.

An example of the computation of multi-dimensional size functions is shown in Figure 24. Two surface models are analyzed using a two-dimensional measuring function $\vec{f} = (f_1, f_2)$, and the domain of the two-dimensional size function is partitioned into half-planes. Details on the definition of \vec{f} and the partition can be found in [Biasotti et al. 2007b]. In this figure, we show the behavior of the corresponding

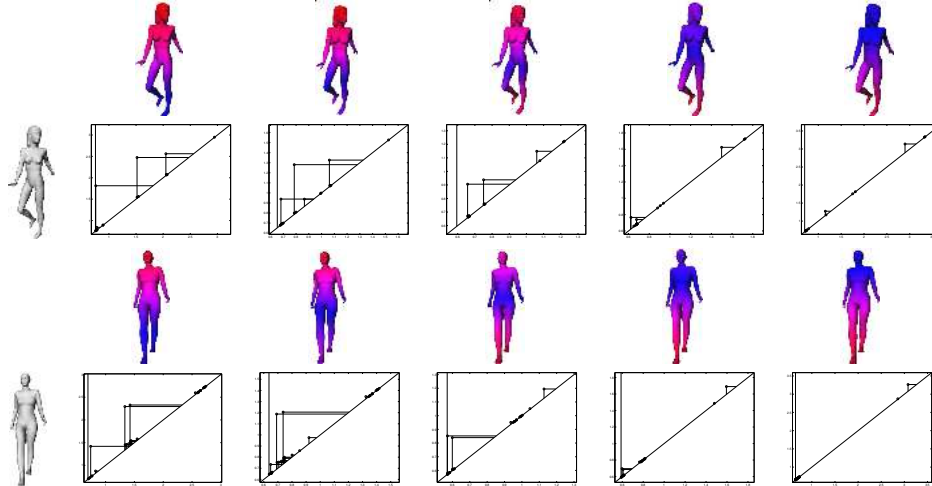


Fig. 24. Example of the computation of a two-dimensional size function on two surface models, on five half-planes of the domain partition defined in [Biasotti et al 2007b].

two-dimensional size function on five half-planes of the partition, depicted from left to right. On each half-plane, the two-dimensional size function coincides with a classical one-dimensional size function. The underlying one-dimensional measuring functions, derived from the components of \vec{f} , are depicted on top of the corresponding size functions (red corresponds to high values of the measuring function, blue corresponds to low values). Then, the two-dimensional matching distance can be computed, based on the classical one-dimensional matching distances between the size functions on each of the half-planes.

Size functions are not the sole tool introduced in size theory. Indeed, algebraic topology has been used to obtain generalizations of size functions that give a more complete description of a pair (S, f) , since they take into account not only the number of connected components but also the presence of other features such as holes, tunnels and voids. The first development in this sense can be found in [Frosini and Mulazzani 1999] where *size homotopy groups* are introduced, inspired by the classical mathematical notion of homotopy group. They are shown to provide a lower bound for the natural pseudo-distance, much in the same way as size functions do.

The study of size functions in the algebraic topology setting was also developed in [Cagliari et al. 2001a] by observing that $\ell_{(S,f)}(x, y)$ can be seen as the rank of the image of $H_0(j_{xy}) : H_0(S_x) \rightarrow H_0(S_y)$ where j_{xy} is the inclusion of S_x into S_y . This observation has led to the definition of the *size functor*, which studies the maps $H_k(j_{xy}) : H_k(S_x) \rightarrow H_k(S_y)$ for every k . In other words, it studies the process of the birth and death of homology classes as the lower level set changes. When M is smooth and compact and f is a Morse function, the functor can be described by oriented trees, called *H_k -trees* [Cagliari et al. 2001b].

7.2 Computational aspects

From the computational point of view, the main efforts have been devoted to the development of techniques for the computation of size functions [Frosini 1992], while no algorithms are available to compute the size homotopy groups or the size functor.

From the application side, using size functions for shape analysis requires two steps: (i) the choice of the size pair (S, f) and (ii) the computation of $\ell_{(S,f)}$.

In the following we discuss each of these steps.

7.2.1 Choosing the size pair. One needs to choose both the space $S = S(X)$, associated with the object X under study, and the measuring function f . Since each pair (S, f) conveys information about X from a different viewpoint, the choice is driven by the set of shape properties that the user wants to capture.

In the applications, X can be a binary, a gray-scale or a full-color image, a continuous curve, a closed surface or a simplicial complex (in particular a graph). In [Uras and Verri 1997] the concept of auxiliary manifold $S(X)$ is introduced, i.e. an object of fixed and known topological structure that can be linked to the shape X under study. The definition of the measuring functions on the auxiliary manifold permits the solution of problems related to small topological changes in the original shape due to noise and perturbations. The idea of an auxiliary space as a domain for the measuring functions has also been proposed in the case of binary images in [Cerri et al. 2006] and of surface meshes in [Biasotti et al. 2006b].

In [Verri and Uras 1996] the authors discuss the need for heuristic criteria to select adequate measuring functions, and propose using parameterized families of measuring functions. The key problem of the invariance requirements in Pattern Recognition tasks is explored in [Landi and Frosini 2002], [Verri and Uras 1994] and [Dibos et al. 2004], where the ability of size functions to deal with Euclidean, affine and projective invariance is discussed.

7.2.2 Computing $\ell_{(S,f)}$. In recent years three algorithms have been proposed to compute the size functions, once the size pair (S, f) has been chosen.

The first two algorithms [Frosini 1992; 1996] involve the definition of two labeled graphs designed to discretize the size pair under study. The third algorithm [d'Amico 2000] starts with a given labeled graph and directly computes the corner-points and cornerlines, which completely describe the size function.

The algorithm in [Frosini 1992] (see also [Frosini 1991]) requires S to be a compact and arcwise connected subset of \mathbb{R}^m , and the measuring function f to be the restriction to S of a continuous function $\bar{f} : \mathbb{R}^m \rightarrow \mathbb{R}$. The first step of this algorithm consists of the discretization of the space S . For this purpose, the concept of δ -covering is introduced, i.e. a collection of open balls $\{B(P_i, \delta)\}_i$ of radius δ and centered at P_i , whose union contains S , and such that the intersection between each ball and S is a non-empty arcwise connected set. We can define the *size graph approximating* (S, f) as the labeled graph $(G, \bar{f}|_G)$, where the set of vertices of G is equal to $\{P_i\}_i$, and two vertices, P_i and P_j , are adjacent if the intersection between S and the union of the balls $B(P_i, \delta)$ and $B(P_j, \delta)$ is an arcwise connected set (see Figure 25). Once the approximating size graph has been obtained, the size function of $(G, \bar{f}|_G)$ is computed. Theoretical results are available [Frosini 1992], relating the size function defined in the continuous case to the approximating one computed

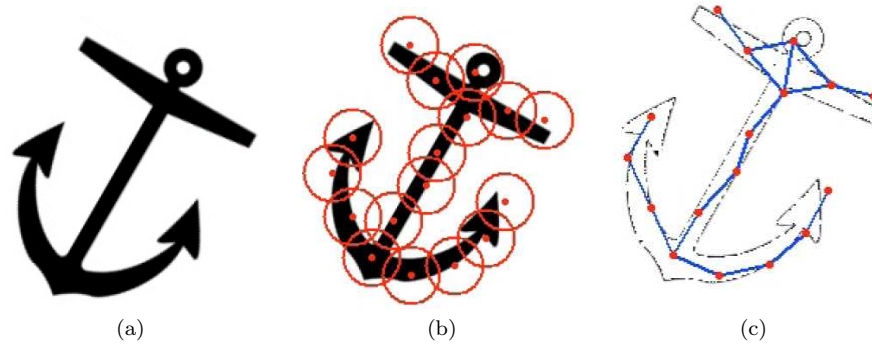


Fig. 25. The discretization of a space described in [Frosini 1992]. (a) The original space. (b) The process of δ -covering. (c) The approximating graph.

on the graph. These results allow us to restrict the computation of $\ell_{(G, \bar{f}_G)}(x, y)$ to a finite subset of values (x, y) , and permit an explicit calculation of the accuracy of the approximation, depending on the function f .

The algorithm proposed in [Frosini 1996] assumes S to be a compact and smooth manifold without boundary, and the measuring function f to be Morse. This method is based on the construction of a *Morse graph*, whose vertices correspond to the critical points of f , and where an arc exists between two vertices if and only if they are connected by at least a gradient flow line. It has been shown that the size function of the Morse graph is identical to the size function of the original space.

Both the preceding algorithms reduce the computation of the size function of the chosen size pair (S, f) to that of a labeled graph. The approximating size graph and the Morse graph are examples of *size graphs*. In general, a *size graph* is a size pair (G, f) , where $G = (V(G), E(G))$ is a finite graph, with $V(G)$ and $E(G)$ the set of vertices and edges respectively, and $f : V(G) \rightarrow \mathbb{R}$ is any function labeling the nodes of the graph. Denoting by G_y the subgraph of G obtained by erasing all vertices of G at which f takes a value strictly greater than y , and all edges that connect those vertices to other vertices, the size function $\ell_{(G, f)}(x, y)$ of (G, f) is equal to the number of connected components of G_y , containing at least one vertex of G_x .

Since the number of vertices and edges of size graphs can be very large, procedures to reduce them are needed. In [Frosini and Pittore 1999] the instruments of *L-reduction* and Δ -*reduction* have been introduced. Their main feature is that they permit a simplification of the graph without changing the corresponding size function.

The *L-reduction* is a global reduction method, since it requires knowledge of the entire size graph. Basically, the vertices of the *L-reduced* graph are of two types: vertices corresponding to local minimum values of the measuring function, and vertices corresponding to saddles of lowest elevation between two minima. The edges connect the minima with their corresponding saddles.

The Δ -*reduction* is a local method, in the sense that only the knowledge of the local structure of the size graph is needed. Moreover, Δ -reduction has been

extended in [d’Amico 2000] to Δ^* -reduction, a procedure defined by recursively applying three different editing moves. It has been proven that a *total* Δ^* -reduction exists, i.e. that the editing process cannot proceed infinitely, and that the totally Δ^* -reduced graph has the simple structure of a tree (or a forest, i.e. a disjoint union of trees, when G has several connected components). A possible implementation for this algorithm is based on the union-find structure [Tarjan and von Leeuwen 1984], so that its computational cost is $O(n + m \cdot \alpha(2m + n, n))$, where m and n are the number of vertices and edges in the graph, respectively, and α is the inverse of Ackermann’s function. Note that m is $O(n^2)$ in the worst case.

The value of the Δ^* -reduction relies not only on reducing the numbers of vertices and edges in the graph, but also on permitting faster computation of size functions [d’Amico 2000]. In fact, the Δ^* -reduction permits direct computation of the cornerpoints and cornerlines, that completely describe the size functions (see Section 7.1). To achieve this, one has to orient the reduced graph obtained through the process of Δ^* -reduction by orienting each edge from the vertex with higher value to the other one. The resulting configuration is an arborescence, i.e. an oriented tree in which no two edges are directed to the same vertex. Finally, the cornerpoints and cornerlines are computed from this arborescence by applying a recursive procedure. The cost of computing the size function of the reduced graph is $O(n' \log n')$, with n' the number of vertices in the reduced graph; usually n' is considerably smaller than n .

Notice that also multi-dimensional size functions can be computed using the algorithm in [d’Amico 2000], as a consequence of the results in [Cerri et al. 2007; Biasotti et al. 2007b] that reduce the computation of multi-dimensional size functions to that of one-dimensional ones through a suitable change of variables (see Section 7.1).

7.3 Applications

The most explored field of application concerning size theory is the field of Pattern Recognition, where size functions have been used as a tool for shape comparison, retrieval and classification, especially in the case of natural or articulated objects. The relationship between the comparison of shapes and the comparison of size functions is studied in [Donatini and Frosini 2004a].

In [Verri and Uras 1996] and [Uras and Verri 1994] the authors proposed a recognition scheme for the signs in the International Alphabet Sign Language (ISL). A one-parameter family of 72 measuring functions is introduced, together with 72 corresponding feature vectors, to describe the signs belonging to the ISL, represented by the curves defining their outline. The problem of recognizing the sign language by means of size functions has also been addressed in [Handouyaya et al. 1999], where a pair of moment-based size functions is used as an input to a neural network classifier.

Size functions are also useful in the biomedical field. In [Ferri et al. 1994] a system for the automatic classification of white blood cells, represented by gray-level images, is presented. In this particular case, the choice of a set of adequate and effective measuring functions is driven by the need to take into account the specific morphological features of the leukocyte classes. Similar principles guided the choice of the features in the ADAM (Automatic Data Analysis for Melanoma early

detection) project. In [d'Amico et al. 2004], [Stanganelli et al. 2005] size functions and Support Vector Machines are combined to implement an automatic classifier of melanocytic lesions; the system is mainly based on a qualitative assessment of asymmetry, as a parameter to distinguish between nevi and melanomas.

Recently a strategy to address the problem of figurative images was proposed. In [Cerri et al. 2006] a complete system for automatic trademark retrieval based on size functions was implemented, in order to deal with the actual problem of preserving product identity and avoiding copyright infringement. Experiments were performed on a database to retrieve binary trademark images provided by the UK Patent Office.

The problem of image retrieval on the Internet is dealt with in [Cerri et al. 2005]. Ferri and Frosini [Ferri and Frosini 2005] suggest equipping each image on a Web site with a simplified drawing called *keypic* (in alternative to *keyword*). Cerri et al. [Cerri et al. 2005] carry out an experimentation on a set of keypics, thus proposing size functions as a possible ingredient in the solution to the problem of image searching and retrieval on the Internet.

The comparison and retrieval of 3D objects has been dealt with using both classical (i.e. one-dimensional) and multi-dimensional size functions. In [Biasotti et al. 2006b] object surfaces are described through size graphs approximating a centerline skeleton, that is enriched with geometrical attributes. The computation of the size functions of these size graphs allows for the comparison and retrieval of the objects. In [Cerri et al. 2007; Biasotti et al. 2007b] some sets of bi-variate and tri-variate measuring functions are introduced to describe triangle meshes and voxelized digital models, respectively. Two-dimensional and three-dimensional size functions are then computed on a small set of models, and experimental results are carried out to test the feasibility of the approach.

8. PERSISTENT HOMOLOGY

We now introduce persistent homology, which provides a tool to study topological features of spaces endowed with possibly varying real functions, completely based on algebraic topology.

Algebraic topology is useful for detecting the number and type of topological features, such as holes, in a space. Among the various tools offered by algebraic topology to this aim, homology groups have the advantage of being computable. However, standard homology is not able to decide to what extent a topological attribute of a space is relevant for the shape description.

As we have seen in Section 3.1.5, Morse Theory explores the topological attributes of an object in an evolutionary context. In [Edelsbrunner et al. 2000] and [Edelsbrunner et al. 2002] this evolutionary approach has been revisited. The authors introduce a technique, called *persistence*, which grows a space incrementally and analyzes the topological changes that occur during this growth. In particular, they produce a tool, called *persistent homology*, for controlling the placement of topological events (such as the merging of connected components or the filling of holes) within the history of this growth. The aim is to furnish a scale to assess the relevance of topological attributes. The main assumption of persistence is that longevity is equivalent to significance. In other words, a significant topological at-

tribute must have a long life-time in a growing complex. In this way, one is able to distinguish the essential features from the fine details.

Much of the material about persistent homology has been incorporated in [Zomorodian 2005]. Recently, persistent homology was surveyed in [Edelsbrunner and Harer 2007].

8.1 Theoretical aspects

The first concept related to persistent homology theory is that of a filtered complex, that is, a complex equipped with a filtration. A *filtration* of a complex is a nested sequence of subcomplexes that ends with the complex itself. Formally, a complex K is filtered by a filtration $\{K^i\}_{i=0,\dots,n}$ if $K^n = K$ and K^i is a subcomplex of K^{i+1} for each $i = 0, \dots, n-1$. An example of filtered complex is given in Figure 26 (Top). Since the sequence of subcomplexes K^i is nested, one can think of K as a complex that grows from an initial state K_0 to a final state $K^n = K$. Therefore it is often referred to as a growing complex.

Filtered complexes arise naturally in many situations. The simplest example of filtration is the *age filtration* [Edelsbrunner et al. 2002]: The complex K is filtered by giving an ordering $\Delta_0, \Delta_1, \dots, \Delta_m$ to its simplices and by defining the sequence of its subcomplexes K^i as $K^i = \{\Delta_j \in K : 0 \leq j \leq i\}$. In other words the complex grows from $K^0 = \{\Delta_0\}$ adding each simplex one by one according to the given order. It is assumed that if Δ_i is a face of Δ_j then Δ_i enters the filtration before Δ_j .

A filtered complex also arises when some space (e.g. a curve or a surface) is known only through a finite sample X of its points. Since the knowledge of the original space is necessarily imprecise, a multi-scale approach may be suited to describe the topology of the underlying space (see also the approach in [Niyogi et al. 2006] to compute the homology of submanifolds from random samples). The idea is to construct, for a real number $\epsilon > 0$, an abstract simplicial complex $R_\epsilon(X)$, called the *Rips complex*, whose abstract k -simplices are exactly the subsets $\{x_0, x_1, \dots, x_k\}$ of X such that $d(x_i, x_j) \leq \epsilon$ for all pairs x_i, x_j with $0 \leq i, j \leq k$. Whenever $\epsilon < \epsilon'$, there is an inclusion $R_\epsilon(X) \rightarrow R_{\epsilon'}(X)$ that reveals a growing complex (cf. [Collins et al. 2004a], [Collins et al. 2004b]).

Another example of filtration is provided by a complex filtered by the increasing values of a real piece-wise linear function defined on it. In other words, suppose we are given a complex K and a piece-wise linear real function f on K (that is, a function defined by its values on the vertices of K). Denoting by A_0, A_1, \dots, A_n the vertices of K , it is possible to filter K by the subcomplexes K^i of K , consisting of the vertices where the function f takes values not greater than $f(A_i)$, together with all the simplices (edges, triangles, etc.) connecting them (cf. [Edelsbrunner et al. 2001], [Edelsbrunner et al. 2003a]).

In general, given a filtered complex, its topological attributes change through the filtration, since new components appear or connect to the old ones, tunnels are created and closed off, voids are enclosed and filled in, etc.

In particular, as for 0-homology, each homology class corresponds to a connected component, and a homology class is born when a point is added, forming a new connected component, thus being a 0-cycle. A homology class dies when two points belonging to different connected components, thus belonging to two different 0-

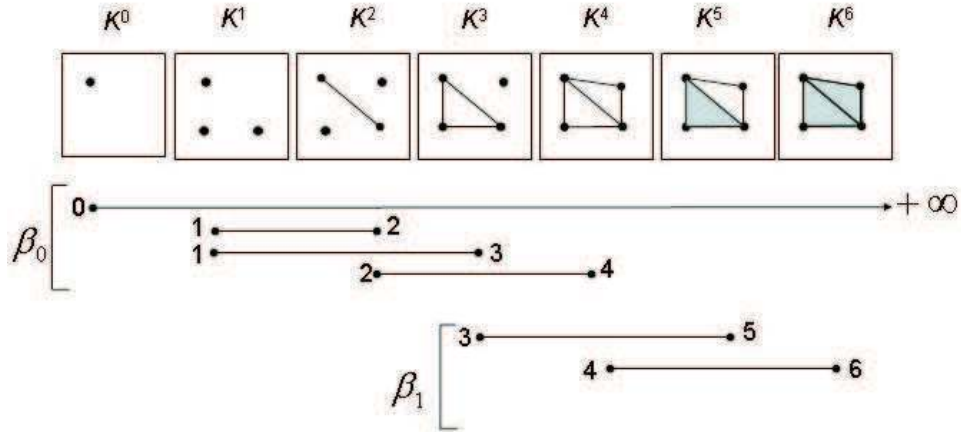


Fig. 26. The persistent homology of a filtered complex can be represented by \mathcal{P} -intervals.

cycles, are connected by a 1-chain, thus becoming a boundary. As an example, consider the filtered complex of Figure 26 (top): One 0-homology class is born at K^0 , two other homology classes are born at K^1 ; at K^2 , a new homology class is born while one of the classes born at K^1 dies, since it is merged to the class born at K^0 ; at K^3 another class dies, and the same happens at K^4 , where we are left with just one class that survives forever.

As for 1-homology, a homology class is born when a 1-chain is added, forming a 1-cycle (for instance, a 1-simplex is added, completing a circle), while it dies when a 2-chain is added so that the 1-cycle becomes a boundary (for instance, a 2-simplex fills a circle). In the example of Figure 26 (top), a homology class is born at K^3 , another one at K^4 , then at K^5 the homology class born at K^3 dies and at K^6 also the homology class born at K^4 dies, so that no 1-cycle survives any longer.

The argument goes on similarly for higher degree homology. Persistent homology algebraically captures this process of the birth and death of homology classes.

Given a filtered simplicial complex $\{K^i\}_{i=0,\dots,n}$, the j -persistent k -th homology group of K^i can be defined as a group isomorphic to the image of the homomorphism $\eta_k^{i,j} : H_k(K^i) \rightarrow H_k(K^{i+j})$ induced by the inclusion of K^i into K^{i+j} . In other words, the j -persistent homology group of K^i counts how many homology classes of K^i still survive in K^{i+j} . Persistence represents the life-time of cycles in the growing filtration (much in the same way as in the definition of the size functor described in Section 7.1). Note that, in more recent papers, the notation for persistent homology groups has changed: the variables i, j have been replaced by $i, i + j$.

The persistent homology of a filtered complex can be represented by a set of intervals, called persistence intervals (briefly \mathcal{P} -intervals), as in Figure 26 (bottom). More precisely, a \mathcal{P} -interval is a pair (i, j) , with $i, j \in \mathbb{Z} \cup \{+\infty\}$ and $0 \leq i < j$, such that there exists a cycle that is completed at level i of the filtration and becomes a boundary at level j .

More recently [Cohen-Steiner et al. 2005], \mathcal{P} -intervals have been described as sets of points in the extended plane. These sets of points are called *persistence diagrams*. In the case of the 0-degree homology of complexes filtered by the lower

level sets of a function f they substantially coincide with the representation of size functions by cornerpoints and cornerlines (cf. Section 7.1). The difference lies in the assumptions about the space and function allowing for their definition, that in size theory are more general. In the same way that the representation of size functions through cornerpoints and cornerlines satisfies the property of stability under perturbations of the data, so this remains true for persistence diagrams, as proven in [Cohen-Steiner et al. 2005; Cohen-Steiner et al. 2007b] with respect to Hausdorff and bottleneck distances.

Considering persistence for complexes filtered by the values of a Morse function f , the process of capturing births and deaths of homology classes naturally establishes a pairing for critical points of f . For example, when f is defined on a curve, passing a local minimum creates a component, while passing a local maximum merges two components represented by two local minima: the maximum is paired with the higher of the two local minima. Clearly, some critical points will not be paired in this process. They correspond to essential features of the shape under study, not depending on the function f . In [Cohen-Steiner et al. 2007a], the persistence pairing is extended to include the homology classes that cannot be paired by ordinary persistent homology since their lifetime is infinite. Another way to pair essential homology classes has been proposed in [Dey and Wenger 2007], where *interval persistence* is introduced focusing on the stability of critical points rather than critical values. In the case of a function defined on a 2-manifold, the pairing of essential classes can be dealt with also using the Reeb graph [Agarwal et al. 2004] (see also Section 8.3).

A recent advance in persistent homology theory is the use of a family of real functions continuously varying in time [Cohen-Steiner et al. 2006]. The structures obtained, referred to as *vines* and *vineyards*, seem to be suitable tools for studying continuous processes. Roughly speaking, a vineyard is a bunch of possibly intersecting curves (the vines) generated by the continuous evolution of persistence diagrams over time.

Generalizing persistent homology to the case of a multi-variate situation in which two or more functions are used to characterize the shape leads to the definition of *multi-filtration* [Zomorodian and Carlsson 2007]. However, it is not possible to extract from this structure a complete and concise representation generalizing the concept of persistence intervals. An alternative solution to this problem is given in [Cagliari et al. 2007].

We refer the reader to [Grandis 2003], [Robins 1999], [Robins 2002] for different developments of the multi-scale approach to describe the topology of a space, which we do not describe here in more detail, since the filtration of the complex does not vary according to the choice of a real function defined on the space.

8.2 Computational aspects

Applying persistent homology requires the user to model the shape under study with a filtered complex. In each application, the choice of the most suitable filtration is left to the user, much in the same way as the choice of the size pair in size theory.

Once a filtered complex has been obtained, two different algorithms are available to compute persistent homology. Both take as input a filtered complex. However, they greatly differ both in the techniques used and in the scope of applications.

The first algorithm reported ([Edelsbrunner et al. 2000]) required the complex to be embedded in \mathbb{R}^3 and filtered by the age filter. Moreover, it worked only with homology coefficients in the field \mathbb{Z}_2 of integers modulo 2. An advantage of this algorithm is that its rationale is rather intuitive. The second algorithm, presented in [Zomorodian and Carlsson 2004], is much more general since it allows us to compute the persistent homology of any filtered complex with coefficients over an arbitrary field in any dimension. The drawback is that the algebraic machinery needed to obtain this algorithm is rather sophisticated, although the final procedure is based on Gaussian elimination. Both the algorithms take at most $O(m^3)$ in time, where m is the number of simplices in the filtration.

We now describe the first algorithm ([Edelsbrunner et al. 2000; 2002]). The idea is to pair the creation of a cycle with its conversion to a boundary. Such a pairing algorithm takes as input a complex $K = \{\Delta_0, \Delta_1, \dots, \Delta_m\}$ embedded in \mathbb{R}^3 and filtered with the age filtration, and returns a list of simplex pairs (Δ_i, Δ_j) , where Δ_i is a k -simplex and Δ_j is a $(k+1)$ -simplex ($0 \leq k \leq 2$). Each pair represents a k -cycle created by Δ_i and turned into a k -boundary by Δ_j . The algorithm initially needs to decide whether the addition of a k -simplex creates a k -cycle (the cycle question). This question can be answered using the incremental algorithm presented in [Delfinado and Edelsbrunner 1995]. Here the assumption that K is embedded in \mathbb{R}^3 plays an important role. Indeed, this implies that the only interesting case is $k = 1$, because any 0-simplex belongs to a cycle, no 3-simplex belongs to a cycle and the case $k = 2$ can be reduced to the case $k = 1$ using a dual graph (that is, the graph whose vertices correspond to the 3-simplices of K and where there is an edge connecting two vertices if and only if the corresponding 3-simplices share a common face). The algorithm is called incremental because it consists of adding one simplex of K at a time. A 1-simplex creates a cycle if and only if its two endpoints belong to the same component. This can be decided by implementing a union-find structure [Tarjan and von Leeuwen 1984]. Once the cycle question is decided for each simplex, the idea is to pair simplices as follows: if a $(k+1)$ -simplex Δ_j does not belong to a $(k+1)$ -cycle, consider its boundary $d = \partial_{k+1}(\Delta_j)$. Since d is a k -cycle, we can consider all the k -simplices in d that have been previously marked as creating a k -cycle. The youngest one, that is, the one with the largest index is the simplex Δ_i that must be paired with Δ_j .

The second algorithm that computes persistent homology ([Zomorodian and Carlsson 2004], [Zomorodian and Carlsson 2005]) works on any filtered d -dimensional simplicial complex $\{K^i\}_{i=0, \dots, n}$ and over any field \mathbb{F} . Taking the homology coefficients in a field, the homology groups are actually vector spaces. The key observation is that the computation of persistence requires compatible bases for $H_k(K^i)$ and $H_k(K^{i+p})$. The idea underlying the algorithm is to represent the lifetime of a simplex by utilizing multiplication by a formal parameter u : if a simplex Δ enters the filtration at time i , at time $i+1$ it becomes $u \cdot \Delta$, at time $i+2$ it becomes $u^2 \cdot \Delta$, and so on. We underline that the letter u is only a formal parameter, while the actual information about the lifetime is given by the power of u . With this representation, the boundary operator has coefficients in the ring $\mathbb{F}[u]$ of polynomials over \mathbb{F} in the letter u . It follows that the boundary operator can be represented by a matrix with polynomial entries. A standard reduction algorithm on this matrix,

more precisely Gaussian elimination by elementary operations on the columns, reduces the matrix to column-echelon form. By non-trivial algebraic arguments, the \mathcal{P} -intervals for the $(k - 1)$ -th persistent homology can be directly read from the pivots of the boundary operator ∂_k , reduced into column-echelon form.

Finally, the computation of *vineyards* is carried out in [Cohen-Steiner et al. 2006] through an algorithm that maintains an ordering of the simplices in the filtration as time varies. This can be achieved in time $O(n)$ per transposition of two simplices in the filtration, where n is the number of simplices used to represent the topological space.

8.3 Applications

One of the first applications of persistent homology reported in the literature [Edelsbrunner et al. 2001] addresses the problem of topological simplification viewed as a process that decreases Betti numbers. We have seen that a by-product of the computation of persistent homology is a set of pairs of simplices (Δ_i, Δ_j) , where Δ_i is the simplex that, on entering the filtration, creates a cycle, and Δ_j is the simplex that, entering the filtration, adds the cycle to the group of boundaries. In other words, attaching Δ_i corresponds to the birth of a new homology class, while attaching Δ_j leads to its death. Thus, topological simplification can be achieved by removing simplices in pairs, in the order of increasing importance: cycles whose persistence is below some threshold can be removed, since they correspond to noise or non-relevant features; only cycles with a longer life-time are considered important. However, the meaning of the simplification changes according to the context. Topological simplification of a complex filtered by a Morse function corresponds to geometric smoothing of the Morse function. The application of this topological simplification to complexes generated by sample points provides a method for noise reduction in sample data. An example of this simplification process can be found in [Edelsbrunner et al. 2002], carried out on a molecular structure represented as a complex endowed with an age filtration. A further application is the simplification of Morse-Smale complexes for 2-manifolds proposed in [Edelsbrunner et al. 2001], where the sequence of cancellation of pairs of critical points is driven by the persistence of the pairs. This approach is also followed in [Bremer et al. 2004], where an algorithm is proposed which allows the simultaneous application of independent cancellations (see also the formalization in [Danovaro et al. 2006]). Applications are shown for terrain models. The extension of the idea of the persistence-driven simplification of Morse-Smale complexes to functions defined on 3-manifolds is performed in [Gyulassy et al. 2005].

The problem of simplifying the function itself, rather than the underlying space, has recently been addressed in [Edelsbrunner et al. 2006]. The notion of ϵ -*simplification* of a function f is introduced, that is, a function g such that $\|f - g\|_\infty \leq \epsilon$ and whose persistence diagrams are the same as those of f , except that all points within L_1 -distance less than ϵ from the diagonal of \mathbb{R}^2 are removed. It is also shown, through a constructive proof, that ϵ -simplifications exist for 2-manifolds. Beside the problem statement, the main novelty in [Edelsbrunner et al. 2006] is that, in previous works on simplification, all points of the persistence diagrams could be moved towards the diagonal, regardless of their distance from it, while this approach ensures that points with persistence higher than ϵ are not moved. The order of removal of

pairs of critical points no longer follows the order of increasing persistence, but the increasing value of the function on the second point of the pair.

A study of shape description and classification via the application of persistent homology is carried out in [Collins et al. 2004a; 2004b] for curve point cloud data, and in [Carlsson et al. 2004; 2005], where examples are shown for geometric surfaces and surfaces of revolution. The general idea is to describe the shape of a complex K , filtered by the increasing values of a real function f , defined on K : the topological changes occurring through the filtration, that, according to Morse theory, are due to the presence of critical points of f , are captured by persistent homology. In particular, in these works the shape of X is studied by constructing a new complex strictly related to X : the *tangent complex* of X , that is the closure of the space of all tangents to all points in X . The tangent complex contains a large amount of information about the geometry of X . So far, the authors have confined themselves to considering only one function, that is, the curvature at a given point along a specified tangent direction. The choice of this function is directed at capturing those features of a shape that are connected with curvature. From this setup the authors derive the notion of the *barcode* of a shape, that is, the set of \mathcal{P} -intervals for this filtered tangent complex. A pseudo-metric between barcodes allows for a measure of the similarity between shapes: being a pseudo-metric and not a metric, the distance between two different shapes can be vanishing. It is interesting to note that, analogous to what happens in size theory (cf. Section 7.1), in this research the shape of X is studied by constructing a new complex, strictly related to X , which in this case is the filtered tangent complex. Barcodes as descriptors for shape classification have been tested on a small database (80 items) of hand-drawn copies of letters.

The recently introduced concept of vineyards is applied in [Cohen-Steiner et al. 2006] to the study of protein folding trajectories. The analysis of the behaviour of proteins also motivates the work in [Agarwal et al. 2004]. The notion of *elevation* is introduced for points on 2-manifolds smoothly embedded in \mathbb{R}^3 , in order to identify cavities and protrusions in the protein structure, since they play an important role in protein interaction. The definition of elevation is derived from an extension of the classical notion of persistence pairing, which takes into account the pairings between all critical points of the function defined on a genus g embedded 2-manifold. In particular, the $2g$ saddles starting the $2g$ cycles, which remain unpaired once the manifold sweep is complete, are paired making use of the Reeb graph of the manifold. Pairing all critical points allows the elevation of each point to be determined by its persistence, that is, the absolute difference in function values to its paired point. Results of the application of extended persistence pairing to protein docking are presented in [Wang et al. 2005].

Addressing the problem of localizing topological attributes, in [Zomorodian and Carlsson 2007] *localized homology* is proposed as a method to find the most local basis of the homology of a given space. The main tool is the Mayer-Vietoris blowup complex associated with an open covering of the simplicial complex under study. The blowup complex and the original complex have the same homology but the former can be filtered according to the number of open sets that cover each simplex. The persistent homology of this filtered complex gives insights into the relationship

between the local and global homology of the space.

Finally, we point out that the idea of considering the topology of a space at various resolutions by means of persistent homology is inspiring research in new directions, such as the coverage problem in sensor networks [de Silva and Ghrist 2007] and the analysis of the structure of natural images [de Silva and Carlsson 2004].

9. MORSE SHAPE DESCRIPTOR

The next method we survey is an algebraic descriptor for objects modeled as smooth manifolds endowed with a Morse function that measures some metric properties of the given objects. The key ingredient of this descriptor is, once again, Morse theory, allowing for a link between topology and critical points in terms of homology groups. The main difference with respect to persistent homology is the use of relative homology groups instead of ordinary homology.

This topological descriptor, like size functions and persistent homology, provides information that can remain constant, despite the variability in appearance of objects due to noise, deformation and other distortions. At the same time, it allows for a significant reduction in the amount of data, while providing sufficient information to characterize and represent objects.

9.1 Theoretical aspects

Like the previous approaches, the Morse shape descriptor describes the shape of a manifold M by analyzing the changes in topology due to the presence of critical points of a Morse function f , defined on M .

This approach is based on the use of relative homology groups. Roughly speaking, the relative homology groups of a pair of spaces (X, A) , with $A \subseteq X$, count the number of cycles in X , while ignoring all the chains of X contained in A . For the precise definition of relative homology groups we refer the reader to [Spanier 1966].

In [Allili et al. 2004], a descriptor is introduced, defined as the function $R_f : \mathbb{R}^2 \times \mathbb{N} \rightarrow \mathbb{N}$ defined by setting $R_f(x, y, k)$ equal to the Betti number of the relative homology group $H_k(M_y, M_x)$, when $x \leq y$ and y belongs to the range of f , and equal to 0, otherwise. Here M is assumed to be a connected compact manifold without boundary and $f : M \rightarrow \mathbb{R}$ a Morse function. The connectedness assumption guarantees that f ranges over an interval and its compactness ensures that this interval is finite and closed. Moreover, the absence of boundary guarantees that the homology generators are directly related to the critical points of the function. In fact, if the manifold has non-empty boundary, some of the homology generators can be related to the boundary components. The idea under this definition is that the relative homology groups give information about a change in the topology of the lower level set when going through a critical value. Indeed, if a single critical point of index λ is present between two levels, x and y , then the relative homology group $H_k(M_y, M_x)$ has rank equal to 1 if $k = \lambda$, and 0 otherwise.

We point out that the intimate connection between the relative homology groups of the lower level sets and critical levels has been known for a long time in mathematics. For example, it was used in [Kelley and Pitcher 1947] to define critical levels for any bounded real function f , defined on any topological space X , without restricting f any further.

The idea, sketched in [Allili et al. 2004], is more thoroughly developed in [Allili and Corriveau 2007], where the authors distinguish between a *Morse Descriptor (MD)* that corresponds to the function R_f above, and a *Morse Shape Descriptor (MSD)*, that is a Morse Descriptor defined only for Morse functions invariant under rigid motions and scale transformations.

The Morse Descriptor allows for multi-scale analysis of shape, since, as shown in [Allili and Corriveau 2007], the larger the number of the lower level sets studied, the more topological information can be obtained: $MD_f(x, y, k) \leq MD_f(x, z, k) + MD_f(z, y, k)$, for every $x \leq y \leq z$ in \mathbb{R} and every $k \in \mathbb{N}$.

9.2 Computational aspects

In its discrete form, the Morse shape descriptor is encoded in a collection of matrix structures, one matrix for each homology degree. For a manifold of dimension n , there are exactly $n + 1$ significant homology degrees k , $0 \leq k \leq n$. The element in the i -th row and j -th column of the k -matrix is precisely the number $R_f(x_i, y_j, k)$. Therefore, the comparison of shapes reduces to a distance measure between matrices. The integer indexes i, j vary in the set of the first N natural numbers. The value N represents the number of samples in the discretization of the range of f normalized between the absolute minimum and maximum value.

The algorithm presented in [Allili and Corriveau 2007] involves the computation of the relative homology of $N(N - 1)/2$ pairs of complexes.

In view of applications to images, the algorithm used in [Allili and Corriveau 2007] to compute the homology groups is based on cubical complexes, which are complexes whose basic building-blocks are intervals, squares, cubes and their generalizations to higher dimensions [Allili et al. 2001], [Allili and Ziou 2003], [Kaczynski et al. 2004], [Ziou and Allili 2002].

9.3 Applications

The performance of the Morse Shape Descriptor has been evaluated in [Allili and Corriveau 2007] on a 2D image retrieval problem. The experimental dataset contains 1100 2D images clustered in 10 classes. In order to model the shapes as connected compact manifolds without boundary, only the shape contours are considered. For each contour, four Morse Shape Descriptors are produced, associated with four different measuring functions, invariant with respect to rigid motions and scale transformations. The measure of similarity between two shapes is given by a weighted sum of distances between the collection of Morse Shape Descriptors associated with the contours. The effectiveness of the system is evaluated by measuring its precision and recall.

10. DISCUSSION

In this survey we have focused on methods for shape analysis in which the object to be studied is described by a pair (S, f) , where S is a space, often coinciding with the shape itself, and f is a real function defined on S . The shared aim of the methods surveyed is to identify points of interest on a shape, and to capture the shape's connectivity in expressive structures. The approach common to all the methods described finds its roots in classical Morse theory, which combines the topological exploration of S with quantitative measurement of geometrical properties provided

by f . This means that both global and local analysis concur to obtain the final shape description.

Due to the possibility of adopting different functions for describing shapes, the methods surveyed provide a general framework for shape characterization. Changing the properties that one wishes to analyze simply means changing the function, without any other modification in the mathematical model. Moreover, the modularity of the approach can be extended to the choice of the space S used to represent the shape under study.

Although all the methods surveyed are inspired by Morse theory, there are some substantial differences in the shape description and interpretation that they offer. In this section we propose a global comparison among the techniques described, which highlights the differences in terms of properties of the descriptors (Section 10.1), effectiveness of the description and loss of information with respect to the representation (Section 10.2), usefulness and context of applicability (Section 10.3).

10.1 Overall comparison and general remarks

The distinguishing feature of Morse and Morse-Smale complexes is that they express geometric information related to the *gradient flow* of the measuring function f , while the other descriptors encode features captured by f itself. Contour trees and Reeb graphs compactly represent topological information related to the *level sets* of f , expressing the way they are connected. Size theory, persistent homology theory and the Morse shape descriptor explore in a homological setting the growth of a space, according to the placement of topological events in the evolution of the *lower level sets*.

We can also observe how the methods surveyed adopt mathematical structures of different level of abstraction to convey geometrical-topological information. On one hand, Morse and Morse-Smale complexes, contour trees and Reeb graphs are essentially combinatoric structures. On the other hand, methods in size theory, persistent homology theory and the Morse shape descriptor mainly rely on the use of algebraic structures.

With regard to the combinatoric descriptors, the different kind of information they encode is reflected by the differences in the combinatoric structures they produce. While contour trees and Reeb graphs always code the shape in terms of one-dimensional structures (i.e. trees or more general graphs, respectively) disregarding the dimension of the underlying manifold, the decomposition provided by Morse or Morse-Smale complexes is expressed in terms of a cell complex, where the dimension of the complex coincides with the dimension of the manifold. Moreover, Morse and Morse-Smale complexes explicitly induce a shape segmentation. A shape segmentation could also be derived naturally from a Reeb graph, by observing that the counterimages of the simplices in the graph actually define a decomposition of the shapes into critical level sets and ribbons.

The connectivity of the different complexes is also worth consideration. Ascending and descending Morse complexes are dual to each other. For instance, in a 2-manifold the 0-cells in one complex correspond to the 2-cells in the other complex and vice versa, and there is a one-to-one correspondence between the edges in the two complexes. Thus, by using just one representation for a cell complex, such as the winged edge [Baumgart 1975], the half-edge [Mäntylä 1988] or the

quad-edge representation [Guibas and Stolfi 1985], it is possible to encode the combinatorial structure of both the ascending and descending complexes for a given Morse function f into a single representation. In this representation, the minima will be associated with the vertices and the maxima with the 2-cells, while the saddle points will be attached to the edges. Two geometric descriptions will be further associated with each edge e , namely the geometry of e in the descending and ascending complex, respectively.

In the Morse-Smale complex each vertex corresponds to a critical point and has valence less or equal to four. In particular, in the $2D$ case, a saddle is always connected with at most two minima and at most two maxima, and a maximum or a minimum is connected with at most four saddles. The 2-cells have four edges and vertices, the latter corresponding to alternating critical points, namely a minimum, a saddle, a maximum and another saddle. These properties allows a description of the critical net of a Morse-Smale complex, called surface network in the $2D$ case, as a tripartite graph whose nodes are the critical points.

With regard to Reeb graphs, the degree of a vertex, which corresponds to a 0-cell of a Morse-Smale complex, depends on the index of the corresponding critical points. Leaf nodes always represent maxima and minima, and intermediate nodes (i.e., nodes with degree ≥ 2) correspond to saddles of different index (cf. Section 6.1). In the case of 2-manifolds embedded in \mathbb{R}^3 , the degree of vertices representing saddles is always three. Similar properties could be stated for contour trees, which are a special case of Reeb graphs. We recall, however, that contour trees are more application-oriented than Reeb graphs, although they are rooted in the same theory.

There are also important differences in the way the shape is coded in the combinatorial representations defined by contour trees and Reeb graphs. Contour trees are defined for scalar fields $f : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$ where D is a simply-connected subdomain of \mathbb{R}^n . This implies that the connected components of the level sets can be ordered with a nesting criteria, and therefore contour trees do not admit cycles. Reeb graphs are defined for a more general class of shapes, n -dimensional manifolds, and therefore they can have a general connectivity which reflects the topology of the manifold. Moreover, Reeb graphs take into account not only the number but also the changes in the topology of connected components of the level sets, while contour trees do not always (see Section 6.1). To give an example, let us consider a scalar field in \mathbb{R}^3 whose iso-surface changes genus across a critical value of the scalar field. In this case, the classical contour tree would always count one connected component while the Reeb graph would identify one critical point and therefore reflect its presence in the related graph structure. In other words, the contour tree does not necessarily encode all the saddles that might be identified by analyzing the level set evolution. For all these reasons, Reeb graphs can be thought of as a generalization of contour trees, which allows for a more flexible framework for studying shape properties.

Size theory, persistent homology theory and Morse shape descriptors share a different approach to shape analysis, based on the use of algebraic structures. We remark that an increase of abstraction level and richness of topological information about the shape corresponds to the increase of mathematical structure – from size functions to persistent homology groups to the size functor – but at the price of

diminishing the manageability of the descriptors. In particular, persistent homology groups and size homotopy groups rely on the algebraic notion of group. Since these algebraic structures are difficult to handle, the more manageable tools of persistence diagrams and size functions have been introduced.

The unifying key among size functions, persistent Betti numbers (i.e., the rank of the persistent homology groups) and the Morse shape descriptor is the observation that all these tools basically count some homology classes determined by the inclusion of lower level sets of a space S with respect to a function f . Since for $x < y$ the size function $\ell_{(S,f)}(x, y)$ counts the number of connected components of the lower level set S_x which remain disconnected in S_y , we may notice that this is precisely the rank of the corresponding 0th persistent homology group. This means that size functions actually coincide with the 0th persistent Betti numbers. It also follows that, in the case of 0-degree homology, the formal series describing size function substantially coincide with the set of points of persistence diagrams. The connection between size functions and the Morse shape descriptor is stated in [Allili et al. 2004]. An analogous relationship can be stated between the Morse shape descriptor and the rank of the persistent homology group, i.e. the persistent Betti numbers.

It also makes sense to compare persistent homology groups with size homotopy groups since they share the same algebraic structure. In much the same way as the classical result that the first homology group is the abelianization of the fundamental group [Spanier 1966], it can be proved that the first persistent homology group is the abelianization of the first size homotopy group.

Finally, we observe that the size functor furnishes a shape description by a still more general structure, that is that of functor. In particular, persistent homology groups can be seen as the images of the morphisms of the size functor.

10.2 Expressiveness of shape descriptors

The techniques surveyed can be discussed also from the point of view of their potential for describing as well as for discriminating shapes.

According to the adopted tool, we obtain descriptions which store a different amount of information about the pair (S, f) . In general, we can observe that the descriptors retaining a larger amount of information are those which allow better discrimination among shapes. Depending on the application, this can be in turn an advantage or a drawback. The advantage of *forgetting* descriptors is their concise and manageable nature, while their drawback generally is the lack of *completeness*. Completeness here is meant as the property of retaining sufficient data about the shape so that it can be uniquely identified by the descriptor. Conversely, incomplete descriptions might bring up ambiguity in the sense that different shapes may have the same associated representation. These two properties are obviously complementary and the descriptor and abstract representation of the shape has to be devised as a compromise between the two aspects, according to the requirements of the application context.

From a theoretical point of view, the Morse and Morse-Smale complex are complete descriptors among the methods surveyed, since they are naturally associated with a decomposition of the original shape. Notice however that, in applied contexts, it is often better to compute and store the 1-skeleton of the Morse-Smale

complex, i.e. the critical net, which is more manageable and computationally less costly. In contrast, in several applications contour trees and Reeb graphs are not treated as purely combinatoric structures, but instead enriched with geometric information related to the underlying model, such as the arc length [Wood et al. 2004], some contour levels [Shinagawa et al. 1991; Biasotti 2004a], scalar values related to surface [Hilaga et al. 2001; Tung and Schmitt 2005] or volume [Carr et al. 2004; Zhang et al. 2004] segments, or the associated subpart decomposition [Biasotti et al. 2006].

From a general perspective, we can observe that there is a decreasing amount of data retained as we progress from combinatoric to algebraic descriptors. For example, consider the 2-manifolds in Figure 27. The study of their shape, with respect to the height function f in the horizontal direction, can be performed using the persistence intervals or the Reeb graph. However, while the Reeb graph is able to discriminate between the two surfaces, neither the 0th nor the 1st homology persistence intervals are able to discriminate between them.

Notice that the triviality of the 0th homology persistence interval is not intrinsic to the manifold, but depends on the particular choice of function. For instance, the opposite of the chosen height function would generate a non-trivial description.

Since the Reeb graph is a one-dimensional simplicial complex with the ability to capture salient shape features, it makes sense to apply other shape descriptors directly to the Reeb graph representation. Since the Reeb graph codes the variation of the connected components of the level sets, it follows that the size functions, the 0th persistent homology group and the 0th homology Morse shape descriptor can be computed on either the original shape or on its Reeb graph. In contrast, this is not true for the size homotopy groups, the higher degree persistent homology groups and the higher degree Morse shape descriptors, since the reduction of the original shape to a one-dimensional structure causes the loss of higher-dimensional features. In particular, the 1st persistent Betti number of the Reeb graph is a lower bound of the 1st persistent Betti number of the original shape, while the subsequent persistent Betti numbers are always zero if computed on the Reeb graph. Notice also that, for the special case of 2-manifolds embedded in \mathbb{R}^3 , the results on the number of cycles in the Reeb graphs reported by [Cole-McLaughlin et al. 2003] enabled the extension by [Agarwal et al. 2004] of the notion of persistence to form a pairing between *all* the critical points of a function defined on the manifold.

10.3 Suitability for applications

In Table VI we summarize the surveyed methods according to the required properties of the space associated with the shape S , showing what is theoretically allowable for a given method (see the second column of table VI) and what has actually been achieved in practical applications, according to the literature surveyed (see the third column of the same table).

From the point of view of applications, Morse and Morse-Smale complexes have proven to be useful tools in analyzing the morphology of terrains. Moreover they naturally provide a shape segmentation, which is suitable both for cutting a surface into a single flattenable piece and for simplifying the model representation through the extraction of a combinatorial base domain. This is fundamental for several geometry processing tasks, such as parameterization, remeshing, surface texturing

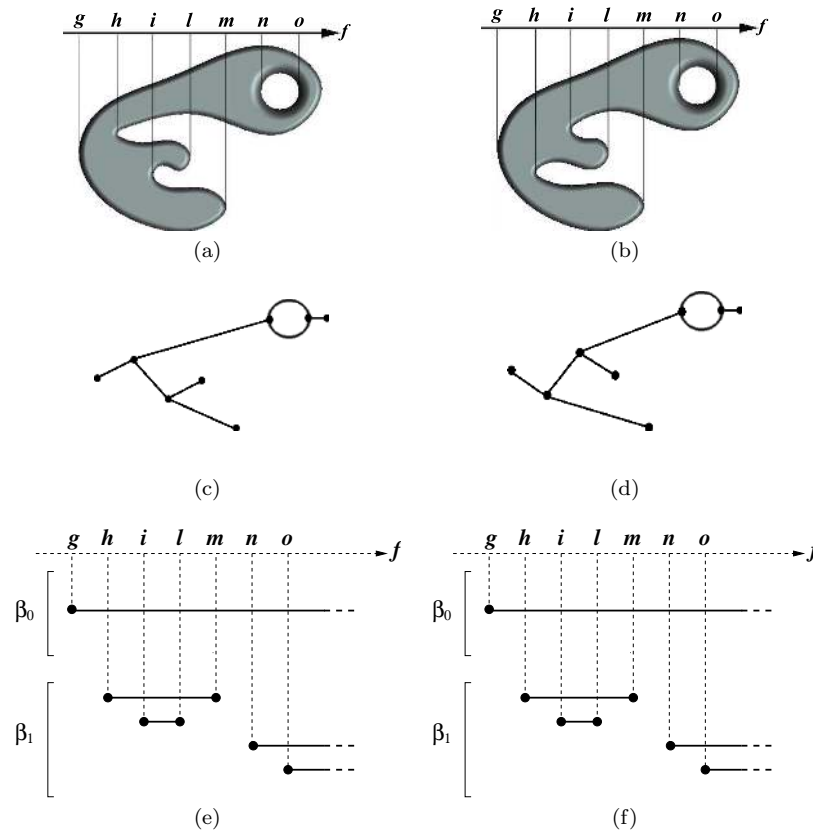


Fig. 27. (a,b) Two surfaces studied with respect to the height function f in the horizontal direction; the values of f are depicted on the arrows. (c,d) Their corresponding Reeb graphs are not isomorphic, and therefore allow to distinguish the original shapes. (e,f) The 0th and the 1st persistent homology groups of the two surfaces coincide, and therefore are not sufficient to discriminate between these objects.

and deformation. In this context, structural problems like over-segmentation are caused by the presence of noise, and efficiency issues arise because of the very large size of existing data sets; these problems have been faced and solved by using generalization techniques and hierarchical representations. Beside the purpose of visual inspection, in recent works the Morse-Smale complex is computed using the eigenfunctions of the discrete Laplacian operator and used to extract surface quadrangulations, that are stable and intrinsic to the model [Dong et al. 2006].

Contour trees are mainly exploited in the visualization context. They have become popular in image processing and topography for their properties that allow a real time navigation of the data. In particular, the recent developments on this topic have highlighted their potential for analyzing high-dimensional and time dependent data, like the visualization of the hemoglobin dynamic and the simulation of galaxy formation in the universe, see for example [Sohn and Bajaj 2006].

Since Reeb graphs generalize to n -dimensional manifolds the concepts behind

	Theoretical side	Computational side
MC	n -manifold	2D and 3D simplicial complexes, 2D grids
MSC	n -manifold	2D and 3D simplicial complexes, 2D grids
CT	n -manifold	simplicial complexes, 2D and 3D grids
RG	n -manifold	simplicial complexes, 3D grids
SFn	Topological spaces	Graphs, 2D simplicial complexes, 2D and 3D grids
SHG	Topological spaces	–
SFr	n -manifolds	–
PH	Simplicial complexes	1-,2-,3D simplicial complexes
MSD	Cubical complexes	2D grids

Table VI. The properties of the input space S for the Morse complexes (MC), the Morse-Smale complex (MSC), the contour tree (CT), the Reeb graph (RG), the size function (SF n), the size homotopy groups (SHG), the size functor (SFr), the persistent homology (PH) and the Morse shape descriptor (MSD).

the contour tree, their application domains partially overlap, for instance in scientific visualization. Despite the more general definition, the existing algorithms for Reeb graph extraction mainly work on 2- and 3-manifolds, and only recently on 3D time-dependent data [Edelsbrunner et al. 2004] and n -dimensional simplicial complexes [Pascucci et al. 2007]. Nevertheless, the definition of Reeb graphs on a domain topologically more “complex” than a scalar field (e.g., with holes, or concavities) emphasizes the compactness of this representation. This fact has stimulated the use of this descriptor in a large number of applications related to surface understanding, simplification, parameterization, segmentation and reconstruction. In addition, the simplicity of the structure (an one-dimensional simplicial complex in every dimension) and the natural link between the properties of the function f and the shape S have lead to a massive use of this descriptor for shape comparison and to development of several shape matching and retrieval tools.

Since the beginning, the declared aim of size theory has been the development of a geometrical-topological framework for comparing shapes. Each shape is viewed as a topological space equipped with a real function describing the shape properties relevant to the comparison problem at hand. Measuring dissimilarity in size theory amounts to minimize the change in the functions due to the application of homeomorphisms between topological spaces, with respect to the L^∞ norm. Size functions are a practical and manageable class of descriptors which allow to provide a lower bound for the dissimilarity measure between shapes. With this theoretical premise, size functions have been extensively used in the field of Pattern Recognition, mainly for image retrieval and classification in the Computer Vision domain, and only recently in Computer Graphics domain [Biasotti et al. 2006a; 2006b; Cerri et al. 2007; Biasotti et al. 2007b].

The idea of persistent homology was originally introduced to assess the relevance of topological attributes in a growing complex. Persistence furnishes a scale to separate out *topological features*, that is attributes with a long life-time in the growing complex, from *topological noise*. The application of this relevance scale to topological simplification is straightforward, leading for example to methods for reducing noise in sample data. At the same time, considering the life-time of topological attributes also induces a powerful description of the shape under study,

that is an invariant reflecting geometrical-topological properties of shapes. This approach to shape comparison has a clear potential which has been demonstrated in a series of examples, although it seems still lacking for an extensive experimentation in comparison with existing techniques. A few years after its introduction, the theory of persistent homology is also becoming one of the basic instruments for solving different application problems, such as protein docking or hole detection in sensor networks.

11. CONCLUSIONS AND FUTURE EMERGING METHODS

The survey has provided an overview of the achievements of computational topology methods for shape analysis using methods rooted in Morse theory. Comparing the emerging challenges in the field identified some years ago in [Bern et al. 1999], we may say that several problems have been successfully solved and many solutions have been developed that also work satisfactorily in practical applications. In particular, the study of the shape of orientable manifolds up to dimension 3 with computational topology methods is now quite mature. However, large multi-dimensional data sets are increasingly available, both static and dynamic, arising for example from scientific simulations. Therefore, one of the main challenges is the definition and the development of mathematical and computational tools to extract knowledge from high dimensional data.

We believe that future trends will mainly focus on two aspects: the concurrent use of more functions to analyse a given shape, and the development of shape description frameworks that improve the effectiveness, in the sense that they better balance the use of geometrical and topological information for characterizing the shape.

Regarding the first aspect, there are already a number of papers that anticipate the trend: the work in [Edelsbrunner and Harer 2002] for example, mentions the importance of developing methods that take into account the evolution and correlation of topological features extracted by more functions applied to the same shape, or alternatively, to study the shape of a scientific data set defined by a multi-valued scalar field. The Jacobi set is introduced in the paper as the mathematical support to the solution of the problem. Given a manifold M and two Morse functions defined on it, the Jacobi set is defined as the set of critical points of the restrictions of one function to the level sets of the other function. The Jacobi set can also be characterized as the set of points where the gradients of the two functions are parallel.

This idea leads to a more general and promising research topic, that considers spaces equipped with multi-valued functions instead of scalar functions. Indeed, a multi-dimensional approach seems more suitable for studying high-dimensional shapes, or alternatively, for studying shapes with a set of functions. The development of a multi-function analysis could bring interesting insights, giving a rich and complete description of the data. While this approach was already suggested in [Frosini and Mulazzani 1999], papers and preprints dealing with this subject have only recently appeared in the framework of size theory [Biasotti et al. 2007b; Cerri et al. 2007] and in persistent homology theory [Carlsson and Zomorodian 2007].

The analysis of dynamical shapes or phenomena also deserves special mention:

from a theoretical point of view, time dependency could be handled as any another variable, but special care has to be taken for the interpretation of dynamical effects. Some of the work reviewed already points out the necessity of defining time-varying descriptions of shapes, with solutions presented for the computation of contour trees and Reeb graphs [Sutton and Hansen 1999; Edelsbrunner et al. 2004; Sohn and Bajaj 2006; Szymczak 2005].

With respect to improvements in effectiveness, we have seen that all methods discussed provide a general framework for studying a shape which is parameterized with respect to the mapping function used, and possibly the space associated to the shape. The mapping function plays the role of the *lens* through which we look at the properties of the shape, and different functions provide different insights. A more interesting problem is how to devise suites of mapping functions that can be formalized beyond a generic best practice or rule of thumb. These functions should be able to capture features that are somehow orthogonal in an abstract conceptual space but also in a geometric sense [Biasotti et al. 2007c]. We think that one of the most promising research direction is the use of the Laplacian eigenfunctions, that are intrinsic to the shape and encapsulate the idea of orthogonality of shape description [Dong et al. 2006; Lévy 2006].

Based on a rich and meaningful set of mapping functions, new descriptions based on Reeb graph could be defined, for example, multi-skeletons or even combined skeletons that capture the most relevant features, augmented with a sufficient geometric data that would allow an efficient processing of the shape data [Marini et al. 2007]. To be noted, indeed, that very interesting results have been obtained using Reeb graphs for shape retrieval applications, due to their capability of reducing the dimensionality of the objects to the linear skeleton representation which supports also partial matching tasks.

Morse and Morse-Smale complexes have proven to be useful tools in analyzing the morphology of terrains and in segmenting $3D$ shapes. In this context, problems due to over-segmentation in the presence of noise, or efficiency issues caused by the very large size of existing data sets, have been faced and solved by using generalization techniques and hierarchical representations. On the other hand, the challenges in scientific visualization are the definition and the development of mathematical tools to extract knowledge from large multi-dimensional data sets arising from scientific simulations. Specifically, volume data sets both static and dynamic (time-varying) are becoming more and more available. Morse and Morse-Smale complexes can be appropriate tools for these purposes having more information content than contour trees, which are sometimes used for the analysis of volume data sets. A completely open issue is how to compute such representations for four-dimensional scalar fields. Generalization and hierarchical representations are even more important in this latter case because of the huge size of available data sets. We need also new techniques for visualizing Morse and Morse-Smale complexes in a way to be effectively helpful for inspecting three and higher-dimensional scientific data sets. Finally, the development of a multi-function analysis also for the Morse and Morse-Smale complex could also bring interesting insights. In these cases, the study of the gradient flows of different measures and their relationship should give a rich and complete description of the dataset.

ACKNOWLEDGMENT

The authors would like to thank Prof. Massimo Ferri and the Vision Mathematics Group at the University of Bologna, Italy, for fruitful discussions on the survey topics.

This work has been developed within the CNR research activities ICT-P03 and DG.RSTL.050.004, and with the support of the EU FP6 Network of Excellence AIM@SHAPE, contract number 506766, by the Italian National Project SHALOM funded by the Italian Ministry of Research under contract number RBIN04HWR8 and by the National Science Foundation under grant number CCF-0541032.

REFERENCES

- ACKERMANN, W. 1928. Zum hilbertschen aufbau der reellen zahlen. *Math. Ann.* 99, 118–133.
- AGARWAL, P. K., EDELSBRUNNER, H., HARER, J., AND WANG, Y. 2004. Extreme elevation on a 2-manifold. In *SCG '04: Proceedings of the 20th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 357–265.
- ALLILI, M. AND CORRIVEAU, D. 2007. Topological analysis of shapes using Morse theory. *Computer Vision and Image Understanding* 105, 3, 188–199.
- ALLILI, M., CORRIVEAU, D., AND ZIOU, D. 2004. Morse homology descriptor for shape characterization. In *ICPR2004: Proceedings of the 17th International Conference on Pattern Recognition*. Vol. 4. 27–30.
- ALLILI, M., MISCHAIKOW, K., AND TANNENBAUM, A. 2001. Cubical homology and the topological classification of 2D and 3D imagery. In *ICIP 2001: IEEE International Conference on Image Processing*. Vol. 2. 173–176.
- ALLILI, M. AND ZIOU, D. 2003. Computational homology approach for topology descriptors and shape representation. In *ICISP'2003: Proceedings of the International Conference on Image and Signal Processing*. Vol. 2. 508–516.
- ATTENE, M., BIASOTTI, S., AND SPAGNUOLO, M. 2001. Re-meshing techniques for topological analysis. In *SMI '01: Proceedings of the International Conference on Shape Modeling and Applications 2001*. IEEE Computer Society Press, Los Alamitos, 142–152.
- ATTENE, M., BIASOTTI, S., AND SPAGNUOLO, M. 2003. Shape understanding by contour-driven retiling. *The Visual Computer* 19, 2-3, 127–138.
- AURENHAMMER, F. 1991. Voronoi diagrams—A survey of a fundamental geometric data structure. *ACM Computing Surveys* 23, 3, 345–405.
- AXEN, U. 1999. Computing Morse functions on triangulated manifolds. In *SODA '99: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM Press, Philadelphia, PA, USA, 850–851.
- AXEN, U. AND EDELSBRUNNER, H. 1998. Auditory Morse analysis of triangulated manifolds. In *Mathematical Visualization*, H. C. Hege and K. Polthier, Eds. Springer-Verlag, 223–236.
- BAJAJ, C. L., GILLETTE, A., AND GOSWAMI, S. 2007. Topology based selection and curation of level sets. In *TopoInVis 2007: Workshop on Topology-based Methods in Visualization 2007*.
- BAJAJ, C. L. AND GOSWAMI, S. 2006. Automatic fold and structural motif elucidation from 3D EM maps of macromolecules. In *ICVGIP 2006: Proceedings of the 5th Indian Conference on Computer Vision, Graphics and Image Processing*. 264–275.
- BAJAJ, C. L., PASCUCCI, V., AND SCHIKORE, D. R. 1997. The contour spectrum. In *VIS '97: Proceedings of the IEEE Visualization 1997*. IEEE Computer Society Press, 167–173.
- BAJAJ, C. L., PASCUCCI, V., AND SCHIKORE, D. R. 1998. Visualization of scalar topology for structural enhancement. In *VIS '98: Proceedings of the IEEE Visualization 1998*. IEEE Computer Society Press, 51–58.
- BAJAJ, C. L. AND SCHIKORE, D. R. 1998. Topology preserving data simplification with error bounds. *Computers and Graphics* 22, 1, 3–12.
- BANCHOFF, T. F. 1967. Critical points and curvature for embedded polyhedra. *Journal of Differential Geometry* 1, 245–256.
- ACM Journal Name, Vol. V, No. N, February 2008.

- BANCHOFF, T. F. 1970. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly* 77, 475–485.
- BANGHAM, J. A., HIDALGO, J. R., AND HARVEY, R. 1998a. Robust morphological scale-space trees. In *Proceedings of Noblesse Workshop on Non-Linear Model Based Image Analysis*, S. Marshall, N. Harvey, and D. Shah, Eds. Glasgow, UK, 133–139.
- BANGHAM, J. A., HIDALGO, J. R., HARVEY, R., AND CAWLEY, G. 1998b. The segmentation of images via scale-space trees. In *Proceedings of the 9th British Machine Vision Conference*. 33–43.
- BAUMGART, B. G. 1975. A polyhedron representation for Computer Vision. In *Proceedings of the AFIPS National Computer Conference*. Vol. 44. 589–596.
- BERGLUND, N. AND SZYMCAK, A. 2004. Making contour trees subdomain-aware. In *Abstracts of the 16th Canadian Conference on Computational Geometry*. 188–191.
- BERN, M. W., EPPSTEIN, D., AGARWAL, P. K., AMENTA, N., CHEW, L. P., DEY, T. K., DOBKIN, D. P., EDELSBRUNNER, H., GRIMM, C., GUIBAS, L. J., HARER, J., HASS, J., HICKS, A., JOHNSON, C. K., LERMAN, G., LETSCHER, D., PLASSMANN, P. E., SEDGWICK, E., SNOEYINK, J., WEEKS, J., YAP, C.-K., AND ZORIN, D. 1999. Emerging challenges in computational topology. *ArXiv Computer Science e-prints cs/9909001*.
- BERRETTI, S., DEL BIMBO, A., AND PALA, P. 2006. Partitioning of 3D meshes using Reeb graphs. In *ICPR '06: Proceedings of the 18th International Conference on Pattern Recognition*. IEEE Computer Society Press, Washington, DC, USA, 19–22.
- BESL, P. AND JAIN, R. 1985. Three-dimensional object recognition. *ACM Computing Surveys* 17, 1, 75–145.
- BESPALOV, D., REGLI, W., AND SHOKOUFANDEH, A. 2003. Reeb graph based shape retrieval for CAD. In *DETC'03: Proceedings of the ASME Design Engineering Technical Conferences 2003*. ASME.
- BEUCHER, S. AND LANTUEJOUL, C. 1979. Use of watersheds in contour detection. In *Proceedings of the Int. Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation*. Rennes, France.
- BIASOTTI, S. 2004a. Computational topology methods for shape modelling applications. Ph.D. thesis, Università degli Studi di Genova.
- BIASOTTI, S. 2004b. Reeb graph representation of surfaces with boundary. In *SMI '04: Proceedings of the International Conference on Shape Modeling and Applications 2004*. IEEE Computer Society Press, Los Alamitos, 371–374.
- BIASOTTI, S. 2005. Topological coding of surfaces with boundary using Reeb graphs. *Computer Graphics and Geometry* 7, 1, 31–45.
- BIASOTTI, S., ATTALI, D., BOISSONNAT, J.-D., EDELSBRUNNER, H., ELBER, G., MORTARA, M., DI BAJA, G. S., SPAGNUOLO, M., AND TANASE, M. 2007a. Skeletal structures. In *Shape Analysis and Structuring*, L. De Floriani and M. Spagnuolo, Eds.
- BIASOTTI, S., CERRI, A., FROSINI, P., GIORGI, D., AND LANDI, C. 2007b. Multidimensional size functions for shape comparison. Tech. Rep. 04-07, IMATI, CNR, Genova (Italy).
- BIASOTTI, S., FALCIDIENO, B., AND SPAGNUOLO, M. 2000a. Extended Reeb Graphs for surface understanding and description. In *DGCI 2000: Proceedings of the 9th Discrete Geometry for Computer Imagery Conference*, G. Borgefors and G. S. di Baja, Eds. Lecture Notes in Computer Science, vol. 1953. Springer Verlag, Uppsala, 185–197.
- BIASOTTI, S., FALCIDIENO, B., AND SPAGNUOLO, M. 2002. Shape abstraction using computational topology techniques. In *From Geometric Modeling to Shape Modeling*, U. Cugini and M. Wozny, Eds. Kluwer Academic Publishers, 209–222.
- BIASOTTI, S., FALCIDIENO, B., AND SPAGNUOLO, M. 2004. Surface shape understanding based on Extended Reeb graphs. In *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*, S. Rana, Ed. John Wiley & Sons, 87–103.
- BIASOTTI, S., GIORGI, D., SPAGNUOLO, M., AND FALCIDIENO, B. 2006a. Computing size functions for 3D models. Tech. Rep. 3-2006, IMATI-CNR.

- BIASOTTI, S., GIORGI, D., SPAGNUOLO, M., AND FALCIDIENO, B. 2006b. Size functions for 3D shape retrieval. In *SGP'06: Proceedings of the 2006 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. 239–242.
- BIASOTTI, S., MARINI, M., SPAGNUOLO, M., AND FALCIDIENO, B. 2006. Sub-part correspondence by structural descriptors of 3D shapes. *Computer Aided Design* 38, 9 (September), 1002–1019.
- BIASOTTI, S. AND MARINI, S. 2005. 3D object comparison based on shape descriptors. *International Journal of Computer Applications in Technology* 23, 2/3/4, 57–69.
- BIASOTTI, S., MARINI, S., MORTARA, M., PATANÉ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 2003. 3D shape matching through topological structures. *Lecture Notes in Computer Science* 2886, 194–203.
- BIASOTTI, S., MORTARA, M., AND SPAGNUOLO, M. 2000b. Surface compression and reconstruction using Reeb graphs and shape analysis. In *SCCG 2000: Proceedings of the 16th Spring Conference on Computer Graphics*. ACM Press, 174–185.
- BIASOTTI, S., PATANÉ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 2007c. Analysis and comparison of real functions on triangulated surfaces, curves and surface. In *Curve and Surface Fitting*, A. Cohen, J.-L. Merrien, and L. L. Schumaker, Eds. Nashboro Press, 41–50.
- BIENIEK, A. AND MOGA, A. 1998. A connected component approach to the watershed segmentation. In *Mathematical Morphology and its Application to Image and Signal Processing*, H. Heijmans and J. Roerdink, Eds. Kluwer Acad. Publ., Dordrecht, 215–222.
- BIENIEK, A. AND MOGA, A. 2000. An efficient watershed algorithm based on connected components. *Pattern Recognition* 33, 907–916.
- BOYELL, R. L. AND RUSTON, H. 1963. Hybrid techniques for real-time radar simulation. In *Proceedings of the 1963 Fall Joint Computer Conference*.
- BREEN, E. AND JONES, R. 1996. Attribute openings, thinnings, and granulometries. *Computer Vision and Image Understanding* 64, 377–389.
- BREMER, P.-T., EDELSBRUNNER, H., HAMANN, B., AND PASCUCCI, V. 2003. A multi-resolution data structure for two-dimensional Morse functions. In *VIS '03: Proceedings of the IEEE Visualization 2003*, G. Turk, J. van Wijk, and R. Moorhead, Eds. IEEE Computer Society Press, 139–146.
- BREMER, P.-T., EDELSBRUNNER, H., HAMANN, B., AND PASCUCCI, V. 2004. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics* 10, 4 (July/August), 385–396.
- BREMER, P. T. AND PASCUCCI, V. 2007. A practical approach to two-dimensional scalar topology. In *Topology-based Methods in Visualization*, H. Hauser, H. Hagen, and H. Theisel, Eds. Springer Berlin Heidelberg, 151–169.
- BREMER, P.-T., PASCUCCI, V., AND HAMANN, B. 2005. Maximizing adaptivity in hierarchical topological models. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*. IEEE Computer Society Press, 300–309.
- BUCHIN, K., DEY, T. K., GIESEN, J., AND JOHN, K. 2007. Recursive geometry of the flow complex and topology of the flow complex filtration. *computational Geometry. Theory and Applications*.
- BUSTOS, B., KEIM, D. A., SAUPE, D., SCHRECK, T., AND VRANIĆ, D. V. 2005. Feature-based similarity search in 3D object databases. *ACM Computing Surveys* 37, 4 (December), 345–387.
- CAGLIARI, F., DI FABIO, B., AND FERRI, M. 2007. One-dimensional reduction of multidimensional persistent homology. *arXiv:math/0702713*.
- CAGLIARI, F., FERRI, M., AND POZZI, P. 2001a. Size functions from the categorical viewpoint. *Acta Applicandae Mathematicae* 67, 225–235.
- CAGLIARI, F., LANDI, C., AND GRASSELLI, L. 2001b. Presentations of Morse homology for studying shape of manifolds. Tech. Rep. 10, DISMI, Univ. di Modena e Reggio Emilia.
- CAIRNS, S. S., 1934. On the triangulation of regular loci. *The Annals of Mathematics, Second series* 35, 3, 579–587.
- CARLSSON, G. AND ZOMORODIAN, A. 2007. The theory of multidimensional persistence. In *SCG '07: Proceedings of the 23rd Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 184–193.

- CARLSSON, G., ZOMORODIAN, A., COLLINS, A., AND GUIBAS, L. 2004. Persistence barcodes for shapes. In *SGP '04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. ACM Press, 127–138.
- CARLSSON, G., ZOMORODIAN, A., COLLINS, A., AND GUIBAS, L. 2005. Persistence barcodes for shapes. *International Journal of Shape Modeling* 11, 2, 149–187.
- CARR, H. 2004. Topological manipulation of isosurfaces. Ph.D. thesis, The University of British Columbia.
- CARR, H. AND SNOEYINK, J. 2003. Path seeds and flexible isosurfaces using topology for exploratory visualization. In *VISSYM '03: Proceedings of the Symposium on Data Visualisation 2003*. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 49–58.
- CARR, H. AND SNOEYINK, J. 2007. Representing interpolant topology for contour tree computation. In *TopoInVis 2007: Workshop on Topology-based Methods in Visualization 2007*.
- CARR, H., SNOEYINK, J., AND AXEN, U. 2000. Computing contour trees in all dimensions. In *SODA '00: Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM Press, Philadelphia, PA, USA, 918–926.
- CARR, H., SNOEYINK, J., AND AXEN, U. 2003. Computing contour trees in all dimensions. *Computational Geometry: Theory and Applications* 24, 2, 75–94.
- CARR, H., SNOEYINK, J., AND VAN DE PANNE, M. 2004. Simplifying flexible isosurfaces using local geometric measures. In *VIS '04: Proceedings of the IEEE Visualization 2004*. 497–504.
- CAYLEY, A. 1859. On Contour and Slope Lines. *London, Edinburgh and Dublin Philosophical Magazine and Journal of Science* XVIII, 264–268.
- CAZALS, F., CHAZAL, F., AND LEWINER, T. 2003. Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *SCG '03: Proceedings of the 19th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 351–360.
- CERRI, A., BIASOTTI, S., AND GIORGI, D. 2007. k -dimensional size functions for shape description and comparison. In *ICIAP 2007: Proceedings of the 14th International Conference on Image Analysis and Processing*. IEEE Computer Society Press, Modena.
- CERRI, A., FERRI, M., AND GIORGI, D. 2005. A complete keypoints experiment with size functions. In *CIVR 2005: Int. Conference on Image and Video Retrieval*. Lecture Notes in Computer Science, vol. 3568. Springer, 357–366.
- CERRI, A., FERRI, M., AND GIORGI, D. 2006. Retrieval of trademark images by means of size functions. *Graphical Models* 68, 5, 451–471.
- CHIANG, Y.-J., LENZ, T., LU, X., AND ROTE, G. 2005. Simple and optimal output-sensitive construction of contour trees using monotone paths. *Computational Geometry: Theory and Applications* 30, 165–195.
- CHIANG, Y.-J. AND LU, X. 2003. Progressive Simplification of Tetrahedral Meshes Preserving All Isosurface Topologies. *Computer Graphics Forum* 22, 3, 493–504.
- COHEN-STEINER, D., EDELSBRUNNER, H., AND HARER, J. 2005. Stability of persistence diagrams. In *SCG '05: Proceedings of the 21st Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 263–271.
- COHEN-STEINER, D., EDELSBRUNNER, H., AND HARER, J. 2007a. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*. To appear.
- COHEN-STEINER, D., EDELSBRUNNER, H., AND HARER, J. 2007b. Stability of persistence diagrams. *Discrete and Computational Geometry* 37, 1, 103–120.
- COHEN-STEINER, D., EDELSBRUNNER, H., AND MOROZOV, D. 2006. Vines and vineyards by updating persistence in linear time. In *SCG '06: Proceedings of the 22nd Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 119–126.
- COLE-MCLAUGHLIN, K., EDELSBRUNNER, H., HARER, J., NATARAJAN, V., AND PASCUCCI, V. 2003. Loops in Reeb graphs of 2-manifolds. In *SCG '03: Proceedings of the 19th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 344–350.
- COLLINS, A., ZOMORODIAN, A., CARLSSON, G., AND GUIBAS, L. 2004a. A barcode shape descriptor for curve point cloud data. *Computers and Graphics* 28, 6, 881–894.

- COLLINS, A., ZOMORODIAN, A., CARLSSON, G., AND GUIBAS, L. 2004b. A barcode shape descriptor for curve point cloud data. In *SPBG'04: Symposium on Point - Based Graphics 2004*. Eurographics Association, 181–191.
- CORNEA, N., SILVER, D., AND MIN, P. 2005. Curve-Skeleton Applications. *VIS '05: Proceedings of the IEEE Visualization 2005*, 95–102.
- COUPRIE, M. AND BERTRAND, G. 1997. Topological grayscale watershed transformation. In *Vision Geometry V, Proceedings of SPIE*. Vol. 3168. 136–146.
- COX, J., KARRON, D. B., AND FERDOUS, N. 2003. Topological zone organization of scalar volume data. *Journal of Mathematical Imaging and Vision* 18, 95–117.
- D'AMICO, M. 2000. A new optimal algorithm for computing size function of shapes. In *CVPRIP Algorithms III*. Proceedings International Conference on Computer Vision, Pattern Recognition and Image Processing. 107–110.
- D'AMICO, M., FERRI, M., AND STANGANELLI, I. 2004. Qualitative asymmetry measure for melanoma detection. In *ISBI2004: Proceedings of the IEEE International Symposium on Biomedical Images*. Arlington VA, 1155–1158.
- D'AMICO, M., FROSINI, P., AND LANDI, C. 2003. Optimal matching between reduced size functions. Tech. Rep. 35, DISMI, Univ. di Modena e Reggio Emilia.
- D'AMICO, M., FROSINI, P., AND LANDI, C. 2005. Natural pseudo-distance and optimal matching between reduced size functions. Tech. Rep. 66, DISMI, Univ. di Modena e Reggio Emilia.
- D'AMICO, M., FROSINI, P., AND LANDI, C. 2006. Using matching distance in size theory: A survey. *International Journal of Imaging Systems and Technology* 16, 5, 154–161.
- DANOVARO, E., DE FLORIANI, L., MAGILLO, P., MESMOUDI, M. M., AND PUPPO, E. 2003a. Morphology-driven simplification and multi-resolution modeling of terrains. In *ACM-GIS 2003: Proceedings of the 11th International Symposium on Advances in Geographic Information Systems*, E. Hoel and P. Rigaux, Eds. ACM Press, 63–70.
- DANOVARO, E., DE FLORIANI, L., AND MESMOUDI, M. M. 2003b. Topological analysis and characterization of discrete scalar fields. In *Theoretical Foundations of Computer Vision, Geometry, Morphology, and Computational Imaging*, T. Asano, R. Klette, and C. Ronse, Eds. Lecture Notes in Computer Science, vol. 2616. Springer Verlag, 386–402.
- DANOVARO, E., DE FLORIANI, L., AND VITALI, M. 2007. Multi-resolution Morse-Smale complexes for terrain modeling. In *ICIAP 2007: Proceedings of the 14th International Conference on Image Analysis and Processing*. IEEE Computer Society Press, Modena.
- DANOVARO, E., FLORIANI, L. D., PAPALEO, L., AND VITALI, M. 2006. A multi-resolution representation of terrain morphology. In *Geographic, Information Science*, M. Raubal, H. Miller, A. Frank, and M. Goodchild, Eds. Lecture Notes in Computer Science, vol. 4197. Springer, Berlin.
- DE BERG, M., CHEONG, O., HAVERKORT, H., LIM, J., AND TOMA, L. 2007. I/O-efficient flow modeling on fat terrains. In *WADS 2007: Proceedings of the 10th International Workshop on Algorithms and Data Structures*, F. Dehne, I. Munro, J.-R. Sack, and R. Tamassia, Eds. Lecture Notes in Computer Science. Springer Verlag, 1–14.
- DE BERG, M. AND VAN KREVELD, M. 1997. Trekking in the Alps without freezing or getting tired. *Algorithmica* 19, 306–323.
- DE FLORIANI, L., MAGILLO, P., AND PUPPO, E. 1999. Multi-resolution representation of shapes based on cell complexes. In *Discrete Geometry for Computer Imagery*, G. Bertrand, M. Couprie, and L. Perrotin, Eds. Lecture Notes in Computer Science, vol. 1568. Springer Verlag, New York, 3–18.
- DE LEEUW, W. AND VAN LIERE, R. 1999. Collapsing flow topology using area metrics. In *VIS '99: Proceedings of the IEEE Visualization 1999*. IEEE Computer Society Press, Los Alamitos, CA, USA, 349–354.
- DE SILVA, V. AND CARLSSON, G. 2004. Topological estimation using witness complexes. In *PBG 2004: Proceedings of the Symposium on Point-Based Graphics 2004*. Eurographics Association, Zurich, 157–166.
- DE SILVA, V. AND GHRIST, R. 2007. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology* 7, 339–358.
- ACM Journal Name, Vol. V, No. N, February 2008.

- DEL BIMBO, A. AND PALA, P. 2006. Content-based retrieval of 3D models. *ACM Transactions on Multimedia Computing, Communications and Applications* 2, 1, 20–43.
- DELFINADO, C. J. A. AND EDELSBRUNNER, H. 1995. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design* 12, 771–784.
- DEY, T. AND WENGER, R. 2007. Stability of critical points with interval persistence. *Discrete and Computational Geometry*. To appear.
- DEY, T. K., EDELSBRUNNER, H., AND GUHA, S. 1999. Computational topology. In *Advances in Discrete and Computational Geometry*, B. Chazelle, J. E. Goodman, and R. Pollack, Eds. Contemporary Mathematics, vol. 223. AMS, Providence, 109–143.
- DEY, T. K., GIESEN, J., AND GOSWAMI, S. 2003. Shape segmentation and matching with flow discretization. In *Algorithms and Data Structures*. Lecture Notes in Computer Science, vol. 2748. Springer, 25–36.
- DEY, T. K. AND SUN, J. 2006. Defining and computing curve-skeletons with medial geodesic function. In *SGP'06: Proceedings of the 2006 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. 143–152.
- DI BATTISTA, G., EADES, P., TAMASSIA, R., AND TOLLIS, I. G. 1999. *Graph drawing: Algorithms for the visualization of graphs*. Prentice-Hall.
- DIBOS, F., FROSINI, P., AND PASQUIGNON, D. 2004. The use of size functions for comparison of shapes through differential invariants. *Journal of Mathematical Imaging and Vision* 21, 107–118.
- DONATINI, P. AND FROSINI, P. 2004a. Lower bounds for natural pseudodistances via size functions. *Archives of Inequalities and Applications* 2, 1–12.
- DONATINI, P. AND FROSINI, P. 2004b. Natural pseudodistances between closed manifolds. *Forum Mathematicum* 16, 5, 695–715.
- DONATINI, P. AND FROSINI, P. 2007. Natural pseudodistances between closed surfaces. *Journal of the European Mathematical Society* 9, 2, 231–253.
- DONATINI, P., FROSINI, P., AND LANDI, C. 1999. Deformation energy for size functions. In *EMM-CVPR'99: Proceedings of the 2nd International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, E. R. Hancock and M. Pelillo, Eds. Lecture Notes in Computer Science, vol. 1654. 44–53.
- DONG, S., BREMER, P.-T., GARLAND, M., PASCUCCI, V., AND HART, J. 2006. Spectral surface quadrangulation. *ACM Transactions on Graphics* 25, 3 (August). Proceedings SIGGRAPH 2006.
- DRYDEN, I. AND MARDIA, K. 1998. *Statistical shape analysis*. John Wiley & Sons New York.
- EDELSBRUNNER, H., FACELLO, M., AND LIANG, J. 1998. On the definition and the construction of pockets in macromolecules. *Discrete Applied Mathematics* 88, 1-3, 83–102.
- EDELSBRUNNER, H. AND HARER, J. 2002. Jacobi sets of multiple Morse functions. In *Foundations in Computational Mathematics*, F. Cucker, R. DeVore, P. Olver, and E. Sueli, Eds. Cambridge univ. Press, 37–57.
- EDELSBRUNNER, H. AND HARER, J. 2007. Persistent homology – a survey. *Contemporary Mathematics*. To appear.
- EDELSBRUNNER, H., HARER, J., MASCARENHAS, A., AND PASCUCCI, V. 2004. Time-varying Reeb graphs for continuous space-time data. In *SCG '04: Proceedings of the 20th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 366–372.
- EDELSBRUNNER, H., HARER, J., NATARAJAN, V., AND PASCUCCI, V. 2003a. Morse-Smale complexes for piecewise linear 3-manifolds. In *SCG '03: Proceedings of the 19th Annual Symposium on Computational Geometry*. ACM Press, 361–370.
- EDELSBRUNNER, H., HARER, J., AND ZOMORODIAN, A. 2001. Hierarchical Morse complexes for piecewise linear 2-manifolds. In *SCG '01: Proceedings of the 17th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 70–79.
- EDELSBRUNNER, H., HARER, J., AND ZOMORODIAN, A. 2003b. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry* 30, 87–107.

- EDELSBRUNNER, H., LETSCHER, D., AND ZOMORODIAN, A. 2000. Topological persistence and simplification. In *IEEE Foundations of Computer Science, Proceedings 41st Annual Symposium*. 454–463.
- EDELSBRUNNER, H., LETSCHER, D., AND ZOMORODIAN, A. 2002. Topological persistence and simplification. *Discrete Computational Geometry* 28, 511–533.
- EDELSBRUNNER, H., MOROZOV, D., AND PASCUCCI, V. 2006. Persistence-sensitive simplification of functions on 2-manifolds. In *SCG '06: Proceedings of the 22nd Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 127–134.
- EDELSBRUNNER, H. AND MÜCKE, E. P. 1990. Simulation of Simplicity: A technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics* 9, 1, 66–104.
- EHRESMANN, C. AND REEB, G. 1944. Sur les champs d'éléments de contact de dimension p complètement intégrable dans une variété continûment différentiable v_n . *Comptes Rendu Hebdomadaires des Séances de l'Académie des Sciences* 218, 955–957.
- FERRI, M. AND FROSINI, P. 2005. A proposal for image indexing: keypics, plastic graphical metadata. In *Internet Imaging VI, Proceedings of SPIE*. Vol. 5670. San Jose, California, USA, 225–231.
- FERRI, M., LOMBARDINI, S., AND PALLOTTI, C. 1994. Leukocyte classification by size functions. In *Proceedings of the 2nd IEEE Workshop on Applications of Computer Vision*. IEEE Computer Society Press. 223–229.
- FORMAN, R. 1998. Morse theory for cell complexes. *Advances in Mathematics* 134, 90–145.
- FORMAN, R. 2002. A user's guide to Discrete Morse Theory. *Seminaire Lotharingien de Combinatoire* 48.
- FREEMAN, H. AND MORSE, S. P. 1967. On searching a contour map for a given terrain profile. *Journal of the Franklin Institute* 248, 1–25.
- FROSINI, P. 1990. A distance for similarity classes of submanifolds of a Euclidean space. *Bulletin of the Australian Mathematical Society* 42, 407–416.
- FROSINI, P. 1991. Measuring shapes by size functions. In *Intelligent Robots and Computer Vision X: Algorithms and Techniques, Proceedings of SPIE*, D. P. Casasent, Ed. Vol. 1607. Boston, MA, 122–133.
- FROSINI, P. 1992. Discrete computation of size functions. *J. Combin. Inform. System Sci.* 17, 232–250.
- FROSINI, P. 1996. Connections between size functions and critical points. *Mathematical Methods in the Applied Sciences* 19, 555–596.
- FROSINI, P. AND LANDI, C. 1997. New pseudo-distances for the size function space. In *Vision Geometry VI, Proceedings of SPIE*, R. A. Melter, A. Y. Wu, and L. J. Latecki, Eds. Vol. 3168. 52–60.
- FROSINI, P. AND LANDI, C. 1999. Size theory as a topological tool for Computer Vision. *Pattern Recognition and Image Analysis* 9, 596–603.
- FROSINI, P. AND LANDI, C. 2001. Size functions and formal series. *Applicable Algebra in Engineering, Communication and Computing* 12, 327–349.
- FROSINI, P. AND MULAZZANI, M. 1999. Size homotopy groups for computation of natural size distances. *Bulletin of the Belgian Mathematical Society* 6, 455–464.
- FROSINI, P. AND PITTORE, M. 1999. New methods for reducing size graphs. *International Journal of Computer Mathematics* 70, 505–517.
- FUCHS, H., KEDEM, Z., AND USELTON, S. 1977. Optimal surface reconstruction from planar contours. *Communications of the ACM* 20, 693–702.
- FUJISHIRO, I., AZUMA, T., AND TAKESHIMA, Y. 1999. Automating transfer function design for comprehensible volume rendering based on 3D field topology analysis. In *VIS '99: Proceedings of the IEEE Visualization 1999*. 467–470.
- FUJISHIRO, I., TAKESHIMA, Y., AZUMA, T., AND TAKAHASHI, S. 2000. Volume data mining using 3D field topology analysis. *IEEE Computer Graphics and Applications*, 46–51.
- ACM Journal Name, Vol. V, No. N, February 2008.

- GERSTNER, T. AND PAJAROLA, R. 2000. Topology preserving and controlled topology simplifying multiresolution isosurface extraction. In *VIS '00: Proceedings of the IEEE Visualization 2000*. IEEE Computer Society Press, Los Alamitos, CA, USA, 259–266.
- GIBLIN, P. AND KIMIA, B. 2004. A formal classification of 3D medial axis points and their local geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26, 2, 238–251.
- GIERTSEN, C., HALVORSEN, A., AND FLOOD, P. 1990. Graph-directed modelling from serial sections. *The Visual Computer* 6, 284–290.
- GIESEN, J. AND JOHN, M. 2003. The flow complex: a data structure for geometric modeling. In *SODA '03: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 285–294.
- GORESKEY, M. AND MACPHERSON, R. 1988. *Stratified Morse Theory*. Springer-Verlag, New York.
- GRAMAIN, A. 1971. *Topologie des surfaces*. Presses Universitaires de France.
- GRANDIS, M. 2003. Ordinary and directed combinatorial homotopy, applied to image analysis and concurrency. *Homology Homotopy Appl.* 5, 211–231.
- GREENBERG, M. AND HARPER, J. R. 1981. *Algebraic topology: A first course*. Addison-Wesley.
- GRIFFITHS, H. B. 1976. *Surfaces*. Cambridge University Press.
- GUIBAS, L. AND STOLFI, J. 1985. Primitives for the manipulation of general subdivisions and computation of Voronoi diagrams. *ACM Transactions on Graphics* 4, 2 (April), 74–123.
- GYULASSY, A., NATARAJAN, V., PASCUCCI, V., BREMER, P.-T., AND HAMANN, B. 2005. Topology-based simplification for feature extraction from 3D scalar fields. In *VIS '05: Proceedings of the IEEE Visualization 2005*. IEEE Computer Society Press, 275–280.
- GYULASSY, A., NATARAJAN, V., PASCUCCI, V., BREMER, P.-T., AND HAMAN, B. 2006. A topological approach to simplification of three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics* 12, 4, 474–484.
- HALLEY, E.. 1702. A new and correct sea chart of the whole world shewing the variations of the compass as they were found in the year M.D.CC.. Mount & Page, London (Cartographic map).
- HANDOUYAYA, M., ZIOU, D., AND WANG, S. 1999. Sign language recognition using moment-based size functions. In *Vision Interface 99, Trois-Rivières*. 210–216.
- HARRISON, J. 2005. Lecture notes on chainlet geometry - new topological methods in geometric measure theory. *ArXiv Mathematical Physics e-prints math-ph/0505063*.
- HELMAN, J. AND HESSELINK, L. 1989. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *Computer*, 27–36.
- HETROY, F. AND ATTALI, D. 2003. Topological quadrangulations of closed triangulated surfaces using the Reeb graph. *Graphical Models* 65, 1-3, 131–148.
- HILAGA, M., SHINAGAWA, Y., KOHMURA, T., AND KUNII, T. L. 2001. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH '01: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*. ACM Press, Los Angeles, CA, 203–212.
- HIRANI, A. N. 2003. Discrete exterior calculus. Ph.D. thesis, California Institute of Technology.
- HIRSCH, M. W. 1997. *Differential Topology*. Springer.
- HUANG, X., FISHER, M., AND SMITH, D. 2003. An efficient implementation of Max Tree with linked list and Hash table. In *Proceedings of the VII Digital Image Computing: Techniques and Applications*, C. Sun, H. Talbot, S. Ourselin, and T. Adriansen, Eds. Sidney, 299–308.
- ITOH, T. AND KOYAMADA, K. 1994. Isosurface Extraction By Using Extrema Graphs. In *VIS '94: Proceedings of Visualization '94*. IEEE Computer Society Press, 77–83.
- ITOH, T. AND KOYAMADA, K. 1995. Automatic isosurface propagation using extrema graph and sorted boundary cell lists. *IEEE Transactions on Visualization and Computer Graphics* 1, 4, 319–327.
- ITOH, T., YAMAGUCHI, Y., AND KOYAMADA, K. 2001. Fast Isosurface Generation Using the Volume Thinning Algorithm. *IEEE Transactions on Visualization and Computer Graphics* 7, 1, 32–46.
- IYER, N., JAYANTI, S., LOU, K., KALYANARAMAN, Y., AND RAMANI, K. 2005. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design* 37, 5, 509–530.

- JONES, M. W., BAERENTZEN, J. A., AND SRAMEK, M. 2006. 3D distance fields: A survey of techniques and applications. *IEEE Transactions on Visualization and Computer Graphics* 12, 4, 581–599.
- JONES, R. 1999. Connected filtering and segmentation using component trees. *Computer Vision and Image Understanding* 75, 215–228.
- KACZYNSKI, T., MISCHAIKOW, K., AND MROZEK, M. 2004. *Computational Homology*. Applied Mathematical Sciences, vol. 157. Springer-Verlag.
- KANONGCHAIYOS, P. AND SHINAGAWA, Y. 2000. Articulated Reeb graphs for interactive skeleton animation. In *Multimedia modeling: Modeling Multimedia Information and System*. World Scientific, 451–467.
- KELLEY, J. AND PITCHER, E. 1947. Exact Homomorphism sequences in Homology Theory. *The Annals of Mathematics*, 2nd Ser. 48, 3, 682–709.
- KETTNER, L., ROSSIGNAC, J., AND SNOEYINK, J. 2003. The Safari Interface for Visualizing Time-Dependent Volume Data Using Iso-surfaces and Contour Spectra. *Computational Geometry: Theory and Applications* 25, 1-2, 97–116.
- KLETTE, R. AND ROSENFELD, A. 2004. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann Series in Computer Graphics and Geometric Modeling. Morgan Kaufmann.
- KOENDERINK, J. J. 1990. *Solid shape*. MIT Press, Cambridge, MA, USA.
- LAM, L., LEE, S., AND SUEN, C. 1992. Thinning Methodologies-A Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14, 9, 869–885.
- LANDI, C. AND FROSINI, P. 2002. Size functions as complete invariants for image recognition. In *Vision Geometry XI, Proceedings of SPIE*, L. J. Latecki, D. M. Mount, and A. Y. Wu, Eds. Vol. 4794. 101–109.
- LANEY, D., BREMER, P. T., MASCARENHAS, A., MILLER, P., AND PASCUCCI, V. 2006. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics* 12, 5, 1053–1060.
- LAZARUS, F. AND VERROUST, A. 1999. Level set diagrams of polyhedral objects. In *SMA '99: Proceedings of the 5th ACM Symposium on Solid Modeling and Applications 1999*, W. Bronsvort and D. Anderson, Eds. ACM Press, 130–140.
- LÉVY, B. 2006. Laplace-Beltrami eigenfunctions: Towards an algorithm that understands geometry. In *SMI '06: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006*. IEEE Computer Society Press, 13–20.
- LEWINER, T., LOPES, H., AND TAVARES, G. 2003. Towards optimality in discrete Morse theory. *Experimental Mathematics* 12, 3, 271–285.
- LEWINER, T., LOPES, H., AND TAVARES, G. 2004. Applications of Forman's discrete Morse theory to topology visualization and mesh compression. *IEEE Transactions on Visualization and Computer Graphics* 10, 5, 499–508.
- LONCARIC, S. 1998. A survey of shape analysis techniques—from automata to hardware. *Pattern Recognition* 31, 8, 983–1001.
- MAGILLO, P., DANOVARO, E., DE FLORIANI, L., PAPALEO, L., AND VITALI, M. 2007. Extracting terrain morphology: A new algorithm and a comparative evaluation. In *Proceedings of the 2nd International Conference on Computer Graphics Theory and Applications*, Barcelona, Spain.
- MANGAN, A. AND WHITAKER, R. 1999. Partitioning 3D surface meshes using watershed segmentation. *IEEE Transaction on Visualization and Computer Graphics* 5, 4, 308–321.
- MÄNTYLÄ, M. 1988. *Introduction to Solid Modeling*. WH Freeman & Co. New York, NY, USA.
- MARINI, S., SPAGNUOLO, M., AND FALCIDIENO, B. 2007. Structural shape prototypes for the automatic classification of 3D objects. *IEEE Computer Graphics and Applications* 27, 4, 28–37.
- MATTES, J. G. AND DEMONGEOT, J. 2000. Efficient algorithms to implement the confinement tree. In *DGCI 2000: Proceedings of the 9th Discrete Geometry for Computer Imagery Conference*, G. Borgefors and G. S. di Baja, Eds. Lecture Notes in Computer Science, vol. 1953. Springer Verlag, Uppsala, 392–405.

- MAXWELL, J. C. 1870. On Hills and Dales. *The London, Edinburgh and Dublin Philosophical* 40, 269, 421–425.
- MCALLISTER, M. 1999. A watershed algorithm for triangulated terrains. In *CCCG99: Proceedings of the 11th Canadian Conference on Computational Geometry*. Vancouver, BC, VIII–A 1–8.
- MEIJSTER, A. AND ROERDINK, J. 1996. Computation of watersheds based on parallel graph algorithms. In *Mathematical Morphology and its Application to Image Segmentation*, P. Maragos, R. W. Shafer, and M. A. Butt, Eds. Kluwer, 305–312.
- MERRILL, R. D. 1973. Representation of contours and regions for efficient computer search. *Communications of the ACM* 16, 2, 69–82.
- MESMOUDI, M. AND DE FLORIANI, L. 2007. Morphology-based representations of discrete scalar fields. In *GRAPP 2007: International Conference on Computer Graphics Theory*. Barcelona, Spain, 137–144.
- MESMOUDI, M. M., DANOVARO, E., DE FLORIANI, L., AND PORT, U. 2007. Surface segmentation through concentrated curvature. In *ICIAP 2007: Proceedings of the 14th International Conference on Image Analysis and Processing*. IEEE Computer Society Press, Modena.
- MEYER, F. 1994. Topographic distance and watershed lines. *Signal Processing* 38, 113–125.
- MEYERS, D., SKINNER, S., AND SLOAN, K. 1992. Surfaces from contours. *ACM Transactions on Graphics* 11, 228–258.
- MILNOR, J. 1963. *Morse Theory*. Princeton University Press, New Jersey.
- MILNOR, J. 1965. *Lectures on h-cobordism*. Princeton University Press.
- MIZUTA, S. AND MATSUDA, T. 2005. Description of Digital Images by Region-based Contour Trees. *Lecture Notes in Computer Science* 3656, 549–558.
- MONASSE, P. AND GUICHARD, F. 2000. Fast computation of a contrast-invariant image representation. *IEEE Transactions on Image Processing* 9, 5, 860–872.
- MORTARA, M. AND PATANÉ, G. 2002. Shape-covering for skeleton extraction. *International Journal of Shape Modelling* 8, 245–252.
- MORTARA, M., PATANÉ, G., SPAGNUOLO, M., FALCIDIENO, B., AND ROSSIGNAC, J. 2004. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica* 38, 1, 227–248.
- MORTENSON, M. E. 1986. *Geometric Modeling*. John Wiley & Sons.
- MUNKRES, J. 2000. *Topology*. Prentice Hall.
- NACKMAN, L. 1984. Two-dimensional Critical Point Configuration Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, 4, 442–450.
- NAJMAN, L. AND COUPRIE, M. 2003. Watershed algorithms and contrast preservation. In *DGCI 2003: Proceedings of the 11th Discrete geometry for computer imagery*, I. Nyström, G. Santini di Baja, and S. Svensson, Eds. Lecture Notes in Computer Science, vol. 2886. Springer, Naples, 62–71.
- NAJMAN, L. AND SCHMITT, M. 1996. Geodesic Saliency of Watershed Contours and Hierarchical Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 18, 12, 1163–1173.
- NATARAJAN, V., WANG, Y., BREMER, P.-T., PASCUCCI, V., AND HAMANN, B. 2006. Segmenting molecular surfaces. *Computer Aided Geometric Design* 23, 6, 495–509.
- NATARAJAN, V. J. AND PASCUCCI, V. 2005. Volumetric data analysis using Morse-Smale complexes. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*. 320–325.
- NI, X., GARLAND, M., AND HART, J. C. 2004. Fair Morse functions for extracting the topological structure of a surface mesh. *ACM Transaction on Graphics* 23, 3, 613–622.
- NIYOGI, P., SMALE, S., AND WEINBERGER, S. 2006. Finding the homology of submanifolds with high confidence from random samples. *Discrete and Computational Geometry*.
- OLIVA, J. M., PERRIN, M., AND COQUILLART, S. 1996. 3D reconstruction of complex polyhedral shapes from contours using a simplified generalised Voronoi diagram. *Computer Graphics Forum* 15, 307–318.

- PAGE, D. L. 2003. Part decomposition of 3D surfaces. Ph.D. thesis, University of Tennessee, Knoxville.
- PALIS, J. AND MELO, W. D. 1982. *Geometric Theory of Dynamical Systems: An Introduction*. Springer-Verlag.
- PAPALEO, L. 2004. Surface reconstruction: online mosaicing and modeling with uncertainty. Ph.D. thesis, University of Genova, Department of Computer Science, DISI-TH-04-04.
- PARABOSCHI, L., BIASOTTI, S., AND FALCIDIENO, B. 2007. Comparing sets of 3D digital shapes through topological structures. In *Gbr'07: Proceedings of the 6th IARP -TC-15 Workshop on Graphbased Representations in Pattern Recognition*, F. Escolano and M. Vento, Eds. Lecture Notes in Computer Science, vol. 4538. 114–125.
- PASCUCCI, V. 2004. Topology diagrams of scalar fields in scientific visualization. In *Topological Data Structures for Surfaces*, S. Rana, Ed. John Wiley & Sons Ltd, 121–129.
- PASCUCCI, V. AND COLE-MCLAUGHIN, K. 2002. Efficient computation of the topology of the level sets. In *VIS '02: Proceedings of the IEEE Visualization 2002*. IEEE Press, 187–194.
- PASCUCCI, V., COLE-MCLAUGHIN, K., AND SCORZELLI, G. 2004. Multi-resolution computation and presentation of contour trees. Tech. rep., LLNL Technical Report number UCRL-PROC-208680.
- PASCUCCI, V. AND COLE-MCLAUGHLIN, K. 2003. Parallel computation of the topology of level sets. *Algorithmica* 38, 1, 249–268.
- PASCUCCI, V., SCORZELLI, G., BREMER, P.-T., AND MASCARENHAS, A. 2007. Robust on-line computation of Reeb graphs: simplicity and speed. *ACM Transactions on Graphics* 26, 3 (August). SIGGRAPH 2007: Proceedings of the 34th Annual Conference on Computer Graphics and Interactive Techniques.
- PATANÉ, G., SPAGNUOLO, M., AND FALCIDIENO, B. 2004. Graph-based parameterization of triangle meshes with arbitrary genus. *Computer Graphics Forum* 23, 4, 783–797.
- PAVLIDIS, T. 1995. A review of algorithms for shape analysis. *Document image analysis table of contents*, 145–160.
- PEUCKER, T. K. AND DOUGLAS, D. H. 1975. Detection of Surface-Specific Points by Local Parallel Processing of Discrete Terrain Elevation Data. *Computer Graphics and Image Processing* 4, 375–387.
- PFALTZ, J. L. 1976. Surface networks. *Geographical Analysis* 8, 77–93.
- PRESS, W., TEUKOLSKY, S., VETTERLING, W., AND FLANNERY, B. 1992. *Numerical recipes in C - 2nd edition*. Cambridge University Press.
- RANA, S., Ed. 2004. *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*. John Wiley & Sons, Europe, London.
- REEB, G. 1946. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences* 222, 847–849.
- REQUICHA, A. 1980. Representations of rigid solids: Theory, methods and systems. *ACM Computing Surveys* 12, 4, 437–464.
- ROBINS, V. 1999. Towards computing homology from finite approximations. In *14th Summer Conference on General Topology and its Applications*, W. Comfort, R. Heckmann, R. Kopperman, and L. Narici, Eds. Topology Proceedings, vol. 24. 503–532.
- ROBINS, V. 2002. Computational topology for point data: Betti numbers of α -shapes. In *Morphology of Condensed Matter: Physics and Geometry of Spatially Complex Systems*, K. Mecke and D. Stoyan, Eds. Springer LNP, vol. 600. 261–274.
- ROERDINK, J. AND MELJSTER, A. 2000. The watershed transform: definitions, algorithms, and parallelization strategies. *Fundamenta Informaticae* 41, 187–228.
- SALEMBIER, P., OLIVERAS, A., AND GARRIDO, L. 1998. Antiextensive connected operators for image and sequence processing. *IEEE Transactions on Image Processing* 7, 555–570.
- SCHNEIDER, B. 2005. Extraction of hierarchical surface networks from bilinear surface patches. *Geographical Analysis* 37, 244–263.
- ACM Journal Name, Vol. V, No. N, February 2008.

- SCHNEIDER, B. AND WOOD, J. 2004. Construction of metric surface networks from raster-based DEMs. In *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*, S. Rana, Ed. John Wiley & Sons Ltd, 53–70.
- SERRA, J. 1983. *Image Analysis and Mathematical Morphology*. Academic Press, Inc., Orlando, FL, USA.
- SHAMIR, A. 2006. Segmentation and Shape Extraction of 3D Boundary Meshes. B. Wyvill and A. Wilkie, Eds. Eurographics Association, Vienna, Austria, 137–149. Eurographics 06 STAR.
- SHATTUCK, D. AND LEAHY, R. 2001. Automated graph based analysis and correction of cortical volume topology. *IEEE Transactions on Medical Imaging* 20, 1167–1177.
- SHINAGAWA, Y. AND KUNII, T. 1991. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications* 11, 44–51.
- SHINAGAWA, Y., KUNII, T. L., AND KERGOSENIEN, Y. L. 1991. Surface coding based on Morse theory. *IEEE Computer Graphics and Applications* 11, 66–78.
- SIDDIQI, K. AND PIZER, S. M., Eds. 2007. *Medial Representations: Mathematics, Algorithms and Applications*. Springer.
- SMALE, S. 1960. Morse inequalities for a dynamical system. *Bulletin of American Mathematical Society* 66, 43–49.
- SOHN, B.-S. AND BAJAJ, C. L. 2006. Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics* 12, 1, 14–25.
- SOILLE, P. 2004. *Morphological Image Analysis: Principles and Applications*. Springer-Verlag, Berlin and New York.
- SPANIER, E. H. 1966. *Algebraic Topology*. McGraw Hill.
- STANGANELLI, I., BRUCALE, A., CALORI, L., GORI, R., LOVATO, A., MAGI, S., KOPF, B., BACCHILEGA, R., RAPISARDA, V., TESTORI, A., ASCIERTO, P. A., SIMEONE, E., AND FERRI, M. 2005. Computer-aided diagnosis of melanocytic lesions. *Anticancer Research* 25, 6C, 4577–4582.
- STEINER, D. AND FISCHER, A. 2001. Topology recognition of 3D closed freeform objects based on topological graphs. In *Pacific Graphics*. 82–88.
- STOEV, S. L. AND STRASSER, W. 2000. Extracting Regions of Interest Applying a Local Watershed Transformation. In *VIS 2000: Proceedings of the IEEE Visualization 2000*. IEEE Computer Society Press, Los Alamitos, 21–28.
- SUTTON, P. AND HANSEN, C. D. 1999. Isosurface extraction in time-varying fields using a temporal branch-on-need tree (TBON). In *VIS '99: Proceedings of the IEEE Visualization 1999*. 147–154.
- SZYMCZAK, A. 2005. Subdomain aware contour trees and contour evolution in time-dependent scalar fields. In *SMI '05: Proceedings of the International Conference on Shape Modeling and Applications 2005*. IEEE Computer Society Press, 136–144.
- TAKAHASHI, S. 2004. *Algorithms for Extracting Surface Topology from Digital Elevation Models*. John Wiley & Sons Ltd, 31–51.
- TAKAHASHI, S., FUJISHIRO, I., AND TAKESHIMA, Y. 2005. Interval volume decomposer: A topological approach to volume traversal. In *Visualization and Data Analysis 2005, Proceedings of SPIE*, R. F. Erbacher, K. Börner, M. Gröhn, and J. C. Roberts, Eds. Vol. 5669. 103–114.
- TAKAHASHI, S., IKEDA, T., SHINAGAWA, Y., KUNII, T. L., AND UEDA, M. 1995. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographic elevation data. *Computer Graphics Forum* 14, 3, 181–192.
- TAKAHASHI, S., NIELSON, G. M., TAKESHIMA, Y., AND FUJISHIRO, I. 2004a. Topological volume skeletonization using adaptive tetrahedralization. In *Geometric Modelling and Processing 2004*. 227–236.
- TAKAHASHI, S., TAKESHIMA, Y., AND FUJISHIRO, I. 2004b. Topological volume skeletonization and its application to transfer function design. *Graphical Models* 66, 1, 24–49.
- TANGELDER, J. AND VELTKAMP, R. 2004. A survey of content based 3D shape retrieval methods. In *SMI '04: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2004*. 145–156.

- TARASOV, S. P. AND VYALYI, M. N. 1998. Construction of contour trees in 3D in $O(n \log n)$ steps. In *SCG '98: Proceedings of the 14th Annual Symposium on Computational Geometry*. ACM Press, 68–75.
- TARJAN, R. E. AND VON LEEUWEN, J. 1984. Worst-case analysis of set union algorithms. *Journal of the Association for Computing Machinery* 31(2), 245–281.
- THEISEL, H., ROESSL, C., AND WEINKAU, T. 2007. Morphological representations of vector fields. In *Shape Analysis and Structuring*, L. De Floriani and M. Spagnuolo, Eds.
- THOM, R. 1949. Sur une partition en cellules associée à une fonction sur une variété. *Comptes Rendu Hebdomadaires des Séances de l'Académie des Sciences* 228, 973–975.
- TORIWAKI, J. AND FUKUMURA, T. 1978. Extraction of structural information from gray pictures. *Computer Graphics and Image Processing* 7, 30–51.
- TRICOCHÉ, X., SCHEUERMANN, G., AND HAGEN, H. 2001. Continuous topology simplification of planar vector fields. In *VIS '01: Proceedings of the IEEE Visualization 2001*. IEEE Computer Society Press, Washington, DC, USA, 159–166.
- TUNG, T. AND SCHMITT, F. 2004. Augmented Reeb graphs for content-based retrieval of 3D mesh models. In *SMI '04: Proceedings of the International Conference on Shape Modeling and Applications 2004*. IEEE Computer Society Press, Los Alamitos, 157–166.
- TUNG, T. AND SCHMITT, F. 2005. The Augmented Multiresolution Reeb Graph approach for content-based retrieval of 3D shapes. *International Journal of Shape Modelling* 11, 1 (June), 91–120.
- TURNER, M. J. 1998. Design of entropy conscious and constrained image operations using a contour tree image definitionⁿ. In *2nd IMA Conference on Image Processing: Mathematical Methods, Algorithms and Applications (Sept. 22-25, 1998)*.
- URAS, C. AND VERRI, A. 1994. On the recognition of the alphabet of the sign language through size functions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition, Jerusalem (Israel)*. Vol. II. IEEE Computer Society Press, Los Alamitos, CA, 334–338.
- URAS, C. AND VERRI, A. 1997. Computing size functions from edge maps. *International Journal of Computer Vision* 23(2), 169–183.
- VAN KREVELD, M., VAN OOSTRUM, R., BAJAJ, C. L., PASCUCCI, V., AND SCHIKORE, D. 1997. Contour trees and small seed sets for isosurface traversal. In *SCG '97: Proceedings of the 13th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 212–220.
- VAN KREVELD, M., VAN OOSTRUM, R., BAJAJ, C. L., PASCUCCI, V., AND SCHIKORE, D. 2004. Contour trees and small seed sets for isosurface generation. In *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*, S. Rana, Ed. John Wiley & Sons, 71–86.
- VEGTER, G. 1997. Computational topology. In *Handbook of Discrete and Computational Geometry*, J. E. Goodman and J. O'Rourke, Eds. CRC Press, Boca Raton, New York, 517–536.
- VEGTER, G. AND YAP, C. K. 1990. Computational complexity of combinatorial surfaces. In *SCG '90: Proceedings of the 6th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 102–111.
- VELTKAMP, R. C. AND HAGENDOORN, M. 2001. State-of-the-Art in shape matching. In *Principles of Visual Information Retrieval*, M. Lew, Ed. Springer-Verlag, 87–119.
- VERRI, A. AND URAS, C. 1994. Invariant size functions. In *Applications of Invariance in Computer Vision*, J. L. Mundy, A. Zisserman, and D. Forsyth, Eds. Lecture Notes in Computer Science, vol. 825. Springer, 215–234.
- VERRI, A. AND URAS, C. 1996. Metric-topological approach to shape representation and recognition. *Image and Vision Computing* 14, 189–207.
- VERRI, A., URAS, C., FROSINI, P., AND FERRI, M. 1993. On the use of size functions for shape analysis. *Biological Cybernetics* 70, 99–107.
- VERROUST, A. AND LAZARUS, F. 2000. Extracting skeletal curves from 3D scattered data. *The Visual Computer* 16, 15–25.
- VINCENT, L. AND SOILLE, P. 1991. Watershed in digital spaces: An efficient algorithm based on immersion simulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 13, 6, 583–598.

- WANG, Y., AGARWAL, P., BROWN, P., EDELSBRUNNER, H., AND RUDOLPH, J. 2005. Coarse and reliable geometric alignment for protein docking. In *Proceedings of Pacific Symposium on Biocomputing 2005*.
- WATSON, L. T., LAFFEY, T. J., AND HARALICK, R. M. 1985. Topographic classification of digital image intensity surfaces using generalized splines and the discrete cosine transformation. *Computer Vision, Graphics, and Image Processing* 29, 143–167.
- WEBER, G. H., DILLARD, S. E., CARR, H., PASCUCCI, V., AND HAMANN, B. 2007. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics* 13, 2, 330–341.
- WEBER, G. H. AND SCHEUERMANN, G. 2004. Automating transfer function design based on topology analysis. In *Geometric Modeling for Scientific Visualization*, G. Brunnett, B. Hamann, H. Mueller, and L. Linsen, Eds. Springer.
- WEBER, G. H., SCHEUERMANN, G., AND HAMANN, B. 2003. Detecting critical regions in scalar fields. In *VISSYM '03: Proceedings of the Symposium on Data Visualisation 2003*, G.-P. Bonneau, S. Hahmann, and C. D. Hansen, Eds. EUROGRAPHICS and IEEE TCVG, Association for Computing Machinery, New York, New York, 85–94.
- WEBER, G. H., SCHUEUERMANN, G., HAGEN, H., AND HAMANN, B. 2002. Exploring scalar fields using critical isovalues. In *VIS '02: Proceedings of the IEEE Visualization 2002*. IEEE Computer Society Press, 171–178.
- WERGHI, N., XIAO, Y., AND SIEBERT, J. P. 2006. A functional-based segmentation of human body scans in arbitrary postures. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* 36, 1, 153–165.
- WILLARD, S. 1970. *General topology*. Addison-Wesley Publishing Company.
- WOLF, G. W. 2004. Topographic surfaces and surface networks. In *Topological Data Structures for Surfaces*, S. Rana, Ed. John Wiley & Sons Ltd, 15–29.
- WOLTER, F. 1992. Cut locus and medial axis in global shape interrogation and representation. Memorandum 92-2.
- WOOD, Z., HOPPE, H., DESBRUN, M., AND SCHROEDER, P. 2004. Removing excess topology from isosurfaces. *ACM Transactions on Graphics* 23, 190–208.
- WOOD, Z. J., DESBRUN, M., SCHROEDER, P., AND BREEN, D. 2000. Semi-regular mesh extraction from volumes. In *VIS '02: Proceedings of the IEEE Visualization 2002*. IEEE Computer Society Press, Los Alamitos, 275–282.
- XIAO, Y., SIEBERT, P., AND WERGHI, N. 2003. A discrete Reeb graph approach for the segmentation of human body scans. In *3DIM 2003: Proceedings of the 4th International Conference on 3-D Digital Imaging and Modeling*. IEEE Computer Society Press, Los Alamitos, 378–385.
- YU, S., VAN, M., AND SNOEYINK, J. 1996. Drainage queries in TINs: from local to global and back again. In *Advances in GIS Research II: Proceedings 7th International Symposium on Spatial Data Handling*. 1–14.
- ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2005. Feature-based surface parameterization and texture mapping. *ACM Transaction on Graphics* 24, 1, 1–27.
- ZHANG, X., BAJAJ, C. L., AND BAKER, N. 2004. Fast Matching of Volumetric Functions Using Multi-resolution Dual Contour Trees. Tech. rep., Texas Institute for Computational and Applied Mathematics, Austin, Texas.
- ZIOU, D. AND ALLILI, M. 2002. Generating cubical complexes from image data and computation of the Euler number. *Pattern Recognition* 35, 2833–2839.
- ZOMORODIAN, A. 2005. *Topology for Computing*. Cambridge University Press, New York, NY.
- ZOMORODIAN, A. AND CARLSSON, G. 2004. Computing persistent homology. In *SCG '04: Proceedings of the 20th Annual Symposium on Computational Geometry*. ACM Press, New York, NY, USA, 255–262.
- ZOMORODIAN, A. AND CARLSSON, G. 2005. Computing persistent homology. *Discrete and Computational Geometry* 33, 2, 249–274.
- ZOMORODIAN, A. AND CARLSSON, G. 2007. Localized homology. In *SMI '07: Proceedings of the IEEE International Conference on Shape Modeling and Applications 2007*. IEEE Computer Society Press, 189–198.

