Computational Topology*

Afra Zomorodian
Dartmouth College

November 3, 2009

1 Introduction

According to the Oxford English Dictionary, the word *topology* is derived of *topos* (τόπος) meaning *place*, and *-logy* (λογια), a variant of the verb λέγειν, meaning *to speak*. As such, topology speaks about places: how local neighborhoods connect to each other to form a space. Computational topology, in turn, undertakes the challenge of studying topology using a computer.

The field of geometry studies intrinsic properties that are invariant under rigid motion, such as the curvature of a surface. In contrast, topology studies invariants under continuous deformations. The larger set of transformations enables topology to extract more qualitative information about a space, such as the number of connected components or tunnels. Computational topology has theoretical and practical goals. Theoretically, we look at the tractability and complexity of each problem, as well as the design of efficient data structures and algorithms. Practically, we are interested in heuristics and fast software for solving problems that arise in diverse disciplines. Our input is often a finite discrete set of noisy samples from some underlying space. This type of input has renewed interest in combinatorial and algebraic topology, areas that had been overshadowed by point set topology in the last one hundred years.

Computational topology developed in response to topological impediments emerging from within geometric problems. In computer graphics, researchers encountered the problem of connectivity in reconstructing *watertight* surfaces from point sets. Often, their heuristics resulted in surfaces with extraneous holes or tunnels that had to be detected and removed before proper geometric processing was feasible. Researchers in computational geometry provided guaranteed surface reconstruction algorithms that depended on sampling conditions on the input. These results required concepts from topology, provoking an interest in the subject. Topological problems also arise naturally in areas that do not deal directly with geometry. In robotics, researchers need to understand the connectivity of the *configuration space* of a robot for computing optimal trajectories that minimize resource consumption. In biology, the thermodynamics hypothesis states that proteins fold to their native states along such optimal trajectories. In sensor networks, determining coverage without localizing the sensors requires deriving global information from local connections. Once again the question of connectivity arises, steering us toward a topological understanding of the problem.

Like topology, computational topology is a large and diverse area. The aim of this chapter is not to be comprehensive, but to describe the fundamental concepts, methods, and structures that permeate the field. We begin with our objects of study, topological spaces and their combinatorial representations, in Section 2. We study these spaces through topological invariants, which we formalize in Section 3, classifying all surfaces, and introducing both combinatorial and algebraic invariants in the process. Sections 4 and 5 focus on homology and persistent homology, algebraic invariants that derive their popularity from their computability. Geometry and topology are intrinsically entangled, as revealed by Morse theory through additional

*Chapter 3, Volume 2 of *Algorithms and Theory of Computation Handbook*, Second Edition, 2009.

structures in Section 6. For topological data analysis, we describe methods for deriving combinatorial structures that represents point sets in Section 7. We end the chapter with a brief discussion of geometric issues in Section 8.

2 Topological Spaces

The focus of this section is topological spaces: their definitions, finite representations, and data structures for efficient manipulation.

2.1 Topology

Metric spaces are endowed with a metric that defines open sets, neighborhoods, and continuity. Without a metric, we must prescribe enough structure to extend these concepts. Intuitively, a topological space is a set of points, each of whom knows its neighbors. A *topology* on a set X is a subset $T \subseteq 2^X$ such that:

1. If $S_1, S_2 \in T$, then $S_1 \cap S_2 \in T$.
2. If $\{S_J \mid j \in J\} \subseteq T$, then $\cup_{j \in J} S_j \in T$.
3. $\emptyset, X \in T$.

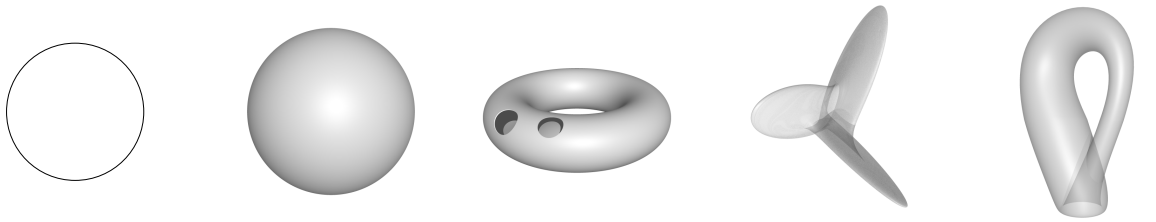
A set $S \in T$ is an *open set* and its complement in X is *closed*. The pair $\mathbb{X} = (X, T)$ is a *topological space*. A set of points may be endowed with different topologies, but we will abuse notation by using $p \in \mathbb{X}$ for $p \in X$.

A function $f : \mathbb{X} \to \mathbb{Y}$ is *continuous* if for every open set A in $\mathbb{Y}$, $f^{-1}(A)$ is open in $\mathbb{X}$. We call a continuous function a *map*. The *closure* $\overline{A}$ of A is the intersection of all closed sets containing A. The *interior* $\mathring{A}$ of A is the union of all open sets contained in A. The *boundary* ∂A of A is $\partial A = \overline{A} - \mathring{A}$. A *neighborhood* of $x \in X$ is any $S \in T$ such that $x \in \mathring{S}$. A subset $A \subseteq X$ with *induced topology* $T_A = \{S \cap A \mid S \in T\}$ is a *subspace* of $\mathbb{X}$. A *homeomorphism* $f : \mathbb{X} \to \mathbb{Y}$ is a 1-1 onto function such that both f and f^{-1} are continuous. We say that $\mathbb{X}$ is *homeomorphic* to $\mathbb{Y}$, $\mathbb{X}$ and $\mathbb{Y}$ have the same *topological type*, and denote it $\mathbb{X} \approx \mathbb{Y}$.

2.2 Manifolds

A topological space may be viewed as an abstraction of a metric space. Similarly, manifolds generalize the connectivity of d-dimensional Euclidean spaces $\mathbb{R}^d$ by being locally similar, but globally different. A *d-dimensional chart* at $p \in \mathbb{X}$ is a homeomorphism $\varphi : U \to \mathbb{R}^d$ onto an open subset of $\mathbb{R}^d$, where U is a neighborhood of p and *open* is defined using the metric [Apostol, 1969]. A *d-dimensional manifold (d-manifold)* is a (separable Hausdorff) topological space $\mathbb{X}$ with a d-dimensional chart at every point $x \in \mathbb{X}$. We do not define the technical terms *separable* and *Hausdorff* as they mainly disallow pathological spaces. The circle or *1-sphere* $\mathbb{S}^1$ in Figure 1(a) is a 1-manifold as every point has a neighborhood homeomorphic to an open interval in $\mathbb{R}^1$. All neighborhoods on the *2-sphere* $\mathbb{S}^2$ in Figure 1(b) are homeomorphic to open disks, so $\mathbb{S}^2$ is a 2-manifold, also called a *surface*. The *boundary* $\partial\mathbb{X}$ of a d-manifold $\mathbb{X}$ is the set of points in $\mathbb{X}$ with neighborhoods homeomorphic to $\mathbb{H}^d = \{x \in \mathbb{R}^d \mid x_1 \geq 0\}$. If the boundary is nonempty, we say $\mathbb{X}$ is a *manifold with boundary*. The boundary of a d-manifold with boundary is always a $(d-1)$-manifold without boundary. Figure 1(c) displays a *torus* with boundary, the boundary being two circles.

We may use a homeomorphism to place one manifold within another. An *embedding* $g : \mathbb{X} \to \mathbb{Y}$ is a homeomorphism onto its image $g(\mathbb{X})$. This image is called an *embedded submanifold*, such as the spaces in Figure 1(a, b, c). We are mainly interested in *compact* manifolds. Again, this is a technical definition that



(a) $\mathbb{S}^1$ (b) $\mathbb{S}^2$ (c) Torus with ∂ (d) $\mathbb{P}^2$ (e) Klein bottle

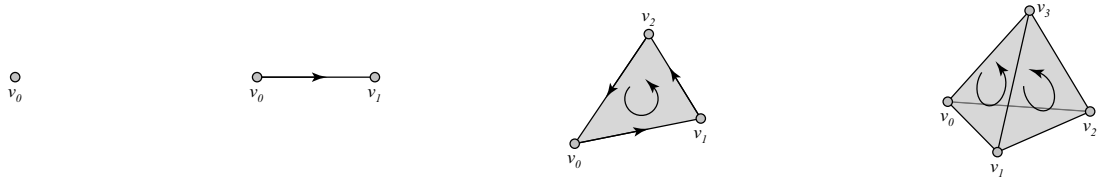
Figure 1: Manifolds. (a) The only compact connected 1-manifold is a circle $\mathbb{S}^1$. (b) The sphere is a 2-manifold. (c) The surface of a donut, a torus, is also a 2-manifold. This torus has two boundaries, so it's a manifold with boundary. Its boundary is two circles, a 1-manifold. (d) A Boy's surface is a geometric immersion of the projective plane $\mathbb{P}^2$, a nonorientable 2-manifold. (e) The Klein bottle is another nonorientable 2-manifold.

requires the notion of a limit. For us, a compact manifold is closed and can be embedded in $\mathbb{R}^d$ so that it has finite extent, i.e. it is bounded. An *immersion* $g : \mathbb{X} \to \mathbb{Y}$ of a compact manifold is a *local* embedding: an immersed compact space may self-intersect, such as the immersions in Figure 1(d, e).

2.3 Simplicial Complexes

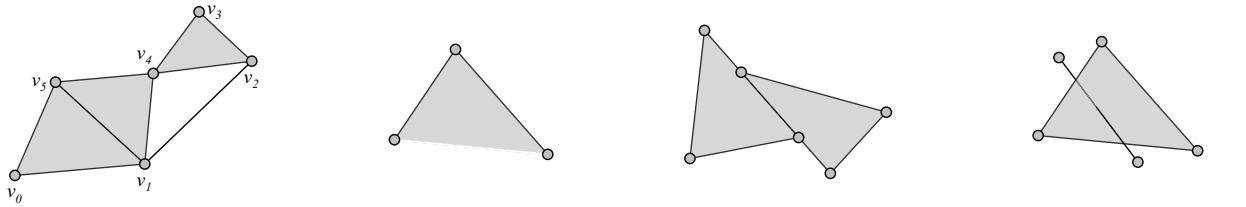
To compute information about a topological space using a computer, we need a finite representation of the space. In this section, we represent a topological space as a union of simple pieces, deriving a *combinatorial* description that is useful in practice. Intuitively, *cell complexes* are composed of Euclidean pieces glued together along seams, generalizing *polyhedra*. Due to their structural simplicity, simplicial complexes are currently a popular representation for topological spaces, so we describe their construction.

A *k-simplex* σ is the convex hull of $k+1$ affinely independent points $S = \{v_0, v_1, \ldots, v_k\}$, as shown in Figure 2. The points in S are the *vertices* of the simplex. A k-simplex σ is a k-dimensional subspace of $\mathbb{R}^d$, $\dim \sigma = k$. Since the defining points of a simplex are affinely independent, so is any subset of them. A simplex τ defined by a subset $T \subseteq S$ is a *face* of σ and has σ as a *coface*. For example, the triangle in Figure 2(c) is a coface of 8 faces: itself, its three edges, three vertices, and the (-1)*-simplex* defined by the empty set, which is a face of every simplex. A *simplicial complex* K is a finite set of simplices such that every face of a simplex in K is also in K, and the nonempty intersection of any two simplices of K is a face of each of them. The *dimension* of K is the dimension of a simplex with the maximum dimension in K. The *vertices* of K are the zero-simplices in K. Figure 3 displays a simplicial complex and several badly formed unions. A simplex is *maximal* if it has no proper coface in K. In the simplicial complex in the figure, the triangles and edge v_1v_2 are maximal.



(a) Vertex $[v_0]$ (b) Edge $[v_0, v_1]$ (c) Triangle $[v_0, v_1, v_2]$ (d) Tetrahedron $[v_0, v_1, v_2, v_3]$

Figure 2: Oriented k-simplices, $0 \le k \le 3$. An oriented simplex induces orientation on its faces, as shown for the edges of the triangle and two faces of the tetrahedron.



(a) A simplicial complex (b) Missing edge (c) Shared partial edge (d) Non-face intersection

Figure 3: A simplicial complex (a) and three unions (b, c, d) that are not complexes. The triangle (b) does not have all its faces, and (c, d) have intersections not along shared faces.

The *underlying space* $|K|$ of a simplicial complex K is the topological space $|K| = \cup_{\sigma \in K} \sigma$, where we regard each simplex as a topological subspace. A *triangulation* of a topological space $\mathbb{X}$ is a simplicial complex K such that $|K| \approx \mathbb{X}$. We say $\mathbb{X}$ is *triangulable* when K exists. Triangulations enable us to represent topological spaces compactly as simplicial complexes. For instance, the surface of the tetrahedron in Figure 2(d) is a triangulation of the sphere $\mathbb{S}^2$ in Figure 1(b). Two simplicial complexes K and L are *isomorphic* iff $|K| \approx |L|$.

We may also define simplicial complexes without utilizing geometry, thereby revealing their combinatorial nature. An *abstract simplicial complex* is a set $\mathcal{S}$ of finite sets such that if $A \in \mathcal{S}$, so is every subset of A. We say $A \in \mathcal{S}$ is an *(abstract) k-simplex* of dimension k if $|A| = k + 1$. A vertex is a 0-simplex, and the face and coface definitions follow as before. Given a (geometric) simplicial complex K with vertices V, let $\mathcal{S}$ be the collection of all subsets $\{v_0, v_1, \ldots, v_k\}$ of V such that the vertices $v_0, v_1, \ldots, v_k$ span a simplex of K. The collection $\mathcal{S}$ is the *vertex scheme* of K and is an abstract simplicial complex. For example, the vertex scheme of the simplicial complex in Figure 3(a) is:

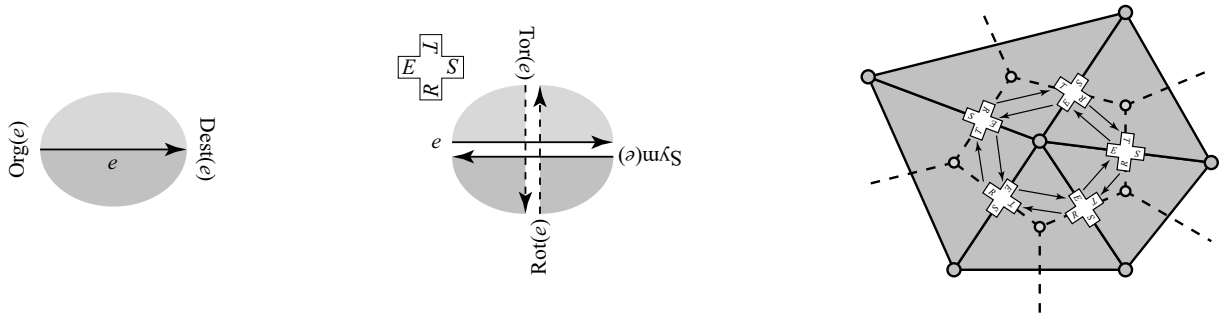
$$\left\{ \begin{array}{l} \emptyset, \\ \{v_0\}, \{v_1\}, \{v_2\}, \{v_3\}, \{v_4\}, \{v_5\}, \\ \{v_0, v_1\}, \{v_0, v_5\}, \{v_1, v_2\}, \{v_1, v_4\}, \{v_1, v_5\}, \{v_2, v_3\}, \{v_2, v_4\}, \{v_3, v_4\}, \{v_4, v_5\}, \\ \{v_0, v_1, v_5\}, \{v_1, v_4, v_5\}, \{v_2, v_3, v_4\} \end{array} \right\} \tag{1}$$

Let $\mathcal{S}_1, \mathcal{S}_2$ be abstract simplicial complexes with vertices V_1, V_2, respectively. An *isomorphism* between $\mathcal{S}_1, \mathcal{S}_2$ is a bijection $\varphi : V_1 \to V_2$, such that the sets in $\mathcal{S}_1$ and $\mathcal{S}_2$ are the same under the renaming of the vertices by φ and its inverse.

There is a strong relationship between the geometric and abstract definitions. Every abstract simplicial complex $\mathcal{S}$ is isomorphic to the vertex scheme of some simplicial complex K, which is its *geometric realization*. For example, Figure 3(a) is a geometric realization of vertex scheme (1). Two simplicial complexes are isomorphic iff their vertex schemes are isomorphic. As we shall see in Section 7, we usually compute simplicial complexes using geometric techniques, but discard the realization and focus on its topology as captured by the vertex scheme. As such, we refer to abstract simplicial complexes simply as simplicial complexes from now on.

A *subcomplex* is a subset $L \subseteq K$ that is also a simplicial complex. An important subcomplex is the *k-skeleton*, which consists of simplices in K of dimension less than or equal to k. The smallest subcomplex containing a subset $L \subseteq K$ is its *closure*, $\operatorname{Cl} L = \{\tau \in K \mid \tau \subseteq \sigma \in L\}$. The *star* of L contains all of the cofaces of L, $\operatorname{St} L = \{\sigma \in K \mid \sigma \supseteq \tau \in L\}$. The *link* of L is the boundary of its star, $\operatorname{Lk} L = \operatorname{Cl} \operatorname{St} L - \operatorname{St} (\operatorname{Cl} L - \{\emptyset\})$. Stars and links correspond to open sets and boundaries in topological spaces.

Suppose we fix an order on the set of vertices. An *orientation* of a k-simplex $\sigma \in K$, $\sigma = \{v_0, v_1, \ldots, v_k\}$,



(a) Directed edge (b) Quadedge (c) Overlay

Figure 4: The Quadedge data structure. (a) A directed edge e from $\mathrm{Org}(e)$ to $\mathrm{Dest}(e)$. (b) A quadedge is composed of four edges: e, $\mathrm{Sym}(e)$ from $\mathrm{Dest}(e)$ to $\mathrm{Org}(e)$, and dual edges $\mathrm{Rot}(e)$ and $\mathrm{Tor}(e)$ that go from right to left and in reverse, respectively. The edges are stored together in a single data structure. (c) A portion of a triangulation overlaid with dashed dual edges and quadedge data structure.

$v_i \in K$ is an equivalence class of orderings of the vertices of σ, where $(v_0, v_1, \ldots, v_k) \sim (v_{\tau(0)}, v_{\tau(1)}, \ldots, v_{\tau(k)})$ are equivalent orderings if the parity of the permutation τ is even. An *oriented simplex* is a simplex with an equivalence class of orderings, denoted as a sequence $[\sigma]$. We may show an orientation graphically using arrows, as in Figure 2. An oriented simplex *induces* orientations on its faces, where we drop the vertices not defining a face in the sequence to get the orientation. For example, triangle $[v_0, v_1, v_2]$ induces oriented edge $[v_0, v_1]$. Two k-simplices sharing a $(k-1)$-face τ are *consistently oriented* if they induce different orientations on τ. A triangulable d-manifold is *orientable* if all d-simplices in any of its triangulations can be oriented consistently. Otherwise, the d-manifold is *nonorientable*.

2.4 Data Structures

There exist numerous data structures for storing cell complexes, especially simplicial complexes. In triangulations, all simplices are maximal and have the same dimension as the underlying manifold. For instance, all the maximal simplices of a triangulation of a 2-manifold, with or without boundary, are triangles, so we may just store triangles and infer the existence of lower-dimensional simplices. This is the basic idea behind *triangle soup* file formats for representing surfaces in graphics, such as *OBJ* or *PLY*.

For topological computation, we require quick access to our neighbors. The key insight is to store both the complex and its dual at once, as demonstrated by the *quadedge* data structure for surfaces [Guibas and Stolfi, 1985]. This data structure focuses on a directed edge e, from its origin $\mathrm{Org}(e)$ to its destination $\mathrm{Dest}(e)$. Each directed edge separates two regions to its left and right, as shown in Figure 4(a). The data structure represents each edge in the triangulation with four edges: the original edge and edges $\mathrm{Rot}(e)$ directed from right to left, $\mathrm{Sym}(e)$ from $\mathrm{Dest}(e)$ to $\mathrm{Org}(e)$, and $\mathrm{Tor}(e)$ from left to right, as shown in Figure 4(b). The edges e and $\mathrm{Sym}(e)$ are in *primal* space and the edges $\mathrm{Rot}(e)$ and $\mathrm{Tor}(e)$ are in *dual* space. Intuitively, we get these edges by rotating by $90°$ counter-clockwise repeatedly, moving between primal and dual spaces. In addition to storing geometric data, each edge also stores a pointer to the next counter-clockwise edge with the same origin, returning it with operation $\mathrm{Onext}(e)$. Figure 4(c) shows a portion of a triangulation, its dual, and its quad-edge data structure. The quadedge stores the four edges together in a single data structure, so all operations are $O(1)$. The operations form an *edge algebra* that allows us to walk quickly across the surface. For example, to get the next clockwise edge with the same origin, we can define $\mathrm{Oprev} \equiv \mathrm{Rot} \circ \mathrm{Onext} \circ \mathrm{Rot}$, as can be checked readily on Figure 4(c).

For arbitrary-dimensional simplicial complexes, we begin by placing a full ordering on the vertices. We

then place all simplices in a hash table, using their vertices as unique keys. This allows for $O(1)$ access to faces, but not to cofaces, which require an inverse lookup table.

Computing orientability. Access to the dual structure enables easy computation of many topological properties, such as orientability. Suppose K is a cell complex whose underlying space is d-manifold. We begin by orienting one cell arbitrarily. We then use the dual structure to consistently orient the neighboring cells. We continue to spread the orientation until either all cells are consistently oriented, or we arrive at a cell that is inconsistently oriented. In the former case, K is orientable, and in the latter, it is nonorientable. This algorithm takes time $O(n_d)$, where n_d is the number of d-cells.

3 Topological Invariants

Last section, we focused on topological spaces and their combinatorial representations. In this section, we examine the properties of topologically equivalent spaces. The equivalence relation for topological spaces is the homeomorphism, which places spaces of the same topological type into the same class. We are interested in *intrinsic* properties of spaces within each class, that is, properties that are invariant under homeomorphism. Therefore, a fundamental problem in topology is the *Homeomorphism Problem*: Can we determine whether two objects are homeomorphic?

As we shall see, the homeomorphism problem is computationally intractable for manifolds of dimension greater than three. Therefore, we often look for partial answers in the form of invariants. A *topological invariant* is a map f that assigns the same object to spaces of the same topological type, that is:

$$\mathbb{X} \approx \mathbb{Y} \implies f(\mathbb{X}) = f(\mathbb{Y}) \tag{2}$$

$$f(\mathbb{X}) \neq f(\mathbb{Y}) \implies \mathbb{X} \not\approx \mathbb{Y} \quad \text{(contrapositive)} \tag{3}$$

$$f(\mathbb{X}) = f(\mathbb{Y}) \implies \text{nothing (in general)} \tag{4}$$

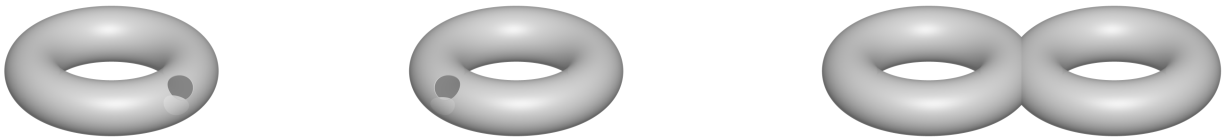
Note that an invariant is only useful through the contrapositive. The *trivial* invariant assigns the same object to all spaces and is therefore useless. The *complete* invariant assigns different objects to non-homeomorphic spaces, so the inverse of implication (2) is true. Most invariants fall in-between these two extremes. A topological invariant implies a classification that is coarser than, but respects, the topological type. In general, the more powerful an invariant, the harder it is to compute it, and as we relax the classification to be coarser, the computation becomes easier. In this section, we learn about several combinatorial and algebraic invariants.

3.1 Topological Type for Manifolds

For compact manifolds, the homeomorphism problem is well understood. In one dimension, there is only a single manifold, namely $\mathbb{S}^1$ in Figure 1(a), so the homeomorphism problem is trivially solved. In two dimensions, the situation is more interesting, so we begin by describing an operation for connecting manifolds. The *connected sum* of two d-manifolds $\mathbb{M}_1$ and $\mathbb{M}_2$ is

$$\mathbb{M}_1 \# \mathbb{M}_2 = \mathbb{M}_1 - \mathbb{D}_1^d \bigcup_{\partial\mathbb{D}_1^d = \partial\mathbb{D}_2^d} \mathbb{M}_2 - \mathbb{D}_2^d, \tag{5}$$

where $\mathbb{D}_1^d, \mathbb{D}_2^d$ are d-dimensional disk in $\mathbb{M}_1, \mathbb{M}_2$, respectively. In other words, we cut out two disks and glue the manifolds together along the boundary of those disks using a homeomorphism, as shown in Figure 5 for two tori. The connected sum of g tori or g projective planes (Figure 1(d)) is a surface of *genus* g, and the former has g *handles*.

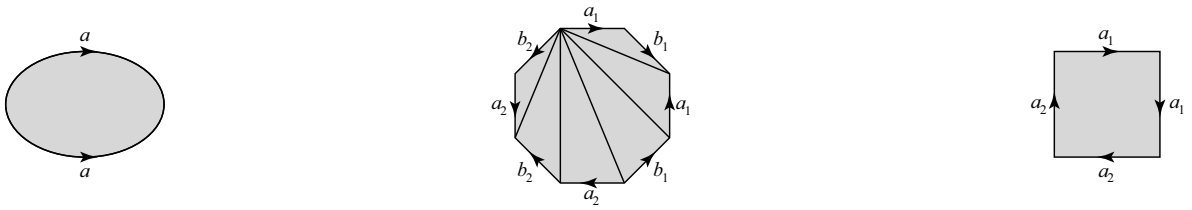


(a) Torus minus disk (b) Torus minus disk (c) Double torus

Figure 5: Connected sum. We cut out a disk from each 2-manifold and glue the manifolds together along the boundary of those disks to get a larger manifold. Here, we sum two tori to get a surface with two handles.

Using the connected sum, the homeomorphism problem is fully resolved for compact 2-manifolds: Every triangulable compact surface is homeomorphic to a sphere, the connected sum of tori, or the connected sum of projective planes. Dehn and Heegaard [1907] give the first rigorous proof of this result. Rado [1925] completes their proof by showing that all compact surfaces are triangulable. A classic approach for this proof represents surfaces with *polygonal schema*, polygons whose edges are labeled by symbols, as shown in Figure 6 [Seifert and Threlfall, 2003]. There are two edges for each symbol, and $\overline{a}$ represents an edge in the direction opposite to edge a. If we glue the edges according to direction, we get a compact surface. There is a *canonical form* of the polygonal schema for all surfaces. For the sphere, it is the special 2-gon in Figure 6(a). For orientable manifolds with genus g, it is a $4g$-gon with form $a_1b_1\overline{a_1}\overline{b_1}\ldots a_gb_g\overline{a_g}\overline{b_g}$, as shown for the genus two torus in Figure 6(b). For nonorientable manifolds with genus g, it is a $2g$-gon with form $a_1a_1\ldots a_ga_g$, as shown for the genus two surface (Klein bottle) in Figure 6(c). On the surface, the edges of the polygonal schema form loops called *canonical generators*. We can cut along these loops to flatten the surface into a topological disk, that is, the original polygon. We may also triangulate the polygon to get a minimal triangulation with a single vertex called the *canonical triangulation*, as shown in Figure 6(b) for the double-torus.

In dimensions four and higher, Markov [1958] proves that the homeomorphism problem is undecidable. For the remaining dimension, $d = 3$, the problem is harder to resolve. Thurston [1982] conjectures that 3-manifolds may be decomposed into pieces with uniform geometry in his *Geometrization Program*. Grigori Perelman completes Thurston's program in 2003 by eliminating geometric singularities that may arise from Ricci flow on the manifolds [Morgan and Tian, 2007].



(a) $a\overline{a}$ (b) $a_1b_1\overline{a_1}\overline{b_1}a_2b_2\overline{a_2}\overline{b_2}$ (c) $a_1a_1a_2a_2$

Figure 6: Canonical forms for polygonal schema. The edges are glued according to the directions. (a) The Sphere $\mathbb{S}^2$ has a special form. (b) A genus g orientable manifold is constructed of a $4g$-gon with the specified gluing pattern, so this is the double torus in Figure 5(c). Adding diagonals, we triangulate the polygon to get a canonical triangulation for the surface. (c) A genus g nonorientable manifold is constructed out of a $2g$-gon with the specified gluing pattern. This 4-gon is the Klein bottle in Figure 1(e), the connected sum of two projective planes in Figure 1(d).

Computing schema and generators. Brahana [1921] gives an algorithm for converting a polygonal schema into its canonical form. Based on this algorithm, Vegter and Yap [1990] present an algorithm for triangulated surfaces that runs in $O(n \log n)$ time, where n is the size of the complex. They also sketch an optimal algorithm for computing the canonical generators in $O(gn)$ time for a surface of genus g. For orientable surfaces, Lazarus et al. [2001] simplify this algorithm, describe a second optimal algorithm for computing canonical generators, and implement both algorithms. Canonical generators may be viewed as a one-vertex graph whose removal from the surface gives us a polygon. A *cut graph* generalizes this notion as any set of edges whose removal transforms the surface to a topological disk, a non-canonical schema. Dey and Schipper [1995] compute a cut graph for cell complexes via a breadth-first search of the dual graph.

3.2 The Euler Characteristic

Since the homeomorphism problem can be undecidable, we resort to invariants for classifying spaces. Let K be any cell complex and n_k be the number of k-dimensional cells in K. The *Euler characteristic* $\chi(K)$ is

$$\chi(K) = \sum_{k=0}^{\dim K} (-1)^k n_k. \tag{6}$$

The Euler characteristic is an integer invariant for $|K|$, so we get the same integer for different complexes with the same underlying space. For instance, the surface of the tetrahedron in Figure 2(d) is a triangulation of the sphere $\mathbb{S}^2$ and we have $\chi(\mathbb{S}^2) = 4 - 6 + 4 = 2$. A cube is a cell complex homeomorphic to $\mathbb{S}^2$, also giving us $\chi(\mathbb{S}^2) = 8 - 12 + 6 = 2$.

For compact surfaces, the Euler characteristic, along with orientability, provide us with a complete invariant. Recall the classification of surfaces from Section 3.1. Using small triangulations, it is easy to show $\chi(\text{Torus}) = 0$ and $\chi(\mathbb{P}^2) = 1$. Similarly, we can show that for compact surfaces $\mathbb{M}_1$ and $\mathbb{M}_2$, $\chi(\mathbb{M}_1 \# \mathbb{M}_2) = \chi(\mathbb{M}_1) + \chi(\mathbb{M}_2) - 2$. We may combine these results to compute the Euler characteristic of connected sums of manifolds. Let $\mathbb{M}_g$ be the connected sum of g tori and $\mathbb{N}_g$ be the connected sum of g projective planes. Then, we have

$$\chi(\mathbb{M}_g) = 2 - 2g, \tag{7}$$

$$\chi(\mathbb{N}_g) = 2 - g. \tag{8}$$

Computing topological types of 2-manifolds. We now have an two-phase algorithm for determining the topological type of an unknown triangulated compact surface $\mathbb{M}$: 1) Determine orientability and 2) compute $\chi(\mathbb{M})$. Using the algorithm in Section 2.4, we determine orientability in $O(n_2)$, where n_2 is the number of triangles in $\mathbb{M}$. We then compute the Euler characteristic using Equation (6) in $O(n)$, where n is the size of $\mathbb{M}$. The characteristic gives us g, completing the classification of $\mathbb{M}$ in linear time.

3.3 Homotopy

A *homotopy* is a family of maps $f_t : \mathbb{X} \to \mathbb{Y},\ t \in [0,1]$, such that the associated map $F : \mathbb{X} \times [0,1] \to \mathbb{Y}$ given by $F(x,t) = f_t(x)$ is continuous. Then, $f_0, f_1 : \mathbb{X} \to \mathbb{Y}$ are *homotopic* via the homotopy f_t, denoted $f_0 \simeq f_1$. A map $f : \mathbb{X} \to \mathbb{Y}$ is a *homotopy equivalence* if there exists a map $g : \mathbb{Y} \to \mathbb{X}$, such that $f \circ g \simeq 1_{\mathbb{Y}}$ and $g \circ f \simeq 1_{\mathbb{X}}$, where $1_{\mathbb{X}}, 1_{\mathbb{Y}}$ are the identity maps on the respective spaces. Then, $\mathbb{X}$ and $\mathbb{Y}$ are *homotopy equivalent* and have the same *homotopy type*, denoted $\mathbb{X} \simeq \mathbb{Y}$. The difference between two spaces being *homeomorphic* versus being *homotopy equivalent* is in what the compositions of the two functions f and g are forced to be: In the former, the compositions must be *equivalent* to the identity maps; in the latter, they only need to be *homotopic* to them. A space with the homotopy type of a point is *contractible* or is *null-homotopic*. Homotopy is a topological invariant since $\mathbb{X} \approx \mathbb{Y} \Rightarrow \mathbb{X} \simeq \mathbb{Y}$. Markov's undecidability proof for

the homeomorphism problem extends to show undecidability for homotopy equivalence for d-manifolds, $d \geq 4$ [Markov, 1958].

A *deformation retraction* of a space $\mathbb{X}$ onto a subspace $\mathbb{A}$ is a family of maps $f_t : \mathbb{X} \to \mathbb{X}$, $t \in [0, 1]$ such that f_0 is the identity map, $f_1(\mathbb{X}) = \mathbb{A}$, and $f_t|_{\mathbb{A}}$, the restriction of f_t to $\mathbb{A}$, is the identity map for all t. The family f_t should be continuous in the sense describe above. Note that f_1 and the inclusion $i \colon \mathbb{A} \to \mathbb{X}$ give a homotopy equivalence between $\mathbb{X}$ and $\mathbb{A}$. For example, a cylinder is not homeomorphic to a circle, but homotopy equivalent to it, as we may continuously shrink a cylinder to a circle via a deformation retraction.

Computing homotopy for curves. Within computational geometry, the *homotopy problem* has been studied for curves on surfaces: Are two given curves homotopic? Suppose the curves are specified by n line segments in the plane, with n point obstacles, such as holes. Cabello et al. [2004] give an efficient $O(n \log n)$ time algorithm for simple paths, and an $O(n^{3/2} \log n)$ time algorithm for self-intersecting paths. For the general case, assume we have two closed curves of size k_1 and k_2 on a genus g 2-manifold $\mathbb{M}$, represented by a triangulation of size n. Dey and Guha [1999] utilize combinatorial group theory to give an algorithm that decides if the paths are homotopic in $O(n + k_1 + k_2)$ time and space, provided $g \neq 2$ if $\mathbb{M}$ is orientable, or $g \neq 3, 4$ if $\mathbb{M}$ is nonorientable.

When one path is a point, the homotopy problem reduces to asking whether a closed curve is contractible. This problem is known also as the *contractibility problem*. While the above algorithms apply, the problem has been studied on its own. For a curve of size k on a genus g surface, Dey and Schipper [1995] compute a non-canonical polygonal schema to decide contractibility in suboptimal $O(n + k \log g)$ time and $O(n + k)$ space.

3.4 The Fundamental Group

The Euler characteristic is an invariant that describes a topological space via a single integer. *Algebraic topology* gives invariants that associate richer algebraic structures with topological spaces. The primary mechanism is via *functors*, which not only provide algebraic images of topological spaces, but also provide images of maps between the spaces. In the remainder of this section and the next two sections, we look at three functors as topological invariants.

The fundamental group captures the structure of homotopic loops within a space. A *path* in $\mathbb{X}$ is a continuous map $f : [0, 1] \to \mathbb{X}$. A *loop* is a path f with $f(0) = f(1)$, that is, a loop starts and ends at the same *basepoint*. The *trivial loop* never leaves its basepoint. The equivalence class of a path f under the equivalence relation of homotopy is $[f]$. A loop that is the boundary of a disk is a *boundary* and since it is contractible, it belongs to the class of the trivial loop. Otherwise, the loop is *non-bounding* and belongs to a different class. We see a boundary and two non-bounding loops on a torus in Figure 7. Given two paths $f, g : [0, 1] \to \mathbb{X}$ with $f(1) = g(0)$, the *product path* $f \cdot g$ is a path which traverses f and then g. The *fundamental group* $\pi_1(\mathbb{X}, x_0)$ of $\mathbb{X}$ and basepoint x_0 has the homotopy classes of loops in $\mathbb{X}$ based at x_0 as

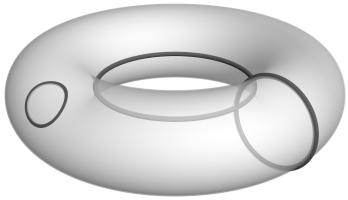


Figure 7: Loops on a torus. The loop on the left is a boundary as it bounds a disk. The other two loops are non-bounding and together generate the fundamental group of a torus, $\pi_1(\text{torus}) \cong \mathbb{Z} \times \mathbb{Z}$.

its elements, the equivalence class of the trivial loop as its identity element, and $[f][g] = [f \cdot g]$ as its binary operation. For a path-connected space $\mathbb{X}$, the basepoint is irrelevant, so we refer to the group as $\pi_1(\mathbb{X})$.

Clearly, any loop drawn on $\mathbb{S}^2$ is bounding, so $\pi_1(\mathbb{S}^2) \cong \{0\}$, where $\cong$ denotes group isomorphism and $\{0\}$ is the trivial group. For any $n \in \mathbb{Z}$, we may go around a circle n times, going clockwise for positive and counter-clockwise for negative integers. Intuitively then, $\pi_1(\mathbb{S}^1) \cong \mathbb{Z}$. Similarly, the two non-bounding loops in Figure 7 generate the fundamental group of a torus, $\pi_1(\text{torus}) \cong \mathbb{Z} \times \mathbb{Z}$, where the surface of the torus turns the group Abelian.

The fundamental group was instrumental in two significant historical problems. Markov [1958] used the fundamental group to reduce the homeomorphism problem for manifolds to the *group isomorphism* problem, which was known to be undecidable [Adyan, 1955]. Essentially, Markov's reduction is through the construction of a d-manifold, $d \geq 4$, whose fundamental group is a given group. In 1904, Poincaré conjectured that if the fundamental group of a three-manifold is trivial, then the manifold is homeomorphic to $\mathbb{S}^3$: $\pi_1(\mathbb{M}) \cong \{0\} \implies M \approx \mathbb{S}^3$. Milnor [2008] provides a detailed history of the *Poincaré Conjecture*, which was subsumed by Thurston's Geometrization program and completed by Perelman.

The fundamental group is the first in a series of *homotopy groups* $\pi_n(\mathbb{X})$ that extend the notion of loops to n-dimensional cycles. These invariant groups are very complicated and not directly computable from a cell complex. Moreover, for an n-dimensional space, only a finite number may be trivial, giving us an infinite description that is not computationally feasible.

Computation. For orientable surfaces, canonical generators or one-vertex cut graphs (Section 3.1) generate the fundamental group. While the fundamental group of specific spaces have been studied, there is no general algorithm for computing the fundamental group for higher-dimensional spaces.

4 Simplicial Homology

Homology is a topological invariant that is quite popular in computational topology as it is easily computable. Homology groups may be regarded as an algebraization of the first layer of geometry in cell complexes: how cells of dimension n attach to cells of dimension $n-1$ [Hatcher, 2002]. We define homology for simplicial complexes, but the theory extends to arbitrary topological spaces.

4.1 Definition

Suppose K is a simplicial complex. A *k-chain* is $\sum_i n_i[\sigma_i]$, where $n_i \in \mathbb{Z}$, $\sigma_i \in K$ is a k-simplex, and the brackets indicate orientation (Section 2.3). The *kth chain group* $C_k(K)$ is the set of all chains, the free Abelian group on oriented k-simplices. For example, $[v_1, v_2] + 2[v_0, v_5] + [v_2, v_3] \in C_1(K)$ for the complex in Figure 3(a) on Page 4. To reduce clutter, we will omit writing the complex below but emphasize that all groups are defined with respect to some complex. We relate chain groups in successive dimensions via the boundary operator. For k-simplex $\sigma = [v_0, v_1, \ldots, v_k] \in K$, we let

$$\partial_k \sigma \quad = \quad \sum_i (-1)^i [v_0, v_1, \ldots, \hat{v_i}, \ldots, v_n], \tag{9}$$

where $\hat{v_i}$ indicates that v_i is deleted from the sequence. The *boundary homomorphism* $\partial_k : C_k \to C_{k-1}$ is the linear extension of the operator above, where we define $\partial_0 \equiv 0$. For instance,

$$\begin{aligned} \partial_1([v_1, v_2] + 2[v_0, v_5] + [v_2, v_3]) &= \partial_1[v_1, v_2] + 2\partial_1[v_0, v_5] + \partial_1[v_2, v_3] \\ &= (v_2 - v_1) + 2(v_5 - v_0) + (v_3 - v_2) \\ &= -2v_0 - v_1 + v_3 + 2v_5, \end{aligned}$$

where we have omitted brackets around vertices. A key fact, easily shown, is that $\partial_{k-1}\partial_k \equiv 0$ for all $k \geq 1$. The boundary operator connects the chain groups into a *chain complex* C_*:

$$\ldots \to C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \to \ldots. \tag{10}$$

A k-chain with no boundary is a *k-cycle*. For example, $[v_1, v_2] + [v_2, v_3] + [v_3, v_4] - [v_1, v_4]$ is a 1-cycle as its boundary is empty. The k-cycles constitute the kernel of ∂_k, so they form a subgroup of C_k, the *kth cycle group* Z_k:

$$Z_k = \ker \partial_k = \{c \in C_k \mid \partial_k c = 0\}. \tag{11}$$

A k-chain that is the boundary of a $(k+1)$-chain is a *k-boundary*. A k-boundary lies in the image of ∂_{k+1}. For example, $[v_0, v_1] + [v_1, v_4] + [v_4, v_5] + [v_5, v_0]$ is a 1-boundary as it is the boundary of the 2-chain $[v_0, v_1, v_5] + [v_1, v_4, v_5]$. The k-boundaries form another subgroup of C_k, the *kth boundary group* B_k:

$$B_k = \operatorname{im} \partial_{k+1} = \{c \in C_k \mid \exists d \in C_{k+1} : c = \partial_{k+1} d\}. \tag{12}$$

Cycles and boundaries are like loops and boundaries in the fundamental group, except the former may have multiple components and are not required to share basepoints. Since $\partial_{k-1}\partial_k \equiv 0$, the subgroups are nested, $B_k \subseteq Z_k \subseteq C_k$. The *$k$th homology group* H_k is:

$$H_k = Z_k \,/\, B_k = \ker \partial_k \,/\, \operatorname{im} \partial_{k+1}. \tag{13}$$

If $z_1, z_2 \in Z_k$ are in the same homology class, z_1 and z_2 are *homologous*, denoted $z_1 \sim z_2$. From group theory, we know that we can write $z_1 = z_2 + b$, where $b \in B_k$. For example, the 1-cycles $z_1 = [v_1, v_2] + [v_2, v_3] + [v_3, v_4] + [v_4, v_1]$ and $z_2 = [v_1, v_2] + [v_2, v_4] + [v_4, v_1]$ are homologous in Figure 3(a) as they both describe the hole in the complex. We have $z_1 = z_2 + b$, where $b = [v_2, v_3] + [v_3, v_4] + [v_4, v_2]$ is a 1-boundary.

The homology groups are invariants for $|K|$ and for homotopy equivalent spaces. Formally, $\mathbb{X} \simeq \mathbb{Y} \Rightarrow H_k(\mathbb{X}) \cong H_k(\mathbb{Y})$ for all k. The invariance of simplicial homology gave rise to the *Hauptvermutung* (principle conjecture) by Poincaré in 1904, which was shown to be false in dimensions higher than three [Ranicki, 1997]. Homology's invariance was finally resolved through the axiomatization of homology and the general theory of *singular homology*.

4.2 Characterization

Since H_k is a finitely-generated group, the standard structure theorem states that it decomposes uniquely into a direct sum

$$\bigoplus_{i=1}^{\beta_k} \mathbb{Z} \;\oplus\; \bigoplus_{j=1}^{m} \mathbb{Z}_{t_j}, \tag{14}$$

where $\beta_k, t_j \in \mathbb{Z}$, $t_j | t_{j+1}$, $\mathbb{Z}_{t_j} = \mathbb{Z}/t_j\mathbb{Z}$ [Dummit and Foote, 2003]. This decomposition defines key invariants of the group. The left sum captures the free subgroup and its rank is the *kth Betti number* β_k of K. The right sum captures the torsion subgroup and the integers t_j are the *torsion coefficients* for the homology group. Table 1 lists the homology groups for basic 2-manifolds.

For torsion-free spaces in three-dimensions, the Betti numbers have intuitive meaning as a consequence of the *Alexander Duality*. β_0 counts the number of connected *components* of the space. β_1 is the dimension of any basis for the *tunnels*. β_2 counts the number of enclosed spaces or *voids*. For example, the torus is one connected component, has two tunnels that are delineated by the cycles in Figure 7, and encloses one void. Therefore $\beta_0 = 1$, $\beta_1 = 2$, and $\beta_2 = 1$.

2-manifold	H_0	H_1	H_2
sphere	$\mathbb{Z}$	$\{0\}$	$\mathbb{Z}$
torus	$\mathbb{Z}$	$\mathbb{Z} \times \mathbb{Z}$	$\mathbb{Z}$
projective plane	$\mathbb{Z}$	$\mathbb{Z}_2$	$\{0\}$
Klein bottle	$\mathbb{Z}$	$\mathbb{Z} \times \mathbb{Z}_2$	$\{0\}$

Table 1: Homology groups of basic 2-manifolds.

4.3 The Euler-Poincaré Formula

We may redefine our first invariant, the Euler characteristic from Section 3.2, in terms of the chain complex C_*. Recall that for a cell complex K, $\chi(K)$ is an alternating sum of n_k, the number of k-simplices in K. But since C_k is the free group on oriented k-simplices, its rank is precisely n_k. In other words, we have $\chi(K) = \chi(C_*(K)) = \sum_k (-1)^k \operatorname{rank}(C_k(K))$. We now denote the sequence of homology functors as H_*. Then, $H_*(C_*(K))$ is another chain complex:

$$\ldots \to H_{k+1} \xrightarrow{\partial_{k+1}} H_k \xrightarrow{\partial_k} H_{k-1} \to \ldots. \tag{15}$$

The maps between the homology groups are induced by the boundary operators: We map a homology class to the class that contains it. According to our new definition, the Euler characteristic of our chain is $\chi(H_*(C_*(K))) = \sum_k (-1)^k \operatorname{rank}(H_k) = \sum_k (-1)^k \beta_k$, the alternating sum of the Betti numbers. Surprisingly, the homology functor preserves the Euler characteristic of a chain complex, giving us the *Euler-Poincaré* formula:

$$\chi(K) = \sum_k (-1)^k n_k = \sum_k (-1)^k \beta_k, \tag{16}$$

where $n_k = |\{\sigma \in K \mid \dim \sigma = k\}|$ and $\beta_k = \operatorname{rank} H_k(K)$. For example, we know $\chi(\mathbb{S}^2) = 2$ from Section 3.2. From Table 1, we have $\beta_0 = 1$, $\beta_1 = 0$, and $\beta_2 = 1$ for $\mathbb{S}^2$, and $1 - 0 + 1 = 2$. The formula derives the invariance of the Euler characteristic from the invariance of homology. For a surface, it also allows us to compute one Betti number given the other two Betti numbers, as we may compute the Euler characteristic by counting simplices.

4.4 Computation

We may view any group as a $\mathbb{Z}$-module, where $\mathbb{Z}$ is the *ring of coefficients* for the chains. This view allows us to use alternate rings of coefficients, such as finite fields $\mathbb{Z}_p$ for a prime p, reals $\mathbb{R}$, or rationals $\mathbb{Q}$. Over a field F, a module becomes a vector space and is fully characterized by its dimension, the Betti number. This means that we get a full characterization for torsion-free spaces when computing over fields.

Computing connected components. For torsion-free spaces, we may use the *union-find* data structure to compute connected components or β_0 [Cormen et al., 2001, Chapter 21]. This data structure maintains a collection of disjoint dynamic sets using three operations that we use below. For a vertex, we create a new set using MAKE-SET and increment β_0. For an edge, we determine if the endpoints of the edge are in the same component using two calls to FIND-SET. If they are not, we use UNION to unite the two sets and decrement β_0. Consequently, we compute β_0 in time $O(n_1 + n_0\alpha(n_0))$, where n_k is the number of k-simplices, and α is the inverse of the Ackermann function which is less than the constant 4 for all practical values. Since the algorithm is incremental, it also yields Betti numbers for filtrations, as defined in Section 5. Using duality, we may extend this algorithm to compute higher-dimensional Betti numbers for surfaces and

three-dimensional complexes whose highest Betti numbers are known. Delfinado and Edelsbrunner [1995] use this approach to compute Betti numbers for subcomplexes of triangulations of $\mathbb{S}^3$.

Computing β_k over fields. Over a field F, our problem lies within linear algebra. Since $\partial_k \colon C_k \to C_{k-1}$ is a linear map, it has a matrix M_k with entries from F in terms of bases for its domain and codomain. For instance, we may use oriented simplices as respective bases for C_k and C_{k-1}. The cycle vector space $Z_k = \ker \partial_k$ is null space of M_k and the boundary vector space $B_k = \operatorname{im} \partial_{k+1}$ is the range space of M_{k+1}. By Equation (13), $\beta_k = \dim H_k = \dim Z_k - \dim B_k$. We may compute these dimensions with two Gaussian eliminations in time $O(n^3)$, where n is the number of simplices in K [Uhlig, 2002]. Here, we are assuming that the field operations take time $O(1)$, which is true for simple fields. The matrix M_k is very sparse for a cell complex and the running time is faster than cubic in practice.

Computing β_k over PIDs. The structure theorem also applies for rings of coefficients that are *principal ideal domains (PIDs)*. For our purposes, a PID is simply a ring in which we may compute the greatest common divisor of a pair of elements. Starting with the matrix M_k, the *reduction algorithm* uses elementary column and row operations to derive alternate bases for the chain groups, relative to which the matrix for ∂_k has the diagonal *Smith normal form*:

$$\left[\begin{array}{ccc|c} b_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & b_{l_k} & \\ \hline & 0 & & 0 \\ & & & \end{array}\right], \tag{17}$$

where $b_j > 1$ and $b_j | b_{j+1}$. The torsion coefficients for H_{k-1} are now precisely those diagonal entries b_j that are greater than one. We may also compute the Betti numbers from the ranks of the matrices, as before.

Over $\mathbb{Z}$, neither the size of the matrix entries nor the number of operations in $\mathbb{Z}$ is polynomially bounded for the reduction algorithm. Consequently, there are sophisticated polynomial algorithms based on modular arithmetic [Storjohann, 1998], although reduction is preferred in practice [Dumas et al., 2003]. It is important to reiterate that homology computation is easy over fields, but hard over $\mathbb{Z}$, as this distinction is a source of common misunderstanding in the literature.

Computing generators. In addition to the Betti numbers and torsion coefficients, we may be interested in actual descriptions of homology classes through representative cycles. To do so, we may modify Gaussian elimination and the reduction algorithm to keep track of all the elementary operations during elimination. This process is almost never used since applications for generators are motivated geometrically, but the computed generators are usually geometrically displeasing. We discuss geometric descriptions in Section 8.2.

5 Persistent Homology

Persistent homology is a recently developed invariant that captures the homological history of a space that is undergoing growth. We model this history for a space X as a *filtration*, a sequence of nested spaces $\emptyset = X_0 \subseteq X_1 \subseteq \ldots \subseteq X_m = X$. A space with a filtration is a *filtered space*. We see a filtered simplicial complex in Figure 8. Note that in a filtered complex, the simplices are always added, but never removed, implying a *partial order* on the simplices. Filtrations arise naturally from many processes, such as the

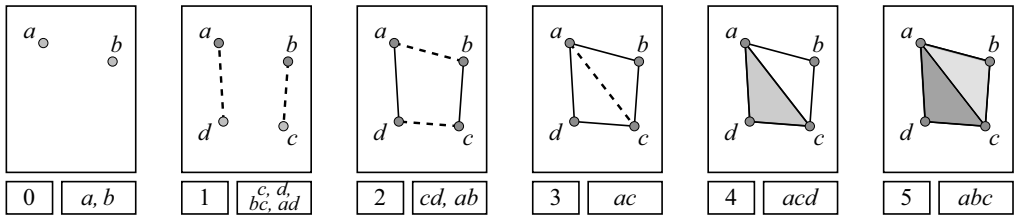


Figure 8: A simple filtration of a simplicial complex with newly added simplices highlighted and listed.

excursion sets of Morse functions over manifolds in Section 6 or the multiscale representations for point sets in Section 7.

Over fields, each space X_j has a kth homology group $H_k(X_j)$, a vector space whose dimension $\beta_k(X_j)$ counts the number of topological attributes in dimension k. Viewing a filtration as a growing space, we see that topological attributes appear and disappear. For instance, the filtered complex in Figure 8 develops a 1-cycle at time 2 that is completely filled at time 5. If we could track an attribute through the filtration, we could talk about its *lifetime* within this growth history. This is the intuition behind the theory of *persistent homology*. Initially, Edelsbrunner et al. [2002] developed the concept for subcomplexes of $\mathbb{S}^3$ and $\mathbb{Z}_2$ coefficients. The treatment here follows later theoretical development that extended the concept to arbitrary spaces and field coefficients [Zomorodian and Carlsson, 2005].

5.1 Theory

A filtration yields a *directed space*

$$\emptyset = X_0 \xhookrightarrow{i} \cdots \xhookrightarrow{i} X_j \cdots \xhookrightarrow{i} X, \tag{18}$$

where the maps i are the respective inclusions. Applying the H_k to both the spaces and the maps, we get another directed space

$$\emptyset = H_k(X_0) \xrightarrow{H_k(i)} \cdots \xrightarrow{H_k(i)} H_k(X_j) \xrightarrow{H_k(i)} \cdots \xrightarrow{H_k(i)} H_k(X) \tag{19}$$

where $H_k(i)$ are the respective maps induced by inclusion. The *persistent homology* of the filtration is the structure of this directed space, and we would like to understand it by classifying and parameterizing it. Intuitively, we do so by building a single structure that contains all the complexes in the directed space. Suppose we are computing over the ring of coefficients R. Then, $R[t]$ is the *polynomial ring*, which we may grade with the *standard grading* Rt^n. We then form a graded module over $R[t]$, $\oplus_{i=0}^{n} H_k(i)$, where the R-module structure is simply the sum of the structures on the individual components, and where the action of t is given by $t \cdot (m_0, m_1, m_2, \ldots) = (0, \varphi_0(m_0), \varphi_1(m_1), \varphi_2(m_2), \ldots)$. and φ_i is the map induced by homology in the directed space. That is, t simply shifts elements of the module up in the gradation. In this manner, we encode the ordering of simplices within the coefficients from the polynomial ring. For instance, vertex a enters the filtration in Figure 8 at time 0, so $t \cdot a$ exists at time 1, $t^2 \cdot a$ at time 2, and so on.

When the ring of coefficients is a field F, $F[t]$ is a PID, so the standard structure theorem gives the unique decomposition of the $F[t]$-module:

$$\bigoplus_{i=1}^{n} \Sigma^{\alpha_i} F[t] \;\oplus\; \bigoplus_{j=1}^{m} \Sigma^{\gamma_j} F[t]/(t^{n_j}), \tag{20}$$

where $(t^n) = t^n F[t]$ are the ideals of the graded field $F[t]$ and Σ^α denotes an α-shift upward in grading. Except for the shifts, the decomposition is essentially the same as the decomposition (14) on Page 11 for groups: the left sum describes free generators and the right sum the torsion subgroup.

Barcodes. Using decomposition (20), we may parameterize the isomorphism classes with a multiset of intervals:

1. Each left summand give us $[\alpha_i, \infty)$, corresponding to a topological attribute that is *created* at time α_i and exists in the final structure.

2. Each right summand gives us $[\gamma_j, \gamma_j + n_j)$, corresponding to an attribute that is created at time γ_j, lives for time n_j, and is *destroyed.*

We refer to this multiset of intervals as a *barcode*. Like a homology group, a barcode is a homotopy invariant. For the filtration in Figure 8, the β_0-barcode is $\{[0, \infty), [0, 2)\}$ with the intervals describing the lifetimes of the components created by simplices a and b, respectively. The β_1-barcode is $\{[2, 5), [3, 4)\}$ for the two 1-cycles created by edges ab and ac, respectively, provided ab enters the complex after cd at time 2.

5.2 Algorithm

While persistence was initially defined for simplicial complexes, the theory extends to singular homology, and the algorithm extends to a broad class of filtered cell complexes, such as simplicial complexes, simplicial sets, Δ-complexes, and cubical complexes. In this section, we describe an algorithm that computes both persistence barcodes, but also descriptions of the representatives of persistent homology classes [Zomorodian and Carlsson, 2008]. We present the algorithm for cell complexes over $\mathbb{Z}_2$ coefficients, but the same algorithm may be easily adapted to other complexes or fields.

Recall that a filtration implies a partial order on the cells. We begin by sorting cells within each time snapshot by dimension, breaking other ties arbitrarily, getting a *full order*. The algorithm, listed in pseudocode in the procedure PAIR-CELLS, takes this full order as input. Its output consists of persistence information: It partitions the cells into *creators* and *destroyers* of homology classes, and pairs the cells that are associated to the same class. Also, the procedure computes a generator for each homology class.

The procedure PAIR-CELLS is incremental, processing one cell at a time. Each cell σ stores its $partner$, its paired cell, as shown in Table 2 for the filtration in Figure 8. Once we have this pairing, we may simply read off the barcode from the filtration. For instance, since vertex b is paired with edge cd, we get interval $[0, 2)$ in the β_0-barcode. Each cell σ also stores its $cascade$, a k-chain, which is initially σ itself. The algorithm focuses the impact of σ's entry on the topology by determining whether $\partial\sigma$ is already a boundary in the complex using a call to the procedure ELIMINATE-BOUNDARIES. After the call, there are two possibilities on line 5:

1. If $\partial(cascade[\sigma]) = 0$, we are able to write $\partial\sigma$ as a sum of the boundary basis elements, so $\partial\sigma$ is already a $(k-1)$-boundary. But now, $cascade[\sigma]$ is a new k-cycle that σ completed. That is, σ creates a new homology cycle and is a *creator*.

2. If $\partial(cascade[\sigma]) \neq 0$, then $\partial\sigma$ becomes a boundary after we add σ, so σ destroys the homology class of its boundary and is a *destroyer*. On lines 6 through 8, we pair σ with the *youngest* cell τ in $\partial(cascade[\sigma])$, that is, the cell that has most recently entered the filtration.

During each iteration, the algorithm maintains the following invariants: It identifies the ith cell as a creator or destroyer and computes its cascade; if σ is a creator, its cascade is a generator for the homology class it creates; otherwise, the boundary of its cascade is a generator for the boundary class.

PAIR-CELLS(K)

1 **for** $\sigma \leftarrow \sigma_1$ **to** $\sigma_n \in K$
2 **do** $partner[\sigma] \leftarrow \emptyset$
3 $cascade[\sigma] \leftarrow \sigma$
4 ELIMINATE-BOUNDARIES(σ)
5 **if** $\partial(cascade[\sigma]) \neq 0$
6 **then** $\tau \leftarrow$ YOUNGEST($\partial(cascade[\sigma])$)
7 $partner[\sigma] \leftarrow \tau$
8 $partner[\tau] \leftarrow \sigma$

The procedure ELIMINATE-BOUNDARIES corresponds to the processing of one row (or column) in Gaussian elimination. We repeatedly look at the youngest cell τ in $\partial(cascade[\sigma])$. If it has no partner, we are done. If it has one, then the cycle that τ created was destroyed by its partner. We then add τ's partner's cascade to σ's cascade, which has the effect of adding a boundary to $\partial(cascade[\sigma])$. Since we only add boundaries, we do not change homology classes.

ELIMINATE-BOUNDARIES(σ)

1 **while** $\partial(cascade[\sigma]) \neq 0$
2 **do** $\tau \leftarrow$ YOUNGEST($\partial(cascade[\sigma])$)
3 **if** $partner[\tau] = \emptyset$
4 **then return**
5 **else** $cascade[\sigma] \leftarrow cascade[\sigma] + cascade[partner[\tau]]$

Table 2 shows the stored attributes for our filtration in Figure 8 after the algorithm's completion. For example, tracing ELIMINATE-BOUNDARIES(cd) through the iterations of the **while** loop, we get:

#	$cascade[cd]$	τ	$partner[\tau]$
1	cd	d	ad
2	$cd + ad$	c	bc
3	$cd + ad + bc$	b	$\emptyset$

Since $partner[b] = \emptyset$ at the time of bc's entry, we pair b with cd in PAIR-CELLS upon return from this procedure, as shown in the table. Table 2 contains all the persistence information we require. If $\partial(cascade[\sigma]) = 0$, σ is a creator and its cascade is a representative of the homology class it created. Otherwise, σ is a destroyer and its cascade is a chain whose boundary is a representative of the homology class σ destroyed.

σ	a	b	c	d	bc	ad	cd	ab	ac	acd	abc
$partner[\sigma]$		cd	bc	ad	c	d	b	abc	acd	ac	ab
$cascade[\sigma]$	a	b	c	d	bc	ad	cd	ab	ac	acd	abc
							ad	cd	cd		acd
							bc	ad	ad		
								bc			

Table 2: Data structure after running the persistence algorithm on the filtration in Figure 8. The simplices without partners, or with partners that come after them in the full order, are creators; the others are destroyers.

6 Morse Theoretic Invariants

In the last three sections, we have looked at a number of combinatorial and algebraic invariants. In this section, we examine the deep relationship between topology and geometry. We begin with the following question: Can we define any smooth function on a manifold? Morse theory shows that the *topology* of the manifold regulates the *geometry* of the function on it, answering the question in the negative. However, this relationship also implies that we may study topology through geometry. Our study yields a number of combinatorial structures that capture the topology of the underlying manifold.

6.1 Morse Theory

In this section, we generally assume we have a 2-manifold $\mathbb{M}$ embedded in $\mathbb{R}^3$. We also assume the manifold is *smooth*, although we do not extend the notions formally due to lack of space, but depend on the reader's intuition Boothby [1986]. Most concepts generalize to higher-dimensional manifolds with Riemannian metrics. Let p be a point on $\mathbb{M}$. A *tangent vector v_p to* $\mathbb{R}^3$ consists of two points of $\mathbb{R}^3$: its *vector part* v and its *point of application* p. A tangent vector v_p to $\mathbb{R}^3$ at p is *tangent to* $\mathbb{M}$ *at* p if v is the velocity of some curve φ in $\mathbb{M}$. The set of all tangent vectors to $\mathbb{M}$ at p is the *tangent plane* $T_p(\mathbb{M})$, the best linear approximation to $\mathbb{M}$ at p. Figure 9 shows selected tangent planes and their respective normals on the torus.

Suppose we have a smooth map $h\colon \mathbb{M} \to \mathbb{R}$. The *derivative $v_p[h]$ of h with respect to v_p* is the common value of $(d/dt)(h \circ \gamma)(0)$, for all curves $\gamma \in \mathbb{M}$ with initial velocity v_p. The *differential* dh_p is a linear function on $T_p(\mathbb{M})$ such that $dh_p(v_p) = v_p[h]$, for all tangent vectors $v_p \in T_p(\mathbb{M})$. The differential converts vector fields into real-valued functions. A point p is *critical* for h if dh_p is the zero map. Otherwise, p is *regular*. Given local coordinates x, y on $\mathbb{M}$, the *Hessian* of h is

$$H(p) = \begin{bmatrix} \frac{\partial^2 h}{\partial x^2}(p) & \frac{\partial^2 h}{\partial y \partial x}(p) \\ \frac{\partial^2 h}{\partial x \partial y}(p) & \frac{\partial^2 h}{\partial y^2}(p) \end{bmatrix}, \tag{21}$$

in terms of the basis $(\partial/\partial x(p), \partial/\partial y(p))$ for $T_p(\mathbb{M})$. A critical point p is *nondegenerate* if the Hessian is nonsingular at p, i.e., $\det H(p) \neq 0$. Otherwise, it is *degenerate*. A map h is a *Morse function* if all its critical points are nondegenerate. By the *Morse Lemma*, it is possible to choose local coordinates so that the neighborhood of a nondegenerate critical point takes the form $h(x, y) = \pm x^2 \pm y^2$. The *index* of a nondegenerate critical point is the number of minuses in this parameterization, or more generally, the number of negative eigenvalues of the Hessian. For a d-dimensional manifold, the index ranges from 0 to d. In two dimensions, a critical point of index 0, 1, or 2 is called a *minimum*, *saddle*, or *maximum*, respectively, as shown in Figure 10. The *monkey saddle* in the figure is the neighborhood of a degenerate critical point, a surface with three ridges and valleys. By the Morse Lemma, it cannot be the neighborhood of a critical point of a Morse function. In higher dimensions, we still have maxima and minima, but more types of saddle points. In Figure 11, we define a Gaussian landscape on a 2-manifold with boundary (a closed disk). The landscape has 3 lakes (minima), 15 passes (saddles), and 13 peaks (maxima), as shown in the

Figure 9: Selected tangent planes, indicated with disks, and normals on a torus.

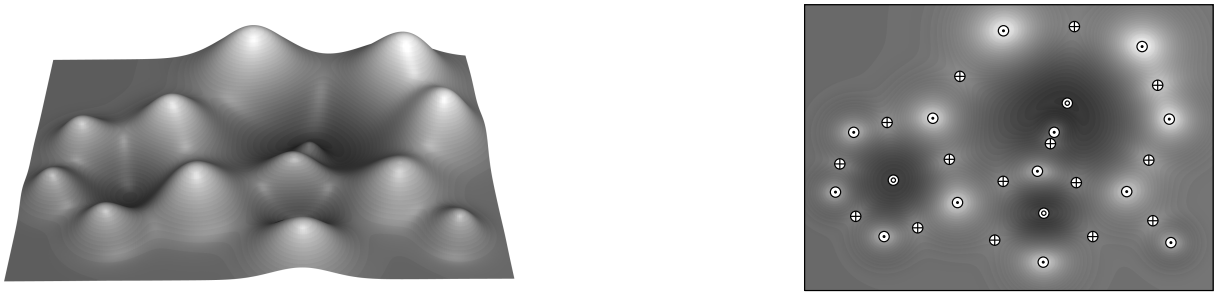
(a) Minimum, 0, $\circledcirc$ (b) Saddle, 1, $\oplus$ (c) Maximum, 2, $\odot$ (d) Monkey Saddle, $\circledast$

Figure 10: Neighborhoods of critical points, along with their indices, and mnemonic symbols. All except the Monkey saddle are nondegenerate.

orthographic view from above with our mnemonic symbols listed in Figure 10. To eliminate the boundary in our landscape, we place a point at $-\infty$ and identify the boundary to it, turning the disk into $\mathbb{S}^2$, and adding a minimum to our Morse function. This *one-point compactification* of the disk is done combinatorially in practice.

Relationship to homology. Given a manifold $\mathbb{M}$ and a Morse function $h\colon \mathbb{M} \to \mathbb{R}$, an *excursion set of h at height r* is $\mathbb{M}_r = \{m \in \mathbb{M} \mid h(m) \leq r\}$. We see three excursion sets for our landscape in Figure 12. Using this figure, we now consider topology of the excursion sets $\mathbb{M}_r$ as we increase r from $-\infty$:

- At a minimum, we get a new component, as demonstrated in leftmost excursion set. So, β_0 increases by one whenever r passes the height of a minimum point.
- At a saddle point, we have two possibilities. Two components may connect through a saddle point, as shown in the middle excursion set. Alternatively, the components may connect around the base of a maxima, forming a 1-cycle, as shown in the rightmost excursion set. So, either β_0 increases or β_1 decreases by one at a saddle point.
- At a maximum, a 1-cycle around that maximum is filled in, so β_1 is decremented.
- At regular points, there is no change to topology.



(a) Gaussian Landscape (b) Critical Points

Figure 11: A Gaussian landscape. A Morse function defined on a 2-manifold with boundary (a) has 3 minima, 15 saddles, and 13 peaks, as noted on the view from above (b). We get an additional minimum when we compactify the disk into a sphere.

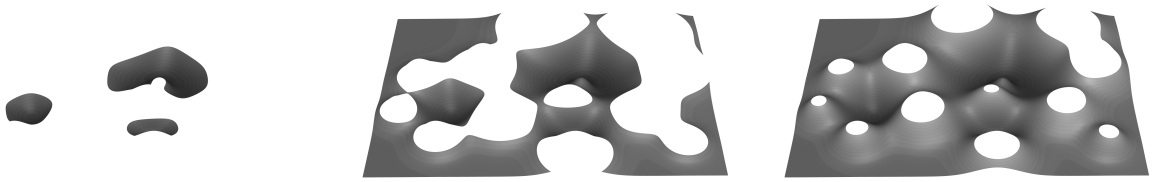


Figure 12: Excursion sets for the Gaussian landscape in Figure 11.

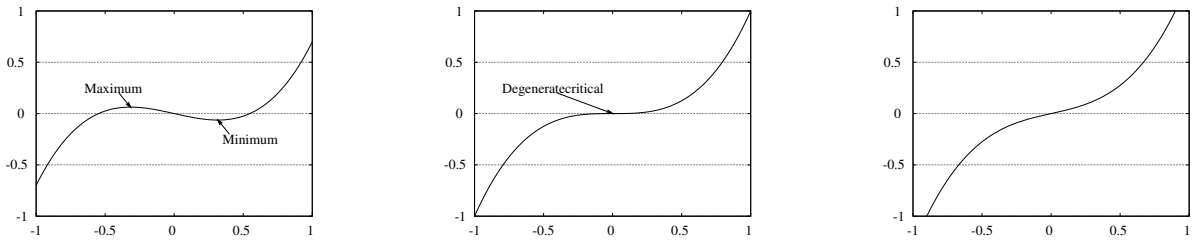
Therefore, minima, saddles, and maxima have the same impact on the Betti numbers as vertices, edges, and triangles. Given this relationship, we may extend the Euler-Poincaré formula (Equation (16) on Page 12):

$$\chi(\mathbb{M}) = \sum_k (-1)^k n_k = \sum_k (-1)^k \beta_k = \sum_k (-1)^k c_k, \tag{22}$$

where n_k is the number of k-cells for any cellular complex with underlying space $\mathbb{M}$, $\beta_k = \operatorname{rank} H_k(\mathbb{M})$, and c_k is the number of Morse critical points with index k for any Morse function define on $\mathbb{M}$. This beautiful equation relates combinatorial topology, algebraic topology, and Morse theory at once. For example, our compactified landscape is a sphere with $\chi(\mathbb{S}) = 2$. We also have $c_0 = 4$ (counting the minimum at $-\infty$), $c_1 = 15$, $c_2 = 13$, and $4 - 15 + 13 = 2$. That is, the topology of a manifold implies a relationship on the number of critical points of any function defined on that manifold.

Unfolding and cancellation. In practice, most of our functions are non-Morse, containing degenerate critical points, such as the function in Figure 13(b). Fortunately, we may perturb any function infinitesimally to be Morse as Morse functions are dense in the space of smooth functions. For instance, simple calculus shows that we may perturb the function in Figure 13(b) to either a function with two nondegenerate critical points (a) or zero critical points (c). Going from (b) to (a) is *unfolding* the degeneracy and its geometric study belongs to singularity theory [Bruce and Giblin, 1992]. Going from (a) to (c) describes the *cancellation* of a pair of critical points as motivated by excursion sets and Equation (22). Both concepts extend to higher-dimensional critical points.

The excursion sets of a Morse function form a filtration, as required by persistent homology in Section 5. The relationship between critical points and homology described in the last section implies a pairing of critical points. In one dimension, a minimum is paired with a maximum, as in Figure 13. In two dimensions,



(a) $y = x^3 - \epsilon x$, $\epsilon > 0$ (b) $y = x^3$ (c) $y = x^3 + \epsilon x$, $\epsilon > 0$

Figure 13: The function $y = x^3$ (b) has a single degenerate critical point at the origin and is not Morse. We may unfold the degenerate critical point into a maximum and a minimum (a), two nondegenerate critical points. We may also smooth it into a regular point (c). Either perturbation yields a Morse function arbitrarily close to (b). The process of going from (a) to (c) is cancellation.

a minimum is paired with a saddle, and a saddle with a maximum. This pairing dictates an ordering for *combinatorial* cancellation of critical points and identification of the significant critical points in the landscape. To remove the canceled critical points *geometrically* as in Figure 13(c), we need additional structure in higher dimensions, as described in Section 6.3.

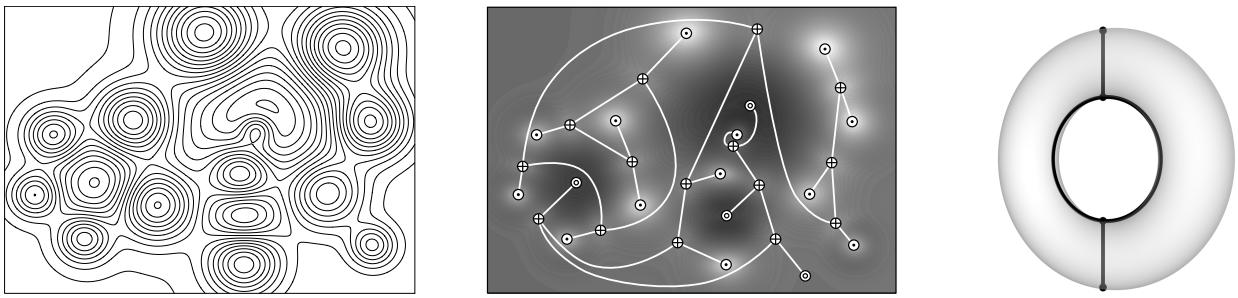
Non-smooth manifolds. While Morse theory is defined only for smooth manifolds, most concepts are not intrinsically dependent upon smoothness. Indeed, the main theorems have been shown to have analogues within discrete settings. Banchoff [1970] extends critical point theory to polyhedral surfaces. Forman [1998] defines discrete Morse functions on cell complexes to create a discrete Morse theory.

6.2 Reeb Graph and Contour Tree

Given a manifold $\mathbb{M}$ and a Morse function $h\colon \mathbb{M} \to \mathbb{R}$, a *level set of h at height r* is the preimage $h^{-1}(r)$. Note that level sets are simply the boundaries of excursion sets. Level sets are called *isolines* and *isosurfaces* for two- and three-manifolds, respectively. A *contour* is a connected component of a level set. For 2-manifolds, a contour is a loop and for 3-manifolds, it is a void. In Figure 14(a), we see the contours corresponding to twenty level sets of our landscape. As we change the level, contours appear, join, split, and disappear. To track the topology of the contours, we contract each contour to a point, obtaining the *Reeb graph* [Reeb, 1946]. When the Reeb graph is a tree, it is known as the *contour tree* [Boyell and Ruston, 1963]. In Figure 14(b), we superimpose the contour tree of our landscape over it. In Figure 14(c), we see the Reeb graph for a torus standing on its end, where the z coordinate is the Morse function.

Computation. We often have three-dimensional Morse functions, sampled on a regular voxelized grid, such as images from CT scans or MRI. We may linearly interpolate between the voxels for a piece-wise linear approximation of the underlying function. Lorensen and Cline [1987] first presented the *Marching cubes* algorithm for extracting isosurfaces using a table based approach that exploited the symmetries of a cube. The algorithm has evolved considerably since then [Newman and Hong, 2006].

Carr et al. [2003] give a simple $O(n \log n + N\alpha(N))$ time algorithm for computing the contour tree of a discrete real-valued field interpolated over a simplicial complex with n vertices and N simplices. Not surprisingly, the algorithm first captures the joining and splitting of contours by using the union-find data structure, described in Section 4.4. Cole-McLaughlin et al. [2004] give an $O(n \log n)$ algorithm for com-



(a) Contours (b) Contour tree (c) Reeb graph

Figure 14: Topology of contours. (a) Contours from 20 level sets for the landscape in Figure 11. (b) The contour tree for the landscape, superimposed on the landscape. The bottommost critical point is the minimum at $-\infty$. (c) The Reeb graph for the torus with Morse function equivalent to its z coordinate. The function has one minimum, two saddles, and one maximum.

puting Reeb graphs for triangulated 2-manifolds, with or without boundary, where n is the number of edges in the triangulation.

6.3 Morse-Smale Complex

The critical points of a Morse function are locations on the 2-manifold where the function is stationary. To extract more structure, we look what happens on regular points. A *vector field* or *flow* is a function that assigns a vector $v_p \in T_p(\mathbb{M})$ to each point $p \in \mathbb{M}$. A particular vector field is the *gradient* ∇h which is define by

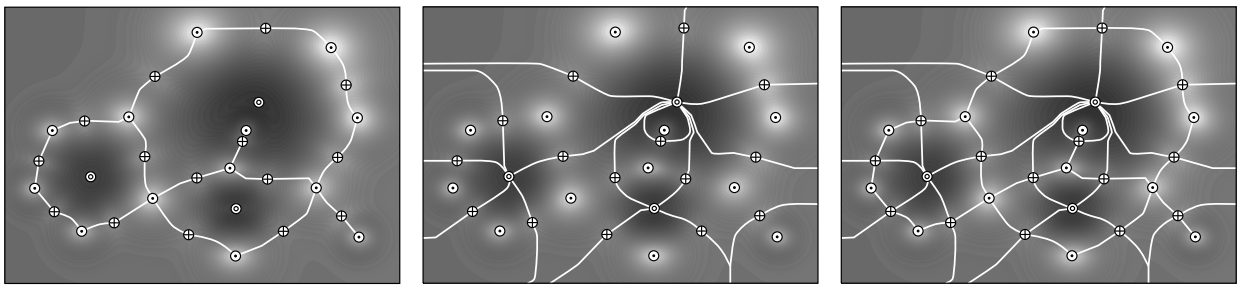
$$\langle \frac{d\gamma}{dt} \cdot \nabla h \rangle = \frac{d(h \circ \gamma)}{dt}, \tag{23}$$

where γ is any curve passing through $p \in \mathbb{M}$, tangent to $v_p \in T_p(\mathbb{M})$. The gradient is related to the derivative, as $v_p[h] = v_p \cdot \nabla h(p)$. It is always possible to choose coordinates (x, y) so that the tangent vectors $\frac{\partial}{\partial x}(p)$, $\frac{\partial}{\partial y}(p)$ are orthonormal with respect to the chosen metric. For such coordinates, the gradient is given by the familiar formula $\nabla h = (\frac{\partial h}{\partial x}(p), \frac{\partial h}{\partial y}(p))$. We integrate the gradient to decompose $\mathbb{M}$ into regions of uniform flow. An *integral line* $\ell : \mathbb{R} \to \mathbb{M}$ is a maximal path whose tangent vectors agree with the gradient, that is, $\frac{\partial}{\partial s}\ell(s) = \nabla h(\ell(s))$ for all $s \in \mathbb{R}$. We call $\operatorname{org} \ell = \lim_{s \to -\infty} \ell(s)$ the *origin* and $\operatorname{dest} \ell = \lim_{s \to +\infty} \ell(s)$ the *destination* of the path ℓ. The integral lines are either disjoint or the same, cover $\mathbb{M}$, and their limits are critical points, which are integral lines by themselves. For a critical point p, the *unstable manifold* $U(p)$ and the *stable manifold* $S(p)$ are:

$$U(p) = \{p\} \cup \{y \in \mathbb{M} \mid y \in \operatorname{im} \ell, \operatorname{org} \ell = p\}, \tag{24}$$

$$S(p) = \{p\} \cup \{y \in \mathbb{M} \mid y \in \operatorname{im} \ell, \operatorname{dest} \ell = p\}. \tag{25}$$

If p has index i, its unstable manifold is an open $(2-i)$-cell and its stable manifold is an open i-cell. For example, the minima on our landscape have open disks as unstable manifolds as shown in Figure 15(a). In *geographic information systems*, a minimum's unstable manifold is called its *watershed* as all the rain in this region flows toward this minimum. The stable manifolds of a Morse function h, shown in Figure 15(b) are the unstable manifolds of $-h$ as ∇ is linear. If the stable and unstable manifolds intersect only transversally, as in we say that our Morse function h is *Morse-Smale*. We then intersect the manifolds to obtain the *Morse-Smale Complex*, as shown in Figure 15(c). In two-dimensions, the complex is a quadrangulation, where each quadrangle has one maximum, two saddle points, and one minimum as vertices, and uniform flow from its maximum to its minimum. In three-dimensions, the complex is composed of hexahedra.



(a) Unstable 1-manifolds (b) Stable 1-manifolds (c) Morse-Smale Complex

Figure 15: The Morse-Smale Complex. The unstable manifolds (a) and stable manifolds (b) each decompose the underlying manifolds into a cell-complex. If the function is Morse-Smale, the stable and unstable manifolds intersect transversally to form the Morse-Smale Complex (c).

Cayley [1859] was the first to conceive of this structure in one of the earliest contributions to Morse theory. Independently, Maxwell [1870] gave full details of the complex as well as relationship between the number of Morse critical points (Equation (22)), but gave Cayley priority generously upon learning about the earlier contribution.

Computation. Using Banchoff's critical point theory, Edelsbrunner et al. [2003b] extend Morse-Smale complexes to piece-wise linear 2-manifolds, giving an algorithm for computing the complex as well as simplifying it through persistent homology. The complex has also been extended to 3-manifolds [Edelsbrunner et al., 2003a], although its computation remains challenging [Gyulassy et al., 2007].

7 Structures for Point Sets

A principle problem within computational topology is recovering the topology of a finite point set. The assumption is that the point set, either acquired or simulated, is sampled from some underlying topological space, whose connectivity is lost during the sampling process. Also, one often assumes that the space is a manifold. In this section, we look at techniques for computing structures that topologically approximate the underlying space of a given point set.

7.1 A Cover and its Nerve

We assume we are given a point set X embedded in $\mathbb{R}^d$, although the ambient space could also be any Riemannian manifold. Since the point set does not have any interesting topology by itself, we begin by approximating the underlying space by pieces of the embedding space. An *open cover* of X is $\mathcal{U} = \{U_i\}_{i \in I}$, $U_i \subseteq \mathbb{R}^d$, where I is an indexing set and $X \subseteq \cup_i U_i$. The *nerve* N of $\mathcal{U}$ is

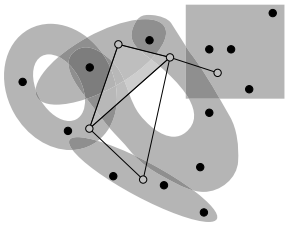
1. $\emptyset \in N$, and
2. If $\cap_{j \in J} U_j \neq \emptyset$ for $J \subseteq I$, then $J \in N$.

Clearly, the Nerve is an abstract simplicial complex. Figure 16(a) displays a point set, an open cover of the points, and the nerve of the cover. We say $\mathcal{U}$ is a *good* cover if all U_i are contractible and so are all of their nonempty finite intersections. The cover in Figure 16(a) is not good, as the leftmost set is homotopy equivalent to a circle. According to Leray's classical *Nerve Lemma*, the nerve of a good cover is homotopy equivalent to the cover. Rotman [1988, Lemma 7.26] shows this result for homology groups and Bott and Tu [1982, Theorem 13.4] prove it for the fundamental group, but given some technical conditions, one gets a homotopy equivalence. This lemma is the basis of most methods for representing point sets. We search for good covers whose nerve will be our representation.

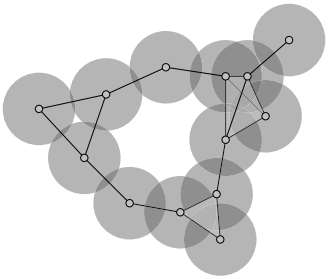
7.2 Abstract Complexes

Our first cover is the union of ϵ-balls centered around the input points, as shown in Figure 16(b). The cover is clearly good. More generally, any cover consisting of geodesically convex neighborhoods will be good. The implicit assumption here is that each ball approximates the space *locally* whenever the sampling is dense enough. The nerve of this cover is called the *Čech complex*. The complex may have dimension higher than the original embedding space and be as large as the power-set of the original point set in the worst case. In our example, the point set is in $\mathbb{R}^2$, but the complex contains a tetrahedron.

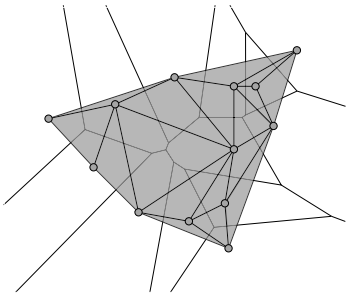
The *Vietoris-Rips complex* relaxes the Čech condition for simplex inclusion by allowing a simplex provided its vertices are pairwise within distance ϵ [Gromov, 1987]. We first compute the 1-skeleton of the complex, a graph on the vertices. We then perform a series of *expansions* by dimension, where we add a



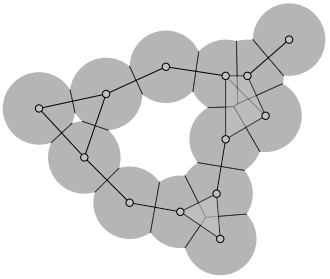
(a) Points, open cover, and nerve



(b) Cover: Union of ϵ-balls, Nerve: Čech Complex



(c) Cover: Voronoi diagram, Nerve: Delaunay complex



(d) Cover: Restricted Voronoi diagram, Nerve: Alpha complex

Figure 16: Representing point sets. (a) Given a point set (black points), we represent it using the nerve of an open cover. (b) The Čech complex is the nerve of a union of ϵ-balls. (c) The Delaunay complex is the nerve of the Voronoi diagram. (d) The Alpha complex is the nerve of the restricted Voronoi cells.

simplex provided all its faces are in the complex. This expansion fills in the triangular hole in the Čech complex in Figure 16(b), so the complex is not homotopy equivalent to the Čech complex.

7.3 Geometric Complexes

We next look at three methods that use techniques from *computational geometry* to compute complexes that are smaller than abstract complexes, but at the cost of intricate geometric computation that often do not extend to higher dimensions.

For $p \in X$, the *Voronoi cell* is the set of points in the ambient space closest to p, $V(p) = \{x \in \mathbb{R}^d \mid \mathrm{d}(x,p) \leq \mathrm{d}(x,y),\ \forall p \in X\}$. The *Voronoi diagram* decomposes $\mathbb{R}^d$ into Voronoi cells and is a good cover for $\mathbb{R}^d$ as all the cells are convex. The *Delaunay complex* is the nerve of the Voronoi diagram, as shown in Figure 16(c) [de Berg et al., 2000]. The *restricted Voronoi diagram* is the intersection of the the union of ϵ-balls and the Voronoi diagram, and the *alpha complex*, shown in Figure 16(d), is the nerve of this cover [Edelsbrunner and Mücke, 1994]. This complex is homotopy equivalent to the Čech complex, but is also embedded and has the same dimension as the ambient space. By construction, the alpha complex is always a subcomplex of the Delaunay complex, so we may compute the former by computing the latter. A deficiency of the methods that use ϵ-balls is that they assume uniform sampling for the point set, which is rarely true in practice. Recently, Cazals et al. [2006] address this deficiency of the alpha complex by defining the *conformal alpha complex* which features a global scale parameter.

Giesen and John [2003] define a distance function based on the point set and define the *flow complex* to be the stable manifolds of this function (See Section 6.3 for definitions.) Like the alpha complex, the flow complex is homotopy equivalent to the Čech complex [Dey et al., 2003]. de Silva and Carlsson [2004] define the *witness complex* using a relaxation of the Delaunay test. The complex is built on *landmarks*, a subset of

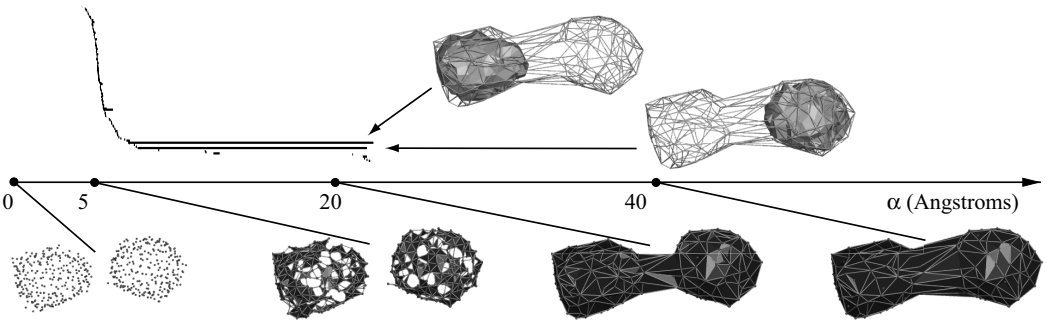


Figure 17: Application to biophysics. Below the α-axis are the point set ($\alpha = 0$) and three complexes from its alpha complex filtration. Above is the persistence barcode: a multiset of 146 intervals. Most intervals are very short, but there are two long intervals corresponding to the two visualized voids. The point set contains the phosphate group coordinates of the unfused inner membranes of a phospholipid molecule.

the point sets. The key idea is that the non-landmark points participate in the construction of the complex by acting as *witnesses* to the existence of simplices. While the witness complex utilizes additional geometry, it is not embedded like the alpha and flow complex, and only approximates the Čech complex. We usually build the 1-skeleton of the witness complex and perform Vietoris-Rips expansions for higher-dimensional skeletons.

Computing nearest neighbors. An essential task in computing complexes is enumerating exact or approximate nearest neighbors. This is a well-studied problem with a rich array of results. Geometric solutions [Arya et al., 1998] are efficient and feasible in low dimensions (less than 20) as they have an exponential dependence on dimension, the so-called *curse of dimensionality*. In higher dimensions, recent hashing techniques have resulted in practical algorithms [Andoni and Indyk, 2006]. While lower bounds exist, they often depend on the model of computation [Chakrabarti and Regev, 2004].

7.4 Using Persistent Homology

All the complexes described in the last two sections describe one-parameter families of spaces. In each case, the parameter is a notion of scale or local feature size in the space. For instance, the Čech complexes requires a radius ϵ for its cover. As we increase ϵ, we get filtered complexes as defined on Page 13. Therefore, we may apply persistent homology to capture the topology of the underlying point set. For instance, Kasson et al. [2007] use topological techniques for an application in *biophysics*. In Figure 17, we see four complexes from the alpha complex filtration of the point set at $\alpha = 0$, which describes the inner membrane of a bilayer vesicle, a primary cellular transport vehicle. We also see the β_2 persistence barcode above the axis, along with descriptions of the two significant voids that correspond to the two long intervals, both of which are computed with the persistence algorithm in Section 5.2. By analyzing more than 170,000 snapshots of fusion trajectories, the authors give a systematic topology-based method for measuring structural changes in membrane fusion.

8 Interactions with Geometry

As we noted in the introduction, computational topology was motivated initially by geometric problems that contained topological subproblems. Having discussed topological techniques, we end this chapter with a sample of geometric problems that involve topology.

8.1 Manifold Reconstruction

Given a finite set of (noisy) samples from a manifold, we would like to recover the original manifold. For 2-manifolds, Dey [2007] surveys results from *computational geometry* that guarantee a reconstruction *homeomorphic* to the original surface, provided the sampling satisfies certain local geometric conditions. Recently, Boissonnat et al. [2007] use witness complexes to reconstruct manifolds in arbitrary dimensions. There is also a new focus on finding homotopy equivalent spaces. Niyogi et al. [2008] give an algorithm to learn the underlying manifold with high confidence, provided the data is drawn from a sampling probability distribution that has support on or near a submanifold of a Euclidean space.

8.2 Geometric Descriptions

Topological noise often creates significant problems for subsequent geometry processing of surfaces, such as mesh *decimation*, *smoothing*, and *parameterization* for *texture mapping* or *remeshing*. To simplify the surface topologically, we need descriptions of topological attributes that also take geometry into account. A deciding factor is how one *measures* the attributes geometrically, as the complexity of the resulting optimal geometric problem is often dependent on the nature of the underlying geometric measure.

Non-canonical polygonal schema. Erickson and Har-Peled [2004] show that the problem of finding the *minimum* cut graph (Page 8) is NP-hard, when the geometric measure is either the total number of cut edges or their total lengths. They also describe a greedy algorithm that gives an $O(\log^2 g)$-approximation of the minimum cut graph in $O(g^2 n \log n)$ for a surface of genus g and combinatorial complexity n.

Homotopic paths in the plane. For a single given path of size k_{in} in the plane with n point obstacles, Hershberger and Snoeyink [1994] give an optimal $O(nk_{\text{in}})$ time algorithm for finding the shortest homotopic path, minimizing either the Euclidean length or number of segments. Efrat et al. [2006] present an output sensitive algorithm that runs in $O(n^{3/2} + k_{\text{in}} \log n + k_{\text{out}})$ time, where n is the number of obstacles, and k_{in} and k_{out} are the complexity of the input and output paths, respectively. Bespamyatnikh [2003] gives an $O(n \log^{1+\epsilon} n + k_{\text{in}} \log n + k_{\text{out}})$ time algorithm for simple paths and an $O(n^{2+\epsilon} + k \log^2 n)$ time algorithm for non-simple paths, where $\epsilon > 0$ is an arbitrary small constant. The constants hidden in the notation depend on ϵ.

Generators for the fundamental group. Colin de Verdière and Lazarus [2005] consider the special case of one-vertex cut graphs, sets of loops which are also a minimum presentations for the fundamental group. Given a set of loops, they give a polynomial-time algorithm that computes the shortest system of loops in the same homotopy class on an orientable surface, provided that the lengths of edges are uniform. For an orientable combinatorial surface with complexity n and a given basepoint, Erickson and Whittlesey [2005] show that a greedy algorithm can find the shortest set of loops that generate the fundamental group of the surface in $O(n \log n)$ using an application of Dijkstra's algorithm.

Generators for homology groups. Erickson and Whittlesey [2005] also give a greedy algorithm that computes the shortest set of cycles that generate the first homology group over $\mathbb{Z}_2$ in time $O(n^2 \log n + n^2 g + n g^3)$, where g is the genus of the surface and n is the complexity of the surface. Zomorodian and Carlsson [2008] consider the problem of *localization*: finding local homology groups for arbitrary spaces and dimensions. Rather than using a particular measure, they include geometry in the input as a cover. They then construct a larger space whose persistent homology localizes the homology generators with respect to the given cover. Freedman and Chen [2008] define the size of a homology class to be the radius of the smallest geodesic ball within the ambient space. Given this measure, they use matroid theory to prove that

an optimal basis may be computed by a greedy method in time $O(\beta^4 n^3 \log^2 n)$, where n is the size of the simplicial complex and β is the Betti number of the homology group.

9 Research Issues and Summary

Suppose we were given a million points in 100 dimension and we wish to recover the topology of the space from which these points were sampled. Currently, none of our tools either scale to this many points or extend to this high a dimension. Yet, we are currently inundated with massive data sets from acquisition devices and computer simulations. Computational topology could provide powerful tools for understanding the structure of this data. However, we need both theoretical results as well as practical algorithms tailored to massive data sets for computational topology to become successful as a general method for data analysis.

10 Defining Terms

Betti number In dimension k, the rank of the kth homology group, denoted β_k.

Contour A connected component of a level set $h^{-1}(c)$ of a Morse function $h\colon \mathbb{M} \to \mathbb{R}$ defined on a manifold $\mathbb{M}$.

Critical point A point on a manifold at which the differential of a Morse function is zero. Non-degenerate critical points on a 2-manifolds are minima, saddles, and maxima.

Euler characteristic An integer invariant that relates combinatorial topology to algebraic topology and Morse theory: $\chi(\mathbb{M}) = \sum_k (-1)^k n_k = \sum_k (-1)^k \beta_k = \sum_k (-1)^k c_k$, where n_k is the number of k-cells for any cellular complex with underlying space $\mathbb{M}$, β_k is the kth Betti number, and c_k is the number of Morse critical points with index k for any Morse function define on $\mathbb{M}$.

Fundamental group The group of homotopy classes of loops in a space.

Homeomorphism A 1-1 onto continuous map whose inverse is continuous.

Homology group An algebraic invariant $H_k = Z_k/B_k$, where Z_k is the group of k-cycles and B_k is the group of k-boundaries in the simplicial complex.

Homotopy A family of maps $f_t : \mathbb{X} \to \mathbb{Y},\ t \in [0,1]$, such that the associated map $F : \mathbb{X} \times [0,1] \to \mathbb{Y}$ given by $F(x,t) = f_t(x)$ is continuous.

Manifold A topological space that is locally Euclidean.

Morse function A function defined on a manifold whose critical points are non-degenerate.

Persistent homology The homology of a growing space, which captures the lifetimes of topological attributes in a multiset of intervals called a barcode.

Reeb graph A graph that captures the connectivity of contours. When it doesn't have cycles, it is called the contour tree.

Simplicial complex A collection of finite sets such that if a set is in the complex, so are all its subsets. It may be visualized as a union of convex hulls of finitely independent points, glued together along shared faces.

Topological space A point set with a topology, a collection of sets that define the open neighborhoods of the points.

References

S. I. Adyan. The algorithmic unsolvability of problems concerning recognition of certain properties of groups. *Doklady Academy Nauk SSSR*, 103:533–535, 1955.

A. Andoni. E^2LSH: Nearest neighbors in high dimensional spaces, 2006. `http://web.mit.edu/andoni/www/LSH/index.html`.

A. Andoni and P. Indyk. Near-optimal hashing algorithms for near neighbor problem in high dimensions. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 459–468, 2006.

T. M. Apostol. *Calculus. Volume II: Multi-variable calculus and linear algebra, with applications to differential equations and probability*. John Wiley & Sons, Inc., New York, NY, second edition, 1969.

S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *Journal of the ACM*, 45(6):891–923, 1998.

T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *The American Mathematical Monthly*, 77(5):475–485, 1970.

W. F. Basener. *Topology and Its Applications*. John Wiley & Sons, Inc., Hoboken, NJ, 2006.

S. Bespamyatnikh. Computing homotopic shortest paths in the plane. *Journal of Algorithms*, 49:284–303, 2003.

J.-D. Boissonnat, L. J. Guibas, and S. Y. Oudot. Manifold reconstruction in arbitrary dimensions using witness complexes. In *Proc. ACM Symposium on Computational Geometry*, pages 194–203, 2007.

W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, San Diego, CA, second edition, 1986.

R. Bott and L. W. Tu. *Differential Forms in Algebraic Topology*. Springer-Verlag, New York, NY, 1982.

R. L. Boyell and H. Ruston. Hybrid techniques for real-time radar simulation. In *Proc. IEEE Fall Joint Computer Conference*, pages 445–458, 1963.

T. Brahana. Systems of circuits on 2-dimensional manifolds. *Annals of Mathematics*, 23:144–168, 1921.

J. W. Bruce and P. J. Giblin. *Curves and singularities: a geometrical introduction to singularity theory*. Cambridge University Press, New York, NY, second edition, 1992.

S. Cabello, Y. Liu, A. Mantler, and J. Snoeyink. Testing homotopy for paths in the plane. *Discrete & Computational Geometry*, 31(1):61–81, 2004.

H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry: Theory and Applications*, 24:75–94, 2003.

A. Cayley. On contours and slope lines. *Philosophical Magazine*, XVIII:264–268, 1859.

F. Cazals, J. Giesen, M. Pauly, and A. Zomorodian. The conformal alpha shape filtration. *The Visual Computer*, 22(8):531–540, 2006.

CGAL. Computational Geometry Algorithms Library, 2008. `http://www.cgal.org`.

A. Chakrabarti and O. Regev. An optimal randomised cell probe lower bound for approximate nearest neighbour searching. In *Proc. IEEE Symposium on Foundations of Computer Science*, pages 473–482, 2004.

CHomP. Computational Homology Project, 2008. `http://chomp.rutgers.edu/`.

K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in Reeb graphs of 2-manifolds. *Discrete & Computational Geometry*, 32:231–244, 2004.

É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete & Computational Geometry*, 33(3):507–534, 2005.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, 2001.

M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, New York, second edition, 2000.

V. de Silva and G. Carlsson. Topological estimation using witness complexes. In *Proc. Symposium on Point-Based Graphics*, pages 157–166, 2004.

M. Dehn and P. Heegaard. Analysis situs. *Enzyklopädie der Mathematischen Wissenschaften*, IIAB3:153–220, 1907.

C. J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for Betti numbers of simplicial complexes on the 3-sphere. *Computer Aided Geometric Design*, 12:771–784, 1995.

T. K. Dey. *Curve and Surface Reconstruction*, volume 23 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, New York, NY, 2007.

T. K. Dey and S. Guha. Transforming curves on surfaces. *Journal of Computer and System Sciences*, 58(2): 297–325, 1999.

T. K. Dey and H. Schipper. A new technique to compute polygonal schema for 2-manifolds with application to null-homotopy detection. *Discrete & Computational Geometry*, 14(1):93–110, 1995.

T. K. Dey, J. Giesen, and M. John. Alpha-shapes and flow shapes are homotopy equivalent. In *Proc. ACM Symposium on Theory of Computing*, pages 493–502, 2003.

J.-G. Dumas, F. Heckenbach, B. D. Saunders, and V. Welker. Computing simplicial homology based on efficient Smith normal form algorithms. In *Algebra, Geometry, and Software Systems*, pages 177–207, 2003.

D. Dummit and R. Foote. *Abstract Algebra*. John Wiley & Sons, Inc., New York, NY, third edition, 2003.

H. Edelsbrunner. Biological applications of computational topology. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 32, pages 1395–1412. Chapman & Hall/CRC, Boca Raton, FL, 2004.

H. Edelsbrunner and E. P. Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13: 43–72, 1994.

H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28:511–533, 2002.

H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proc. ACM Symposium on Computational Geometry*, pages 361–370, 2003a.

H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry*, 30:87–107, 2003b.

A. Efrat, S. G. Kobourov, and A. Lubiw. Computing homotopic shortest paths efficiently. *Computational Geometry: Theory and Applications*, 35(3):162–172, 2006.

J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.

J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1046, 2005.

R. Forman. Morse theory for cell complexes. *Advances in Mathematics*, 134(1):90–145, 1998.

D. Freedman and C. Chen. Measuring and localizing homology, 2008. arXiv:0705.3061v2.

E. Gawrilow and M. Joswig. `polymake`, 2008. `http://www.math.tu-berlin.de/polymake/`.

R. Ghrist. Barcodes: the persistent topology of data. *American Mathematical Society Current Events Bulletin*, 2007.

J. Giesen and M. John. The flow complex: A data structure for geometric modeling. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 285–294, 2003.

M. Gromov. Hyperbolic groups. In S. Gersten, editor, *Essays in Group Theory*, pages 75–263. Springer-Verlag, New York, NY, 1987.

L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *ACM Transactions on Graphics*, 4:74–123, 1985.

A. Gyulassy, V. Natarajan, B. Hamann, and V. Pascucci. Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. In *Proc. IEEE Visualization*, pages 1440–1447, 2007.

A. Hatcher. *Algebraic Topology*. Cambridge University Press, New York, NY, 2002. `http://www.math.cornell.edu/~hatcher/AT/ATpage.html`.

M. Henle. *A Combinatorial Introduction to Topology*. Dover Publications, Inc., New York, 1997.

J. Hershberger and J. Snoeyink. Computing minimum length paths of a given homotopy class. *Computational Geometry: Theory and Applications*, 4(2):63–97, 1994.

M. Joswig. Computing invariants of simplicial manifolds, 2004. arXiv:math/0401176v1.

P. M. Kasson, A. Zomorodian, S. Park, N. Singhal, L. J. Guibas, and V. S. Pande. Persistent voids: a new structural metric for membrane fusion. *Bioinformatics*, 23(14):1753–1759, 2007.

F. Lazarus, G. Vegter, M. Pocchiola, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proc. ACM Symposium on Computational Geometry*, pages 80–89, 2001.

W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proc. SIGGRAPH*, pages 163–169, 1987.

A. A. Markov. Insolubility of the problem of homeomorphy. In *Proc. International Congress of Mathematics*, pages 14–21, 1958.

Y. Matsumoto. *An Introduction to Morse Theory*, volume 208 of *Iwanami Series in Modern Mathematics*. American Mathematical Society, Providence, RI, 2002.

J. C. Maxwell. On hills and dales. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 40(269):421–425, December 1870.

J. Milnor. The Poincaré conjecture, 2008. `http://www.claymath.org/millennium/Poincare_Conjecture/`.

J. Morgan and G. Tian. *Ricci Flow and the Poincaré Conjecture*, volume 3 of *Clay Mathematics Monographs*. American Mathematical Society and Clay Mathematics Institute, Cambridge, MA, 2007.

D. M. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching, 2006. `http://www.cs.umd.edu/~mount/ANN/`.

T. S. Newman and Y. Hong. A survey of the marching cubes algorithm. *Computers & Graphics*, 30(5): 854–879, 2006.

P. Niyogi, S. Smale, and S. Weinberger. Finding the homology of submanifolds with high confidence from random samples. *Discrete & Computational Geometry*, 39(1):419–441, 2008.

PLEX. Simplicial complexes in MATLAB, 2006. `http://math.stanford.edu/comptop/programs/`.

T. Rado. Über den begriff den Riemannschen fläschen. *Acta Szeged*, 2:101–121, 1925.

A. A. Ranicki, editor. *The Hauptvermutung Book*. Kluwer Academic Publishers, New York, NY, 1997.

G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Les Comptes rendus de l'Académie des sciences*, 222:847–849, 1946.

J. J. Rotman. *An Introduction to Algebraic Topology*. Springer-Verlag, New York, NY, 1988.

H. Seifert and W. Threlfall. *Lehrbuch der Topologie*. AMS Chelsea Publishing, Providence, RI, 2003.

A. Storjohann. Computing Hermite and Smith normal forms of triangular integer matrices. *Linear Algebra and Its Applications*, 282(1–3):25–45, 1998.

W. Thurston. Three-dimensional manifolds, Kleinian groups and hyperbolic geometry. *Bulletin of the American Mathematical Society (New Series)*, 6(3):357–381, 1982.

F. Uhlig. *Transform Linear Algebra*. Prentice Hall, Upper Saddle River, NJ, 2002.

G. Vegter. Computational topology. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 63, pages 719–742. Chapman & Hall/CRC, Boca Raton, FL, 2004.

G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proc. ACM Symposium on Computational Geometry*, pages 102–111, 1990.

A. Zomorodian. *Topology for Computing*, volume 16 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, New York, NY, 2005.

A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33 (2):249–274, 2005.

A. Zomorodian and G. Carlsson. Localized homology. *Computational Geometry: Theory and Applications*, 41:126–148, 2008.

Further Information

Surveys. The survey by Vegter [2004] contains results on knots, embeddings, and immersions not in this chapter. Edelsbrunner [2004] considers biological applications of computational topology. Joswig [2004] focuses on computation of invariants, including cohomology. Ghrist [2007] surveys recent results in persistent homology and its applications.

Books. Henle [1997] is an accessible introduction to topology. Hatcher [2002] is an excellent textbook on algebraic topology and is already in its 7th printing although it is available freely on the web. Basener [2006] covers topology and its applications. Matsumoto [2002] is a monograph on Morse theory. Zomorodian [2005] expands on persistent homology and Morse-Smale complexes.

Software. PLEX [2006] is a library of MATLAB routines that include construction of Čech, Vietoris-Rips and witness complexes, as well as computation of homology and persistent homology barcodes. CHomP [2008] computes homology of cubical complexes using reduction with tailored heuristics. CGAL [2008] is a computational geometry library that includes construction of alpha complexes. `polymake`'s `TOPAZ` application supports computing simplicial homology and cohomology modules, as well as more sophisticated invariants, such as cup products [Gawrilow and Joswig, 2008]. ANN [Mount and Arya, 2006] and E^{2}LSH [Andoni, 2006] are software packages for finding approximate nearest neighbors in high dimensions.