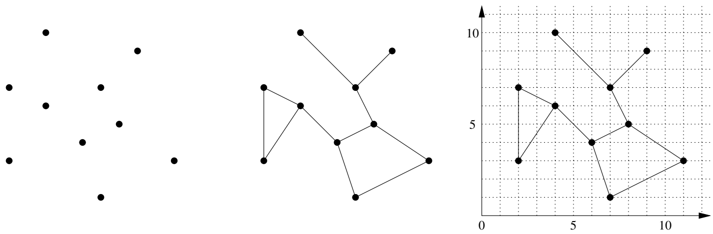
1

Introduction

The focus of this book is capturing and understanding the topological properties of spaces. To do so, we use methods derived from exploring the relationship between geometry and topology. In this chapter, I will motivate this approach by explaining what spaces are, how they arise in many fields of inquiry, and why we are interested in their properties. I will then introduce new theoretical methods for rigorously analyzing topologies of spaces. These methods are grounded in homology and Morse theory, and generalize to high-dimensional spaces. In addition, the methods are robust and fast, and therefore practical from a computational point of view. Having introduced the methods, I end this chapter by discussing the organization of the rest of the book.

1.1 Spaces

Let us begin with a discussion of spaces. A *space* is a set of points as shown in Figure 1.1(a). We cannot define what a *set* is, other than accepting it as a primitive notion. Intuitively, we think of a set as a collection or conglomeration of objects. In the case of a space, these objects are *points*, yet another primitive notion in mathematics. The concept of a space is too weak to be interesting, as it lacks structure. We make this notion slightly richer with the addition of a *topology*. We shall see in Chapter 2 what a topology formally means. Here, we think of a topology as the knowledge of the connectivity of a space: Each point in the space knows which points are near it, that is, in its *neighborhood*. In other words, we know how the space is connected. For example, in Figure 1.1(b), neighbor points are connected graphically by a path in the graph. We call such a space a *topological space*. At first blush, the concept of a topological space may seem contrived, as we are very comfortable with the richer *metric spaces*, as in Figure 1.1(c). We are introduced to the prototypical metric space, the *Euclidean space* $\mathbb{R}^d$, in secondary school, and we often envision our



(a) A space (b) A topological space (c) A metric space

Fig. 1.1. Spaces.

world as $\mathbb{R}^3$. A metric space has an associated *metric*, which enables us to measure distances between points in that space and, in turn, implicitly define their neighborhoods. Consequently, a metric provides a space with a topology, and a metric space is a topological one. Topological spaces feel alien to us because we are accustomed to having a metric. The spaces arise naturally, however, in many fields.

Example 1.1 (graphics) We often model a real-world object as a set of elements, where the elements are triangles, arbitrary polygons, or B-splines.

Example 1.2 (geography) Planetary landscapes are modeled as elevations over grids, or triangulations, in *geographic information systems*.

Example 1.3 (robotics) A robot must often plan a path in its world that contains many obstacles. We are interested in efficiently capturing and representing the *configuration space* in which a robot may travel.

Example 1.4 (biology) A protein is a single chain of amino acids, which folds into a globular structure. The *Thermodynamics Hypothesis* states that a protein always folds into a state of minimum energy. To predict protein structure, we would like to model the folding of a protein computationally. As such, the *protein folding* problem becomes an optimization problem: We are looking for a path to the global minimum in a very high-dimensional energy landscape.

All the spaces in the above examples are topological spaces. In fact, they are metric spaces that derive their topology from their metrics. However, the questions raised are often topological in nature, and we may solve them easier by focusing on the topology of the space, and not its geometry. I will refer to topological spaces simply as spaces from this point onward.

1.2 Shapes of Spaces

We have seen that spaces arise in the process of solving many problems. Consequently, we are interested in capturing and understanding the *shapes* of spaces. This understanding is really in the form of classifications: We would like to know how spaces agree and differ in shape in order to categorize them. To do so, we need to identify intrinsic properties of spaces. We can try transforming a space in some fixed way and observe the properties that do not change. We call these properties the *invariants* of the space. Felix Klein gave this famous definition for geometry in his *Erlanger Programm* address in 1872. For example, *Euclidean geometry* refers to the study of invariants under rigid motion in $\mathbb{R}^d$, e.g., moving a cube in space does not change its geometry. Topology, on the other hand, studies invariants under continuous, and continuously invertible, transformations. For example, we can mold and stretch a play-doh ball into a filled cube by such transformations, but not into a donut shape. Generally, we view and study geometric and topological properties separately.

1.2.1 Geometry

There are a variety of issues we may be concerned with regarding the geometry of a space. We usually have a finite representation of a space for computation. We could be interested in measuring the quality of our representation, trying to improve the representation via modifications, and analyzing the effect of our changes. Alternatively, we could attempt to reduce the size of the representation in order to make computations viable, without sacrificing the geometric accuracy of the space.

Example 1.5 (decimation) The Stanford Dragon in Figure 1.2(a) consists of 871,414 triangles. Large meshes may not be appropriate for many applications involving real-time rendering. Having *decimated* the surface to 5% of its original size (b), I show that the new surface approximates the original surface quite well (c). The maximum distance between the new vertices and the original surface is 0.08% of the length of the diagonal of the dragon's bounding box.

(a) Stanford Dragon, represented by a triangulated surface

(b) Decimated to 5% of the number of triangles

(c) Normalized distance to original surface, in increasing intensity

Fig. 1.2. Geometric simplification.

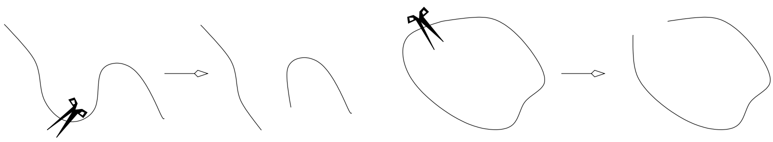


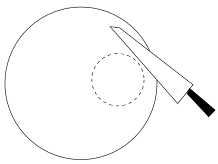
Fig. 1.3. The string on the left is cut into two pieces. The loop string on the right is cut but still is in one piece.

1.2.2 Topology

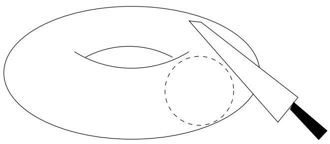
While Klein's unifying definition makes topology a form of geometry, we often differentiate between the two concepts. Recall that when we talk about topology, we are interested in how spaces are connected. Topology concerns itself with how things are connected, not how they look. Let's start with a few examples.

Example 1.6 (loops of string) Imagine we are given two pieces of strings. We tie the ends of one of them, so it forms a loop. Are they connected the same way, or differently? One way to find out is to cut both, as shown in Figure 1.3. When we cut each string, we are obviously changing its connectivity. Since the result is different, they must have been connected differently to begin with.

Example 1.7 (sphere and torus) Suppose you have a hollow ball (a sphere) and the surface of a donut (a torus.) When you cut the sphere anywhere, you get two pieces: the cap and the sphere with a hole, as shown in Figure 1.4(a). But there are ways you can cut the torus so that you only get one



(a) No matter where we cut the sphere, we get two pieces



(b) If we're careful, we can cut the torus and still leave it in one piece.

Fig. 1.4. Two pieces or one piece?

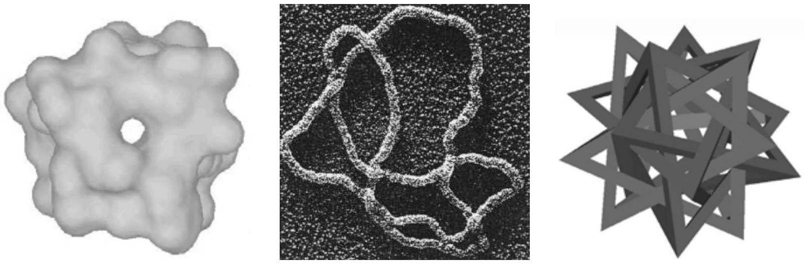
piece. Somehow, the torus is acting like our string loop and the sphere like the untied string.

Example 1.8 (holding hands) Imagine you're walking down a crowded street, holding somebody's hand. When you reach a telephone pole and have to walk on opposite sides of the pole, you let go of the other person's hand. Why?

Let's look back to the first example. Before we cut the string, the two points near the cut are near each other. We say that they are *neighbors* or in each other's *neighborhoods*. After the cut, the two points are no longer neighbors, and their neighborhood has changed. This is the critical difference between the untied string and the loop: The former has two ends. All the points in the loop have two neighbors, one to their left and one to their right. But the untied string has two points, each of whom has a single neighbor. This is why the two strings have different connectivity. Note that this connectivity does not change if we deform or stretch the strings (as if they are made of rubber.) As long as we don't cut them, the connectivity remains the same. Topology studies this connectivity, a property that is *intrinsic* to the space itself.

In addition to studying the *intrinsic* properties of a space, topology is concerned not only with how an object is connected (intrinsic topology), but how it is *placed* within another space (extrinsic topology.) For example, suppose we put a knot on a string and then tie its ends together. Clearly, the string has the same connectivity as the loop we saw in Example 1.6. But no matter how we move the string around, we cannot get rid of the knot (in topology terms, we cannot unknot the knot into the *unknot*.) Or can we? Can we prove that we cannot?

So, topological properties include having tunnels, as shown in Figure 1.5(a), being knotted (b), and having components that are linked (c) and cannot be taken apart. We seek computational methods to detect these properties. Topo-



(a) Gramicidin A, a protein, with a tunnel

(b) A knotted DNA

(c) Five pairwise-linked tetrahedral skeletons

Fig. 1.5. Topological properties. (b) Reprinted with permission from S Wasserman et al., SCIENCE, 229:171–174 (1985). © 1985 AAAS.

(a) Sampled point set from a surface

(b) Recovered topology

(c) Piece-wise linear surface approximation

Fig. 1.6. Surface reconstruction.

logical questions arise frequently in many areas of computation. Tools developed in topology, however, have not been used to address these problems traditionally.

Example 1.9 (surface reconstruction) Usually, a computer model is created by sampling the surface of an object and creating a point set, as in Figure 1.6(a). *Surface reconstruction*, a major area of research in computer graphics and computational geometry, refers to the recovery of the lost topology (b) and, in turn, geometry of a space. Once the connectivity is reestablished, the surface is often represented by a piece-wise linear approximation (c).

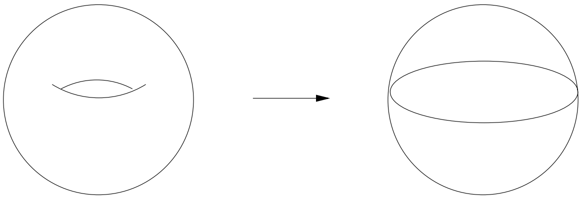


Fig. 1.7. Topological simplification.

As for geometry, we would also like to be able to simplify a space topologically, as in Figure 1.7. I have intentionally made the figures primitive compared to the previous geometric figures to reflect the essential structure that topology captures. To simplify topology, we need a measure of the importance of topological attributes. I provide one such measure in this book.

1.2.3 Relationship

The geometry and topology of a space are fundamentally related, as they are both properties of the same space. Geometric modifications, such as decimation in Example 1.5, could alter the topology. Is the simplified dragon in Figure 1.2(c) connected the same way as the original? In this case, the answer is yes, because the decimation algorithm excludes geometric modifications that have topological impact. We have changed the geometry of the surface without changing its topology.

When creating photo-realistic images, however, appearance is the dominant issue, and changes in topology may not matter. We could, therefore, allow for topological changes when simplifying the geometry. In other words, geometric modifications are possible with, and without, induced changes in topology. The reverse, however, is not true. We cannot eliminate the "hole" in the surface of the donut (torus) to get a sphere in Figure 1.7 without changing the geometry of the surface. We further examine the relationship between topology and geometry by looking at contours of terrains.

Example 1.10 (contours) In Figure 1.8, I show a flooded terrain with the water receding. The boundaries of the components that appear are the *iso-lines* or *contours* of the terrain. Contour lines are used often in map drawings. Noise in sampled data changes the geometry of a terrain, introducing small mountains and lakes. In turn, this influences how contour lines appear and merge as the water recedes.

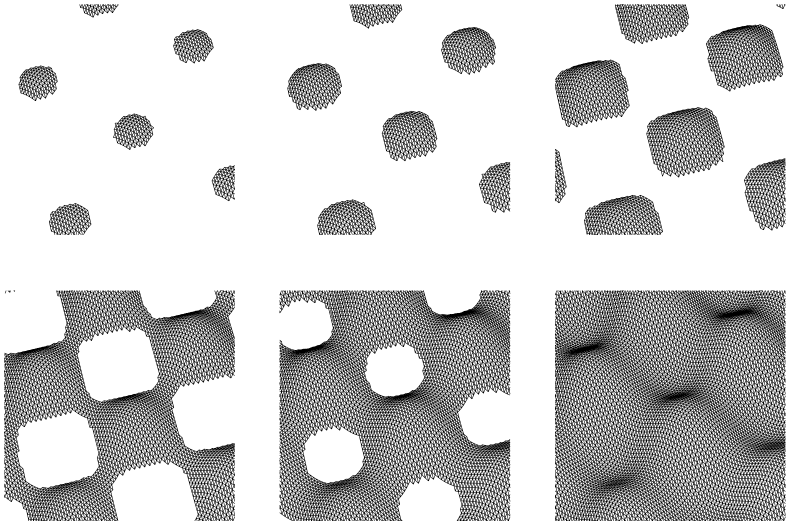


Fig. 1.8. Noah's flood receding.

We may view the spaces shown in Figure 1.8 as a single growing space undergoing topological and geometric changes. The history of such a space, called a *filtration*, is the primary object for this book. Note that the topology of the iso-lines within this history is determined by the geometry of the terrain. Generalizing to a $(d+1)$-dimensional surface, we see that there is a relationship between the topology of d-dimensional *level sets* of a space and its geometry, one dimension higher. This relationship is the subject of *Morse theory*, which we will encounter in this book.

1.3 New Results

We will also examine some new results in the area of computational topology. There are three main groups of theoretical results: persistence, Morse complexes, and the linking number.

Persistence. *Persistence* is a new measure for topological attributes. We call it persistence, as it ranks attributes by their life time in a filtration: their persistence in being a feature in the face of growth. Using this definition, we look at the following:

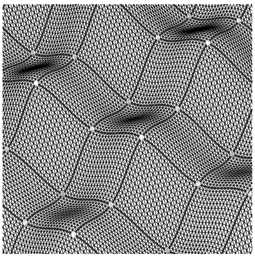


Fig. 1.9. A Morse complex over a terrain.

- **Persistence:** efficient algorithms for computing persistence over arbitrary coefficients.
- **Topological Simplification:** algorithms for simplifying topology, based on persistence. The algorithms remove attributes in the order of increasing persistence. At any moment, we call the removed attributes topological noise, and the remaining ones topological features.
- **Cycles and Manifolds:** algorithms for computing representations. The persistence algorithm tracks the subspaces that express nontrivial topological attributes, in order to compute persistence. We show how to modify this algorithm to identify these subspaces (cycles), as well as the subspaces that eliminate them (manifolds.)

Morse complexes. A *Morse complex* gives a full analysis of the behavior of flow over a space by partitioning the space into cells of uniform flow. In the case of a two-dimensional surface, such as the terrain in Figure 1.8, the Morse complex connects maxima (peaks) to minima (pits) through saddle points (passes) via edges, partitioning the terrain into quadrangles, as shown in Figure 1.9. Morse complexes are defined, however, only for smooth spaces. In this book, we will see how to extend this definition to piece-wise linear surfaces, which are frequently used for computation. In addition, we will learn how to construct hierarchies of Morse complexes.

- **Morse complex:** We give an algorithm for computing the Morse complex by first constructing a complex whose combinatorial form matches that of the Morse complex and then deriving the Morse complex via local transformations. This construction reflects a paradigm we call the *Simulation of Differentiability*.
- **Hierarchy:** We apply persistence to a filtration of the Morse complex to get a hierarchy of increasingly coarser Morse complexes. This corresponds to

modifying the geometry of the space in order to eliminate noise and simplify the topology of the contours of the surface.

Linking number. The *linking number* is an integer invariant that measures the separability of a pair of knots. We extend the definition of the linking number to simplicial complexes. We then develop data structures and algorithms for computing the linking numbers of the complexes in a filtration.

1.4 Organization

The rest of this book is divided into three parts: mathematics, algorithms, and applications. Part One, *Mathematics*, contains background on algebra, geometry, and topology, as well as the new theoretical contributions. In Chapter 2, we describe the spaces we are interested in exploring, and how we examine them by encoding their geometries in filtrations of complexes. Chapter 3 provides enough group theory background for the definition of homology groups in Chapter 4. We also discuss other measures of topology and justify our choice of homology. Switching to smooth manifolds, we review concepts from Morse Theory in Chapter 5. In Chapter 6, we give the mathematics behind the new results in this book.

Part Two, *Algorithms*, contains data structures and algorithms for the mathematics presented in Part I. In each chapter, we motivate and present algorithms and prove they are correct. In Chapter 7, we introduce algorithms for computing persistence: over $\mathbb{Z}_2$ coefficients, arbitrary fields, and arbitrary principal ideal domains. We then address topological simplification using persistence in Chapter 8. In Chapter 9, we describe an algorithm for computing two-dimensional Morse complexes. We end this part by showing how one may compute linking numbers in Chapter 10.

Part Three, *Applications*, contains issues relating to the application of the theory and algorithms presented in Parts I and II. To apply theoretical ideas to real-world problems, we need implementations and software, which we present in Chapter 11. We give empirical proof of the speed of the algorithms through experiments with our implementations in Chapter 12. We devote Chapter 13 to applications of the work in this book and future work.