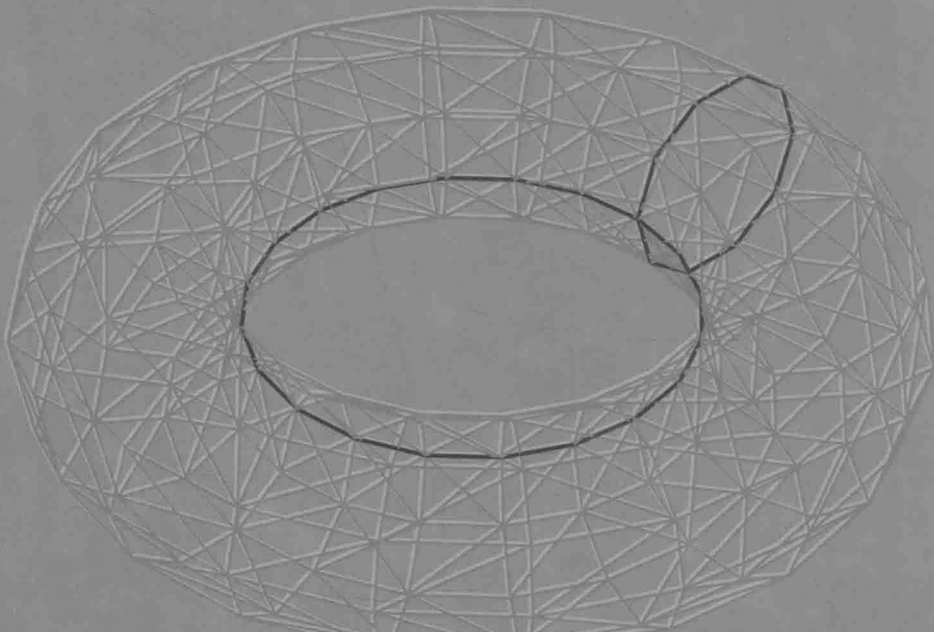
Cambridge Monographs on Applied and Computational Mathematics

Topology for Computing

Afra J. Zomorodian



Topology for Computing

AFRA J. ZOMORODIAN
Stanford University

CAMBRIDGE
UNIVERSITY PRESS

PUBLISHED BY THE PRESS SYNDICATE OF THE UNIVERSITY OF CAMBRIDGE
The Pitt Building, Trumpington Street, Cambridge, United Kingdom

CAMBRIDGE UNIVERSITY PRESS
The Edinburgh Building, Cambridge CB2 2RU, UK
40 West 20th Street, New York, NY 10011-4211, USA
477 Williamstown Road, Port Melbourne, VIC 3207, Australia
Ruiz de Alarcón 13, 28014 Madrid, Spain
Dock House, The Waterfront, Cape Town 8001, South Africa

http://www.cambridge.org

First published 2005

Printed in the United States of America

Typeface Times Roman 9/11 pt. *System* LaTeX 2_ε [AU]

A catalog record for this book is available from the British Library.

Library of Congress Cataloging in Publication Data

Zomorodian, Afra J., 1974–
Topology for computing / Afra J. Zomorodian.
p. cm. – (Cambridge monographs on applied and computational mathematics ; 16)
Includes bibliographical references and index.
ISBN 0-521-83666-2 (hardback)
1. Topology. I. Title. II. Series.
QA611.Z65 2004
514–dc22
2004047311

ISBN 0 521 83666 2 hardback

CAMBRIDGE MONOGRAPHS ON APPLIED AND COMPUTATIONAL MATHEMATICS

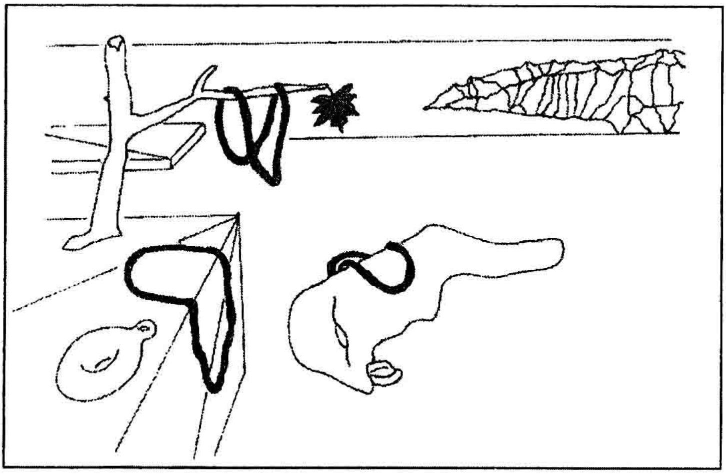
16 Topology for Computing

The *Cambridge Monographs on Applied and Computational Mathematics* reflects the crucial role of mathematical and computational techniques in contemporary science. The series publishes expositions on all aspects of applicable and numerical mathematics, with an emphasis on new developments in this fast-moving area of research.

State-of-the-art methods and algorithms as well as modern mathematical descriptions of physical and mechanical ideas are presented in a manner suited to graduate research students and professionals alike. Sound pedagogical presentation is a prerequisite. It is intended that books in the series will serve to inform a new generation of researchers.

Also in this series:

1. A Practical Guide to Pseudospectral Methods, *Bengt Fornberg*
2. Dynamical Systems and Numerical Analysis, *A. M. Stuart and A. R. Humphries*
3. Level Set Methods and Fast Marching Methods, *J. A. Sethian*
4. The Numerical Solution of Integral Equations of the Second Kind, *Kendall E. Atkinson*
5. Orthogonal Rational Functions, *Adhemar Bultheel, Pablo González-Vera, Erik Hendiksen, and Olav Njåstad*
6. The Theory of Composites, *Graeme W. Milton*
7. Geometry and Topology for Mesh Generation, *Herbert Edelsbrunner*
8. Schwarz–Christoffel Mapping, *Tofin A. Driscoll and Lloyd N. Trefethen*
9. High-Order Methods for Incompressible Fluid Flow, *M. O. Deville, P. F. Fischer, and E. H. Mund*
10. Practical Extrapolation Methods, *Avram Sidi*
11. Generalized Riemann Problems in Computational Fluid Dynamics, *Matania Ben-Artzi and Joseph Falcovitz*
12. Radial Basis Functions: Theory and Implementations, *Martin D. Buhmann*
13. Iterative Krylov Methods for Large Linear Systems, *Henk A. van der Vorst*
14. Simulating Hamiltonian Dynamics, *Ben Leimkuhler and Sebastian Reich*
15. Collocation Methods for Volterra Integral and Related Functional Equations, *Hermann Brunner*



— *Persistence of Homology* — Afra Zomorodian (After Salvador Dali)

TO MY PARENTS

On the left, a double-torus and a 1-cycle lie on a triangulated 2-manifold. There is a box-shaped cell-complex above. An unknot hangs from the large branch of the sapless withering tree. Through some exertion, the tree identifies itself as a maple by bearing a single green leaf. A deformed two-sphere, a torus, and a nonbounding loop form a pile in the center. Near the horizon, a 2-manifold is embedded by an associated height field. It divides itself into regions using the 1-cells of its Morse-Smale complex.

Preface

My goal in this book is to enable a non-specialist to grasp and participate in current research in computational topology. Therefore, this book is not a compilation of recent advances in the area. Rather, the book presents basic mathematical concepts from a computer scientist's point of view, focusing on computational challenges and introducing algorithms and data structures when appropriate. The book also incorporates several recent results from my doctoral dissertation and subsequent related results in computational topology.

The primary motivation for this book is the significance and utility of topological concepts in solving problems in computer science. These problems arise naturally in computational geometry, graphics, robotics, structural biology, and chemistry. Often, the questions themselves have been known and considered by topologists. Unfortunately, there are many barriers to interaction:

- Computer scientists do not know the language of topologists. Topology, unlike geometry, is not a required subject in high school mathematics and is almost never dealt with in undergraduate computer science. The axiomatic nature of topology further compounds the problem as it generates cryptic and esoteric terminology that makes the field unintelligible and inaccessible to non-topologists.
- Topology can be very unintuitive and enigmatic and therefore can appear very complicated and mystifying, often frightening away interested computer scientists.
- Topology is a large field with many branches. Computer scientists often require only simple concepts from each branch. While there are certainly a number of offerings in topology by mathematics departments, the focus of these courses is often theoretical, concerned with deep questions and existential results.

Because of the relative dearth of interaction between topologists and computer scientists, there are many opportunities for research. Many topological questions have large complexity: the best known bound, if any, may be exponential. For example, I once attended a talk on an algorithm that ran in quadruply exponential time! Let me make this clear. It was

$$O\left(2^{2^{2^{2^n}}}\right).$$

And one may overhear topologists boasting that their software can now handle 14 tetrahedra, not just 13. But better bounds may exist for specialized questions, such as problems in low dimensions, where our interests chiefly lie. We need better algorithms, parallel algorithms, approximation schemes, data structures, and software to solve these problems within our lifetime (or the lifetime of the universe.)

This book is based primarily on my dissertation, completed under the supervision of Herbert Edelsbrunner in 2001. Consequently, some chapters, such as those in Part Three, have a thesis feel to them. I have also incorporated notes from several graduate-level courses I have organized in the area: *Introduction to Computational Topology* at Stanford University, California, during Fall 2002 and Winter 2004; and *Topology for Computing* at the Max-Planck-Institut für Informatik, Saarbrücken, Germany, during Fall 2003.

The goal of this book is to make algorithmically minded individuals fluent in the language of topology. Currently, most researchers in computational topology have a mathematics background. My hope is to recruit more computer scientists into this emerging field.

Stanford, California A. J. Z.
June 2004

Acknowledgments

I am indebted to Persi Diaconis for the genesis of this book. He attended my very first talk in the Stanford Mathematics Department, asked for a copy of my thesis, and recommended it for publication. To have my work be recognized by such a brilliant and extraordinary figure is an enormous honor for me. I would like to thank Lauren Cowles for undertaking this project and coaching me throughout the editing process and Elise Oranges for copyediting the text.

During my time at Stanford, I have collaborated primarily with Leonidas Guibas and Gunnar Carlsson. Leo has been more than just a post-doctoral supervisor, but a colleague, a mentor, and a friend. He is a successful academic who balances research, teaching, and the mentoring of students. He guides a large animated research group that works on a manifold of significant problems. And his impressive academic progeny testify to his care for their success.

Eleven years after being a freshman in his "honors calculus," I am fortunate to have Gunnar as a colleague. Gunnar astounds me consistently with his knowledge, humility, generosity, and kindness. I continue to rely on his estimation, advice, and support.

I would also like to thank the members of Leo and Gunnar's research groups as well as the Stanford Graphics Laboratory, for inspired talks and invigorating discussions. This book was partially written during a four-month stay at the Max-Planck-Institut. I would like to thank Lutz Kettner and Kurt Mehlhorn for their sponsorship, as well as for coaxing me into teaching a mini-course.

Finally, I would like to thank my research collaborators, whose work appears in this book: Gunnar Carlsson, Anne Collins, Herbert Edelsbrunner, Leonidas Guibas, John Harer, and David Letscher. My research was supported, in part, by ARO under grant DAAG55-98-1-0177, by NSF under grants CCR-00-86013 and DMS-0138456, and by NSF/DARPA under grant CARGO 0138456.

Contents

	Preface	*page* xi
	Acknowledgments	xiii
1	**Introduction**	1
1.1	Spaces	1
1.2	Shapes of Spaces	3
1.3	New Results	8
1.4	Organization	10
	Part One: Mathematics	
2	**Spaces and Filtrations**	13
2.1	Topological Spaces	14
2.2	Manifolds	19
2.3	Simplicial Complexes	23
2.4	Alpha Shapes	32
2.5	Manifold Sweeps	37
3	**Group Theory**	41
3.1	Introduction to Groups	41
3.2	Characterizing Groups	47
3.3	Advanced Structures	53
4	**Homology**	60
4.1	Justification	60
4.2	Homology Groups	70
4.3	Arbitrary Coefficients	79
5	**Morse Theory**	83
5.1	Tangent Spaces	84
5.2	Derivatives and Morse Functions	85
5.3	Critical Points	86
5.4	Stable and Unstable Manifolds	88
5.5	Morse-Smale Complex	90

6 New Results 94
6.1 Persistence 95
6.2 Hierarchical Morse-Smale Complexes 105
6.3 Linking Number 116

Part Two: Algorithms

7 The Persistence Algorithms 125
7.1 Marking Algorithm 125
7.2 Algorithm for $\mathbb{Z}_2$ 128
7.3 Algorithm for Fields 136
7.4 Algorithm for PIDs 146

8 Topological Simplification 148
8.1 Motivation 148
8.2 Reordering Algorithms 150
8.3 Conflicts 153
8.4 Topology Maps 157

9 The Morse-Smale Complex Algorithm 161
9.1 Motivation 162
9.2 The Quasi Morse-Smale Complex Algorithm 162
9.3 Local Transformations 166
9.4 Algorithm 169

10 The Linking Number Algorithm 171
10.1 Motivation 171
10.2 Algorithm 172

Part Three: Applications

11 Software 183
11.1 Methodology 183
11.2 Organization 184
11.3 Development 186
11.4 Data Structures 190
11.5 *CView* 193

12 Experiments 198
12.1 Three-Dimensional Data 198
12.2 Algorithm for $\mathbb{Z}_2$ 204
12.3 Algorithm for Fields 208
12.4 Topological Simplification 215
12.5 The Morse-Smale Complex Algorithm 217
12.6 The Linking Number Algorithm 220

13 Applications 223
13.1 Computational Structural Biology 223
13.2 Hierarchical Clustering 227

13.3 Denoising Density Functions 229
13.4 Surface Reconstruction 231
13.5 Shape Description 232
13.6 I/O Efficient Algorithms 233

Bibliography 235
Index 240

Color plates follow page 154

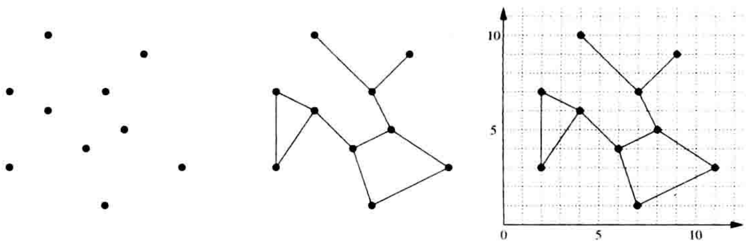
1

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.

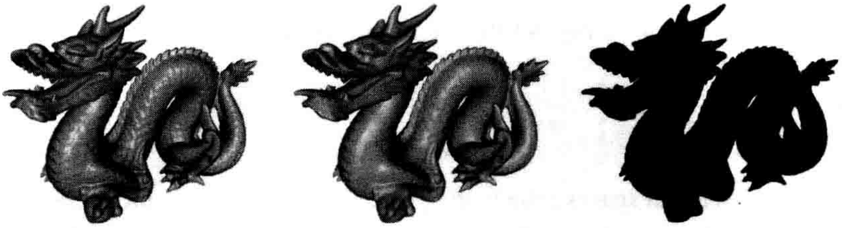
1.2 Shapes of Spaces

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.

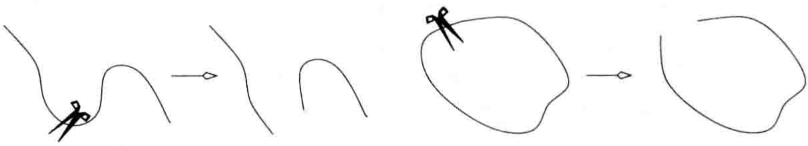


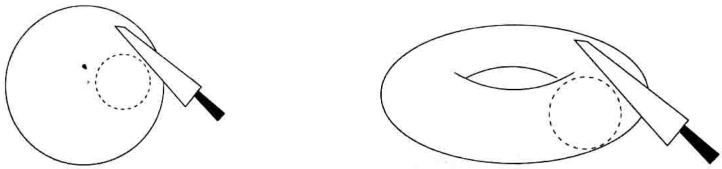
Fig. 1.3. The string on the left is cut into two pieces. The loop string on the right is cut but still is in one piece.

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-