Algebraic topology for computer vision

Daniel Freedman, Chao Chen

HP Laboratories

HPL-2009-375

Keyword(s):
algebraic topology, persistent homology, computer vision, image processing

Abstract:

Algebraic topology is generally considered one of the purest subfields of mathematics. However, over the last decade two interesting new lines of research have emerged, one focusing on algorithms for algebraic topology, and the other on applications of algebraic topology in engineering and science. Amongst the new areas in which the techniques have been applied are computer vision and image processing. In this paper, we survey the results of these endeavours. Because algebraic topology is an area of mathematics with which most computer vision practitioners have no experience, we review the machinery behind the theories of homology and persistent homology; our review emphasizes intuitive explanations. In terms of applications to computer vision, we focus on four illustrative problems: shape signatures, natural image statistics, image denoising, and segmentation. Our hope is that this review will stimulate interest on the part of computer vision researchers to both use and extend the tools of this new field.

External Posting Date: December 17, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: December 17, 2009 [Fulltext]

© Copyright 2009 Hewlett-Packard Development Company, L.P.

ALGEBRAIC TOPOLOGY FOR COMPUTER VISION

DANIEL FREEDMAN* AND CHAO CHEN†

Abstract. Algebraic topology is generally considered one of the purest subfields of mathematics. However, over the last decade two interesting new lines of research have emerged, one focusing on algorithms for algebraic topology, and the other on applications of algebraic topology in engineering and science. Amongst the new areas in which the techniques have been applied are computer vision and image processing. In this paper, we survey the results of these endeavours. Because algebraic topology is an area of mathematics with which most computer vision practitioners have no experience, we review the machinery behind the theories of homology and persistent homology; our review emphasizes intuitive explanations. In terms of applications to computer vision, we focus on four illustrative problems: shape signatures, natural image statistics, image denoising, and segmentation. Our hope is that this review will stimulate interest on the part of computer vision researchers to both use and extend the tools of this new field.

Key words. algebraic topology, persistent homology, computer vision, image processing

AMS subject classifications. 55U99, 68T45, 68U10, 65D18

1. Introduction.

Algebraic topology, with roots dating to Poincare at the turn of the twentieth century, has traditionally been considered one of the purest subfields of mathematics, with very few connections to applications. The last decade, however, has witnessed an explosion of interest in computational aspects of algebraic topology, and with the development of this computational machinery, a concomitant interest in applications. In this paper, we review some of these developments, and show how methods of computational algebraic topology may be fruitfully applied to problems of computer vision and image processing. We hope that this review will stimulate interest on the part of computer vision researchers to both use and extend these tools.

As we have noted, the new interest in algebraic topological tools has been fueled by two parallel developments: the design of new algebraic topological algorithms, and the application of these algorithms to various scientific and engineering fields. The new algorithms have focused almost exclusively on computations involving homology groups; without going into detail at this point (we defer a formal description of homology theory until Section 2), we may note that homology groups are topological invariants which, roughly speaking, count the number of holes of various dimensions that a topological space has. The great advantage conferred by homology groups is that these groups are relatively straightforward to compute with, as opposed to other topological concepts such as homotopy groups, or worse, homeomorphism equivalence classes. In particular, the concept of *persistent homology* as introduced by Edelsbrunner *et al.* [20], has proven to be very useful. The new framework has been applied to a number of fields, including molecular biology [16, 10], sensor networks [25], robotics [13], graphics [14], geometric modeling [17], machine learning [35], as well as computer vision and image processing.

Why use topological tools in computer vision? One answer is that, generally speaking, topological invariants tend to be very robust. If the topological space is

*Hewlett-Packard Laboratories, Haifa, Israel and the Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, 12180. (daniel.freedman@hp.com). Questions, comments, or corrections to this document may be directed to that email address.

†Department of Computer Science, Rensselaer Polytechnic Institute, Troy, NY, 12180. (chenc3@cs.rpi.edu).

stretched and deformed continuously, without any tearing or gluing, then the topological invariants of the space will be preserved. This is, of course, by design: topology is essentially the study of invariants of spaces under this group of transformations – continuous bijections with continuous inverses, also known as homeomorphisms. This property might be quite useful in the design of shape signatures, where the goal is to find a compact description of a shape which does not change much (if it all) when the shape undergoes some types of deformation.

In fact, though, topological invariants do suffer from some problems. On the one hand, such invariants are perhaps "too robust:" in the shape signature example, it is easy to see that quite different shape may share the same invariant. For example, the curve describing a hand-like shape and a circle are topologically equivalent, see Figure 1.1. Thus, the topological space often needs to be augmented by extra geometric information before the invariant becomes useful; we will see an example of this type of augmentation in Section 3.1. On the other hand, topological invariants can – strangely – be quite sensitive to noise. An example of this phenomenon is shown in Figure 1.1, in which a space (a torus), and the same space with a tiny handle attached, are shown. Clearly, the two objects are very similar, but due to the presence of the tiny handle, their topological invariants will not agree. Persistent homology is the tool that deals with this type of "topological noise," by examining not only the topological features of a space, but also their "lifetime," or significance.

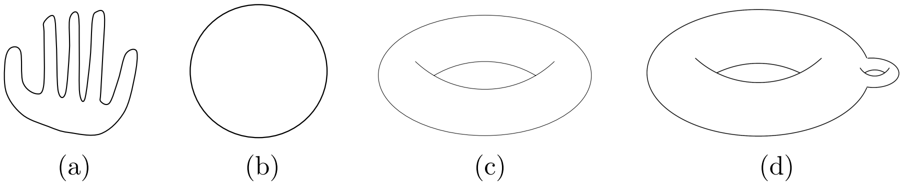


FIG. 1.1. *The problems with topological invariants. The hand-like shape in (a) and the circle in (b) are quite different shapes, but they possess the same topology. On the flip side, the ordinary torus in (c) is similar to the two-handled torus in (d), due to the fact that the second handle is very small (i.e. is "topological noise"). However, their topological descriptors will be different.*

The fields of applied and computational algebraic topology are quite new, with most of the key developments having taken place in the last decade. The applications of these ideas to computer vision and image processing are even newer, with most of the relevant work have appeared in the last five years. As a result, this paper is perhaps slightly different from a traditional survey or review of the literature. We focus on two goals: presenting the mathematical material – which can be quite daunting at first blush – in an accessible fashion, and reviewing some of the most interesting applications of this material to computer vision and image processing. The main goal of the paper is to stimulate interest in these new algebraic topological tools on the part of the computer vision community, so that the tools may be applied to new problems, and perhaps computationally and theoretically extended as well.

The remainder of the paper is organized as follows. In Section 2, we review the relevant material from algebraic topology, focusing in particular on homology theory and persistent homology. Although the material is presented in a way to make it as intuitive as possible, references to various standard texts and papers are given, to aid the reader interested in pursuing the topic further. Section 3 presents four interesting and illustrative applications of the topological techniques to computer vision and image processing. In Section 3.1, the problem of designing a shape signature is considered. Section 3.2 examines the problems of natural image statistics, and shows that the new topological techniques can contribute to a more accurate characterization of these statistics. Section 3.3 discusses the traditional problem of noise reduction, and simultaneously examines the problem of image segmentation. Persistent homology leads to a Mean Shift like algorithm, but one which has a rigorous way of merging segments. Finally, Section 4 concludes.

2. A Review of Persistent and Computational Homology. In this section, we provide the necessary background in algebraic topology, including a discussion of simplicial complexes, homology groups, and persistent homology. We will try to give an accessible introduction to the relevant notions, but given the space limitations our discussion will necessarily be somewhat brief. The interested reader is referred to [34, 27] for further details in general algebraic topology and [18, 24] for surveys of persistent homology.

2.1. Simplicial Complex. A *d-dimensional simplex* or *d-simplex*, σ, is the convex hull of $d+1$ *affinely independent vertices*, which means for any of these vertices, v_i, the d vectors $v_j - v_i$, $j \neq i$, are linearly independent. In other words, given a set of $(d+1)$ vertices such that no m-dimensional plane contains more than $(m+1)$ of them, a simplex is the set of points each of which is a linear combination of these vertices, with all coefficients nonnegative and summing to 1. A 0-simplex, 1-simplex, 2-simplex and 3-simplex are a vertex, edge, triangle and tetrahedron, respectively (Figure 2.1). The convex hull of a nonempty subset of vertices of σ is its *face*.

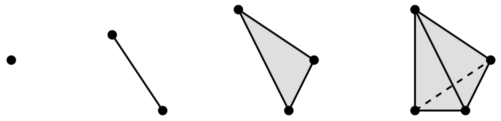


FIG. 2.1. *Simplices of dimension 0, 1, 2 and 3.*

A *simplicial complex* K is a finite set of simplices that satisfies the following two conditions.

1. Any face of a simplex in K is also in K.
2. The intersection of any two simplices in K is either empty or is a face for both of them.

The *dimension* of a simplicial complex is the highest dimension of its simplices. If a subset $K_0 \subseteq K$ is a simplicial complex, it is a *subcomplex* of K.

Please see Figure 2.2 for an example simplicial complex. The triangulation of the cube provides 3-dimensional simplices. Therefore the simplicial complex is 3-dimensional.

2.2. The Chain Group. In this paper, we only use simplicial homology of $\mathbb{Z}_2$ coefficients, which is introduced in this section. For completeness, in Section 2.5, we briefly discuss simplicial homology of other coefficient rings.

Within a given simplicial complex K, a *d-chain* is a linear combination of d-simplices in K, formally,

$$c = \sum_{\sigma \in K} a_\sigma \sigma, \quad a_\sigma \in \mathbb{Z}_2.$$

Note that since the field is $\mathbb{Z}_2$, the set of d-chains is in one-to-one correspondence with the set of subsets of d-simplices. A d-chain corresponds to a n_d-dimensional vector,

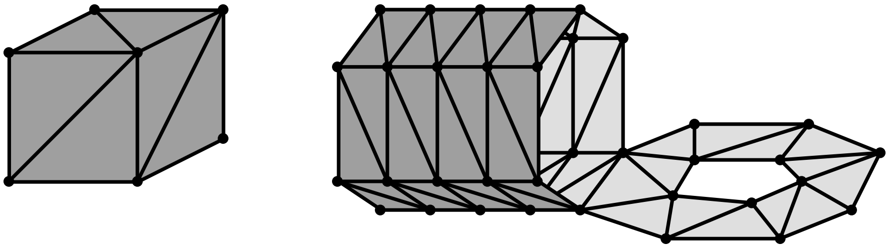


FIG. 2.2. *An example simplicial complex. It is the combination of the triangulation of a tube (open in both ends), an annulus and a cube. Note that the tube and the annulus share a common edge.*

whose nonzero entries correspond to the included d-simplices. Here n_d is the number of d-simplices in K. For example, in Figure 2.3, the red edges form a 1-chain and the dark grey triangles form a 2-chain.

If we define the addition of chains as the addition of these vectors, all the d-chains form the *group of d-chains*, $\mathsf{C}_d(K)$. Note that addition is using $\mathbb{Z}_2$ (i.e. mod 2) arithmetic.

2.3. The Cycle and Boundary Groups. The *boundary* of a d-chain is the mod 2 sum of the $(d-1)$-faces of all the d-simplices in the chain. For example, in Figure 2.3, the green edges form the boundary of the 2-chain formed by the three dark grey triangles. Two out of seven 1-dimensional faces (edges) of the triangles do not appear in the boundary due to the mod 2 addition. Similarly, tetrahedra of the cube form a 3-chain whose boundary is the triangles of the box bounding the cube.

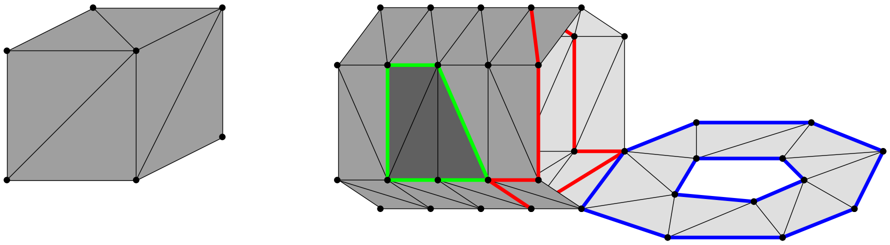


FIG. 2.3. *The green cycle is a boundary. The two blue cycles belong to a nontrivial class. The red cycle represents another nontrivial class.*

The boundary operator $\partial_d : \mathsf{C}_d(K) \to \mathsf{C}_{d-1}(K)$ is a group homomorphism, which means that the boundary of the sum of any two d-chains is equal to the sum of their boundaries, formally,

$$\partial_d(c_1 + c_2) = \partial_d(c_1) + \partial_d(c_2), \quad \forall c_1, c_2 \in \mathsf{C}_d(K).$$

A *d-cycle* is a d-chain with zero boundary. The set of d-cycles forms a subgroup of the chain group, which is the kernel of the boundary operator, $\mathsf{Z}_d(K) = \ker(\partial_d)$. The set of d-boundaries forms a group, which is the image of the boundary operator, $\mathsf{B}_d(K) = \mathrm{img}(\partial_{d+1})$. A d-cycle which is not a d-boundary, $z \in \mathsf{Z}_d(K) \backslash \mathsf{B}_d(K)$, is a *nonbounding cycle*. In Figure 2.3, both the green and red chains are 1-cycles. (The red chain goes around the interior of the tube, but some parts are necessarily occluded in the rendering.) But only the red chain is nonbounding. It is not hard to see that a d-boundary is also a d-cycle. Therefore, $\mathsf{B}_d(K)$ is a subgroup of $\mathsf{Z}_d(K)$.

In our case, the coefficients belong to a field, namely $\mathbb{Z}_2$; when this is the case, the groups of chains, boundaries and cycles are all vector spaces.[1] Computing the boundary of a d-chain corresponds to multiplying the chain vector with a boundary matrix $[b_1, ..., b_{n_d}]$, whose column vectors are boundaries of d-simplices in K. By slightly abusing notation, we call the boundary matrix ∂_d.

See Figure 2.4 for a simplicial complex whose boundary matrices are

$$\partial_1 = \left[\begin{array}{c|ccccccc} & ab & ac & ad & ae & bc & cd & ce \\ \hline a & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ b & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ c & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ d & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ e & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] \quad \text{and} \quad \partial_2 = \left[\begin{array}{c|cc} & acd & ace \\ \hline ab & 0 & 0 \\ ac & 1 & 1 \\ ad & 1 & 0 \\ ae & 0 & 1 \\ bc & 0 & 0 \\ cd & 1 & 0 \\ ce & 0 & 1 \end{array}\right].$$

This simplicial complex has 1 nontrivial homology class, represented by three different nonbounding cycles, $(ab + bc + cd + da)$, $(ab + bc + ca)$ and $(ab + bc + ce + ea)$, whose corresponding vectors are $(1, 0, 1, 0, 1, 1, 0)^T$, $(1, 1, 0, 0, 1, 0, 0)^T$ and $(1, 0, 0, 1, 1, 0, 1)^T$ respectively.

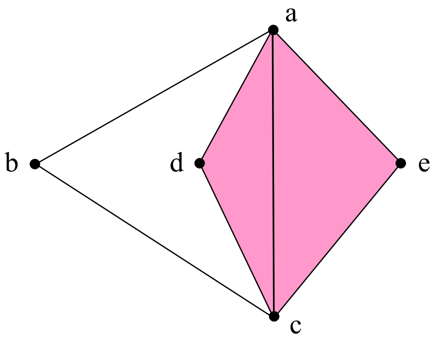


FIG. 2.4. *A simplicial complex K containing five 0-simplices, seven 1-simplices and two 2-simplices.*

2.4. The Homology Group. In algebraic topology, we want to capture all the nonbounding cycles, and more importantly, to classify them. We classify cycles into equivalence classes, each of which contains the set of cycles whose difference is a boundary. A *homology class* is the set of cycles

$$\{z \mid z = z_0 + \partial_{d+1} c,\ c \in \mathsf{C}_{d+1}(K)\},$$

for a fixed z_0. It is not hard to verify that this set is closed, that is, for any two elements of the same set, their sum also belongs to the set. This set, denoted as $[z_0] = z_0 + \mathsf{B}_d(K)$, is called a *coset*. Any cycle belonging to the class can be the *representative cycle*, z_0. When the representative cycle z_0 is a boundary, $[z_0] = 0 + \mathsf{B}_d(K)$ is the boundary group itself.

[1] Note that this is not true when the homology is over a ring which is not a field, such as $\mathbb{Z}$.

The set of equivalence classes, together with the boundary group, under addition defined by the addition of their representative cycles, forms a nice group structure. This group of equivalent classes is the quotient group $\mathsf{H}_d(K) = \mathsf{Z}_d(K)/\mathsf{B}_d(K)$, and is called the *d-dimensional homology group*. The boundary group $0 + \mathsf{B}_d(K)$ is the identity element of $\mathsf{H}_d(K)$. Otherwise, when z_0 is a nonbounding cycle, $[z_0]$ is a *nontrivial homology class* represented by z_0. Cycles of the same homology class are said to be *homologous* to each other, formally, $z_1 \sim z_2$.

In Figure 2.3, the two blue cycles are homologous to each other, but not to the red and green cycles. The 1-dimensional homology group has four different members, represented by the green cycle (corresponding to the boundary group), the red cycle, one of the blue cycles, and the sum of a red cycle and a blue cycle, respectively.

The dimension of the homology group is referred to as the *Betti number*,

$$\begin{aligned}\beta_d &= \dim(\mathsf{H}_d(K)) \\ &= \dim(\mathsf{Z}_d(K)) - \dim(\mathsf{B}_d(K)).\end{aligned}$$

By linear algebra, the Betti number can be computed by computing the ranks of all boundary matrices.

$$\beta_d = (n_d - \mathrm{rank}(\partial_d)) - \mathrm{rank}(\partial_{d+1}).$$

As the dimension of the chain group is upper bounded by the cardinality of K, n, so are the dimensions of $\mathsf{B}_d(K)$, $\mathsf{Z}_d(K)$ and $\mathsf{H}_d(K)$.

2.5. Extensions of Homology Theory. Whereas the simplicial homology studies a topological space by studying its triangulation, for a general topological space, we could use the *singular homology*. In singular homology, a simplex is defined as a continuous mapping (not necessarily injective) from the standard simplex to the topological space. The definition is extended to chains, boundary operations and singular homology groups. It can be proven that the simplicial homology of a simplicial complex is isomorphic to the singular homology of its geometric realization (the underlying space). This implies, in particular, that the simplicial homology of a space does not depend on the particular simplicial complex chosen for the space. In the figures in this paper, we may sometime ignore the simplicial complex and only show the continuous images of chains.

We restrict our discussion of simplicial homology to be over $\mathbb{Z}_2$ field. In general, the coefficients may belong to arbitrary abelian groups. In such cases, the group structure of the homology can be more complicated. See [34] for more details.

2.6. Computation of Homology. Through a sequence of row and column operations, we can transform the boundary matrices into the so-called Smith Normal Form. We do not explain the structure of the Smith Normal Form [30] here; instead, we simply note that in the example of the simplicial complex of Figure 2.4, the boundary matrices can be rewritten as

$$\partial_1' = \left[\begin{array}{c|ccccccc} & ab & ac & ad & ae & ab+ac+bc & ac+ae+ce & ac+ad+cd \\ \hline a+b & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ a+c & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ a+d & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ a+e & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \quad \text{and}$$

$$\partial_2' = \left[\begin{array}{c|cc} & acd & ace \\ \hline ac+ad+cd & 1 & 0 \\ ac+ae+ce & 0 & 1 \\ ab+ac+bc & 0 & 0 \\ ae & 0 & 0 \\ ad & 0 & 0 \\ ac & 0 & 0 \\ ab & 0 & 0 \end{array}\right].$$

Note that we have effectively changed the bases of the chain groups (0- and 1-dimensional chain groups in the case of ∂_1, and 1- and 2-dimensional chain groups in the case of ∂_2) by the above operations. For the d-dimensional boundary matrix, the set of zero columns corresponds to a basis of the d-dimensional cycle group, i.e. $\{ab + ac + bc, ac + ae + ce, ac + ad + cd\}$. The set of nonzero rows corresponds to a basis of the $(d-1)$-dimensional boundary group. There is a one-to-one correspondence between this boundary basis and the set of d-chains corresponding to nonzero columns, specified by these nonzero diagonal entries in the new boundary matrix.[2] Each element of this $(d-1)$-dimensional boundary basis is the boundary of its corresponding d-chain. In our example, the chains $ac + ad + cd$ and $ac + ae + ce$ are the boundaries of the 2-chains acd and ace, respectively.

The idea of reducing the boundary matrices into canonical forms [34] has been extended to various reduction algorithms for different purposes [29, 18, 44]. Next, we will introduce one specific reduction, namely, the persistent homology reduction.

2.7. Persistent Homology. We first give the intuition. Given a topological space $\mathbb{X}$ and a *filter function* $f : \mathbb{X} \to \mathbb{R}$, *persistent homology* studies changes in the topology of the sublevel sets, $\mathbb{X}^t = f^{-1}(-\infty, t]$. In Figure 2.5 the topological space is the 2-dimensional plane $\mathbb{R}^2$ and the filter function is the `peaks` function in matlab. Sublevel sets with different threshold t appear to have different topology (Figure 2.6).

As we increase the threshold t from $-\infty$ to $+\infty$, the sublevel set grows from the empty set to the entire topological space. During the growth, different homology classes may be born and then die. For example, in Figure 2.6(a), a new component is born. This component dies later (Figure 2.6(b)), when it merges into some component born earlier. In Figure 2.6(c), a new hole is born when a same component contacts itself. The newborn hole dies (Figure 2.6(d)) when it is sealed up.

The purpose of persistent homology is to capture the birth and death times of these components (0-dimensional homology classes) and holes (1-dimensional classes), and more generally, higher dimensional homology classes. By birth, we mean a homology class comes into being; by death, we mean it either becomes trivial or becomes identical to some other class born earlier. The persistence, or lifetime of a class is the

[2] The relationship may be more complicated if the homology is not over $\mathbb{Z}_2$ field.

difference between its death and birth times. Those with longer lives tell us something about the global structure of the space $\mathbb{X}$, as described by the filter function.

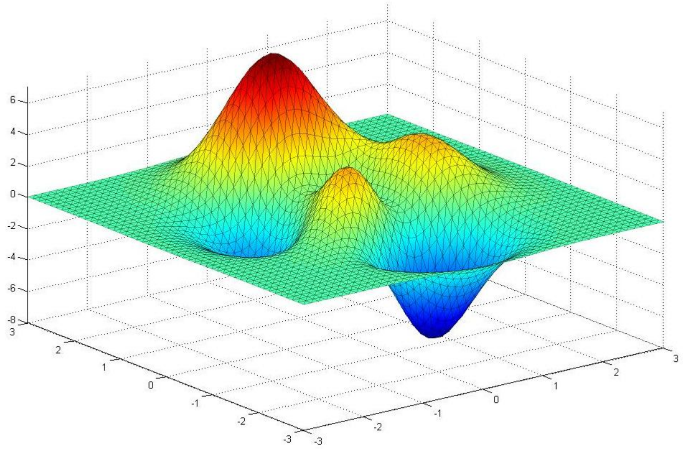
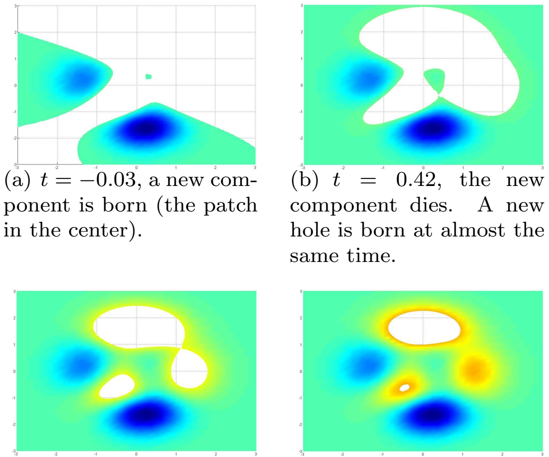


FIG. 2.5. *Persistent homology of the 2D plane filtered by a filter function (the* `peaks` *function in matlab).*



(a) $t = -0.03$, a new component is born (the patch in the center).

(b) $t = 0.42$, the new component dies. A new hole is born at almost the same time.

(c) $t = 2.24$, a new hole (on the right) is born.

(d) $t = 3.58$, the new hole dies.

FIG. 2.6. *Sublevel sets and their homologies. We draw the continuous sublevel sets whereas the persistence is computed through the simplicial complex.*

Next, we introduce formal definition. Edelsbrunner et al. [20, 43] defined the persistent homology of a simplicial complex K filtered by a scalar function. (In the example of Figure 2.6, we may imagine a triangulation of the relevant topological space, i.e. the plane.) A *filter function* $f : K \to \mathbb{R}$ assigns each simplex in K a real value, such that the function value of a simplex is no smaller than those of its faces. Without loss of generality, we assume that the filter function values of all simplices are different. Simplices of K are sorted in ascending order according to their filter function values,

$$(\sigma_1, \sigma_2, \cdots, \sigma_m), \quad f(\sigma_i) < f(\sigma_{i+1}), \quad \forall 1 \leq i \leq m-1,$$

namely, the *simplex-ordering* of K with regard to f. This is the order in which simplices enter the sublevel set $f^{-1}(-\infty, t]$ as t increases. Any sublevel set is a subcomplex, denoted as K_i, which has and only has $\sigma_1, \cdots, \sigma_i$ as its simplices. The nested sequence of sublevel sets

$$\emptyset = K_0 \subset K_1 \subset \cdots \subset K_m = K$$

is called a *filtration* of K. Let $f_i = f(\sigma_i)$ and $f_0 = -\infty$, $K_i = f^{-1}(-\infty, f_i]$.

For any $0 \leq i < j \leq m$, the inclusion mapping of K_i into K_j induces a group homomorphism of the corresponding homology groups,

$$F_d^{i,j} : \mathsf{H}_d(K_i) \to \mathsf{H}_d(K_j).$$

A *persistent homology class*, h, is born at the time f_i if $h \in \mathsf{H}_d(K_i)$ but $h \notin \mathrm{img}(F_d^{i-1,i})$. Given h is born at f_i, h dies at time f_j if $F_d^{i,j-1}(h) \notin \mathrm{img}(F_d^{i-1,j-1})$ but $F_d^{i,j}(h) \in \mathrm{img}(F_d^{i-1,j})$. Any class in the coset $h + \mathrm{img}(F_d^{i-1,i})$ is born at f_i and dies at f_j.

The *persistence* of a persistent homology class is defined as the difference between its death and birth times, which quantifies the significance of the feature. Not all the persistent homology classes die. Those which never die are *essential* classes, which correspond to nontrivial homology classes of K. An essential homology class has the $+\infty$ death time, and thus, an infinite persistence. For example, in the example of Figure 2.5, there are three 0-dimensional persistent classes. Only one of them has infinite persistence and the other two are relatively less significant and eventually die. The three 1-dimensional persistent classes also have different significances, measured by their persistences. In next section, we will discuss this in further detail.

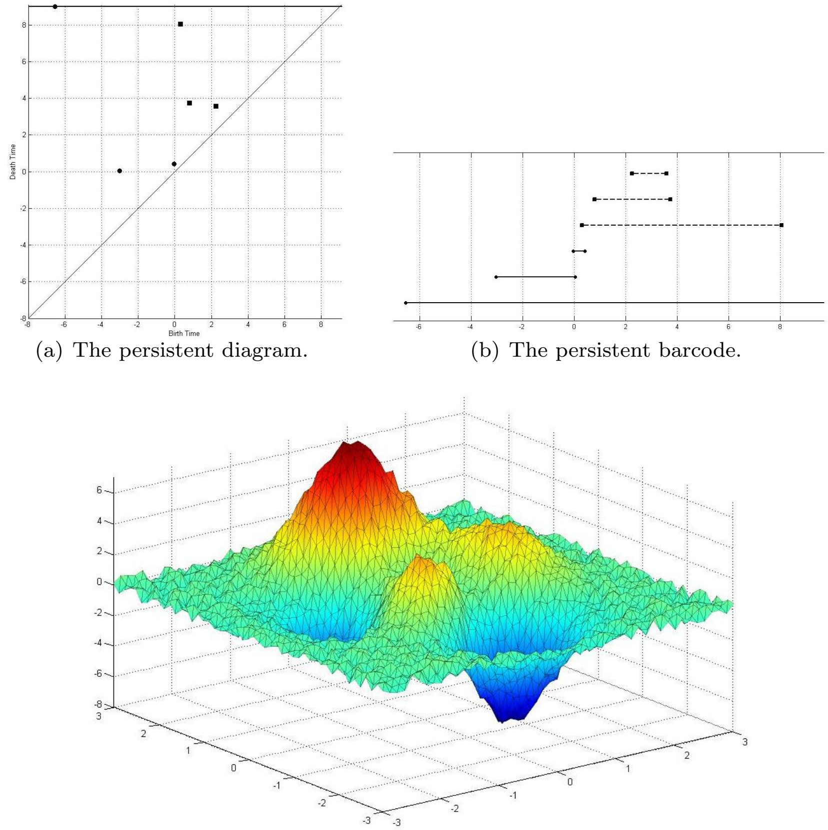
An essential justification of the usefulness of persistence is its stability [9]. It has been proved that for a given topological space, the difference between the persistent homologies of two separate filter functions is upper bounded by the difference between the filter functions, as measured by the sup-norm. The distance between persistent homologies is defined as the distance between their persistent diagrams, which will be introduced in the next section. In a recent work [5], restrictions on the space and filter functions have been relaxed. Furthermore, the stability has been extended to two different topological spaces, e.g. a manifold and its finite sampling.

The definition of persistent homology can be naturally extended to general topological space with mild assumptions. The stability guarantees that the persistence of a general topological spaces filtered by a scalar function can be approximated by the persistence of its finite approximation (triangulation of the space and finite sampling of the filter function).

2.8. The Persistence Diagram and Barcodes. The persistent homology can be visualized and studied using a *persistence diagram*, in which each persistent homology class corresponds to a point whose x and y coordinates are its birth and death times, respectively. Its persistence is equal to its vertical or horizontal distance from the diagonal. Important features correspond to points further away from the diagonal in the persistence diagram. Please see Figure 2.7(a) for the persistent diagram of the previous example. We plot 0-dimensional and 1-dimensional persistent classes with round and square marks, respectively. From the diagram, we call see that there are two important persistent classes, corresponding to the component of the space itself, and the hole which is sealed up very late (when $t = 8.06$). For convenience, persistence classes with infinite death time (essential) are plotted on a horizontal line, whose y coordinate is infinity (the thickened line in the figure).

Formally, the persistence diagram includes all the points corresponding to persistent homology classes, as well as the diagonal line. It is proven to be stable to changes in the filter function [9]. This stability of persistence implies that the persistence diagram remains almost the same if we introduce noise into the filter function (Figure 2.7(c)).

Alternatively, we could plot the life intervals of persistent classes in the real line. For a persistent class, the birth and death time are the start and end points of the interval. We call this representation a *persistent barcode*. See Figure 2.7(b) for the persistent barcode representation of the persistent homology in the preceding example. 0-dimensional classes (resp. 1-dimensional) are drawn in solid (resp. dashed) lines with round (resp. square) marks on the start and end points. The essential 0-dimensional class has no death time.



(a) The persistent diagram.

(b) The persistent barcode.

(c) Noise is introduced.

Fig. 2.7. *Persistent diagram, barcode and the filter function with noise.*

2.9. Computing Persistence. Edelsbrunner et al. [20] devised an $O(n^3)$ algorithm to compute the persistent homology. Its input are a simplicial complex and a filter function f. Its output are the birth and death times of all the persistent homology classes. The latest version of this algorithm [10, 18] unifies boundary matrices of different dimensions into one overall *incidence matrix* D. Rows and columns of D correspond to simplices of K, indexed in the simplex-ordering. An entry of D is 1 if and only if its corresponding entry is 1 in the corresponding boundary matrix. The algorithm performs column reductions on D from left to right. Each new column is reduced by addition with the already reduced columns, until its lowest nonzero entry is as high as possible.

More specifically, during the reduction, record $\mathrm{low}(i)$ as the lowest nonzero entry of each column i. To reduce column i, we repeatedly find column j satisfying $j < i$ and $\mathrm{low}(j) = \mathrm{low}(i)$; we then add column j to column i, until column i becomes a zero column or we cannot find a qualified j anymore. If column i is reduced to a zero column, $\mathrm{low}(i)$ does not exist. This is equivalent to reducing each boundary matrix into a canonical form, whose nonzero columns all have different lowest nonzero entries, and thus are linearly independent. Despite the order of reduction, as long as we only use columns on the left for reduction, the pair of each column and its lowest nonzero entry is unique.

The reduction of D can be written as a matrix multiplication,

$$R = DV, \tag{2.1}$$

where R is the *reduced matrix* and V is an upper triangular matrix. The reduced matrix R provides $\mathrm{rank}(D)$ many pairs of simplices, $(\sigma_i, \sigma_j) : \mathrm{low}(j) = i$. In such a pair, we say σ_j is paired on the right, σ_i is paired on the left. Each simplex appear in at most one pair, either on the left or on the right (cannot be both). For simplices that are not paired with any other simplex, we say they are paired with infinity: (σ_k, ∞). Simplices paired on the right are *negative* simplices. Simplices paired on the left, with other simplices or with infinity, are *positive* simplices.

A pair (σ_i, σ_j) corresponds to a persistent class, whose birth and death time are $f_i = f(\sigma_i)$ and $f_j = f(\sigma_j)$, respectively. A pair (σ_k, ∞) corresponds to an essential class, whose birth time is $f_k = f(\sigma_k)$.

The reduction is completely recorded in the matrix V. Columns of V corresponding to positive simplices form bases of cycle groups. Columns corresponding to positive simplices paired with $+\infty$ are cycles representing essential classes and form homology cycle bases.

3. Applications to Computer Vision. In this section, we sketch out applications of the algebraic topological apparatus from the previous section to problems in computer vision and image processing. We focus on four illustrative applications: computation of shape signatures, the statistics of natural images, noise reduction, and image segmentation. In the case of the latter two, we treat them simultaneously, as the topological treatments of the two problems are closely related. For each of the problems of shape signatures and natural image statistics, we describe the technique of a single paper; in the case of noise reduction and image segmentation, we describe the results of a number of papers, as these latter problems have received somewhat more treatment in the still nascent literature.

3.1. Shape Signatures. A shape signature is a compact representation of the geometry of an object. An ideal signature should be the same for all of the objects within a particular class of objects. For example, if the class is the set of objects which are rigid motions (rotations plus translations) of a given template curve in the plane, then the curvature function is a good signature. In fact, in this example, the curvature function is the ideal signature, as it is equal for all objects in the set, and is different from the signature for an object from any other such set. Such ideal signatures[3] are difficult to find in most cases of interest, so instead researchers have settled on a compromise: signatures which are similar within a class of interest, and dissimilar between classes, where similarity is measured by a particular distance function on signatures. For a survey of shape signatures, see [42, 39] and references therein.

In this section, we review the work of Collins *et al.* [4, 11] on shape signatures. The method presented in this work has the advantage that it is applicable to manifolds of any dimension; and with small modifications, to non-manifold spaces as well. Before delving into the details of this method, a natural question may arise: is the homology, on its own, sufficient to act as a shape signature? The answer is no, for two reasons. The first reason is that homology groups are sensitive to topological noise: as we have already seen in Figure 1.1, adding a small handle to a surface will completely change the homology of that surface. However, as we have noted, persistence is able to deal neatly with this problem. Thus, we may wonder whether persistent homology – with an appropriate filtration – is sufficient to act as a shape signature. The answer again is no, and this is the second reason homology (and persistent homology) is insufficient: homology is too coarse a description of an object, as very different objects may have the same or similar homology groups. (This issue was also illustrated in Figure 1.1.) The key to the method of Collins *et al.* is to *augment* the underlying space to create a geometrically more informative space, and then to use the tools of persistence to compute signature of this space.

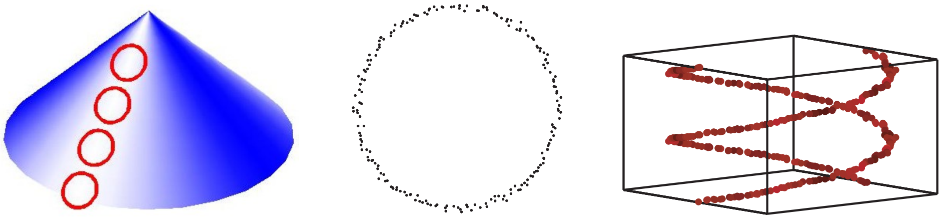


FIG. 3.1. *Two visualizations of the tangent complex. Left: the space is the blue cone, and the unit tangent vectors, which augment each point, are visualized as red circles. Middle and Right: the space is circle, here represented as a point cloud (middle); in this* 1*-dimensional case, the tangent complex can be represented explicitly (right). (Left figure taken from [4]. Copyright ©2005 World Scientific. Reprinted with permission. All rights reserved. Middle and right figures taken from [11]. Copyright ©2004 Elsevier. Reprinted with permission. All rights reserved.)*

We begin by describing the augmented space. Let $\mathbb{X}$ be the space of interest, which is a subset of $\mathbb{R}^n$. Define $T^0(\mathbb{X}) \subset \mathbb{X} \times \mathbb{S}^{n-1}$ as

$$T^0(\mathbb{X}) = \left\{ (x, \xi) \,\middle|\, \lim_{t \to 0} \frac{d(x + t\xi, \mathbb{X})}{t} = 0 \right\}$$

Then the *tangent complex* of $\mathbb{X}$, $T(\mathbb{X})$, is the closure of $T^0(\mathbb{X})$. In the case of a

[3] Note, however, that the curvature function has many problems of its own: in particular, it amplifies noise.

manifold, the tangent complex is similar to the tangent bundle of the manifold[4] – that is, each point is augmented with the set of unit tangent vectors to the manifold at that point. Thus, for example, in the case of a smooth surface living in $\mathbb{R}^3$, each point is augmented by a circle of tangent vectors, see Figure 3.1 (left). In the case of a smooth closed curve, each point is augmented by exactly two vectors (i.e., the two elements of $\mathbb{S}^0$), and the tangent complex can be visualized more explicitly, see Figure 3.1 (middle and right). The case of non-manifold behaviour is somewhat different. If a surface has a crease – e.g. imagine two planes meeting in a line – then at the crease, each point is augmented with not one, but two circles of tangent vectors. See Figure 3.2 for an illustration.

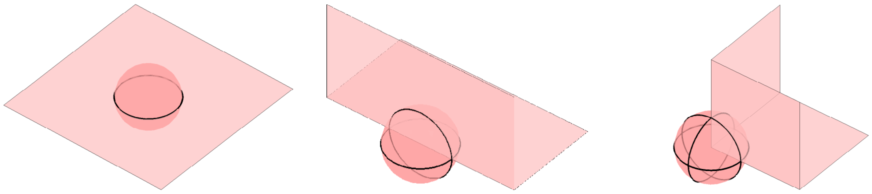


FIG. 3.2. *The tangent complex for non-manifolds. Left: where the space is a 2-manifold, each point is augmented by a single circle of unit tangent vectors. Middle and right: where there is non-manifold behaviour, the space may be augmented by more than one circle of unit tangent vectors. (Figures taken from [4]. Copyright ©2005 World Scientific. Reprinted with permission. All rights reserved.)*

In order to apply the tools of persistent homology, we will need a filter function for the space $T(\mathbb{X})$; ideally, the filter function should be geometric. Let us begin by focusing on the case when $\mathbb{X}$ is a curve, and consider the curvature $\kappa(x)$ at each point $x \in \mathbb{X}$. For any point in the tangent complex, $t = (x, \xi) \in T(\mathbb{X})$, we extend the curvature from the curve itself to the tangent complex in the natural way, $\kappa(t) = \kappa(x, \xi) \equiv \kappa(x)$. Then we may use the curvature as our filtration function. In the case of curves, this has the effect of focusing on the flat parts of the curve first, while adding in increasingly more curvy segments as we increase the value of κ. This idea is illustrated in Figure 3.3 (middle column), for a family of ellipses. To extend this definition to arbitrary manifolds – and indeed, arbitrary spaces – one may, for each $t = (x, \xi)$, define a circle of second order contact (akin to a classical osculating circle). The reciprocal of the radius of this circle gives an analogue to the curvature, which we may then use as a filter function. The interested reader is referred to [4, 11] for further details.

Finally, the shape signature for the space $\mathbb{X}$ is found by computing the persistent homology of the filtered tangent complex. This leads to a set of persistent barcodes, one set for each dimension. Recall, from Section 2.7, that the barcodes consist of intervals of the real line: the beginning of the interval is the birthtime and the end of the interval is the deathtime of the feature in question. These barcodes can be visualized by stacking the intervals, see Figure 3.3 (right column). (A historical note: the language of persistence diagrams has by and large replaced that of barcodes, although the two formalisms are equivalent. The reason that the papers [4, 11] use barcodes is that they preceded the development of the compact language of persistence

[4] Though not exactly the same, due to the use of unit vectors $\mathbb{S}^{n-1}$ in the definition of the tangent complex.

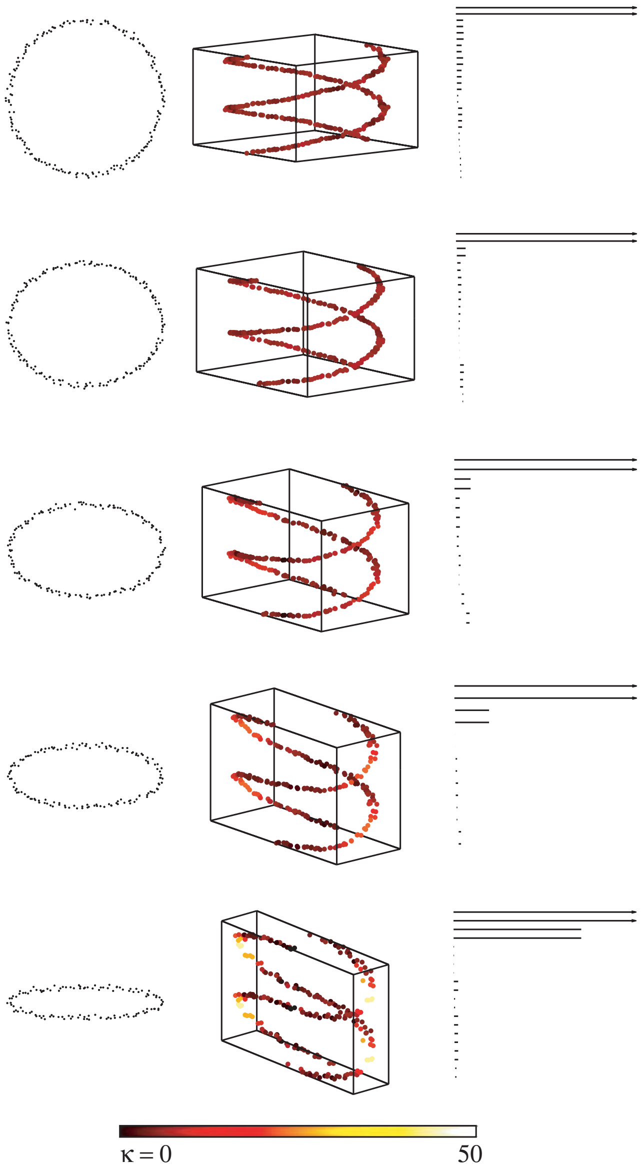


Fig. 3.3. *The persistence barcodes of shapes. Left: various ellipses, represented as point clouds. Middle: the filtered tangent complexes of these point shapes. The filter function is represented using colour, where the colour code is given at the bottom of the figure. Right: the corresponding barcodes (dimension 0), computing using persistent homology. (Figures taken from [11]. Copyright ©2004 Elsevier. Reprinted with permission. All rights reserved.)*

diagrams [9].)

In order to compare two shapes, the barcode of a given dimension of the first shape is compared with the barcode of the same dimension of the second shape. The metric between barcodes is given by a matching algorithm: the cost of matching two intervals is given by the length of their symmetric difference, while the cost of unmatched intervals is simply their length. The distance between two barcodes is then given by the cost of the minimal matching; this distance, which can be computed by a bipartite graph matching algorithm, can be shown to be a metric, as desired.

We briefly mention some of the implementation details that are necessary for computing this signature when the shape of interest is given by a point cloud, which is a common case in practice. The tangent complex can be computed by using PCA on points in the neighbourhood of a given point (taken by the k nearest neighbours for a parameter k); if the space is not smooth, this information can be backed out of the eigenanalysis of the PCA, though this is more complicated. To compute the curvature at a given point, a circle of second order contact is fit to the data. To compute the persistence, a simplicial complex is first computed as the Cech complex [34, 27], which is the dual to the union of balls where each ball (of a fixed radius ε) is centered around a point in the tangent complex. The complex is the dual in the sense that where two balls intersect, an edge is placed; where three balls intersect, a triangle is placed; and so on. This complex can be shown to be homotopy equivalent, and hence homologically equivalent, to the the union of balls [15]. Finally, the standard persistent homology algorithm (see Section 2.9) can be applied to this simplicial complex. For further details, the reader is referred to [4, 11].

Results of applying the algorithm to the problem of computing shape signatures of a set of handwritten letters are shown in Figure 3.4. Quite obviously, there are mature technologies available for the OCR problem, and this technique does not outperform them. Nonetheless, the results illustrate the power of the approach. In this example, there are eight letters, of which there are ten examples of each; in the experiments, the algorithm described above is augmented with simple information about the Betti numbers β_0 and β_1 (see [11] for details). The resulting distance matrix for these 80 elements is shown in Figure 3.4: clearly, the shape signatures have the ability to capture the shape characteristics of the handwritten letters.

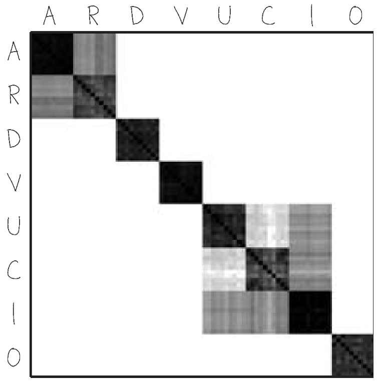


Fig. 3.4. *The distance matrix for the 80 letters (10 examples of each of 8 letters); dark represents a small value, and light represents a large value. (Figures taken from [11]. Copyright ©2004 Elsevier. Reprinted with permission. All rights reserved.)*

3.2. Statistics of Natural Images. The problem of characterizing the statistics of natural images is a traditional topic in computer vision [22, 36, 40, 28, 31, 41]. The goal is to find the basic "rules" which describe images of natural scenes; these rules, once found, serve two purposes. The first purpose is an engineering one: the rules serve as a prior on images, which can be used in probabilistic or energy formulations of a variety of problems. For example, in the case of the so called Patch Transform [8], the goal is to reassemble the patches of an image in a jigsaw puzzle-like fashion while satisfying some user constraints. The natural image statistics provided by the Gaussian Mixture Model Field of Experts model [41] are used to ensure that the assembly process leads to a sensible image. In contrast to such an engineering view, the second purpose of the study of image statistics is a scientific one. In this setting, the characterization of natural image statistics is interesting in its own right, and can lead to information about the way in which animals process visual data.

In this section, we review the work of Carlsson *et al.* [3], which uses a persistent homological approach to characterizing natural image statistics. The data used in this paper is the same as that of Lee *et al.* [31], which we briefly review. A set of 3×3 image patches is taken from a collection of more than $4,000$ images of natural scenes. The top 20% of the patches, as measured by within-patch contrast, are retained, leading to a collection of about 8 million patches. Each patch is naturally represented as a vector in $\mathbb{R}^9$; however, in order to better elucidate the structure two further transformations are performed on each patch. First, the mean intensity is subtracted off of each patch, leading to the patches inhabiting an 8-dimensional subspace of $\mathbb{R}^9$. Second, the patches are normalized so that each has unit norm. (Note that the process by which this normalization occurs is more complex than simply dividing by the norm of the vector; the interested reader is referred to [31] for details.) Finally, then, the transformed patches live in $\mathbb{S}^7$, the 7-sphere.

In this setting, the goal is thus to characterize the statistics of the dense point cloud lying on $\mathbb{S}^7$. In order to do so, a filter function related to the density of the points is introduced. For each point x in the point cloud, let $\delta_k(x)$ be the distance from x to its k^{th} nearest neighbour. Thus, $\delta_k(x)$ is inversely related to the density: the smaller is $\delta_k(x)$, the more densely represented is the area around a point. k itself is a parameter: setting k to have a small value results in a focus on the fine-scale structure of the data; whereas for k large, the coarse-scale structure of the point cloud becomes more apparent. For a fixed k, the function $\delta_k(x)$ is used as the filter function, with one caveat: the filtration ends when we have accumulated a fraction T of the points, where T is usually taken as 0.25. The reason for this latter restriction is that the high density points (i.e., the fraction T of points with the *smallest* values of $\delta_k(x)$) are said to form a "stable core," which best represent the image statistics. Finally, to speed up the computation, $5,000$ points at random are sampled from the data, and the persistence computation is performed on this subset. Many random samplings are taken to ensure consistency.

Given this filter function, what statistics are discovered? The first important discovery is that with a k value of 300 – that is, a large k value corresponding a relatively coarse scale – there is a single long-lived 1-dimensional homology class, see Figure 3.5 (top). To what does this circle on $\mathbb{S}^7$ correspond? An examination of the patches making up this circle indicates that they are patches with a light region on one side of the patch, and a dark region on the other: that is, they are edges. The circular structure turns out to be derived from the angle of the line separating the dark and the light regions; the angle can effectively take on all value from 0 to π. See Figure 3.5 (bottom) for an illustration.

The second important discovery uses a k value of 25, for a more fine scale analysis. At this scale, it is observed that there are 5 long-lived generators of the 1-dimensional homology group, see Figure 3.6 (top). There are several structures which can give

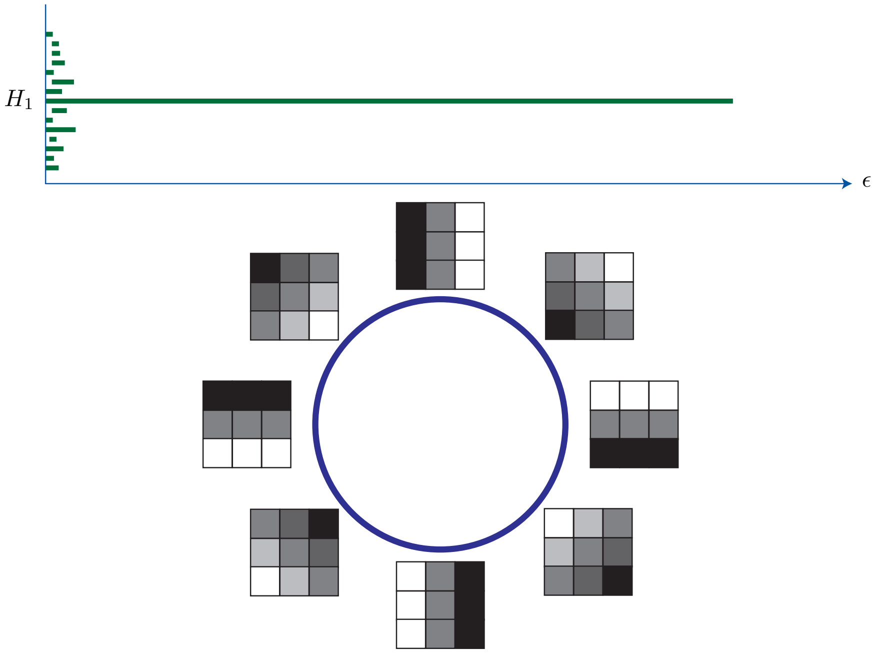


FIG. 3.5. *Coarse scale structure of the image statistics, corresponding to* $k = 300$. *Top: at this scale, the persistence barcode indicates that the data has one long-lived* 1*-dimensional homology class. Bottom: this single circle in* $\mathbb{S}^7$ *corresponds to edges, with the angle in the circle giving the angle of the edge. (Figures taken from [24]. Copyright ©2008 Robert Ghrist. Reprinted with permission. All rights reserved.)*

rise to $\beta_1 = 5$ on the 7-sphere; on examination of the data, it turns out that the relevant structure is a series of three interlocking circles, see Figure 3.6 (bottom). The first circle is the same as that already discovered at the coarse scale; the other two each intersect the original circle twice, but are disjoint from each other. It turns out that these two new circles represent patches which include three stripes, in which the stripes are not necessarily monotonic by grayscale, see Figure 3.7. Because they intersect the original circle, they include the edge patches described above (which are three stripe patterns, but where the stripes *are* monotonic by grayscale); but they also include non-edge patches, such as a patch consisting of two dark stripes sandwiching a light stripe. What is the difference between the two new circles? One circle represents horizontal stripes, and the other vertical stripes. Note that these horizontal and vertical stripes are *not* due to pixellization effects – if the images are all rotated by an angle of $\pi/4$, the true horizontal and vertical directions, represented as diagonals in this coordinate system, are still discovered [3].

Two more points deserve mention. First, more complex information can also be gleaned from the data, by looking at higher-order homology groups. For example, in looking at the 2-dimensional persistent homology, one finds a Klein Bottle structure. This structure has an explanation in terms of the underlying patches, but one which is somewhat involved to explain. The interested reader is referred to [3, 24] for a more in depth exposition. Second, this type of analysis has not been limited to natural image statistics; in a recent paper [38], Singh *et al.* have applied the same set of techniques to visual cortex data from experiments on primates. The latter should be of interest

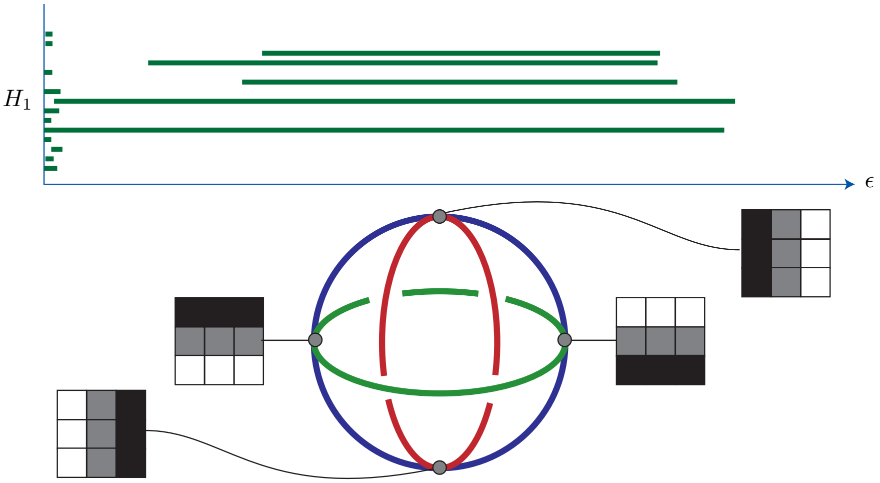


FIG. 3.6. *Fine scale structure of the image statistics, corresponding to $k = 25$. Top: at this scale, the persistence barcode indicates that the data has five long-lived 1-dimensional homology classes. Bottom: $\beta_1 = 5$ results from a series of three interlocking circles, in which the first circle is the coarse scale circle, while the other two each intersect the original circle twice, but are disjoint from each other. (Figures taken from [24]. Copyright ©2008 Robert Ghrist. Reprinted with permission. All rights reserved.)*

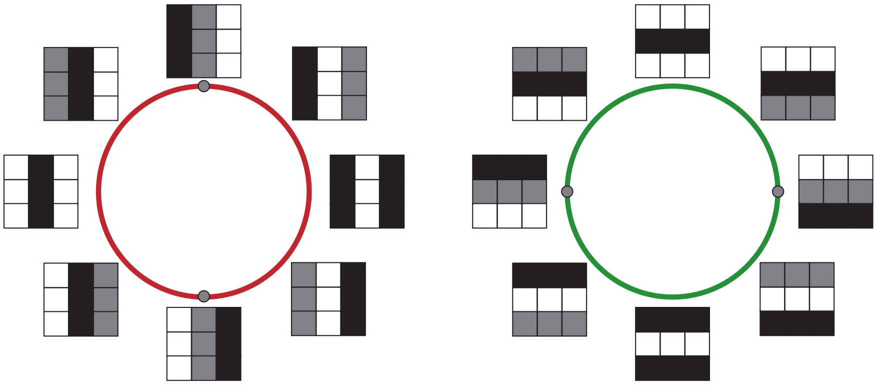


FIG. 3.7. *Analyzing the fine scale structure. The two new circles represent patches which include three stripes, in which the stripes are not necessarily monotonic by grayscale. (Figures taken from [24]. Copyright ©2008 Robert Ghrist. Reprinted with permission. All rights reserved.)*

to the biological vision community.

3.3. Noise Reduction and Segmentation. The problem of noise reduction in images is an old one. The literature on this subject is vast, so no attempt will be made to survey it here; instead, let us simply note that classical approaches tend to be based on ideas from signal processing, estimation theory, and diffusion. In some instances, there are distinct features which one wishes to preserve in the image. For example, in terrain images, it is critical to preserve the large peaks, valleys, and passes, which correspond to maxima, minima, and saddles, respectively; see Figure 3.8. In ordinary images, it may also be useful to preserve "important" critical points, as this allows the image to retain a certain sharpness while noise is removed.

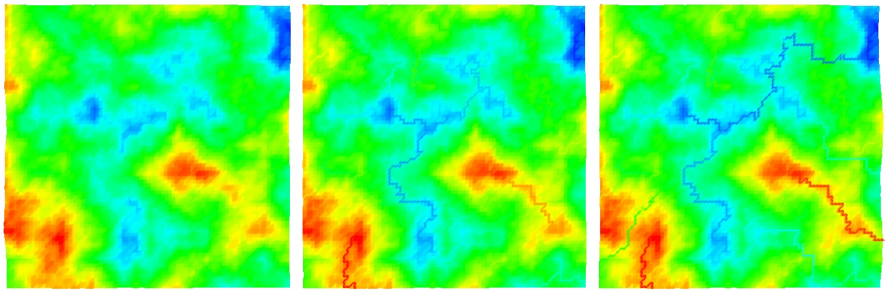


Fig. 3.8. *Simplification of a terrain image. Left: the original terrain image, in which heights range from blue (low) through cyan, green, yellow, and to red (high). Middle and right: simplifications of the terrain image which preserve large critical points. (Figures taken from [1]. Copyright ©2009 Dominique Attali. Reprinted with permission. All rights reserved.)*

What is the connection between critical points and the homological tools that we have thus far discussed? The answer is that the critical points of a function and the persistent homology of that function are intimately related. In particular, the topology of the sublevel set of the function changes whenever there is a critical point; this can be seen in the example of Figure 2.6 and corresponding discussion in Section 2.7. As we know, the persistent homology itself is defined based on the changing homology of the sublevel sets. In fact, then, it turns out that the critical points of a function are in two-to-one correspondence with the points of the persistence diagram. That is, every birth time and every death time of a homological feature corresponds to a critical point of the relevant function. (Recall that each point in the persistence diagram consists of both a birth and death coordinate, which is why the correspondence is *two*-to-one.)

Note that this relationship between the homology of a space and the critical points of a function on that space dates to Morse and his eponymous Morse Theory [33, 32]. Classical Morse Theory assumes a smooth function, which in addition satisfies a mild condition known as the Morse condition. The advantage of the persistent homology approach is that no smoothness is assumed for the function, so that a sensible definition of critical points exists even when the underlying function is not smooth. That is, a homological critical value [9] of a function is a value at which the homology of the sublevel set of the function changes. This definition corresponds with the traditional critical point (at which the derivative vanishes) for smooth functions, but is more general. For example, it can be applied to a piecewise linear function, which is a standard case seen in applications.

Having established the relationship between critical points and points in the persistence diagram, we may now formulate the problem of feature-sensitive noise reduction as that of removing points with small persistence from the persistence diagram, while leaving other points in the persistence diagram alone. This has the effect of retaining important critical points, while discarding other, less significant ones. This problem has been addressed in the persistent homology literature, and goes under the title *persistence simplification*. The formal problem of persistence simplification, as described in [21, 1] is as follows:

Definition 3.1. *Given a topological space $\mathbb{X}$ and function $f : \mathbb{X} \to \mathbb{R}$, a function $g : \mathbb{X} \to \mathbb{R}$ is an ε-simplification of f if the two functions are close, $\|f - g\|_\infty \leq \infty$,* *and the persistence diagram $D(g)$ contains only those points in the diagram $D(f)$ that are more than ε away from the diagonal.*

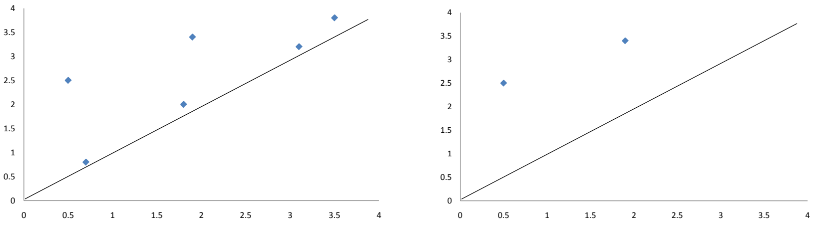


FIG. 3.9. *Persistence simplification. Left: the original persistence diagram for a given function f. Right: the persistence diagram for the ε-simplification g of the function f. Here $\varepsilon = 0.3$.*

It should be clear that if such a g can be found, it will have solved the problem of feature-sensitive noise reduction, see Figure 3.9. Before examining algorithms for this general problem, however, we observe an interesting connection with the seemingly unrelated problem of segmentation. It will turn out that an important problem in segmentation admits a simpler version of persistence simplification, for which an algorithm has been developed. We will then return to a high-level discussion of results for the more general persistence simplification problem.

3.3.1. Segmentation. The connection between persistence simplification and segmentation comes about through the Mean Shift algorithm. In Mean Shift segmentation [23, 7, 12], each pixel in the image is assigned a feature vector – generally speaking, colour, texture, or some combination of the two, which is then often augmented by position. A non-parametric estimate of the probability density in this feature space – the Kernel Density Estimate (KDE) – is then constructed. Given this KDE, the image is segmented according to the *modes*, or local maxima, of the KDE. In particular, each local maximum of the KDE represents a segment of the image, and each pixel is assigned to the local maximum in whose basin of attraction the pixel lies.

Despite its success in many applications, the Mean Shift algorithm is known to generally yield an oversegmentation; that is, it produces too many segments. In some applications, this is tolerable; Mean Shift is sometimes used as preprocessing for a more sophisticated clustering algorithm, and is used simply to reduce the complexity of this second algorithm. On the other hand, it would be desirable if Mean Shift were able to yield a more precise segmentation on its own. Since Mean Shift tends to oversegment, the results might be corrected by forcing the algorithm to produce fewer clusters. Since each cluster corresponds to a local maximum of the KDE, the problem of mitigating oversegmentation is equivalent to the problem of filtering the KDE so as to preserve the most important local maxima, while eliminating smaller ones. In this sense, this problem is formally similar to a simpler version of the feature-sensitive noise reduction and persistence simplification outlined above. In particular, we are interested not in preserving all large critical points, but rather, only large local maxima.

Chazal *et al.* [6] present an elegant and practical method for attacking this problem. Before discussing their method, however, it is worth pointing out the contribution of Paris and Durand [37], who also attempt to tackle this problem. The basic idea of the paper is roughly in consonance with the approach of persistence simplification, and is the first work, to our knowledge, to try to tackle this problem in relation to segmentation. However, there are problems: the method is very "digital," in that it attempts to find basins of attraction and then to use a topological persistence oriented criterion for eliminating modes, by just using the digital grid, without a true simplicial complex (or other appropriate cell complex) underlying the analysis. The digitality proves to be a problem, both in theory and in practice. In theory, there is not much one can prove here; and in practice, some grid points are never classified as belonging to any basin of attraction, and a heuristic must be used.

The method of Chazal *et al.* [6], by contrast, is well-founded theoretically, and provides some nice practical properties as well. The set up is as follows. It is assumed that the topological space $\mathbb{X}$ is given to the user as a sampled space, that is as a finite subset L. In fact, the representation of L is entirely coordinate-free; all that is needed is the set of pairwise distances between all points in L.[5] The goal is to compute the persistence of this sampled representation, and to simplify it, in the sense above: to eliminate local maxima whose persistence is smaller than a given threshold.

The first contribution is to show how the persistence diagram itself may be computed for the sampled representation. The details of this procedure are highly technical (relying on many algebraic concepts, such as persistence modules, not described here), so we will only give a rough sketch; the interested reader is referred to [6] for the full treatment. The *Rips Complex* of a finite point set L and a positive real number δ is denoted $R_\delta(L)$ and is defined as follows. A k-simplex σ with vertex set $v_0, \ldots, v_k \in L$ belongs to $R_\delta(L)$ if the distance between all pairs of vertices is less than or equal to δ: $d(v_i, v_j) \leq \delta$ for all i, j. It turns out that there is no value of δ for which the persistent homology of the Rips Complex $R_\delta(L)$ matches that of $\mathbb{X}$, even for well-sampled spaces. Instead, the relationship between *a pair* of Rips Complexes, $R_\delta(L)$ and $R_{2\delta}(L)$, is sufficient to yield the persistent homology of the original space $\mathbb{X}$. We do not formally define the nature of this relationship here, as it depends on very advanced algebraic concepts; we merely note that it can proven that if the space is well sampled, the persistence diagram of $\mathbb{X}$ can be computed.

Note that this result on its own is quite important, as it allows for the computation of persistence in high dimensional spaces, when an approach based on a simplicial complex would be too expensive (note that the size of the simplicial complex generally grows exponentially with the embedding dimension). In addition, though, the authors show how to use this scheme to simply deal with the oversegmentation problem from Mean Shift. First, a sampled version of Mean Shift is proposed: at any point, a steepest ascent vector is defined by looking at the highest (by function value) point in the neighbourhood of the point, where the neighbourhood is defined using 1-skeleton of the the Rips Complex $R_{2\delta}(L)$. The point is then moved according to this steepest ascent vector, unless the point itself is the highest point in its own neighbourhood, in which case it is deemed a local maximum. The persistences of each such local maximum have already been computed using the algorithm sketched in the previous paragraph. In fact, the result of that algorithm is a diagram in which neighbouring maxima are linked. More formally, the diagram consists of pairs (v, e), where v is a local maximum and e is an edge of the 1-skeleton of the Rips Complex $R_{2\delta}(L)$ that links the connected component created by v in $R_{2\delta}(L)$ to the one created by some higher maximum u. If the lifespan of the connected component of v is shorter than some threshold, then the cluster of v is merged into that of u.

[5] Thus, $\mathbb{X}$ must also be a metric space; in practice, this is never a restriction, and most relevant spaces have even more structure, that is they are Riemannian manifolds.

This algorithm, in addition to its provable properties (under appropriate sampling), can be shown to have a reasonable complexity. In particular, the complexity of the first part of the algorithm, the computation of persistence, is $O(n^3)$, where n is the number of points in L, whereas the second part is close to linear in n. Results of applying the algorithm are illustrated in Figure 3.10.

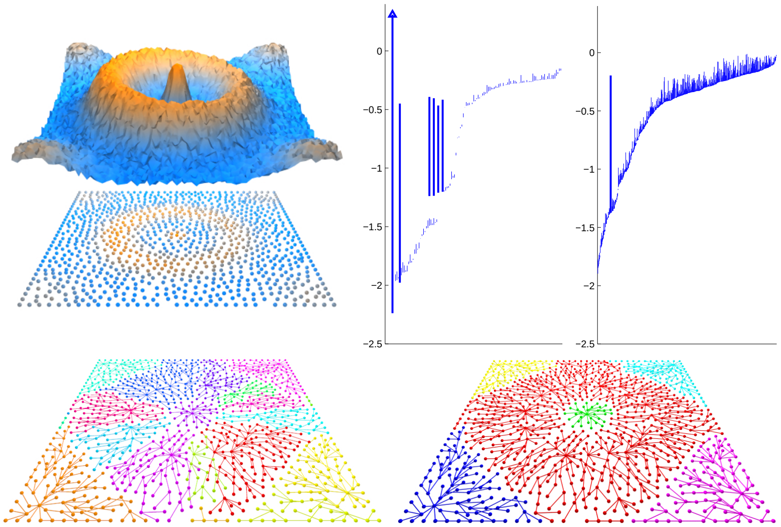


FIG. 3.10. *Segmentation. Top left: the function. Top right: the persistence barcode, illustrating the six large local maxima. Bottom left: the results of (discrete) Mean Shift. Bottom right: the results after merging clusters using persistence. (Figures taken from [6]. Copyright ©2009 Society for Industrial and Applied Mathematics. Reprinted with permission. All rights reserved.)*

3.3.2. General Persistence Simplification. As opposed to our treatment of the problems in the previous sections, in which we focused on one particular algorithm, we will, in this case, proceed to summarize the state of existing results in this field. This is mainly due to the fact that a somewhat larger literature has developed to tackle the problem of persistence simplification, though open problems certainly remain.

The first paper to introduce persistent homology [20] had already considered the problem of persistence simplification. In this paper, the setting is purely simplicial, and the filtration itself is given by an ordering of the simplices. An algorithm is given there for reordering of the simplices which simplifies the persistence. The main problem with this algorithm is that it reduces the persistence of *all* features; that is, all points in the persistence diagram are moved closer to the diagonal, not just the smaller ones. This is not really desirable, as we wish to preserve the large features as precisely as possible, while removing the smaller ones.

Another series of papers tackle the problem of simplification of Morse-Smale complexes [19, 2, 26]. Briefly, the Morse-Smale complex bears a close relationship to the problem of Mean-Shift segmentation. The stable manifolds which are key ingredients in the construction of the Morse-Smale complex are essentially the same as the basins of attraction of the modes in Mean Shift; however, the Morse-Smale complex also uses the concept of unstable manifolds, which do not have a direct analogy in Mean Shift. (Unstable manifolds essentially allow points to run "down the hill" instead of up; this can be seen as performing mean shift on the negative KDE. The Morse-Smale complex then takes intersections of stable and unstable manifolds to build up a cell complex.) In any case, these methods are somewhat different than what we are after, as they simplify the complex, rather than the filter function. That is, they change the domain of the function, rather than the function itself. This can sometime be useful, for example when we are interested in simplifying a surface; but in the case of noise reduction and reducing the effects of oversegmentation, the main issue is to simplify the function itself. Another issue related to these results for simplification is that they are highly dependent on the low-dimension of the underlying topological space; the cases of dimension 2 and 3 are considered in the previously cited papers. In many cases, clustering occurs in spaces of somewhat higher dimension; for example, a common choice in image segmentation is $d = 5$, where each vector comprises 3 colour and 2 spatial dimensions.

The original paper to pose the problem of persistence simplification as in Definition 3.1 was that of Edelsbrunner et al. [21]. In this paper, the problem was solved for piecewise linear 2-manifolds (surface meshes); the main problem with the approach is simply that it is very complicated, with many subcases considered. A more recent paper of Attali et al. [1] also tackles the problem of simplification on surfaces and provides a simple algorithm, which is relatively simple to implement, and has a low complexity – linear in the number of simplices. Both of these papers are restricted to the low-dimensional setting.

4. Conclusions and Future Directions.

In this survey, we have reviewed the new algorithms for computing with algebraic topology, in particular those of persistent homology; and the application of these algorithms to problems in computer vision and image processing. These techniques require some effort to master, but we believe that the effort is worth it: the techniques represent powerful new ways to attack interesting problems in vision. Furthermore, the methods have an inherent elegance which should be appealing to many vision researchers.

We believe that this is just the beginning of the application of the new topological ideas to image related problems. This paper ought merely to be an entryway for interested researchers into the exciting new developments in computational and applied algebraic topology. It is our hope that in five years, another survey will be required to cover the much larger number of developments that will have taken place over that time.

REFERENCES

[1] D. Attali, M. Glisse, S. Hornus, F. Lazarus, and D. Morozov, *Persistence-sensitive simplification of functions on surfaces in linear time*, Submitted to SoCG, 9.

[2] P.T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci, *A topological hierarchy for functions on triangulated surfaces*, IEEE Transactions on Visualization and Computer Graphics, 10 (2004), pp. 385–396.

[3] G. Carlsson, T. Ishkhanov, V. De Silva, and A. Zomorodian, *On the local behavior of spaces of natural images*, International Journal of Computer Vision, 76 (2008), pp. 1–12.

[4] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas, *Persistence barcodes for shapes*, International Journal of Shape Modeling, 11 (2005), pp. 149–187.

[5] F. Chazal, D. Cohen-Steiner, M. Glisse, L.J. Guibas, and S.Y. Oudot, *Proximity of persistence modules and their diagrams*, in Proceedings of the 25th annual symposium on Computational geometry, ACM New York, NY, USA, 2009, pp. 237–246.

[6] F. Chazal, L.J. Guibas, S.Y. Oudot, and P. Skraba, *Analysis of scalar fields over point cloud data*, in Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2009, pp. 1021–1030.

[7] Y. Cheng, *Mean shift, mode seeking, and clustering*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 17 (1995), pp. 790–799.

[8] T.S. Cho, M. Butman, S. Avidan, and W.T. Freeman, *The patch transform and its applications to image editing*, in IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, 2008, pp. 1–8.

[9] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, *Stability of persistence diagrams*, Discrete and Computational Geometry, 37 (2007), pp. 103–120.

[10] D. Cohen-Steiner, H. Edelsbrunner, and D. Morozov, *Vines and vineyards by updating persistence in linear time*, in Proceedings of the twenty-second annual symposium on Computational geometry, ACM New York, NY, USA, 2006, pp. 119–126.

[11] A. Collins, A. Zomorodian, G. Carlsson, and L.J. Guibas, *A barcode shape descriptor for curve point cloud data*, Computers & Graphics, 28 (2004), pp. 881–894.

[12] D. Comaniciu and P. Meer, *Mean Shift: A Robust Approach Toward Feature Space Analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (2002), pp. 603–619.

[13] V. de Silva, R. Ghrist, and A. Muhammad, *Blind swarms for coverage in 2-d*, in Robotics: Science and Systems, 2005.

[14] T.K. Dey, K. Li, J. Sun, and D. Cohen-Steiner, *Computing geometry-aware handle and tunnel loops in 3D models*, (2008).

[15] H. Edelsbrunner, *The union of balls and its dual shape*, Discrete and Computational Geometry, 13 (1995), pp. 415–440.

[16] ———, *63 BIOLOGICAL APPLICATIONS OF COMPUTATIONAL TOPOLOGY*, Handbook of Discrete and Computational Geometry, (2004), p. 1395.

[17] ———, *Surface tiling with differential topology*, in ACM International Conference Proceeding Series; Vol. 255: Proceedings of the third Eurographics symposium on Geometry processing: Vienna, Austria, Association for Computing Machinery, Inc, One Astor Plaza, 1515 Broadway, New York, NY, 10036-5701, USA,, 2005.

[18] H. Edelsbrunner and J. Harer, *Persistent homology — a survey.*, In Twenty Years After, eds. J. E. Goodman, J. Pach and R. Pollack, AMS., (2007).

[19] H. Edelsbrunner, J. Harer, and A. Zomorodian, *Hierarchical Morse–Smale Complexes for Piecewise Linear 2-Manifolds*, Discrete and Computational Geometry, 30 (2003), pp. 87–107.

[20] H. Edelsbrunner, D. Letscher, and A. Zomorodian, *Topological persistence and simplification*, Discrete and Computational Geometry, 28 (2002), pp. 511–533.

[21] H. Edelsbrunner, D. Morozov, and V. Pascucci, *Persistence-sensitive simplification functions on 2-manifolds*, in Proceedings of the twenty-second annual symposium on Computational geometry, ACM New York, NY, USA, 2006, pp. 127–134.

[22] D.J. Field, *Relations between the statistics of natural images and the response properties of cortical cells*, Journal of the Optical Society of America A, 4 (1987), pp. 2379–2394.

[23] K. Fukunaga and L. Hostetler, *The estimation of the gradient of a density function, with applications in pattern recognition*, IEEE Transactions on Information Theory, 21 (1975), pp. 32–40.

[24] R. Ghrist, *Barcodes: The persistent topology of data*, BULLETIN-AMERICAN MATHEMATICAL SOCIETY, 45 (2008), p. 61.

[25] R. Ghrist and A. Muhammad, *Coverage and hole-detection in sensor networks via homology*, in Proceedings of the 4th international symposium on Information processing in sensor networks, IEEE Press Piscataway, NJ, USA, 2005.

[26] A. Gyulassy, V. Natarajan, V. Pascucci, P.T. Bremer, and B. Hamann, *Topology-based simplification for feature extraction from 3d scalar fields*, in Proc. IEEE Conf. Visualization, 2005, pp. 535–542.

[27] A. Hatcher, *Algebraic topology*, Cambridge University Press, 2002.

[28] J. Huang and D. Mumford, *Statistics of natural images and models*, in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 1999, pp. 541–547.

[29] T. Kaczynski, M. Mrozek, and M. Slusarek, *Homology computation by reduction of chain complexes.*, Computers and Math. Appl., 35 (1998), pp. 59–70.

[30] R. Kannan and A. Bachem, *Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix*, SIAM Journal on Computing, 8 (1979), p. 499.

[31] A.B. Lee, K.S. Pedersen, and D. Mumford, *The nonlinear statistics of high-contrast patches in natural images*, International Journal of Computer Vision, 54 (2003), pp. 83–103.

[32] Y. Matsumoto and K. Hudson, *An introduction to Morse theory*, Amer Mathematical Society, 2002.

[33] J.W. Milnor, *Morse theory*, Princeton university press, 1963.

[34] J. R. Munkres, *Elements of Algebraic Topology.*, Addison-Wesley, Redwook City, California, 1984.

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

[36] B. Olshausen and D. Field, *Natural image statistics and efficient coding*, Network: Computation in Neural Systems, 7 (1996), pp. 333–339.

[37] S. Paris and F. Durand, *A topological approach to hierarchical segmentation using mean shift*, in Proc. CVPR, 2007.

[38] Gurjeet Singh, Facundo Memoli, Tigran Ishkhanov, Guillermo Sapiro, Gunnar Carlsson, and Dario L. Ringach, *Topological analysis of population activity in visual cortex*, J. Vis., 8 (2008), pp. 1–18.

[39] J.W.H. Tangelder and R.C. Veltkamp, *A survey of content based 3D shape retrieval methods*, Multimedia Tools and Applications, 39 (2008), pp. 441–471.

[40] A. Van der Schaaf and JH Van Hateren, *Modelling the power spectra of natural images: statistics and information*, Vision Research, 36 (1996), pp. 2759–2770.

[41] Y. Weiss and WT Freeman, *What makes a good model of natural images?*, in IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07, 2007, pp. 1–8.

[42] D. Zhang and G. Lu, *Review of shape representation and description techniques*, Pattern Recognition, 37 (2004), pp. 1–19.

[43] Afra Zomorodian, *Topology for Computing.*, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2005.

[44] Afra Zomorodian and Gunnar Carlsson, *Computing persistent homology.*, Discrete & Computational Geometry, 33 (2005), pp. 249–274.