Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Graphics 28 (2004) 881–894
COMPUTERS & GRAPHICS
www.elsevier.com/locate/cag

A barcode shape descriptor for curve point cloud data

Anne Collins[a,*,1], Afra Zomorodian[b,2], Gunnar Carlsson[a,1], Leonidas J. Guibas[b,2]

[a] *Department of Mathematics, Stanford University, 450 Serra Mall, Bldg. 380, Stanford, CA 94305 2125, USA*
[b] *Department of Computer Science, Stanford University, Stanford, CA 94305, USA*

Abstract

In this paper, we present a complete computational pipeline for extracting a compact shape descriptor for curve point cloud data (PCD). Our shape descriptor, called a *barcode*, is based on a blend of techniques from differential geometry and algebraic topology. We also provide a metric over the space of barcodes, enabling fast comparison of PCDs for shape recognition and clustering. To demonstrate the feasibility of our approach, we implement our pipeline and provide experimental evidence in shape classification and parametrization.
© 2004 Elsevier Ltd. All rights reserved.

Keywords: Barcode; Descriptor; Persistence; Tangent complex; Point cloud data; Curves

1. Introduction

In this paper, we present a complete computational pipeline for extracting a compact shape descriptor for curve point cloud data (PCD). Our shape descriptor, called a *barcode*, is based on a blend of techniques from differential geometry and algebraic topology. We also provide a metric over the space of barcodes, enabling fast comparison of PCDs for shape recognition and clustering. To demonstrate the feasibility of our approach, we implement our pipeline and provide experimental evidence in shape classification and parametrization.

1.1. Prior work

Shape analysis is a well-studied problem in many areas of computer science, such as vision, graphics, and pattern recognition. Researchers in vision first introduced the idea of using compact representations of shapes, or *shape descriptors*, for two-dimensional data or images. They derived descriptors using diverse methods, such as topological invariants, moment invariants, morphological methods for skeletons or medial axes, and elliptic Fourier parameterizations [1–3]. More recently, the availability of large sets of digitized three-dimensional shapes has generated interest in 3D descriptors [4,5], with techniques such as shape distributions [6] and multi-resolution Reeb graphs [7]. Ideally, a shape descriptor should be invariant to rigid transformations and coordinatize the shape space in a meaningful way.

The idea of using *point cloud data* or *PCD* as a display primitive was introduced early [8], but did not become popular until the recent emergence of massive datasets. PCDs are now utilized in rendering [9,10], shape representation [11,12], and modeling [13,14], among

*Corresponding author.
E-mail addresses: collins@math.stanford.edu (A. Collins), afra@cs.stanford.edu (A. Zomorodian), gunnar@math.stanford.edu (G. Carlsson), guibas@cs.stanford.edu (L.J. Guibas).
[1]Research supported, in part, by NSF under grant DMS 0101364.
[2]Research supported, in part, by NSF/DARPA under grant CARGO 0138456 and by NSF under grant ITR 0086013.

0097-8493/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2004.08.015

other uses. Furthermore, PCDs are often the only possible primitive for exploring shapes in higher dimensions [15–17].

1.2. Our work

In a previous paper, we initiated a study of shape description via the application of persistent homology to tangential constructions [18]. We proposed a robust method that combines the differentiating power of geometry with the classifying power of topology. We also showed the viability of our method through explicit calculations for one- and two-dimensional mathematical objects (curves and surfaces.) In this paper, we shift our focus from theory to practice, illustrating the feasibility of our method in the PCD domain. We focus on curves in order to explore the issues that arise in the application of our techniques. We must emphasize, however, that we view curves as one-dimensional manifolds, and insist that all our solutions extend to n-dimensional manifolds. Therefore, we avoid heuristics based on abusing characteristics of curve PCDs and search for general techniques that will be suitable in all dimensions. We briefly discuss computing our structures for two-dimensional manifolds, or surfaces, in Section 5.5.

1.3. Overview

The rest of the paper is organized as follows. In Section 2 we review the theoretical background for our shape descriptor. We believe that an intuitive understanding of this material is sufficient for appreciating the results of this paper. As such, this section is brief in its treatment, and we refer the interested reader to our previous paper for a detailed description [18]. Section 3 contains the algorithms for computing barcodes for PCDs sampled from closed smooth curves. We also describe the computation of the metric over the space of barcodes. We apply our techniques to families of algebraic curves in Section 4 to demonstrate their effectiveness. In Section 5, we extend our system to general PCDs that may include non-manifold points, singularities, boundary points, or noise. We then illustrate the power of our methods through applications to shape classification and parametrization in Section 6.

2. Background

In this section, we review the theoretical background necessary for our work. To make the discussion accessible to the non-specialist, our exposition will have an intuitive flavor.

2.1. Filtered simplicial complex

Let S be a set of points. A *k-simplex* is a subset of S of size $k+1$ (19). A simplex may be realized geometrically as the convex hull of $k+1$ affinely independent points in $\mathbb{R}^d, d \geqslant k$. A realization gives us the familiar low-dimensional k-simplices: *vertices*, *edges*, and *triangles*. A *simplicial complex* is a set K of simplices on S such that if $\sigma \in K$ then $\tau \subset \sigma$ implies $\tau \in K$. A *subcomplex* of K is a simplicial complex $L \subseteq K$. A *filtration* of a complex K is a nested sequence of complexes $\emptyset = K^0 \subseteq K^1 \subseteq \cdots \subseteq K^m = K$. We call K a *filtered complex* and show a small example in Fig. 1.

2.2. Persistent homology

Suppose we are given a shape X that is embedded in $\mathbb{R}^3$. *Homology* is an algebraic invariant that counts the topological attributes of this shape in terms of its *Betti numbers* β_i [19]. Specifically, β_0 counts the number of components of X. β_1 is the rank of a basis for the *tunnels* through X. These tunnels may be viewed as forming a graph with cycles [20]. β_2 counts the number of *voids* in X, or spaces that are enclosed by the shape. In this

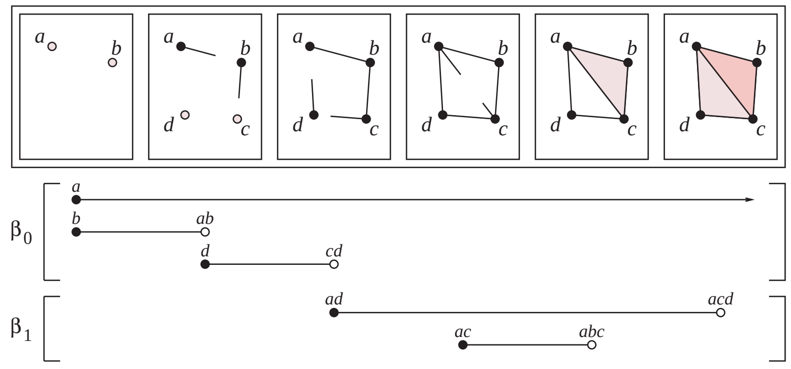

Fig. 1. A filtered complex with newly added simplices highlighted. We show the persistent interval set in each dimension below the filtration. Each persistent interval shown is the lifetime of a topological attribute, created and destroyed by the simplices at the low and high endpoints, respectively.

manner, homology gives a finite compact description of the connectivity of the shape. Since homology is an invariant, we may represent our shape combinatorially with a simplicial complex that has the same connectivity to get the same result.

Suppose now that we are also given a process for constructing our shape from scratch. Such a growth process gives an evolving shape that undergoes topological changes: new components appear and connect to the old ones, tunnels are created and closed off, and voids are enclosed and filled in. *Persistent homology* is an algebraic invariant that identifies the birth and death of each topological attribute in this evolution [21,22]. Each attribute has a lifetime during which it contributes to some Betti number. We deem important those attributes with longer lifetimes, as they *persist* in being features of the shape. We may represent this lifetime as an interval, as shown in Fig. 1 for our small example. A feature, such as the first component in any filtration, may live forever and therefore have a half-infinite interval as its lifetime. Persistent homology describes the connectivity of our evolving shape via a multiset of intervals in each dimension. If we represent our shape with a simplicial complex, we may also represent its growth with a filtered complex.

2.3. Filtered tangent complex

We examine the geometry of our shape by looking at the tangents at each point of the shape. Although our approach extends to any dimension, we restrict our definitions to curves as they are the focus of this paper and simplify the description.

Let X be a curve in $\mathbb{R}^2$. We define $T^0(X) \subseteq X \times \mathbb{S}^1$ to be the set of the tangents at all points of X. That is,

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

A point (x, ζ) in $T^0(X)$ represents a tangent vector at a point $x \in X$ in the direction $\zeta \in \mathbb{S}^1$. The *tangent complex* of X is the closure of $T^0, T(X) = \overline{T^0(X)} \subseteq \mathbb{R}^2 \times \mathbb{S}^1$. $T(X)$ is equipped with a projection $\pi : T(X) \to X$ that projects a point $(x, \zeta) \in T(X)$ in the tangent complex onto its *basepoint* $x \in X$, and $\pi^{-1}(x) \subseteq T(X)$ is the *fiber at* x.

We may filter the tangent complex using the curvature at each point. We let $T^0_\kappa(X)$ be the set of points $(x, \zeta) \in T^0(X)$ where the curvature $\kappa(x)$ at x is less than κ, and define $T_\kappa(X)$ be the closure of $T^0_\kappa(X)$ in $\mathbb{R}^2 \times \mathbb{S}^1$. We call the κ-parametrized family of spaces $\{T_\kappa(X)\}_{\kappa \geqslant 0}$ the *filtered tangent complex*, denoted by $T^{filt}(X)$.

2.4. Barcodes

We get a compact descriptor by applying persistent homology to the filtered tangent complex of our shape. That is, the descriptor examines the connectivity of not the shape itself, but that of a derived space that is enriched with geometric information about the shape. We define a *barcode* to be the resulting set of persistence intervals for $T^{filt}(X)$ in each dimension. For curves, the only interesting barcode is usually the β_0-barcode which describes the lifetimes of the components in the growing tangent complex. We also define a quasi-metric, a metric that has ∞ as a possible value, over the collection of all barcodes. Our metric enables us to utilize barcodes as shape descriptors, as we can compare shapes by measuring the difference between their barcodes.

Let I, J be any two intervals in a barcode. We define their dissimilarity $\delta(I, J)$ to be the length of their symmetric difference: $\delta(I, J) = |I \cup J - I \cap J|$. Note that $\delta(I, J)$ may be infinite. Given a pair of barcodes B_1 and B_2, a *matching* is a set $M(B_1, B_2) \subseteq B_1 \times B_2 = \{(I, J) \mid I \in B_1 \text{ and } J \in B_2\}$, so that any interval in B_1 or B_2 occurs in at most one pair (I, J). Let M_1, M_2 be the intervals from B_1, B_2, respectively, that are matched in M, and let N be the non-matched intervals $N = (B_1 - M_1) \cup (B_2 - M_2)$. Given a matching M for B_1 and B_2, we define the *distance of* B_1 *and* B_2 *relative to* M to be the sum

$$\mathscr{D}_M(B_1, B_2) = \sum_{(I,J) \in M} \delta(I, J) + \sum_{L \in N} |L|. \tag{1}$$

We now look for the best possible matching to define the quasi-metric:

$$\mathscr{D}(B_1, B_2) = \min_M \mathscr{D}_M(B_1, B_2).$$

3. Computing barcodes

In this section, we present a complete pipeline for computing barcodes for a PCD. Throughout this section, we assume that our PCD P contains samples from a smooth closed curve $X \subset \mathbb{R}^2$. Before we compute the barcode, we need to construct the tangent complex. Since we only have samples from the original space, we can only approximate the tangent complex. We begin by computing a new PCD, $\pi^{-1}(P) \subset T(X)$, that samples the tangent complex for our shape. To capture its homology, we first approximate the underlying space and then compute a simplicial complex that represents its connectivity. We filter this complex by estimating the curvature at each point of $\pi^{-1}(P)$. We conclude this section by describing the barcode computation and giving an algorithm for computing the metric on the barcode space.

3.1. Fibers

Suppose we are given a PCD P, as shown in Fig. 2(a). We wish to compute the fiber at each point to generate a

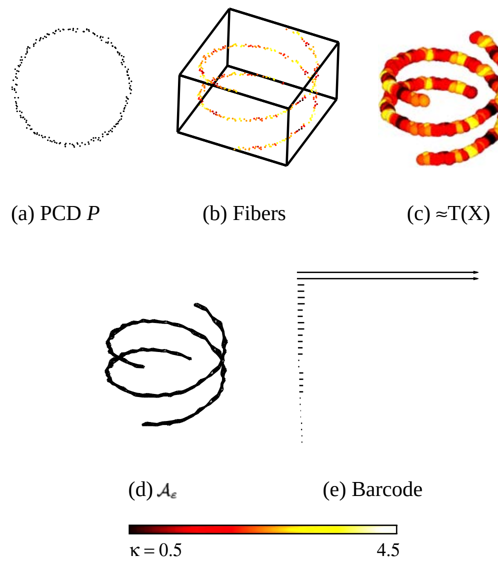

Fig. 2. Given a noisy PCD P (a), we compute the fibers $\pi^{-1}(P)$ (b) by fitting lines locally. In the volume, the z-axis corresponds to tangent angle, so the top and the bottom of the volume are glued. We center ε-balls with $\varepsilon = 0.05$ at the fiber points to get a space (c) that approximates the tangent complex $T(X)$. The fibers and union of balls are colored according to curvature using the hot colormap shown. The curvature estimates appear noisy because of the small variation. We capture the topology of the union of balls using the simplicial complex (d). We show the α-complex $\mathscr{A}_\varepsilon$ for the union of balls in the figure (with $\alpha = \varepsilon$) as it has a nice geometric realization. In practice, we utilize the Rips complex. Applying persistent homology, we get the β_0-barcode (e).

new PCD $\pi^{-1}(P)$ that samples the tangent complex $T(X)$. Naturally, we must estimate the tangent directions, as we do not have the underlying shape X from which P was sampled. We do so by approximating the tangent line to the curve X at point $p \in P$ via a *total least squares* fit that minimizes the sum of the squares of the perpendicular distances of the line to the point's nearest neighbors. Let S be the k nearest neighbors to p, and let $x_0 = \frac{1}{k}\sum_{i=1}^{k} x_i$ be the average of the points in S. We assume that the best line passes through x_0. In general, the hyperplane $\mathscr{P}(n, x_0)$ in $\mathbb{R}^n$ which is normal to n and passes through the point x_0 has equation $(x - x_0) \cdot n = 0$. The perpendicular distance from any point $x_i \in S$ to this hyperplane is $|(x_i - x_0) \cdot n|$, provided than $|n| = 1$. Let M be the matrix whose ith row is $(x_i - x_0)^{\mathrm{T}}$. Then Mn is the vector of perpendicular distances from points in S to the hyperplane $\mathscr{P}(n, x_0)$, and the total least squares (TLS) problem is to minimize $|Mn|^2$. The eigenvector corresponding to the smallest eigenvalue of the covariance matrix $M^{\mathrm{T}}M$ is the normal to the hyperplane $\mathscr{P}(n, x_0)$ that best approximates the neighbor set S. Therefore, for a point p in two dimensions, the fiber $\pi^{-1}(p)$ contains the eigenvector corresponding to the larger eigenvalue, as well as the vector pointing in the reverse direction.

We note that it is better to use TLS here than *ordinary least squares* (OLS), as the optimal line found by the former method is independent of the parametrization of the points. Also, when the underlying curve is not smooth, we may use TLS to identify points near the singularities by observing when the eigenvalues are close to each other.

Choosing a correct neighborhood set is a fundamental issue in PCD computation and relates to the correct recovery of the lost topology and embedding of the underlying shape. The neighbor set S may contain either the k nearest neighbors to p, or all points within a disc of radius ε. The appropriate value of k or ε depends on local sampling density, local feature size, and noise, and may vary from point to point. It is standard practice to set these parameters empirically [14,15,17], although recent work on automatic estimation of neighborhood sizes seems promising [23]. In our current software, we estimate k for each data set independently. We hope to incorporate automatic estimation into our software in the near future.

3.2. *Approximated $T(X)$*

We now have a sampling $\pi^{-1}(P)$ of the tangent complex $T(X)$, as shown in Fig. 2(b) for our example. This set is discrete and has no interesting topology. The usual approach is to center an *ε-ball* $B_\varepsilon(p) = \{x \mid d(p,x) \leqslant \varepsilon\}$, a ball of radius ε, at each point of $\pi^{-1}(P)$. This approach is based on the assumption that the underlying space is a manifold, or locally flat. Our approximation to $T(X)$ is the union of ε-balls around the fiber points:

$$T(X) \approx \bigcup_{p \in \pi^{-1}(P)} B_\varepsilon(p).$$

Two issues arise, however: first, we need a metric d on $\mathbb{R}^2 \times \mathbb{S}^1$ so that we can define what an ε-ball is, and second, we need to determine an appropriate value for ε.

We define a Euclidean-like metric generally on $\mathbb{R}^n \times \mathbb{S}^{n-1}$ as $ds^2 = dx^2 + \omega^2 d\zeta^2$. That is, the squared distance between the tangent vectors $\tau = (x, \zeta)$ and $\tau' = (x', \zeta')$ is given by

$$d^2(\tau, \tau') = \sum_{i=1}^{n} (x_i - x_i')^2 + \omega^2 \sum_{i=1}^{n} (\zeta_i - \zeta_i')^2,$$

where ω is a scaling factor. Here, the distance between the two directions ζ, $\zeta' \in \mathbb{S}^{n-1}$ is the chord length as opposed to the arc length. The first measure approximates the second quite well when the distances are small, and is also much faster computationally. The choice of the scaling factor ω in our metric depends on the nature of the PCD and our goals in computing the tangent complex. A large value of ω will spread the points of $\pi^{-1}(P)$ out in the angular directions. This is useful for segmenting an object composed of flat pieces, such as the letter 'V'. However, too much separation can lead to errors for smooth curves with high curvature regions, such as an eccentric ellipse. In such regions, the angular separation at neighboring basepoints changes rapidly, yielding points that are further apart in $\pi^{-1}(P)$. In these cases, a smaller value of ω maintains the connectivity of X, while still separating the directions enough to compute the barcodes for $T(X)$. Setting $\omega = 0$ projects the fibers $\pi^{-1}(P)$ back to their basepoints P.

There is, of course, no perfect choice for ε, as it depends not only on the factors described in the previous section, but also on the value of the scale factor ω in the metric. We need to choose ε to be at least large enough so that the basepoints are properly connected when $\omega = 0$. When ω is small, then the starting ε is usually sufficient. When ω is large, the union of ε-balls is less connected, which may be precisely what we want, such as for the letter 'V'. We have devised a rule of thumb for setting ε. Recall that curvature is defined to be $\kappa = \frac{d\varphi}{ds}$, where φ is the tangent angle and s is arc-length along X. Then, two points that are Δx apart in a region with curvature κ have tangent angles roughly $\Delta\varphi \approx \kappa\Delta x$ apart. Since the chord length $\Delta\zeta$ approximates the arc length $\Delta\varphi$ on $\mathbb{S}^1$ for small values, the squared distance between neighboring points in $\pi^{-1}(P)$ is approximately $\Delta x^2(1 + (\omega\kappa)^2)$. So,

$$\varepsilon \approx \frac{\sqrt{\Delta x^2(1 + (\omega\kappa)^2)}}{2}. \qquad (2)$$

3.3. Complex

We now have an approximation to the tangent complex as a union of balls. To compute its topology efficiently, we require a combinatorial representation of this union as a simplicial complex. This simplicial complex $T(P)$ must have the same connectivity as the union of balls, or the same *homotopy type*.

A commonly used complex in algebraic topology is the *Čech complex*. For a set of m points M, the Čech complex looks at the intersection pattern of the union of ε-balls:

$$\mathscr{C}_\varepsilon(M) = \left\{ \operatorname{conv} T \mid T \subseteq M,\ \bigcap_{t \in T} B_\varepsilon(t) \neq \emptyset \right\}.$$

Clearly, the Čech complex is homotopic to the union of balls. Unfortunately, it is also expensive to compute, as we need to examine all subsets of the point set for potentially $\sum_{k=0}^{m} \binom{m}{k} = 2^{m+1} - 1$ simplices. Furthermore, the complex may have high-dimensional simplices even for low-dimensional point sets. If four balls have a common intersection in two dimensions, the Čech complex for the point set will include a four-dimensional simplex.

A common approximation to the Čech complex is the *Rips complex* [24]. Intuitively, this complex only looks at the intersection pattern between pairs of balls, and adds higher simplices whenever all of their lower sub-simplices are present:

$$\mathscr{R}_\varepsilon(M) = \{\operatorname{conv} T \mid T \subseteq M,\ d(s,t) \leqslant \varepsilon,\ s,t \in T\}.$$

Note that $\mathscr{C}_{\varepsilon/2}(M) \subseteq \mathscr{R}_\varepsilon(M)$ for all ε, and that the Rips complex may have different connectivity than the union of balls. The Rips complex is also large and requires $O(\binom{m}{k})$ time for computing k-simplices. However, it is easier than the Čech complex to compute and is often used in practice.

Since we are computing β_0-barcodes in this paper, we only require the vertices and edges in $T(P)$. At this level, the Čech and Rips complexes are identical. For higher dimensional PCDs such as points from surfaces, however, we will need triangles and, at times, tetrahedra, for computing the barcodes. We are therefore examining methods for computing small complexes that represent the union of balls. A potential approach utilizes *α-complexes* $\mathscr{A}_\alpha$, subcomplexes of the *Delaunay complex*, the dual to the Voronoï diagram of the points [25,26]. These complexes are small and geometrically realizable, and their highest-dimensional simplices have the same dimension as the embedding space. We may view our metric $\mathbb{R}^n \times \mathbb{S}^{n-1}$ as a Euclidean metric by first scaling the tangents on $\mathbb{S}^{n-1}$ to lie on a sphere of radius ω. Then, we may compute α-complexes easily, provided we connect the complex correctly in the tangent dimension across the top/bottom boundary. Fig. 2 displays renderings of our space with the correct scaling as well as an α-complex with $\alpha = \varepsilon$. A fundamental problem with this approach, however, is that we need to filter α-complexes by curvature. Currently, we do not know whether this is possible. An alternate but attractive method is to compute the *witness complex* [27]. This complex utilizes a subsample of *landmark points* to compute small complexes that approximate the topology of the underlying ball-set.

3.4. Filtered tangent complex

We now have a combinatorial representation of the tangent complex. We next need to filter the tangent complex using the curvature at the basepoint. Recall that the *curvature* at a point $x \in X$ in direction ζ is $\kappa(x,\zeta) = 1/\rho(x,\zeta)$, where ρ is the radius of the *osculating circle* to X at x in direction ζ. We need to estimate this curvature at each point of $\pi^{-1}(P)$, in order to construct the filtration on $T(P)$ required to compute barcodes. We then assign to each simplex the maximum of the curvatures at its vertices.

Rather than estimating the osculating circle, we estimate the *osculating parabola* as this estimation is computationally more efficient. Two curves $y = f(x)$ and $y = g(x)$ in the plane have *second order contact* at x_0 iff $f(x_0) = g(x_0), f'(x_0) = g'(x_0)$ and $f''(x_0) = g''(x_0)$. So, if X admits a circle of second-order contact, then it also admits a parabola of second-order contact. Consider the coordinate frame centered at $x \in X$ with vertical axis normal to X. Suppose the curvature at x is $\kappa = 1/\rho$, that is, the osculating circle has equation $x^2 + (y - \rho)^2 = \rho^2$. This circle has derivatives $y' = 0$ and $y'' = 1/\rho$ at x. Integrating, we find that the parabola which has second-order contact with this circle, and hence with X, has equation $y = x^2/2\rho$, as shown in Fig. 3.

We again approximate the shape locally using a set of neighborhood points for each point in P. To find the best-fit parabola, we do not utilize the TLS approach as in Section 3.1, as the equations that minimize the perpendicular distance to a parabola are rather unpleasant. Instead, we use OLS which minimizes the vertical distance to the parabola. Naturally, the resulting parabola depends upon the coordinate frame in which the points are expressed. Fortunately, we have already determined the appropriate frame to use in computing the fibers in Section 3.1. Once we compute the fiber at p, we move the nearest neighbors S to a coordinate frame with vertical axis the TLS best-fit normal direction. We set the origin to be x_0 in this coordinate frame (the average of the points in S) although we do not insist that the vertex of the parabola lies precisely there. We then fit a vertical parabola $f(x) = c_0 + c_1 x + c_2 x^2$ as follows. Suppose the collection S of k neighbor points is $S = \{(x_1, y_1), \ldots, (x_k, y_k)\}$. Let

$$A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_k & x_k^2 \end{pmatrix},$$

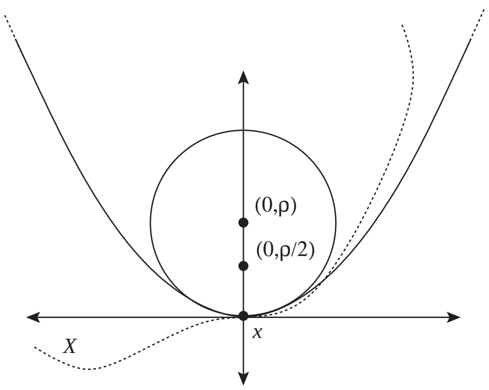

Fig. 3. The osculating circle and parabola to X (dashed) at x. The circle has center $(0, \rho)$, the parabola has focus $(0, \rho/2)$. The curvature of X at x is $\kappa = 1/\rho$.

$$C = (c_0, c_1, c_2)^{\mathrm{T}},$$

$$Y = (y_1, \ldots, y_k)^{\mathrm{T}}.$$

If all points of S lie on f, then $AC = Y$; thus $\eta = AC - Y$ is the vector of errors that measures the distance of f from S. We wish to find the vector C that minimizes $|\eta|$. Setting the derivatives of $|\eta|^2$ to zero with respect to $\{c_i\}$, we solve for C to get

$$C = (A^{\mathrm{T}} A)^{-1} A^{\mathrm{T}} Y.$$

The curvature of the parabola $f(x) = c_0 + c_1 x + c_2 x^2$ at its vertex is $2c_2$, and this is our curvature estimate κ at p. We use this curvature to obtain a filtration of the simplicial complex $T(P)$ that we computed in the last section. This filtered complex approximates $T^{filt}(X)$, the filtered tangent complex described in Section 2.3. For our example, Fig. 2(b) and 2(c) show the fibers $\pi^{-1}(P)$ and union of ε-balls colored according to curvature, using the *hot* colormap.

3.5. Metric space of barcodes

We now have computed a filtered simplicial complex that approximates T^{filt} for our PCD. We next compute the β_0 barcodes using an implementation of the persistence algorithm [22]. Fig. 2(e) shows the resulting β_0-barcode for our sample PCD. As expected, the barcode contains two long intervals, corresponding to the two persistent components of the tangent complex that represent the two tangent directions at each point of a circle. The noise in our PCD is reflected in small intervals in the barcode, which we can discard easily.

To compute the metric, we modify the algorithm that we gave in a previous paper [18] so it is robust numerically. Given two barcodes B_1, B_2, our algorithm computes the metric in three stages. In the first stage, we simply compare the number of half-infinite intervals in the two barcodes and return ∞ in the case of inequality. Note that this makes our measure a quasi-metric. In the second stage, we compute the distance between the half-infinite intervals. This problem now reduces into finding a matching that minimizes the distance between two pointsets, namely the low endpoints of the two sets of half-infinite intervals. We sort the intervals according to their low endpoints and match them one-to-one according to their ranks. Given a matched pair of half-infinite intervals $I \in B_1, J \in B_2$, their dissimilarity is $\delta(I, J) = |\mathrm{low}(I) - \mathrm{low}(J)|$, where $\mathrm{low}(\cdot)$ denotes the low endpoint of an interval.

In the third stage, we compute the distance between finite intervals using a matching problem. Minimizing the distance is equivalent to maximizing the intersection length [18]. We accomplish the latter by recasting the problem as a graph problem. Given sets B_1 and B_2, we define $G(V, E)$ to be a weighted bipartite graph [20]. We place a vertex in V for each interval in $B_1 \cup B_2$. After sorting the intervals, we scan the intervals to compute all intersecting pairs between the two sets [28]. Each pair $(I, J) \in B_1 \times B_2$ adds an edge with weight $|I \cap J|$ to E. Maximizing the similarity is equivalent to the well-known *maximum weight bipartite matching problem*. In our software, we solve this problem with the function `MAX_WEIGHT_BIPARTITE_MATCHING` from the LEDA graph library [29,30]. We then sum the dissimilarity of each pair of matched intervals, as well as the length of the unmatched intervals, to get the distance.

4. Algebraic curves

Having described our methods for computing the metric space of barcodes, we examine our shape descriptor for PCDs of families of algebraic curves. Throughout this section, we use a neighborhood of $k = 20$ points for computing fibers and estimating curvature.

4.1. Family of ellipses

Our first family of spaces are ellipses given by the equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$. We compute PCDs for the five ellipses shown in Fig. 4 with semi-major axis $a = 0.5$ and semi-minor axes b equal to 0.5, 0.4, 0.3, 0.2, and 0.1, from top to bottom. To generate the point sets, we select 50 points per unit length spaced evenly along the x- and y-axis, and then project these samples onto the true curve. Therefore, the points are roughly $\Delta x = 0.02$ apart. We then add Gaussian noise to each point with mean 0 and standard deviation equal to half the inter-point distance or 0.01. For our metric, we use a scaling factor $\omega = 0.1$. To determine an appropriate value for ε for computing the Rips complex, we utilize our rule-of-thumb: Eq. (2) from Section 3.2. The maximum curvature for the ellipses shown is $\kappa_{max} = 50$, so $\varepsilon \approx 0.02\sqrt{1 + 5^2}/2 \approx 0.05$. This value successfully connects points with close basepoints and tangent directions, while still keeping antipodal points in the individual fibers separated.

4.2. Family of cubics

Our second family of spaces are cubics given by the equation $y = x^3 - ax$. The five cubics shown in Fig. 5 have a equal to 0, 1, 2, 3, and 4, respectively. In this case, the portion of the graph sampled is approximately three by three. In order to have roughly the same number of points as the ellipses, we select 15 points per unit length spaced evenly along the x- and y-axis, and project them as before. The points of P are now roughly 0.06 apart. We add Gaussian noise to each point with mean 0 and standard deviation half the inter-point distance or 0.03. For our metric, we use $\omega = 0.5$, primarily for aesthetic

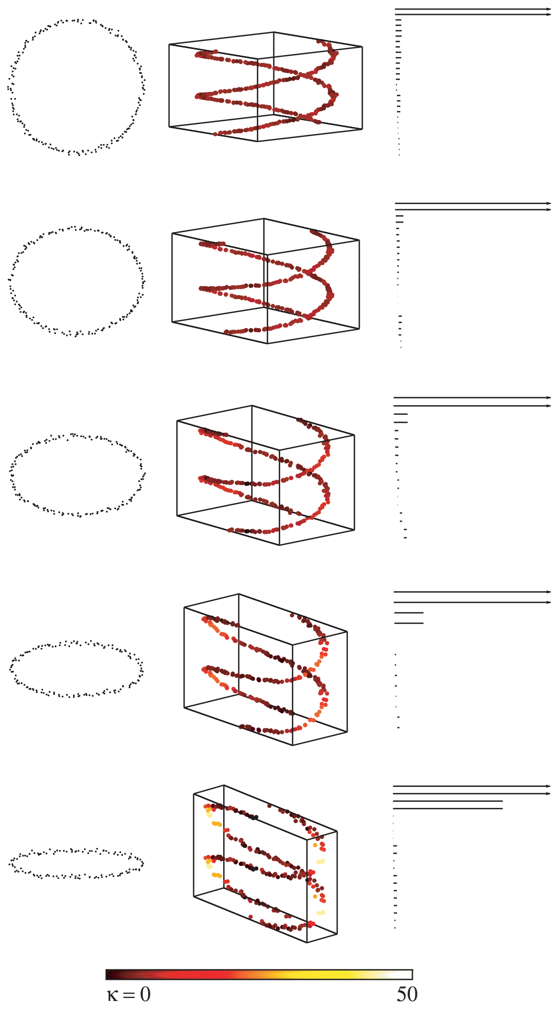

Fig. 4. Family of ellipses: PCD P, fibers $\pi^{-1}(P)$ colored by curvature, and β_0-barcode.

reasons as the fibers are then more spread out. The maximum curvature on the cubics is $\kappa_{max} \approx 8$, and our rule-of-thumb suggests that we need $\varepsilon \approx 0.4$. However, $\varepsilon = 0.2$ is sufficient in this case.

5. Extensions

In Section 3, we assumed that our PCD was sampled from a closed smooth curve in the plane. Our PCDs in the last section, however, violated our assumption as both families had added noise, and the family of cubics featured boundary points. Our method performed quite well, however, and naturally, we would like our method to generalize to other misbehaving PCDs. In this section, we characterize several such phenomena. For each problem, we describe possible solutions that are restrictions of methods that work in arbitrary dimensions. In

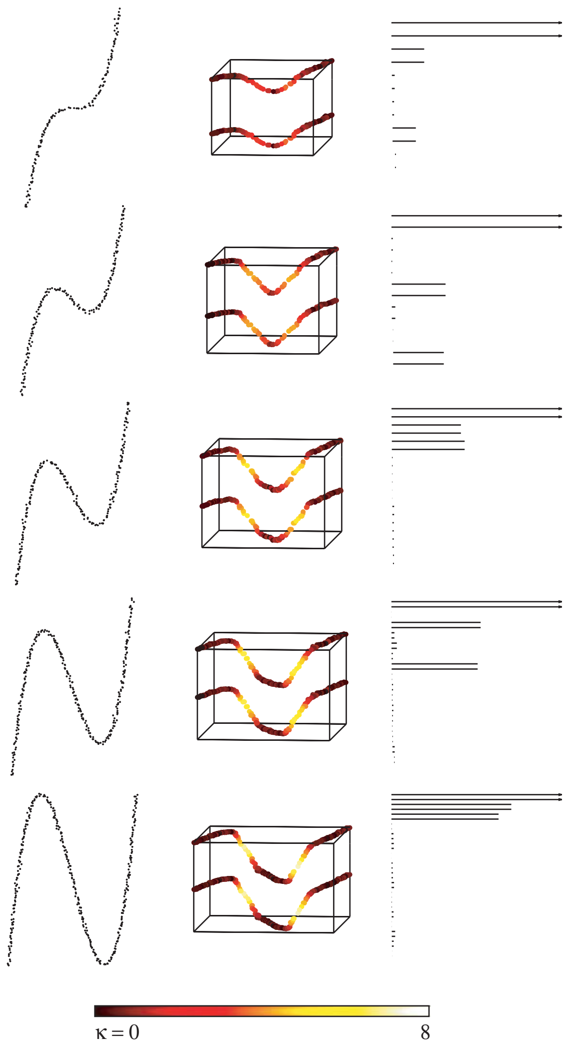

Fig. 5. Family of cubics: PCD P, fibers $\pi^{-1}(P)$ colored by curvature, and β_0-barcode.

the final section, we briefly discuss computing our structures for point cloud data sampled from surfaces embedded in three dimensions.

5.1. Non-manifold points

Suppose that our PCD P is sampled from a geometric object X that is not a manifold. In other words, there are points in the object that are not contained in any neighborhood that can be parametrized by a Euclidean space of some dimension. In the case of curves, a non-manifold point appears at a *crossing*, where two arcs intersect transversally. For example, the junction point of the letter ‘T’ is a non-manifold point. We would like our method to manage nicely in the presence of non-manifold points.

Our approach is to create the tangent complex for P as before, but remove points for which there is no well-defined linear approximation due to proximity to a singular point in X. Such points are identified by a relatively large ratio between the eigenvalues of the TLS covariance matrix constructed for computing fibers in Section 3.1. The point removal effectively segments the tangent complex into pieces. With appropriately large values of ε and ω, we can still connect the remaining pieces correctly.

Fig. 6 shows a PCD for the letter ‘T’. Near the non-manifold point, the tangent direction (height) and curvature (color) estimates deviate from the correct values, and the fiber over the crossing point appears as two rogue points away from the main segments. By removing all points whose eigenvalue ratio is greater than 0.25, we successfully eliminate both rogue and high-curvature points. The gap introduced in the fibers over the crossbar of the ‘T’ is narrower than the vertical (angular) spacing between components. With a well-chosen value of ε, the ε-balls will bridge this gap yet leave four components, as desired. For the images here, we perturbed points 0.01 apart by Gaussian noise with mean 0 and standard deviation 0.005—half the inter-point distance. The tangent complexes are displayed with angular scaling factor $\omega = 0.2$. Balls of radius $\varepsilon = 0.1$ give the correct $T(P)$.

5.2. Singularities

Our PCD may be sampled from a non-smooth manifold. For curves, a non-smooth point typically appear as a “kink”, such as in the letter ‘V’. We say a corner is a *singular point* in the PCD. If the goal is simply to detect the presence of a singular point, then our solution to non-manifold points above—to snip out those points with bad linear approximation—works quite well here, as Fig. 7 displays for the letter ‘V’ and parameters as above.

Sometimes, however, we would like to study a family of spaces that contain singular points to understand shape variability. Since our curvature estimates at a non-smooth point are large, they are included in the filtered tangent complex relatively late, breaking the complex into many components early on. Moreover, the curvature estimates correlate well with the “kinkiness” of the

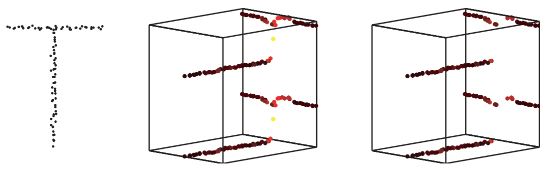

Fig. 6. PCD for the letter ‘T’, all fibers in $\pi^{-1}(P)$, and fibers with eigenvalue ratio less than 0.25.

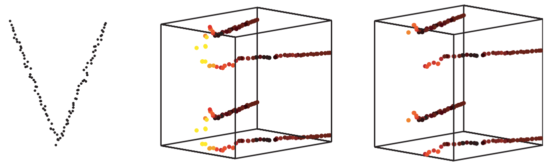
Fig. 7. PCD for the letter 'V', all fibers in $\pi^{-1}(P)$, and fibers with eigenvalue ratio less than 0.25.

singularity, and enable a parametrization of the family, as an example illustrates in the next section. This method extends easily to higher dimensions with higher-dimensional barcodes.

5.3. Boundary points

We may have a PCD sampled from a space with boundary. Counting boundary points of curves could be an effective tool for differentiating between them. Currently, our method does not distinguish boundary points, but simply allows them to get curvature estimates similar to their neighboring points in the PCD, as seen for the shapes in Figs. 5, 6, and 7. We propose a method, however, that distinguishes boundary points via one-dimensional *relative homology*. Around each point p, we may construct $B_\varepsilon(p)$ with its boundary $S_\varepsilon(p)$. For a manifold point, the relative homology group $H_1(B_\varepsilon(p), S_\varepsilon(p))$ has rank 1. Around non-manifold points, the group has rank greater than 1. At a boundary point, the group has rank 0. This strategy would empower our method, for example, to distinguish between the letters 'I' and 'J' with serifs. We plan to implement this strategy in the near future.

5.4. Noise

Our PCD samples may contain noise, which affects our method in two different ways:

(1) Noise may effectively thicken a curve so that it is no longer a one-dimensional object. Once the curve is thick enough, it becomes significantly difficult to compute reliable tangent and curvature estimates.
(2) Noise may also create outliers that disrupt homology calculations by introducing spurious components that result in long barcode intervals that are indistinguishable from genuine persistent intervals.

We resolve the first problem in part by averaging the estimated curvature values over neighborhoods of each point. This averaging smooths the curvature calculations, but does not fix incorrect tangent estimates, which can result in a mis-connected tangent complex. For some real-world datasets, our technique encounters problems in computing reliable tangent estimates. Fig. 8 gives an example of the resulting misconnected tangent complex for a scanned-in number from the MNIST database of handwritten digits [31].

We may resolve the second problem by considering the density of points in the point cloud, and preprocessing the PCD by removing points with low density values. Another strategy which shows promise is to postpone including a point in the filtration of $T(P)$ until it is part of a component with at least k points, for some threshold size k. Not only does this omit singleton outliers, but it also reduces the number and size of the noisy short intervals we see for small κ in our barcodes.

5.5. Surfaces

We end this section with a short discussion on computing tangent complexes for PCDs that are sampled from two-dimensional manifolds or *surfaces*. We have a formal presentation of our techniques for surfaces in our previous paper, where we also analyze several families of mathematical surfaces [18]. Recall that on a curve, there is at most a single osculating circle. On a smooth surface, however, the tangent space at each point is a plane, giving us a whole circle of tangent directions, as shown in Fig. 9. For each point x and direction ζ, there exists an osculating circle. The radius of this circle determines the curvature at that point and direction. As before, we use this curvature to filter the points (x, ζ) in the tangent complex.

In practice, we only have a PCD sampled from a surface and need to approximate the tangent complex using the fibers of the points. At each point, we may approximate the surface locally fitting a quadratic

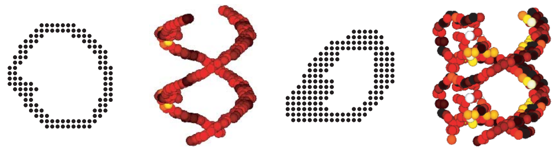
Fig. 8. Two examples of hand printed scanned-in digits '0' from the MNIST Database. We successfully construct $T(P)$ on the left, but the mishappen left side of the right '0' is too thick, resulting in tangent estimate errors and an incorrect $T(P)$.

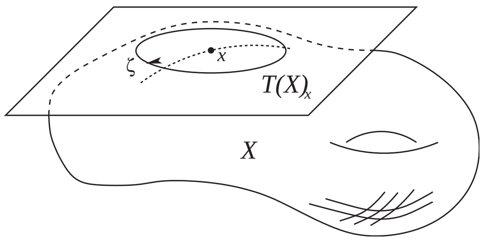

Fig. 9. Surface X with the tangent plane at x and the unit tangent circle $T(X)_x$. We also show a dotted portion of the osculating circle in the direction ζ.

surface, sample the circle of directions, and compute curvatures for those directions. This computation gives us the fibers $\pi^1(P) \subset T(X)$ that sample the tangent complex. We may then compute barcodes for the surface by completing the pipeline outlined in Fig. 2. Observe that the tangent complex for a surface will be embedded in $\mathbb{R}^5$, making the Rips complex rather large and the computation potentially expensive. We are exploring utilizing alternative structures, such as the witness complex [27], as well as alternative techniques to make the computation manageable.

6. Applications

In this section, we discuss the application of our work to shape parametrization and classification. We have implemented a complete system for computing and visualizing filtered tangent complexes, and for computing, displaying, and comparing barcodes.

6.1. Parameterizing a family of shapes

The two families of shapes we saw in Section 4 may be easily parametrized via barcodes. For a family of ellipses

$$\frac{x^2}{a^2}+\frac{y^2}{b^2}=1,$$

with parameters $a \geqslant b > 0$, we can show mathematically that the β_0-barcode consists of two half-infinite intervals

$$\left(\frac{b}{a^2},\infty\right)$$

and two long finite intervals

$$\left(\frac{b}{a^2},\frac{a}{b^2}\right).$$

For fixed a and decreasing b, the intervals should grow longer, and this is precisely the behavior of the barcodes in Fig. 4. Similarly, for a family of cubics with equation $y = x^3 - ax$ parametrized by a, the barcode should contain two half-infinite intervals and four long finite ones, with the exact equations being rather complex [18]. As the parameter a grows in value, the length of the finite intervals should increase. Once again, the barcode captures this behavior in practice as seen in Fig. 5.

An interesting application to shape parametrization is the recovery of the motion of a two-link articulated arm, as shown in Fig. 10. Suppose we have PCDs for the arm at angles from 0 to 90° in 15° intervals, and we wish to recover the sequence that describes the bending motion of this arm. As the figure illustrates, sorting the PCDs by the length of the longest finite interval in their β_0-barcodes recovers the motion sequence. The noise in the data creates many small intervals. The intrinsic shape of the arm, however, is described by the two half-infinite intervals and the two long finite ones.

To illustrate the robustness of our barcode metric, we compute ten random copies of each of the seven articulations and compute the distance between them. Fig. 11 displays the resulting distance matrix, where the distances are computed using the barcode metric of Section 3.5. The distance between each pair of points is mapped to gray-scale, with black corresponding to zero distance. Pairs whose matrix entry is near the diagonal of the matrix are close in the sequence, and consequently have close articulation angles. They are also close in the barcode metric, making the diagonal of the matrix dark.

We generate each arm by placing 100 points 0.02 apart, and perturbing each by Gaussian noise with mean 0 and standard deviation 0.01. We use $\omega = 0.1$ and $\varepsilon = 0.005$ as for the ellipses in Fig. 4. In addition, we utilize the curvature averaging strategy of Section 5.4 using the twenty nearest neighbors to cope with the noise.

6.2. Classifying shapes

To demonstrate the power of our technique for shape classification, we apply it to a collection of hand printed scanned-in letters of the alphabet. We stress that our aim here is not to outperform existing OCR techniques, but to present an instructive example that illustrates the power of our techniques. It is clear that a pure topological classification cannot distinguish between the letters of the alphabet as it partitions the Roman alphabet into three classes based on the number of holes. The letter ‘B’ has two holes, the letters ‘A’, ‘D’, ‘O’, ‘P’, ‘Q’, and ‘R’ have one hole, and the remaining letters have none.

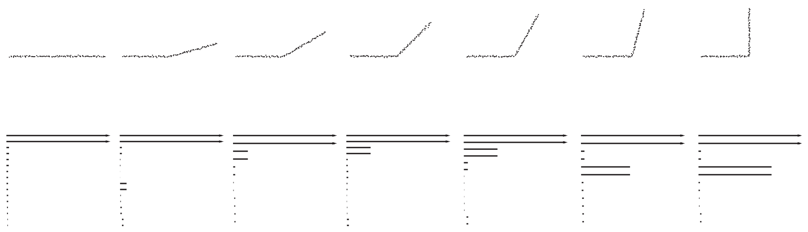
Fig. 10. A bending two-link articulated arm. The β_0-barcodes enable the recovery of the sequence, and hence the motion.

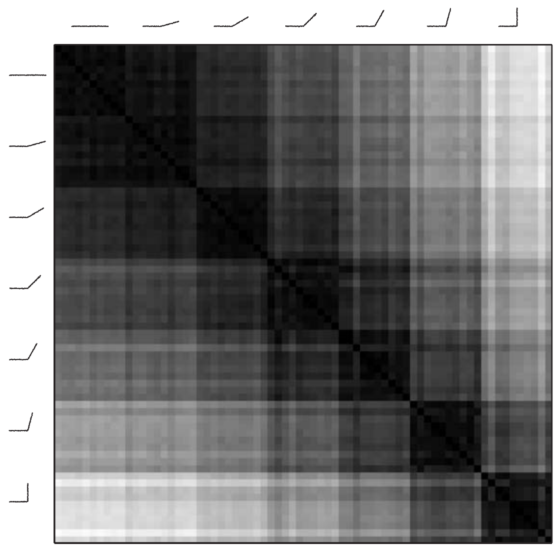
Fig. 11. Distance matrix for the two-link articulated arm in Fig. 10. We have 10 copies of each of the seven articulations of the arm. We map the distance between each pair to gray-scale, with black indicating zero distance.

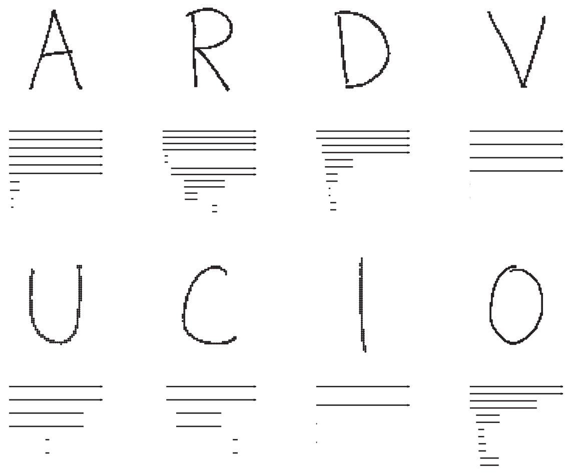

Fig. 12. Hand-drawn scanned-in copies of the letters discussed in Section 6.2, together with their β_0-barcodes.

However, we can utilize the techniques of this paper to glean more information. For example, the letters 'U' and 'V' have the same topology, but 'U' is smooth while 'V' has a kink. This singularity splits the tangent complex for 'V' into four pieces, as in Fig. 7, while the tangent complex for 'U' has only two components. In turn, these components translate into the half-infinite intervals in the β_0-barcodes of the letters: four for 'V', and two for 'U', as shown in Fig. 12. Similarly, although 'O' and 'D' have the same topology, they are distinguishable by the number of half-infinite intervals in their barcodes, with the singularities in 'D' generating two additional intervals.

Even when two letters are both smooth, we may use their curvature information to distinguish between them. For example, the difference in curvature between the letters 'C' and 'I' results in different low endpoints for intervals in their barcodes, as seen in Fig. 12. Even though the tangent complexes for the two letters have the same number of components, our barcode metric can distinguish between them. Finally, we consider the letters 'A' and 'R', both of which have tangent complexes that split into six components that are represented via the six half-infinite intervals in their respective barcodes. Again, the curved portion of 'R' results in a different low endpoint for one pair of half-infinite intervals, and hence a different barcode than for 'A'.

For our experiments, we scan ten hand printed copies of each of the eight letters discussed. Each letter has between 69–294 points, spaced 0.025 apart. We remove points whose eigenvalue ratio is greater than 0.1, as discussed in Section 5.1. We then use $k = 20$ nearest neighbors to estimate the normal direction and curvature at each point, and use ball radius $\varepsilon = 0.2$ and scale factor $\omega = 0.1$ to compute the filtered tangent complex. Fig. 13(a) displays the resulting distance matrix, where distance is mapped as before. The letters are grouped according to the number of infinite components in $T(P)$:6, 4, and 2, respectively. The distance between letters from different groups is infinite, as reflected in the large white regions of the matrix. The matrix illustrates that we may exploit tangent information to distinguish between letters that have the same topology.

6.3. Multiple signatures

In the last section, we showed how tangent information provides additional power for discriminating between letters. It is clear from the resulting matrix in Fig. 13(b), however, that the tangent information, by itself, is insufficient for a full classification. In the tangent domain, the letters 'D' and 'V' look very similar. But we already know how to distinguish between them using a pure topological method: 'D' has one hole, and 'V' has none. So, we must employ information from both domains for classifying letters. We generalize this insight to advocate a framework for distinguishing shapes via *multiple* barcodes. Within our framework, each shape has several associated barcodes derived from different geometric filtrations. For example, we may consider the shape itself, various metrics or Morse functions on the space, or derived complexes, such as the tangent complex. Each barcode captures a different shape invariant. Since we have a single shape descriptor throughout our system, we utilize our metric to find

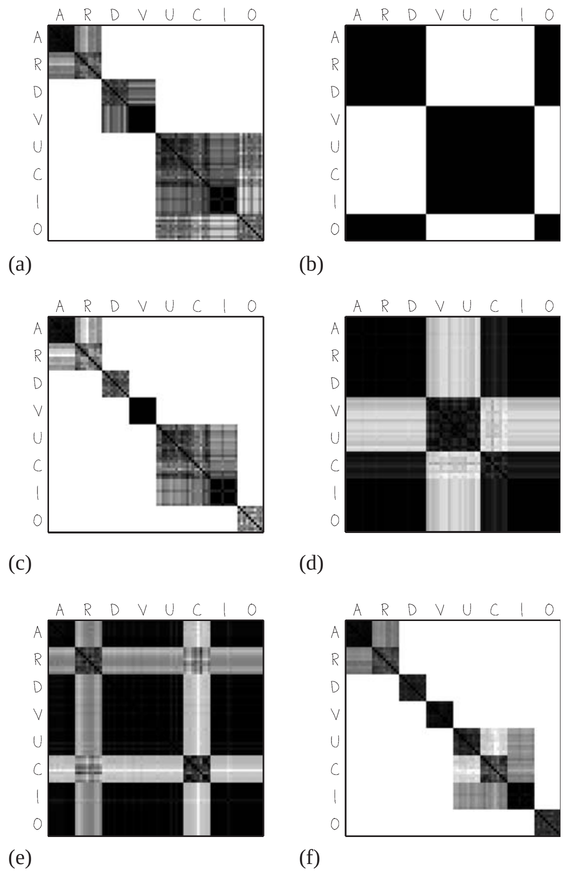

Fig. 13. Distance matrices for 10 scanned instances of the letters ‘A’, ‘R’, ‘D’, ‘V’, ‘U’, ‘C’, ‘I’ and ‘O’. We map the distance between each pair to gray-scale. (a) β_0-barcodes for the tangent complex $T(P)$ filtered by curvature, (b) a mask matrix, where distance is 0 if the letters have the same β_1, and ∞, otherwise, (c) the combination of (a) and (b), (d) β_0 of original space filtered top-down distinguishes ‘U’ from ‘C’, (e) β_0 of original space filtered right-left distinguishes pairs {‘A’, ‘R’} and {‘C’, ‘I’}, (f) the combination of all four distance functions distinguishes all letters.

distances between respective barcodes of two shapes. We then combine the information from the different barcodes for shape comparison.

In the rest of this section, we demonstrate our framework through the letter classification example. We must emphasize, however, that the key aspect of our framework is its generality: it does not rely on detailed ad-hoc analysis of a particular classification problem, but rather on a family of signatures that are potentially useful for solving other problems. Our methods also have a conceptual nature that extend naturally to higher dimensional settings.

We begin by extracting information from the topology of the letters. As already discussed, the letters are partially classifiable using the number of loops or β_1. We compute β_1 for each letter via a simple method that does not require a filtration, although we may employ more robust methods for its calculation. We include all points of the PCD and use balls of radius 0.08 to construct a complex. We then create a *mask* matrix, shown in Fig. 13(b), where the distance between two shapes is 0 if they have identical β_1 and infinite otherwise. Observe that the mask allows us to distinguish ‘D’ from ‘V’, and ‘O’ from ‘U’, ‘C’, and ‘I’. We now apply this mask to the matrix of tangent information in Fig. 13(a) that we obtained in the last section to get Fig. 13(c). Our new matrix classifies ‘D’, ‘V’, and ‘I’ correctly, but still has two blocks that group ‘A’ and ‘R’, and ‘U’, ‘C’, and ‘I’, respectively.

We next examine separating ‘U’ and ‘C’. An important characteristic of our metric is that it is invariant under both small elastic and large rigid motions. Since a ‘C’ is basically a ‘U’ turned on its side, we should not expect any topological method to separate them. However, as the alphabet illustrates, there are situations where it is necessary to distinguish between an object and a rotated version of itself. One way to do so is to employ a directional Morse function and examine the evolution of the excursion sets $X_f = \{(x, y) \in X \mid y > f\}$. As with all our techniques, this method for directional distinction extends in an obvious way to higher dimensional point clouds. In this case, we consider a vertical top-down filtration. Note that $\beta_0(U_f) = 2$ while $\beta_0(C_f) = 1$ for most values of f. Therefore, the corresponding β_0-barcodes allow us to distinguish between ‘U’ and ‘C’, as seen in Fig. 14. This filtration gives us the distance matrix shown in Fig. 13(d).

Our final filtration employs a horizontal right-left Morse function. This filtration sharpens the distinction between the pairs {‘A’, ‘R’} and {‘C’, ‘I’}. We show the resulting β_0-barcodes for representatives of each pair in Fig. 15, and the resulting distance matrix is shown in Fig. 13(e).

Having described our filtrations, we combine the multiple signatures into a single measure, depicted by the distance matrix in Fig. 13(f). This measure is based on the four invariant signatures:

(1) β_0-barcodes of the tangent complex, filtered by curvature, in Fig. 13(a),
(2) β_1 of the letters in Fig. 13(b),

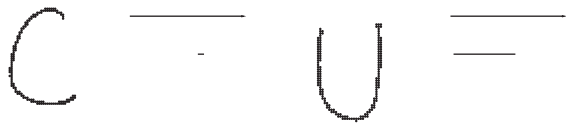

Fig. 14. Filtering the original point clouds from top to bottom gives different β_0 barcodes for ‘C’ and ‘U’.

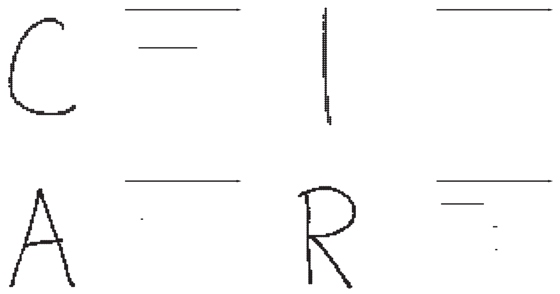
Fig. 15. Filtering the original point clouds from right to left distinguishes between the pairs {‘C’, ‘T’} and {‘A’, ‘R’}.

(3) β_0-barcodes of the letters, filtered top-down, in Fig. 13(d),
(4) β_0-barcodes of the letters, filtered right-left, in Fig. 13(e).

Let $\{M_1, M_2, M_3, M_4\}$ denote the corresponding distance matrices, respectively. Since the distance range varies for each matrix, we re-scale the matrices prior to combining them. Specifically, if max(M) is the maximum value of the finite entries in the matrix M, we set $\lambda_{1,3} = \max(M_1)/\max(M_3)$ and $\lambda_{1,4} = \max(M_1)/\max(M_4)$. Then our combined distance matrix is

$$M = M_1 + M_2 + \lambda_{1,3} \cdot M_3 + \lambda_{1,4} \cdot M_4.$$

By combining all four signatures, we can readily distinguish between the eight letters discussed above. Fig. 13(f) depicts the final result.

7. Conclusion

In this paper we apply ideas of our earlier paper to provide novel methods for studying the qualitative properties of one-dimensional spaces in the plane [18]. Our method is based on studying the connected components of a complex constructed from a curve using its tangential information. Our method generates a compact shape descriptor called a barcode for a given PCD. We illustrate the feasibility of our methods by applying them for classification and parametrization of several families of shape PCDs with noise. We also provide an effective metric for comparing shape barcodes for classification and parametrization. Finally, we discuss the limitations of our methods and possible extensions. An important property of our methods is that they are applicable to any curve PCD without any need for specialized knowledge about the curve. The salient feature of our work is its reliance on theory, allowing us to extend our methods to shapes in higher dimensions, such as families of surfaces embedded in $\mathbb{R}^3$, where we utilize higher-dimensional barcodes.

Our work suggests a number of enticing research directions:

- Implementing density estimation techniques to remove spurious components arising from noise,
- A systematic study of the thickness problem of scanned curves,
- Implementing the strategy for identifying boundary points, further strengthening our method,
- Applying our methods to surface point cloud data.

It is also potentially very useful to study higher dimensional persistence, that is, situations where a complex is filtered by two or more variables. In this situation, the classification of multiply filtered vector spaces is much more difficult than the single variable persistence. Indeed, we do not expect to find a complete classification in terms of simple combinatorial data like a barcode. However, we still hope to find interesting computable invariants that would be quite useful in studying shape recognition.

We should note that an eventual goal of this work is automatic shape classification via learning algorithms. Our approach replaces the raw shape data with information-rich barcodes that contain geometric and topological invariants. Learning algorithms could then discover significant intervals within a barcode, recover parameterizations, and provide proper weights for combining barcodes derived from different filtrations. We believe the combination of our techniques with learning theory represents an exciting new avenue of research.

References

[1] Duda RO, Hart PE, Stork DG. Pattern classification and scene analysis, 2nd ed. Hoboken, NJ: Wiley-Interscience; 2000.
[2] Gonzalez RC, Woods RE. Digital image processing, 2nd ed. Boston, MA: Addison-Wesley; 2002.
[3] Sebastian T, Klein P, Kimia B. Recognition of shapes by editing shock graphs. In: International Conference on Computer Vision; 2001. p. 755–62.
[4] Fan T-J. Describing and recognizing 3D objects using surface properties. New York, NY: Springer; 1990.
[5] Fisher RB. From surfaces to objects: computer vision and three-dimensional scene analysis. New York, NY: John Wiley; 1989.
[6] Osada R, Funkhouser T, Chazella B, Dobkin D. Matching 3D models with shape distributions. In: International Conference on Shape Modeling & Application; 2001. p. 154–66.
[7] Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the ACM SIGGRAPH 2001; 2001. p. 203–12.

[8] Levoy M, Whitted T. The use of points as display primitives, Tech. rep., The University of North Carolina at Chappel Hill, 1985.
[9] Adamson A, Alexa M. Ray tracing point set surfaces. In: Proceedings of Shape Modeling International; 2003.
[10] Rusinkiewicz S, Levoy M. QSplat: a multiresolution point rendering system for large meshes. In: Proceedings of ACM SIGGRAPH 2000; 2000. p. 343–52.
[11] Alexa M, Behr J, Cohen-Or D, Fleishman S, Levin D, Silva CT. Point set surfaces. In: Proceedings of Vision; 2001.
[12] Zwicker M, Pauly M, Knoll O, Gross M. Pointshop 3D: an interactive system for point-based surface editing. In: Proceedings of ACM SIGGRAPH 2002, vol. 21; 2002. p. 322–9.
[13] Adams B, Dutré P. Interactive boolean operations on surfel-bounded solids. ACM Trans Graph 2003;22(3):651–6.
[14] Pauly M, Keiser R, Kobbelt LP, Gross M. Shape modeling with point-sampled geometry. In: Proceedings of ACM SIGGRAPH 2003, vol. 22; 2003. p. 641–50.
[15] Donoho DL, Grimes C. Hessian eigenmaps: locally linear embedding techniques for high-dimensional data. PNAS 2003;100:5591–6.
[16] Lee AB, Pedersen KS, Mumford D. The non-linear statistics of high contrast patches in natural images, Tech. rep., Brown University, Available online, 2001.
[17] Tenenbaum JB, de Silva V, Langford JC. A global geometric frame-work for nonlinear dimensionality reduction. Science 2000;290:2319–23.
[18] Carlsson G, Zomorodian A, Collins A, Guibas L. Persistence barcodes for shapes. In: Proceedings of Geometry Process; 2004. p. 127–38.
[19] Munkres JR. Elements of algebraic topology. Red-wood City, California: Addison-Wesley; 1984.
[20] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to algorithms. Cambridge, MA: MIT Press; 2001.
[21] Edelsbrunner H, Letscher D, Zomorodian A. Topological persistence and simplification. Discrete Computational Geometry 2002;28:511–33.
[22] Zomorodian A, Carlsson G. Computing persistent homology. Discrete Computational Geometry, to appear.
[23] Mitra NJ, Nguyen A, Guibas L. Estimating surface normals in noisy point cloud data. International Journal of Computational Geometry and Applications 2004, to appear.
[24] Gromov M. Hyperbolic groups. In: Gersten S editor. Essays in group theory. New York, NY: Springer; 1987. p. 75–263.
[25] de Berg M, van Kreveld M, Overmars M, Schwarzkopf O. Computational geometry: algorithms and applications. New York: Springer; 1997.
[26] Edelsbrunner H. The union of balls and its dual shape. Discrete Computational Geometry 1995;13:415–40.
[27] de Silva V, Carlsson G. Topological estimation using witness complexes. In: Proceedings of Symposium on Point-Based Graphics; 2004. p. 157–66.
[28] Zomorodian A, Edelsbrunner H. Fast software for box intersection. International Journal of Computational Geometry and Applications 2002;12:143–72.
[29] Algorithmic Solutions Software GmbH, LEDA, 2004.
[30] Mehlhorn K, Näher S. The LEDA platform of combinatorial and geometric computing. Cambridge, UK: Cambridge University Press; 1999.
[31] LeCun Y. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/.