

Branching and Circular Features in High Dimensional Data

Bei Wang, Brian Summa, Valerio Pascucci, and Mikael Vejdemo-Johansson

UUSCI-2011-005

Scientific Computing and Imaging Institute University of Utah Salt Lake City, UT 84112 USA

May 30, 2011

Abstract:

Large observations and simulations in scientific research give rise to high-dimensional data sets that present many challenges and opportunities in data analysis and visualization. Researchers in the application domains such as engineering, computational biology, climate study, imaging and motion capture are faced with the problem of how to discover compact representations of high-dimensional data while preserving their intrinsic structure. In many applications, the original data is projected onto low-dimensional space via dimensionality reduction techniques prior to modeling. One problem with this approach is that the projection step in the process can fail to preserve structure in the data that is only apparent in high dimensions. Conversely, such techniques may create structural illusions in the projection, implying structure not present in the original high-dimensional data. Our solution is to utilize topological techniques to recover important structures in high-dimensional data that contains non-trivial topology. Specifically, we are interested in two types of features in high dimensions: local branching structures and global circular structures. We construct local and global circle-valued coordinate functions to represent such features. Subsequently, we perform dimensionality reduction on the data while ensuring such structures are visually preserved. Our results reveal never-before-seen structures on real-world data sets from a variety of applications.



Branching and Circular Features in High Dimensional Data

Bei Wang, Brian Summa, Valerio Pascucci, Member, IEEE, and Mikael Vejdemo-Johansson



Fig. 1. Given point cloud data in high-dimensional space, we detect and visualize branching structures in a neighborhood surrounding a given point of interest. Here we give two simple examples with point clouds sampled from surfaces embedded in 3-dimensional space. In (a), given a genus-3 surface, we analyze the branching structure around one of its corners, x. We apply color-mapping transfer functions to local circle-valued coordinate functions to visualize the structure. Specifically, the color scale indicates the "direction" of the branches. As illustrated in (b), there is a local two-way branching around x, where the coordinate function of each branch is visualized in (e) and (f), respectively. In (c), given a genus-4 surface, we detect a seven-way branching around x (d), where three of the coordinate functions are shown in (g), (h) and (i), respectively.

Abstract— Large observations and simulations in scientific research give rise to high-dimensional data sets that present many challenges and opportunities in data analysis and visualization. Researchers in the application domains such as engineering, computational biology, climate study, imaging and motion capture are faced with the problem of how to discover compact representations of high-dimensional data while preserving their intrinsic structure. In many applications, the original data is projected onto low-dimensional space via dimensionality reduction techniques prior to modeling. One problem with this approach is that the projection step in the process can fail to preserve structure in the data that is only apparent in high dimensional data. Our solution is to utilize topological techniques to recover important structures in high-dimensional data that contains non-trivial topology. Specifically, we are interested in two types of features in high dimensions: local branching structures and global circular structures. We construct local and global circle-valued coordinate functions to represent such features. Subsequently, we perform dimensionality reduction on the data while ensuring such structures are visually preserved. Our results reveal never-before-seen structures on real-world data sets from a variety of applications.

Index Terms— Dimensionality reduction, circular coordinates, visualization, topological analysis.

1 INTRODUCTION

Many scientific investigations depend on exploratory data analysis and visualization of high-dimensional data sets that represent complex phenomena. Given a collection of high-dimensional data points, dimensionality reduction techniques are typically applied prior to modeling and feature detection. These techniques find a low-dimensional representation of the data with simple guarantees, by assuming that real-valued low-dimensional coordinates are sufficient to capture its underlying intrinsic structure.

- Bei Wang, Brian Summa, and Valerio Pascucci are with SCI Institute, University of Utah, E-mails: beiwang@sci.utah.edu, bsumma@sci.utah.edu, pascucci@sci.utah.edu.
- Mikael Vejdemo-Johansson is with Stanford University, E-mail: mik@math.stanford.edu.

In mathematical terms, given a collection of high-dimensional data points $X \in \mathbb{R}^d$, dimensionality reduction techniques obtain an embedding that maps a point $x = (x_1, x_2, ..., x_d) \in X$ to a point $y = (y_1, y_2, ..., y_m)$, where $m \ll d$, through a set of real-valued coordinate functions $\phi = (\phi_1, \phi_2, ..., \phi_m) : X \to \mathbb{R}$, where $y_i = \phi_i(x)$, with the assumption that the data typically has the topological structure of a convex domain [13]. However, if the underlying space in high dimensions contains nontrivial topology, either globally or locally, dimensionality reduction alone is no longer sufficient to preserve the topology. In particular, we consider two types of topological features in high dimension: local branching and global circular structures. We use topological techniques to construct local and global circle-valued coordinate functions that represent such features. We then ensure they are visually preserved in the low dimensional projection.

The authors in [13] challenge the convex domain assumption in dimensionality reduction through topological analysis. They consider data sets whose underlying spaces contain circular structures, such as circle, annulus and torus. They describe a topological procedure that enlarges the class of coordinate functions to include global circle-valued functions, that map the point cloud to a closed circle, $\theta : X \to \mathbb{S}^1$. We adapt and build upon this work to construct local circle-valued coordinate functions on a subset of points $U \subseteq X$, that encode branching structures in data. Formally, we compute functions that map subsets of points onto a closed circle, $\theta' : U \to \mathbb{S}^1$. We visualize these local circular coordinate functions by applying a color map transfer function.

The advantage with local and global circle-valued coordinate functions is two-fold. First, they enrich data representations by revealing branching and circular features in the data. Second, they differentiate intrinsic structures of the data from structural illusions.

For example, for a point cloud sampled from a torus embedded in 2D as shown in Figure 2 top, dimensionality techniques alone can always visualize one of its essential loops represented by θ_1 , while fail to showcase the other essential loop revealed by θ_2 without tearing or cutting. In Figure 2 bottom, global circular-valued coordinate functions differentiate a trefoil knot from two closed circles based upon seemly similar projections.

On the other hand, local circle-valued coordinate functions reveal topological features within a sub-region of the point cloud, as shown in Figure 3. It is able to capture the three-way branching structure surrounding the crossing point in figure eight (Figure 3 bottom), while detecting structural illusion of a figure eight created by projecting a circle in a certain direction (Figure 3 top).

Our main contributions are as follows.

- We introduce local circle-valued coordinate functions that facilitate local structural analysis, especially the detection of branching features in data. We construct these functions in a local neighborhood through topological analysis of 1-dimensional cohomology. That is, we choose a subset of points U that are with close proximity of a given point, and construct coordinate functions $\theta: U \to \mathbb{S}^1$.
- On the technical level, we develop a local version of the persistent cohomology machinery through local cohomology computed on point cloud data. Persistence enables the detection of significant local features and separates features from noise within the data. That is, we obtain parameterization of *U* through coordinate functions $\theta_1, \theta_2, ..., \theta_n : U \to S^1$, where *n* indicates the number of significant local features.
- We present the first technique that approximates topological circular and branching structures in high-dimensional space to aid visualization in the low-dimensional projection.
- We present empirical evidence demonstrating that both the local and global circle-valued coordinate functions, for the first time, permit more precise analysis on a large collection of real-world data sets.

2 RELATED WORK

Various algorithms have been proposed to compute loops on surfaces, or homology generators that satisfy certain geometric optimality [27, 9, 38, 23]. [27] computes shortest set of homology generators for 2-manifolds. [16] uses topological persistence [22] to computes topologically correct loops on surfaces, that wrap around their "handles" and "tunnels". Given a weighted simplicial complex and a nontrivial cycle, [15] computes its homologous cycle with minimal weight. [11] approximates a shortest basis of the one dimensional homology group of a manifold in \mathbb{R}^d from its point sample. Algorithms have also been developed to compute shortest cycles, minimum cuts, or maximum flow related to graphs embedded on surfaces [26, 28, 25, 24].

In terms of revealing circular structures or essential loops within data, several approaches have been taken to find alternative representations. [17] studies cylindrical manifolds - data whose generative model includes a cyclic and a linear parameter, and tries to find embedding functions that map them onto a cylinder $\mathbb{S}^1 \times \mathbb{R}$. [33] projects



Fig. 2. Visualized global circular coordinate functions. Top: two global circle-valued coordinate functions for a point cloud sampled from a torus. Top left: $\theta_1 : X \to \mathbb{S}^1$. Top right: $\theta_2 : X \to \mathbb{S}^1$. Bottom left, projection of a trefoil knot; bottom middle and right, projection of two closed circles. Figures are reproductions from [13].



Fig. 3. Visualized local circle-valued coordinate function. Top left: local circle-valued coordinate function for a point cloud sampled from a torus. Top right: projection of a circle on 2D that gives an illusion of a figure eight, local circle-valued coordinate function indicates there is no local branching structure. Bottom: projection of a figure eight on 2D, three circle-valued coordinate functions are visualized to describe the local branching structures.

data with non-trivial topology by destroying essential loops via tearing and cutting. [36] maps data to a pre-chosen non-flat target space, such as a cylinder or a sphere, using multidimensional scaling. The work in [13] represents the original topology with significant circular structures without tearing or cutting, and gives a number of circlevalued coordinate functions determined experimentally through persistence. Recent work in [5] defines a modified version of topological persistence (level persistence) for 1-cocycles, and shows that such 1-cocycles can be interpreted as a circle valued map. We notice similarities between [13] and [5], however they use different notions of persistence.

Algorithms that focus on cohomology computation, especially persistent cohomology, has been proposed in recent years [13, 5]. [19, 18] designs efficient algorithm to compute cohomology basis, with applications in computational electromagnetic. [14] addresses duality in persistent homology and cohomology computation, while [10] compares efficiencies of these algorithms. Local persistent homology has been used in stratification learning [2, 3].

Our work is the first that constructs local circle-valued coordinates on high-dimensional data to detect branching structures. We discover and visualize topological structures such as circles and branches on some data sets that have never been realized before.

3 TECHNICAL BACKGROUND

We now introduce several key ingredients (algebraic and algorithmic) behind our algorithm. Our work deals with homology and cohomology groups, the art of counting "cycles" and "pseudo-cycles" in a topological space. Cohomology groups are algebraically "dual" to the homology groups, while less geometrical, they are important in theory and practice. According to Bott and Tu [4], "one of the hallmarks of a topologist is a sound intuition for the coboundary operator." We review necessary background on these concepts for non-specialists, with intuitive examples. We then describe local cohomology that is important in our branching structural analysis. For a readable mathematical introduction, see [35, 31] for algebraic topology and [21] for persistent homology.

3.1 Homology and cohomology

Homology. Homology deals with topological features such as "holes" or "cycles"; 0-, 1- and 2- dimensional homology groups correspond to components, tunnels and voids in a topological space. Here, we discuss its simplest and most concrete definition, at the level of simplicial homology.

Consider the simplicial complex *K* pictured in Figure 4, which is a triangulation of an annulus. The 0-, 1- and 2-simplexes in *K* are the vertices, edges and triangles, denoted by the sets $K^0 = \{v_i\}, K^1 = \{e_i\}$ and $K^2 = \{\Delta_i\}$ respectively. We assume all simplices are arbitrarily oriented. Let $G = \mathbb{Z}$ be the group of integers. We define the 0-chains, 1-chains and 2-chains as formal sums of 0-, 1- and 2-simplexes with integer coefficients, respectively,

$$C_0 = C_0(K;\mathbb{Z}) = \{b = \sum g_i v_i \mid g_i \in \mathbb{Z}\},\$$

$$C_1 = C_1(K;\mathbb{Z}) = \{a = \sum g_i e_i \mid g_i \in \mathbb{Z}\},\$$

$$C_2 = C_1(K;\mathbb{Z}) = \{c = \sum g_i \Delta_i \mid g_i \in \mathbb{Z}\}.$$

By abuse of notation, each 0-, 1- and 2-simplex in *K* corresponds to an *elementary chain* of the same dimension. Then 0-, 1- and 2-chains can be considered as sums of elementary chains. Here, 0-chain $b_1 =$ $v_3 + v_4 + v_5$ (green). 1-chain $a_1 = e_1 + e_2 + e_3 + e_4 + e_5 + e_6 + e_7 + e_8$ (red). 2-chain $c_1 = \Delta_1 + \Delta_2$ (pink). We now define *boundary maps*, $\partial_2 : C_2 \rightarrow C_1$ and $\partial_1 : C_1 \rightarrow C_0$. Representing an oriented *p*-simplex by its vertices $[v_0, ..., v_p]$, we have,

$$\begin{aligned} \partial_2([v_0, v_1, v_2]) &= [v_1, v_2] - [v_0, v_2] + [v_0, v_1]. \\ \partial_1([v_0, v_1]) &= v_1 - v_0. \end{aligned}$$

It is easy to verify that $\partial \circ \partial = 0$. For example, $\partial(\Delta_1) = \partial([v_4, v_5, v_6]) = [v_5, v_6] - [v_4, v_6] + [v_4, v_5]$, $\partial([v_4, v_5]) = v_5 - v_4$, $\partial \circ \partial(\Delta_1) = 0$. Let $a \in C_1$. *a* is a 1-cycle if $\partial a = 0$. It is a 1-boundary if it is the boundary of some 2-chain *c*, that is, $\partial(c) = a$. Since the 1-boundaries are always 1-cycles, im $\partial_2 \subseteq \ker \partial_1$. The 1-homology of *K* is the quotient group, $H_1 = H_1(K; \mathbb{Z}) = \ker \partial_1 / \operatorname{im} \partial_2$. For example, a_1 (red) is a 1-cycle since $\partial(a_1) = 0$. a_2 (cyan) is a 1-boundary since it is the boundary of the 2-chain c_1 . a_1 is a 1-cycle, but not a 1-boundary, therefore $a_1 \in H_1$. It can be used as a representative of the homology class that generates the first homology group of *K*. Two elements $a, a' \in H_1$ are homologous iff $a - a' = \partial c$, for some 2-chain *c*, denoted as $a \sim a'$. Here $a_1(red) \sim a_3(orange)$.

Consider the torus, in a triangulation *K* in Figure 6 top left, its 1homology group is generated by the 1-chains a_1 (red) and a_2 (blue), that is, $a_1 = [a,b] + [b,c] + [c,a]$ and $a_2 = [a,d] + [d,e] + [e,a]$. We can verify that both a_1 and a_2 are 1-cycles, as $\partial(a_1) = \partial(a_2) = 0$. They are not 1-boundaries since neither $\partial b = a_1$ nor $\partial b = a_2$ admit a solution $b \in C_2$. In addition, a_1 and a_2 are not homologous.

Cohomology. Now we associate K with another sequence of groups called *cohomology groups*, whose origins lie in algebra rather than geometry [35]. In many ways, they are considered "dual" to homology groups, and are important in practice. Following our previous introduction to homology groups, we bring intuitions to cohomology via simple examples.

Consider our example in Figure 4, cohomology deals with functions (more precisely, homomorphisms) on 0-, 1- and 2-chain groups. By



Fig. 4. The triangulation of an annulus. 1-chain $a_1 = e_1 + e_2 + ... + e_8$ (red) is a generator of H₁.

abuse of notation, each 0-, 1- and 2-dimensional simplex in *K* corresponds to an *elementary cochain* of the same dimension. For example, 1-simplex *e* has a corresponding elementary 1-cochain e^* , which is a function on 1-chain whose value is 1 on *e* and 0 on all other edges. In other words, $e^* : C_1 \to \mathbb{Z}$, where $e^*(e) = 1$ and $e^*(e') = 0$ for all $e' \in K^1, e' \neq e$. Similarly, we have elementary 0-cochains, v^* associated with 0-simplex *v*; and elementary 2-cochains Δ^* associated with 2-simplex Δ . Then 0-, 1- and 2-cochains can be considered as sums of elementary cochains, that is,

$$C^{0} = C^{0}(K;\mathbb{Z}) = \{\beta : C_{0} \to \mathbb{Z}, \beta = \sum g_{i}v_{i}^{*} \mid g_{i} \in \mathbb{Z}\},\$$

$$C^{1} = C^{1}(K;\mathbb{Z}) = \{\alpha : C_{1} \to \mathbb{Z}, \alpha = \sum g_{i}e_{i}^{*} \mid g_{i} \in \mathbb{Z}\},\$$

$$C^{2} = C^{1}(K;\mathbb{Z}) = \{\gamma : C_{2} \to \mathbb{Z}, \gamma = \sum g_{i}\Delta_{i}^{*} \mid g_{i} \in \mathbb{Z}\}.$$

We then define the *coboundary maps*, $\delta_0 : C^0 \to C^1$ and $\delta_1 : C^1 \to C^2$,

$$\begin{aligned} (\delta_0\beta)([v_0,v_1]) &= \beta(v_1) - \beta(v_0), \\ (\delta_1\alpha)([v_0,v_1,v_2]) &= \alpha([v_1,v_2]) - \alpha([v_0,v_2]) + \alpha([v_0,v_1]). \end{aligned}$$

These notations are convenient in computing coboundaries. For example, If $\alpha = \sum g_i e_i^*$, then $\delta(\alpha) = \sum g_i (\delta e_i^*)$. To compute δe^* for each oriented simplex *e*, we have $\delta e^* = \sum \varepsilon_j \Delta_j^*$, where the summation extends over all Δ_j having *e* as a face, and $\varepsilon_j = \pm 1$ is the sign with which *e* appears in the expression for $\partial \Delta_j$. Similar rule applies to computing δv^* .

Let $\alpha \in C^1$, it is a 1-cocycle if $\delta_1(\alpha) = 0$. It is a 1-coboundary if there exists a $\beta \in C^0$ such that $\delta_0(\beta) = 0$. It is easy to verify that $\delta \circ \delta = 0$. 1-coboundaries are always 1-cocycles, we have $\operatorname{im}(\delta_0) \subseteq \operatorname{ker}(\delta_1)$. We define the 1-cohomology of K to be the quotient group, $\mathsf{H}^1 = \mathsf{H}^1(K;\mathbb{Z}) = \operatorname{ker}(\delta_1)/\operatorname{im}(\delta_0)$. Two 1-cocycles α and α' are cohomologous if $\alpha - \alpha'$ is a coboundary.

In Figure 5 (left), assuming all triangles are oriented counterclockwise, we compute δe_5^* . $e_5^* : C_1 \to \mathbb{Z}$ has value 1 on e_5 and 0 on other edges. δe_5^* has value -1 on Δ_1 and 1 on Δ_2 , because e_5 appears in $\partial \Delta_2$ and $\partial \Delta_1$ with signs +1 and -1, respectively. Therefore, $\delta e_5^* = \Delta_2^* - \Delta_1^*$. A similar remark shows that $\delta v_1^* = e_2^* - e_1^*$ and $\delta v_3^* = e_3^* - e_2^* - e_5^*$. The 1-cochain $\alpha = e_1^* + e_5^* - e_3^*$ is a 1-cocycle since $\delta(\alpha) = \delta(e_1^*) + \delta(e_5^*) - \delta(e_3^*) = (\Delta_1^*) + (\Delta_2^* - \Delta_1^*) - (\Delta_2^*) = 0$. Meanwhile, α is also a 1-coboundary since $\alpha_1 = \delta(-v_1^* - v_3^*)$. In terms of generators, in 5 (right), the 1-chain $\alpha_1 = e_6^* + e_7^* + e_8^* + e_9^* + e_{10}^*$ is a 1-cocycle, since $\delta(\alpha_1) = \delta(e_6^*) + ... + \delta(e_{10}^*) = \Delta_3^* + (\Delta_4^* - \Delta_3^*) + (\Delta_5^* - \Delta_4^*) + (\Delta_6^* - \Delta_5^*) - \Delta_6^* = 0$. It is not a 1-coboundary. Therefore, $\alpha_1 \in H^1$, and it can be used as the representative of the 1-st cohomology class. α_1 (red) is cohomologous to α_2 (orange), as we can check $\alpha_1 - \alpha_2 = \delta(v_4^* + v_5^* + v_6^*)$.

Consider the torus example in Figure 6 bottom, its 1-cohomology group is generated by the 1-cochains α_1 (red) and α_2 (blue). We can



Fig. 5. Left: simple examples of cochains. Right: the triangulation of an annulus, 1-cochain $\alpha_1 = e_6^* + e_7^* + e_8^* + e_9^* + e_{10}^*$ is a generator of H¹.

check that both α_1 and α_2 are 1-cocycles, not 1-coboundaries, and are not cohomologous. It is important to note the duality between cohomology and homology generators, which is slightly counter-intuitive. Here, $\alpha_1 \in H^1$ (red) is dual to $a_1 \in H_1$ (red), while $\alpha_2 \in H^1$ (blue) is dual to $a_2 \in H_1$ (blue).



Fig. 6. The triangulation of a torus. Top: $a_1 = [a,b] + [b,c] + [c,a]$ (red) and $a_2 = [a,d] + [d,e] + [e,a]$ (blue) are the generators of H₁. Bottom: 1-cochains α_1 (red) and α_2 (blue) are generators of H¹.

3.2 Local cohomology

The notation $H^1(\mathbb{X}, \mathbb{Y})$ is commonly referred to as *relative cohomology*, which is the computation of the cohomology groups of quotient space \mathbb{X}/\mathbb{Y} . Intuitively, imaging gluing all points in \mathbb{Y} to a dummy vertex *w* at infinity, any non-trivial topology within \mathbb{Y} is destroyed, therefore $H^1(\mathbb{X}, \mathbb{Y})$ only cares about topological features that are in \mathbb{X} and not in \mathbb{Y} .

Suppose we have a topological space \mathbb{X} embedded in \mathbb{R}^N , the 1-*dimensional local homology group* of \mathbb{X} at a point $x \in \mathbb{X}$ is defined as $H_1(\mathbb{X}, \mathbb{X} - x)$ [35]. We define the 1-*dimensional local co-homology groups* as the (vector-space) dual of the local homology, that is, $H^1(\mathbb{X}, \mathbb{X} - x)$. Taking a small enough radius r and using excision, the above local homology groups in question are in fact $H_1(\mathbb{X} \cap B_r(x), \mathbb{X} \cap \partial B_r(x))$, where $B_r(x)$ and $\partial B_r(x)$ denote a ball of radius r centered at x and its boundary. We therefore obtain our local cohomology as $H^1(\mathbb{X} \cap B_r(x), \mathbb{X} \cap \partial B_r(x))$. (Note that we use the term local cohomology differently from the traditional concept introduced by Alexander Grothendieck [30].) $H^1(\mathbb{X} \cap B_r(x), \mathbb{X} \cap \partial B_r(x))$

computes topological features of X within a local neighborhood $B_r(x)$, hence the term "local cohomology".

To put the above formal definition into context, see Figure 7 (left). The space \mathbb{X} is an annulus. Given a point $x \in \mathbb{X}$ and a radius *r*, we draw a ball of radius *r* around *x*. The space that is inside the ball is $\mathbb{X} \cap B_r(x)$ (pink shaded region), and the space that is on the boundary is $\mathbb{X} \cap \partial B_r(x)$ (black). We therefore compute $H^1(\mathbb{X} \cap B_r(x), \mathbb{X} \cap \partial B_r(x))$.



Fig. 7. Local homology and cohomology simple example. Left: computing local (co)homology through coning operation. Right: illustration of the coning operation.



Fig. 8. Top: local cohomology computation under simplicial setting for annulus indicating 0-way branching. Bottom: local cohomology computation under simplicial setting for figure eight indicating 3-way branching.

3.3 Homotopy theory

We rely on the following principle from homotopy theory, which relates circular coordinates with cohomology. Let $[X, \mathbb{S}^1]$ be the set of equivalence classes of continuous maps from space X to \mathbb{S}^1 . Let $H^1(X;\mathbb{Z})$ be the group of 1-dimensional cohomology classes with integer coefficients. For topological spaces with the homotopy type of a cell complex, there is an isomorphism $H^1(X;\mathbb{Z}) \cong [X, \mathbb{S}^1]$ [31]. This implies that *if* X has non-trivial 1-dimensional cohomology class $\alpha \in H^1(X;\mathbb{Z})$, we can construct a continuous function $\theta : X \to \mathbb{S}^1$ from α (see [13] for a formal proof).

Suppose we represent our point cloud data X with a simplicial complex K that contains vertices, edges and triangles. In a nutshell, 1-dimensional cohomology classes are functions that map a collection of edges in K to integers. In an algebraic way, global 1-dimensional cohomology represents circular structures while local 1-dimensional cohomology captures branching structures in data.

4 OVERVIEW

With the technical tools described early, we now give an overview of our algorithm. We detect branching structures by computing local circular coordinates. Given a point cloud *X* and a point of interest $x \in X$, we choose a subset $U \subseteq X \subset \mathbb{R}^d$ in the neighborhood of *x*, and output local circular coordinate functions $\theta : U \to \mathbb{S}^1$, that give the values for points in the neighborhood of *x*. Our overall pipeline is as follows:

- 1. Turn the point cloud data X in the local neighborhood of x into a simplicial complex K, where the vertices in $K, K^0 = U \subseteq X$.
- 2. Use the local version of persistent cohomology to detect significant cohomology class in data, $\alpha_p \in H^1(K, \mathbb{F}_p)$, where \mathbb{F}_p is the field of integers modulo a fixed prime *p*.
- Lift α_p to α ∈ H¹(K, Z), smooth α to α ∈ C¹(K, R), and integrate a to a circle-valued function θ : U → S¹.
- 4. Approximate topological circular and branching structures represented in $\bar{\alpha} \in C^1(K, \mathbb{R})$ in high dimension to aid visualization in the projection.
- 5. Encode each local circular coordinates with a color map transfer function to highlight true structures and rule out structural illusions.

Here, step 1 and 2 adapts and build upon previous work, while step 3 uses well-established procedures in [13]. Step 4 introduces approximations of circular and branching structures to help visualization. We find it extreme useful in practice.

5 ALGORITHM DETAILS

5.1 Data points to simplicial complexes

We now describe the algorithm step 1 in detail. **Coning operation**. To compute local cohomology at $x \in X$ under the simplicial setting, we represent $H^1(\mathbb{X} \cap B_r(x), \mathbb{X} \cap \partial B_r(x))$ as $H^1(K', L')$, with a pair of simplicial complexes K' and L', where K' represents the space $\mathbb{X} \cap B_r(x)$, and L' represents its boundary.

To construct K' and L', we use the following approximations. Let K_0 be the simplicial complex constructed over the entire point cloud X. Fix a point $x \in X$ and a neighborhood parameter r. A simplex $\sigma \in K_0$ is in K' if some (or all) of its vertices are in $B_r(x)$, or it is the boundary of some simplex in K'. A simplex $\sigma \in K_0$ is in L' if all its vertices are outside $B_r(x)$, and it is the boundary of some simplex in K'.

Let *K* be a simplicial complex in \mathbb{R}^d and let $w \in \mathbb{R}^N$ be a dummy point at infinity (a vector not affinely dependent on the vertices of *K*). The *cone on K with vertex w*, denoted *CK*, is the simplicial complex whose simplexes are of the form: [w] or $[w, v_0, ..., v_p]$ or $[v_0, ..., v_p]$ for $[v_0, ..., v_p] \in K$ [29]. *K* is the *base* of the cone.

Relative cohomology groups can be interpreted as the absolute cohomology groups of an associated simplicial complex [29]. That is, we can prove by excision theorem, $H^1(K', L') \cong H^1(K' \cup CL')$. In other words, we can construct a new simplicial complex $K = K' \cup CL'$ that represents the local structure around *x*, and consequently compute its dimension one cohomology.

Consider the annulus, in a slightly simplified version of the triangulation considered earlier, as shown in Figure 8 top right. Suppose K_0 is the entire triangulation. The approximated L' contains all the blue vertices and edges. K' contains all the light shaded triangles, edges and vertices. The local structure around x can be represented by simplicial complex $K = K' \cup CL'$. Similarly, a slightly more involved example is shown in Figure 8 bottom, with a triangulation of a figure eight.

Sequence of simplicial complexes. Given a collection of data points $X \in \mathbb{R}^d$ with a distance metric, a point of interest $x \in X$, and a radius r, we convert the local neighborhood of x into a simplicial complex K, where $U = K^0 - w$. The default distance metric is the L_2 distance, while Hamming distance and Edit distance are also used in a few examples in Section 6.

Point cloud data can be represented as a single simplicial complex, or more usefully as a nested family of simplicial complexes [12]. We use Vietoris-Rips complex Rips (X, ε) , where there is a *p*-simplex for



Fig. 9. Circular structure on the left has high persistence while circular structure on the right is considered topological noise [13].

every finite set of p + 1 points in X that has a diameter at most ε . Since we are only interested in computing H¹, we only use its 2-skeleton. For $\varepsilon_1 \subseteq \varepsilon_2 ... \subseteq \varepsilon_n$, we obtain a nested family of simplicial complexes, $\mathcal{K}_0 : K_0(\varepsilon_1) \subseteq ... \subseteq K_0(\varepsilon_n)$, where $K_0(\varepsilon_i) = \text{Rips}(X, \varepsilon_i)$. For larger data sets, we can also use Witness complex [12], which is constructed from a subset of points in X.

Naively, we first construct a nested family of simplicial complexes on *X*, \mathcal{K}_0 . We then filter simplexes in \mathcal{K}_0 by their proximity to *x* in certain precise sense (as described in Section 3.2). For a fixed ε_i , $K_0(\varepsilon_i)$ is the simplicial complex constructed from the entire point cloud. We construct the corresponding $K(\varepsilon_i) = K'(\varepsilon_i) \cup CL'(\varepsilon)$. This leads to a nested family of simplicial complexes that represents local structure around *x*, that is, $\mathcal{K} : K(\varepsilon_1) \subseteq ... \subseteq K(\varepsilon_n)$, where $K(\varepsilon_n)^0 - w = U$.

5.2 Persistence cohomology in its local version

At step 2 of the algorithm, we are now given a nested family of simplicial complexes that represent the local structure at different parameter values ε . We introduce the notion of *scale* for learning this local structure through the concept of *persistence*. Persistence studies the evolution of vectors in a sequence of vector spaces [8]. One main example of such a sequence comes from the cohomology groups of a nested sequence of simplicial complexes constructed at different scale. Persistence provides a way of ranking the significance of the cohomological classes and is essential to achieve the robustness of the proposed methods.

Formally, let $K_i = K(\varepsilon_i)$, we are given a nested family of simplicial complexes connected by inclusions,

$$\mathscr{K}: K_1 \to K_2 \to \ldots \to K_n.$$

For $\varepsilon_i \leq \varepsilon_j$, the inclusion of spaces $K_i \subseteq K_j$ induces a map between cohomology groups, $f : H^1(K_j) \to H^1(K_i)$, and we consider the sequence,

$$\mathsf{H}^1(K_1) \leftarrow \mathsf{H}^1(K_2) \leftarrow \ldots \leftarrow \mathsf{H}^1(K_n).$$

A class $\alpha \in H^1(K_a)$ is *born* at the time *a* if it appears for the first time as a cohomology class, and such a class *dies* entering $H^1(K_b)$ when it disappears as a cohomology class. We call b - a the *persistence* of α . We consider classes with high persistence as representing significant topological structure. It is important to note here that $H^1(K_1) = H^1(K_1; \mathbb{F}_p)$ is computed using \mathbb{F}_p coefficient for technical reasons detailed in [13].

Intuitively, persistence separates features from noise by measuring the significance of circular or branching structures. An illustrative examples is shown in Figure 9 where global circle-valued coordinate function on the left corresponds to high persistent, or significant circle structures, while the circle-valued coordinate function on the right might be considered as topological noise.

The algorithm that computes persistent cohomology of a sequence of simplicial complexes, is a modified version of the persistent homology algorithm [22, 7], which in turn is a variation of the classic Smith normal form algorithm [35]. In a nutshell, it involves a specific ordering in conducting matrix reduction on the coboundary matrices of the nested simplicial complexes. After the matrix reduction, we obtain a collection of cocyles, each is represented as a set of edges with coefficients. For a detailed treatment and discussion of persistent cohomology algorithm, see [14]. For the story behind persistent homology, see [20, 21].

5.3 Lifting, smoothing and integration

For step 3 of our algorithm, we are given a collection of cocyles obtained from step 2. Each cocyle is represented as a collection of edges with coefficients in \mathbb{F}_p . We then modify their coefficients to be integers (\mathbb{Z}), and later to be reals (\mathbb{R}). Once we have a collection of edges with real coefficient, we perform integration that "concentrate" the values onto the vertices, therefore constructing our circle-valued coordinate function. Formally, once we obtained a cohomology class $\alpha_p \in H^1(K; \mathbb{F}_p)$, we need to lift it to $\alpha \in H^1(K; \mathbb{Z})$, we then smooth α to $\bar{\alpha} \in C^1(K;\mathbb{R})$, and further integrate $\bar{\alpha}$ to a circle-valued coordinate function $\theta: U \to \mathbb{S}^1$. The lifting, smoothing and integration procedures are detailed in [13]. Here we review some of their key ideas. It is important to note that due to the coning operation in the previous step, we ignore the circle-valued coordinates at the dummy vertex w. **Lifting.** Given α_p , we lift α_p to α , from \mathbb{F}_p coefficient to integer coefficient. That is, $\alpha_p = \sum n_i e_i^*$, where $n_i \in \mathbb{F}_p$, and $\alpha = \sum g_i e_i^*$, where $g_i \in \mathbb{Z}$. We use a heuristic lifting scheme as follows: $g_i = n_i$ if $|n_i| \leq 1$ p/2, otherwise, $g_i = n_i - p$. This enables choosing g_i within range [-(p-1)/2, ..., -1, 0, 1, ..., (p-1)/2]. We expect that p-torsion is rare in practice (see [13] for technical details), therefore in almost all cases tested, the lifted α is still a valid cocycle by satisfying $\delta_1 \alpha = 0$. **Smoothing.** Given α , we find the "smoothest" cocycle $\bar{\alpha} \in C^1(K;\mathbb{R})$ that is cohomologous to α . By smoothness we mean that $\bar{\alpha}$ has a small total variation defined as $||\bar{\alpha}||^2 = \sum_{e \in K^1} |\bar{\alpha}(e)|^2$. α and $\bar{\alpha}$ are cohomologous if there exists a $f \in C^0(K;\mathbb{R})$ such that $\bar{\alpha} - \alpha = \delta_0 f$. Therefore, we obtain $\bar{\alpha}$ by solving the following minimization prob-lem: $\bar{\alpha} = \arg \min_{\bar{\alpha}} \{ ||\bar{\alpha}||^2 \mid \bar{\alpha} - \alpha = \delta_0 f, \exists f \in C^0(K; \mathbb{R}) \}.$

Integration. Given $\bar{\alpha}$, we integrate $\bar{\alpha}$ to obtain the local circle-valued coordinate function $\theta: U \to \mathbb{S}^1$. Given $f: K^0 \to \mathbb{R}$ from the smoothing step, θ can be obtained as the side-effect of smoothing by allowing $\theta = f \mod \mathbb{Z}$. On the other hand, θ can be constructed via bruce force searching on *K*. Starting from an arbitrary $x_0 \in K^0$, set $\theta(x_0) = 0$, for all other vertices $x \in K^0$, finding its shortest path from x_0 . Assign $\theta(b) = \theta(a) + \bar{\alpha}([a,b])$ if a vertex *b* enters the structure through edge [a,b].

5.4 Generator approximations

To aid visualization, we provide two methods to give a fast approximation to $\bar{\alpha}$, the cocycle generator cohomologous to α . Due to the highdimensionality and complexity of the data, often many data points may map to the same value on \mathbb{S}^1 , or close in parameterization by a small ε . Therefore tracing out all edges with non-zero coefficient in $\bar{\alpha}$ can lead to a messy visualization when projected onto a lower dimensional space.

One fast and simple approximation of $\bar{\alpha}$ computes a minimum weight cycle which spans bins of values. Here we assume a binning of parametrized values, where points lie in a common bin if their distance (difference in value) is less than a predetermined ε . Bins are ordered according to their values with respect to \mathbb{S}^1 . Between two neighboring bins, we assumes a complete graph where all pair-wise edges across bins are possible, with edge weights reflecting value differences at their end nodes. Our problem reduces to computing a minimum weight cycle across all bins. We demonstrate in Section 6 that even this simple approximation reveals much information on the structure of the parametrization.

On the other hand, we know that $\bar{\alpha}$ must operate on edges of the underlying Rips complex. Therefore we can augment the minimum weight cycle approximation to enforce this constraint. Given a collection of bins constructed before, we build a graph between two neighboring bins by only allowing edges in the Rips complex. Meanwhile, for a bin with *n* points $v_1, ..., v_n$, we build an internal *n* by *n* bipartite graph, where edge weight $w(v_i, w_j)$ equals the shortest path between v_i and v_j following edges within the Rips complex, and $w(v_i, v_j) = 0$.

To reduce this complexity, using the Rips edges as guides, we preprocess each bin and mark at which points the cycle is allowed to enter or exit a bin of points. In this way, only the pairs of enter and exit points need to be computed for the intra-bin shortest paths. These edges are combined with the the inter-bin edges to form the constrained graph from which we compute the minimum weight cycle.

5.5 Algorithm summary

The algorithm described above detects local 1-cocyles with the following highlights. First, computing local cohomology groups can be approximated by coning operations on Rips complexes. Second, persistent cohomology detects significant features from noise. Third, the above procedure leads to a local circular parameterization that emphasizes branching structure. Last but not least, generator approximations correspond to approximating circular and branching structures in high dimensions to aid visualization. It is important to note that these branching and circular structures are detected in the high dimensional space via cohomology computation. We only use dimensionality reduction techniques overlaying color-mapped coordinate functions to visualize them in their low dimensional projections. Furthermore, we can construct circle-valued coordinate functions locally even if topology is trivial globally.

6 RESULTS

6.1 Software and data sets

The present results are obtained by our implementation of local cohomology computation on top of the C++ library Dionysus [34]. For classic dimension reduction techniques such as Isomap and Laplacian eigenmaps, we use a toolbox from [37].

We construct local and global circle-valued coordinate functions for a variety of synthetic data and real-world examples. Through these experiments, we demonstrate that both the global and local circular coordinates provide a detailed analysis on the intrinsic structure and are extremely beneficial for many applications.

The expensive part of the computation is generating Rips complexes and computing persistent cohomology by reducing coboundary matrixes. The complexity of computing persistent cohomology is worst case $O(n^3)$, where *n* is the number of simplexes in the simplicial complex [10].

6.2 Genus-3 surface and genus-4 surface

We test our methods on several synthetic data sets with known branching structures. The first data set is a point cloud X sampled from a genus-3 surface as shown in Figure 1 (a). We focus on a point $x \in X$ from one of its four corners and construct local circle-valued coordinates in its neighborhood. Its two-way branching structure is illustrated in 1 (b). We construct their corresponding circle-valued coordinate functions from the point cloud, both of which are shown in Figure 1 (e) and (f).

The second data set is sampled from a genus-4 surface in Figure 1 (c), where seven-way branching exists in the neighborhood of x as indicated in Figure 1 (d). We construct seven corresponding circle-valued coordinate functions, three of which are shown in Figure 1, the rest are shown in Figure 10.

6.3 Virus Outbreak

For a small exercise, we use the VAST 2010 mini challenge data set involving Drafa virus genetic sequences. 58 mutated genetic sequences form a collection of outbreak sequences rooted at the ancestor sequence named "Nigeria B". Each genetic sequence contains 1045 nucleotides, and corresponds to a point in 1045 dimensional space. We focus on studying the local structure surrounding x ="Nigeria B", using circle-valued coordinates and Hamming distance metric. We then embed all 58 sequences into 3D space, with Multidimensional Scaling. As shown in Figure 11, local circle-valued coordinates reveal the branching structures surrounding x. To interpret the color-coded circle-valued coordinate function better, we overlaying the points with the phylogenetic tree among these sequences.



Fig. 10. Genus-4 surface data set, where four of its seven local circlevalued coordinate functions are shown.



Fig. 11. Local structures among virus genetic sequences. Points are overlaid with a phylogenetic tree. The red arrow points to the ancestor sequence.

6.4 Motion Capture Data

For this example, we construct both global and local circular coordinates on a couple of motion capture data sets freely distributed online. Motion capture data is the recorded movement of a live actor over time. In the following we will show that that there are interesting features captured by our methods that are worth further investigation. For our testing, we have analyzed motion capture data saved in the Biovision BVH format. BVH is a hierarchical set of relative joint angles rooted at a node that is traditionally centered at hips of the live actor. All translational motion of the actor in world space is also encoded at this node, to give a complete representation of the motion. Since the space for translational motion is small, finding circular structure due to this motion should not be difficult. A much more interesting problem is finding structure and correlation in the joint angles, therefore the translational motion is ignored in our testing.

6.4.1 Local illusions: walk, hop and walk.

The first data set from OSU Motion Capture Lab data repository [32] involves a female actor walking, hopping over an obstacle and then walking again. It contains 189 frames with 66 joint angles per frame. For our tests, each frame is considered a point in 66 dimensional space with a L_2 distance metric. Figure 12 shows this point set embedded onto 3D Euclidean space using Laplacian eigenmaps. This 3D embedding appears to reveal some branching structure, denoted in the top image of this figure. Contrary to this, the local circle-valued coordi-

nate functions computed in high dimension indicate that it's a visual illusion, shown in the rest of images. This example emphasizes the fact that traditional dimensionality reduction can introduce structure where none is present. Local circle-valued coordinates cannot suffer from this flaw and therefore can help in uncouple illusions from actual structure in visualizations.



Fig. 12. Motion capture sequences from a female actor walking, hopping over an obstacle and then walking again. First image on top: Laplacian eigenmaps appears to reveal a branching structure in the local neighborhood of a point. Second image: when we visualize local-circle valued coordinate functions in the same neighborhood, we obtain two independent parametrizations. This indicate that no branching structures exist in that local neighborhood. Third and fourth image: we approximate the generators associated with the local cohomology classes to aid visualization of the branching structure. The approximated generator travels along a given branch where the color indicates its direction. Notice that there are extra edges pointed by arrows that are artifacts from the approximation.



Fig. 13. Motion capture sequences from a Ballet dancer where global circle-valued coordinates reveal moments of hesitation or pauses before a point of inflection in the arm movement.





Fig. 15. Data from a combustion simulation. Top: 3D Isomap embedding of circular structure with highest persistence. We aid the visualization using approximated cocycle generator. Bottom: 3D rotated view with approximated generator to guide visualization. This circular structure can be used to discover correlations between parameters.

Fig. 14. 1995 democrat house representatives voting data. Top: visualizing through transfer function two of the top five global circle-valued coordinate function (ranked by persistence). Middle: the circle structure with highest persistence, red-arrows point to representatives who switched parties or resigned. We use approximated cocycle to aid the visualization of the circular structure. Bottom: the fifth significant circle that may represent some interesting intra-party structure, again cocyle is approximated to guide the visualization.

6.4.2 Ballet dancer

The Ballet dancer data set, obtained from [1], involves a ballet dancer performing a traditional stretch. The dataset contains 471 frames of 54 joint angles. A subsampled (every 2nd frame) compilation of the

frames is shown in Figure 13 top. In this figure, time flows in rowmajor order first from left to right then top to bottom by rows. We construct global circle-valued coordinate functions on the data. The parameterized value bins for the second largest 1-cocycle in terms of is drawn over the frames of Figure 13 top. As this shows, the cycles coincide with the points of the motion data where there is hesitation or pauses before a point of inflection in the arm movement, see Figure 13 bottom right. Additionally, the global-circular coordinates give a binning of similar motions in this parameterization. In bins highlighted in blue in this figure share a common parameterization and, as the frames show, have very similar movement in the arms. Finally, Figure 13 bottom left shows the Laplacian eigenmaps and isomap projection of the



Fig. 16. Climate data: the second most significant circle-valued coordinate function (top) overlaid with approximated cocycle generator (bottom).

joint angles on 3D Euclidean space, left and right respectively. This example emphasizes the fact that without the circular-coordinate parameterization, it would be impossible to infer this cycle from the embedded points alone.

6.5 Voting

For this example, we are interested in detecting intra-party structures in political data. In particular we look at house voting records obtained from house.gov. Figure 14 shows voting data of democrat house representatives in 1995, when 205 representatives voted on 885 issues over the entire year. Using hamming distance metric, we construct global circle-valued coordinate functions with respect to significant 1-cohomology classes. We then embed all 205 points onto 3D using Multidimensional Scaling. In Figure 14 top, the most significant circular structure is actually formed representatives Deal (D-GA), Laughlin (D-TX), and Reynolds (D-IL) who switched political affiliation or resigned during the year and therefore had an incomplete voting record for the end of the year. These data points are marked by red-arrows in the figure. We believe some of the other circular structures as shown in Figure 14 bottom, reflect intra-party structures that are interesting for further investigation.

6.6 Combustion

We use a subset of points from a combustion simulation and analyze its output parameter space. Points are located on a 16 by 16 grid with 4 simulation time steps. Each point is considered 16 dimensional, including parameters such as mixture fraction, dissipation rate, heat release rate and temperature. We embed the points into 3D using Isomap and visualize the significant circular features in the data. We also use our circular structural approximations to guide the visualization. This is shown in Figure 15.

6.7 Climate simulation

We are interested in finding features in the output parameter space of a climate simulation. This data set comprises 1612 simulation points, each with 8 output parameters, including total cloud percentage, precipitation rate, sea-level pressure, surface stress and temperature . We construct global circle-valued coordinate functions and visualize them in Landmark Isomap 3D projection. This is shown in Figure 16, where the significant circular structure is concentrated at the basin of the projection. We believe the circular structure in parameter space indicates correlations among parameters. We will work with domain scientists to confirm such findings.

7 DISCUSSIONS

We consider our work as a first step towards a more ambitious goal of combining dimensionality reduction with local or regionalized topological analysis of intrinsic structure. Our local circle-valued coordinates functions are similar to local dimensionality estimation in a way that they reflect detailed structural information which might have global effects. We ask the following questions: how do the local and global structures of data interact with one another? How can local analysis infer global structure? There are various open questions, and we address a few here.

Shortest local cocycle. Many algorithms exist to compute 1-cycles with geometric constraints, such as shortest by length or minimum by weight. Are these algorithm extendable to compute the shortest (local) 1-cocycles? The smoothing step described earlier obtains a 1-cocycle with minimum total variance. While persistent homology computes representative homology-generating cycles, these cycles can fluctuate drastically due to changes in the filtration or in the simplicial complex. Work in [6] tracks these cycles so that the changes are local with temporal coherence. We believe this line of work can be extendable to (local) persistence cohomology computations.

Extending local parameterization. Using our algorithm, a point set $U \subseteq X$ in the *r*-neighborhood of $x \in X$ is parameterized by circlevalued coordinates function $\theta : U \to \mathbb{S}^1$. We can extend such a parameterization by gradually increasing *r* until non-trivial topological changes take place. That is, we can extend $\theta : U \to \mathbb{S}^1$ to $\theta' : U' \to \mathbb{S}^1$ where $U \subseteq U'$. We can also obtain a total partial ordering of points in *X* by concatenating multiple local parameterizations. That is, given two circle-valued functions $\theta_1 : U_1 \to \mathbb{S}^1$ and $\theta_2 : U_2 \to \mathbb{S}^1$, where $U_1 \cap U_2 \neq \emptyset$. It might be possible to construct $\theta : U_1 \cup U_2 \to \mathbb{S}^1$ in a coherent manner. The notion of 1-cocycle is not only important in our context of circular coordinates, but also shows up in data ranking and discrete vector fields [5]. Does a total partial ordering obtained from "gluing" local 1-cocycles play a role in data ranking?

Computing local cohomology. To guarantee theoretical correctness in computing local (co)homology, we need to use the Delaunay complex as detailed in [2, 3]. However it is unpractical to compute Delaunay complexes in high dimensions. We believe that using Rips or Witness complexes to compute local cohomology in high dimensions is the best available option. Its correctness guarantee remains as an open theoretical question.

Visualizing branching and circular structures in high dimensions. As shown in Figure 15 and Figure 16, the default projection and viewing angles can not visualize the circular structures clearly. We obtain better visualization on these structures by approximating cocycles in high dimensional space and choose proper viewing angles, using algorithm presented in Section 5.4. We believe it is an interesting open question to develop visualization techniques that preserve and emphasize topological structures recovered in high-dimensions, such as branching and circular features discussed in the paper.

REFERENCES

- [1] Free motion capture. http://gfx-motion-capture.blogspot.com/.
- [2] P. Bendich, D. Cohen-Steiner, H. Edelsbrunner, J. Harer, and D. Morozov. Inferring local homology from sampled stratified spaces. In *Proceedings 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 536–546, 2007.
- [3] P. Bendich, B. Wang, and S. Mukherjee. Towards stratification learning through homology inference. AAAI 2010 Fall Symposium on Manifold Learning and its Applications, 2010.

- [4] R. Bott and L. W. Tu. *Differential Forms in Algebraic Topology*. Springer, 1982.
- [5] D. Burghelea and T. K. Dey. Defining and computing topological persistence for 1-cocycles. Manuscript, December 2010.
- [6] O. Busaryev, T. K. Day, and Y. Wang. Tracking a generator by persistence. *Discrete Mathematics, Algorithms and Applications*, 2(4):539– 552, 2010.
- [7] G. Carlsson, A. J. Zomorodian, A. Collins, and L. J. Guibas. Persistence barcodes for shapes. In *Proceedings Eurographs/ACM SIGGRAPH Symposium on Geometry Processing*, pages 124–135, 2004.
- [8] F. Chazal, D. Cohen-Steiner, M. Glisse, L. J. Guibas, and S. Y. Oudot. Proximity of persistence modules and their diagrams. In *Proceedings* 25th Annual Symposium on Computational Geometry, pages 237–246, 2009.
- C. Chen and D. Freedman. Quantifying homology classes. Proceedings 25th International Symposium on Theoritical Aspects of Computer Science, 1:169–180, 2008.
- [10] C. Chen and M. Kerber. Persistent homology computation with a twist. Proceedings 27th European Workshop on Computational Geometry, 2011.
- [11] T. K. Day, J. Sun, and Y. Wang. Approximating loops in a shortest homology basis from point data. *Proceedings Annual Symposium on Computational Geometry*, pages 166–175, 2010.
- [12] V. de Silva and G. Carlsson. Topological estimation using witness complexes. Symposium on Point-Based Graphics, pages 157–166, 2004.
- [13] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Persistent cohomology and circular coordinates. In *Proceedings 25th Annual Symposium* on Computational Geometry, pages 227–236, 2009.
- [14] V. de Silva, D. Morozov, and M. Vejdemo-Johansson. Dualities in persistent (co)homology. Manuscript, 2010.
- [15] T. K. Dey, A. N. Hirani, and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. *Proceedings 42nd* ACM Symposium on Theory of Computing, pages 221–230, 2010.
- [16] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometryaware handle and tunnel loops in 3d models. SIGGRAPH, 45:1–9, 2008.
- [17] M. Dixon, N. Jacobs, and R. Pless. Finding minimal parametrizations of cylindrical image manifolds. *Proceedings Conference on Computer Vision and Pattern Recognition Workshop*, page 192, 2006.
- [18] P. Dlotko. A fast algorithm to compute cohomology group generators of orientable 2-manifolds. 3rd International Workshop on Computational Topology in Image Context, 2010.
- [19] P. Dlotko and R. Specogna. Efficient cohomology computations for electromagnetic modeling. *Computer Modeling in Engineering and Sciences*, 60(3):247–277, 2010.
- [20] H. Edelsbrunner and J. Harer. Persistent homology a survey. Contemporary Mathematics, 453:257–282, 2008.
- [21] H. Edelsbrunner and J. Harer. Computational Topology: An Introduction. American Mathematical Society, Providence, RI, USA, 2010.
- [22] H. Edelsbrunner, D. Letscher, and A. J. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511– 533, 2002.
- [23] Éric Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete Computational Geometry*, 33:627–636, 2005.
- [24] J. Erickson, E. W. Chambers, and A. Nayyeri. Homology flows, cohomology cuts. *Proceedings 41st Annual ACM Symposium on Theory of Computing*, pages 273–282, 2009.
- [25] J. Erickson, E. W. Chambers, and A. Nayyeri. Minimum cuts and shortest homologous cycles. *Proceedings 25th Annual ACM Symposium on Computational Geometry*, pages 377–385, 2009.
- [26] J. Erickson and A. Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. Manuscript, 2011.
- [27] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. *Proceedings 16th Annual ACM-SIAM symposium on Discrete Algorithms*, pages 1038–1046, 2005.
- [28] J. Erickson and P. Worah. Computing the shortest essential cycle. *Discrete and Computational Geometry*, 2010.
- [29] P. Giblin. Graphs, Surfaces and Homology. Cambridge University Press, New York, NY, USA, 2010.
- [30] R. Hartshorne. Local Cohomology: A Seminar Given by A. Groethendieck, Harvard University, Fall, 1961. Springer, 1967.
- [31] A. Hatcher. Algebraic Topology. Cambridge University Press, 2002.
- [32] O. M. C. Lab. Motion capture data sets.
- [33] J. A. Lee and M. Verleysen. Nonlinear dimensionality reduction of data

manifolds with essential loops. Neurocomputing, 67:29-53, 2005.

- [34] D. Morozov. Dionysus library for computing persistent homology.
- [35] J. R. Munkres. *Elements of algebraic topology*. Addison-Wesley, Redwood City, CA, USA, 1984.
- [36] R. Pless and I. Simon. Embedding images in non-flat spaces. In Conference on Imaging Science Systems and Technology, pages 182–188, 2002.
- [37] L. van der Maaten. Matlab toolbox for dimensionality reduction. http://homepage.tudelft.nl/19j49/.
 - [38] A. J. Zomorodian and G. Carlsson. Localized homology. Computational Geometry: Theory and Applications, 2008.