Multivariate Data Analysis Using Persistence-Based Filtering and Topological Signatures

Bastian Rieck, *Student Member*, *IEEE*, Hubert Mara, and Heike Leitte, *Member*, *IEEE*

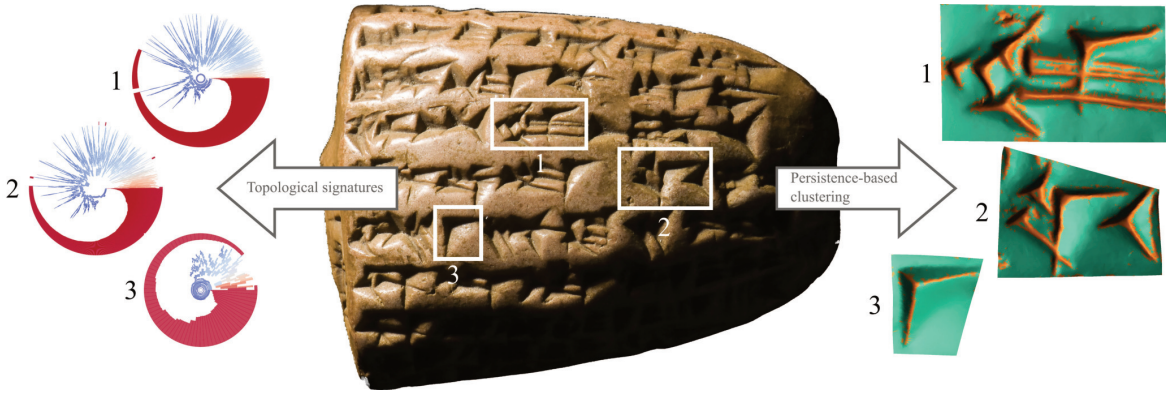


Fig. 1. Analysis of a cuneiform tablet found in the Middle East. The high-resolution surface scan of the clay tablet with Assyrian writing (center) can be described by 16-dimensional *Multiscale Integral Invariant Volume Descriptors*. The persistence rings (left) reveal characteristic patterns in the high-dimensional space that aid in the automatic segmentation of characters (right).

Abstract—The extraction of significant structures in arbitrary high-dimensional data sets is a challenging task. Moreover, classifying data points as noise in order to reduce a data set bears special relevance for many application domains. Standard methods such as clustering serve to reduce problem complexity by providing the user with classes of similar entities. However, they usually do not highlight relations between different entities and require a stopping criterion, e.g. the number of clusters to be detected. In this paper, we present a visualization pipeline based on recent advancements in algebraic topology. More precisely, we employ methods from *persistent homology* that enable topological data analysis on high-dimensional data sets. Our pipeline inherently copes with noisy data and data sets of arbitrary dimensions. It extracts central structures of a data set in a hierarchical manner by using a persistence-based filtering algorithm that is theoretically well-founded. We furthermore introduce *persistence rings*, a novel visualization technique for a class of topological features—the persistence intervals—of large data sets. Persistence rings provide a *unique* topological signature of a data set, which helps in recognizing similarities. In addition, we provide interactive visualization techniques that assist the user in evaluating the parameter space of our method in order to extract relevant structures. We describe and evaluate our analysis pipeline by means of two very distinct classes of data sets: First, a class of synthetic data sets containing topological objects is employed to highlight the interaction capabilities of our method. Second, in order to affirm the utility of our technique, we analyse a class of high-dimensional real-world data sets arising from current research in *cultural heritage*.

Index Terms—Topological persistence, multivariate data, clustering.

1 Introduction

In an age of ever-increasing information density, scientists in many domains generate large amounts of high-dimensional data during their experiments. From a researcher's perspective, the amount and dimensionality of experimental data are both "boon and bane": On the one hand, large data sets with many variables contain more information that might be relevant during analysis and aid in detecting correlations between objects. On the other hand, finding relevant structures in high-dimensional data sets (regardless of their size) is non-trivial. Projection techniques, while worthwhile, may occlude salient structures, thereby creating a wholly different appearance of a data set. Alternatively, clustering algorithms can be used for complexity reduction. Briefly put, the idea of these algorithms [19] is to group *similar objects*, thus supporting a researcher in finding hitherto hidden structures in data sets. However, clustering does not necessarily emphasize any structures in high-dimensional data sets—it yields a set of similar objects, leaving much information in the input data unused. The inclusion of *topological information* such as the Betti numbers, on the other hand, helps uncover differences and similarities between clusters. To this end, we exploit recent advantages in algebraic topology. In short, algebraic topology does not rely on the actual coordinates of data points (as opposed to geometrically-motivated methods). Instead, *distances* (i.e. relations) between the points are considered important. Hence, these methods are ideally suited for analysing high-dimensional, *unstructured* data sets. The long-standing conversion problem of unstructured data sets into the realm of algebraic topology was recently addressed by the development of *persistent homology*. Recognizing the inherent multiscale topology of such data, persistent homology is also somewhat noise-resistant.

In this paper, we introduce a visualization pipeline that couples a recent *persistence-based clustering scheme* [5] with the calculation of topological signatures for clusters. Our visualization pipeline works on data sets of arbitrarily high dimensions and helps uncover hidden relationships between clusters. Our main **contributions** are:

- *Bastian Rieck, Hubert Mara, and Heike Leitte are with the Interdisciplinary Center for Scientific Computing (IWR), Heidelberg University, Germany. E-mail: {bastian.rieck, hubert.mara, heike.leitte}@iwr.uni-heidelberg.de.*

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

- Extending persistent homology analysis to work on large-scale, high-dimensional real-world data sets.
- Introduction of *persistence rings*, a novel visualization technique for topological features of high-dimensional data sets, providing an abstract and concise characterization of their structures.
- Description of an analysis workflow (based on persistent homology) that is suitable for the interactive exploration of high-dimensional data sets.
- Application of our methods to a high-dimensional data set from *cultural heritage* results in the discovery of hitherto unknown structures. We use these structures to provide the *automatic* extraction of cuneiform characters from digitized cuneiform tablets.

2 Related Work

For the exploration of multivariate data sets, common algorithms either try to find relevant dimensions in the data or project all dimensions to 2D or 3D. *Projection pursuit* [18] and `Isomap` [23, p. 243] are typical representatives of the former class of methods. For the latter class, *parallel coordinates* [33, pp. 247–249] and *scatterplot matrices* [33, 238–239] are very popular. Further useful projection methods include *principal component analysis* [23, pp. 226–234], *linear discriminant analysis* [23, Chapter 2], and *multidimensional scaling* [23, pp. 242–245].

The persistence-based clustering algorithm by Chazal et al. [5] is related to density-based clustering methods such as `DBSCAN` [13] and `DENCLUE` [17]. However, both methods use a number of parameters that require fine-tuning and their cluster detection is not based on the topology of a data set.

Another approach towards the exploration of multivariate data exploits the *topological structure* of data sets, thereby obtaining qualitative geometrical information. Furthermore, as Carlsson [1] remarks, topological methods are less sensitive to the choice of metrics and depend on intrinsic geometric properties of data sets. Due to their advantages, topological methods have seen widespread use in recent years: Gyulassy et al. [15] use topologically-motivated extraction of features from 3D scalar fields while also performing simplification. Weber et al. [34] visualize the topology of scalar functions as a terrain while preserving critical points. Building on this, Oesterling et al. [26] visualize generic point clouds. Singh et al. [30] presented `Mapper`, a method for extracting simplified representations of high-dimensional data sets using simplicial complexes and clustering algorithms. Its output is consistent with the `Isomap` algorithm. The generalized *contour tree* of Carr et al. [3] is commonly used for the analysis of scalar fields. Several extensions help explore multivariate data sets: Pascucci and Cole-McLaughlin [27] augment contour trees with Betti numbers of isocontours, while Weber et al. [35] use contour trees to control and improve volume rendering. In order to provide multi-resolution analysis of larger data sets, Pascucci et al. [28] present a topologically-motivated simplification of the contour tree.

Recently, the pool of topological methods has been expanded by persistent homology, a multiscale method for dealing with unstructured high-dimensional data sets containing noise. As a tool, persistent homology has already been used for the analysis of special data sets: Carlsson et al. [2] used persistent homology to investigate the space of natural images, thereby identifying certain submanifolds in the data. Singh et al. [31] applied persistent homology to analyse the topological structure of neural activity of the primary visual cortex. Their experiments confirm that spontaneous activity is statistically different from noise. de Silva and Ghrist [8, 9] used persistent homology to detect *holes* in sensor network coverage.

3 Mathematical Background

In the following, we will expand on the concepts that are relevant for understanding both the clustering algorithm and the calculation of topological signatures. We will not be able to cover every topic in detail but strive to give a short and interesting overview of the matter. For brevity, some concepts will only be defined very loosely. The reader is referred to Hatcher [16] and Munkres [25] for a general introduction to algebraic topology. Zomorodian [37] and Edelsbrunner [11], on the other hand, provide excellent introductions to persistent homology and topological data analysis.

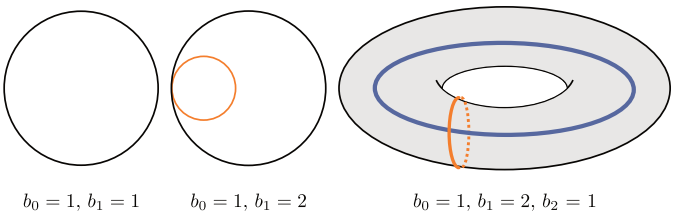


Fig. 2. Non-zero Betti numbers for idealized (i.e. perfectly sampled) topological objects. For the torus, we also show the two generators of the first homology group. We have $b_2 = 1$ because the torus encloses a void, i.e. the space inside its "tube". Comparing the nested circles with the torus highlights the importance of higher-dimensional homology groups: Only when calculating 2-dimensional homology groups can we distinguish the nested circles from the torus by purely topological means.

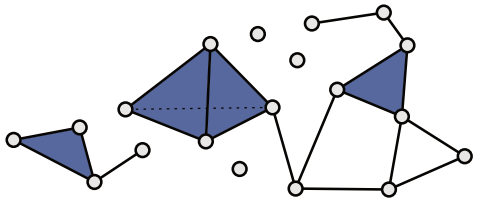


Fig. 3. A simplicial complex containing simplices of dimensions 0–3, i.e. vertices, edges, triangles, and a tetrahedron. Note that only the *filled* faces in the figure are considered simplices. For real data sets, we obtain the *Vietoris-Rips* complex $\mathscr{V}_\varepsilon$ from a *Rips graph* $\mathscr{R}_\varepsilon$—see Sec. 3.2.

3.1 Algebraic topology

The core concept of algebraic topology is a *topological space*. We may think of this as a set of data points from $\mathbb{R}^m$, along with some notion of *connectivity*, i.e. we require that each point "knows" its neighbours. It is the task of a topologist to obtain *invariants* of a topological space, i.e. criteria that help distinguish different topological spaces from each other. The most common invariant is given by *simplicial homology*. Put somewhat simplified, homology assigns a topological space a set of groups, denoted by H_i, where $i = 0, 1, \ldots, m-1$. Each *generator* (in the group-theoretic sense) of a homology group of dimension n then describes an n-dimensional "hole" in the data set. Thus, the topological features detected by simplicial homology correspond to n-dimensional "holes", e.g. connected components (dimension 0), tunnels (dimension 1), and voids (dimension 2). The *number* of "holes" in dimension n is also known as the nth *Betti number* b_n of the data set. A *sphere* in $\mathbb{R}^3$, for example, has $b_0 = 1$ (because a sphere has a single connected component), $b_1 = 0$ (because the surface of a sphere does not have any holes), and $b_2 = 1$ (because the sphere *encloses* a void). In contrast to this, a "filled" sphere, i.e. a *ball*, in $\mathbb{R}^3$ has $b_2 = 0$ because it does not enclose a void. Fig. 2 shows more topological objects and their corresponding Betti numbers. In Sec. 6.1, we will recover their topological properties from non-idealized, noisy samples as well.

Since homology is an invariant, the homology groups of two homeomorphic spaces *coincide* (two spaces are said to be homeomorphic if they can be transformed into each other by "stretching" and "bending"; a classical example is the homeomorphism between a *doughnut* and a *coffee mug*—see [20], p. 5, for an example). Equal homology groups, however, do *not* imply that two spaces are homeomorphic.

There are several (mostly equivalent) notions of defining homology groups. From the perspective of computer science, *simplicial homology* is the most appealing one because it can be calculated effectively—see [25, pp. 56–61] for the standard algo-

rithm. For these calculations, the topological space needs to be given as a *simplicial complex*, i.e. a space constructed from vertices (0-dimensional simplices), edges (1-dimensional simplices), triangular faces (2-dimensional simplices), and their corresponding n-dimensional generalizations—see Fig. 3 for an example. For simplicial homology, the generators of the homology groups are represented as formal sums of simplices, the *simplicial chains*. Our topological signature calculation algorithm uses these chains in order to highlight the detected features of a data set.

3.2 Persistent homology

Since real-world data sets are not commonly endowed with a topological structure, it needs to be obtained by some means. The standard construction for this is to calculate a *Rips graph* $\mathscr{R}_\varepsilon$ (also known as *neighbourhood graph*) for the data set. $\mathscr{R}_\varepsilon$ is then expanded into a simplicial complex, yielding the *Vietoris-Rips* complex $\mathscr{V}_\varepsilon$ [32]. Note that $\mathscr{V}_\varepsilon$ is built from the input data set and each of its simplices is a *subset* of the data. Thus, if given a 1-simplex (u,v) of $\mathscr{V}_\varepsilon$, for example, it is justified to say that data point u and v *create* the simplex.

Usually, the Rips graph is defined as follows: Given a set of n data points from $\mathbb{R}^m$, the Rips graph $\mathscr{R}_\varepsilon$ is a graph whose vertices are given as $V = \{0,\dots,n-1\}$ and whose edges are given as $E = \{(i,j) | d(i,j) \leq \varepsilon\}$, where d refers to a distance function such as the Euclidean metric.

The problem of $\mathscr{R}_\varepsilon$ is its distance parameter ε: Different values will yield different Rips graphs, which in turn will associate different homology groups to the data set. Recognizing that there is no single "correct" parameter value for real-world data sets, persistent homology now calculates homology groups for a *range* of values for ε. To this end, the simplices of the Vietoris-Rips complex are partitioned into *positive* and *negative* simplices—see the seminal paper of Edelsbrunner et al. [12] for details on this calculation. Briefly put, positive simplices *create* a topological feature—i.e. a "hole"—while negative simplices *destroy* one. By assigning each simplex a weight (for example the value of ε for which the simplex "appears" in the simplicial complex), topological features can be assigned a *persistence* value. For a feature created by a k-dimensional simplex σ, this value is given as the difference between the weight of the negative $(k+1)$-dimensional simplex τ destroying it and the weight of σ, its creator. If a feature never gets destroyed, it is assigned an *infinite* persistence. For each dimension k, the persistence values are combined into *persistence intervals* of the form $[a,b]$ with $a \in \mathbb{R}$ and $b \in \mathbb{R} \cup \{\infty\}$. Informally put, a is the "creation time" of a topological feature, while b is the time of its "destruction". The *persistence* or *prominence* of a persistence interval is given as $b-a \in \mathbb{R} \cup \{\infty\}$. Intuitively, high persistence indicates *relevant* features in the data set, while low persistence indicates topological noise. Thus, *persistent homology* defines a multiscale homology theory for unstructured data sets (usually given as point clouds) by assigning them a set of persistence intervals.

3.3 Filtrations

At the heart of all persistent homology calculations lies the concept of a *filtration*. A filtration of a given simplicial complex is a sequence of valid subcomplexes. We may think of this as a simplicial complex that is built by incrementally adding new simplices. By keeping track in which simplicial complex a given simplex appears *first*, a filtration naturally imposes a *partial order* on the set of simplices. This partial order can be extended to a *total order* by sorting conflicting simplices by dimension and, in case of ties, lexicographically. By varying the filtration of a simplicial complex, we can influence the type and number of topological features that are detected. This is done by assigning low-dimensional simplices (usually, either 0-simplices or 1-simplices) a weight w that is calculated by some means (such as the Euclidean distance between two 1-simplices). The weight of higher-dimensional simplices is then calculated as the *maximum* of the weight of the lower-dimensional simplices it comprises. Persistent homology calculations then use these weights as the respective *creation* and *destruction* times for features. For this paper, we require two filtrations.

Upper star filtration The persistence-based clustering scheme (see Sec. 3.4 for details) uses the *upper star* filtration, which assigns each 0-simplex v a density value $f(v)$, and each edge (u,v) the value $\max(f(u),f(v))$. Simplices are now sorted by *decreasing* function value and *increasing* dimension. Zomorodian [37] showed that the *upper star* filtration corresponds to exploring the *superlevel sets* of the weight function, i.e. sets of the form $f^{-1}[\alpha,\infty]$ for some value $\alpha \in \mathbb{R}$.

Distance filtration We use the *distance filtration* for the calculation of topological signatures. This filtration assigns 0-simplices a weight of 0, while a 1-simplex (u,v) is assigned the value $d(u,v)$, i.e. the Euclidean distance between data points u and v. This filtration is more geometrically-inclined and simplifies the detection of *geometrical* features in a data set—the meaning of this will become clear when analysing the example data sets.

3.4 Persistence-based clustering

The persistence-based clustering scheme of Chazal et al. [5] uses a density estimator to obtain a density value $f(p)$ for each data point p. Thus, their algorithm fits into the class of *density-based* clustering schemes: Given a density function f, the assumption is that the input data are samples drawn from f—see Kotsiantis and Pintelas [19] for more information. Hence, in order to understand the structure of the data set, the scheme of Chazal et al. analyses the *topology* of f by associating clusters with the regions of attraction of the local maxima. Since a density function is usually not directly available, it needs to be estimated before initializing the clustering scheme. Our method estimates the density using the *distance to a measure* method introduced by Chazal et al. [4]. Requiring only a distance function $d(\cdot,\cdot)$, this density estimator is well-suited for the density estimation of point clouds. For a query point p in the data set, we have $f(p) = -1/k\sqrt{\sum_{i=1}^{k} d^2(p,n_i)}$, where n_i refers to the ith neighbour of the query point p and k is the number of neighbouring points that are used for the density estimation. Note that the minus sign is required to ensure that larger function values correspond to higher densities.

Taking the input points as vertices of a Rips graph $\mathscr{R}_\varepsilon$, each vertex p is then assigned its density value $f(p)$. Chazal et al. consider $\mathscr{R}_\varepsilon$ to be a 1-dimensional simplicial complex and sort it by an *upper-star filtration*. Their calculation of persistent homology now involves looking at the *superlevel sets* of f, i.e. sets of the form $f^{-1}[\alpha,\infty]$ for a density value $\alpha \in \mathbb{R}$. From Morse theory, we know that a new component appears in the superlevel set of f whenever a *local maximum* v is reached. Persistent homology now prescribes that v creates a new connected component. The component created by v is destroyed when it gets connected to another component that has been created by an even larger density value. This can only happen at data points that are *not* a local maximum. Letting α_c be the density value upon creation of the component and α_d the density value upon destruction, we assign a persistence interval of $[\alpha_c,\alpha_d]$. Following Chazal et al. [5], we display the intervals in a persistence diagram—see Sec. 4.1 for their definition. Note that, by construction, each persistence interval belongs to the 0-dimensional homology group. Thus, it describes a *connected component* in the data set, and the number of intervals with a large persistence is taken as an approximation to the number of *clusters* in the data set.

3.5 Topological signatures

Having extracted structures from the data set using the persistence clustering algorithm that is the core of our scheme, we *partition* the data set according to its persistent homology in dimension 0. This operation yields several small sets, whose total number depends on the current parameters of the algorithm. We now expand each of this smaller sets into its corresponding Vietoris-Rips complex, using the fast expansion algorithm described by Zomorodian [38]. We purposefully *ignore* the density estimates for this expansion. Instead, we select a distance-based filtration for the complex. More precisely, this filtration will assign each 0-simplex a weight of zero, while each 1-simplex (u,v) is assigned the Euclidean distance $d(u,v)$ between u

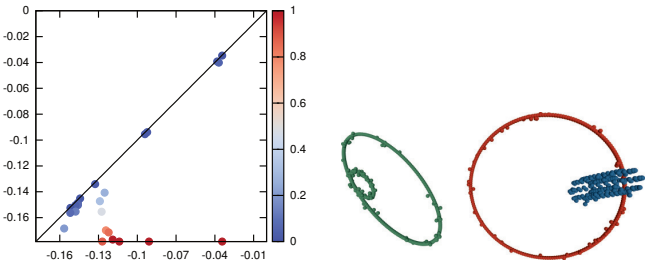


Fig. 4. Persistence diagram (left, showing H_0) and 3D view of the synthetic data set (right). The object colours on the right indicate the cluster a data point belongs to. Our algorithm recovers the three distinct objects correctly. We created the objects at different scales in order to prove that our persistence visualizations handle this correctly. For the persistence diagram (left), we colour-coded points by their *relative* persistence values, using a continuous diverging colour map suggested by Moreland [24]. Note that the points in the upper right corner have a very high multiplicity but as their persistence values are roughly equal, they become occluded.

and v. We selected this filtration because it represents multi-scale features better—features at smaller scales, for example, may not conform to the surrounding density, which would make them harder to detect. For each expanded complex, we now apply the *persistent homology* calculation algorithm described by Zomorodian and Carlsson [39]. In addition to the calculation of the persistence intervals as outlined in Section 3.2, this algorithm has been extended to calculate the *generators* of the detected features. Each generator corresponds to a simplicial chain that creates a certain topological feature in the data set—we show examples for this in Sec. 6.1. As a result, we obtain a set of *topological signatures* depicting the topology of each structure that was detected by the persistence clustering algorithm.

The use of topological signatures depends on the selected data set. At a coarse level, even without the calculation of homology generators, they help in *similarity detection*. Similar topological signatures indicate similarities in the detected structures. At a fine level, by using the homology generators, the signatures may *uncover* hidden or nested structures in the detected clusters.

4 Persistence Visualizations

Persistent homology calculations yield a set of persistence intervals for each dimension. Currently, there are two common methods for visualizing these intervals: Drawing them as *intervals* in the plane, which is done by *persistence barcodes* [14], or drawing them as *points* in the plane, which is done by *persistence diagrams* [6]. In this section, we show both visualizations and describe their drawbacks. In addition, we motivate our *persistence rings* visualization, which uses a radial visualization of persistence intervals, thereby combining the advantages of persistence diagrams and barcodes. In the following, we assume that we have a set I_k of persistence intervals for each dimension k of the data set. Intervals from I_k describe generators from the kth persistent homology group, and we will hence say that the corresponding visualization shows H_k (while it actually shows the *generators* of H_k).

4.1 Persistence diagrams

For each $[a,b] \in I_k$, we obtain a persistence diagram by considering the interval to signify a point in the Euclidean plane—see Fig. 4a for an example. Intervals with $b = \infty$ are usually placed slightly *outside* the diagram or on its (upper or lower) border. Note that, depending on the filtration, values are either situated all *above* the diagonal (e.g. for the distance-based filtration) or *below* the diagonal (e.g. for the upper-star filtration used for persistence clustering). Points with a close distance to the diagonal indicate features that have a rather brief existence. Consequently, a dense area of points around the diagonal indicates topological noise. In total, the persistence diagram stresses the *lifespan*, i.e. the persistence of topological features, highlighting those that do not exist for very long.

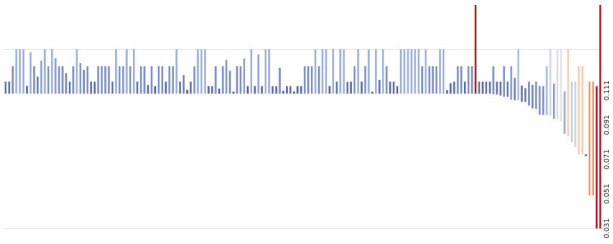


Fig. 5. H_1 of the second cluster (a torus) of the synthetic data set, shown as a barcode (which we rotated for layout reasons). Even for an object as topologically simple as a torus, the amount of persistence intervals is substantial. See Sec. 6.1 for a detailed discussion of the data set.

The drawback of this visualization is that users expect the dense areas around the diagonal to represent *significant structures* in the data set—instead, the converse is true. We solve this by colouring each point in the diagram based on its (relative) persistence value. *Relevant* points, i.e. points with a large persistence, are highlighted in red by this colour map. A further problem of the diagram is the *occlusion* of features with (almost) equal values. Users cannot easily determine how many features are represented by a point in the diagram. For qualitative queries, e.g. for comparing which point represents more features in the diagram, colour-coding could be used. However, this still makes it harder for users to grasp how many features are present. Despite all this, the persistence diagram has its merits because it greatly simplifies highlighting all features whose lifespan is bounded by some upper value. Thus, the persistence diagram is a good choice for displaying the results of the clustering algorithm.

4.2 Persistence barcodes

For each $[a,b] \in I_k$, we obtain a persistence barcode by drawing a horizontal line or bar from a to position b. The resulting bars are then placed on top of each other—see Fig. 5 for an example. Again, intervals with $b = \infty$ are handled by extending their lines beyond the borders of the barcode. Thus, barcodes represent persistence intervals as bars of different lengths, thereby highlighting both the lifespans of attributes as well as their creation times.

The disadvantage of barcodes is their increase in height for larger data sets—which is why we will *not* be using them for the analysis of our data sets in Sec. 6. Since even a very small interval occupies some vertical space, users quickly lose their overview over larger data sets. For example, users cannot simply estimate the number of features present at a given "time". A further problem of barcodes is that very small intervals may not be represented adequately—they may simply be lost in the clutter of surrounding intervals. On the other hand, since each feature is represented by a single bar, the barcode does not suffer from any occlusion problems. Using zooming and filtering of intervals, interactive barcodes may somewhat remedy the shortcomings. However, barcodes cannot easily provide users with a sufficient overview of the data set.

4.3 Persistence rings

Motivated by requiring a visualization that is *compact* while still providing a good overview of the data set, we introduce *persistence rings*. This visualization displays persistence intervals *radially*. More precisely, we assign $[a,b] \in I_k$ an annular sector from radius a to radius b. If $b = \infty$, we use a radius that is larger than the radii used for the finite intervals. Consequently, we still have two degrees of freedom for drawing the annular sectors, namely its *opening angle* θ (which determines the size of the sector) and its *angular offset* ϕ (which determines the radial position of the sector).

Depending on the desired view on the data set, different ways of selecting these angles can be established. Ideally, we would like the angles to reflect the *relevance* of a given feature. Foremost, however, we require that all annular sectors are placed without any overlaps. Satisfying both constraints is not easily possible because the problem

is inherently a *global* optimization problem—local optimization of angles cannot guarantee that there are no overlaps. In addition, the intersections between different intervals are *intransitive*, making it harder to check whether an interval can be placed correctly. During the implementation of the persistence rings, we experimented with different approaches. *Global* optimization of both angles (using a simple function that relates the *area* of a segment to its *persistence*) proved to be very costly due to missing gradients. Furthermore, the layout was not aesthetically pleasing and contained some overlaps, thereby not alleviating the occlusion problem. Our current implementation thus uses a simple heuristic to ensure that there are no overlaps and that the size of each segment and its persistence are (roughly) correlated—see Alg. 1 for details. However, the heuristic cannot guarantee this correlation: Segments with *few* neighbours may be assigned a large θ even though their persistence is small.

Input : Set $\mathscr{I}$ of persistence intervals, interval tree $\mathscr{T}$
Output: Persistence intervals with assigned angles
Partition $\mathscr{I}$ into finite and semifinite intervals. Sort the finite intervals by increasing persistence, i.e. by increasing interval length. Sort the semifinite intervals by increasing creation time.
foreach Interval $I \in \mathscr{I}$ **do**
  $\mathscr{N} \leftarrow$ FINDOVERLAPPINGINTERVALS($\mathscr{T}, I$)
  numNeighbours $\leftarrow 1$, sumLengths $\leftarrow 0$, $\phi_{\max} \leftarrow 0$
  foreach Interval $N \in \mathscr{N}$ **do**
    if ALREADYPLACED(N) **then**
      $\phi_{\max} \leftarrow \text{MAX}(\phi_{\max}, \phi(N) + \theta(N))$
    else
      numNeighbours $\leftarrow$ numNeighbours $+1$
      sumLengths $\leftarrow$ sumLengths $+$ GETLENGTH(N)
    end
  end
  $\alpha \leftarrow 2\pi - \phi_{\max}$, $\phi(I) \leftarrow \phi_{\max}$
  if I is finite **then**
    $\theta(I) \leftarrow \alpha \cdot$ GETLENGTH(I)/sumLengths
  else
    $\theta(I) \leftarrow \alpha$/numNeighbours
  end
end

Alg. 1: Placement heuristic for persistence rings

In order to give users more visual clues concerning the persistence of the annular sectors, we colourize each sector according to its persistence, thereby facilitating the detection of relevant features in the data set. The colours of the annular sectors can also be used for other purposes: For the calculation of topological signatures, we assign colours based on the length of the *simplicial chain* of a persistence interval. This colour assignment highlights topological features that occur at *different scales*, regardless of their actual persistence values—see our analysis in Sec. 6.1 for details.

4.4 Comparison

Fig. 6 contains different persistence visualizations for a noisy example data set. The data set (Fig. 6a) contains two nested circles with different sizes that have been sampled unevenly with approximately 150 points per circle. Below the data set, the corresponding barcode for H_1, i.e. the first homology group of the data set, is shown. Here, the barcode contains little noise and indicates that there are *two* persistent generators, described by the first bar (corresponding to the circle denoted by the number 2) and the third bar (corresponding to the circle denoted by the number 1). Two other generators with a smaller lifetime also appear, but are destroyed later on. These correspond to smaller degenerate circles caused by the additional noise in the data set. The persistence diagram (Fig. 6b) showing the same information contains a lot of topological noise, which results in many points clustered around the diagonal. Note that since we used a distance filtration to calculate the persistence diagram, all points are situated *above* the diagonal. The persistence ring (Fig. 6c) clearly shows that the first persistent generator (the larger of the two outer "slices" of the rings)

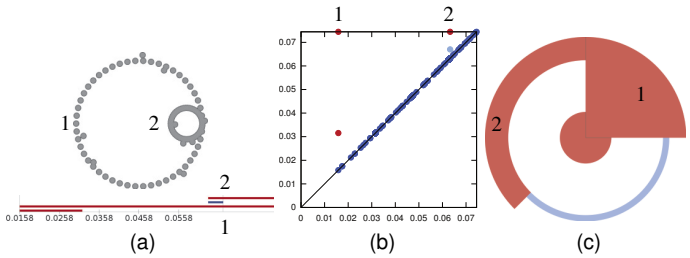


Fig. 6. A comparison between different persistence visualizations for H_1. The underlying data set contains a noisy sampling of two nested circles (among others, we will also use these objects in Sec. 6.1). The two relevant generators of the first homology group are indicated by the numbers 1 and 2. See Sec. 4.4 for more details.

is created very early, while the second persistent generator is only created for a rather large distance value. This indicates that the *scale* of the features described by the generators varies—we can see that there is a size difference between both circles. In essence, the persistence diagram shows (Fig. 6b) the same information. However, the scale of the features is not immediately obvious to users.

5 Workflow

In short, our persistence-based visualization and analysis pipeline consists of the following steps: (i) We cluster input point clouds of arbitrary dimensions using the algorithm described in Sec. 3.4. (ii) The user is presented the clustering results via projections of the original data. (iii) Additional persistence visualizations depict central topological features of the data set. (iv) The different visualizations are interactive and coupled to enable a "brushing and linking"-like data exploration. More precisely, a typical analysis session results in the following steps:

1. Obtain information about the (potential) number of connected components in the data set by using *dendrograms* created with *single-linkage hierarchical clustering* [5]. Use this information to select a distance threshold ε for the *persistence clustering* algorithm. Note that *without* the dendrogram information, this parameter would be very volatile: A wrong value for ε will result in *oversampling* the input data—in the worst case, if ε is set to the maximum distance of two data points, the clustering scheme will *always* detect a single large cluster.

2. Set parameter k, which represents the number of neighbours used for the density estimator. Usually, our default value of 15 yields satisfactory results. The user may verify the choice for k through a point cloud or a histogram of the density values. Values for k that result in little variation should be avoided. If k is set too high, there will be no spread in densities any more. If k is set too low, too many points of maximum density will be found by the clustering algorithm.
   However, we did not experience any significant changes in the clustering quality when modifying k. Thus, as long as care is taken that there is some variation of densities, the actual parameter value is not relevant.

3. Use information from the previous steps to apply the *persistence clustering* algorithm. This yields a (coarse) segmentation of the point cloud, resembling a density-based clustering. We provide a low-dimensional projection of the input point cloud, making it possible for the user to fine-tune parameters of the clustering scheme—although our software contains sensible *default values* for each parameter, they might not be applicable to any situation. Furthermore, we *augment* the projection using colours based on density values or cluster associations. If preferred by the user, the data view may also use projection techniques such as multidimensional scaling or parallel coordinates.

4. The user may now choose to either *explore* the segmented data set (using topological signatures) or *refine* the segmentation (by changing the parameters of the clustering algorithm). The *topological sig-*

natures as described in Sec. 3.5 uncover similarities of the detected clusters.

5. Having obtained information using the topological signatures, the user may choose to perform further topological analysis on either the whole point cloud or parts of it. Due to the fast runtime of the persistence clustering algorithm for moderately sized data sets, this allows an almost real-time exploration of multivariate data sets.

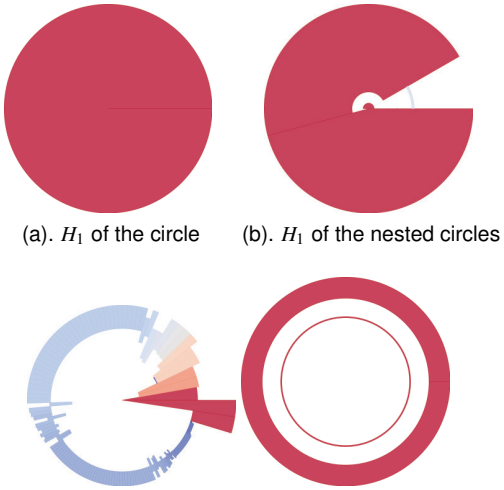
6 Results

In the following, we present the results of applying our visualization pipeline to two classes of data sets. We first analyse a synthetic data set containing a variety of interesting topological structures. We also use this data set to highlight how topological signatures help users explore a high-dimensional data set. After this, we show the results of our analysis of high-dimensional feature space data from *cultural heritage*. Our method enables the identification of relevant structures in the feature space. We use these structures to distinguish the *writing*, i.e. cuneiform characters, from the *background* of cuneiform tablets.

6.1 Synthetic data set

In order to evaluate our analysis pipeline and highlight its interaction capabilities, we generated a synthetic data set that contains a number of topological features: We sample three distinct topological objects at random—a *circle*, two *nested circles*, and a *torus*. The center of each of these objects is placed at random in the data set. We then embed each object in a high-dimensional space by using the *subgroup algorithm* of Diaconis and Shashahani [10]. Briefly put, this algorithm can be used to calculate a random element of $SO(n)$, i.e. the *special orthogonal* group of $n \times n$ matrices, which describes rotations in an n-dimensional space. We now pad the coordinates of each object with zeros and apply the rotation matrix. The result is a seemingly high-dimensional point cloud containing discrete samples of objects from $\mathbb{R}^2$ and $\mathbb{R}^3$. As a last step, in order to simulate real-world data sets, we add 25% noisy data points using Gaussian noise with $\mu = 0$ and σ equal to half the average distance of the points around each object.

For the analysis, we first apply the clustering algorithm to the point cloud. Since the data set is not very complicated, the clustering algorithm *perfectly* recovers the three objects, yielding a classification rate of 100%. See Fig. 4a for the resulting persistence diagram and Fig. 4b for the point cloud and its cluster associations. The three objects show up as three points of infinite persistence in the persistence diagram. This is indicated by the three red points on the x-axis of the diagram. Note that the remaining points are of finite persistence because they are coloured differently. The information that three distinct objects have been identified already conveys some insight into the data set. In order to improve this insight, we also calculate a *topological signature* for each object—see Fig. 7 for a combined display and Fig. 9 for the corresponding persistence diagrams (which we included for comparison reasons). The signature clearly highlights in what ways these objects differ: For the first cluster (Fig. 7a), we have a single large persistence generator in dimension 1. Thus, this cluster has the topology of a *circle* and the generator corresponds to the "hole" that is bounded by the circle. For the second cluster (Fig. 7c), we obtain two generators of large persistence in dimension 1 and many generators of lesser persistence. Further interaction with the persistence rings shows that the plethora of less-persistent generators corresponds to circular structures of the torus. These structures are detected at a very early distance level by persistent homology—hence their relatively large persistence—but they are *destroyed* as soon as the simplicial complex has been expanded. The two prominent generators, on the other hand, correspond to the two circles that create the torus. In dimension 2, the second cluster has a single persistent generator, indicating an enclosed cavity or void. All in all, this suggests the topology of a *torus* for the second cluster. For the third cluster (Fig. 7b), we again identify two generators of large persistence in dimension 1, as well as some generators of low persistence. Interaction again shows that each of the persistent generators corresponds to the "holes" bounded by each circle while the less-persistent generators correspond to topological noise that occurs during the expansion of the simplicial complex. Note that



(a). H_1 of the circle (b). H_1 of the nested circles

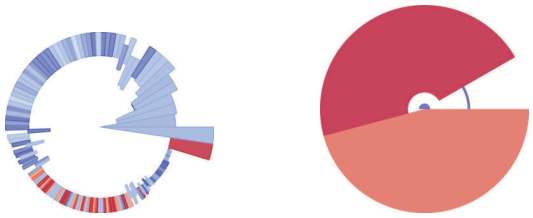
(c). H_1 and H_2 of the torus

Fig. 7. Persistence visualizations for the detected clusters of the synthetic data set. In comparison to a barcode, the persistence rings show the data in a more compact and aesthetically pleasing way, allowing faster comparisons between different signatures. As described in Fig. 4a, each sector is colourized according to its relative persistence value. See Sec. 6.1 for a detailed discussion of the data set and its persistent homology.

the third cluster is shown to behave differently than the second cluster. Our signature does *not* show any topological features in dimension 2, which is justified because the nested circles do not enclose a cavity. Hence, even without any further inspection of the data set, persistence rings serve to distinguish the torus from the nested circles, both *quantitatively* (more persistence intervals) and *qualitatively* (non-empty persistent homology in dimension 2).

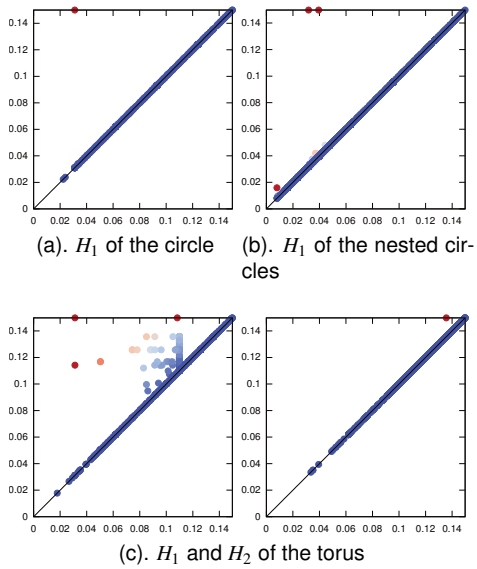
In order to obtain information about the *scale* of the detected features, we added the option of colouring persistence rings according to the lengths of their associated boundary chains. More precisely, the persistent homology calculation [39] yields a boundary chain for each persistence interval. By counting the number of simplices the boundary chain consists of, we obtain an approximation of the size of the corresponding feature. Fig. 8 depicts the size of several features of the synthetic data set.

In summary, we obtain an understanding of *intrinsic* properties of structures in a data set *without* overly relying on their actual coordi-



(a). Persistence ring of H_1 for the torus (b). Persistence ring of H_1 for the nested circles

Fig. 8. For these persistence ring visualizations, we colourized the annular sectors by the lengths of their boundary chains (using the colour map from Fig. 4a), thereby obtaining a visualization of the *scales* of the detected topological features. For the torus cluster on the left, we see that its first persistent generator does not contain many points because it corresponds to the circle that goes around the "hole" bounded by the torus. The second generator, on the other hand, contains more points as it goes around the "tube" of the torus. For the cluster of nested circles (right), we see that both circular features have roughly the same number of data points.



(a). H_1 of the circle (b). H_1 of the nested circles

(c). H_1 and H_2 of the torus

Fig. 9. Persistence diagrams for the detected clusters of the synthetic data set. Note the presence of topological noise around the diagonal. See Fig. 7 for the corresponding persistence rings.

nates: Note that our method recovers the topological objects and their signatures correctly *without* using any prior information about the dimensionality of the point cloud. Also, we do not make any assumptions about the structures that are embedded in the data set.

6.2 Analysing the 16-dimensional feature space of *Multiscale Integral Invariant* (MSII) filters

Our second application is rooted in the field of *cultural heritage*: Cuneiform tablets are among mankind's earliest form of written documents. These are clay tablets of varying shapes and sizes that have been inscribed using blunt reeds. The impressions and indentations left by the writing device are shaped like *wedges*, hence the name *cuneiform* (from the Latin word *cuneus*, meaning "wedge"). At present, there are several hundreds of thousands of different cuneiform tablets. Only few experts in assyriology are capable of their transcription, transliteration, and translation. Since the tablets are often damaged, this process is very time-consuming.

Previous approaches for automatic character extraction involved the use of photography of 2D scanners, which proved to be very error-prone. Recently, the rise of precise 3D scanners facilitated the creation of *digitized* versions of the cuneiform tablets. These meshes *preserve* geometrical and topological information, making them suitable for further analysis. Our work builds on methods introduced by Mara et al. [22, 21]. Briefly put, this work involves the calculation of a multiscale filter for meshes of cuneiform tablets.

This *Multiscale Integral Invariant* (MSII) filter was introduced by Pottmann et al. [29] for feature detection on 2D manifolds embedded in $\mathbb{R}^3$. The filter requires input data in the form of a polygonal mesh, i.e. a triangulated point cloud. At any point $\mathbf{p} \in \mathbb{R}^3$ of the input mesh, the *volume integral invariant* $V_r(\mathbf{p})$ is defined as the integral of the indicator function of the mesh domain D, evaluated within a Euclidean ball B of radius r around $\mathbf{p}$, i.e. $V_r(\mathbf{p}) = \int_{\mathbf{p}+rB} 1_D(\mathbf{x}) d\mathbf{x}$. The volume descriptor is then *normalized* such that $V_r(\mathbf{p}) \in [-1,1]$. Instead of calculating $V_r(\mathbf{p})$ for a *single* radius r, Pottmann et al. [29] now define a scale of 16 decreasing radii $r_1, \ldots, r_{16}$. Then, they evaluate $V_r(\mathbf{p})$ for each radius, and assign each point $\mathbf{p} \in \mathbb{R}^3$ its *feature vector* $f_\mathbf{p} = (V_{r_1}(\mathbf{p}), \ldots, V_{r_{16}}(\mathbf{p})) \in \mathbb{R}^{16}$. The set of all 16-dimensional feature vectors is called the *feature space* of the mesh. Pottmann et al. furthermore show that the feature space yields a very accurate measure of the local *convexity* or *concavity* of a point in the mesh. The multiple scales ensure that even small variations in the shape of the mesh can be detected.

Mara et al. [22] now suggest a preliminary character extraction algorithm operating on this feature space. Their proposed technique uses the *convolution* of feature vectors, combined with a *thresholding* procedure. However, there are several problems: (i) The calculation of feature vectors requires a start radius and a number of scales. Incorrect choices will make the character extraction process error-prone. (ii) A threshold cannot be easily chosen. Furthermore, the extraction is very unstable with respect to small changes to the threshold. (iii) The results of the extraction process cannot be validated automatically.

Our topological exploration method now operates on the 16-dimensional feature space (i.e. we do not use the connectivity of the input mesh), providing solutions to the problems outlined above: First, using persistence-based clustering, we partition the feature space into several clusters. These clusters describe different regions in the mesh of a cuneiform tablet. Second, using topological signatures, we can quickly decide whether a data set contains meaningful structures. This is a first step towards automatic validation of the extraction process. Third, we highlight instabilities in the feature vector calculation, which warrants further research into integral invariants. At last, by interacting with the persistence rings, we discover a previously unknown complicated *nested relationship* of several classes of feature vectors.

In the following sections, we describe the results of our topological analysis. We deliberately only use homological information from dimension 0 (for the clustering) and dimension 1 (for the signature calculations). Although our method handles arbitrary dimensions, the restriction to H_0 and H_1 reduces the complexity of our analysis, while retaining sufficient information about the high-dimensional space to enable us to derive valuable clues about the spatial structure of different features.

6.2.1 "Kaskal" data set

We first analyse a *synthetic* mesh depicting the cuneiform character "Kaskal". In contrast to real-world data, this mesh has fully planar parts. In addition, the indentations are executed perfectly and there are no damaged parts. Fig. 12a depicts an orthographic projection of the mesh with virtual illumination. Using the dendrogram from Fig. 10, we experimented with values for ε in the range of 0.075–0.125. All these parameters roughly yielded the same persistence diagram, which is shown by Fig. 10. Application of the persistence clustering algorithm resulted in 12 clusters—see Fig. 12b. The clusters provide a very good partition of the cuneiform character into several parts. Firstly, note that the majority of points are assigned to the largest cluster (shown in orange). It comprises the *locally* planar parts of the mesh. Thus, parts of each V-shaped wedge are also assigned to this cluster. Secondly, the red cluster describes the "bottom" part of each V-shaped wedge. Finally, the remaining clusters contain points that describe various substructures of the mesh—the green cluster, for example, contains points where two or more wedges intersect.

Fig. 11 shows three representative persistence rings for the data set, describing the orange cluster (Fig. 11a), the green cluster (Fig. 11b), and the red cluster (Fig. 11c). Note that even by visual inspection, the differences in the clusters are apparent. Thus, without relying on information such as the cluster size, which might be misleading for noisy data sets, users can reliable distinguish different clusters from each other.

Furthermore, we found that the large number of generators for the orange cluster indicates literal "holes" in the parameter space: There are several sets of points belonging to the orange cluster that *bound* other clusters in the data set. In contrast, for the remaining clusters, we discovered that the number of generators of large persistence varies only between 2 and 3. Again, we identified points that *bound* other clusters. Points of the orange cluster *permeate* the complete feature space, while the smaller clusters only bound some smaller parts of it. So far, these findings indicate a complex *nested* relationship between the different clusters, making further studies necessary. This relationship was hitherto *not* explored and we are convinced that it can be exploited to further improve the quality of the segmentations.

In summary, with the "Kaskal" data set, we studied several relevant surface structures of a cuneiform tablet while keeping data artefacts as

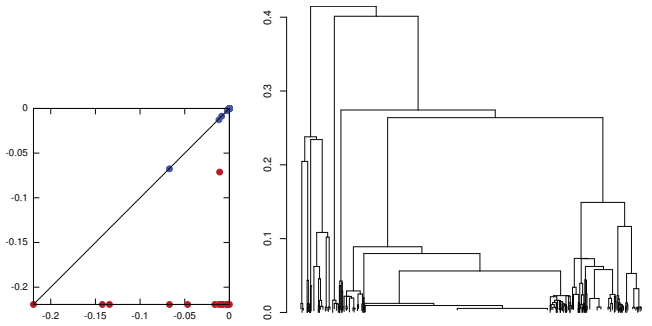
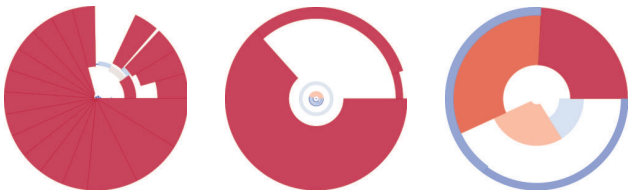


Fig. 10. Persistence diagram (left, showing H_0) and dendrogram (right) for the "Kaskal" data set. The clearly separated areas in the persistence diagram are due to the synthetic origin of the data set: There is only a single off-diagonal point of *finite* persistence; the remaining points of finite persistence are all situated along the diagonal, indicating that they have a persistence of 0. We use the dendrogram to choose $\varepsilon \in [0.075, 0.125]$.



(a). H_1 for cluster 1 (orange) (b). H_1 for cluster 4 (green) (c). H_1 for cluster 8 (red)

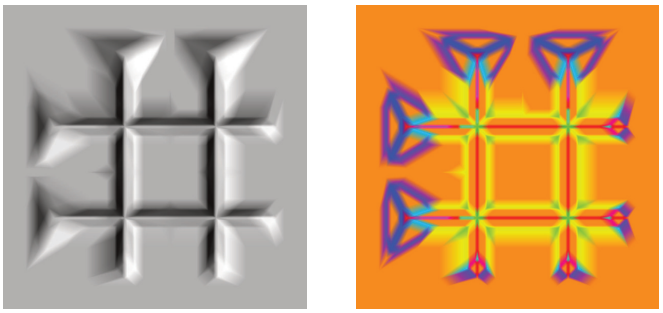
Fig. 11. Representative persistence ring visualizations of the first homology group H_1 for the "Kaskal" data set shown in Fig. 12. Fig. 11a shows the topology corresponding to the cluster containing all *planar* points. By inspecting the simplicial chains, we found that each generator corresponds to a sequence of data points bounding one of the non-planar clusters. Fig. 11b shows the topology of the cluster containing the points at the *intersections* of the cross-like structures of the "Kaskal" sign. Here, we identified a single persistent generator that bounds cluster 8. Finally, cluster 8 contains points at the "ridges" of the cuneiform signs. Again, its topology is similar to that of the other clusters.

small as possible. The topological signatures, represented as *persistence rings*, demonstrate that the multivariate feature space contains complex high-dimensional structures. These structures differ for different parts of the data set (such as the cuneiform signs themselves and the background of the clay tablet). In this analysis, we explored and explained the resulting structures. We found typical characteristics that manifest themselves in the topological signature, i.e. the persistence rings. These characteristics look alike even for *different* cuneiform signs, as we will demonstrate in the next section.

We separated the feature vectors of the "Kaskal" data set into several clusters with *unique* signatures, which we hope to further exploit in the future—ultimately, we strive to obtain a reconstruction of impressions of individual wedges or cuneiform signs.

6.2.2 "HOS_G8" data set

To test our methods on real-world data, we selected a digitized cuneiform tablet, designated "HOS_G8" by the *Heidelberger Objektsammlung*. With 344694 vertices and 689384 faces, the local resolution of the cuneiform characters is very high and covers even small details such as fingerprint. However, the high resolution also renders automatic character extraction very difficult. Our initial analysis showed that the feature vectors for real-world data contain much noise. The noise is not only caused by instabilities of the feature vectors, but also by imperfectly imprinted cuneiform characters. For the complete data set, we encountered many spurious persistence intervals that aggravated our analysis. Hence, we manually selected several regions of interest in the mesh and analysed them instead of the larger data set.



(a). Mesh with virtual lighting (b). Segmented mesh without lighting

Fig. 12. For validating our topological approach, we used a synthetic mesh depicting the "Kaskal" cuneiform sign. With 1233 vertices and 2432 faces, the underlying mesh is small enough for the feature vector calculation to work in real-time. The image to the left depicts an orthographic projection of the mesh using virtual illumination to highlight the ridges. The image to the right shows the result of our clustering algorithm. We detect 12 distinct clusters that describe regions of different curvature in the mesh—see Sec. 6.2.1 for a detailed discussion.

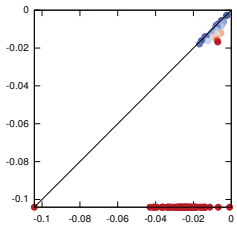
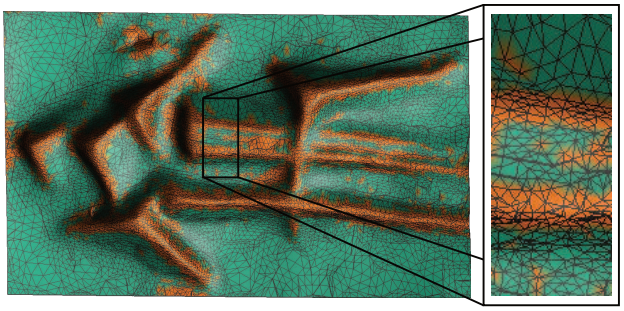


Fig. 13. Persistence diagram of H_0 for the "HOS_G8" cuneiform tablet. Note that there is much topological noise of little persistence clustered along the diagonal and many features of infinite persistence.

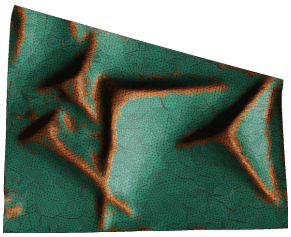
In addition to less noise, this also effectively reduced the running time of our algorithm from 300s for the complete data set to a mere 5s–10s. Despite the reduced data set, however, dendrograms proved to be *impractical* for choosing ε. We thus resorted to estimating the average distance of points in the feature space by randomly selecting a fraction of points and querying their k nearest neighbours (with $k \in [10, 20]$). For each selected point p, we calculated the mean distance m_p to its neighbours. The mean of all m_p values then yielded a good estimate of the average point distance for the complete point cloud. We then proceeded heuristically by setting ε to multiples of this estimate. While this procedure proved to be sufficiently easy to follow and provided us with several partitions, we aim to improve the selection process for ε in a future publication.

Fig. 13 shows an example persistence diagram that is obtained for one of the regions of interest of the cuneiform tablet. Note that this diagram also contains a clear separation of topological noise (along the diagonal) and topological features (along the bottom). Upon increasing ε further, these components will quickly be merged into a single large cluster. This is due to the skewed density distribution of the data set: Each mesh contains a large amount of "regular", i.e. planar or almost planar, data points, which results in a large density. The non-planar points, however, are very different from each other, and will rather be merged into the cluster of regular points. To counteract this, we selected a value for ε that resulted in a fair distribution of points and *merged* each cluster into its nearest neighbouring cluster. This resulted in a segmentation of the data set such that the *skeleton* of a cuneiform sign is extracted. Fig. 1 and Fig. 14 show our segmentations.

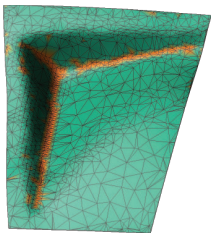
The analysis of the topology of each data set agrees with our analysis from Sec. 6.2.1. First of all, we note that those clusters that contain points in the background exhibit the same topology regardless of the number of points: Fig. 15a (for a larger region of interest) and Fig. 15c (for a smaller region of interest) show the persistence rings



(a). First region of interest

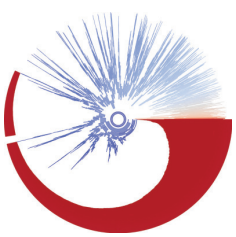


(b). Second region of interest

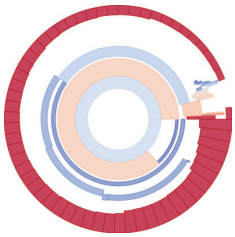


(c). Third region of interest

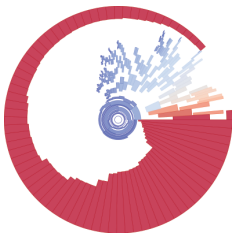
Fig. 14. Several regions of interest for the "HOS_G8" data set. Due to the high mesh density, as depicted in Fig. 14a, we removed parts of the mesh (hence the uneven borders) in order to obtain a smaller data set.



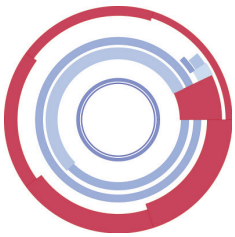
(a). H_1 of the cluster describing the background of Fig. 14a



(b). H_1 of the cluster describing the writing of Fig. 14a



(c). H_1 of the cluster describing the background of Fig. 14c



(d). H_1 of the cluster describing the writing of Fig. 14c

Fig. 15. Persistence rings for H_1, i.e. the first homology group, of the different regions of interest shown in Fig. 14.

corresponding to points in the background of Fig. 14a and Fig. 14c, respectively. The large number of generators describes the previously encountered phenomenon: The feature space contains a plethora of "holes" that bound smaller clusters (again corresponding to parts of the cuneiform characters). In contrast to that, less generators show up in the persistence rings that corresponding to the writing on the tablet—see Fig. 15b and Fig. 15d. By *merging* the smaller clusters into a single large one, we combine their topological features. Just as in the case of the "Kaskal" data set, the persistence rings suggest a *nested* relationship—which strengthens our previous analysis. Note that the topological structure of these clusters is not as *rich* as for the "background" cluster. This is caused by too much noise in the feature vectors, which explains the instabilities of the thresholding approach as well as the failure of automated character extraction. Thus, our topological analysis suggest new directions for the calculation of feature vectors. Namely, by varying the feature vector radii based on local densities, we expect better results, indicated by a richer topological structure.

7 Conclusion & Future Work

We introduced a novel method for exploring a high-dimensional data set. To this end, we coupled a topologically-based clustering algorithm with the calculation of topological signatures. We used these signatures to *distinguish* clustered objects by their topology. In addition, we introduced *persistence rings*, a novel visualization technique for the persistence intervals that are created by persistent homology calculations. We combined our new visualization with persistence diagrams and persistence barcodes, thereby allowing an *interactive* exploration of a high-dimensional data set.

We demonstrated the viability of our method using two classes of data sets: First, we provided an in-depth analysis of a synthetic data set that contains several topological objects. Our persistence rings visualization detects the differences between the individual clusters, thereby showing the advantages of coupling clustering algorithms with topological signatures. Second, we analysed the 16-dimensional feature space of filter responses for *cultural heritage* data. Our method detects hitherto unknown structures in the feature space, revealing complicated nested relationships. Moreover, by clustering the feature space, we were able to segment virtual cuneiform tablets into parts containing *writing* (i.e. cuneiform characters) and parts containing only the background. While feature space information was previously exploited by different means, our method does not suffer from instabilities in the data and requires no complicated thresholding. Thus, by operating on a 16-dimensional feature space, our segmentation facilitates cuneiform character extraction for 3-dimensional data and allows further post-processing. Automatic character extraction is an important goal for assyriologists, who have to deal with the transcription of several hundreds of thousands of (often damaged) cuneiform tablets.

For future work in this area, topological signatures should include a better *localization* of features, i.e. assigning each homology generator feature a *geometrical* meaning. In the synthetic torus data set, for example, one generator should correspond to points around the "tube" of the torus, while the other one should correspond to points around the "hole". In the video accompanying this publication, we show an implementation of a simple localization procedure. It works, however, only for extremely simple data sets. A potential solution for the localization problem was recently presented by Zomorodian and Carlsson [40].

Furthermore, the different ways of creating simplicial complexes should be examined—witness complexes [7], for example, are an alternative to the Vietoris-Rips expansion. At last, future research should investigate several *metrics* for $\mathscr{R}_\varepsilon$. Throughout this paper, only the Euclidean metric was used. However, using approaches such as *metric learning*, domain knowledge may be added to the creation of $\mathscr{R}_\varepsilon$, thereby further improving the quality of the results—see Yang [36] for a comprehensive survey.

Acknowledgments

This work was partially funded by the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences* (HGS MathComp) and the *Excellence Initiative* of the *German Research Foundation* (DFG). The cuneiform tablet data were kindly provided by Prof. Dr. Stefan M. Maul and Dr. Stefan Jakob, *Assyriologie Heidelberg*, *Assur-Forschungsstelle* of the *Heidelberg Academy of Sciences and Humanities* (HAW).

REFERENCES

[1] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46:255–308, 2009.

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

[3] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms*, SODA '00, pages 918–926. SIAM, 2000.

[4] F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Geometric inference for measures based on distance functions. Rapport de recherche 6930, INRIA, May 2009.

[5] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Persistence-based clustering in Riemannian manifolds. In *Proc. 27th Annual ACM Symposium on Computational Geometry*, pages 97–106, 2011.

[6] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, Jan. 2007.

[7] V. de Silva and G. Carlsson. Topological estimation using witness complexes. *IEEE/Eurographics Symposium on Point-Based Graphics*, pages 157–166, 2004.

[8] V. de Silva and R. Ghrist. Coordinate-free coverage in sensor networks with controlled boundaries via homology. *International Journal of Robotics Research*, 25(12):1205–1222, Dec. 2006.

[9] V. de Silva and R. Ghrist. Coverage in sensor networks via persistent homology. *Algebraic & Geometric Topology*, 7:339–358, 2007.

[10] P. Diaconis and M. Shahshahani. The subgroup algorithm for generating uniform random variables. *Probability in the Engineering and Informational Sciences*, 1(1):15–32, 1987.

[11] H. Edelsbrunner and J. L. Harer. *Computational topology*. American Mathematical Society, Providence, RI, 2010.

[12] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, 2002.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[14] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45:61–75, 2008.

[15] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3d scalar fields. In *Proceedings of IEEE Conference on Visualization*, 2005.

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

[17] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[18] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–475, 1985.

[19] S. B. Kotsiantis and P. E. Pintelas. Recent advances in clustering: A brief survey. *WSEAS Transactions on Information Science and Applications*, 1:73–81, 2004.

[20] J. M. Lee. *Introduction to topological manifolds*. Graduate Texts in Mathematics. Springer, 2000.

[21] H. Mara. *Multi-Scale Integral Invariants for Robust Character Extraction from Irregular Polygon Mesh Data*. PhD thesis, Heidelberg University, 2012 (submitted).

[22] H. Mara, S. Krömker, S. Jakob, and B. Breuckmann. GigaMesh and Gilgamesh - 3D Multiscale Integral Invariant Cuneiform Character Extraction. In *Proc. VAST Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 131–138, Palais du Louvre, Paris, France, 2010.

[23] S. Marsland. *Machine learning - An algorithmic perspective*. Chapmann & Hall / CRC Press, 2009.

[24] K. Moreland. Diverging color maps for scientific visualization. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II*, ISVC '09, pages 92–103, Berlin, Heidelberg, 2009. Springer-Verlag.

[25] J. R. Munkres. *Elements of algebraic topology*. Addison-Wesley Publishing Company, Inc., 1984.

[26] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. Visualization of high-dimensional point clouds using their density distribution's topology. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1547–1559, November 2011.

[27] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Visualization, 2002. VIS 2002. IEEE*, pages 187–194, November 2002.

[28] V. Pascucci, K. Cole-McLaughlin, and G. Scorzelli. The TOPORRERY: Computation and presentation of multi-resolution topology. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, pages 19–40. Springer, 2009.

[29] H. Pottmann, J. Wallner, Q.-X. Huang, and Y.-L. Yang. Integral invariants for robust geometry processing. *Computer Aided Geometric Design*, 26:37–60, January 2009.

[30] G. Singh, F. Memoli, and G. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *Eurographics Symposium on Point Based Graphics*, pages 91–100, 2007.

[31] G. Singh, F. Mémoli, T. Ishkhanov, G. Sapiro, G. Carlsson, and D. Ringach. Topological analysis of population activity in visual cortex. *Journal of Vision*, 8(8), 2008.

[32] L. Vietoris. Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. *Mathematische Annalen*, 97(1):454–472, 1927.

[33] M. Ward, G. Grinstein, and D. Keim. *Interactive data visualization: Foundations, techniques, and applications*. A K Peters, Ltd., 2010.

[34] G. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1416–1423, 2007.

[35] G. Weber, S. Dillard, H. Carr, V. Pascucci, and B. Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):330–341, 2007.

[36] L. Yang. Distance metric learning: A comprehensive survey. Technical report, Michigan State University, May 2006.

[37] A. Zomorodian. *Topology for computing*. Cambridge monographs on applied and computational mathematics. Cambridge University Press, 2005.

[38] A. Zomorodian. Fast construction of the Vietoris-Rips complex. *Computers & Graphics*, 34(3):263–271, 2010.

[39] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.

[40] A. Zomorodian and G. Carlsson. Localized homology. *Computational Geometry*, 41(3):126–148, 2008.