Interactive Multi-scale Mapper

Tripti Agarwal (u1319433)

May 17, 2021

1 Abstract

Producing summary of the topological information and a map defined on the data is the main aim of topological data analysis. There are various tools defined in the literature that can produce these summaries. Mapper is one of these tools with which we can produce these summaries of the data. But these summaries are always produced at a fixed scale for which the cover is created. To overcome this problem a multi-scale mapper is proposed with which multiple mappers at different scales can be created. A lot of work has been done for the implementation of mapper graph and one such method is mapper interactive. But there is no implementation available for the multi-scale mapper. We create a user interactive version of multi-scale mapper with which one can produce multiple mappers for a given dataset and find the relationship between each mapper. For quantitative analysis of the tool we also calculate the distance between each mapper.

2 Introduction

In recent years, a lot of progress has been done in creating various topological skeletons that can produce topological summaries. Various skeletons such as contour trees [3], Reeb graphs [2], mappers [7] with which we can analyze the complex and diverse data. Another tool is multi-scale mapper [4] with which we can understand the data at various scale. It builds on top of the mapper algorithm [7], by obtaining a tower of covers at different intervals or overlaps, and then constructing a link between simplices at different levels [6].

In this work, the aim is to create a visualization tool based on the MapperInteractive framework [9] to explore mapper at multiple scales. The tool can highlight related nodes at different scales based on user selection. This allows for visualizing how the feature is born or merges into a higher-level feature as a function of resolution. The evaluation of the tool is done on synthetic point cloud data.

The motivation and contribution of working on this project is to allow users to better understand their data using topological summaries. The project provides interactive analysis and visualization of high-dimensional point cloud data using multi-scale mapper.

Following are the contribution for this project:

1. We use the MapperInteractive implementation as the base of our work. This work is an extension of MapperInteractive, which leverages the Kepler-mapper python library [8]; to produce multiple mappers at different scales.

2. We also provide new parameters such as the number of mappers to produce an interval difference between each mapper. A detailed discussion of these parameters is provided in section 4.

3. Our implementation can link nodes of mapper at one interval to nodes of mapper in another interval. Additionally, we provide means to guide users towards the scales at which prominent features are created or destroyed.

4. In a multi-scale mapper, the data is divided into covers at one level, and then these covers are further divided to obtain mappers at another level [4]. In our implementation, for simplicity, we compute mappers at each level (instead of dividing the covers), and then we make links between the nodes of each mapper based on how data points are present in these covers (see section 4 for details).

5. For the quantitative analysis of the data, we calculate the distance between each pair of mapper graphs. We use approximate graph edit distance [1] to find the similarity/dissimilarity between graphs at different levels.

3 Background

For the proposed work, we need to understand some basic topological data structures.

1. **Mapper:** Mapper [7] is a tool for summarizing topological information of the data by defining a map on it. Mapper takes high-dimensional data as input, a map defined on it, and produces a summary of the data by using a cover of the codomain of the map. This cover, via a pullback operation to the domain, produces a simplicial complex connecting the data points. There are various open-source implementation for mappers available, for example Kepler mapper[8], and gitto tda mapper. Figure 1 shows a simple example of mapper graph obtained from black points marked in the figure. A filter fuction is applied on these data points (here height function), which is then projected on y-axis. We obtain 4 different covers U_1, U_2, U_3 and U_4 which compose the covering of the projected space. A pull back of these intervals i.e. preimage of the filter function is then obtained, then cluster of each preimage is obtained. Finally a nerve of the clusters is obtained. The nerve of a cover is a simplicial complex such that each cover element becomes a vertex and every time two or more cover elements intersect, their vertices are connected. Mapper is able to estimate important connectivity aspects of the underlying space of the data so that it can be explored in a visual forum. This is an unsupervised method of generating a visual representation of the data that can often reveal new insights of the data that other methods cannot.

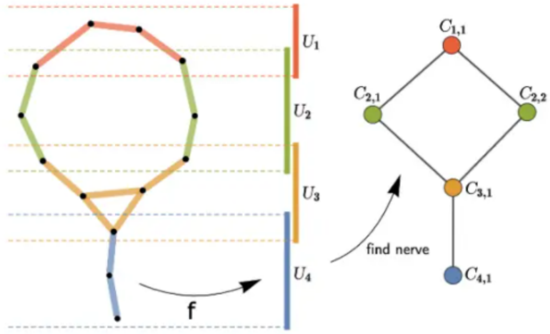

Figure 1: (left) data is represented by 2-dim black points, lens - filter function is a height function projected on y-axis, (center) covers - four intervals compose the covering of the projected space, after pull back of these intervals, i.e. taking the preimage of the filter function, we cluster each preimage in a nearest neighbour fashion. and (right) the resulting Mapper graph.

2. **MapperInteractive :** MapperInteractive [9] is a web-based framework for interactive analysis and visualization of high-dimensional point cloud build on mapper algorithm. The framework allows to import a point cloud data, change the number of intervals, overlap percentage, and mapping algorithm to produce a mapper graph. It also provides other functionality such as node selection and path selection options on the graph. Other attributes include changing the color and size of the nodes based on input data. An open-source implementation for MapperInteractive is available. Figure 2 shows an example of interactive version of mapper, in which a point cloud representing the snowman is visualized using MapperInteractive. The nodes in the graph are colored and the size of the nodes is changed based on the input data. Other parameters such number of intervals, overlap percentage and mapping algorithm is set using the interactive tool.

Figure 2: MapperInteractive on snowman point cloud.

3. **Multi-scale Mapper :** Multi-scale mapper is proposed by Dey et al [4]. Mapper graphs are implemented at a fixed scale and hence do not provide flexibility in analyzing the data. Multi-scale mapper is found by computing the nerve of each cover, of a tower of covers. Figure 3 shows the tower of covers. Here, the nerve of a tower of covers is a tower of simplicial complexes. With multi-scale mappers, we can create multiple graphs at different intervals, which helps to better understand the data. As proposed in [4], multi-scale mapper works on one concept, i.e., changing the number of covers. An extension of the multi-scale mapper is proposed by Piekenbrock et al. [5], in which we can produce multiple mappers by keeping the number of intervals fixed by changing the overlap percentage.

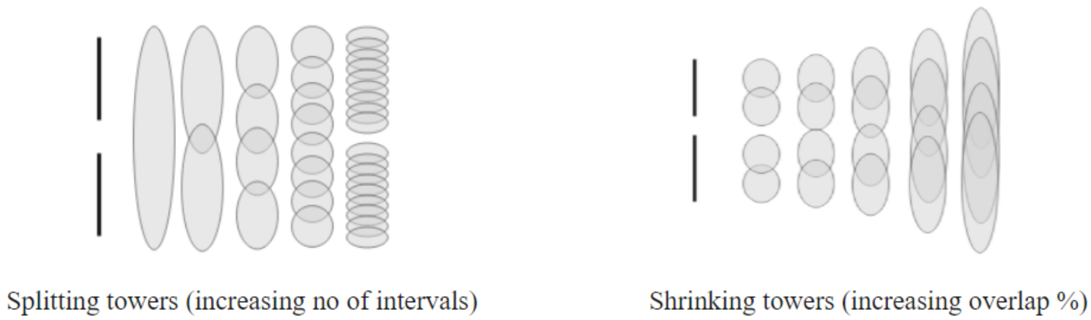

Figure 3: (Left) Increasing number of intervals. (Right) Increasing overlap percentage.

Due to the instability when changing the scale, the clusters and the connection between these clusters can change. We need to use multi-scale mappers to better understand the data as it allows us to visualize the mappers at different scales. But there are no interactive tools for multi-scale mappers. This gives us the motivation to implement an interactive visual tool that can take point cloud data and produce multiple mappers at different scales as per the user's choice.

4 Method and my major contribution

We start by analysing the work already available in the literature. We studied two tools before starting to work for the project i.e Kepler mapper implementation and mapper interactive. We used mapper interactive which is build using Kepler mapper implementation.

1. To create a multi-scale mapper following new parameters were added in the front end to produce appropriate results.

   - Start interval: What is the number of intervals in the coarsest mapper graph?
   - Number of intervals: How many intervals to compute mapper graphs for?
   - Stride: What is the spacing between intervals?

   This is completely done by me.

   For example if start interval is 3 , number of intervals is 5 and stride is 2 then following intervals [3,5,7,9,11] are computed. This maps to "number of cubes" in the cover as defined in Kepler mapper. Interval n implies that the range of the lens is split into n^2 intervals.

2. Next using the input parameters different mapper graphs are produced. These graphs are then passed to front end and displayed in a grid.

   The work division here is that, I produced the graphs (back-end work), and then produced it on the front-end. These graphs were overlapping initially, so Karthik (project partner) created a grid for proper display.

3. Next we find the relationship between the nodes of the mapper graphs at different scales. Each node corresponds to a cluster of data points, as determined by the underlying clustering algorithm (DBSCAN is used in all our experiments). Given two mapper graphs at different scales, M_A and M_B. A node N_A in M_A is linked to a node N_B in M_B iff:

$$|D(N_A) \cap D(N_B)| > 0 \tag{1}$$

   Here, $D(N)$ is a function that returns data points corresponding to the nodes N. In other words, we establish a link between nodes at different scales if the two nodes share at least one data point. We use this information to highlight linked nodes in the frontend.

   Here the proposal of the above technique was proposed by me and Karthik did the implementation.

4. Since mapper graphs define a simplicial complex, we can compute Betti numbers on them. In our work, this corresponds to β_0, the number of connected components; and β_1, the number of loops in the dataset. This information is relayed to the user in the frontend with indications where it changes from one scale to the next. This allows the user to determine what scales potentially have useful information.

   The proposal and implementation of Betti numbers is completely mine.

5. Finally, we evaluated the result of multiple graphs by comparing the distance between mappers at different intervals. However, there is no consensus on a good distance metric between mapper graphs. We treat each mapper graph as a node and set of edges and then compute approximate graph edit distance(exact solutions are NP-hard and can take a long time) between graphs. This corresponds to the number of changes(insertions/deletions) required to transform one graph to the other.

   The implementation is done completely done by me.

5 Evaluation

Our work must primarily be evaluated qualitatively: based on the implementation and usability of our tools. While we do not have a comprehensive user study, we have demonstrated that our project helps users better understand topological summaries of their dataset.

- While a dataset can look like a simple loop from a coarse scale, the user can use this higher-level topological summary to zone in on finer details using our tool. If the user were to simply run mapper with a fine-scale interval, it'd be very difficult to identify what a fine-scale feature represents in a sea of nodes.
- We show that two near-identical datasets can be differentiated when studied at different scales.

- We use visual aids to indicate to the user the scales at which a significant topological difference is likely. This is via the Betti number indicators.

We evaluated the result of multiple graphs by comparing the distance between mappers at different intervals. However, there is no consensus on a good distance metric between mapper graphs. We treat each mapper graph as a node and set of edges and then compute approximate graph edit distance(exact solutions are NP-hard and can take a long time) between graphs. This corresponds to the number of changes(insertions/deletions) required to transform one graph to the other.

6 Datasets and their results

Data
The dataset consists of point clouds stored in CSV files. In our experiments, we apply the method to 2D point clouds. The first being a point cloud data of a ring. The ring consists of a big central hole. We used the second data set as a point cloud data of a ring with a small hole in it. This makes the data consisting of two holes in it. With our tool, we aim to achieve multiple mappers with fixed attributes to show the difference between the two datasets. Another dataset that we used is a point cloud data of the deer. With this data we aim to achieve finer details of the deer like the horns and legs as the mapper level changes. Overall we aim to capture the shape of the data and various topological properties of the data at different mapper levels.

Results
We demonstrate our results on three datasets.

1. **Ring:** In order to study this dataset, we use a stride of 2 and an overlap of 23%. The overlap % is dependent on the data being studied and was determined by trial and error. (This could potentially be alleviated by running a multi-scale mapper on different overlaps in addition or instead of multiple intervals). Once this is loaded into our tool, we can click on different nodes to visualize how these features split into other nodes at finer scales.

   We can observe how different parts of the ring are changed based on the resolution at which we observe the data. Since this only contains a single loop, we only observe the number of nodes increasing as we go to finer scales, but there is no difference in overall topology.

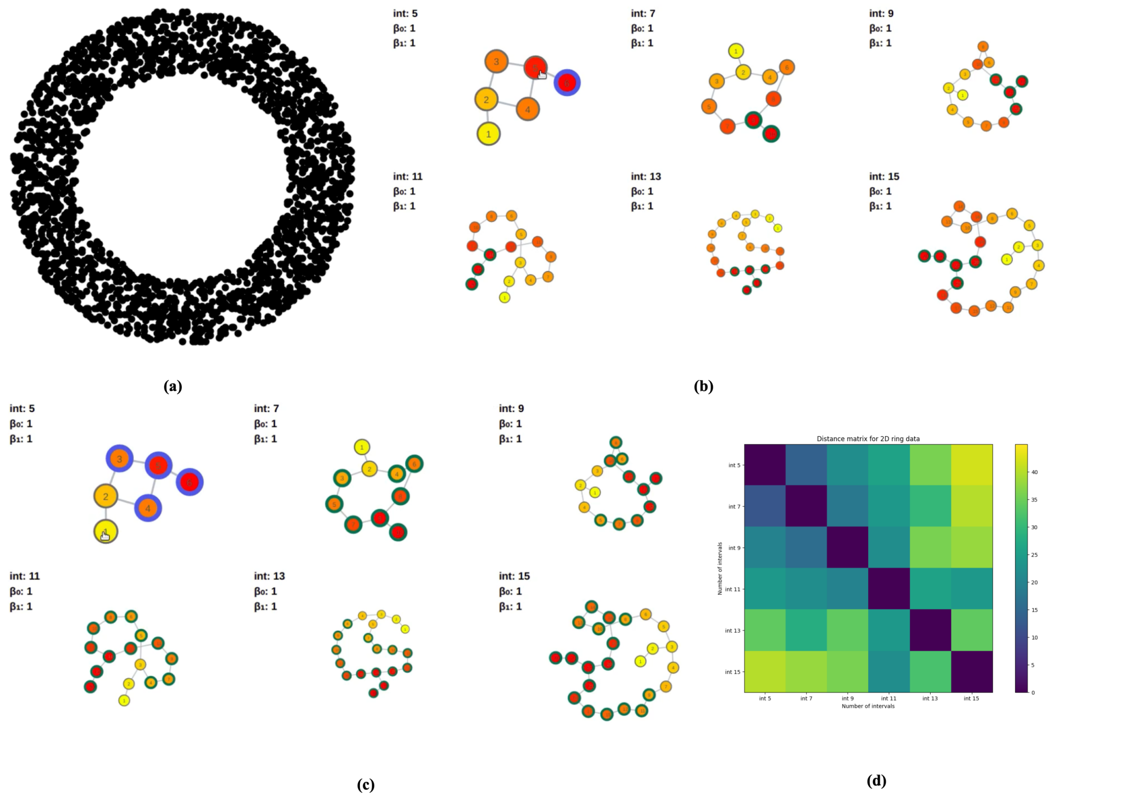

Figure 4: (a) Point Cloud of 2D Ring Dataset. (b)A single node selected from the end of the ring at the coarsest resolution. In the finer scales, we can see that this corresponds to other parts of the ends of the loop. Blue: Selected by user, Green: Highlighted by tool. (c) Multiple nodes selected from the coarsest resolution. In the finer scales, we can see how these features evolve. Blue: Selected by user, Green: Highlighted by tool. (d) Approximate Graph Edit Distance between Mapper graphs at different intervals for the 2D Ring dataset.

2. **Ring with hole:** Next, we study a nearly identical dataset, except for a hole in the lower-left portion of the rink. We use the same hyperparameters as the previous dataset.

   We observe that the β_1 increases by 1 in the finest scale. This implies that an additional loop has been formed. We can zoom in on that particular scale's mapper graph to verify this. If we select the nodes in that loop, we can observe how it maps to other parts of the ring at coarser scales. Finally, by selecting the entire loop in the coarsest scale, we can visualize how it eventually splits into a smaller sub-loop. In this dataset, we can see the sub-loop form at interval 15. Plotting a distance matrix of graph edit distances between graphs at different intervals in each of the two datasets, we see a slightly larger increase in the distance at the 15th interval in the dataset with holes.

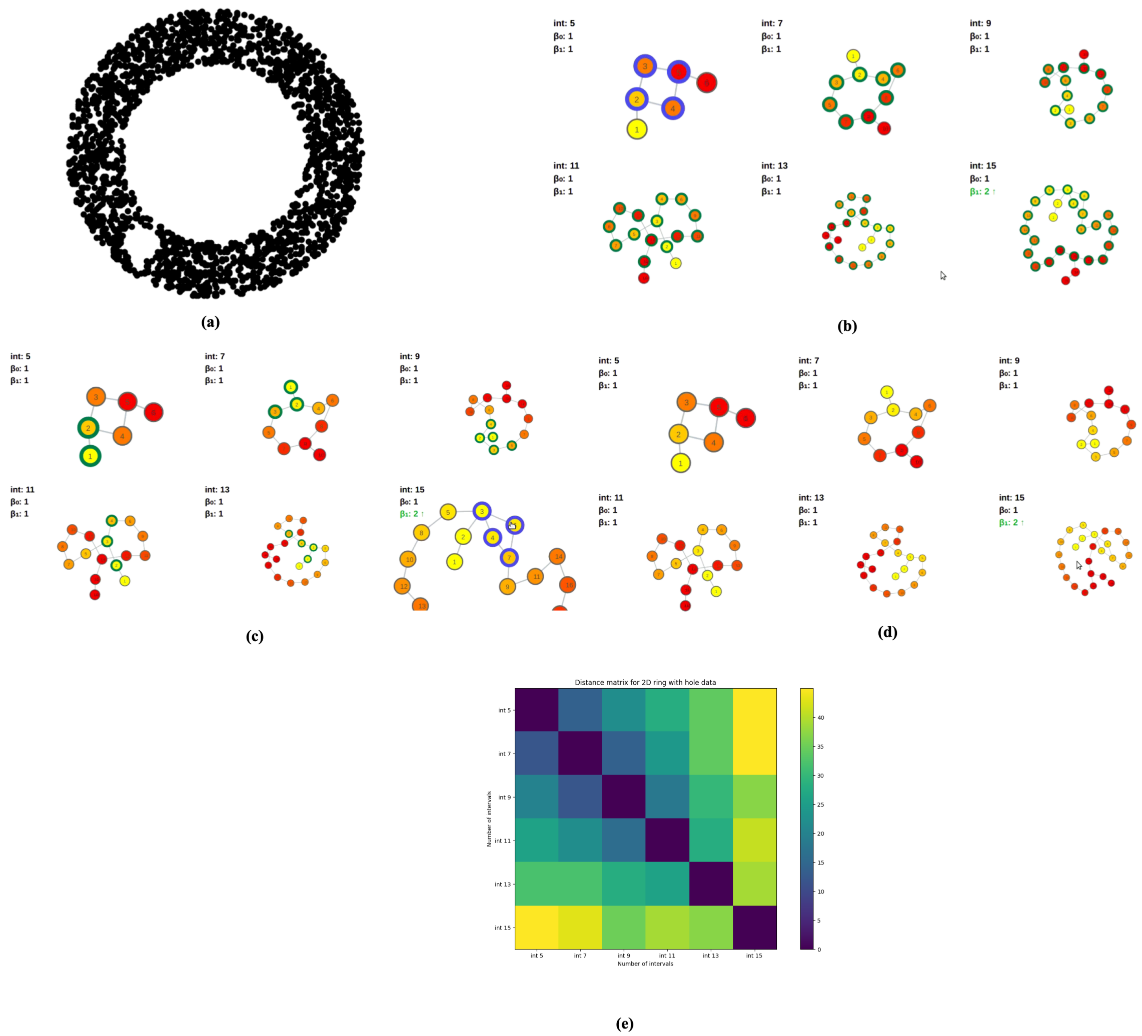

Figure 5: (a) Point Cloud of 2D Ring-with-hole Dataset. (b)Selecting the entire large loop at the coarser scale, we can see how it evolves in finer scales, and the fact that it contains the smaller inner loop. Blue: Selected by user, Green: Highlighted by tool. (c) Zooming in at the finest scale, we can select the smaller loop and observe how it evolved from coarser scales. Blue: Selected by user, Green: Highlighted by tool. (d) Notice that β_1 increases in the finest resolution. This indicates that there may be an interesting topological feature at that scale. (e) Approximate Graph Edit Distance between Mapper graphs at different intervals for the 2D Ring with hole dataset.

3. **Deer:** We study results obtained from running mappers at different scales on a point cloud obtained from the silhouette of a deer. This was chosen from the MPEG7 dataset. On the coarse scale, the door is topologically equal to the ring. However, as the scale increase, we see that it devolves into a confusing sea of nodes. By clicking individual nodes on the coarse-

scale, we can explore how the finer-scale features originate. When a node from the head is selected, it maps the head plus antlers in finer scales. Similarly, the origin of the two hind legs can be observed. Due to the nature of the graph edit distance metric, the difference is very subtle. We can compute the same distance metric for the deer dataset, but the metrics are less explanatory.

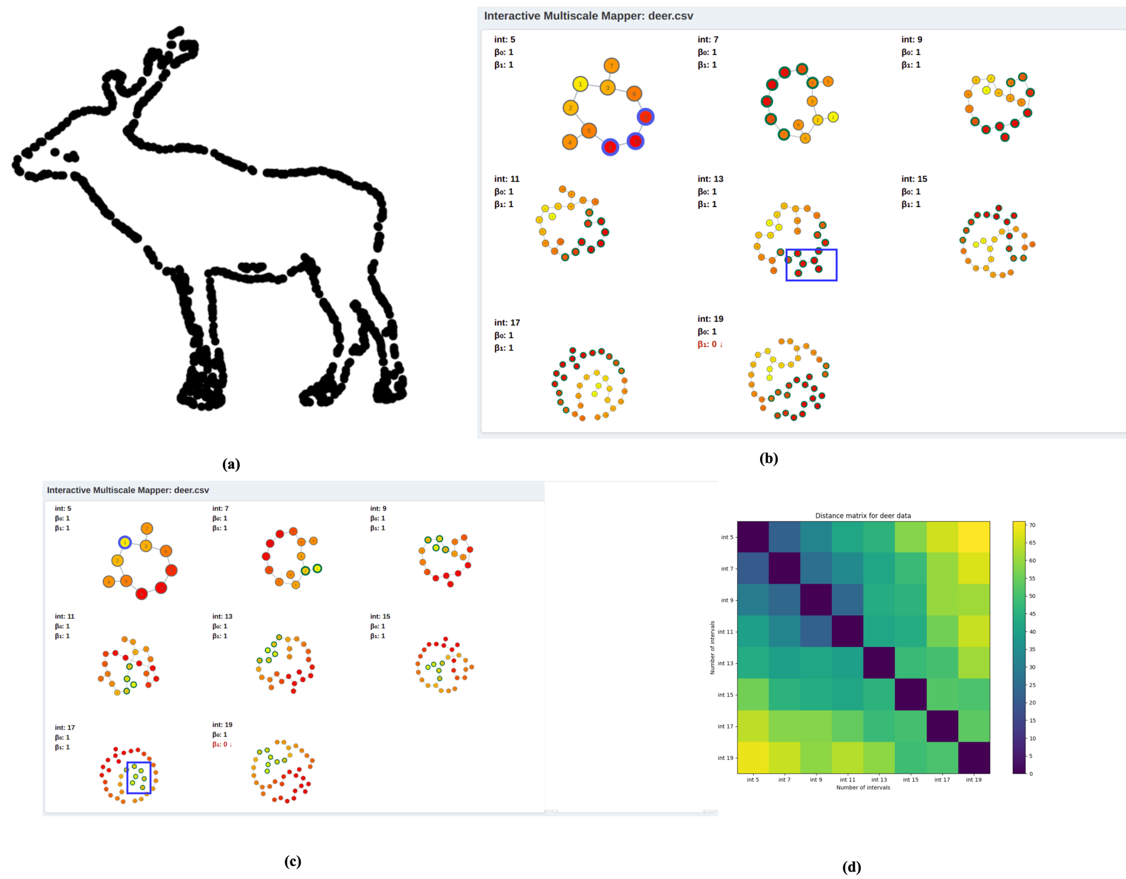

Figure 6: (a) Point cloud for deer dataset. (b) We can see how the hind-legs and hooves of the deer evolve from the finer scale. Blue: Selected by user, Green: Highlighted by tool. (c) Observing the head of the deer, we can see how it evolves into the antlers at finer scales. Blue: Selected by user, Green: Highlighted by tool. A blue marking is done to show the two horns of the deer. (d) Approximate Graph Edit Distance between Mapper graphs at different intervals for the deer dataset.

7 Conclusion and Future Work

This project extends the MapperInteractive framework to allow for the study of data(typically point clouds) at multiple scales. We have designed intuitive means to track a feature/node at different scales and highlight to the user the scales at which the topology of the underlying data changes. We design an interactive visualization tool to apply the mapper framework at different scales, as well as study the relationship between nodes at various scales. We also demonstrate the need for such a tool on synthetic datasets. We discover that point clouds often have features that are glossed over when Mapper users select intervals that are too large. Selecting a very fine scale would have the opposite effect: there are too many topological features to make sense. Our work demonstrates that by running mapper at different scales, we can relate a higher-level feature to what it maps into as we reduce the scale.

As with most visualization tools, user studies are required to better understand what they would require as far as interactivity and presentation go. While we have implemented a single means of specifying intervals and a fixed overlap, it is worthwhile to explore if changing other hyperparameters to obtain finer scales is preferable in certain scenarios. Lastly, these extensions could be merged with the MapperInteractive codebase to increase adoption and obtain feedback.

References

[1] Zeina Abu-Aisheh et al. "An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems". In: *4th International Conference on Pattern Recognition Applications and Methods 2015*. Lisbon, Portugal, Jan. 2015. DOI: 10.5220/0005209202710278. URL: https://hal.archives-ouvertes.fr/hal-01168816.

[2] Silvia Biasotti et al. "Reeb graphs for shape analysis and applications". In: *Theoretical computer science* 392.1-3 (2008), pp. 5–22.

[3] Hamish Carr, Jack Snoeyink, and Michiel Van De Panne. "Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree". In: *Computational Geometry* 43.1 (2010), pp. 42–58.

[4] Tamal K. Dey, Facundo Mémoli, and Yusu Wang. "Mutiscale Mapper: A Framework for Topological Summarization of Data and Maps". In: *CoRR* abs/1504.03763 (2015). arXiv: 1504.03763. URL: http://arxiv.org/abs/1504.03763.

[5] Matt Piekenbrock, Derek Doran, and Ryan Kramer. "Efficient Multi-Scale Simplicial Complex Generation for Mapper". In: ().

[6] Nathaniel Saul. *Nerves and Towers III: Towers*. https://sauln.github.io/blog/towers-towers/.

[7] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. "Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition". In: *Eurographics Symposium on Point-Based Graphics*. Ed. by M. Botsch et al. The Eurographics Association, 2007. ISBN: 978-3-905673-51-7. DOI: 10.2312/SPBG/SPBG07/091-100.

[8] Hendrik Jacob van Veen et al. *Kepler Mapper: A flexible Python implementation of the Mapper algorithm*. Version 1.4.1. Oct. 2020. DOI: 10.5281/zenodo.4077395. URL: https://doi.org/10.5281/zenodo.4077395.

[9] Youjia Zhou et al. *Mapper Interactive: A Scalable, Extendable, and Interactive Toolbox for the Visual Exploration of High-Dimensional Data*. 2020. arXiv: 2011.03209 [cs.CG].