Localized Topological Simplification of Scalar Data [2]

Joy Lim, Hongyuan Zhu

Abstract—In topological data analysis, it is common to first simplify scalar data. The paper proposes a localized topological simplification(LTS) method. Given a set of wanted local extremes and a scalar field f. The LTS method generates another function g, approximating the original scalar field f. Meanwhile, g keeps all desired local extremes and smooth regions around unwanted extreme points. Since the proposed method focuses on unwanted areas, shared-memory parallelism can boost the performance compared to traditional methods. The goal of our final project is to implement the algorithm proposed in this paper.

1 Team members

Joy Lim
Hongyuan Zhu

2 Introduction

The growth in size and complexity of datasets advanced analysis tools to extract, interpret, and visualize meaningful information effectively. Topological data analysis (TDA) offers efficient techniques to extract structural information from data. An advantage of TDA is its ability to representations that separate noise from significant features. Simplification techniques can be categories into two strategies: bridging, which connects undesired contours to desired ones, and flattening, which removes undesired contours entirely. However, both methods are limited to handling extremum-saddle pairs and can change the shape of the data. Thus, the localized topological simplification algorithm has been proposed. We propose to implement the localized topological simplification algorithm to remove unwanted extremes while maintaining the geometry.

3 Background

3.1 General Topological Simplification

General topological simplification reduces the critical points of f to form a subset $C_g \subseteq C_f$ in a simplified scalar field g, while maintaining identical indices and locations. This simplification ensures that g remains close to f with a small $\|f-g\|_\infty$ for data fitting.

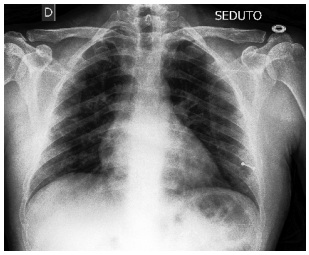
3.2 Topological Persistence

Persistence assesses the importance of a critical point by measuring how long a connected component in the sub-level or super-level set persists as the isovalue decreases. When two connected components (created at maxima) merge at a saddle, the younger component dies in favor of the older one, forming a persistence pair. The persistence value $p(\langle s, m_0\rangle) = f(m_0) - f(s)$, defined as the difference between the function values of the critical points in the pair, quantifies the significance of the feature.

4 Previous Work

Topological simplification techniques can be applied either as pre-simplifications of scalar data. Pre-simplification methods are classified into numerical and combinatorial approaches. Numerical methods optimize scalar fields with constraints on preserved extrema, often focusing on smoothness using techniques like Laplacian or bi-Laplacian smoothing. Unfortunately, this can be unstable and computationally expensive. Combinatorial methods guarantee correctness by construction and simplify features often using bridging. However, this method can introduce undesirable artifacts and post-processing steps. Compared to the traditional approaches, LTS method improves efficiency and parallelization by focusing only on undesired extrema.

5 Results



(a) Downsized CT scan

Fig. 1: Visualization of CT Scan.

5.1 Data Preprocessing

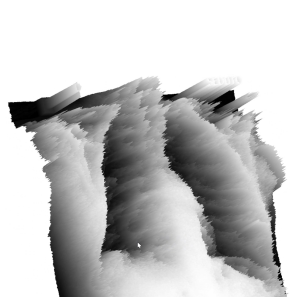
We are using COVID-19 dataset which contains CT scan pictures of patients. To make it align with homework code, we use greyscale of the picture as scalar, and pixel-wise gradient as vx,vy,vz. Since the original pictures have size of 4kx3k, we downsample its size to 424x348. We use this as original data for our project and its visualization is shown in (a) and (b).

As no smoothing was applied to the original data. There are massive local extremes, which makes the size of super level set extremely small. In other words, most simplification would be hard to visualize as shown in (b). To reduce the number of local extreme points, we first apply Gaussian filters to the original data so that curves are smoothed. another side effect of Gaussian filter is that the size of super super-level set increases.

Runtime: Non parallel runtime: 24 seconds
Parallel runtime: 24 seconds with 49 threads in total.
sigma for gaussian filter $\sigma = 2$.

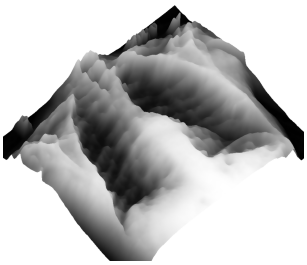
5.2 Things Not Implemented

The paper mentioned the cases when super-level set overlap with each other. In this case, each super-level set simplifies the nonoverlapping part and merges the final outcome at the end. We did not implement that, which may be the main reason why our simplification is not visually straightforward on covid-19 CT scan. Besides that, we did not implement the auto-extreme selecting process as mentioned in the paper mainly because the project would be too big for our group. In our implementation, we randomly select some extremes to simplify.

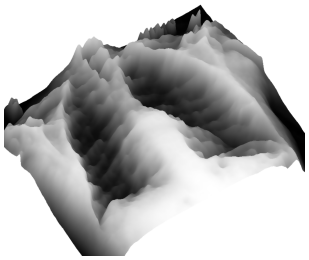


(a) Original data without topological simplification

(b) Original data with topological simplification

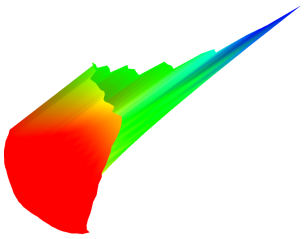


(c) Topological simplification without Gaussian smoothing



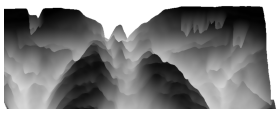
(d) Topological simplification with Gaussian smoothing

(e) Visualization of original r10.ply



(f) Visualization of topological simplified r10.ply

(g) Visualization of word area simplified



(h) Visualization of word area original

Fig. 2: Visual Evaluation of Topological Simplification.

6 Evaluation

The data sets to which we wish to apply our visual techniques are COVID-19 images, which are collected from the Covid-19 Image Dataset of Kaggle open data sets [1]. We evaluated the implementation by comparing the before and after of the simplification process using the project template visualization options.

6.1 Runtime

In our implementation, the runtime is the same and not as efficient as we expected because most of the super-level sets are small, with several being large, thus the thread with the large super-level set dominates the runtime. Since our implementation randomly selects super-level sets to simplify and most selected superlevel sets are small, we should be able to see larger performance gains if we select several larger superlevel sets.

6.2 r10

Since most super-level sets are small, visualization of simplified super-level sets is not straightforward. To illustrate the effect of our implementation, we use r10 as one example. (e) is the original r10, it might look different because we modify the height function. In (f) the local maximum in front is simplified and all smaller scalars between two local maximums are raised to the level of the original local maximum. Now simplified r10 has only one local maximum, which is also the global maximum. This follows the scheme of local topological simplification as mentioned in the background. LTS method raises points in the super-level set to the level of the original local maximum because this scheme has less impact on overall topological features like topology persistency.

6.3 CT Scan

we cannot observe large simplification in our outcome. However, our implementation does simplify some small areas. In (g) we can see that the word SEDUTO in the original picture is simplified compared to the original visualization in (h). This align with our observation that most super-level sets are small. Therefore, most simplifications our implementation performed are small.

7 Division of Tasks

Joy Lim: Worked on the data preprocess for the CT scans (jpeg). In Python, each file is processed individually and is downsized, normalized, and made injective. I attempted to generate a quad mesh, but ended up with a triangle mesh. The extraction is exported to a ply file. Hongyuan revised it.

Hongyuan Zhu: implement the super-level set propagation process. Implement reordering, order validation, and final merge process.

8 Conclusion

The localized topological simplification of scalar data is proposed as an efficient and fast part of topological data analysis by reducing time spent on data simplification. Although our implementation has its limitations, the LTS method is beneficial and an essential part of data pre-processing because of its ability to processes regions that actually need to be simplified, remove unwanted extrema, and preserve the scalar field.

Author Information

Joy Lim is with Oregon State University.
E-mail: limjoy@oregonstate.edu

Hongyuan Zhu is with Oregon State University.
E-mail: zhuhon@oregonstate.edu

REFERENCES

[1] Kaggle. Covid-19 image dataset, 2019. 2

[2] J. Lukasczyk, C. Garth, R. Maciejewski, and J. Tierny. Localized topological simplification of scalar data. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):572–582, 2021. doi: 10.1109/TVCG.2020.3030353 1