TTK as a Teaching Platform

Joshua A. Levine, University of Arizona, Oct. 2018

josh@email.arizona.edu
http://www.cs.arizona.edu/~josh

Courses Where TTK Has Been Deployed

Graduate Student Courses:

- Levine: https://jalevine.bitbucket.io/csc630/
  - Fall 2017, TTK only
- Tierny/Chazal: https://www-pequan.lip6.fr/~tierny/topologicalDataAnalysisClass.html
  - Spring 2018, TTK + Gudhi
- Wang: http://www.sci.utah.edu/~beiwang/teaching/cs6965-spring-2018.html
  - Spring 2018, Multiple advanced VIS topics, including a module on topology

Student Experience Levels

Courses targeted at graduate students, MS and PhD students

Minimum prerequisites:

- Limited Mathematical Background
- Reasonable Coding Background (C/C++)

In some cases, students not majoring in CS invited to attend

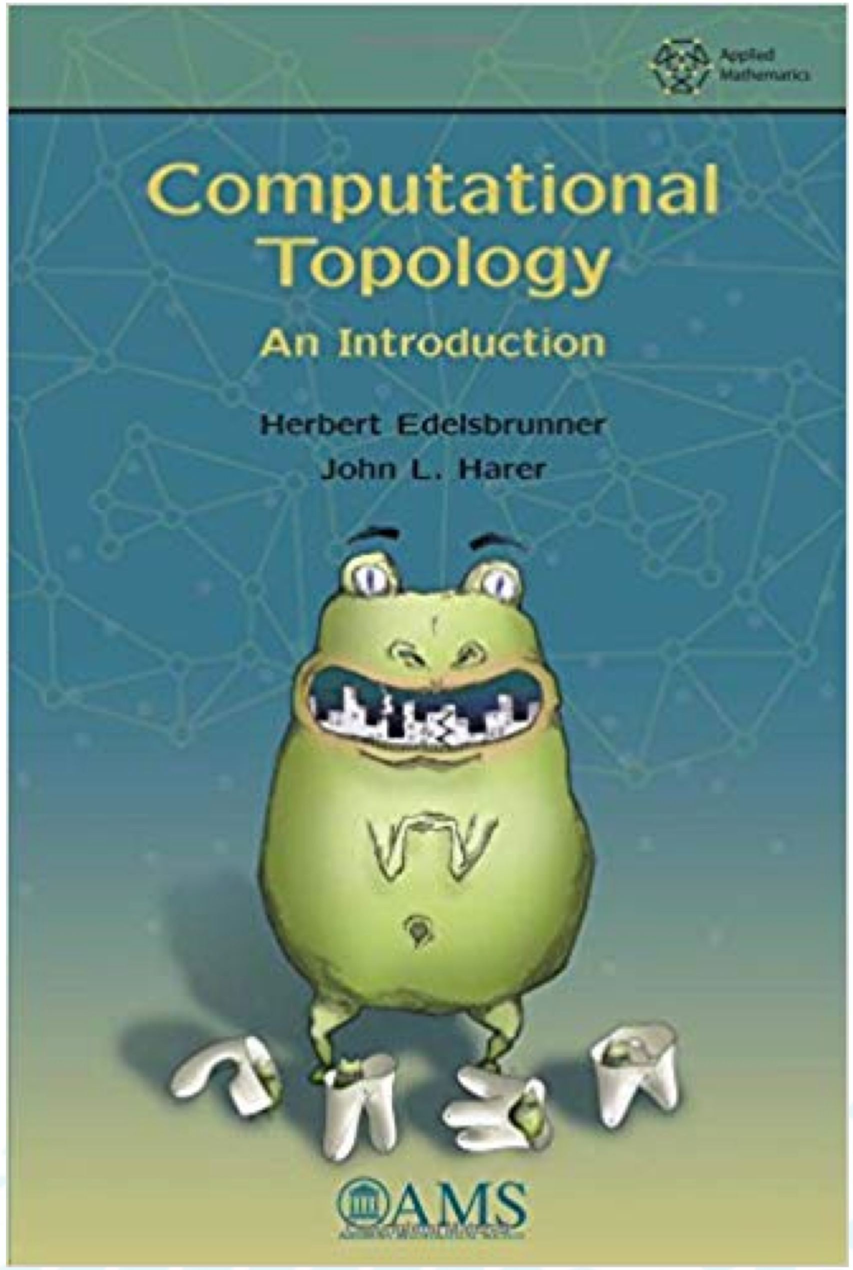
Why Did I Use TTK?

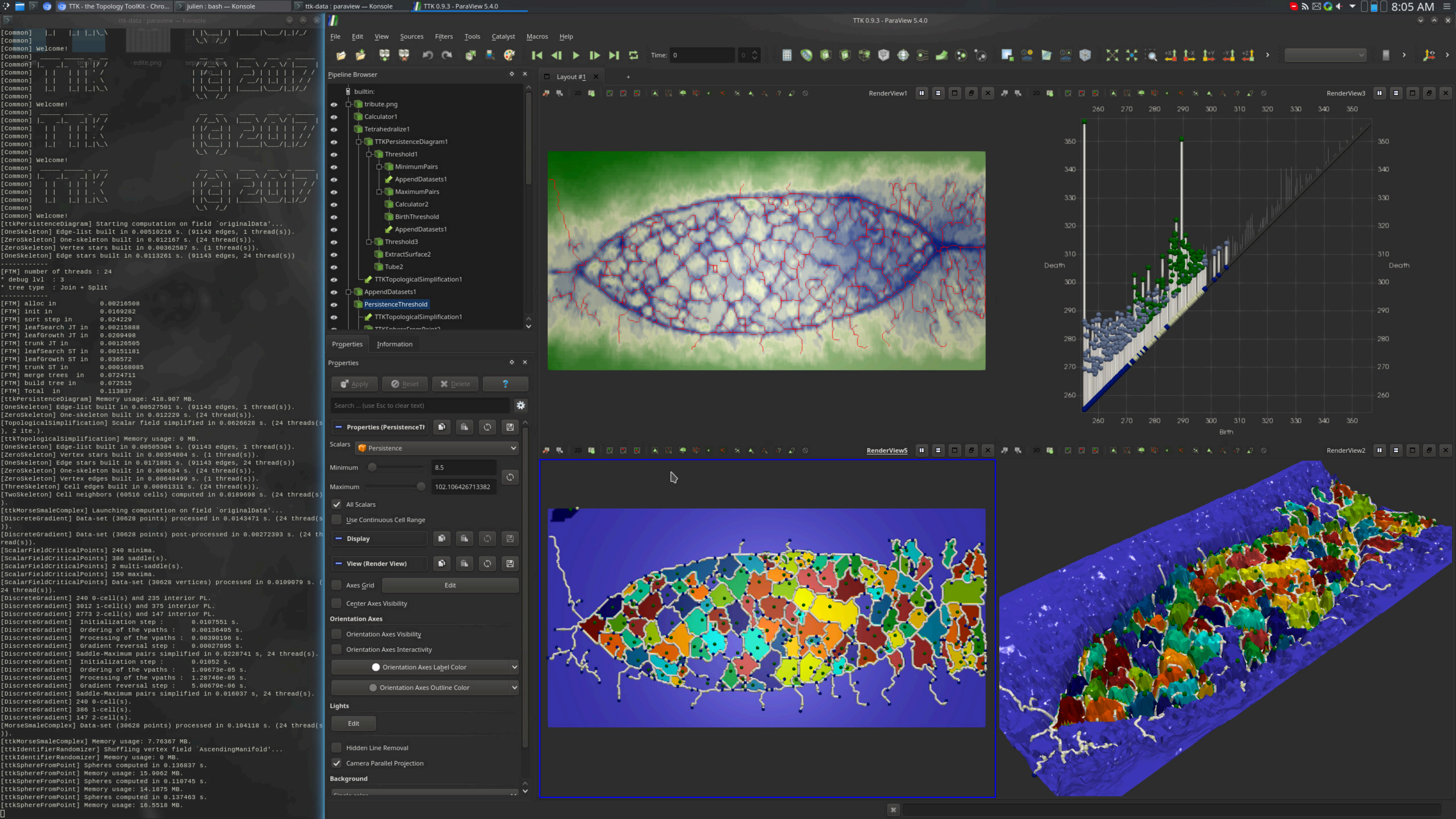
Goals of My Course at Univ. of Arizona

1.) Introduce students to topological analysis

Approach: We spent ~10 weeks (about 30 hours total) working as a reading group and covered most of Edelsbrunner and Harer's Computational Topology: An Introduction

Occasionally, we demoed actual examples from the book!




TTK - the Topology ToolKit - Chro...
julien : bash — Konsole
ttk-data : paraview — Konsole
TTK 0.9.3 - ParaView 5.4.0
8:05 AM
File
Edit
View
Sources
Filters
Tools
Catalyst
Macros
Help
Time: 0
Pipeline Browser
builtin:
tribute.png
Calculator1
Tetrahedralize1
TTKPersistenceDiagram1
Threshold1
MinimumPairs
AppendDatasets1
MaximumPairs
Calculator2
BirthThreshold
AppendDatasets1
Threshold3
ExtractSurface2
Tube2
TTKTopologicalSimplification1
AppendDatasets1
PersistenceThreshold
TTKTopologicalSimplification1
Properties
Information
Apply
Reset
Delete
Search ... (use Esc to clear text)
Properties (PersistenceTh
Scalars
Persistence
Minimum
8.5
Maximum
102.106426713382
All Scalars
Use Continuous Cell Range
Display
View (Render View)
Axes Grid
Edit
Center Axes Visibility
Orientation Axes
Orientation Axes Visibility
Orientation Axes Interactivity
Orientation Axes Label Color
Orientation Axes Outline Color
Lights
Edit
Hidden Line Removal
Camera Parallel Projection
Background
Layout #1
RenderView1
RenderView3
RenderView5
RenderView2
Death
Birth
[Common] Welcome!
[ttkPersistenceDiagram] Starting computation on field `originalData'...
[FTM] number of threads : 24
* debug lvl : 3
* tree type : Join + Split
[ttkMorseSmaleComplex] Launching computation on field `originalData'...
[ScalarFieldCriticalPoints] 240 minima.
[ScalarFieldCriticalPoints] 386 saddle(s).
[ScalarFieldCriticalPoints] 2 multi-saddle(s).
[ScalarFieldCriticalPoints] 150 maxima.
[ttkIdentifierRandomizer] Shuffling vertex field `AscendingManifold'...
[ttkSphereFromPoint] Spheres computed in 0.136837 s.

IX.3 Persistence for Image Segmentation

In this section, we discuss image data and, in particular, the problem of segmenting the data into meaningful pieces. A popular approach is the watershed method, but it is sensitive to noise in the data, tending to overdo the segmentation. We show how to use persistent homology to cope with this difficulty.

Background. When we collect data about a physical phenomenon, we do so to varying degrees of resolution. Images are high-resolution data sets, representing shapes and scenes in great detail. A large part of biological and medical research, as well as medical practice, depends on technology that produces 2- and 3-dimensional images. But we can go beyond three dimensions, eg. with video sequences that unwind in time. There are also reasons for generating images synthetically, using methods such as Fourier transforms, for finding symmetries and for other purposes. The high resolution of image data suggests we think of it as a continuous object and apply methods from continuous rather than discrete mathematics for its analysis. By its nature, an image contains more than the desired information. Therefore the first task is often the extraction of interesting features. Capturing and describing these features is the province of image analysis. It includes tasks such as denoising, segmentation, registration, comparison, and more.

Segmentation. Once an image is acquired, it becomes a mathematical object that we can study. In particular, a 2-dimensional image is a function that is piecewise constant on a rectangular configuration of pixels; see Figure IX.8. Inside a pixel, the function is constant and measures intensity or gray value, and similarly for color pictures, except that we get three separate images, one for red, one for green, and one for blue. Given an image, the *segmentation problem* looks to identify

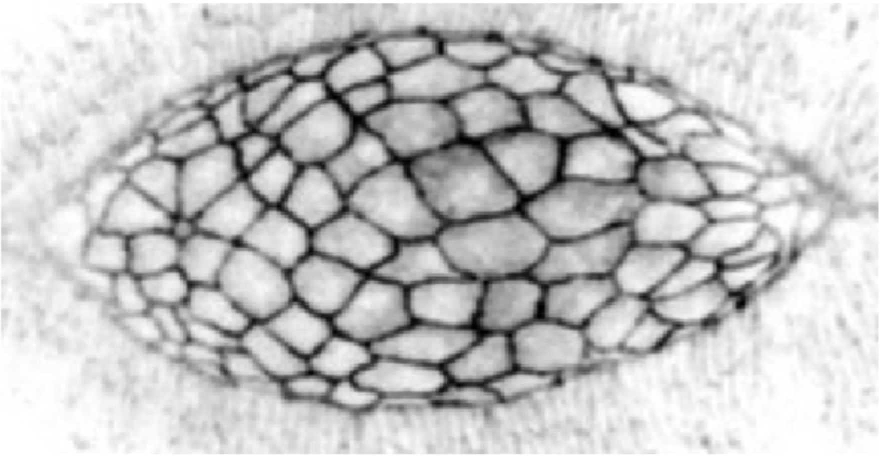


Figure IX.8: Confocal microscopy image of a cross-section of the cells in a Drosophila embryo during the developmental process known as dorsal closure [image courtesy of Daniel P. Kiehart and Adrienne Wells, Biology, Duke University].

regions of interest from these values. In Figure IX.8, these regions would be the cells imaged by microscopy. The general problem is hopelessly ill-defined but of major importance. As a result, the state of the art in the field is at best imperfect.

Goals of My Course at Univ. of Arizona

2.) Gain knowledge about how topological analysis is used to analysis data in VIS

Approach: We spent ~5 weeks (about 15 hours total) surveying research papers

Students completed final projects often based on these papers.

How Did Students Use TTK?

Assignments

Usage/Analysis Tasks:

- Persistent Homology
- Exploring Topological Structures
- Segmentation of Scientific Data

Goals for these assignments typically were to better understand key concepts and structures in topological analysis

In the course at UA, I also had written exercises based on EH's book

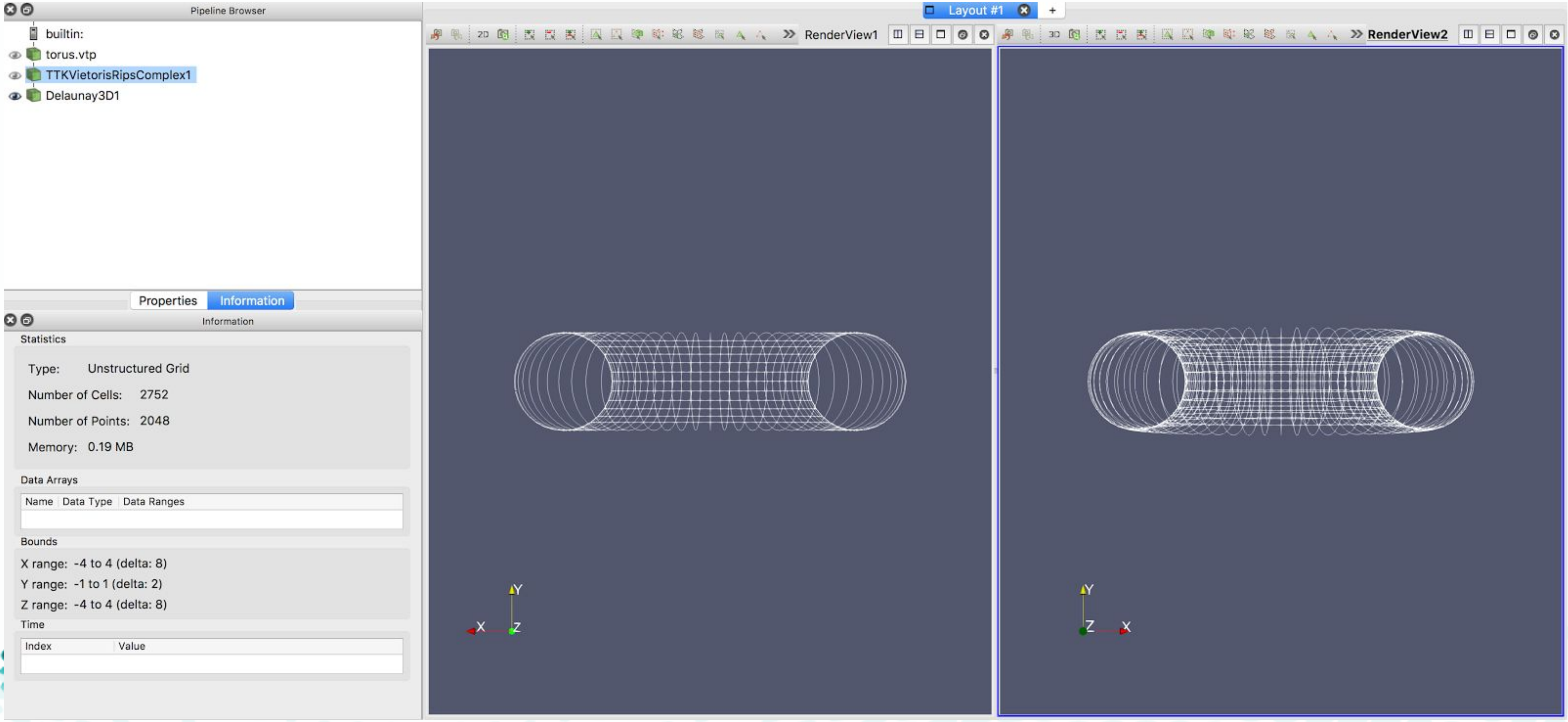
Assignments

Tasks for Coding New Modules:

- Betti Numbers / Euler Characteristics
- Vietoris-Rips Complexes
- Critical Point Extraction/Classification
- PL Morse Complex Approximation

Goals of these assignments typically were to gain mastery with implementing topological tools

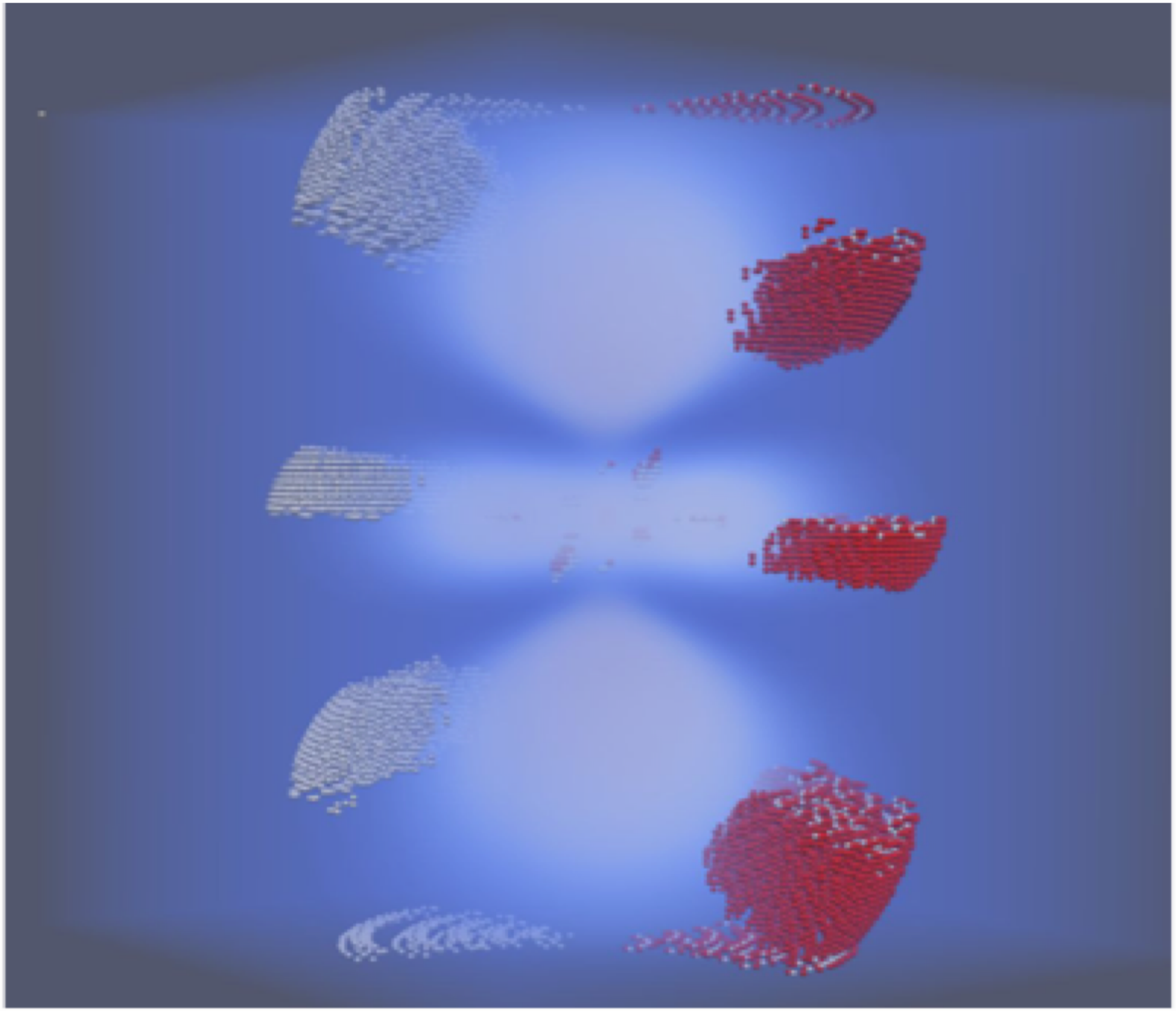
Comparing V-R Complexes to Alpha Shapes



Critical Point Extraction / Classification

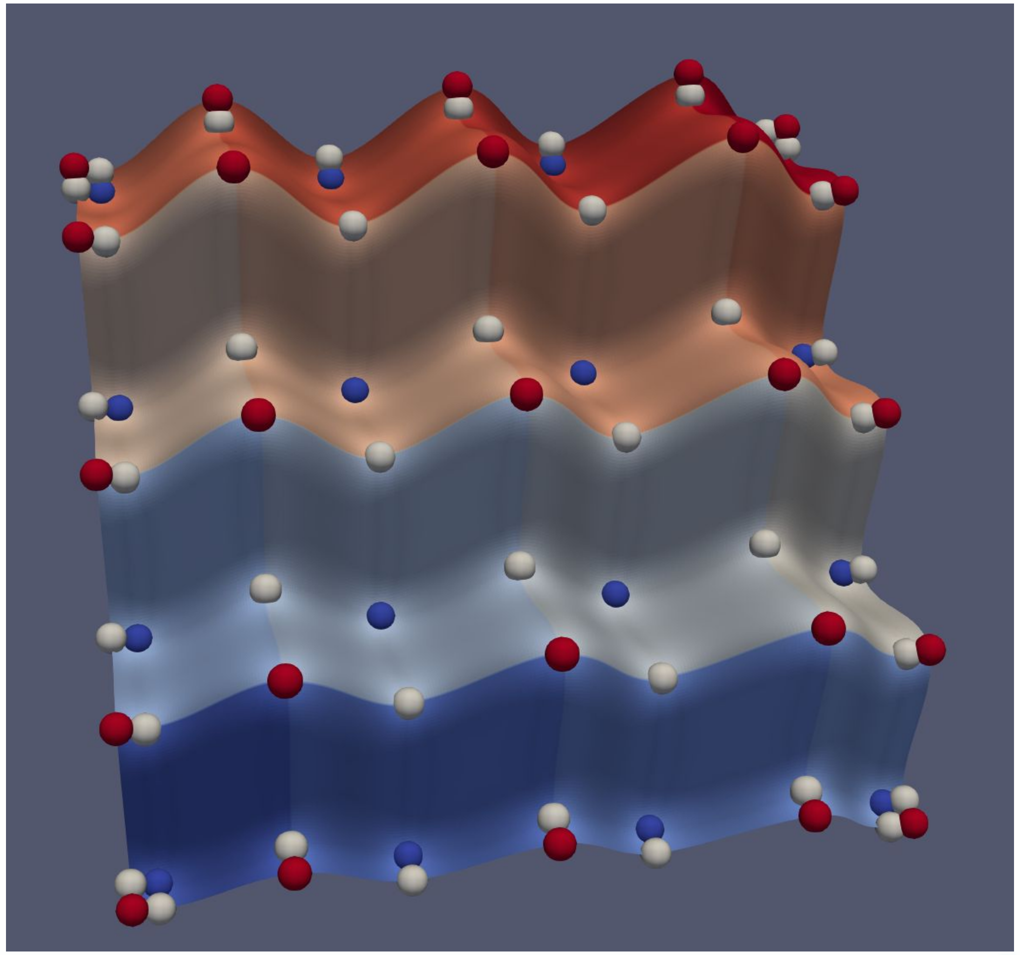
Students found it useful to compare against TTK's built in filter for this same task

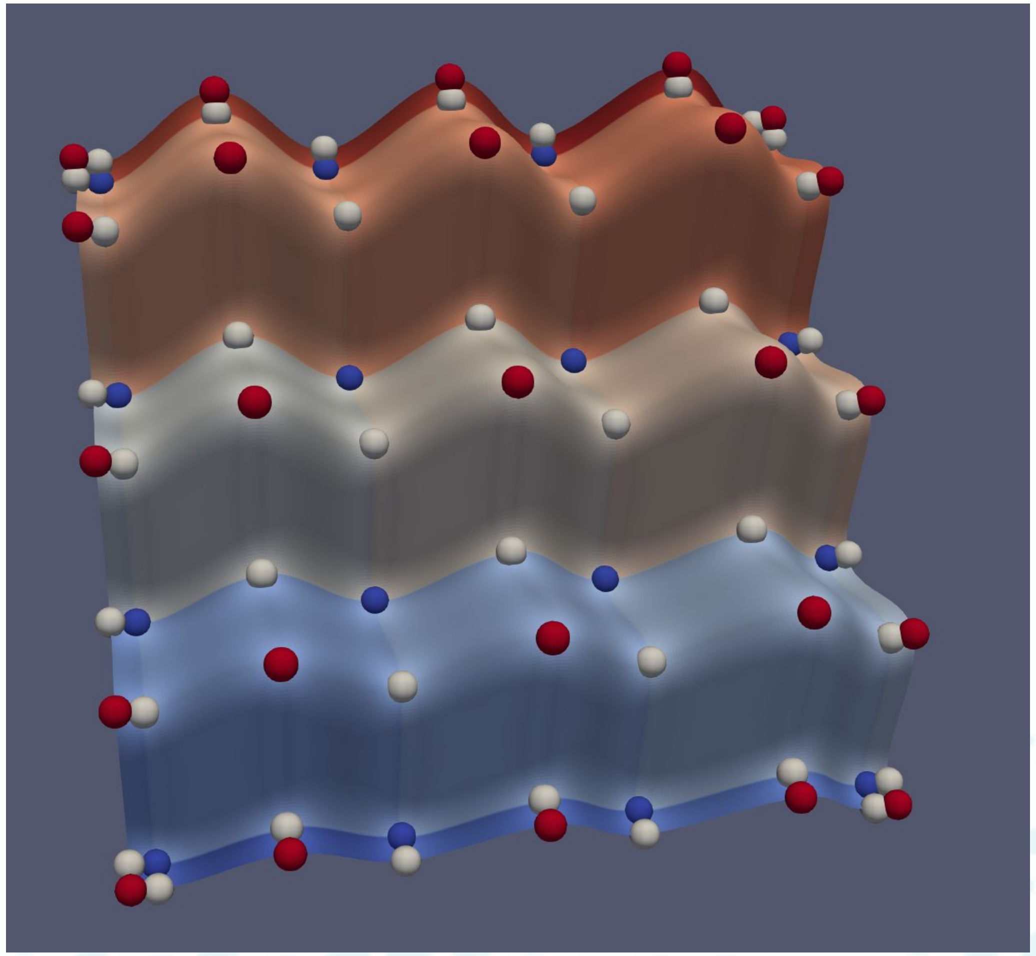
Also saw the complexities and nuances on degenerate cases



Ascending and Descending Manifolds

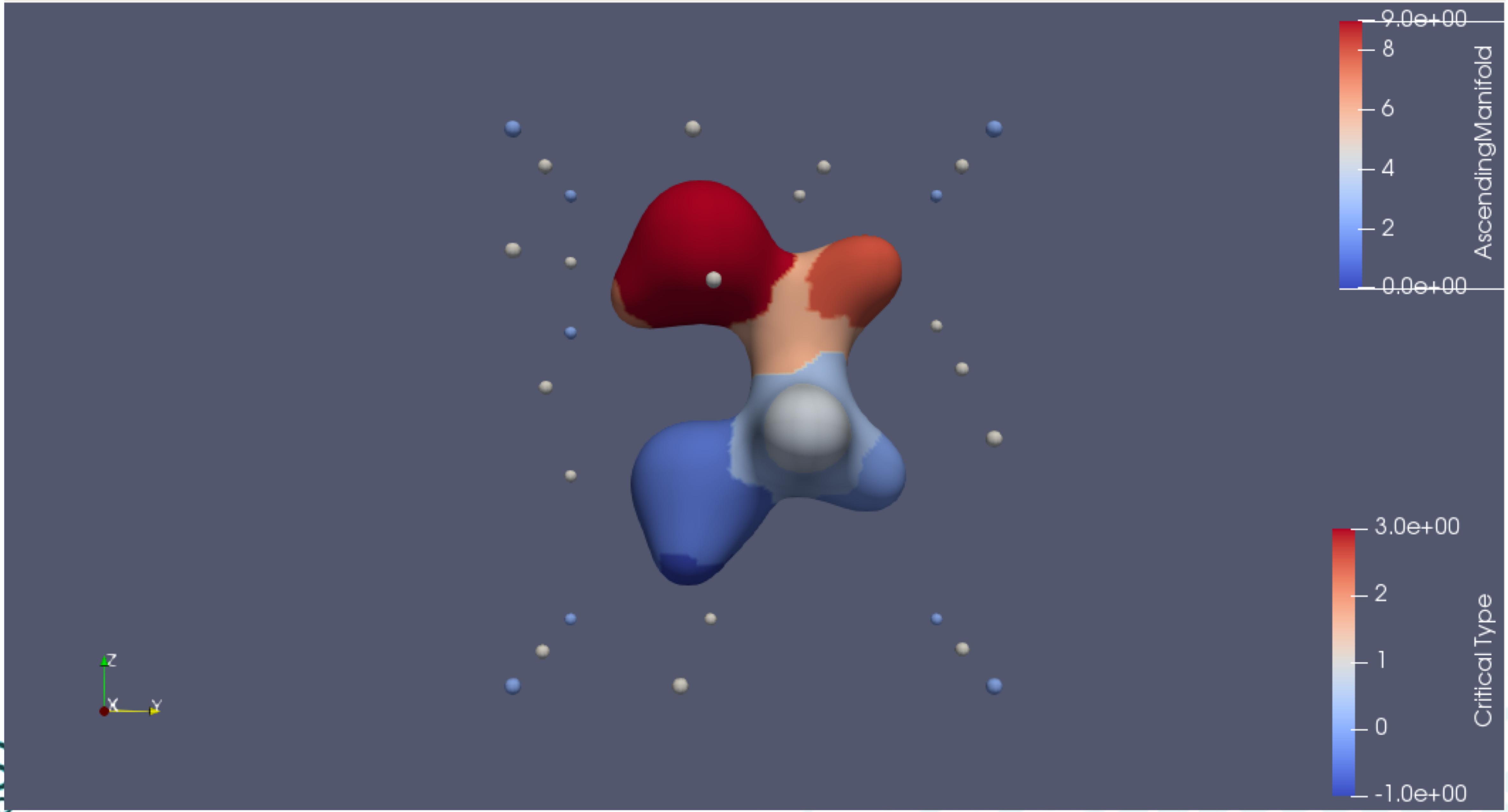
This module expected as input a set of of critical points from
TTKScalarFieldCriticalPoints





VIS 2018

Combining Morse Complexes with Isosurfacing



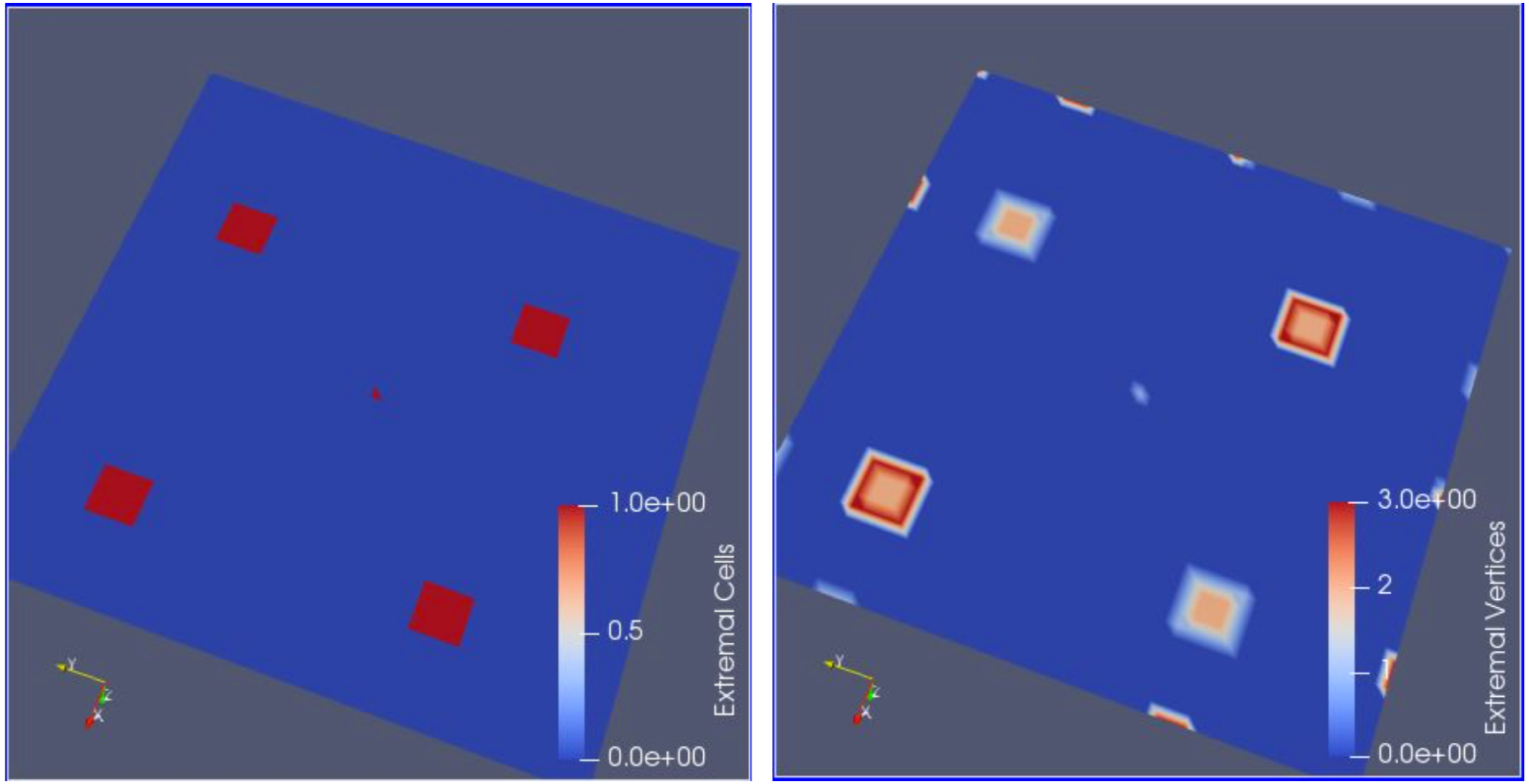
VIS 2018

Final Projects in TTK

Example student projects:

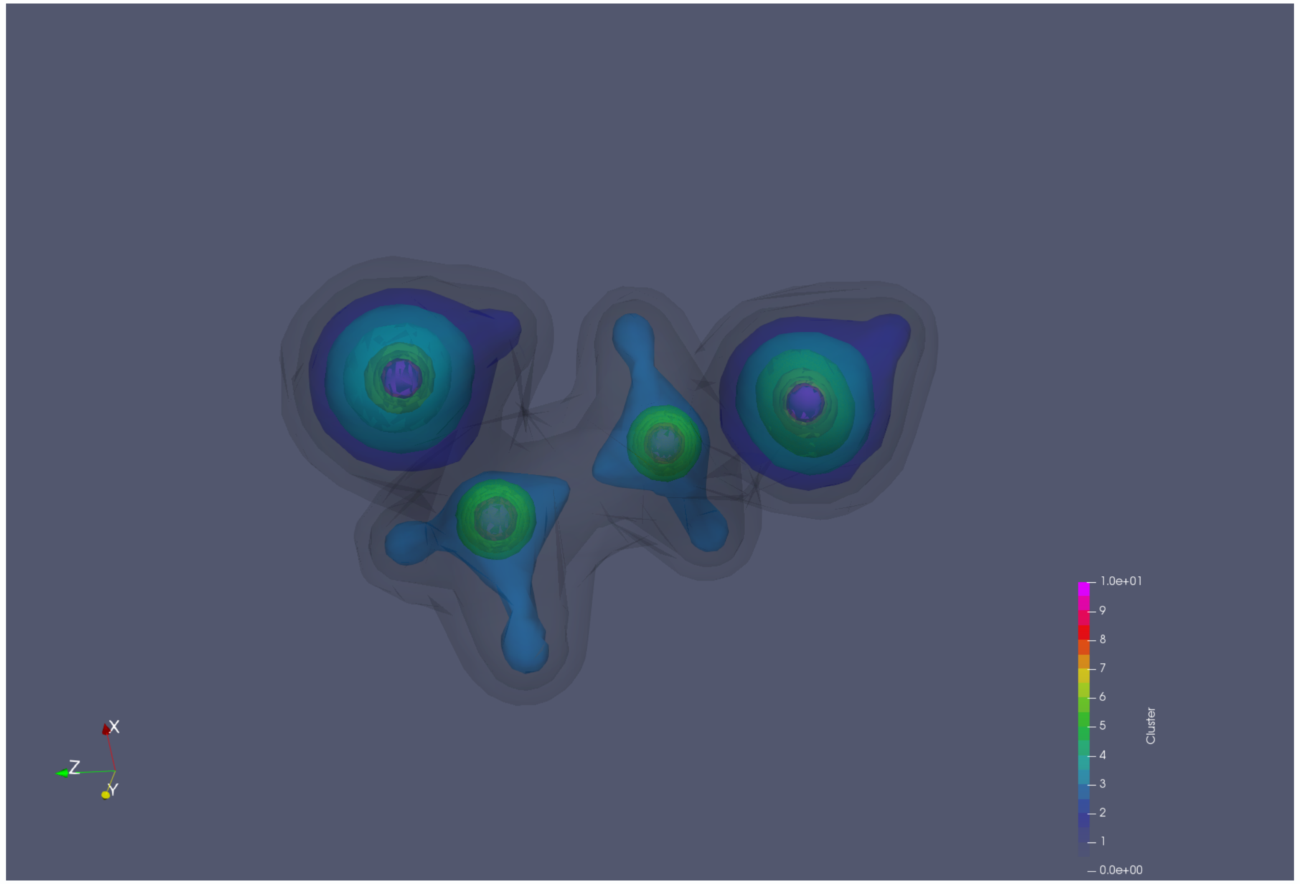
- Fast Extraction of Čech / Rips Complexes
- Generating Loops
- Variants of algorithms Reeb Graphs / Contour Trees
- Shape Matching based on Topological Features
- Pareto Extrema Extraction

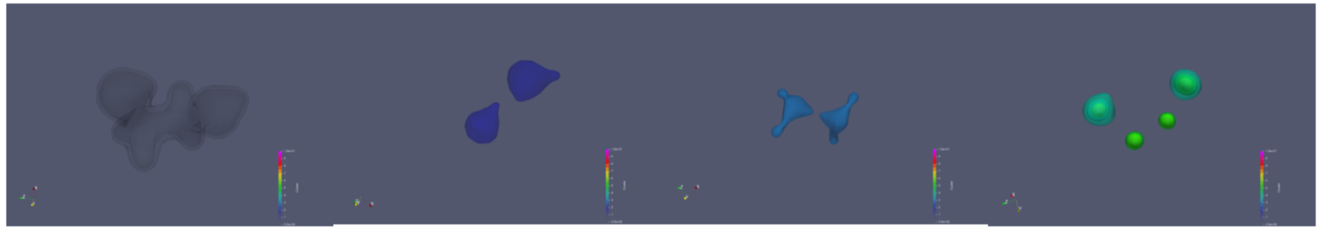
Pareto Extrema



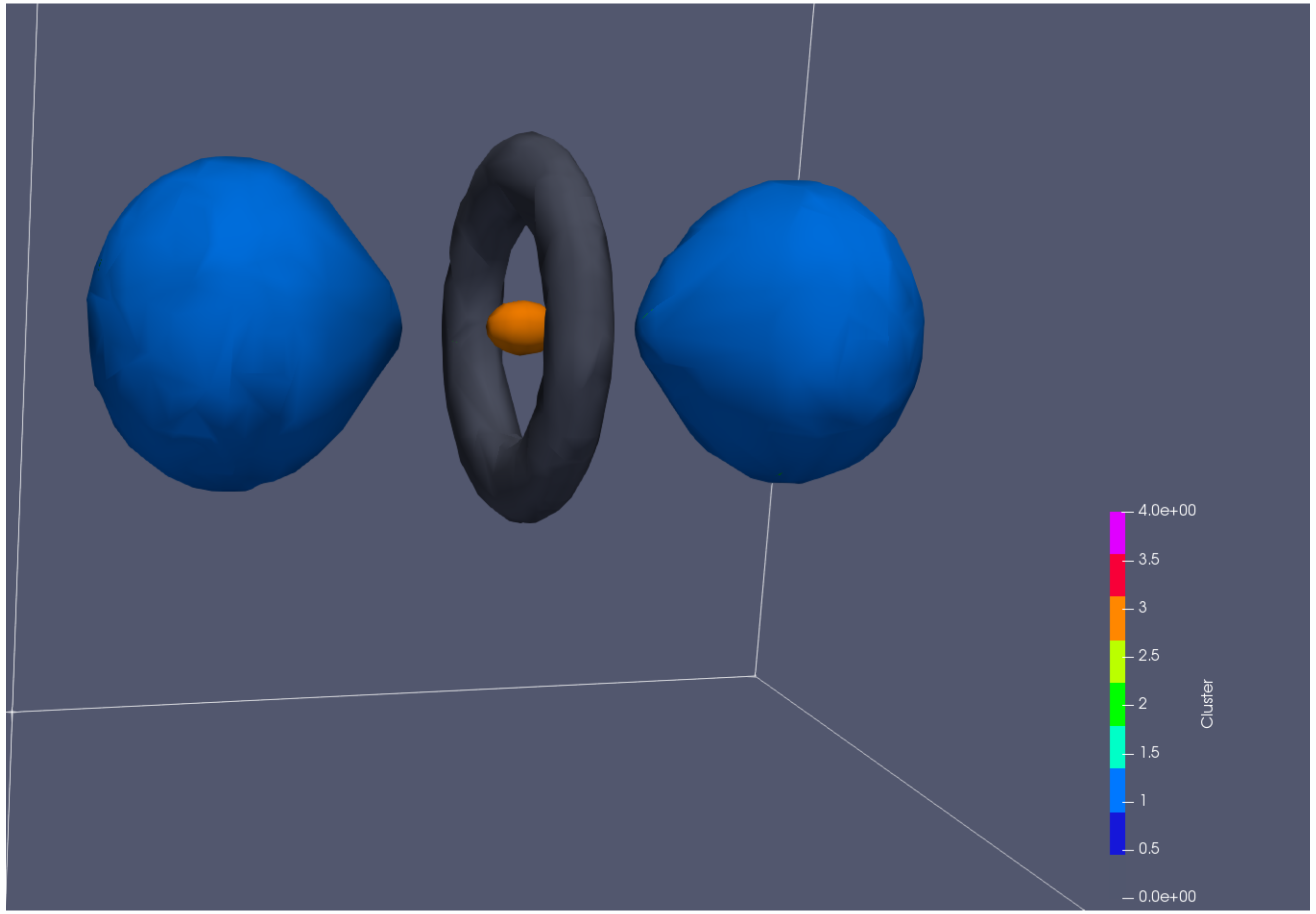
Extremal Vertices and Cells of four fields, based on *Huettenberger et al., Towards Multifield Scalar Topology Based on Pareto Optimality, CGF 2013.*

Shape Matching / Symmetry





Cluster 0 Cluster 2 Cluster 3 Cluster 4-10



Clustering contours inspired by *Thomas and Natarajan, Multiscale symmetry detection in scalar fields by clustering contours, IEEE TVCG 2014.*

Feedback and Discussion

Student Feedback (Positives)

What did you especially like about this course?

“I found the programming assignments to be interesting and useful for learning the material.”

“The projects”

“Topology” =)

Student Feedback (Suggestions)

What suggestions would you make to improve this course?

"I would suggest paired programming for one of the coding assignments. I would have benefitted from being matched with a more experience student - I think I would have made fewer time-consuming mistakes, would have been much less frustrated, and would have better learned how to use C and TTK."

"More projects"

VIS 2018

TTK as a Teaching Platform

Joshua A. Levine, University of Arizona

THE UNIVERSITY OF ARIZONA

josh@email.arizona.edu
http://www.cs.arizona.edu/~josh