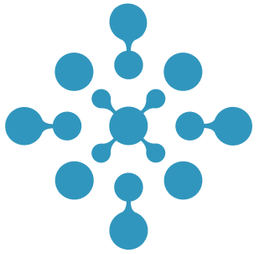


VIS 2019

TTK Integration into Inviwo

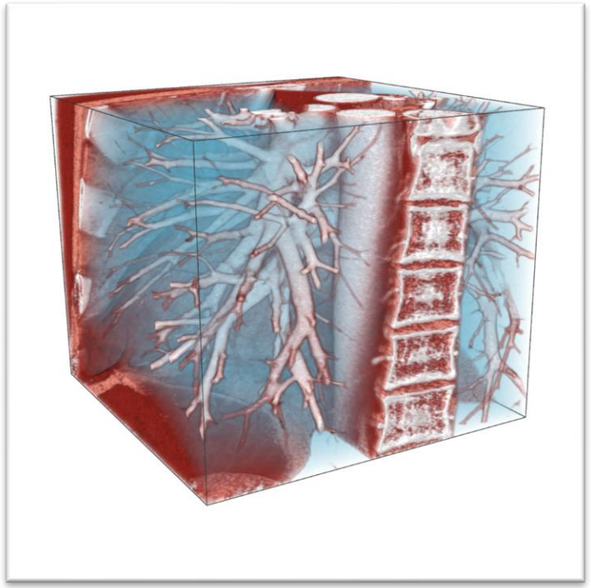
Martin Falk, Linköping University

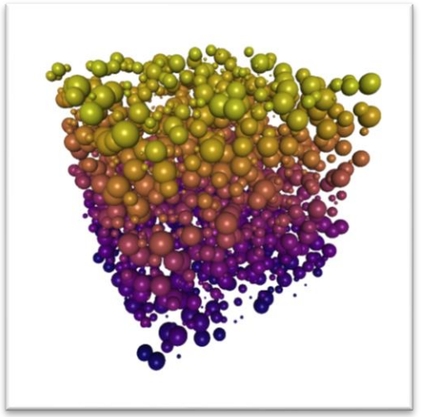
LINKÖPING UNIVERSITY

SeRC
Swedish e-Science Research Centre

Inviwo

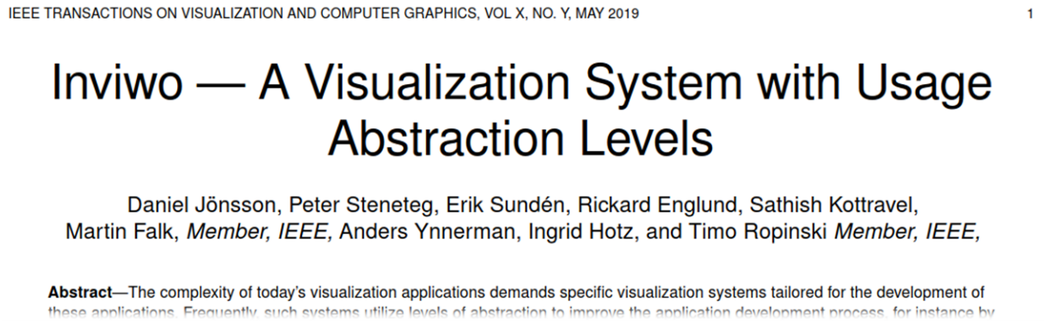
What is Inviwo?





Overview

- Research software
- Developed by visualization groups at
  - LiU, KTH, UULM
- Liberal license (Simplified BSD)
  - Commercial use permitted



IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS, VOL X, NO. Y, MAY 2019 1

Inviwo — A Visualization System with Usage Abstraction Levels

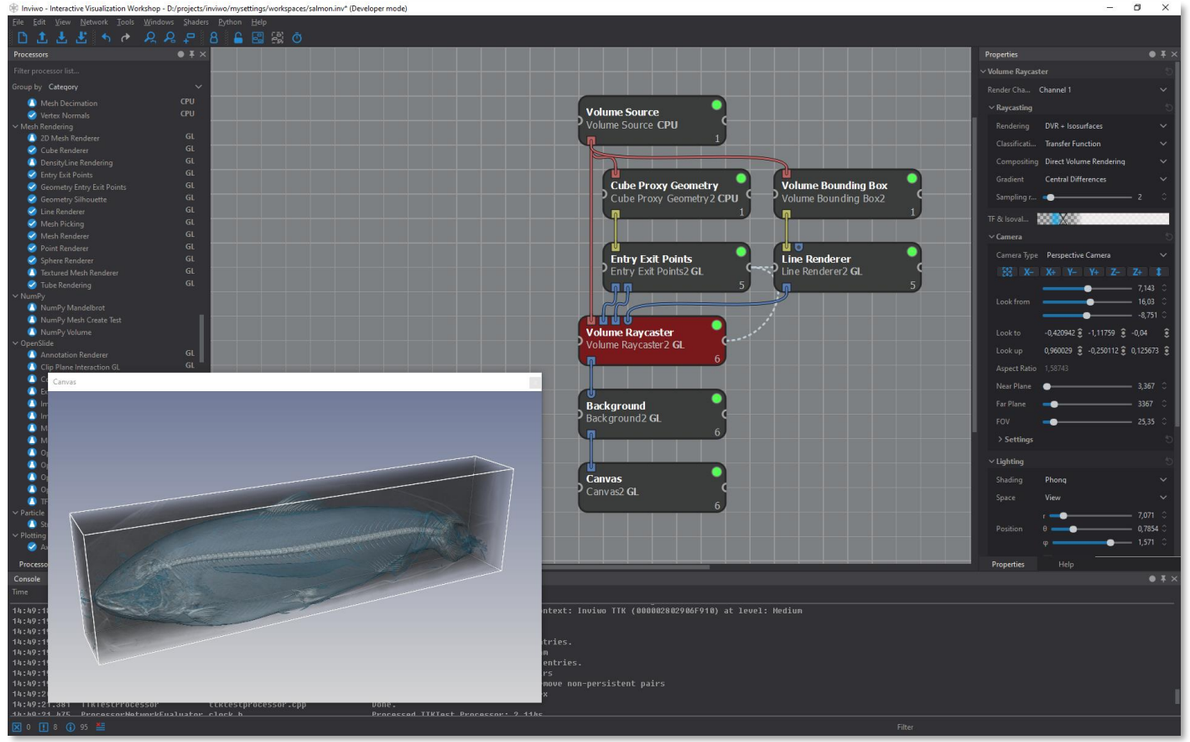
Daniel Jönsson, Peter Steneteg, Erik Sundén, Rickard Englund, Sathish Kottravel, Martin Falk, *Member, IEEE,* Anders Ynnerman, Ingrid Hotz, and Timo Ropinski *Member, IEEE,*

Abstract—The complexity of today's visualization applications demands specific visualization systems tailored for the development of these applications. Frequently, such systems utilize levels of abstraction to improve the application development process, for instance by

D. Jönsson, *et al.*, "Inviwo – A Visualization System with Usage Abstraction Levels" in *IEEE Transactions on Visualization & Computer Graphics*, 2019. doi: 10.1109/TVCG.2019.2920639

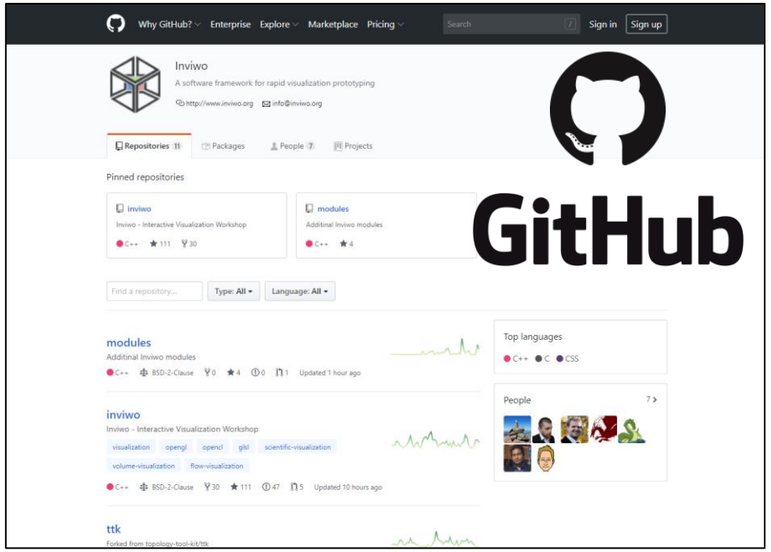
Key Features

- Framework for rapid prototyping
  - C++, Python, JavaScript/HTML
- Visualization pipeline
  - Models data flow
- Transparent data handling
  - RAM ↔ OpenGL ↔ OpenCL ↔ …
- Linked views, Brushing & Linking
- Application building



Getting started

- Recommended way: github.com/inviwo
- Just want to try it out (binaries): inviwo.org
- TTK support available via: github.com/inviwo/modules

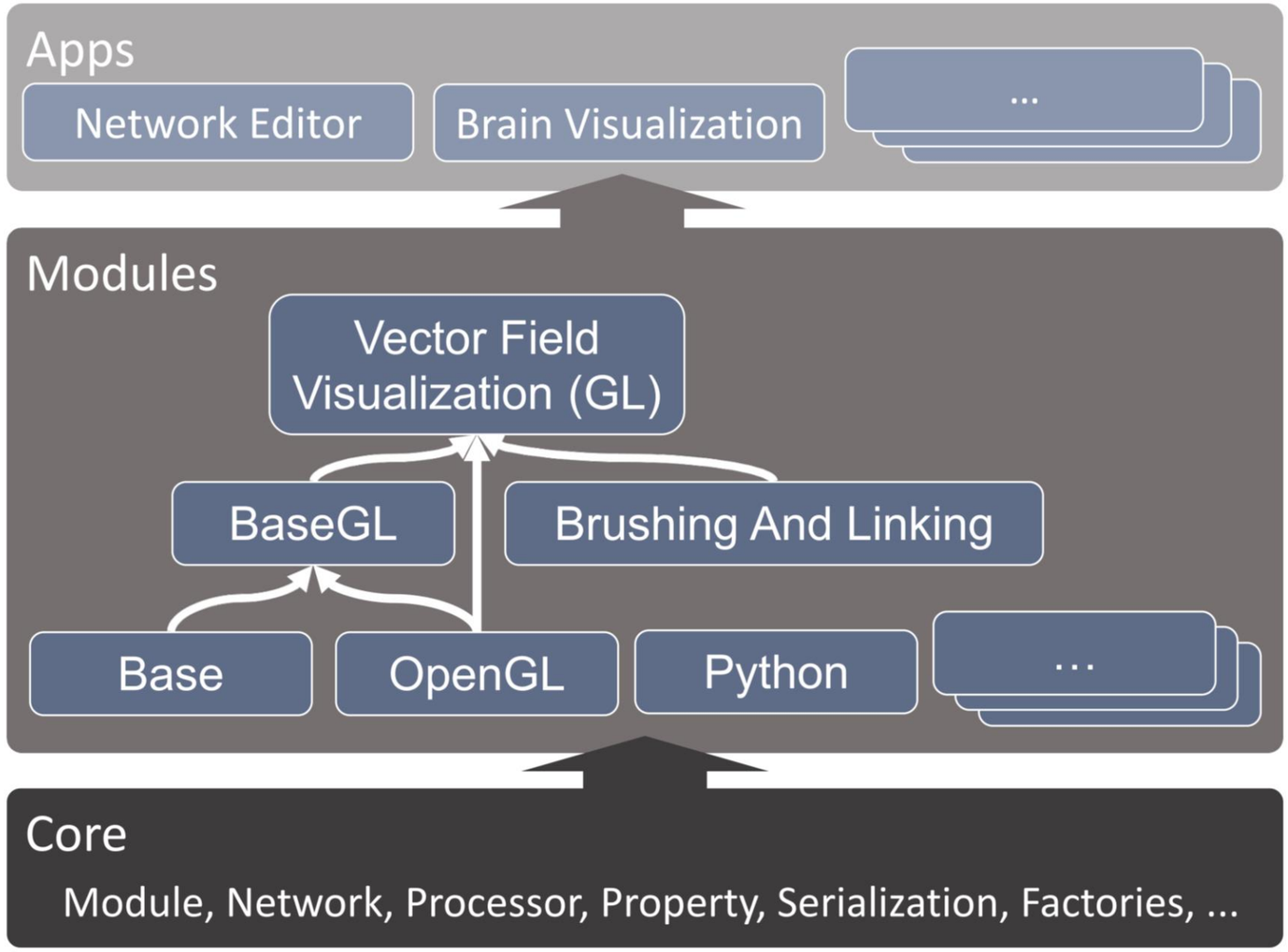


github.com/inviwo

inviwo.org

Architecture

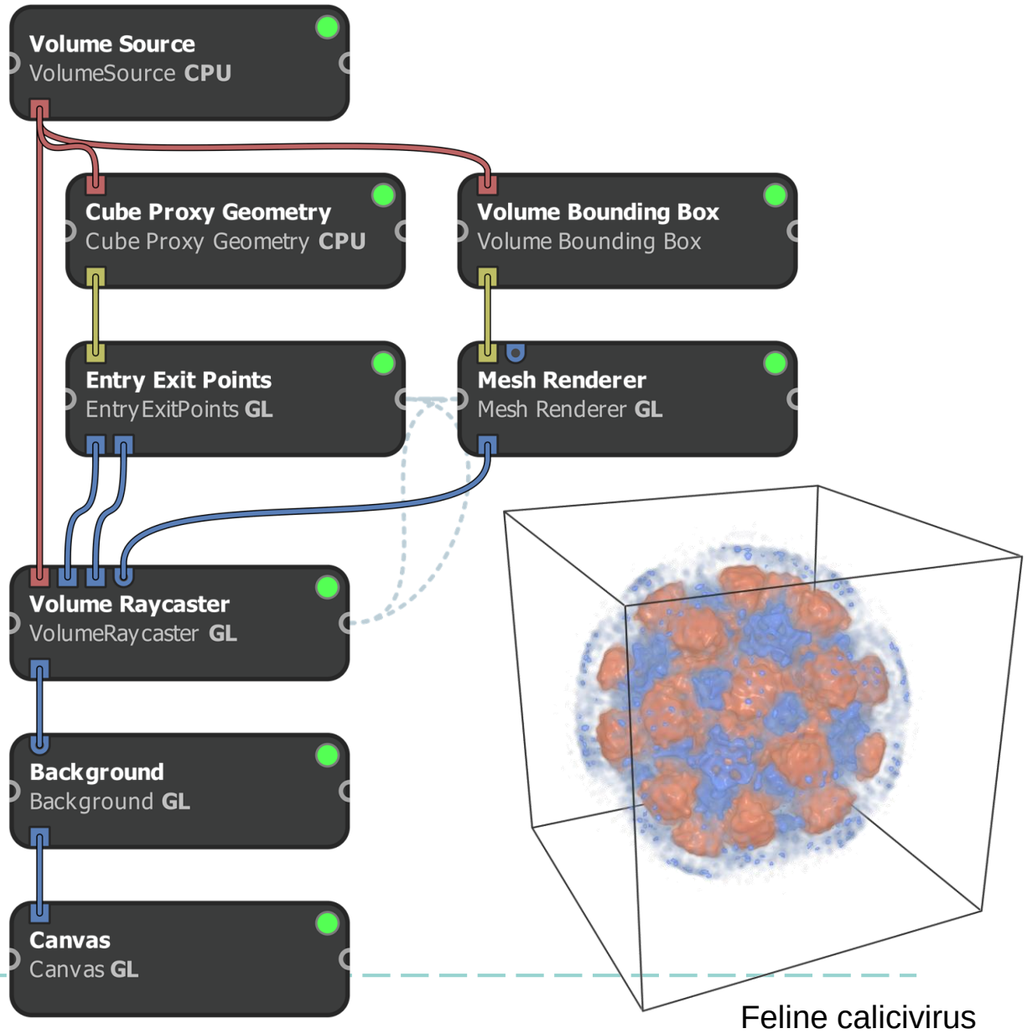
- Pure C++ Core
- Modular system (plugins)
  - Can wrap external libs
- Multiple Apps
  - Qt network editor
  - Python interface
  - …



Visualization pipeline – processor network

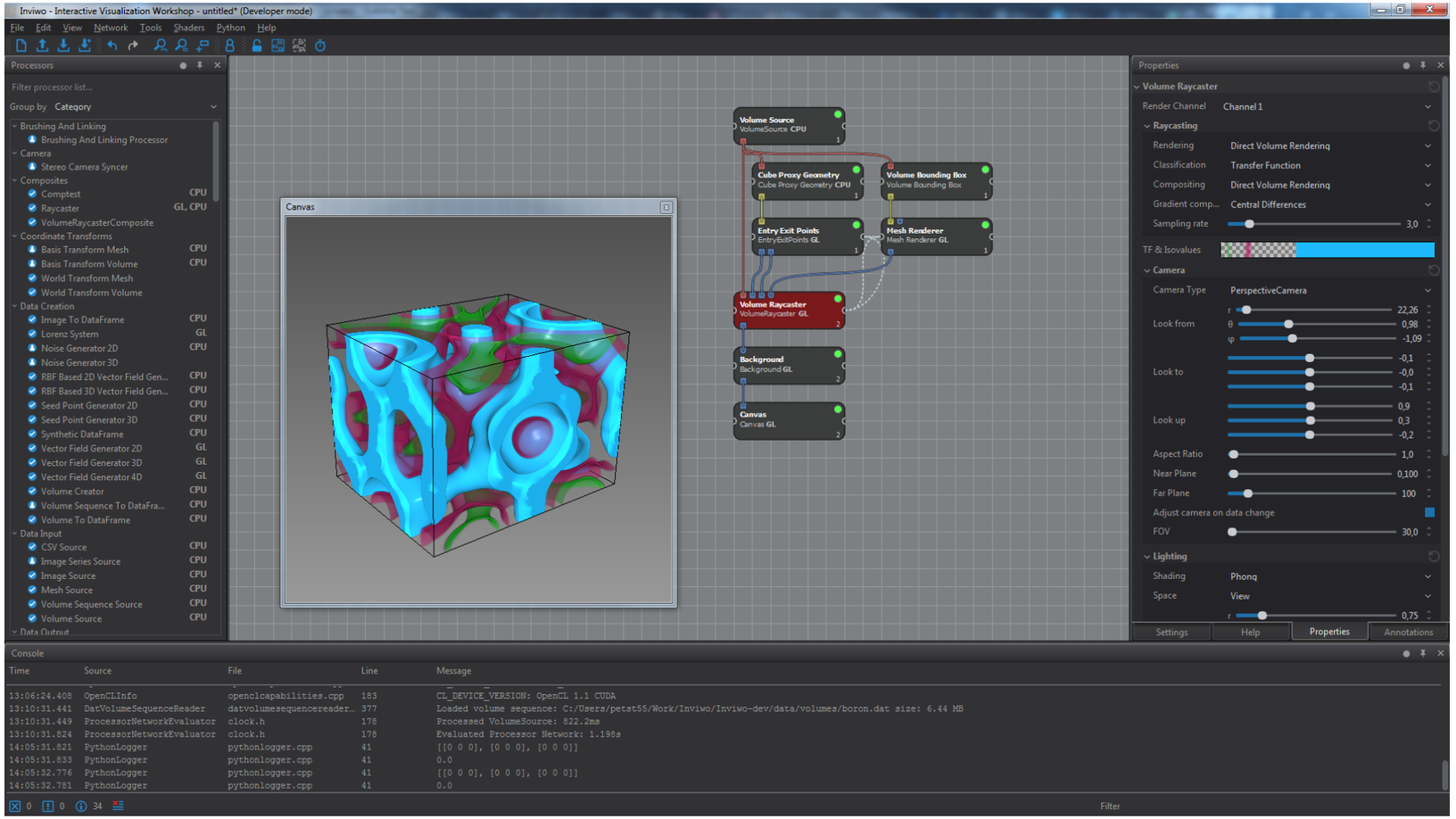
- Models a data-flow graph
- Data enters at the top
- Result at the bottom
- “Stateless”

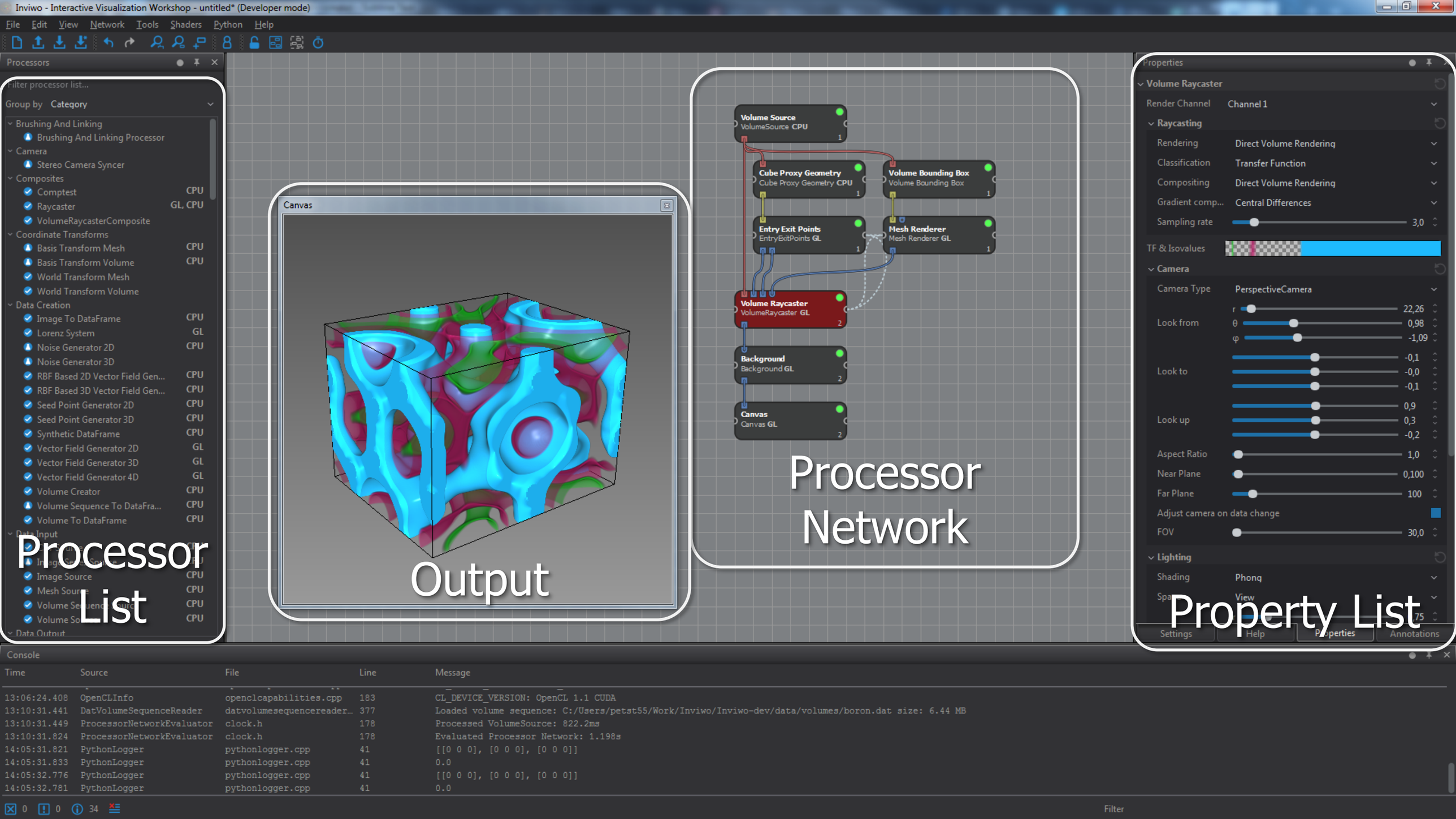
$$f_{p4}\left(g_{p3}\left(h_{p1}(input_1), k_{p2}(input_2)\right)\right)$$



Feline calicivirus

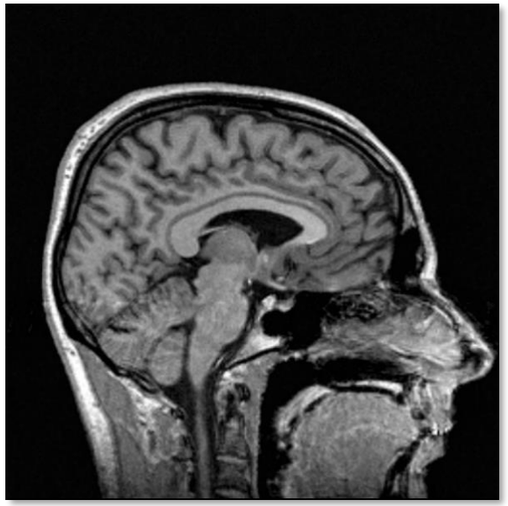
Network editor – Create/edit visualization pipelines

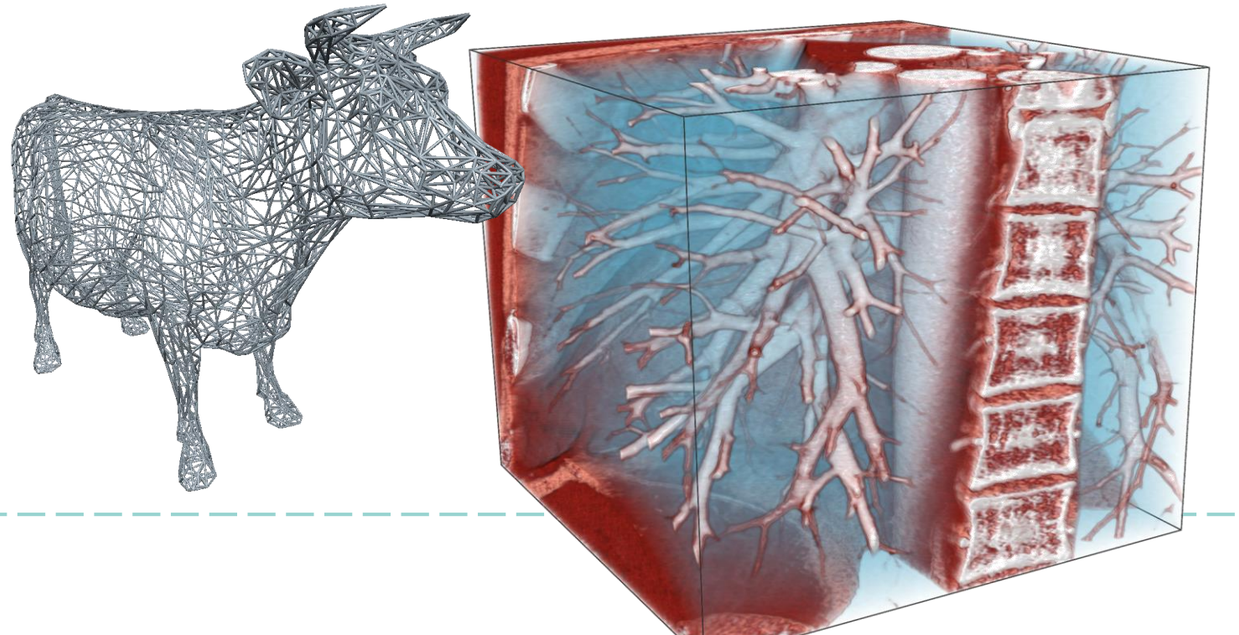



Inviwo - Interactive Visualization Workshop - untitled* (Developer mode)
File Edit View Network Tools Shaders Python Help
Processors
Filter processor list...
Group by Category
Brushing And Linking
Brushing And Linking Processor
Camera
Stereo Camera Syncer
Composites
Comptest CPU
Raycaster GL, CPU
VolumeRaycasterComposite
Coordinate Transforms
Basis Transform Mesh CPU
Basis Transform Volume CPU
World Transform Mesh
World Transform Volume
Data Creation
Image To DataFrame CPU
Lorenz System GL
Noise Generator 2D CPU
Noise Generator 3D
RBF Based 2D Vector Field Gen... CPU
RBF Based 3D Vector Field Gen... CPU
Seed Point Generator 2D CPU
Seed Point Generator 3D CPU
Synthetic DataFrame CPU
Vector Field Generator 2D GL
Vector Field Generator 3D GL
Vector Field Generator 4D GL
Volume Creator CPU
Volume Sequence To DataFra... CPU
Volume To DataFrame CPU
Image Source CPU
Mesh Source CPU
Volume Sequence Source CPU
Volume Source CPU
Data Output
Processor List
Canvas
Output
Volume Source
VolumeSource CPU
Cube Proxy Geometry
Cube Proxy Geometry CPU
Volume Bounding Box
Volume Bounding Box
Entry Exit Points
EntryExitPoints GL
Mesh Renderer
Mesh Renderer GL
Volume Raycaster
VolumeRaycaster GL
Background
Background GL
Canvas
Canvas GL
Processor Network
Properties
Volume Raycaster
Render Channel Channel 1
Raycasting
Rendering Direct Volume Rendering
Classification Transfer Function
Compositing Direct Volume Rendering
Gradient comp... Central Differences
Sampling rate 3,0
TF & Isovalues
Camera
Camera Type PerspectiveCamera
Look from r 22,26 θ 0,98 φ -1,09
Look to -0,1 -0,0 -0,1
Look up 0,9 0,3 -0,2
Aspect Ratio 1,0
Near Plane 0,100
Far Plane 100
Adjust camera on data change
FOV 30,0
Lighting
Shading Phong
Settings Help Properties Annotations
Property List
Console
Time Source File Line Message
13:06:24.408 OpenCLInfo openclcapabilities.cpp 183 CL_DEVICE_VERSION: OpenCL 1.1 CUDA
13:10:31.441 DatVolumeSequenceReader datvolumesequencereader... 377 Loaded volume sequence: C:/Users/petst55/Work/Inviwo/Inviwo-dev/data/volumes/boron.dat size: 6.44 MB
13:10:31.449 ProcessorNetworkEvaluator clock.h 178 Processed VolumeSource: 822.2ms
13:10:31.824 ProcessorNetworkEvaluator clock.h 178 Evaluated Processor Network: 1.198s
14:05:31.821 PythonLogger pythonlogger.cpp 41 [[0 0 0], [0 0 0], [0 0 0]]
14:05:31.833 PythonLogger pythonlogger.cpp 41 0.0
14:05:32.776 PythonLogger pythonlogger.cpp 41 [[0 0 0], [0 0 0], [0 0 0]]
14:05:32.781 PythonLogger pythonlogger.cpp 41 0.0
0 0 34
Filter

Supported data formats

- Imaging data
  - DICOM, H5, TIFF stack, RAW, nifty, pvm, …
- Mesh data via Assimp
  - Collada, 3Ds, fbx, obj, …
- Various image formats
  - Jpeg, TIFF, PNG, …
- VTK support in progress

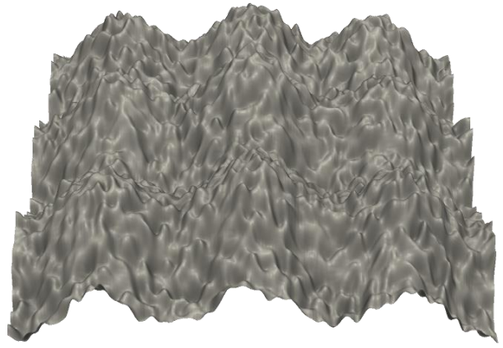




TTK integration

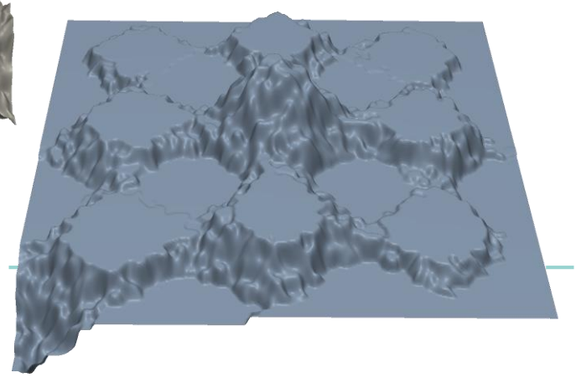
TTK C++ example

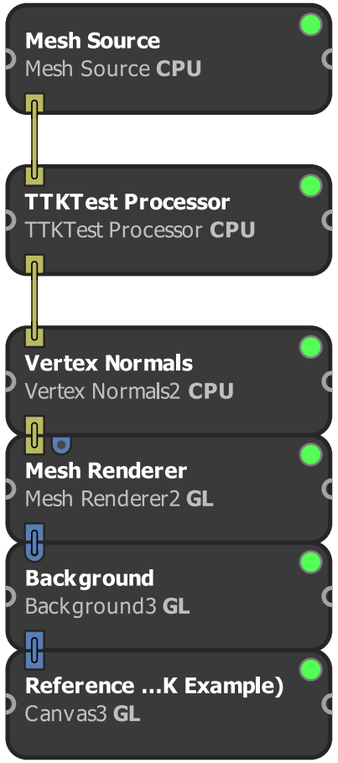
- One processor contains entire example
  - Persistence diagram
  - Simplification
  - …



Input geometry

Output






```
void TTKTestProcessor::process() {
    // convert input mesh to TTK triangle data
    auto ttkData = topology::meshToTTKTriangulation(*inport

    // NOW, do the TTK processing
    LogInfo("1. computing some elevation");
    std::vector<float> height(ttkData.getPoints().size());
    std::vector<int> offsets(height.size());
    // use the z-coordinate here
    for (size_t i = 0; i < ttkData.getPoints().size(); ++i)
        height[i] = ttkData[i].z;
        offsets[i] = static_cast<int>(i);
    }

    LogInfo("2. computing the persistence curve");
    ttk::PersistenceCurve curve;
    std::vector<std::pair<float, ttk::SimplexId> > outputCu
    curve.setupTriangulation(&ttkData.getTriangulation());
    curve.setInputScalars(height.data());
    curve.setInputOffsets(offsets.data());
    curve.setOutputCTPlot(&outputCurve);
    curve.execute<float, int>();
    LogInfo("  ttk::PersistenceCurve has " << outputCurve.s

    LogInfo("3. computing the persistence diagram");
    ttk::PersistenceDiagram diagram;
    std::vector<std::tuple<ttk::SimplexId, ttk::CriticalTyp
                           float, ttk::SimplexId> > diagram
    diagram.setupTriangulation(&ttkData.getTriangulation())
    diagram.setInputScalars(height.data());
    diagram.setInputOffsets(offsets.data());
    diagram.setOutputCTDiagram(&diagramOutput);
    diagram.execute<float, int>();
    ...

    outport_.setData(topology::ttkTriangulationToMesh(ttkDa
```

Interfacing Inviwo with TTK

- Convert from/to TTK
  - Mesh, volumes, …

```
TriangulationData meshToTTKTriangulation(const Mesh& mesh);
std::shared_ptr<Mesh> ttkTriangulationToMesh(const TriangulationData& data,
                                             const vec4& color = vec4(1.0f),
                                             bool applyScalars = false, size_t component = 0);

TriangulationData volumeToTTKTriangulation(const Volume& volume, size_t channel);
std::shared_ptr<Volume> ttkTriangulationToVolume(const TriangulationData& data);
```

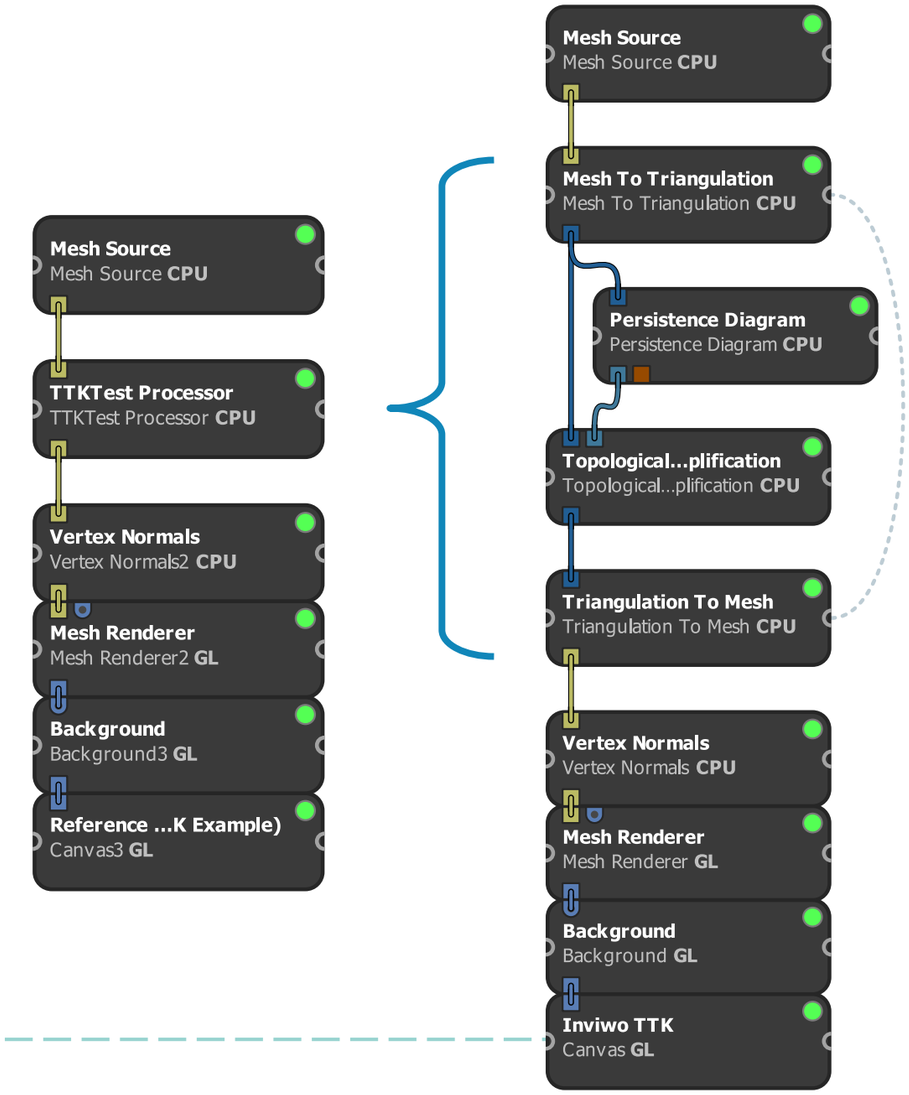
- Encapsulate TTK data for subsequent processors

```
class IVW_MODULE_TOPOLOGYTOOLKIT_API TriangulationData : public SpatialEntity<3> {
public:
    TriangulationData(const size3_t& dims, const vec3& origin, const vec3& extent, ...);
    TriangulationData(std::vector<vec3> points, ...);

private:
    ttk::Triangulation triangulation_;
};
```

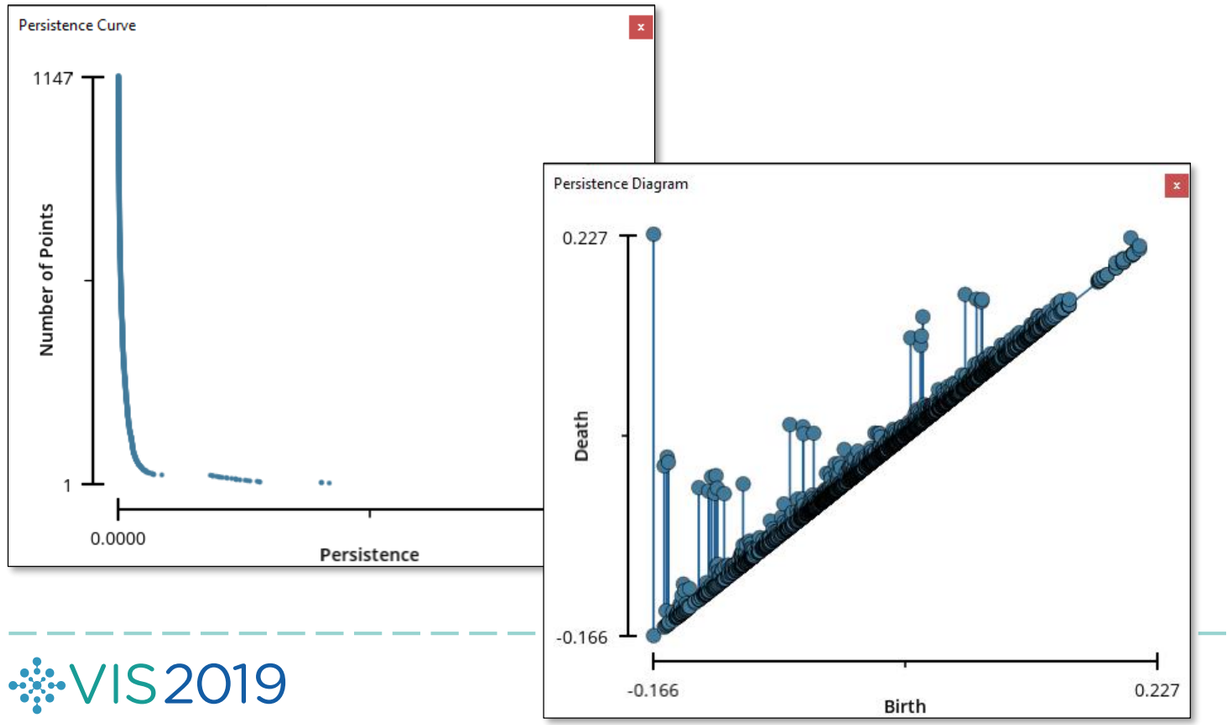
Interfacing Inviwo with TTK

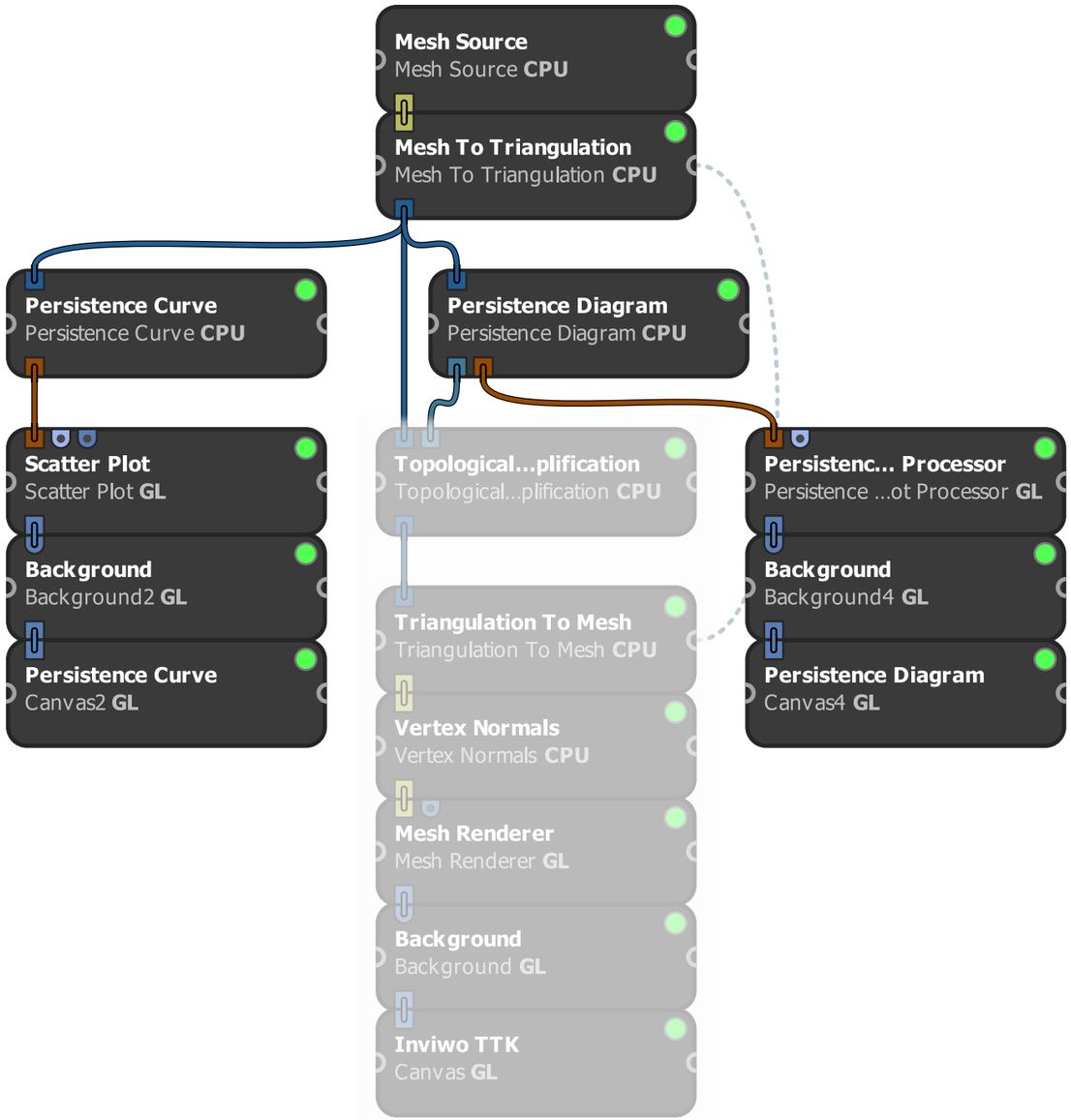
- Separating functionality into processors
  - Topological simplification
  - Persistence diagram and curve
  - Contour tree
  - Morse-Smale complex
- Inviwo/TTK interface
  - Volume/Mesh To Triangulation
  - Triangulation To Volume/Mesh
  - Morse-Smale To Mesh
  - …



Plotting

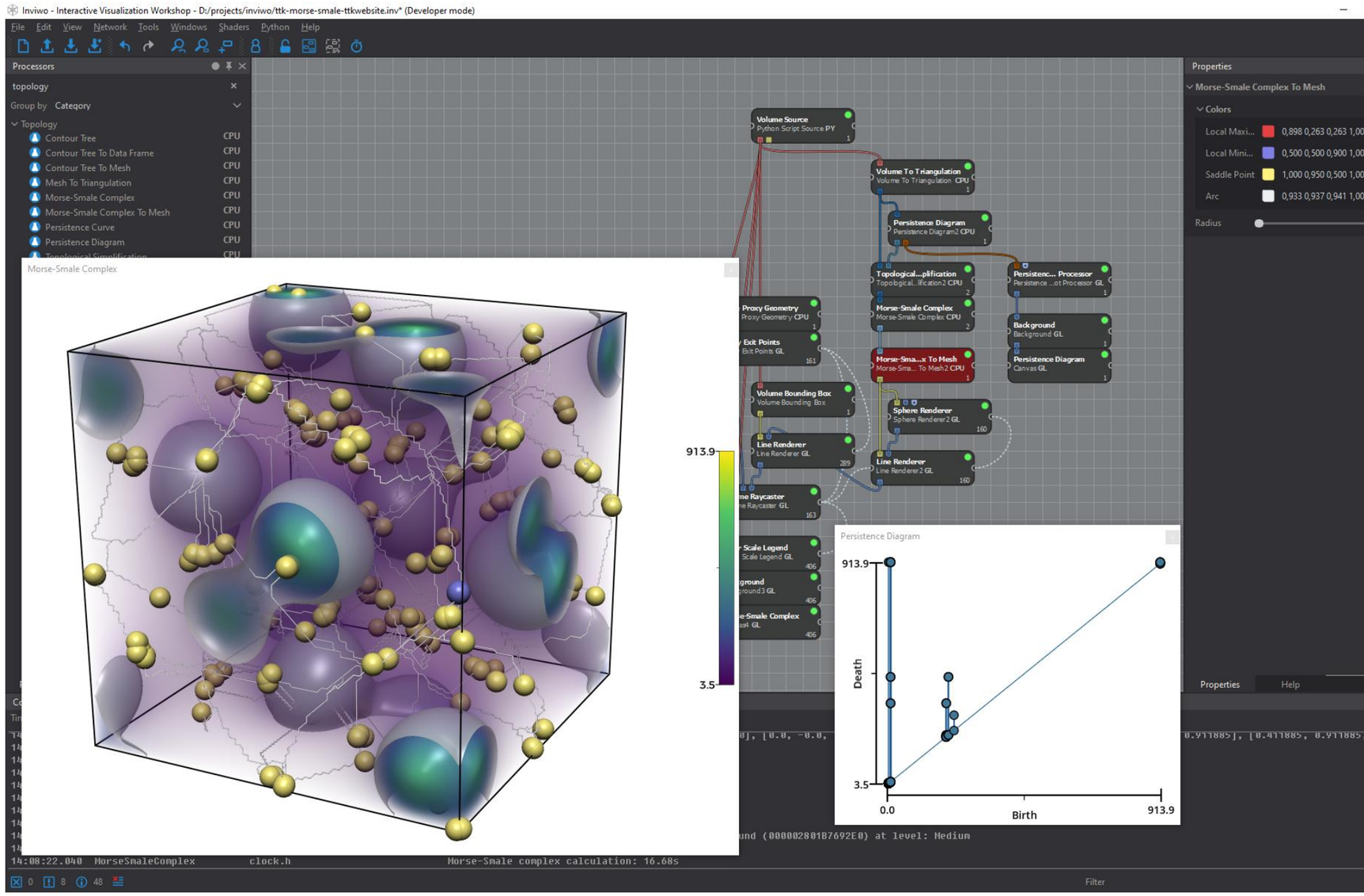
- Conversion to DataFrame
  - Persistence diagram
  - Persistence curve





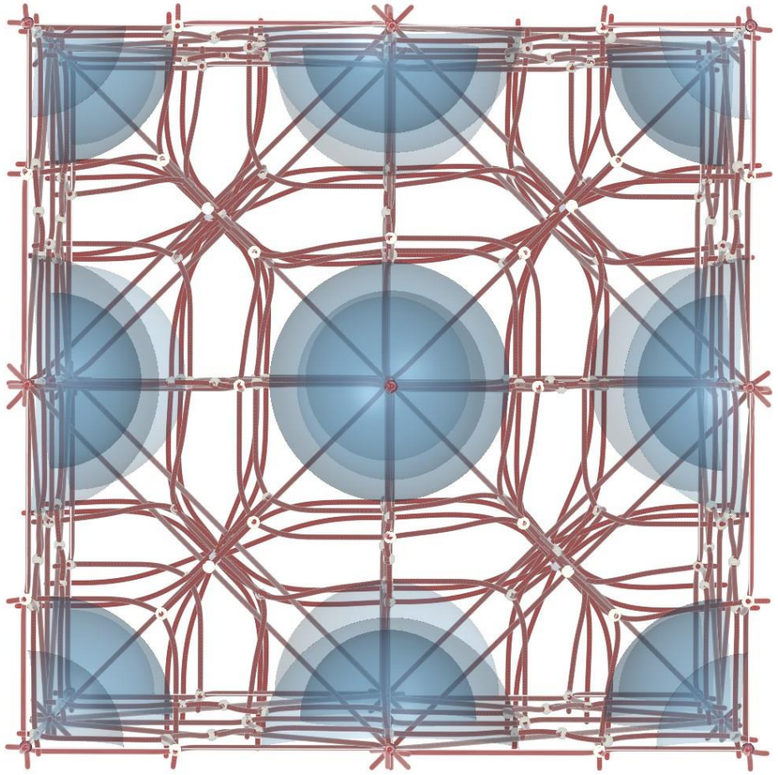
Charge density

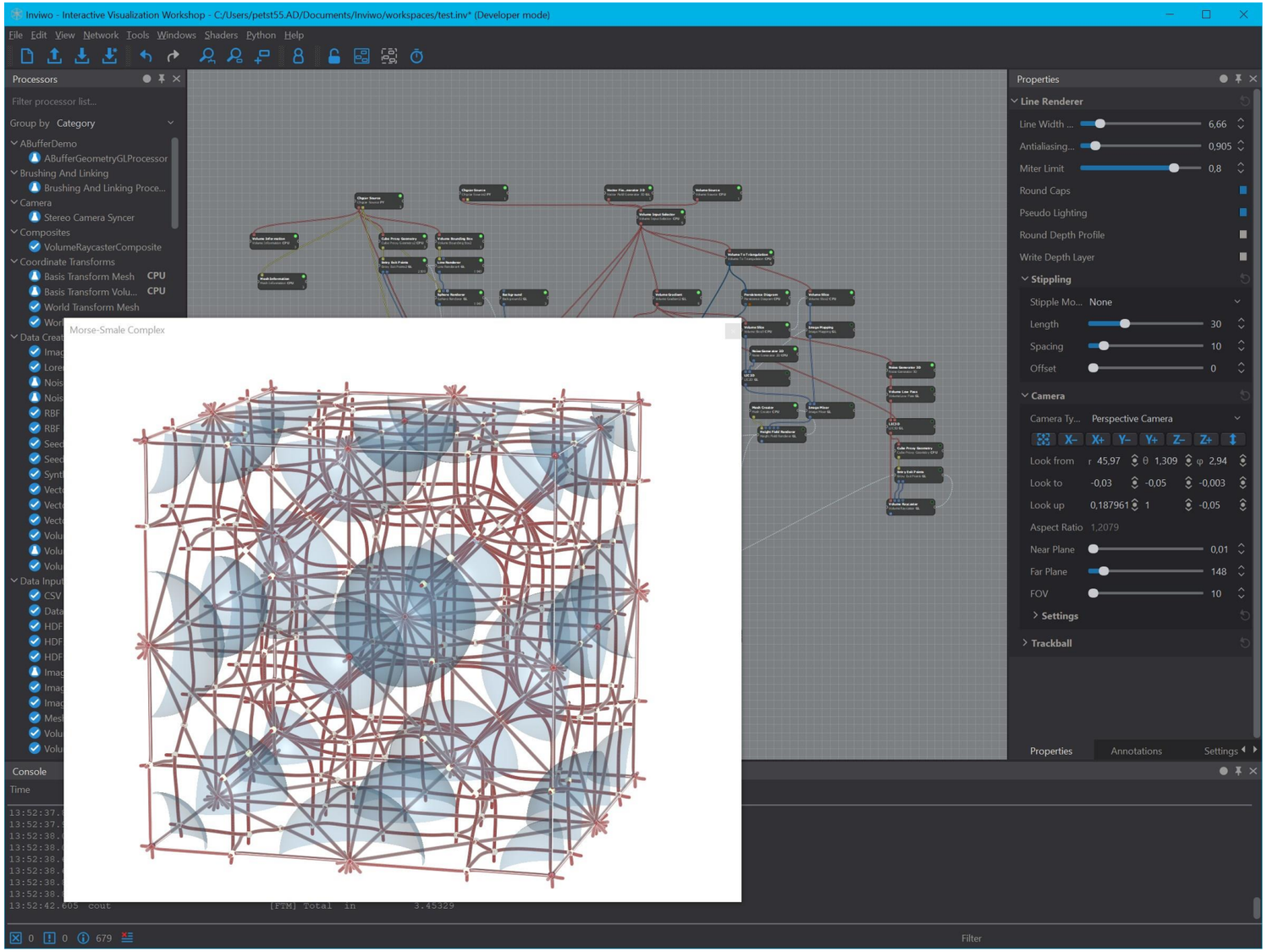
- Iron Oxide



Charge density

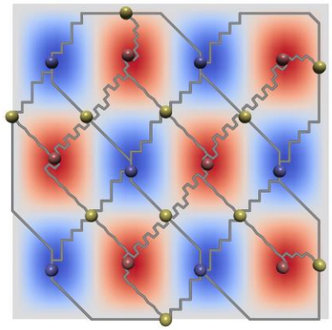
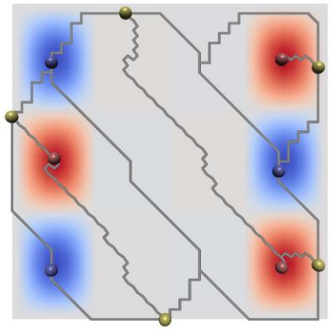
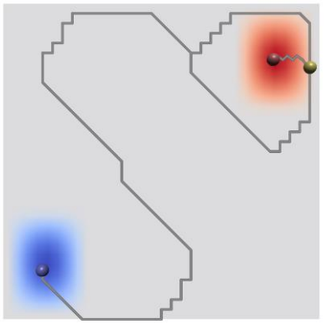
- Sodium chloride





Experiences & Challenges

- TTK is very modular
  - Targeted toward VTK / ParaView
- Interface can be quite extensive
  - Morse-Smale requires 15 vectors (6 for critical points, 3+6 for 1-separatrices)
  - Would be handy to access internal data structures
- TTK Developers are very supportive ☺

Inviwo.org

Need help?

inviwo.slack.com

Presented on Thursday
9:00 – 10:30AM
Ballroom B

SeRC
Swedish e-Science Research Centre

ELLIIT

LINKÖPING UNIVERSITY

KTH VETENSKAP OCH KONST

universität ulm

D. Jönsson, *et al.*, "Inviwo – A Visualization System with Usage Abstraction Levels" in *IEEE Transactions on Visualization & Computer Graphics*, 2019.
doi: 10.1109/TVCG.2019.2920639