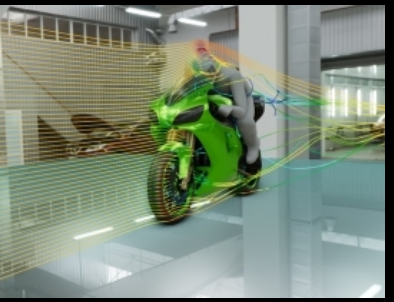
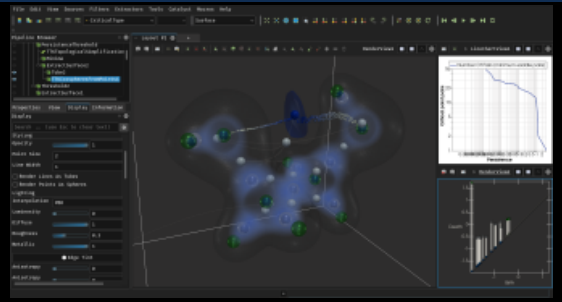
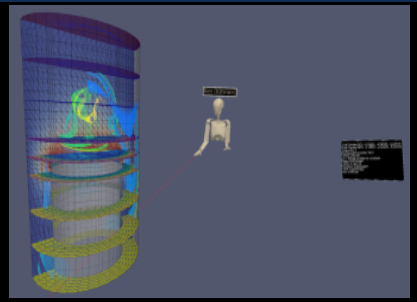
This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2023-02690C

Exceptional service in the national interest

Sandia National Laboratories



What's New in ParaView

DOECGF 2023

April 25, 2023

Mark Bolstad Sandia National Laboratories

with lots of contributions from Cory Quammen and the ParaView Community

Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. SAND xxxx-xxxxx

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Acknowledgements

- This research was supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of two U.S. Department of Energy organizations (Office of Science and the National Nuclear Security Administration) responsible for the planning and preparation of a capable exascale ecosystem, including software, applications, hardware, advanced system engineering, and early testbed platforms, in support of the nation's exascale computing imperative.
- Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.
- **Thanks to many, many partners in labs, universities, and industry.**

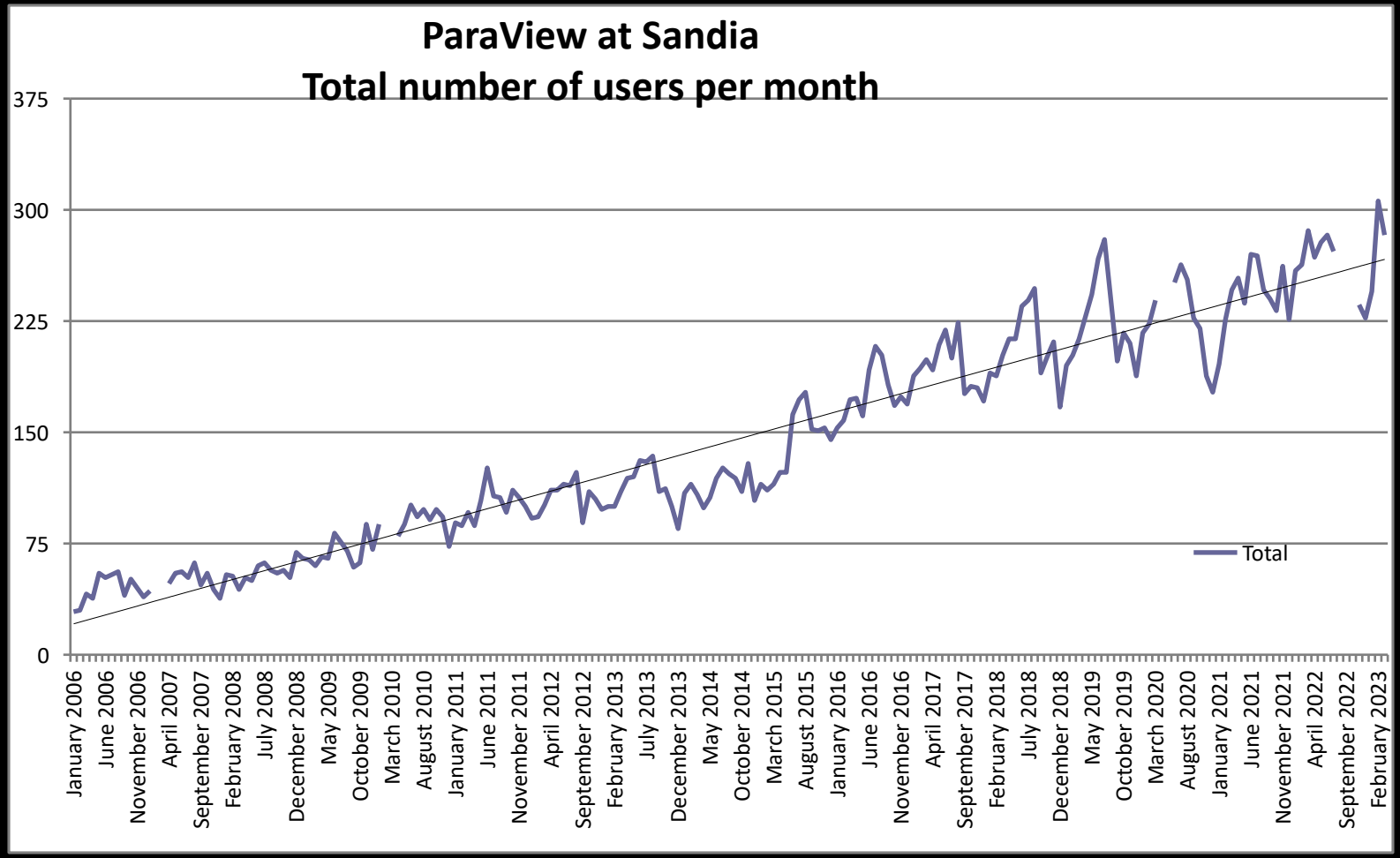
The Numbers

Sandia National Laboratories

- Releases during the last 12+ months: 5.10.1 (March, 2022), 5.11.0 (November 2022), 5.11.1 (March 2023)
- Users at Sandia
  - ~250/month (DART metric: 2+ uses, non-support)
  - ~500 total during 2022 (unclassified use only)
- Downloads from Kitware past year: 302K, 7.6k sources

Sandia Monthly ParaView Usage

Sandia National Laboratories

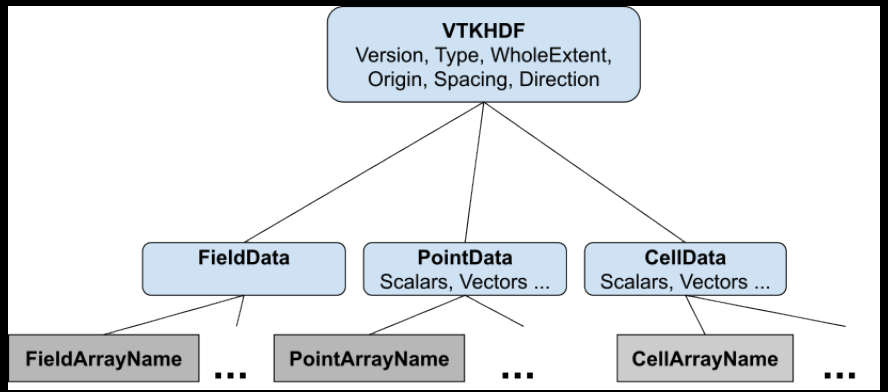


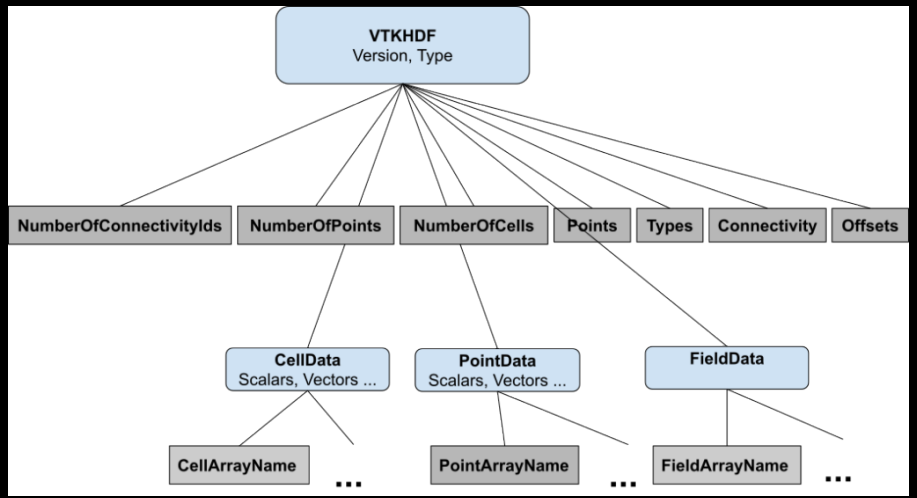
Sandia National Laboratories

Data Management Improvements

VTK HDF format

- Storing VTK data in HDF is complicated because HDF does not have a **standard** way to attach meaning to its arrays
- Traditionally the method was to use XDMF
  - Need to write XML (light-weight data)
  - Not actively maintained
- So, created VTKHDF to address the issue

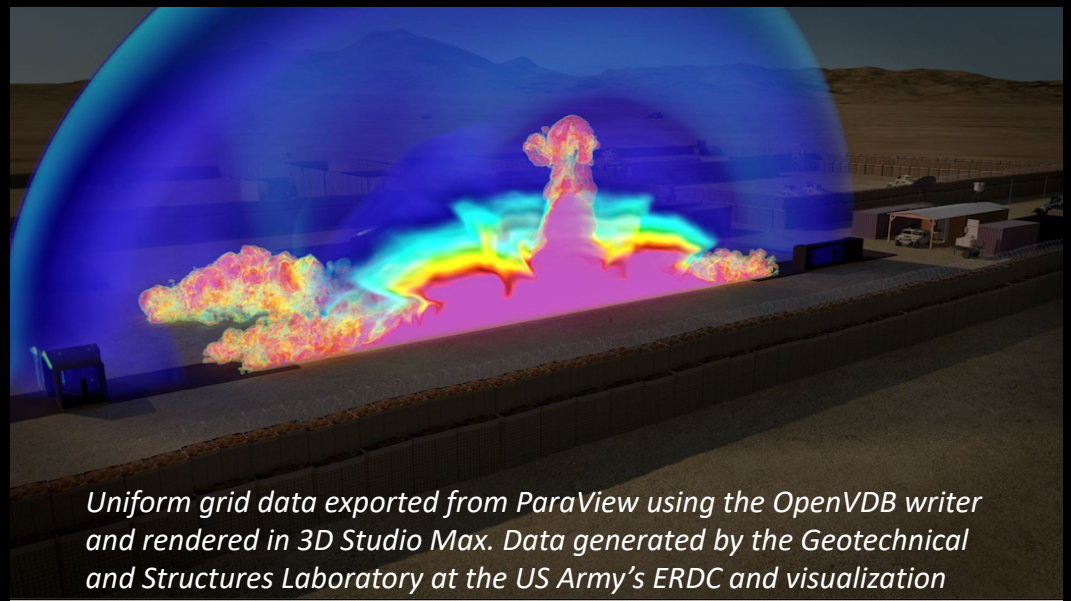




Others

Sandia National Laboratories

- IOSS Improvements
  - Writer for Exodus files (beta)
  - Reader now shows block/set ids in the Blocks/Sets selection widgets shown on the *Properties Panel*.
- OpenVDB Reader (5.11)/Writer (5.10)
  - Allows for interchange between DCC tools/OmniVerse and ParaView

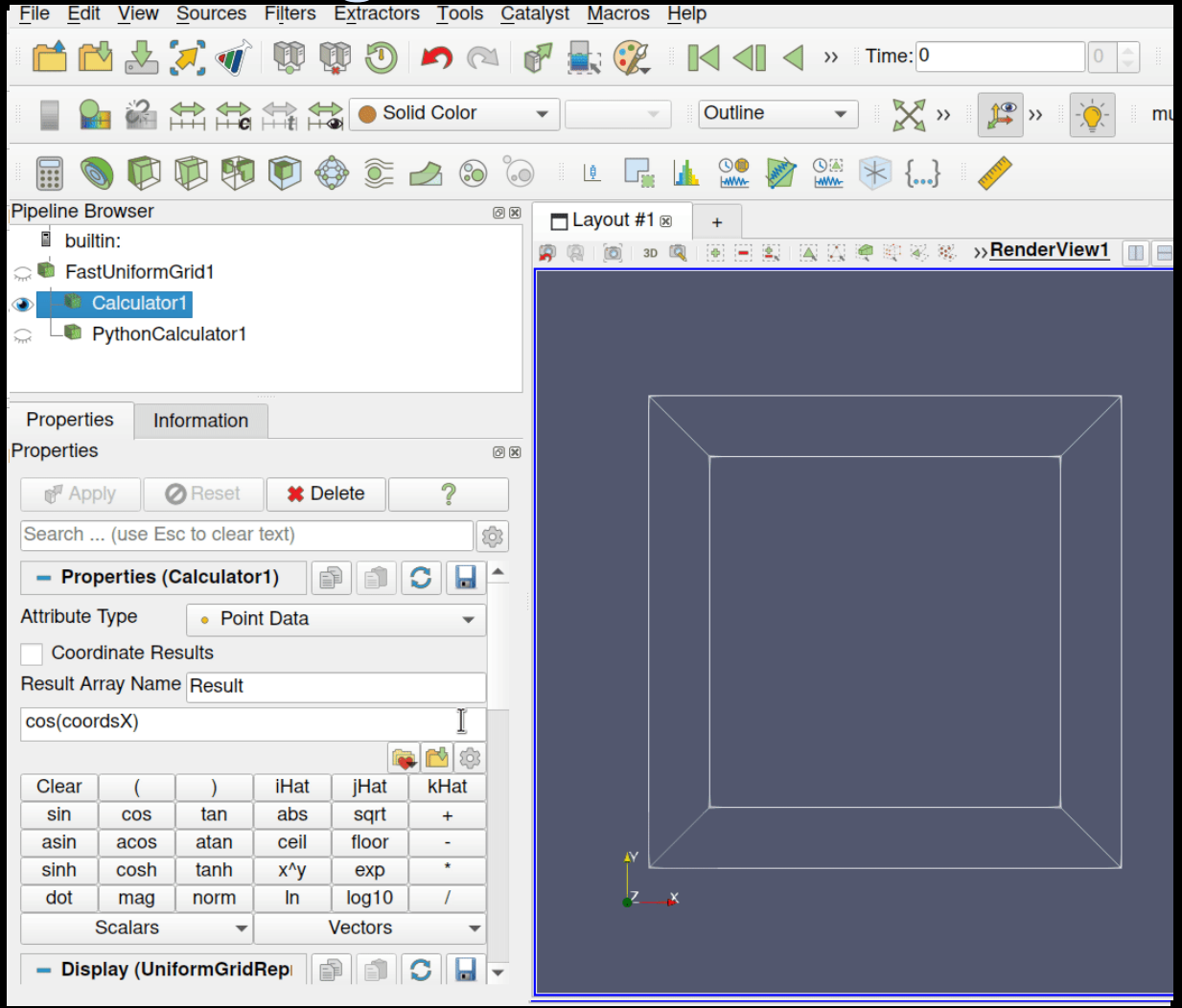


Uniform grid data exported from ParaView using the OpenVDB writer and rendered in 3D Studio Max. Data generated by the Geotechnical and Structures Laboratory at the US Army's ERDC and visualization generated by the US DoD HPCMP.

Sandia National Laboratories

New/Improved Functionality

Expression Manager



Selection Editor

Sandia National Laboratories

Selection Editor

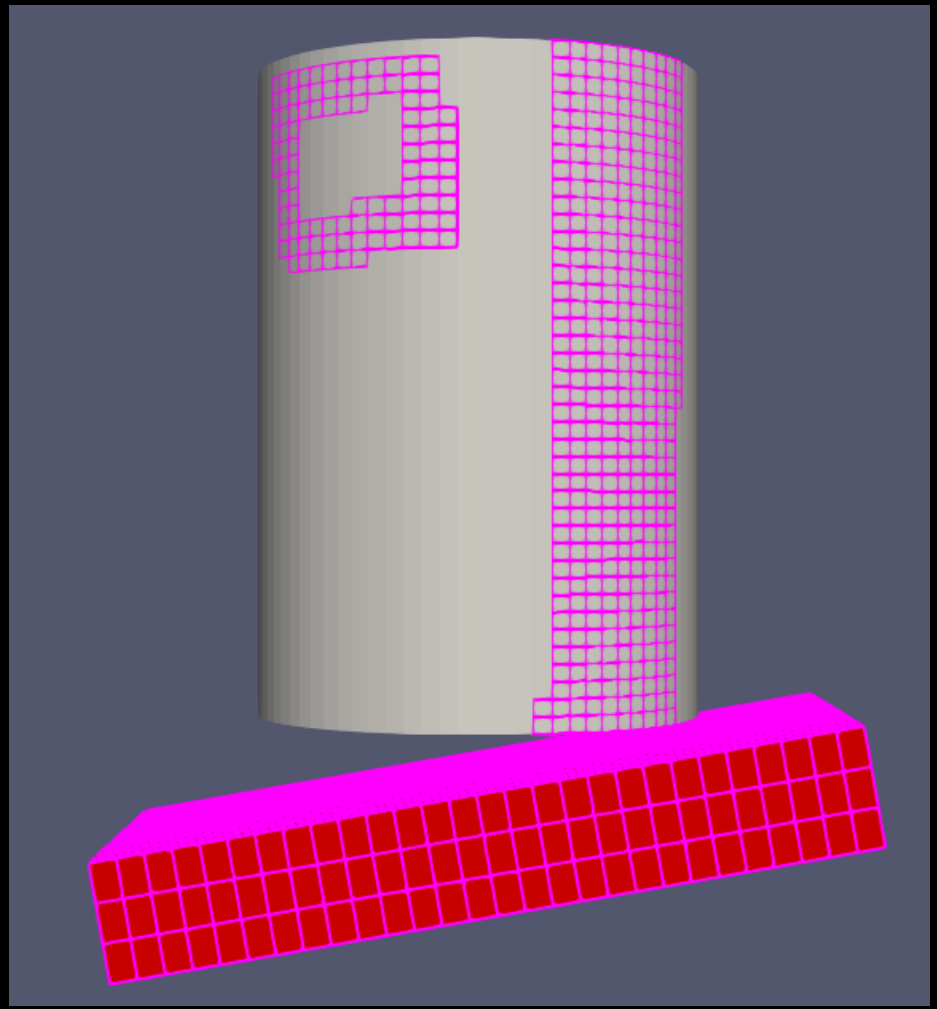
Data Producer: can.ex2

Element Type: Cell

Expression: s1|(s0^s2)|s3

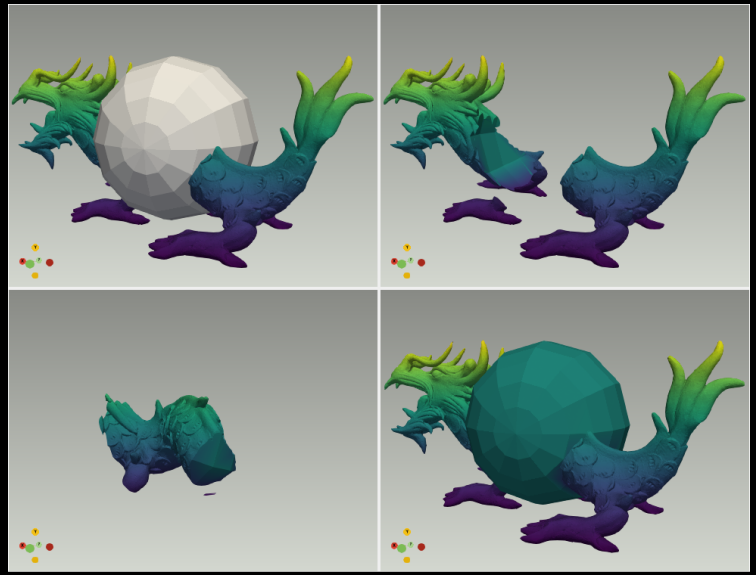
	Name	Type
1	s0	Frustum Selection
2	s1	Block Selectors Selection
3	s2	Composite ID Selection
4	s3	Query Selection

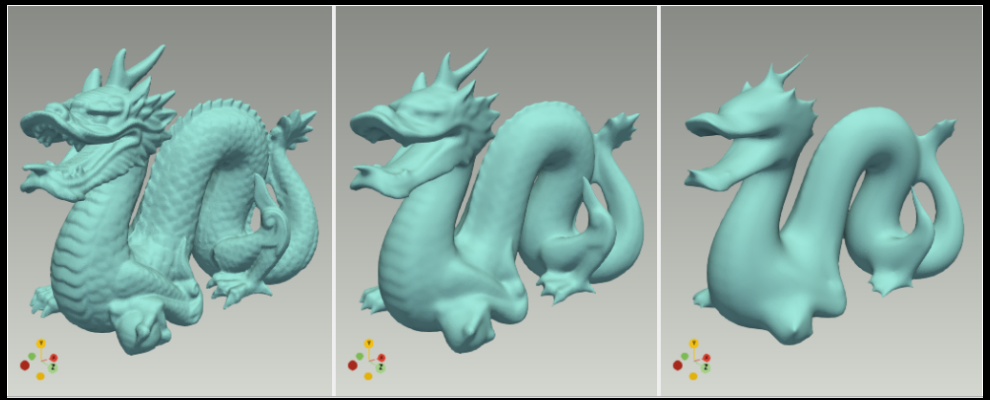
Activate Combined Selections



VESPA

- VTK Enhanced with Surface Processing Algorithms
  - Wrapping over the CGAL library
    - VTK Library
    - ParaView Plugin
- Alpha Wrapping, Boolean Operations, Hole Filling, Isotropic Remesher, Mesh Deformation, Mesh Subdivision, Region Fairing, Shape Smoothing, 2D Delaunay





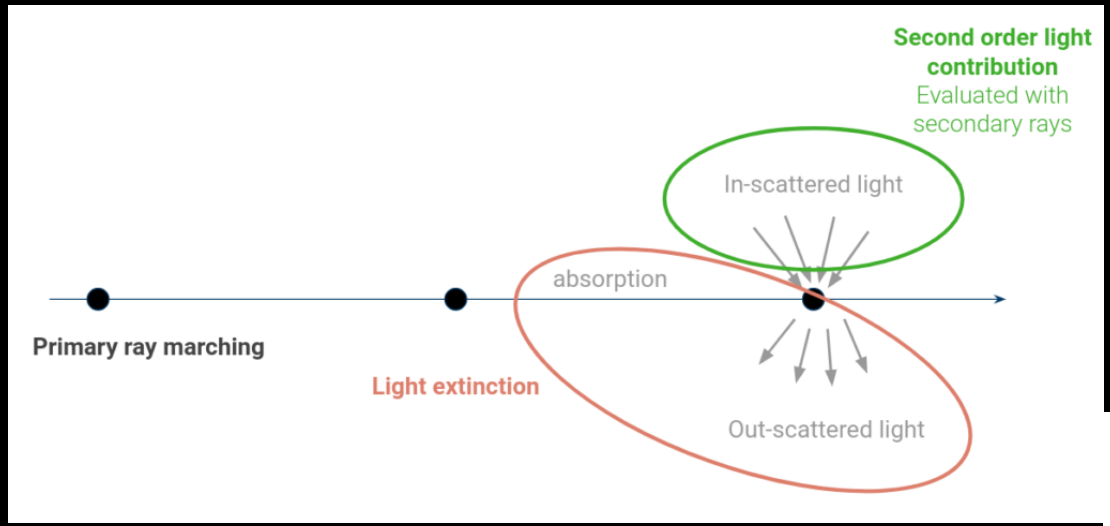
Sandia National Laboratories

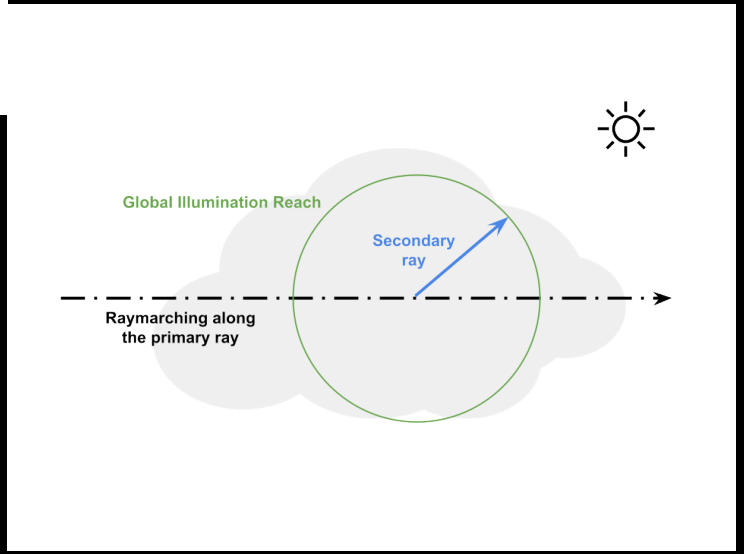
Sandia National Laboratories

Rendering Improvements

Volume Rendering with Scattering Model

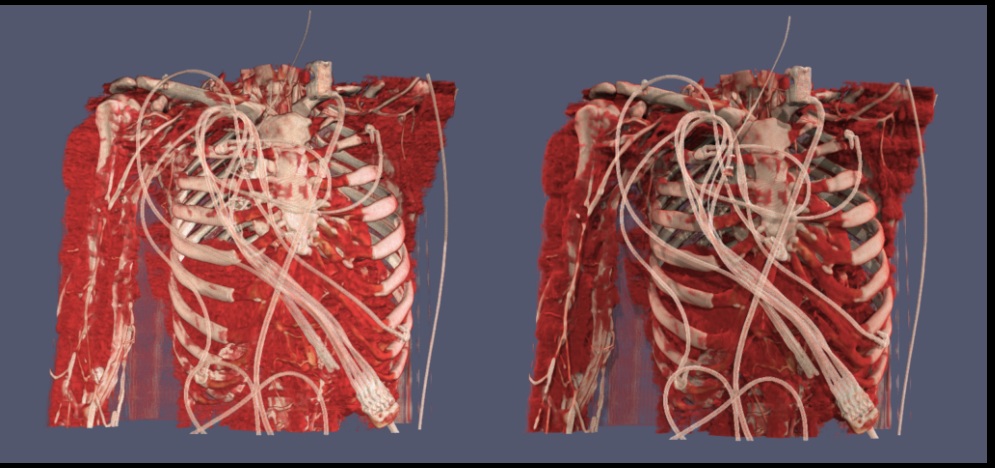
Sandia National Laboratories

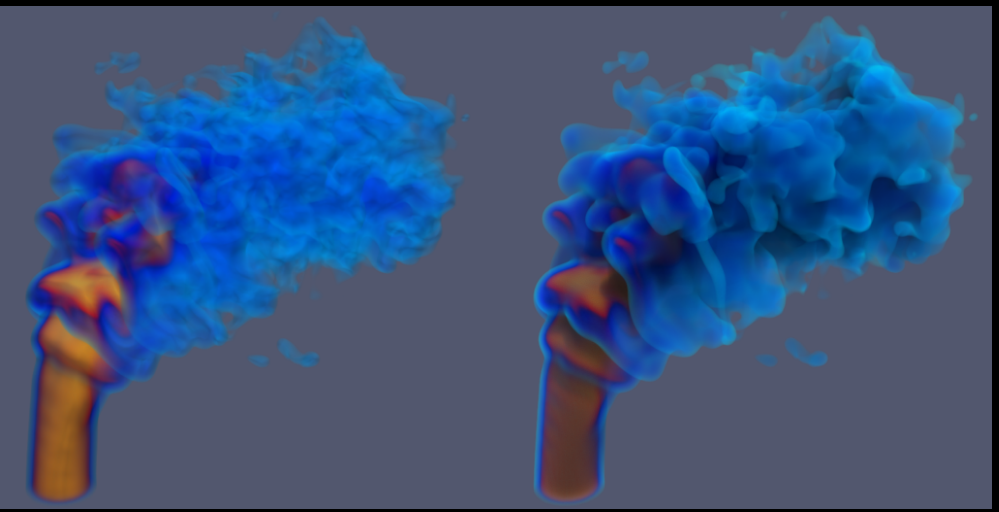


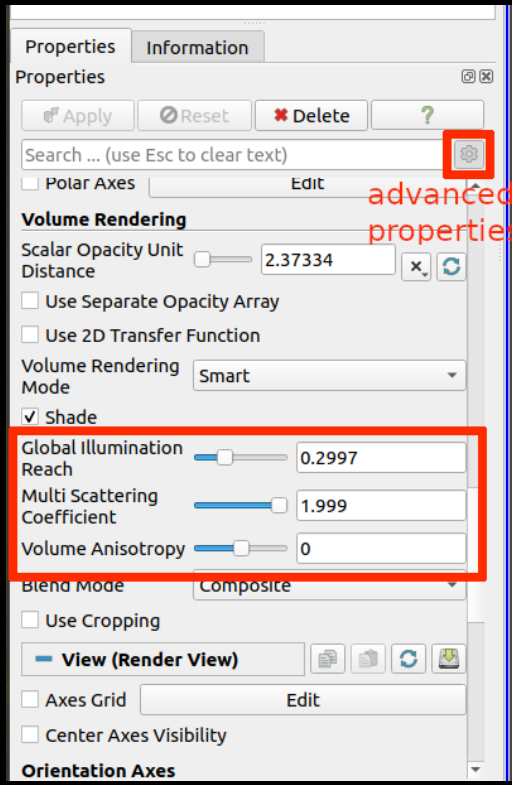


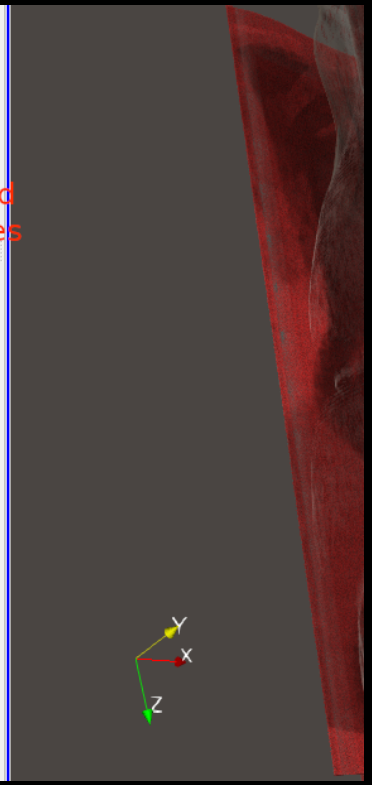
Volume Rendering with Scattering Model

Sandia National Laboratories



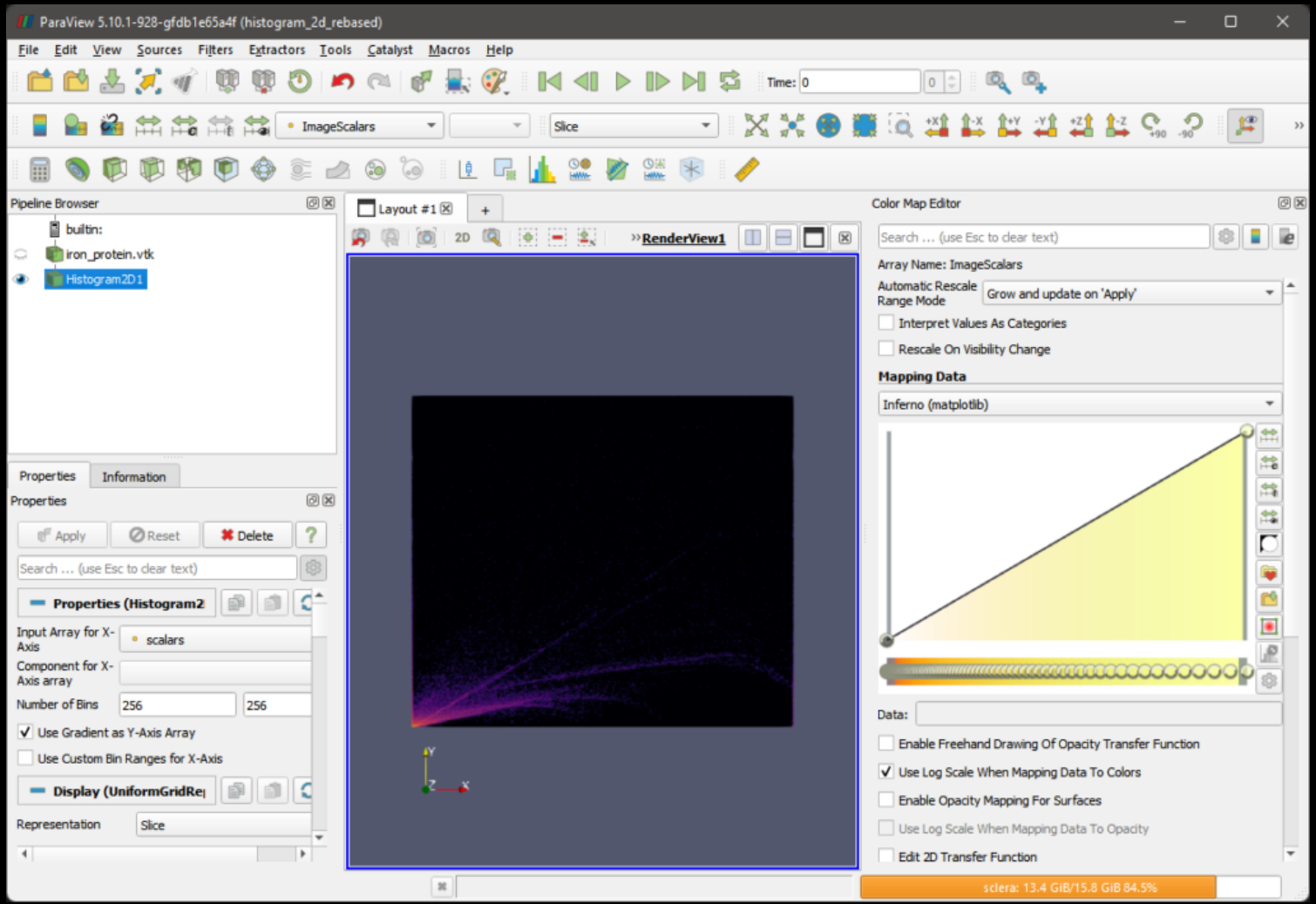






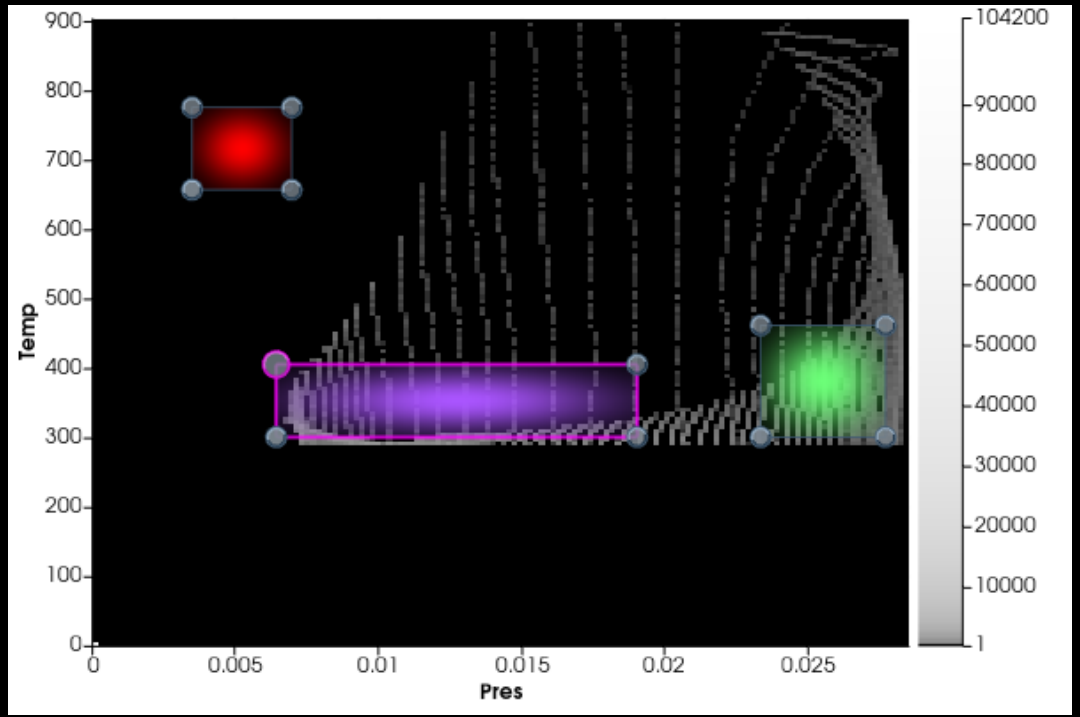
2D Transfer Functions

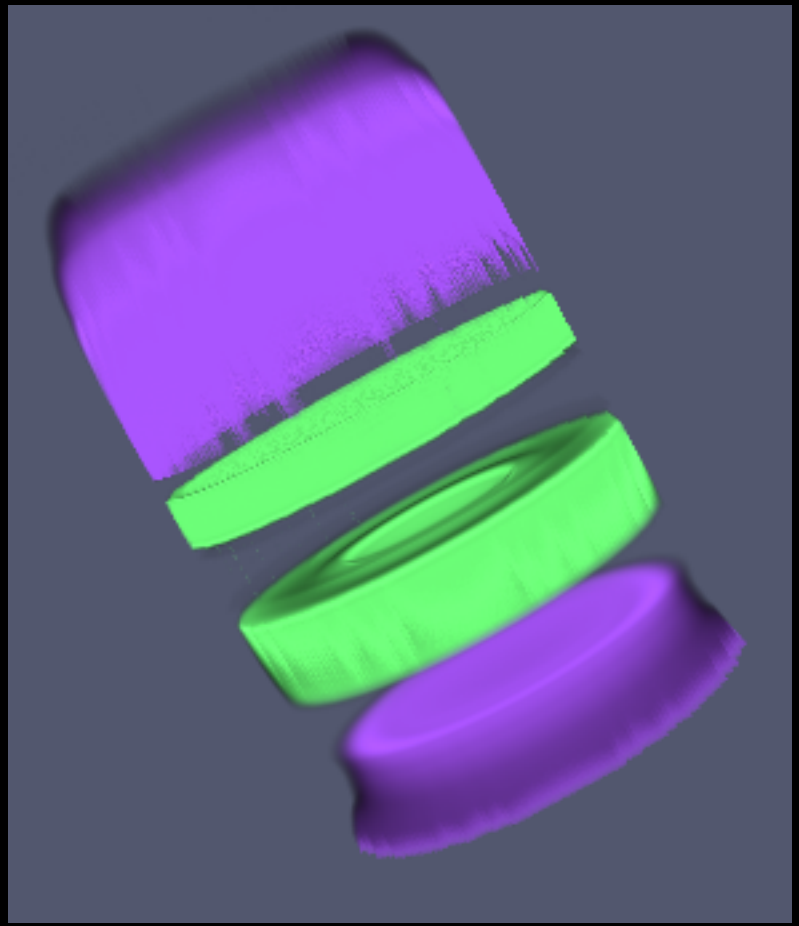
Sandia National Laboratories



2D Transfer Functions

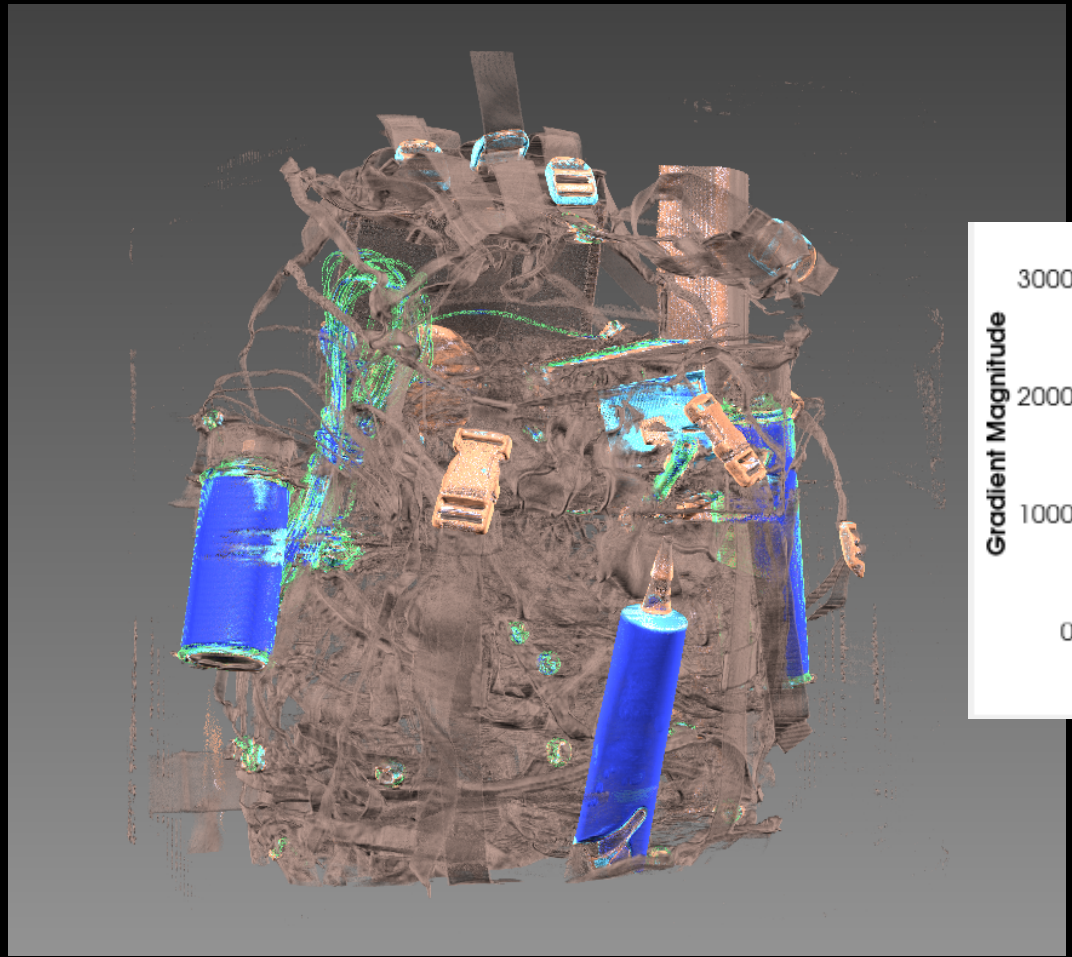
Sandia National Laboratories

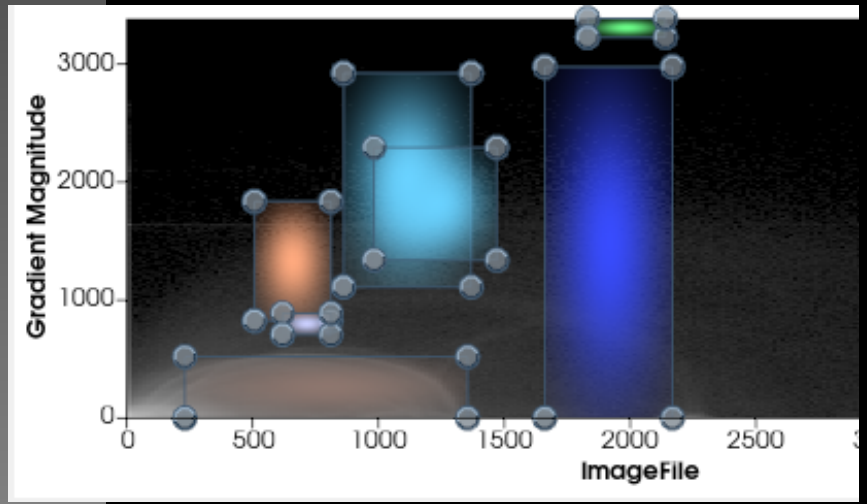




2D Transfer Functions (backpack dataset)

Sandia National Laboratories





Sandia National Laboratories

Performance Improvements

Performance Improvements

Sandia National Laboratories

- Work that built on recent improvements to vtkSMPTools
- Several approaches have contributed to the performance improvements shown below which include:
  - employing multithreading using vtkSMPTools
  - using caching whenever it's deemed beneficial
  - utilizing memory pools
  - improving certain cell/point locator operations
- Highlight a few, the full list is here:
  - https://www.kitware.com/vtk-paraview-filters-performance-improvements/

Performance Improvements

Sandia National Laboratories

1. **vtkMergeVectorComponents** is a filter that is responsible for merging three different PointData/CellData arrays into one array with tuple size = 3. Its performance has been improved by employing multithreading using vtkSMPTools.

1 Thread	0.510 sec
8 Threads	0.152 sec
Parallel Efficiency	42%
Speed-up	**x3.35**

2. **vtkExtractVectorComponents** is a filter that is responsible for creating three different arrays PointData/CellData out of one array with tuple size = 3. Its performance has been improved by employing multithreading using vtkSMPTools.

1 Thread	0.561 sec
8 Threads	0.150 sec
Parallel Efficiency	47%
Speed-up	**x3.74**

Performance Improvements

Sandia National Laboratories

8. **vtkStreamTracer** is a filter that integrates vector fields to generate streamlines. Its performance has been improved by 1) employing multithreading using vtkSMPTools and 2) minimizing the usage of the locators used for integration.

Surface DataSet	**Before with Point Locator**	**After with Point Locator**	**After with Cell Locator**
1 Thread	71.8 sec	35.259 sec	11.241 sec
8 Threads	–	5.892 sec	1.857 sec
Parallel Efficiency	–	75%	76%
Speed-up	–	x5.98	x6.05
Before-After Speed-up	–	**x12.18**	**x38.66**

Volume DataSet	**Before with Point Locator**	**After with Point Locator**	**After with Cell Locator**
1 Thread	8.936 sec	5.843 sec	7.189 sec
8 Threads	–	1.179 sec	2.431 sec
Parallel Efficiency	–	62%	37%
Speed-up	–	x4.96	x2.96
Before-After Speed-up	–	**x7.6**	**x3.67**

Performance Improvements

Sandia National Laboratories

9. **vtkParticleTracer** is a filter that integrates a vector field to advect particles (changes in this filter also affect **vtkParticlePathFilter/vtkStreaklineFilter**). Its performance has been improved by 1) employing multithreading using vtkSMPTools, 2) caching the cell bounds of the input mesh, and 3) utilizing transformation techniques to avoid rebuilding the cell locator when the input dataset at each timestep is a linear transformation of the first timestep.

	Before with Cell Locator	**After with Point Locator and Linear Transform Optimization = Off**	**After with Cell Locator and Linear Transform Optimization = Off**	**After with Point Locator and Linear Transform Optimization = On**	**After with Cell Locator and Linear Transform Optimization = On**
1 Thread	1362.5 sec	1301.5 sec	1752.5 sec	200.66 sec	109.25 sec
8 Threads	–	165.68 sec	223.7 sec	54.425 sec	17.244 sec
Parallel Efficiency	–	98%	98%	46%	79%
Speed-up	–	x7.85	x7.83	x3.68	x6.33
Before-After Speed-up	–	**x8.22**	**x6.09**	**x25.03**	**x79.01**

Performance Improvements

Sandia National Laboratories

10. **vtkGeometryFilter** is a filter that extracts the boundary geometry (surface) of a dataset. Its performance has been improved by 1) replacing all the existing algorithms with the ones found in **vtkDataSetSurfaceFilter** and utilizing the existing multithreaded infrastructure, 2) improving the existing HashTables query speed by removing duplicate faces when found, 3) consuming 50% less memory by identifying if vtkIdType (long long) or int cell point IDs are sufficient, and 4) converting a vtkUnstructuredGrid to vtkPolydata almost instantaneously if it contains only vertices, lines, polys, or strips—therefore it's already a surface. vtkGeometryFilter delegates to **vtkDataSetSurfaceFilter** when input is vtkUnstructuredGrid and it contains nonlinear cells. **vtkDataSetSurfaceFilter** will be removed when **vtkGeometryFilter** is extended to support nonlinear cells.

UnstructuredGrid with Tetra	**vtkDataSetSurfaceFilter**	**Before vtkGeometryFilter**	**After vtkGeometryFilter**
1 Thread	4.749 sec	64.820 sec	2.650 sec
8 Threads	–	9.875 sec	0.440 sec
Parallel Efficiency	–	82%	75%
Speed-up	–	x6.56	x6.02
Before-After Speed-up	–	–	**x22.44**
vtkDataSetSurfaceFilter-After Speed-up	–	–	**x10.79**

UnstructuredGrid with triangles	**vtkDataSetSurfaceFilter**	**Before vtkGeometryFilter**	**After vtkGeometryFilter**
1 Thread	1.599 sec	1.801 sec	0.008 sec
8 Threads	–	0.363 sec	0.002 sec
Parallel Efficiency	–	62%	50%
Speed-up	–	x4.96	x4
Before-After Speed-up	–	–	**x191.5**
vtkDataSetSurfaceFilter-After Speed-up	–	–	**x799.5**

Lots More Features/Fixes

- https://www.kitware.com/tag/paraview/

Sandia National Laboratories