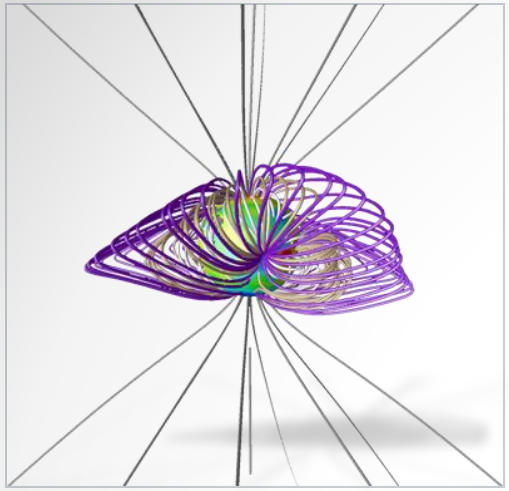
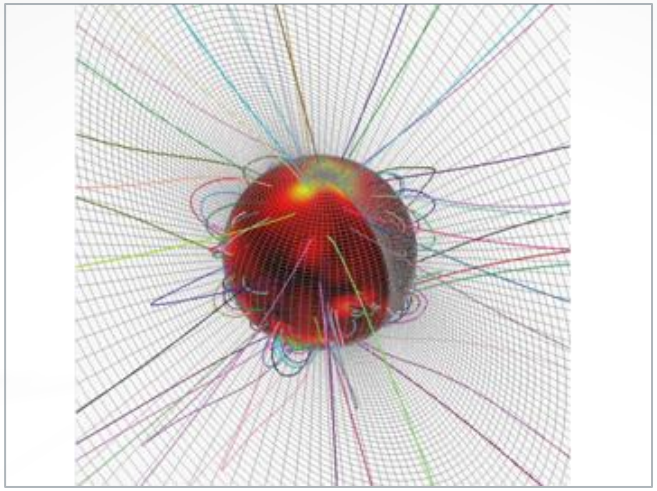
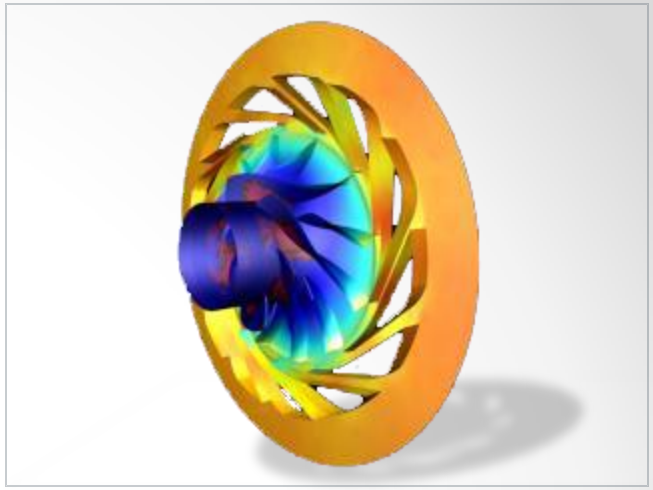
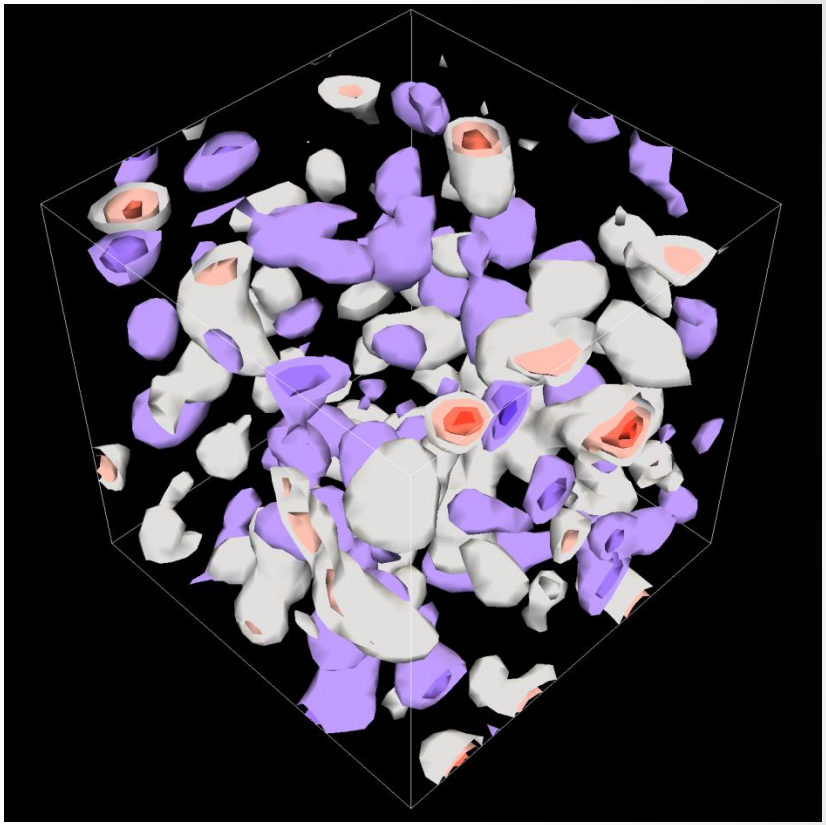




Scientific Visualization

Katia Oleinik: koleinik@bu.edu

*Scientific Visualization (or **SciVis**)* is a branch of computer science which is concerned with graphical representation of real-world objects and scientific data. It allows researches to better understand and study the systems that normally cannot be seen or studied.

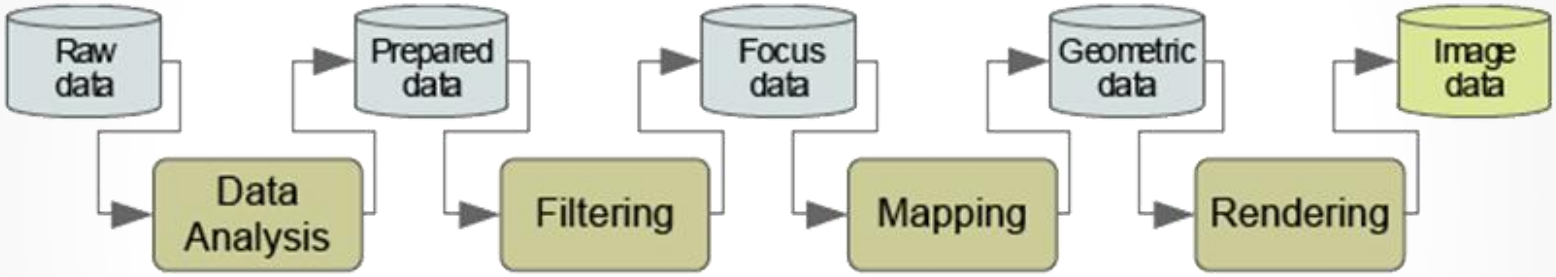


Contours of constant topological charge density for a quantum field theory on a lattice
Claudio Rebbi, Adam Avakian, et al.
BU Physics (2009)

Principal Challenges

- Vast amount of data to process requires extensive computing resources
- Interactive display and data analysis are necessary to establish previously unknown correlation between data in space and/or time.
- Time-dependent simulations require substantial amount of CPU and GPU resources
- Data often include many additional spatial and temporal dimensions which make it difficult to analyze them with conventional 2D graphs.

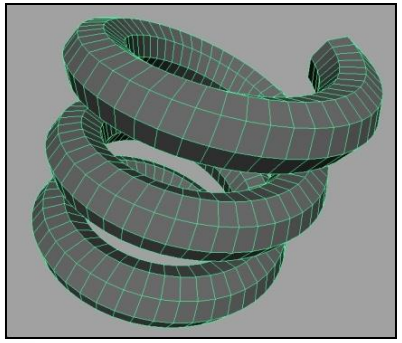
Workflow



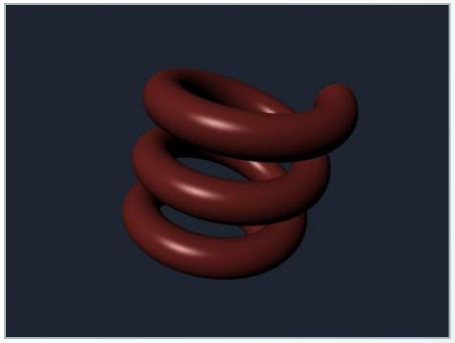
Data Analysis/Preparation

191	214	206	113	134	135	101	120	135	102	119	147	113	129	165
117	131	170	125	138	170	158	168	195	219	227	246	244	251	255
249	254	255	197	202	222	157	163	187	156	161	190	165	172	200
218	224	246	250	255	255	252	255	251	251	255	252	245	249	250
251	255	255	188	192	204	196	200	212	234	238	250	177	181	192
111	114	133	122	121	155	154	151	194	182	176	220	243	240	255
253	252	255	250	251	246	254	254	252	255	255	255	255	255	255
254	254	254	255	255	255	255	255	255	254	254	254	255	255	255
249	249	249	249	249	249	252	252	252	255	221	155	255	255	255
255	255	255	255	255	255	255	255	255	255	255	255	255	255	255
255	255	255	255	255	255	255	255	255	253	253	253	252	252	252
252	252	252	253	253	253	255	255	255	255	255	255	255	255	255

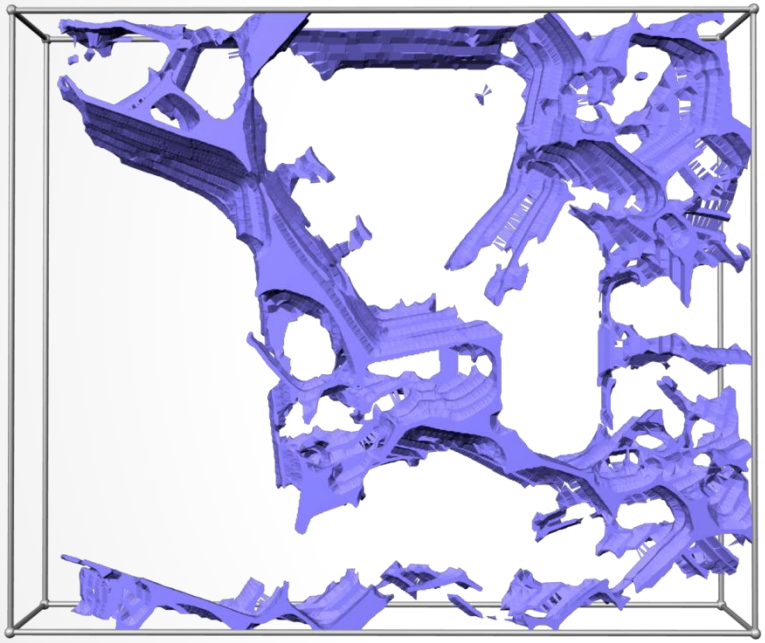
Scientific Visualization



Computer Graphics



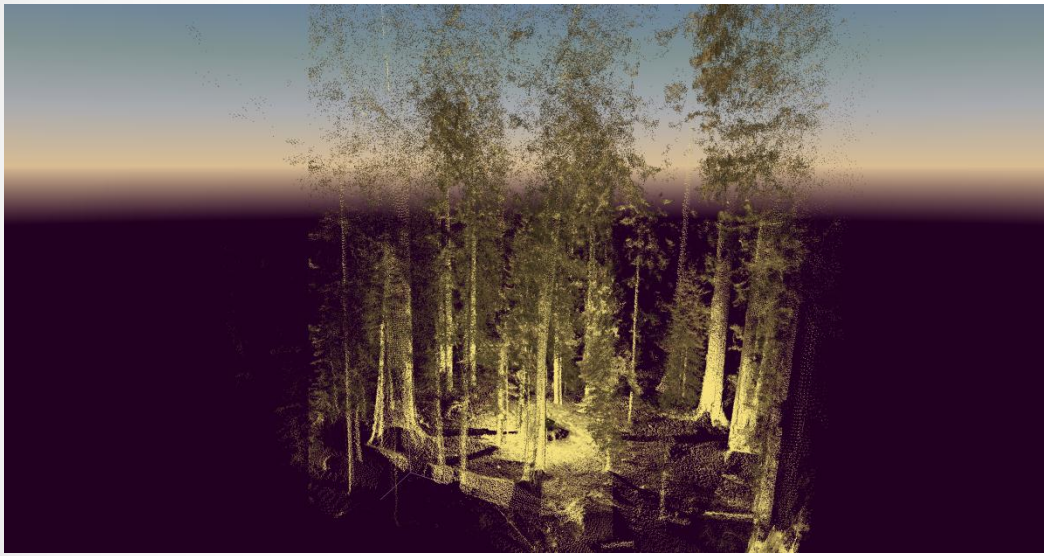
SciVis Topics: Surface Rendering



Melt boundaries produced using alpha-shape based reconstruction.
Gordana Garapic, Ulrich Faul, and Erik Brisson, BU (2009)

Surface rendering is a process of generating a computer surface representation from a given data. The computer surface model is described in terms of primitive geometric objects such as points, polygons, normals. Often an additional information such as texture, lighting and shading used to improve the quality of the result image.

SciVis Topics: Volume Rendering

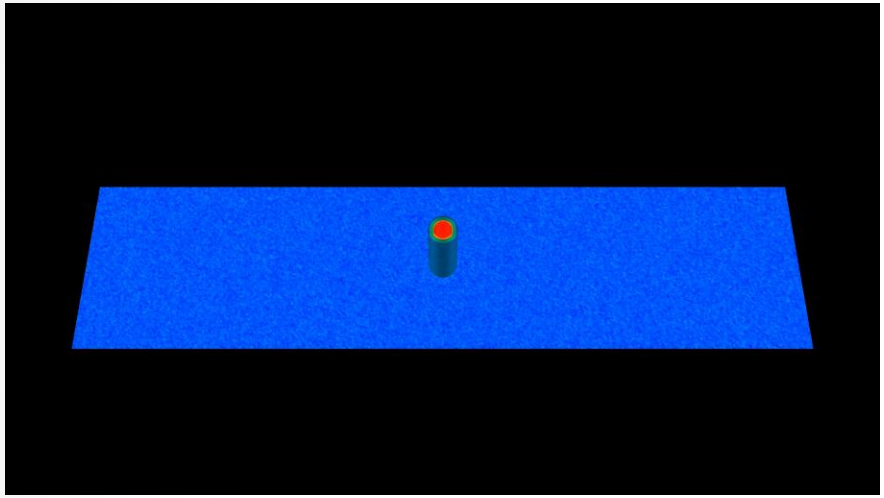


Tomographic Reconstruction of Forest Structure.
Xiaoyuan Yang1, Alan Strahler1, and Erik Brisson, BU (2008)

Volume Rendering is a set of techniques used to display a 3d Volume using 2D discretely sampled data set. A typical 3D data set is a group of 2D slice images acquired by a CT or MRI scanner.

Modern GPUs with their highly parallel structure and modern shading capabilities greatly improve the quality and speed of volume rendering.

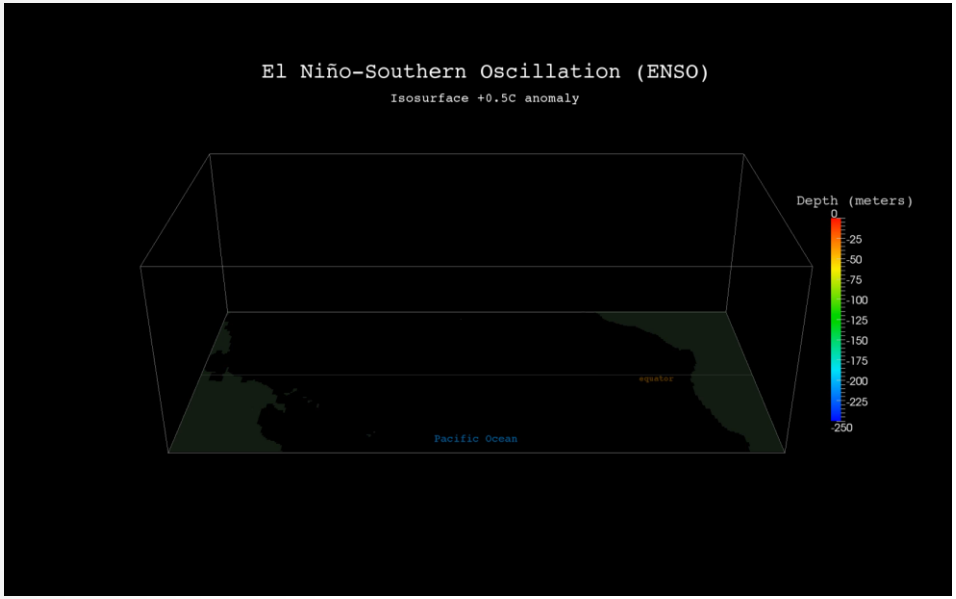
SciVis Topics: Animation



SciVis animation is a technique of creating moving images by means of 2D and 3D computer graphics.

Kinetic Farley-Buneman Turbulence.
M. Oppenheim, L. Dyrud, A. vom Endt, R. Putnam, R. Gasser, and E. Brisson, BU (2010)

SciVis Topics: Computer Simulations

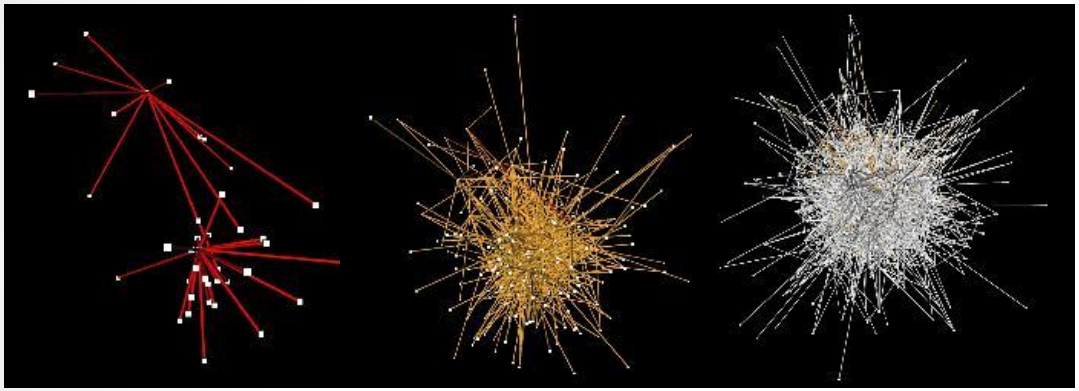


Computer Simulation is a computer program, that attempts to simulate an abstract model of a particular system.

Simulations are used to explore and gain new insights into new technology, and to estimate the performance of systems too complex for analytical solutions.

Triggering of El Niño Onset by Northern Hemisphere Sea-level Pressure Variations.
Bruce T. Anderson, Renellys Perez, and Katia Oleinik, BU (2011)

SciVis Topics: Information Visualization

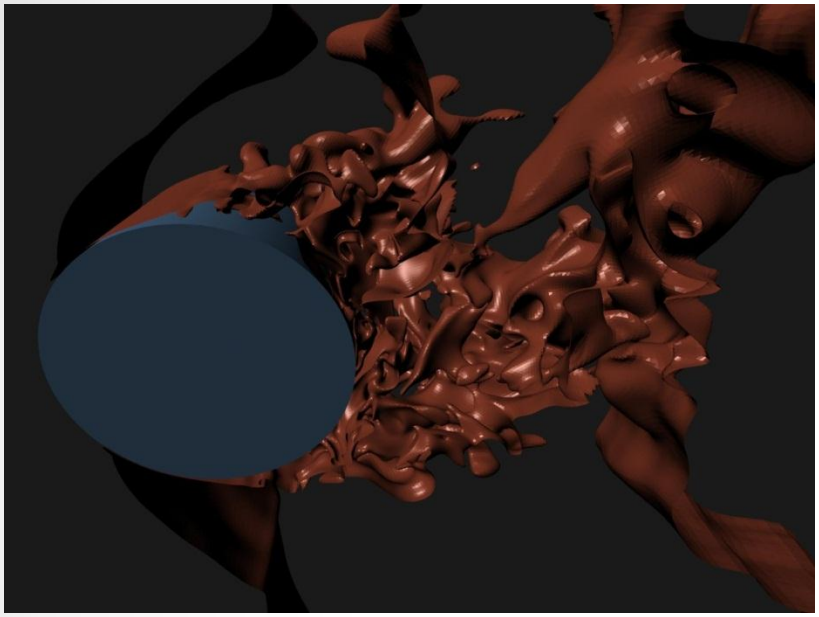


Structural Hole Theory: The Relationship between Corporate Ties and Corporate Innovation.
Erik Noyes, *BU (2008)*

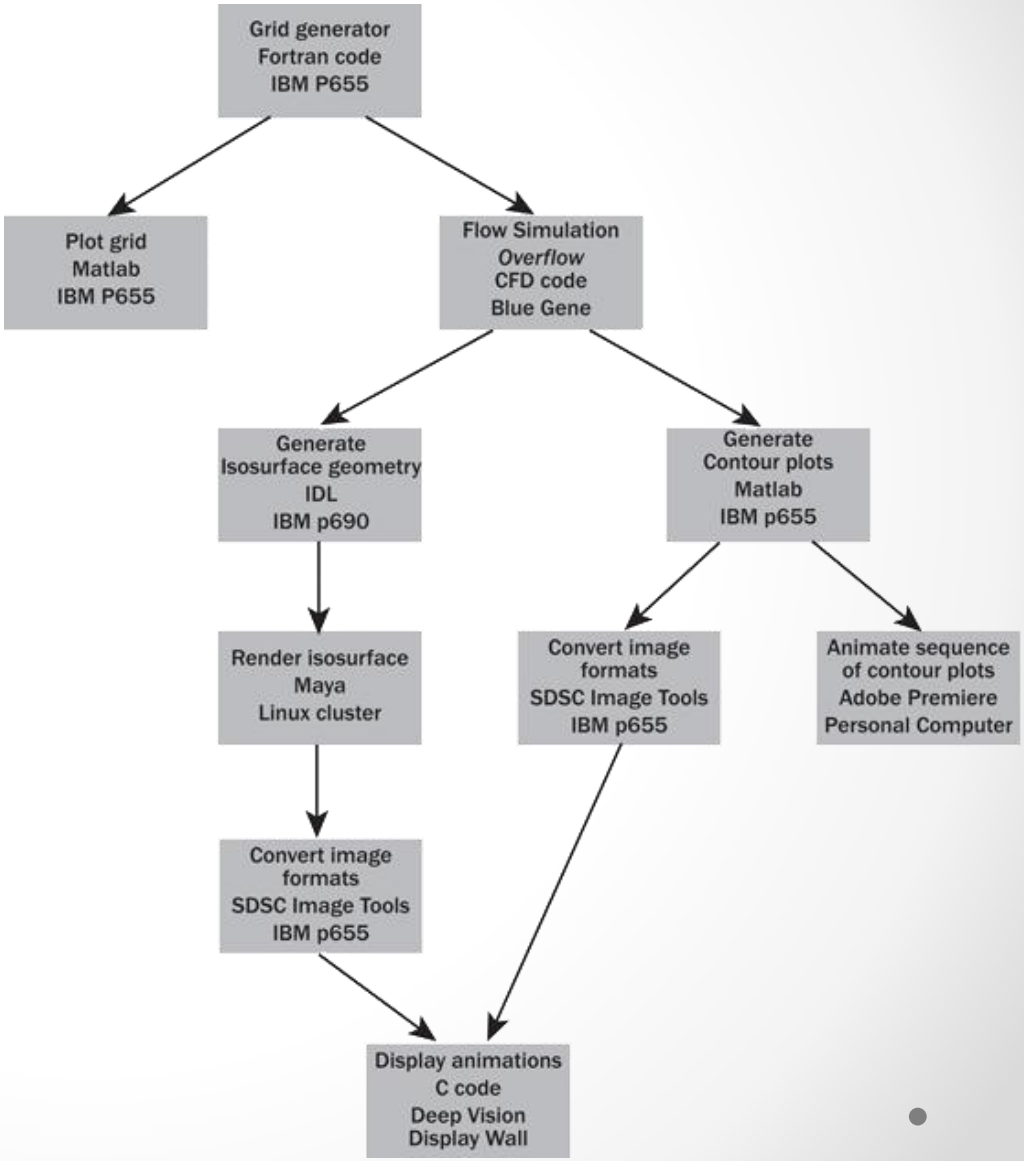
Info. Visualization is a technique of the visual representation of large-scale collections of non-numerical information, such as databases, networks, multi-dimensional relations, etc.

It helps to explore and understand large amounts of information at once, discover and study the clusters in large networks and understand the relationships between them.

Workflow Example



Simulation of vortex shedding over a right circular cylinder.
Douglas L. Sondak and Erik Brisson BU IS&T SCV (2009)



Data Analysis / Computation

The *Scientific Computing and Visualization group* (SCV) at Boston University provides a number of advanced, multiprocessor supercomputing systems for research computing. These systems are available to all University faculty, their students, and their collaborators for research and for educational use in courses related to computational science.

Data Analysis / Computation

Hardware Requirements

- Powerful multi-core CPUs and GPUs
- Increasing amount of external and internal memory

BU Facilities

- Massively parallel computing systems, including 1024-node IBM Blue Gene/L
- large shared memory systems including IBM pSeries p655
- large high-performance storage system (e.g., GPFS, a parallel file system) shared across all the platforms

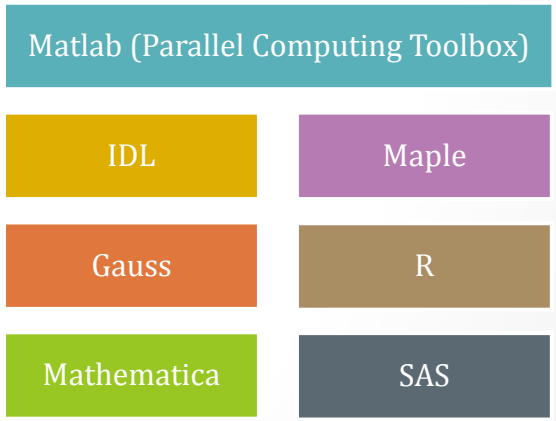
Data Analysis / Computation

Software Requirements

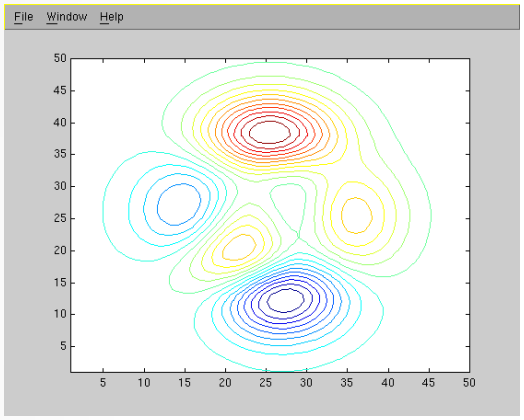
- Depending on the type of the data and the required analysis special software might be necessary to process the data

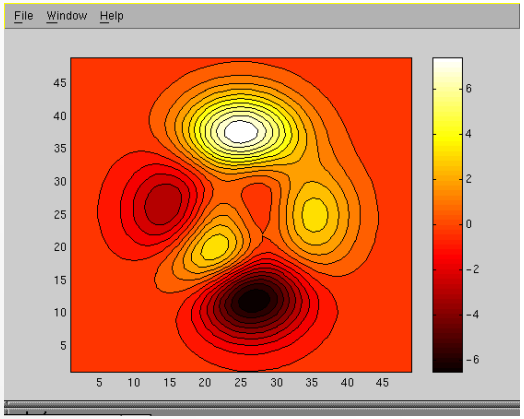
BU Facilities

- broad set of open source and commercial scientific and mathematical software packages including:



Data Analysis / Computation: Matlab





MATLAB is an interactive environment in which you can program as well as visualize your computations. It includes a set of high-level graphical functions for:

- Line plots
- Bars, histograms, pies and other graphs
- Surface plots
- Mesh plots
- Contour plots
- Animation

MATLAB also provides for low-level control in the event that the user wishes to have better control of the graphical output.

Often used in Scientific Visualization for initial processing of data including misc. mathematical operations on raw datasets.

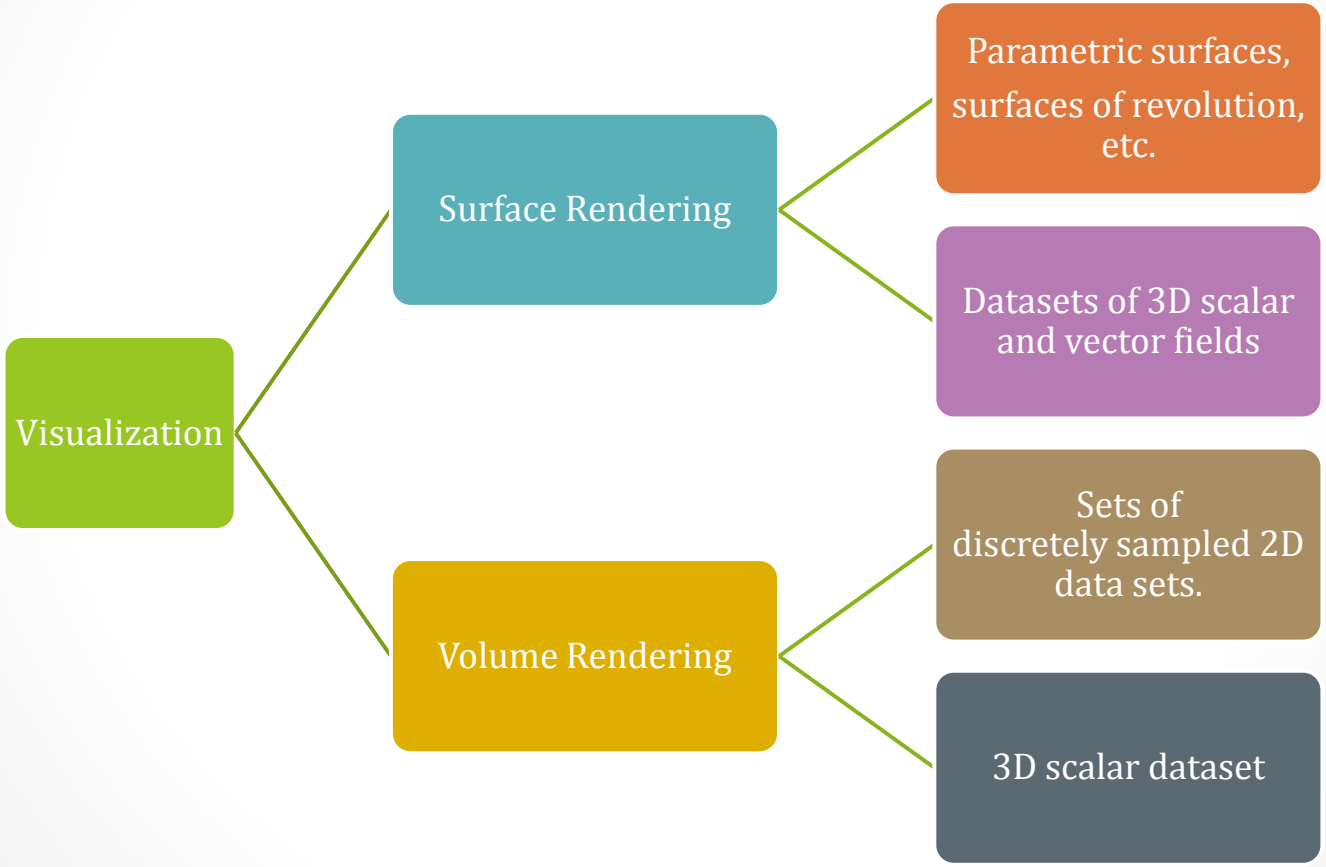
Data Analysis / Computation: MPI, OpenMP

On contemporary computers, speeding up computations is most often achieved by employing multiprocessors concurrently on shared-memory multi-cored nodes or multiprocessor distributed-memory clusters.

MPI (Message Passing Interface) is a library of communication functions to enable and enhance multiprocessing on these computer architectures.

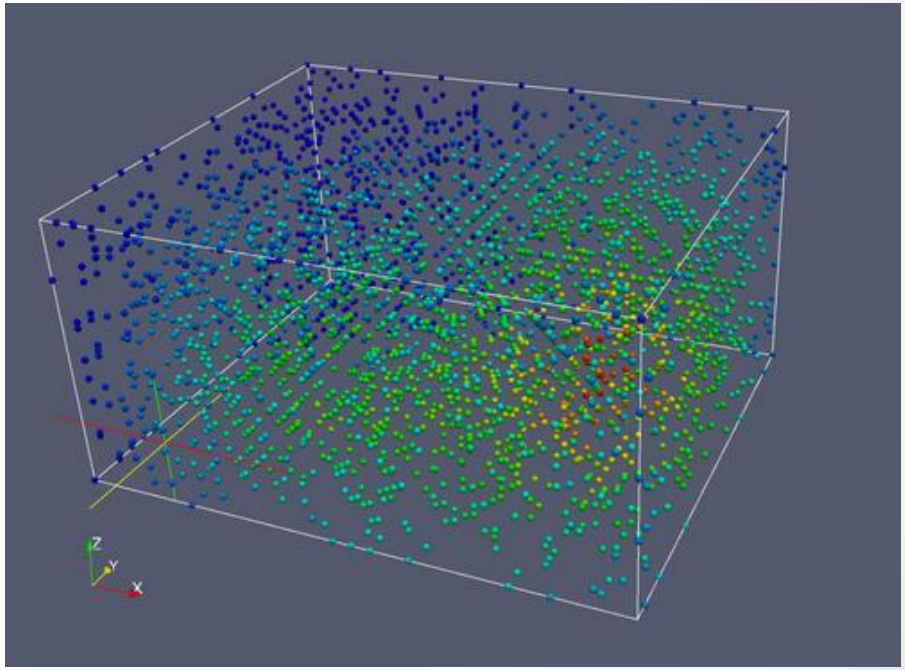
OpenMP (Open Multi-Processing) is a method for parallelizing C, C++, and Fortran programs on shared-memory systems, including multi-core PCs and blade servers.

Visualization Techniques



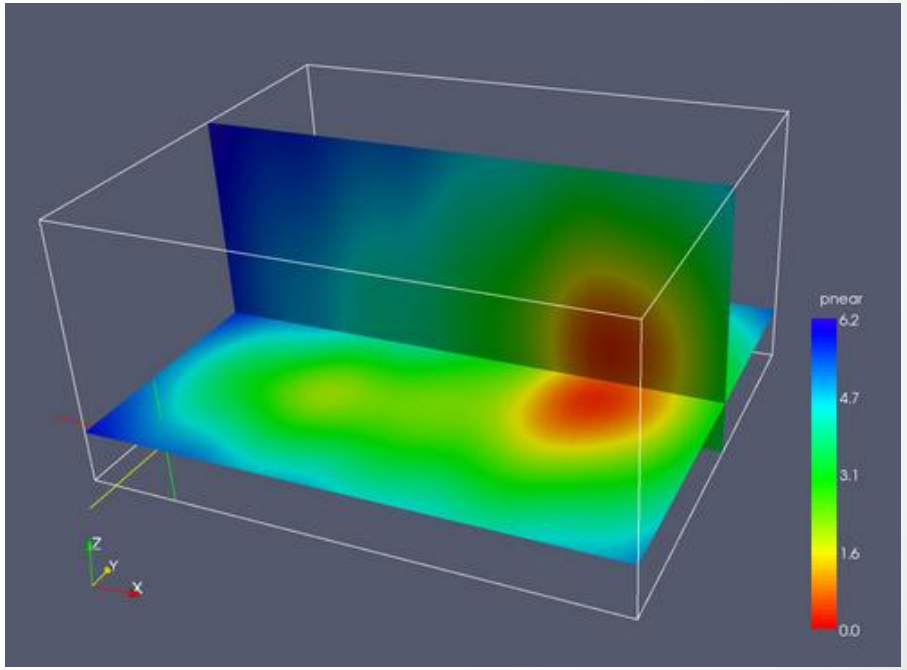
Visualization Techniques of Data Analysis

Scalar data in 3D as points



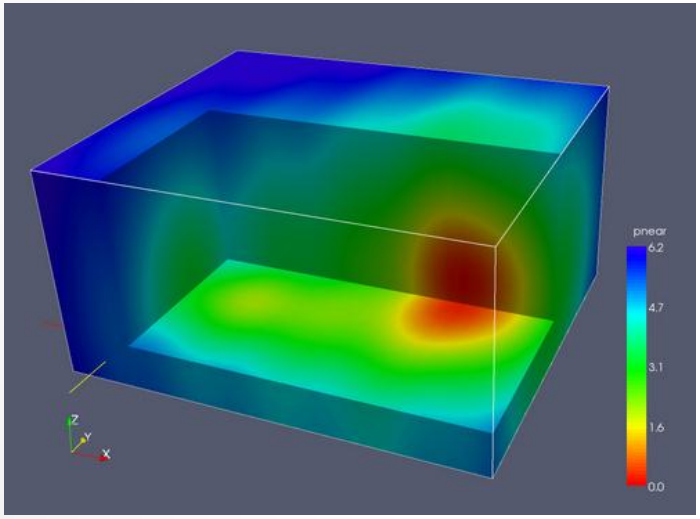
Visualization Techniques of Data Analysis

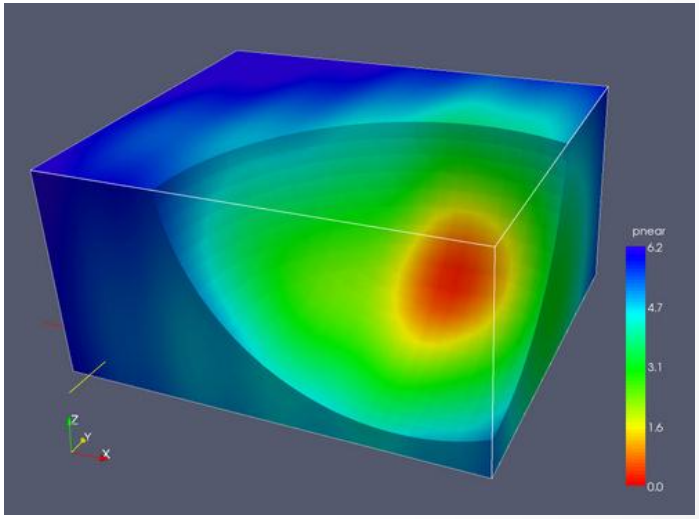
Scalar data slicing



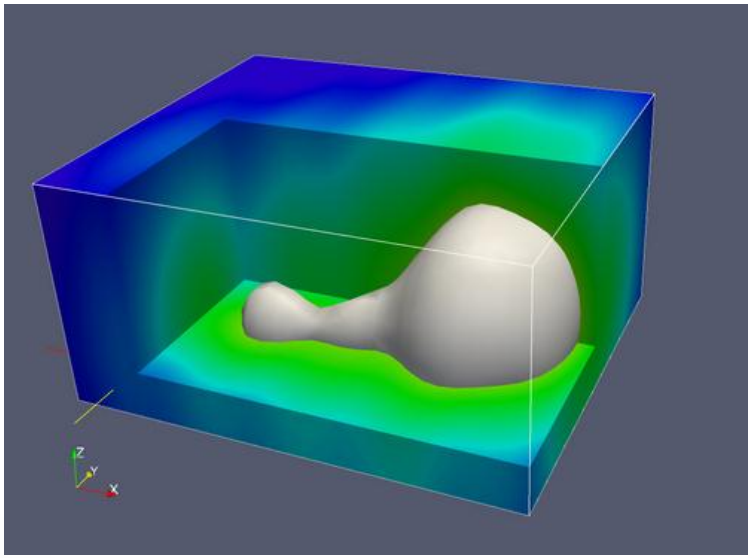
Visualization Techniques of Data Analysis

Scalar data clipping, trimming and subsetting.



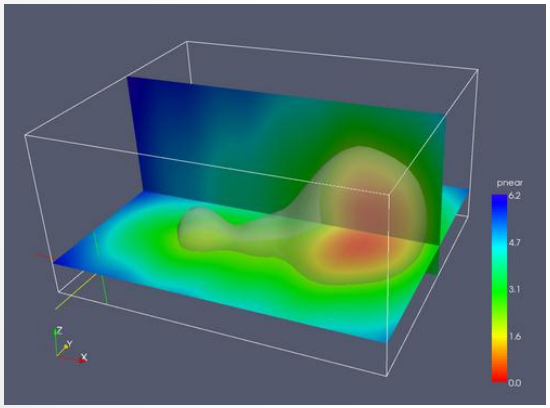


Isocurves and Isosurfaces

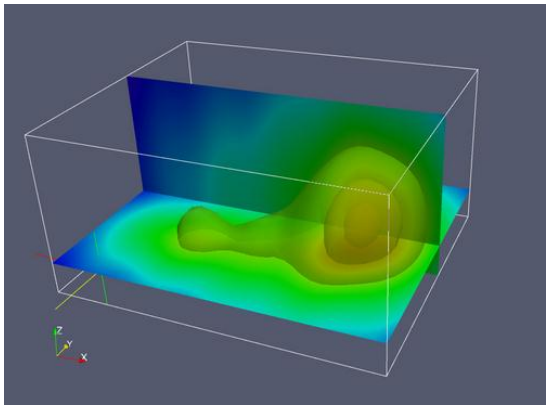


Combination of different techniques

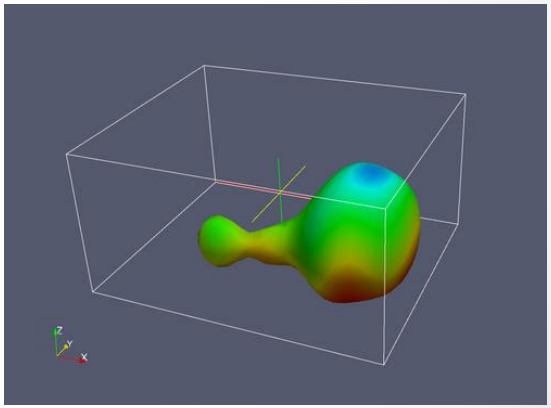
Transparent isosurface



Multiple isosurfaces

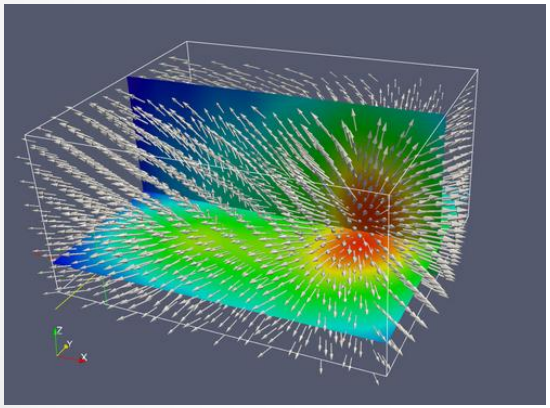


Isosurface with color mapping

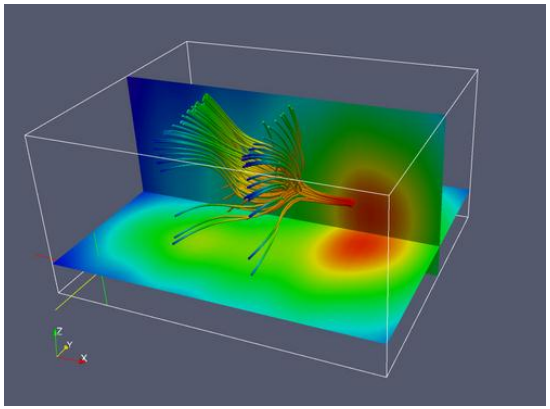


Vector analysis techniques

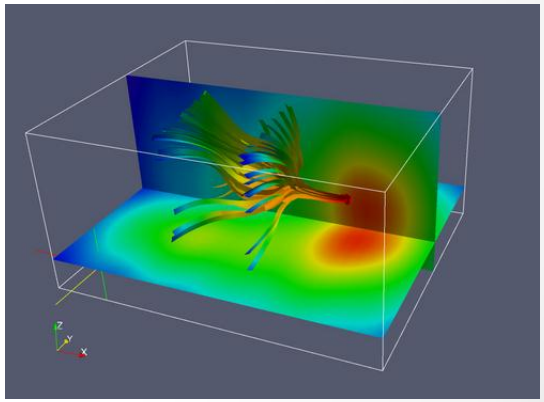
Glyphs



Streamlines

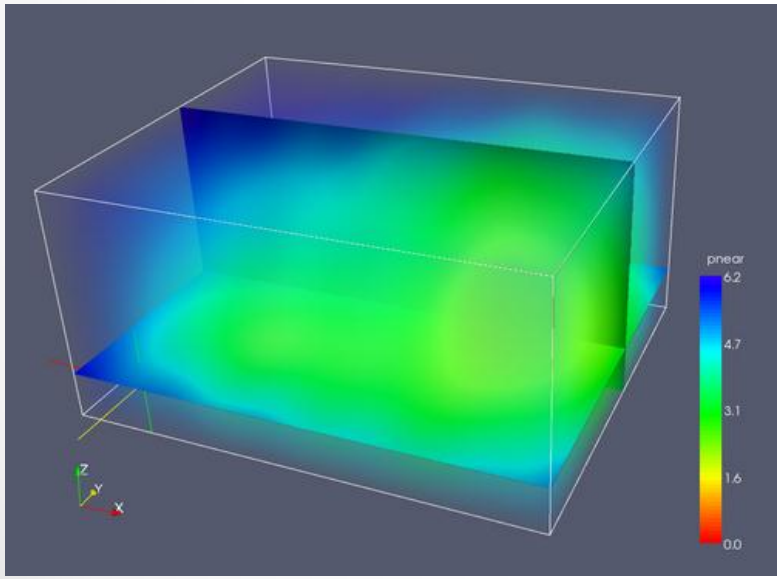


Ribbons



Volume Rendering

Transparent volume of 3D dataset



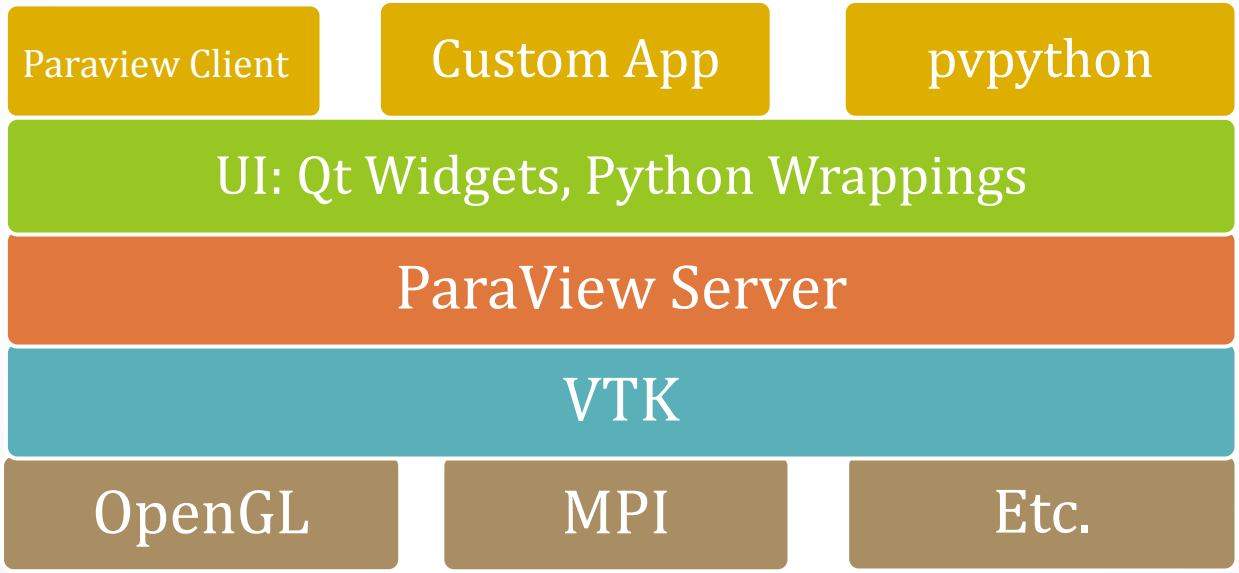
Volume created from 2D slice images acquired by a CT, MRI, or Micro CT scanner

The Visualization Toolkit (VTK) is an open-source, freely available software system for 3D computer graphics, image processing and visualization. VTK consists of a C++ class library and several interpreted interface layers including Tcl/Tk, Java, and Python.

visualization algorithms	Advanced Modeling Techniques	Cross-platform	Supports
• Scalar • Vector • Tensor • Texture • Volumes	• Implicit modeling • Polygon reduction • Mesh Smoothing • Cutting • Contouring • Delaunay triangulation	• Linux • Windows • Mac • Unix	• Parallel processing • Various databases • 3D interaction widgets

ParaView is an open-source, multi-platform application designed to visualize data sets of varying sizes from small to very large. The goals of the ParaView project include developing an open-source, multi-platform visualization application that supports distributed computational models to process large data sets.


ParaView
Parallel Visualization Application

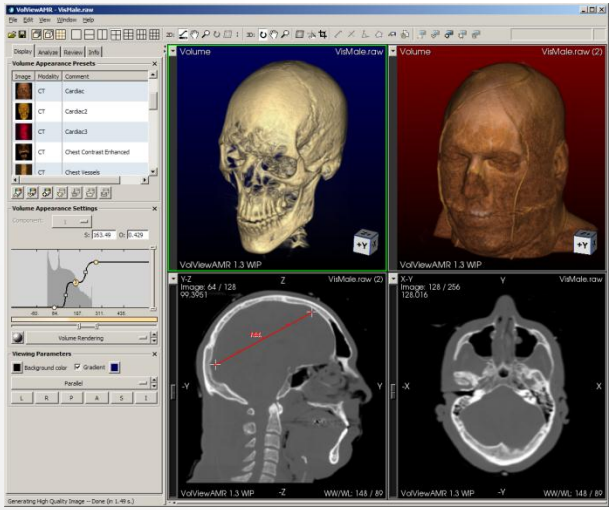

Paraview Client
Custom App
pvpython
UI: Qt Widgets, Python Wrappings
ParaView Server
VTK
OpenGL
MPI
Etc.

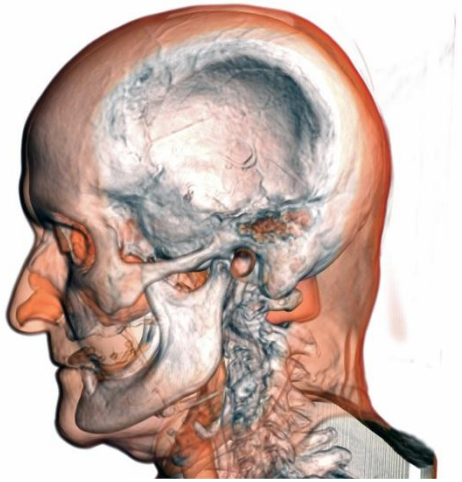
ParaView supports many input file types and the list grows as more types are added:

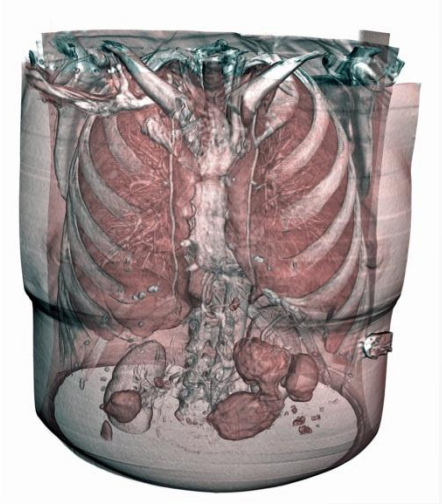
- ParaView Data (.pvd)
- VTK (.vtk, .vtu, .vti, .vts, .vtr, .vtm, …)
- CCSM (.nc, .cdf, .ncd …)
- Image Data and Raw Image Data (.jpg, .png, .tiff, .nrrd, nhrd, …)
- Plot3D and Plot2D (.p2d, .xyz, …)
- PLY polygonal file format
- NetCDF(.nc)
- LODI (.nc, .cdf, .ncd, …)
- Flash multiblock files
- Many others

VolView is an open-source, intuitive, interactive system for volume visualization that allows researchers to quickly explore and analyze complex 3D medical or scientific data on Windows, Mac and Linux computers. Users can easily load and interactively explore datasets using 2D and 3D display methods and tools. 3D tools include volume rendering, maximum intensity projections, and oblique reformatting.

VolView


Volume
VisMale.raw
VolViewAMR 1.3 WIP
Volume
VisMale.raw (2)
VolViewAMR 1.3 WIP
Y-Z
VisMale.raw (2)
VolViewAMR 1.3 WIP
X-Y
VisMale.raw
VolViewAMR 1.3 WIP

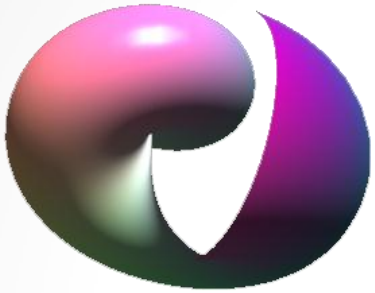


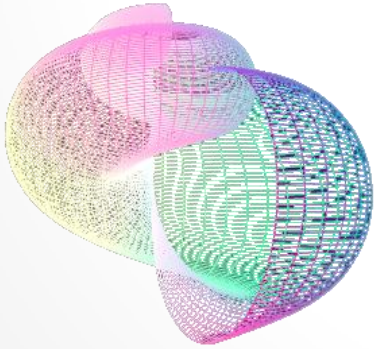


Rendering

BU DVD Wall

Rendering

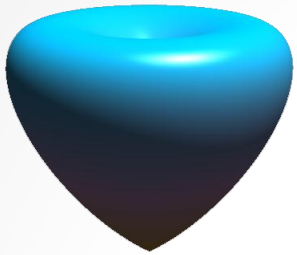


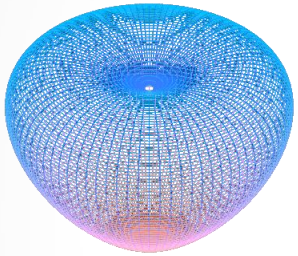


OpenGL is a standard API developed by Silicon Graphics and managed by Khronos Group.

- Cross-language
- Cross-platform (hardware and OS independent)
- 2D and 3D graphics
- More than 250 different function calls
- Used in CAD, virtual reality, SciVis, flight simulation, video games.

Rendering





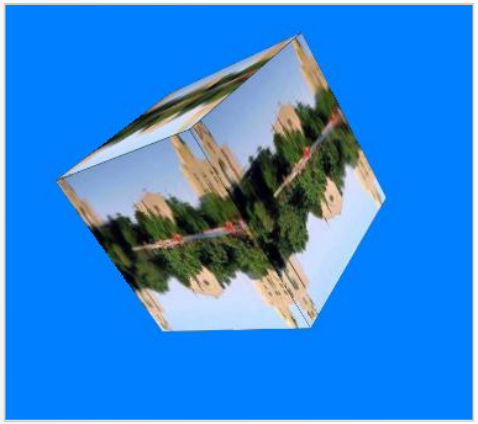
OpenGL presents a single, uniform interface with different 3D accelerators, hides different capabilities of hardware platforms, requiring full support of all OpenGL features for all implementations.

Open GL accepts primitives such as points, lines, polygons and converts them into pixels.

Several libraries are built on top of OpenGL to provide features not available in OpenGL itself.

OpenGL standard allows individual vendors to provide additional functionalities through extensions.

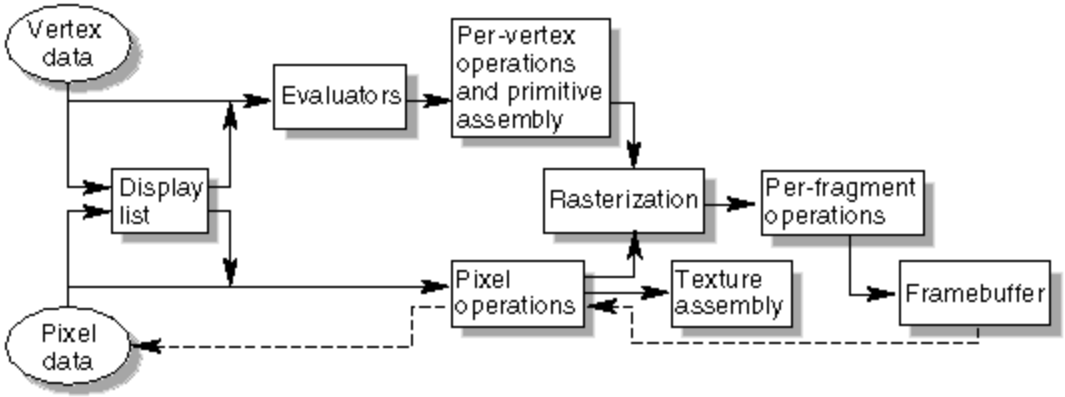
Rendering



OpenGL features:

- Geometric and raster primitives
- RGBA or color index mode
- Immediate drawing mode or Display Lists
- Model and Viewing transformations
- Lighting
- Texture mapping
- Depth buffer (hidden surface removal)
- Transparency (alpha – blending)
- Anti-aliasing

Rendering


Vertex data
Evaluators
Per-vertex operations and primitive assembly
Display list
Rasterization
Per-fragment operations
Pixel operations
Texture assembly
Framebuffer
Pixel data

Rendering

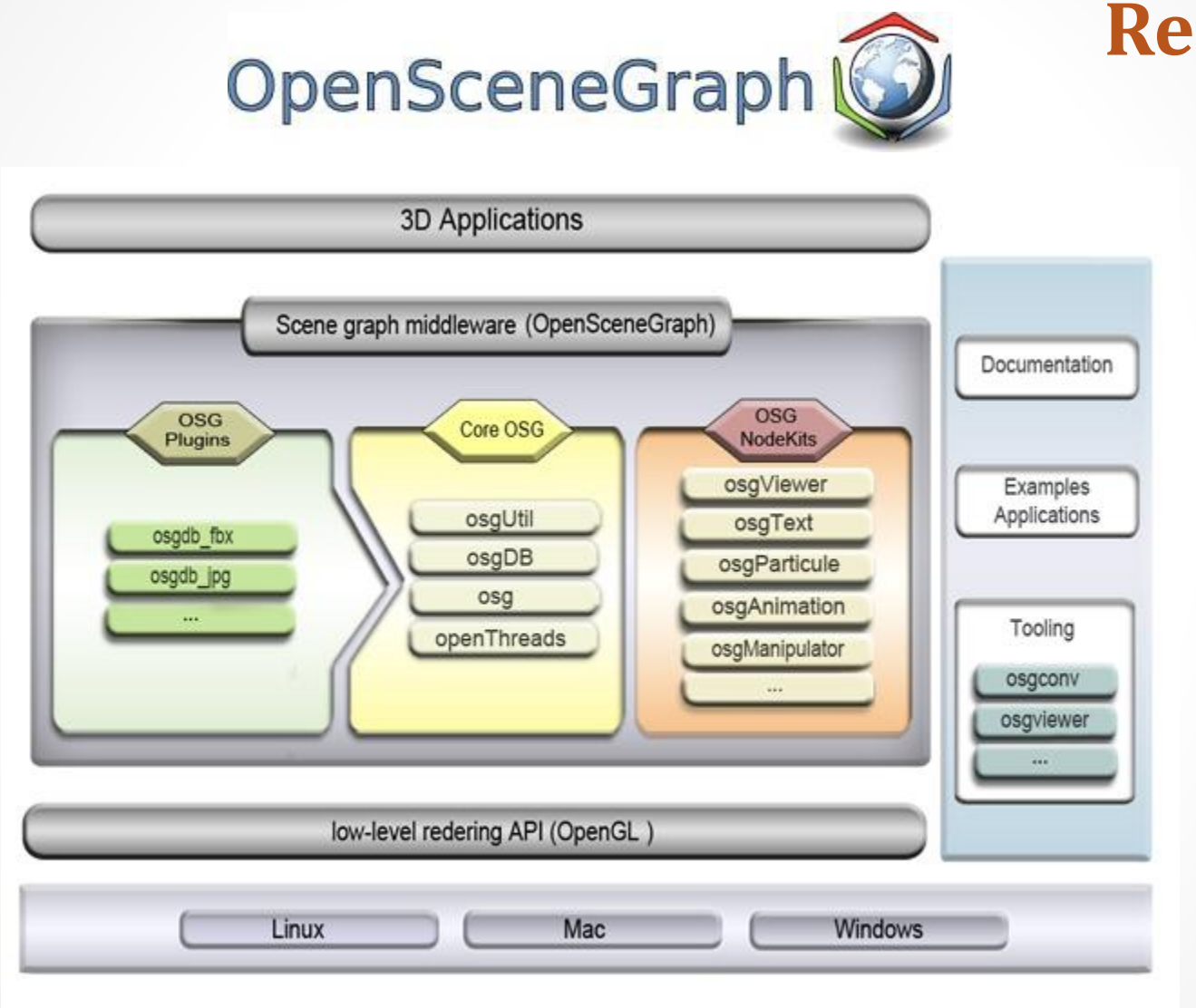
Animation is probably one of the most exciting feature of computer capabilities. OpenGL provides double-buffering support, which enables a smooth and fast transition between frames.

Rendering

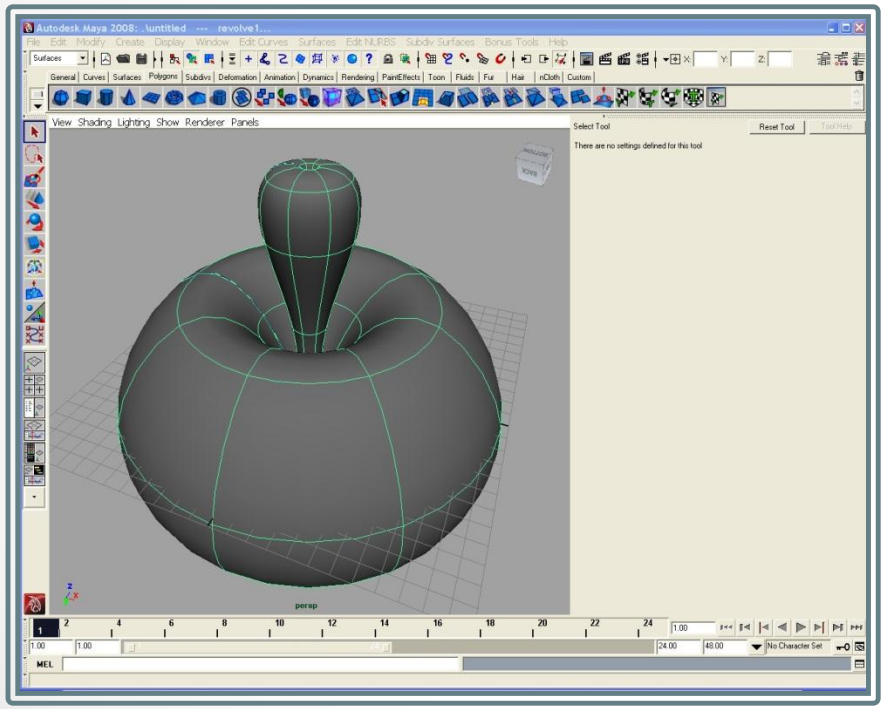
OpenSceneGraph (OSG) is a well-established open-source scene graph toolkit supporting high-performance 3D graphics, used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modeling.

Written entirely in Standard C++ and OpenGL, it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris, HP-Ux, AIX and FreeBSD operating systems.

Rendering


OpenSceneGraph
3D Applications
Scene graph middleware (OpenSceneGraph)
OSG Plugins
osgdb_fbx
osgdb_jpg
...
Core OSG
osgUtil
osgDB
osg
openThreads
OSG NodeKits
osgViewer
osgText
osgParticule
osgAnimation
osgManipulator
...
Documentation
Examples Applications
Tooling
osgconv
osgviewer
...
low-level redering API (OpenGL)
Linux
Mac
Windows

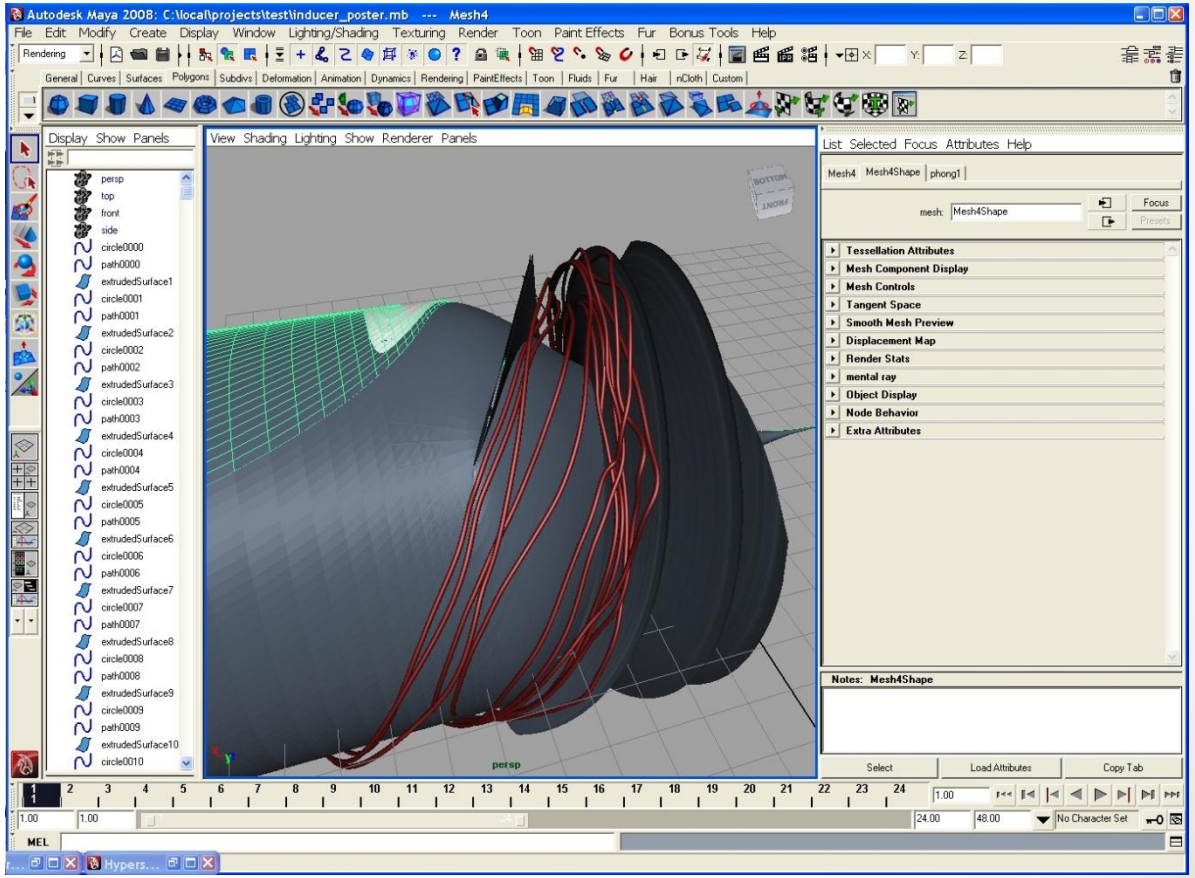
Rendering



Maya is an application used to generate 3D images for use in film, television, game development and architecture.

Alongside its more recognized visual workflow, Maya is equipped with a cross-platform scripting language, called Maya Embedded Language. MEL, is provided for scripting and a means to customize the core functionality of the software

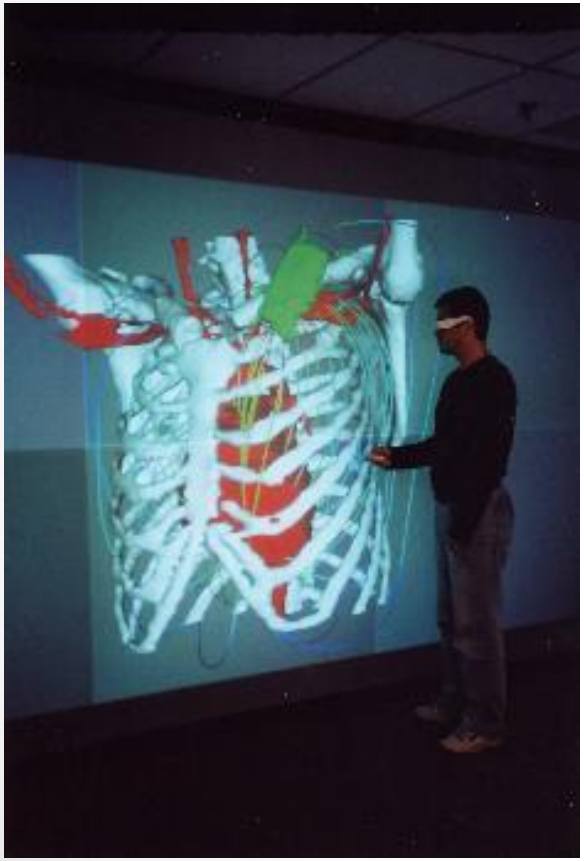
Rendering



Interactive Streakline Rendering in a Rocket Engine Turbopump Inducer.

Douglas L. Sondak and Erik Brisson – BU
Daniel J. Dorney – NASA Marshall Space Flight Center

BU Deep Vision Display Wall



SCV Tutorials

- Introduction to Scientific Visualization
- Scientific Visualization Using ParaView
- Scientific Visualization Using VTK
- Scientific Visualization Using MATLAB
- Graphics Programming in C/C++: OpenGL
- OpenSceneGraph
- Introduction to Maya
- Graphics and Images for Publication and Presentation

Contact me: Katia Oleinik: koleinik@bu.edu

Tutorial presentations and examples online:
www.bu.edu/tech/research/training/presentations/list/

Online evaluation:
http://scv.bu.edu/survey/tutorial_evaluation.html

Resources

- BU Scientific Computing and Visualization:
  http://www.bu.edu/tech/research/scv/

- Matlab: http://www.mathworks.com/products/matlab/
- VTK: http://www.vtk.org/
- Paraview: http://www.paraview.org/
- OpenGL: http://www.opengl.org/
- OpenSceneGraph: http://www.openscenegraph.org/
- Maya : http://usa.autodesk.com/maya/