Remote Visualization

How to make pretty pictures *for science!*

Aaron Knoll
Research Scientist
SCI Institute, University of Utah

"The purpose of computing is insight not numbers."

-- R. W. Hamming (1961)

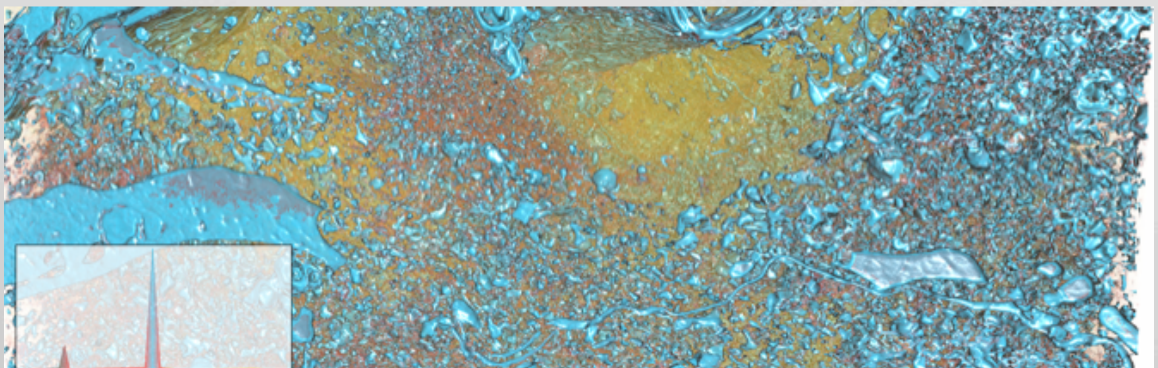
Roadmap

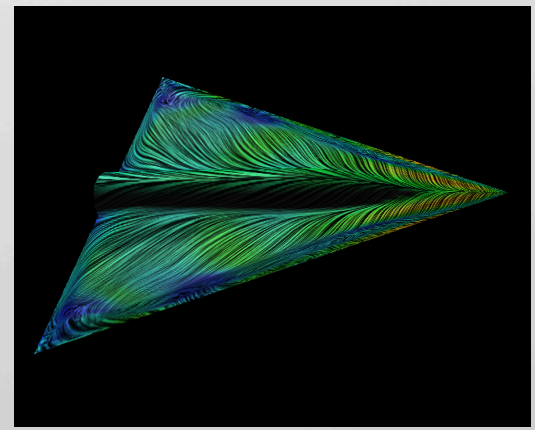
- What is visualization
- Why do it?
- How do we do visualization?
- Visualization software
- Remote and Parallel visualization

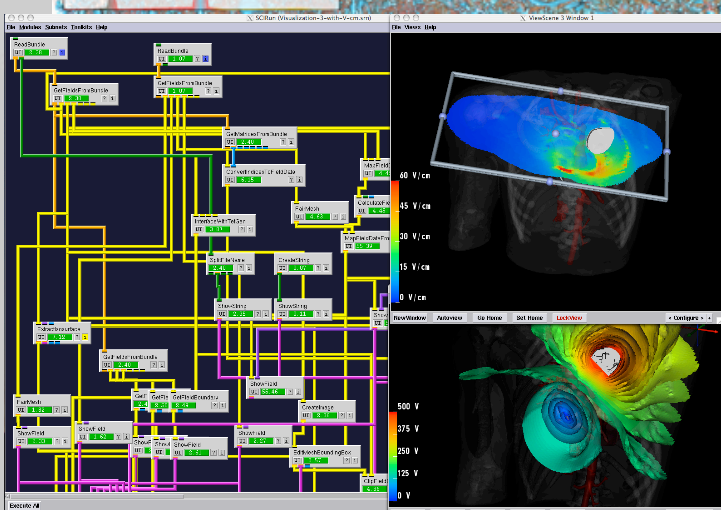
What is visualization?

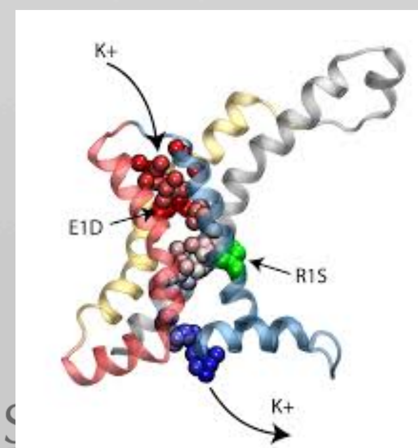
Two kinds of visualization

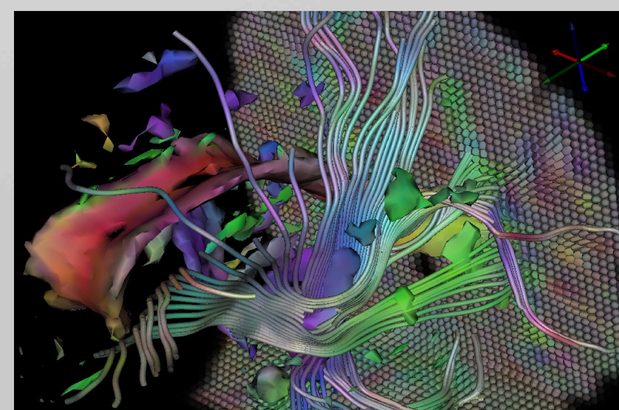
1. Scientific visualization
   - Data have spatial context
   - Map data features to colors in a way that makes visual sense
     - F(space, time) ➔ RGBA
   - *i.e., "rendering stuff"*











Two kinds of visualization

2. Information visualization
   - Data have no spatial context (or, space is secondary)
   - Illustrate relationships between diverse attributes
   - *i.e., "plots, graphs and diagrams"*

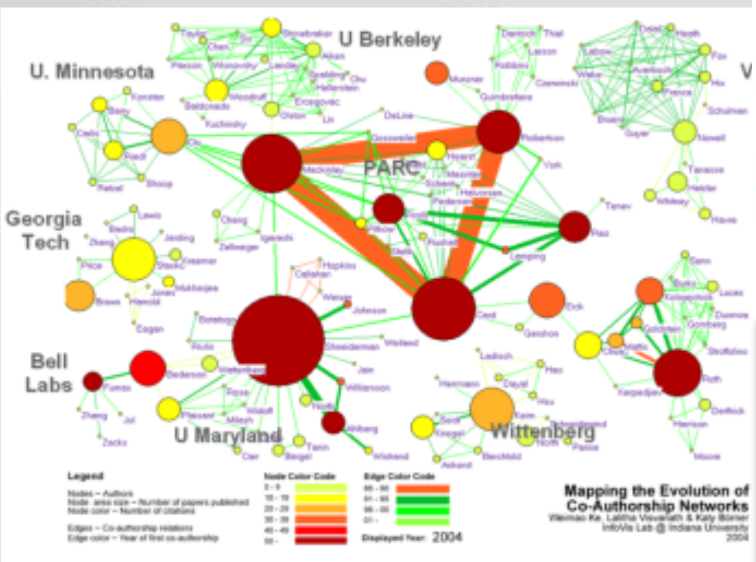
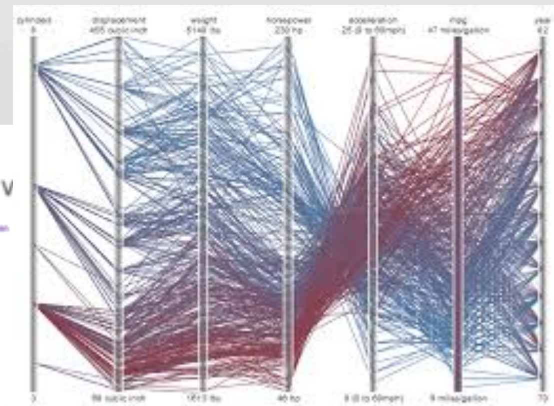
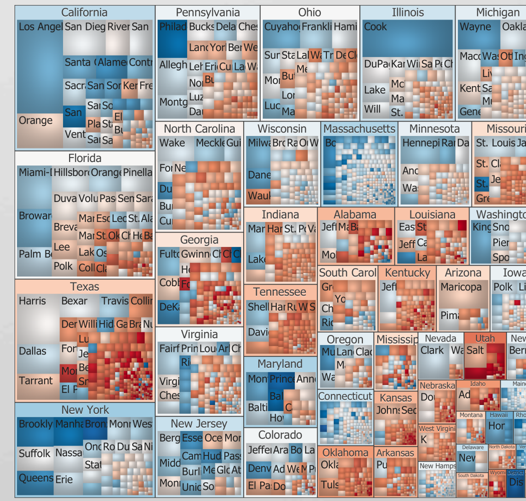
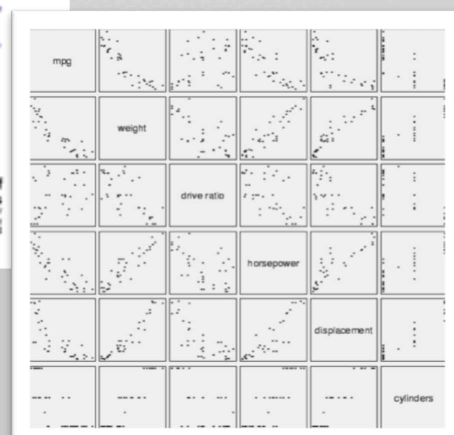


Image: Katy Borner (IU Bloomington)



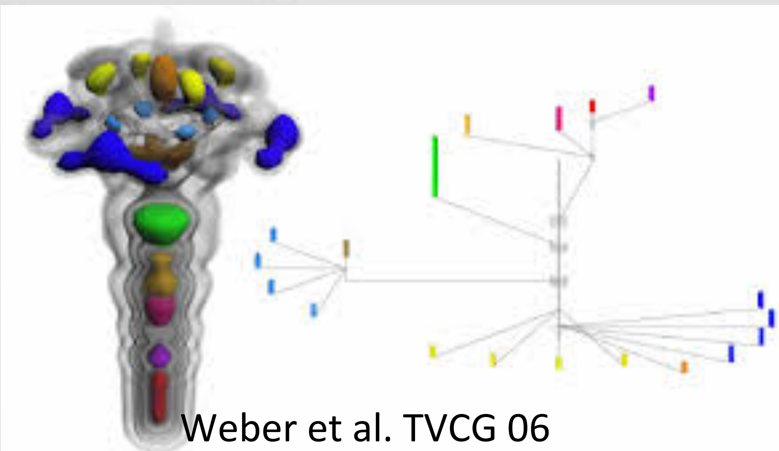




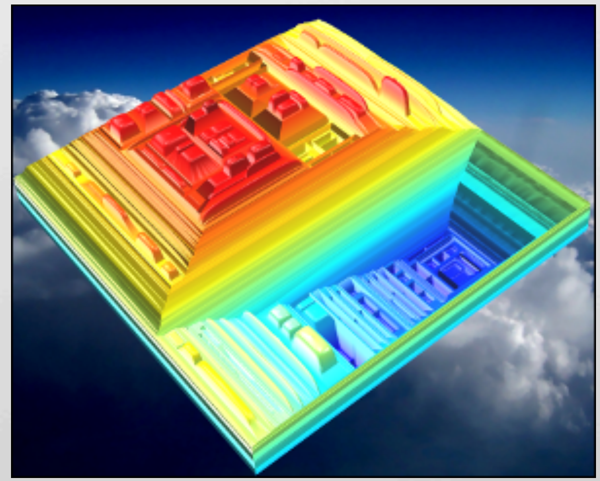
Meyer et al. Vis 2009

Two kinds of visualization

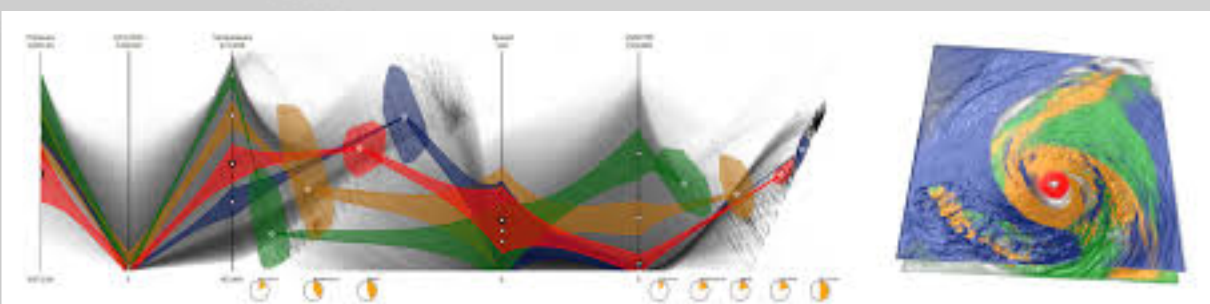
- There is no "line in the sand" between InfoVis and SciVis.
- Today we'll mostly talk about SciVis...



Weber et al. TVCG 06



Harvey and Wang, Eurovis 2010



Guo et al. Pacific Vis 2011

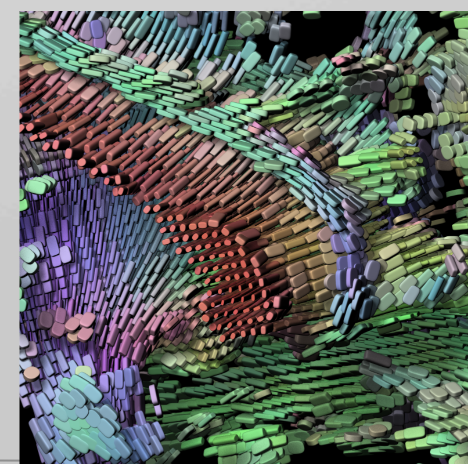


Image: Gordon Kindlmann, University of Chicago

Why do visualization?

Why do visualization?

- Illustration

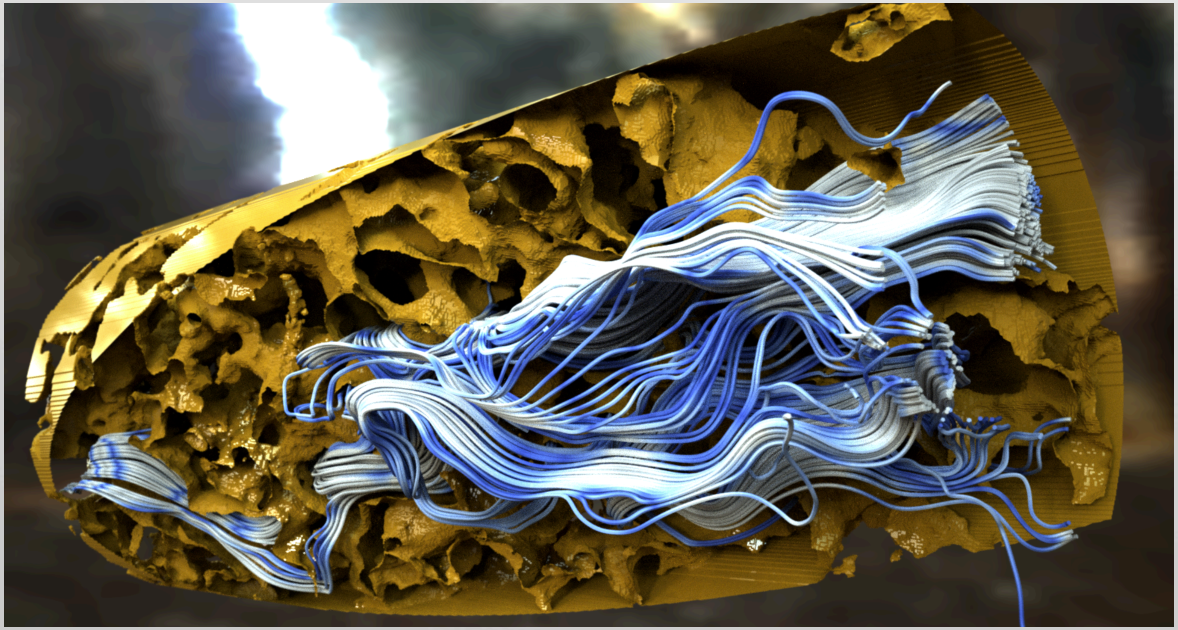
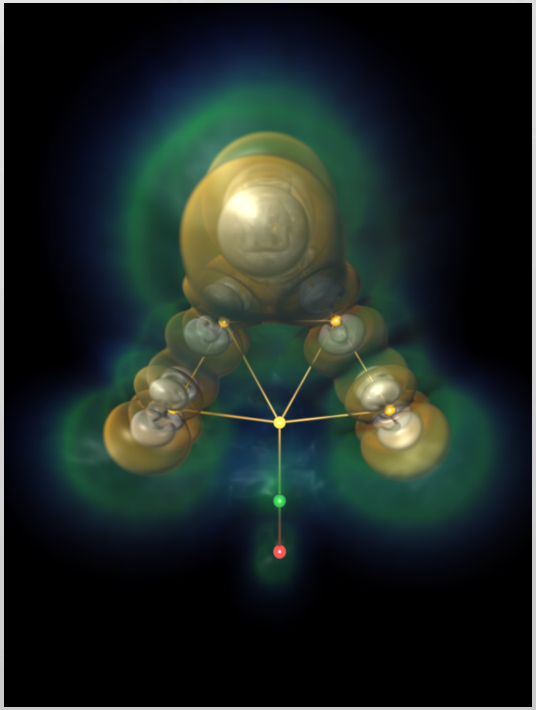


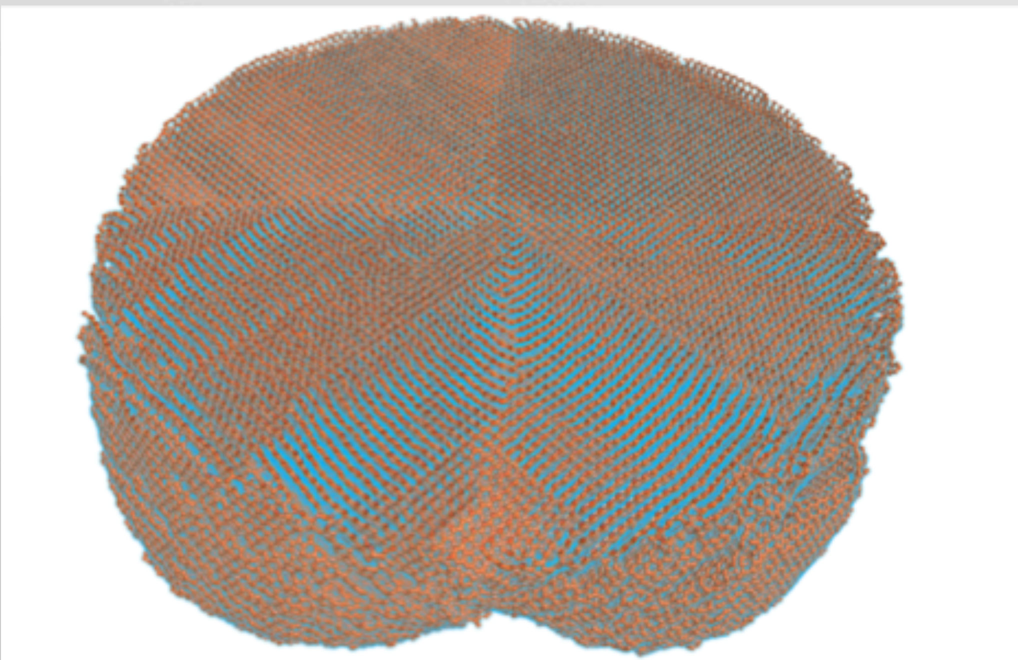
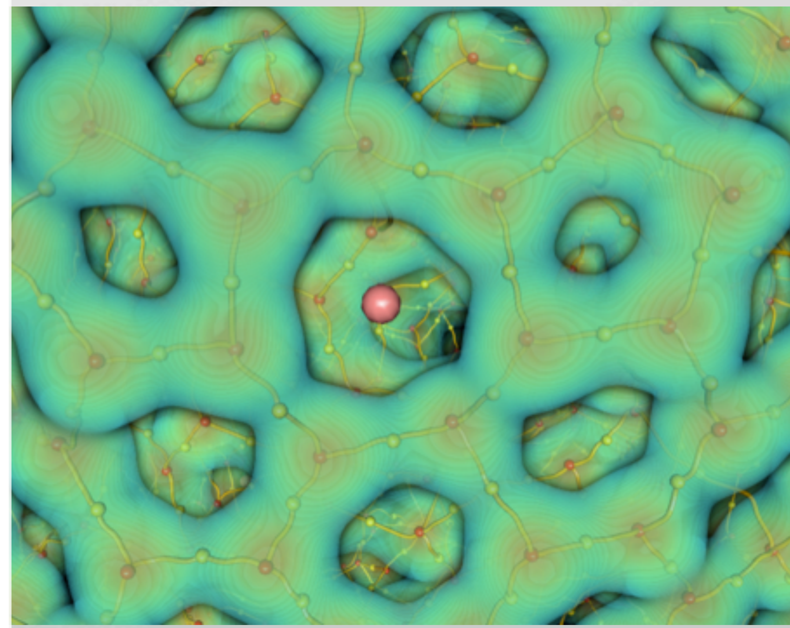
Image: Carson Brownlee (TACC)
Data: Sade Garcia, Michael Sukop (FIU)



Data: Aslihan Sumer (ARI), Julius Jellinek (ANL)

Why do visualization?

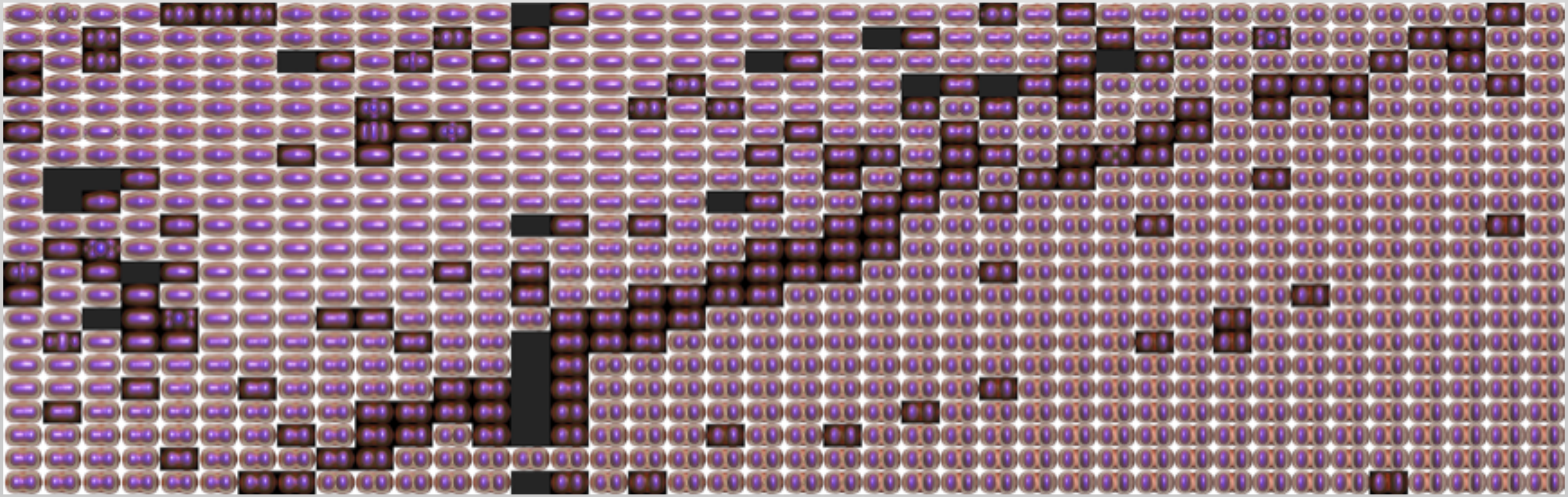
- Illustration
- Analysis

Images: Aaron Knoll and Attila Gyulassy (SCI), Data: KC Lau (Argonne National Lab)

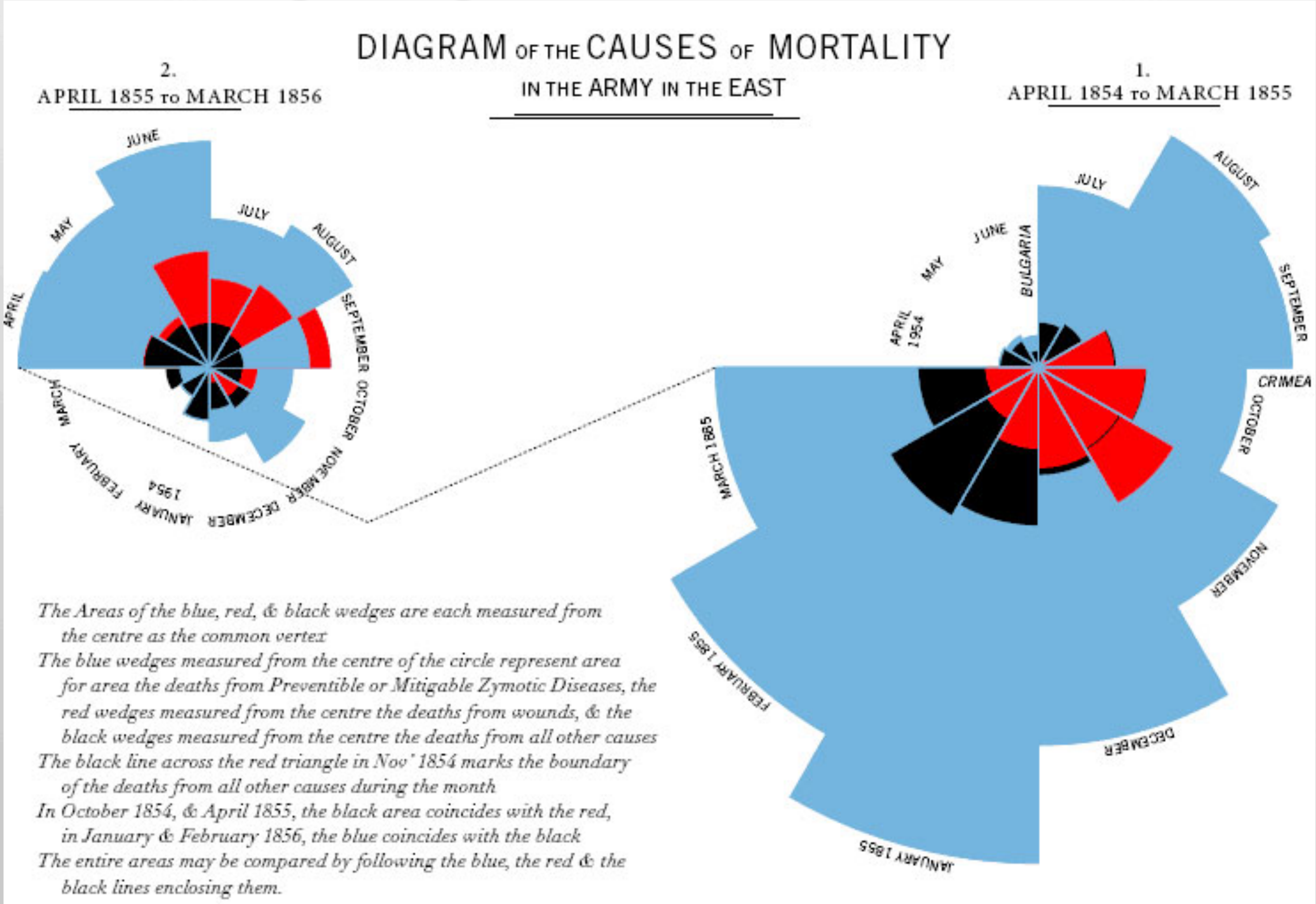
Why do visualization?

- Illustration
- Analysis
- Validation

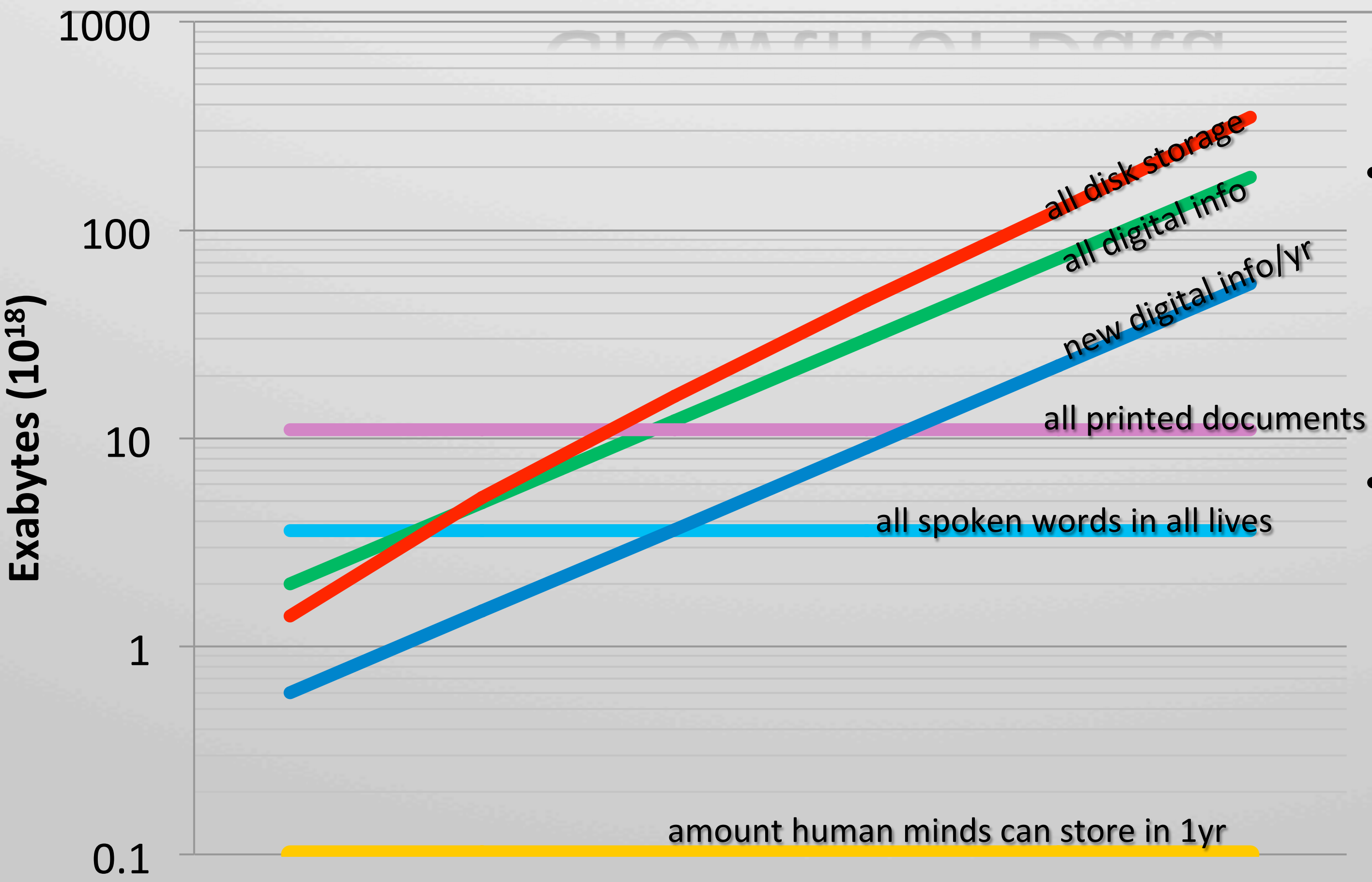


DFT computation - nuclear fission of Plutonium (Data: Nicolas Schunck, LLNL)

Florence Nightingale Cox Comb



Growth of Data



- As of February 2011, approximately 295 exabytes of data exist
- **Every two days we create as much data as we did from the beginning of mankind until 2003**

Sources: Lesk, Berkeley SIMS, Landauer, EMC, TechCrunch, Smart Planet

www.sci.utah.edu THE UNIVERSITY OF UTAH

How do we do visualization?

Scientific Visualization basics

Computational data

scalar/vector/tensor fields
structured, unstructured, atomistic

+

Computer graphics

Camera
Lighting model
Rendering algorithm

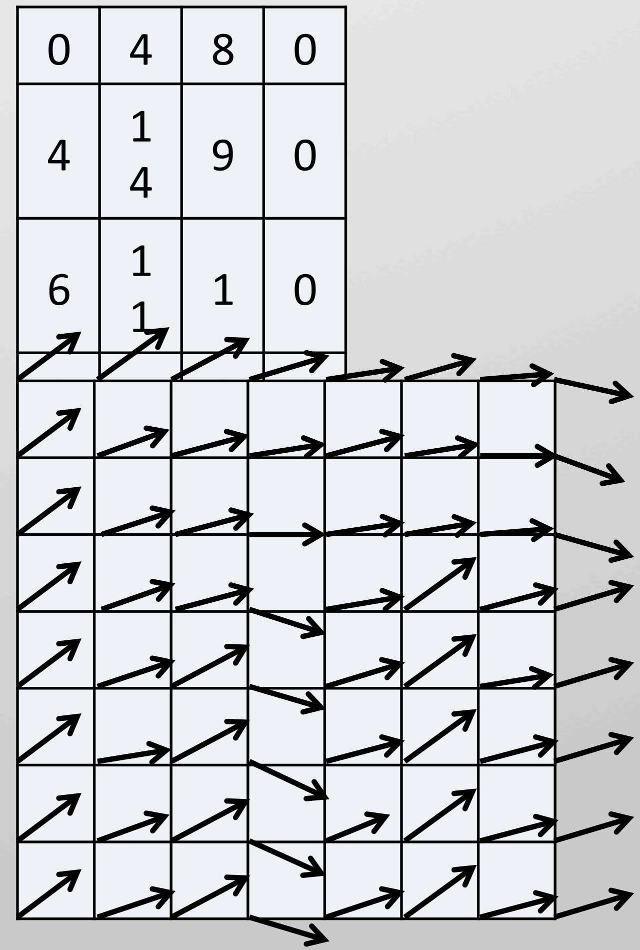
=

Visualization

Reader
Filters
Color maps
Rendering!

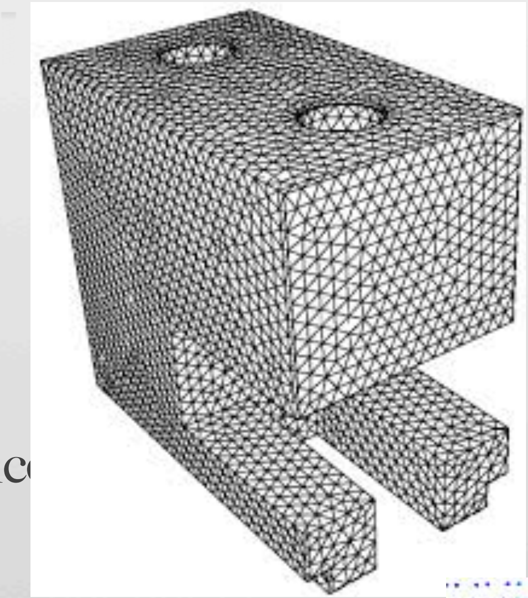
Computational Data

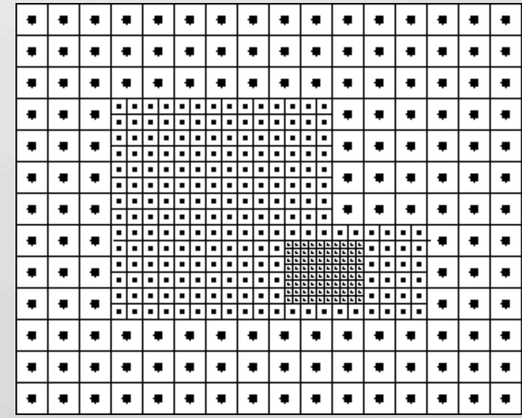
- Typically continuous fields
  - Scalar, vector, tensor
  - 2D, 3D, 2.5D

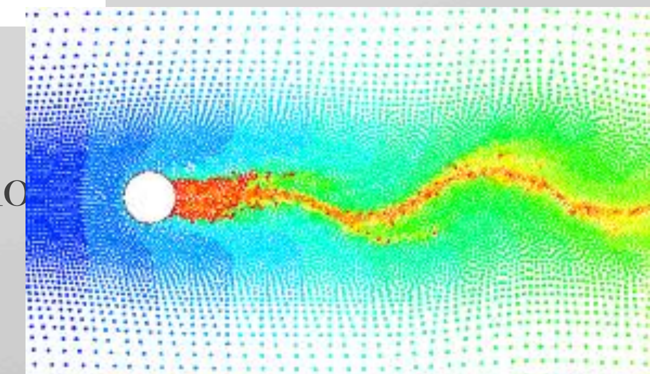


Computational Data

- Typically continuous fields
  - Scalar, vector, tensor
  - 2D, 3D, 2.5D
- Geometry (“topology”)
  - Structured (finite differenc
  - Unstructured
    - Tet,hex meshes (finite elements)
    - Particles -- n-body simulations, (molecular dynamics, cosmology, blood flo
    - AMR
  - Other (spectral / functional / wavelet)
    - DFT, FMM, DNS
    - Usually generate large structured data as postprocess

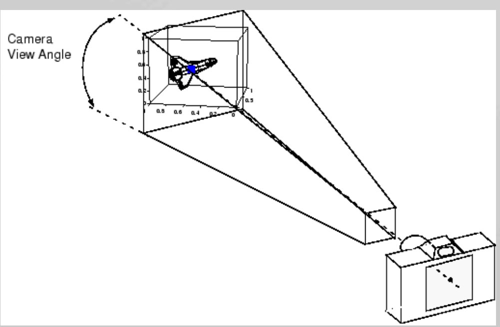


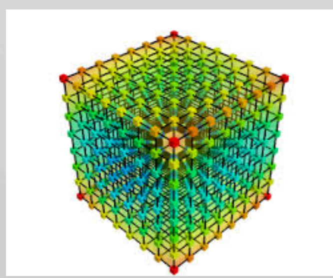


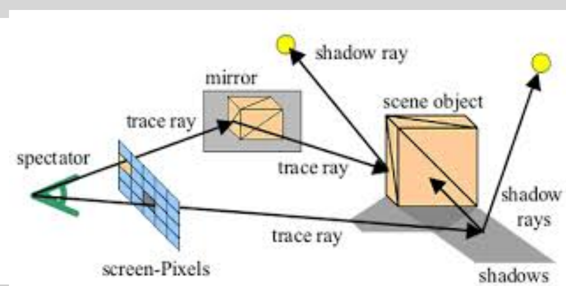


Computer graphics (ideally)

Camera (perspective, orthographic) + **Primitives** (points, lines, triangles, voxels, elements) + **Rendering** (Rasterization, ray tracing, splatting) = **Display** (pixels)







Computer graphics (reality)

Camera
(perspective, orthographic)

+

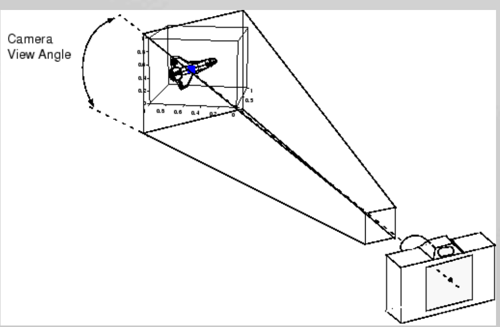
Primitives
(triangles)

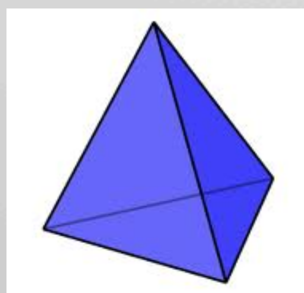
+

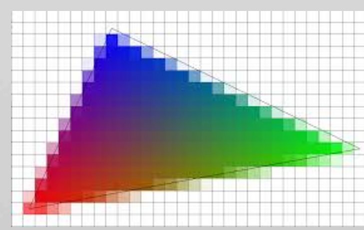
Rendering
(rasterization)

=

Display
(pixels)





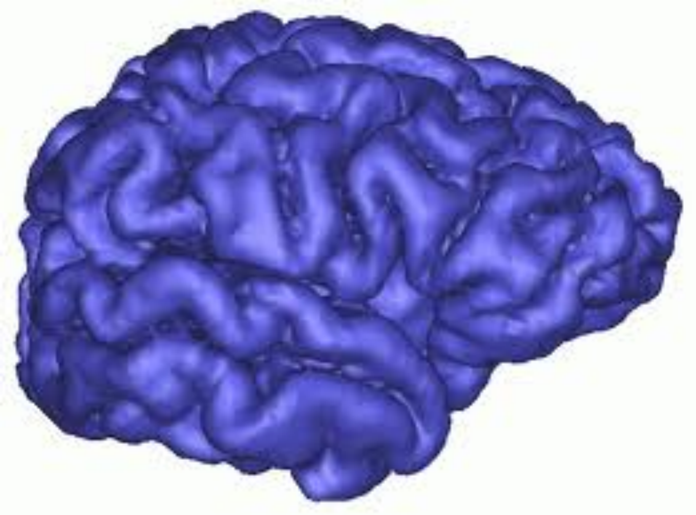


The visualization pipeline

- Convert (volume) vis data into renderable (triangle) primitives

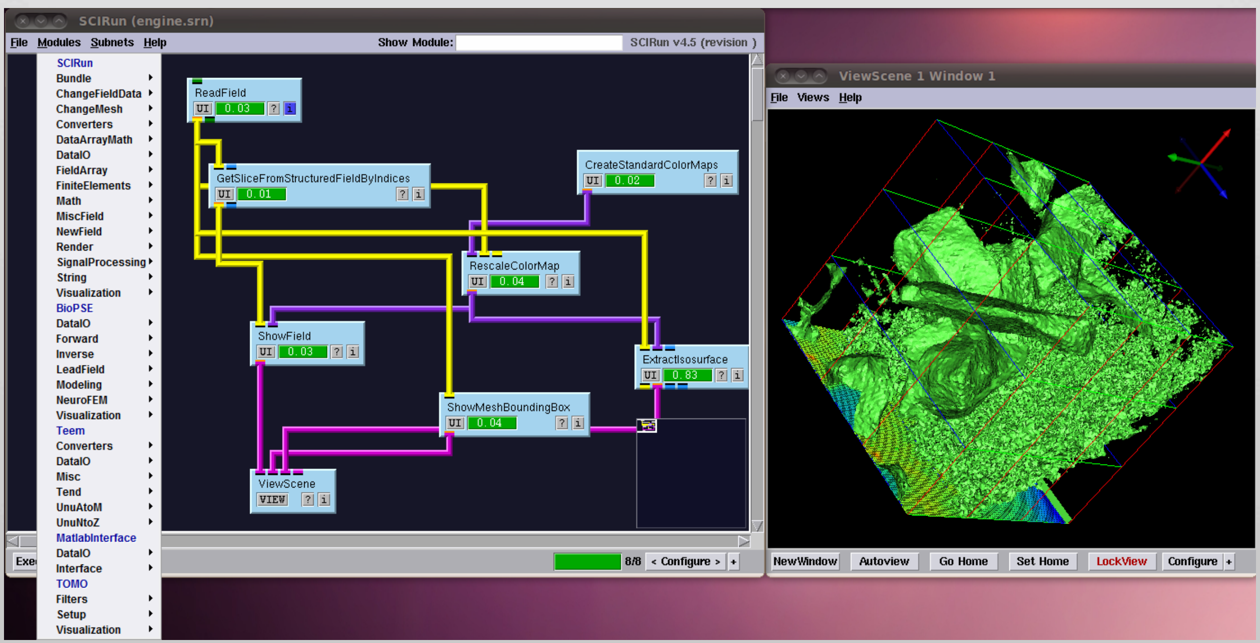
The visualization pipeline

- Convert (volume) vis data into renderable (triangle) primitives
- Simplify data to visualize what we want in the process.



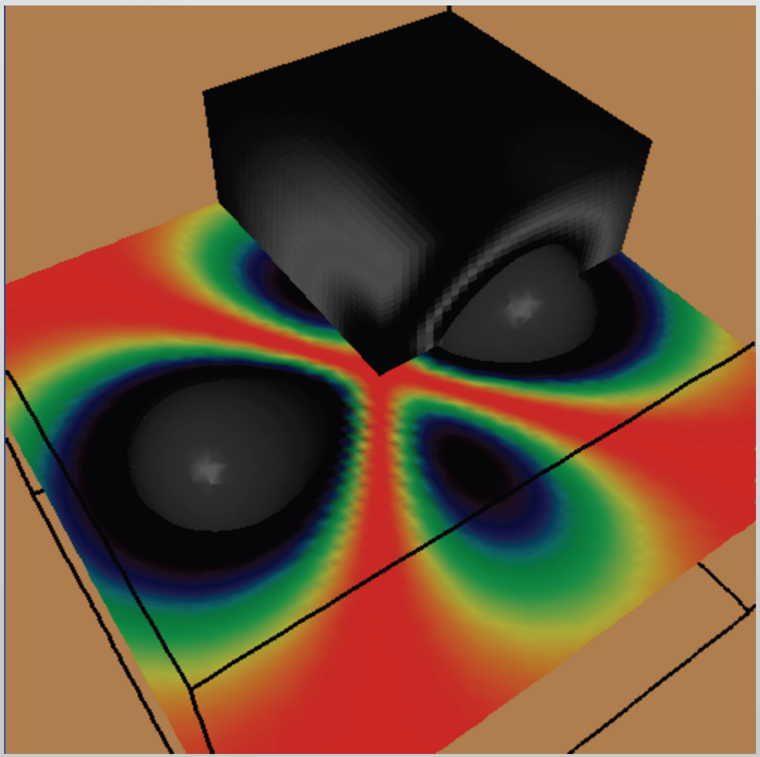
The visualization pipeline

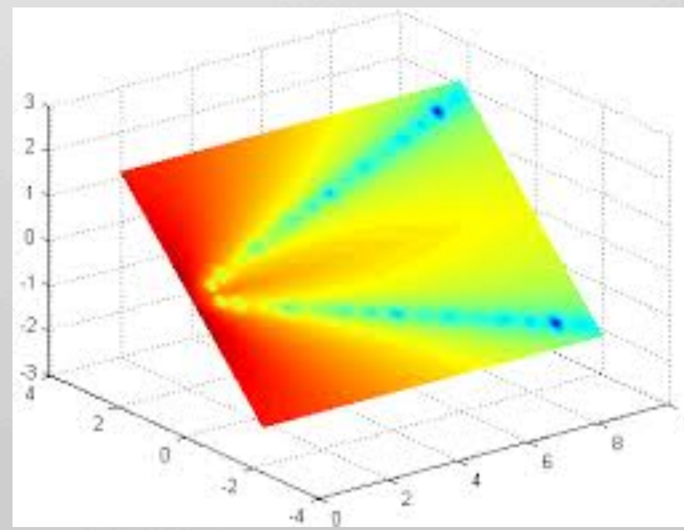
- Convert (volume) vis data into renderable (triangle) primitives
- Simplify data to visualize what we want in the process.
- Each step in the pipeline is a **filter**.



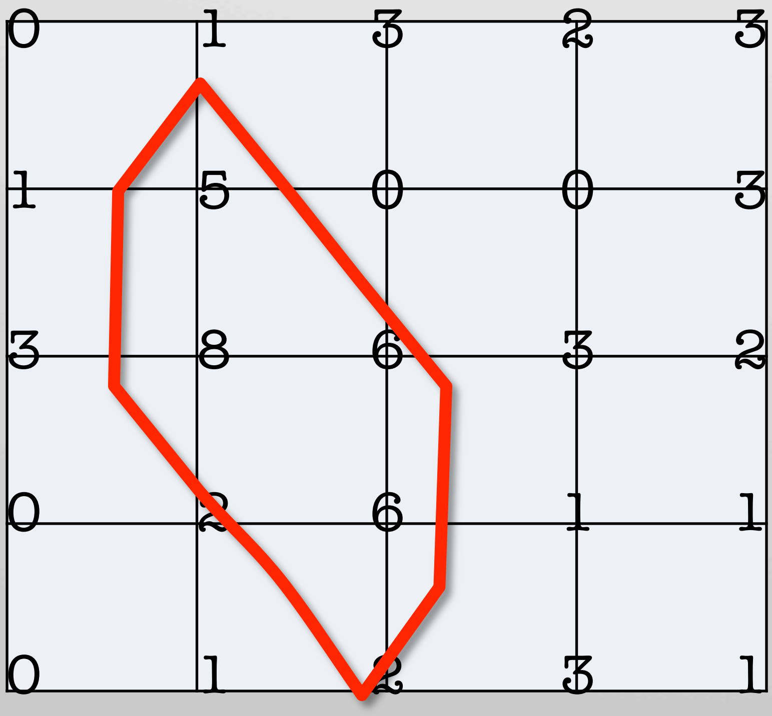
Scalar field visualization

Pseudocolor and cut planes.





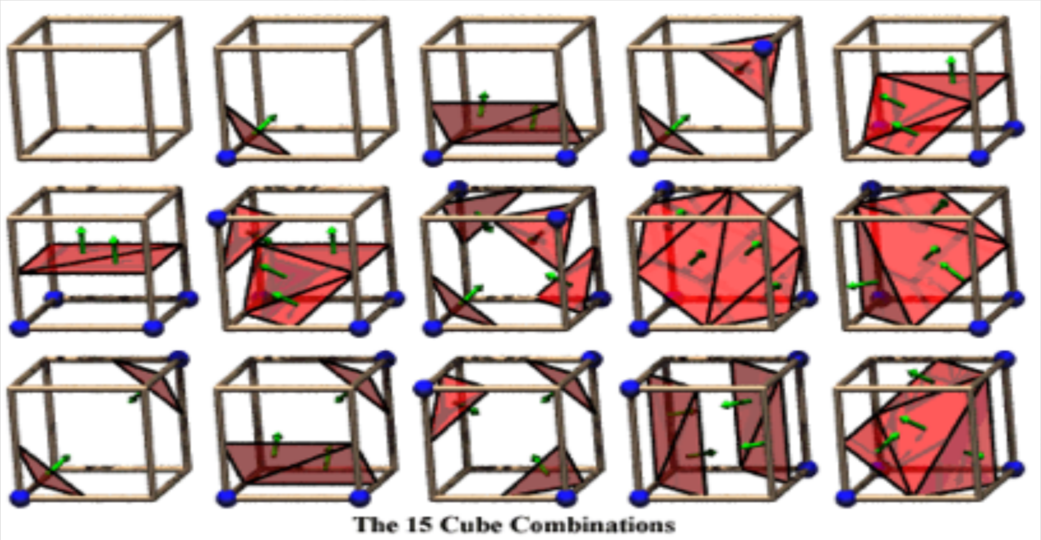
Isocontours

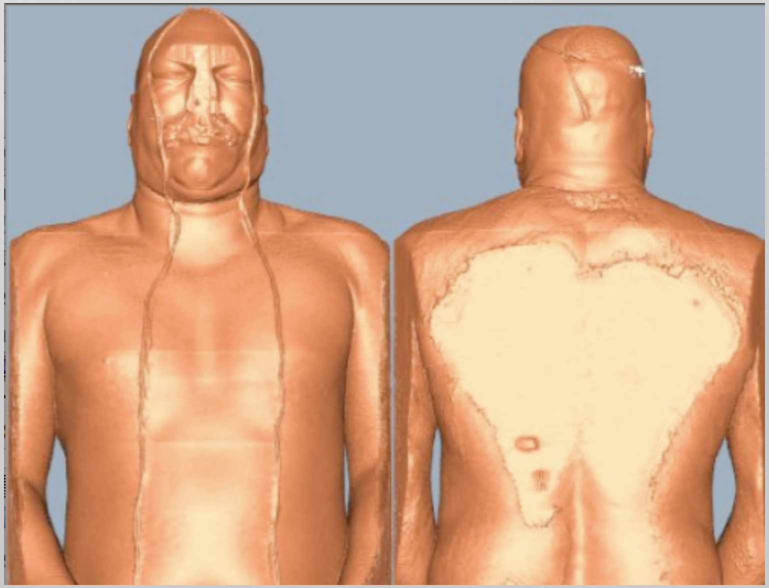


Isocontour for isovalue = 4?

Isosurfaces

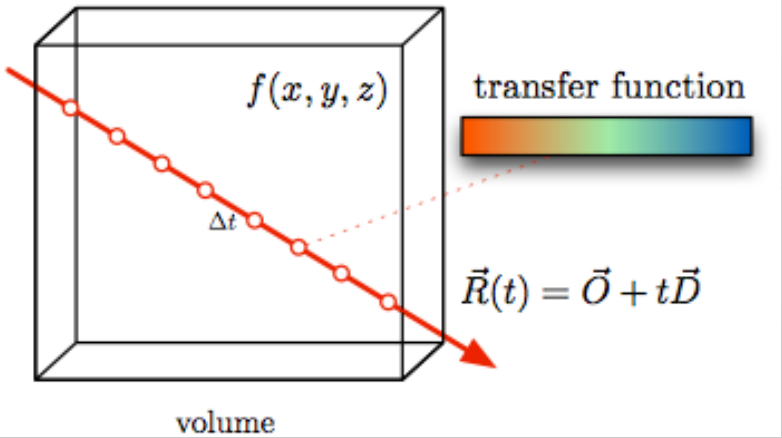
- An *isosurface* is an isocontour of a 3D volume.
- Marching cubes
  - in each voxel, extract the isosurface as triangles
  - Wyvill & McPeeters 86 (blobbies), Lorensen & Cline 87 (structured volumes)
  - cited over 10,000 times!





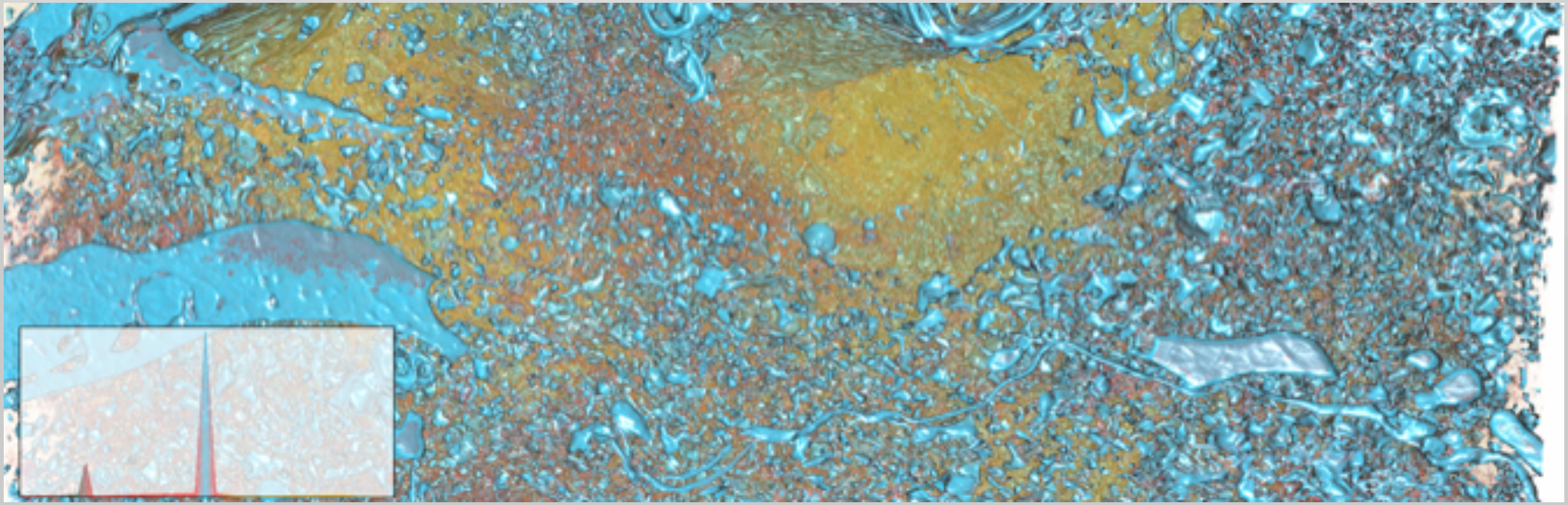
Volume rendering

- Alternative to isosurfacing.
- Less of a filter, more of a full rendering technique
  - Filtering is done through the *transfer function*.
- Can be implemented numerous ways (slicing, splatting, ray casting)



Volume rendering

- Alternative to isosurfacing.
- Less of a filter, more of a full rendering technique
  - Filtering is done through the *transfer function*.
- Can be implemented numerous ways (slicing, splatting, ray casting)
- Renders "all" isosurfaces at once, with control over opacity
  - Disadvantages: slow, hard to design transfer functions...



Flow vis

- So far, we have discussed scalar fields.
- What do we do for vector field data?

- Seed + advect particles:

$$P_{new} = P_{current} + V_P \Delta t$$

- **Streamlines** advect in space
- **Pathlines** advect in time

- Other variants:
  – Streaklines, timelines
  – Stream/streak surfaces

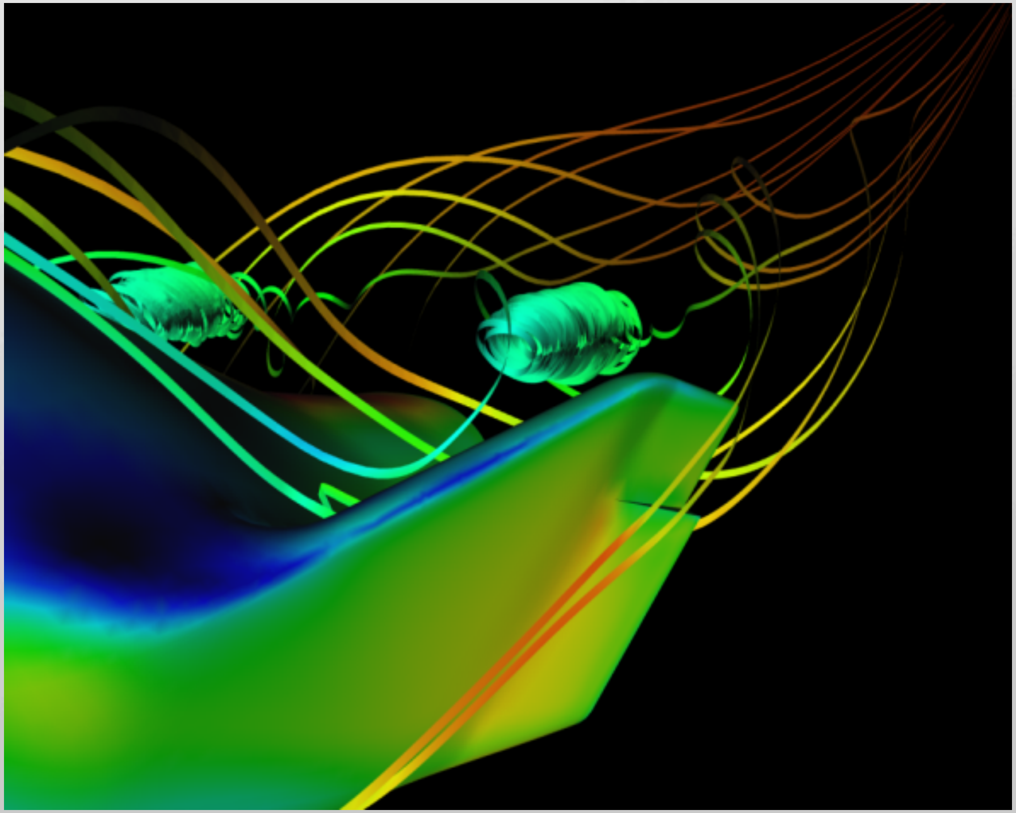
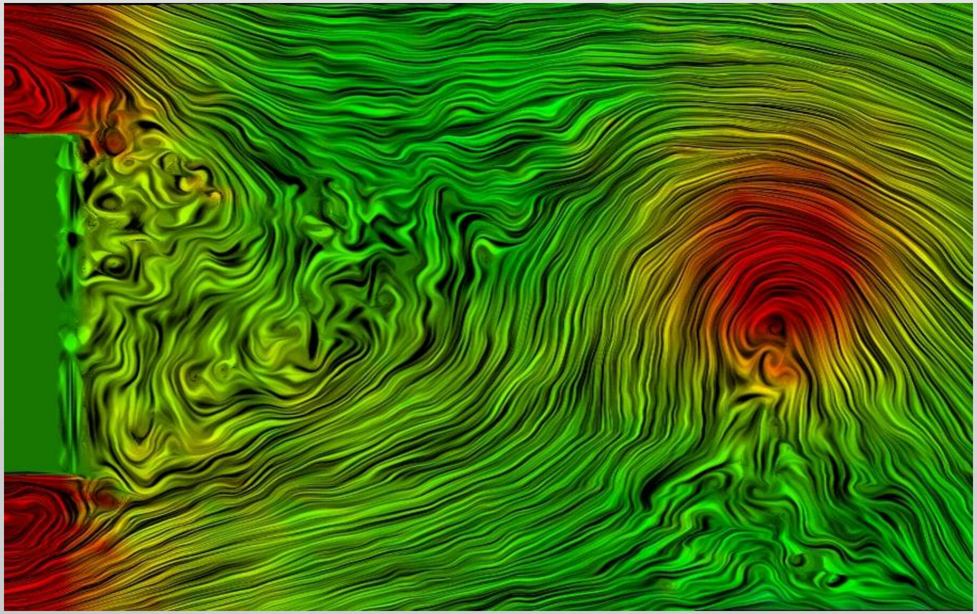
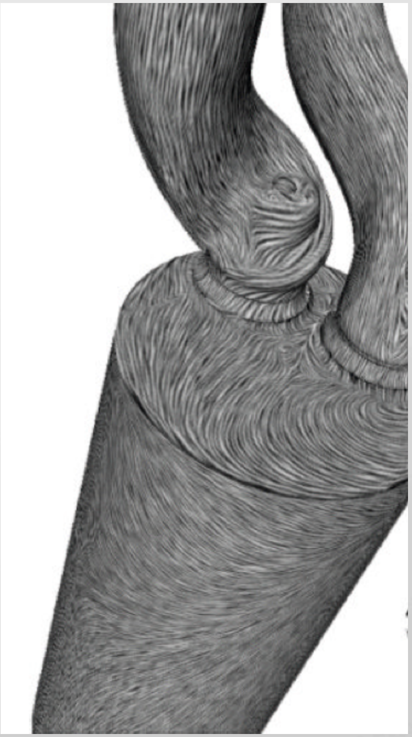


Image: Kelly Gaither, TACC

Flow vis – texture methods (2D)

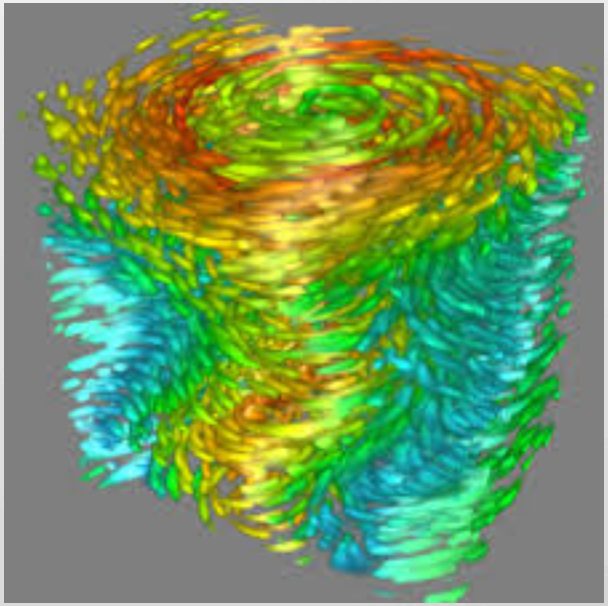


Spot noise (image: Wim De Leeuw, CWI)



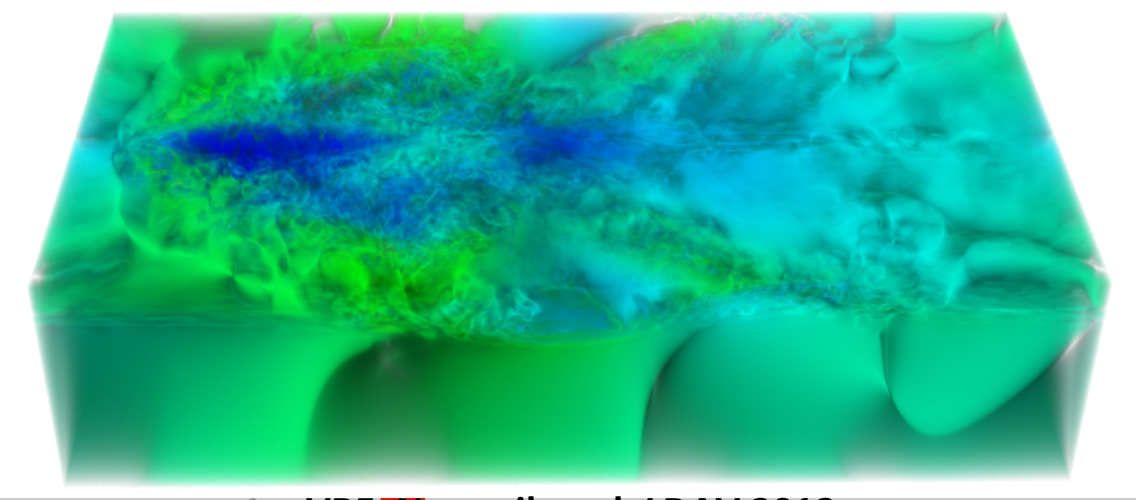
LIC (image: Guo Shi Li, SCI)

Flow vis – volumetric methods



3D LIC, Li et al. Vis 2004

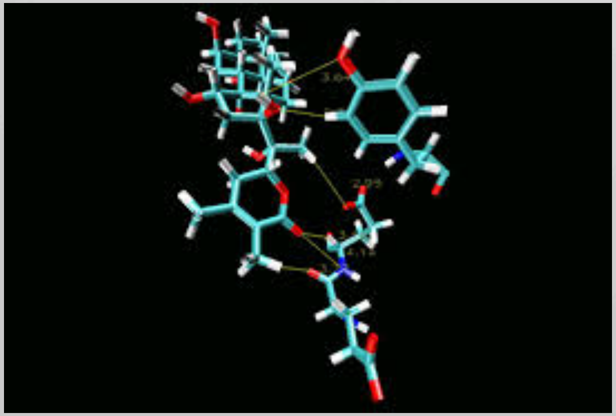
FTLE, Garth et al. Vis 06



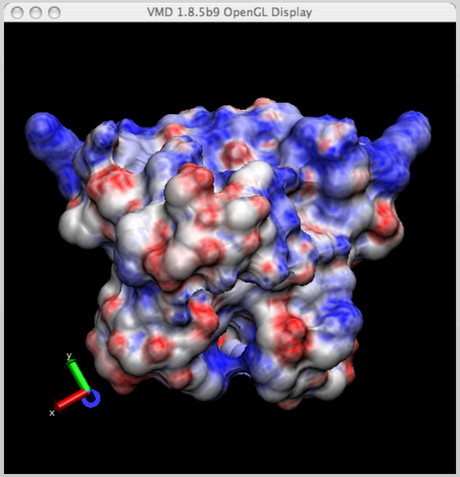
VRF, Navratil et al. LDAV 2012

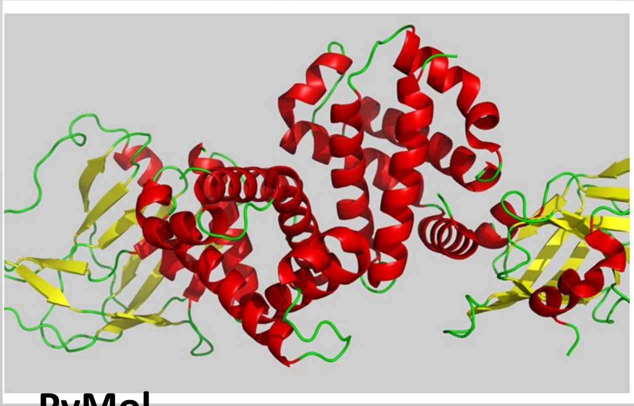
Molecular visualization

- Largely separate techniques, tools from the rest of sci vis
  - VMD, Chimera, PyMol, Avogadro, Megamol
- Glyphs
- Molecular surfaces
  - Van der Waals, CPK
  - Connolly surfaces, SES, SAS
- Ribbons
  - Primary, secondary, etc. protein structures
- Some need for volume and vector field visualization



VMD





PyMol

www.sci.utah.edu

THE UNIVERSITY OF UTAH

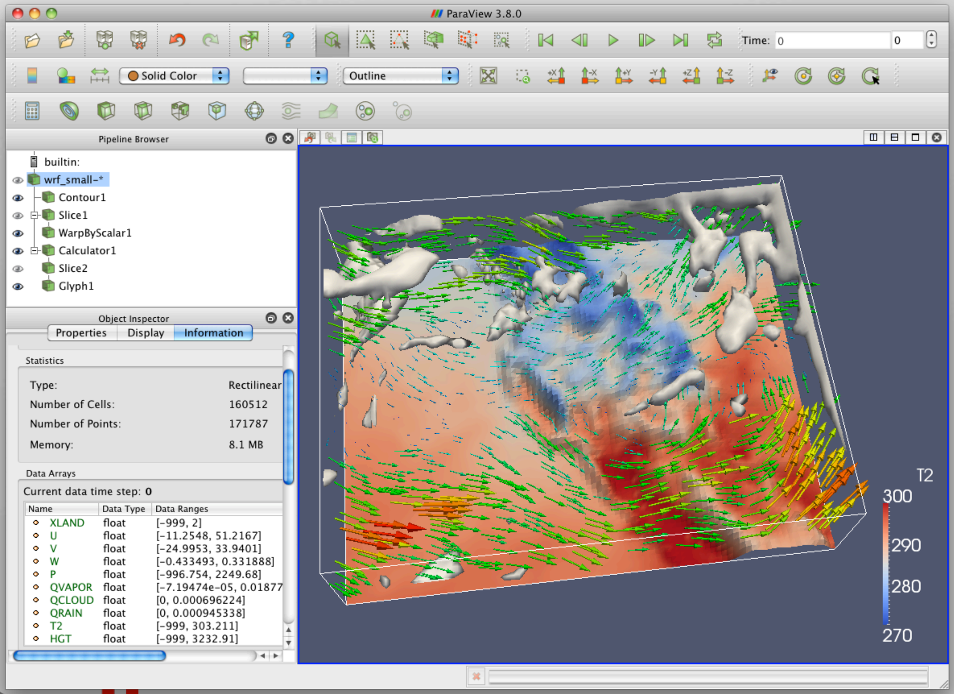
Visualization software

VTK

- http://www.vtk.org
- Open source, developed by Kitware
- The standard bearer for scientific visualization
  - Marching cubes, streamlines, cut/clip planes, etc.
  - Hundreds of filters for data resampling, geometry processing, analysis
- VTK data model
  - Supports any structured/unstructured data, fully-featured C++/templates data model
- Call as a library from C++, Java, Python, Tcl/Tk
- What VTK does not do:
  - Full-featured UI
  - Exploit latest OpenGL 4.X features
  - Use GPUs for filter computation (VTK-M initiative)
  - Parallel rendering

ParaView

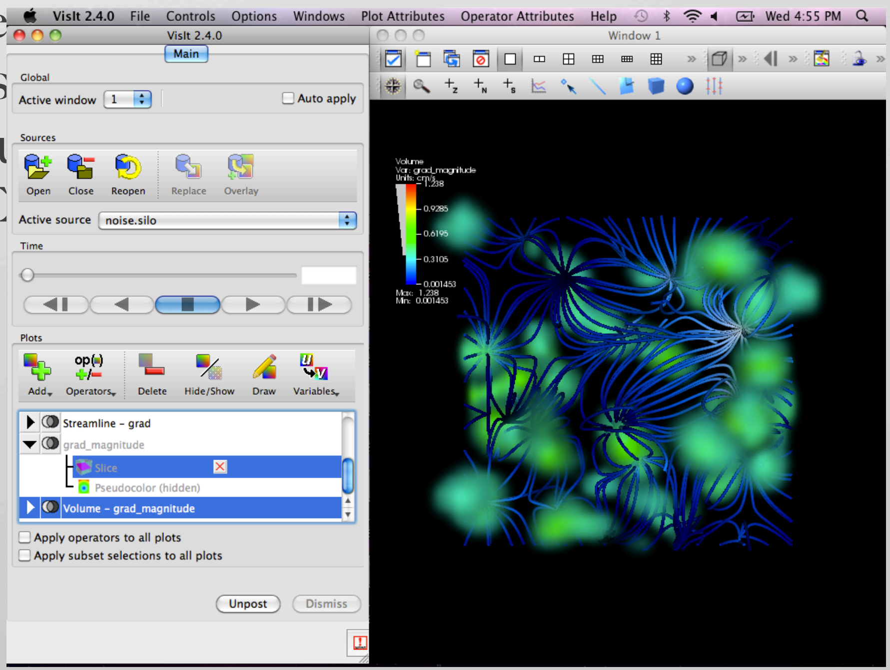
- http://www.paraview.org/
- Open source, built on top of VTK
- Maintained by Kitware, Sandia, LANL, ARL
- Provides data-parallel reading, filtering and rendering
- Qt UI, tree hierarchy of filters, camera controls, etc.
- Many, many parallel readers
- Reasonably good volume rendering



* Image courtesy of Paul Navratil and Kelly Gaither (TACC)

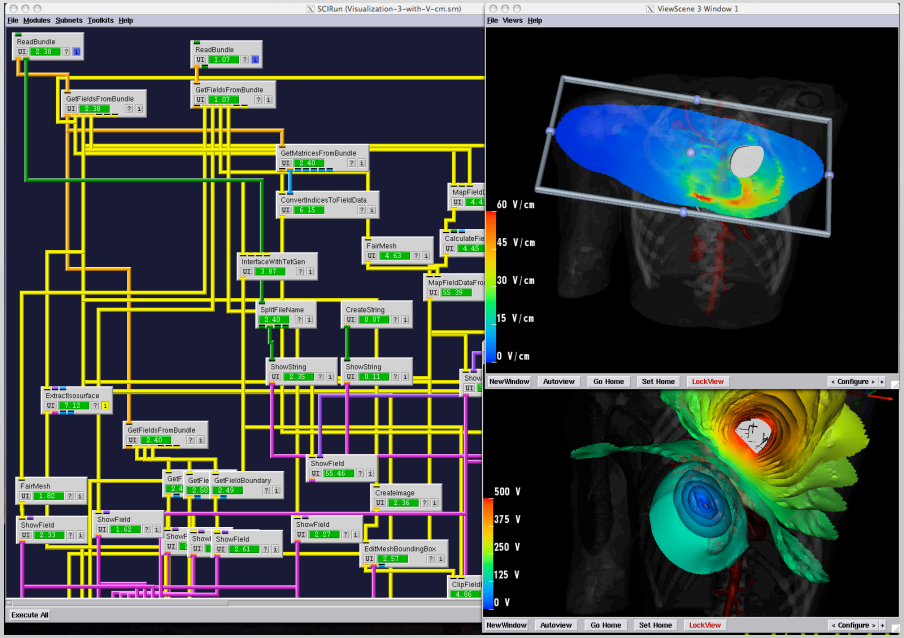
VisIt

- http://visit.llnl.gov
- Contributors at LLNL, LBNL and many others
- Also built on top of VTK
- Also provides parallel reading, filtering and rendering!
- Additional "AVT" layer
  - Capabilities outside of the VTK pipeline
- Filters replaced with plots and operators
- Huge array of flow visualization techniques
  - Fast streamlines, particle tracing, FTLE
- Also, many different readers
  - Enzo AMR, other data



SCIRun

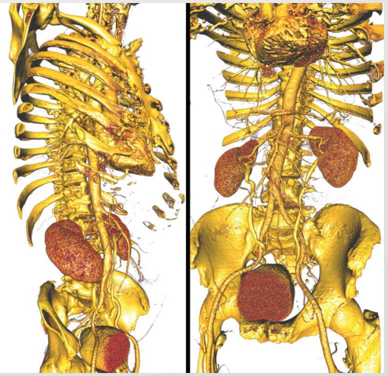
- http://www.sci.utah.edu/cibc-software/scirun.html
- Open source, maintained by the University of Utah
- Designed for efficient operation on multithreaded workstations (SMP)
- "Computational workbench": a single pipeline for both simulation and visualization
  - Bioelectric field FE simulation codes
  - Computer aided surgery
- Unique visualization modalities
  - Volume rendering with 2D transfer functions



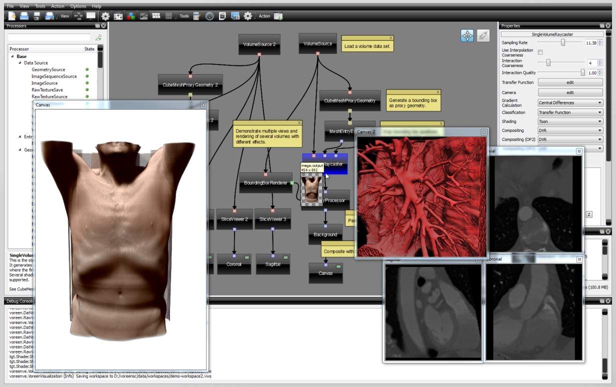
Visus

- http://www.pascucci.org/visus/
- Maintained by Valerio Pascucci, SCI Institute, University of Utah
- Streaming visualization framework for remote and in situ visualization
  - IDX file format for efficient data processing and rendering.
  - Fast out-of-core queries and analysis
- 2D and 3D gigapixel imagery, large structured data, AMR

Volume renderers



- ImageVis3D
  - http://www.sci.utah.edu/cibc-software/imagevis3d
  - Maintained by SCI and TU Duisberg
  - Fast, out-of-core GPU volume renderer with Qt UI, transfer function editors
- Voreen
  - http://www.voreen.org
  - University of Munster and Linkoping University
  - Supports complicated workflows, variety of rendering and shading methods



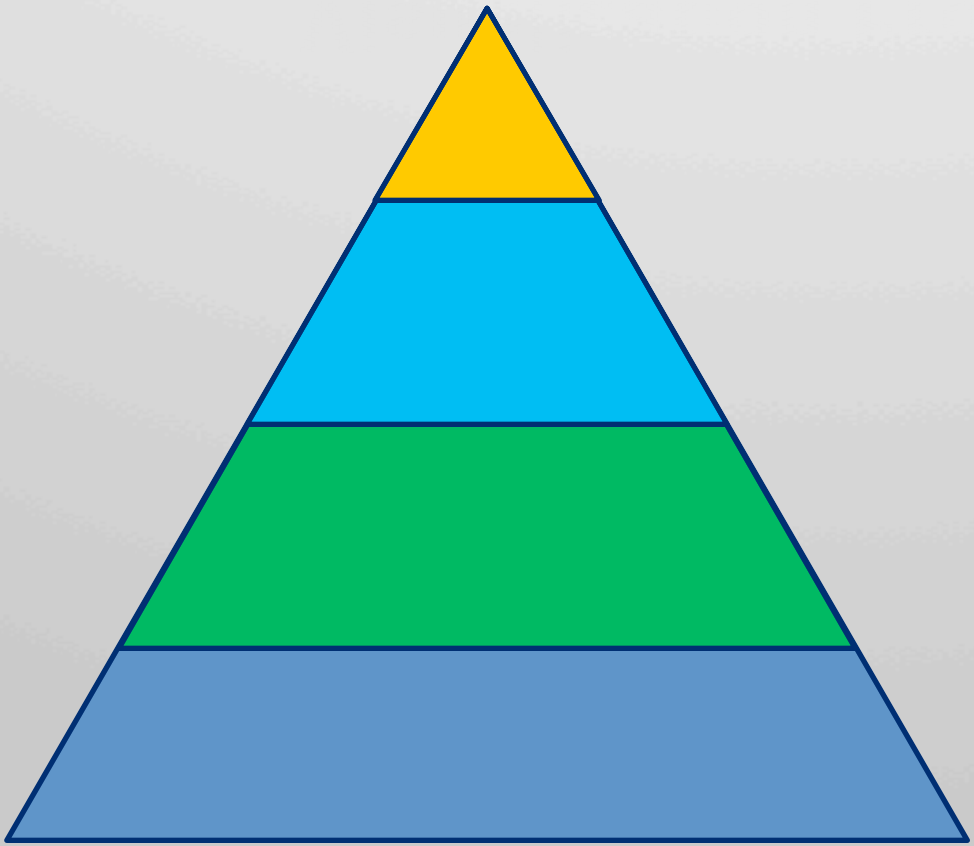
- Exposure renderer
  - https://code.google.com/p/exposure-render/
  - CUDA-based photorealistic volumetric path tracing
- These only support structured volume data.

Many, many other vis packages

- Molecular (VMD, PyMol, Avogadro, Chimera, Megamol)
- Rendering frameworks with vis capabilities
  - Intel Embree
  - NVIDIA Optix
- Info Vis / statistics
  - Microsoft Excel
  - Google charts
  - Matplotlib (python)
  - gnuplot
  - R
  - Matlab / Octave
  - Mathematica
  - IVTK

Remote and Parallel Visualization

Visualization Problem Size



Huge problems
- Data cannot be moved off supercomputer
- “true in situ”

Large problems:
- Data are impractical to move, already in a parallel container format
- Parallel vis cluster, ParaView, VisIt

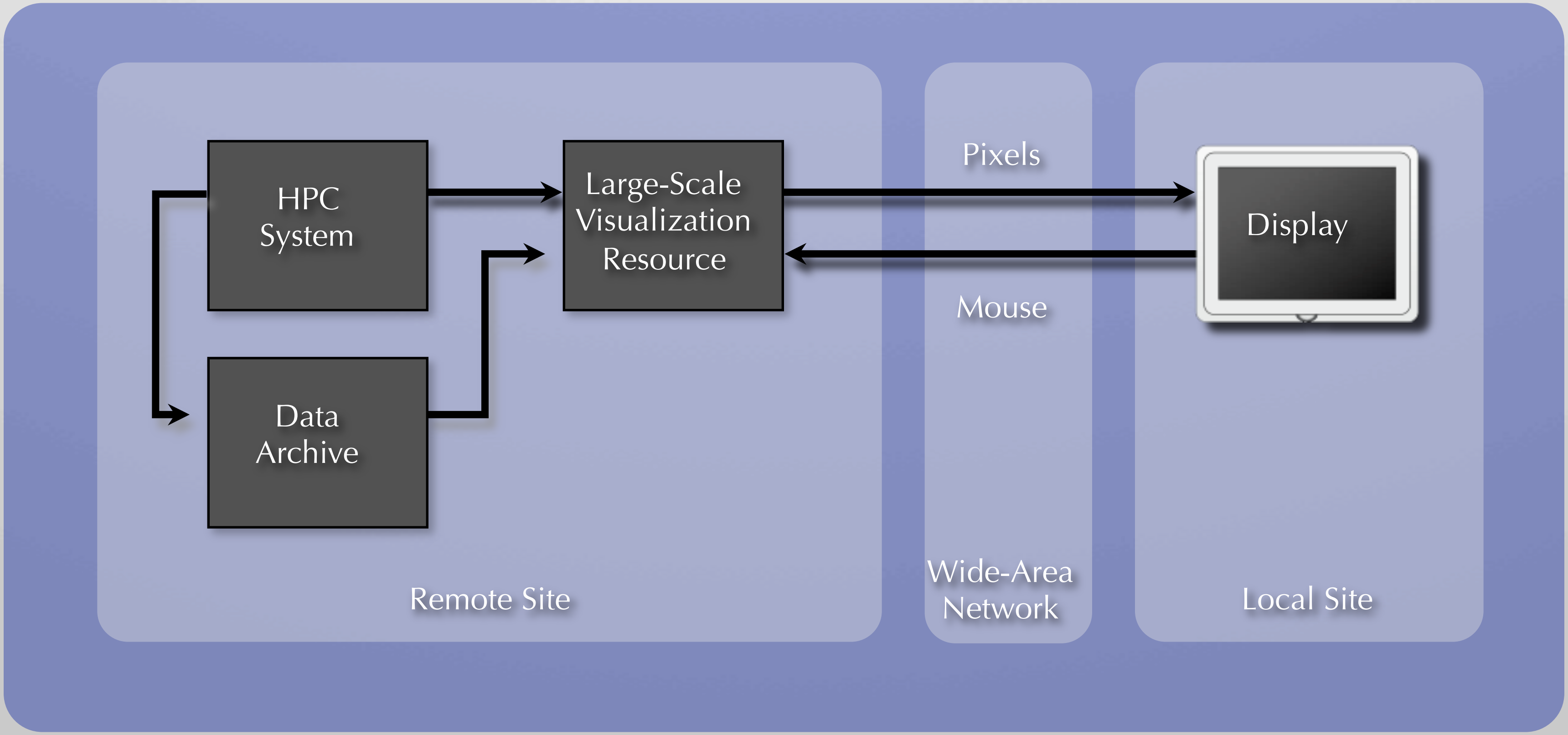
Medium problems:
- Data are costly to move over WAN
- Large memory node with a good GPU, remote vis

Small problems
- Data are small and easily moved
- Office machines and laptops are adequate for visualization

*** Slide thanks to Sean Ahern (NICS) and Kelly Gaither (TACC)**

Remote visualization

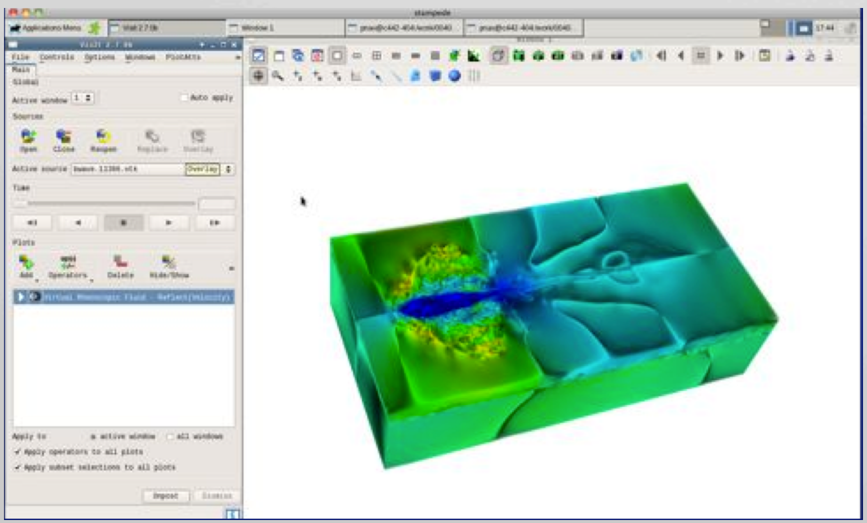
- Leave your data on the HPC resource!
- Remote interactive vis first, then batch job to make a movie

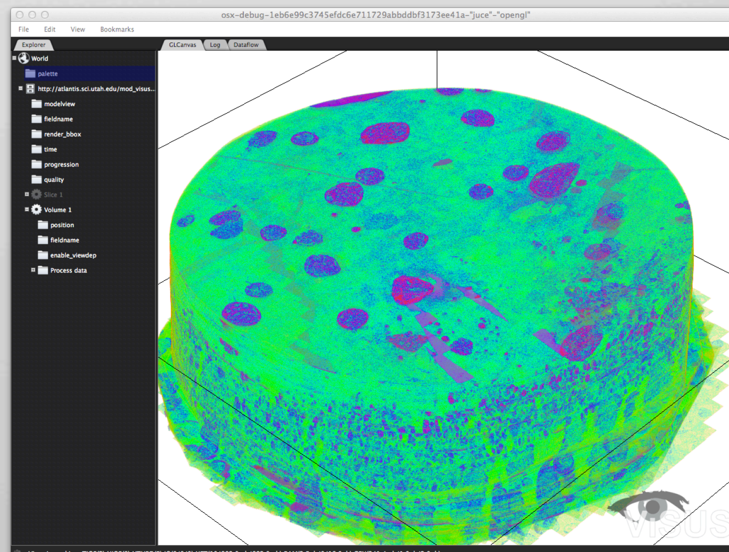


* Slide courtesy of Paul Navratil and Kelly Gaither (TACC)

Remote vis methods

- X11 (ssh –X)
- Client-server vis applications (ParaView, VisIt, Visus)
- VNC client, vglrun + xvnc on cluster (for accelerated XGL)
- Proprietary solutions (FastX, NX)





Example: VNC at TACC

Get a VNC client:
I like TigerVNC, http://tigervnc.org

From a unix shell:
ssh stampede.tacc.utexas.edu
sbatch –n 4 –N 4 -p vis -A [Allocation] /share/doc/slurm/job.vnc -geometry 1600x1200
touch vncserver.out
tail –f vncserver.out

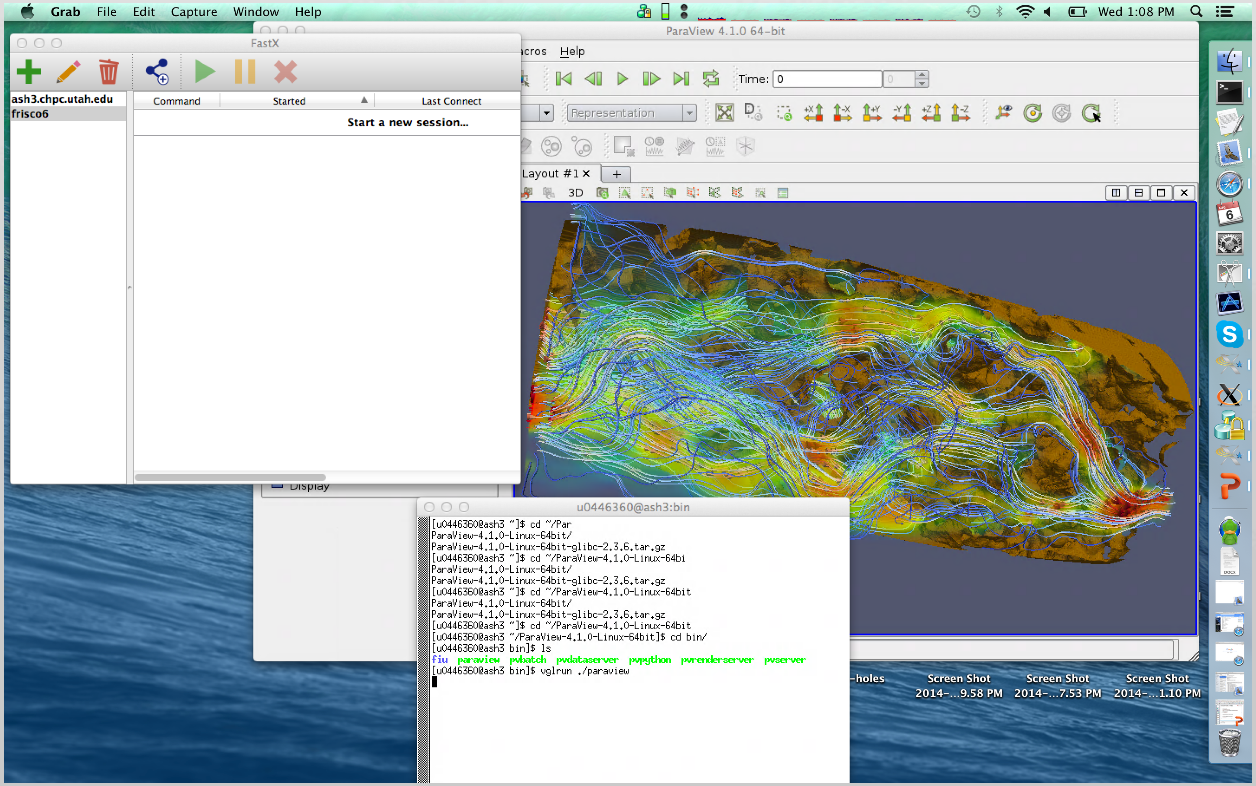
In another terminal on the client machine:
vncviewer stampede.tacc.utexas.edu:14214

Run any GLX application on the resource:
vglrun paraview

Example: FastX at CHPC

Same effect as VNC+vglrun but without a VNC client.

FastX client lets you launch and resize terminals and individual windows.



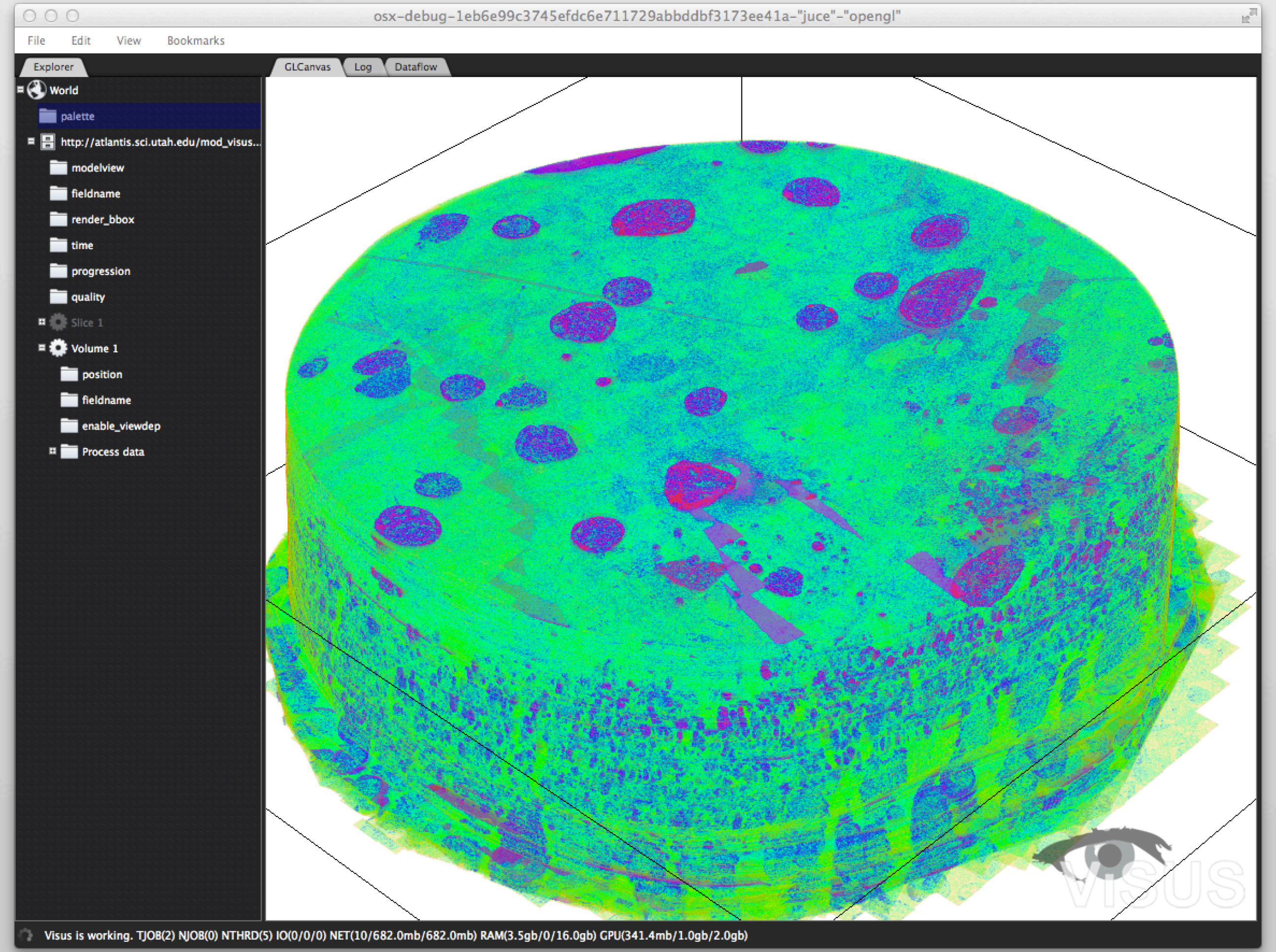
Data-parallel visualization

- One option for client-server remote visualization
- Partition the **data** across nodes.
- ParaView and VisIt were designed to execute VTK in data-parallel
- Many advantages:
  - Distributed memory (larger effective total memory)
  - Much, much faster file IO
  - Production versions of PV/VisIt already deployed at most DOE and NSF sites
    - See user guides for how to run in client-server mode
- Some disadvantages
  - Occasional bugs in the parallel implementations
  - Parallel compositing slow for interactive vis

Image-parallel visualization

- Partition the **screen** across nodes.
- Visus and ImageVis3D
- Each node computes a separate portion of the screen, with a local portion of global data
- Multiple options for data distribution
  - Level of detail (LOD) + "out-of-core" : Visus, ImageVis3D, Neurotrace
  - Shared cache (e.g., using MPI one-sided, MPI3 RMA) : Manta, OSPRay
- Advantages:
  - Always interactive!
  - Great for tiled displays
  - Can handle HUGE data in a scalable fashion
- Some disadvantages
  - You have to wait for the full-resolution data to appear
  - LOD data requires its own "streaming" format (e.g., IDX, UVF)

Image Parallel Visualization: Visus



Future of vis

- More (and easier) in situ visualization
- New data and programming models
  - DAX, Piston, EAVL, VTK-M
- CPU vs GPU?
  - Shared-memory vs distributed
  - Large memory helps visualization more than HPC…
- Better, more scalable rendering options
  - There will always be a need for better parallel vis on clusters!
- Merging infovis & sci vis

Challenging visualization is a major focus of efforts at SCI!
If you have questions: knolla@sci.utah.edu