Overview and Introduction to Scientific Visualization

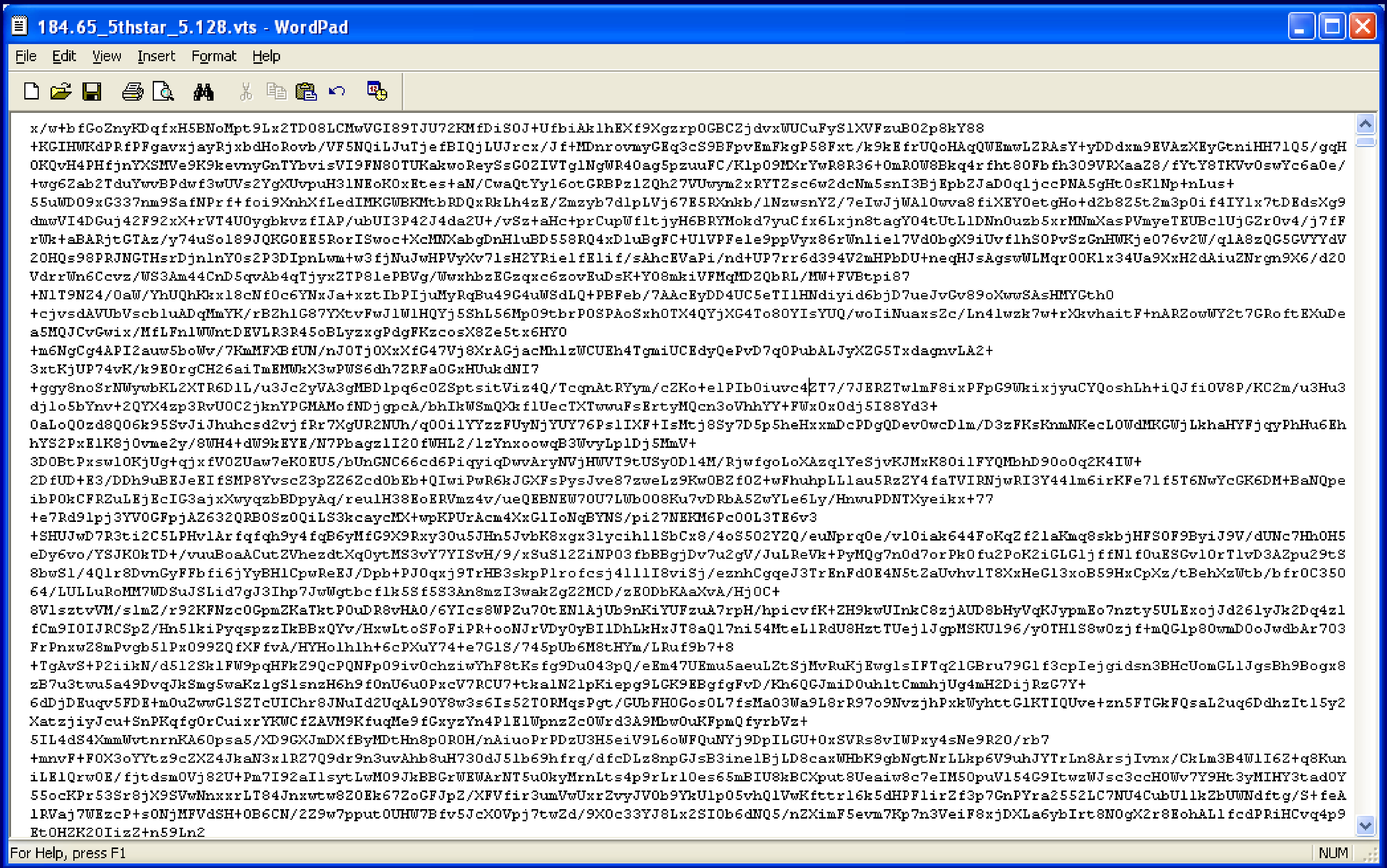
Paul Navrátil, Ph.D.
Texas Advanced Computing Center
The University of Texas at Austin

Visualization Training
September 8, 2011

Scientific Visualization

"The purpose of computing is insight not numbers."

-- R. W. Hamming (1961)


184.65_5thstar_5.128.vts - WordPad
File Edit View Insert Format Help
For Help, press F1
NUM

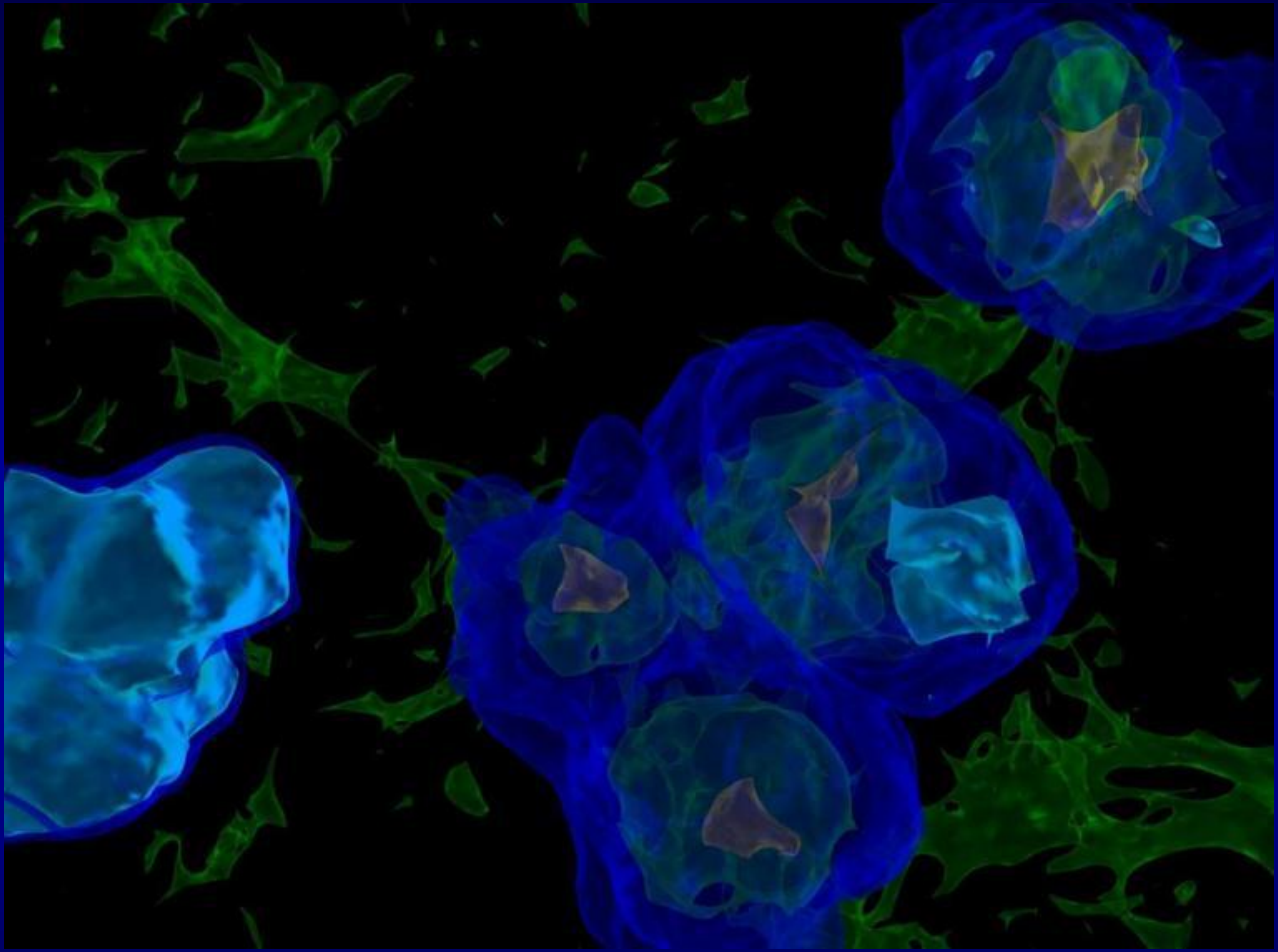
184.65_5thstar_5.128.vts - WordPad

File Edit View Insert Format Help

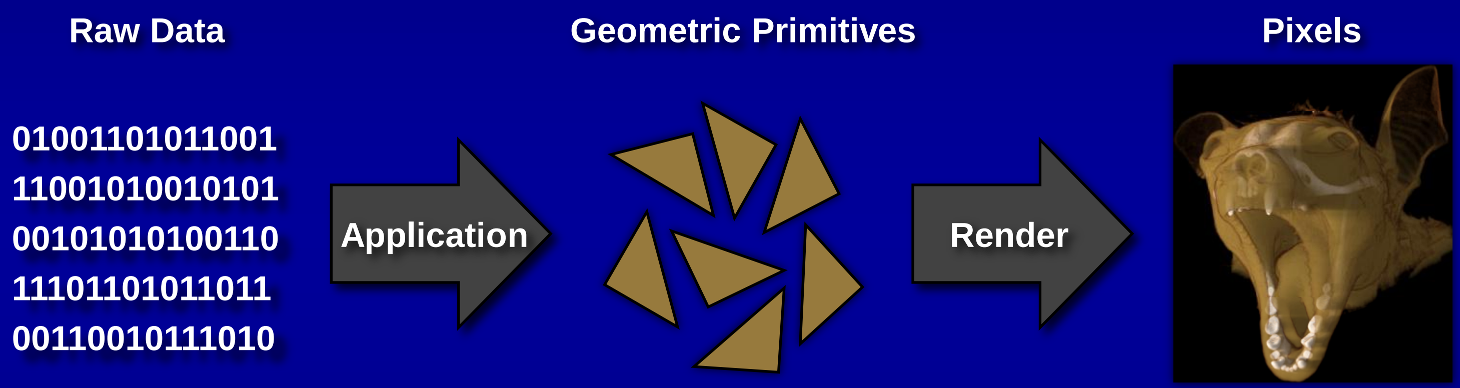

```
<?xml version="1.0"?>
<VTKFile type="StructuredGrid" version="0.1" byte_order="LittleEndian" compressor="vtkZLibDataCompressor">
  <StructuredGrid WholeExtent="0 127 0 127 0 127">
    <Piece Extent="0 127 0 127 0 127">
      <PointData Scalars="sink_identifier">
        <DataArray type="Float32" Name="sink_identifier" format="appended" offset="0"/>
        <DataArray type="Float32" Name="temperature" format="appended" offset="19136"/>
        <DataArray type="Float32" Name="density" format="appended" offset="4709980"/>
        <DataArray type="Float32" Name="electron_fraction" format="appended" offset="14273720"/>
        <DataArray type="Float32" Name="H2_fraction" format="appended" offset="23144016"/>
        <DataArray type="Float32" Name="HD_fraction" format="appended" offset="32285480"/>
        <DataArray type="Float32" Name="He3_fraction" format="appended" offset="41430172" />
        <DataArray type="Float32" Name="He2_fraction" format="appended" offset="41745020" />
        <DataArray type="Float32" Name="He1_fraction" format="appended" offset="42055764" />
        <DataArray type="Float32" Name="particle_mass" format="appended" offset="42329188" />
      </PointData>
      <CellData>
      </CellData>
      <Points>
        <DataArray type="Float32" NumberOfComponents="3" format="appended" offset="42499596" />
      </Points>
    </Piece>
  </StructuredGrid>
  <AppendedData encoding="base64">

_AAEAAACAAAAAAAAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0A
AAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA
0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAA
AA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQ
AAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAA
DQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAA
AADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAAN
AAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAA
ANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0A
AAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAADQAAAA0AAAANAAAAA==eJztwQEBAAAAgJD+r+
4ICgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYgAAAAXic7cEBAQAAAICQ/q/uCAoAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGIAAAAF4n03BAQEAAACAkP6v7ggK
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABiAAAABeJztwQEBAAAAgJD+r+
4ICgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAYgAAAAXic7cEBAQAAAICQ/q/uCAoAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAGIAAAAF4n03BAQEAAACAkP6v7ggK
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABiAAAABeJztwQEBAAAAgJD+r+
```


For Help, press F1

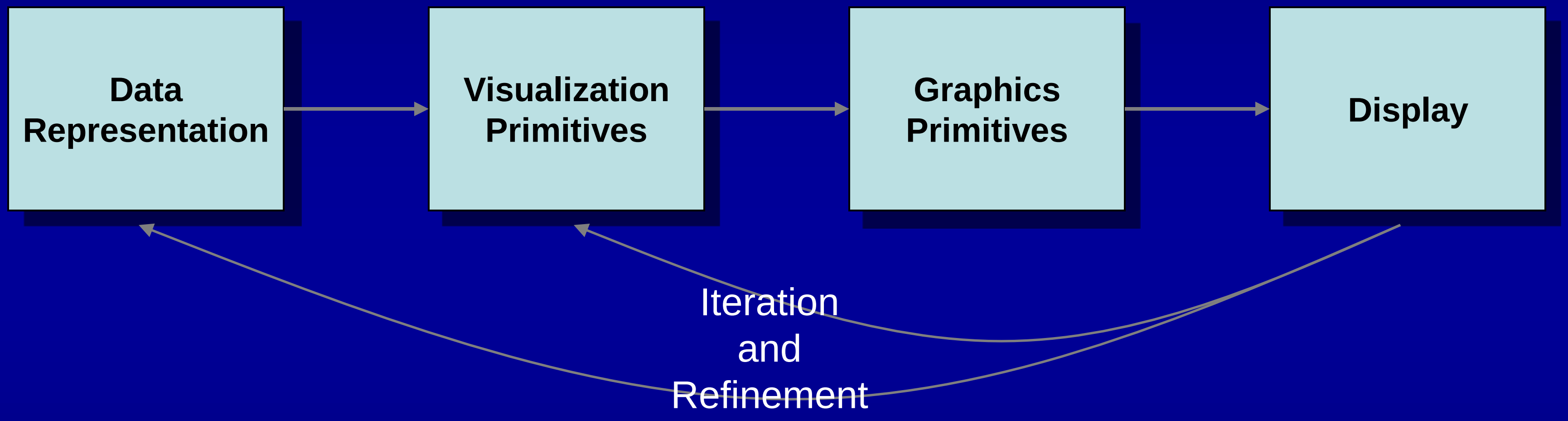
NUM



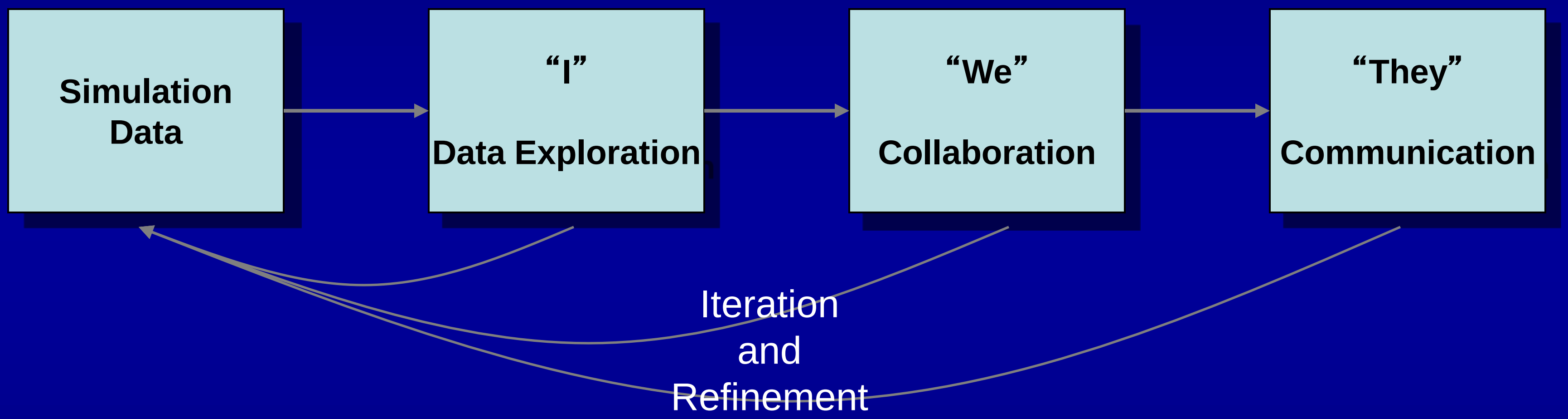
Visualization Allows Us to “See” the Science



Getting from Data to Insight



“I, We, They” Development Path



Visualization Process Summary

- The primary goal of visualization is ***insight***
- A picture is worth not just 1000 words,
  but potentially tera- or peta-bytes of data
- Larger datasets demand not just visualization, but advanced visualization resources and techniques
- Visualization system technology improves with advances in GPUs and LCD technology
- Visualization software slower to adapt

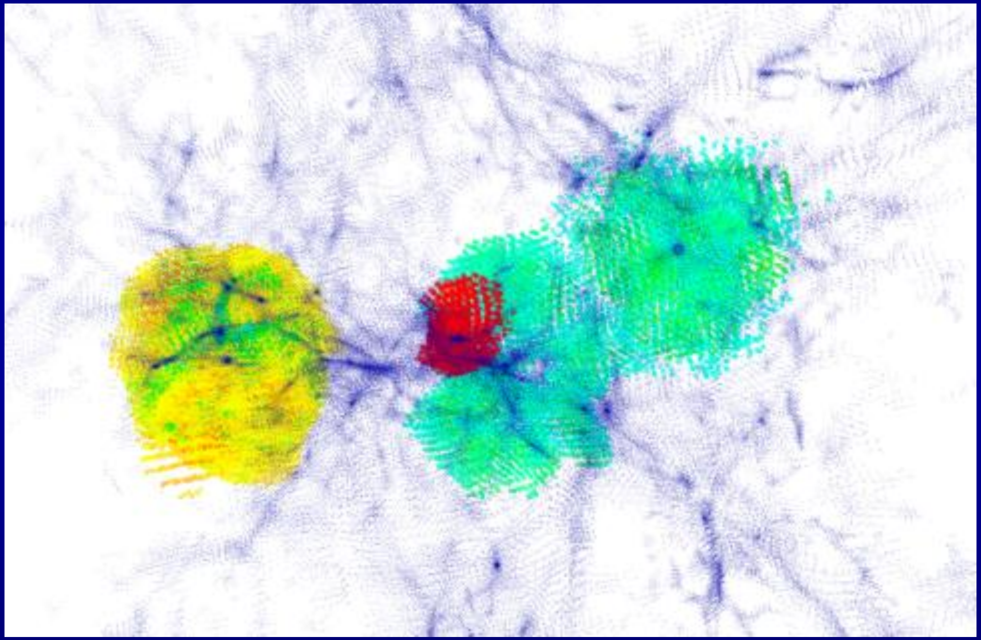
Types of Input Data

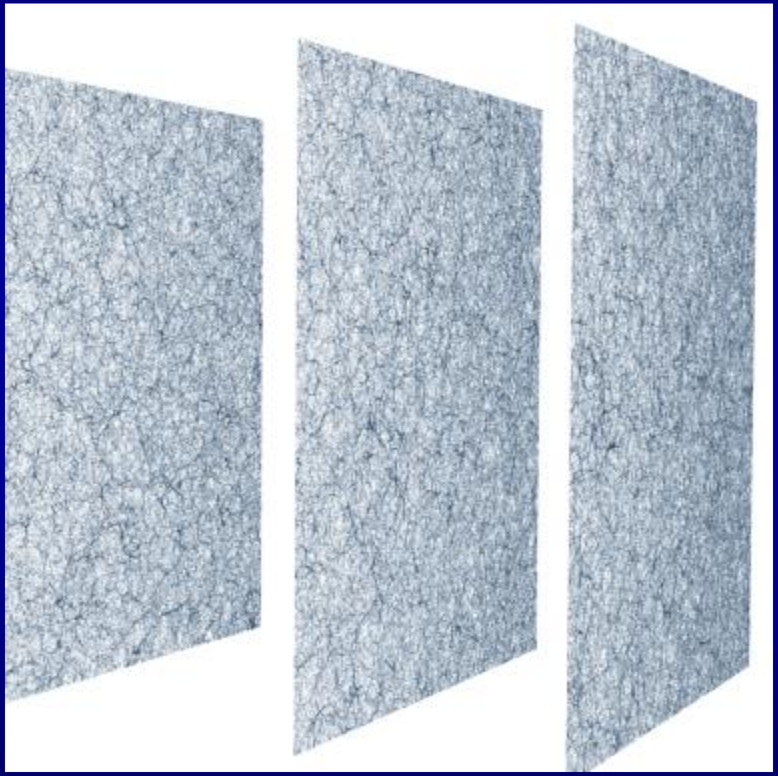
- Point / Particle
  - N-body simulation
- Regular grid
  - Medical scan
- Curvilinear grid
  - Engineering model
- Unstructured grid
  - Extracted surfaces

TACC

Types of Input Data

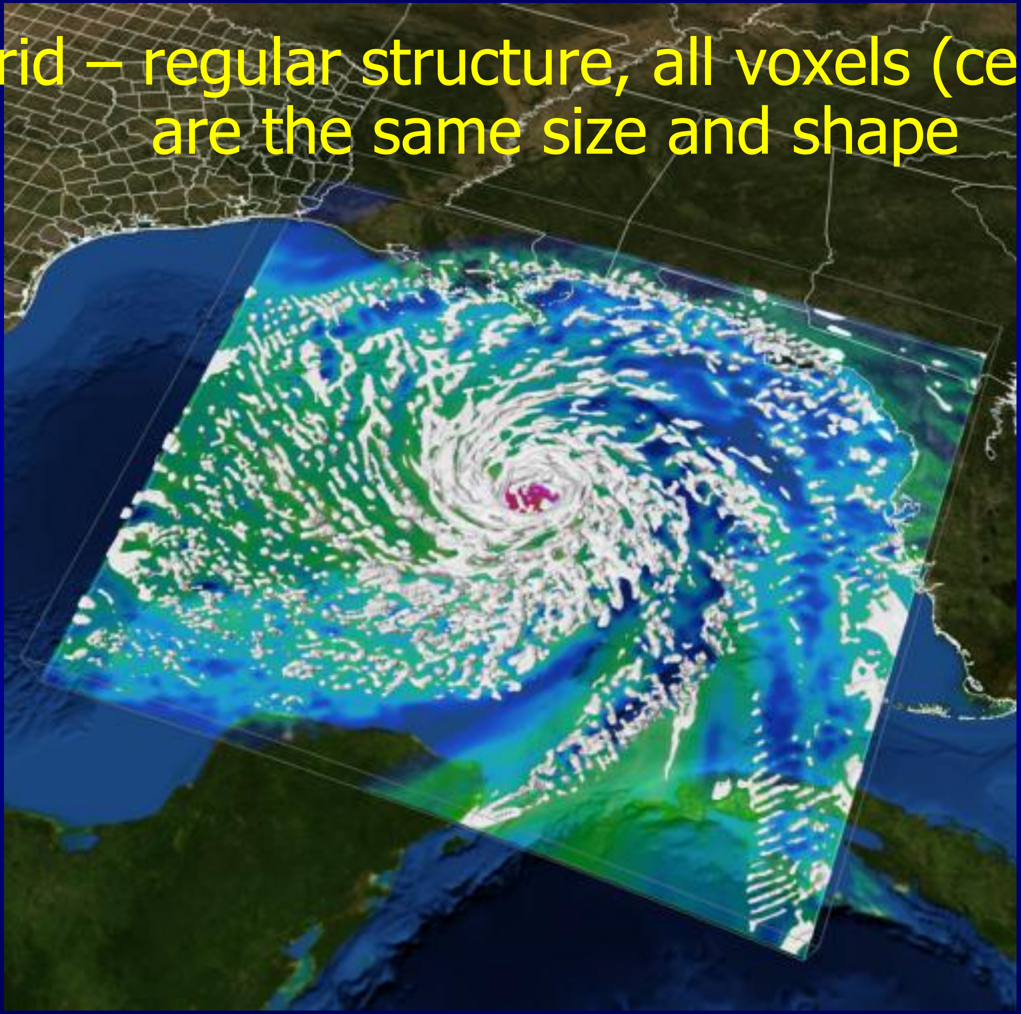
Point – scattered values with no defined structure





Types of Input Data

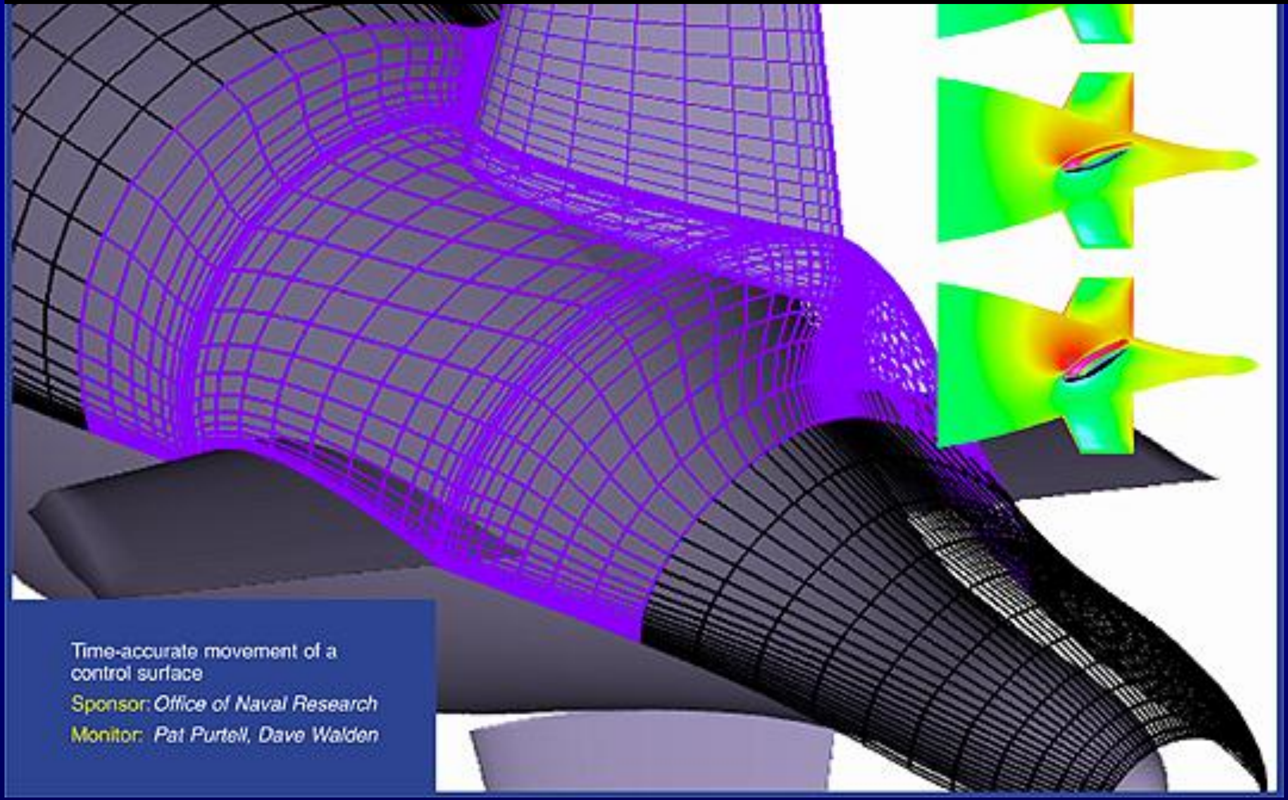
Grid – regular structure, all voxels (cells) are the same size and shape



TACC

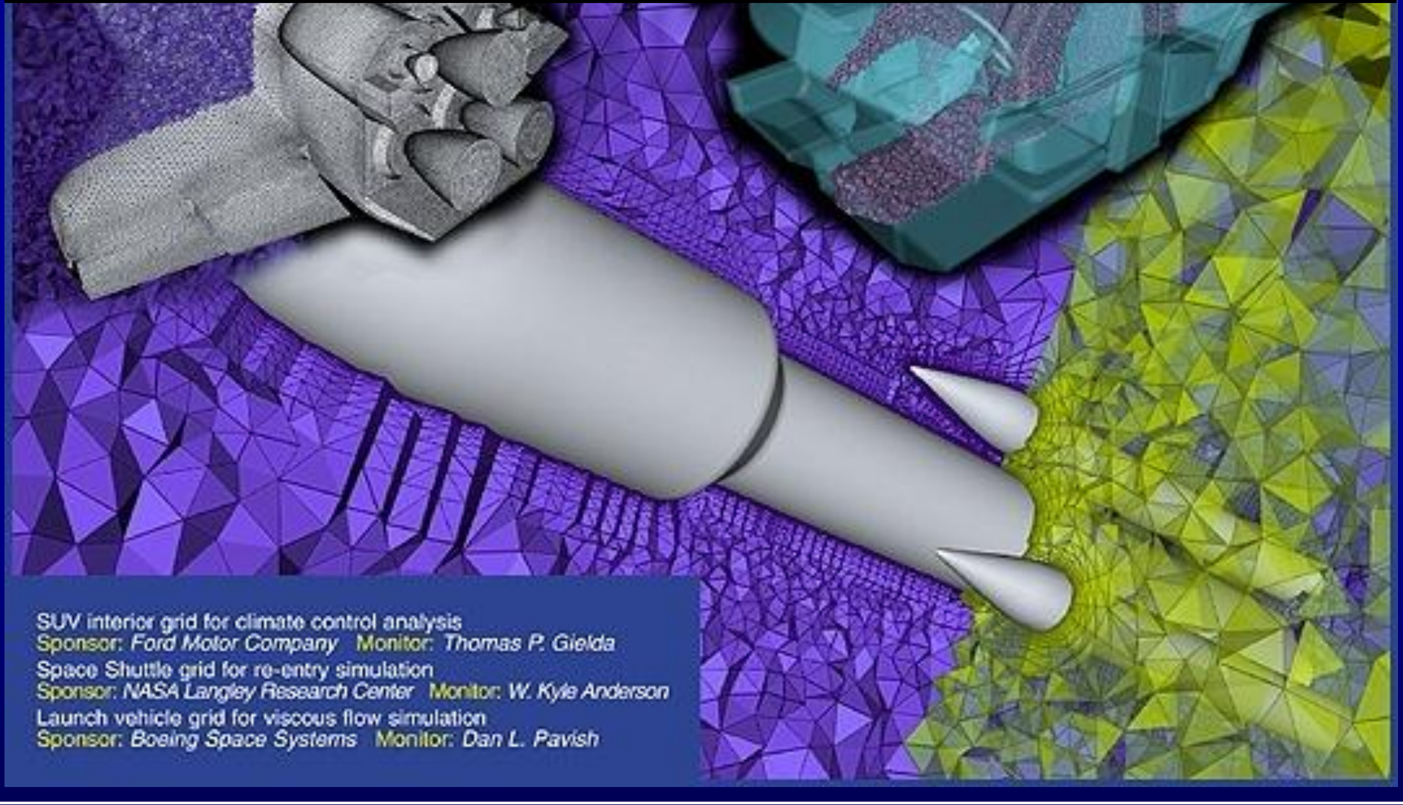
Types of Input Data

Curvilinear – regularly grided mesh
shaping function applied



Types of Input Data

Unstructured grid – irregular mesh typically composed of tetrahedra, prisms, pyramids, or hexahedra.



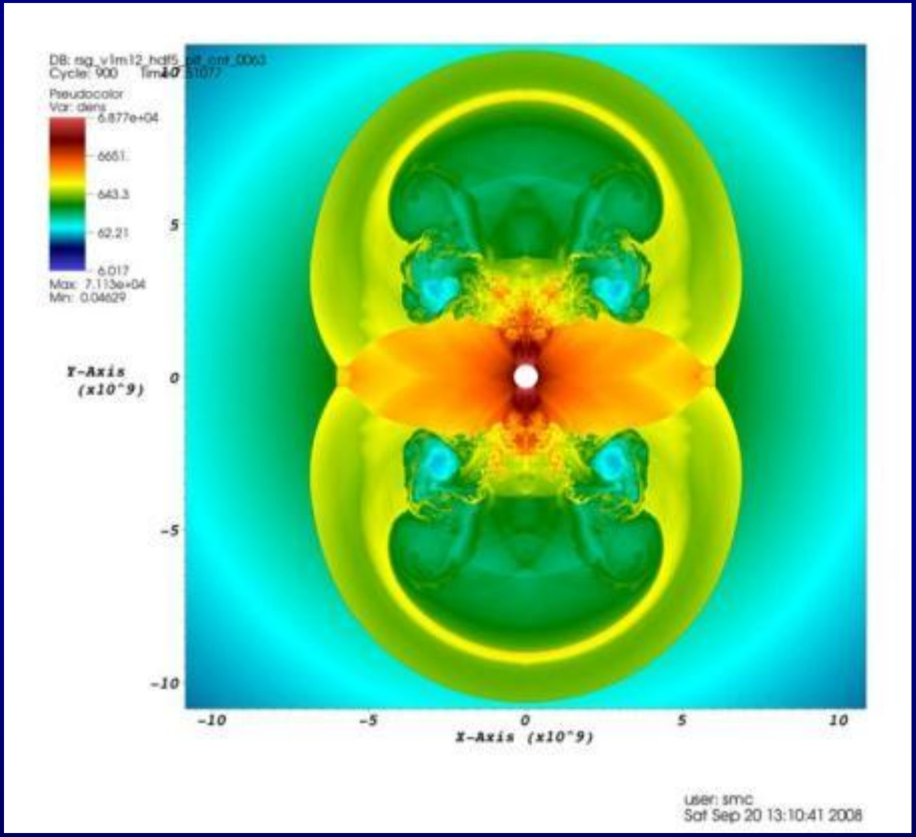
Visualization Operations

- Surface Shading (Pseudocolor)
- Isosufacing (Contours)
- Volume Rendering
- Clipping Planes
- Streamlines

Surface Shading (Pseudocolor)

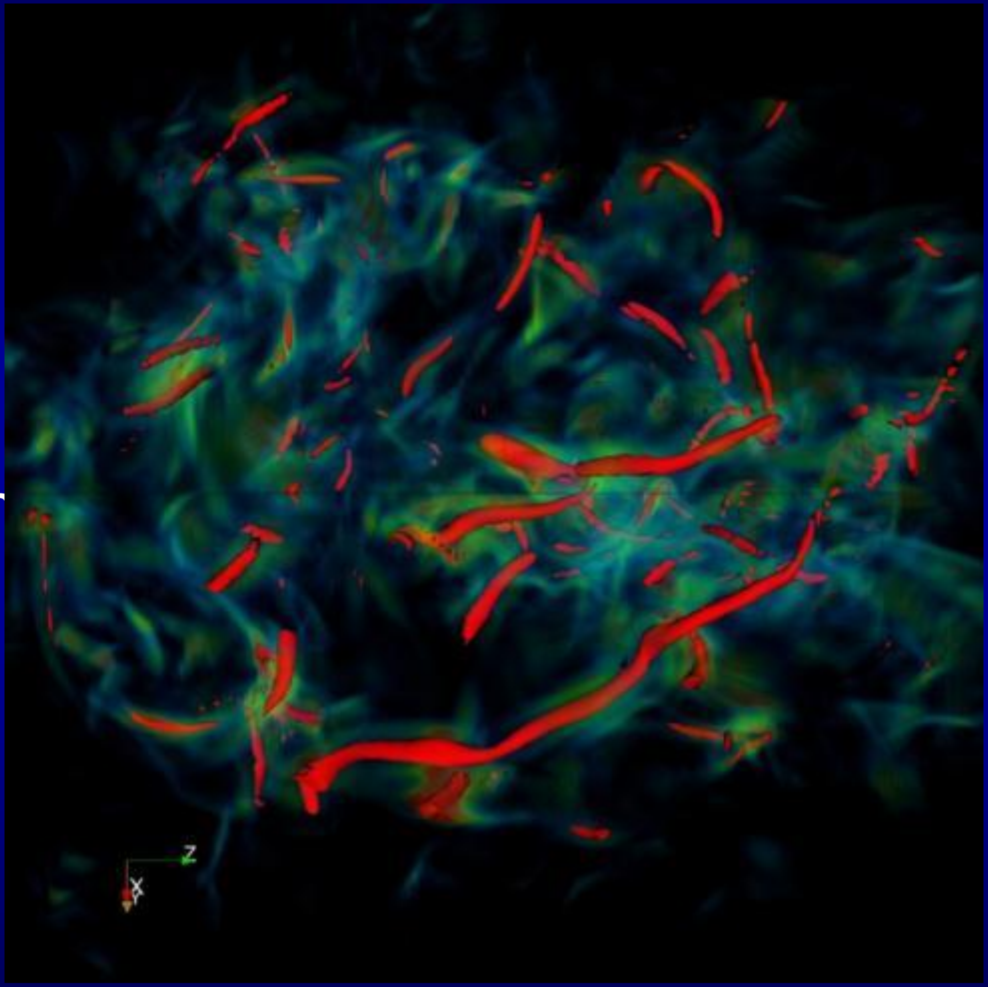
Given a scalar value at a point on the surface and a color map, find the corresponding color (and opacity) and apply it to the surface point.

Most common operation, often combined with other ops



Isosurfaces (Contours)

- Surface that represents points of constant value with a volume
- Plot the surface for a given scalar value.
- Good for showing known values of interest
- Good for sampling through a data range

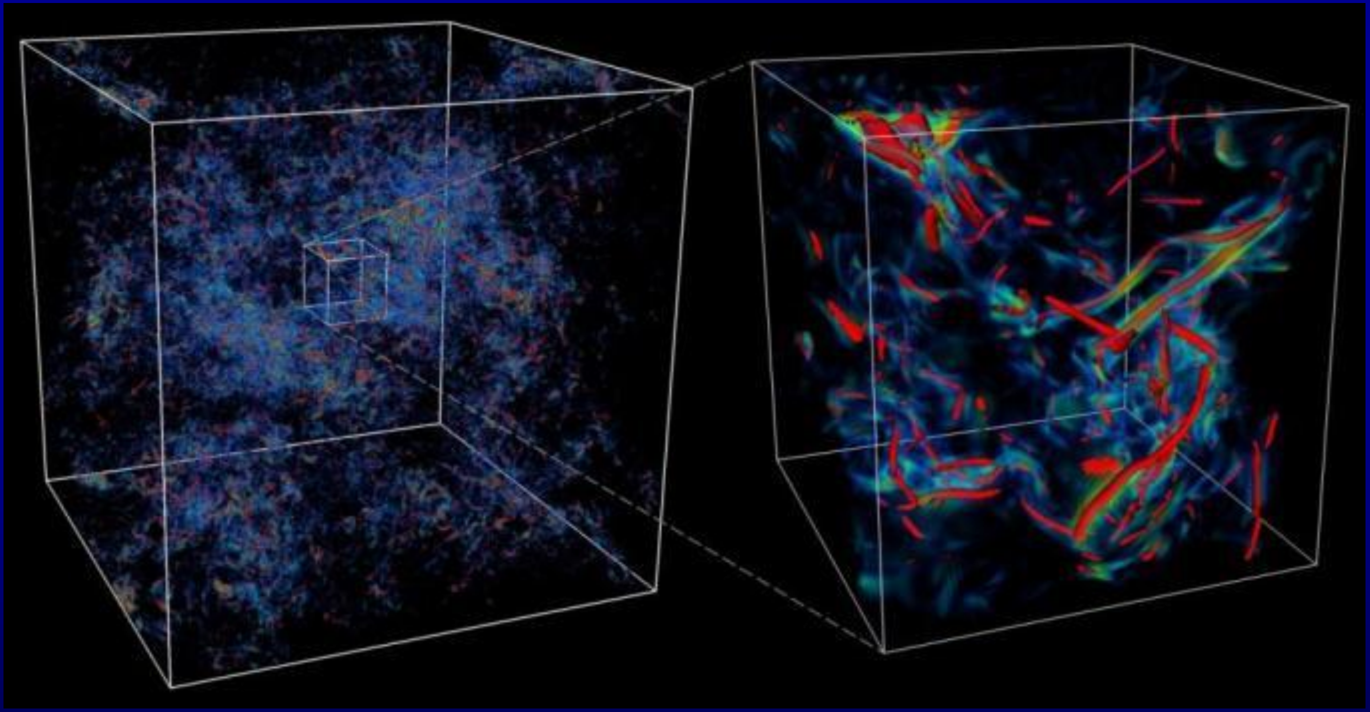


Volume Rendering

Expresses how light travels through a volume

Color and opacity controlled by transfer function

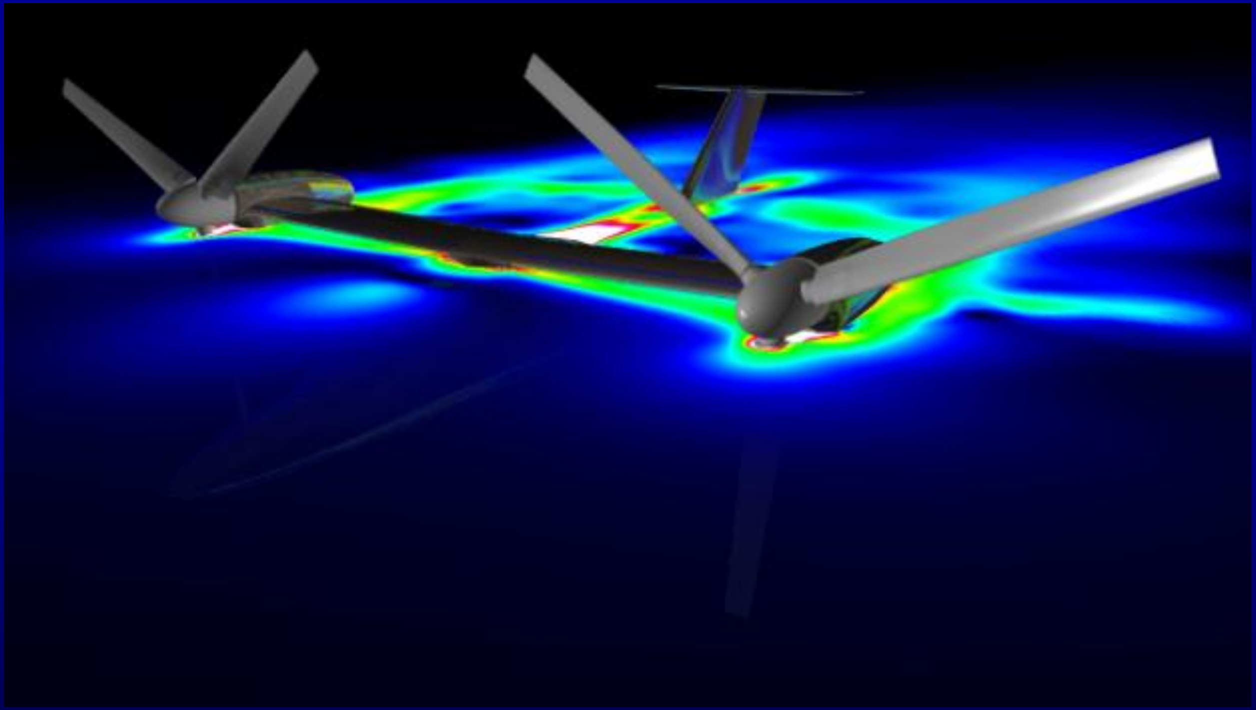
Smoother transitions than isosurfaces



Clipping / Slicing Planes

Extract a plane from the data to show features

Hide part of dataset to expose features

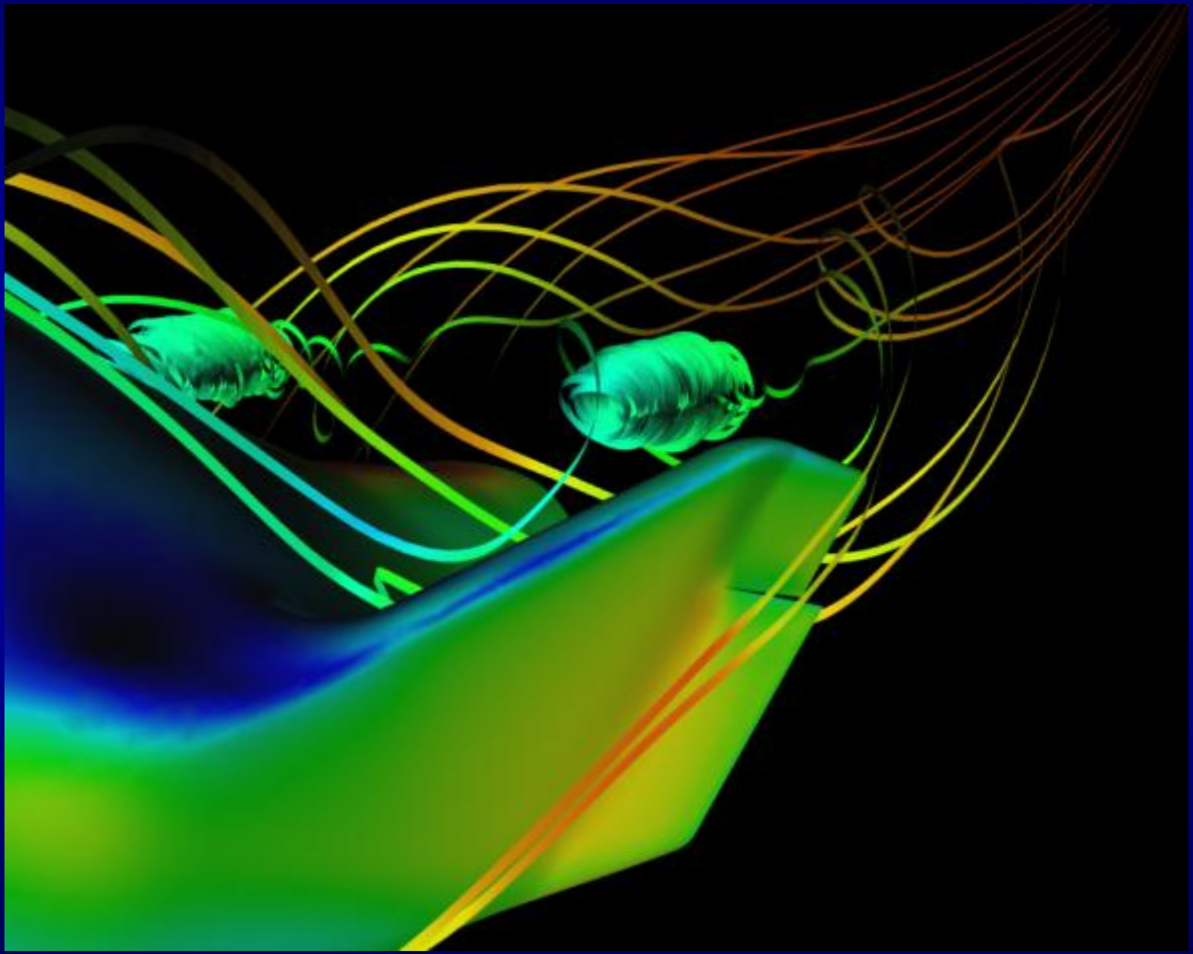


Particle Traces (Streamlines)

Given a vector field, extract a trace that follows that trajectory defined by the vector.

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

Streamlines – trace in space
Pathlines – trace in time



Visualization Resources

- Personal machines
  - Most accessible, least powerful
- Projection systems
  - Seamless image, high purchase and maintenance costs
- Tiled-LCD displays
  - Lowest per-pixel costs, bezels divide image
- Remote visualization
  - Access to high-performance system, latency can affect user experience

XSEDE Visualization Resources

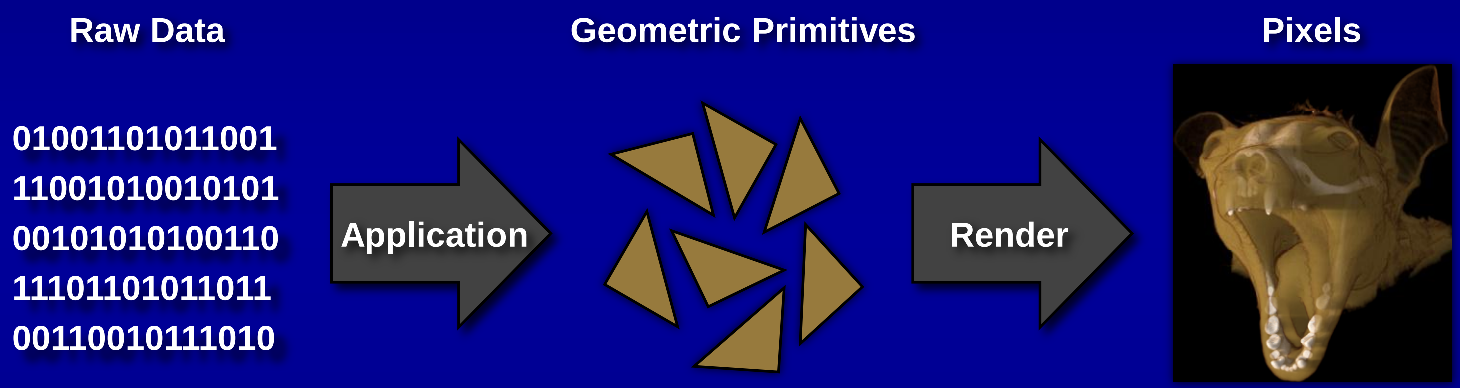
- Longhorn (TACC)
  - 256 nodes, 2048 total cores, 512 total GPUs
    13.5 TB aggregate memory, QDR InfiniBand interconnect
  - Ranger file system mounted read-only across system
  - Longhorn Visualization Portal
    - https://portal.longhorn.tacc.utexas.edu/
    - Visualization job submission and monitoring
    - Remote, interactive, web-based visualization
    - Guided visualization using EnVision
- Spur (TACC)
  - 8 nodes, 128 total cores, 32 total GPUs
    1 TB aggregate memory, SDR InfiniBand interconnect
  - Shares interconnect and file system with Ranger
- Lonestar (TACC)
  - 8 nodes, 96 total cores, 32 total GPUs (expand to 72 nodes soon)192 GB aggregate memory, QDR InfiniBand interconnect
  - Shares interconnect and file system with Lonestar HPC nodes

TeraGrid Visualization Resources

- Nautilus (NICS)
  - SMP, 1024 Total Cores, 8 GPUs
    4 TB Global Shared Memory, SGI NUMAlink 5 interconnect
- TeraDRE Condor Pool (Purdue)
  - 1750 Nodes, 14000 Total Cores, 48 Nodes with GPUs
    28 TB Aggregate Memory, no interconnect

Visualization Challenges

Visualization Allows Us to “See” the Science



But what about large, distributed data?

01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

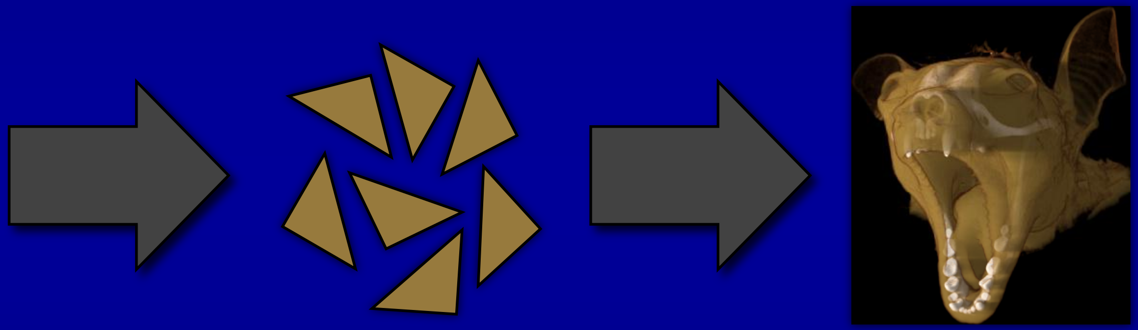
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

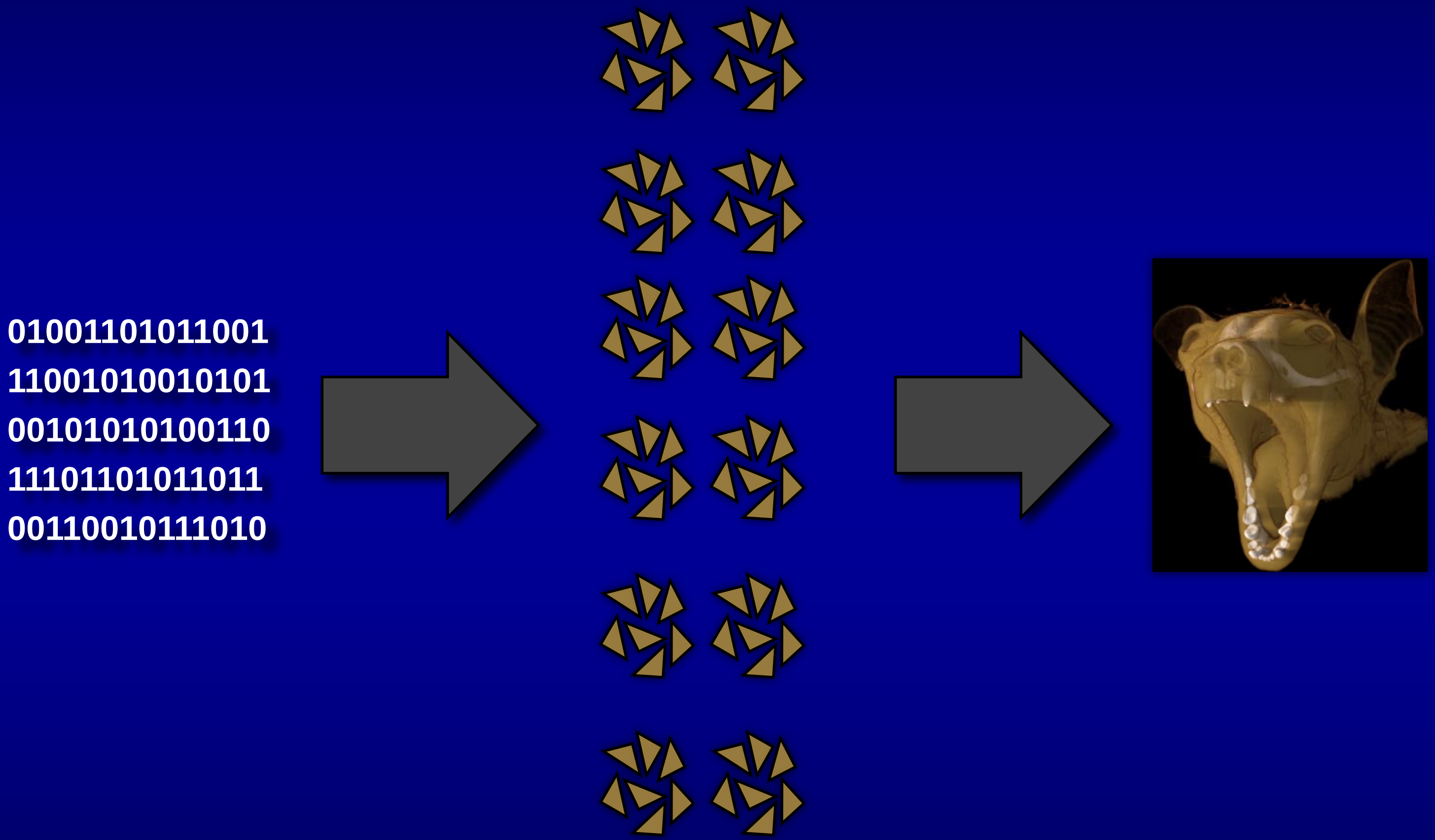
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

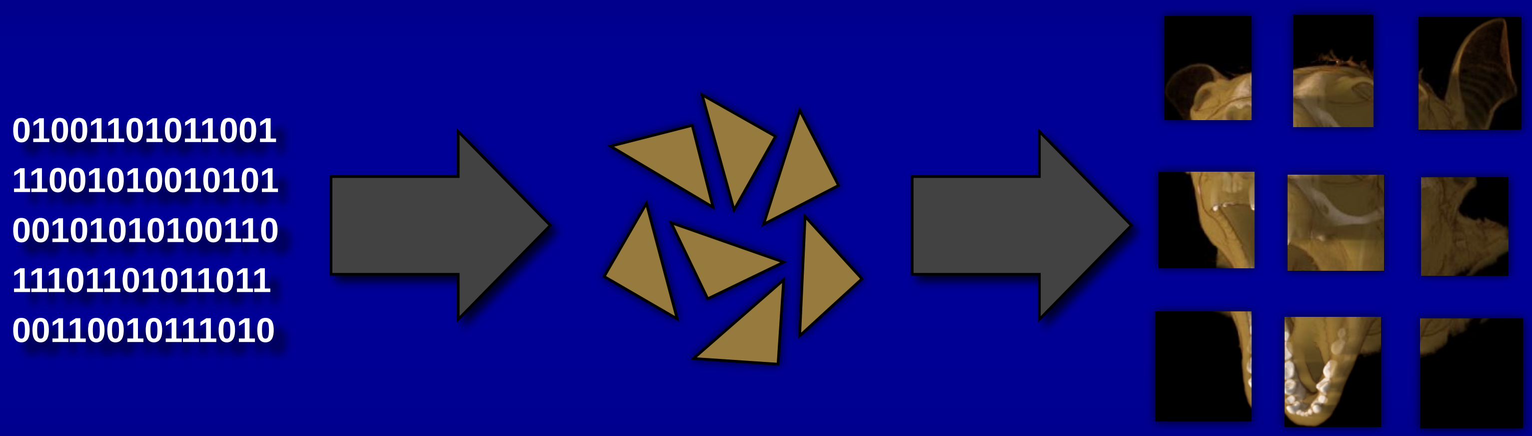


TACC

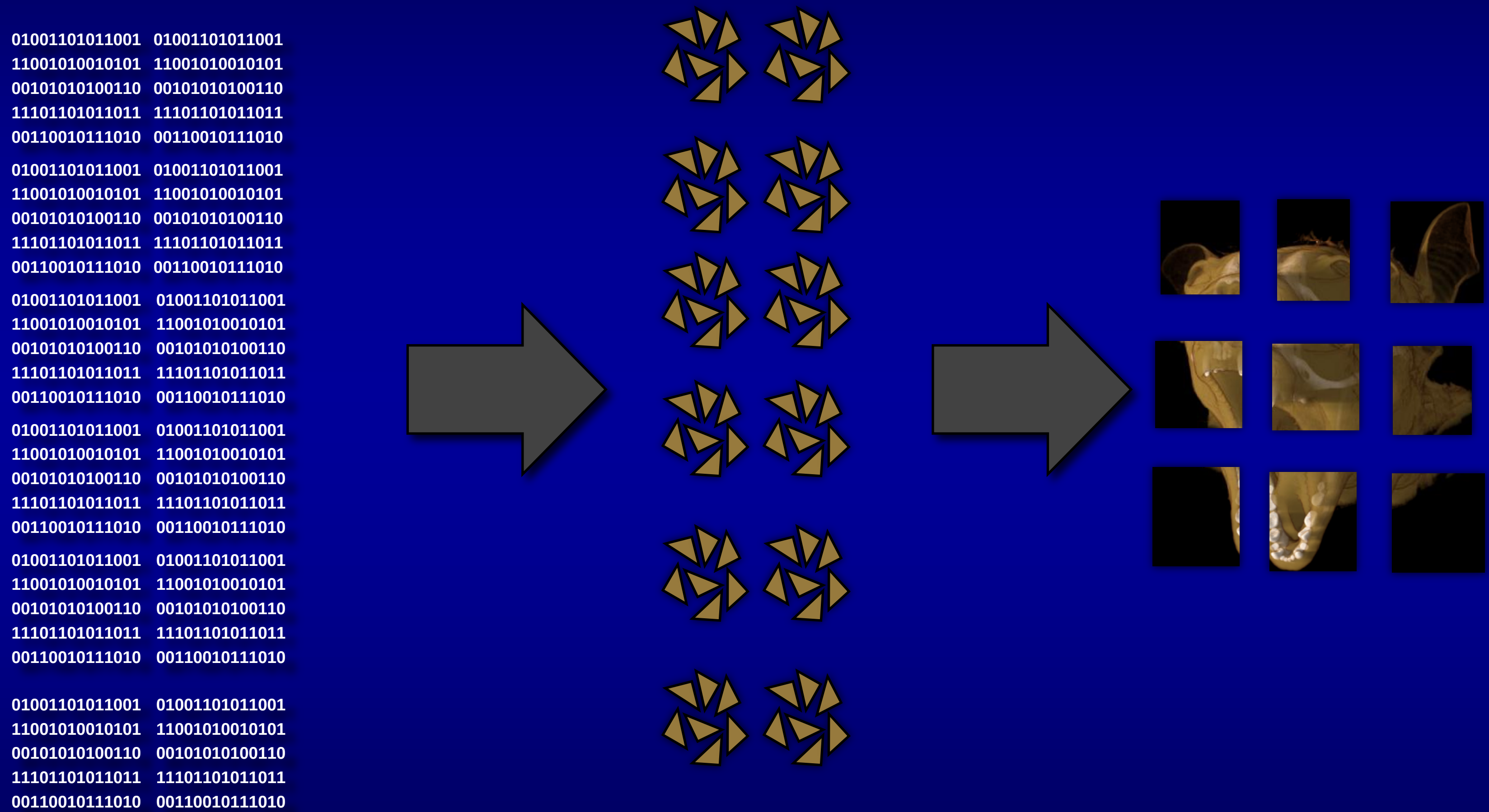
Or distributed rendering?



Or distributed displays?



Or all three?



01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010
01001101011001 01001101011001
11001010010101 11001010010101
00101010100110 00101010100110
11101101011011 11101101011011
00110010111010 00110010111010

Visualization Scaling Challenges

- Moving data to the visualization machine
- Most applications built for shared memory machines, not distributed clusters
- Image resolution limits in some software cannot capture feature details
- Displays cannot show entire high-resolution images at their native resolution

Visualization scales with HPC

Large data produced by large simulations require large visualization machines and produce large visualization results

Moving Data

- How long can you wait?

File Size	10 Gbps	54 Mbps
1 GB	1 sec	2.5 min
1 TB	~17 min	~43 hours
1 PB	~12 days	~5 years

Analyzing Data

- Visualization programs only beginning to efficiently handle ultrascale data
  - 650 GB dataset -> 3 TB memory footprint
  - Allocate HPC nodes for RAM not cores
  - N-1 idle processors per node!
- Stability across many distributed nodes
  - Rendering clusters typically number N <= 64
  - Data must be dividable onto N cores
    Remember this when resampling!

Imaging Data

Image: NASA Blue Marble Project

<u>Hypothetical fly-around movie</u>

4096 x 2160 PNG	~ 10 MB
x 360 degrees	~ 3.6 GB
x 30 days	~ 108 GB
x 12 months	~ 1.3 TB
@ 10 fps	3.6 hours
@ 60 fps	36 min

TACC

Displaying Data

Dell 30" flat-panel LCD

4 Megapixel display

2560 x 1600 resolution

Displaying Data

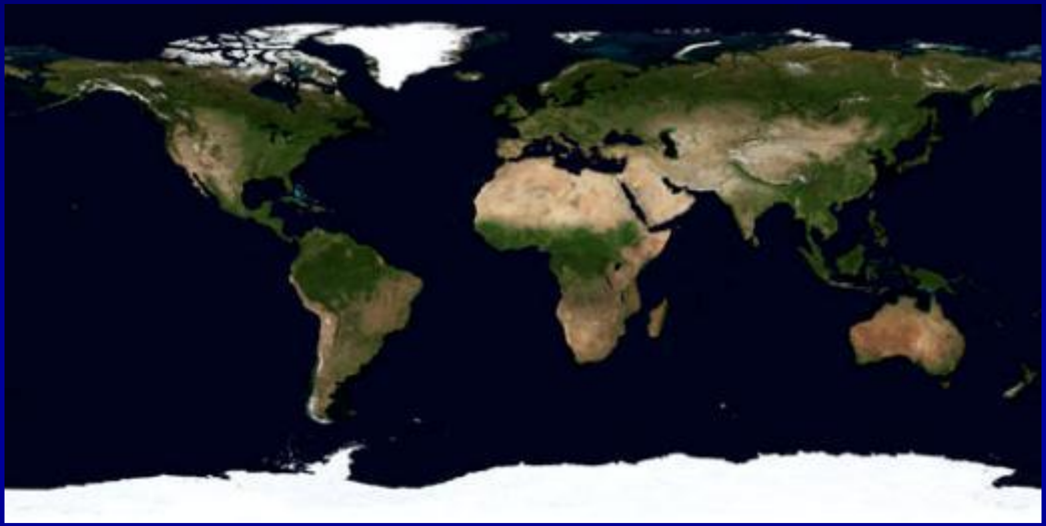
Stallion – currently world’s highest-resolution tiled display

307 Megapixels
38400 x 8000 pixel resolution

 Dell 30” LCD

TACC

Displaying Data



NASA Blue Marble
0.5 km^2 per pixel

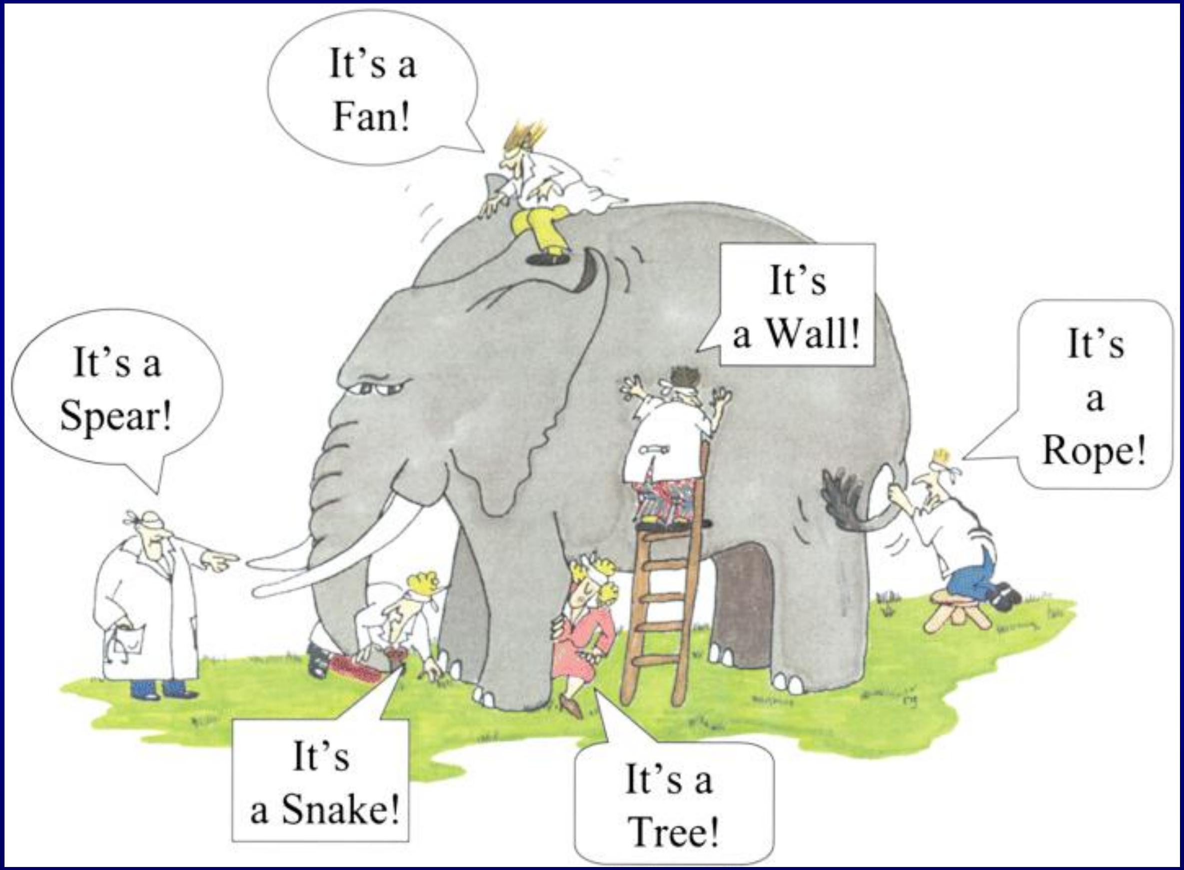
3732 Mpixel
(86400 x 43200)

Stallion – 307 Mpixel (38400 x 8000)

- Dell 30” LCD – 4 Mpixel (2560 x 1600)

TACC

What's the solution?



Solution by Partial Sums

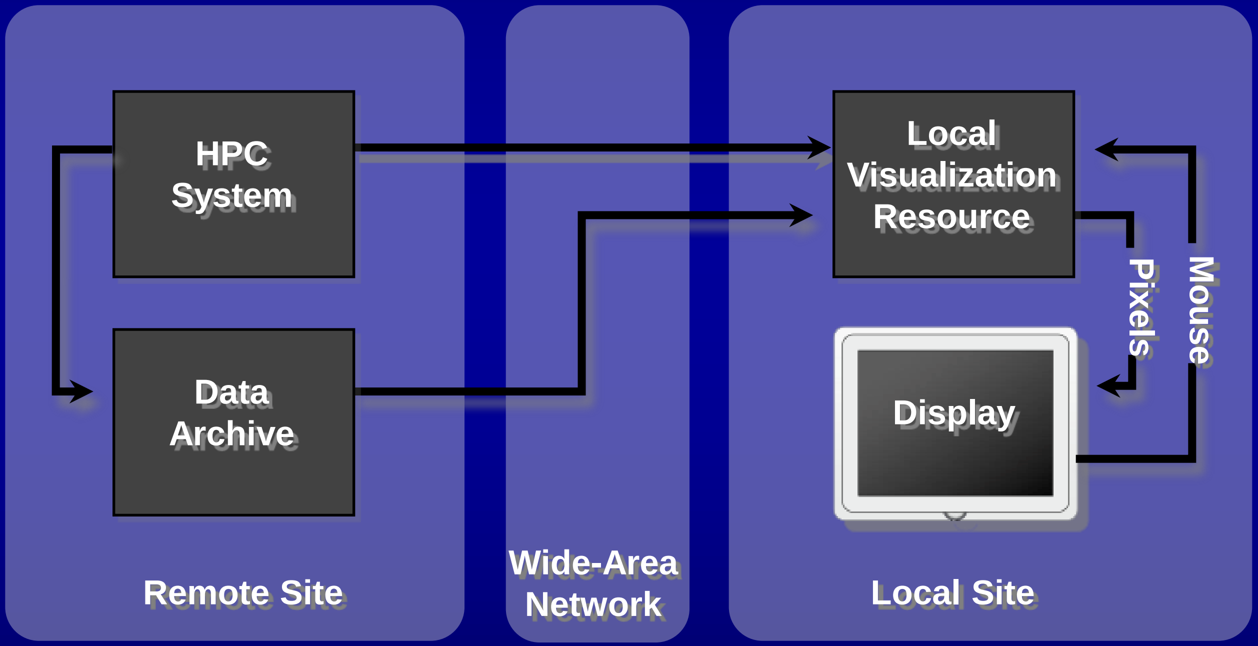
- Moving data – integrate vis machine into simulation machine. Move the machine to data!
  - Ranger + Spur: shared file system and interconnect
- Analyzing data – create larger vis machines and develop more efficient vis apps
  - Smaller memory footprint
  - More stable across many distributed nodes

Until then, the simulation machine is the vis machine!

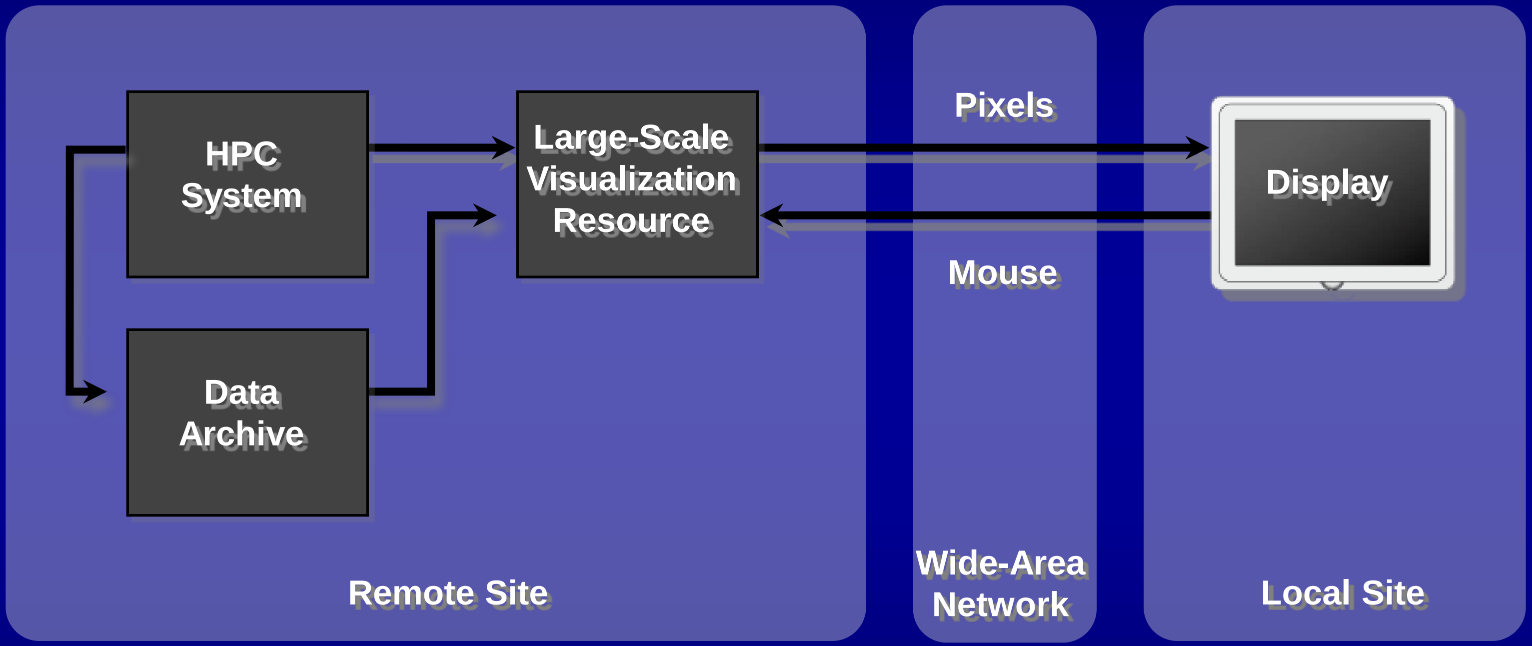
Solution by Partial Sums

- Imaging data – focus vis effort on interesting features
  parallelize image creation
  - Feature detection to determine visualization targets
    but can miss "unknown unknowns"
  - Distribute image rendering across cluster
- Displaying data – high resolution displays
  multi-resolution image navigation
  - Large displays need large spaces
  - Physical navigation of display provides better insights

Old Model
(No Remote Capability)



New Model
Remote Capability



Using GPUs

- More than for just rendering!
  - HPC applications and Visualization algorithms
- Parallelism – kernel should be highly SIMD/SIMT
  - Switching kernels is expensive!
  - Fermi *hardware* supports multiple kernel execution
- Control Flow – avoid conditionals in kernels
  - Implemented with predication, harms utilization
- Job size – high workload per thread + many threads
  - amortize thread initialization and memory transfer costs
  - GPU is a throughput machine, must keep it busy!
- Memory footprint – task must decompose well
  - local store per GPU core is low (16 KB on Longhorn)
  - card-local RAM is limited (4GB on Longhorn)
  - access to system RAM is slow (treat like disk access)
- More on this Thursday during CUDA for HPC

Summary

- Challenges at every stage of visualization when operating on large data
- Partial solutions exist, though not integrated
- Problem sizes continue to grow at every stage
- Vis software community must keep pace with hardware innovations

Thank you!

pnav@tacc.utexas.edu