THE VISH VISUALIZATION ENVIRONMENT

And Application to Computing Streamlines in Multibock Datasets

Werner Benger
Scientific Visualization Group
Center for Computation & Technology
at
Louisiana State University

Outline

- What is VISH? - Design and structure
- Features of VISH – Status Quo
- Application to computing streamlines in multiblock vectorfield data

What stands "VISH" for?

- A **Vi**sualization **Sh**ell
- A framework for realizing **Vi**sualization Wi**sh**es
- Something else... (free to imagination)
- Pronouncation (proposal):
  - Even times: "fish"
  - Odd times: "wish"
  - Maritime naming convention

What is VISH?

- A highly modular infrastructure to implement visualization (and more) algorithms
  - Strong encapsulation between components
  - Abstract interface that allows to integrate VISH components into existing applications
- Everything is a plugin
  - "microkernel" – defines objects an their relationships
  - "plugins" – OpenGL rendering, data I/O layers, GUI
- Academic open source licensing
  - not formally open source, but freely available to academic community

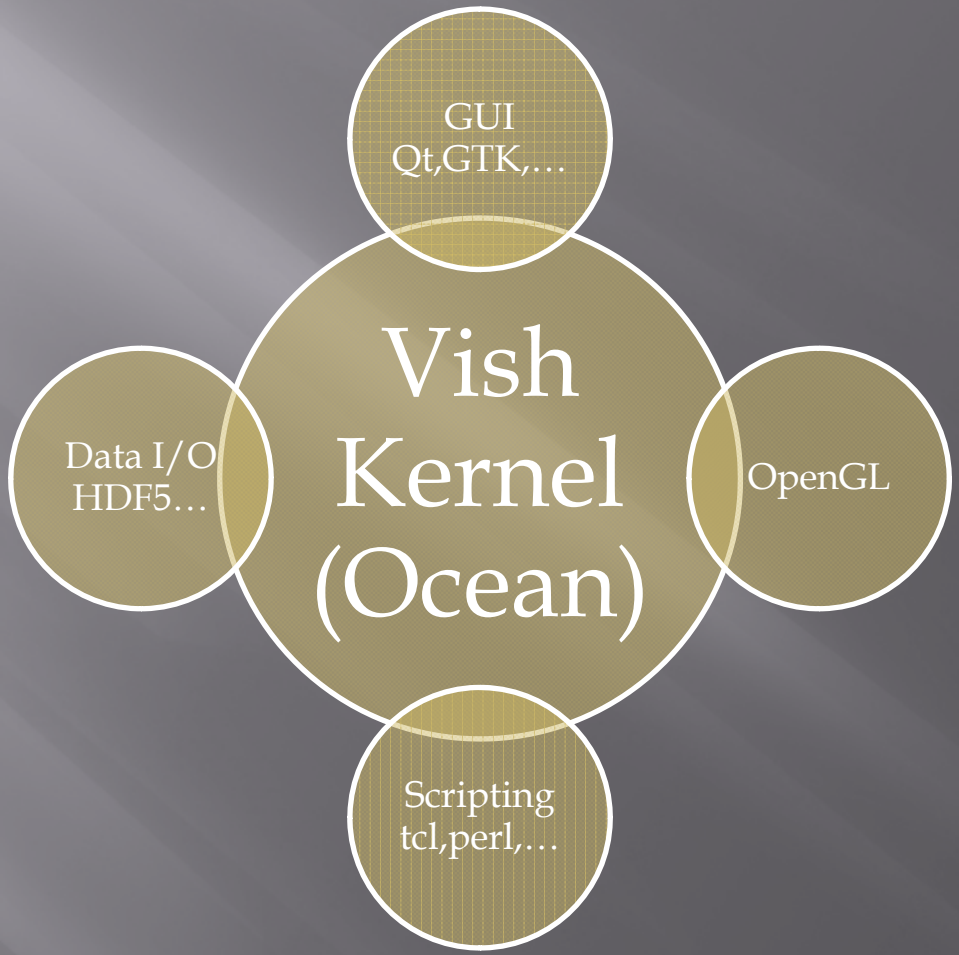
Vision of VISH

- Primary domain: scientific visualization
- Closing the gap between "one-shot" research algorithms and complex end-user convenient applications
  - Infrastructure powerful such as in established commercial applications
  - Not bound to a specific application or platform
    - A set of libraries, can be added to existing applications
    - Standalone reference application - qVISH

VISH Components

- Kernel with object management and (runtime) plugin mechanisms
- User Interface plugins (e.g., QT frontend)
- Data model (systematic treatment of scientific data via a common approach)
  - I/O layer as runtime plugins (file formats, streaming)
- Visualization infrastructure
  - OpenGL caching mechanisms

VISH Components

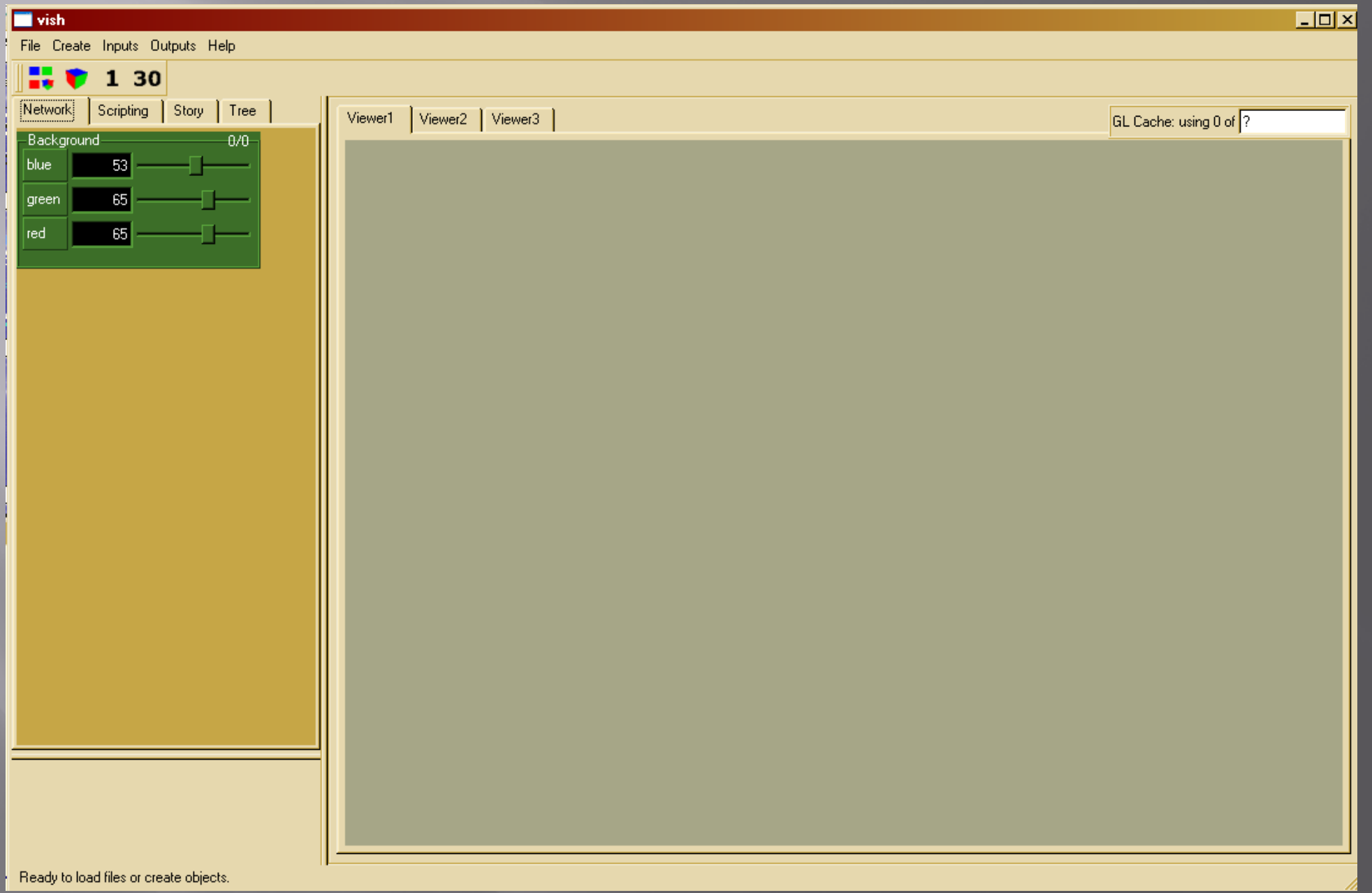


VISH Kernel: ocean

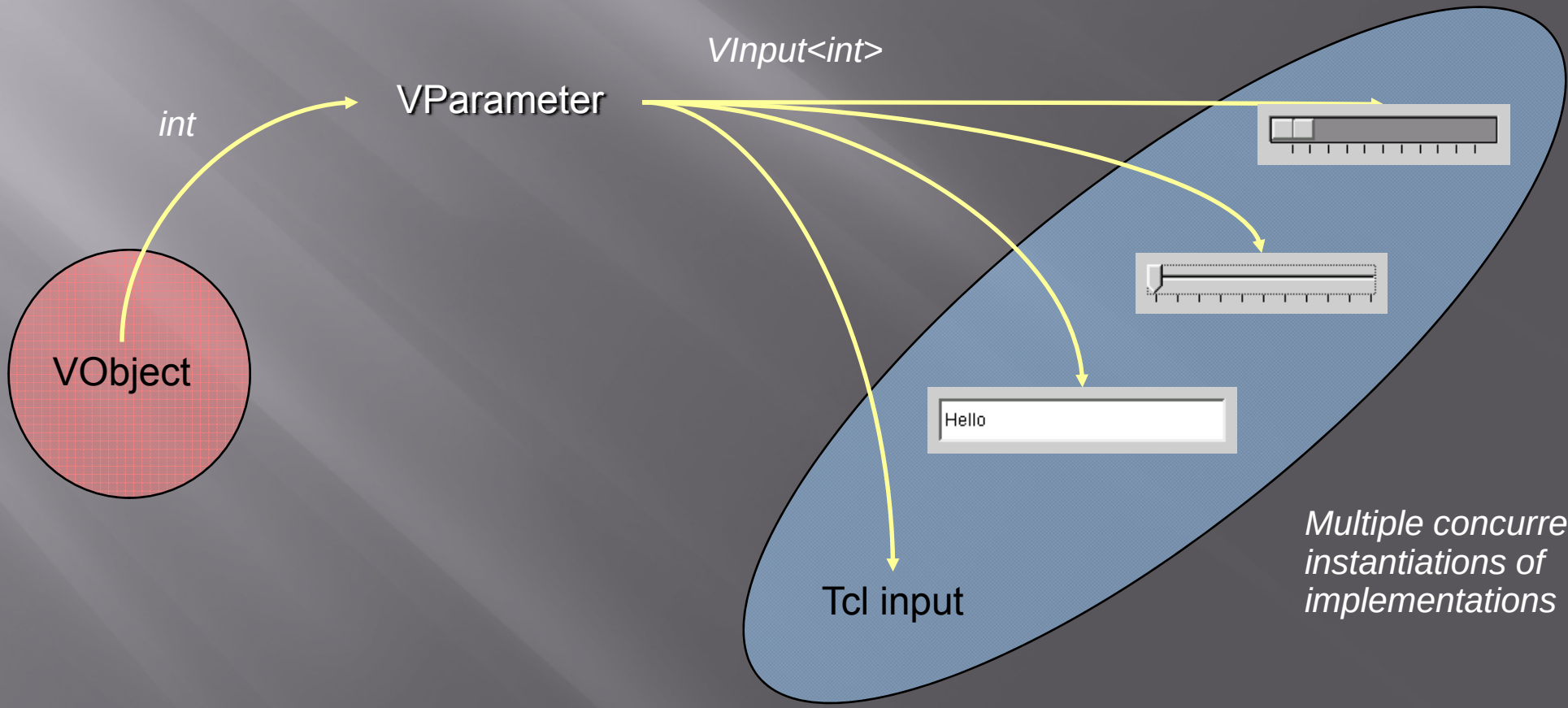
- Database-like kernel
  - Abstract objects with inputs and outputs
  - Runtime plugins based on C++ type query
  - Data and control flow management
- OpenGL support library
  - Layered rendering
  - OpenGL memory and multidimensional cache management

Background.cpp

```
#include <ocean/plankton/VCreator.hpp>
#include <ocean/GLvish/VRenderObject.hpp>

using namespace Wizt;

namespace
{

class   MyBackground : public VRenderObject
{
      // Input slots for a type "double"
      TypedSlot<double>      Red, Green, Blue;

      // The virtual callback functions for rendering
      override void render(VRenderContext&Context) const
      {
      // Local variables
      double red=1, green=1, blue=1, alpha=1;

            // Evaluate from input slots relative to Context
            // Note that this is special VISH syntax
            Red    << Context >> red;
            Green << Context >> green;
            Blue   << Context >> blue;

            // Do some OpenGL stuff here
            glClearColor( red,
                        green,
                        blue,
                        alpha );

            glClear(GL_COLOR_BUFFER_BIT);
      }

public:
      MyBackground(const string&name, int, const RefPtr<VCreationPreferences>&VP)
      : VRenderObject(name, BACKGROUND_OBJECT, VP)
      , Red    (this, "red"   , 0.5)
      , Green(this, "green", 0.5)
      , Blue   (this, "blue" , 0.7)
      {}
};

// Define creation object for the given object
static Ref<VCreator<MyBackground> > myBackground("Private/MyBackground");

}
```

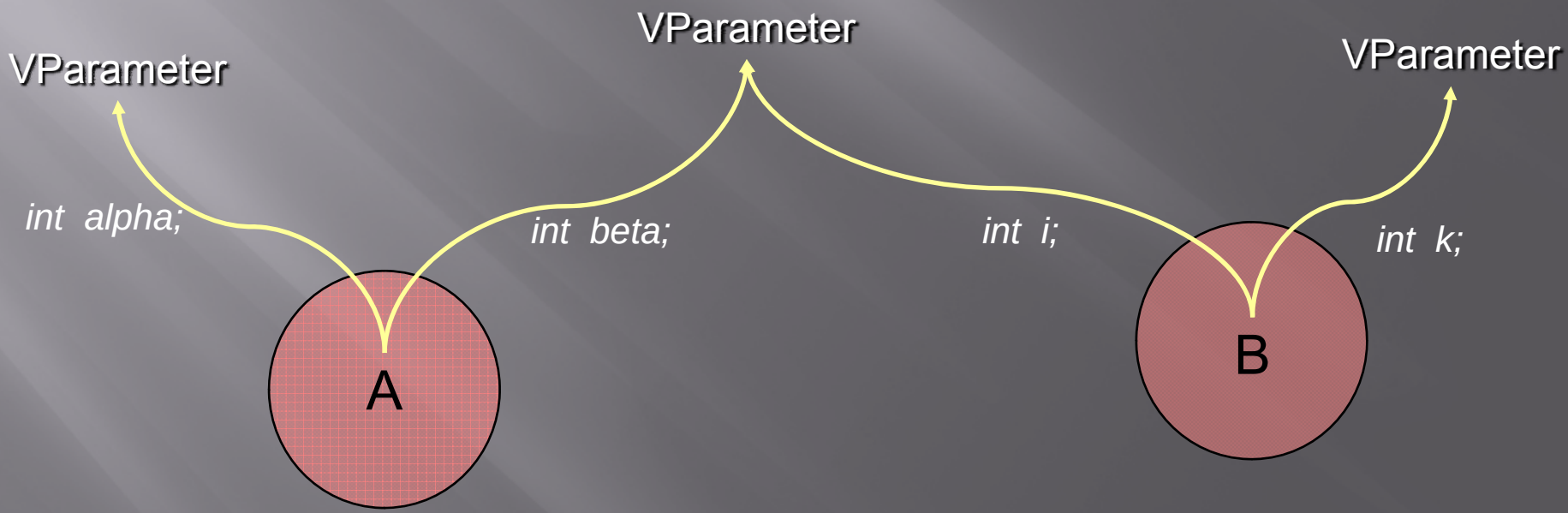

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
Background 0/0
blue 53
green 65
red 65
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
Ready to load files or create objects.

VObjects and Parameters
Multiple Inputs



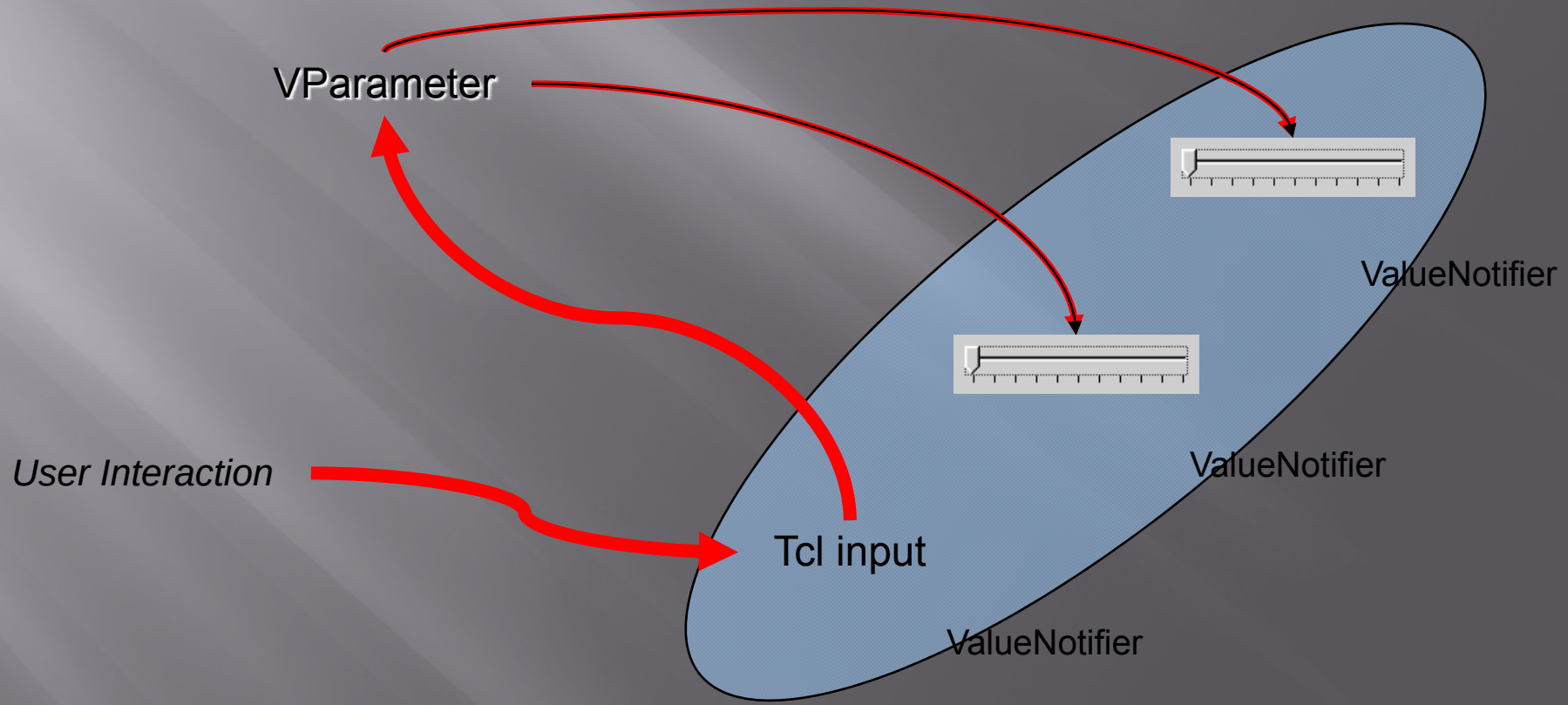
VObjects and Parameters
Shared Inputs

VObject “A” requests input parameter “alpha, beta”
VObject “B” requests input parameter “i,j”
Input **A.beta** is identical to **B.i**



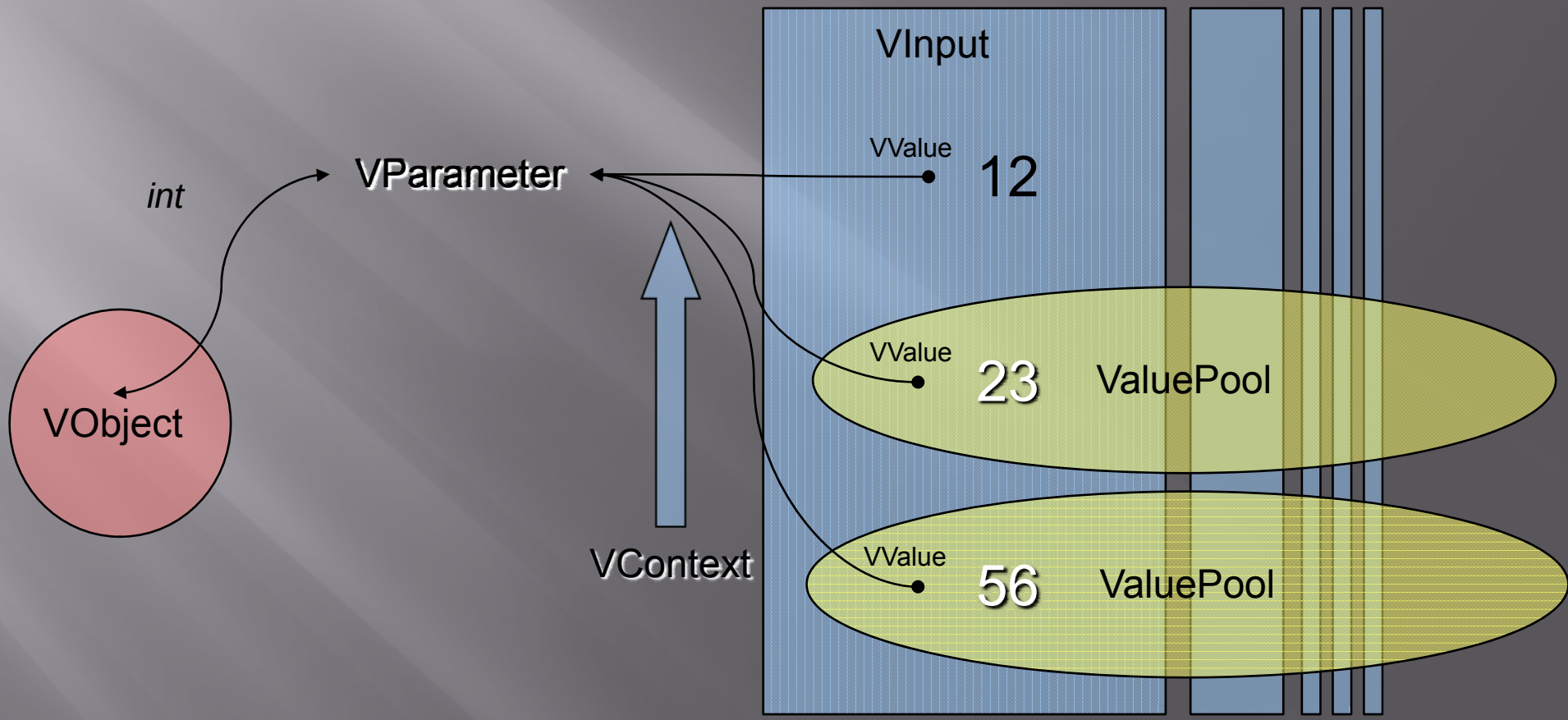
Parameter Notification

Changing one parameter notifies many values



Local Values of Variables

Value of input objects may be stored relative to a ValuePool



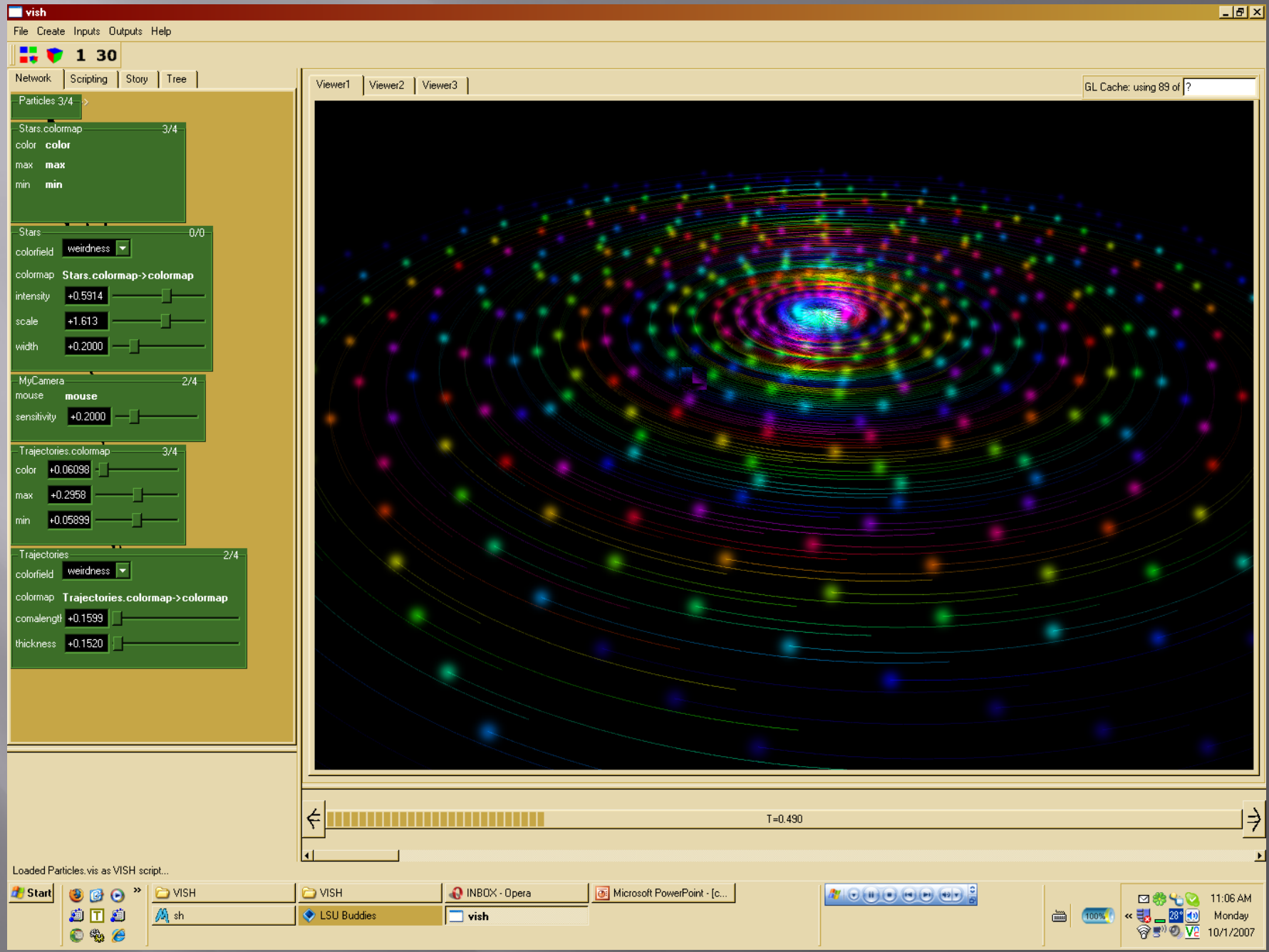
VISH GUI: qVISH

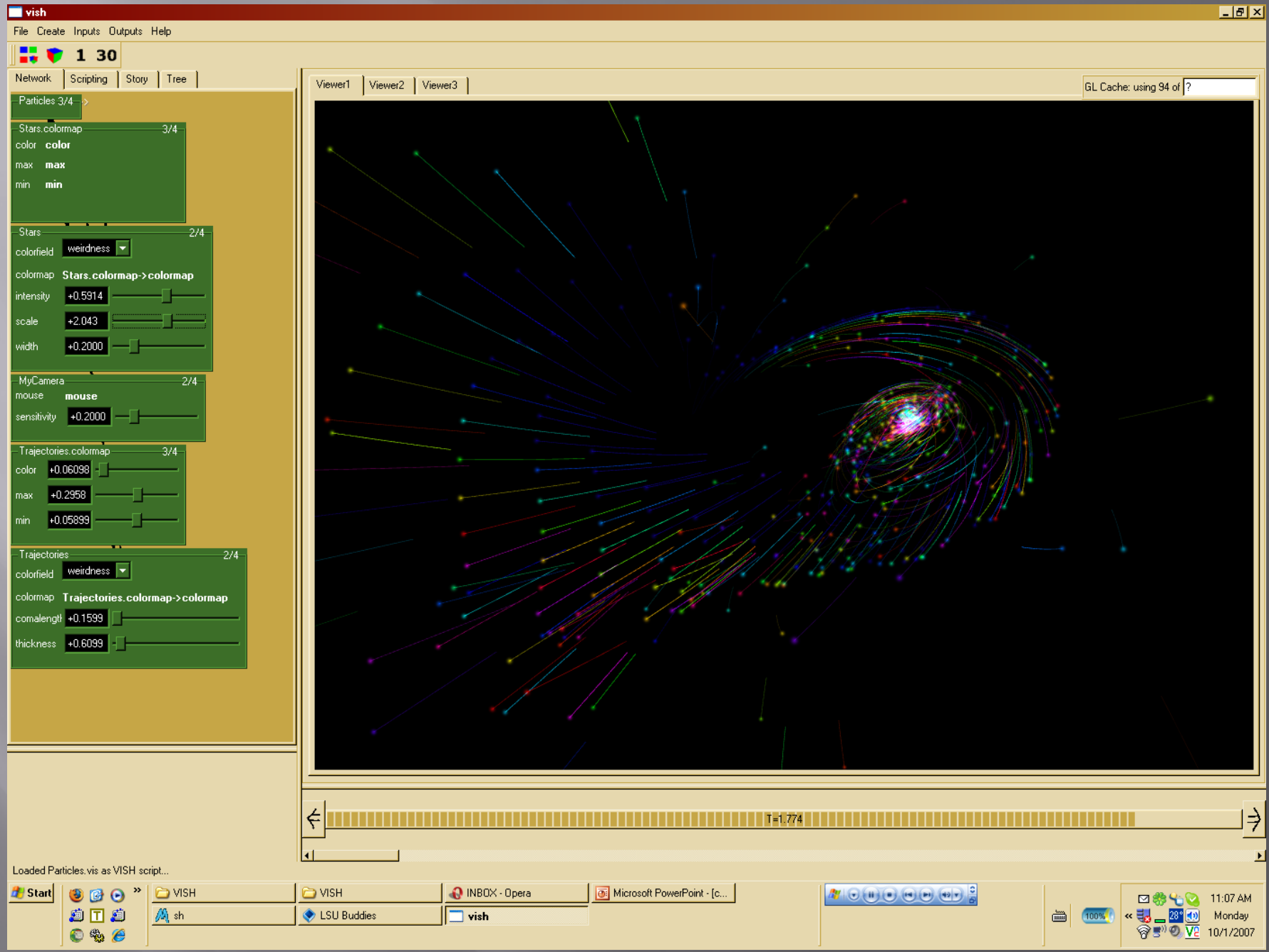
- Reference prototype implementation
  - based on QT (both qt3 and qt4 possible)
  - Plugin to the kernel
  - Consists of several plugins itself, such as
    - Network, scripting, tree, story representation
- Other GUI's possible: GTK, wxWindows, FLTK, none (batch mode)
- Interfacing existing applications possible:
  - E.g. run VISH within Amira, or Scirun, or AVS or …??

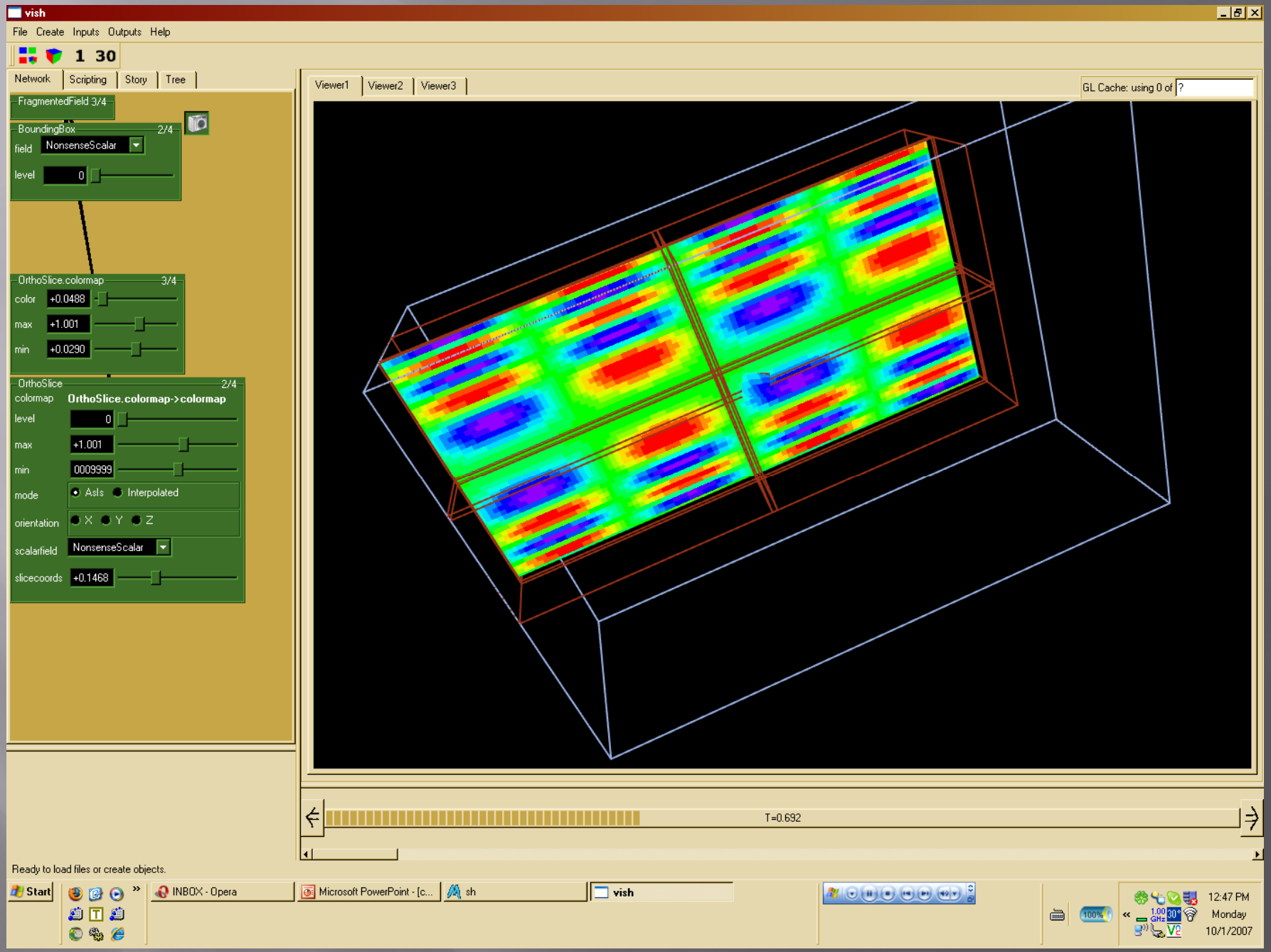
GUI DEMO

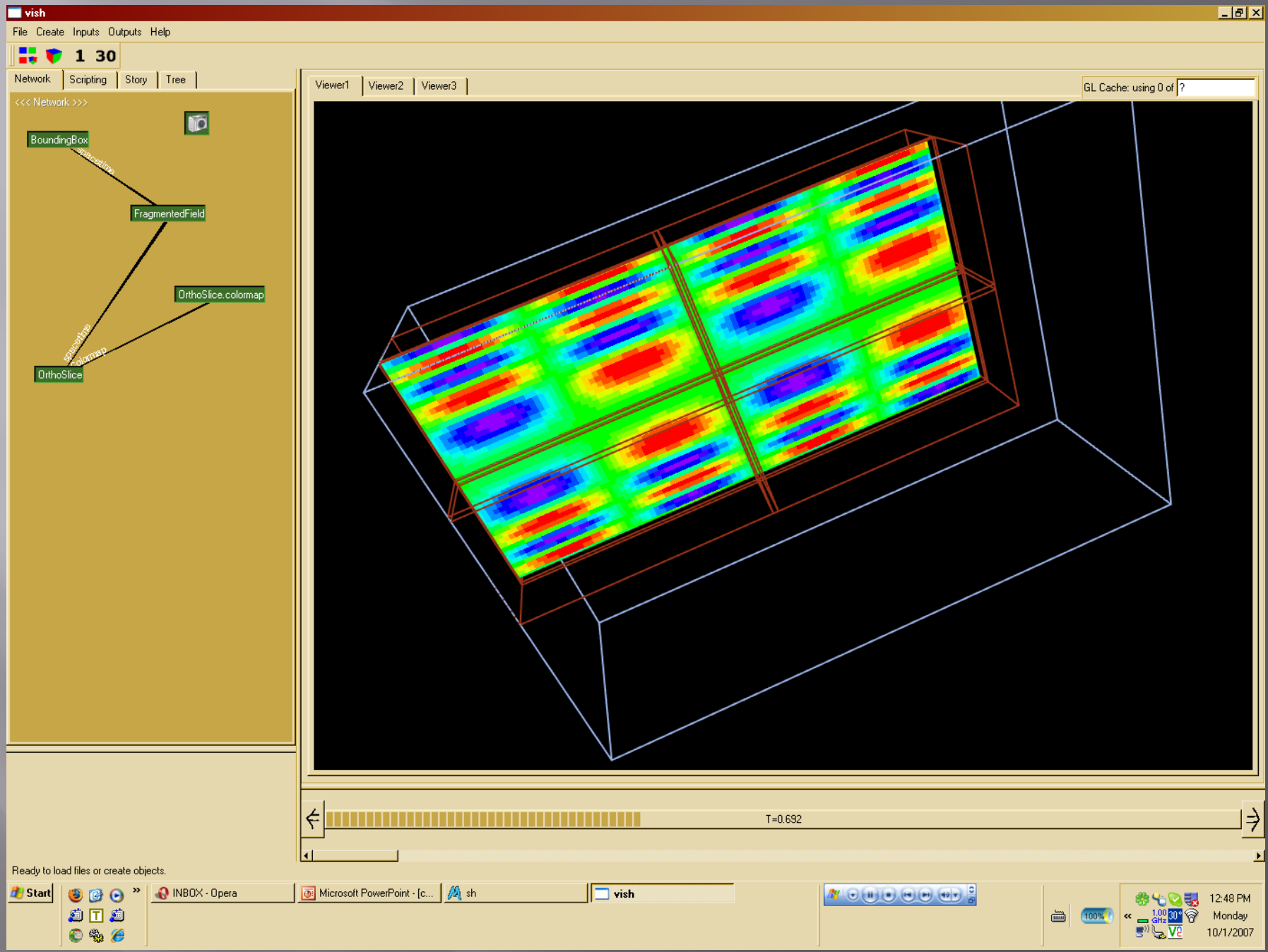
VISH Status

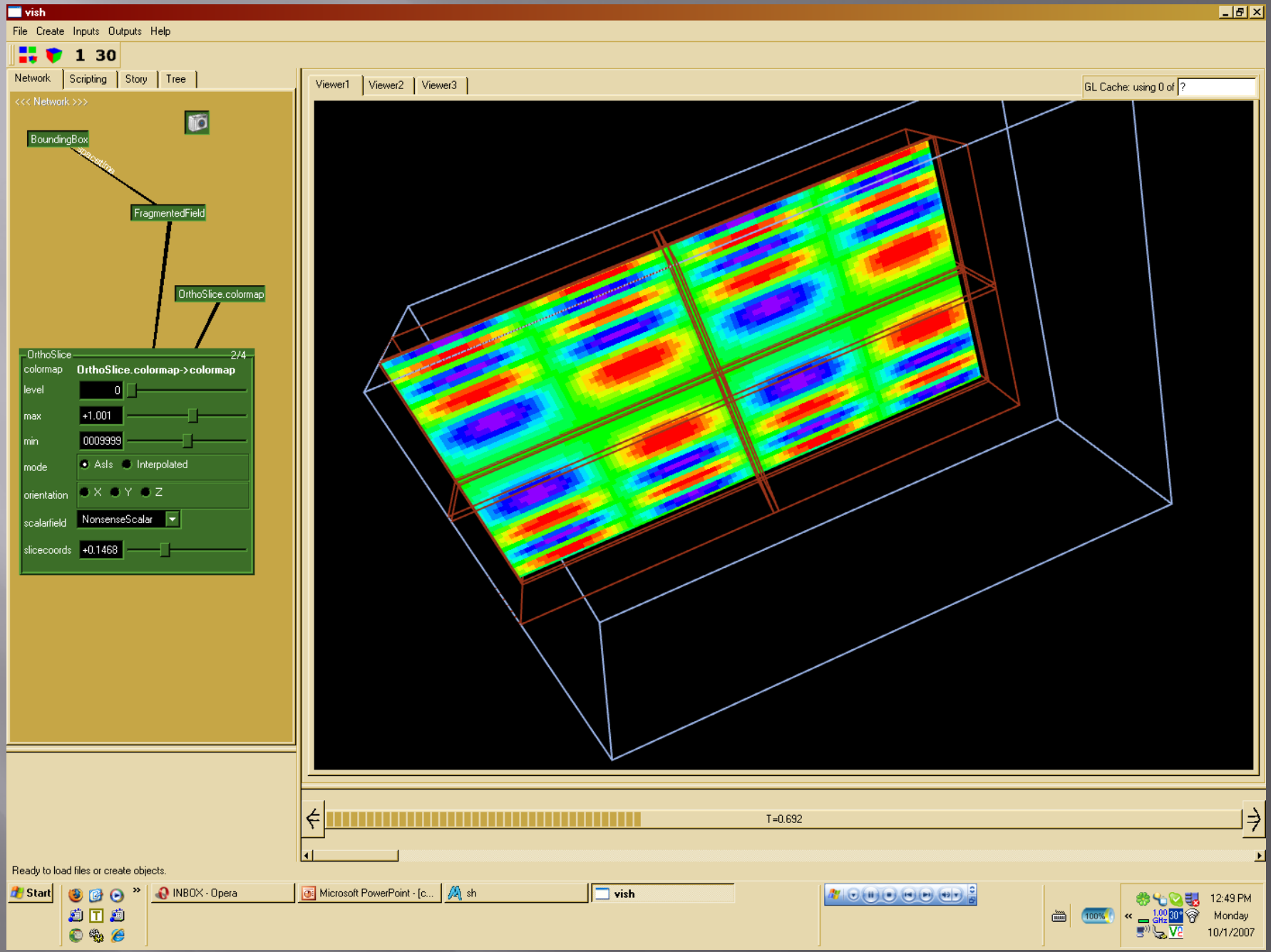
- Screenshots!

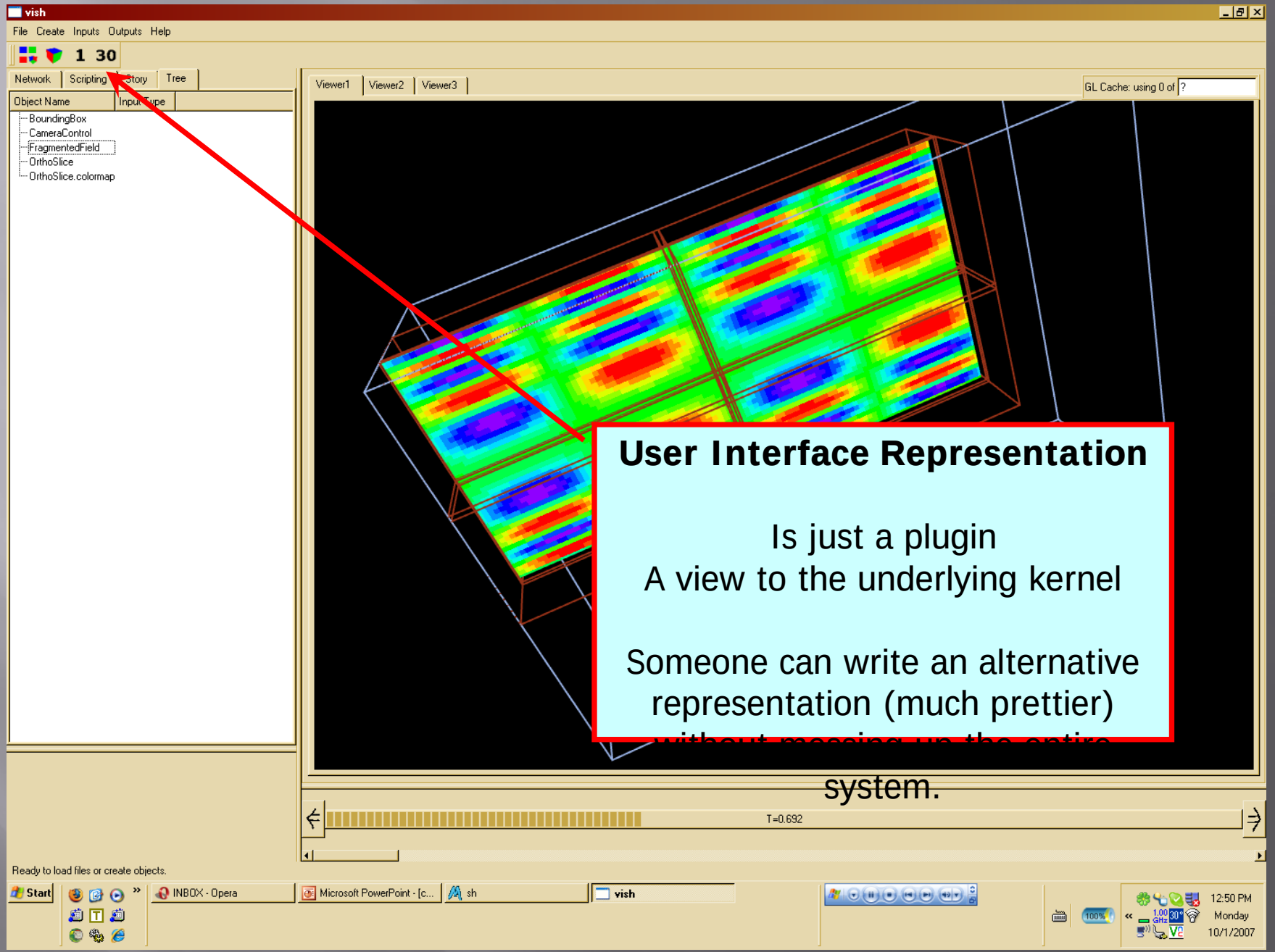

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
Particles 3/4
Stars.colormap 3/4
color color
max max
min min
Stars 0/0
colorfield weirdness
colormap Stars.colormap->colormap
intensity +0.5914
scale +1.613
width +0.2000
MyCamera 2/4
mouse mouse
sensitivity +0.2000
Trajectories.colormap 3/4
color +0.06098
max +0.2958
min +0.05899
Trajectories 2/4
colorfield weirdness
colormap Trajectories.colormap->colormap
comalength +0.1599
thickness +0.1520
Viewer1 Viewer2 Viewer3
GL Cache: using 89 of ?
T=0.490
Loaded Particles.vis as VISH script...
Start
VISH
VISH
INBOX - Opera
Microsoft PowerPoint - [c...
sh
LSU Buddies
vish
11:06 AM
Monday
10/1/2007

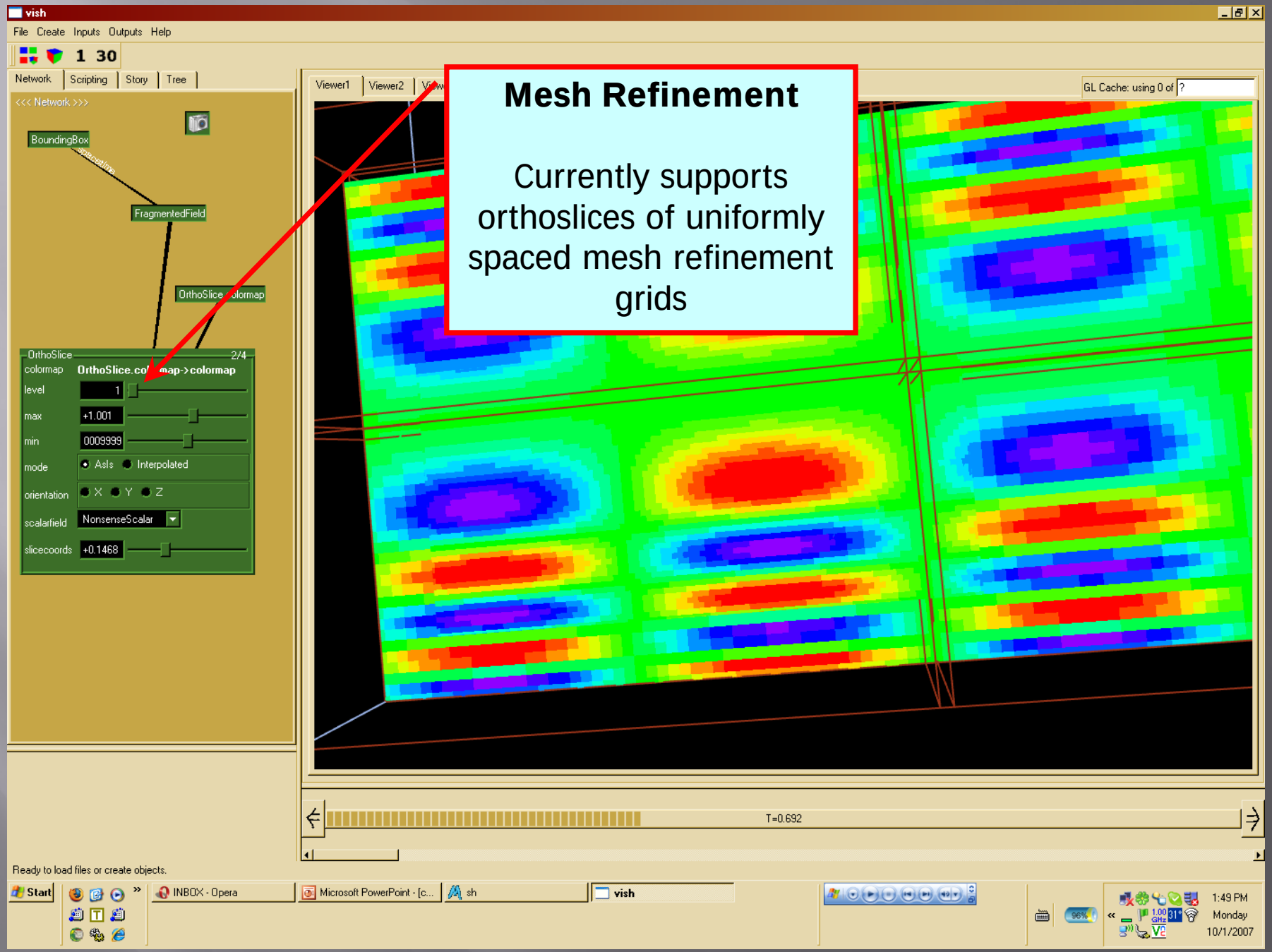

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
Particles 3/4
Stars.colormap 3/4
color color
max max
min min
Stars 2/4
colorfield weirdness
colormap Stars.colormap->colormap
intensity +0.5914
scale +2.043
width +0.2000
MyCamera 2/4
mouse mouse
sensitivity +0.2000
Trajectories.colormap 3/4
color +0.06098
max +0.2958
min +0.05899
Trajectories 2/4
colorfield weirdness
colormap Trajectories.colormap->colormap
comalength +0.1599
thickness +0.6099
Viewer1 Viewer2 Viewer3
GL Cache: using 94 of ?
T=1.774
Loaded Particles.vis as VISH script...
Start
VISH
VISH
INBOX - Opera
Microsoft PowerPoint - [c...
sh
LSU Buddies
vish
11:07 AM
Monday
10/1/2007

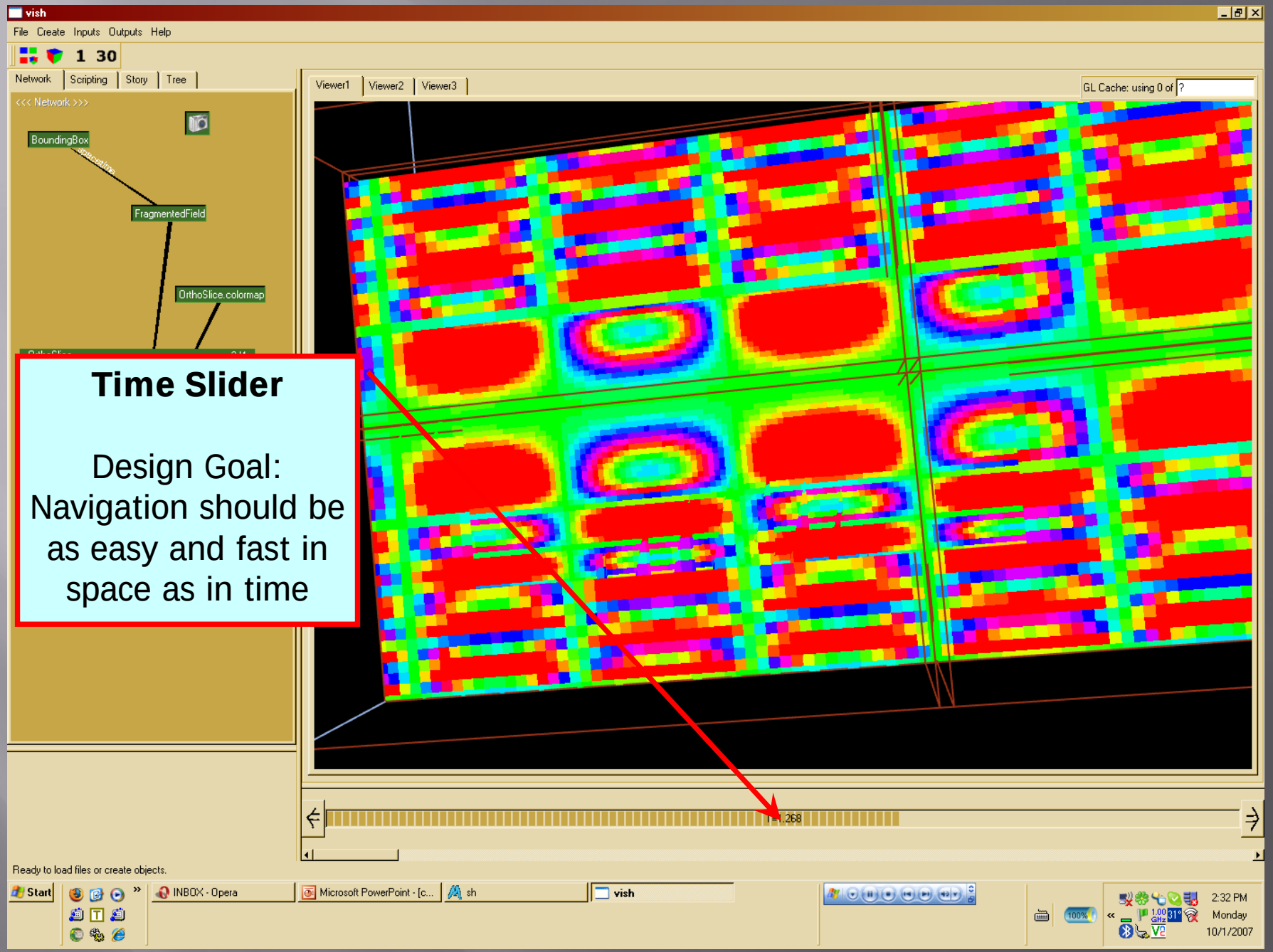

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
FragmentedField 3/4
BoundingBox 2/4
field NonsenseScalar
level 0
OrthoSlice.colormap 3/4
color +0.0488
max +1.001
min +0.0290
OrthoSlice 2/4
colormap OrthoSlice.colormap->colormap
level 0
max +1.001
min 0009999
mode Asis Interpolated
orientation X Y Z
scalarfield NonsenseScalar
slicecoords +0.1468
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
T=0.692
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
12:47 PM
Monday
10/1/2007

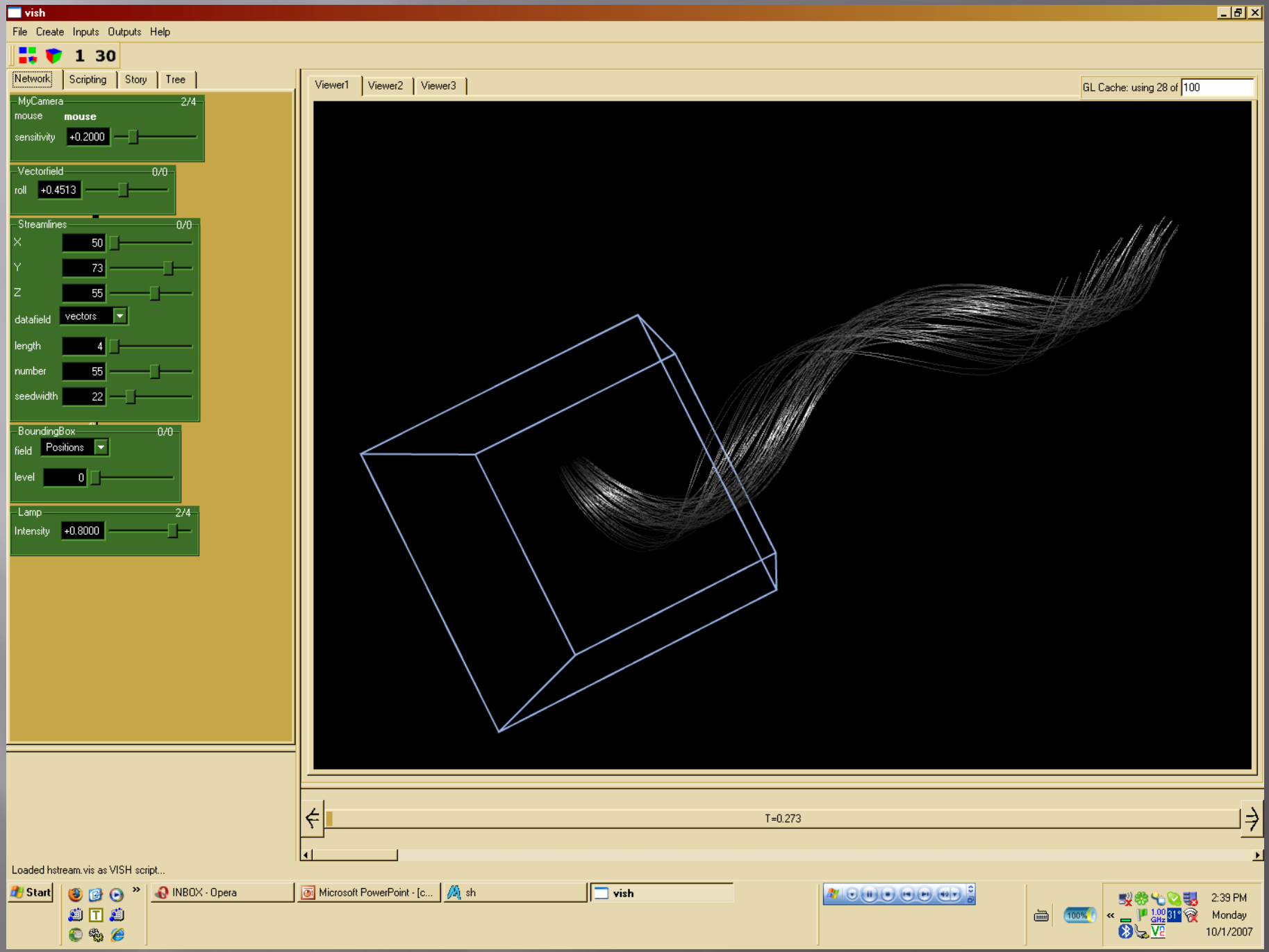

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
<<< Network >>>
BoundingBox
spacetime
FragmentedField
OrthoSlice.colormap
spacetime
colormap
OrthoSlice
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
T=0.692
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
12:48 PM
Monday
10/1/2007

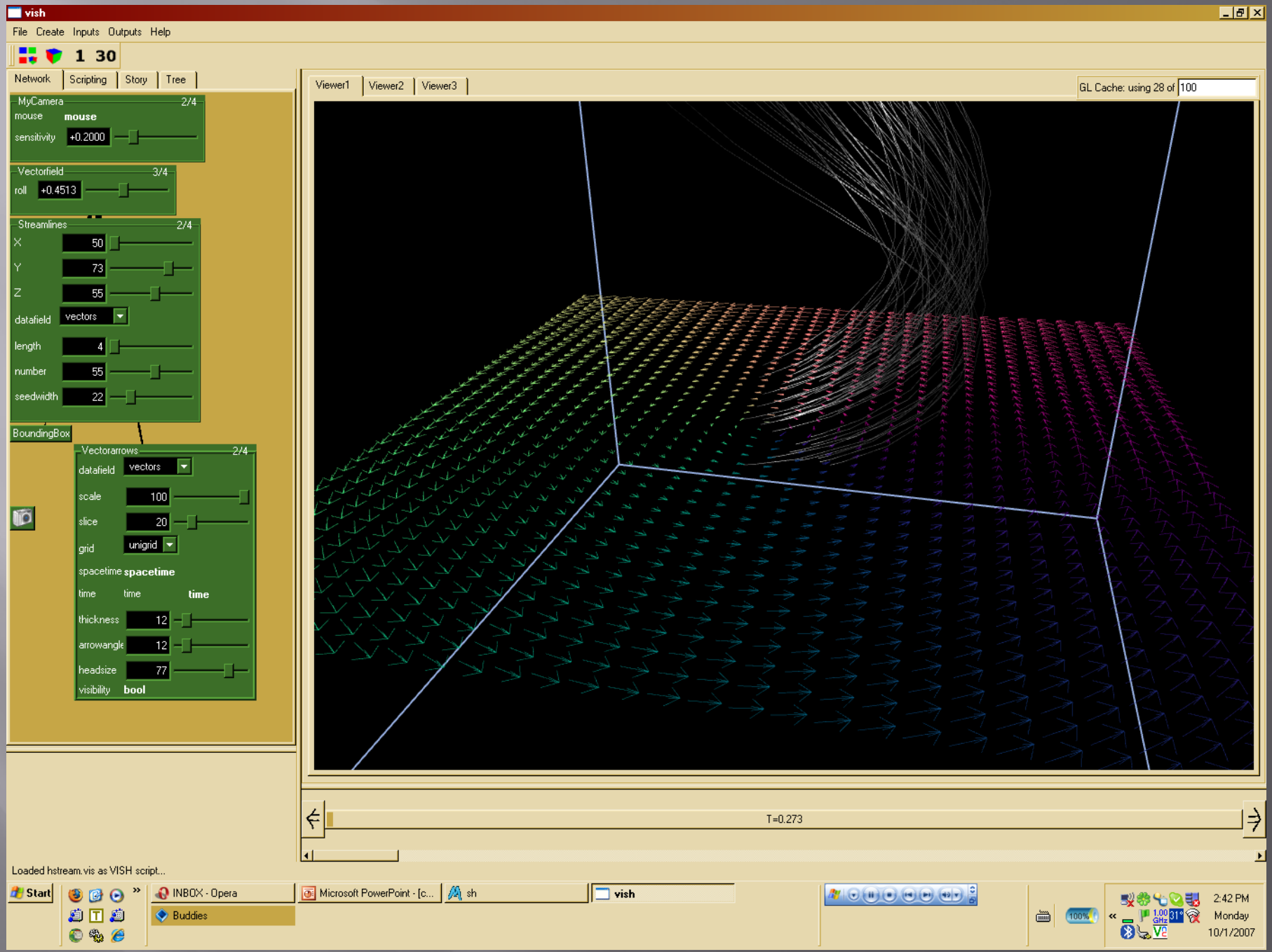

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
<<< Network >>>
BoundingBox
spacestime
FragmentedField
OrthoSlice.colormap
OrthoSlice 2/4
colormap OrthoSlice.colormap->colormap
level 0
max +1.001
min 0009999
mode Asis Interpolated
orientation X Y Z
scalarfield NonsenseScalar
slicecoords +0.1468
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
T=0.692
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
12:49 PM
Monday
10/1/2007


vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
Object Name Input Type
BoundingBox
CameraControl
FragmentedField
OrthoSlice
OrthoSlice.colormap
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
User Interface Representation
Is just a plugin
A view to the underlying kernel
Someone can write an alternative representation (much prettier) without messing up the entire system.
T=0.692
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
12:50 PM
Monday
10/1/2007


vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
<<< Network >>>
BoundingBox
FragmentedField
OrthoSlice
OrthoSlice
colormap
level
max
min
mode
orientation
scalarfield
slicecoords
Asis
Interpolated
X Y Z
NonsenseScalar
+0.1468
Viewer1
Viewer2
GL Cache: using 0 of
Mesh Refinement
Currently supports orthoslices of uniformly spaced mesh refinement grids
T=0.692
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
1:49 PM
Monday
10/1/2007

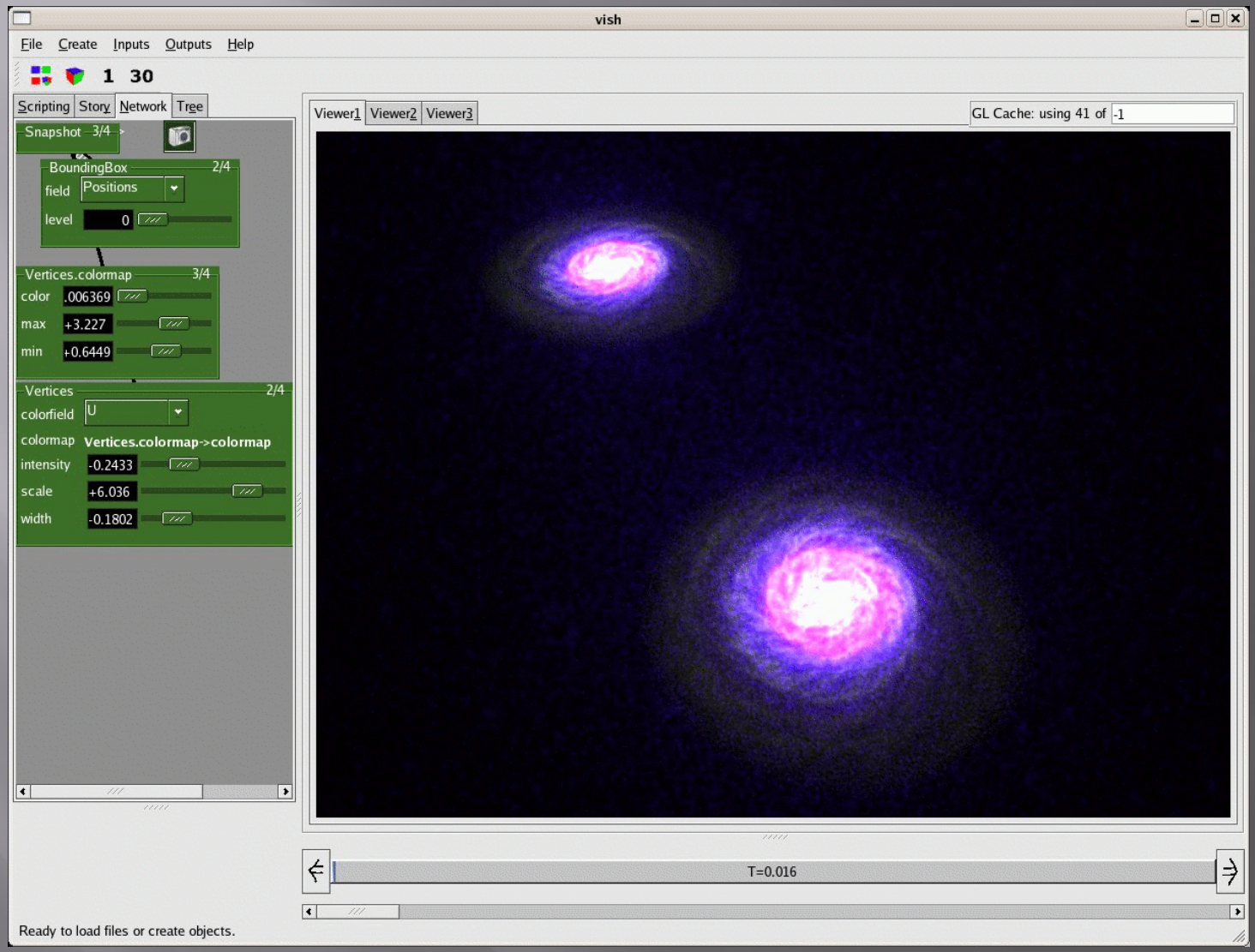

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
<<< Network >>>
BoundingBox
FragmentedField
OrthoSlice.colormap
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
Time Slider
Design Goal:
Navigation should be as easy and fast in space as in time
T = 1.268
Ready to load files or create objects.
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
2:32 PM
Monday
10/1/2007


vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
MyCamera 2/4
mouse mouse
sensitivity +0.2000
Vectorfield 0/0
roll +0.4513
Streamlines 0/0
X 50
Y 73
Z 55
datafield vectors
length 4
number 55
seedwidth 22
BoundingBox 0/0
field Positions
level 0
Lamp 2/4
Intensity +0.8000
Viewer1 Viewer2 Viewer3
GL Cache: using 28 of 100
T=0.273
Loaded hstream.vis as VISH script...
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
2:39 PM
Monday
10/1/2007

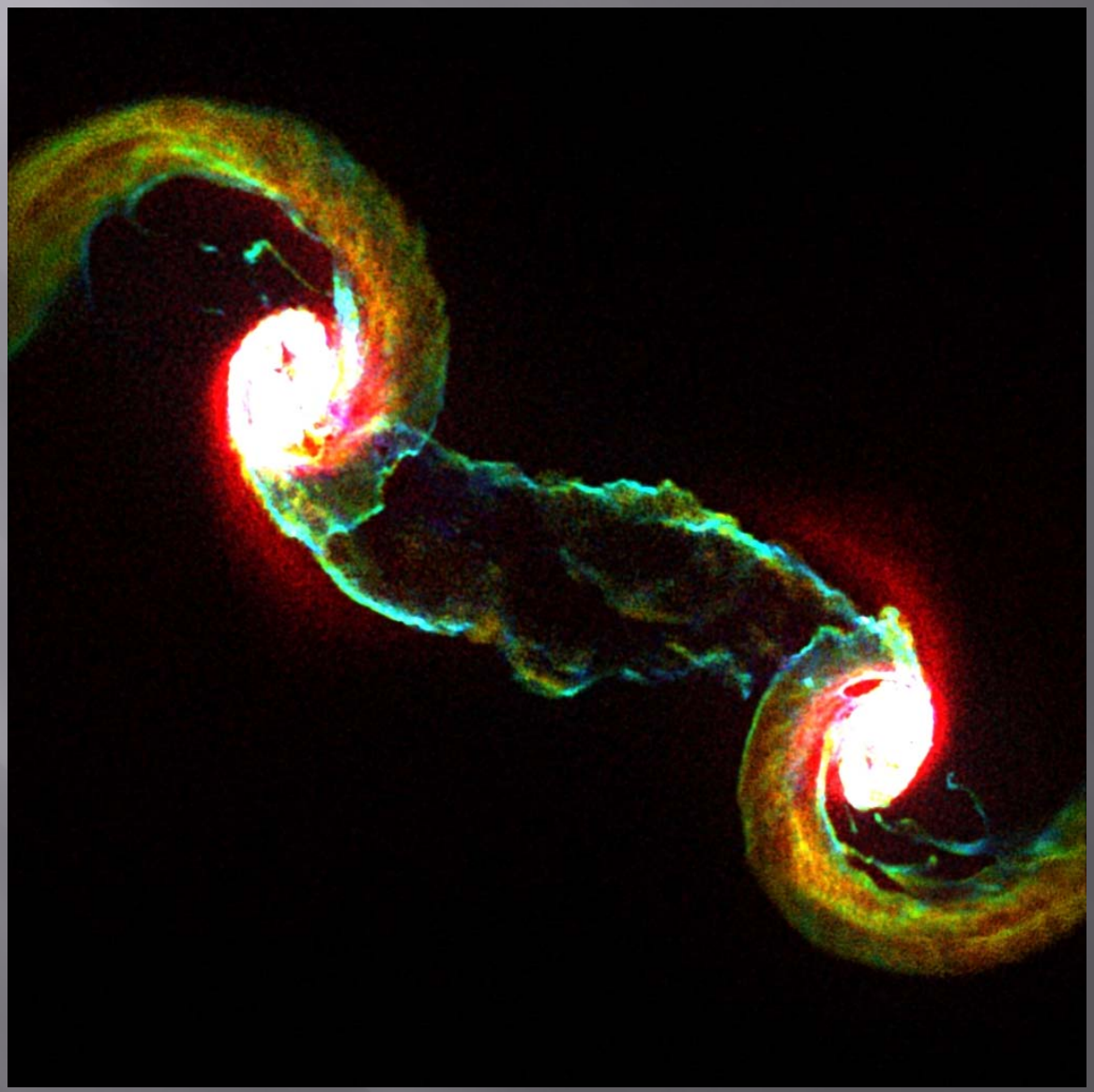

vish
File Create Inputs Outputs Help
1 30
Network Scripting Story Tree
MyCamera 2/4
mouse mouse
sensitivity +0.2000
Vectorfield 3/4
roll +0.4513
Streamlines 2/4
X 50
Y 73
Z 55
datafield vectors
length 4
number 55
seedwidth 22
BoundingBox
Vectorarrows 2/4
datafield vectors
scale 100
slice 20
grid unigrid
spacetime spacetime
time time time
thickness 12
arrowangle 12
headsize 77
visibility bool
Viewer1 Viewer2 Viewer3
GL Cache: using 28 of 100
T=0.273
Loaded hstream.vis as VISH script...
Start
INBOX - Opera
Microsoft PowerPoint - [c...
sh
vish
Buddies
2:42 PM
Monday
10/1/2007

SPH Particle Data set

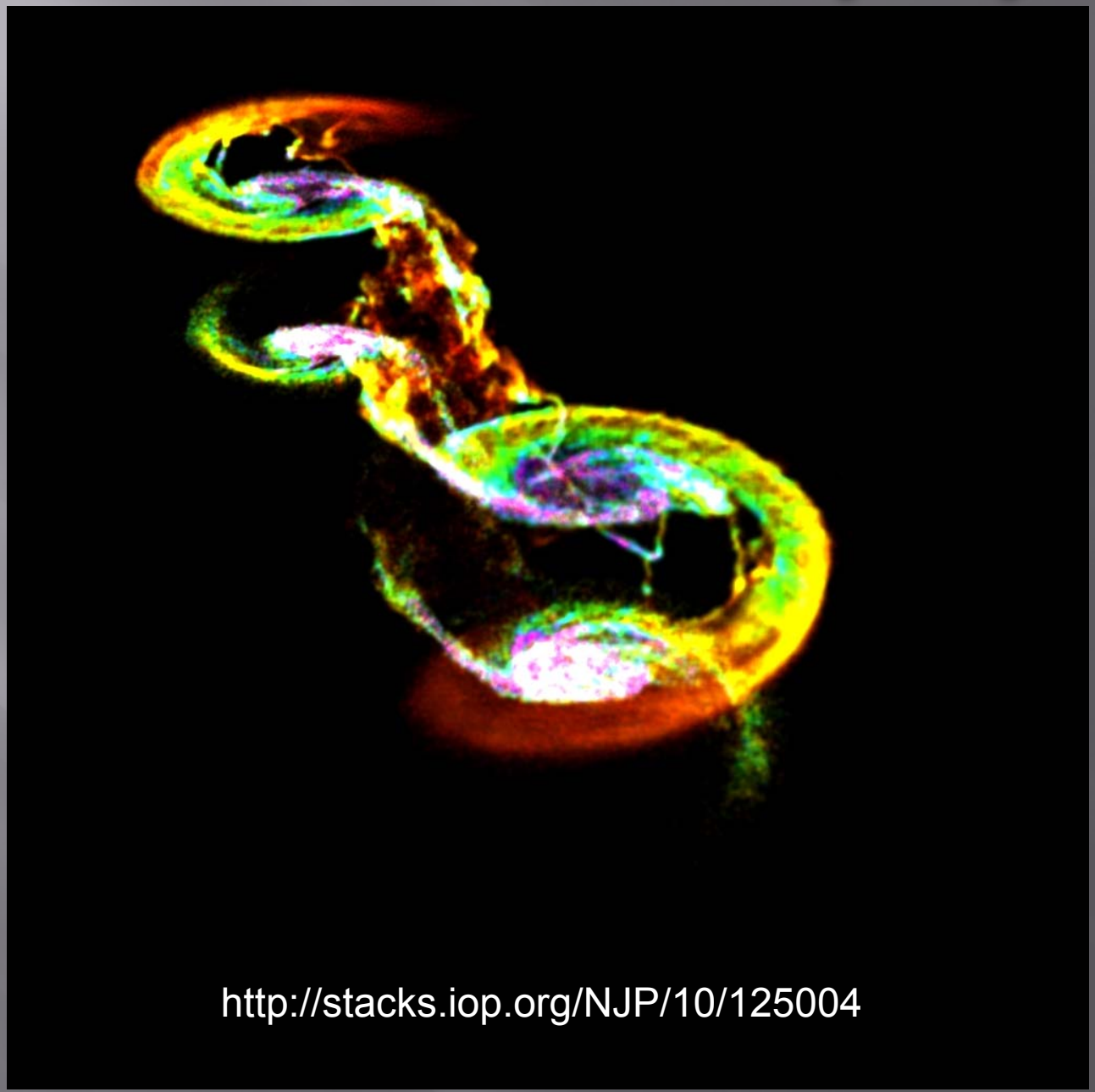
(2.6mio particles, interactive)



Particles Colored by Fields

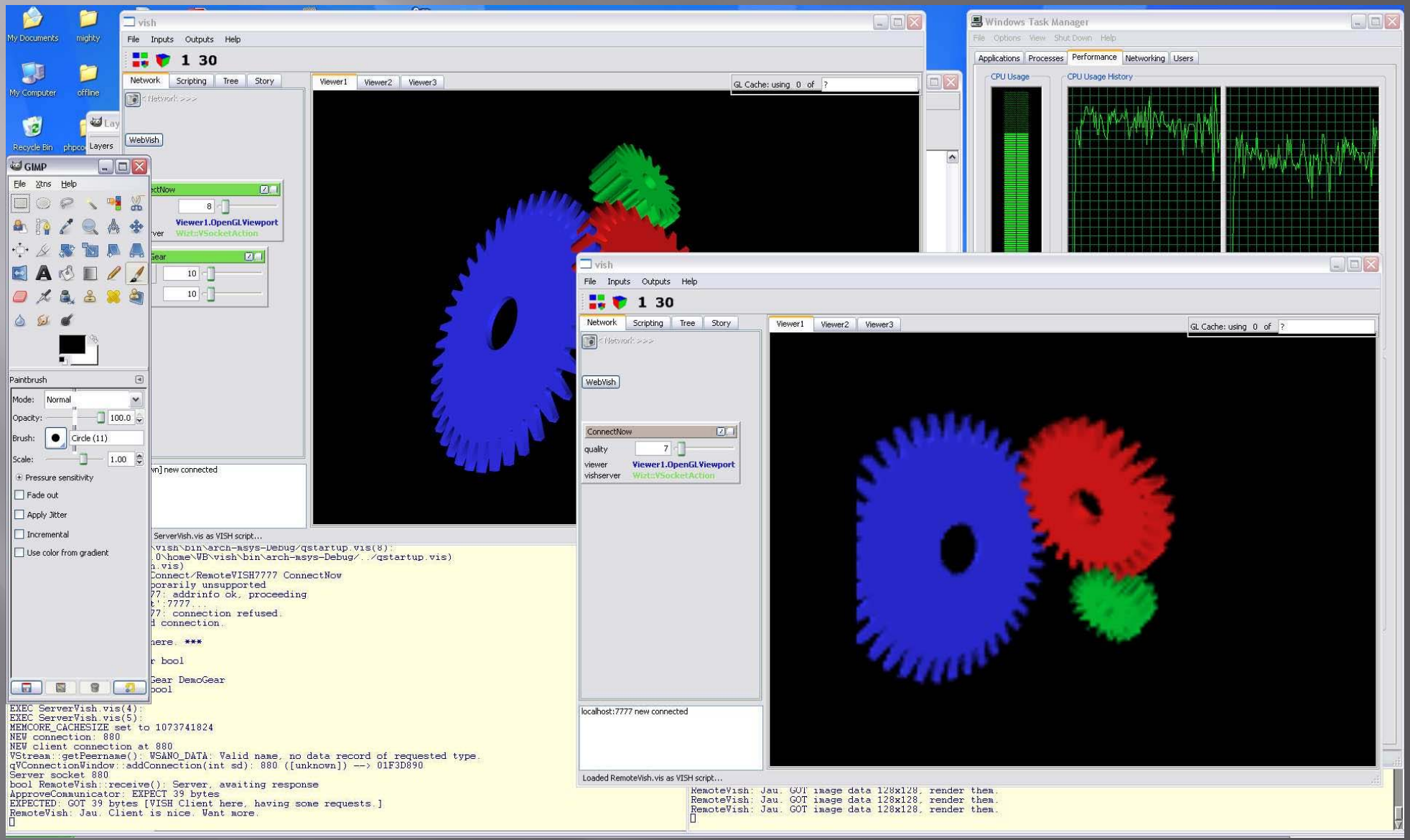


Multifield - Display

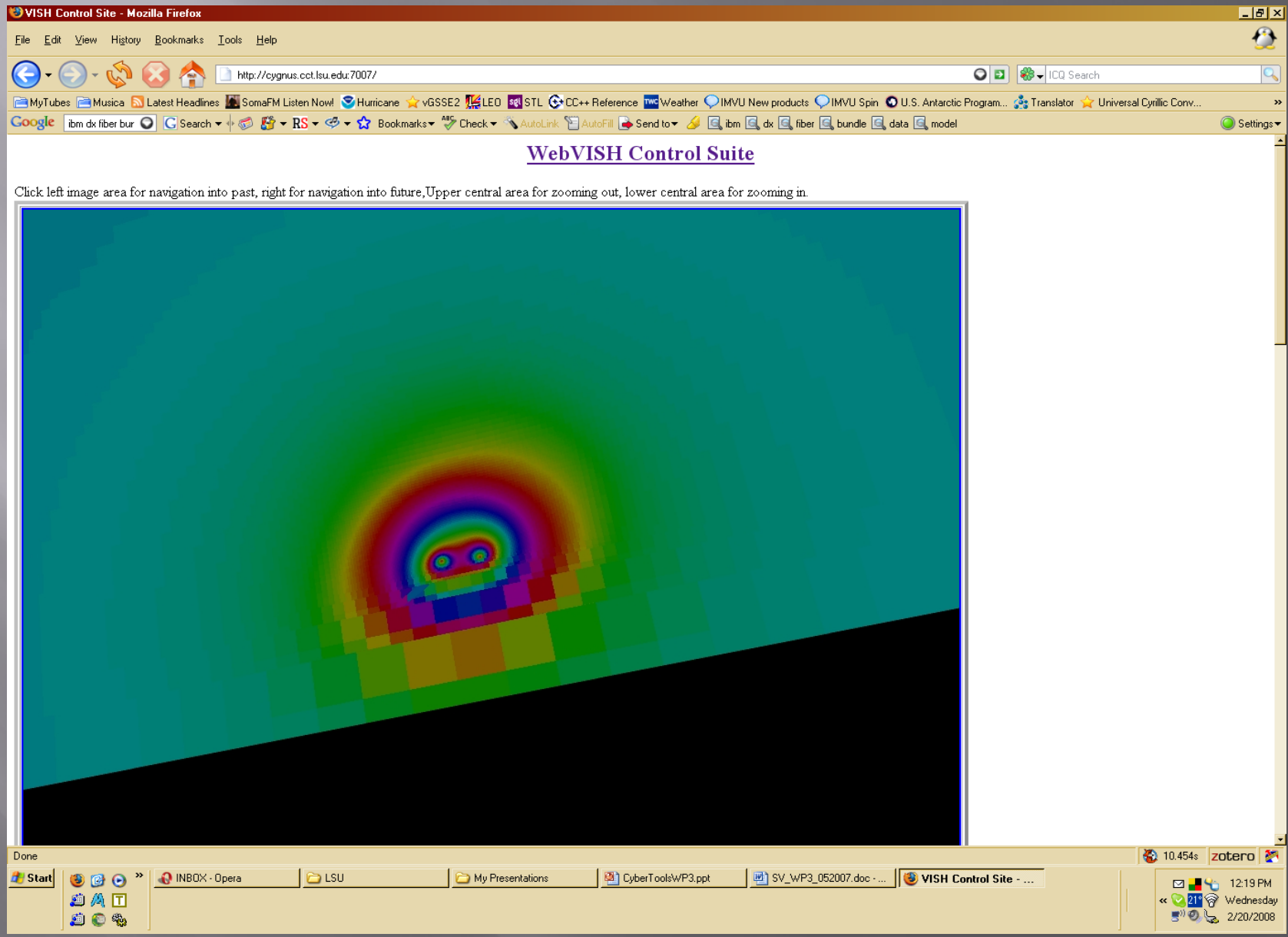


http://stacks.iop.org/NJP/10/125004

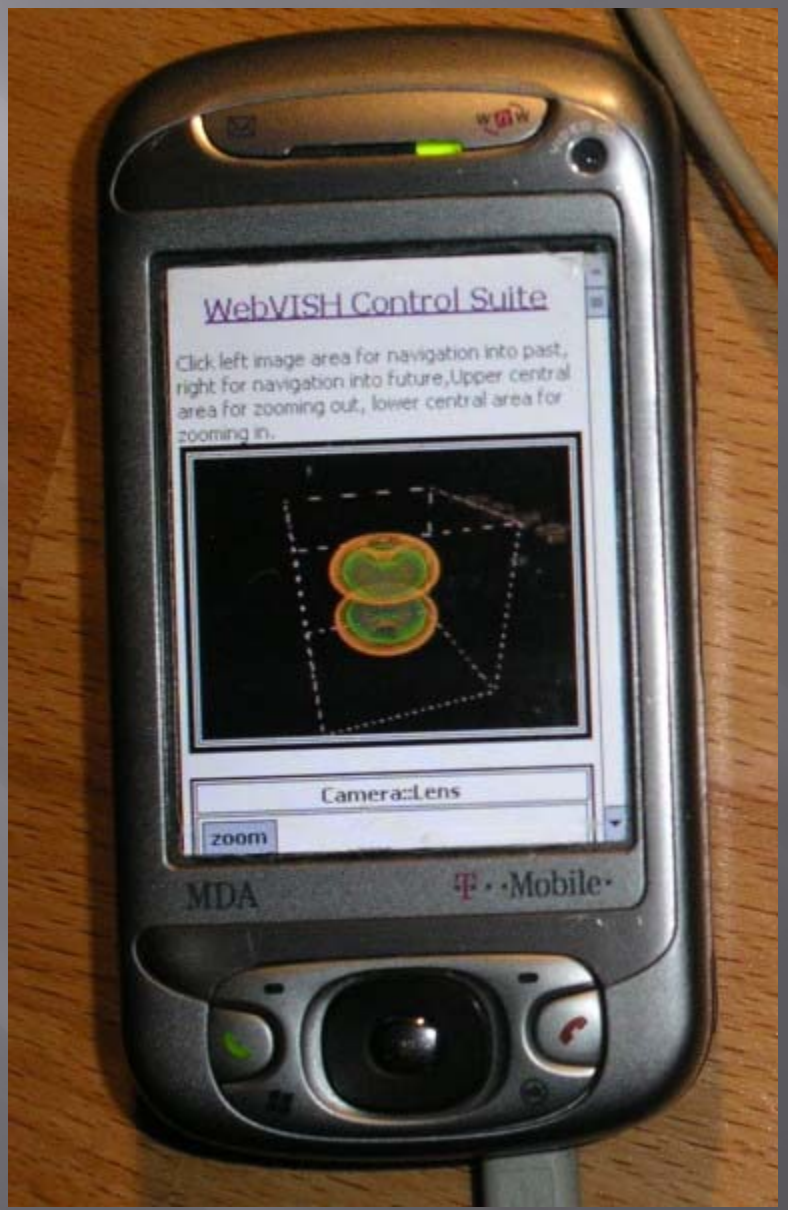
Remote Vish


vish
File Inputs Outputs Help
1 30
Network Scripting Tree Story
Viewer1 Viewer2 Viewer3
GL Cache: using 0 of ?
WebVish
GIMP
File Xtns Help
Paintbrush
Mode: Normal
Opacity: 100.0
Brush: Circle (11)
Scale: 1.00
Pressure sensitivity
Fade out
Apply Jitter
Incremental
Use color from gradient
ConnectNow
Viewer1.OpenGLViewport
Wizt::VSocketAction
Windows Task Manager
File Options View Shut Down Help
Applications Processes Performance Networking Users
CPU Usage
CPU Usage History
quality
viewer
vishserver
localhost:7777 new connected
Loaded RemoteVish.vis as VISH script...
EXEC ServerVish.vis(4):
EXEC ServerVish.vis(5):
MEMCORE_CACHESIZE set to 1073741824
NEW connection: 880
NEW client connection at 880
VStream::getPeername(): WSANO_DATA: Valid name, no data record of requested type.
qVConnectionWindow::addConnection(int sd): 880 ([unknown]) --> 01F3D890
Server socket 880
bool RemoteVish::receive(): Server, awaiting response
ApproveCommunicator: EXPECT 39 bytes
EXPECTED: GOT 39 bytes [VISH Client here, having some requests.]
RemoteVish: Jau. Client is nice. Want more.
RemoteVish: Jau. GOT image data 128x128. render them.
RemoteVish: Jau. GOT image data 128x128. render them.
RemoteVish: Jau. GOT image data 128x128. render them.

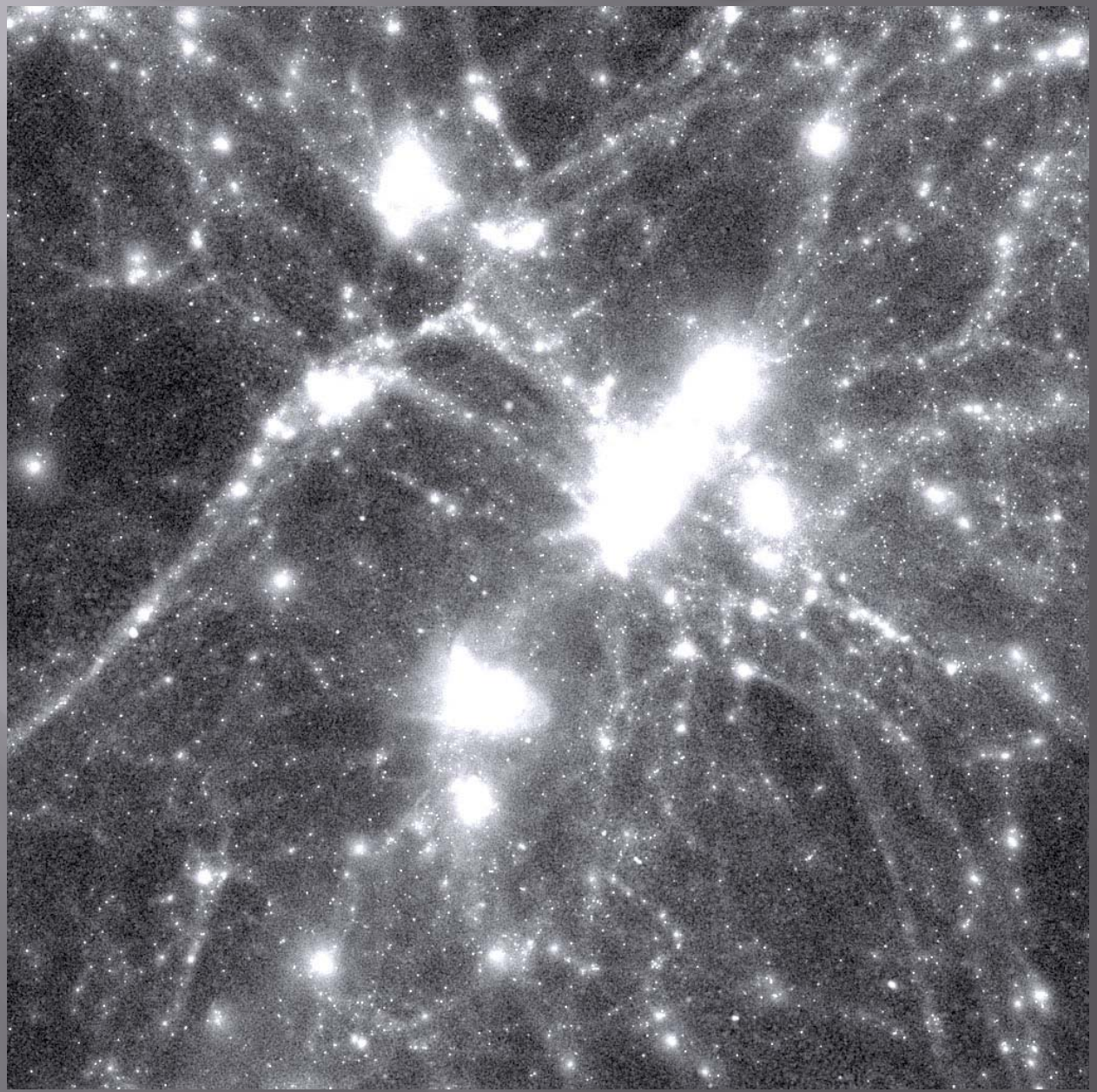
WebVISH


VISH Control Site - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://cygnus.cct.lsu.edu:7007/
WebVISH Control Suite
Click left image area for navigation into past, right for navigation into future,Upper central area for zooming out, lower central area for zooming in.
Done
Start
INBOX - Opera
LSU
My Presentations
CyberToolsWP3.ppt
SV_WP3_052007.doc - ...
VISH Control Site - ...
12:19 PM
Wednesday
2/20/2008

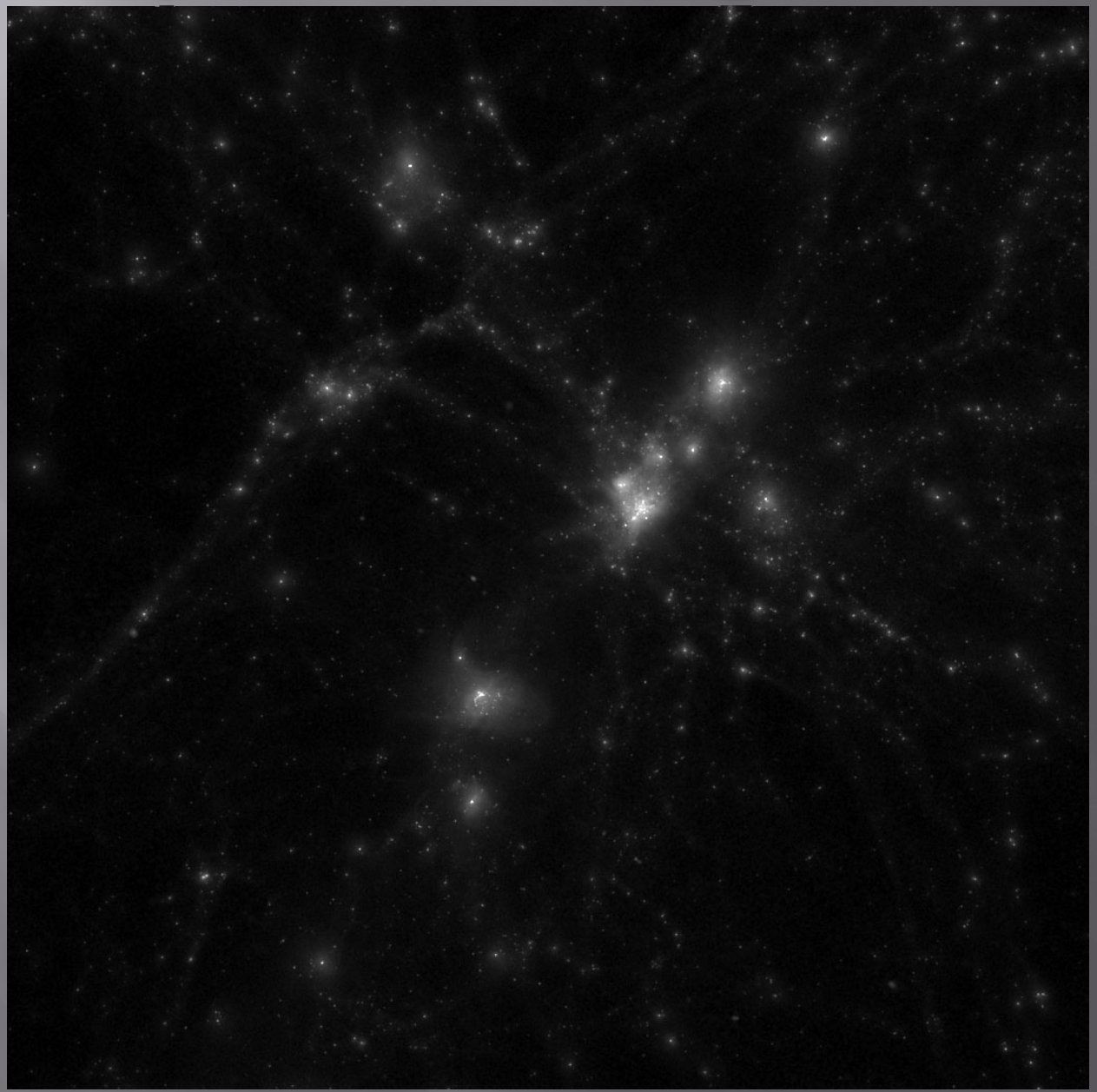
Remote Access



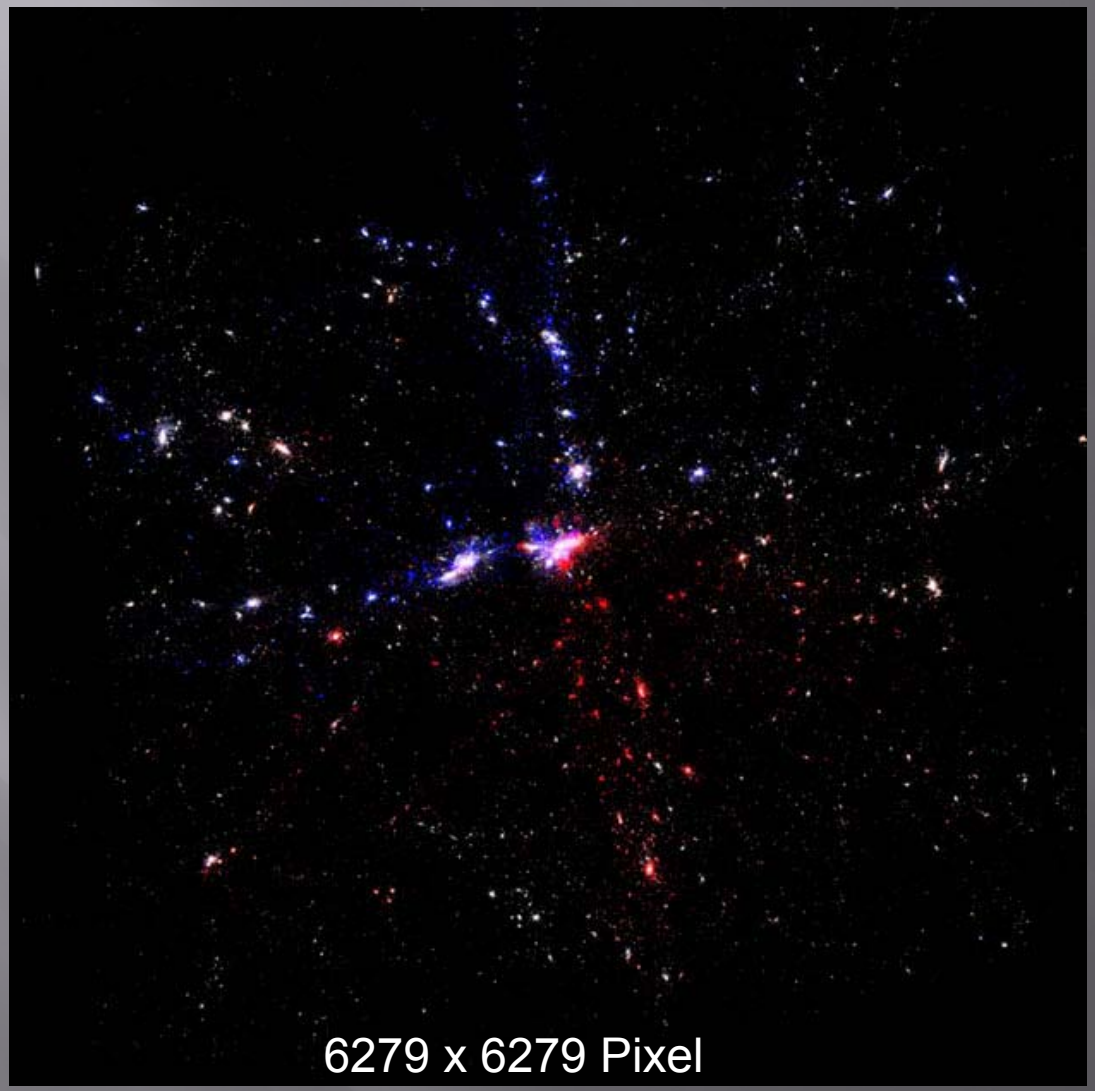
High Dynamic Range Rendering



High Dynamic Range Rendering

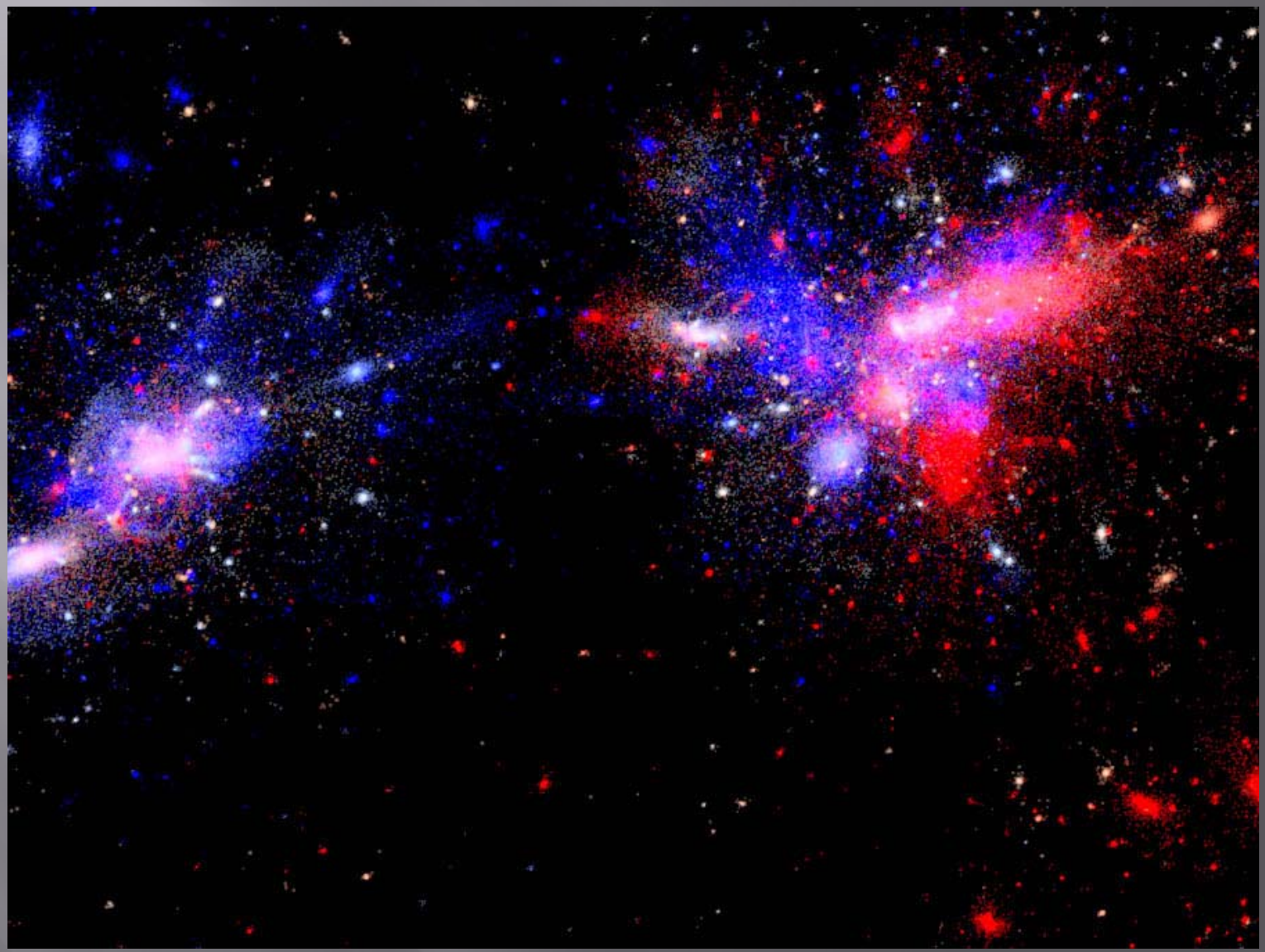


High Resolution

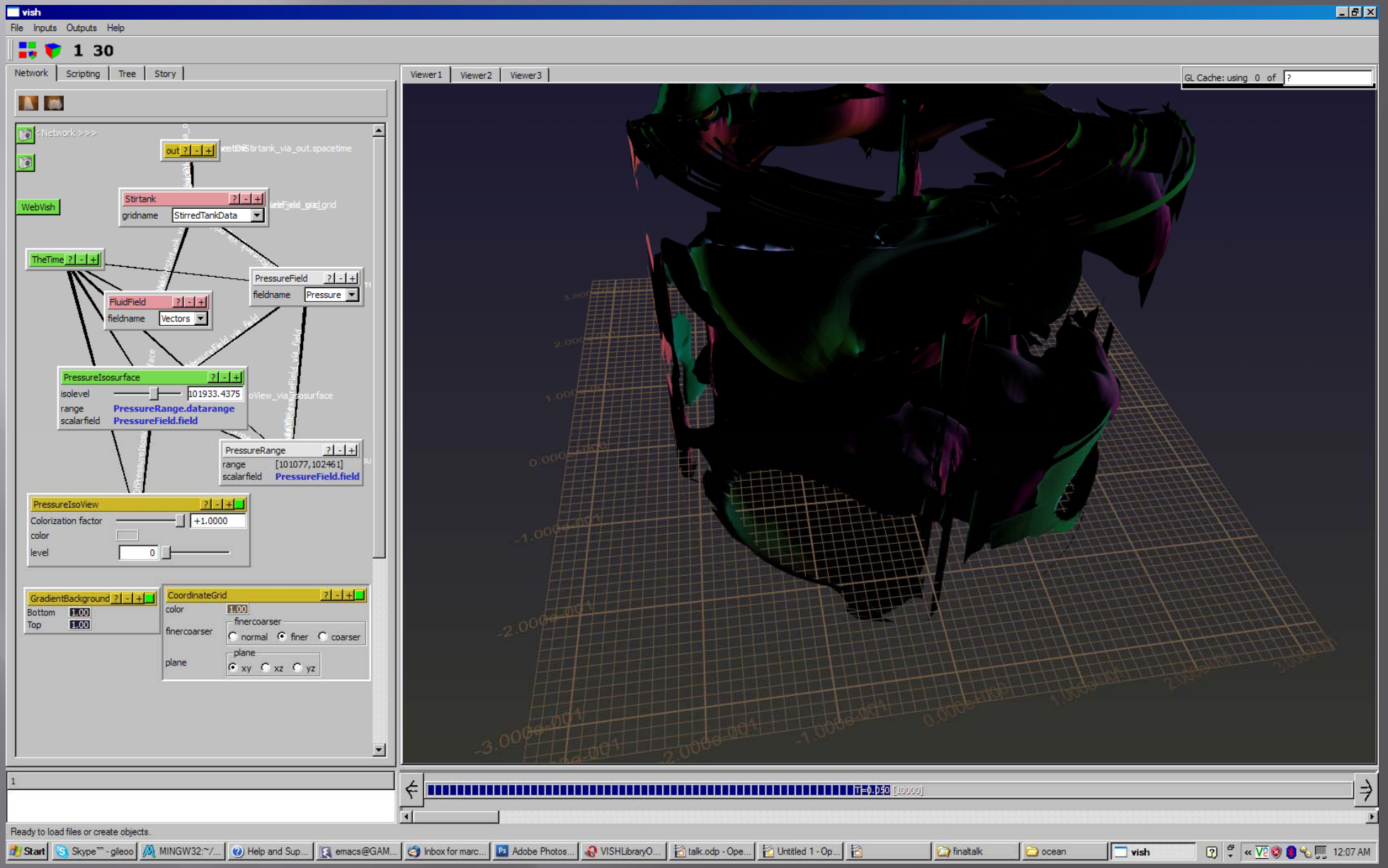


6279 x 6279 Pixel

Central Region of HR-Image

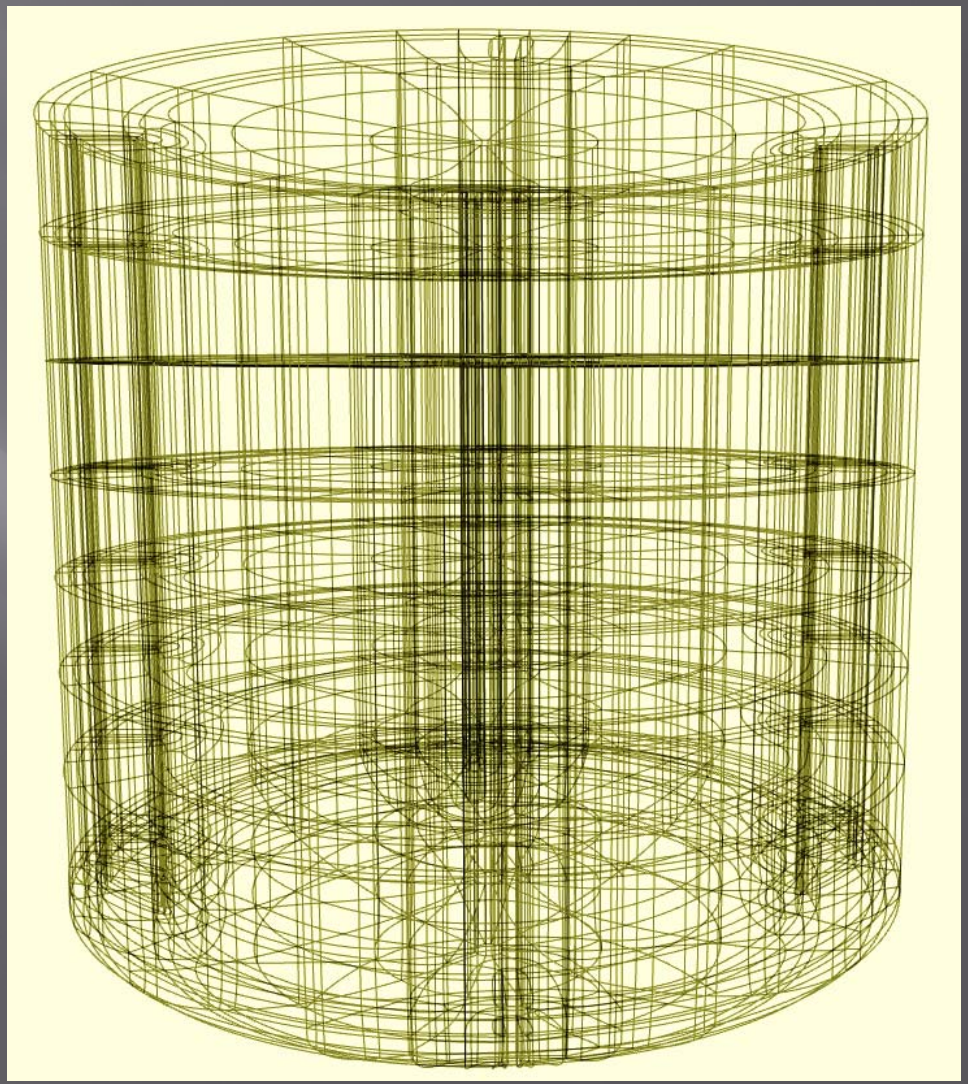


Practical Application
Student's work of 6 months

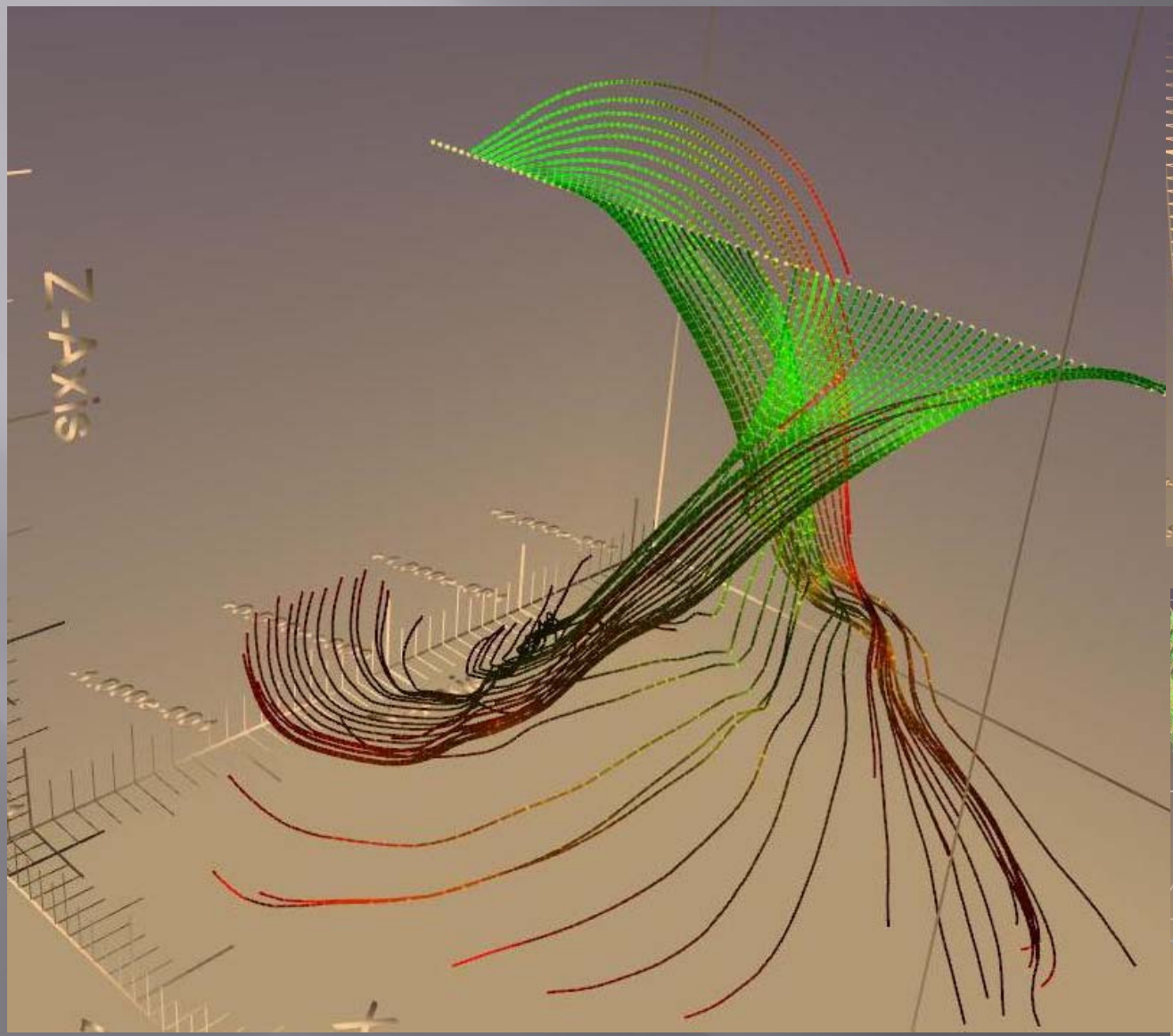


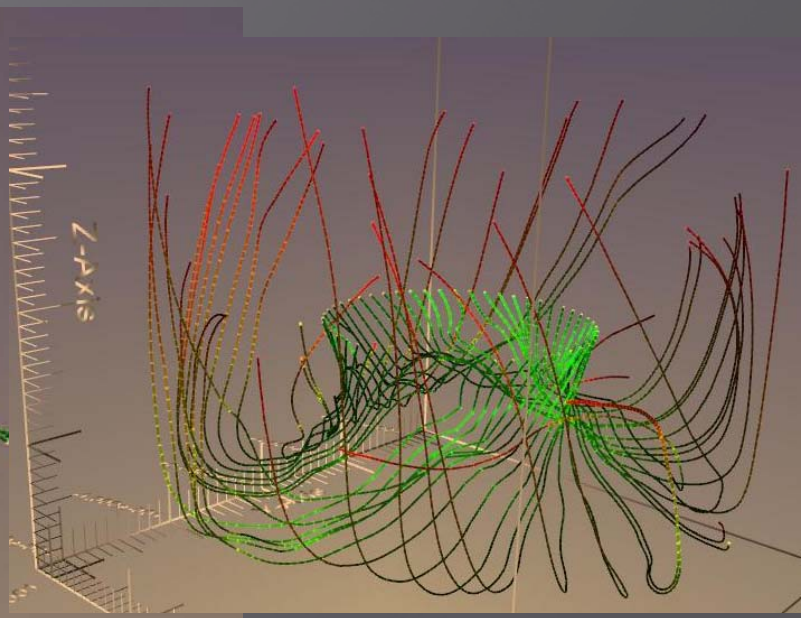
Multiblock Curvilinear Data

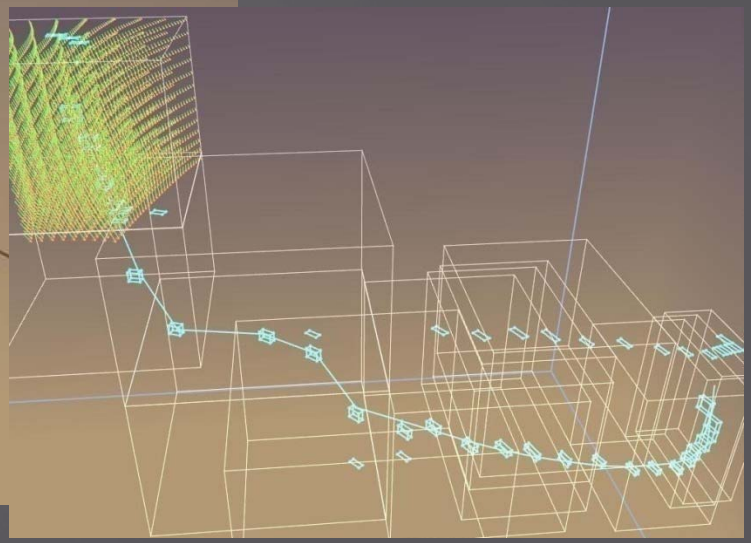
- The Stirtank Dataset
  - Department of Mechanical
  - Engineering
    - Sumanta Acharya
    - Somnath Roy
  - 2088 curvilinear blocks
  - Vectorfield describing velocity
  - Scalarfield describing pressure



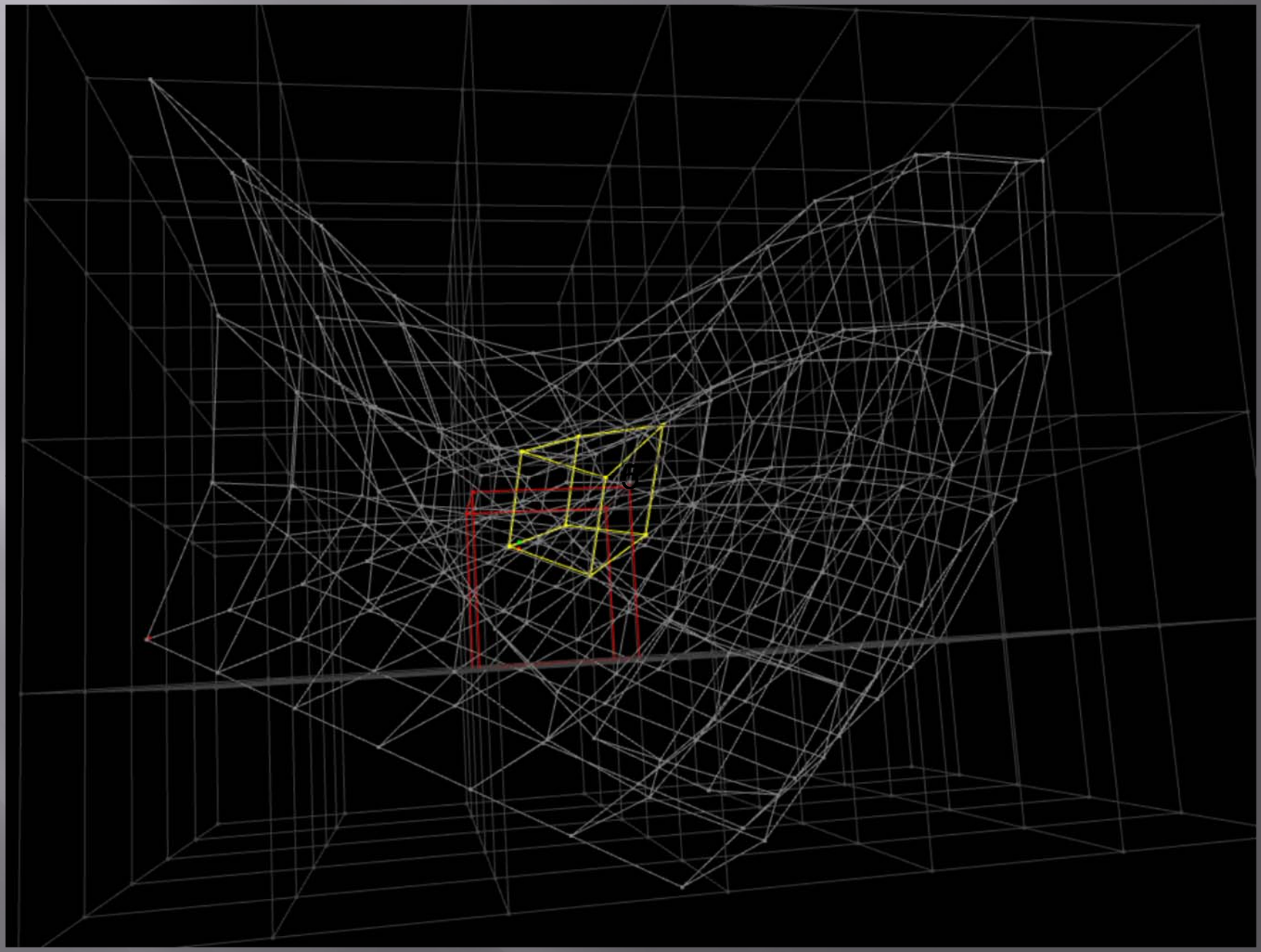
Streamlines





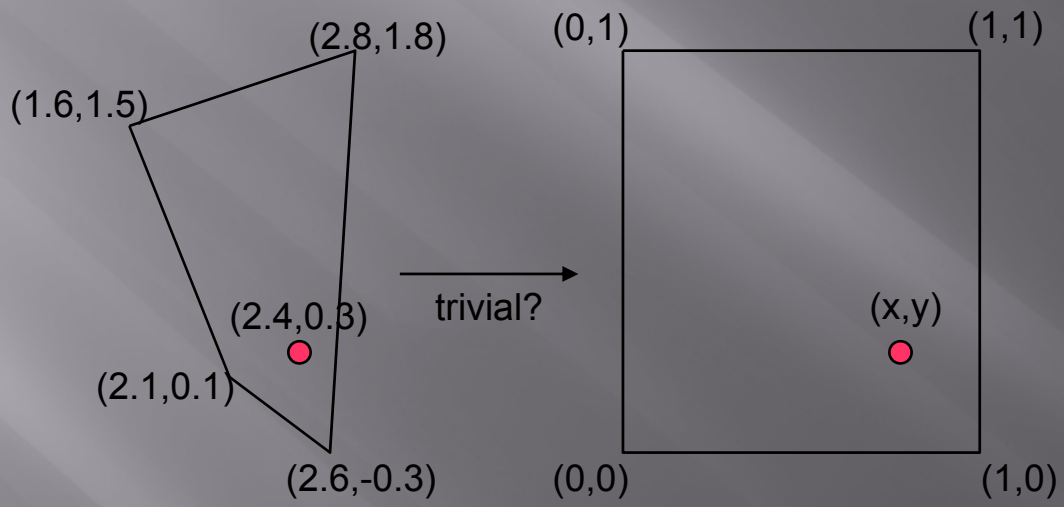


Complexity: how to interpolate



Interpolation non-trival

- Calculate local coordinates in a general Hexahedral Cell
  - Seems similar for interpolation using barycentric coordinates in triangles



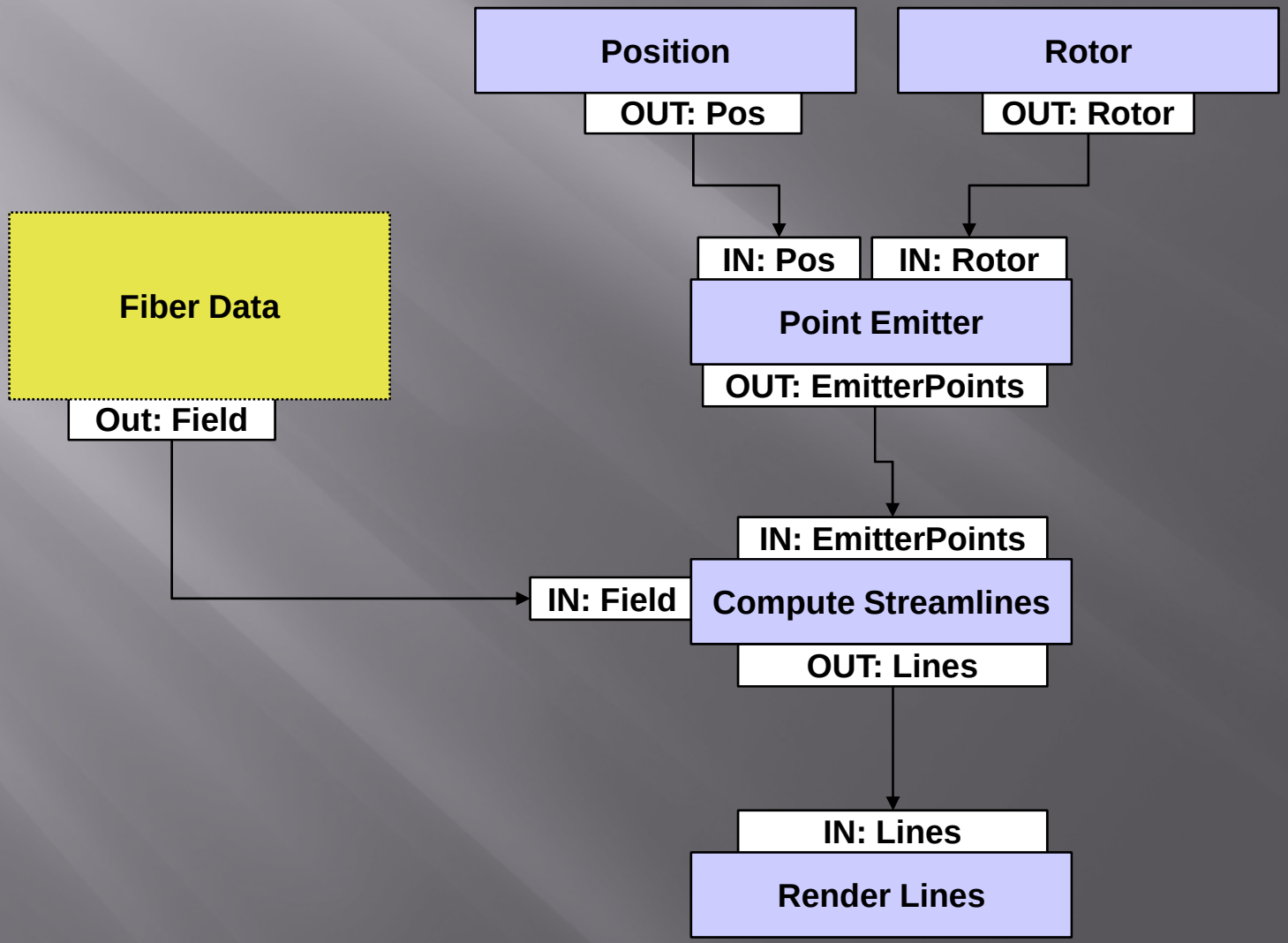
"Must be trivial, that's a **bilinear** mapping, or?"

"Wait no, it's not is it? But still it's **conform**."

"Now it slowly fulfills me with fear. It's even **not conform**."

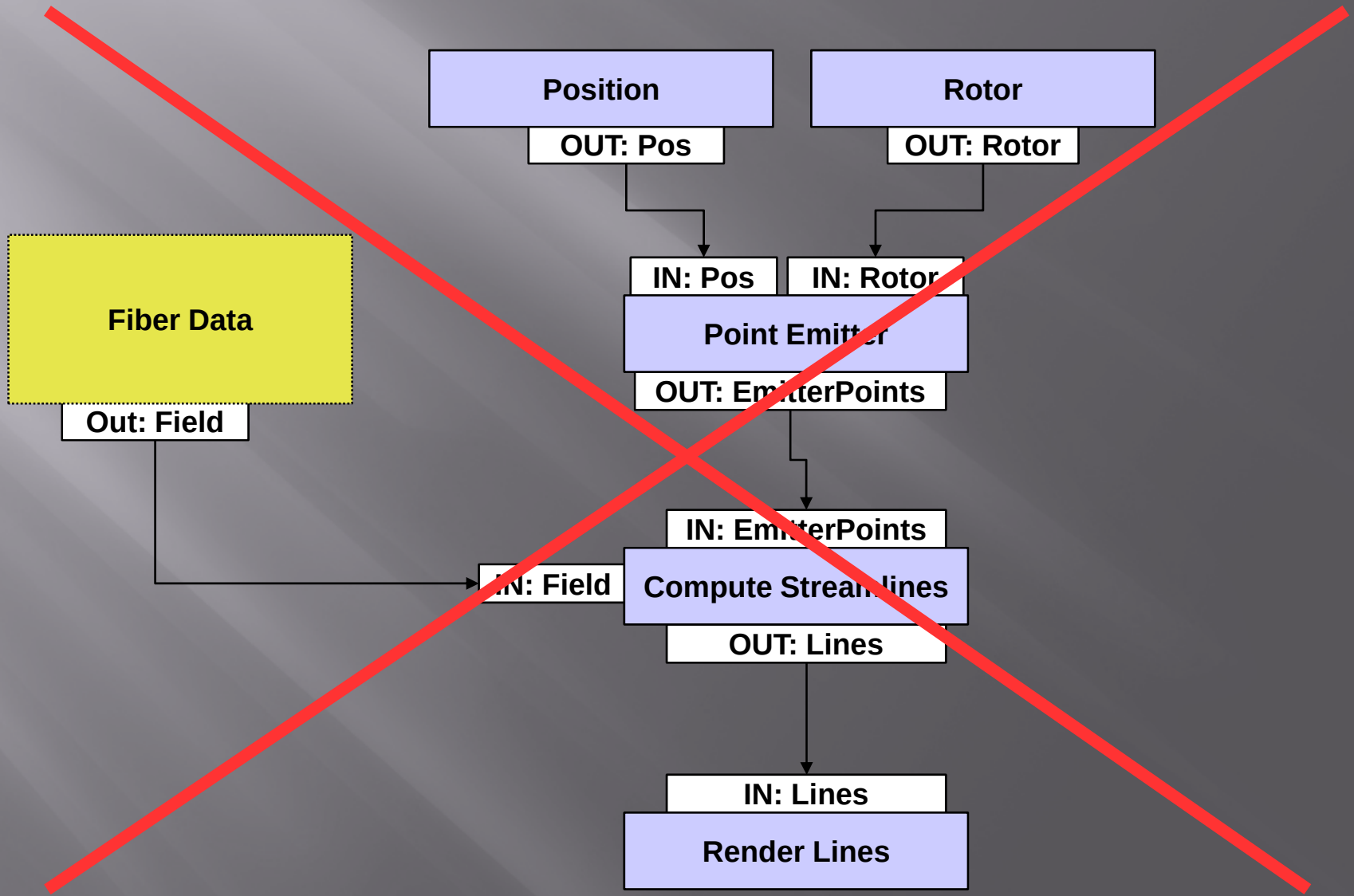
Using the input/output types

- Use bundles, grids and field for data flow



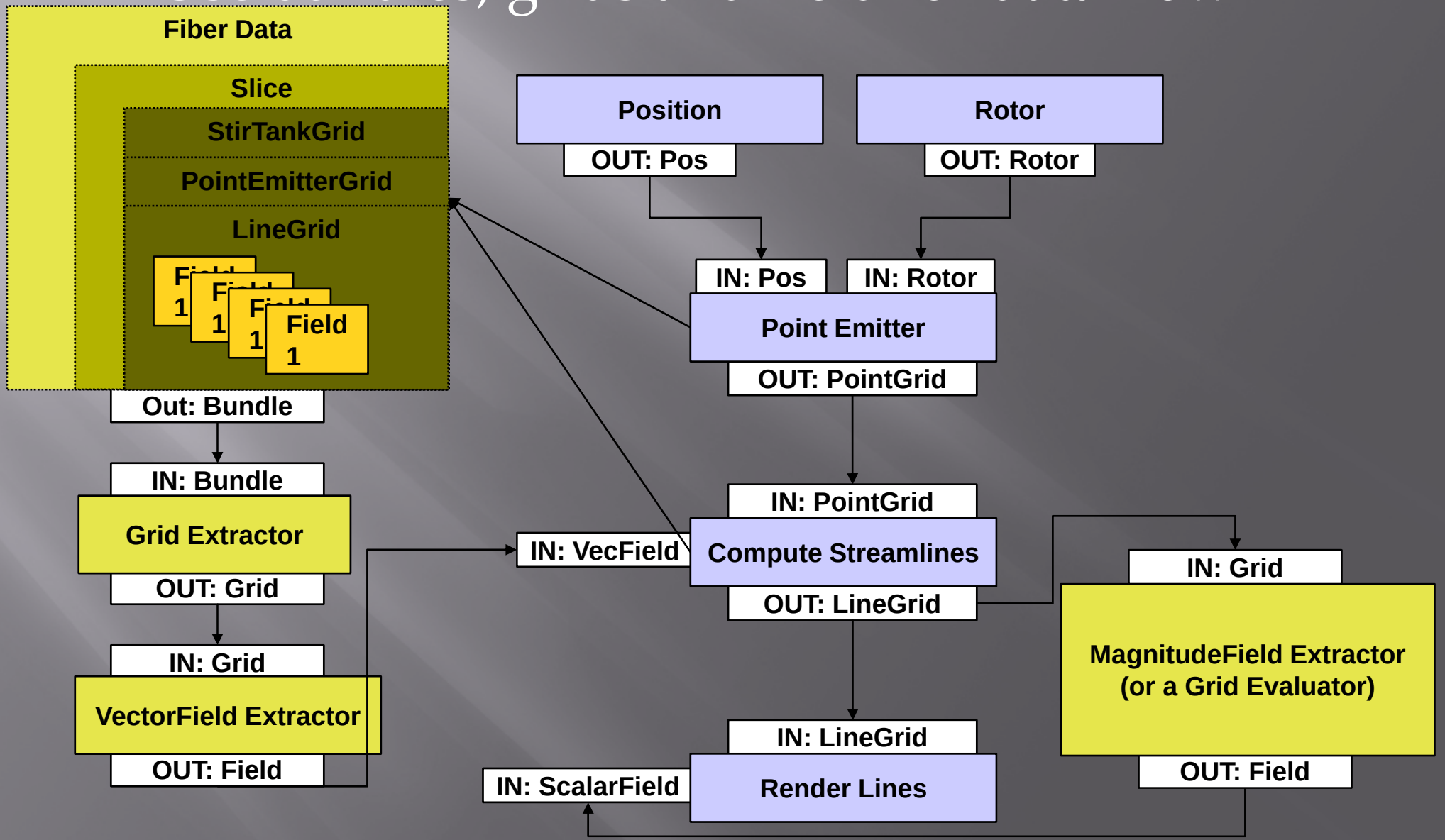
Using the data model

- Use bundles, grids and field for data flow

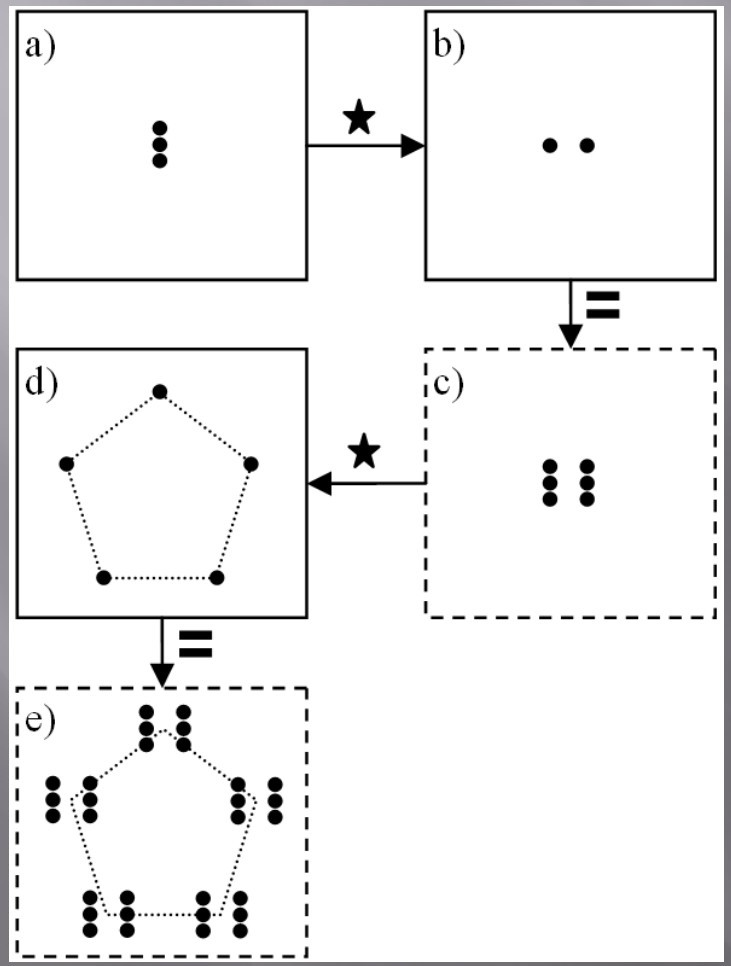


All objects described by Fiber Bundles

- Use bundles, grids and field for data flow



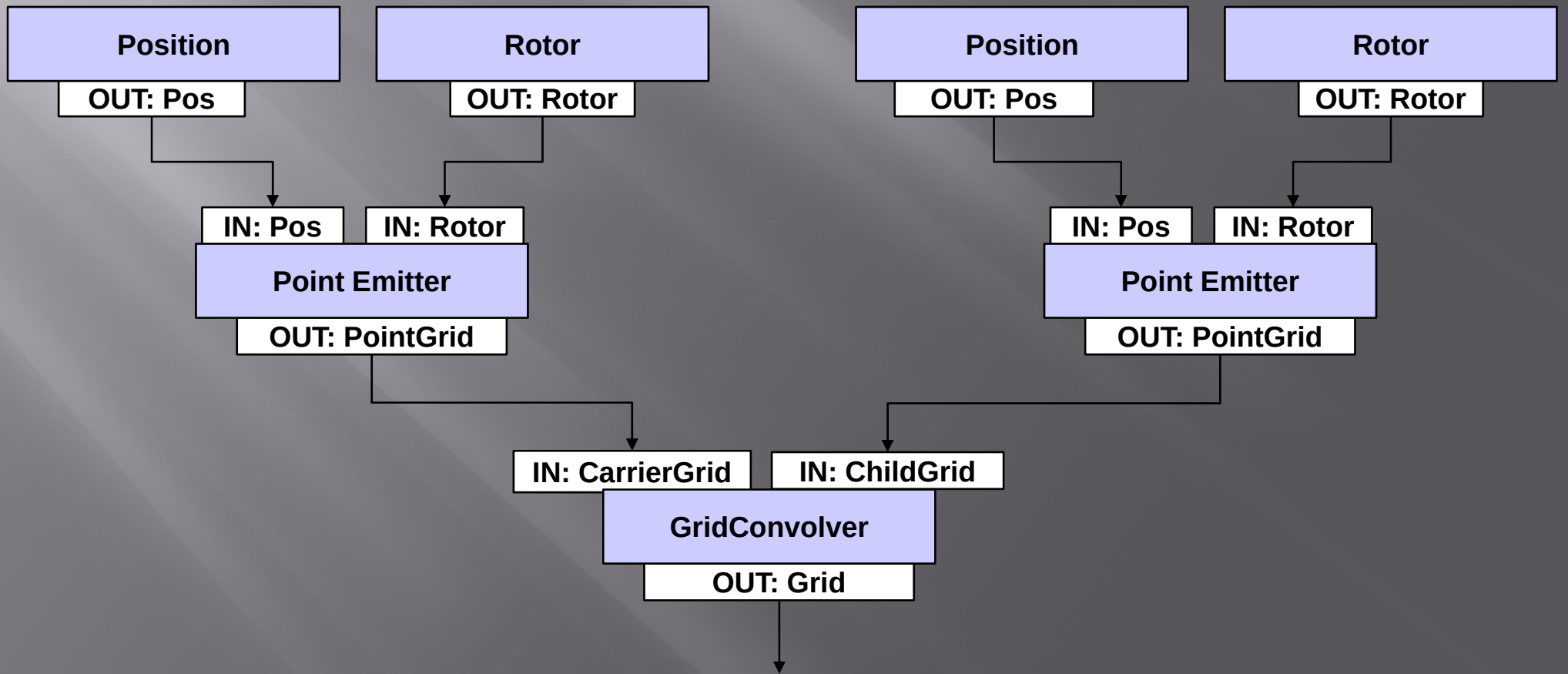
Point Emission by Convolution

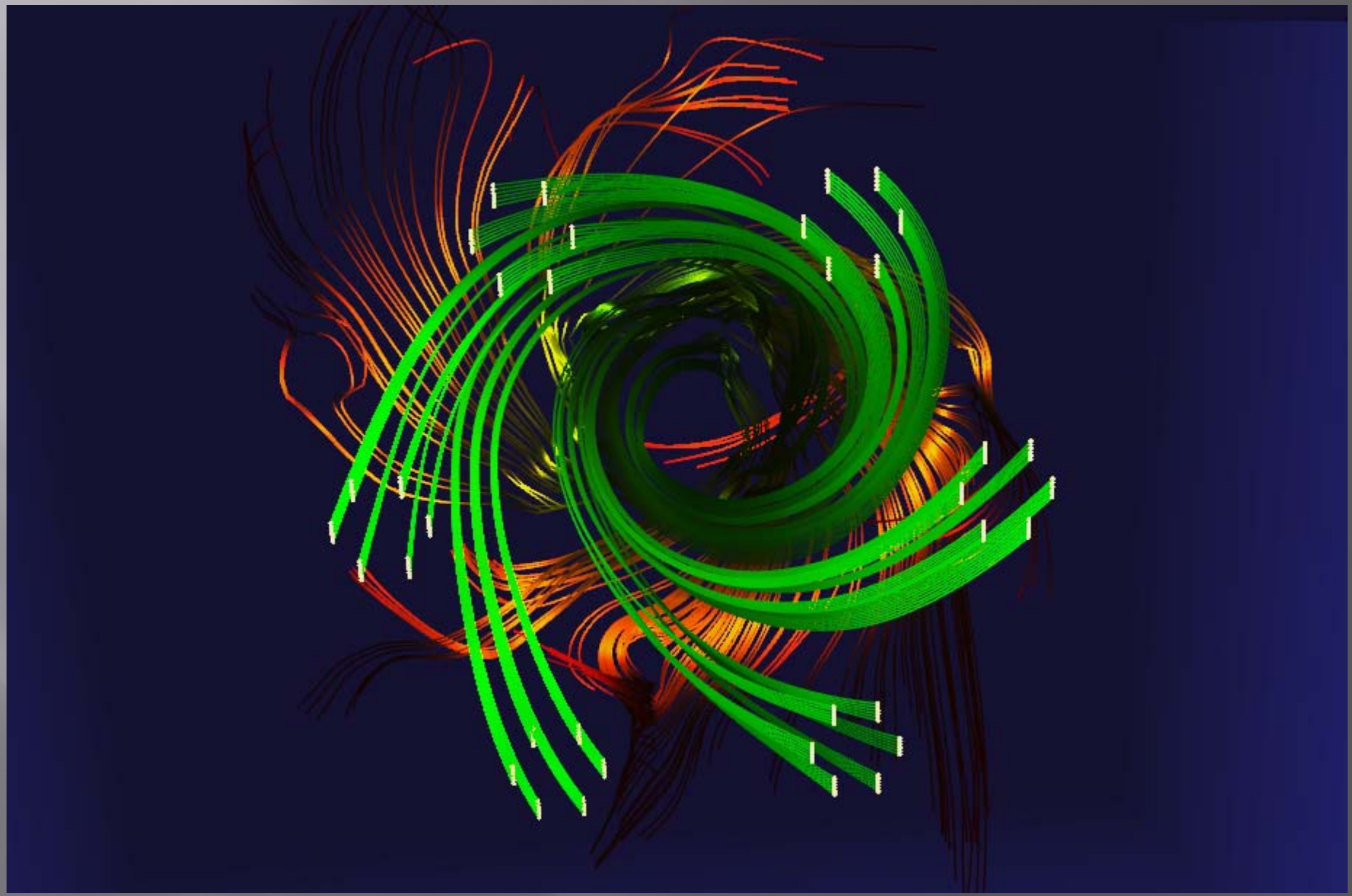


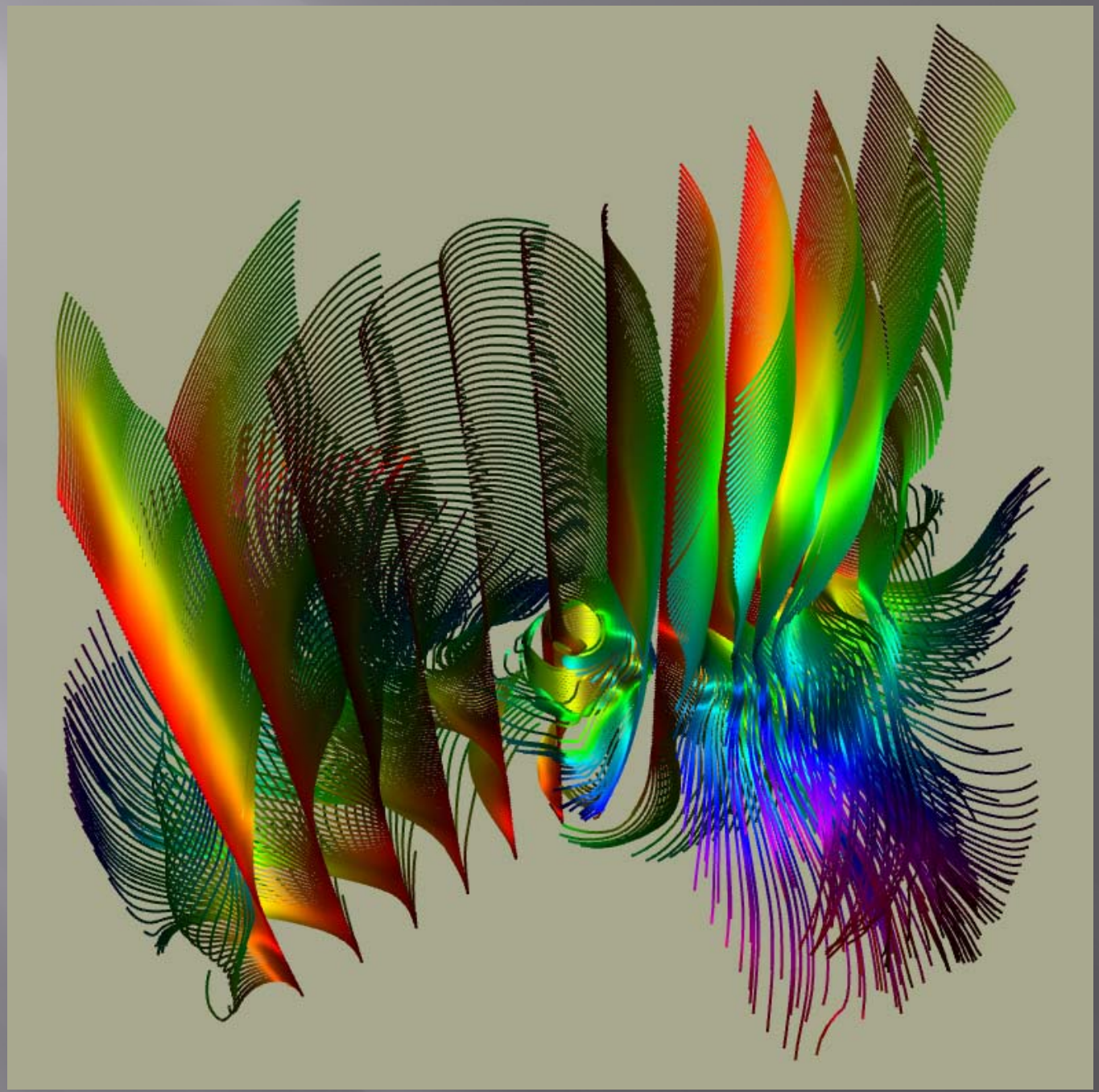
- Point Emitter Revisited
  - Basic Tasks
    - Create points on different geometric shapes.
    - Transform them
  - Idea
    - Instead of create points, copy a number of points on different geometric shapes (grids?!?)
    - Connect the output of a point Emitter to another point emitter

→ Grid Convolution (?)

- Module Separation
  - PointEmitter becomes a PointGridCreator
  - Additional module: *GridConvolver*

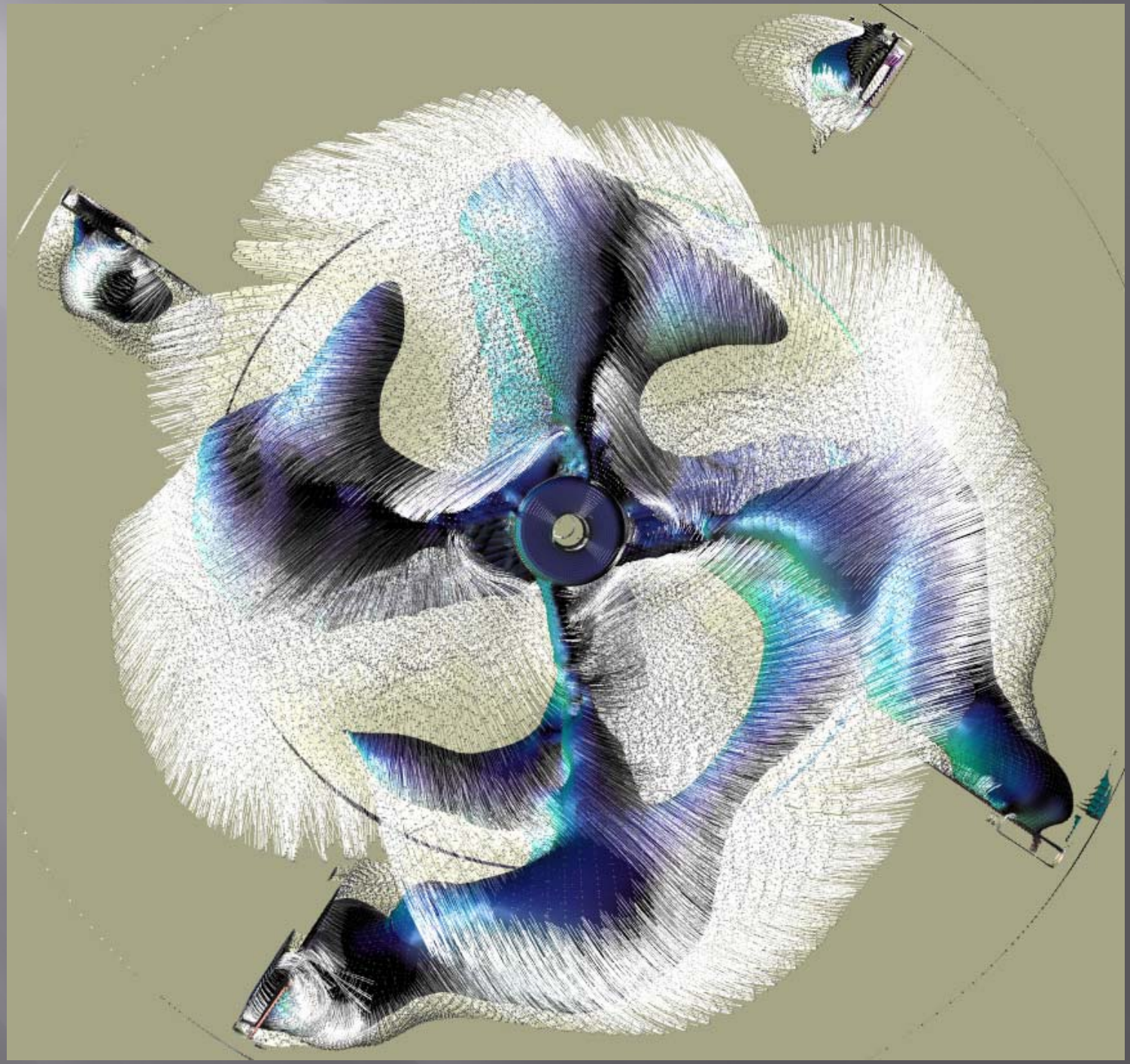




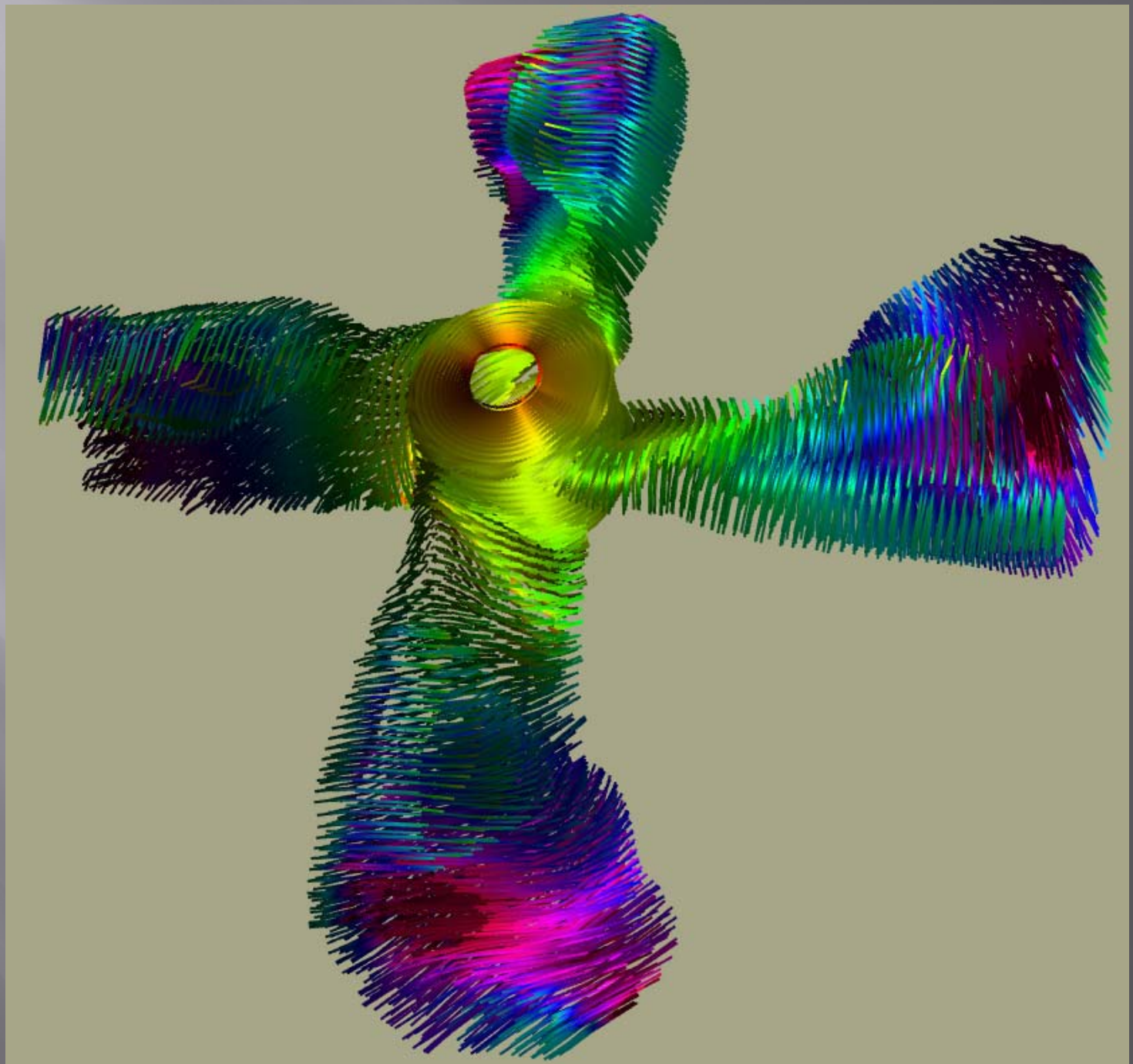


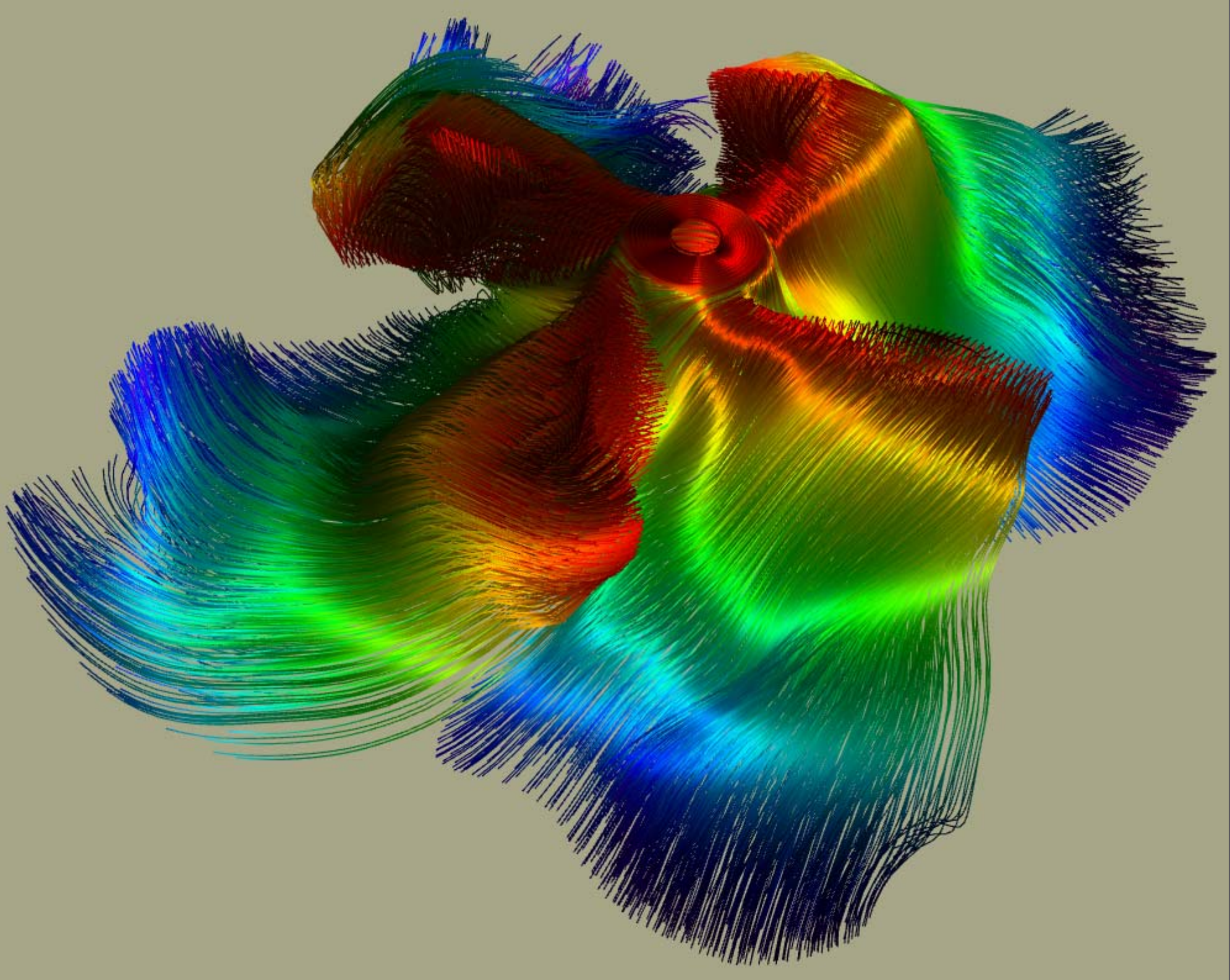
The Systematic Approach

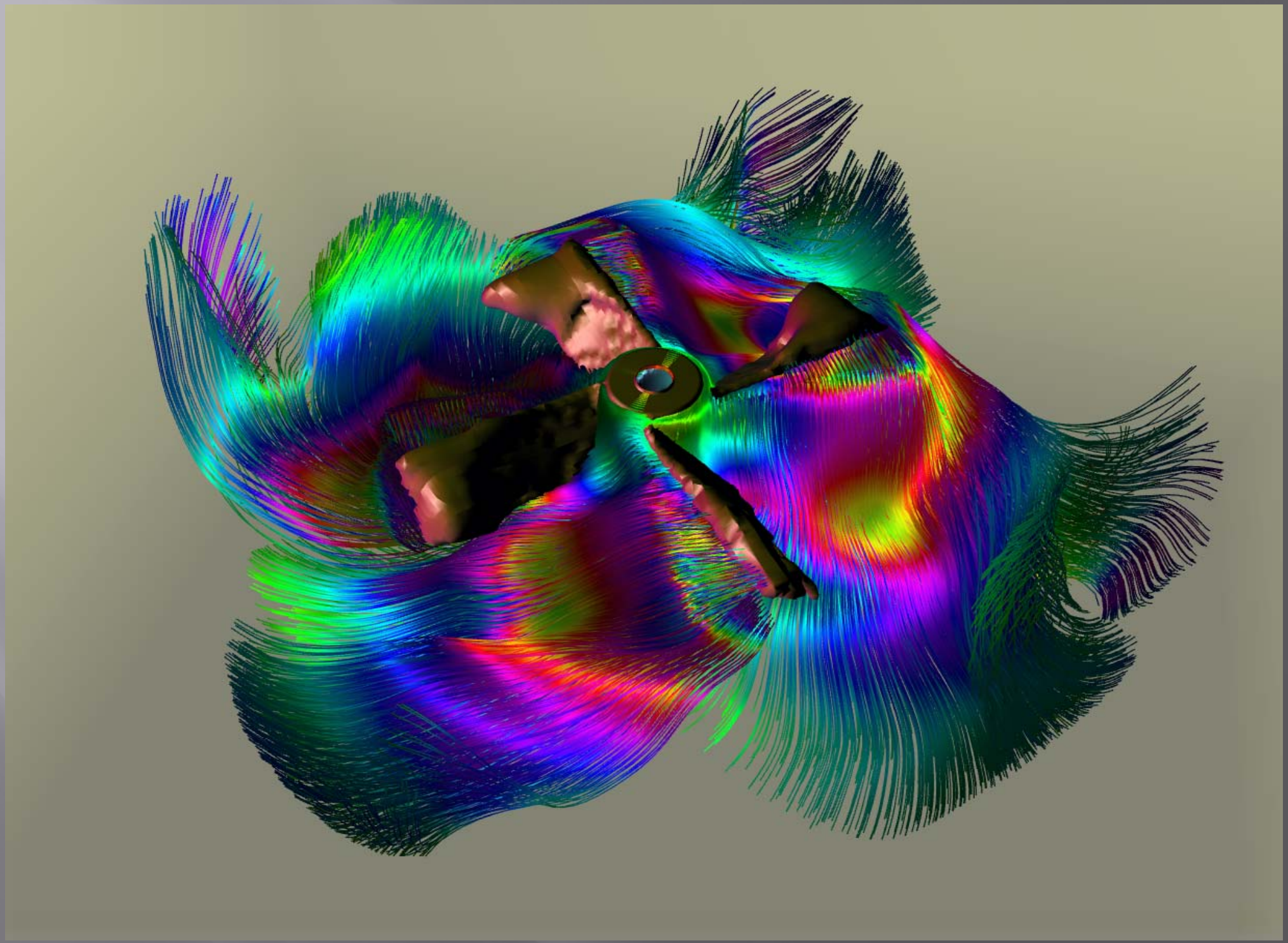
- Using Grids and Fields in Computation Module (1)
  - Use an input grid for defining the start-points (seed points) of streamlines
  - → Any grid object can be used!
    - e.g. the grid points of a computed iso-surface



Now do it the right way







The End

http://sciviz.cct.lsu.edu/projects/Vish/

For now ☺

Unified Data Model

- Point Sets
- Unstructured Cell Data
- Tetrahedral Grids
- Regular Grids
- Uniform Grids
- Uniform Cartesian Grids
- Uniform Polar Grids

HDF5 formulation available via

http://www.fiberbundle.net/

- Triangular Surfaces
- Quad-based Surfaces
- Irregular Surfaces
- Hierarchical Grids (AMR)
- Streamlines, Particle Trajectories, Geodesics
- Apparent Horizons
- Embedding Surfaces

Approach in VISH: "Fiber Bundle Data Model"

- A property-based description of scientific data
- A specific data type is built from "construction blocks"
  - Construction blocks are concepts from differential geometry, reduced to practical application domain in SciViz
  - Hierarchy of six levels

Integrated synergy effect

Not each type independent (VTK, Amira):

***Not* a list of myriads of independent cases – similar cases are modeled similarly**

- Eg. Uniform grid → regular grid ← curvilinear grid
- triangular surfaces / regular surfaces / points

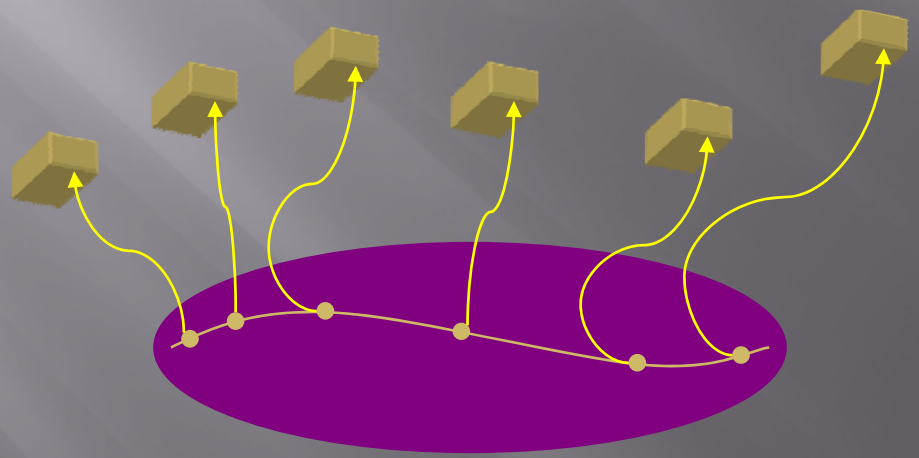
- Not necessarily the most straightforward modeling for a specific problem, but only gradual increase of complexity for closely related problem

- Inspired by mathematical concept of fiber bundles

- Data model similar to OpenDX, but extended/more systematic/self-constrained

What is a "fiber bundle"?

- Short: a space E that can (locally) be written as the product of a base space B and a fiber space F, e.g. $E = B \quad F$ (trivial bundle)



fiber space at each point of base space

base space

VISH data model: FiberLib2

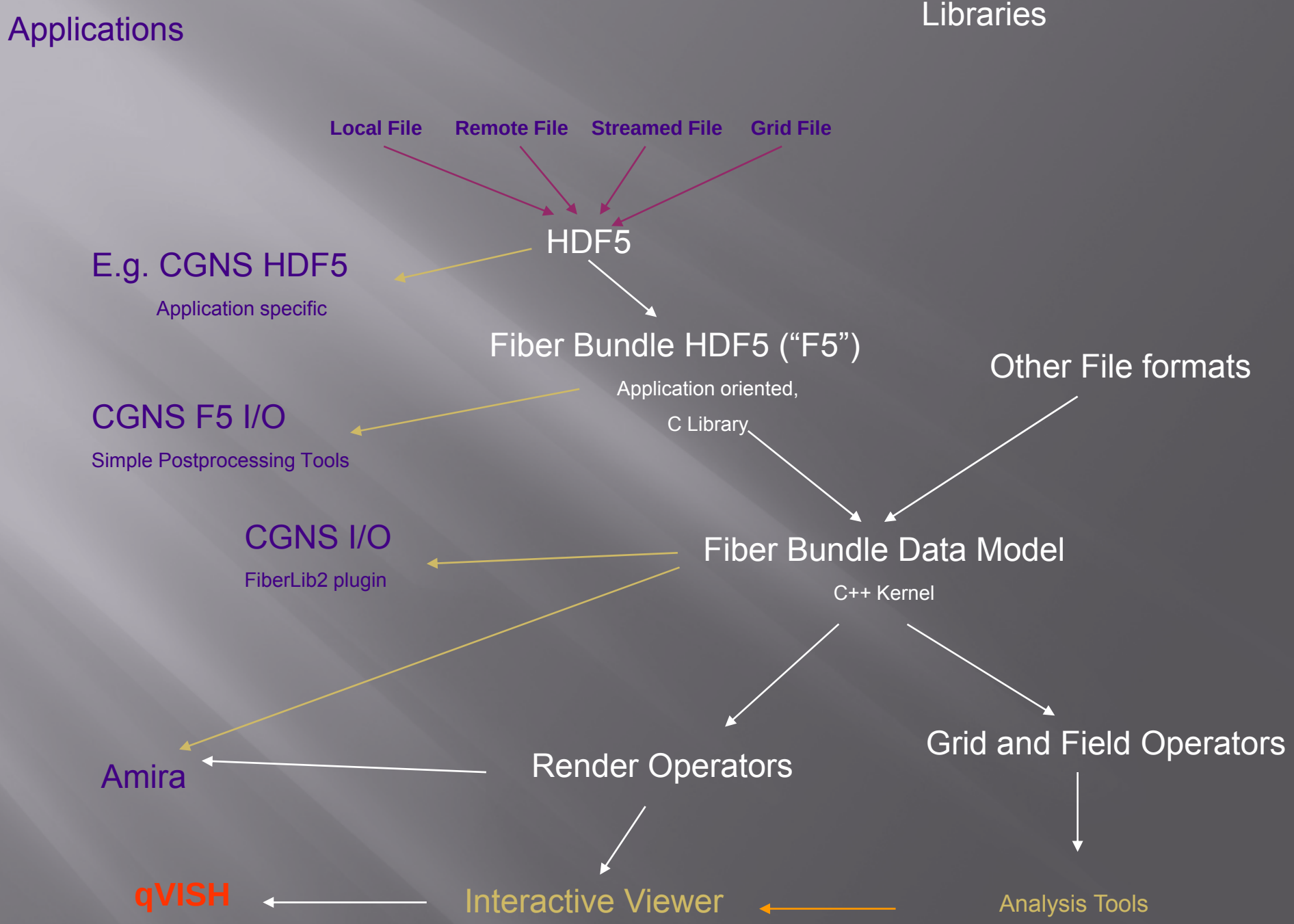
- Systematic treatment of a wide category of scientific data, based on the mathematics of fiber bundles
- Common denominator for otherwise diverse grid types
- Plugin to VISH (“fiber-VISH” or “FISH”)
- No need to use it, customized data types also possible in VISH (but bypassing the FISH infrastructure then)

FiberLib2

- Systematic approach for scientific data:
  - Particle systems → unstructured grid → regular grids → uniform grids → block-structured uniform grids → curvilinear multi-block grids …
  - Incremental transition from one such category to next one
  - Can cover multiple timesteps, grids, fields…

FiberLib2 I/O features

- I/O layers are plugins (shared libraries) independent of core implementation
  - Distinction among data and metadata
  - on-demand loading and creation of data
  - Cache-management
- Most powerful I/O layer is “F5”
  - 1:1 representation of the FiberLib2 into HDF5


Applications
Libraries
Local File
Remote File
Streamed File
Grid File
HDF5
E.g. CGNS HDF5
Application specific
Fiber Bundle HDF5 (“F5”)
Application oriented,
C Library
Other File formats
CGNS F5 I/O
Simple Postprocessing Tools
CGNS I/O
FiberLib2 plugin
Fiber Bundle Data Model
C++ Kernel
Amira
Render Operators
Grid and Field Operators
qVISH
Interactive Viewer
Analysis Tools

FiberLib2 Usage

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Simplest case: Equidistant static scalar field (float data[X][Y][Z])

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Multiple fields on same domain

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Time-dependent fields on same domain

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Multiple blocks (multiprozessor output)

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Mesh refinement or unstructured grids

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Multiblock Curvilinear grids

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Multiblock grids with refinement

- Hierarchical tree of substructures, five levels:
  1. Time dependency (parameter space)
  2. Grid object (computational domain/mesh)
  3. Topological information (vertices, cells, …)
  4. Coordinate representations & relationships
  5. Fields (scalar, vector, tensor)
  6. (field fragments)

Availability

- Code development management:
  - http://sciviz.cct.lsu.edu/projects/vish
  - Available via SVN in source code for registered users at http://vish.origo.ethz.ch/