Introduction to Scientific Visualisation

Paul Bourke
WASP, UWA

Definitions

- Definition 1: Gain insight into data by using computer graphics.

  Data acquired from physical measurement or simulation in scientific research.

- Definition 2: Provide scientists with insight into their research. Common secondary benefit: data checking and verification.

- Simplified: Turning often large and complicated datasets into images and movies.

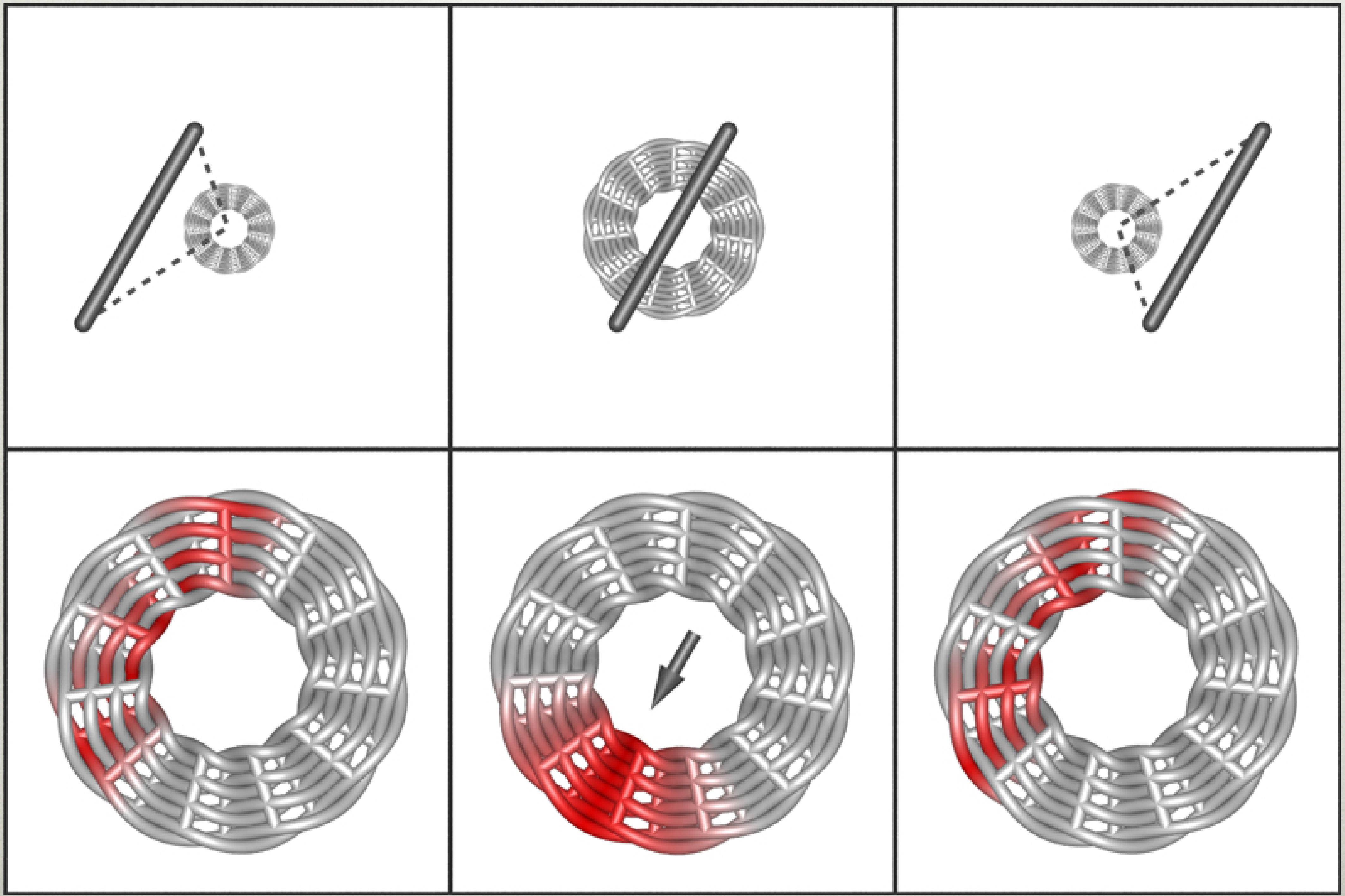
Types of Scientific Visualisation

- Illustrative visualisation
  Convey an understanding of scientific principles, often targeted at a non-expert audience.

- Data visualisation
  Convey an understanding of a dataset, usually intended for experts in the research discipline.

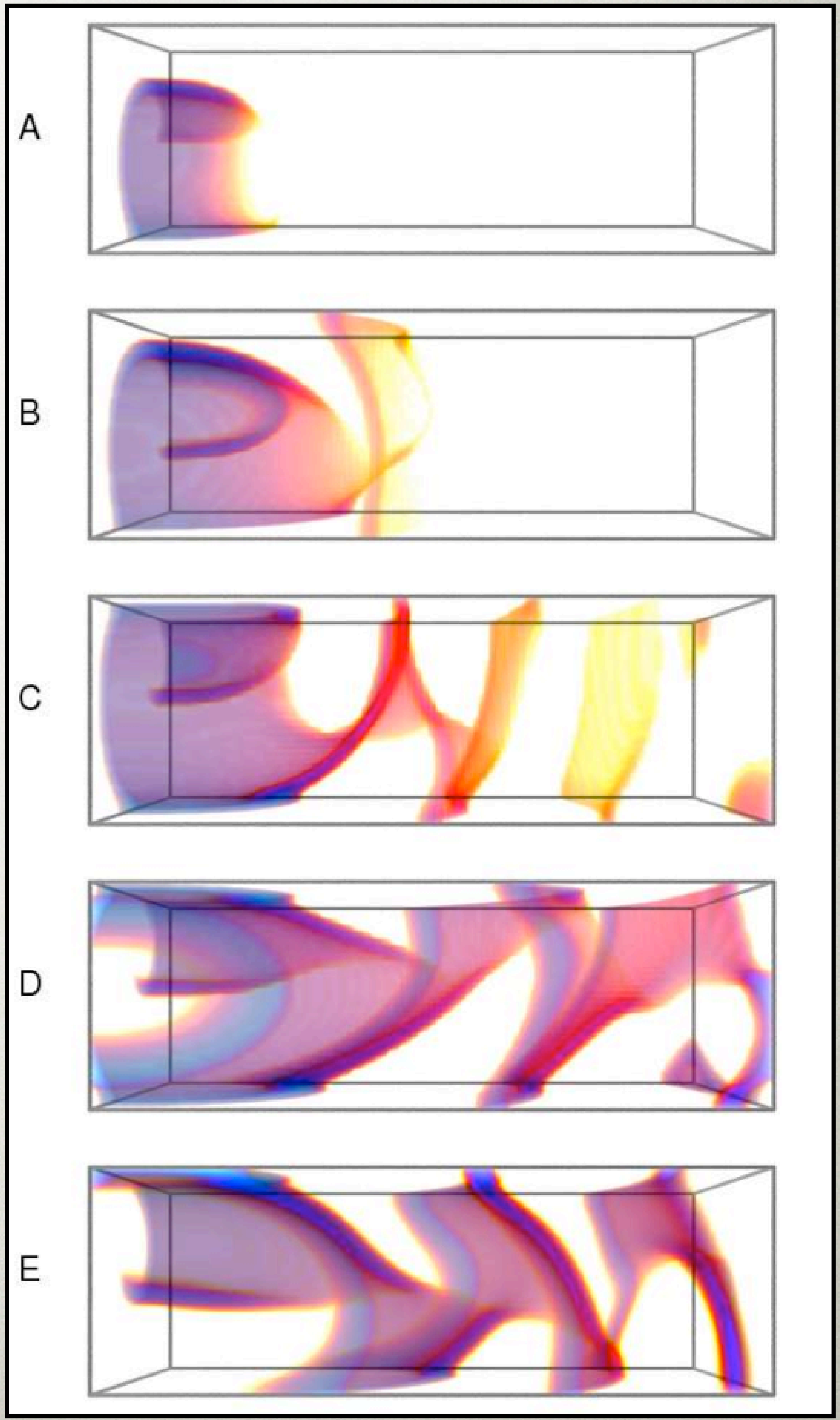
Example: Illustrative visualisation

- Model of activation in the visual cortex.
- No data involved, illustration of mathematical ideas.



Example: Data visualisation

- Simulation of helical waves.
- Volumentric dataset at each time step.
- Velocity mapped to colour and transparency.



DATA CHARACTERISTICS/CLASSIFICATIONS

- How many independent variables? For example what is the dimension of the data?
- Variable type: scalars, vectors, ... dimension?
- Regular sampling or not?
- Variables discrete or continuous?
- Is the data time varying or static?
- Sometimes techniques are categorised by whether they involve surface rendering or more direct rendering algorithms.

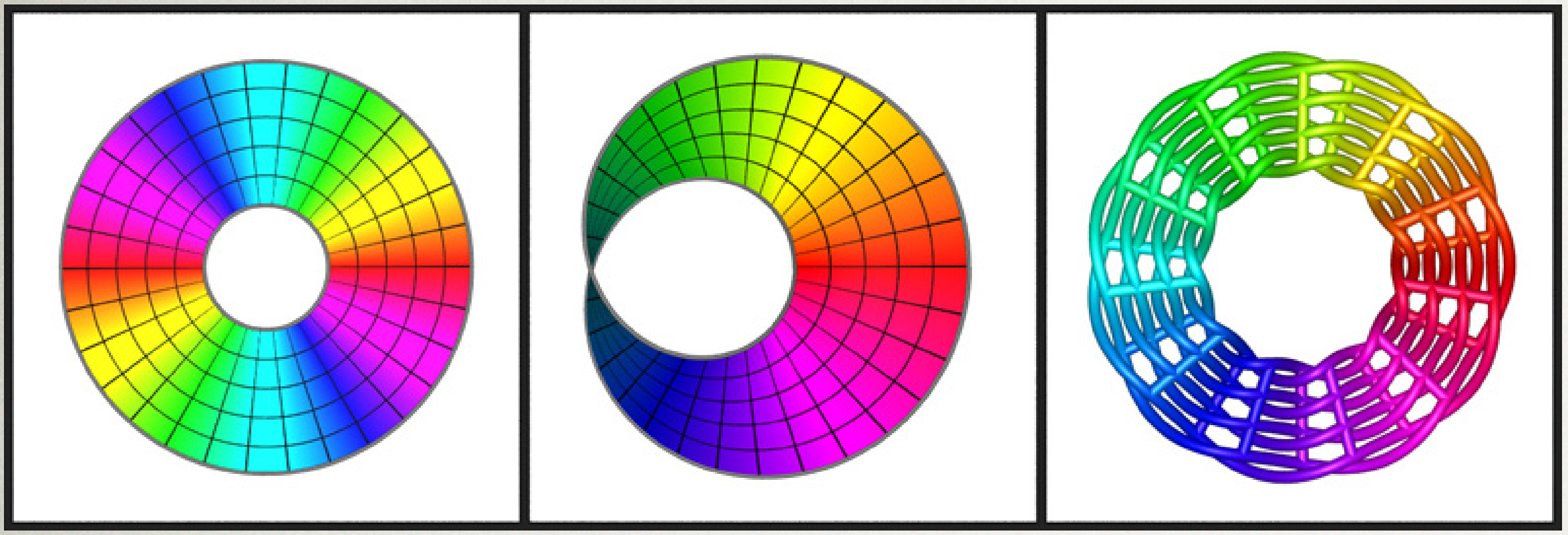
Audience considerations

- Experts in the field and researchers who created the data. Usually want interactivity, are prepared to become familiar with a graphical/visual language, expect quite precise representations of the data.
- Presentation to a more general expert audience, for example: seminars, conferences, peers.
- Visuals for the general public as part of a science/museum exhibitions, children and adult education courses. Interaction possibilities and high-end hardware may be limited, the information must be more clearly/simply presented.

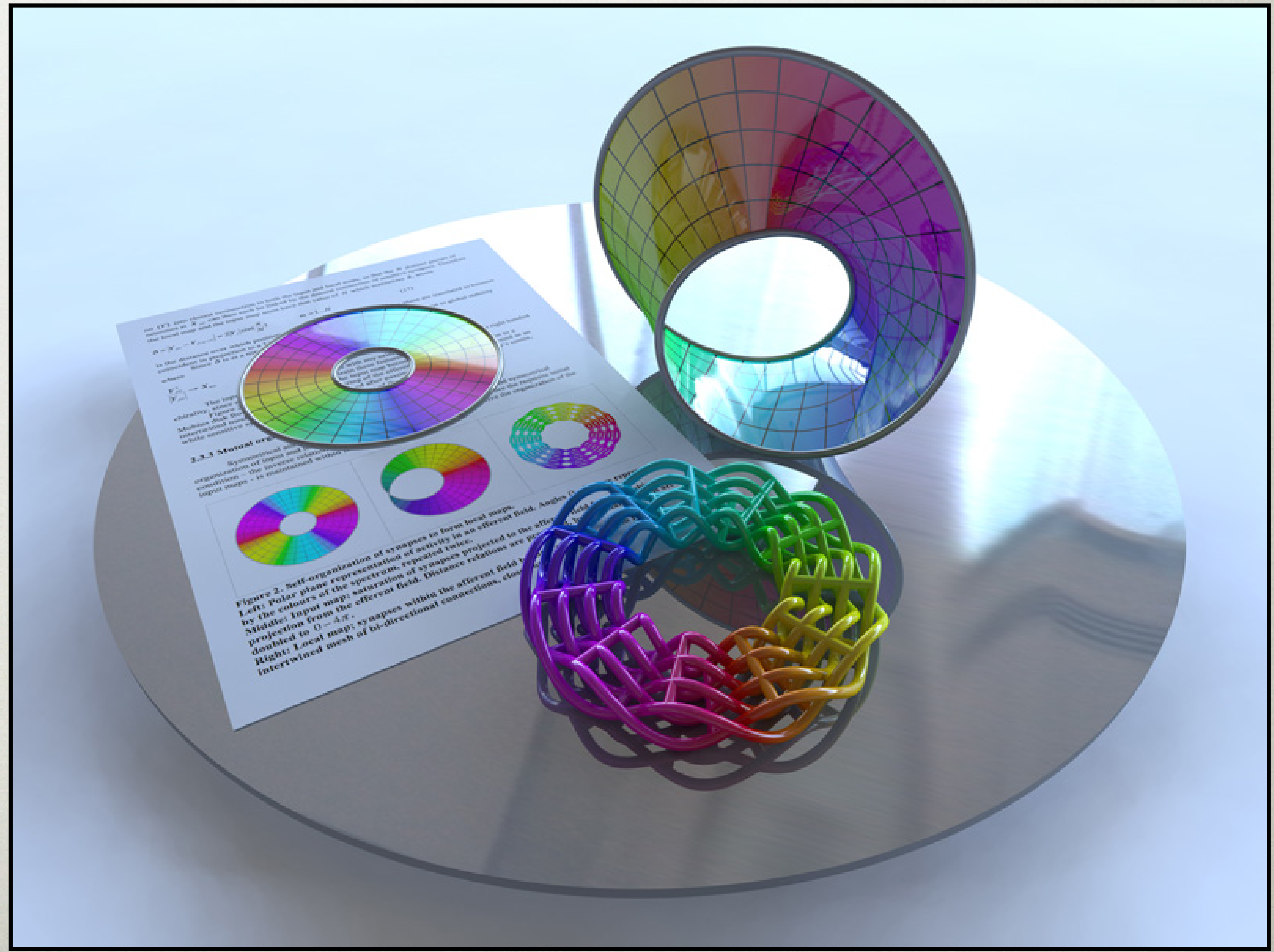
Media/display considerations

- Usually output of visualisation is viewed on 2D digital displays.
- Stereoscopic viewing and other exotic displays are not widespread.
- Video and movie versions will typically be lower resolution.
- Publication versions are obviously still images, perhaps with a limited colour palette and perhaps even black and white. They will be higher resolution than digital displays.
- Promotional graphics (Journal covers, posters), high resolution and visual impact.

Example: interaction->journal cover



Versions of the mathematical models used by the researcher. (Interactive)



Version rendered for the journal cover. Radiosity + high dynamic range environment map. (Pre-rendered)

Standard Techniques / Issues

- Create 2D and 3D geometric representations of data.
- Mapping variables to colour and/or opacity.
- Mapping variables to glyphs.
- Dimension reduction: contours, isosurfaces.
- Data representation and file format conversion.
- Use of novel display technologies.
- Interactive vs pre-rendered visualisation.
- Software, homegrown, open source, commercial.

Geometric Representations

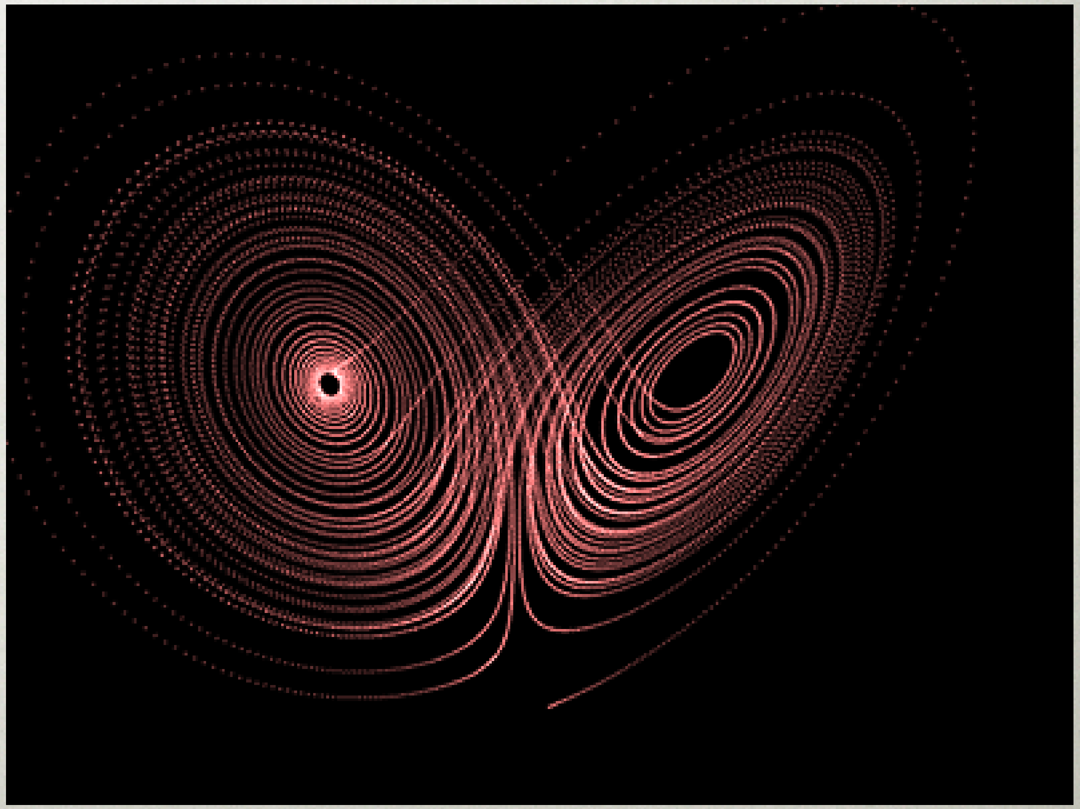
- Points, lines, surfaces, clouds.
- Usually there are some coordinate variables in the data or at least some natural mappings of variables to spatial coordinates.
- Not necessarily cartesian coordinates, for example: polar, cylindrical, hyperbolic, complex ....
- While not limited to 3 geometric dimensions, we are constrained to projections into 2 or 3 dimensions.

Example

- Simple representation of points in 3 dimensions in a 3D data format (geom) and viewing with software that interactively renders the geometry. Simple tools like this are often used to get a general “feel” of the data.
- Go to the lorenz directory “cd lorenz”
- “make”
- Run by typing “lorenz1 > 1.geom”
- View results “geomviewer -og 1.geom”

.geom file primitives:

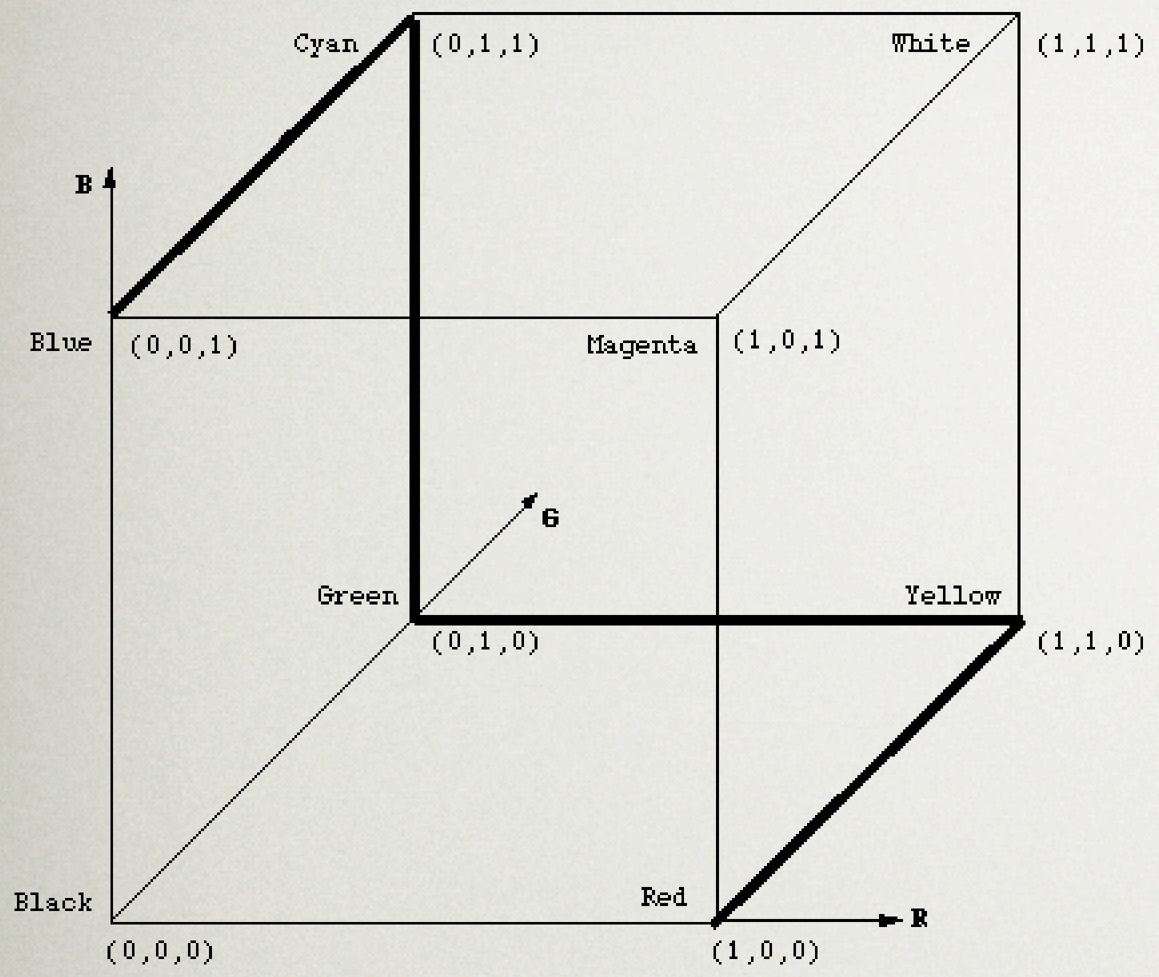
s - sphere	
d - disk	
c - cone	
t - text	
p - point	
l - line	
f3 - triangle	
f4 - quad	
m - marker	



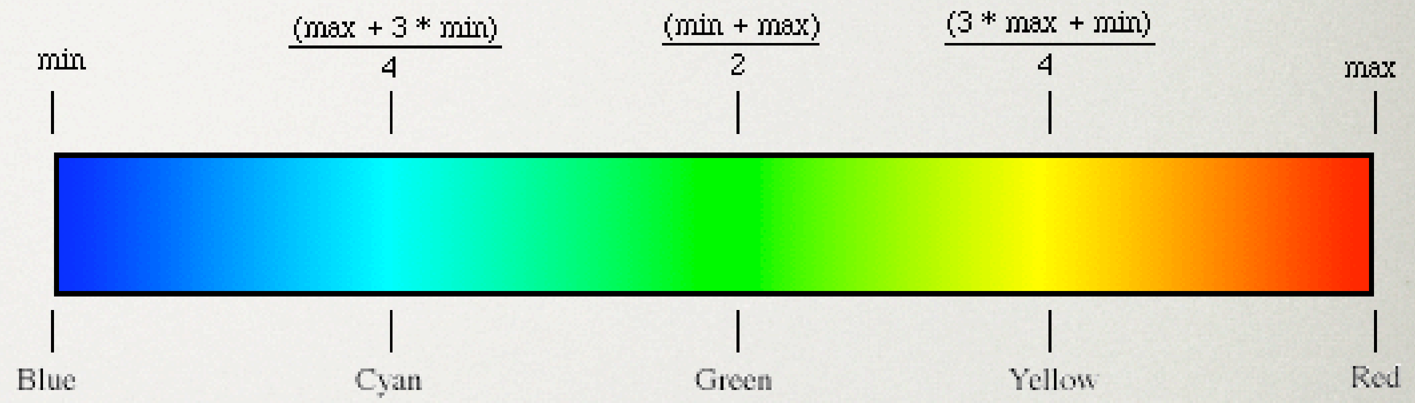
Colour Mapping

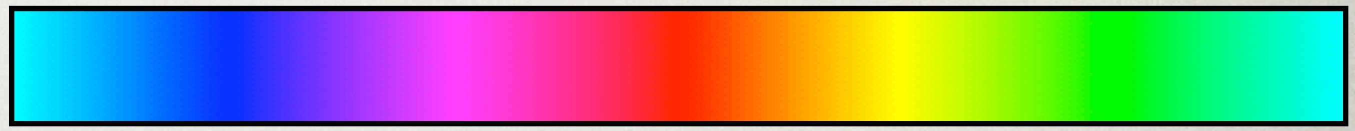
- Colour ramps, one directional or circular.
- Continuous or discrete (eg: colour lookup table).
- Applied between the range of a variable, linear or otherwise, clamping variable to a limited range.
- Some colours have cultural meaning (red=hot, blue=cold).
- There are characteristics of our visual system to consider.
- Derived in different colour spaces:
  RGB, HSV, HSB, YUV, CMY, ...
- Sometimes useful to represent variables geometrically and by colour ramps at the same time.
- Relationship is not always linear, eg: power or logarithmic.

Example: blue-cyan-green-yellow-red

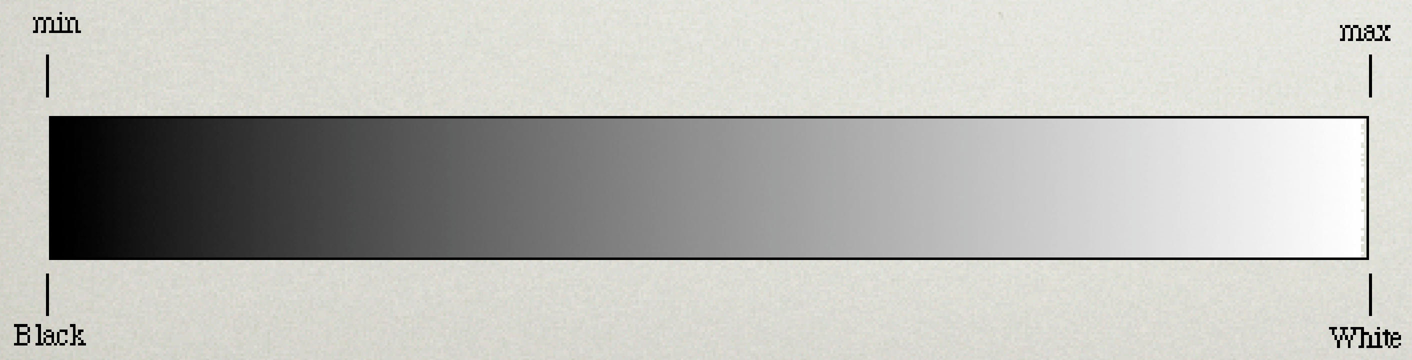


Directional





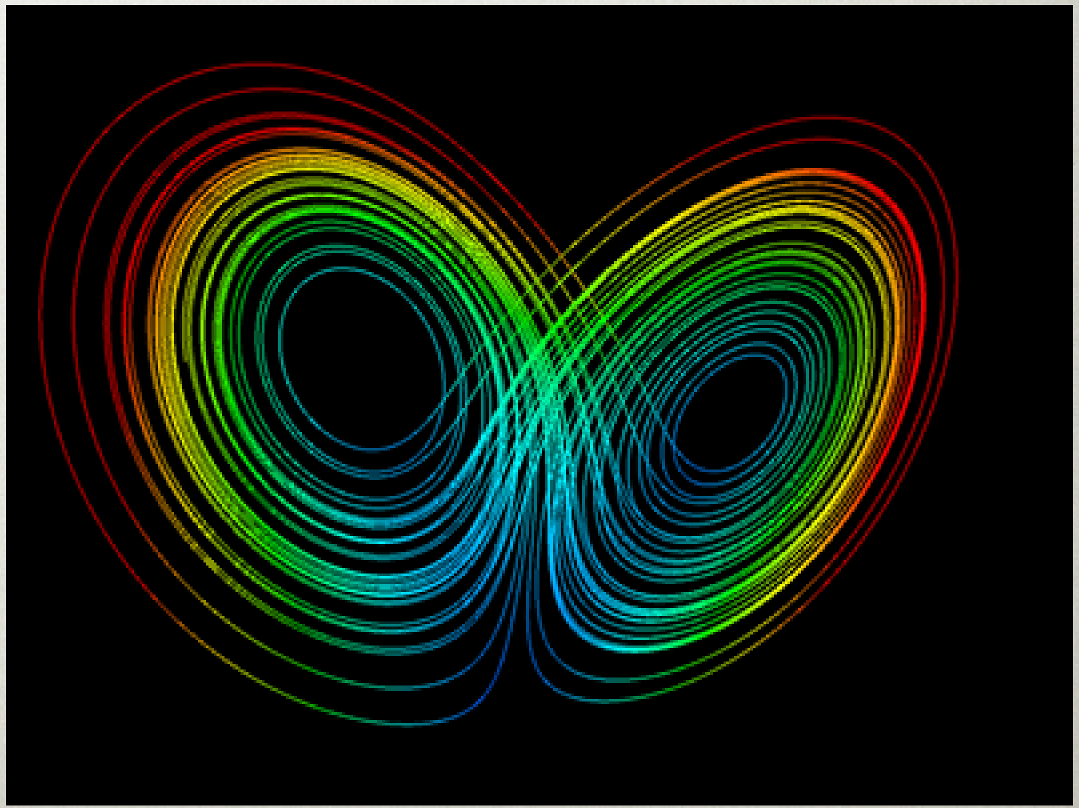
Circular



Greyscale - line from black to white in RGB colour cube.

Example

- Shade lorenz attractor in the last exercise by speed, red is high speed, blue is low speed (speed is just proportional to the distance between successive points). Colour maps normally require calculation of the range of variable values, if not known beforehand.
- Go to the lorenz directory “cd lorenz”
- “make”
- Create data “lorenz2 > 2.geom”
- View results “geomviewer -og 2.geom”
- “make clean”

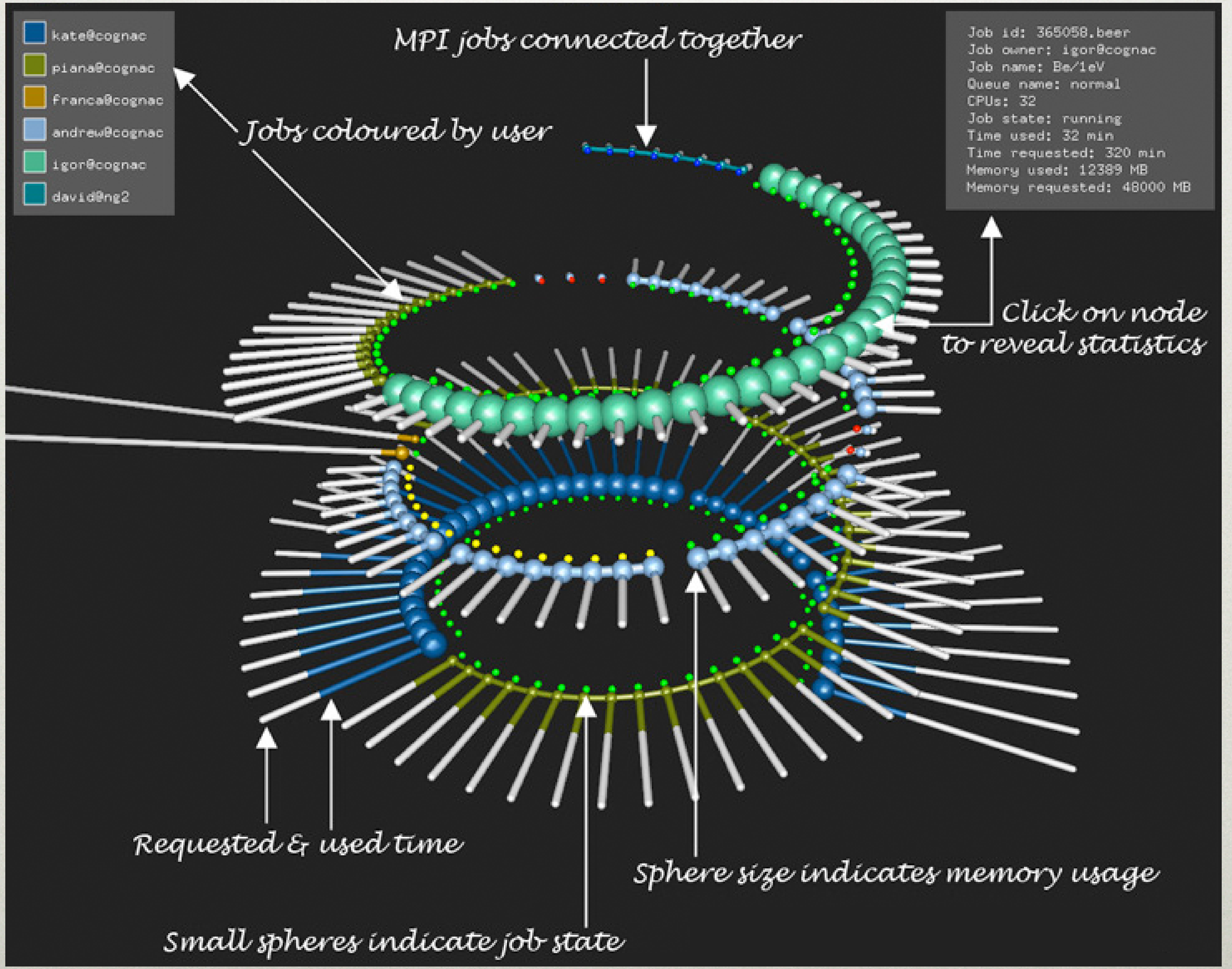


Glyphs

- A glyph is a symbol or graphical primitive with variables mapped to various geometric and colour attributes.
- The variables mapped to the glyph do not necessarily have an inherent geometric meaning but most successful mapping do.
- Examples: vector represented by an arrow, scalar represented by length, a variable describing an amount mapped to glyph size, and so on.

Example

- Turns data from PBS (Portable Batch System for Supercomputers) into a graphical representation.



```
Job id: 365058.beer
Job owner: igor@cognac
Job name: Be/1eV
Queue name: normal
CPUs: 32
Job state: running
Time used: 32 min
Time requested: 320 min
Memory used: 12389 MB
Memory requested: 48000 MB
```

🟦 kate@cognac
🟩 piana@cognac
🟧 franca@cognac
🟦 andrew@cognac
🟩 igor@cognac
🟦 david@ng2
MPI jobs connected together
Jobs coloured by user
Click on node to reveal statistics
Requested & used time
Sphere size indicates memory usage
Small spheres indicate job state

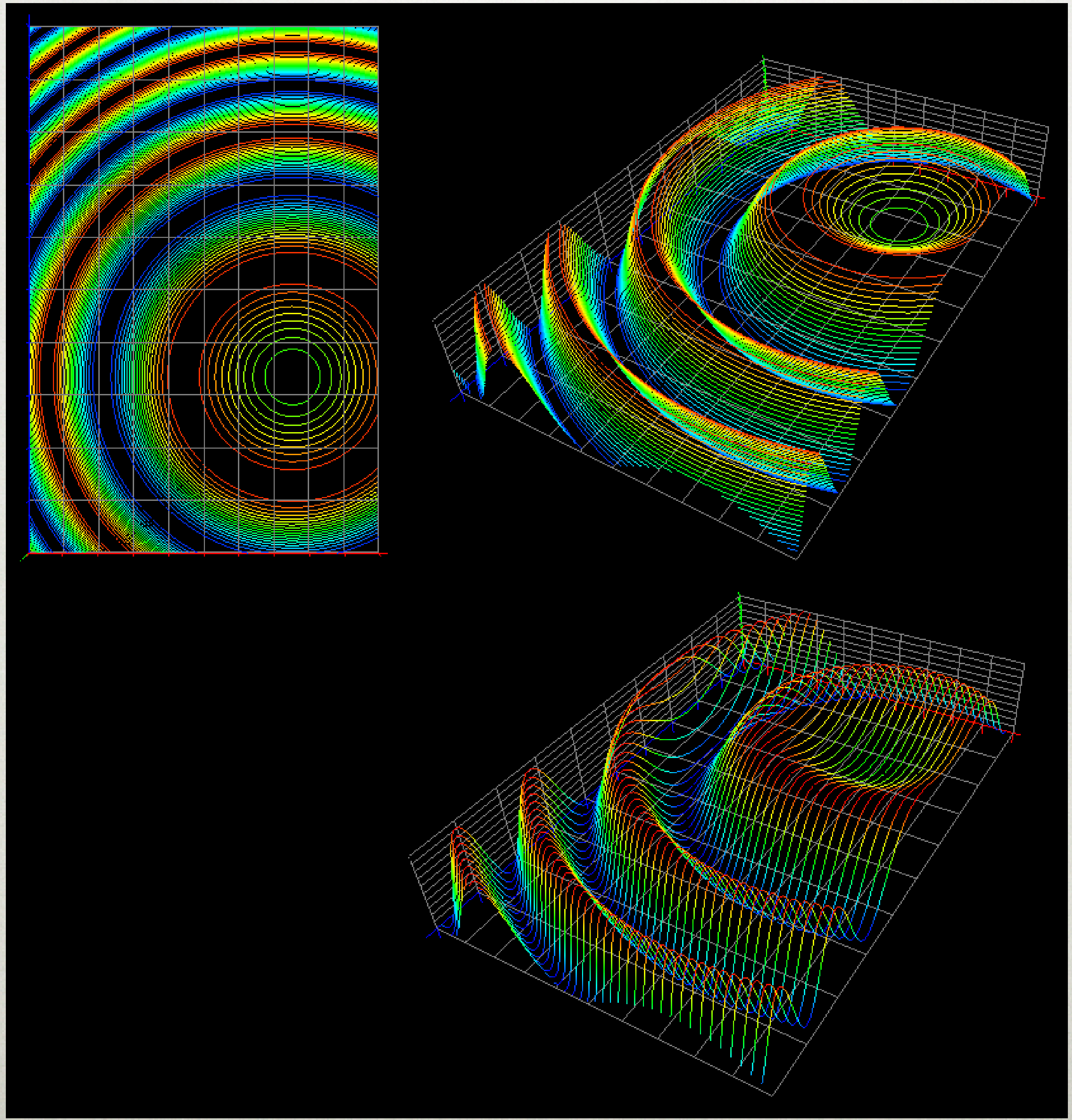
Contours: 3D -> 2D

- Contour is a curve of constant value on a 3D surface.
- A 3D surface is represented with 2D graphical elements, namely lines ... a projection from 3D to 2D.
- A form of dimension reduction we are all familiar with: weather map isobars, landscape height contours, etc.
- Need not be contours in just one direction or parallel to an axis, dimension reduction equally applies to slices in any plane. Also need not be a functional surface.
- Not necessarily used as often today because we have devices that can interactively represent 3D surfaces.

Example

- Simple wave phenomena.

Traditional contour representation

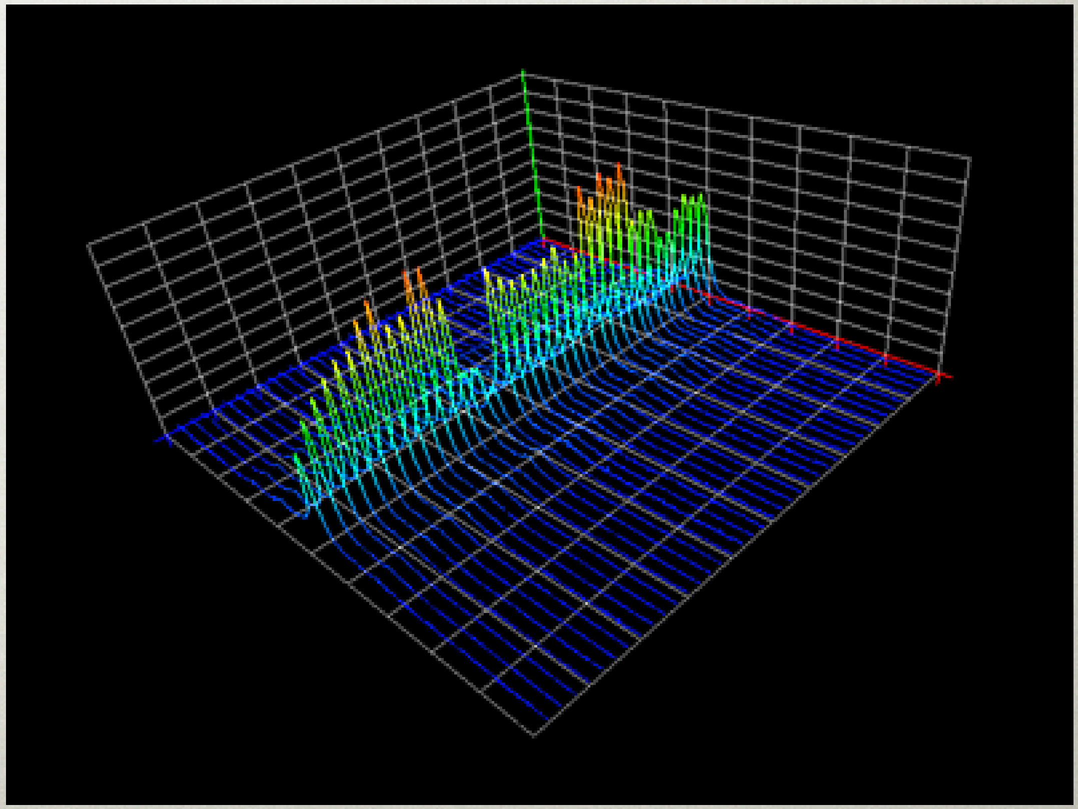


View not perpendicular to contour plane

Contour planes along arbitrary axis

Example

- cd to the pulsar directory
- Inspect raw data in "datastrips"
- Understand what the "combine.c" code does
- Merges data into a single 2D grid file, type "make" and run resulting program "combine"
- Inspect data file "pulsardata.grid"
  grid dimensions nx and ny
  axes spacing for x and y
  height data nx * ny z axis values
- View the surface "glgraph -og pulsardata.grid"
- Experiment changing rendering modes, changing height scaling, colour maps, rendering modes, etc
  (Popup menus with right mouse button)

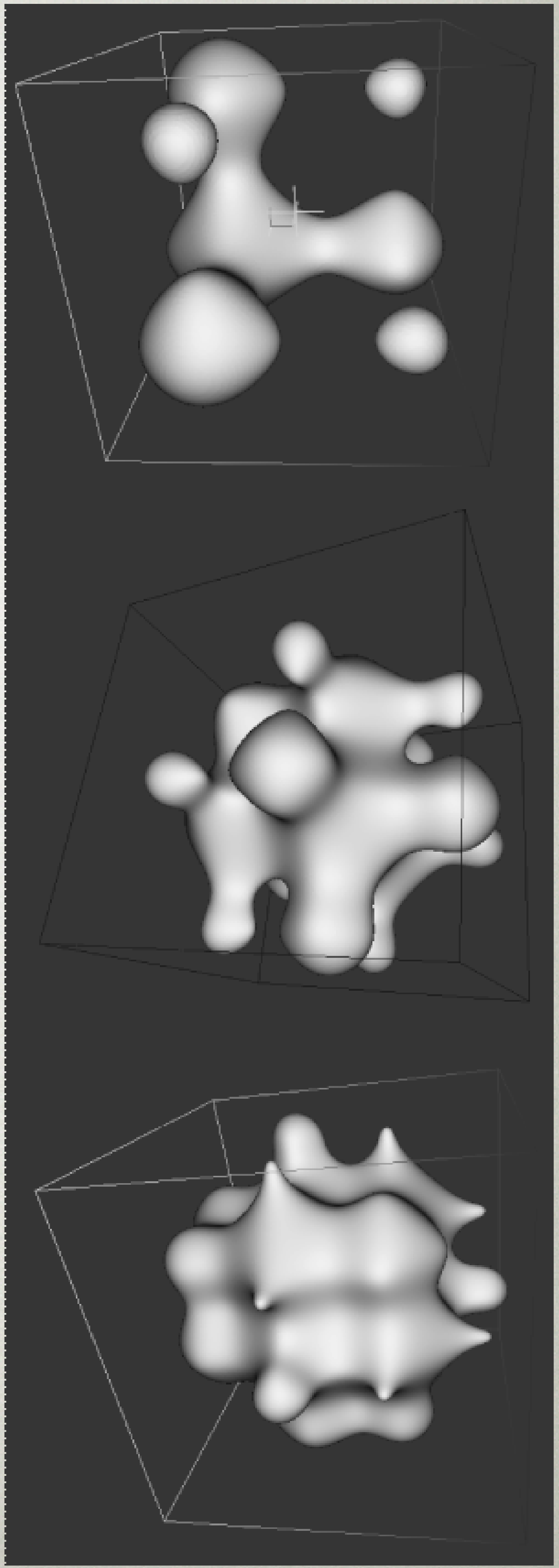


Isosurfaces: 4D -> 3D

- Isosurfaces are the 3D equivalent to 2D contours. Surface of constant value within a volumentric dataset (4D).
- Terminology: voxel - the 3D equivalent of a 2D pixel.
- Standard technique is known as “Marching Cubes”.
- Result is a surface made up of triangular faces that can usually be viewed interactively (OpenGL) or rendered (raytraced).
- The surface representation isn’t necessarily efficient, there are a number of techniques that reduce the triangle count without adversely affecting the usefulness or accuracy of the representation.

Example

- Topology example - “the blob”
  $f(x,y,z) = x^2 + y^2 + z^2 + \sin(4x) + \sin(4y) + \sin(4z) - 1$
- cd to the “blob” directory
- Type “make” to build “makeblob” and create the dataset by running the resulting program “makeblob” that creates “blob.raw”.
- Volumetric dataset is 256x256x256 and each voxel is an unsigned short integer. Total volume size is 35MB, would only be considered an average size volumetric dataset.
- “runiso” will create an isosurface using “polyr”
- View the isosurface using “offviewer”
  “offviewer -o 1 30000.off”
- Right mouse menus to change the shading.
- Create another isosurface at a different value and view it.

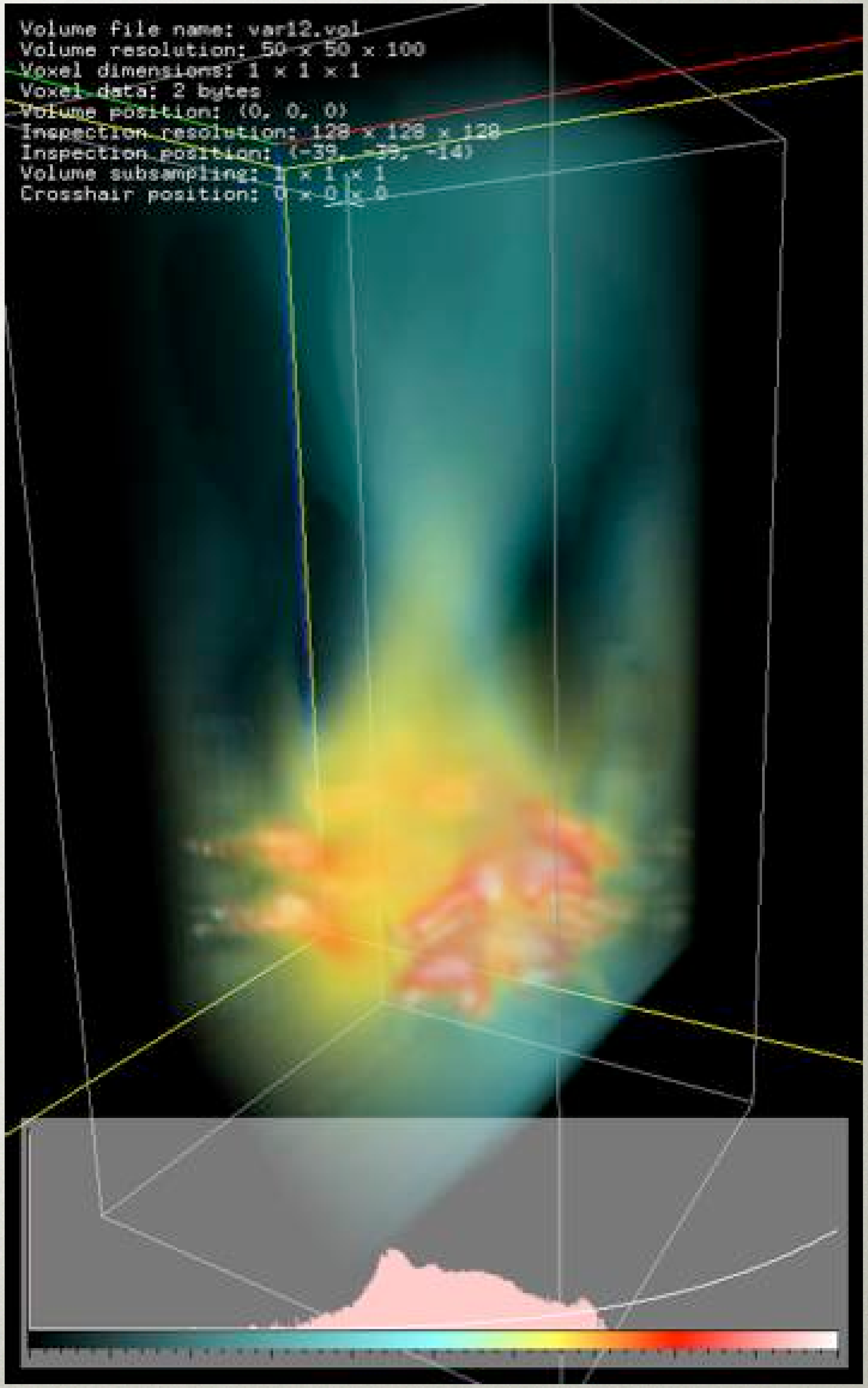


Direct Volume Rendering

- Problem with isosurfaces is that they throw so much information away. OK if there is only one isolevel of interest, eg: skeleton.
- View the whole volume, map voxel values to colour and transparency (opacity), employ rendering model.
- Interactive frame rates are possible with todays graphics cards.
- A number of algorithms available: multiple 2D texture maps, 3D texture maps, ray casting as a shader.
- Improved visual quality by considering gradient not just the voxel values.

Example: coal burning furnaces

- Volumetric data with 12 variables computed from simulations of the furnace behavior. eg: temperature, oxidant, nitrous oxide, pressure, etc.
- Assign colour and opacity to ranges of voxel values, see histogram at base of the image on right.



```
Volume file name: var12.vol
Volume resolution: 50 x 50 x 100
Voxel dimensions: 1 x 1 x 1
Voxel data: 2 bytes
Volume position: (0, 0, 0)
Inspection resolution: 128 x 128 x 128
Inspection position: (-39, -39, -14)
Volume subsampling: 1 x 1 x 1
Crosshair position: 0 x 0 x 0
```

Example

- cd to the furnace directory
- View the furnace temperature volume “glvol temperature.vol”
- Right mouse click and use the “voxel options” display box to change the inspection volume to 64x64x128
- Remove subsampling (set to 1x1x1)
- Using the colour map and opacity designer create informative visualisation of the temperature distribution.
- For more information on the usage of glvol just type the program name with no command line arguments.

```
Usage: glvol [options] volfile
Options
            -h     this text
            -f     full screen
            -s     active stereo
           -ss     dual screen stereo
            -m s   load map file
            -M s   load marker file
            -r n   initial resolution (1,2,3,4)
           -vx n   initial view volume x size
           -vy n   initial view volume y size
           -vz n   initial view volume z size
            -c s   load camera file
            -a     start in auto rotate mode
            -A n   set tween frames for animation
Key Strokes
  arrow keys       move inspection box
       mouse       rotate/roll camera
           h       move camera to home position
           f       focus camera to current inspection cube
           c       center inspection volume
           w       write current window to a TGA file
           W       write frames to TGA file continuously
         i,k       translate, up/down
         j,l       translate left/right
 x,X,y,Y,z,Z       automatic scanning along axes
         <,>       move camera forward/backward
         +,-       change inspection box size
          f1       toggle decoration display
          f2       toggle texture display
          f3       toggle histogram display
          f4       toggle text information
          f5       toggle camera tracking
          f6       toggle autorotate
          f7       toggle 3d cursor
     1,2,3,4       set supsampling level to 1,2,4,8
           q       quit
```

File Formats

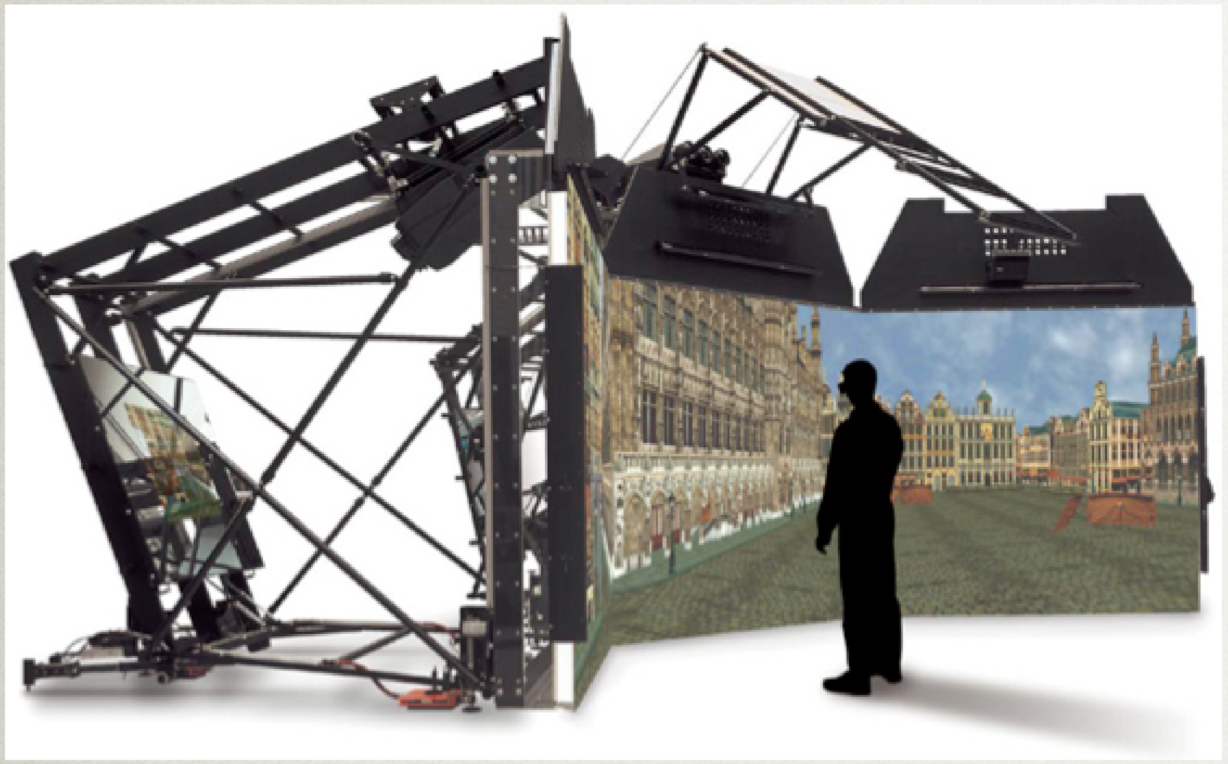
- Researchers like to save data in their own format.
- There are some industry specific standards:
  CDF, FITS, HDF, DEM, VICAR, DICOM, ...
- Libraries usually available to assist with reading/writing.
- Fact of life: a significant amount of time is spent dealing with the reading/writing/conversion of data files.
- Big or small endian.
- Reading Fortran files from C/C++!

Quote: “Standards are good, lets have lots of them.”

Display technologies

- There are some visualisation problems that can benefit from something more than a standard computer display.
- Example 1: Stereoscopic display that presents the viewer with an enhanced sense of depth of 3D geometry. Autostereoscopic display technology "around the corner".
- Example 2: High resolution display for cases when data density exceeds display resolution.
- Example 3: Immersive displays that place the observer inside the data and fills our peripheral vision.
- All about characteristics of our visual system: two eyes horizontally offset and wide field of view.

Display gallery



Software tools: a few examples

- Many packages are domain specific: eg: AIPS++ for Astronomy, RasMol for Chemistry, Visual3 for finite element, Analyze for medical, Vis5D for atmospheric science, VMD and RasMol for molecular visualisation, and so on.
- General commercial toolkits: eg: AVS Express, IRIS Explorer.
- Open source toolkits: OpenDX used to be called the IBM Visualisation Data Explorer.
- Libraries: VTK toolkit, NAG Visualisation software.
- Miscellaneous tools: GeomView, POVRay, GIMP ....

Challenges in Visualisation

- Datasets are getting larger due to:
  - improved imaging technologies (acquisition data)
  - higher performance computing (simulation data)
- Resolution is increasing, data size: 8 -> 16 -> 32 -> 64 ...
- Real time interactivity: ~30+ fps or higher.
- High levels of graphical quality: antialiasing, improved lighting models, higher resolution imagery.
- Solutions often required/requested at the end users desktop rather than in the visualisation laboratory.
- Visual impact for public education, competing with Hollywood.