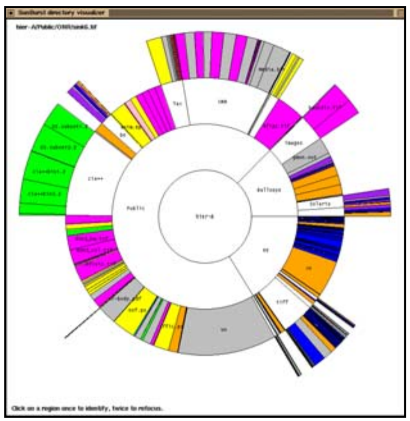
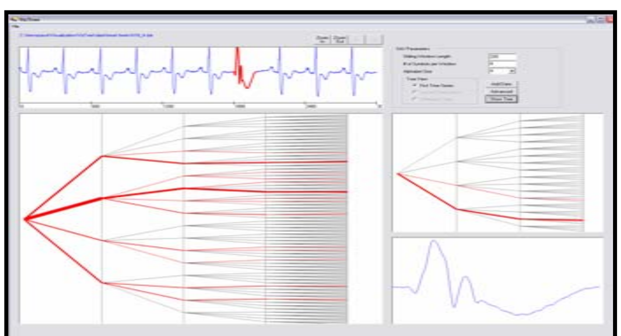
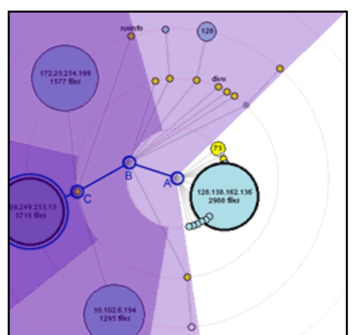
An Overview of Visualization techniques for the Analysis of Large datasets

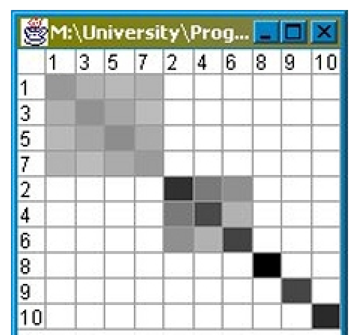


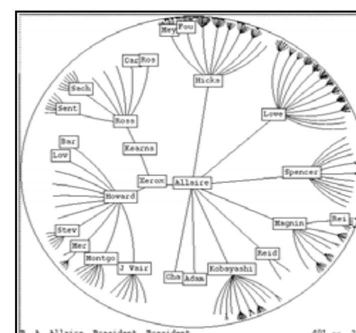
Liz Stuart
The Visualization Lab
University of Plymouth

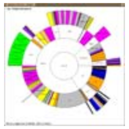
www.plymouth.ac.uk/infovis





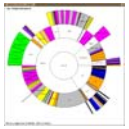






Overview

- Visualization
- Information Visualization (IV)
- Data Types
  - Trees
  - Multidimensional data
- Case Study from Neurophysiology
- Conclusions

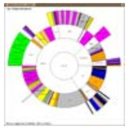


Visualization – a mental process

- Visual adj. (Oxford English Dictionary)
  - Of or used in seeing

- Visualize v. (also –ise)(-zing or –sing)
  - Imagine visually

- Definition by Vandoni 1989
  - “the process of **transforming information** into a visual form, enabling users to **observe** the information. The resulting visual display enables the scientist or engineer to **perceive** visually **features** which are **hidden** in the data but nevertheless are needed for **data analysis** and exploration”



The Visualization Umbrella

From the **literature** the **scope** of visualisation is clearly diverse and incorporates, or intersects with, nine substantial areas of computer science research and development:

- animation
- software visualisation
- visualisation environments
- volume rendering and visualization
- computer graphics
- visual programming
- image processing
- virtual reality
- **Scientific Visualisation** (interesting)
- **Information Visualization** (approx. 15 years)

These may apply to data, program execution or code

Scientific Visualization

- Definition
  - Use of computer-supported interactive **visual** representation of data to **amplify cognition**

- Representation of **data** with physical ‘shape’
  - the data items **already** have a 2D/3D structure
  - **not abstract** data representation
  - **additional** information needs to be represented
  - data needs to be made **accessible**

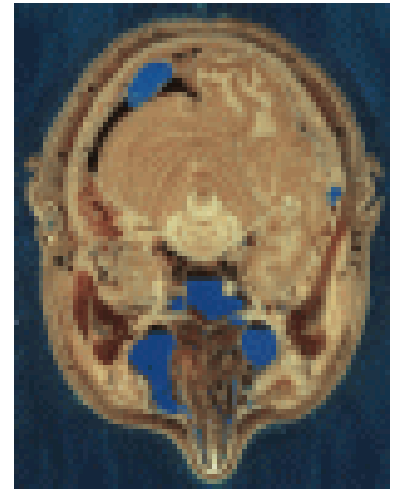
Visible Human 1

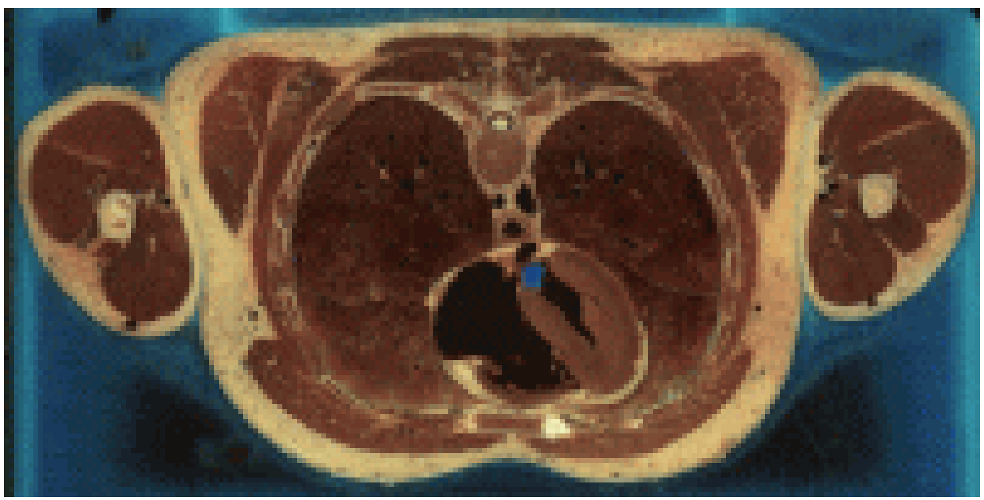
- Dynamic image of the **human body**
- Constructed form a set of **cross-section photographs**
  - Body frozen
  - Sliced
  - Photographed
- Images combined to form **3D** version for **exploration**

http://www.nlm.nih.gov/research/visible/visible_human.html

Visible Human 2

A section through Visible Human Male - head, including cerebellum, cerebral cortex, brainstem, nasal passages

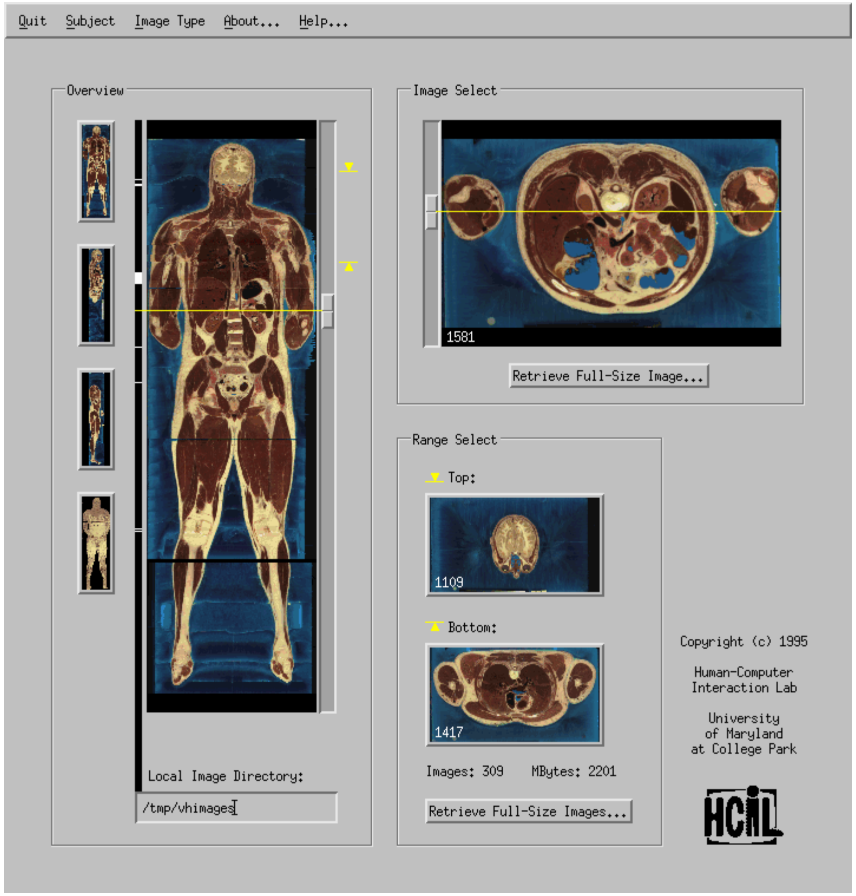




The thorax, including heart, lungs, spinal column, major vessels, musculature

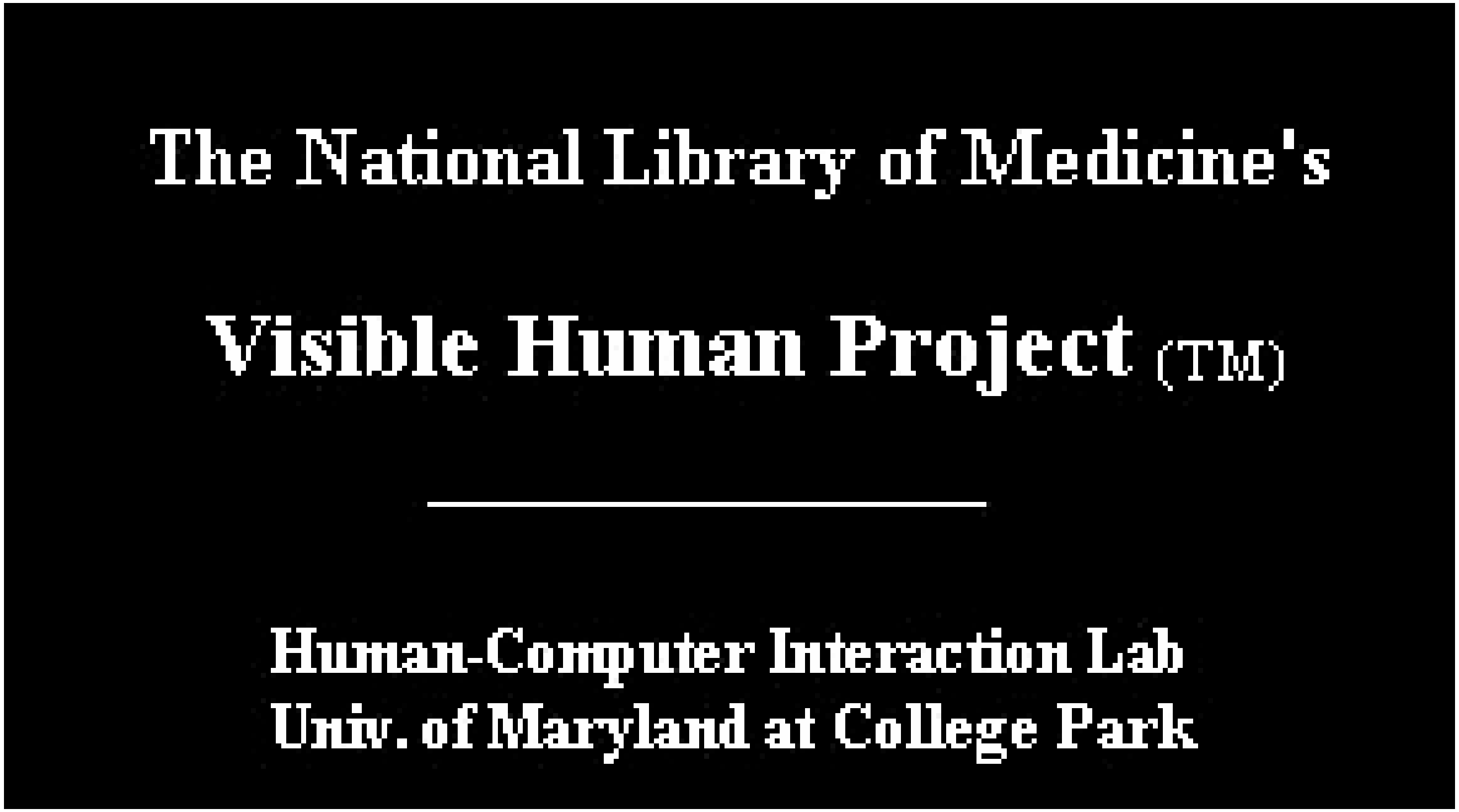
http://www.nlm.nih.gov/research/visible/visible_human.html

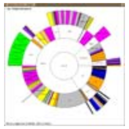
Visible Human - GUI



http://www.cs.umd.edu/hcil/

Visible Human - animation

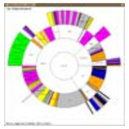




Information Visualization

Definition – The use of computer-supported interactive **visual** representation of **abstract** data to amplify **cognition**

- Compact graphical presentation
  - Large number of items
  - Extracted from even larger dataset
- Enable users to make:
  - Discoveries
  - Decisions
  - Explanations
- Relates to:
  - Patterns (trends, clusters, gaps, outliers)
  - Groups
  - Individual items

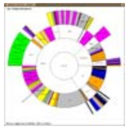


Shneiderman's Mantra

overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand
overview, zoom, filter, details-on-demand

Our informal design code

Shneiderman 1996 - The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations, Proceedings of the 1996 IEEE Symposium on Visual Languages

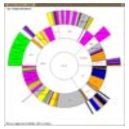


Shneiderman’s Mantra

- **Overview** **-** Gain an overview of the entire collection
- **Zoom** **-** Zoom in of items of interest
- **Filter** **-** Filter out uninteresting items
- **Details-on-demand** **-** Select an item or group and get details when needed

... and ...

- **Relate** **-** View relationships among items
- **History** **-** Keep a history of actions to support undo, redo and progressive refinement
- **Extract** **-** Allow extraction of sub-collections and the query parameters

Data Types

- 1D Linear
- 2D Map
- 3D World
- Network
- Workspace
- **Tree**
- **Multi-Dim**

1D Linear, 2D Map, 3D World — Traditional Visualization problems

Network — Uses 2d/3d, Animation, Distortion

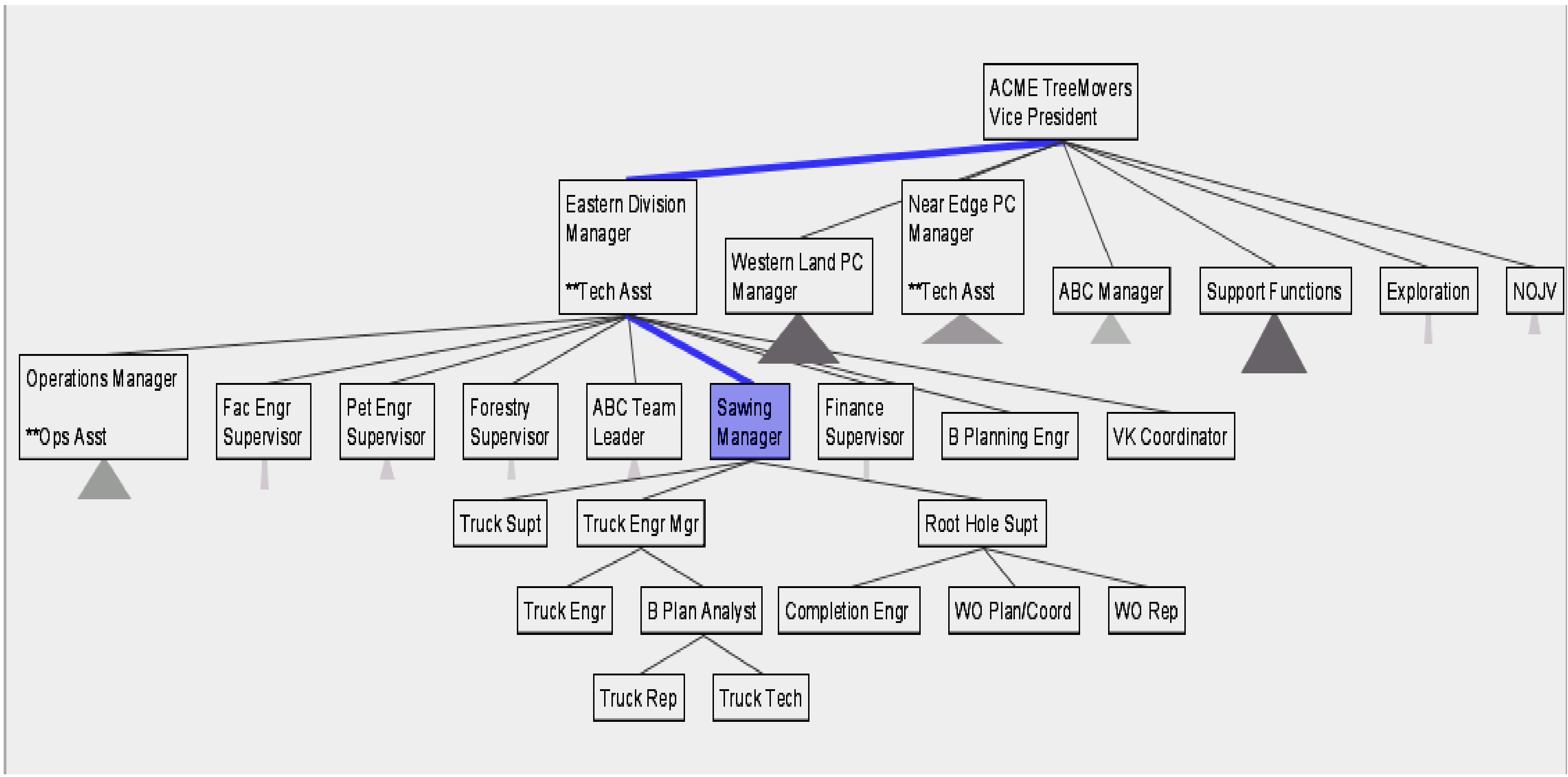
Workspace — Better display organisation (elastic windows)

Tree, Multi-Dim — Interesting challenges!

Each data type requires **different** considerations for representation

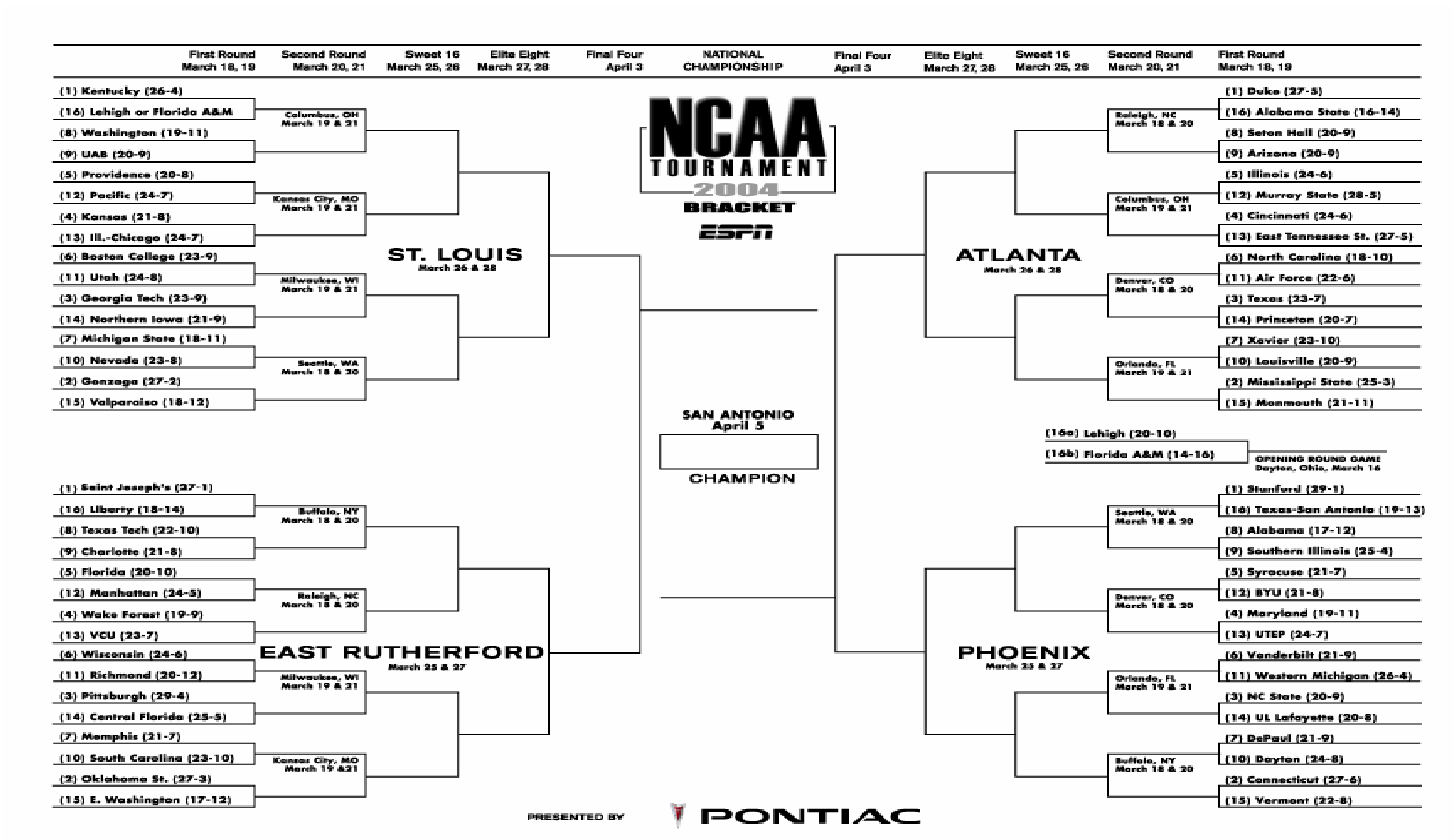
http://otal.umd.edu/Olive/

Tree Data 1 – organizational charts



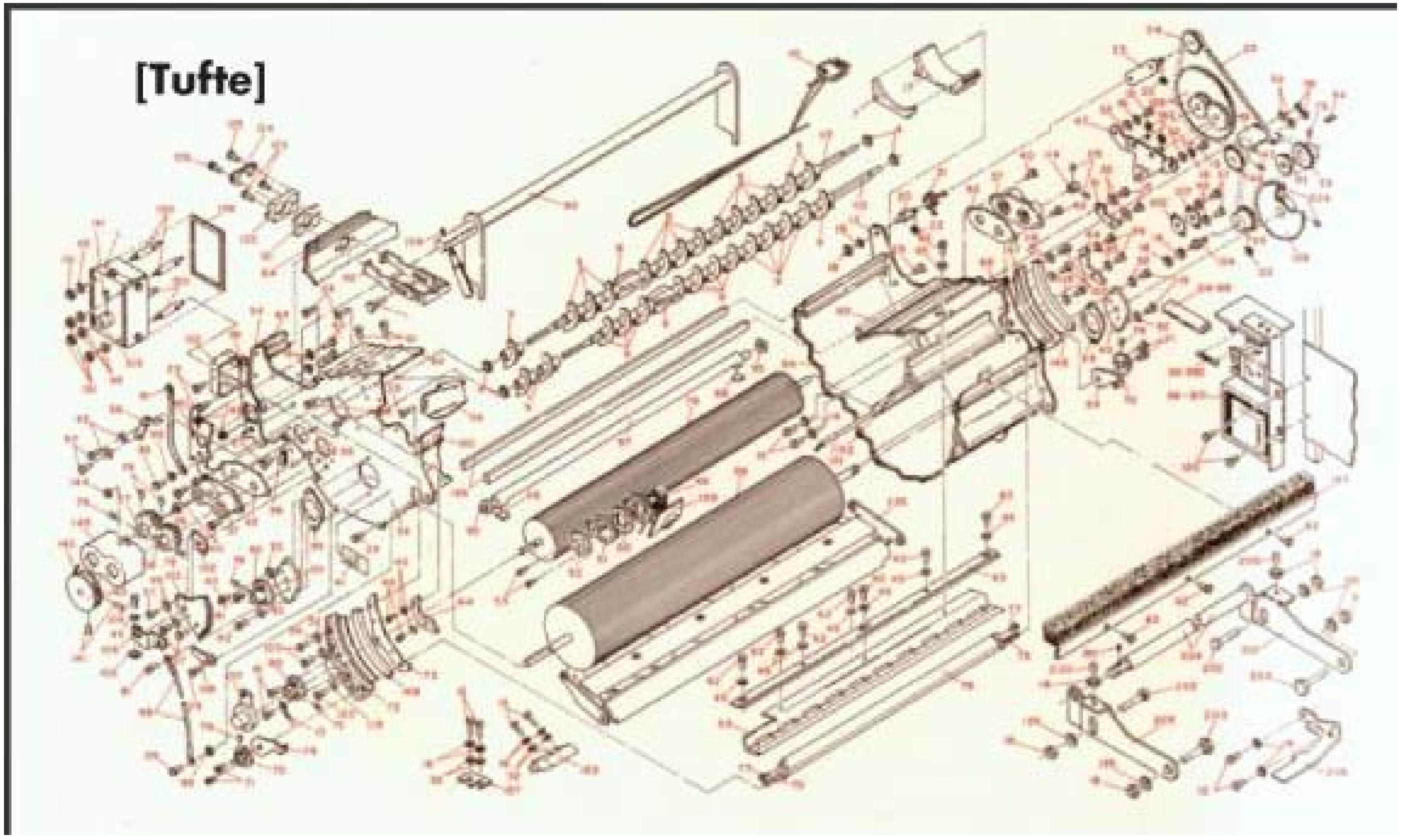
http://www.cs.umd.edu/hcil/spacetree/

Tree Data 2 - tournaments



http://graphics.stanford.edu/~hanrahan/talks/todrawatree/

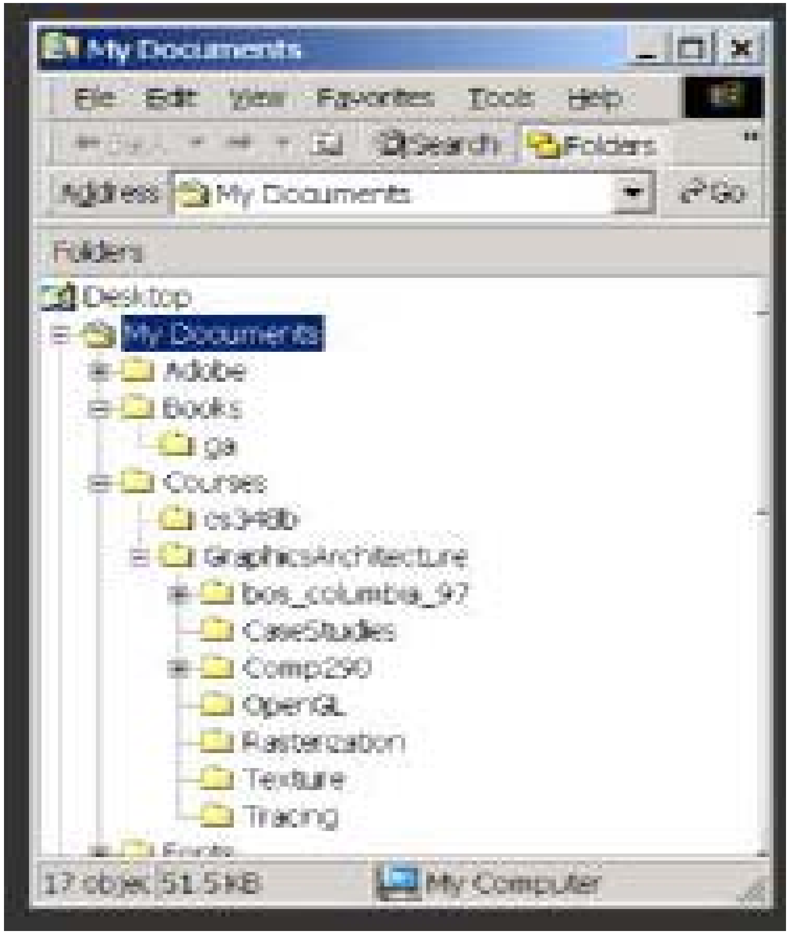
Tree Data 3 – assembly instructions

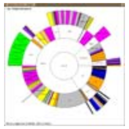


http://graphics.stanford.edu/~hanrahan/talks/todrawatree/

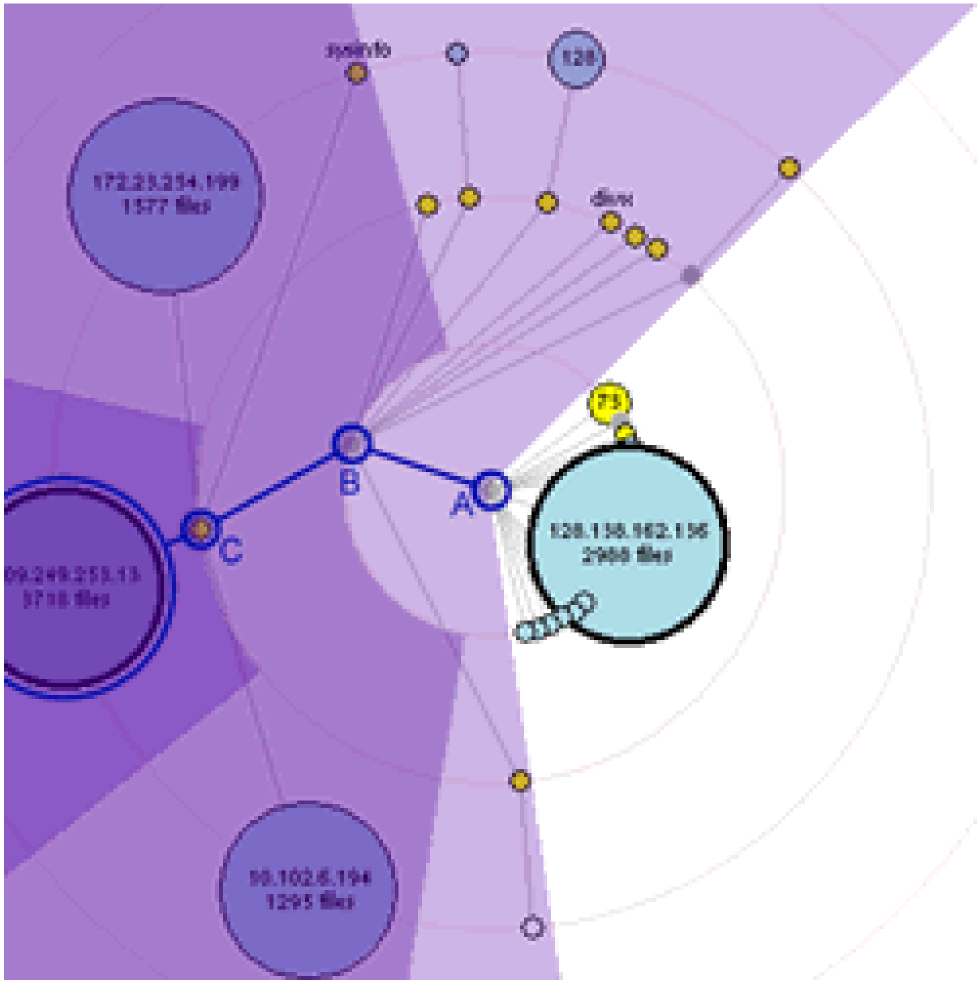
Tree Visualization

- File hierarchy in computer
- Need to navigate through the hierarchy to find a particular file
- Two common problems: “Where am I?” and “Where is the file that I am looking for?”





Radial Layout

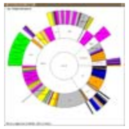


- Places children on increasing larger radii
- A top-down layout converted into polar co-ordinates
- breadth --> angle
  depth --> radius

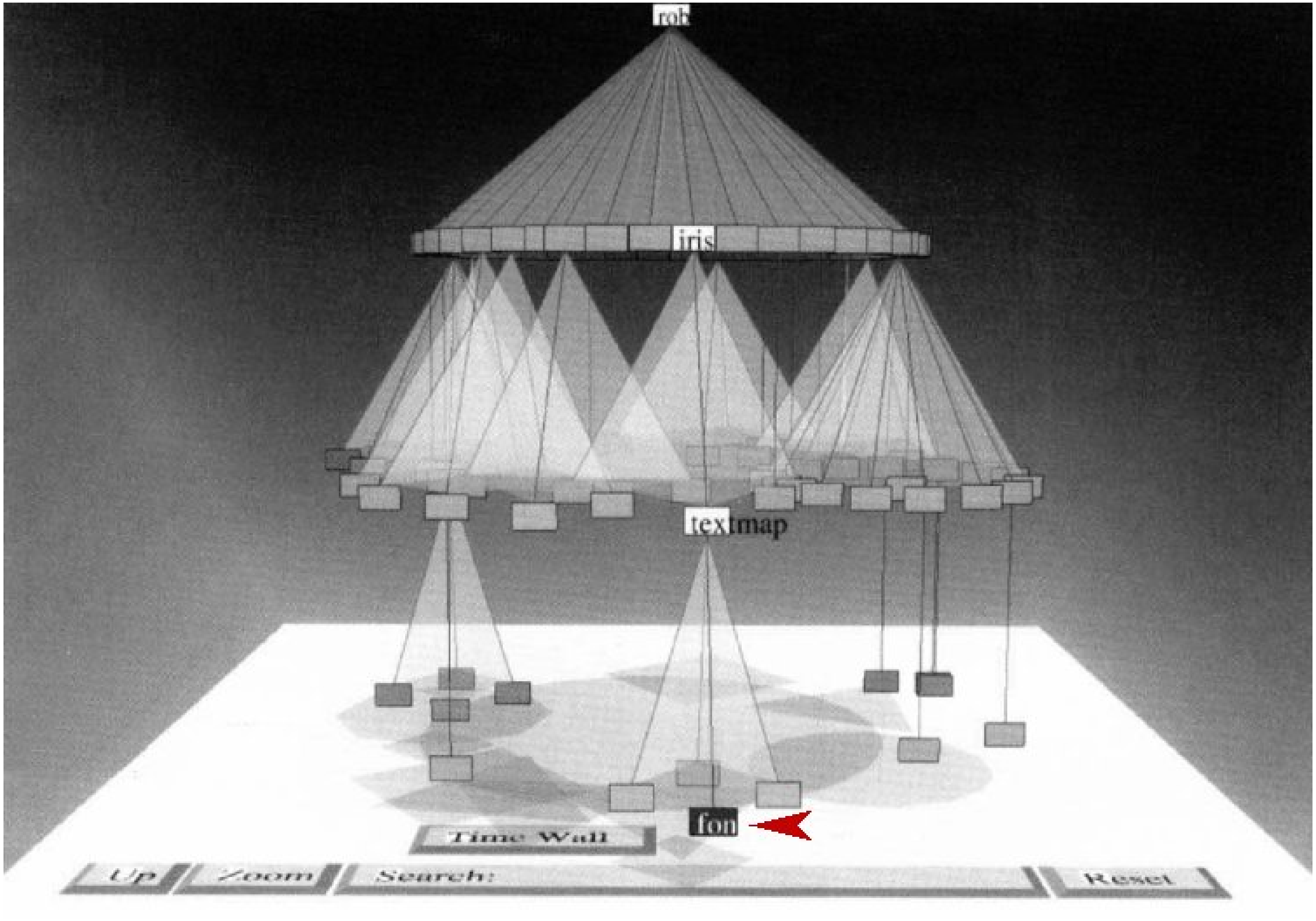
http://bailando.sims.berkeley.edu/papers/infovis01.htm

Cone Trees

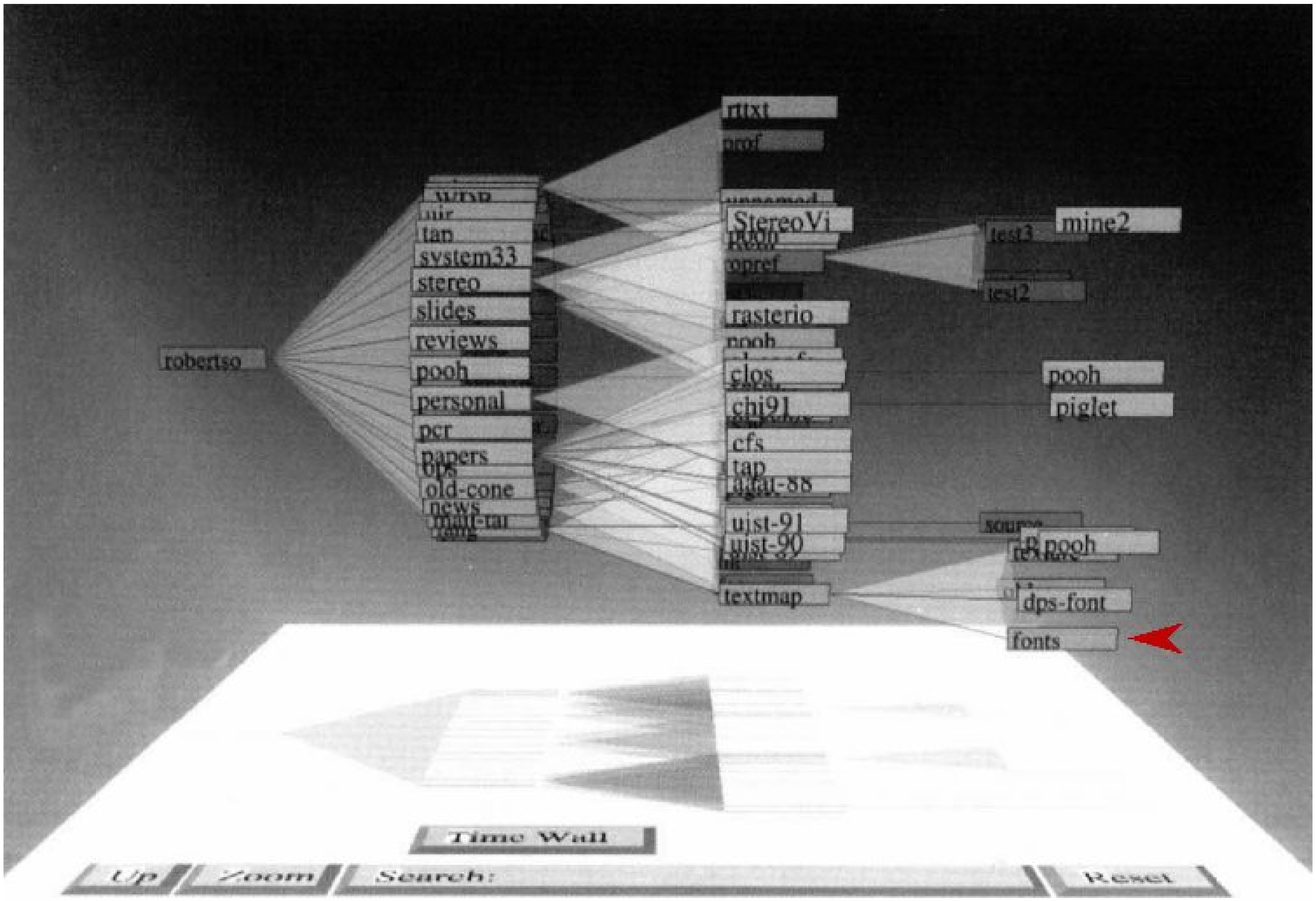
- **3D** representation of **hierarchy**
  - more nodes than 2D
- **Smooth animation** to exploit perceptual tracking
  - object constancy
- **Occlusion** problems
- **Limit** to number of nodes clearly displayed
- Rotation keep **changing the view** of the data



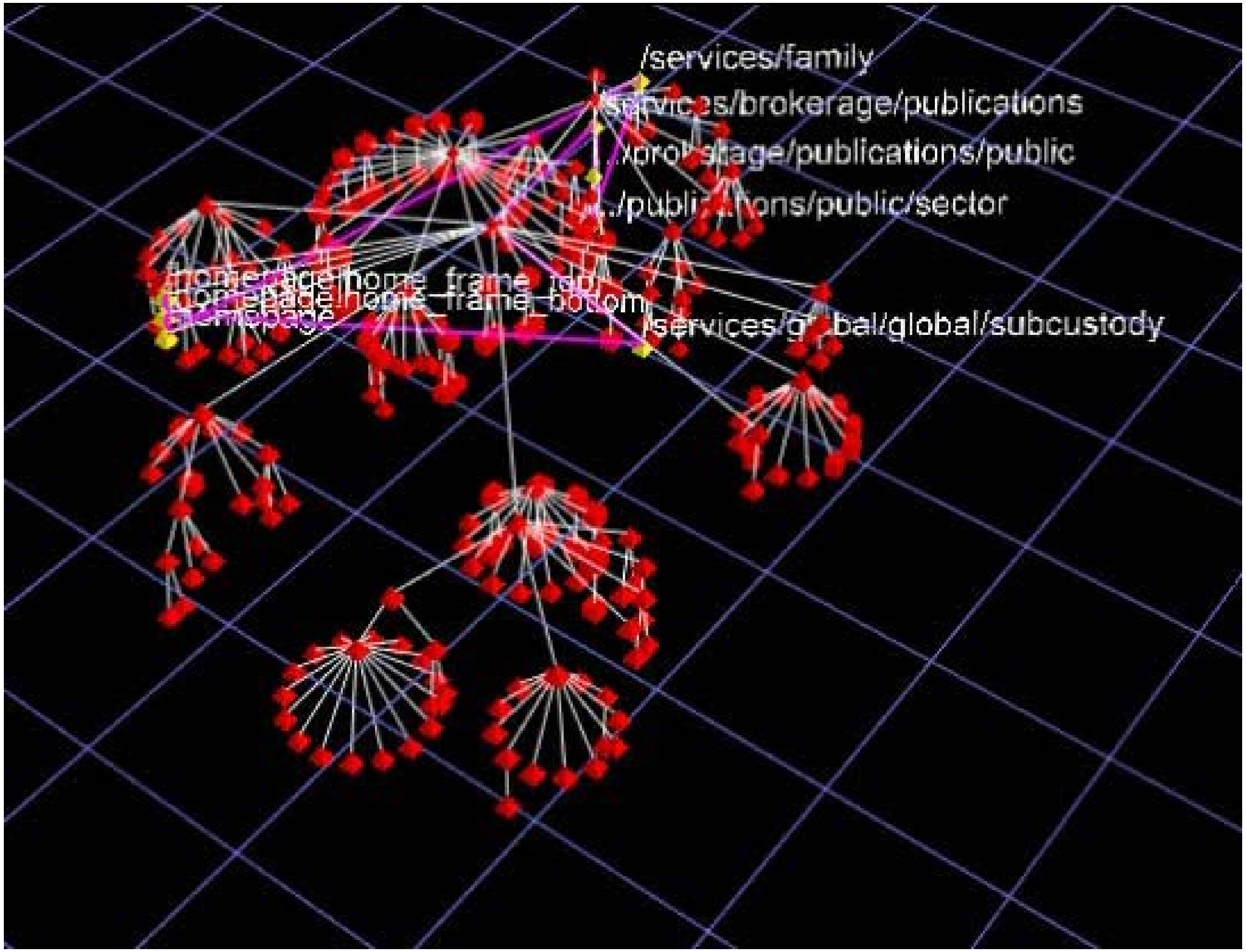
Cone Trees – vertical

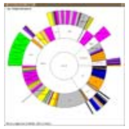


Cone Trees - horizontal



Cone Trees - animation



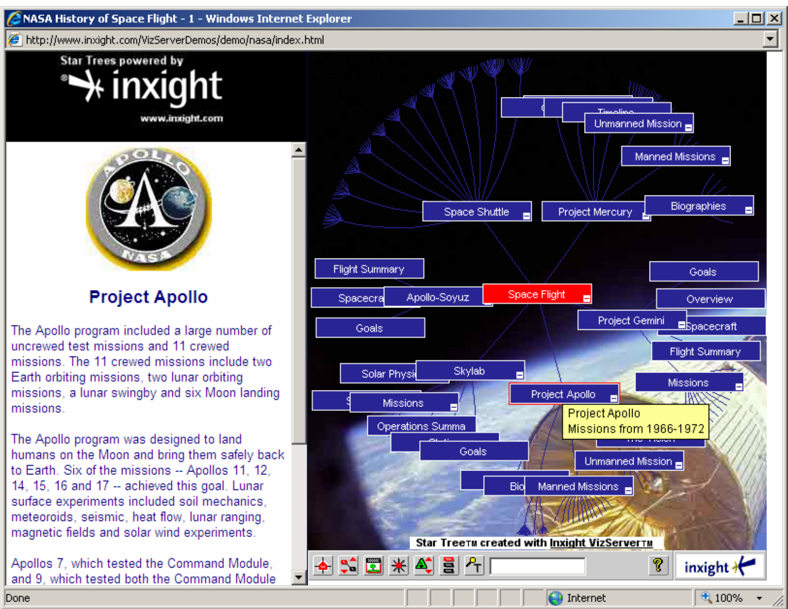


Hyperbolic Browser

- **Focus** and **Context** view
  - Fish Eye
  - Area of high magnification to **examine detail**
  - Maintain **context** with ‘whole’ dataset

Hyperbolic Browser

- Nodes are positioned using hyperbolic geometry
- Subsequently, nodes are projected into “normal” space



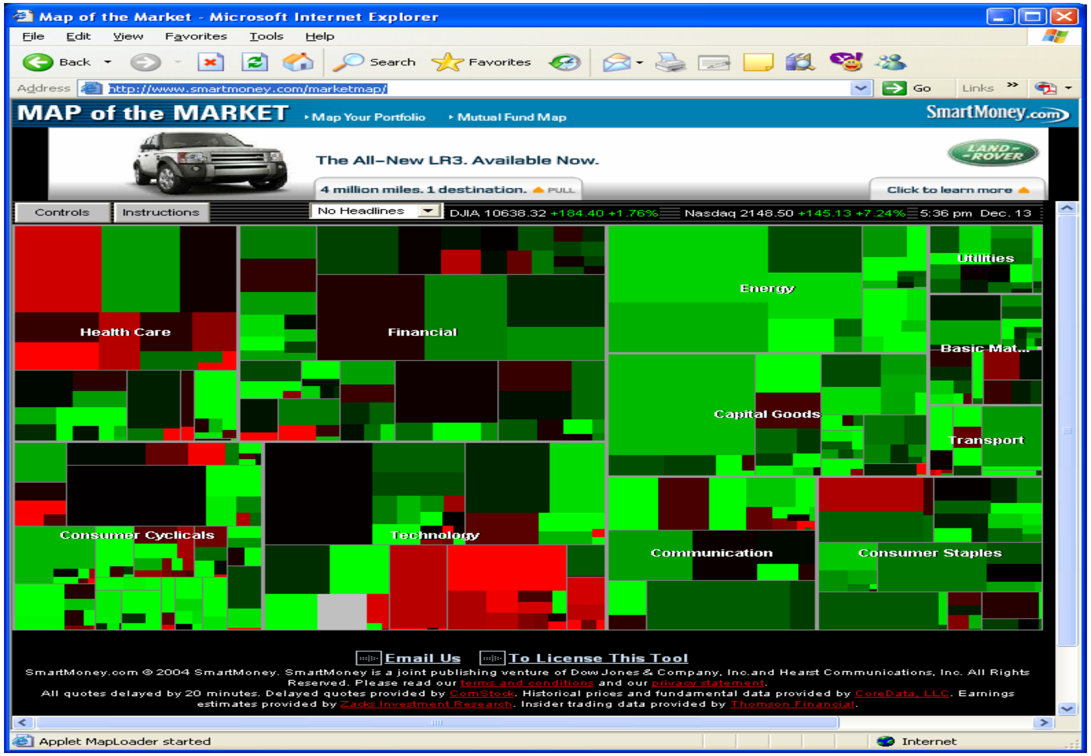
- NASA – History of Space flight

http://www.inxight.com/VizServerDemos/demo/nasa/index.html

Treemaps

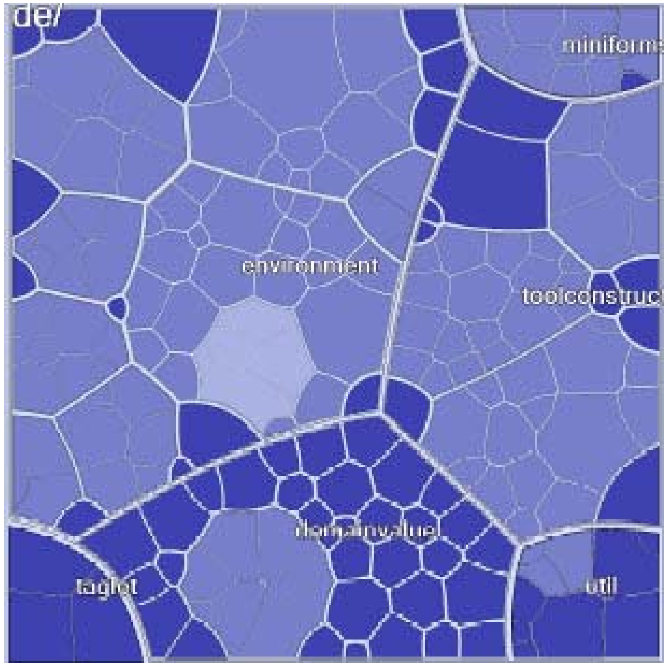
- Hierarchical, grouped and non hierarchical data
- **2D space-filing representation**
  - Each node represented by a block
  - Each level represented by a frame
- **Size** and **colour** of block can represent data attributes

SmartMoney

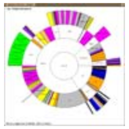


You can see **SmartMoney** live at http://www.smartmoney.com/marketmap/

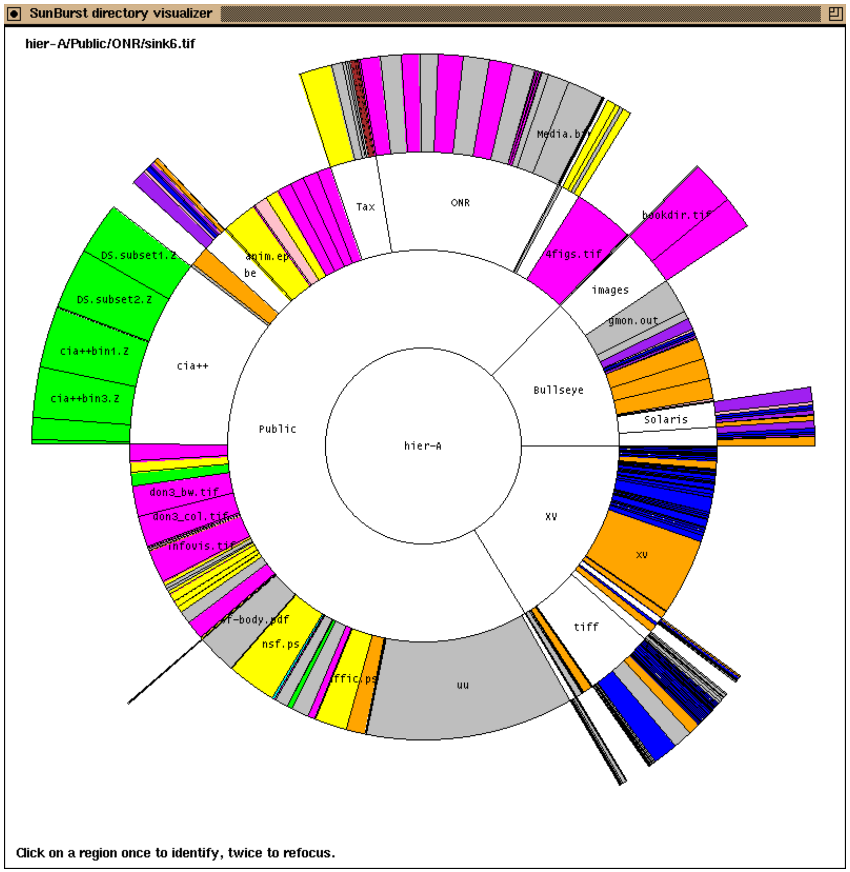
Voronoi Treemaps



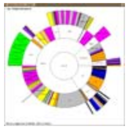
- variation of the treemap
- displays software metrics
- uses arbitrary polygons instead of rectangles
- The layouts are computed by the iterative relaxation of Voronoi tessellations.



Sunburst Diagrams



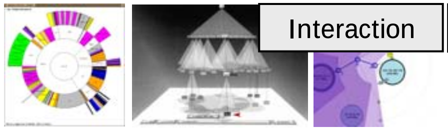
- a radial, rather than planar, layout of the tree.
- provides more room to display information about non-leaf nodes than traditional treemaps
- a line from any node to the center of the SunBurst map will intersect all ancestors of that node.

Multidimensional Data/Visualization

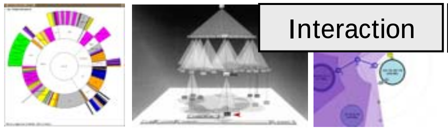
- Multi-dimensional information visualizations present **data** that is **not primarily spatial**.
- The number of **attributes** of a given item in the collection is **more than three**.
- Example applications of multi-dimensional visualization schemes may use
  - **stock market** statistics
  - **factory** production line data sets
  - a set of books in a **library**
  - a movie **database**
  - almost any abstract and statistical information about any phenomenon.

http://www.otal.umd.edu/Olive/Multi-D.html



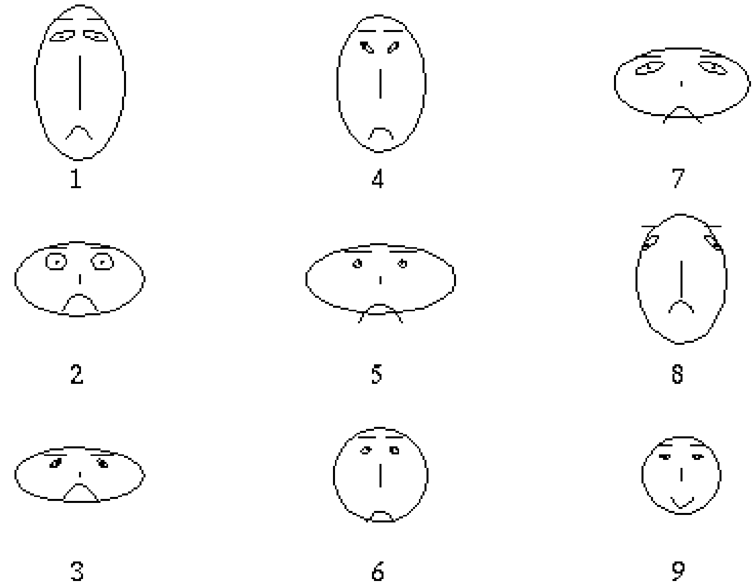
Glyphs

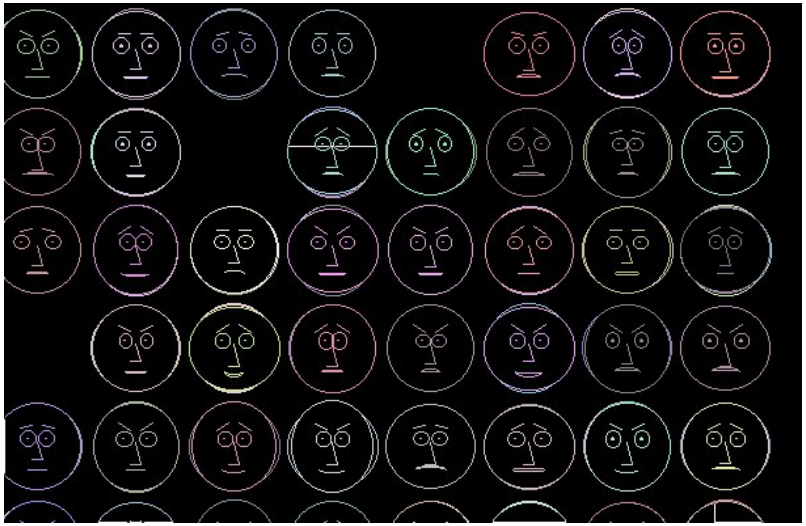
- A dot is **not always** a dot
- Data **attributes** can be **encoded** into the geometric and presentation primitives of an object
- For example
  - Colour = heat (as seen earlier)
  - Shape = type
  - Size = modification date
- Chernoff faces
- Star plots



Chernoff faces

- Chernoff faces map **data** to **facial** characteristics.
- Since humans are particularly well-suited to note **differences in human faces** and facial expressions, this seems like it **takes advantage** of our natural powers of perception.

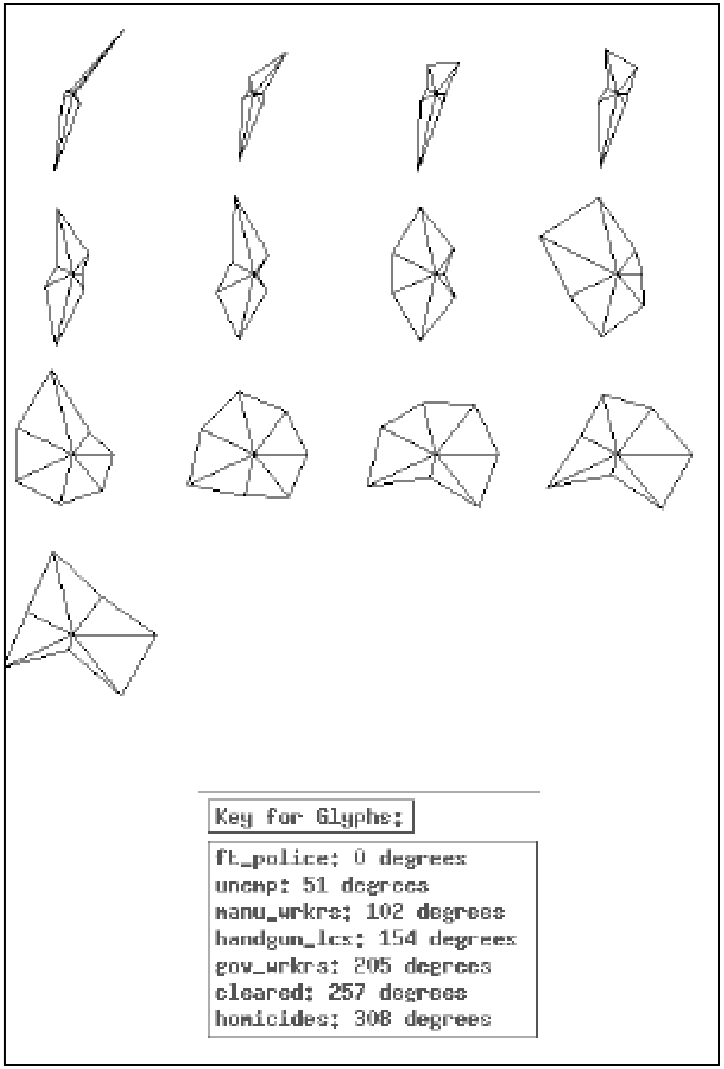




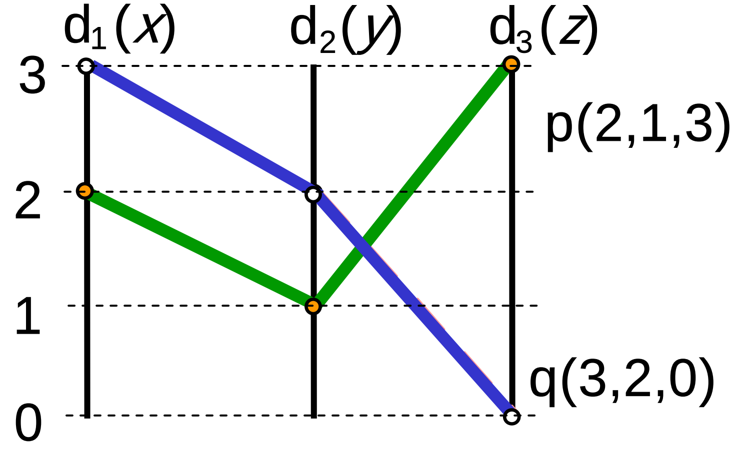
Star Plots

- Star plots
  - Each observation represented as a ‘star’
  - Each spike represents a variable
  - Length of spike indicates the value

Crime in Detroit

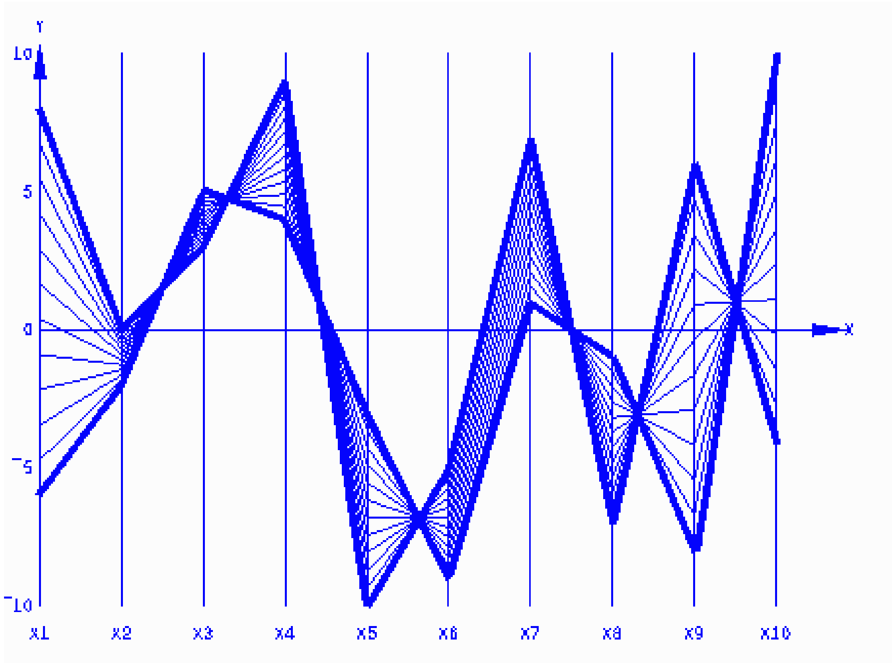


Parallel Coordinates



- The **diagram** shows the three dimensional coordinates **p(2,1,3)** and **q(3,2,0)** as parallel coordinates (pc's) as opposed to standard Cartesian coordinates
- Each point is **plotted** on the three **dimensions** (axes) and a **line** is drawn to connect these intersection points thus, creating the parallel coordinate.
- Parallel coordinates are often used in contrast to **Cartesian** coordinates as they can easily **expand beyond** three dimensions.
- Thus, they are very useful for **multivariate** data analysis.

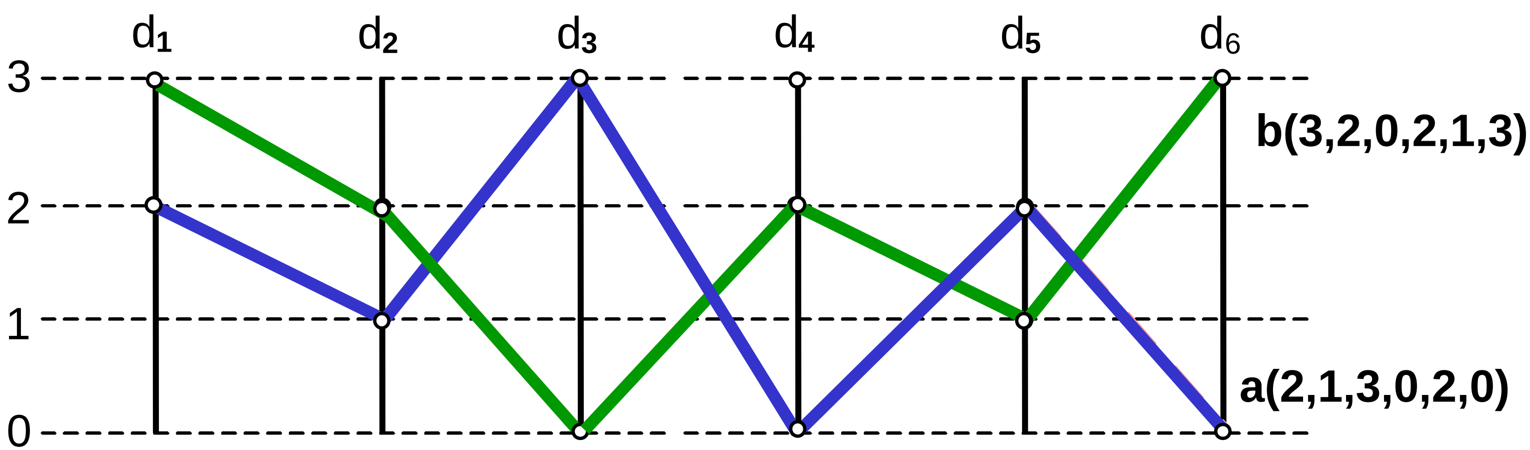
Parallel Coordinates



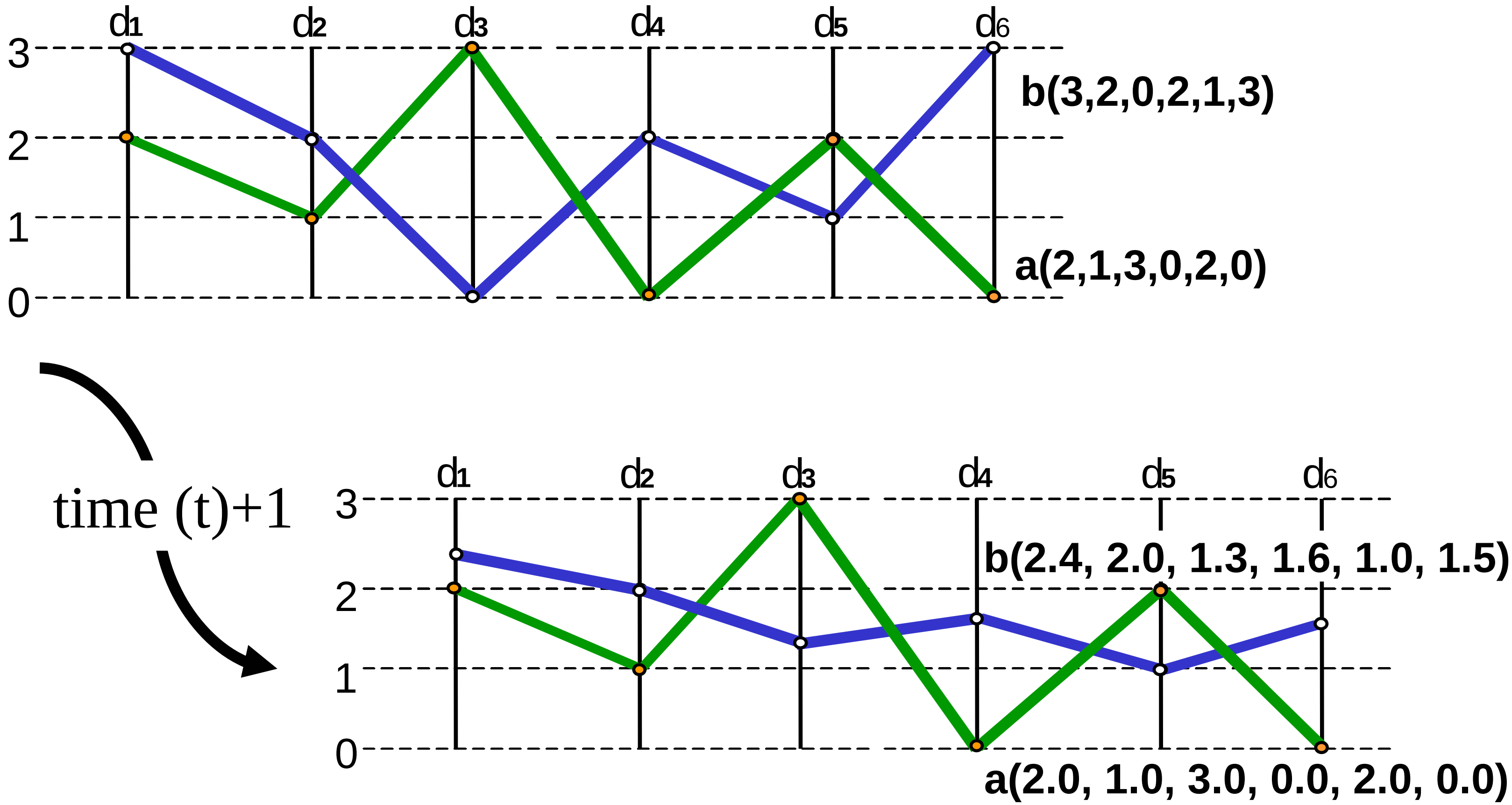
- One of the greatest strengths of parallel coordinates is their ability to, relatively simply, **represent statistical relationships** such as correlation and clustering
- A **positive** correlation is depicted as lines **converging** to a point somewhere **outside** the axes. A **negative** correlation is graphed as **converging** lines to a single point **between** the two axes corresponding to the two variables
- Note the importance of the **order of the axes** in determining relationships!

Using PC to show movement in *n*d space

- We can do 3d animation but **what about** 10d data, 15d data and beyond this ...?
- We are using parallel coordinates to **represent multi-dimensional space**: 10 dimensional space, 15 dimensional space, higher ...
- See the diagram showing the points "**a**" and "**b**" in a six dimensional space

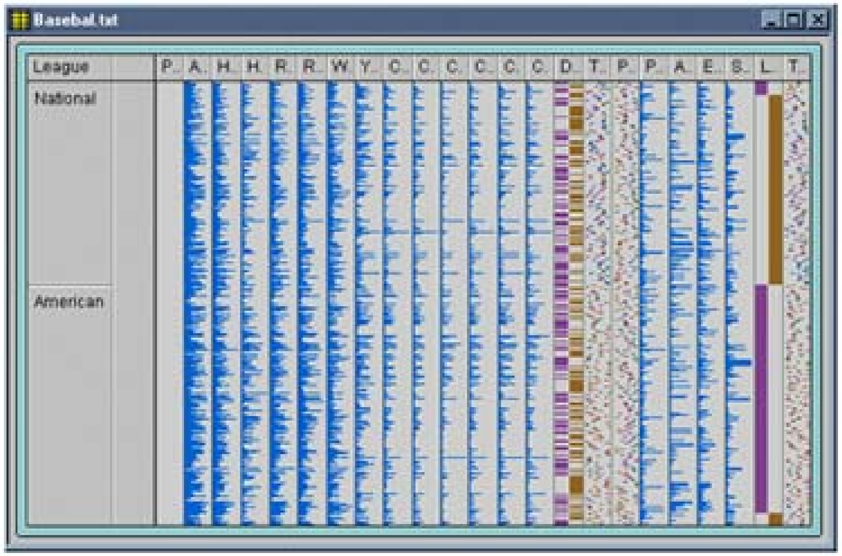


Animation of *n*d Space



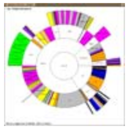
- Note the **intersection / crossover** of lines does not inherently infer collisions

TableLens

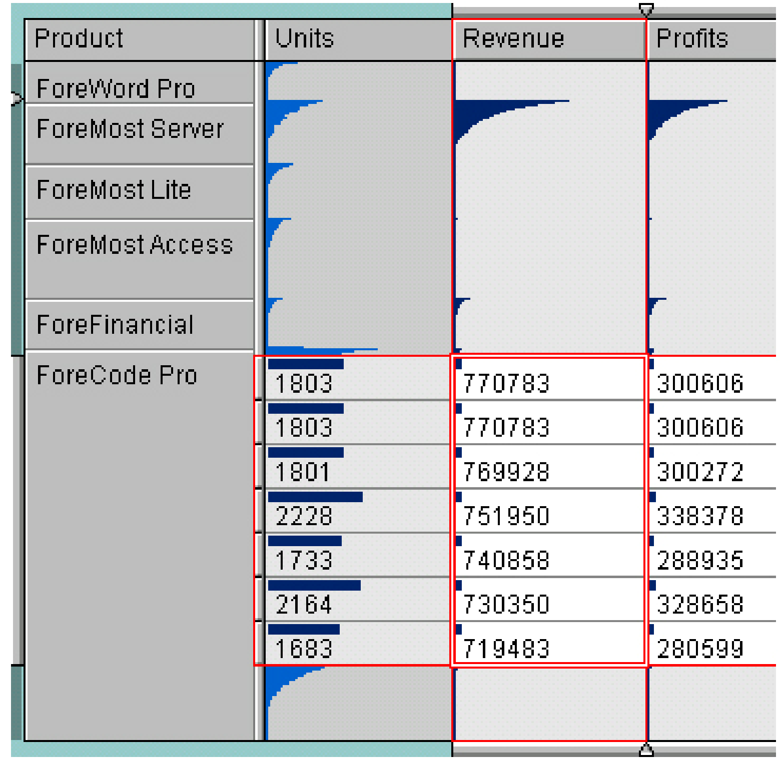


- Used to visualize large amounts of **tabular data** at once
- Displays **record-oriented data** in columns and rows.
- But instead of displaying numbers and words, **TableLens fills the cells** with scaled and coloured horizontal bars.
- You **see much more data** - So where a spreadsheet might be able to show 50 rows and 20 columns on a 19-inch screen, TableLens can show up to 1,000 rows and 40 to 50 columns, without scrollbars, and without obscuring any data points.

http://www.inxight.com/demos/tl_flightdelay/tl_flightdelay.html

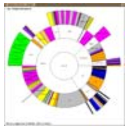


TableLens



- TableLens can also be used as a **presentation and reporting tool**.
- Users can:
  - **Sort** by clicking on columns.
  - **Rearrange** columns by drag-and-drop.
  - **Promote** columns to create subgroups.
  - **Focus** by clicking on a cell or by clicking and dragging to “focus” a whole row or multiple rows.
  - **Filter** subsets to create smaller more specific datasets.
  - **Spotlight** data (rows/columns) to track particular information as you sort.

http://www.inxight.com/demos/tl_flightdelay/tl_flightdelay.html

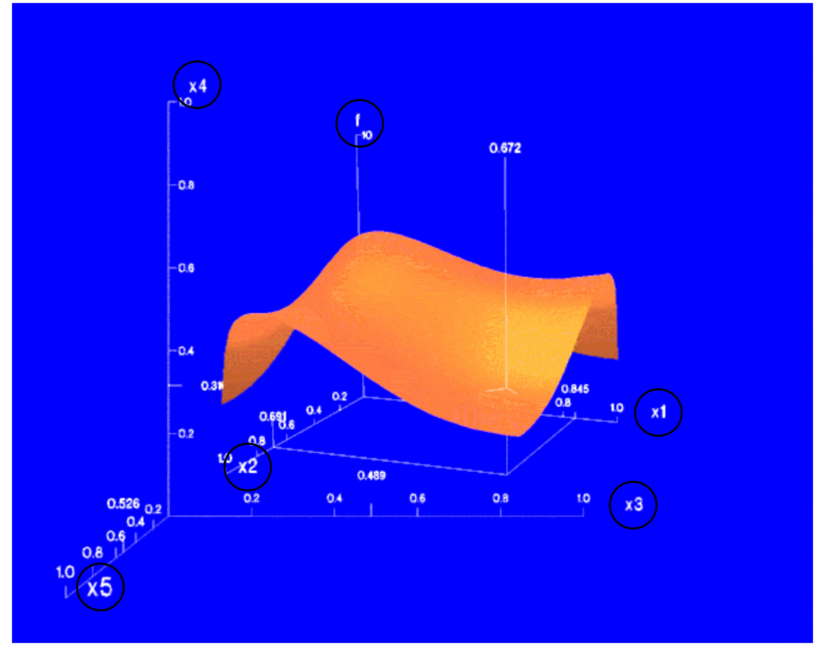


Worlds within worlds

- ***AutoVisual*** designs interactive virtual worlds for visualizing and exploring multivariate relations.
- ***n-Vision***, a real-time, 3D, interactive visualization system
- The primary interaction metaphor in *n*-Vision is **worlds within worlds**, an interactive visualization technique that exploits nested, heterogeneous coordinate systems to map multiple variables onto each of the three spatial dimensions.
- Worlds within worlds is a powerful method, yet it is readily understandable

http://www1.cs.columbia.edu/graphics/projects/AutoVisual/AutoVisual.html#figure_dipstick

An example world



- Consider the function

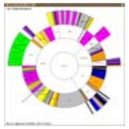
  f (x1, x2, x3, x4, x5)

- Following the description above, we first select **constant values** for three variables **x3, x4, x5**, call them c3, c4, c5 giving

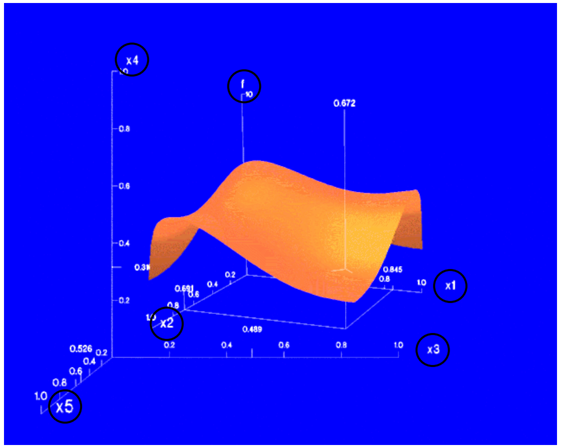
  f '(x1, x2) = f (x1, x2, c3, c4, c5)

- The function f ' is visualized as a surface plot, with x1 on the X-axis, x2 on the Z-axis, and the value of the function on the vertical axis (Y-axis).

http://www1.cs.columbia.edu/graphics/projects/AutoVisual/AutoVisual.html#figure_dipstick



Example continued ...

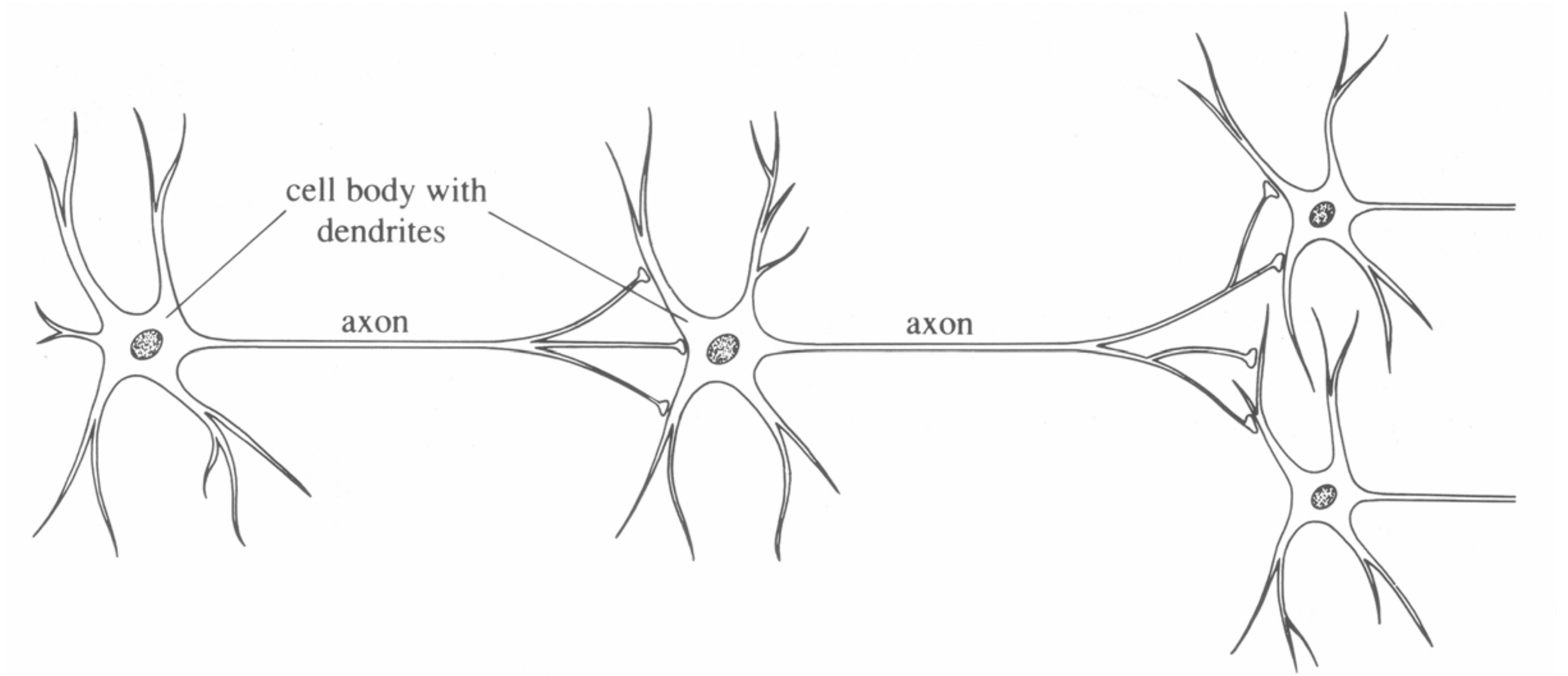


- To let the user select particular values for x3, x4 and x5, they are graphed on a **separate set of axes**, bound to the X, Y and Z axes respectively.
- **Selecting a point** within this larger graph **determines** the particular **values of c3, c4 and c5** used in the smaller graph.
- Thus, the **contents of the smaller graph** depend on the location of some **interactive mark** in the larger graph.
- We represent this dependency explicitly by attaching the **origin** of the **smaller** graph (the surface plot) to the **interactive point** in the **larger** graph.
- To change the values of x3, x4 and x5, the user grabs the **surface plot** and **translates** it relative to the larger world. That is, the surface plot is nested within the graph of x3, x4 and x5.

http://www1.cs.columbia.edu/graphics/projects/AutoVisual/AutoVisual.html#figure_dipstick

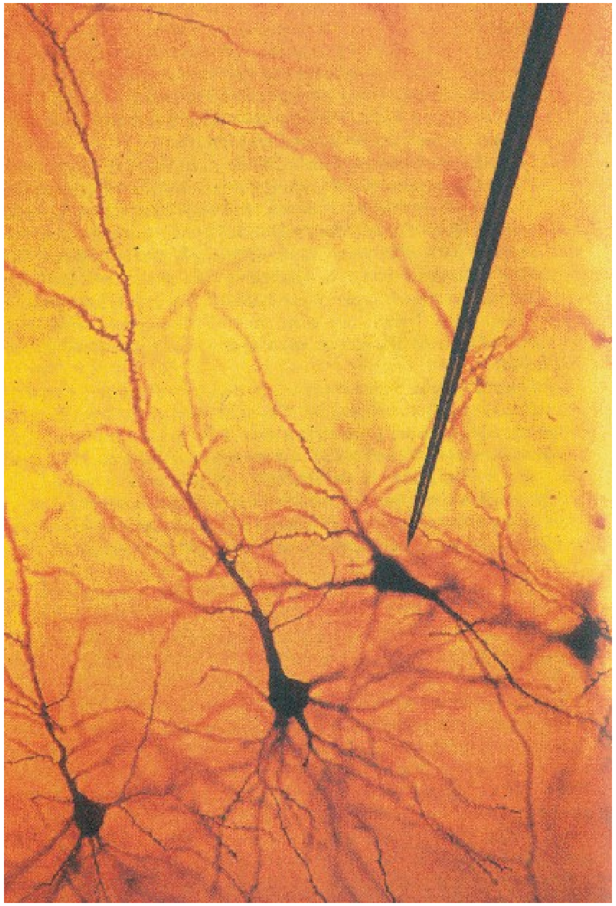
Case Study – some background

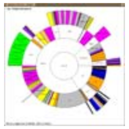
- Neurons (aka nerve cells) **carry information** in the brain
- **Dendrites** - **receive** information into the cell
- **Axons** – **pass out** information (via the cell body) via axon
- For communication between neurons to occur, an electrical impulse travels down an axon to the synaptic terminal



Case Study – spikes and spike trains

- At the synapse, neurotransmitters trigger **"go"** signals that **allow** the message to be passed to the next neuron
- At this point we say that the neuron **SPIKES** (passes the message to the next neuron)
- Neurons also produce **"stop"** signals that **prevent** the message from being forwarded.
- The recording of a neuron’s activity over time is known as the **SPIKE TRAIN**.

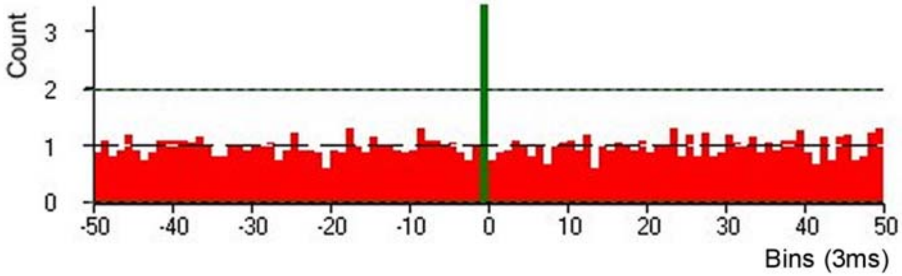




Multidimensional spike train datasets

- Neurophysiologists want to understand more about **information processing** in the **brain**
- The human brain contains **approx. 10^{10} neurons**
- Temporal Coding – where connected neurons have **temporally** correlated spike trains
  - **But we want to know more about this temporal synchrony !!!**
- No longer **limited by hardware** ability to record – it has increased to 100's simultaneously = massive quantities of data (**gigabytes, terabytes**)
- When numerous neurons are recorded simultaneously, we record **multi-dimensional spike train datasets**. `

Cross Correlation



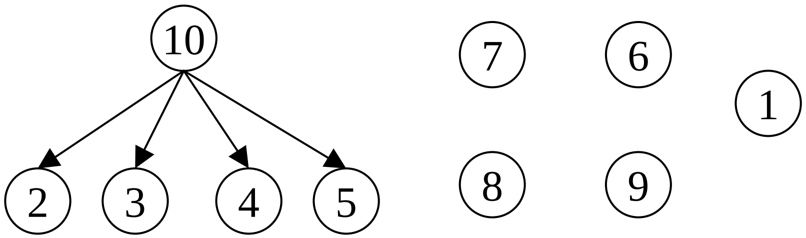
- Classic method for **identifying temporal synchrony** between **two** spike trains
- Analysis (histogram of spikes) of how one train (**target**) fires compared to another (**reference**)
- Correlation time frame specified
  - Number of equal time segments (**bins**)
- Normalise [**Brillinger** 1979]
- Inspect for **significant** peak(s) in **correlogram** (shown above)

So what's the Problem?

- Need to **identify temporal correlation** (if any) for all pairs in data file
  - Many, many, cross-correlograms
  - Must **examine all** to 'understand' any relationships
  - Very time-consuming
  - Difficult to '**see**' the pattern
- Need a better way...

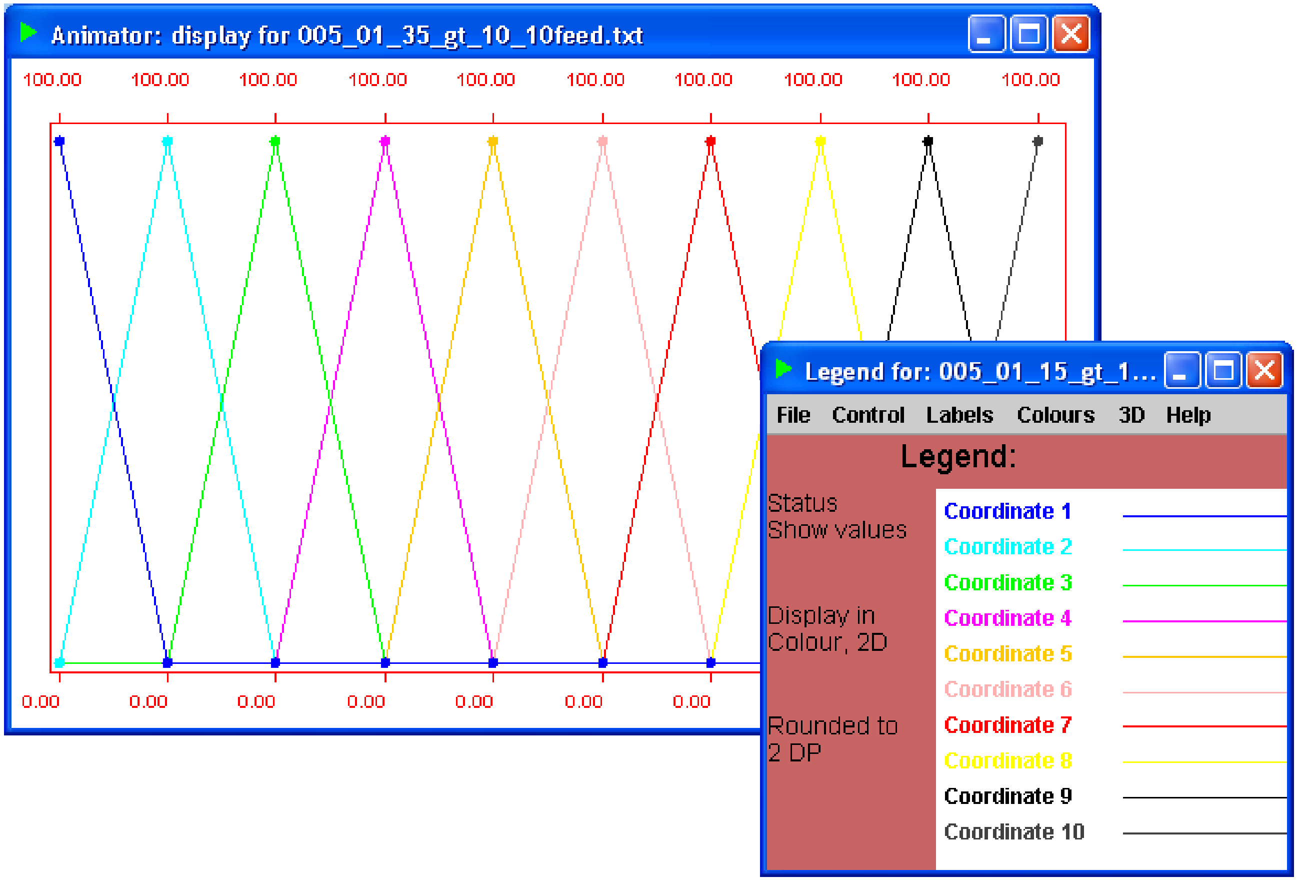
Solution 1: Using Parallel coordinates

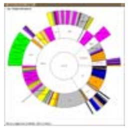
- **Animate** the n (10) dimensional **activity** of the data from the **Gravity Transform**
- For example, assume 10 neurons in the **topology** shown:



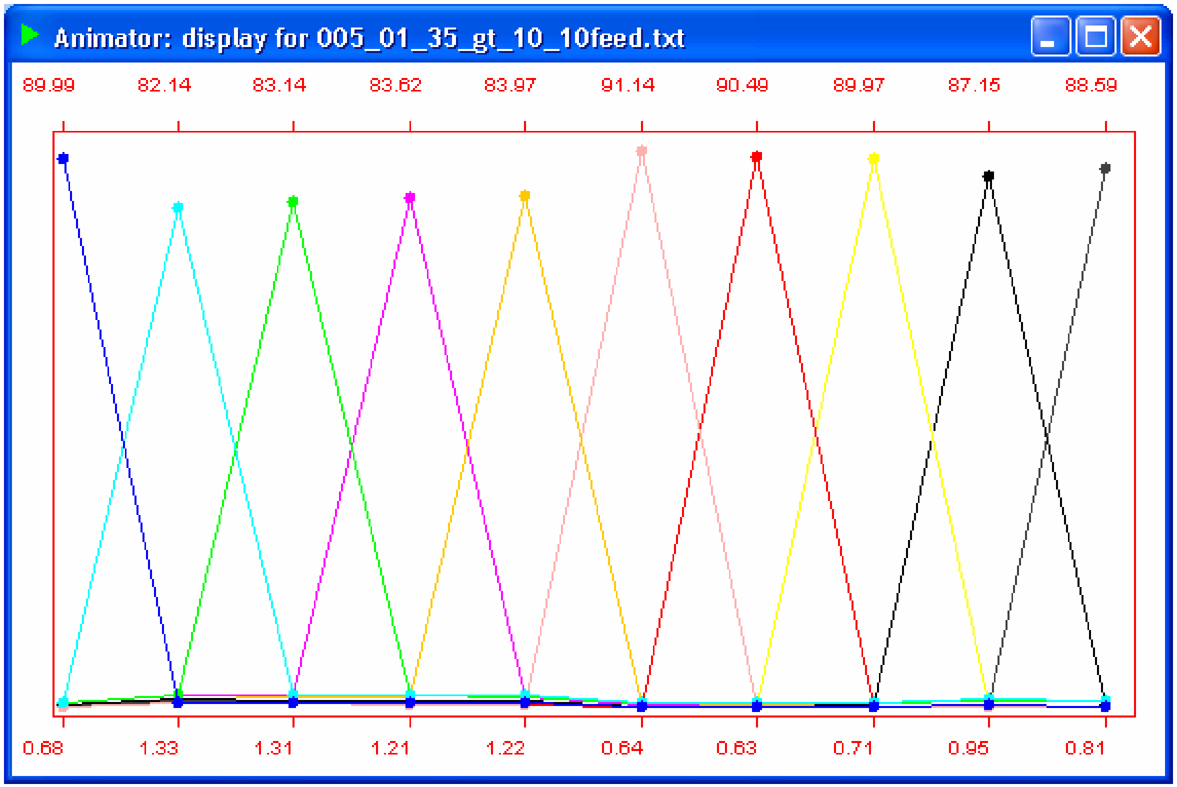
- Note that **neuron 10** stimulates neurons 2–5 inclusive and that **neurons 1, 6-9 inclusive and 10** receive no stimulation from any other neuron.
- Thus we have **two groups of neurons** receiving significantly different levels of stimulation.

Animator: Iteration 1



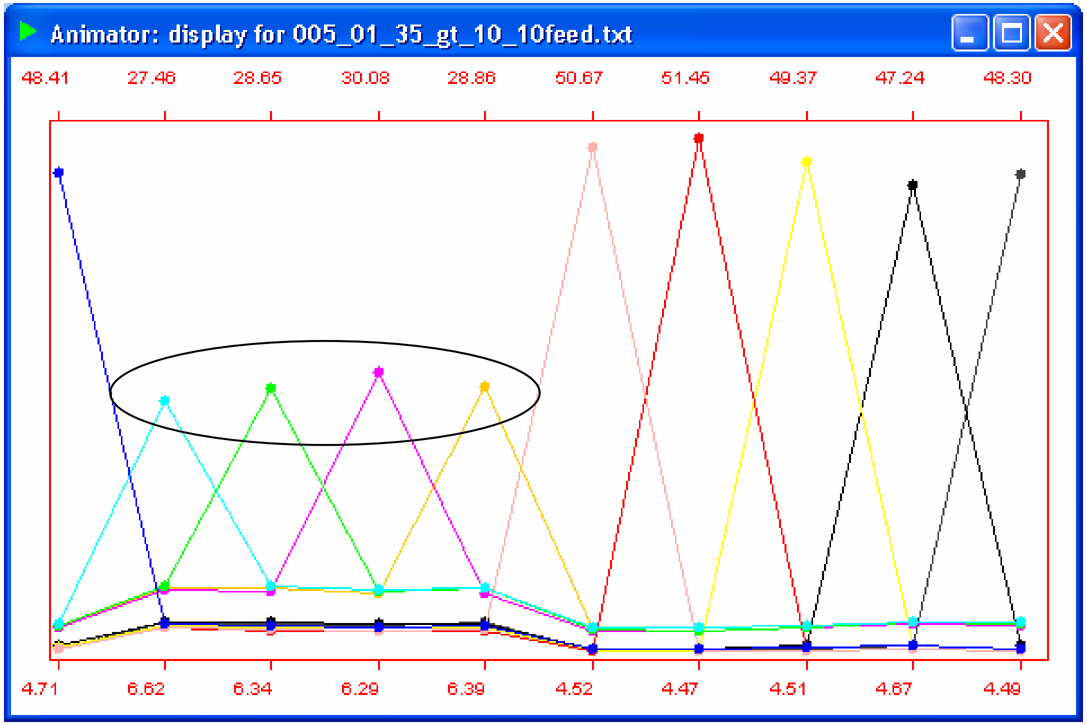


Animator: Iteration 80

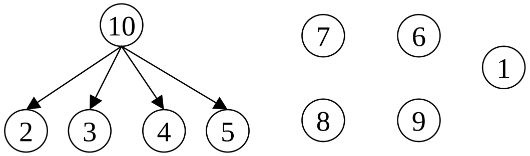


Note the changing scales = optional setting!

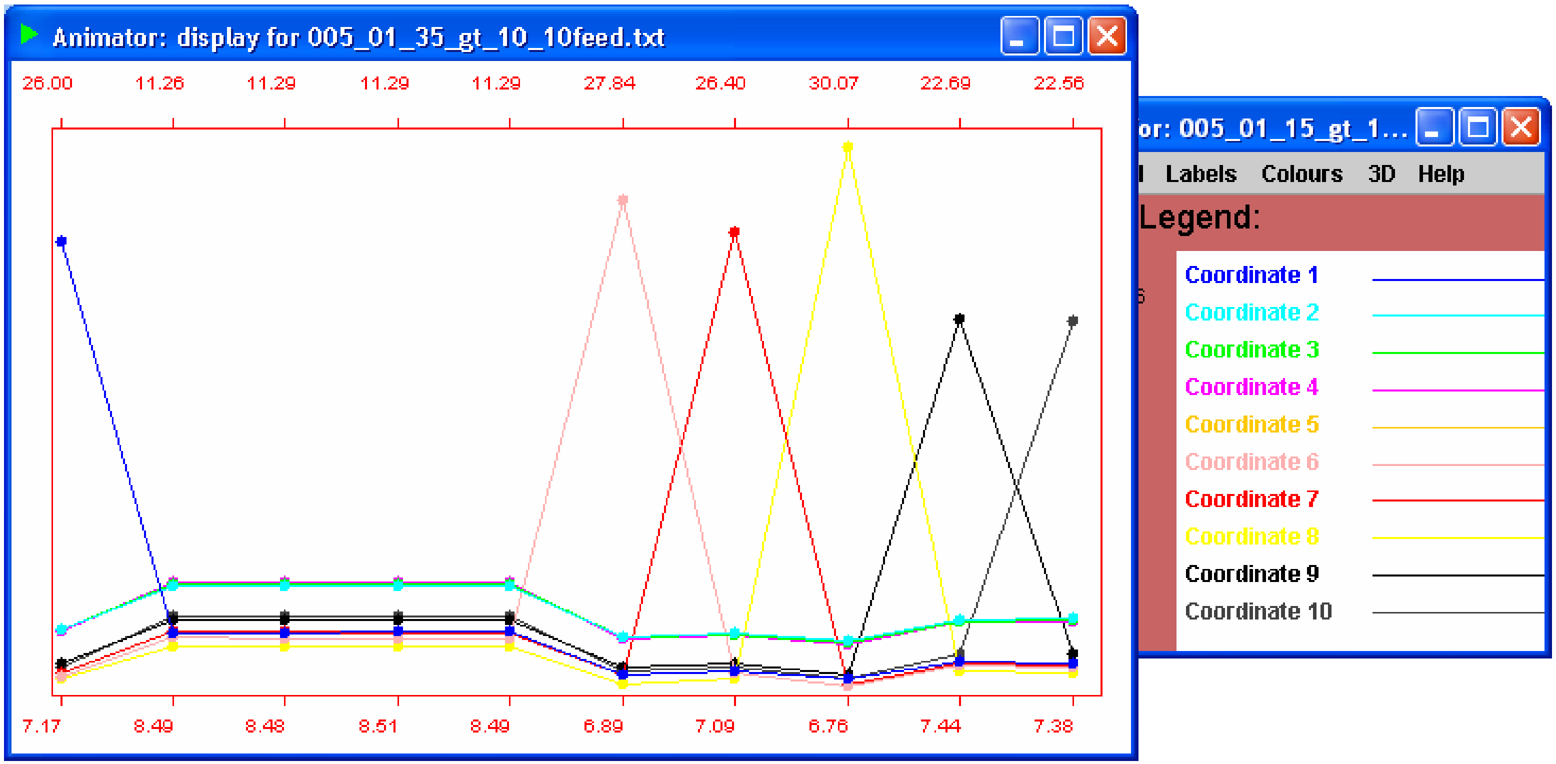
Animator: Iteration 350



See a group appearing …



Animator: Iteration 500

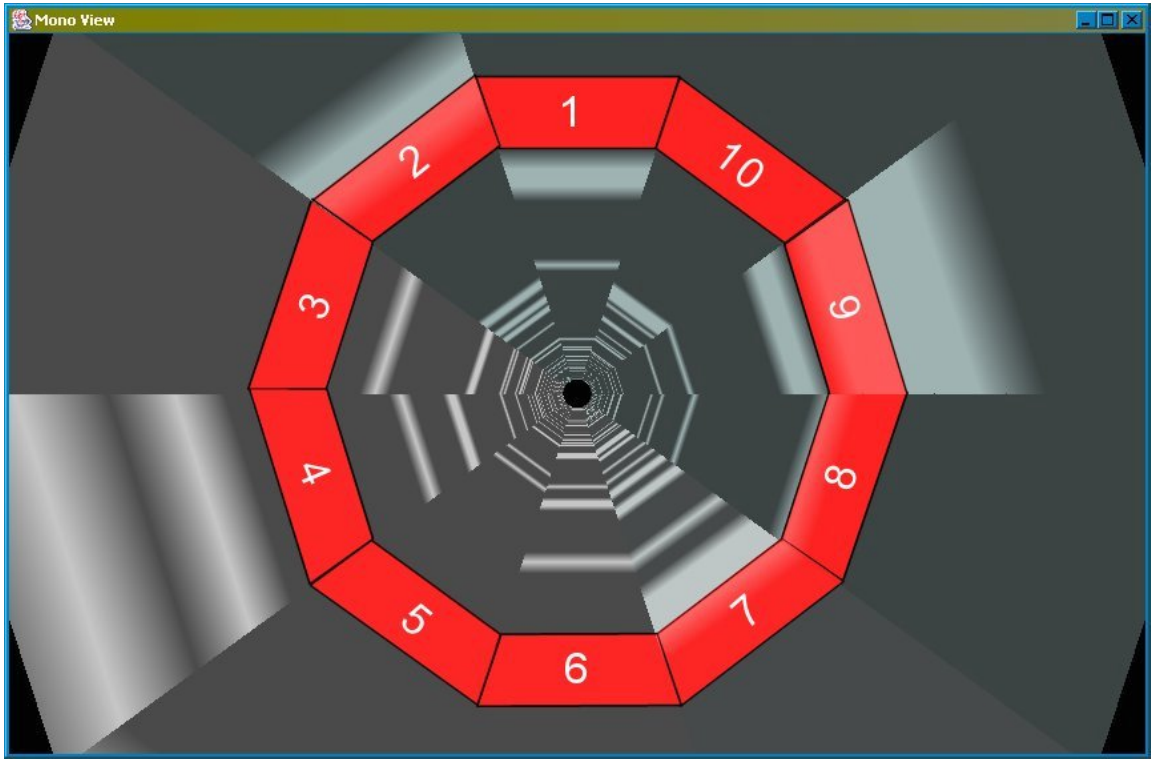


- Neurons 2,3,4 and 5 = definitely a sub-group
- Reasonably useful visualization for verification but not good enough to identify assembly structures

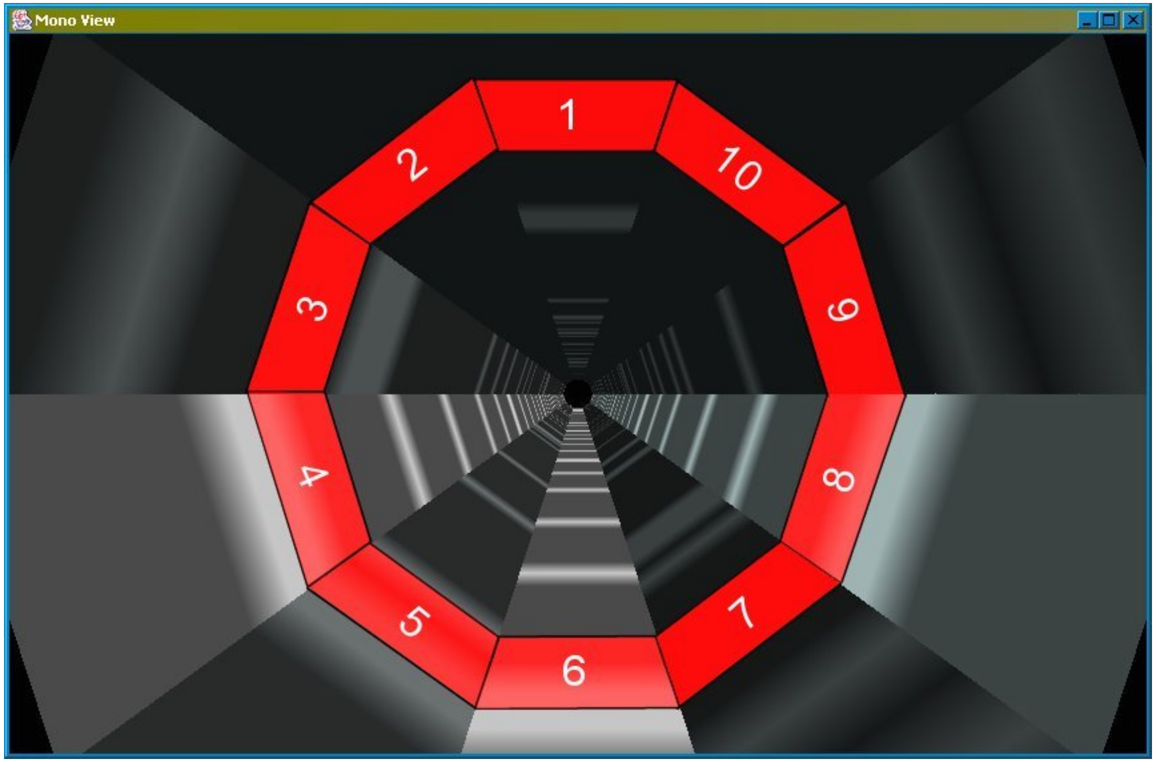
Solution 2: The Spike Train Tunnel

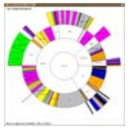
- **Directly** represents spike times
- **Comparison** of a number of trains
- Enables **direct user** manipulation
- **Virtual Environment**
- Users can:
  - **Filter** and **Dim** trains
  - **Reorder** by coincident spikes
  - **Sort**, using **cluster** analysis
  - **Fly** through the tunnel

Tunnel of 10 spike trains

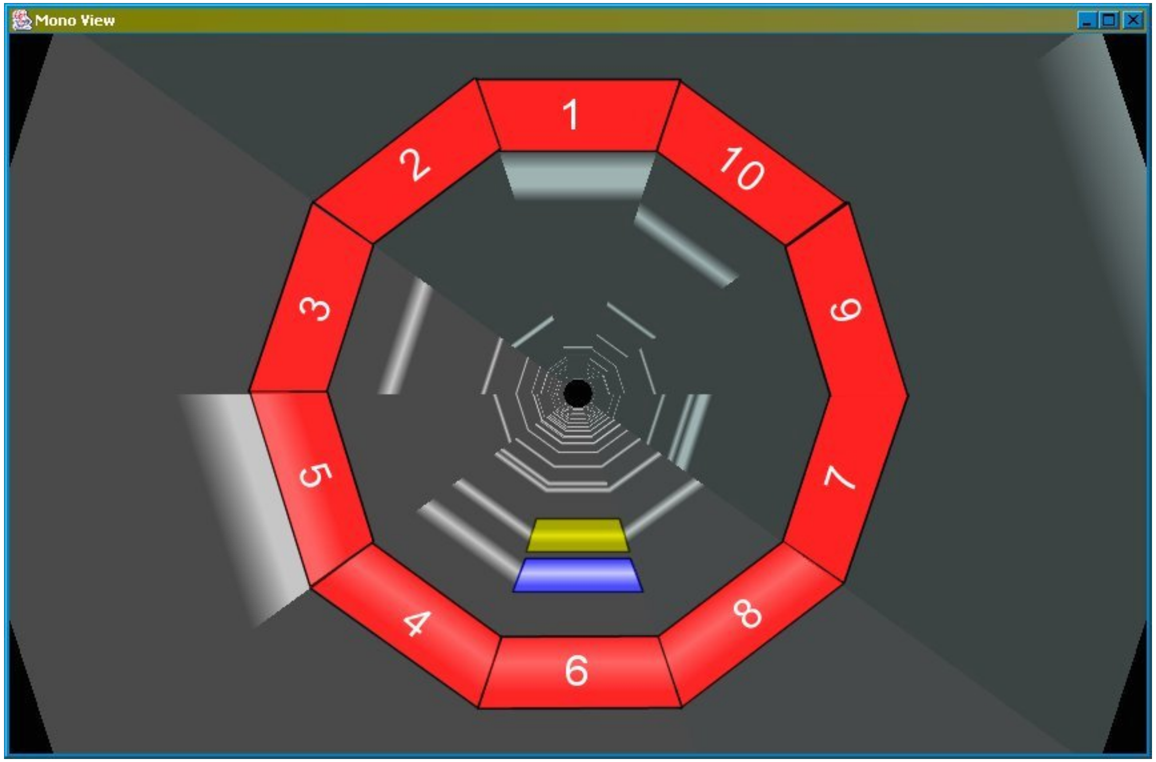


Spike Train Filtering





Progressive coincidence sorting



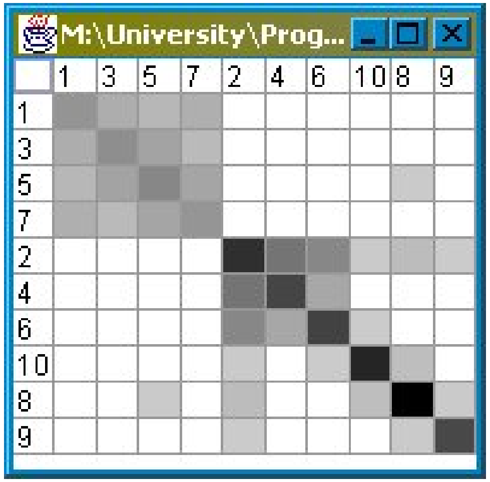
- Useful or is it just eye candy?

Solution 3: The Correlation Grid

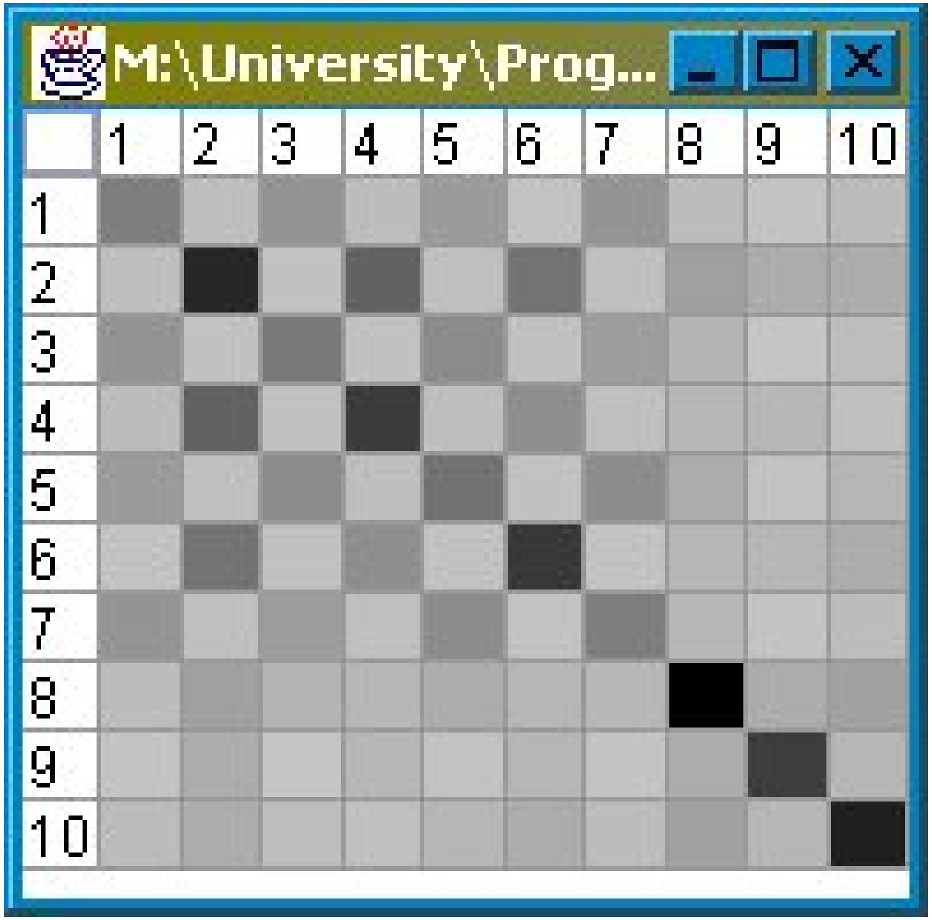
- Back to BASICS

- Presents an overview of **all cross-correlations** for a selected dataset in one **compact 2D representation**
  - Uses **glyph** representation
  - **Grey-scale** encoding

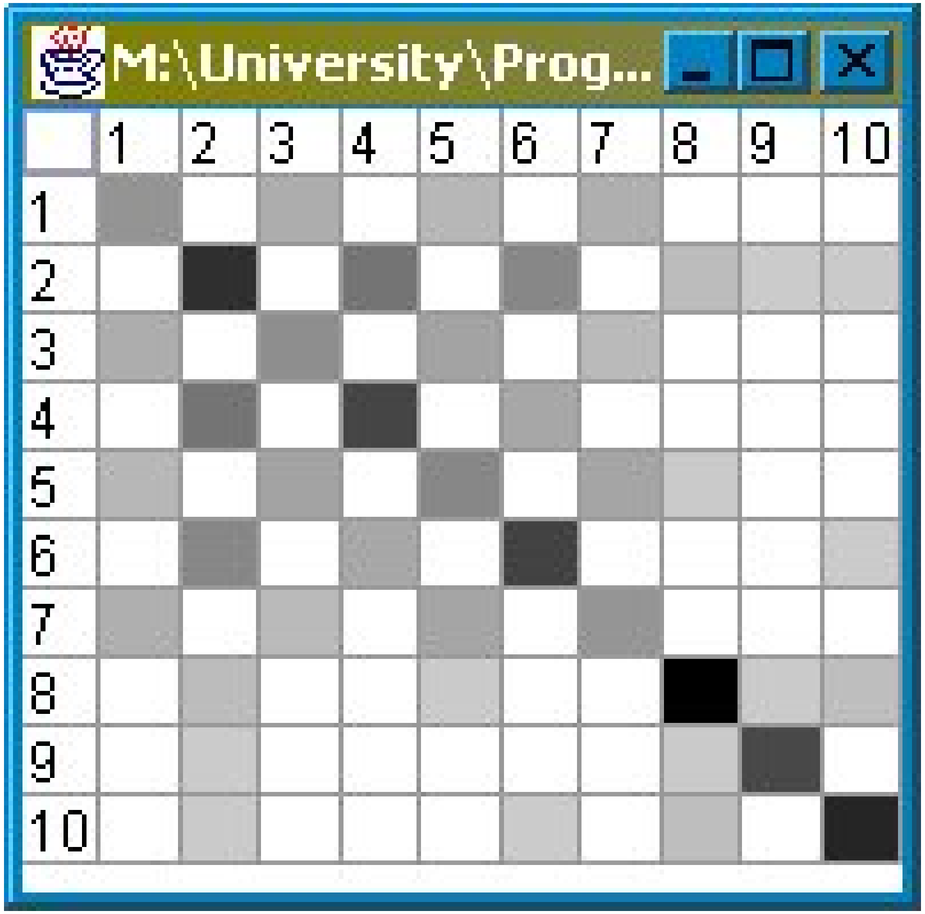
- Facilitates **direct user interaction**



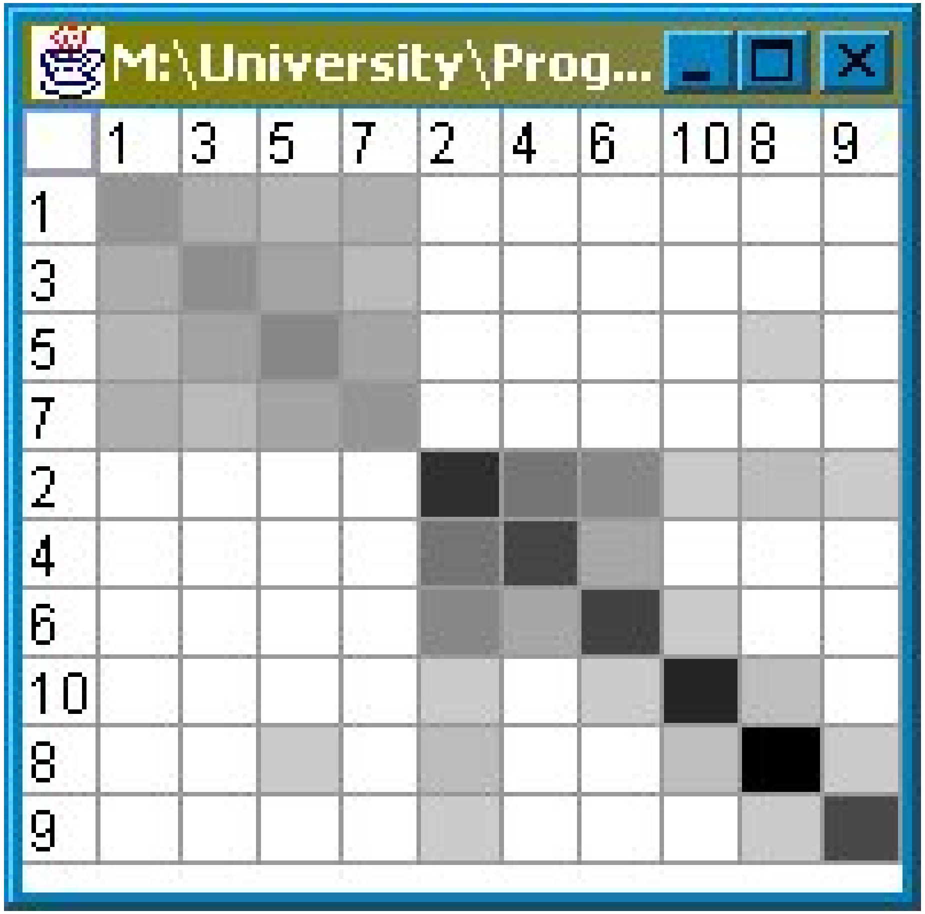
Raw data in Grid



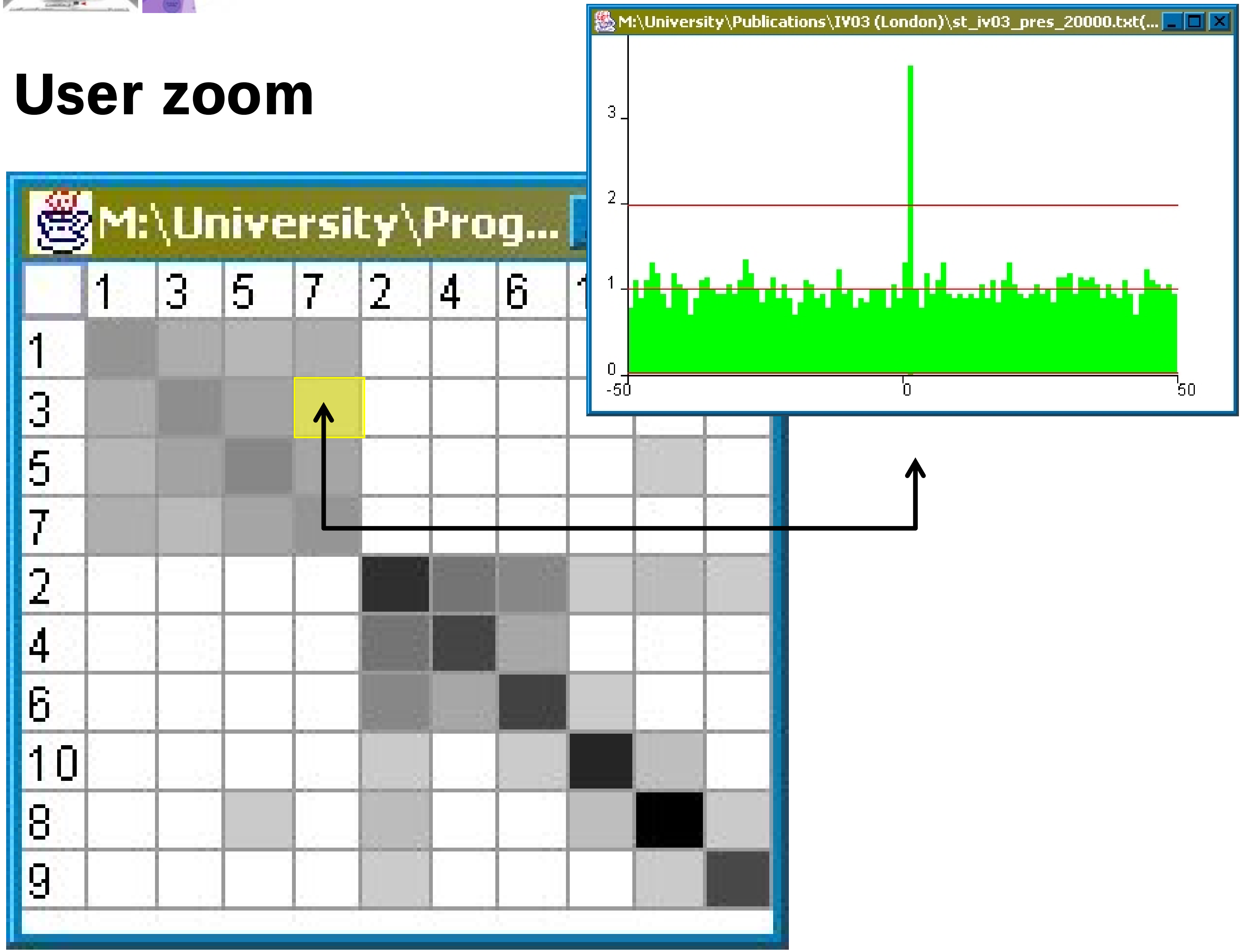
Filter – normalise data using Brillinger



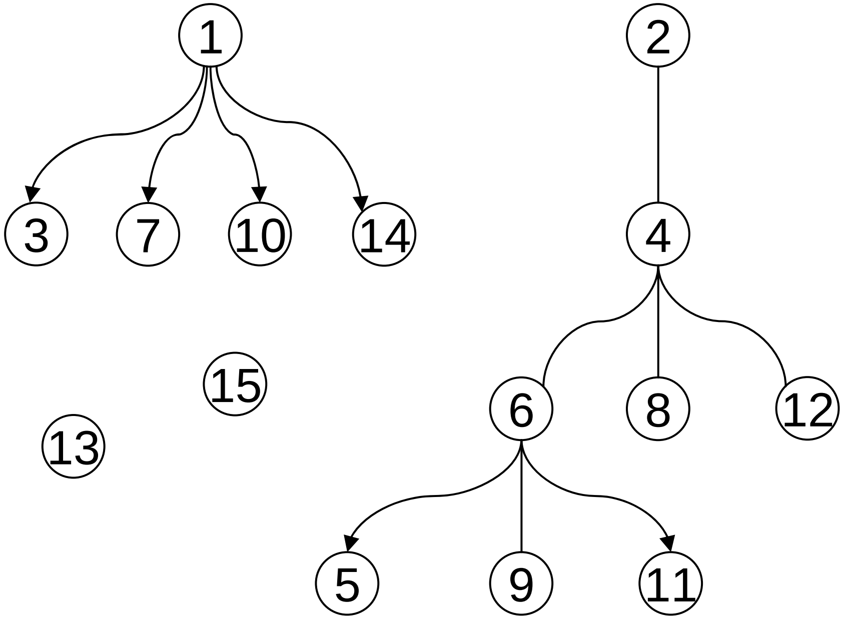
Sort using Cluster Analysis



User zoom

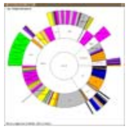

M:\University\Prog...
1 3 5 7 2 4 6
1
3
5
7
2
4
6
10
8
9
M:\University\Publications\IV03 (London)\st_iv03_pres_20000.txt(...
3
2
1
0
-50
0
50

Lets test it – A “blind” example

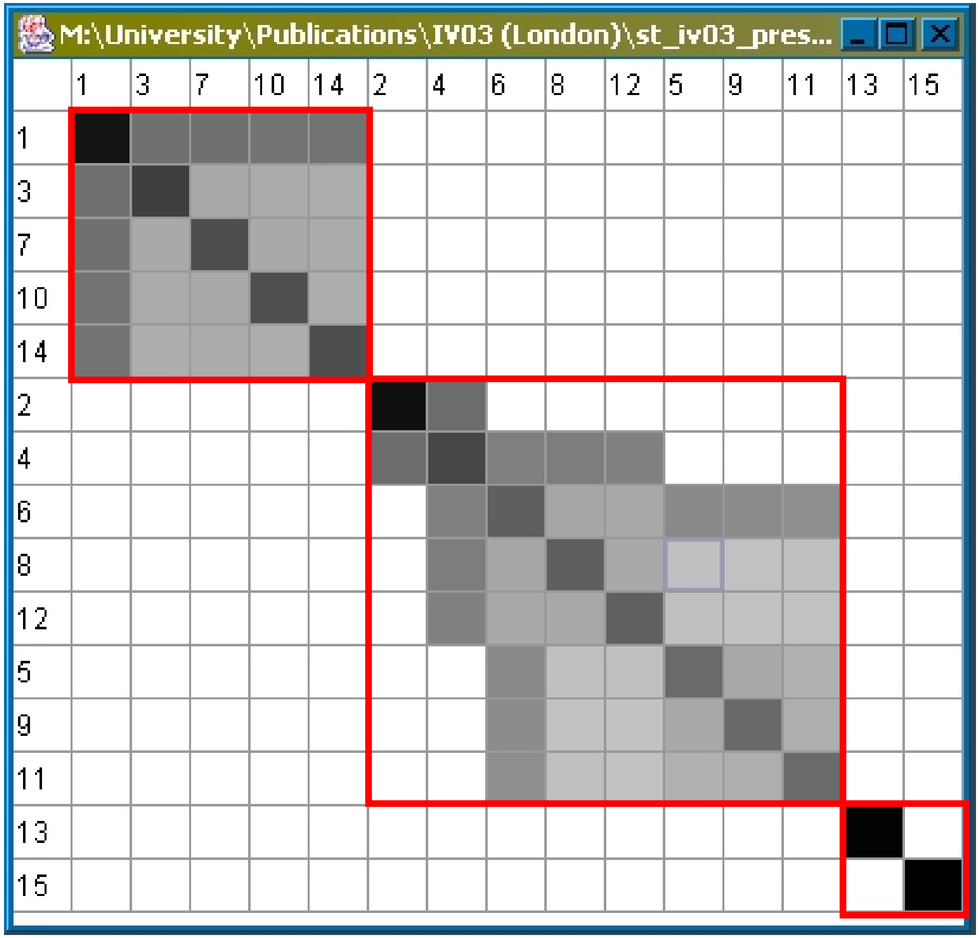


- This topology was **not known** to the analysts beforehand.
- The input to the system is **solely** the multi-dimensional spike train dataset.

Generated for 20000ms using an ‘Enhanced Integrate and Fire’ generator [Borisyuk 2001]

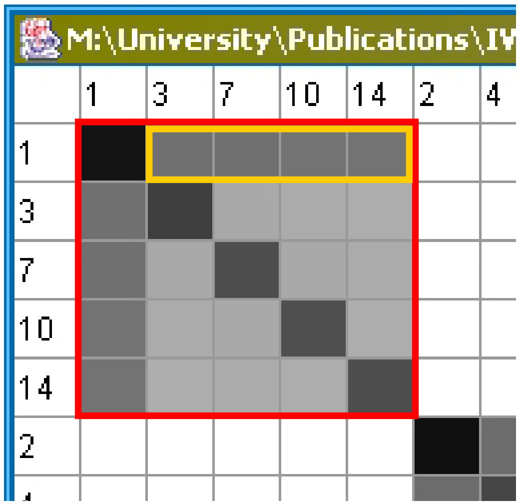


The Grid

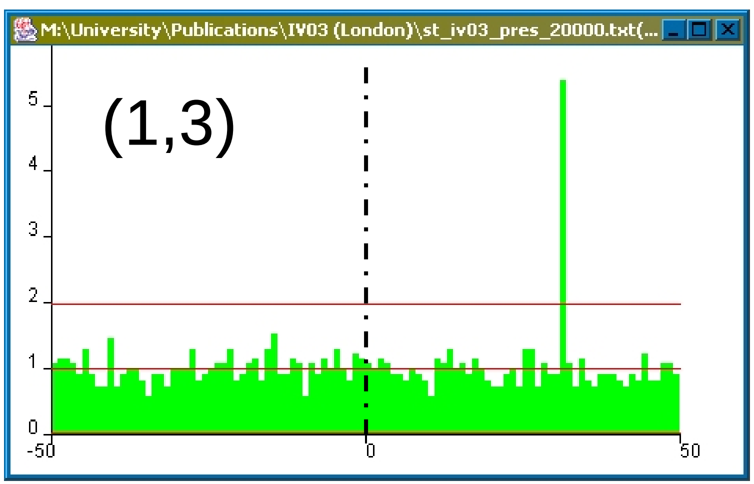


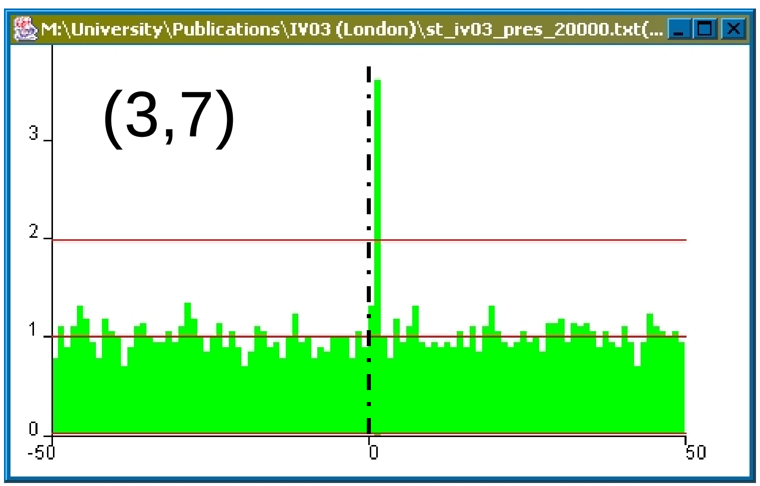
- Instantly, **three groups** are identified
- Ignore **auto** correlation
- Note, **symmetry**
- Lets focus on the **top group** first ...

Top Group

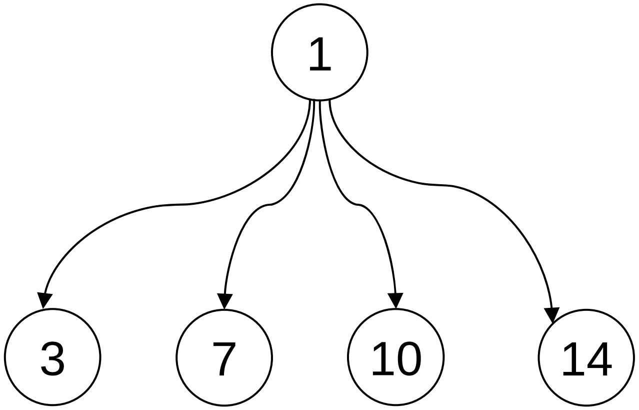


- Stronger correlation between spike train 1 and 3,7,10,14 (yellow highlight)
- Indicates correlation due to connection
  - see any Cross-Correlogram, say (1,3)
- Correlation between rest due to co-stimulation
  - see any Cross-Correlogram, say (3,7)





Top Group - assembly

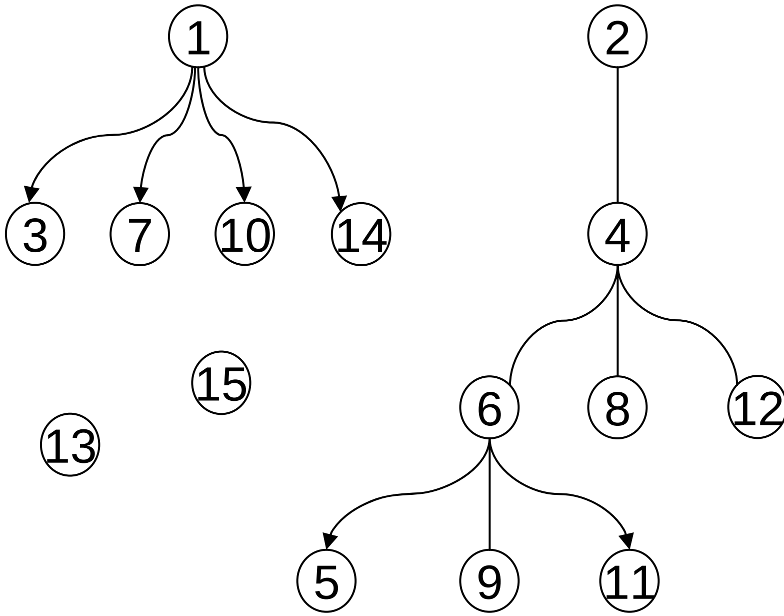


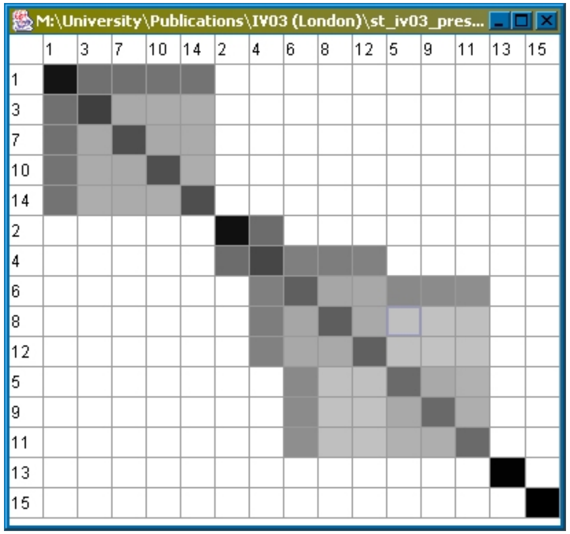
and so on with the rest of the assembly ...

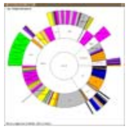
Finally

- After analysis of the middle and bottom groups, we **re-created the assembly (blindly!)**
  - Overview patterns
  - Details of correlation

- **Faster** than looking at all correlations





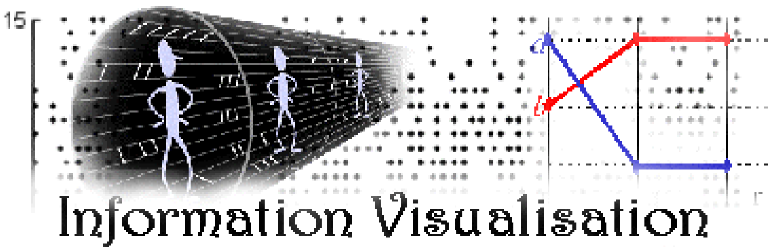


Conclusions

- In the Correlation Grid, **Visualization** provides a means of visually exploring the data after it has been processed **mathematically**.
- We need **both** the **visualization** and the **core mathematics** to solve the problems of large scale data analysis

Mathematics + IV = powerful analysis tools

Contact details



Dr. Liz Stuart
The Visualization Lab
University of Plymouth

lstuart@plymouth.ac.uk
www.plymouth.ac.uk/infovis