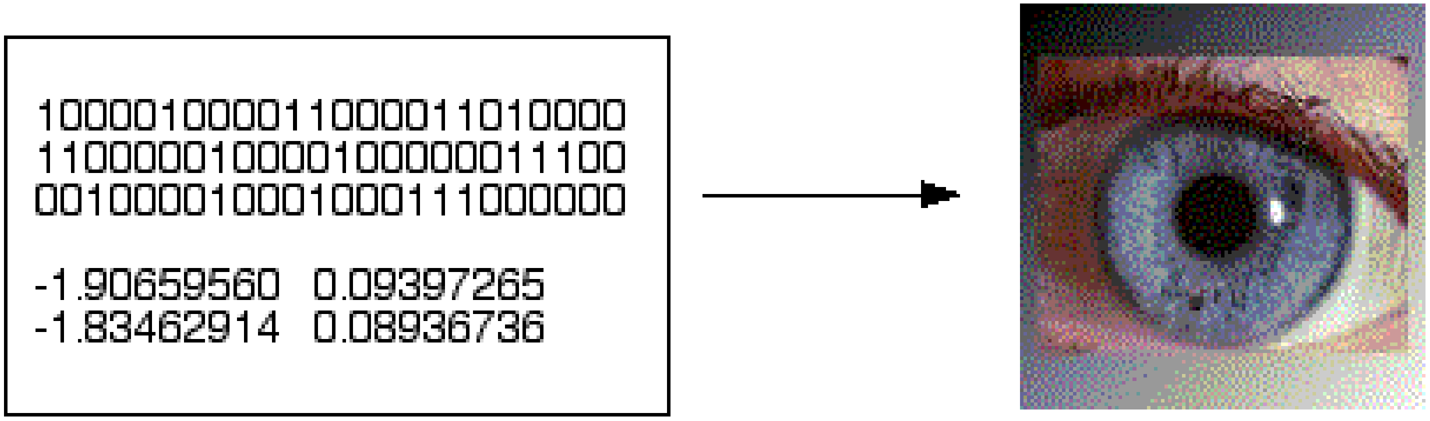
Znanstvena vizualizacija



```
10000100001100001101000
11000001000010000011100
00100001000100011100000

-1.90659560   0.09397265
-1.83462914   0.08936736
```

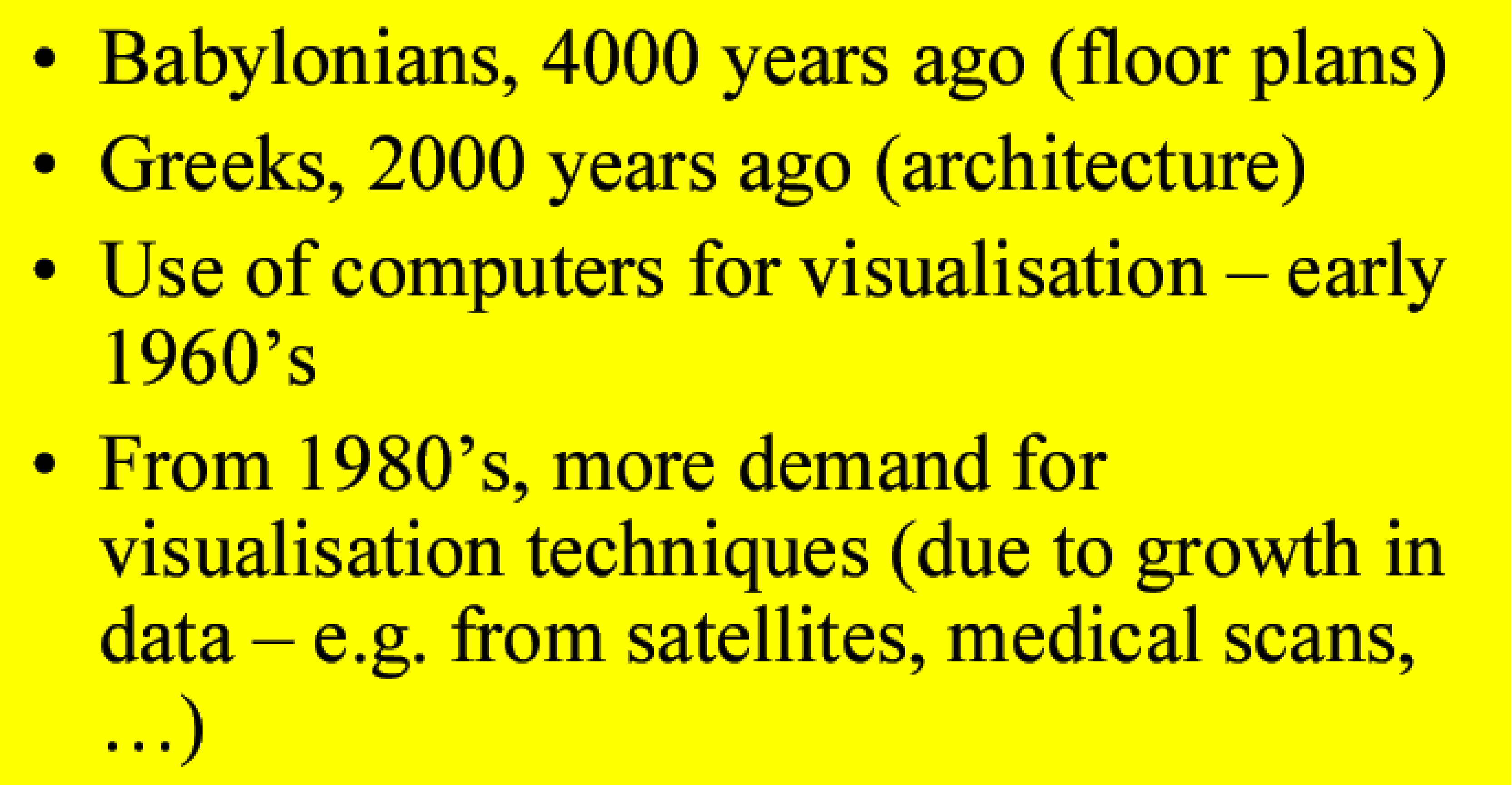
Visualizacija, računalniška grafika, znanost

Visualization brings together two disparate fields, the traditional sciences (like physics, chemistry, biology, for example) and computer graphics

Cilji vizualizacije

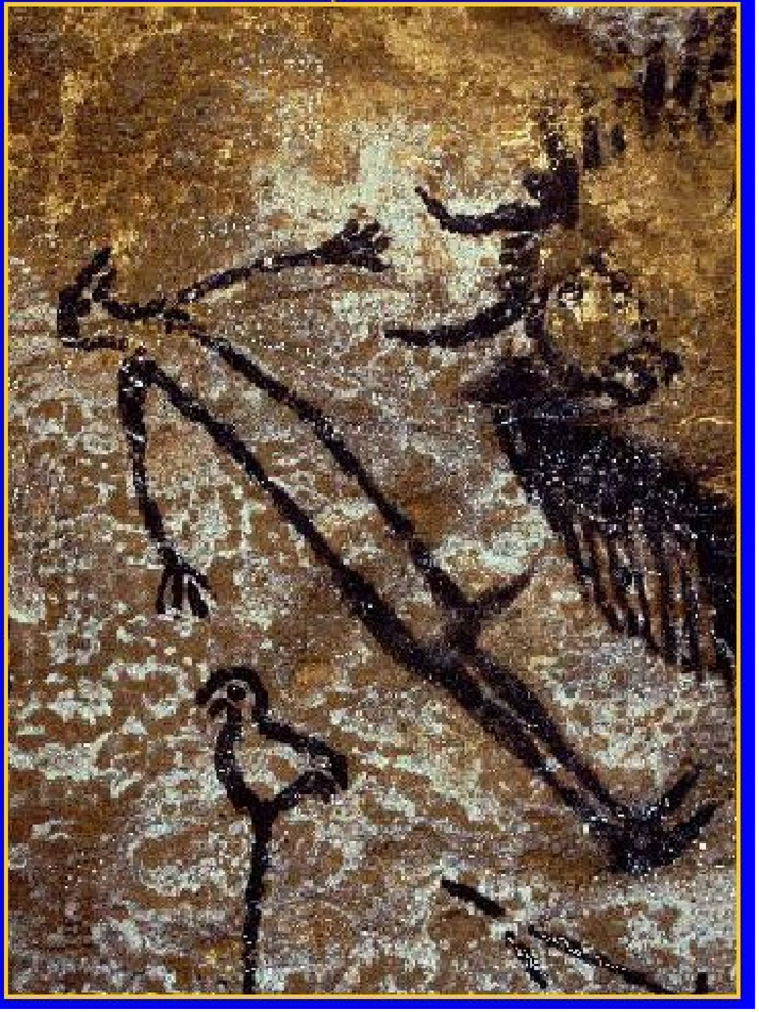
- Comparing: images, positions, data sets, subsets of data.
- Distinguishing: importance, objects, activities, range of value.
- Indicating directions: orientations, order, direction of flow.
- Locating: position relative to axis, object, map.
- Relating: concepts, e.g. value and direction, position and shape, temperature and velocity, object type and value.
- Representing values: numeric value of data.
- Revealing objects: exposing, highlighting, bringing to the front, making visible, enhancing visibility.

Malo zgodovine



- Babylonians, 4000 years ago (floor plans)
- Greeks, 2000 years ago (architecture)
- Use of computers for visualisation – early 1960’s
- From 1980’s, more demand for visualisation techniques (due to growth in data – e.g. from satellites, medical scans, …)

Lascaux (12,000 BC)



www.culture.fr/culture/arcnat/lascaux/en/

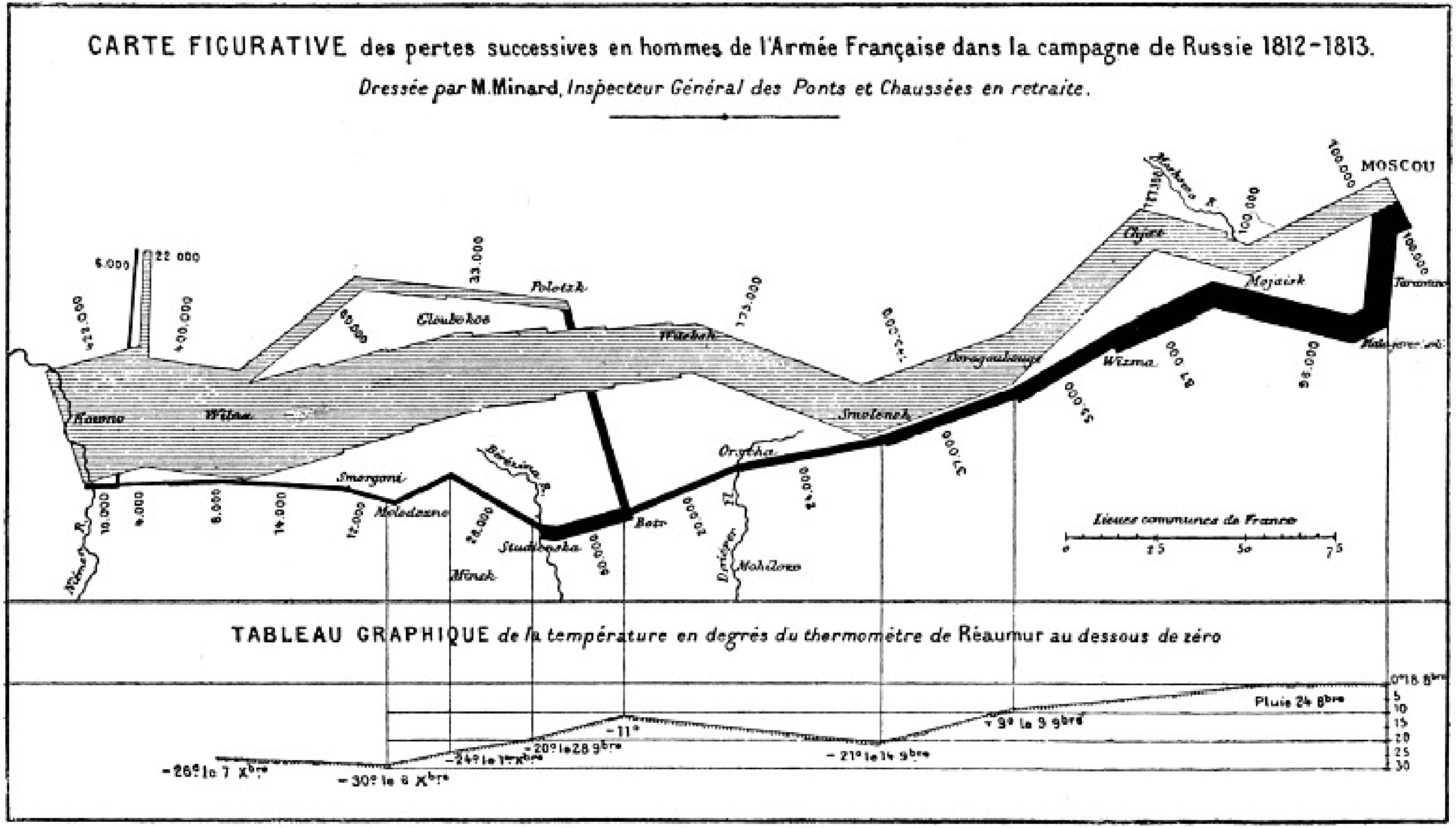
Dr John Snow

- Plotted locations of cholera deaths, London, September 1854
- Marked the eleven water pumps
- Observed correlation of deaths with proximity to Broad Street water pump
- Pump handle replaced, death rate decreases.

Napoleon's Campaign

- Depicts 1812 invasion of Russia
- 6 variables
  - Size of army
  - 2D location of army
  - Direction of movement
  - Temperature
  - Date
- Graphic also by Charles Minard

Napoleon's Russia Campaign



X^{bre} = December   9^{bre} = November   8^{bre} = October

Ena od definicij

- The broad class of computer graphic and imaging techniques used to create visual representations of data.
- Wolff and Yaeger, Visualization of Natural Phenomena, ELOS, Springer-Verlag, 1993

Kaj je znanstvena vizualizacija?

- the *mapping* of data and/or information to *images* to gain or present *understanding* and *insight*
- the *visual analysis* of scientific data. It transforms the numeric representation into a geometric (image) representation, enabling a scientist, engineer, or mathematician to observe his simulated or measured data
- Scientific Visualization encompasses and unifies the fields of computer graphics, image processing, computer vision, computer-aided design, signal processing, and human-computer interaction
- Scientific Visualization is a set of software tools coupled with a powerful 3D graphical computing environment that allows any geometric object or concept to be visualized by anyone. The software provides an easy to use interface for the user. The hardware must be able to manipulate complex, geometrically-described, 3D environments in motion, color, and with any level of “realism” called for to better communicate the essence of the computation (in foreword of An Introductory Guide to Scientific Visualization. R. A. Earnshaw & N. Wiseman, Springer-Verlag, 1992)
- SciVi, SciViz, SciVis, SV, ...

Zakaj znanstvena vizualizacija?

- “The purpose of computing is insight, not numbers”

  Richard Hamming (1962)

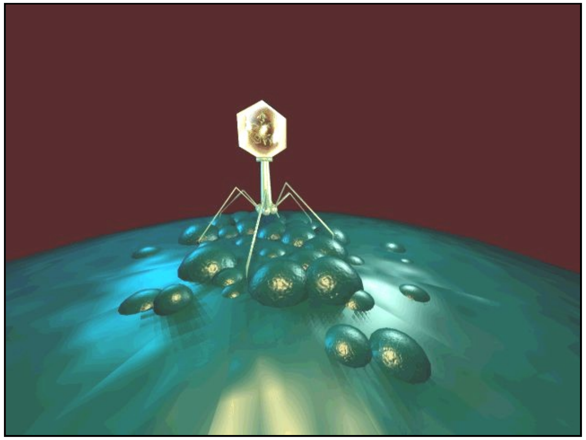
- “*Maintain a Healthy Skepticism:* Never believe the results of your calculations. Always look for errors, limitations, unnatural constraints, inaccurate inputs, and places where algorithms break down. Nothing works until it is tested, and it probably does not really work even then. This skepticism should help avoid embarrassing situations, such as occur when a neat physical explanation is painstakingly developed for a computational blunder.”

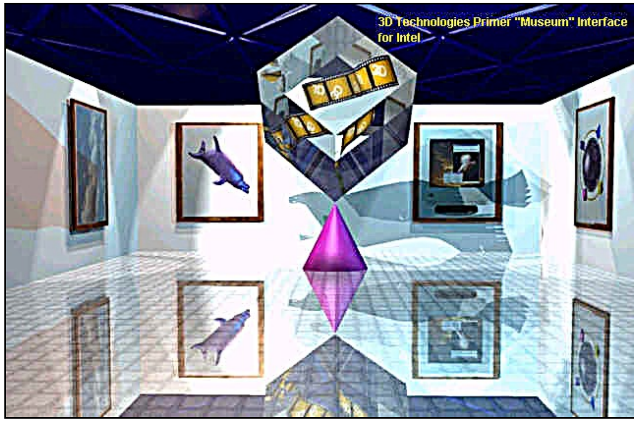
  E. S. Oran & J. P. Boris (1987)

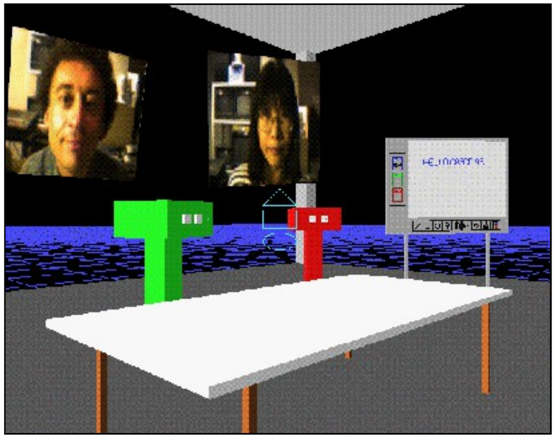
- “In decreasing order of importance, scientific visualization:
  - shows you where you are screwing up
  - allows discoveries
  - is useful for show-and-tell”

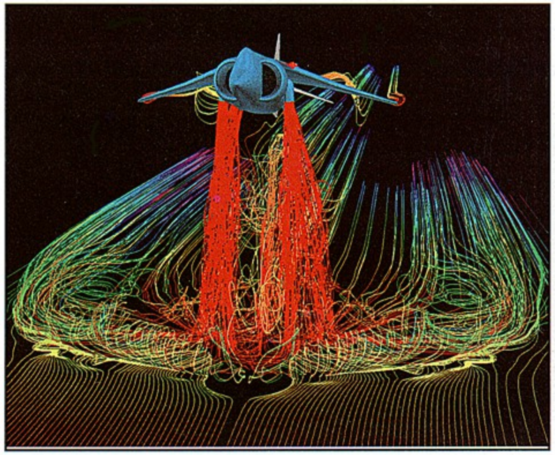
  J. J. O’Brien (1993)

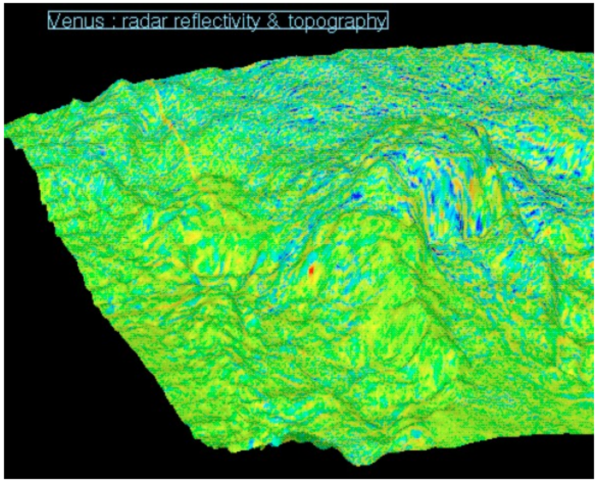
Aplikacije



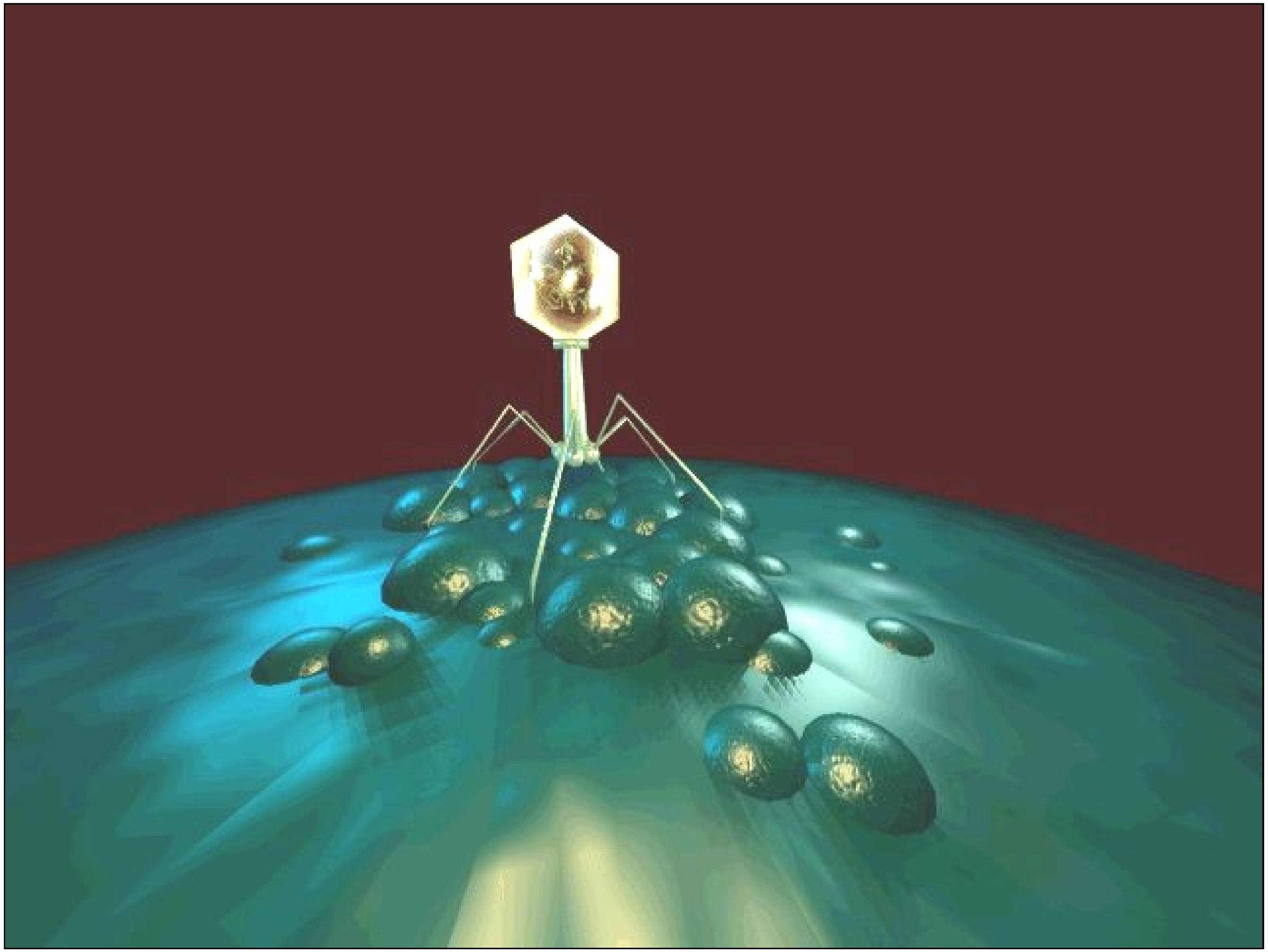

3D Technologies Primer "Museum" Interface for Intel





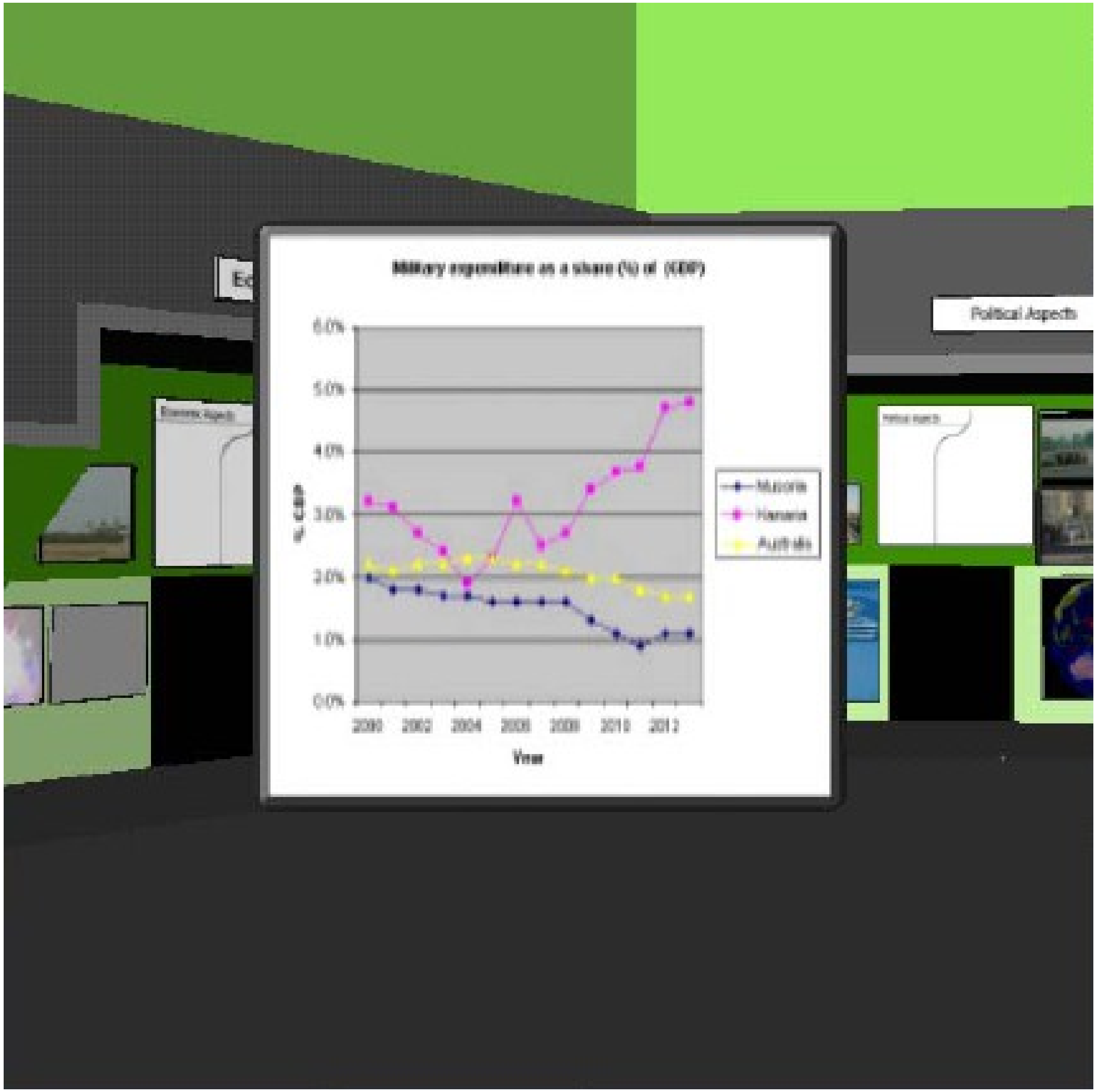

Venus : radar reflectivity & topography

Visualisation for Education

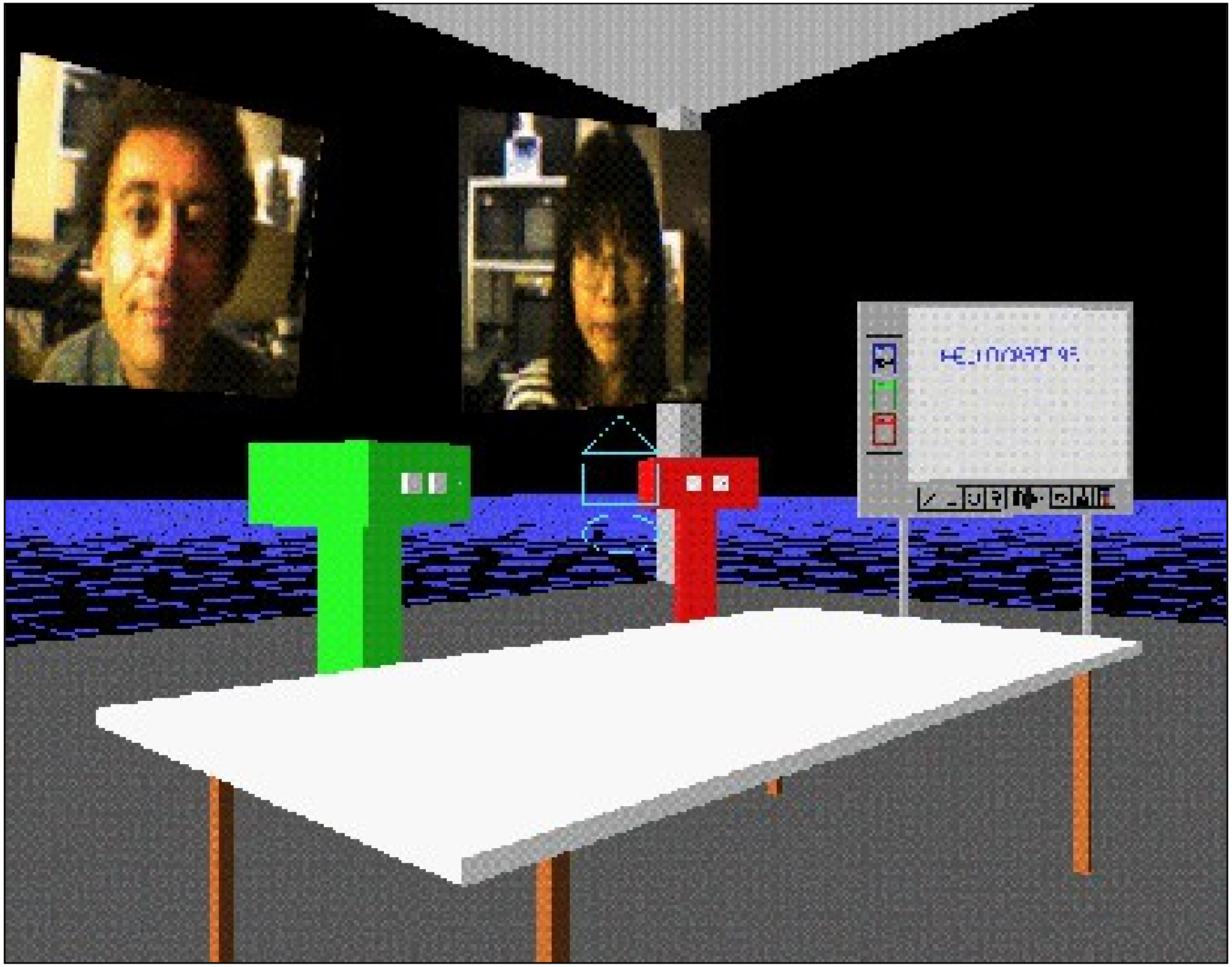


Visualisation for Architecture

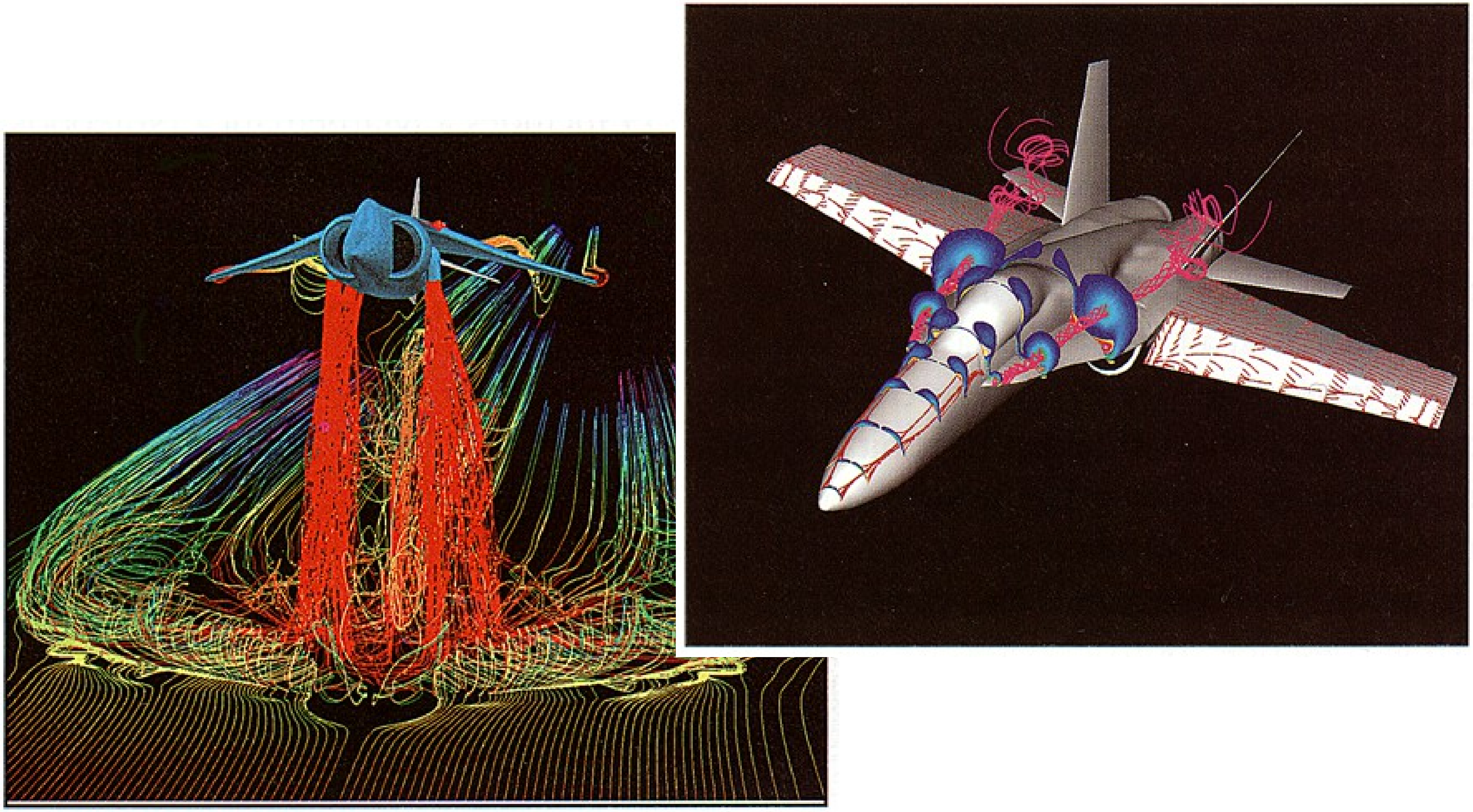
Visualisation in Interface Design



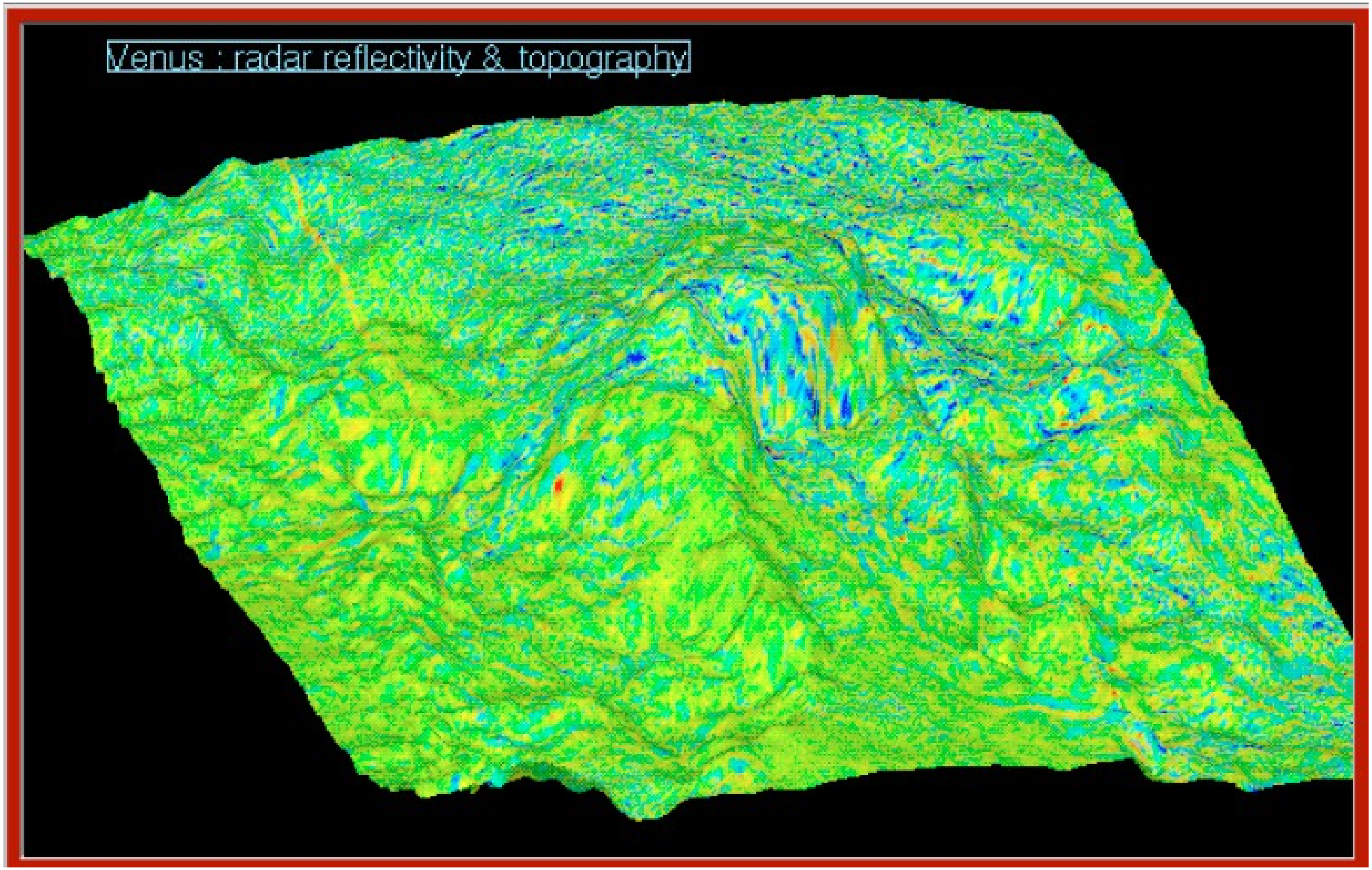
Visualising Group Work



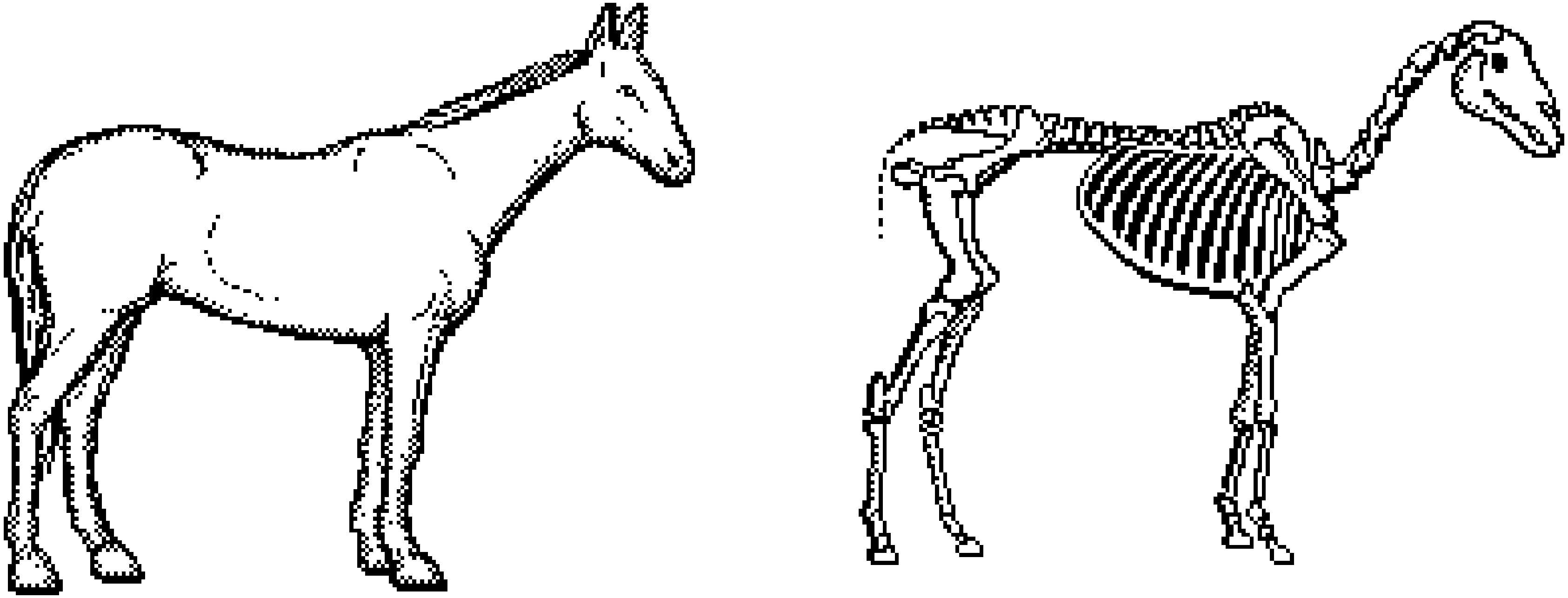
Visualising Complexity



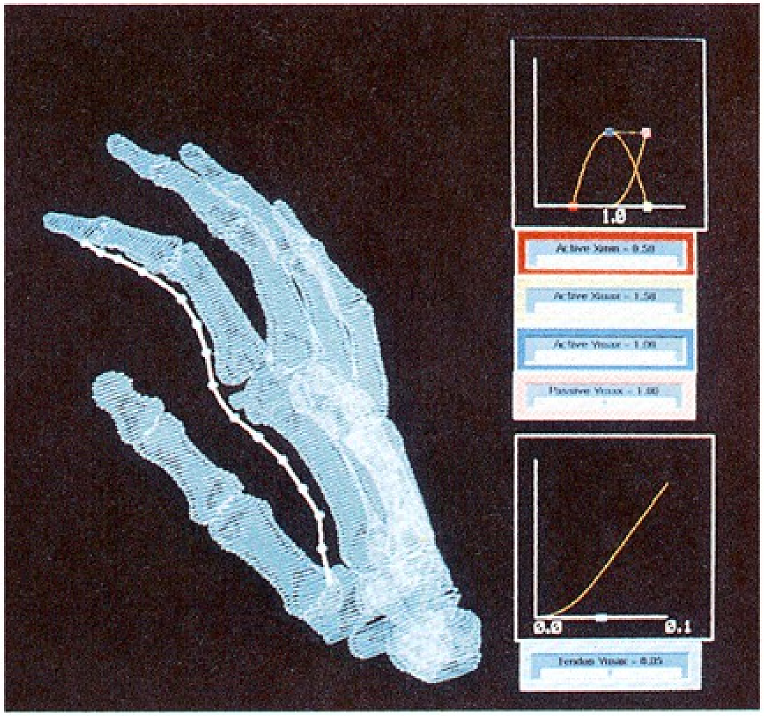
Scientific Visualization in the Geosciences

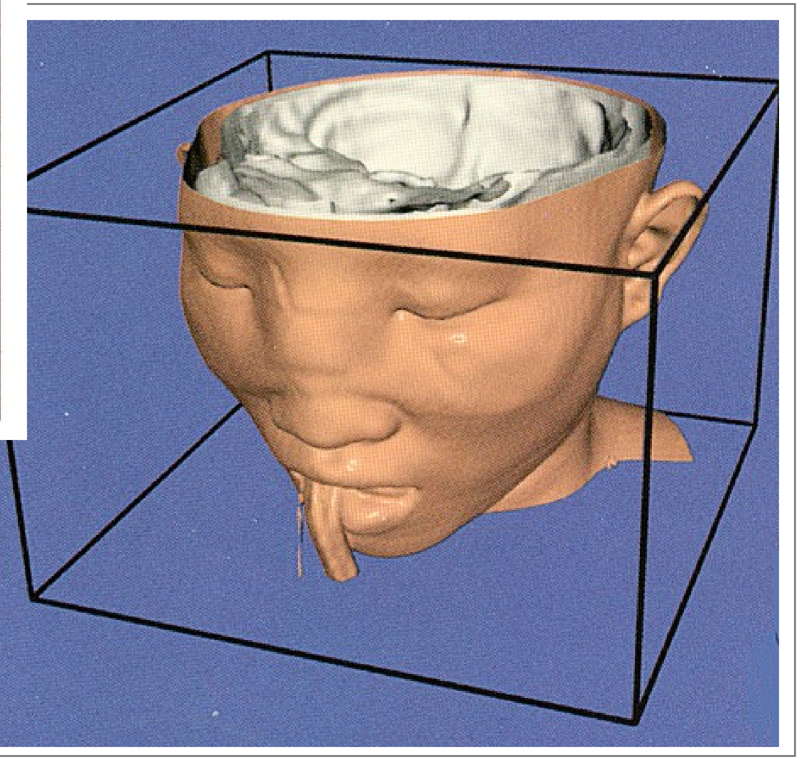


Visualization for Biological and Physical Scientists



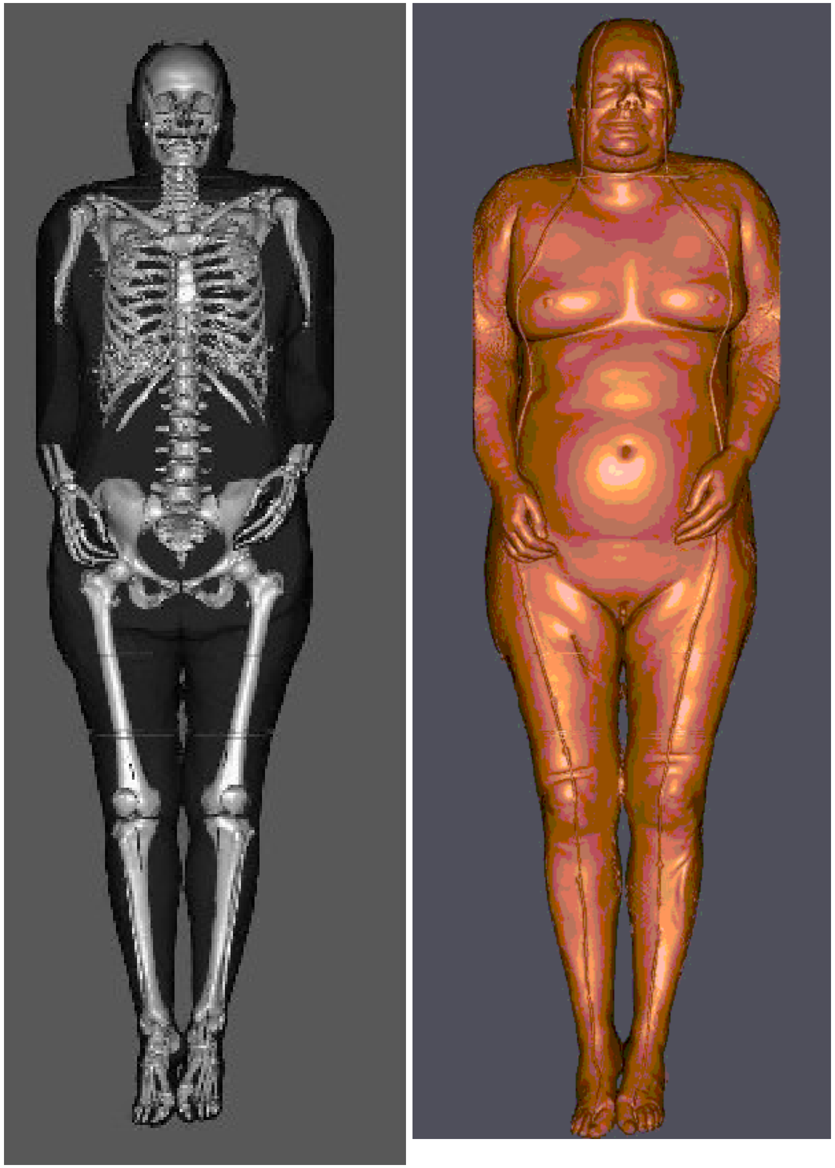
Medical/Bioinformatics



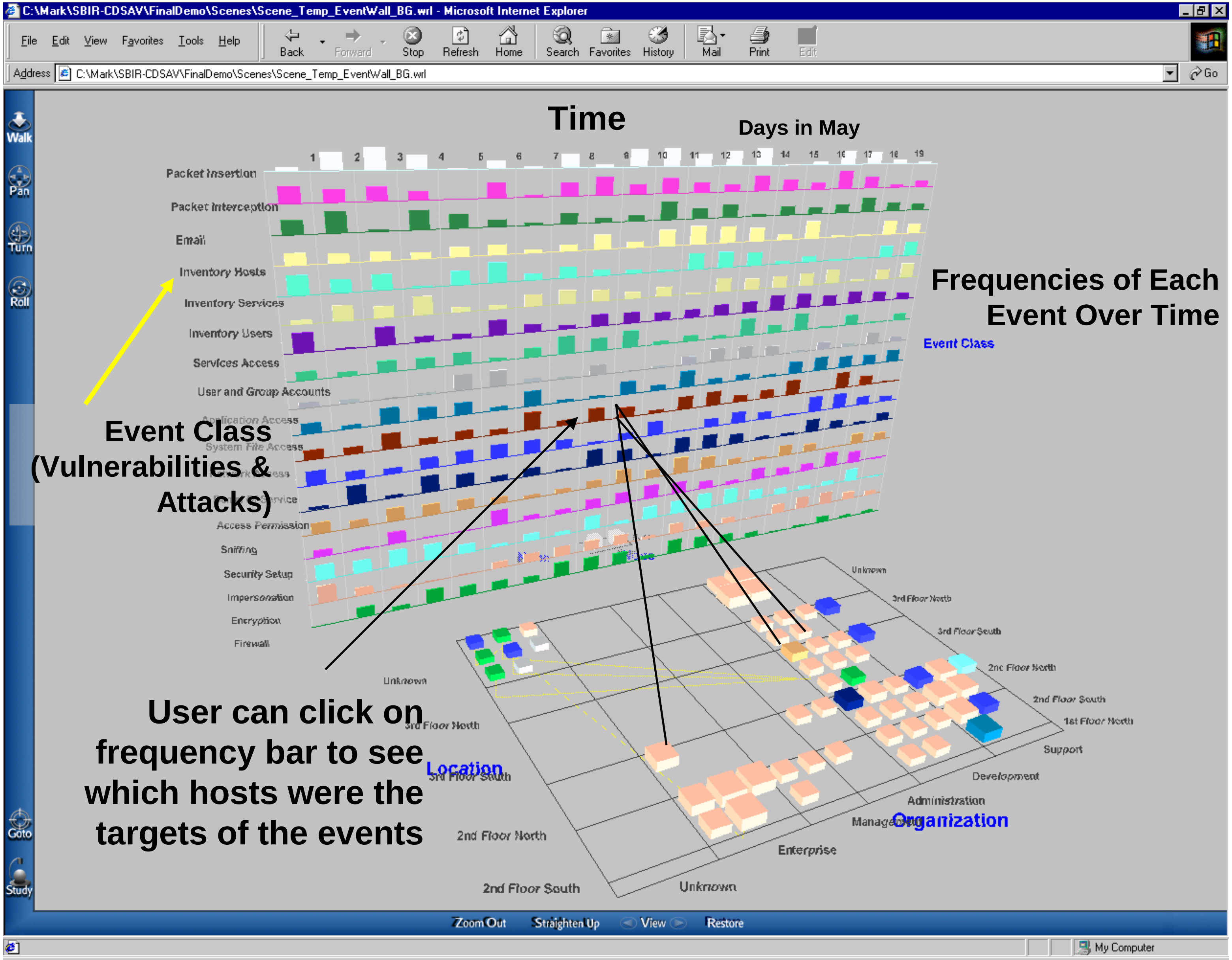


NLM Visible Human Project

- An outgrowth of the NLM's 1986 Long-Range Plan.
- Creating a complete, anatomically detailed, three-dimensional representations of the male and female human body.
- The current phase of the project is collecting transverse CT, MRI and cryosection images of representative male and female cadavers at one millimeter intervals.



Visualising Non-Physical Systems



Time

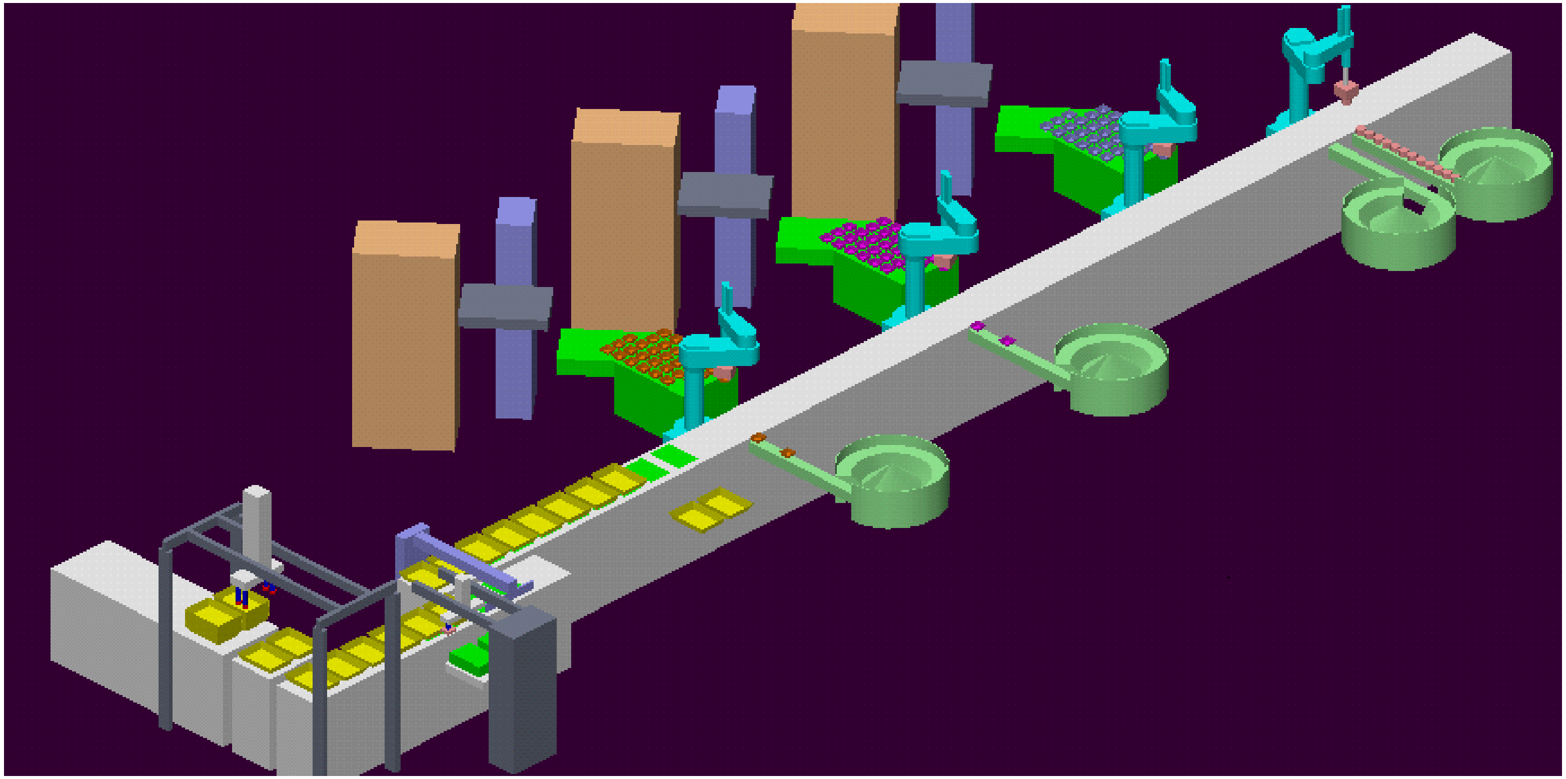
Days in May

Frequencies of Each Event Over Time

Event Class (Vulnerabilities & Attacks)

User can click on frequency bar to see which hosts were the targets of the events

Visualising Non-existing Systems



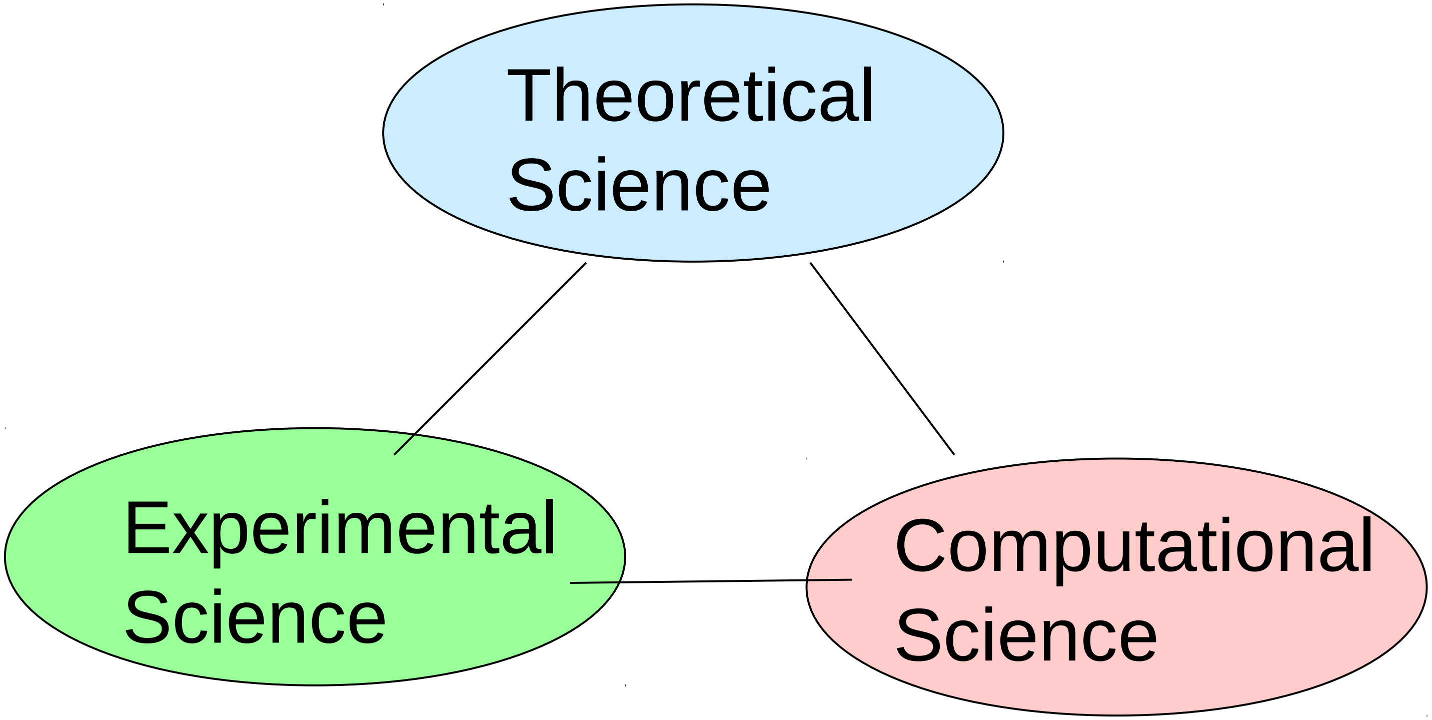
Eg Visual Interactive Simulation

Systems Modelling

Visualization in Computational Science

- Computational Science refers to the knowledge and techniques required to perform computer simulations and tackling computationally intensive problems.
- *Main goal* of Computational Science:
  - To understand the workings of nature
- *Steps*
  - Observations, Physical Model, Mathematical Model, Numerical Model, Simulation and Analysis

The Research Triangle

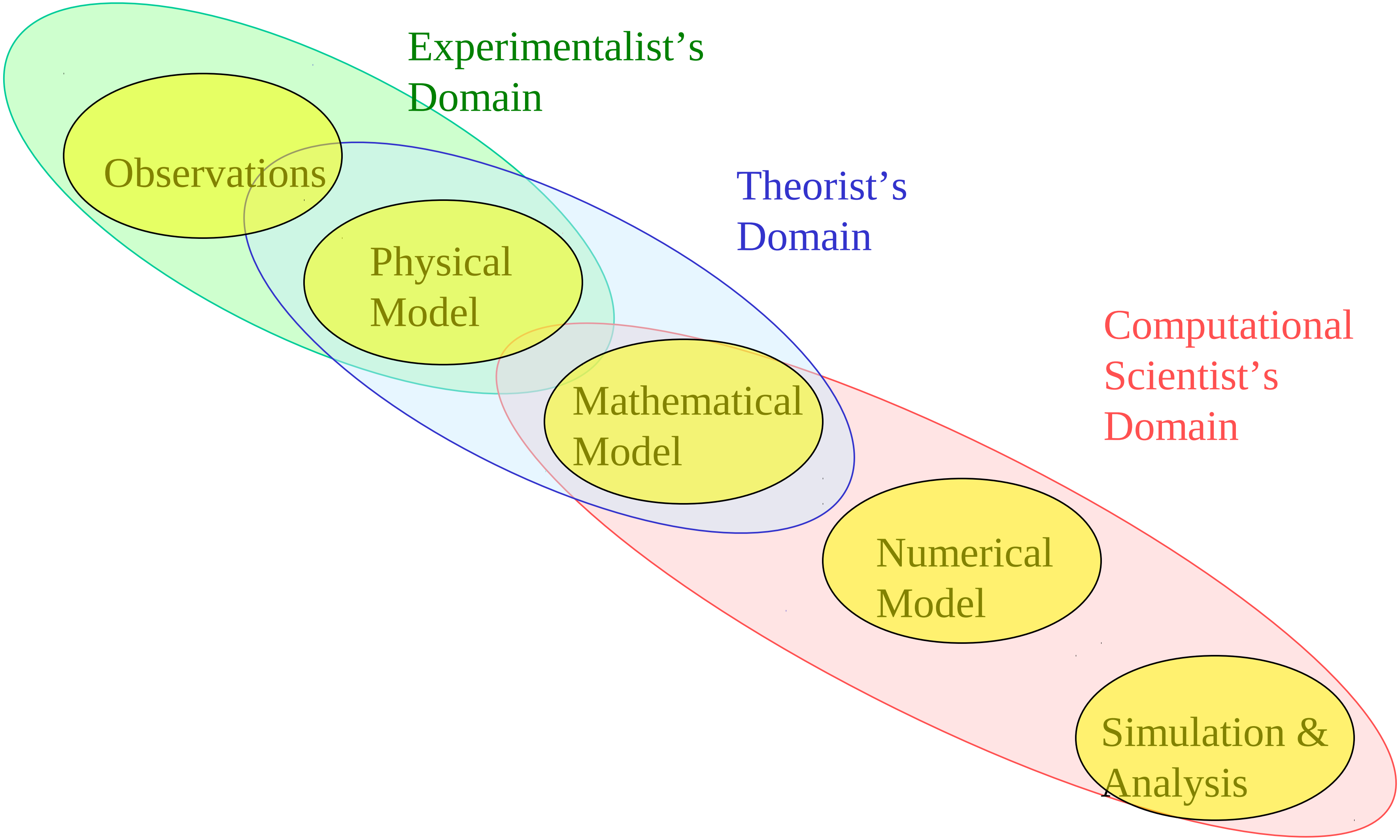


Theoretical Science

Experimental Science

Computational Science

Domene raziskovanja



Experimentalist's Domain

Theorist's Domain

Computational Scientist's Domain

Observations

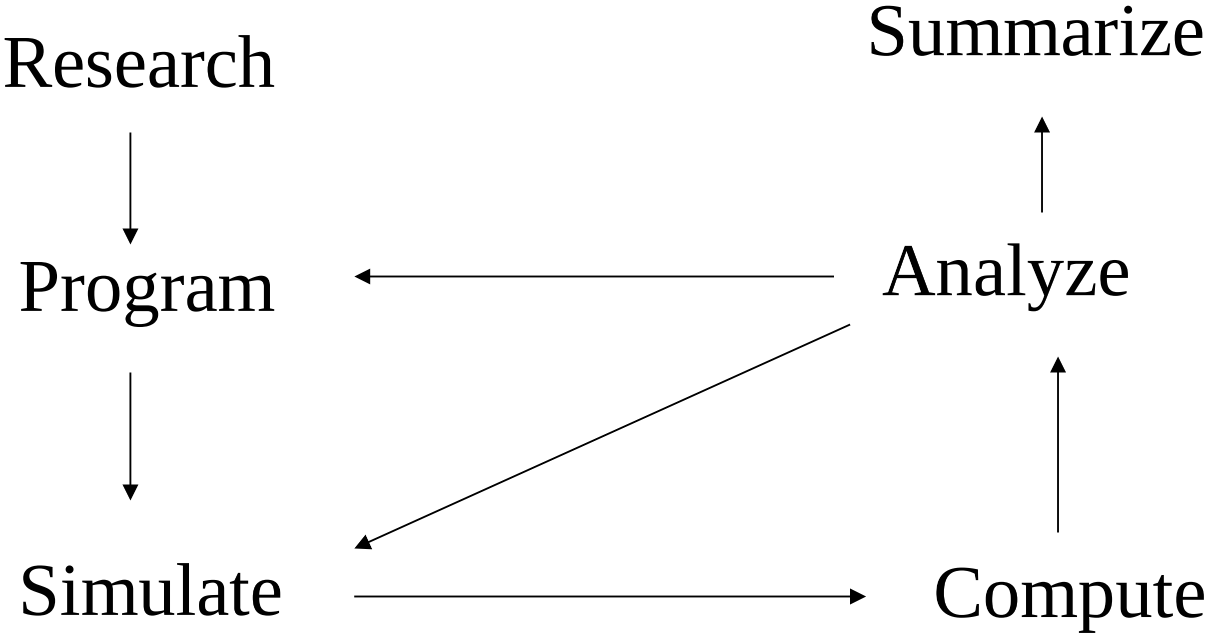
Physical Model

Mathematical Model

Numerical Model

Simulation & Analysis

Computational Cycle



Research

Program

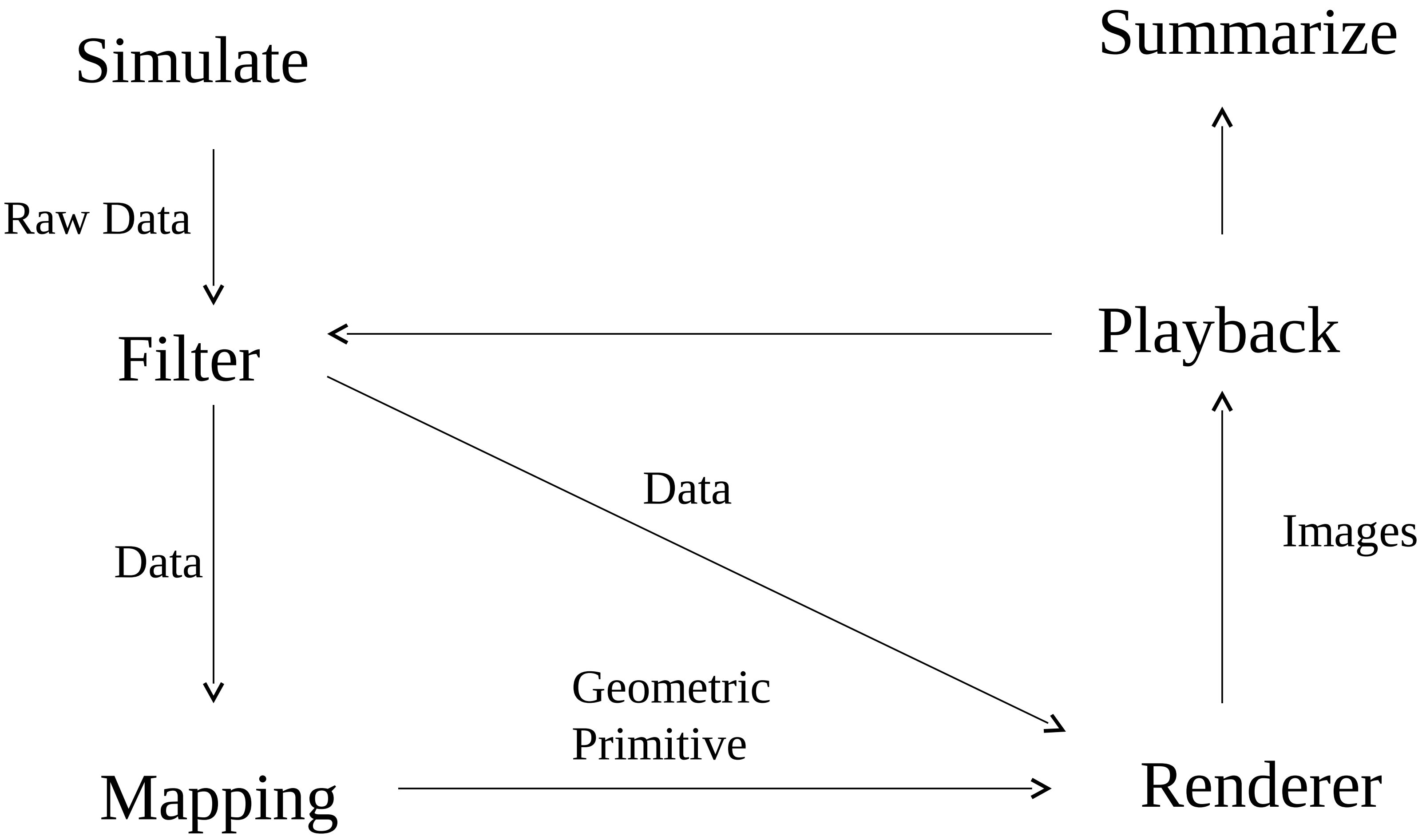
Simulate

Compute

Analyze

Summarize

Analysis Cycle



Simulate

Raw Data

Filter

Data

Mapping

Geometric
Primitive

Data

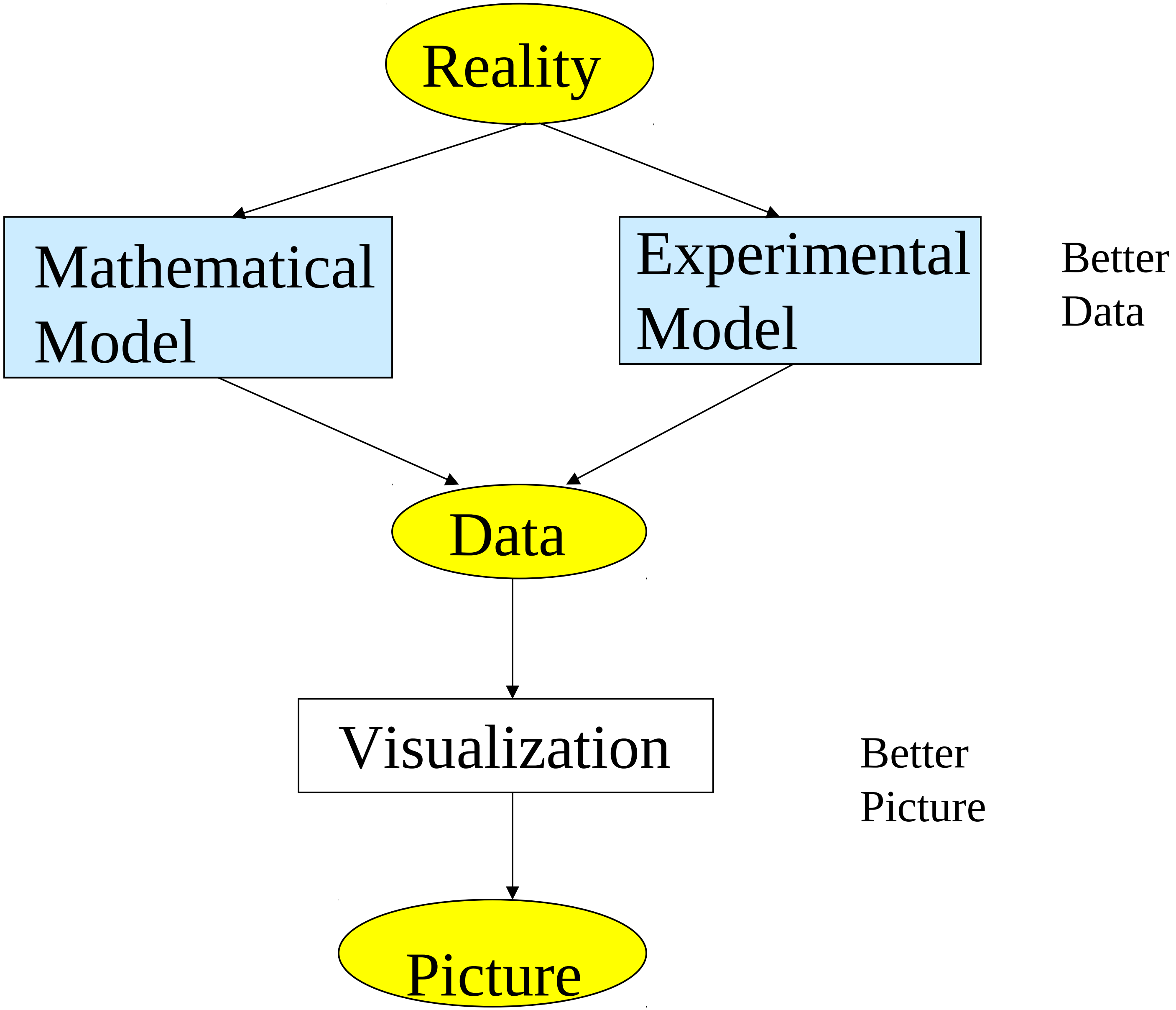
Renderer

Images

Playback

Summarize

Od resničnosti do slike



Reality

Mathematical Model

Experimental Model

Better Data

Data

Visualization

Better Picture

Picture

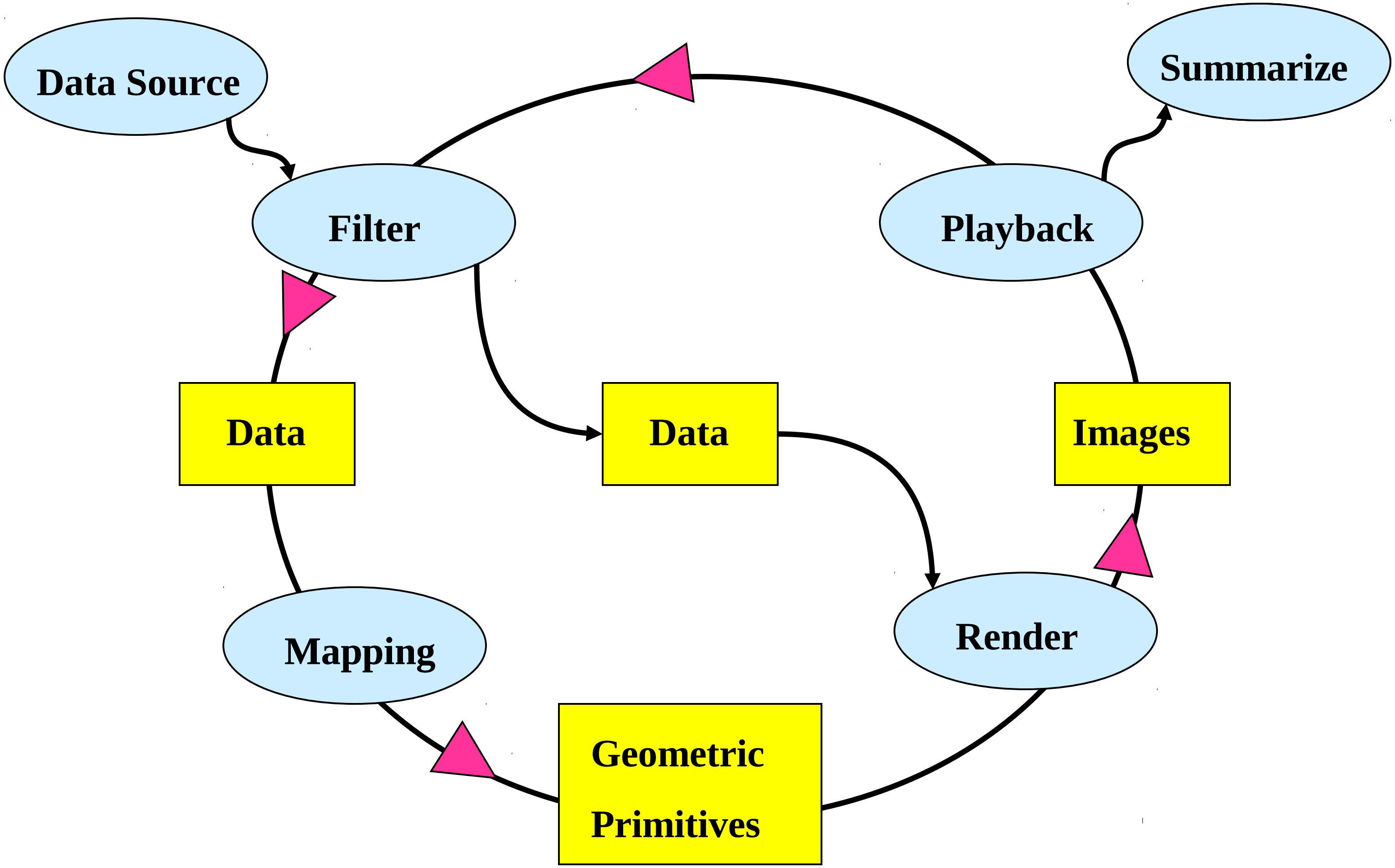
Zakaj vizualizacija podatkov?

A Picture is worth a Million Bytes

```
34   4F   D2   73   89    2E

12    90  E1   45    36    FF

FF    6A   B4   78   54   23

D2   AA   38   90    87   54

DD   C2   FF   89   00    76
```

Data → Information

Proces znanstvene vizualizacije



Data Source

Summarize

Filter

Playback

Data

Data

Images

Mapping

Render

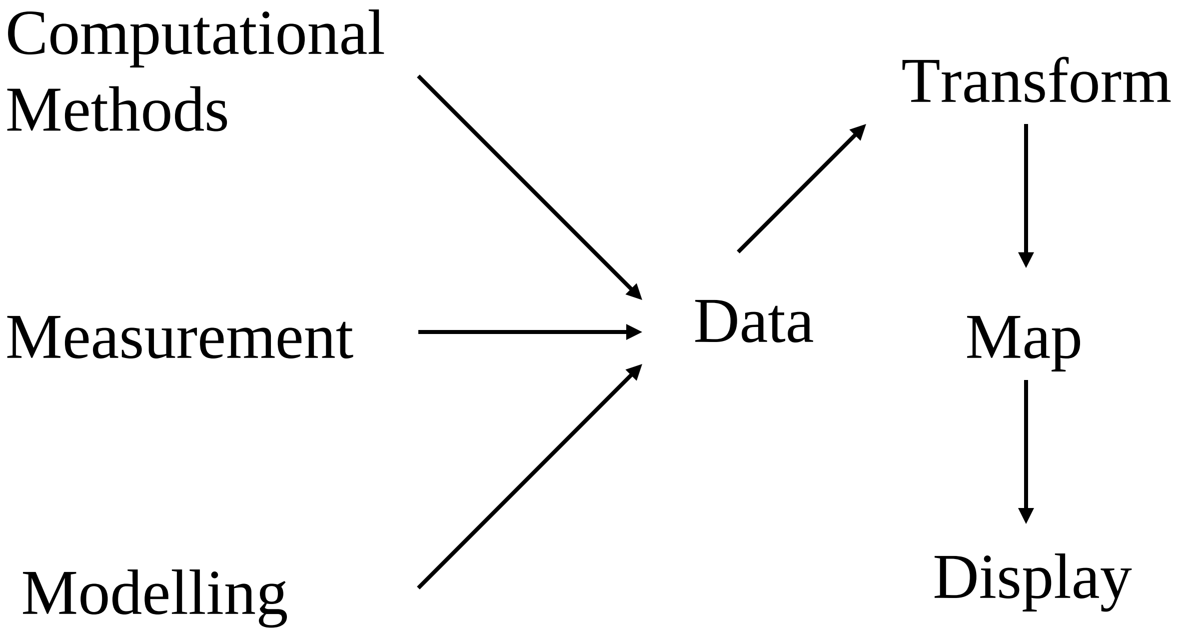
Geometric

Primitives

Types of Source Data

- Measured (sensed, observed, experimental)
- Simulated (computed)

Predstavitev podatkov za vizualizacijo



Computational Methods

Measurement

Modelling

Data

Transform

Map

Display

Possible Research

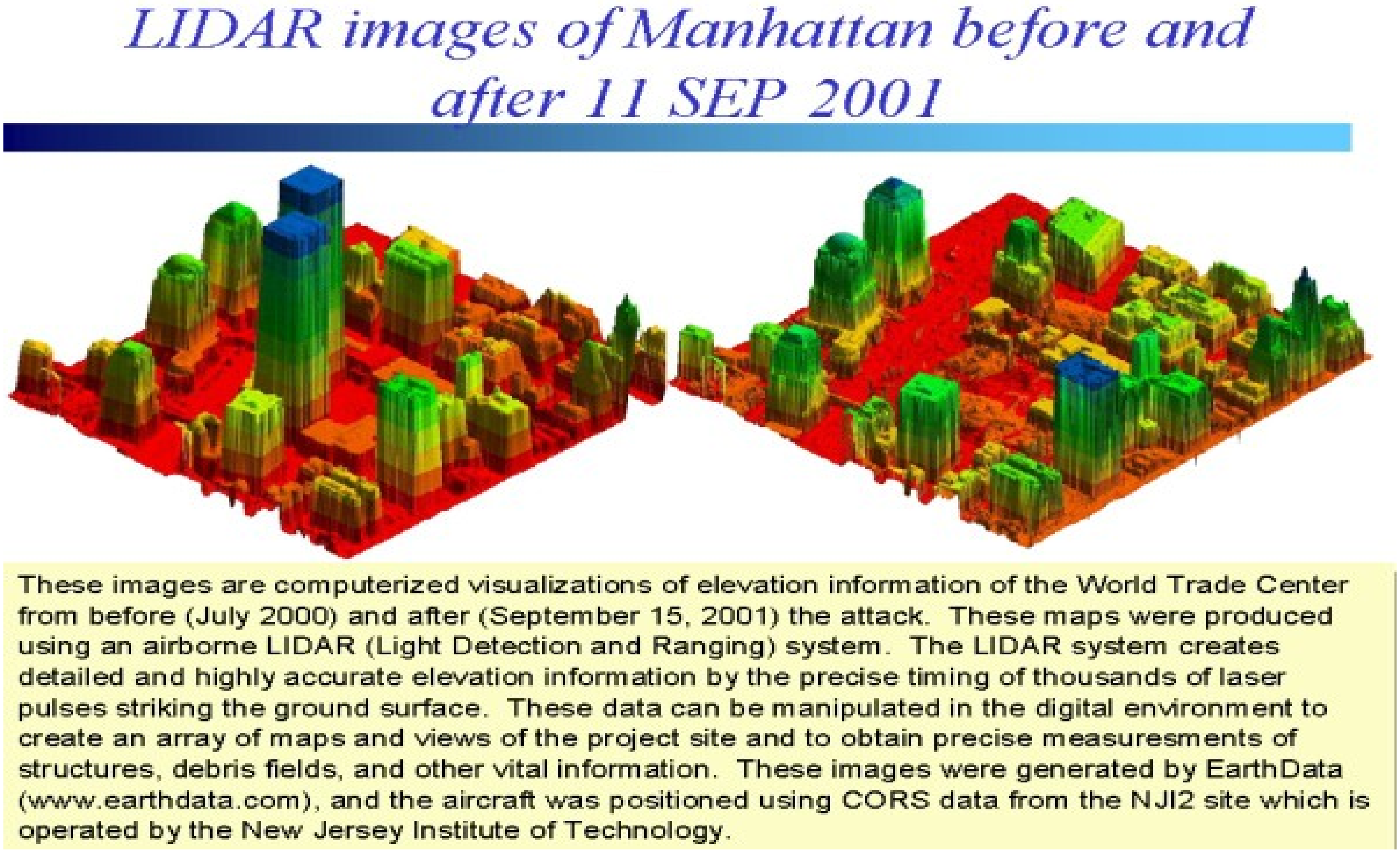
- Visualization of time-dependent motion
- Change of topology
- Interactive feature extraction
- Interactive exploration
- Use of force feedback in visualization
- Handling of Multi-Gigabyte datasets
- Exploration of high-dimensional spaces

Types of Scientific Visualization

- I See
  - interactive
  - exploratory
  - softcopy display
  - minimal contextual information
- We See
  - interactive
  - exploratory
  - softcopy display
  - more context/explanation needed
- They See
  - presentation usually
  - not interactive
  - not exploratory
  - hardcopy display
  - lots of context/explanation needed

Data and Scientific Visualisation

Measured Data – LIDAR (Light Detection and Ranging)

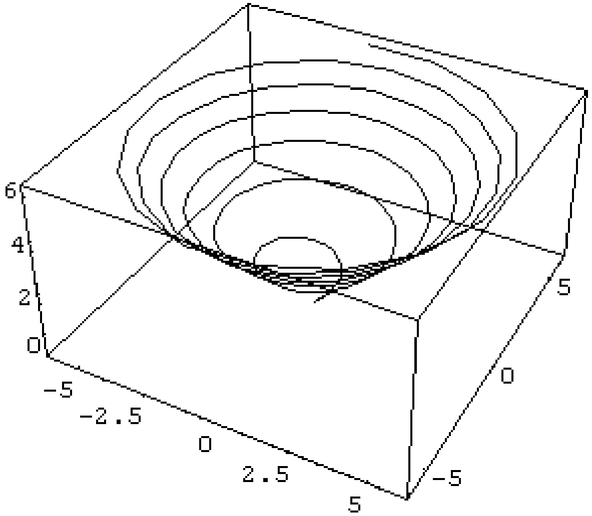


LIDAR images of Manhattan before and after 11 SEP 2001

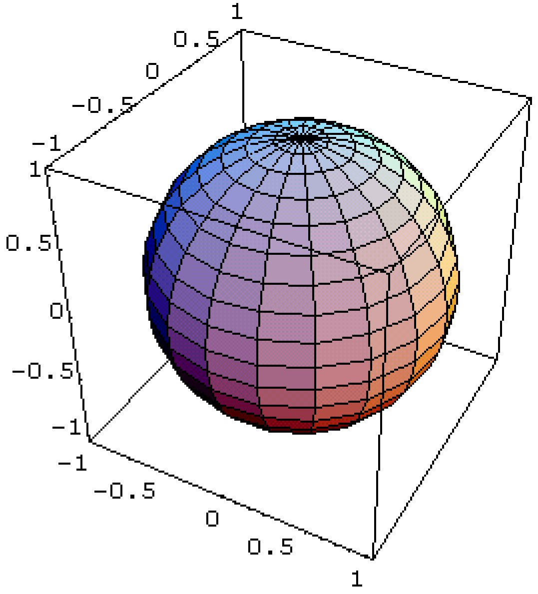
These images are computerized visualizations of elevation information of the World Trade Center from before (July 2000) and after (September 15, 2001) the attack. These maps were produced using an airborne LIDAR (Light Detection and Ranging) system. The LIDAR system creates detailed and highly accurate elevation information by the precise timing of thousands of laser pulses striking the ground surface. These data can be manipulated in the digital environment to create an array of maps and views of the project site and to obtain precise measuresments of structures, debris fields, and other vital information. These images were generated by EarthData (www.earthdata.com), and the aircraft was positioned using CORS data from the NJI2 site which is operated by the New Jersey Institute of Technology.

Primeri iz programa Mathematica

```
ParametricPlot3D[{t*Cos[t^2],
 t*Sin[t^2],t},
{t,0,2Pi},PlotPoints->200]
```

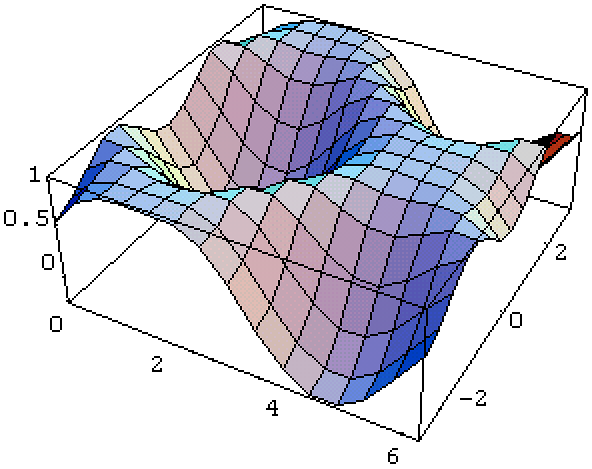


```
ParametricPlot3D[{Cos[u] Cos[v],
Sin[u] Cos[v],Sin[v]},{u,0,2Pi},
{v,-Pi/2,Pi/2}]
```

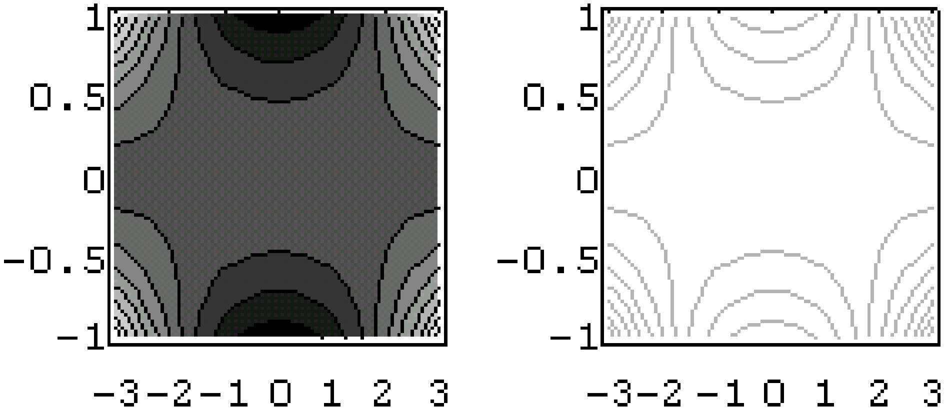


Primeri iz programa Mathematica

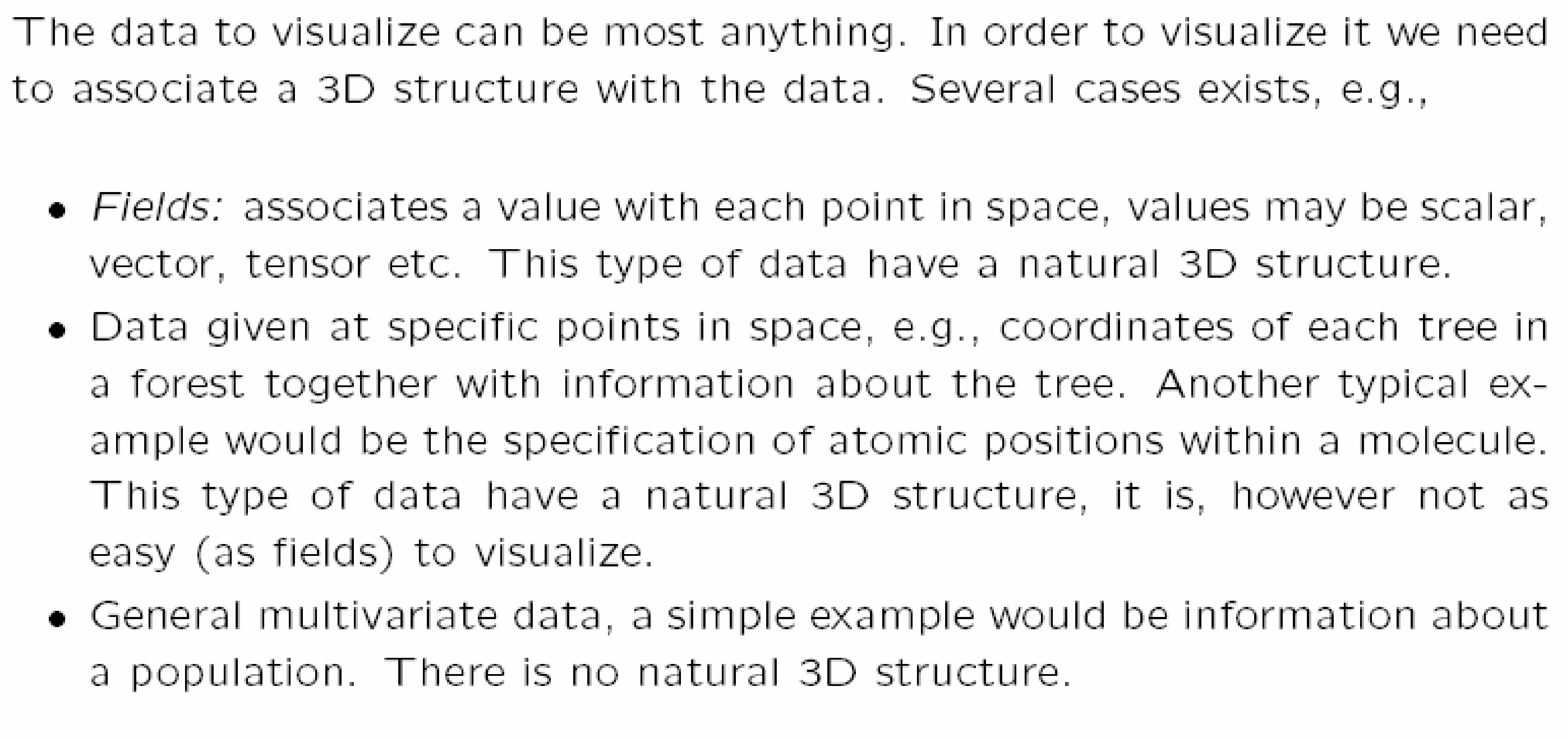
```
Plot3D[Cos[Sin[x]+Cos[y]],
{x,0,2Pi},{y,-Pi,Pi}]
```



```
f[x_,y_]=(x^2-3)y^2;
cpone=ContourPlot[f[x,y],{x,-3,3},
        {y,-1,1},
        DisplayFunction->Identity];
cptwo=ContourPlot[f[x,y],{x,-3,3},
        {y,-1,1},
        ContourStyle->GrayLevel[.7],
        ContourShading->False,
        DisplayFunction->Identity];
Show[GraphicsArray[{cpone, cptwo}]]
```



Kaj je podatek?



The data to visualize can be most anything. In order to visualize it we need to associate a 3D structure with the data. Several cases exists, e.g.,

- *Fields:* associates a value with each point in space, values may be scalar, vector, tensor etc. This type of data have a natural 3D structure.
- Data given at specific points in space, e.g., coordinates of each tree in a forest together with information about the tree. Another typical example would be the specification of atomic positions within a molecule. This type of data have a natural 3D structure, it is, however not as easy (as fields) to visualize.
- General multivariate data, a simple example would be information about a population. There is no natural 3D structure.

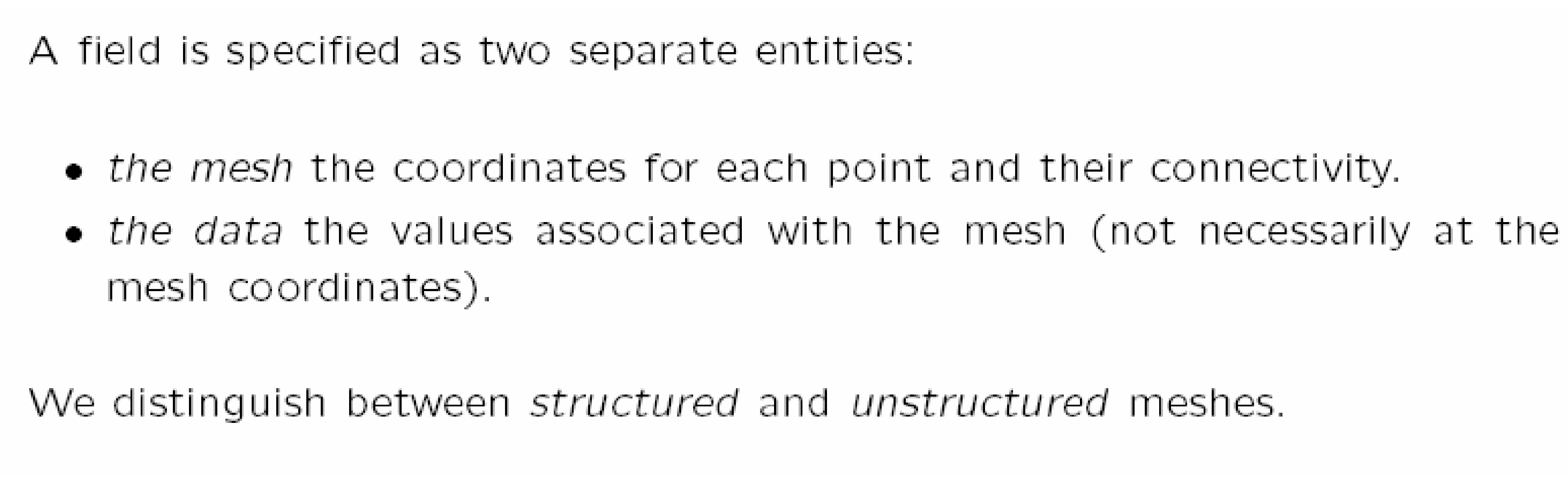
Polja (fields)

Fields occur in many typical physics or engineering problems, in these contexts they usually appear as the solution to some partial differential equation. A typical example would be the velocity field of a liquid.

Depending on the type of object described by the field we distinguish

- *Scalar:* a field which associates a single number with each point in space, examples are: temperature, pressure in a gas or liquid, concentration, wave function etc.
- *Vector:* a field which associates a vector with each point in space, examples are: velocity field in a gas or liquid, magnetic field, electric field etc.
- *Tensor:* a field which associates a tensor (a 3×3 matrix) with each point in space, the typical example is: stress (and strain) tensor in a solid.

Mreže in podatki (mesh and data)

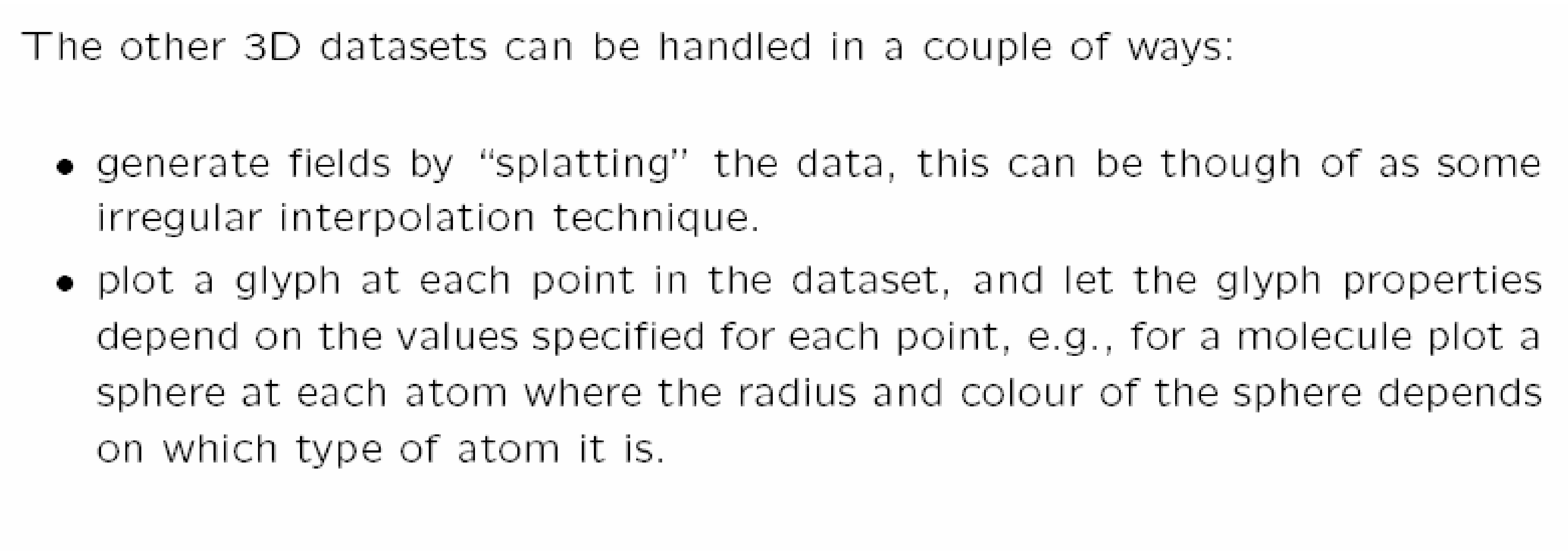


A field is specified as two separate entities:

- *the mesh* the coordinates for each point and their connectivity.
- *the data* the values associated with the mesh (not necessarily at the mesh coordinates).

We distinguish between *structured* and *unstructured* meshes.

Drugi tipi 3D podatkov



The other 3D datasets can be handled in a couple of ways:

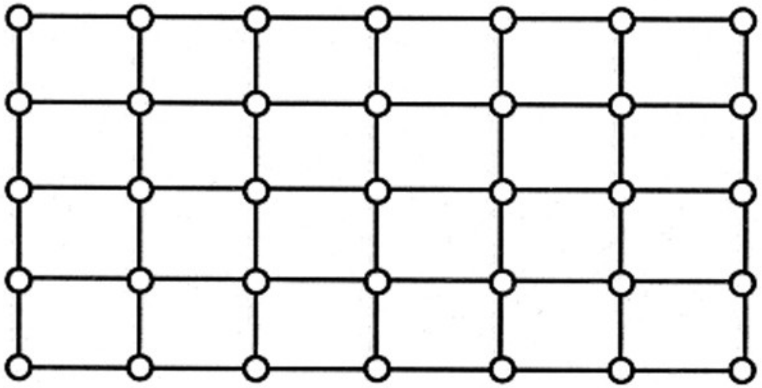
- generate fields by "splatting" the data, this can be though of as some irregular interpolation technique.
- plot a glyph at each point in the dataset, and let the glyph properties depend on the values specified for each point, e.g., for a molecule plot a sphere at each atom where the radius and colour of the sphere depends on which type of atom it is.

Dimenzije in tipi podatkov

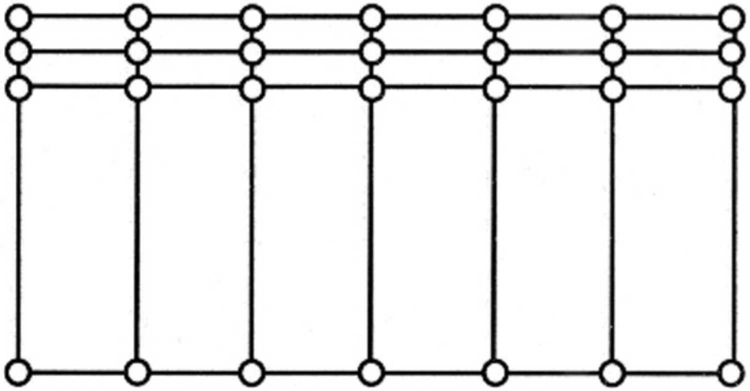
- domain (independent variables)
  - 2D, 3D, 4D, etc.
  - if the data points are not connected: scattered
  - if the domain is gridded:
    - rectilinear (uniform [regular] or non-uniform [irregular])
      - cubes: with data points at vertices, on faces, or in the center
    - curvilinear
      - distorted cubes, but 6-connectivity at nodes, analogy is a distorted sponge
    - unstructured
      - 2D - triangles
      - 3D – tetrahedra, prisms, pyramids, or hexahedra
- range (dependent variables)
  - scalar (pressure, density, temperature, voltage, height, salinity)
  - vector (velocity, current, gradient, etc.)
  - tensor (stress, strain, etc.)

Grids

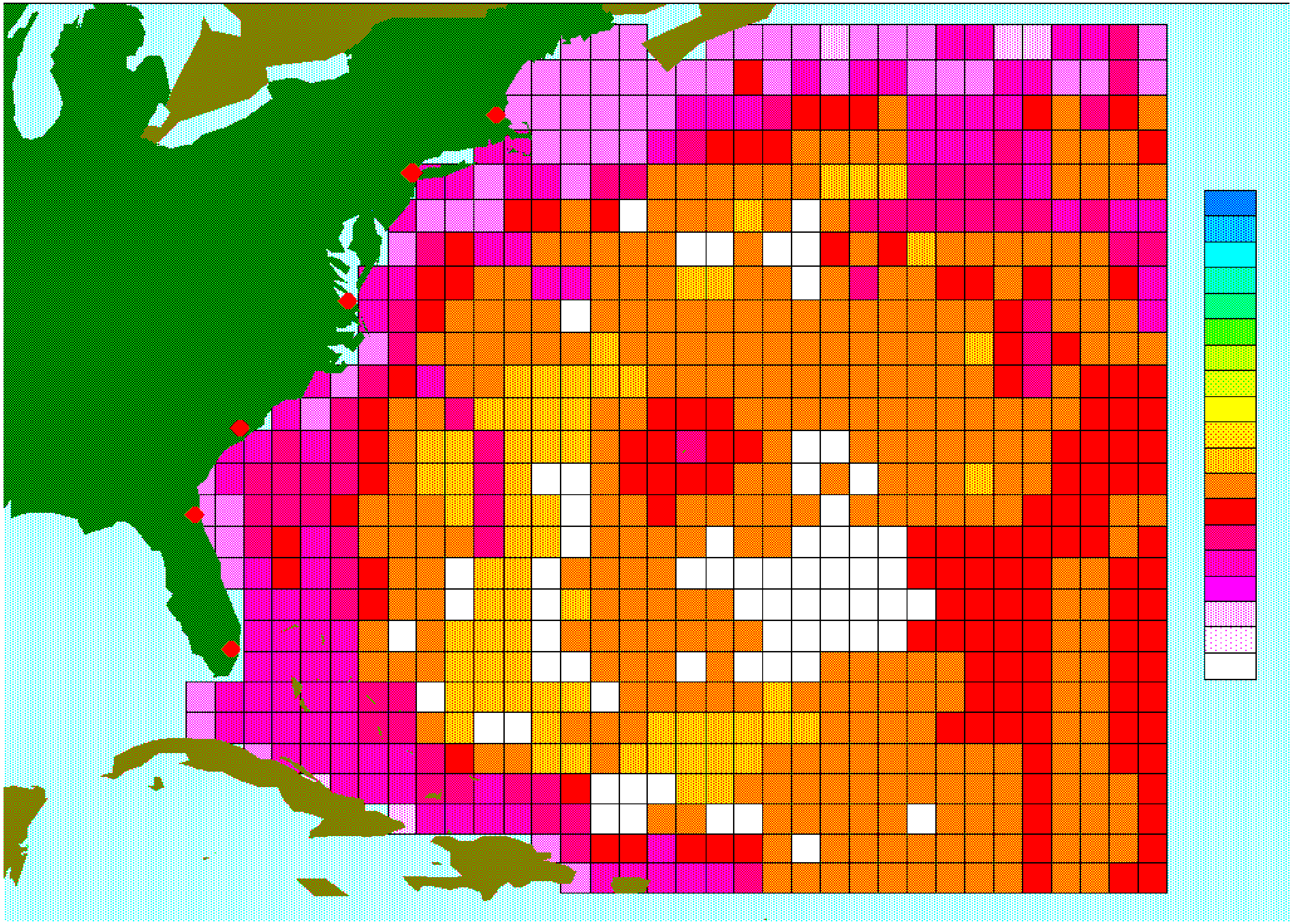
Uniform Rectilinear



Non–uniform Rectilinear

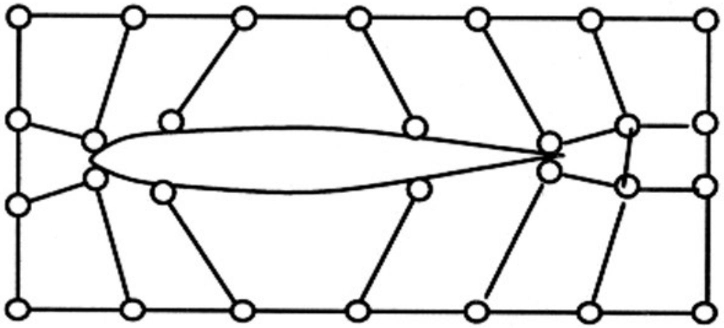


Uniform Data

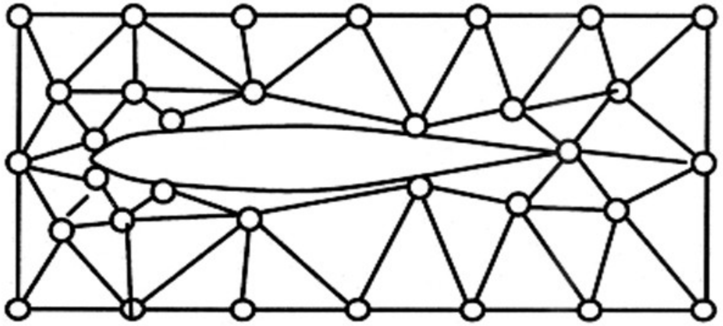


Grids

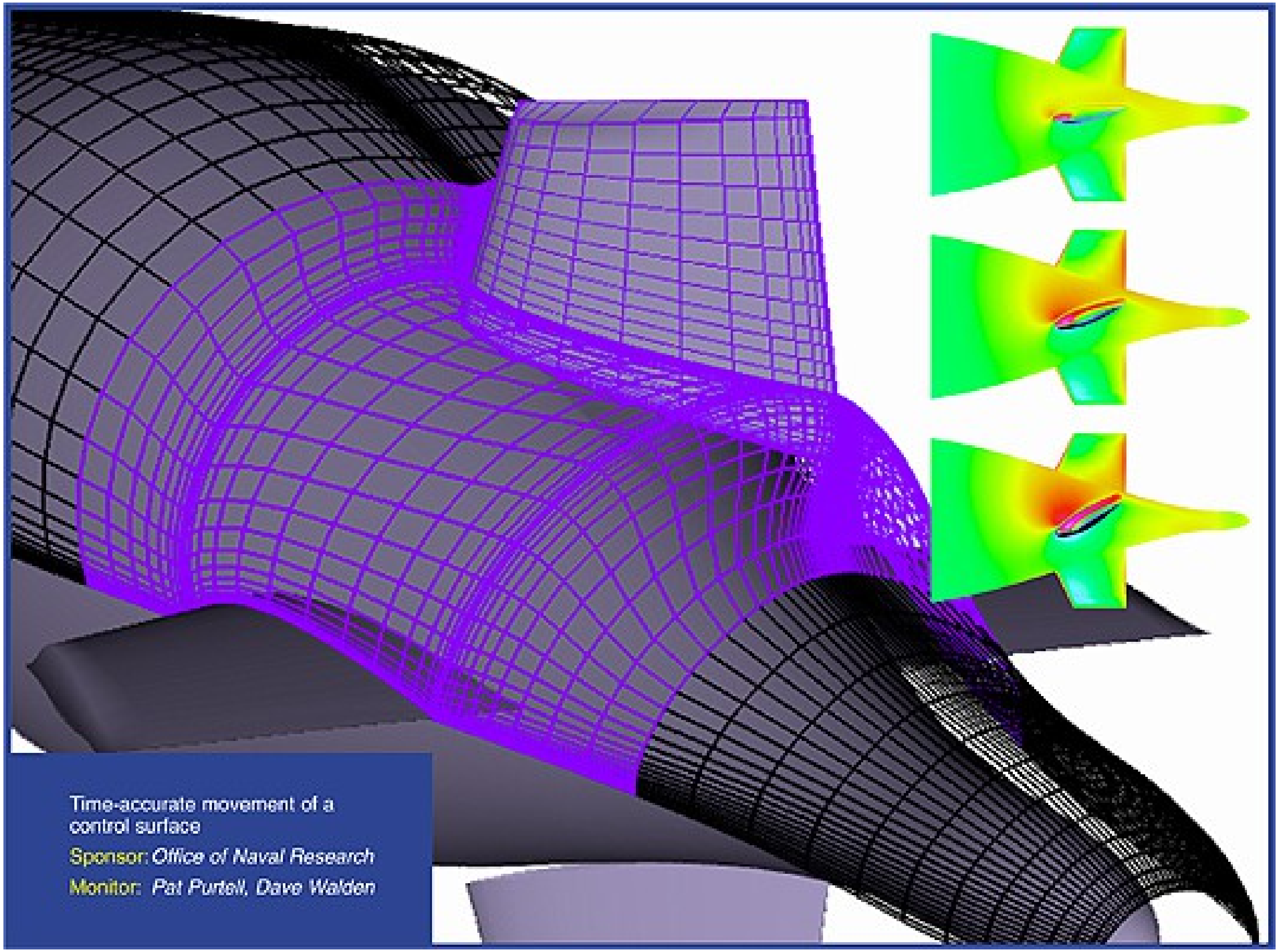
Curvilinear



Triangular



Curvilinear Data

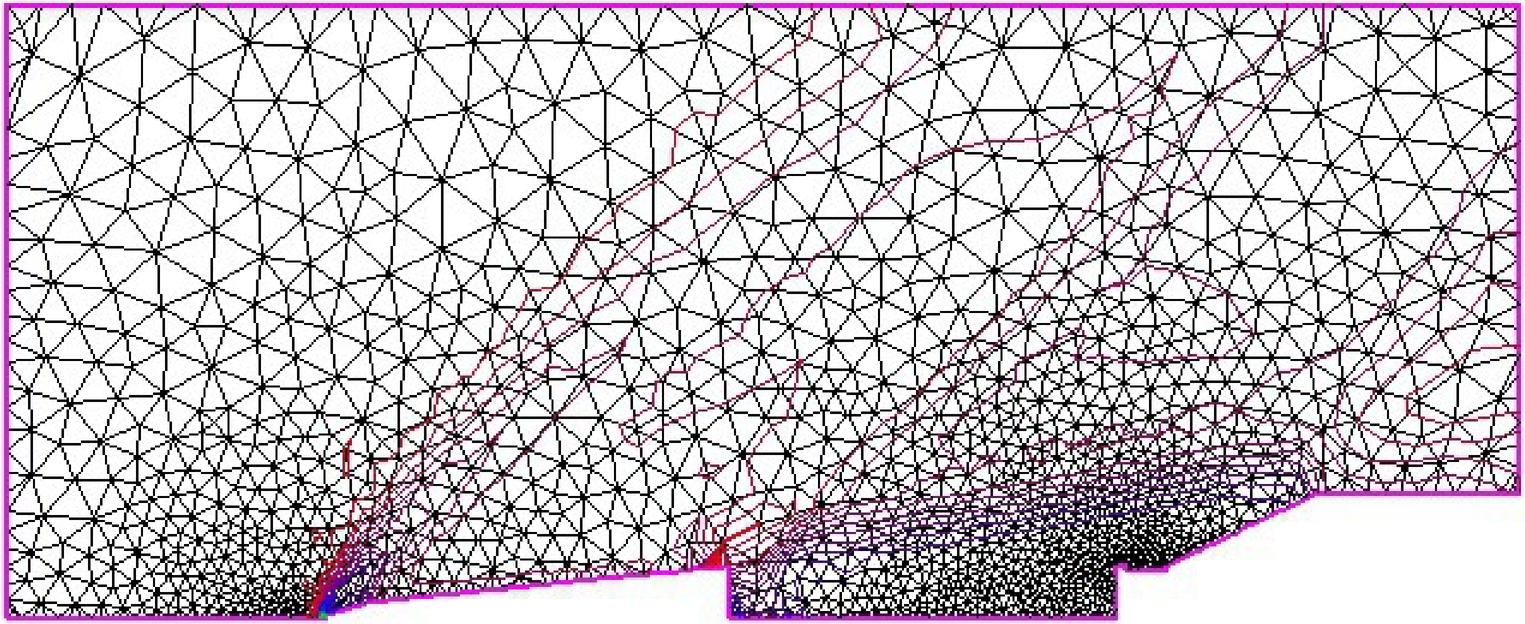


Time-accurate movement of a control surface

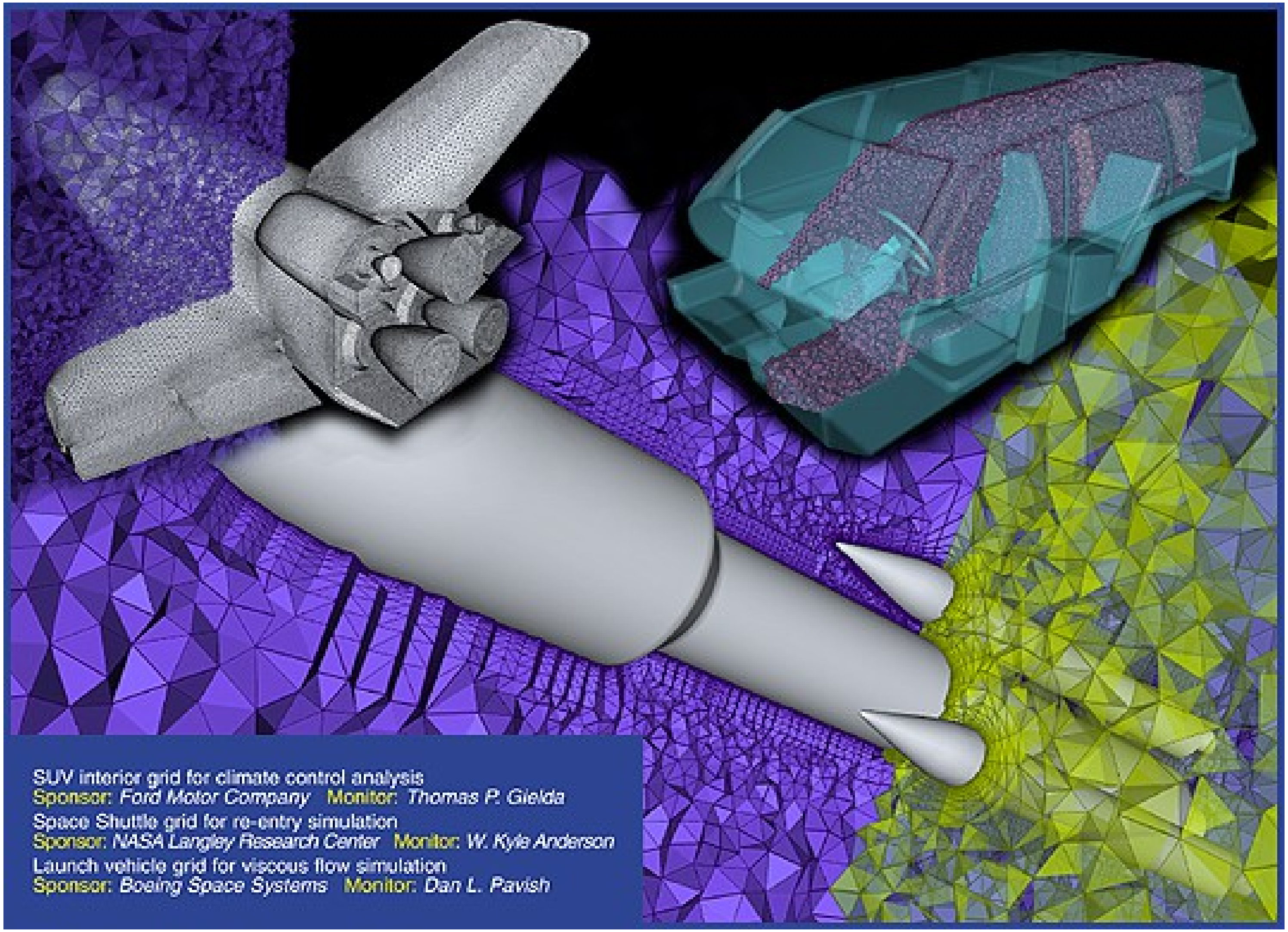
Sponsor: *Office of Naval Research*

Monitor: *Pat Purtell, Dave Walden*

Unstructured Data



Unstructured Data

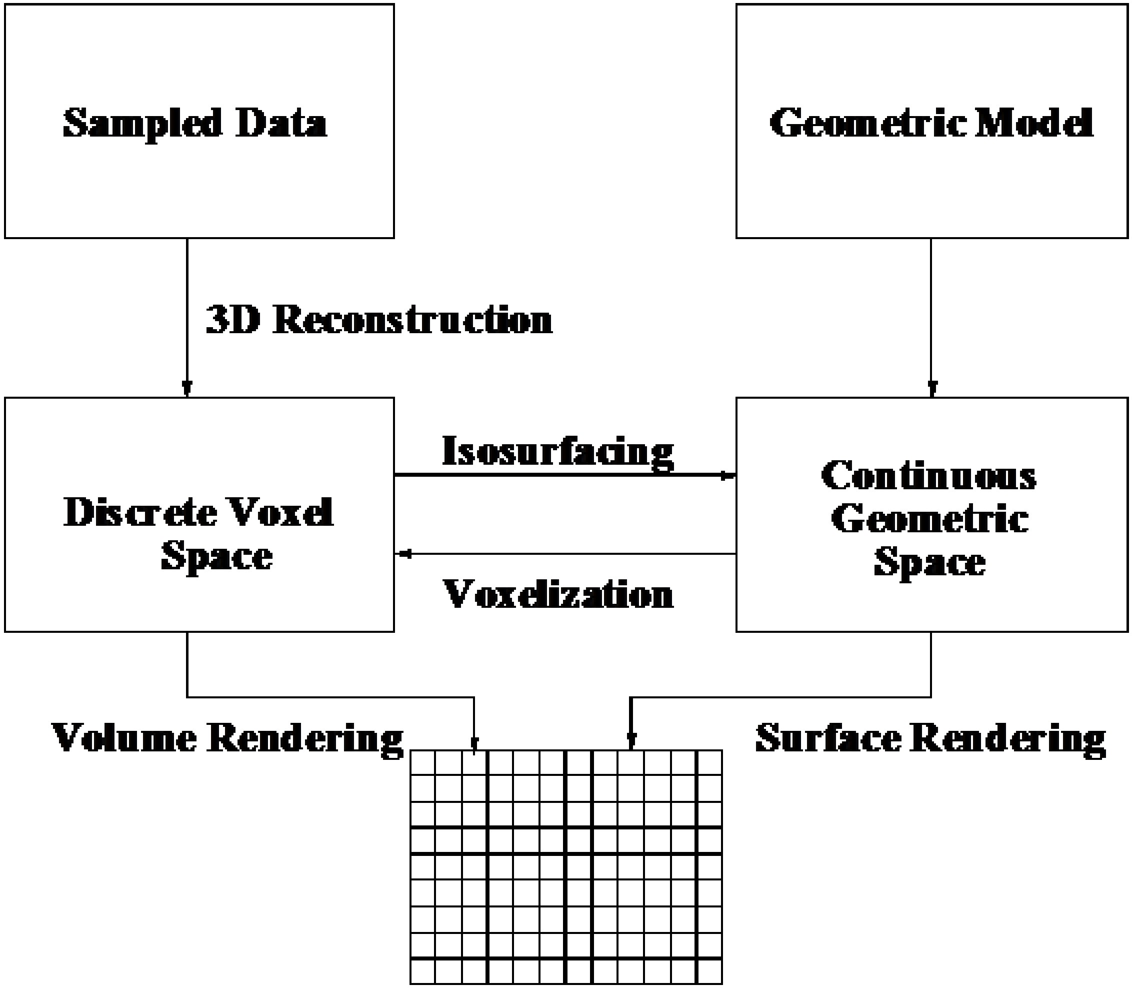


SUV interior grid for climate control analysis
Sponsor: *Ford Motor Company* Monitor: *Thomas P. Gielda*
Space Shuttle grid for re-entry simulation
Sponsor: *NASA Langley Research Center* Monitor: *W. Kyle Anderson*
Launch vehicle grid for viscous flow simulation
Sponsor: *Boeing Space Systems* Monitor: *Dan L. Pavish*

Common steps in volume visualization algorithms

- data acquisition
- slice processing
- dataset reconstruction
- dataset classification
- mapping into primitives

Tehnike za 3D skalarne veličine



Taxonomy of Volume Visualization

Klasifikacija podatkov

- most difficult task
- choose threshold if SF algorithm
- choose color and opacity for range of data values if DVR
- other factors
  - user’s knowledge of the material (data) being visualized
    - if the f(**x**) are density values and you know bone has a density of 0.8 and you want to see bone, you set the opacity of f(**x**)=0.8 to 1.0
  - the location of the classified elements in the volume
    - often want outer layers more transparent, thus element position is factored into the formula
  - the material occupancy of the element
    - When the data values are changing slowly, only a narrow range of values is acceptable as the isovalue surface. When the data values are changing rapidly, a wide range of values is acceptable as the isovalue surface.

How to turn data into an image

- Ultimately you turn on a discrete point on the computer screen or you put a finite quantity of colored material on some material or …wet chemistry.
- You have to make a lot of choices in how you want that done.
- The process can be divided in mathematical steps, scientific steps, and engineering steps.

Klasifikacija tehnik vizualizacije

- Glyph techniques – use symbols to represent values or states within a field of information.
- Surface Methods – extracts polygonal versions of calculated components.
- Direct Volume Rendering – generates a rendered image of the volume where volume elements are projected directly onto the image plane.

Metode vizualizacije prostornin

- Cutting Planes
- Isosurfaces
- Direct Volume Rendering

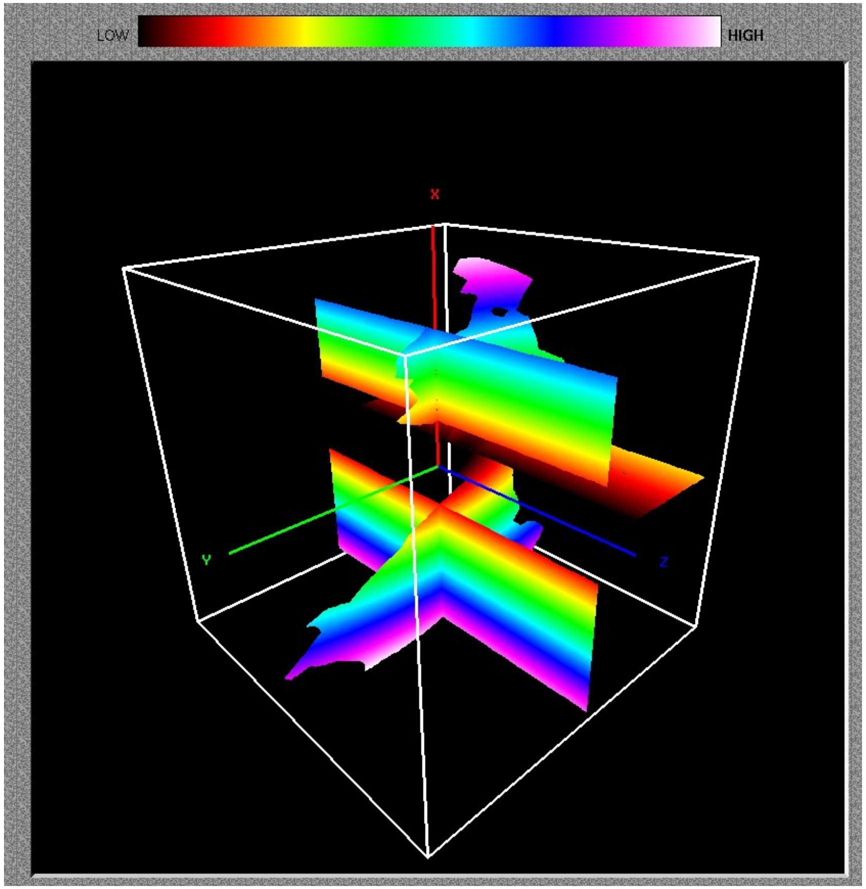
Metode vizualizacije prostornin

The fundamental algorithms are of two types: **direct volume rendering (DVR)** algorithms and **surface-fitting (SF)** algorithms.

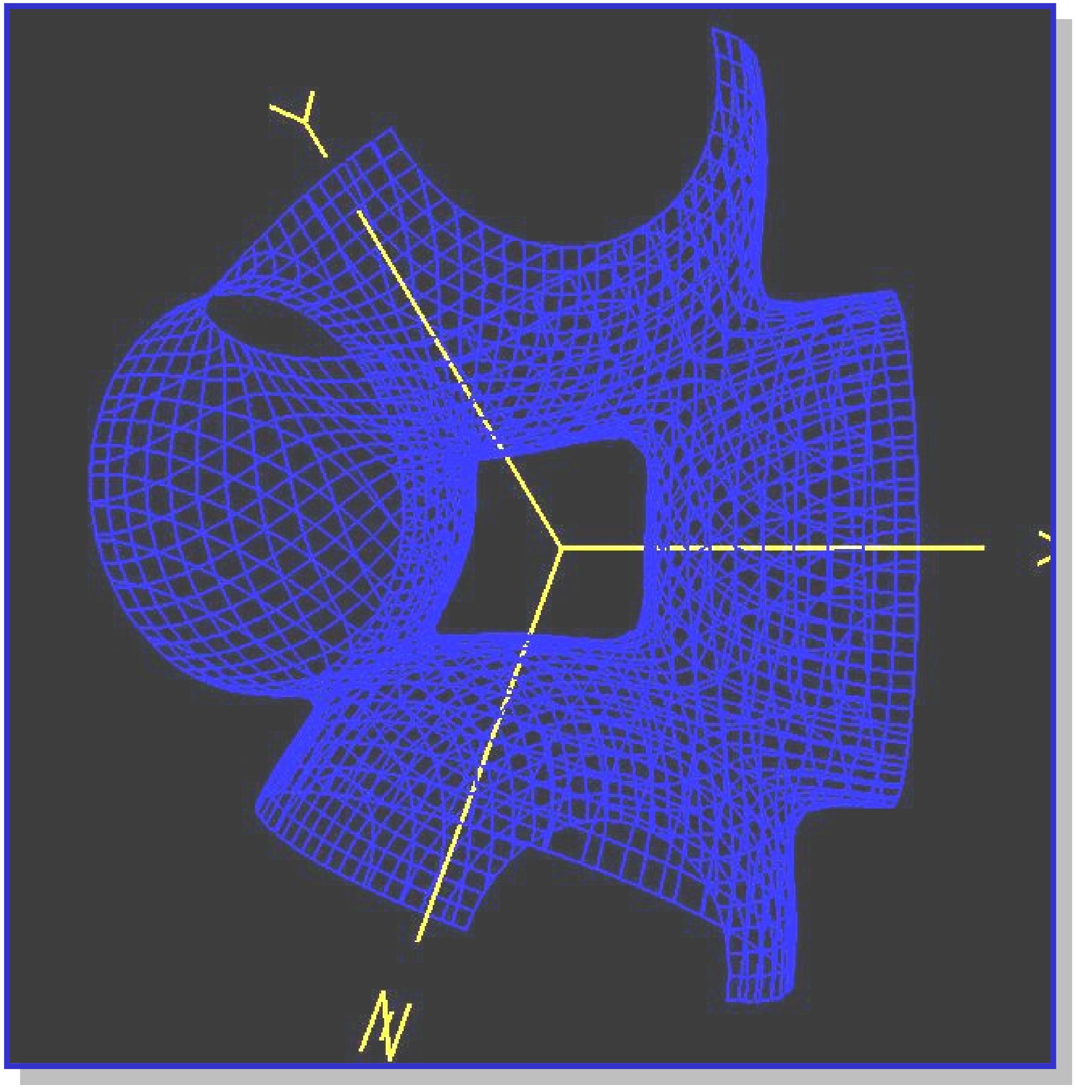
DVR methods map elements directly into screen space without using geometric primitives as an intermediate representation. DVR methods are especially good for datasets with amorphous features such as clouds, fluids, and gases. A disadvantage is that the entire dataset must be traversed for each rendered image. Sometimes a low resolution image is quickly created to check it and then refined. The process of increasing image resolution and quality is called "progressive refinement".

SF methods are also called feature-extraction or iso-surfacing and fit planar polygons or surface patches to constant-value contour surfaces. SF methods are usually faster than DVR methods since they traverse the dataset once, for a given threshold value, to obtain the surface and then conventional rendering methods (which may be in hardware) are used to produce the images. New views of the surface can be quickly generated. Using a new threshold is time consuming since the original dataset must be traversed again.

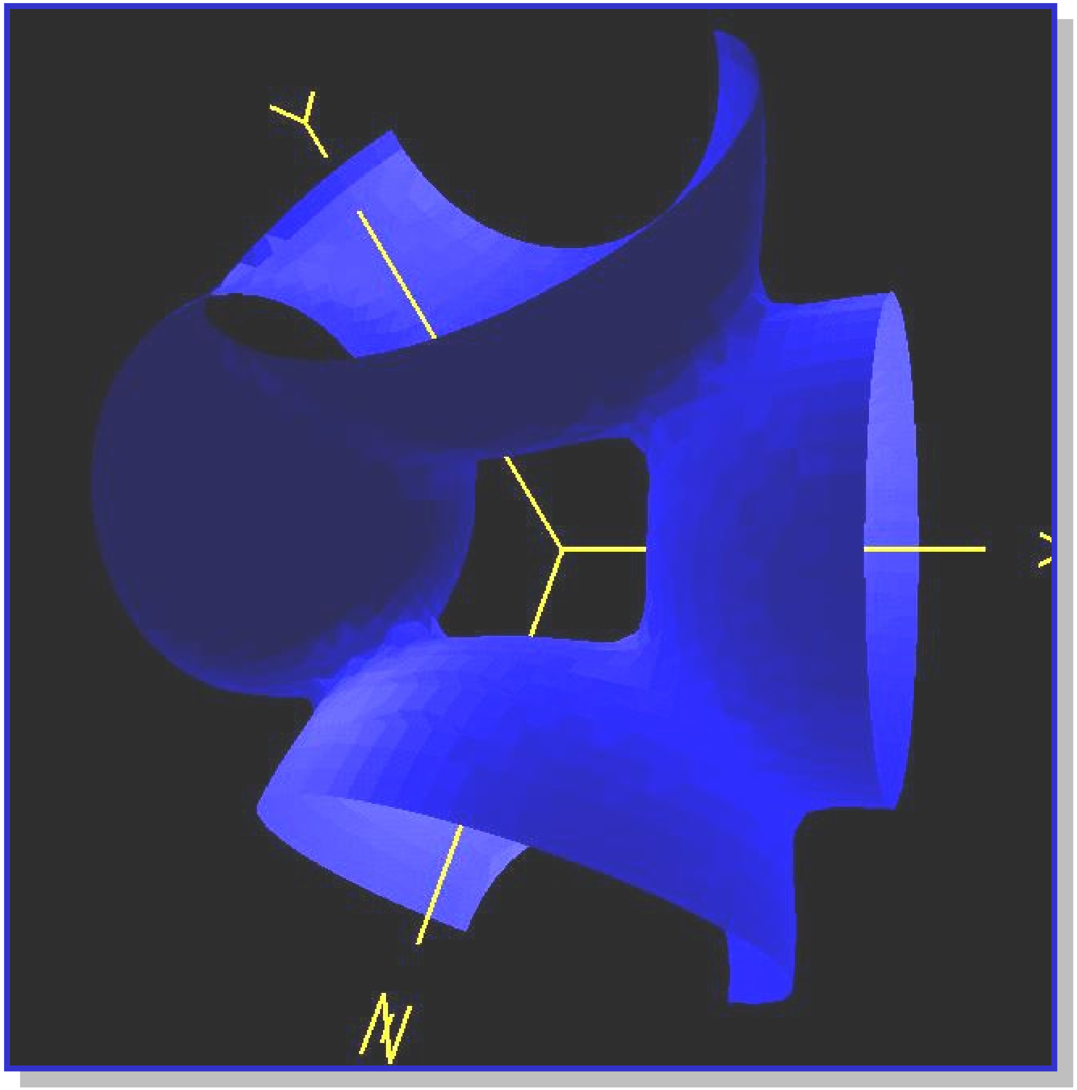
Cutting Planes



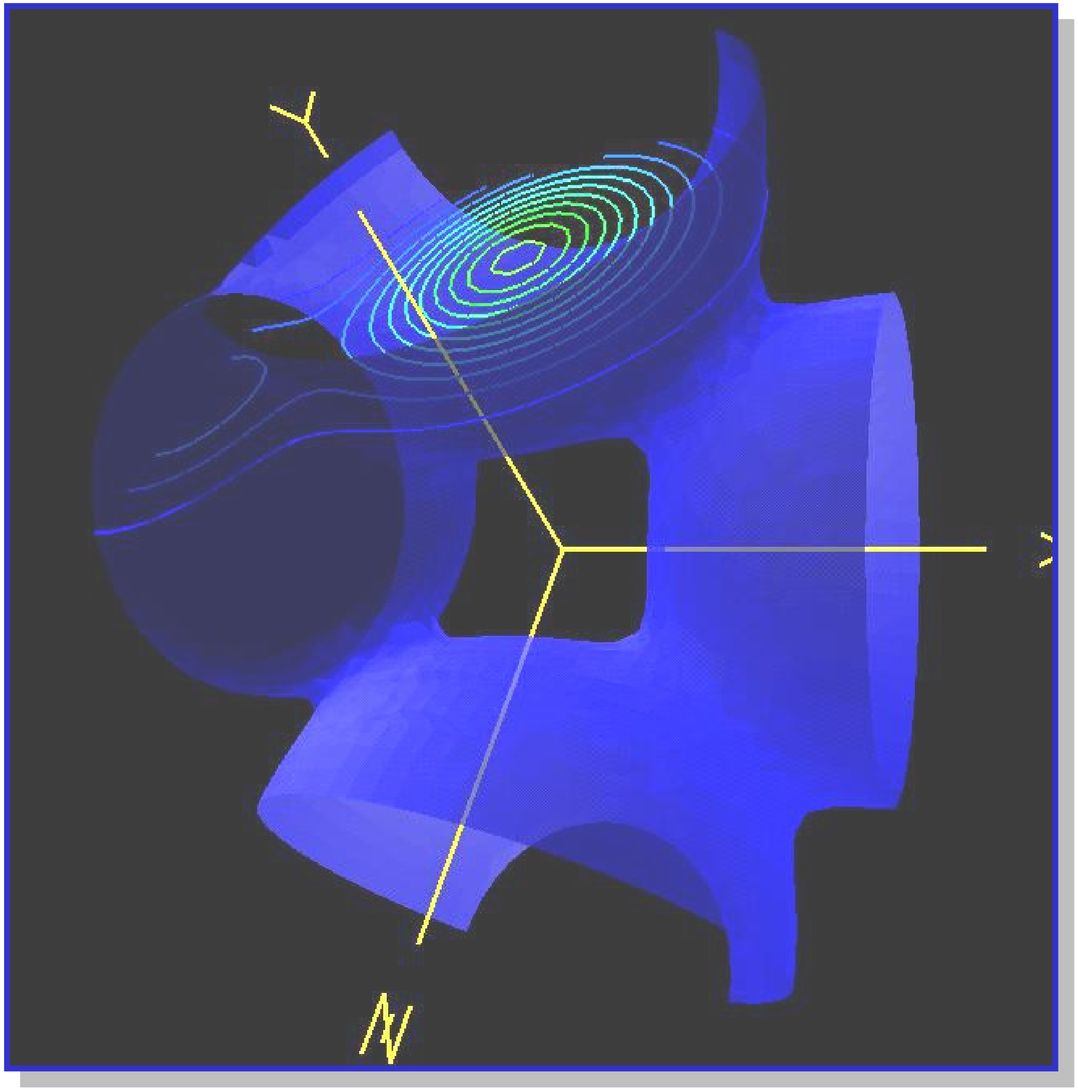
3D Wireframe Isosurface



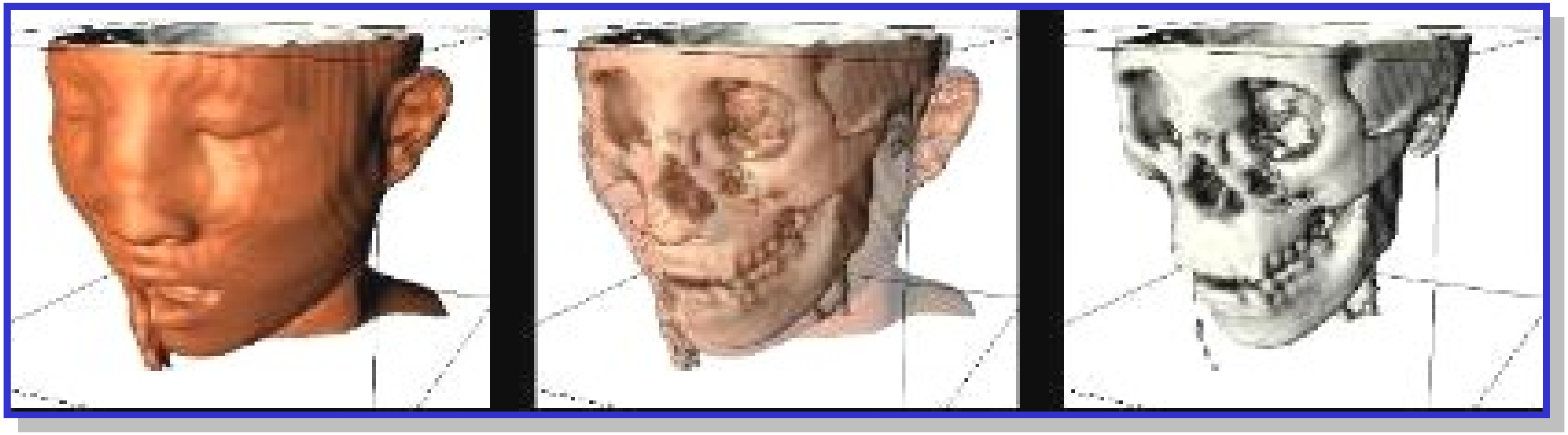
3D Polygonal Isosurface



Isosurface + Contour Plane



Isosurfaces of Medical Data



Konstrukcija Izo-površin (isosurfaces)

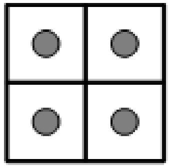
- The three steps to isosurface construction are:
  - detect surface(s)
  - fit geometric primitives to the detected surface(s)
  - render surface
- two methods for fitting geometric primitives:
  - contour connection methods
  - voxel intersection methods
    - dividing cubes
    - marching cubes
- *Marching Cubes* works by classifying the eight nodes of a voxel as being greater-than or less-than a threshold. These eight bits are used as an index into an edge intersection table. Linear interpolation is used to determine edge-intersection location and triangles are fitted to the edges. The triangles and the surface normals at vertices are added to a linked list for rendering.

Voksli in celice

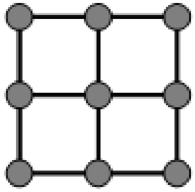
- The amount of structure in the dataset usually determines which volume visualization algorithm will create the most informative images.

Voxels and Cells

- Volumes of data can be treated as either an array of volume elements (voxels) or an array of cells.
  - **voxel - the area around a data point**



  - cell - views volume as a collection of hexahedra whose corners are the data points



Almost every volume visualization algorithm requires resampling which requires interpolation (or reconstruction).

Prečkanja (traversals)

- image-order - for each pixel see what objects (elements) project onto the pixel (remember, a pixel covers an area on the screen).
- object-order - for each object (element) see where it intersects the image plane
  - **can progress front-to-back (saves some computation potentially), or**
  - **back-to-front (can see scene evolve)**

Fotorealizem

Photorealism is a hotly debated issue in DVR. The issue is should objects look like they are made from “real” material? The problem is that plausible objects may be misleading, but non-plausible may be hard to interpret. No consensus exists.

Scalar Data in 3D

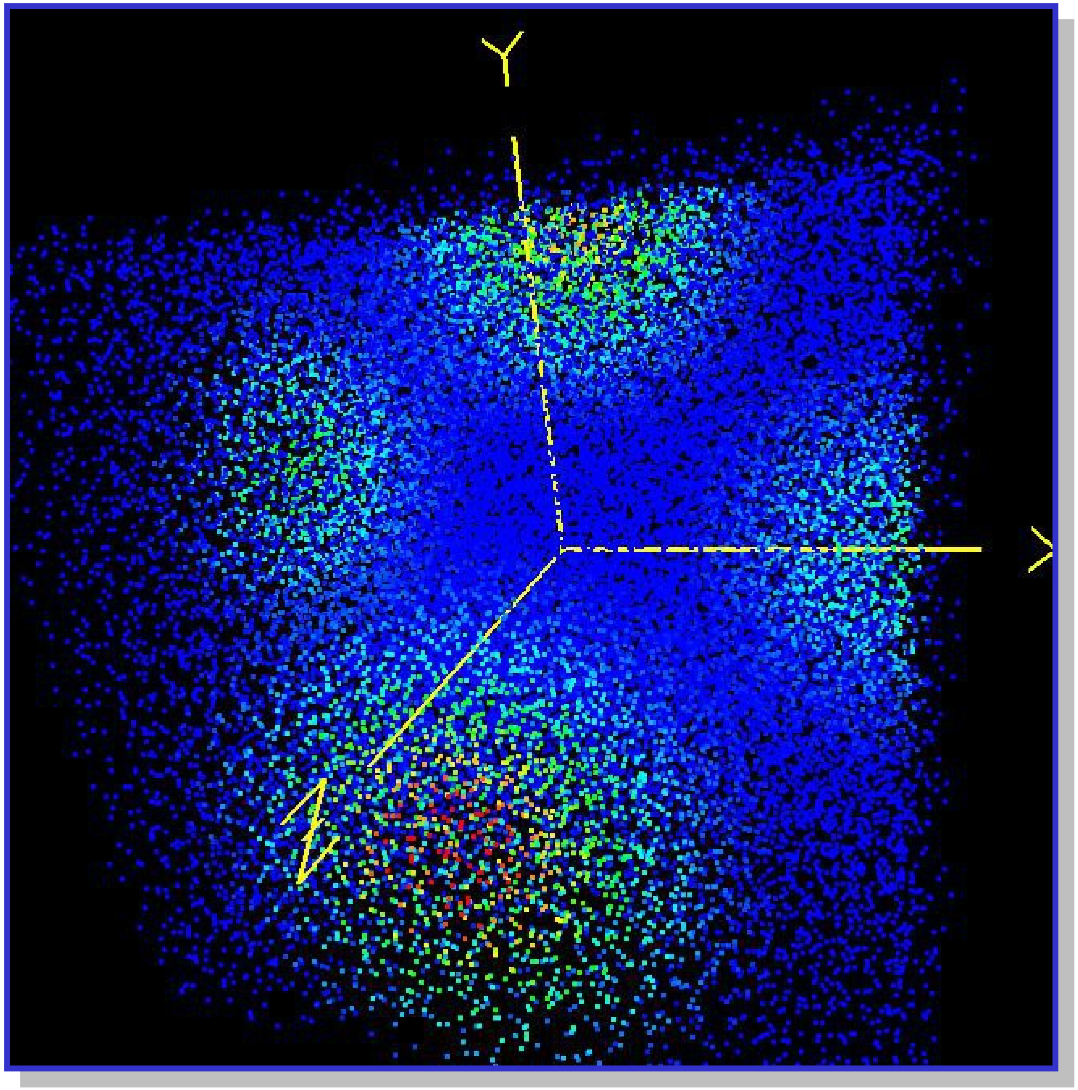
Examples:

- Temperatures in a room
- Molecular potentials

Techniques:

- Colored dots (“point cloud”)
- Cutting plane
- Isosurfaces

3D Jittered Point Cloud



Vector Data in 3D

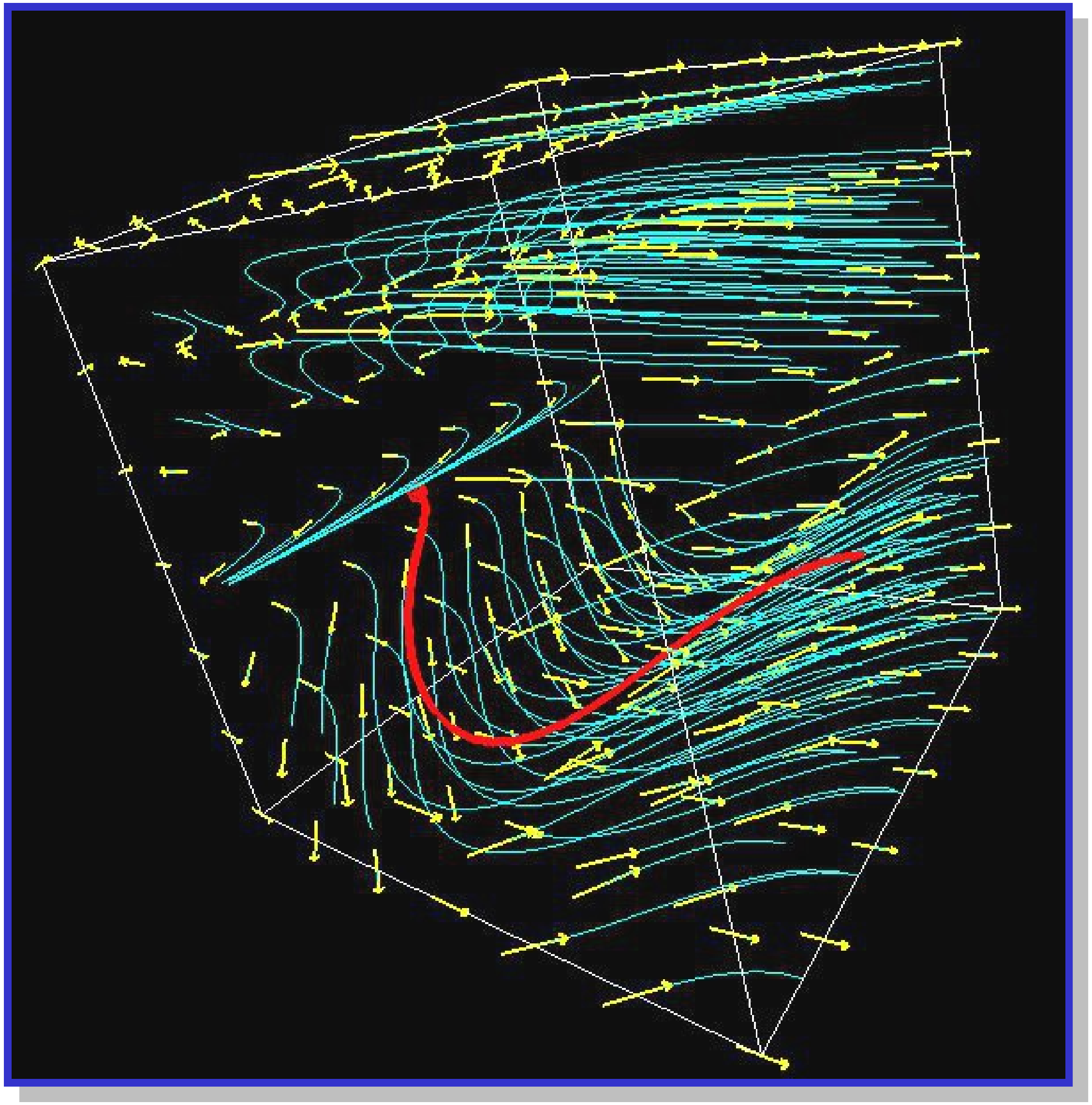
Examples:

- **3D flow field**

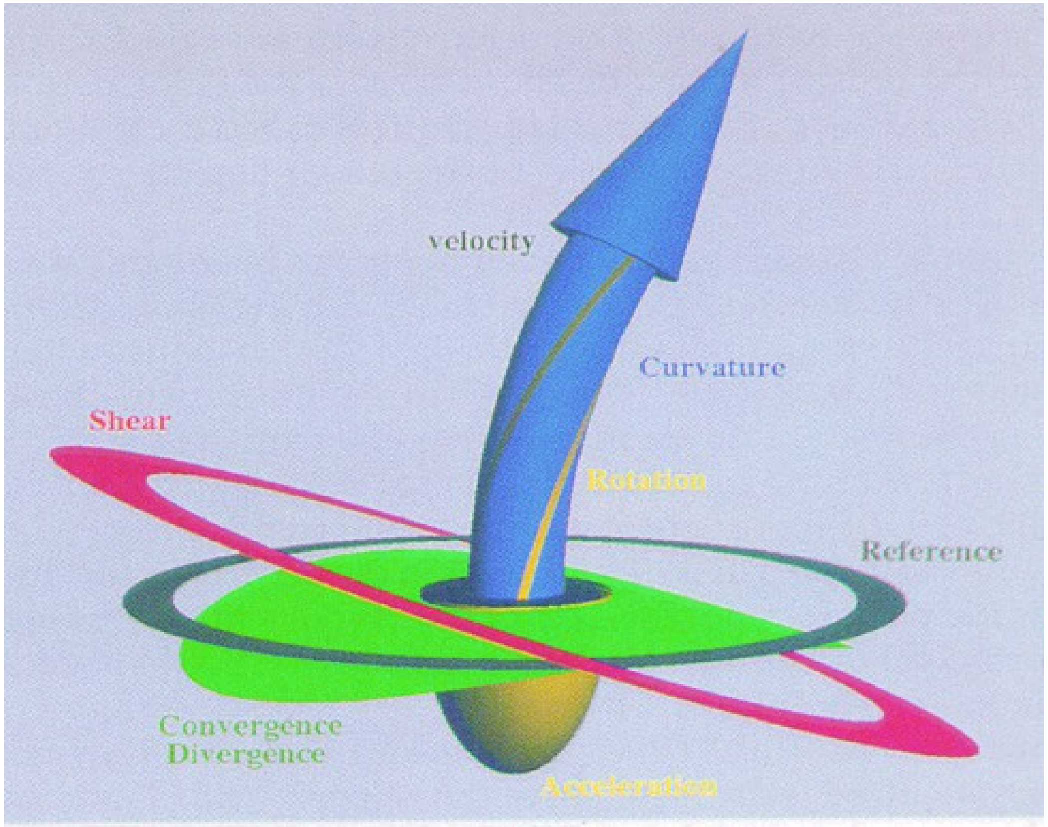
Techniques:

- **Arrows (“vector cloud”)**
- **Streamlines**

3D Flow Field

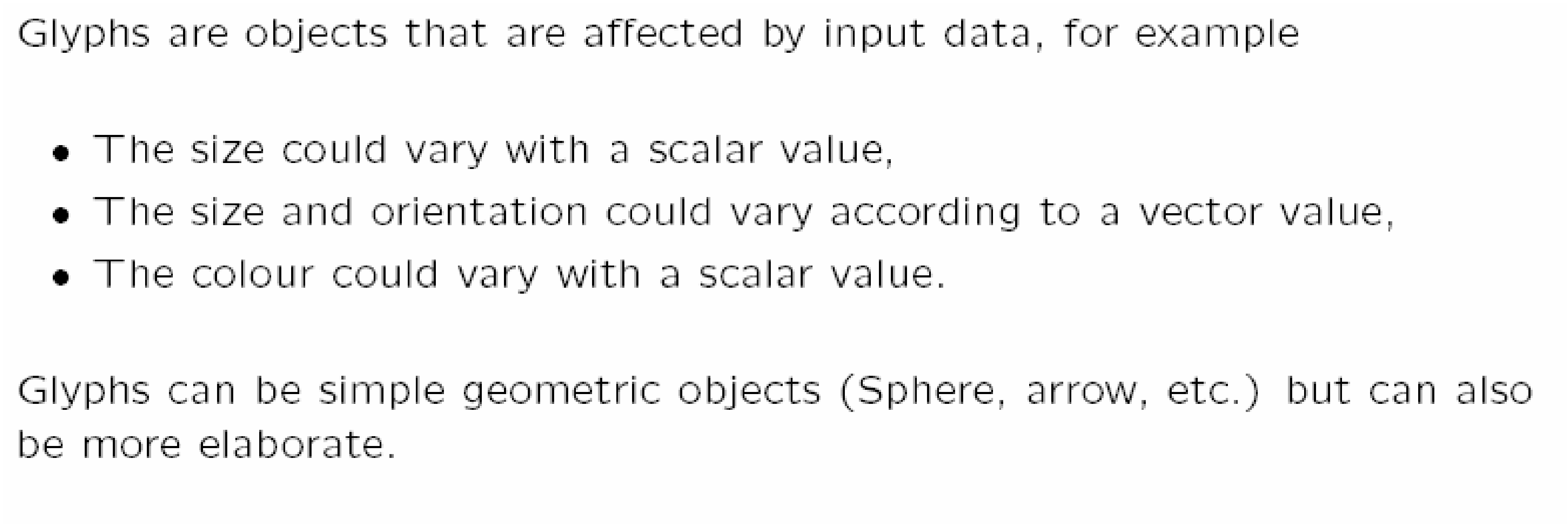


Tehnike s figurami (Glyph Techniques)



Glyph techniques – use symbols to represent values or states within a field of information.

Figure (glyphs)

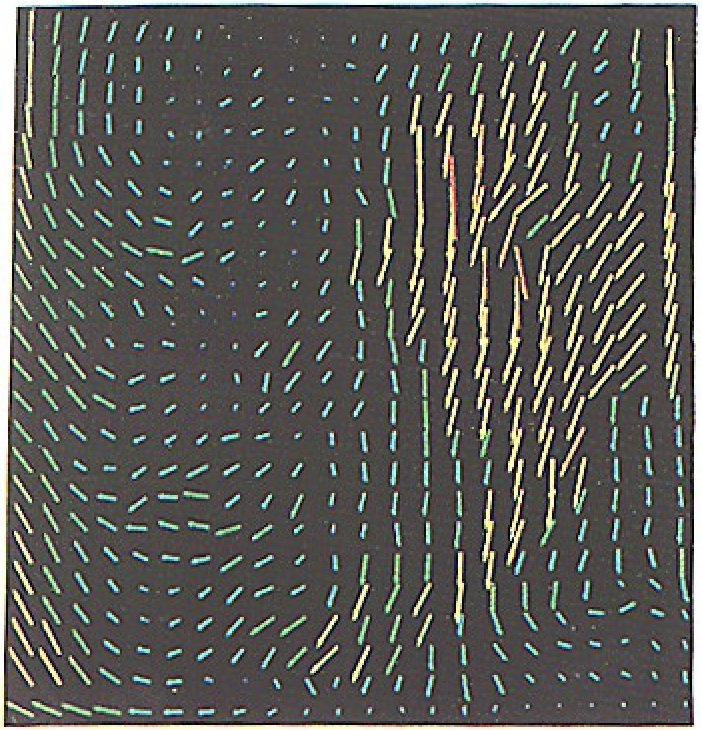


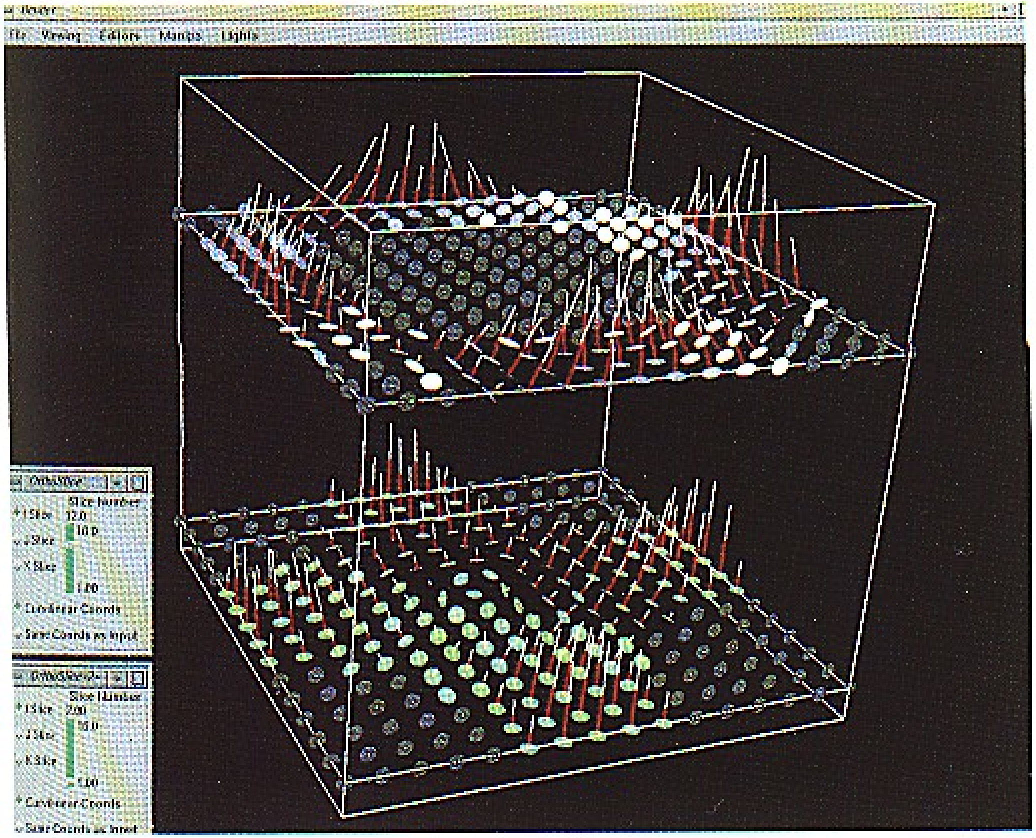
Glyphs are objects that are affected by input data, for example

- The size could vary with a scalar value,
- The size and orientation could vary according to a vector value,
- The colour could vary with a scalar value.

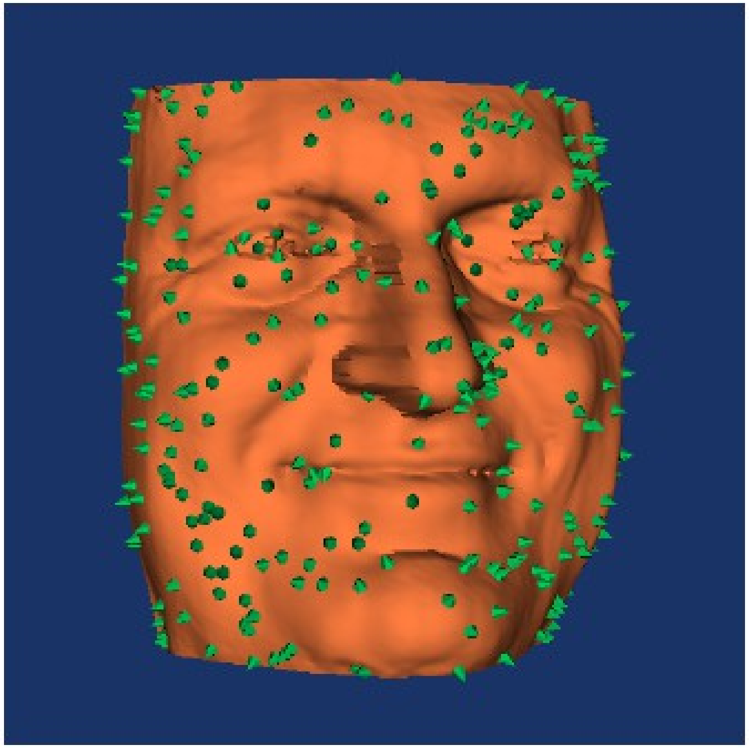
Glyphs can be simple geometric objects (Sphere, arrow, etc.) but can also be more elaborate.

Figure (Glyphs)

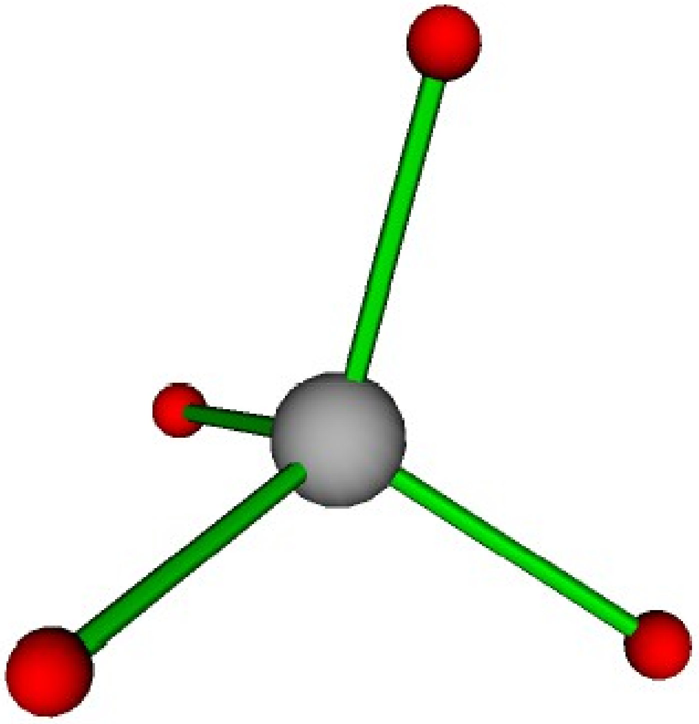




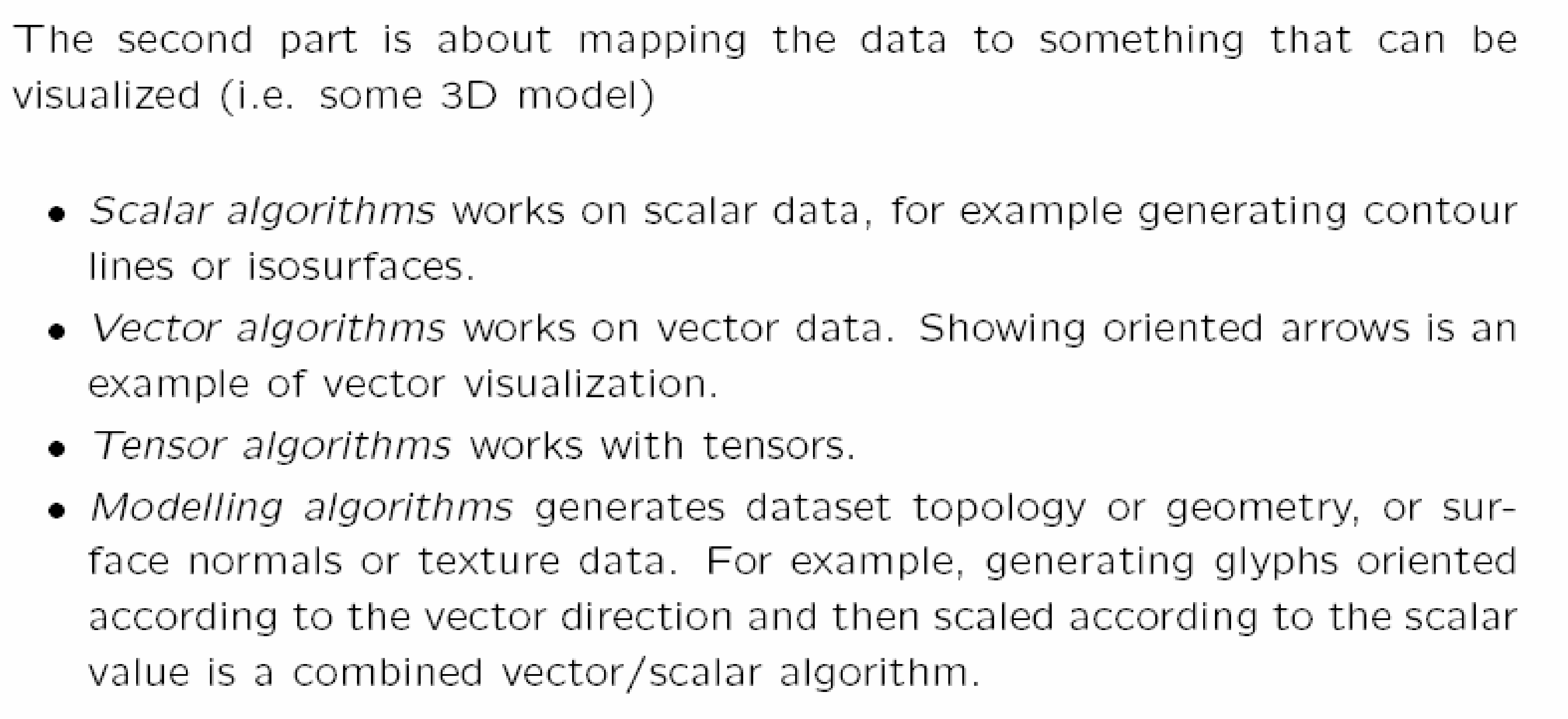
Glyph example: face



Glyph example: molecule



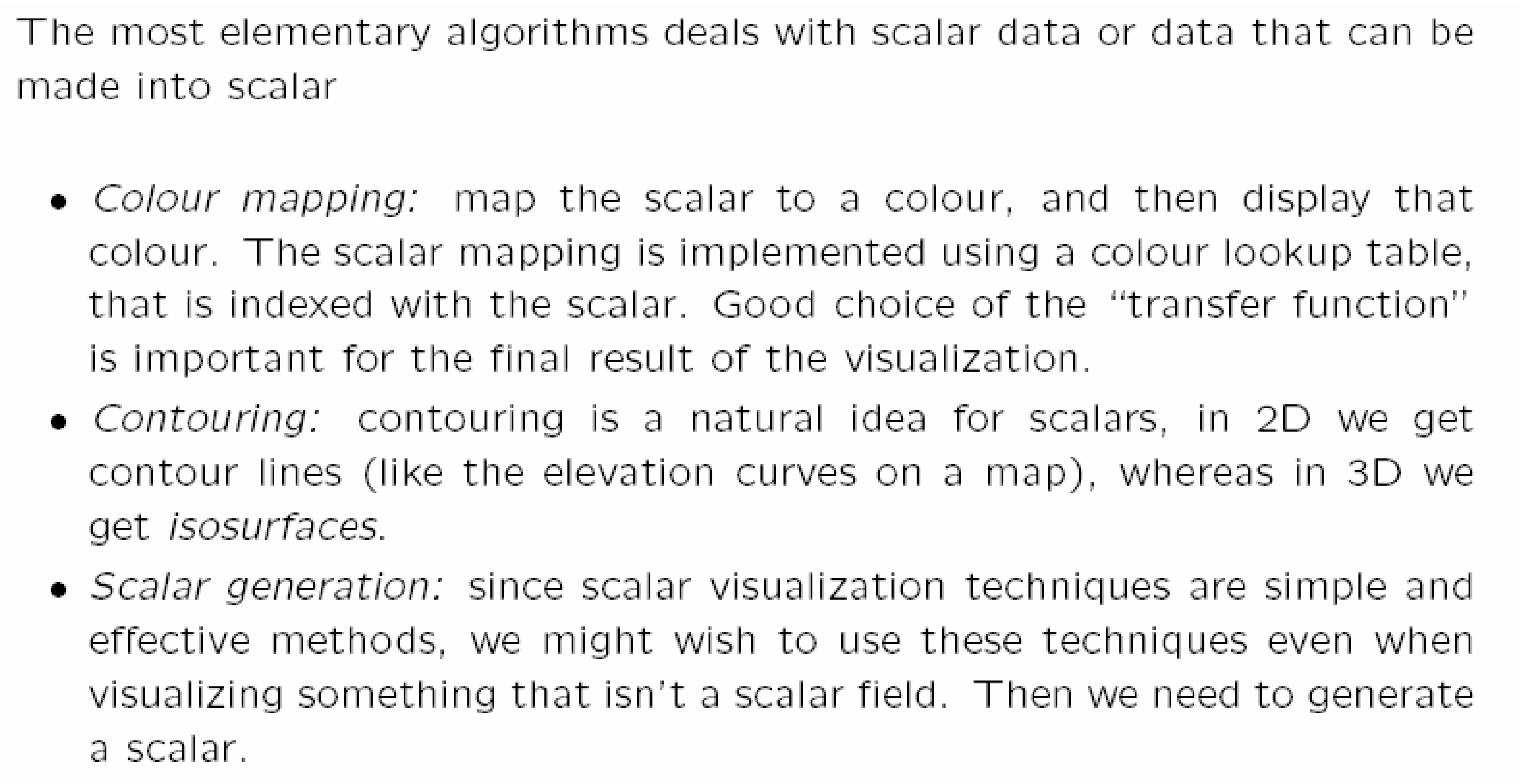
Še o algoritmih



The second part is about mapping the data to something that can be visualized (i.e. some 3D model)

- *Scalar algorithms* works on scalar data, for example generating contour lines or isosurfaces.
- *Vector algorithms* works on vector data. Showing oriented arrows is an example of vector visualization.
- *Tensor algorithms* works with tensors.
- *Modelling algorithms* generates dataset topology or geometry, or surface normals or texture data. For example, generating glyphs oriented according to the vector direction and then scaled according to the scalar value is a combined vector/scalar algorithm.

Skalarni algoritmi



The most elementary algorithms deals with scalar data or data that can be made into scalar

- *Colour mapping:* map the scalar to a colour, and then display that colour. The scalar mapping is implemented using a colour lookup table, that is indexed with the scalar. Good choice of the "transfer function" is important for the final result of the visualization.
- *Contouring:* contouring is a natural idea for scalars, in 2D we get contour lines (like the elevation curves on a map), whereas in 3D we get *isosurfaces*.
- *Scalar generation:* since scalar visualization techniques are simple and effective methods, we might wish to use these techniques even when visualizing something that isn't a scalar field. Then we need to generate a scalar.

Barvne preslikave in izo površine (color maps and isosurfaces)

Primeri barvnih preslikav

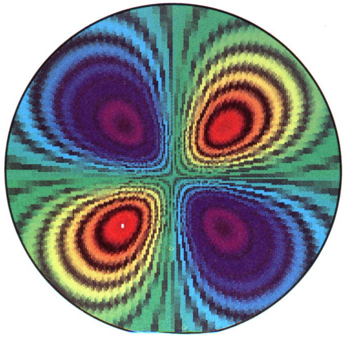


Figure 2.4a

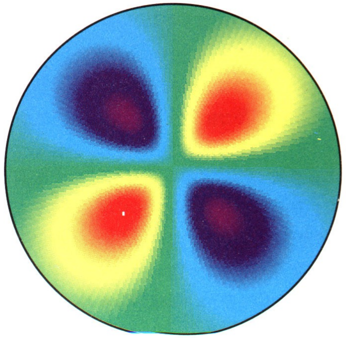


Figure 2.4b

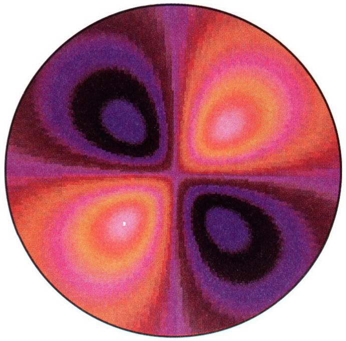


Figure 2.4c

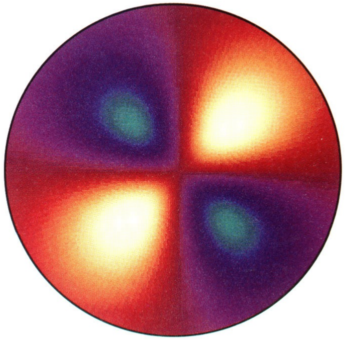


Figure 2.4d

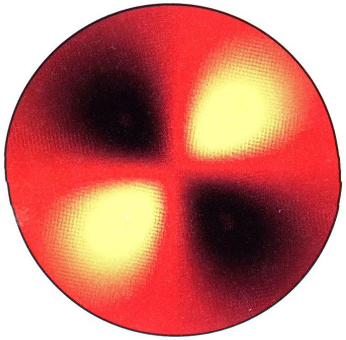


Figure 2.4e

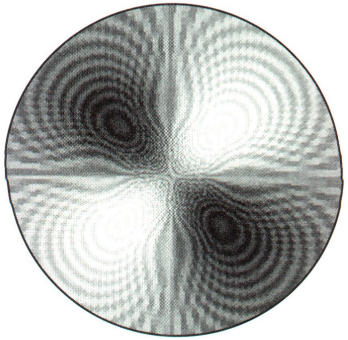
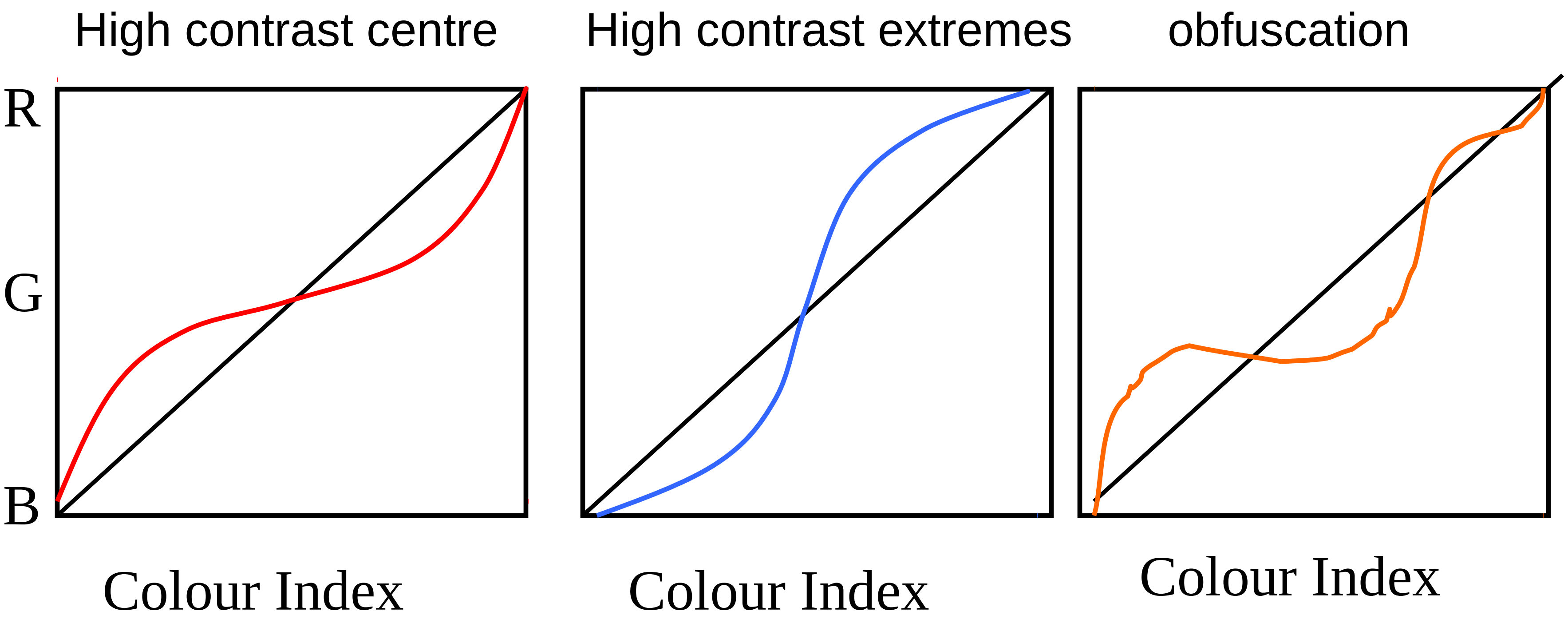


Figure 2.4f

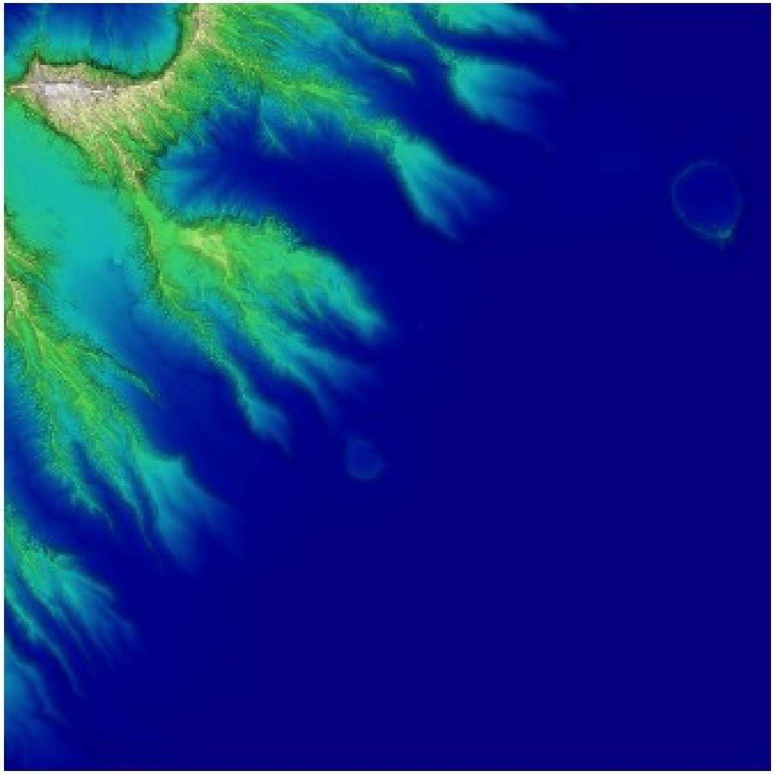
Barvne preslikave (Colour Maps)

Transfer function maps data/colour colours

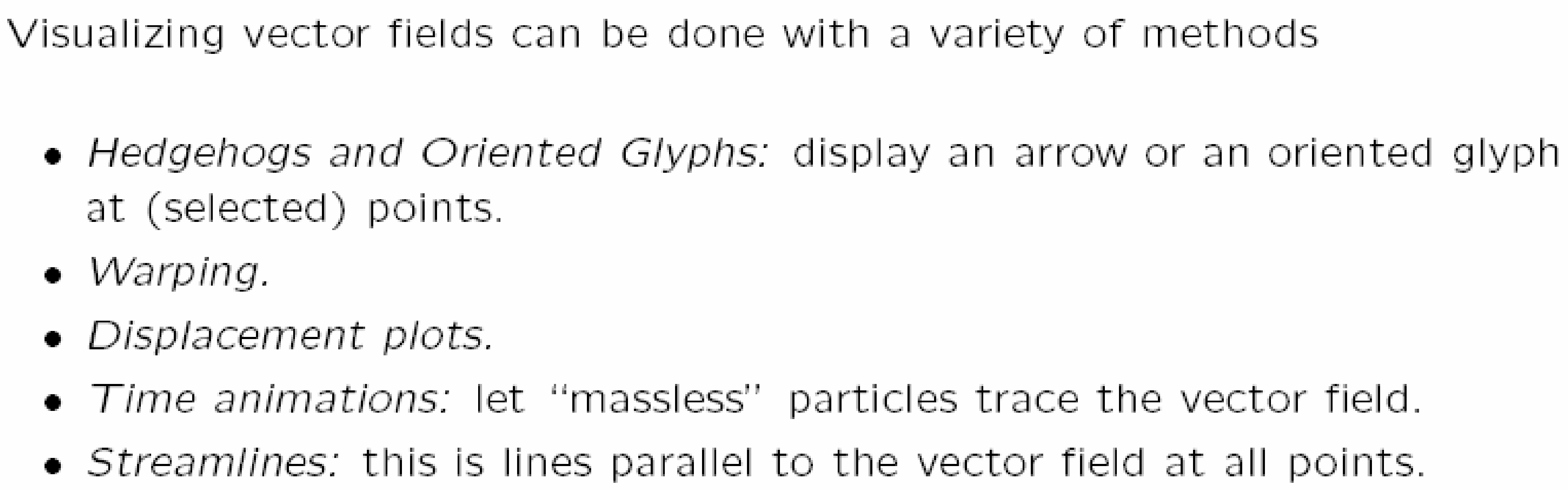
Non-linear transfer functions enhance ranges of data



Scalar Generation



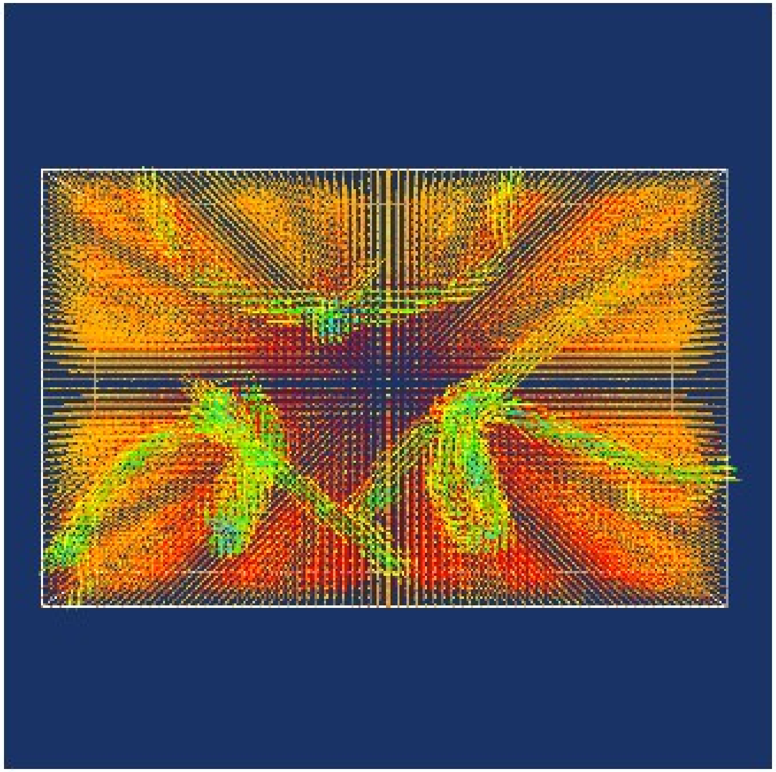
Vektorski algoritmi



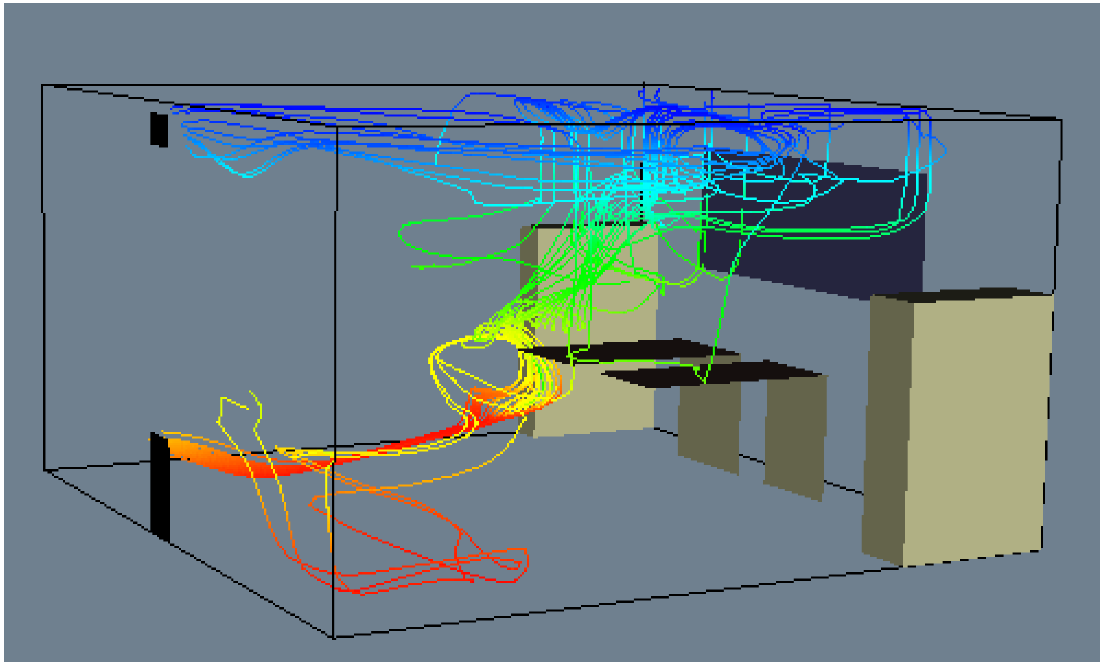
Visualizing vector fields can be done with a variety of methods

- *Hedgehogs and Oriented Glyphs:* display an arrow or an oriented glyph at (selected) points.
- *Warping.*
- *Displacement plots.*
- *Time animations:* let “massless” particles trace the vector field.
- *Streamlines:* this is lines parallel to the vector field at all points.

Hedgehogs and Oriented Glyphs



Streamlines



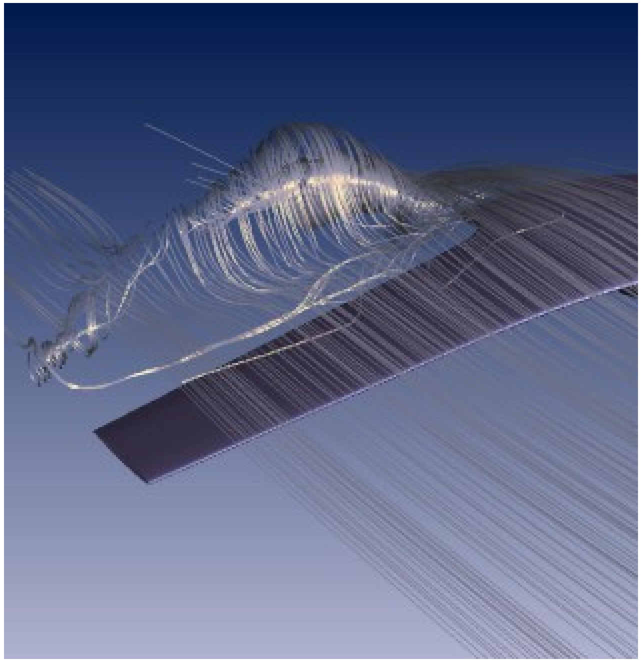
FEM and 3D data visualization

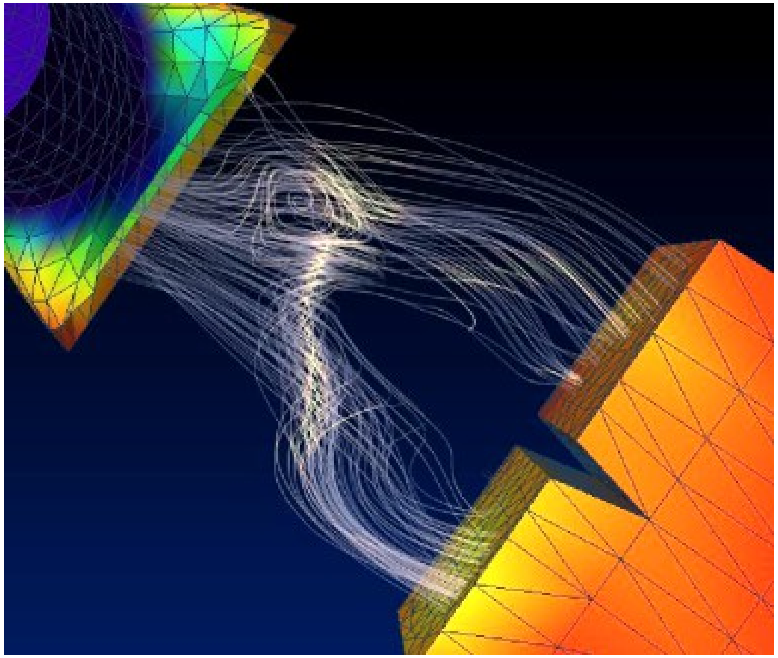
Visualized with Amira

(courtesy TGS)

Vector Fields:

- illuminated field lines

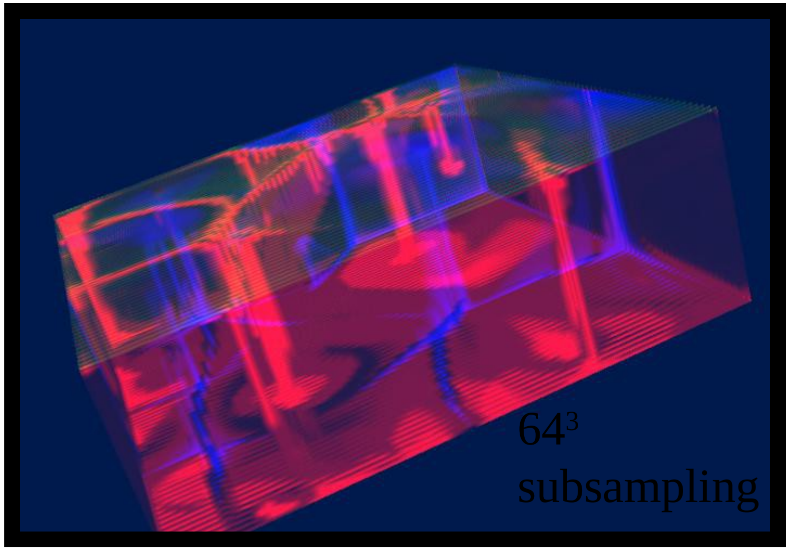




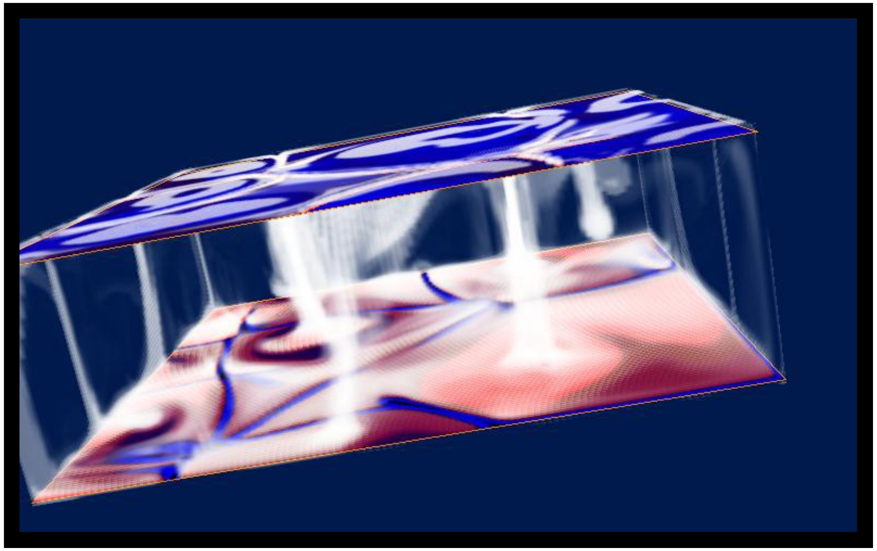
Heat Convection between Two Plates

257^3 dataset

Data, courtesy David Yeun



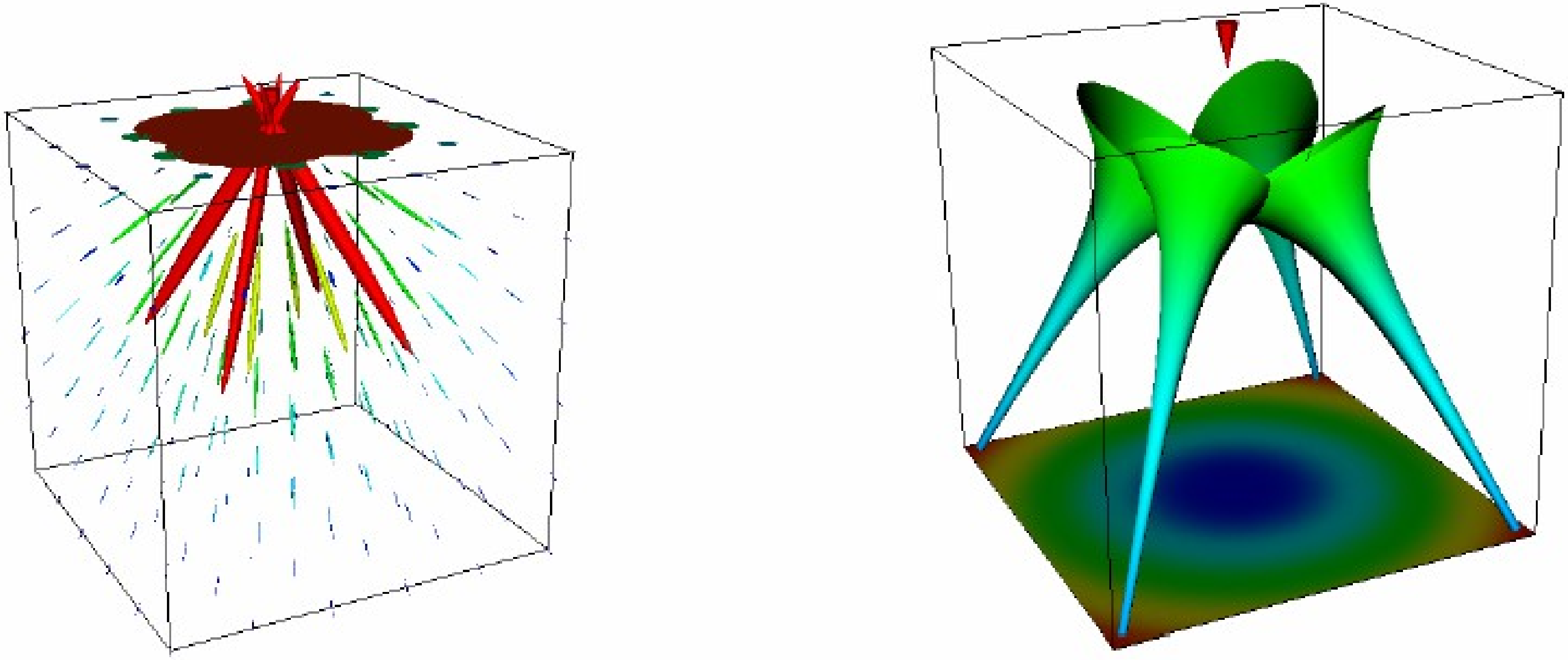
64^3 subsampling



Heat flow between two plates at constant temperature

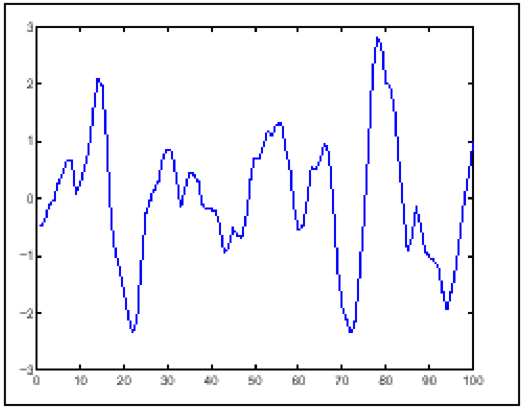
Tenzorski algoritmi

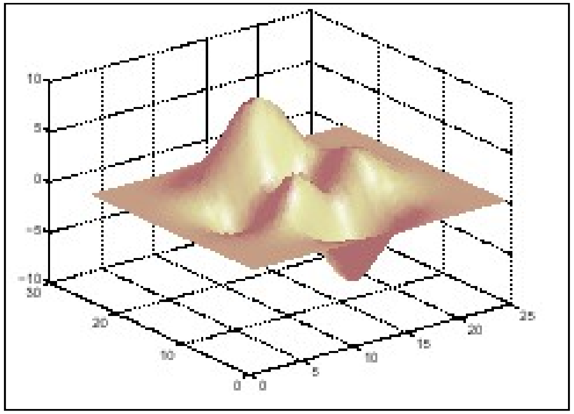
Below are two example tensor visualizations

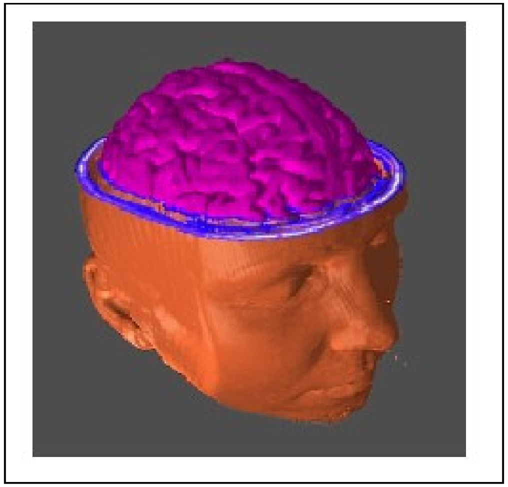


To the left *tensor ellipsoids*, to the right *Hyperstreamlines*

Dimenzije vizualizacije







$y = f(x)$

Graphs/Histograms

Charts

$z = f(x,y)$

Contours

False Colour

$f = f(x,y,z)$

Volume visualisation

Isosurfaces

Dimenzije atributov

- **Scalar**

  Temperature u(t), u(x,y,z)

  Price p(t)

  Height h(x,y)

- **Vector**

  eg. Velocity (u,v), (u,v,w)

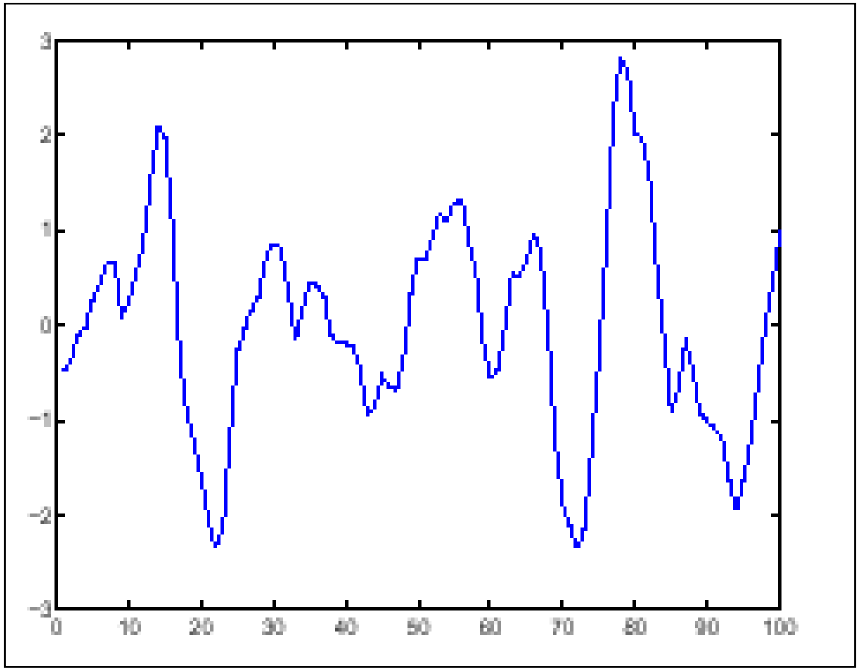
  Share Price and Volume p(t),V(t)

  Gradient of a Scalar $\nabla u(x,y) = (du/dx , du/dy)$

- **Tensor** eg. Stress and Strain

$$\begin{bmatrix} a11, & a12, & a13 \\ a21, & a22, & a23 \\ a31, & a32\ , & a33 \end{bmatrix}$$

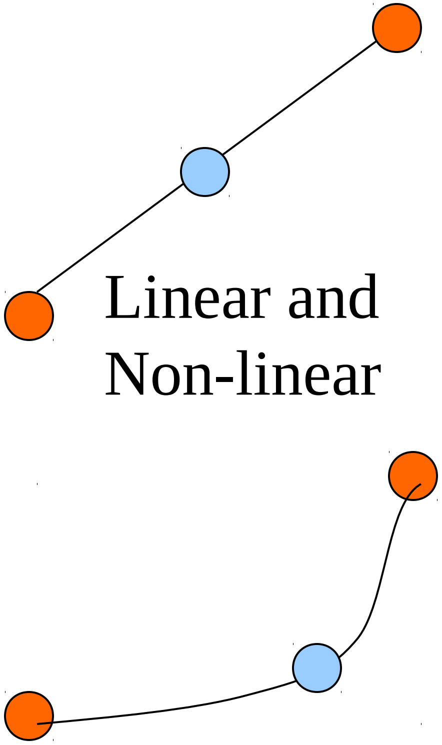
One Dimensional Visualisation



- •Ancient Technique
- •Plots, Graphs, Bar charts etc.
- •Log./Linear Scales

- •Join data points
- •Label Axes

Linear Graphing Techniques



Linear and
Non-linear

Interpolation

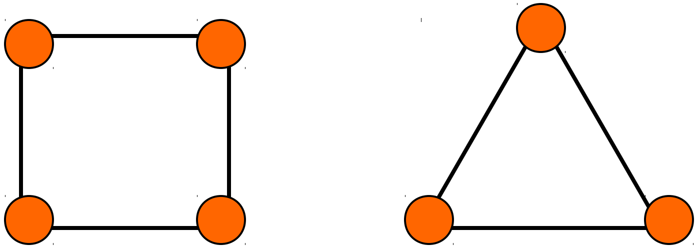
Linear - fast, simple

Non-Linear - eg polynormal, Spline

Used where clear trend exits

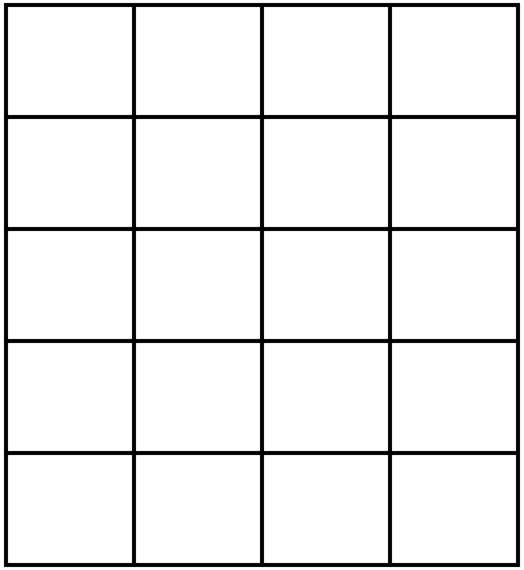
Can distort data

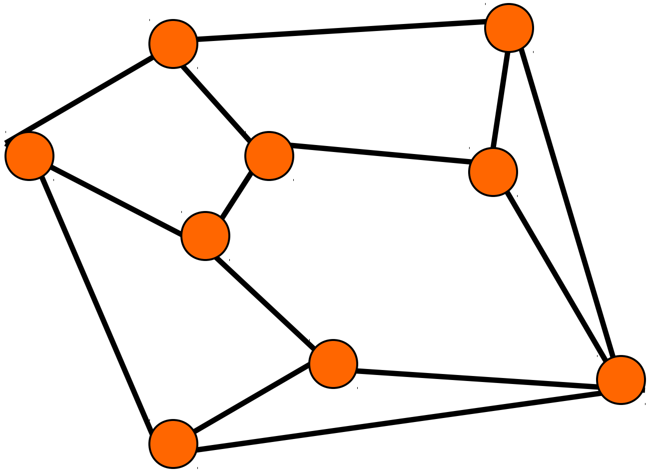
Two Dimensional Visualisation



Cells

Grids

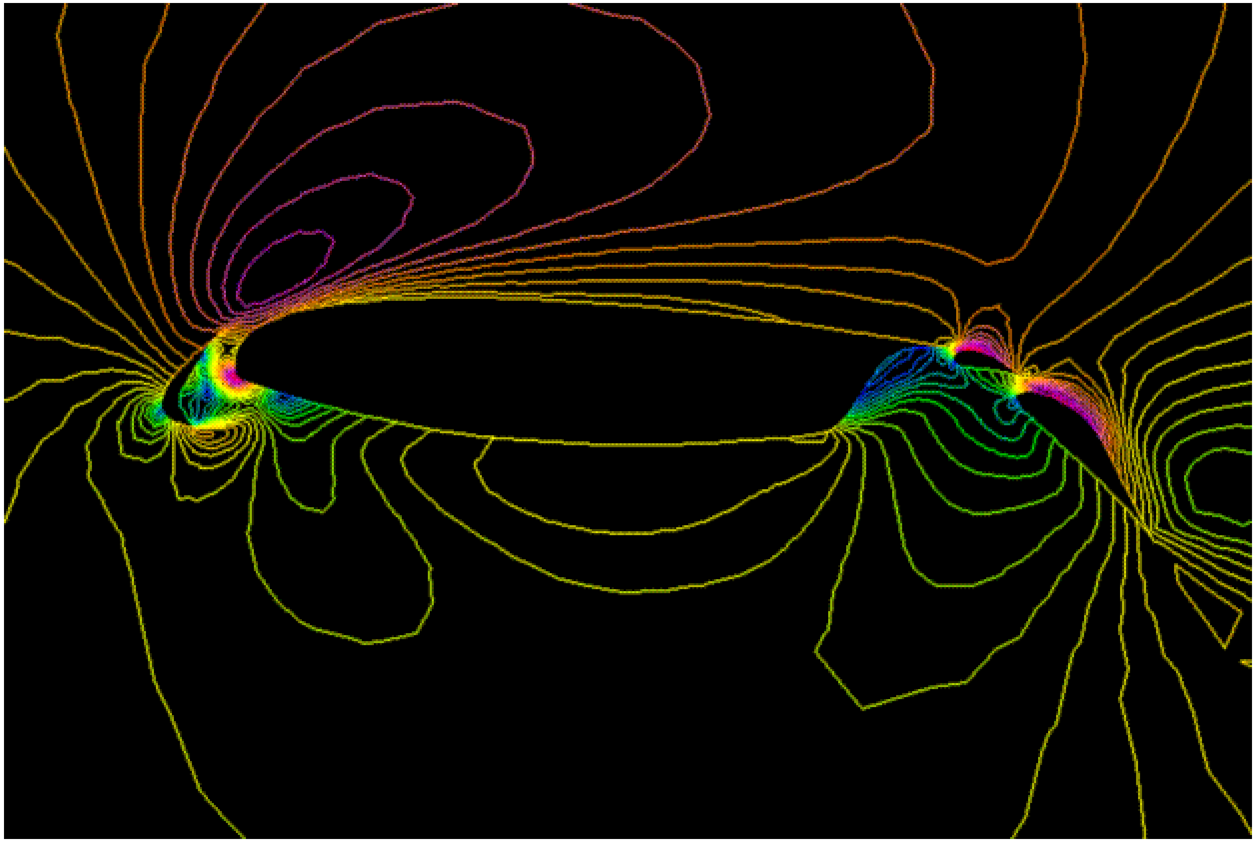




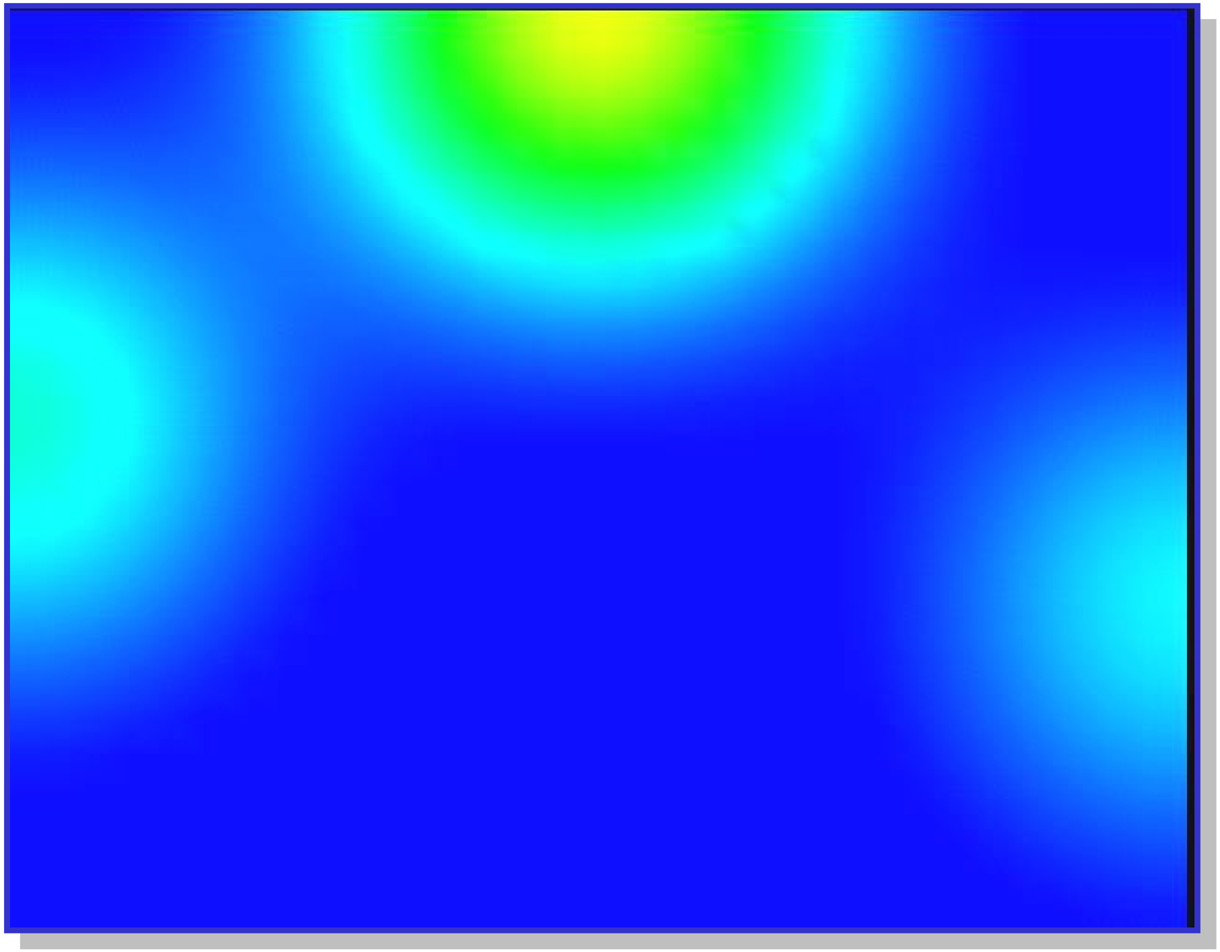
Unstructured

2D Visualizations

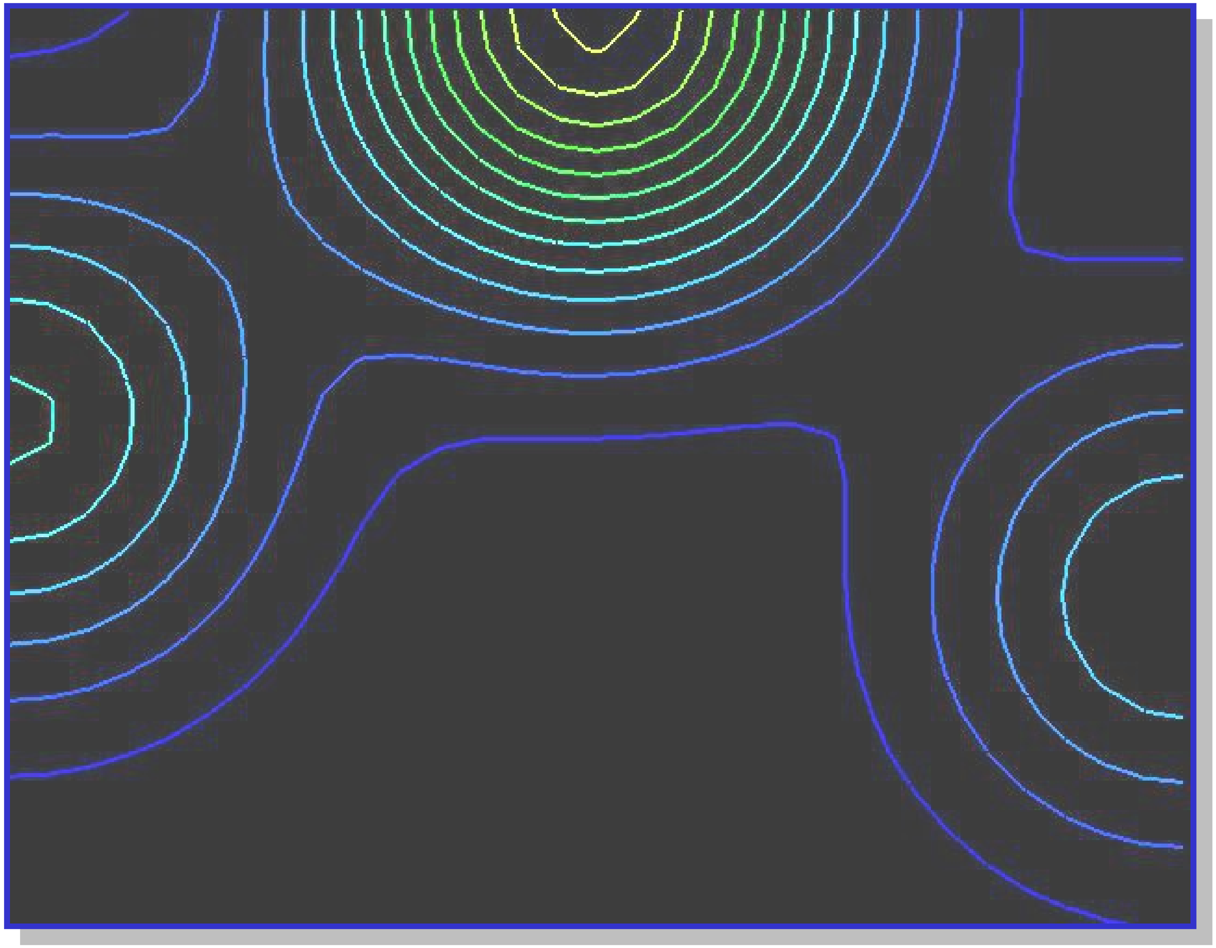
- Contour Lines - *f(x,y)* = *constant*



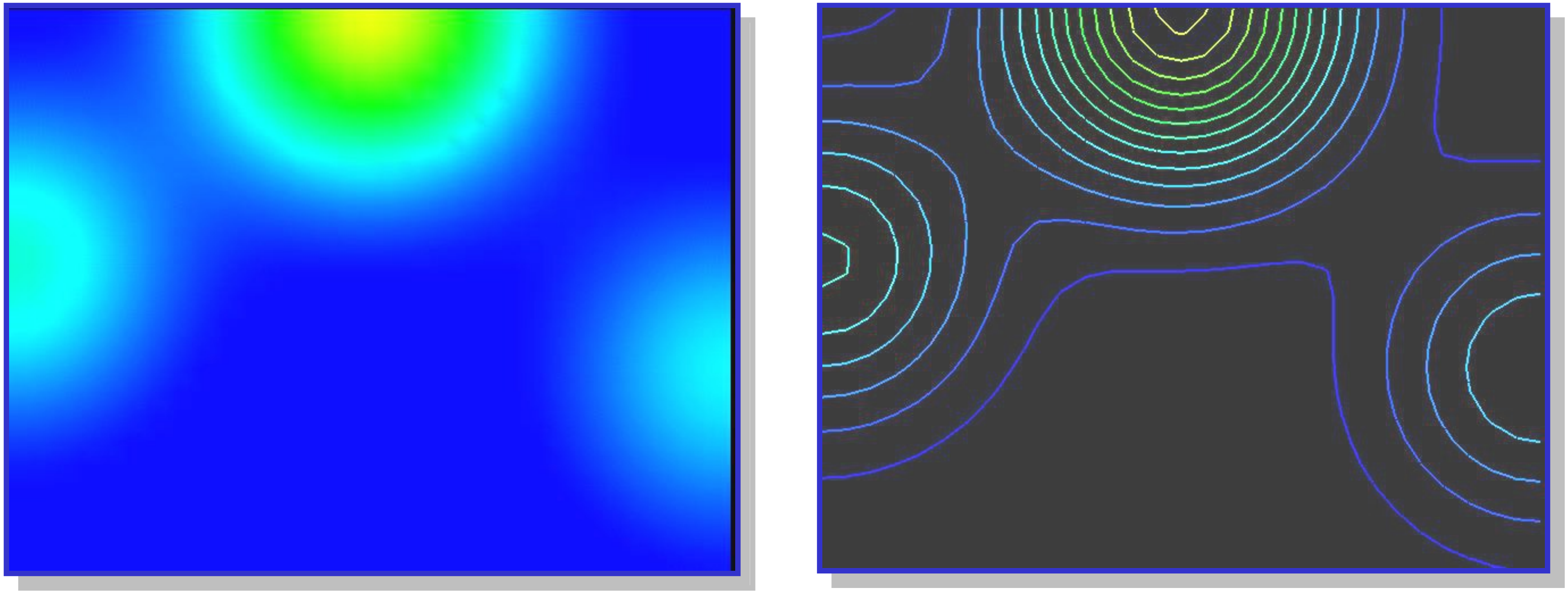
2D Color Interpolation



2D Contours

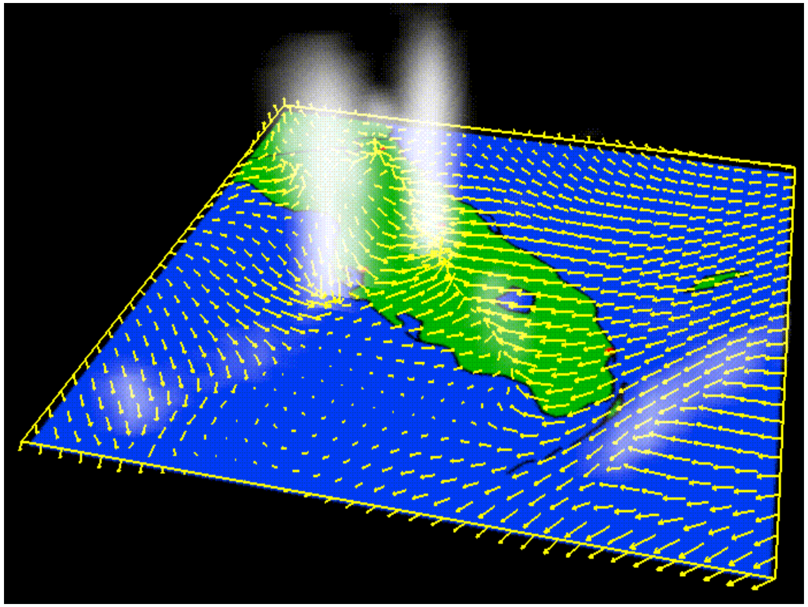


Same Data, Different Geometrization



2D Visualization

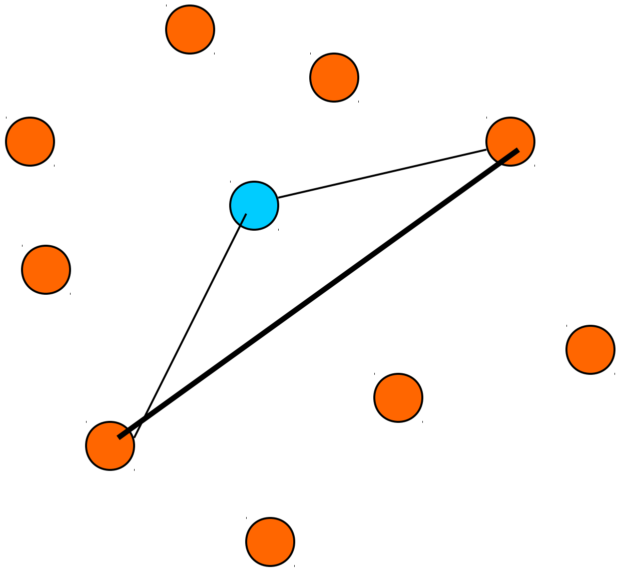
- Vector Fields
  - Hedgehogs
  - Streamlines



2D Data Techniques

Kriging

Technique for inferring points on a 2D plane



Data points are weighted according to their proximity to the interpolant

2D Data Techniques

Contouring

Data:Contours are lines of constant function value
Contour, path (x,y) so that f(x,y) = constant

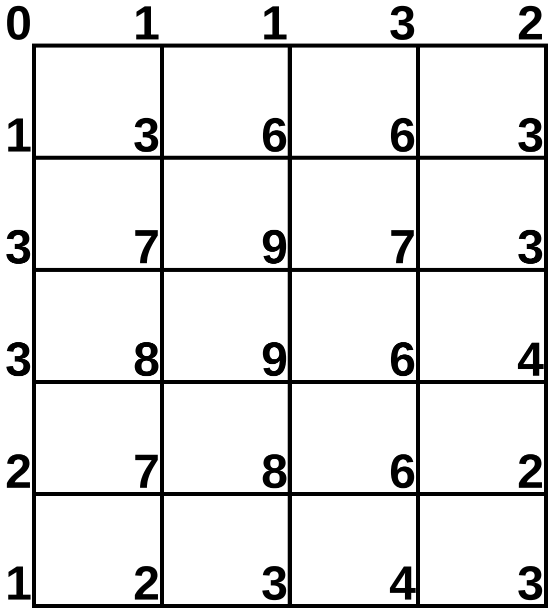
Examples:
c = 100m above sea level.
C is a constant intensity in MRI scan.
C is a constant ozone concentration.

2D Data Techniques

Contouring

Select height (e.g. c = 5)

- Scan cells until an edge crossing c is found
- Find location of crossing with inverse linear interpolation.
- Locate exit edge
- Mark checked edges
- move to next cell (exit cell becomes entrance edge)

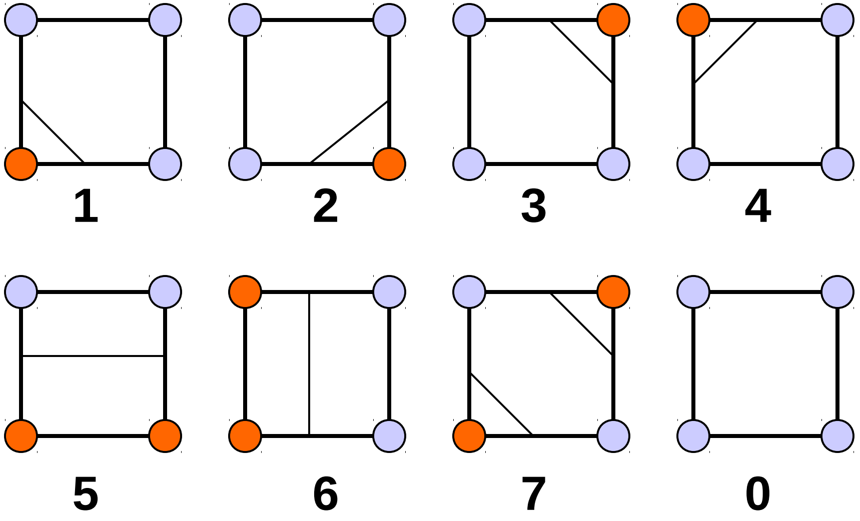


0	1	1	3	2
1	3	6	6	3
3	7	9	7	3
3	8	9	6	4
2	7	8	6	2
1	2	3	4	3

2D Data Techniques

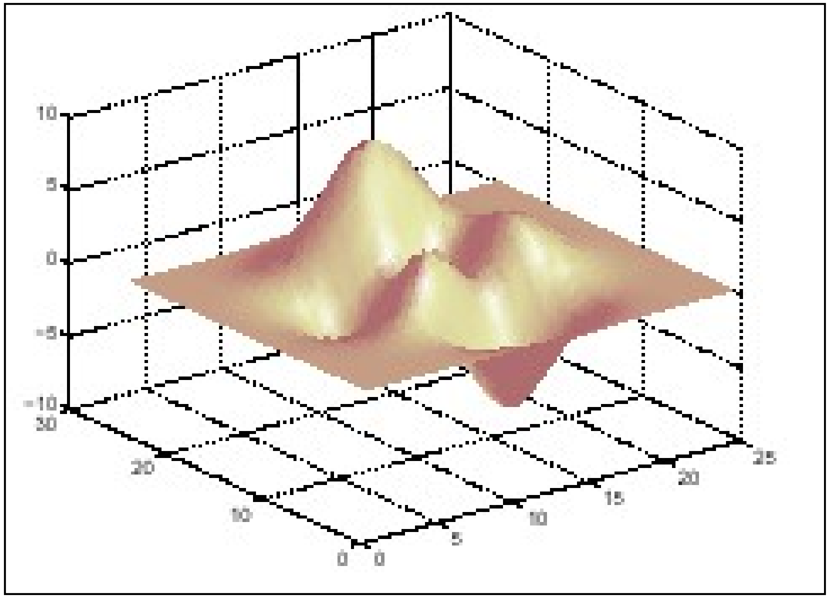
Contouring

Marching Squares



0	1	1	3	2
1	3	6	6	3
3	7	9	7	3
3	8	9	6	4
2	7	8	6	2
1	2	3	4	3

Surface Modelling



Each (triangular) defines a planar polygon.
Coordinates of vertices
x ; y ; z
Calculate polygon normal
Render with a lighting model
Colour may indicate height or be spurious!

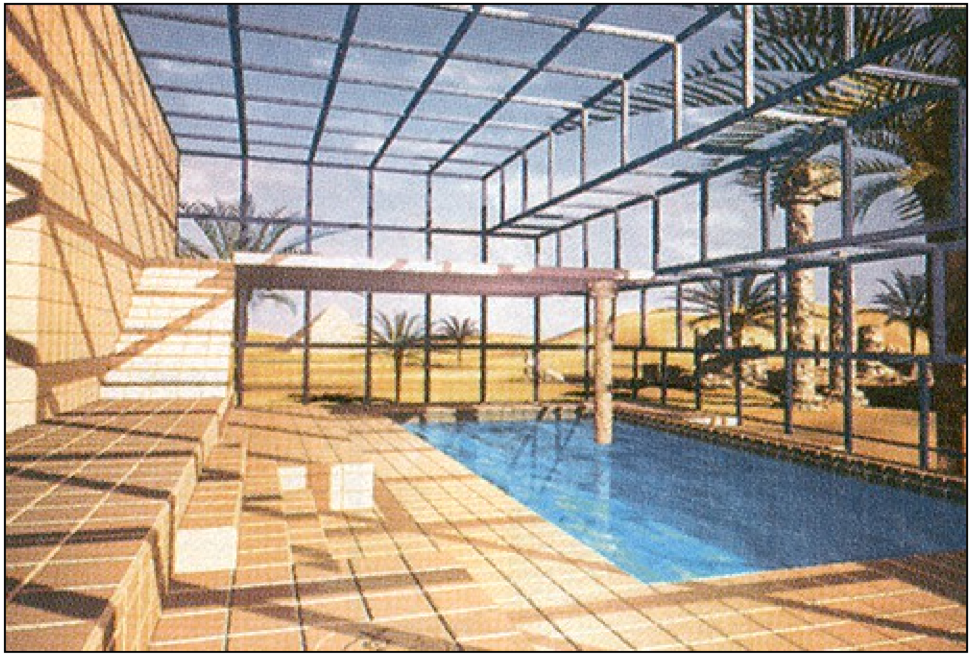
Looks pretty
Difficult to obtain quantitative information

Three Dimensional Visualisation

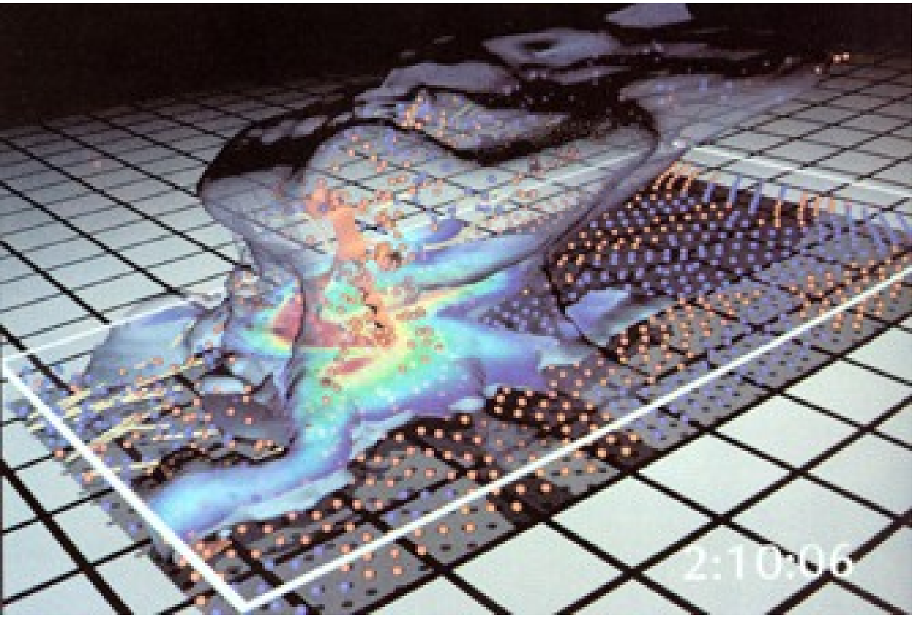
Used especially in Medical Applications

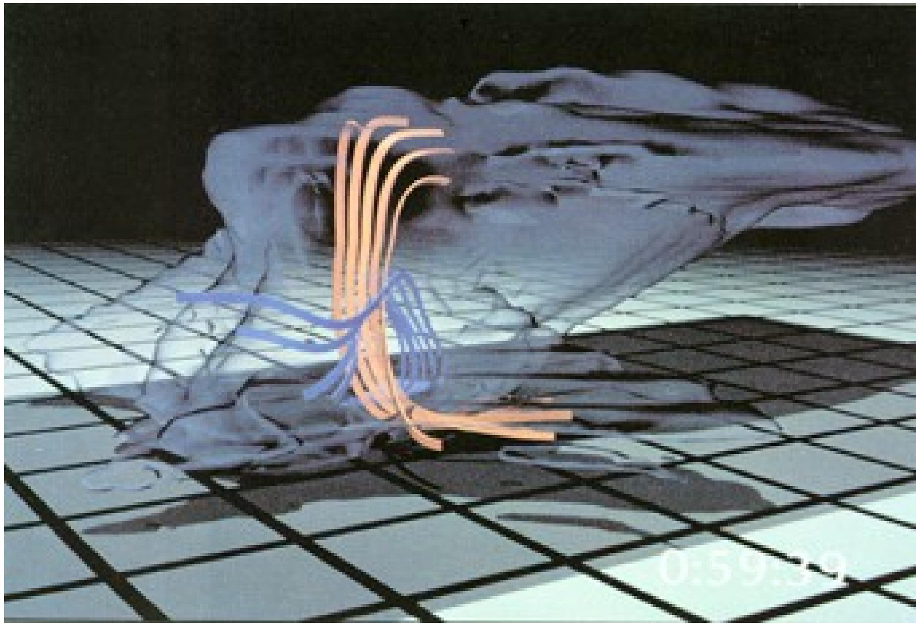
Simulations (eg flight training)

Architectural Walk-throughs

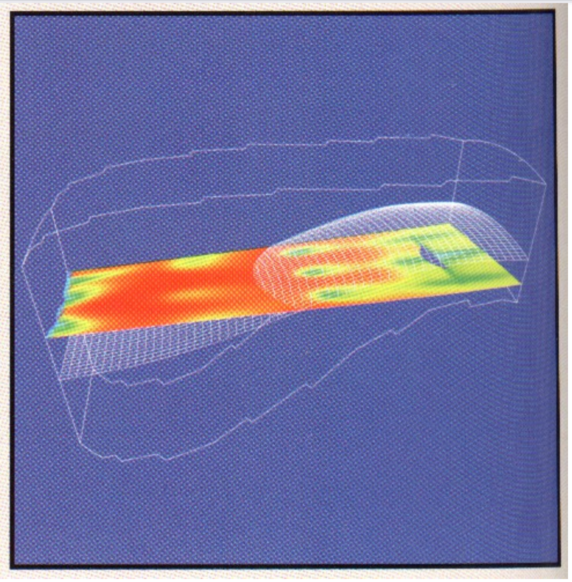


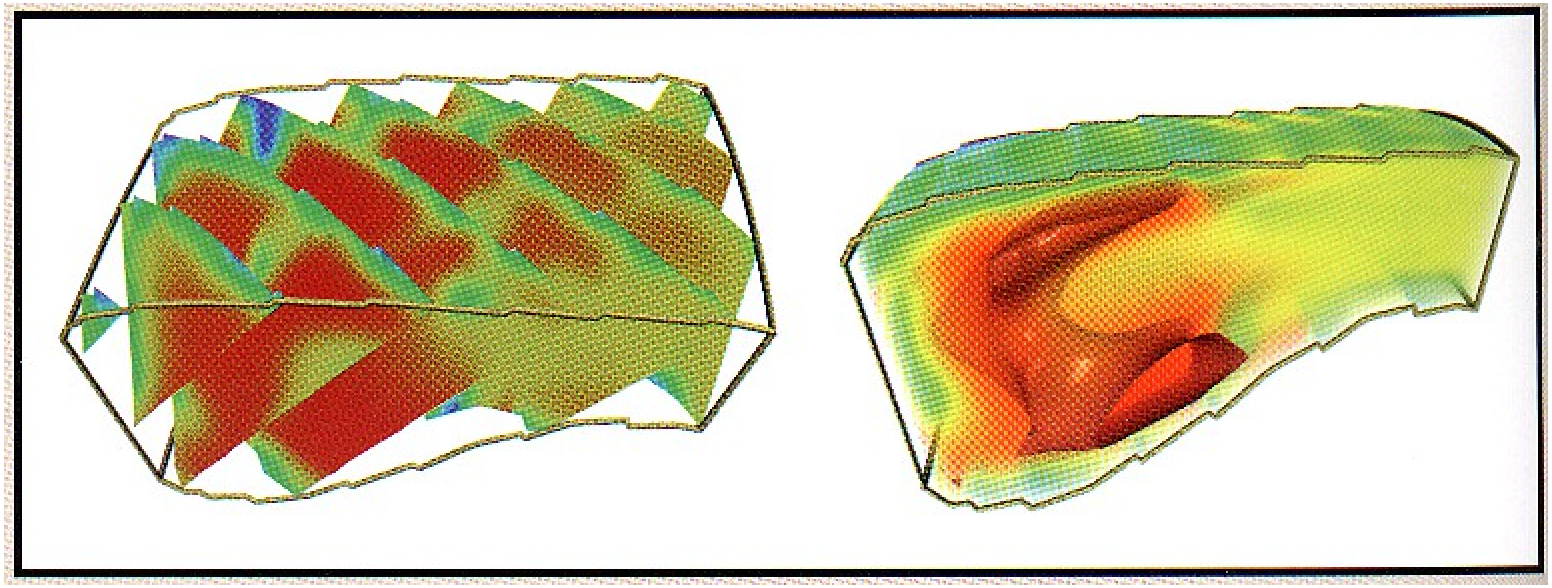
Example 3D Visualisation





Example 3D Visualisation





Zaključki

- Emerging Field of computing
- Needs high processing and graphics capabilities
- Wide ranging application
- No dominant standards and norms (exploratory area)
- Large design input
- Potential to obfuscate as well as enlighten
- Needs to reflect needs of end users (not technology for its own sake)

Zaključki

- Extract from large datasets more meaningful components (called *data extracts*)
  - Isosurface, streamlines, streaklines, vector field topology, vortex tubes, cracks, fault lines, etc.
  - Sedimentation layers, free-surfaces, edge and surface extraction
- Render this data with comprehension in mind, as opposed to visual realism

Zaključki

- Strike balance between
  - High- resolution versus interactive speed
- How to
- Time-dependent visualization
- Describe and view change of data topology
  - Vector and scalar fields
  - Tensor fields (i.e., rate of strain tensor)
- How to navigate a Terabyte dataset?