Local and Remote Visualisation Techniques

UvA High Performance Computing course

Robert Belleman, UvA/Ivl
Paul Melis, SURFsara
Program for today

09:00 – 09:15  Who’s who?
09:15 – 09:45  Introduction to Data Visualization (Robert Belleman)
09:45 – 10:45 Visualization at SURFsara and an Introduction to ParaView (Paul Melis)

10:45 – 11:45 Hands-on: Scientific Visualization with ParaView

11:45 – 12:15 Remote Visualization, the Collaboratorium (Paul Melis)

12:45 – 13:00 Closing
Introduction to Data Visualization

Robert G. Belleman, PhD
Informatics Institute
University of Amsterdam

Email: R.G.Belleman@uva.nl
• **Scientific Visualization and Virtual Reality** team
  - part of *Computational Science* at UvA/IvI
  - close collaboration with SURFsara
• Research theme: **interactive visual exploration**
  - Software solutions and architectures, Problem Solving Environments, Interactive graphics devices
• Application areas: computational science
  - (astro)physics, medicine, biology, finance, architecture, computer science, …
<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.00</td>
<td>8.04</td>
<td>10.00</td>
<td>9.14</td>
<td>10.00</td>
<td>7.46</td>
<td>8.00</td>
<td>6.58</td>
</tr>
<tr>
<td>8.00</td>
<td>6.95</td>
<td>8.00</td>
<td>8.14</td>
<td>8.00</td>
<td>6.77</td>
<td>8.00</td>
<td>5.76</td>
</tr>
<tr>
<td>13.00</td>
<td>7.58</td>
<td>13.00</td>
<td>8.74</td>
<td>13.00</td>
<td>12.74</td>
<td>8.00</td>
<td>7.71</td>
</tr>
<tr>
<td>9.00</td>
<td>8.81</td>
<td>9.00</td>
<td>8.77</td>
<td>9.00</td>
<td>7.11</td>
<td>8.00</td>
<td>8.84</td>
</tr>
<tr>
<td>11.00</td>
<td>8.33</td>
<td>11.00</td>
<td>9.26</td>
<td>11.00</td>
<td>7.81</td>
<td>8.00</td>
<td>8.47</td>
</tr>
<tr>
<td>14.00</td>
<td>9.96</td>
<td>14.00</td>
<td>8.10</td>
<td>14.00</td>
<td>8.84</td>
<td>8.00</td>
<td>7.04</td>
</tr>
<tr>
<td>6.00</td>
<td>7.24</td>
<td>6.00</td>
<td>6.13</td>
<td>6.00</td>
<td>6.08</td>
<td>8.00</td>
<td>5.25</td>
</tr>
<tr>
<td>4.00</td>
<td>4.26</td>
<td>4.00</td>
<td>3.10</td>
<td>4.00</td>
<td>5.39</td>
<td>19.00</td>
<td>12.50</td>
</tr>
<tr>
<td>12.00</td>
<td>10.84</td>
<td>12.00</td>
<td>9.11</td>
<td>12.00</td>
<td>8.15</td>
<td>8.00</td>
<td>5.56</td>
</tr>
<tr>
<td>7.00</td>
<td>4.82</td>
<td>7.00</td>
<td>7.26</td>
<td>7.00</td>
<td>6.42</td>
<td>8.00</td>
<td>7.91</td>
</tr>
<tr>
<td>5.00</td>
<td>5.68</td>
<td>5.00</td>
<td>4.74</td>
<td>5.00</td>
<td>5.73</td>
<td>8.00</td>
<td>6.89</td>
</tr>
</tbody>
</table>

\[ \mu_x = 9.00, \sigma_x = 3.32 \]
\[ \mu_y = 7.50, \sigma_y = 2.03 \]
linear regression: \( y = \frac{1}{2}x + 3 \)

Visualization taxonomy

- **Scientific visualization** ("scivis" or "datavis")
  - Data with an implicit or explicit geometric structure
    - Measurements, results from simulations or experiments

- **Information visualization** ("infovis" or "infographics")
  - Data with an abstract structure
    - Relations, graphs and networks

- **Visual analytics**
  - Interactive environments for the detection of the expected and discovery of the unexpected
Scientific visualization

“Scientific visualization deals with all aspects that are connected with the visual representation of data sets from scientific experiments or simulations to achieve a deeper understanding or a simpler representation of complex phenomena.”

Information visualization

- “In information visualization, the graphical models may represent abstract concepts and relationships that do not necessarily have a counterpart in the physical world.”

How much water is there on, in, and above the Earth?

http://ga.water.usgs.gov/edu/earthhowmuch.html
Information visualization

Charles Minard, 1869
Software: Tableau
Software: Spotfire
DIY Software: D3.js

Data-Driven Documents
The scientific visualization pipeline

Data source → Enrichment → Mapping → Render

files  raw data  derived data  visual object  image

Haber and McNabb reference model
Example: extracting a contour from medical data

DICOMReader → Contour → Triangulate → Render
Example: visualizing flow data with streamlines
Example:
visualizing flow data with streamlines

CSVReader → TableToPoints → streamtracer → tube → Render

TableToPoints → glyphs

pointdata → lines → polygons

image
Example: visualizing flow data with streamlines

CSVReader → TableToPoints → streamtracer → tube → Render

glyphs

outline

TableToPoints → pointdata → lines

 polygons → image
Pipeline creation

- Filters are connected together to form a “visualization pipeline” or “dataflow network”
- The input port of a filter may only be connected to the output port(s) of (an)other filter(s) if the port types are “similar"
Pipeline behaviour

- Filters in a pipeline only execute when necessary
  - When data at the input has changed, or a parameter changes
- Data flows downstream, update checks go upstream
Data model

- Data sets are represented by a *mesh* and *attributes*
Data model

- Data sets are represented by a *mesh* and *attributes*
- A mesh consists of interconnected vertices (topology)
Data model

- Data sets are represented by a *mesh* and *attributes*
- A mesh consists of interconnected vertices (topology)
- Collections of vertices form cells (regions, zones)
Data model

- Data sets are represented by a mesh and attributes
- A mesh consists of interconnected vertices (topology)
- Collections of vertices form cells (regions, zones)
- Vertices can have attributes
Data model

- Data sets are represented by a *mesh* and *attributes*
- A mesh consists of interconnected vertices (topology)
- Collections of vertices form cells (regions, zones)
- Vertices can have attributes
- Cells can have attributes
Cell types

Linear cell types

Non-linear cell types
Uniform Rectilinear Grid (image data)

Implicit topology and point coordinates, all cells of same type.

Properties:
- Extent: min/max indices
- Origin
- Spacing

Examples:
- Images (JPEG, PNG, TIFF, etc)
- (Bio-)medical data (CT, MRI, CLSM, etc)
Rectilinear grid

Implicit topology, semi-implicit point coordinates, all cells of same type.

Properties:
- Extent: min/max indices
- Vertex coordinates

Examples:
- Data structure in simulations with non-uniform density
Curvilinear Grid

Implicit topology, explicit point coordinates, all cells of same type.

Properties:
- Extent: min/max indices
- Vertex coordinates

Examples:
- Data structure in simulations with non-uniform density on non-rectangular domain
Adaptive Mesh Refinement (AMR)

Collection of non-uniform rectilinear grids (a.k.a. Berger-Oliger mesh).

Examples:
- Data structure in simulations with irregular non-uniform density
Unstructured grid

Explicit topology, explicit point coordinates, all possible cell types.

Examples:
- Finite Element/Volume Models
- CAD/CAM
Polygon data

Explicit topology, explicit point coordinates, restricted cell types (vertices, lines, polygons).

Examples:
- Game models (OBJ, STL, PLY)
- Molecule models (PDB)
Software

- Often domain-specific
- Almost all based on visualization pipeline / dataflow concept

Commercial:
- AVS (Advanced Visual Systems)
- IRIS Explorer
- Amira
- Matlab, Mathematica, IDL
- Spotfire
- …

Public domain:
- VTK
- ParaView
- VolView
- VisIt
- DeVIDE
- MeVisLab
- SCIRun
- Gephi
- Cytoscape
- R
- …
The Visualization Toolkit (VTK)

VTK is:

- open source visualization library
  - C++ library with > 1500 classes
  - Language “bindings” to Java, Python, Tcl, Ruby
- works on Unix/Linux, Windows, MacOS
- object-oriented design

VTK provides:

- Visualization methods to turn data into geometry
- Graphics model to turn geometry into images (OpenGL)
- Image processing methods
The Visualization Toolkit (VTK)

VTK is *not*:
- VTK is *not* a programming language
- VTK is *not* an application
  - No drag-and-drop “visual program editor” as with AVS, Iris Explorer, OpenDX, etc.
  - You have to *program*

More info:
- [http://www.vtk.org/](http://www.vtk.org/)
Robert G. Belleman, PhD
Informatics Institute
Universiteit van Amsterdam

Science Park 904
1098 XH Amsterdam

Email: R.G.Belleman@uva.nl
DIY Software: D3.js

Data-Driven Documents
D3.js tutorial

- [http://bost.ocks.org/mike/d3/workshop/#0](http://bost.ocks.org/mike/d3/workshop/#0)
Processing

```java
// Brownian motion.
// Recording random movement as a continuous line.

int num = 2000;
int range = 6;

float[] ax = new float[num];
float[] ay = new float[num];

void setup()
{
    size(200, 200);
    for(int i = 0; i < num; i++)
    {
        ax[i] = width/2;
        ay[i] = height/2;
    }
    frameRate(30);
}

void draw()
{
    background(0);
    noStroke();
    for(int i = 0; i < num; i++)
    {
        line(ax[i], ay[i], ax[i+1], ay[i+1]);
    }
}
```