CSE 564
Visualization and Visual Analytics

Visualization of Hierarchies

Klaus Mueller

Computer Science Department
Stony Brook University
<table>
<thead>
<tr>
<th>Lecture</th>
<th>Topic</th>
<th>Projects</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Intro, schedule, and logistics</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Applications of visual analytics, basic tasks, data types</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Introduction to D3, basic vis techniques for non-spatial data</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Data assimilation and preparation</td>
<td>Project #1 out</td>
</tr>
<tr>
<td>5</td>
<td>Data assimilation and preparation</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Bias in visualization</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Data reduction and dimension reduction</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Visual perception</td>
<td>Project #2(a) out</td>
</tr>
<tr>
<td>9</td>
<td>Visual cognition</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Visual design and aesthetics</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster analysis: numerical data</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Cluster analysis: categorical data</td>
<td>Project #2(b) out</td>
</tr>
<tr>
<td>13</td>
<td>High-dimensional data visualization</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Dimensionality reduction and embedding methods</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Principles of interaction</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Midterm #1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Visual analytics</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>The visual sense making process</td>
<td>Final project proposal call out</td>
</tr>
<tr>
<td>19</td>
<td>Maps</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Visualization of hierarchies</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Visualization of time-varying and time-series data</td>
<td>Final project proposal due</td>
</tr>
<tr>
<td>22</td>
<td>Foundations of scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Volume rendering</td>
<td>Project 3 out</td>
</tr>
<tr>
<td>24</td>
<td>Scientific and medical visualization</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Visual analytics system design and evaluation</td>
<td>Final Project preliminary report due</td>
</tr>
<tr>
<td>26</td>
<td>Memorable visualization and embellishments</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Infographics design</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Midterm #2</td>
<td></td>
</tr>
</tbody>
</table>
Hierarchies = Trees
Mapping publications to a tree

- major leaves are papers
- minor leaves are co-authors
- height is time
- fruit are comments
- size or color is number of paper’s citations
- journal papers on right side
- conference papers left side
PRODUCTIVE VS. UNPRODUCTIVE RESEARCHERS

Productive

Unproductive
Visualizing hard drives with tree cartoons

Kleiber et al., IEEE InfoVis 2001
Color maps to file type
- blue are pdf files, red are image files
Standard Node-Edge layout for a hierarchical network

- 3 levels
- color maps to quantitative information (here population)
DENDROGRAM

Typically used to depict classification hierarchies

- split-off points visualize proximity
BIRDS AND DINOSAURS
CIRCLES ARE MORE SPACE-EFFICIENT
Chord Diagrams

Represents flows or connections between several entities

- for example the number of people migrating from one country to another
Can we make it easier to read?

- yes
- via edge bundling
- see soon
Hierarchy of the Flare ActionScript visualization library

- elements are organized in several folders, such as query, data, scale...
- each folder is then subdivided in subfolders and so on.
- can be visualized as a radial dendrogram
Hierarchical Chord Diagram

Visualize dependencies in the library

bad: straight line
better: follow a hierarchical edge bundling line
Apply the bundling to every adjacency connection of the dataset

- show the hierarchy of the dataset
- decrease the clutter as much as possible
- bundling the electrical wires together in order to reduce clutter
- and fan them out at their terminus in order to connect them to the terminals
RADIAL PLOTS AND EDGE BUNDLES

Original Graph
Edges are represented by splines with tension $\beta$

Setting $\beta$
- low values mainly provide low-level, node-to-node connectivity information
- high values provide high-level information

Holten, IEEE TVCG 2006
Software system call graph

- green is caller, red is callee

balloon layout (isolated processes)  radial layout (more integrated)
Without Edge Bundling

balloon layout

radial layout
Curved edges are represented as *splines*
- a spline is a smooth curve defined by some control points
- moving the control points changes the curve

- **Interpolation**
- **Bézier (approximation)**
- **BSpline (approximation)**
A B-Spline curve is defined as follows:

- $n$ is the total number of control points
- $d$ is the order of the curves, $2 \leq d \leq n+1$, $d$ typically 3 or 4
- $B_{k,d}$ are the uniform B-spline blending functions of degree $d-1$
- $P_k$ are the control points
- Each $B_{k,d}$ is only non-zero for a small range of $t$ values, so the curve has local control

Or in matrix form:
- $t$ is the parametric variable
- defined on $[0,1]$

$$X(t) = \sum_{k=0}^{n} P_k B_{k,d}(t)$$

$$x(t) = \frac{1}{6} \begin{bmatrix} P_0 & P_1 & P_2 & P_3 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} t^3 \\ t^2 \\ t \\ 1 \end{bmatrix}$$
Four basis functions $B$ must be active to define the B-Spline curve.
The locations of the control points scale the basis functions

- in this simple example we see a continuous 1D function generated from 6 control points and basis functions

\[ X(t) = \sum_{k=0}^{n} P_k B_{k,4}(t) \]

The curve can’t start until there are 4 basis functions active
Cubic B-Spline Animated

![Cubic B-Spline Diagram](image)
EXTENSION TO SURFACES

B-spline surface
One straightforward way of reducing clutter is to replace polylines with polycurves:

Each line segment is replaced with an end-point interpolating, quadratic B-spline. A tension parameter can be controlled by the user.

Let $m$ be the mid-point in viewport coordinates of $v_{i,j}$ and $v_{i+1,j}$, end-points of a line segment.

Let $c_k$ be the cluster to which this segment belongs and $c_{k,\mu}$ be its mid-point in viewport coordinates.

Let $\lambda$ and $\beta$ be tension parameters (usually $\lambda = 0.75$) and $0 \leq \beta \leq 1$ is set by the user.

The control points of the spline are given by:

- $(-1, v_{i,j})$
- $(0, \beta m + (1 - \beta)p)$
- $(1, v_{i+1,j})$
The tension can be changed to control the amount of clutter reduction.

In our implementation, the $\lambda$ parameter is fixed, but the $\beta$ parameter can be changed in the GUI.

Examples of medium and low tension, respectively:
Recall that clusters are often rendered as heavy line segments on top of the dataset.
In IPC we render the clusters as polygonal meshes. They help to show the ranges of each cluster along axes. The vertical “spread” can be controlled by the user.
Draw curves at different opacities

- long curves: low opacities (high transparencies)
- short curves: high opacity (makes short curves visible)
Alpha (Opacity) Blending

Alpha blending also enables visualization of sub-bundles and differentiation of lines.
Another bundling technique

- flow diagram
- the width of the arrows is proportional to the flow rate

Use cases:

- where money came from and went to (budgets, contributions)
- flows of energy from source to destination
- flows of goods from place to place
Hierarchies with Sun Burst Displays
SAME DATA WITH TREEMAP
Treemap Construction
Size is mapped to market cap, yellow boxes are investor’s holdings.
Advantages

- due to perceived discontinuity in texture between nodes, lines are no longer necessary to separate nodes
- more of the space can be used for the actual node display
- much smaller nodes can be shown than in a flat treemap
Used in programs like

- WinDirStat (Windows)
- KDirStat (Linux)
- DiskInventory (Mac)
Squarified treemap is preferred
- it’s difficult to visually compare long slivery tiles with tiles that have a more even aspect ratio
- a squarified treemap makes the map more globally comparable

Voronoi treemap
- based on Voronoi tesselation
Constructing a Squarified Tree map

Optimization criterion

- keep aspect ratio of boxes close to 1

Sequence:

- steps 1, 2, 4, 5, 7, 9, 10
- steps 3, 6, 8 would increase the aspect ratios of the boxes
- start a new row

Can greatly improve

- ability to compare the magnitude of different leaf nodes
- at the same time maintain some level of the original hierarchy