<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>Project #1 out</td>
</tr>
<tr>
<td>4</td>
<td>Data assimilation and preparation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Data reduction and notion of similarity and distance</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Visual perception and cognition</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Visual design and aesthetics</td>
<td>Project #1 due</td>
</tr>
<tr>
<td>8</td>
<td>Dimension reduction</td>
<td>Project #2 out</td>
</tr>
<tr>
<td>9</td>
<td>Data mining techniques: clusters, text, patterns</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Cluster analysis: numerical data</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Cluster analysis: categorical data</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Spatial data origins: medical imaging, scientific simulation</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Techniques to visualize spatial data: volume visualization</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Intro to GPU programming</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Techniques to visualize spatial data: flow visualization</td>
<td>Project #3 out</td>
</tr>
<tr>
<td>16</td>
<td>Midterm #1</td>
<td>Project #2 due</td>
</tr>
<tr>
<td>17</td>
<td>Illustrative rendering</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>High-dimensional data</td>
<td>Project #3 due</td>
</tr>
<tr>
<td>19</td>
<td>Hierarchies</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Principles of interaction</td>
<td>Final project proposal due</td>
</tr>
<tr>
<td>21</td>
<td>Visual analytics and the visual sense making process</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Evaluation and user studies</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Visualization of time-varying, time-series, streaming data</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Visualization of graph data</td>
<td>Final Project preliminary report due</td>
</tr>
<tr>
<td>25</td>
<td>Visualization of text data</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Midterm #2</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Data journalism</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Final project presentations</td>
<td>Final Project slides and final report due</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

Fung and Ma, Personal VIS 2015
PRODUCTIVE VS. UNPRODUCTIVE RESEARCHERS

Productive

Unproductive
Visualizing hard drives with tree cartoons

one file  many files

Kleiberg et al., IEEE InfoVis 2001
BOTANICAL–INSPIRED VISUALIZATIONS

Color maps to file type

- blue are pdf files, red are image files
Conventional

Standard Node-Edge layout for a hierarchical network

- 3 levels
- color maps to quantitative information (here population)
Typically used to depict classification hierarchies

- split-off points visualize proximity
BIRDS AND DINOSAURS

SAURISCHIANS
- Sauropods
- Theropods
   - Carnosaurs
   - Tyrannosauroids
- Compsognathids
- Therizinosaurs
- Alvarezsaurids
- Oviraptorosaurs
- Troodontids
- Dromaeosaurs
- Scanoraptoropterygids

FOSSIL FILAMENTS AND FEATHERS
- Dinosaur fossils reveal nine featherlike forms, as well as bumps similar to the feather supporting quill knobs of living birds. Only the four feather types in the right column are seen in living birds.

- No skin or feathers found
- Filaments branching from a central filament
- Filaments attached at the base
- Filaments arising from a membrane
- Simple filaments
- Simple broad filaments
- Filaments joined at the base
- Filaments joined at the base or a central filament
- Filaments joined at the base or a central filament
- Filaments branched
- Filaments branched
- Filaments branched
- Filaments branched
- Simple feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers
- Simple broad feathers

Aves (birds)
- Archaeopteryx
- Jeholornis
- Confuciusornis
- Neornithines (modern birds)
Circles Are More Space-Efficient
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
Smooth curve defined by some control points
Moving the control points changes the curve
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

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

Or in matrix form:

- $t$ is the *parametric variable*
- defined on $[0,1]$
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.
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:
Cluster Rendering

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.
Alpha (Opacity) Blending

Draw curves at different opacities

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

alpha blending disabled            alpha blending enabled
Alpha blending also enables visualization of sub-bundles and differentiation of lines

alpha blending disabled  alpha blending enabled
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
HIERARCHIES WITH SUN BURST DISPLAYS
Sunburst with Partition of Unity
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
Tree map for Disk Drives

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