Data Visualization (CIS 468)

Multiple Views

Dr. David Koop
Multiple Views

[Improvise, Weaver, 2004]
Multiple Views

- Why have just one visualization?
- Sometimes data is best examined in more than one view
  - Clutter/visual overload
  - Different attributes (cannot show all attributes in one view)
  - Different scales (task requires overview or detail)
  - Different encodings (no single encoding is optimal for all tasks)
- Eyes Beat Memory (Ch. 6)
  - Aiding working memory:
    side-by-side/layers > animated > jump cuts
  - Showing all visual elements at once → don't need to remember
Summary of Choices (Scatterplot + Bar Chart)

(a) Juxtaposed views.
(b) Integrated views.
(c) Superimposed views.
(d) Overloaded views.
(e) Nested views.

[W. Javed and N. Elmqvist, 2012]
Juxtaposition

[ComVis, K. Matkovic et al., 2008]
Integration

"best statistical graphic ever"

[later known as a Sankey Diagram]

[Napoleon's March to Moscow, C. J. Minard, 1869]
Multiple Views

- **Juxtapose and Coordinate Multiple Side-by-Side Views**
  - **Share Encoding: Same/Different**
    - *Linked Highlighting*

- **Share Data: All/Subset/None**

- **Share Navigation**

[Munzner (ill. Maguire), 2014]
## Multiple Views

The figure below illustrates the combination of data and encoding to create different views. The table shows the possible combinations:

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Data</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Redundant <a href="image1.png"><img src="image1.png" alt="Graph" /></a></td>
</tr>
<tr>
<td></td>
<td>Subset</td>
<td>Overview/Detail <a href="image2.png"><img src="image2.png" alt="Graph" /></a></td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>Small Multiples <a href="image3.png"><img src="image3.png" alt="Graph" /></a></td>
</tr>
</tbody>
</table>

- **Same**: Redundant
- **Different**: Multiform

[![Image1](image1.png)](image1.png) [![Image2](image2.png)](image2.png) [![Image3](image3.png)](image3.png)

[Munzner (ill. Maguire), 2014]
Multiform

[Improvise, Weaver, 2004]
Small Multiples

• Same encoding, but different data in each view (e.g. SPLOM)
Multiple Views

Partition into Side-by-Side Views

Superimpose Layers

[Munzner (ill. Maguire), 2014]
I page. In Figure 2 there are 6 panels, I column, 6 rows, and 1 page. Later, we will show a Trellis display with more than one page. We refer to the rectangular array as the trellis because it is reminiscent of a garden trelliswork.

Each panel of a trellis display shows a subset of the values of panel variables; these values are formed by conditioning on the values of conditioning variables. In Figure 1 the panel variables are variety and yield, and the conditioning variables are site and year. On each panel, values of yield and variety are displayed for one combination of year and site.
Partitioning: Recursive Subdivision

[Slingsby et al., 2009]
Assignment 4

- Interaction, Network, and Multiple Views
- Due before Thanksgiving, but remember Quiz 2!
Superimposition

Superimposed views overlay two or more visual spaces on top of each other (Figures 6 and 7). The resulting visualization becomes the visual combination of the component visualizations, often using transparency to enable seeing all views. Superimposed views are generally used to highlight spatial relations in the component visualizations. In other words, the spatial linking present in these views is one-to-one, i.e., all the overlay visualizations share the same underlying visual space. Line graph visualizations with several data series, where more than one graph is superimposed in a single chart (e.g., [19]), is a very commonly used example of this design pattern.

The spatial linking in the superimposed views allows for easy comparison across different datasets because the user does not have to split their attention between different parts of the visual space. Furthermore, the fact that visualizations are stacked means that they can each use the full available space in the view. However, because the composition simply adds the component visualizations together, the visual clutter may become significant, and it is also likely to cause conflicts arising from one visualization occluding another.

5.1 Mapgets

Mapgets [38] is a geographic visualization system that allows users to interactively perform map editing and querying of geographical datasets. The maps generated using Mapgets are built on an underlying presentation stack that superimposes multiple dataset layers on top of each other. The users can dynamically select the dataset to use for each layer and the total number of layers to compose. Different layers in the presentation stack allow users to independently interact with each of the associated visualization and control the layer attributes. The technique also allows the users to reorder layers in the presentation stack to achieve the desirable map result. Figure 6 shows an example of a European map generated in Mapgets. The presentation stack associated with this map consists of three layers: the bottom layer visualizes rivers, the center layer is used to depict the country borders, and the topmost layer is used to display the country labels.

5.2 GeoSpace

GeoSpace [22] allows users to interactively explore complex visual spaces using superimposed views. It permits progressively overlaying different datasets, based on the user queries, in a single view. Beyond allowing users to explore datasets through dynamic queries, GeoSpace also supports pan and zoom operations for navigation. Figure 7 shows GeoSpace system being used for exploring crime around the Cambridge, MA area. The figure shows a 2D view of the visualization, where red dots that are spatially coupled to the underlying layer show the reported crime cases in the region.

Figure 8: SPPC [45] (Overloaded Views). This tool overloads points into the region bounded by two axes in the parallel coordinate plot.

Figure 9: Links on treemaps [14] (Overloaded Views). The tool identifies a tree structure in a graph and visualizes it using a treemap.

[Mapgets, A. Voisard, 1995]
Superimposed Layers

- Put different layers in the same spatial region, overlay information
- Usually each layer spans the entire view
- Must be identifiable: visually distinguishable
- Cartography has to deal with this a lot
- May be static or dynamic (user controls which layers are shown)
Example: Superimposed Line Charts
Example: Hierarchical Edge Bundles

Fig. 13. A software system and its associated call graph (caller = green, callee = red). (a) and (b) show the system with bundling strength $\beta = 0.85$ using a balloon layout (node labels disabled) and a radial layout, respectively. Bundling reduces visual clutter, making it easier to perceive the actual connections than when compared to the non-bundled versions (figures 2a and 11a). Bundled visualizations also show relations between sparsely connected systems more clearly (encircled regions); these are almost completely obscured in the non-bundled versions. The encircled regions highlight identical parts of the system for (a), (b), and figure 15.

Fig. 14. Using the bundling strength $\beta$ to provide a trade-off between low-level and high-level views of the adjacency relations. The value of $\beta$ increases from left-to-right; low values mainly provide low-level, node-to-node connectivity information, whereas high values provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

The majority of the participants regarded the technique as useful for quickly gaining insight in the adjacency relations present in hierarchically organized systems. In general, the visualizations were also regarded as being aesthetically pleasing. SIG and FEI Company Eindhoven are currently supporting further development by providing us with additional data sets and feedback regarding the resulting visualizations.

More specifically, most of the participants particularly valued the fact that relations between items at low levels of the hierarchy were automatically lifted to implicit relations between items at higher levels by means of bundles. This quickly gave them an impression of the high-level connectivity information while still being able to inspect the low-level relations that were responsible for the bundles by interactively manipulating the bundling strength.

This is illustrated in figure 14, which shows visualizations using different values for the bundling strength $\beta$. Low values result in visualizations that mainly provide low-level, node-to-node connectivity information. High values result in visualizations that provide high-level information as well by implicit visualization of adjacency edges between parent nodes that are the result of explicit adjacency edges between their respective child nodes.

Another aspect that was commented on was how the bundles gave an impression of the hierarchical organization of the data as well, thereby strengthening the visualization of the hierarchy. More specifically, a thick bundle shows the presence of two elements at a fairly
D3 Multiple Views and Interaction

• [https://codepen.io/dakoop/pen/oQxxmx](https://codepen.io/dakoop/pen/oQxxmx)

• Process `mouseover` and `mouseout` events
  - Get selected element
  - Provide **feedback** (e.g. highlighting)

• Find matching items in other view(s)
  - Can use `filter` for this
  - Highlight them
  - Make sure that if they overlap, the highlighted item is on top