Data Visualization (CSCI 627/490)

Interaction & Multiple Views

Dr. David Koop
What is wrong with here and how can it be fixed?
Good: Data magnitude $\iff$ Mark magnitude
Show when the baseline is not zero

See also: "Tear Up Your Baseline" [RJ Andrews]

[W. C. Brinton via RJ Andrews]
Tufte's Lie Factor

• Size of effect = \( \frac{\text{2nd value} - \text{1st value}}{\text{1st value}} \)

• Lie factor = \( \frac{\text{size of effect in graphic}}{\text{size of effect in data}} \)

• In the graphic:

\[
\text{Lie Factor} = \frac{\frac{5.3 - 0.6}{0.6}}{\frac{27.5 - 18}{18}} = 14.8
\]
Avoid Chartjunk

Extraneous visual elements that distract from the message

[T. Brey via A. Lex]
No Unjustified 3D

- Occlusion hides information
- Perspective distortion dangers
- Tilted text isn't legible

- Can help with shape perception
Validation at each level

- ** Threat ** Wrong problem
  - ✓ ** Validate ** Observe and interview target users

- ** Threat ** Wrong task/data abstraction
  - ✓ ** Validate ** Justify encoding/interaction design
    - ✓ ** Validate ** Analyze computational complexity
    - Implement system
    - ✓ ** Validate ** Measure system time/memory
    - ✓ ** Validate ** Qualitative/quantitative result image analysis
      - Test on any users, informal usability study
    - ✓ ** Validate ** Lab study, measure human time/errors for task
    - ✓ ** Validate ** Test on target users, collect anecdotal evidence of utility
    - ✓ ** Validate ** Field study, document human usage of deployed system

- ✓ ** Validate ** Observe adoption rates

[Munzner, 2014]
Five Design Sheet Method

Fig. 4: Five stages to the FdS: (S1) meet with client and consider the task; or deliberate on own. (S2) Ideate and sketch small ideas. (S3) Select users, prepare test scenarios, practice these scenarios, and allocate roles (greeter, facilitator, computer, observers). (S4) Select users, prepare test scenarios, practice these scenarios, and allocate roles (greeter, facilitator, computer, observers). (S5) Generate realization sheet, and plan a prototype. Discuss with client and re-iterate if necessary.

The user thinks divergently and considers many alternative ideas. Buxton et al. [33] present a series of visual sketching methods as a way to solve problems in business and visualization interface development. Roam [34] encourage storyboarding and sketching prototypes for rapid evaluation, and because it maximizes the number of times you get to refine your design before you must commit to code.

Another inspirational idea from architecture design was the idea of parti pris (the big idea). The word comes from the French parti, a bias or a mind-made-up. In architectural-criticism the parti is an assumption that informs the design; it is therefore the central, defining element. In the context of sketching this means that one must pre-judge the outcome of the idea in order to allow it to happen.

Sketching is also used by Walny et al. [23], demonstrating the power of sketching; they explore one designer's sketching practices based on excerpts of their sketchbooks. The use of lo-fidelity sketching frees the user from worry about technical limitations or assumptions and encourages them to externalize and visualize problems. Many creative industries use sketching as a way to investigate, experiment, make decisions, and refine existing solutions.

In visualization, this has been less formally used. Users often sketch multiple ideas, and sketching is an important method for generating several solutions, and make comparison to other lo-fidelity design ideas. Sketches to generate concepts, to externalize and visualize problems, and to consider the different parts of the data at this stage. Buxton et al. [34], for example, presents a series of visual sketching methods as a way to solve problems in business and user scenarios. Craft and Cairns [24] presents a series of visual sketching methods as a way to solve problems in business and user scenarios.

Rettig gave users a pragmatic set of instructions for programmers to develop lo-fi prototypes on paper: assemble a kit (pens, paper, ruler, scissors, etc.), set a deadline, draw models not illustrations. He also develops lo-fi prototypes on paper: assemble a kit (pens, paper, ruler, scissors, etc.), set a deadline, draw models not illustrations. He also develops lo-fi prototypes on paper: assemble a kit (pens, paper, ruler, scissors, etc.), set a deadline, draw models not illustrations.

In the visualization domain, for instance, Craft and Cairns [24] and Curtis and Vertel [25] comprising of five sheets (collect, relate, donate and create) and Curtis and Vertel [25]. The latter four sheets are similar in function, but differ in purpose. The former is the brainstorm (ideas) sheet (Fig. 3), each sheet containing five parts. Explicitly, the first sheet is a computer to create different 3D models, they often render the output in a sketchy appearance (e.g., [20]). Similarly prototype visualization tools can be rendered in a sketchy appearance. Similarly prototype visualization tools can be rendered in a sketchy appearance.
Sheets 2-4

...discovery of penicillin and it is these that are actually good ideas: e.g., sticky-note glue or the well written examples where scientists make errors or have accidents. Either try to fix these mistakes or use the result to your favor. There are...interpretation.

...metaphors work both ways: they both inspire and are needed for in-...similar ideas together. E.g., Johnson with different skills and knowledge...fine:

Re-work:

...and careful thought, says Johnson...theory is summarized in...questioning

...How do new concepts get 'born'? The five parts of sheet 1 leads the...Ideation is the process of creating new ideas. But where do ideas come...3.3 FdS Sheet 1: Ideation

...user to think divergently, to first generate new ideas...user needs to gather resources together...Finally the user needs to gather resources together...[J. Roberts et al., 2016]
Project Design

• Work on turning your visualization ideas into designs
• Turn in:
  - Three Designs Sketches
  - One Bad Design
  - Progress on Implementation
• Options:
  - Try vastly different options
  - Refine an initial idea
• Due Friday, Nov. 13
Assignment 4

• Geospatial Visualizations & Treemap
  - Choose colormaps carefully
  - Add legend
• Due Nov. 2
Guidelines for Interaction Design
Interaction

- The view changes over time
- Changes can affect almost any aspect of the visualization
  - encoding
  - arrangement
  - ordering
  - viewpoint
  - attributes being shown
  - aggregation level
Interaction Overview

- **Change over Time**
  - Select

- **Navigate**
  - Item Reduction
    - Zoom
    - Geometric or Semantic
  - Pan/Translate
  - Constrained
  - Attribute Reduction
    - Slice
    - Cut
    - Project

[Munzner (ill. Maguire), 2014]
 Sorting

• Allow user to find patterns by reordering the data
• Do this with tabular data all the time
• Note that categorical attributes don't really need sorting
  - We can compare these attributes no matter what order
  - Instead, sort categorical attribute based on an ordered attribute
Example: LineUp

[Gratzl et al., 2013]
Example: LineUp

[Gratzl et al., 2013]
Slope Graphs

- Connection marks
- Link the same item appearing in different rows
- Show changes for different attributes (parallel coordinates idea) but with one highlighted item
- Also called bump charts
Animation: Jump Cut vs. Animated Transitions

<table>
<thead>
<tr>
<th>Q♠</th>
<th>A♠</th>
<th>Q♣</th>
<th>K♠</th>
</tr>
</thead>
<tbody>
<tr>
<td>K♦</td>
<td>J♦</td>
<td>Q♦</td>
<td>A♦</td>
</tr>
<tr>
<td>A♥</td>
<td>J♣</td>
<td>Q♥</td>
<td>J♥</td>
</tr>
<tr>
<td>A♦</td>
<td>A♠</td>
<td>Q♦</td>
<td>J♥</td>
</tr>
<tr>
<td>Q♦</td>
<td>K♥</td>
<td>J♦</td>
<td>A♠</td>
</tr>
<tr>
<td>K♠</td>
<td>J♠</td>
<td>K♠</td>
<td>J♠</td>
</tr>
<tr>
<td>K♥</td>
<td>K♠</td>
<td>J♣</td>
<td>A♥</td>
</tr>
<tr>
<td>A♠</td>
<td>Q♥</td>
<td>K♥</td>
<td>Q♥</td>
</tr>
<tr>
<td>J♥</td>
<td>Q♣</td>
<td>K♠</td>
<td>A♠</td>
</tr>
<tr>
<td>Q♣</td>
<td>K♠</td>
<td>A♥</td>
<td>K♥</td>
</tr>
<tr>
<td>K♦</td>
<td>J♠</td>
<td>A♠</td>
<td>K♥</td>
</tr>
</tbody>
</table>

D. Koop, CSCI 627/490, Fall 2020
## Animation: Jump Cut vs. Animated Transitions

<table>
<thead>
<tr>
<th>♦️</th>
<th>♣️</th>
<th>♡️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>A</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>Q</td>
<td>Q</td>
</tr>
<tr>
<td>Q</td>
<td>J</td>
<td>Q</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>A</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>K</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>A</td>
</tr>
<tr>
<td>J</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
<td>K</td>
</tr>
</tbody>
</table>

---

D. Koop, CSCI 627/490, Fall 2020
# Animation: Jump Cut vs. Animated Transitions

<table>
<thead>
<tr>
<th>♠️</th>
<th>♠️</th>
<th>♠️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>K</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>J</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>♥️</th>
<th>♥️</th>
<th>♥️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>K</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>J</td>
</tr>
</tbody>
</table>
## Animation: Jump Cut vs. Animated Transitions

<table>
<thead>
<tr>
<th>♦️</th>
<th>♠️</th>
<th>♣️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>A</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>J</td>
<td>Q</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>J</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>J</td>
<td>K</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>K</td>
<td>A</td>
</tr>
<tr>
<td>J</td>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>Q</td>
<td>K</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>A</td>
<td>K</td>
</tr>
<tr>
<td>J</td>
<td>K</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
<td>J</td>
</tr>
</tbody>
</table>

---

D. Koop, CSCI 627/490, Fall 2020
## Side-by-side views

<table>
<thead>
<tr>
<th>♣️</th>
<th>♣️</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>A</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
</tr>
<tr>
<td>A</td>
<td>Q</td>
</tr>
<tr>
<td>A</td>
<td>♠️</td>
</tr>
<tr>
<td>Q</td>
<td>♠️</td>
</tr>
<tr>
<td>Q</td>
<td>J</td>
</tr>
<tr>
<td>A</td>
<td>K</td>
</tr>
<tr>
<td>K</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>J</td>
<td>A</td>
</tr>
<tr>
<td>Q</td>
<td>♠️</td>
</tr>
<tr>
<td>K</td>
<td>♠️</td>
</tr>
<tr>
<td>J</td>
<td>♠️</td>
</tr>
<tr>
<td>J</td>
<td>♠️</td>
</tr>
<tr>
<td>J</td>
<td>♠️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>♥️</th>
<th>♥️</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Q</td>
<td>Q</td>
</tr>
<tr>
<td>K</td>
<td>K</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
</tr>
</tbody>
</table>
### Side-by-side views

<table>
<thead>
<tr>
<th>♠️ Q</th>
<th>♠️ A</th>
</tr>
</thead>
<tbody>
<tr>
<td>♠️ K</td>
<td>♠️ Q</td>
</tr>
<tr>
<td>♠️ A</td>
<td>♠️ Q</td>
</tr>
<tr>
<td>♠️ A</td>
<td>♠️ Q</td>
</tr>
<tr>
<td>♠️ Q</td>
<td>♠️ J</td>
</tr>
<tr>
<td>♠️ Q</td>
<td>♠️ J</td>
</tr>
<tr>
<td>♠️ A</td>
<td>♠️ K</td>
</tr>
<tr>
<td>♠️ K</td>
<td>♠️ Q</td>
</tr>
<tr>
<td>♠️ Q</td>
<td>♠️ K</td>
</tr>
<tr>
<td>♠️ A</td>
<td>♠️ J</td>
</tr>
<tr>
<td>♠️ J</td>
<td>♠️ J</td>
</tr>
<tr>
<td>♠️ J</td>
<td>♠️ J</td>
</tr>
</tbody>
</table>
Animated Transitions

Stacked  Grouped

[M. Bostock]
Animated Transitions

- Stacked
- Grouped
Animated Transitions

• "Jump cuts" are hard to follow
• Animations help users maintain sense of context between two states
• Empirical study showed that they work (Heer & Robertson, 2007)
Studying Animated Transitions

[Heer and Robertson, 2007]
Studying Animated Transitions

[Heer and Robertson, 2007]
Design Considerations

• Based on Tversky et al.'s Congruence and Apprehension Principles

• Congruence (Expressiveness):
  - Use consistent semantic-syntactic mappings
  - Respect semantic correspondence
  - Avoid ambiguity

• Apprehension (Effectiveness):
  - Group similar transitions
  - Minimize occlusion
  - Maximize predictability
  - Use simple transitions
  - Use staging for complex transitions
  - Transitions as long as needed, but no longer

[Heer and Robertson, 2007]
Experiment 1 (Syntactic)

- Object Tracking: Follow objects across a transition and identify the locations of the objects in the final graphic
  - Tests: bar chart to donut chart, stacked to grouped bars, sorting a bar chart, scatter plot to bar chart, timestep in a scatterplot
  - Either a jump cut or an animated transition
  - Users pick highlighted elements after transition (measure #pixels from correct)

[Heer and Robertson, 2007]
Experiment 2 (Semantic)

- Estimating Changing Values: Follow a single target across transition and estimate the percentage change in value
  - Tests: axis rescaling + timestep animations
  - In stacked bars, each stack level updates separately, donut charts are multi-stage
  - Users asked to enter an estimate of change (increments of 20% from -90% to 90% or click "?" for no idea)
Results/Conclusions

- User Preferences: Staged animation > animation > static transitions

- Animation improves graphical perception
- Staging is better (do axis rescaling before value changes)
- Avoid axis rescaling when possible

[Heer and Robertson, 2007]
Change Blindness

- [https://www.youtube.com/watch?v=uO8wpm9HSB0](https://www.youtube.com/watch?v=uO8wpm9HSB0)
Change Blindness

- https://www.youtube.com/watch?v=uO8wpm9HSB0
Selection

- Selection is often used to initiate other changes
- User needs to select something to drive the next change
- What can be a selection target?
  - Items, links, attributes, (views)
- How?
  - mouse click, mouse hover, touch
  - keyboard modifiers, right/left mouse click, force
- Selection modes:
  - Single, multiple
  - Contiguous?
Highlighting

- Selection is the user action
- Feedback is important!
- How? Change selected item's visual encoding
  - Change color: want to achieve visual popout
  - Add outline mark: allows original color to be preserved
  - Change size (line width)
  - Add motion: marching ants
Highlighting

• Selection is the user action
• Feedback is important!
• How? Change selected item's visual encoding
  - Change color: want to achieve visual popout
  - Add outline mark: allows original color to be preserved
  - Change size (line width)
  - Add motion: marching ants
Highlighting

Obama has 431 ways to win
5 ties
Romney has 76 ways to win

If Romney wins Ohio...

Florida
Ohio
North Carolina
Virginia
Wisconsin
Colorado
Iowa
Nevada
New Hampshire

60% of paths
0.00% of paths
10% of paths

NYTimes
Selection Outcomes

- Selection is usually a part of an action sequence
- Can filter, aggregate, reorder selected items
Responsiveness Required

• Delays are perceived by users
• Visual feedback
  - Show the user they did something (highlighting, etc)
  - Interaction should happen quick!
• Latency: mouse click versus mouse hover
• Popup versus detail displays
Interaction Latency

- The Effects of Interactive Latency on Exploratory Visual Analysis, Z. Liu and J. Heer, 2014
- Brush & link, select, pan, zoom

- 500ms added latency causes significant cost
  - decreases user activity and dataset coverage
  - reduces rate of observations, generalizations, and hypotheses
Interaction Overview

**Change over Time**

**Navigate**

**Select**

- **Item Reduction**
  - Zoom
  - Geometric or Semantic

- **Attribute Reduction**
  - Slice
  - Cut
  - Project

[D. Koop, CSCI 627/490, Fall 2020]

[Munzner (ill. Maguire), 2014]
Navigation

• Fix the layout of all visual elements but provide methods for the viewpoint to change

• Camera analogy: only certain features visible in a frame
  - Zooming
  - Panning (aka scrolling)
  - Translating
  - Rotating (rare in 2D, important in 3D)
Navigation

- Item Reduction
  - Zoom
    - Geometric or Semantic
  - Pan/Translate
  - Constrained

- Attribute Reduction
  - Slice
  - Cut
  - Project

[Munzner (ill. Maguire), 2014]
Zooming
Geometric Zooming
Zooming
Semantic Zooming
Zooming

- Geometric Zooming: just like a camera
- Semantic Zooming: visual appearance of objects can change at different scales
- LiveRAC Example: (focus + context)

Figure 3. LiveRAC shows a full day of system management time-series data using a reorderable matrix of area-aware charts. Over 4000 devices are shown in rows, with 11 columns representing groups of monitored parameters. (a): The user has sorted by the maximum value in the CPU column. The first several dozen rows have been stretched to show sparklines for the devices, with the top 13 enlarged enough to display text labels. The time period of business hours has been selected, showing the increase in the In pkts parameter for many devices. (b): The top three rows have been further enlarged to show fully detailed charts in the CPU column and partially detailed ones in Swap and two other columns. The time marker (vertical black line on each chart) indicates the start of anomalous activity in several of spire’s parameters. Below the labeled rows, we see many blocks at the lowest semantic zoom level, and further below we see a compressed region of highly saturated blocks that aggregate information from many charts.

Principle: multiple views are most effective when coordinated through explicit linking.

The principle of linked views [15] is that explicit coordination between views enhances their value. In LiveRAC, as the user moves the cursor within a chart, the same point in time is marked in all charts with a vertical line. Similarly, selecting a time segment in one chart shows a mark in all of them. This technique allows direct comparison between parameter values at the same time on different charts. In addition, people can easily correlate times between large charts with detailed axis labels, and smaller, more concise charts.

Assertion: showing several levels of detail simultaneously provides useful high information density in context.

Several technique choices are based on this assertion. First, LiveRAC uses stretch and squish navigation, where expanding one or many regions compresses the rest of the view [11, 17]. The accompanying video shows the look and feel of this navigation technique. The stretching and squishing operates on rectangular regions, so expanding a single chart also magnifies the entire row for the device it represents, and the entire column for the parameters that it shows. The edges of the display are fixed so that all cells remain within the visible area, as opposed to conventional zooming where some regions are pushed off-screen. There are rapid navigation shortcuts to zoom a single cell, a column, an aggregated group of devices, the results of a search, or to zoom out to an overview. Users can also directly drag grid lines or resize freely drawn on-screen rectangles. Navigation shortcuts can also be created for any arbitrary grouping, whose cells do not need to be contiguous. This interaction mechanism affords multiple focus regions, supporting multiple levels of detail.

Second, charts in LiveRAC dynamically adapt to show visual representations adapted in each cell to the available screen space. This technique, called semantic zooming [13], allows a hierarchy of representations for a group of device-parameter time-series. In Figure 3, the largest charts have multiple overlaid curves and detailed axis and legend labels. Smaller charts show fewer curves and less labeling, and at smaller sizes only one curve is shown as a sparkline [24]. On each curve, the maximum value over the displayed time period is indicated with a red dot, the minimum with a blue dot, and the current value with a green one. All representation levels color code the background rectangle according to dynamically changeable thresholds of the minimum, maximum, or average values of the parameters within the current time window. The smallest view is a simple block, where this color coding is the only information shown.

Third, aggregation techniques achieve visual scalability by ensuring dense regions show meaningful visual representations. Given our target scale of dozens of parameters and thousands of devices, the size of the matrix could easily surpass 100,000 cells. Stretch and squish navigation allows users to quickly create a mosaic with cells of many different sizes.
Navigation Constraints

- **Unconstrained** navigation: walking around in the world or an immersive 3D environment
  - Fairly standard in computer games to go where you want
  - Constrained by walls, objects (collision detection)

- **Constrained navigation:**
  - 3D: camera must be right-side up
  - Limit pan/zoom to certain areas
  - Comes up often with **multiple views**: want to show an area in one view that corresponds to a selection in another view
van Wijk Smooth Zooming
van Wijk Smooth Zooming

[van Wijk, 2003, M. Bostock]
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
Multiple Views

• Big questions:
  - How to partition display or layer views?
  - How to coordinate views (e.g. navigation, selection)?
  - What data is shared?
Design Space of Composite Visualization

- Composite visualization views (CVVs)
  - Includes Coordinated multiple views (CMV)
  - More!

- Design Patterns:
  - Juxtaposition: side-by-side
  - Superimposition: layers
  - Overloading: vis meshed with another
  - Nesting: vis inside a vis (recursive vis)
  - Integration: "merge" views + links

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

Figure 3: Interior of the ion trap. The green ball is the ion, the white ball is the ion, the red ball is the ion, and the blue ball is the ion. The yellow ball is the ion, and the purple ball is the ion.
Juxtaposition
Juxtaposition Guidelines

• Benefits:
  - The component visualizations are independent and can be composed without interference
  - Easy to implement

• Drawbacks:
  - Implicit visual linking is not always easy to see, particularly when multiple objects are selected
  - Space is divided between the views, yielding less space for each view

• Applications: Use for heterogeneous datasets consisting of many different types of data, or for where different independent visualizations need to be combined.

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

Semantic Substrates

Figure 3 shows a visual exploration of a simulated ion trajectory in a cubic ion trap using Improvise. Improvise is a visualization framework based on the juxtaposition (or tiling) of component visualizations (Figures 4, 5). For this reason, the integrated views design pattern is also based on juxtaposing (or tiling) the component visualizations. Improvise allows users to highlight the corresponding data items.

ComVis

ComVis is a multidimensional visualization system supporting multiple coordinated views for exploring complex datasets (Figure 2). The dataset is shown in the form of a table view at the bottom of the main window. Beyond basic interactions, ComVis also supports three-bin brushing, allowing it to be adapted for virtually any type of data and visualization.
Integration

[VisLink, Collins and Carpendale, 2007]
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]
Overloading

Figure 6: Mapgets [38] (Superimposed Views). Presentation stack, with superimposed layers for rivers, borders, and labels, in Mapgets.

Figure 7: GeoSpace [22] (Superimposed Views). A crime data layer superimposed on a geographical map of the Cambridge, MA area.

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.

[SPCC, X. Yuan et al., 2009]
Nesting

D. Koop, CSCI 627/490, Fall 2020

[ZAME, N. Elmqvist et al., 2008]
Nesting

Fekete et al. [14] proposed a technique for rendering graphs using a treemap visualization. The idea is based on the fact that it is possible to decompose a graph into a tree structure and then overload links corresponding to the remaining graph edges that are not included in the tree. This allows for using a treemap to visualize the tree structure, and then overload links corresponding to the remaining graph edges on the treemap visualization. Even though Fekete et al. [14] proposed a technique for rendering graphs using a treemap with overloaded graph links, the technique is based on the trends in any dimension of a dataset [10]. Combining both techniques, on the other hand, provide an effective way of correlating visual structures themselves must be modified to combine the components. We will see examples of this below.

Parallel coordinates are efficient for visualizing multiple dimensions in a compact 2D visualization, called the parallel coordinates visualization, uses this approach. The base matrix represents the space between pairs of scatterplots on a parallel coordinates visualization [18] (Figure 8). The technique is based on converting the space between pairs of scatterplots through multidimensional scaling [42]. The technique is similar to superimposing views, but with some important differences. Like superimposing views, there exists no means that this space is open for being overloaded.

This design pattern characterizes compositions where one visualization, called the host [26], is rendered inside another, using the same spatial mapping as the client visualization. The technique is also an example of combining two techniques to compensate for their individual shortcomings. Parallel coordinates are simply replaced with the visual marks of the host with the visual structure of the client visualization (Figures 10 and 11). An example of this would be a scatterplot where the individual marks are barchart glyphs [25]. The nested views pattern provides an effective way of relating visual structures across dimensions. We now provide an example of using nested views inside a visual mark, the client visualization (Figures 10 and 11). An example of this would be a scatterplot where the individual marks are barchart glyphs [25]. The nested views pattern provides an effective way of relating visual structures across dimensions.

[NodeTrix, N. Henry et al., 2007]
Midterm