Information Visualization
Manipulate Interactive, Facet into Multiple, Scalable Insets

Ex: Complexity Families

Tamara Munzner
Department of Computer Science
University of British Columbia

Week 7, 3 Nov 2021

https://www.cs.ubc.ca/~tmm/courses/547-21
Plan for today

• small group exercises
  – Complexity Families

• last week reading Q&A
  – paper: Polaris

• this week reading Q&A
  – chap: Manipulate Interactive, Multiple Views. paper: ScalableInsets

• reminder: post-class office hours
  – if you want discussion of your project proposal feedback ASAP
    • faster than waiting for my written comments
Upcoming

• next week (W10): reading week. no class, no readings, no async discussion
  – work on projects!

• week after (W11)
  – **light async reading/discussion (note updated web page)**
    • 1 reading: Ch 13, Reduce
  – due Tue 3pm: project updates
  – in class: project peer reviews
    • each team will be matched with one other
    • read other team's written update before class
    • first A critiques B; then B critiques A
    • record discussion/thoughts in gdoc
  – in class: mini-lecture
Upcoming

• week after that (W12)
  – async: last week of readings / discussion
    • Ch 14: Embed - Focus+Context
      [type: design study]
    – in class: post-update meetings with Tamara
      • oral feedback on project progress, after I've read them

• last week of classes (W13)
  – async: no readings/discussion
  – in class: lecture on research process and final writeup expectations

• final presentations (W15)
Visualization Analysis & Design

Interactive Views (Ch 11/12)

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner
How to handle complexity: 1 previous strategy

→ Derive

• derive new data to show within view
How to handle complexity: 1 previous strategy + 2 more

- Derive

Manipulate

- Change

Facet

- Juxtapose

• derive new data to show within view
• change view over time
• facet across multiple views

- Select

- Navigate

- Partition

- Superimpose
Manipulate View
Manipulate

Change over Time

[Diagram showing data points changing over time with a trend]
Change over time

• change any of the other choices
  – encoding itself
  – parameters
  – arrange: rearrange, reorder
  – aggregation level, what is filtered...

  – interaction entails change

• powerful & flexible
Idiom: Re-encode

made with Tableau, http://tableausoftware.com
Idiom: Change parameters

- widgets and controls
  - sliders, buttons, radio buttons, checkboxes, dropdowns/comboboxes
- pros
  - clear affordances, self-documenting (with labels)
- cons
  - uses screen space
- design choices
  - separated vs interleaved
    - controls & canvas

[Growth of a Nation](http://laurenwood.github.io/)
made with D3
Idiom: **Change order/arrangement**

- **what**: simple table
- **how**: data-driven reordering
- **why**: find extreme values, trends

[Sortable Bar Chart](https://observablehq.com/@d3/sortable-bar-chart)

*made with D3*
Idiom: **Reorder**

- **what:** table with many attributes
- **how:** data-driven reordering by selecting column
- **why:** find correlations between attributes


*made with D3*
Idiom: Change alignment

- stacked bars
  - easy to compare
    - first segment
    - total bar

- align to different segment
  - supports flexible comparison

Idiom: **Animated transitions - visual encoding change**

- smooth transition from one state to another
  - alternative to jump cuts, supports item tracking
    - best case for animation
  - staging to reduce cognitive load

[Stacked to Grouped Bars](https://observablehq.com/@d3/stacked-to-grouped-bars)
Idiom: **Animated transition - tree detail**

- animated transition
  - network drilldown/rollup

[Collapsible Tree] https://observablehq.com/@d3/collapsible-tree
Manipulate

- Change over Time

- Select
Interaction technology

• what do you design for?
  – mouse & keyboard on desktop?
    • large screens, hover, multiple clicks
  – touch interaction on mobile?
    • small screens, no hover, just tap
  – gestures from video / sensors?
    • ergonomic reality vs movie bombast
  – eye tracking?

Data visualization and the news - Gregor Aisch (37 min)
vimeo.com/182590214

I Hate Tom Cruise - Alex Kauffmann (5 min)
www.youtube.com/watch?v=QXLfT9sFcbc
Selection

- selection: basic operation for most interaction
- design choices
  - how many selection types?
    - interaction modalities
      - click/tap (heavyweight) vs hover (lightweight but not available on most touchscreens)
      - multiple click types (shift-click, option-click, …)
      - proximity beyond click/hover (touching vs nearby vs distant)
    - application semantics
      - adding to selection set vs replacing selection
      - can selection be null?
        - ex: toggle so nothing selected if click on background
      - primary vs secondary (ex: source/target nodes in network)
      - group membership (add/delete items, name group, …)
Highlighting

• highlight: change visual encoding for selection targets
  – visual feedback closely tied to but separable from selection (interaction)

• design choices: typical visual channels
  – change item color
    • but hides existing color coding
  – add outline mark
  – change size (ex: increase outline mark linewidth)
  – change shape (ex: from solid to dashed line for link mark)

• unusual channels: motion
  – motion: usually avoid for single view
    • with multiple views, could justify to draw attention to other views
Manipulate

- Change over Time
- Navigate
  - Item Reduction
    - Zoom
      - Geometric or Semantic
  - Pan/Translate
  - Constrained
Manipulate

- Change over Time
- Select

Navigate

- Zoom
- Pan/Translate
- Constrained

Geometric or Semantic Attribute Reduction
Navigate: Changing viewpoint/visibility

• change viewpoint
  – changes which items are visible within view

• camera metaphor
  – pan/translate/scroll
    • move up/down/sideways

→ Navigate

→ Pan/Translate
Idiom: **Scrollytelling**

- **how**: navigate page by scrolling (panning down)
- **pros**:  
  - familiar & intuitive, from standard web browsing  
  - linear (only up & down) vs possible overload of click-based interface choices
- **cons**:  
  - full-screen mode may lack affordances  
  - scrolljacking, no direct access  
  - unexpected behaviour  
  - continuous control for discrete steps

[How to Scroll, Bostock](https://bost.ocks.org/mike/scroll/)
https://eagereyes.org/blog/2016/the-scrollytelling-scourge
Navigate: Changing viewpoint/visibility

• change viewpoint
  – changes which items are visible within view

• camera metaphor
  – pan/translate/scroll
    • move up/down/sideways
  – rotate/spin
    • typically in 3D
  – zoom in/out
    • enlarge/shrink world == move camera closer/further
    • geometric zoom: standard, like moving physical object
Navigate: Unconstrained vs constrained

- unconstrained navigation
  - easy to implement for designer
  - hard to control for user
    - easy to overshoot/undershoot
- constrained navigation
  - typically uses animated transitions
  - trajectory automatically computed based on selection
    - just click; selection ends up framed nicely in final viewport
Idiom: **Animated transition + constrained navigation**

- example: geographic map
  - simple zoom, only viewport changes, shapes preserved

[Zoom to Bounding Box](https://observablehq.com/@d3/zoom-to-bounding-box)
Navigate: Reducing attributes

- continuation of camera metaphor
  - slice
    - show only items matching specific value for given attribute: slicing plane
    - axis aligned, or arbitrary alignment
  - cut
    - show only items on far slide of plane from camera
  - project
    - change mathematics of image creation
      - orthographic
      - perspective
      - many others: Mercator, cabinet, ...

Interaction benefits

• interaction pros
  – major advantage of computer-based vs paper-based visualization
  – flexible, powerful, intuitive
    • exploratory data analysis: change as you go during analysis process
    • fluid task switching: different visual encodings support different tasks
  – animated transitions provide excellent support
    • empirical evidence that animated transitions help people stay oriented
Interaction limitations

• interaction has a time cost
  – sometimes minor, sometimes significant
  – degenerates to human-powered search in worst case
• remembering previous state imposes cognitive load
• controls may take screen real estate
  – or invisible functionality may be difficult to discover (lack of affordances)
• users may not interact as planned by designer
  – NYTimes logs show ~90% don’t interact beyond scrollytelling - Aisch, 2016
Visualization Analysis & Design

Interactive Views (Ch 11/12) II

Tamara Munzner
Department of Computer Science
University of British Columbia
@tamaramunzner
How to handle complexity: 1 previous strategy + 2 more

Derive

- Derive new data to show within view
- Change view over time
- Facet across multiple views

Manipulate

- Change
- Select
- Navigate

Facet

- Juxtapose
- Partition
- Superimpose

Actions

- Derive new data to show within view
- Change view over time
- Facet across multiple views
Multiple Views
Facet

- Juxtapose

- Partition

- Superimpose
Facet

- **Juxtapose**

- **Partition**

- **Superimpose**
Juxtapose and coordinate views

- Share Encoding: Same/Different
  - Linked Highlighting

- Share Data: All/Subset/None

- Share Navigation
Idiom: **Linked highlighting**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom

- encoding: different
  - *multiform*

- data: all shared
  - all *items* shared
  - different *attributes* across the views

- aka: brushing and linking

---

*Visual Exploration of Large Structured Datasets*. Wills. 
Linked views: Directionality

• unidirectional vs bidirectional linking
  – bidirectional almost always better!

http://pbeshai.github.io/linked-highlighting-react-vega-redux/
https://medium.com/@pbesh/linked-highlighting-with-react-d3-js-and-reflux-16e9c0b2210b
Idiom: **Overview-detail views**

- encoding: same or different  
  - ex: same (birds-eye map)
- data: subset shared  
  - viewpoint differences: subset of data items
- navigation: shared  
  - bidirectional linking
- other differences  
  - (window size)

System: **Google Maps**

Idiom: **Overview-detail navigation**

- encoding: same or different
- data: subset shared
- navigation: shared
  - unidirectional linking
  - select in small overview, change extent in large detail view

[https://observablehq.com/@uwdata/interaction](https://observablehq.com/@uwdata/interaction)
Idiom: **Tooltips**

- popup information for selection
  - hover or click
  - specific case of detail view: provide useful additional detail on demand
  - beware: does not support overview!
    - always consider if there's a way to visually encode directly to provide overview
    - “If you make a rollover or tooltip, assume nobody will see it. If it's important, make it explicit.”
      - — Gregor Aisch, NYTimes

[https://www.highcharts.com/demo/dynamic-master-detail]
Idiom: **Small multiples**

- **encoding: same**
  - ex: line charts
- **data: none shared**
  - different slices of dataset
    - items or attributes
    - ex: stock prices for different companies

[https://bl.ocks.org/mbostock/1157787]
Interactive small multiples

• linked highlighting: analogous item/attribute across views
  – same year highlighted across all charts if hover within any chart

[https://bl.ocks.org/ColinEberhardt/3c780088c363d1515403f50a87a87121]
[https://blog.scottlogic.com/2017/04/05/interactive-responsive-small-multiples.html]
[http://projects.flowingdata.com/tut/linked_small_multiples_demo/]
Example: Combining many interaction idioms

System: **Buckets**

- multiform
- multidirectional linked highlighting of small multiples
- tooltips

http://buckets.peterbeshai.com/
Juxtapose views: tradeoffs

• juxtapose costs
  – display area
    • 2 views side by side: each has only half the area of one view

• juxtapose benefits
  – cognitive load: eyes vs memory
    • lower cognitive load: move eyes between 2 views
    • higher cognitive load: compare single changing view to memory of previous state
Juxtapose vs animate

- animate: hard to follow if many scattered changes or many frames
  – vs easy special case: animated transitions

Juxtapose vs animate

• animate: hard to follow if many scattered changes or many frames
  – vs easy special case: animated transitions

• juxtapose: easier to compare across small multiples
  – different conditions (color), same gene (layout)

# View coordination: Design choices

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Data</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Subset</td>
<td>None</td>
<td></td>
</tr>
<tr>
<td>Same</td>
<td>Redundant</td>
<td>Overview/Detail</td>
<td>Small Multiples</td>
<td></td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
<td>Overview/Detail</td>
<td>No Linkage</td>
<td></td>
</tr>
</tbody>
</table>
Idiom: **Reorderable lists**

- list views
  - easy lookup
  - useful when linked to other views

- how many views is ok vs too complex?
  - open research question

Partition into views

- how to divide data between views
  - split into regions by attributes
  - encodes association between items using spatial proximity
  - order of splits has major implications for what patterns are visible
Partitioning: Grouped vs small-multiple bars

- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages

- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states

[https://observablehq.com/@d3/grouped-bar-chart]

[https://bl.ocks.org/mbostock/4679202]
Partitioning: Recursive subdivision

• split by neighborhood
  – flat, terrace, semi-detached, detached

• then by type
  – years as rows
  – months as columns

• then time

• color by price

• neighborhood patterns
  – where it’s expensive
  – where you pay much more for detached type

Partitioning: Recursive subdivision

• switch order of splits
  – type then neighborhood

• switch color
  – by price variation

• type patterns
  – within specific type, which neighborhoods inconsistent

Partitioning: Recursive subdivision

• different encoding for second-level regions
  – choropleth maps

Facet

- **Juxtapose**

- **Partition**

- **Superimpose**
Superimpose layers

• layer: set of objects spread out over region
  – each set is visually distinguishable group
  – extent: whole view

• design choices
  – how many layers, how to distinguish?
    • encode with different, nonoverlapping channels
    • two layers achievable, three with careful design
  – small static set, or dynamic from many possible?
Static visual layering

• foreground layer: roads
  – hue, size distinguishing main from minor
  – high luminance contrast from background

• background layer: regions
  – desaturated colors for water, parks, land areas

• user can selectively focus attention

Idiom: Trellis plots

• superimpose within same frame
  – color code by year

• partitioning
  – split by site, rows are barley varieties

• main-effects ordering
  – derive value of median for group
  – order rows within view by variety median
  – order views themselves by site median

Superimposing limits (static)

- few layers, more lines
  - up to a few dozen lines
  - but not hundreds

- superimpose vs juxtapose: empirical study
  - same size: all multiples, vs single superimposed
    - superimposed: local tasks
    - juxtaposed: global tasks, esp. for many charts

Dynamic visual layering

• interactive, based on selection
• one-hop neighbour highlighting

click (heavyweight)  
hover (fast)

https://mariandoerk.de/edgemaps/demo/
How?

Encode

- **Arrange**
  - Express
  - Separate
- **Order**
  - Align
- **Use**

Map from **categorical** and **ordered** attributes

- **Color**
  - Hue
  - Saturation
  - Luminance
- **Size, Angle, Curvature, ...**

- **Shape**
  - + • ■ ▲

- **Motion**
  - Direction, Rate, Frequency, ... 

Manipulate

- **Change**
- **Select**
- **Navigate**

Facet

- **Juxtapose**
- **Partition**
- **Superimpose**

Reduce

- **Filter**
- **Aggregate**
- **Embed**