Information Visualization
Interactive Views

Tamara Munzner
Department of Computer Science
University of British Columbia

Lect 8/9/10, 30 Jan & 4/6 Feb 2020

https://www.cs.ubc.ca/~tmm/courses/436V-20
Upcoming

• Foundations 3: out Thu Jan 30, due Wed Feb 5 6pm
• Programming 2: out Thu Jan 30, due Wed Feb 12 6pm
• D3 videos/readings week 4
  – The General Update Pattern of D3.js [60 min]
  – Interaction with Unidirectional Data Flow [16 min]
  – Read: Reusable D3 Components
• Quiz 4, due by Fri Jan 31, 8am
Interactive Views
How to handle complexity: 1 previous strategy + 3 more

- Derive
  - derive new data to show within view
  - change view over time
  - facet across multiple views
  - reduce items/attributes within single view

- Manipulate
  - Change
  - Select
  - Navigate

- Facet
  - Juxtapose
  - Partition
  - Superimpose

- Reduce
  - Filter
  - Aggregate
  - Embed

Actions:
- Analyze
- Search
- Query
- Consume
- Present
- Enjoy
- Discover
- Produce
- Annotate
- Record
- Derive
- Identify
- Compare
- Summarise

Target known
Target unknown
Location known
Location unknown
Lookup
Locate
Browse
Explore
Manipulate

- **Change over Time**

- **Select**

- **Navigate**
  - **Item Reduction**
    - **Zoom**
      - Geometric or Semantic
    - **Pan/Translate**
    - **Constrained**
Change over time

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

  – interaction entails change
Idiom: **Re-encode**

System: **Tableau**

made using Tableau, [http://tableausoftware.com](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

slide inspired by: Alexander Lex, Utah
Idiom: **Change order/arrangement**

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

[Sortable Bar Chart](https://bl.ocks.org/mbostock/3885705)
**Idiom:** Reorder

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

**System:** DataStripes

[http://carlmanaster.github.io/datastripes/]
Idiom: Change alignment

- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison

System: LineUp

Shiny example

- APGI genome browser
  - tooling: R/Shiny
  - interactivity
    - tooltip detail on demand on hover
    - expand/contract chromosomes
    - expand/contract control panes

https://gallery.shinyapps.io/genome_browser/
Idiom: **Animated transitions**

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

- example: animated transitions in statistical data graphics

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](http://bl.ocks.org/mbostock/3943967)
Idiom: **Animated transition - tree detail**

- animated transition
  - network drilldown/rollup

[Collapsible Tree](https://bl.ocks.org/mbostock/4339083)
Idiom: **Animated transition - bar detail**

- example: hierarchical bar chart
  - add detail during transition to new level of detail

[Hierarchical Bar Chart](https://bl.ocks.org/mbostock/1283663)
Interactive transitions quiz: 4 Ways Budget

• what changed?

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?

slide inspired by: Alexander Lex, Utah

Data visualization and the news - Gregor Aisch (37 min)
www.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
Tooltips

• popup information for selection
  – hover or click
  – can 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
Rule of thumb: **Responsiveness is required**

- **visual feedback: three rough categories**
  - **0.1 seconds: perceptual processing**
    - subsecond response for mouseover highlighting - ballistic motion
  - **1 second: immediate response**
    - fast response after mouseclick, button press - Fitts’ Law limits on motor control
  - **10 seconds: brief tasks**
    - bounded response after dialog box - mental model of heavyweight operation (file load)

- **scalability considerations**
  - highlight selection without complete redraw of view (graphics frontbuffer)
  - show hourglass for multi-second operations (check for cancel/undo)
  - show progress bar for long operations (process in background thread)
  - rendering speed when item count is large (guaranteed frame rate)
Manipulate

- **Change over Time**

- **Select**

- **Navigate**

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

- **Attribute Reduction**

- **Slice**

- **Cut**

- **Project**
Navigate: Changing viewpoint/visibility

- change viewpoint
  - changes which items are visible within view
- camera metaphor
  - pan/translate/scroll
    - move up/down/sideways

→ Navigate
  → Item Reduction

→ 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

https://eagereyes.org/blog/2016/the-scrollytelling-scourge

[How to Scroll, Bostock](https://bost.ocks.org/mike/scroll/)

*slide inspired by: Alexander Lex, Utah*
Scrollytelling examples


slide inspired by: Alexander Lex, Utah
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://bl.ocks.org/mbostock/4699541)
Idiom: **Animated transition + constrained navigation**

- example: icicle plot
  - transition into containing mark causes aspect ratio (shape) change

[Zoomable Icicle](https://bl.ocks.org/mbostock/1005873)
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
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

- aka: brushing and linking

Linked views

• unidirectional vs bidirectional linking

http://www.ralphstraumann.ch/projects/swiss-population-cartogram/

http://peterbeshai.com/linked-highlighting-react-d3-reflux/
Linked views: Multidirectional linking

http://buckets.peterbeshai.com/

https://medium.com/@pbesh/linked-highlighting-with-react-d3-js-and-reflux-16e9c0b2210b
Video: Visual Analysis of Historical Hotel Visitation Patterns

https://www.youtube.com/watch?v=Tzsv6wkZoiQ

http://www.cs.ou.edu/~weaver/improvise/examples/hotels/
Complex linked multiform views

System: Pathfinder

https://www.youtube.com/watch?v=aZF7AC8aNXo
Idiom: **Overview-detail views**

- encoding: same
- data: subset shared
- navigation: shared
  - bidirectional linking

- differences
  - viewpoint
  - (size)

- special case: **birds-eye map**

---

System: **Google Maps**

Idiom: **Overview-detail navigation**

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

https://www.highcharts.com/demo/dynamic-master-detail

https://bl.ocks.org/mbostock/34f08d5e11952a80609169b7917d4172
Overview-detail

- multiscale: three viewing levels
  - linked views
  - dynamic filtering
  - tooling: processing
    (modern version: p5js.org)

System: MizBee

https://www.youtube.com/watch?v=86p7brwuz2g
Overview-detail

https://www.youtube.com/watch?v=UcKDbGqHsdE
Flows: R/Shiny

https://gallery.shinyapps.io/TSupplyDemand/
Idiom: **Parallel sets**

[Diagram of parallel sets showing categories such as Survived vs. Perished, Sex (Female vs. Male), Age (Child vs. Adult), and Class (Second Class, Crew, First Class, Third Class).]

https://www.jasondavies.com/parallel-sets/

https://eagereyes.org/parallel-sets
Idiom: Mosaic plots

System: Mondrian

http://www.theusrus.de/blog/understanding-mosaic-plots/
http://www.theusrus.de/Mondrian/
http://www.theusrus.de/blog/making-movies/
Idiom: Small multiples

- encoding: same
- data: none shared
  - different attributes for node colors
  - (same network layout)
- navigation: shared

Coordinate views: Design choice interaction

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
</tr>
<tr>
<td>Same</td>
<td>Redundant</td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
</tr>
</tbody>
</table>

- **why juxtapose views?**
  - benefits: eyes vs memory
    - lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
  - costs: display area, 2 views side by side each have only half the area of one view
Why not animation?

• disparate frames and regions: comparison difficult
  – vs contiguous frames
  – vs small region
  – vs coherent motion of group

• safe special case
  – animated transitions
System: **Improvise**

- investigate power of multiple views
  - pushing limits on view count, interaction complexity
  - how many is ok?
    - open research question
- reorderable lists
  - easy lookup
  - useful when linked to other encodings

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

• no strict dividing line
  – view: big/detailed
    • contiguous region in which visually encoded data is shown on the display
  – glyph: small/iconic
    • object with internal structure that arises from multiple marks
Partitioning: List alignment

• 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
Partitioning: Recursive subdivision

• split by neighborhood
• then by type
• then time
  – years as rows
  – months as columns
• 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

System: HIVE

Partitioning: Recursive subdivision

- different encoding for second-level regions
  - choropleth maps

Partitioning: Recursive subdivision

• size regions by sale counts
  – not uniformly
• result: treemap

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 achieveable, 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
- “get it right in black and white”
  - check luminance contrast with greyscale view

Superimposing limits

• few layers, but many lines
  – up to a few dozen
  – but not hundreds

• superimpose vs juxtapose: empirical study
  – superimposed for local, multiple for global
  – tasks
    • local: maximum, global: slope, discrimination
  – same screen space for all multiples vs single superimposed

Idiom: Trellis plots

• superimpose within same frame
  – color code by year

• partitioning
  – split by site, rows are wheat varieties

• main-effects ordering
  – derive value of median for group, use to order
  – order rows within view by variety median
  – order views themselves by site median
Dynamic visual layering

• interactive based on selection
• one-hop neighbour highlighting demos: click vs hover (lightweight)

http://mariandoerk.de/edgemaps/demo/
### How?

<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arrange</strong></td>
<td><strong>Change</strong></td>
<td><strong>Juxtapose</strong></td>
<td><strong>Filter</strong></td>
</tr>
<tr>
<td>Express</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Order</strong></td>
<td><strong>Select</strong></td>
<td><strong>Partition</strong></td>
<td><strong>Aggregate</strong></td>
</tr>
<tr>
<td>Align</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Use</strong></td>
<td><strong>Navigate</strong></td>
<td><strong>Superimpose</strong></td>
<td><strong>Embed</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Map** from categorical and ordered attributes
  - Color:
    - Hue
    - Saturation
    - Luminance
  - Size, Angle, Curvature, ...
  - Shape:
    - +
    - ●
    - ■
    - ▲
  - Motion:
    - Direction, Rate, Frequency, ...

**What?**

**Why?**

**How?**
Credits

• Visualization Analysis and Design (Ch 11, 12)
• Alex Lex & Miriah Meyer, http://dataviscourse.net/