Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

Tableau Software
February 20 2015, Seattle WA

http://www.cs.ubc.ca/~tmm/talks.html#vad15tableau
Defining visualization (vis)

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Why?...
Why have a human in the loop?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

• don’t need vis when fully automatic solution exists and is trusted
• many analysis problems ill-specified
  – don’t know exactly what questions to ask in advance
• possibilities
  – long-term use for end users (e.g. exploratory analysis of scientific data)
  – presentation of known results
  – stepping stone to better understanding of requirements before developing models
  – help developers of automatic solution refine/debug, determine parameters
  – help end users of automatic solutions verify, build trust
Why use an external representation?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- external representation: replace cognition with perception

Why represent all the data?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

• summaries lose information, details matter
  – confirm expected and find unexpected patterns
  – assess validity of statistical model

**Anscombe’s Quartet**

<table>
<thead>
<tr>
<th>Identical statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>x mean</td>
</tr>
<tr>
<td>x variance</td>
</tr>
<tr>
<td>y mean</td>
</tr>
<tr>
<td>y variance</td>
</tr>
<tr>
<td>x/y correlation</td>
</tr>
</tbody>
</table>
Analysis framework: Four levels, three questions

• **domain** situation
  – who are the target users?

• **abstraction**
  – translate from specifics of domain to vocabulary of vis

• **what** is shown? **data abstraction**

• **why** is the user looking at it? **task abstraction**

• **idiom**
  • **how** is it shown?
    • **visual encoding idiom**: how to draw
    • **interaction idiom**: how to manipulate

• **algorithm**
  – efficient computation


Validation methods from different fields for each level

- **Domain situation**
  - Observe target users using existing tools

- **Data/task abstraction**
  - **Visual encoding/interaction idiom**
    - Justify design with respect to alternatives
  - **Algorithm**
    - Measure system time/memory
    - Analyze computational complexity
  - Analyze results qualitatively
  - Measure human time with lab experiment (lab study)
  - Observe target users after deployment (field study)
  - Measure adoption

- **Anthropology/ethnography**
  - design
  - computer science
  - cognitive psychology

- mismatch: cannot show idiom good with system timings
- mismatch: cannot show abstraction good with lab study
Why analyze?

- imposes a structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new

---

**Why?**

- **Tree**
  - **Actions**
    - Present
    - Locate
    - Identify
  - **Targets**
    - Path between two nodes

**How?**

- **SpaceTree**
  - Encode
  - Navigate
  - Select
  - Filter
  - Aggregate

- **TreeJuxtaposer**
  - Encode
  - Navigate
  - Select
  - Arrange

---


### Data Types
- **Items**
- **Attributes**
- **Links**
- **Positions**
- **Grids**

### Data and Dataset Types
<table>
<thead>
<tr>
<th>Tables</th>
<th>Networks &amp; Trees</th>
<th>Fields</th>
<th>Geometry</th>
<th>Clusters, Sets, Lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>Items (nodes)</td>
<td>Grids</td>
<td>Items</td>
<td>Positions</td>
</tr>
<tr>
<td>Attributes</td>
<td>Links</td>
<td>Positions</td>
<td>Attributes</td>
<td>Positions</td>
</tr>
</tbody>
</table>

### Dataset Types
- **Tables**
- **Networks**
- **Fields (Continuous)**
- **Multidimensional Table**
- **Trees**
- **Geometry (Spatial)**

### Attribute Types
- **Categorical**
  - +  •  □  △
- **Ordered**
  - Ordinal
  - +  •  □  △
- **Quantitative**

### Ordering Direction
- **Sequential**
- **Diverging**
- **Cyclic**

### Dataset Availability
- **Static**
- **Dynamic**
Dataset and data types

Dataset Types

- Tables
- Networks

Attribute Types

- Categorical
- Ordered
  - Ordinal
  - Quantitative
• \{action, target\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – browse topology
Actions 1: Analyze

- consume
  - discover vs present
    • classic split
    • aka explore vs explain
  - enjoy
    • newcomer
    • aka casual, social

- produce
  - annotate, record
  - derive
    • crucial design choice
Actions II: Search

• what does user know?
  – target, location

<table>
<thead>
<tr>
<th></th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location known</td>
<td>Lookup</td>
<td>Browse</td>
</tr>
<tr>
<td>Location unknown</td>
<td>Locate</td>
<td>Explore</td>
</tr>
</tbody>
</table>
Actions III: Query

• what does user know?
  – target, location

• how much of the data matters?
  – one, some, all

→ Search

<table>
<thead>
<tr>
<th></th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location known</td>
<td>Lookup</td>
<td>Browse</td>
</tr>
<tr>
<td>Location unknown</td>
<td>Locate</td>
<td>Explore</td>
</tr>
</tbody>
</table>

→ Query

→ Identify

→ Compare

→ Summarize
Targets

- All Data
  - Trends
  - Outliers
  - Features

- Attributes
  - One
    - Distribution
    - Extremes
  - Many
    - Dependency
    - Correlation
    - Similarity

- Network Data
  - Topology
    - Paths

- Spatial Data
  - Shape
### How?

#### Encode

- **Arrange**
  - Express
  - Separate
- **Order**
  - Align
- **Use**
  - ![Map Color](image)
  - ![Map Motion](image)
  - ![Map Size, Angle, Curvature](image)
  - ![Map Shape](image)
  - ![Map Motion](image)

- **Map from categorical and ordered attributes**
  - **Color**
    - Hue
    - Saturation
    - Luminance
  - **Size, Angle, Curvature, ...**
  - **Shape**
    - ![Shape](image)
  - **Motion**
    - Direction, Rate, Frequency, ...

#### Manipulate

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

#### Facet

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

#### Reduce

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

---

**What?**  
- ![Map Color](image)
- ![Map Motion](image)
- ![Map Size, Angle, Curvature](image)
- ![Map Shape](image)
- ![Map Motion](image)

**Why?**  
- ![Map Color](image)
- ![Map Motion](image)
- ![Map Size, Angle, Curvature](image)
- ![Map Shape](image)
- ![Map Motion](image)

**How?**
How to encode: Arrange space, map channels

Encode

**Arrange**
- Express
- Order
- Use

**Map**
- from **categorical** and **ordered** attributes
  - Color
    - **Hue**
    - **Saturation**
    - **Luminance**
  - **Size, Angle, Curvature, ...**
  - **Shape**
    - + • ■ ▲
  - **Motion**
    - *Direction, Rate, Frequency, ...*
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>
<tr>
<td><strong>Map</strong></td>
<td><strong>Hue</strong></td>
<td><strong>Direction</strong>, <strong>Rate</strong>, <strong>Frequency</strong>, ...</td>
<td></td>
</tr>
<tr>
<td>from <em>categorical</em> and <em>ordered</em> attributes</td>
<td><strong>Saturation</strong></td>
<td><strong>Embed</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Color</strong></td>
<td><strong>Luminance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Size, Angle, Curvature, ...</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Shape</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Motion</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Direction, Rate, Frequency, ...*
How to handle complexity: 3 more strategies

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

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

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

- **Derive**

+ 1 previous

- change view over time
- facet across multiple views
- reduce items/attributes within single view
- derive new data to show within view
How to handle complexity: 3 more strategies

+ 1 previous

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

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

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

- **Derive**

- • change over time
  - - most obvious & flexible of the 4 strategies
Idiom: **Animated transitions**

- smooth transition from one state to another
  - alternative to jump cuts
  - support for item tracking when amount of change is limited
- example: multilevel matrix views
  - scope of what is shown narrows down
    - middle block stretches to fill space, additional structure appears within
    - other blocks squish down to increasingly aggregated representations

Facet

- Juxtapose

- Partition

- Superimpose

Coordinate Multiple Side By Side Views

- Share Encoding: Same/Different
  - Linked Highlighting

- Share Data: All/Subset/None

- Share Navigation
How to handle complexity: 3 more strategies

- Manipulate
  - Change
  - Select
  - Navigate

- Facet
  - Juxtapose
  - Partition
  - Superimpose

- Reduce
  - Filter
  - Aggregate
  - Embed

- Derive

+ 1 previous

- facet data across multiple views
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

Idiom: *bird’s-eye maps*

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

- differences
  - viewpoint
  - (size)

- **overview-detail**

System: **Google Maps**

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></th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
</tr>
<tr>
<td><strong>Encoding</strong></td>
<td></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
Partition into views

• how to divide data between views
  – encodes association between items using spatial proximity
  – major implications for what patterns are visible
  – split according to attributes

• design choices
  – how many splits
    • all the way down: one mark per region?
    • stop earlier, for more complex structure within region?
  – order in which attribs used to split
  – how many views
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 type
- then by neighborhood
- then time
  - years as rows
  - months as columns

System: HIVE

Partitioning: Recursive subdivision

- switch order of splits
  - neighborhood then type
- very different patterns

Partitioning: Recursive subdivision

- different encoding for second-level regions
  - choropleth maps

How to handle complexity: 3 more strategies

<table>
<thead>
<tr>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change</td>
<td>Juxtapose</td>
<td>Filter</td>
</tr>
<tr>
<td>Select</td>
<td>Partition</td>
<td>Aggregate</td>
</tr>
<tr>
<td>Navigate</td>
<td>Superimpose</td>
<td>Embed</td>
</tr>
</tbody>
</table>

- Derive

- reduce what is shown within single view
Reduce items and attributes

• reduce/increase: inverses

• filter
  – pro: straightforward and intuitive
    • to understand and compute
  – con: out of sight, out of mind

• aggregation
  – pro: inform about whole set
  – con: difficult to avoid losing signal

• not mutually exclusive
  – combine filter, aggregate
  – combine reduce, facet, change, derive

Reducing Items and Attributes

- Filter
  - Items
  - Attributes

- Aggregate
  - Items
  - Attributes

- Embed
Idiom: boxplot

- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attribs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown

[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]
**Idiom:** *Dimensionality reduction for documents*

- attribute aggregation
  - derive low-dimensional target space from high-dimensional measured space

**Task 1**

<table>
<thead>
<tr>
<th>In: HD data</th>
<th>Out: 2D data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item 1</td>
<td>Item 1</td>
</tr>
<tr>
<td>Item ...</td>
<td>Item ...</td>
</tr>
<tr>
<td>Item n</td>
<td>Item n</td>
</tr>
</tbody>
</table>

**Task 2**

<table>
<thead>
<tr>
<th>In: 2D data</th>
<th>Out: Scatterplot Clusters &amp; points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item 1</td>
<td>Clusters &amp; points</td>
</tr>
<tr>
<td>Item ...</td>
<td></td>
</tr>
<tr>
<td>Item n</td>
<td></td>
</tr>
</tbody>
</table>

**Task 3**

<table>
<thead>
<tr>
<th>In: Scatterplot Clusters &amp; points</th>
<th>Out: Labels for clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>wombat</td>
<td></td>
</tr>
</tbody>
</table>
### Data Types
- **Items**

### Data and Datasets
- **Tables**
- **Items**
- **Attributes**

### Actions
- **Analyze**
  - **Consume**
    - **Discover**
    - **Present**
    - **Enjoy**

### All Data
- **Trends**
- **Outliers**
- **Features**

### Why?
- **Actions**
- **Targets**

### How?
- **Encode**
  - **Arrange**
    - **Express**
    - **Separate**
  - **Map**
    - **Color**
      - **Hue**
      - **Saturation**
      - **Luminance**
    - **Size, Angle, Curvature, ...**
    - **Shape**
    - **Motion**
      - **Direction, Rate, Frequency, ...**
  - **Facet**
    - **Select**
    - **Partition**
    - **Aggregate**
  - **Reduce**
    - **Change**
    - **Juxtapose**
    - **Filter**
    - **Embed**

### Domain
- **abstraction**
- **idiom**
- **algorithm**
More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad15tableau

• book page (including tutorial lecture slides)
  http://www.cs.ubc.ca/~tmm/vadbook
  – 20% promo code for book+ebook combo:
    HVN17

  – illustrations: Eamonn Maguire

• papers, videos, software, talks, full courses
  http://www.cs.ubc.ca/group/infovis
  http://www.cs.ubc.ca/~tmm

Visualization Analysis and Design.