Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

PacificVis 2016 Keynote
April 20 2016, Taipei Taiwan

http://www.cs.ubc.ca/~tmm/talks.html#vad16pacvis

@tamaramunzner
Why talk about a textbook to a room of experts?

• convince you of the value in thinking systematically about vis design
  – decompose into comprehensive framework of principles and design choices
  – situate specific examples within framework as concrete illustrations

• provide unified view that crosscuts entire field of visualization
  – infovis and scivis: addressing different kinds of data
  – visual analytics: interweave data analysis and transformation with interactive visual exploration
  – caveat: my own background in infovis shines through!
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**
      - often don’t just draw what you’re given: transform to new form
    - **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


Why is validation difficult?

- different ways to get it wrong at each level

**Domain situation**
- You misunderstood their needs

**Data/task abstraction**
- You’re showing them the wrong thing

**Visual encoding/interaction idiom**
- The way you show it doesn’t work

**Algorithm**
- Your code is too slow
Why is validation difficult?

• solution: use methods from different fields at 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*)

- **Anthropology/ethnography**
  - Observe target users after deployment (*field study*)
  - Measure adoption

- **Design**

- **Computer science**

- **Cognitive psychology**

- **Anthropology/ethnography**

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

<table>
<thead>
<tr>
<th>What?</th>
<th>Why?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tree</td>
<td>Actions</td>
<td>SpaceTree</td>
</tr>
<tr>
<td></td>
<td>Present</td>
<td>Encode</td>
</tr>
<tr>
<td></td>
<td>Locate</td>
<td>Navigate</td>
</tr>
<tr>
<td></td>
<td>Identify</td>
<td>Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Filter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Aggregate</td>
</tr>
<tr>
<td>Tree</td>
<td>Targets</td>
<td>TreeJuxtaposer</td>
</tr>
<tr>
<td></td>
<td>Path between two nodes</td>
<td>Encode</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Navigate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Select</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Arrange</td>
</tr>
</tbody>
</table>


Why?

What?

How?

Datasets

Attributes

Data Types

Attributes

Links

Positions

Grids

Data and Dataset Types

Tables

Networks & Trees

Fields

Geometry

Clusters, Sets, Lists

Data Types

Attributes

Links

Positions

Grids

Dataset Types

Multidimensional Table

Networks

Trees

Fields (Continuous)

Dataset Availability

Static

Dynamic

Attribute Types

Categorical

Ordered

Ordinal

Quantitative

Ordering Direction

Sequential

Diverging

Cyclic
Types: Datasets and data

➡️ Dataset Types

➡️ Tables

➡️ Networks

➡️ Spatial

➡️ Fields (Continuous)

➡️ Geometry (Spatial)

➡️ Attribute Types

➡️ Categorical

➡️ Ordered

➡️ Ordinal

➡️ Quantitative
• \{\text{action, target}\} pairs
  – \textit{discover distribution}
  – \textit{compare trends}
  – \textit{locate outliers}
  – \textit{browse topology}
Actions: Analyze, Query

- analyze
  - consume
    - discover vs present
      - aka explore vs explain
  - enjoy
    - aka casual, social
- produce
  - annotate, record, derive
- query
  - how much data matters?
    - one, some, all
- independent choices
  - analyze, query, (search)
Derive: Crucial Design Choice

• don’t just draw what you’re given!
  – decide what the right thing to show is
  – create it with a series of transformations from the original dataset
  – draw that

• one of the four major strategies for handling complexity

![Original Data](chart1)

![Derived Data](chart2)

\[ \text{trade balance} = \text{exports} - \text{imports} \]
Analysis example: Derive one attribute

- Strahler number
  - centrality metric for trees/networks
  - derived quantitative attribute
  - draw top 5K of 500K for good skeleton


Task 1

In Tree ➔ Out Quantitative attribute on nodes

What? ➔ In Tree ➔ Out Quantitative attribute on nodes
Why? ➔ Derive

Task 2

In Tree + In Quantitative attribute on nodes ➔ Out Filtered Tree

What? ➔ In Tree ➔ In Quantitative attribute on nodes ➔ Out Filtered Tree
Why? ➔ Summarize ➔ Topology
How? ➔ Reduce ➔ Filter

Removed unimportant parts
Targets

- All Data
  - Trends
  - Outliers
  - Features

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

- Network Data
  - Topology
    - Paths

- Spatial Data
  - Shape
<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Arrange</strong>&lt;br&gt;→ Express&lt;br&gt;→ Separate</td>
<td><strong>Map</strong>&lt;br&gt;from <em>categorical</em> and <em>ordered</em> attributes</td>
<td><strong>Change</strong>&lt;br&gt;→ Hue&lt;br&gt;→ Saturation&lt;br&gt;→ Luminance</td>
<td><strong>Filter</strong>&lt;br&gt;→ Filter</td>
</tr>
<tr>
<td><strong>Order</strong>&lt;br&gt;→ Align</td>
<td><strong>Color</strong>&lt;br&gt;→ Size, Angle, Curvature, ...</td>
<td><strong>Select</strong>&lt;br&gt;→ Navigate</td>
<td><strong>Aggrega</strong>&lt;br&gt;→ Aggregate</td>
</tr>
<tr>
<td><strong>Use</strong></td>
<td><strong>Shape</strong>&lt;br&gt;→ + ● ■ ▲</td>
<td><strong>Partition</strong>&lt;br&gt;→ Superimpose</td>
<td><strong>Embed</strong>&lt;br&gt;→ Embed</td>
</tr>
<tr>
<td></td>
<td><strong>Motion</strong>&lt;br&gt;→ Direction, Rate, Frequency, ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**What?**

**Why?**

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

Encode

 misled

 Arrange

 ➔ Express
 ➔ Order
 ➔ Use

 ➔ Separate
 ➔ Align

 ➔ Map

from categorical and ordered attributes

 ➔ Color
 ➔ Hue ➔ Saturation ➔ Luminance

 ➔ Size, Angle, Curvature, ...

 ➔ Shape

 ➔ Motion

Direction, Rate, Frequency, ...
Definitions: Marks and channels

- **marks**
  - geometric primitives

- **channels**
  - control appearance of marks
Encoding visually with marks and channels

• analyze idiom structure
  – as combination of marks and channels

1: vertical position
mark: line

2: vertical position
horizontal position
mark: point

3: vertical position
horizontal position
color hue
mark: point

4: vertical position
horizontal position
color hue
size (area)
mark: point
# Channels

<table>
<thead>
<tr>
<th>Ordered Attributes</th>
<th>Categorical Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position on common scale</td>
<td>Spatial region</td>
</tr>
<tr>
<td>Position on unaligned scale</td>
<td>Color hue</td>
</tr>
<tr>
<td>Length (1D size)</td>
<td>Motion</td>
</tr>
<tr>
<td>Tilt/angle</td>
<td>Shape</td>
</tr>
<tr>
<td>Area (2D size)</td>
<td></td>
</tr>
<tr>
<td>Depth (3D position)</td>
<td></td>
</tr>
<tr>
<td>Color luminance</td>
<td></td>
</tr>
<tr>
<td>Color saturation</td>
<td></td>
</tr>
<tr>
<td>Curvature</td>
<td></td>
</tr>
<tr>
<td>Volume (3D size)</td>
<td></td>
</tr>
</tbody>
</table>

**Identity Channels:**
- Spatial region
- Color hue
- Motion
- Shape

**Ordered Attributes:**
- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)
Channels: Matching Types

**Magnitude Channels: Ordered Attributes**
- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)

**Identity Channels: Categorical Attributes**
- Spatial region
- Color hue
- Motion
- Shape

- expressiveness principle
  - match channel and data characteristics
Channels: Rankings

Magnitude Channels: Ordered Attributes
- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)

Identity Channels: Categorical Attributes
- Spatial region
- Color hue
- Motion
- Shape

- expressiveness principle
  - match channel and data characteristics
- effectiveness principle
  - encode most important attributes with highest ranked channels
## How?

### Encode

- **Arrange**
  - Express
  - Separate

- **Order**
  - Align

- **Use**

### Manipulate

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

- **Color**
  - Hue
  - Saturation
  - Luminance

- **Size, Angle, Curvature, ...**

- **Shape**
  - +
  - ●
  - □
  - △

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

### Facet

- **Change**

- **Select**

- **Navigate**

### Reduce

- **Filter**

- **Partition**

- **Superimpose**

- **Aggregate**

- **Embed**
How to handle complexity: 3 more strategies

- Manipulate
  - Change
  - Select
  - Navigate

- Facet
  - Juxtapose
  - Partition
  - Superimpose

- Reduce
  - Filter
  - Aggregate
  - Embed

- Derive

• 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

- **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

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
Facet

- **Juxtapose**

- **Partition**

- **Superimpose**

---

**Coordinate Multiple Side By Side 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

Idiom: bird’s-eye maps

• encoding: same
• data: subset shared
• navigation: shared  
  – bidirectional linking

• differences  
  – viewpoint  
  – (size)

• overview-detail

**Idiom: Small multiples**

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

**System: Cerebral**

### Coordinate views: Design choice interaction

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

Partition into Side-by-Side 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 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

Partitioning: Recursive subdivision

• different encoding for second-level regions
  – choropleth maps

How to handle complexity: 3 more strategies

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

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

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

+ 1 previous

- 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
**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**

In HD data ➔ Out 2D data

**Task 2**

In 2D data ➔ Out Scatterplot Clusters & points

**Task 3**

In Scatterplot Clusters & points ➔ Out Labels for clusters
More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad16pacvis

• 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