Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

University of Illinois at Chicago, Distinguished Lecture Series
April 9 2015, Chicago IL

http://www.cs.ubc.ca/~tmm/talks.html#vad15chicago
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>Identiﬁcal statistics</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x mean</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>x variance</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>y mean</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>y variance</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>x/y correlation</td>
<td>1</td>
<td></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


[A Multi-Level Typology of Abstract Visualization Tasks
Validation methods from different fields for each level

<table>
<thead>
<tr>
<th>Domain situation</th>
<th>Data/task abstraction</th>
<th>Design</th>
<th>Computer science</th>
<th>Cognitive psychology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Observe target users using existing tools</td>
<td>Visual encoding/interaction idiom</td>
<td>Design</td>
<td>Visual encoding/interaction idiom</td>
<td>Design</td>
</tr>
<tr>
<td>Justify design with respect to alternatives</td>
<td>Algorithm</td>
<td>Computer science</td>
<td>Algorithm</td>
<td>Computer science</td>
</tr>
<tr>
<td>Measure system time/memory</td>
<td>Analyze computational complexity</td>
<td>Cognitive psychology</td>
<td>Analyze computational complexity</td>
<td>Cognitive psychology</td>
</tr>
<tr>
<td>Analyze results qualitatively</td>
<td>Measure human time with lab experiment ((lab\ study))</td>
<td>Design</td>
<td>Measure human time with lab experiment ((lab\ study))</td>
<td>Design</td>
</tr>
<tr>
<td>Observe target users after deployment ((field\ study))</td>
<td>Measure adoption</td>
<td>Computer science</td>
<td>Measure adoption</td>
<td>Computer science</td>
</tr>
<tr>
<td>Measure adoption</td>
<td></td>
<td>Cognitive psychology</td>
<td>Measure adoption</td>
<td>Cognitive psychology</td>
</tr>
</tbody>
</table>

- 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

SpaceTree

Encode  
Navigate  
Select  
Filter  
Aggregate

TreeJuxtaposer

Encode  
Navigate  
Select  
Arrange

What?

Why?

How?
Datasets

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

Data and Dataset Types
- Tables
  - Items
  - Attributes
- Networks & Trees
  - Items (nodes)
  - Grids
- Fields
  - Positions
- Geometry
  - Items
  - Positions
- Clusters, Sets, Lists
  - Items

Attributes

Attribute Types
- Categorical
- Ordered
  - Ordinal
  - Quantitative

Dataset Types

Dataset Availability
- Static
- Dynamic

Why?

How?
Dataset and data types

**Dataset Types**

- **Tables**

  ![Table Diagram](image)

- **Networks**

  ![Network Diagram](image)

**Attribute Types**

- **Categorical**

  ![Categorical Icons](image)

- **Ordered**

  - **Ordinal**

    ![Ordinal Icons](image)

  - **Quantitative**

    ![Quantitative Icons](image)
• \{action, target\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – browse topology

### Actions

- **Analyze**
  - Consume
    - Discover
    - Present
    - Enjoy
  - Produce
    - Annotate
    - Record
    - Derive

- **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
  - Many
    - Dependency
    - Correlation
    - Similarity

- **Network Data**
  - Topology
    - Paths

- **Spatial Data**
  - Shape
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><img src="image" alt="Look up" /></td>
<td><img src="image" alt="Browse" /></td>
</tr>
<tr>
<td>Location unknown</td>
<td><img src="image" alt="Locate" /></td>
<td><img src="image" alt="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
<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Arrange&lt;br&gt;  ➤ Express&lt;br&gt;  ➤ Separate</td>
<td>- Change&lt;br&gt;  ➤ Map&lt;br&gt;  ➤ from categorical and ordered attributes&lt;br&gt;  ➤ Color&lt;br&gt;  ➤ Hue&lt;br&gt;  ➤ Saturation&lt;br&gt;  ➤ Luminance&lt;br&gt;  ➤ Size, Angle, Curvature, ...&lt;br&gt;  ➤ Shape&lt;br&gt;  ➤ + • □ △</td>
<td>- Juxtapose&lt;br&gt;  ➤ Select&lt;br&gt;  ➤ Partition&lt;br&gt;  ➤ Superimpose</td>
<td>- Filter&lt;br&gt;  ➤ Aggregate&lt;br&gt;  ➤ Embed</td>
</tr>
<tr>
<td>- Order&lt;br&gt;  ➤ Align</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Use</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**How?**

**What?**

- **Why?**

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

Encode

Directions:

- **Arrange**
  - Express
  - Order
  - Use

- **Map**
  - from *categorical* and *ordered* attributes
  - Color
    - Hue
    - Saturation
    - Luminance
  - Size, Angle, Curvature, ...
  - Shape
    - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:

- Color
  - Hue
  - Saturation
  - Luminance
- Shape
  - Direction, Rate, Frequency, ...

Legend:
Encoding visually

• analyze idiom structure
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

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)

Spatial region

Color hue

Motion

Shape

Same
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?

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

#### What?
- **Map**
  - from *categorical* and *ordered* attributes
  - *Color*
    - Hue
  - *Size, Angle, Curvature, ...*
  - *Shape*
    - + • ■ △
  - *Motion*
    - *Direction, Rate, Frequency, ...*

#### Why?

#### How?
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

- **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**

- 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

System: Cerebral

### 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></td>
<td>Subset</td>
</tr>
<tr>
<td></td>
<td>None</td>
</tr>
<tr>
<td>Same</td>
<td>Redundant</td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
</tr>
<tr>
<td></td>
<td>Overview/Detail</td>
</tr>
<tr>
<td></td>
<td>Small Multiples</td>
</tr>
<tr>
<td></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
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

System: HIVE

Partitioning: Recursive subdivision

- different encoding for second-level regions
  – choropleth maps

System: HIVE

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>

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

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

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

- 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 attributes
    - 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>What?</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>In High-dimensional data</td>
<td>Produce Derive</td>
</tr>
<tr>
<td>Out 2D data</td>
<td></td>
</tr>
</tbody>
</table>

**Task 2**

<table>
<thead>
<tr>
<th>What?</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>In 2D data</td>
<td>Discover Identify</td>
</tr>
<tr>
<td>Out Scatterplot Clusters &amp; points</td>
<td>Explore Navigate Select</td>
</tr>
</tbody>
</table>

**Task 3**

<table>
<thead>
<tr>
<th>What?</th>
<th>Why?</th>
</tr>
</thead>
<tbody>
<tr>
<td>In Scatterplot Clusters &amp; points</td>
<td>Produce Annotate</td>
</tr>
<tr>
<td>Out Labels for clusters</td>
<td></td>
</tr>
</tbody>
</table>

- attribute aggregation
More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad15chicago

• 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