Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

UBC STAT 545A Guest Lecture
October 20 2016, Vancouver BC

http://www.cs.ubc.ca/~tmm/talks.html#vad16bryan
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>
Why analyze?

• imposes structure on huge design space
  – scaffold to help you think systematically about choices
  – analyzing existing as stepping stone to designing new
  – most possibilities ineffective for particular task/data combination


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

Datasets

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

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

Attributes

- Attribute Types
  - Categorical
    - +
    - ●
    - □
    - ▲
  - Ordered
    - Ordinal
    - ↑
  - Quantitative

Ordering Direction

- Sequential
- Diverging
- Cyclic

Geometry (Spatial)

Dataset Availability

- Static
- Dynamic
Three major datatypes

Dataset Types

- Tables
  - Attributes (columns)
  - Items (rows)
  - Cell containing value

- Networks
  - Link
  - Node (item)

- Spatial
  - Fields (Continuous)
  - Geometry (Spatial)

- Multidimensional Table
  - Key 1
  - Key 2
  - Attributes

- Trees
  - Attributes (columns)
  - Value in cell

- Spatial
  - Grid of positions
  - Cell
  - Attributes (columns)
  - Value in cell

- Position

• visualization vs computer graphics
  – geometry is design decision
Types: Datasets and data

- **Dataset Types**
  - Tables
  - Networks
  - Spatial
    - Fields (Continuous)
    - Geometry (Spatial)

- **Attribute Types**
  - Categorical
  - Ordered
    - Ordinal
    - Quantitative
• \{action, target\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – 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

• 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

![Original Data Graph]

Derived Data

![Derived Data Graph]

\[ \text{trade balance} = \text{exports} - \text{imports} \]
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**

- **What?**
  - In Tree
  - Out Quantitative attribute on nodes

- **Why?**
  - Derive

**Task 2**

- **What?**
  - In Tree
  - Quantitative attribute on nodes

- **Why?**
  - Summarize

- **How?**
  - Reduce
  - Filter

Out Filtered Tree
Removed unimportant parts

In Tree +
Out Quantitative attribute on nodes

In Tree
Out Filtered Tree
**Targets**

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

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

- **Network Data**
  - Topology
    - Paths

- **Spatial Data**
  - Shape
## How?

<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrange</td>
<td>Map</td>
<td>Change</td>
<td>Filter</td>
</tr>
<tr>
<td>Express</td>
<td>from categorical and ordered attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Separate</td>
<td>Color</td>
<td>Juxtapose</td>
<td></td>
</tr>
<tr>
<td>Order</td>
<td>Hue, Saturation, Luminance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Align</td>
<td>Size, Angle, Curvature, ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use</td>
<td>Shape</td>
<td>Select</td>
<td>Aggregate</td>
</tr>
<tr>
<td></td>
<td>+, ●, ■, ▲</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Motion</td>
<td>Partition</td>
<td>Embed</td>
</tr>
<tr>
<td></td>
<td>Direction, Rate, Frequency, ...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### What?

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

### Why?

- Encode
- Arrange
- Express
- Separate
- Order
- Align
- Use

### How?

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

- Change
- Juxtapose
- Select
- Partition
- Navigate
- Superimpose
- Embed
How to encode: Arrange space, map channels

Encode

Arrange

- Express
- Order
- Use

Separate

Align

Map

from categorical and ordered attributes

- Color
  - Hue
  - Saturation
  - Luminance

- Size, Angle, Curvature, ...

- Shape

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

En code
A rrange
Exp r ess
S ep a ra te
O r der A li g n
U se
M ap
C ol or
M ot ion
S i ze , A ng le , C ur va tu r e , ...
Hue S at ur at ion L umi na nc e
S h a pe
D i re c tion, R a te , F re quen cy , ...
F r om c a te gor i cal a nd o r der ed a t t r ibu t es
Encoding visually

• analyze idiom structure
Definitions: Marks and channels

• marks
  – geometric primitives

  • Points
  • Lines
  • Areas

• channels
  – control appearance of marks

  • Position
    ➔ Horizontal
    ➔ Vertical
    ➔ Both

  • Color

  • Shape

  • Tilt

  • Size
    ➔ Length
    ➔ Area
    ➔ Volume
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
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
Accuracy: Fundamental Theory

Steven’s Psychophysical Power Law: $S = I^N$
Accuracy: Vis experiments

## Separability vs. Integrality

<table>
<thead>
<tr>
<th>Position</th>
<th>Size</th>
<th>Width</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Hue (Color)</td>
<td>+ Hue (Color)</td>
<td>+ Height</td>
<td>+ Green</td>
</tr>
</tbody>
</table>

- **Fully separable**
  - 2 groups each

- **Some interference**
  - 2 groups each

- **Some/significant interference**
  - 3 groups total: integral area

- **Major interference**
  - 4 groups total: integral hue
Grouping

• containment
• connection

Marks as Links

→ Containment

→ Connection

Identity Channels: Categorical Attributes

• proximity
  – same spatial region
• similarity
  – same values as other categorical channels
How to encode: Arrange position and region

Encode

Arrange
- Express
- Order
- Use

Separate
- Align

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

Why?
How?
What?
Arrange tables

- Express Values
- Separate, Order, Align Regions
  - Separate
  - Order
  - Align

- Axis Orientation
  - Rectilinear
  - Parallel
  - Radial

- Layout Density
  - Dense
  - Space-Filling

- 1 Key
  - List
- 2 Keys
  - Matrix
- 3 Keys
  - Volume
- Many Keys
  - Recursive Subdivision
Idioms: dot chart, line chart

• one key, one value
  – data
    • 2 quant attribs
  – mark: points
    • dot plot: + line connection marks between them
  – channels
    • aligned lengths to express quant value
    • separated and ordered by key attrib into horizontal regions
  – task
    • find trend
      – connection marks emphasize ordering of items along key axis by explicitly showing relationship between one item and the next
Choosing bar vs line charts

- depends on type of key attrib
  - bar charts if categorical
  - line charts if ordered
- do not use line charts for categorical key attribs
  - violates expressiveness principle
    - implication of trend so strong that it overrides semantics!
      - “The more male a person is, the taller he/she is”

Idiom: **glyphmaps**

- rectilinear good for linear vs nonlinear trends

- radial good for cyclic patterns

Idiom: **heatmap**

- **two keys, one value**
  - **data**
    - 2 categ attribs (gene, experimental condition)
    - 1 quant attrib (expression levels)
  - **marks: area**
    - separate and align in 2D matrix
      - indexed by 2 categorical attributes
  - **channels**
    - color by quant attrib
      - (ordered diverging colormap)
  - **task**
    - find clusters, outliers
  - **scalability**
    - 1M items, 100s of categ levels, ~10 quant attrib levels
Arrange spatial data

Use Given

Geometry

- Geographic
- Other Derived

Spatial Fields

- Scalar Fields (one value per cell)
  - Isocontours
  - Direct Volume Rendering

- Vector and Tensor Fields (many values per cell)
  - Flow Glyphs (local)
  - Geometric (sparse seeds)
  - Textures (dense seeds)
  - Features (globally derived)
Idiom: choropleth map

- use given spatial data
  - when central task is understanding spatial relationships
- data
  - geographic geometry
  - table with 1 quant attribute per region
- encoding
  - use given geometry for area mark boundaries
  - sequential segmented colormaps

http://bl.ocks.org/mbostock/4060606
Population maps trickiness

- beware!

[https://xkcd.com/1138]
Idiom: **topographic map**

- **data**
  - geographic geometry
  - scalar spatial field
    - 1 quant attribute per grid cell
- **derived data**
  - isoline geometry
    - isocontours computed for specific levels of scalar values
Idioms: **isosurfaces, direct volume rendering**

- **data**
  - scalar spatial field
    - 1 quant attribute per grid cell
- **task**
  - shape understanding, spatial relationships
- **isosurface**
  - derived data: isocontours computed for specific levels of scalar values
- **direct volume rendering**
  - transfer function maps scalar values to color, opacity

Idiom: similarity-clustered streamlines

- data
  - 3D vector field
- derived data (from field)
  - streamlines: trajectory particle will follow
- derived data (per streamline)
  - curvature, torsion, tortuosity
  - signature: complex weighted combination
  - compute cluster hierarchy across all signatures
  - encode: color and opacity by cluster
- tasks
  - find features, query shape
- scalability
  - millions of samples, hundreds of streamlines

Arrange networks and trees

- **Node–Link Diagrams**
  - Connection Marks
  - ![Node–Link Diagram](image)

- **Adjacency Matrix**
  - Derived Table
  - ![Adjacency Matrix](image)

- **Enclosure**
  - Containment Marks
  - ![Enclosure](image)
Idiom: **force-directed placement**

- visual encoding
  - link connection marks, node point marks
- considerations
  - spatial position: no meaning directly encoded
    - left free to minimize crossings
  - proximity semantics?
    - sometimes meaningful
    - sometimes arbitrary, artifact of layout algorithm
    - tension with length
      - long edges more visually salient than short
- tasks
  - explore topology; locate paths, clusters
- scalability
  - node/edge density $E < 4N$

[Image: mbostock.github.com/d3/ex/force.html]
Idiom: **adjacency matrix view**

- **data**: network  
  - transform into same data/encoding as heatmap
- **derived data**: table from network  
  - 1 quant attrib  
    - weighted edge between nodes  
  - 2 categ attribs: node list x 2
- **visual encoding**  
  - cell shows presence/absence of edge
- **scalability**  
  - 1K nodes, 1M edges

![Diagram of adjacency matrix view](image)

**Figure 7.5**: Comparing matrix and node-link views of a five-node network.  
(a) Matrix view. (b) Node-link view. From [Henry et al. 07], Figure 3b and 3a.  
(Permission needed.)

Matrix views of networks can achieve very high information density, up to a limit of one thousand nodes and one million edges, just like cluster heatmaps and all other matrix views that uses small area marks.

**Technique**  
**network matrix view**  
**Data Types**  
**network**  
**Derived Data**  
| table: network nodes as keys, link status between two nodes as values |

**View Comp.**  
**space**: area marks in 2D matrix alignment

**Scalability**  
**nodes**: 1K  
**edges**: 1M

---

7.1.3.3 Multiple Keys: Partition and Subdivide  
When a dataset has only one key, then it is straightforward to use that key to separate into one region.
Connection vs. adjacency comparison

• adjacency matrix strengths
  – predictability, scalability, supports reordering
  – some topology tasks trainable

• node-link diagram strengths
  – topology understanding, path tracing
  – intuitive, no training needed

• empirical study
  – node-link best for small networks
  – matrix best for large networks
    • if tasks don’t involve topological structure!

Idiom: **radial node-link tree**

- data
  - tree
- encoding
  - link connection marks
  - point node marks
  - radial axis orientation
    - angular proximity: siblings
    - distance from center: depth in tree
- tasks
  - understanding topology, following paths
- scalability
  - 1K - 10K nodes

Idiom: treemap

• data
  – tree
  – 1 quant attrib at leaf nodes

• encoding
  – area containment marks for hierarchical structure
  – rectilinear orientation
  – size encodes quant attrib

• tasks
  – query attribute at leaf nodes

• scalability
  – 1M leaf nodes

Connection vs. containment comparison

• marks as links (vs. nodes)
  – common case in network drawing
  – 1D case: connection
    • ex: all node-link diagrams
    • emphasizes topology, path tracing
    • networks and trees
  – 2D case: containment
    • ex: all treemap variants
    • emphasizes attribute values at leaves (size coding)
    • only trees

How to encode: Mapping color

Encode

→ Arrange
  → Express
  → Order
  → Use
→ Separate
→ Align

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

Why?

How?

What?
Color: Luminance, saturation, hue

• 3 channels
  – identity for categorical
    • hue
  – magnitude for ordered
    • luminance
    • saturation

• RGB: poor for encoding
• HSL: better, but beware
  – lightness ≠ luminance
Categorical color: Discriminability constraints

- noncontiguous small regions of color: only 6-12 bins

[Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. BMC Bioinformatics, 8:82, 2007.]
Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable


Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – large-scale structure: fewer hues

Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – large-scale structure: fewer hues
  – fine structure: multiple hues with monotonically increasing luminance [eg viridis R/python]
Viridis

- colorful, perceptually uniform, colorblind-safe, monotonically increasing luminance

https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html
Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – large-scale structure: fewer hues
  – fine structure: multiple hues with monotonically increasing luminance [e.g., viridis R/python]
  – segmented rainbows for binned or categorical


### 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><strong>Hue</strong></td>
<td><strong>Partition</strong></td>
<td><strong>Aggregate</strong></td>
</tr>
<tr>
<td>Separate</td>
<td><strong>Saturation</strong></td>
<td><strong>Superimpose</strong></td>
<td><strong>Embed</strong></td>
</tr>
<tr>
<td>Order</td>
<td><strong>Luminance</strong></td>
<td><strong>Direction, Rate, Frequency, ...</strong></td>
<td><strong>Embed</strong></td>
</tr>
<tr>
<td>Align</td>
<td><strong>Size, Angle, Curvature, ...</strong></td>
<td><strong>Superimpose</strong></td>
<td><strong>Embed</strong></td>
</tr>
<tr>
<td>Use</td>
<td><strong>Shape</strong></td>
<td><strong>Navigator</strong></td>
<td><strong>Embed</strong></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?**
How to handle complexity: 3 more strategies

**Manipulate**

- Change
  - \[ \text{data} \]

- Select
  - \[ \text{select} \]

- Navigate
  - \[ \text{navigate} \]

**Facet**

- Juxtapose
  - \[ \text{data} \]

- Partition
  - \[ \text{partition} \]

- Superimpose
  - \[ \text{superimpos} \]

**Reduce**

- Filter
  - \[ \text{filter} \]

- Aggregate
  - \[ \text{aggregate} \]

- Embed
  - \[ \text{embed} \]

**Derive**

- \[ \text{derive} \]

---

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

Data and D
- Tables
- Items
- Attributes

Dataset Types
- Tables
- Attribute (rows)

Search
- Multidimensional
- Location known
- Location unknown

Query
- Geospatial
- Identify

Encode
- Arrange
  - Express
  - Separate
- Map
  - from categorical and ordered attributes
  - Color
    - Hue
    - Saturation
    - Luminance
  - Size, Angle, Curvature, ...
  - Shape
    - + • ◇
  - Motion
    - Direction, Rate, Frequency, ...

All Data
- Trends
- Outliers
- Features

Manipulate
- Change
- Juxtapose
- Filter

Facet
- Select
- Partition

Reduce
- Aggregate
- Superimpose
- Embed

Actions

Targets

Why?

domain

abstraction What?

idiom How?

algorithm Why?
More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad16bryan

• 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

• grad vis course Jan ’17: CPSC 547, Tue/Thu 3:30
  - students from outside CS are welcome
  http://www.cs.ubc.ca/~tmm/courses/547-17

@tamaramunzner