Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

Guest Lecture, STAT 545
Oct 19 2017, Vancouver BC

www.cs.ubc.ca/~tmm/talks.html#vad17stat545  @tamaramunzner
Visualization (vis) defined & motivated

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.

• human in the loop needs the details
  – doesn't know exactly what questions to ask in advance
  – longterm exploratory analysis
    • speed up through human-in-the-loop visual data analysis
  – presentation of known results
  – stepping stone towards automation: refining, trustbuilding

• intended task, measurable definitions of effectiveness
Nested model: Four levels of vis design

- **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


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

[In a nested model of visualization design and validation, Munzner. IEEE TVCG 15(6):921-928, 2009 (Proc. InfoVis 2009).]

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
Anthropology/ethnography
**What?**

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

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

- **Attributes**
  - **Attribute Types**
    - Categorical
    - Ordered
    - Ordinal
    - Quantitative

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

- **Geometry (Spatial)**

- **Dataset Availability**
  - Static
  - Dynamic

**Why?**

**How?**
Types: Datasets and data

➡️ Dataset Types
➡️ Tables
➡️ Networks

➡️ Attribute Types
➡️ Categorical
➡️ Ordered
➡️ Ordinal
➡️ Quantitative

➡️ Ordering Direction
➡️ Sequential
➡️ Diverging
➡️ Cyclic
• \{\text{action, target}\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – browse topology
Actions: Analyze

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

• produce
  – annotate, record
  – derive
    • crucial design choice

Analyze

Consume

Present

Enjoy

Produce

Annotate

Record

Derive
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

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

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
<th>Quantitative attribute on nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**Why?**
- Derive

### Task 2

<table>
<thead>
<tr>
<th>In</th>
<th>Out</th>
<th>Quantitative attribute on nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**Why?**
- Summarize Topology
- Reduce
- Filter

**How?**
- Filtered Tree
- Removed unimportant parts
**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)
Why: 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 from categorical and ordered attributes
- Color
  - Hue
  - Saturation
  - Luminance
- Size, Angle, Curvature, ...
- Shape
- Motion
  Direction, Rate, Frequency, ...

Manipulate
- Change
- Select
- Navigate

Facet
- Juxtapose
- Partition
- Superimpose

Reduce
- Filter
- Aggregate
- Embed

What?

Why?

How?
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, ...
Definitions: Marks and channels

• marks
  – geometric primitives

• channels
  – control appearance of marks
  – can redundantly code with multiple channels
Visual encoding

• 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
Channels: Expressiveness types and effectiveness 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
  - spatial position ranks high for both
Accuracy: Fundamental Theory

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

Discriminability: How many usable steps?

• must be sufficient for number of attribute levels to show
  – linewidth: few bins but salient

[mappa.mundi.net/maps/maps 014/telegeography.html]
Separability vs. Integrality

Position
- Hue (Color)

Size
- Hue (Color)

Width
- Height

Red
- Green

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

Spatial region

Color hue

Motion

Shape

proximity
  - same spatial region

similarity
  - same values as other categorical channels
Idiom design choices: Encode

**Why?**

**How?**

**What?**

**Arrange**

→ Express

→ Order

→ Use

**Map**

from categorical and ordered attributes

→ Color

→ Hue
→ Saturation
→ Luminance

→ Size, Angle, Curvature, ...

→ Shape

→ Motion

*Direction, Rate, Frequency, ...*
Categorical vs ordered color

Decomposing color

• first rule of color: do not talk about color!
  – color is confusing if treated as monolithic

• decompose into three channels
  – ordered can show magnitude
    • luminance: how bright
    • saturation: how colorful
  – categorical can show identity
    • hue: what color

• channels have different properties
  – what they convey directly to perceptual system
  – how much they can convey: how many discriminable bins can we use?
Categorical color: limited number of discriminable bins

• human perception built on relative comparisons
  – great if color contiguous
  – surprisingly bad for absolute comparisons

• noncontiguous small regions of color
  – fewer bins than you want
  – rule of thumb: 6-12 bins, including background and highlights

  alternatives? this afternoon!

[Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. BMC Bioinformatics, 8:82, 2007.]
ColorBrewer

- [http://www.colorbrewer2.org](http://www.colorbrewer2.org)
- saturation and area example: size affects salience!
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 [eg viridis R/python]
  – segmented rainbows for binned or categorical


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**

- **Aggregate**

- **Embed**

What?

Why?

How?
Encode tables: Arrange space

Arrange

Express

Separate

Order

Align
Arrange tables

Express Values

Separate, Order, Align Regions

Separate
Order
Align

Axis Orientation

Rectilinear
Parallel
Radial

Layout Density

Dense
Space-Filling

1 Key
2 Keys
3 Keys
Many Keys

List
Recursive Subdivision

Matrix
Volume

1 Key
2 Keys
3 Keys
Many Keys

Recursive Subdivision

1 Key
2 Keys
3 Keys
Many Keys

List
Recursive Subdivision

Matrix
Volume

1 Key
2 Keys
3 Keys
Many Keys

Recursive Subdivision

Matrix
Volume

1 Key
2 Keys
3 Keys
Many Keys

Recursive Subdivision
Keys and values

- **key**
  - independent attribute
  - used as unique index to look up items
  - simple tables: 1 key
  - multidimensional tables: multiple keys

- **value**
  - dependent attribute, value of cell

- classify arrangements by key count
  - 0, 1, 2, many...

Express Values

- 1 Key
  - List
- 2 Keys
  - Matrix
- 3 Keys
  - Volume
- Many Keys
  - Recursive Subdivision
Express Values

1 Key
List

2 Keys
Matrix

3 Keys
Volume

Many Keys
Recursive Subdivision

0 Keys
**Idiom: scatterplot**

- **express values**
  - quantitative attributes
- **no keys, only values**
  - data
    - 2 quant attrs
  - mark: points
  - channels
    - horiz + vert position
- **tasks**
  - find trends, outliers, distribution, correlation, clusters
- **scalability**
  - hundreds of items

Some keys

Express Values

1 Key
List

2 Keys
Matrix

3 Keys
Volume

Many Keys
Recursive Subdivision
Some keys: Categorical regions

- **regions**: contiguous bounded areas distinct from each other
  - using space to *separate* (proximity)
  - following expressiveness principle for categorical attributes
- use ordered attribute to *order* and *align* regions

- **1 Key**
  - List
- **2 Keys**
  - Matrix
- **3 Keys**
  - Volume
- **Many Keys**
  - Recursive Subdivision
**Idiom: bar chart**

- one key, one value
  - data
    - 1 categ attrib, 1 quant attrib
  - mark: lines
  - channels
    - length to express quant value
    - spatial regions: one per mark
      - separated horizontally, aligned vertically
      - ordered by quant attrib
        » by label (alphabetical), by length attrib (data-driven)
- task
  - compare, lookup values
- scalability
  - dozens to hundreds of levels for key attrib
LIMITATION: Hard to know rank. What’s the 4\textsuperscript{th} most? The 7\textsuperscript{th}?

[Slide courtesy of Ben Jones]
Separated, Aligned and Ordered

[Slide courtesy of Ben Jones]
Separated but not Ordered or Aligned

LIMITATION: Hard to make comparisons

[Slide courtesy of Ben Jones]
Idiom: **line chart / dot plot**

- one key, one value
  - data
    - 2 quant attribs
  - mark: points
    - 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
- scalability
  - hundreds of key levels, hundreds of value levels
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”

Express Values ➔ 1 Key List ➔ 2 Keys Matrix ➔ 3 Keys Volume ➔ Many Keys Recursive Subdivision

2 Keys
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
Axis Orientation

- Rectilinear
- Parallel
- Radial
Idioms: **pie chart, polar area chart**

- **pie chart**
  - area marks with angle channel
  - accuracy: angle/area less accurate than line length
    - arclength also less accurate than line length

- **polar area chart**
  - area marks with length channel
  - more direct analog to bar charts

- **data**
  - 1 categ key attrib, 1 quant value attrib

- **task**
  - part-to-whole judgements

---

Idioms: **normalized stacked bar chart**

- task
  - part-to-whole judgements

- normalized stacked bar chart
  - stacked bar chart, normalized to full vert height
  - single stacked bar equivalent to full pie
    - high information density: requires narrow rectangle

- pie chart
  - information density: requires large circle

http://bl.ocks.org/mbostock/3886208
http://bl.ocks.org/mbostock/3887235
http://bl.ocks.org/mbostock/3886394
Idiom: **glyphmaps**

- rectilinear good for linear vs nonlinear trends

- radial good for cyclic patterns

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 colormap [*more later*]
  - (geographic heat map)

[http://bl.ocks.org/mbostock/4060606](http://bl.ocks.org/mbostock/4060606)
Population maps trickiness

• beware!
• absolute vs relative again
  • population density vs per capita
• investigate with Ben Jones Tableau
  Public demo
  • http://public.tableau.com/profile/ben.jones#!/vizhome/PopVsFin/PopVsFin

Are Maps of Financial Variables just Population Maps?
  • yes, unless you look at per capita (relative) numbers

[ https://xkcd.com/1138 ]
Idiom: Bayesian surprise maps

• use models of expectations to highlight surprising values
• confounds (population) and variance (sparsity)

[Surprise! Bayesian Weighting for De-Biasing Thematic Maps. Correll and Heer. Proc InfoVis 2016]

Arrange networks and trees

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

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

- **Enclosure**
  - Containment Marks
  - ![Marks](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 $E < 4N$
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


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!

Link marks: Connection and containment

• 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

**Encode**

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

**Map**

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

**Manipulate**

- Change
- Select
- Navigate

**Facet**

- Juxtapose
- Partition
- Superimpose

**Reduce**

- Filter
- Aggregate
- Embed
How to handle complexity: 1 previous strategy + 3 more

- Derive new data to show within view
- Change view over time
- Facet across multiple views
- Reduce items/attributes within single view

Derive

- Change
- Juxtapose
- Filter

Manipulate

- Select
- Partition

Facet

- Navigate
- Superimpose

Reduce

- Embed
More Information

• this talk
  www.cs.ubc.ca/~tmm/talks.html#vad17stat545

• 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, courses
  http://www.cs.ubc.ca/group/infovis
  http://www.cs.ubc.ca/~tmm