Data Visualization Pitfalls to Avoid

Tamara Munzner
Department of Computer Science
University of British Columbia

CBR Arts Meets Science, UBC Centre for Blood Research
Mar 23 2017, Vancouver BC

http://www.cs.ubc.ca/~tmm/talks.html#cbr17
@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
  – presentation of known results
  – stepping stone towards automation: refining, trustbuilding

• intended task, measurable definitions of effectiveness

more at:
Visualization Analysis and Design, Chapter 1.
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

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


Threats to validity differ 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

Evaluate success at each level with methods from different fields

- **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

- **Analytical computer science**
  - Analyze results qualitatively
  - Measure human time with lab experiment (*lab study*)

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

---

Design Study Methodology

Reflections from the Trenches and from the Stacks

http://www.cs.ubc.ca/labs/imager/tr/2012/dsm/

Design Study Methodology: Reflections from the Trenches and from the Stacks.
Design Studies: Lessons learned after 21 of them

Cerebral genomics
MizBee genomics
Pathline genomics
MulteeSum genomics
Vismon fisheries management
QuestVis sustainability
WiKeVis in-car networks

MostVis in-car networks
Car-X-Ray in-car networks
ProgSpy2010 in-car networks
RelEx in-car networks
Cardiogram in-car networks
AutobahnVis in-car networks
VisTra in-car networks

Constellation linguistics
LibVis cultural heritage
Caidants multicast
SessionViewer web log analysis
LiveRAC server hosting
PowerSetViewer data mining
LastHistory music listening
Methodology for Problem-Driven Work

• definitions

• 9-stage framework

• 32 pitfalls and how to avoid them
**What?**

**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>Items</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)
- Shapes
- Grids
- Positions

**Dataset Availability**
- Static
- Dynamic

**Attributes**
- Categorical
- Ordered
- Quantitative

**Ordering Direction**
- Sequential
- Diverging
- Cyclic
Three major datatypes

Dataset Types

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

Networks
- Link
- Node (item)
- Trees

Spatial
- Fields (Continuous)
- Geometry (Spatial)

- Visualization vs computer graphics
- Geometry is design decision
Types: Datasets and data

Dataset Types

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

- **Networks**
  - Link
  - Node (item)

Spatial

- **Fields (Continuous)**
  - Grid of positions
  - Cell
  - Attributes (columns)
  - Value in cell

- **Geometry (Spatial)**
  - Position

Attribute Types

- **Categorical**
- **Ordered**
  - **Ordinal**
  - **Quantitative**

- **Spatial Networks**
  - Nodes (items)
  - Fields (Continuous)
  - Attributes (columns)
  - Value in cell

- **Spatial Tables**
  - Attributes (columns)
  - Items (rows)
  - Cell containing value
• \{action, target\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – browse topology

Data

Actions

- Analyze
  - Consume
    - Discover
      - Trends
  - Produce
    - Annotate
    - Derive
  - Present
  - Enjoy

On the right:

Targets

- All Data
  - Trends
  - Outliers
  - Features
- Attributes
  - One
    - One
    - Distribution
    - Extreme
  - Many
    - Dependency
    - Correlation
    - Similarity
- Network Data
  - Topology
    - Paths
- Spatial Data
  - Shape

Why?

What?

How?
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
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

Derived Data

trade balance = \textit{exports} – \textit{imports}
Targets

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

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

- **Network Data**
  - Topology
    - Paths

- **Spatial Data**
  - Shape
## 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?**
How to encode: Arrange space, map channels

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

- **Express**

- **Separate**

- **Align**

- **Use**

- **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

<table>
<thead>
<tr>
<th>Points</th>
<th>Lines</th>
<th>Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Points" /></td>
<td><img src="image2.png" alt="Lines" /></td>
<td><img src="image3.png" alt="Areas" /></td>
</tr>
</tbody>
</table>

### Position
- Horizontal
- Vertical
- Both

### Color
- ![Color](image4.png)

### Shape
- ![Shape](image5.png)

### Tilt
- ![Tilt](image6.png)

### 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)**

**Attributes**

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

**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)
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
Challenges of Color

• what is wrong with this picture?

@WTFViz

“visualizations that make no sense”

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
    • saturation
  – categorical can show identity
    • hue

• channels have different properties
  – what they convey directly to perceptual system
  – how much they can convey: how many discriminable bins can we use?
Luminance

- need luminance for edge detection
  - fine-grained detail only visible through luminance contrast
  - legible text requires luminance contrast!

- intrinsic perceptual ordering

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

-so what can we do instead?

[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 [e.g., 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


Visual encoding: 2D vs 3D

• 2D good, 3D better?
  – not so fast…

http://amberleyromo.com/images/Bookcover/Animal-Farm.png
Unjustified 3D all too common, in the news and elsewhere

http://viz.wtf/post/137826497077/eye-popping-3d-triangles  
http://viz.wtf/post/13902022202/designer-drugs-ht-ducqn
Depth vs power of the plane

- high-ranked spatial position channels: **planar** spatial position – not depth!

**Magnitude Channels: Ordered Attributes**

- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
Life in 3D?…

• we don’t really live in 3D: we see in 2.05D
  – acquire more info on image plane quickly from eye movements
  – acquire more info for depth slower, from head/body motion

[adapted from Visual Thinking for Design. Ware. Morgan Kaufmann 2010.]
Occlusion hides information

• occlusion
• interaction complexity

Perspective distortion loses information

- perspective distortion
  - interferes with all size channel encodings
  - power of the plane is lost!

[Visualizing the Results of Multimedia Web Search Engines. Mukherjea, Hirata, and Hara. InfoVis 96]
3D vs 2D bar charts

• 3D bars never a good idea!
No unjustified 3D example: Time-series data

- extruded curves: detailed comparisons impossible

[Cluster and Calendar based Visualization of Time Series Data. van Wijk and van Selow, Proc. InfoVis 99.]
No unjustified 3D example: Transform for new data abstraction

- derived data: cluster hierarchy
- juxtapose multiple views: calendar, superimposed 2D curves

[Cluster and Calendar based Visualization of Time Series Data. van Wijk and van Selow, Proc. InfoVis 99.]
Justified 3D: shape perception

• benefits outweigh costs when task is shape perception for 3D spatial data
  – interactive navigation supports synthesis across many viewpoints

No unjustified 3D

• 3D legitimate for true 3D spatial data
• 3D needs very careful justification for abstract data
  – enthusiasm in 1990s, but now skepticism
  – be especially careful with 3D for point clouds or networks

Justified 3D: Economic growth curve

Four strategies to handle complexity

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

- **Manipulate**
  - Change

- **Facet**
  - Juxtapose

- **Reduce**
  - Filter

more at:
Visualization Analysis and Design. 
More Information

• this talk
  www.cs.ubc.ca/~tmm/talks.html#cbr17

• book
  http://www.cs.ubc.ca/~tmm/vadbook
  – 20% off promo code, book+ebook combo: HVN17

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