Visualization Analysis & Design

I & II

Tamara Munzner
Department of Computer Science
University of British Columbia

IAS PITP Summer School Tutorial
July 2016, Princeton NJ

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp

@tamaramunzner
Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp  @tamaramunzner
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>Identical statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x mean</td>
<td>9</td>
</tr>
<tr>
<td>x variance</td>
<td>10</td>
</tr>
<tr>
<td>y mean</td>
<td>8</td>
</tr>
<tr>
<td>y variance</td>
<td>4</td>
</tr>
<tr>
<td>x/y correlation</td>
<td>1</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**
      - 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

- **Anthropology/ethnography**
  - **Domain situation**
    - Observe target users using existing tools

- **Design**
  - **Data/task abstraction**
    - **Visual encoding/interaction idiom**
      - Justify design with respect to alternatives
    - **Algorithm**
      - Measure system time/memory
      - Analyze computational complexity
  - **Cognitive psychology**
    - Analyze results qualitatively
    - Measure human time with lab experiment (*lab study*)
  - **Anthropology/ethnography**
    - Observe target users after deployment (*field study*)
    - Measure adoption

---

# Datasets

**What?**

<table>
<thead>
<tr>
<th>Data Types</th>
<th>Attributes</th>
<th>Links</th>
<th>Positions</th>
<th>Grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>Attributes</td>
<td>Links</td>
<td>Positions</td>
<td>Grids</td>
</tr>
</tbody>
</table>

**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>Positions</td>
</tr>
<tr>
<td>Attributes</td>
<td>Links</td>
<td>Positions</td>
<td>Attributes</td>
<td>Grids</td>
</tr>
</tbody>
</table>

**Dataset Types**

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

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

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

- **Multidimensional Table**

- **Trees**

- **Geometry (Spatial)**

**Attribute Types**

- **Categorical**
  - + • □ ▲

- **Ordered**
  - Ordinal
  - + • □ ▲

- **Quantitative**

**Ordering Direction**

- Sequential
- Diverging
- Cyclic

**Dataset Availability**

- Static
- Dynamic

---

**Why?**

- Dataset Availability
  - Static
  - Dynamic
Three major datatypes

- **Dataset Types**

  - **Tables**
    - Items (rows)
    - Attributes (columns)
    - Cell containing value
  
  - **Networks**
    - Link
    - Node (item)
  
  - **Spatial**
    - Fields (Continuous)
    - Geometry (Spatial)
    - Grid of positions
    - Cell
    - Node (item)
    - Link
    - Position
    - Value in cell
Attribute types

Attribute Types

- Categorical
- Ordered
- Ordinal
- Quantitative

Ordering Direction

- Sequential
- Diverging
- Cyclic
• {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

Derived Data

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

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

- **Attributes**
  - One
    - Distribution
    - Extremes
  - Many

- **Network Data**
  - Topology
    - Paths

- **Spatial Data**
  - Shape
**Encode**

- **Arrangement**
  - Express
  - Separate

- **Ordering**
  - Align

- **Using**
  - [Map](#)

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

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

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

- **Shape**
  - [Shape](#)

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

**Manipulate**

- **Change**

- **Select**

- **Navigate**

**Facet**

- **Juxtapose**

- **Partition**

**Reduce**

- **Filter**

- **Aggregate**

- **Embed**

---

**What?**

- [What](#)

**Why?**

- [Why](#)

**How?**

- [How](#)
Further reading

  – Chap 1: What’s Vis, and Why Do It?
  – Chap 2: What: Data Abstraction
  – Chap 3: Why: Task Abstraction


Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 4 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp  @tamaramunzner
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**
  - **Partition**
  - **Navigate**
  - **Superimpose**

Facet
- **Juxtapose**

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

What?

Why?

How?
Visual encoding

• analyze idiom structure
Definitions: Marks and channels

• marks
  – geometric primitives

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

• interactions
  – point marks only convey position; no area constraints
    • can be size and shape coded
  – line marks convey position and length
    • can only be size coded in 1D (width)
  – area marks fully constrained
    • cannot be size or shape coded
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
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

---

• 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

[mappa.mundi.net/maps/maps_014/telegeography.html]
Separability vs. Integrality

Position
- Hue (Color)

Size
- Hue (Color)

Width
- Height

Red
- Green

Fully separable

Some interference

Some/significant interference

Major interference

2 groups each

2 groups each

3 groups total: integral area

4 groups total: integral hue
Popout

• find the red dot
  – how long does it take?

• parallel processing on many individual channels
  – speed independent of distractor count
  – speed depends on channel and amount of difference from distractors

• serial search for (almost all) combinations
  – speed depends on number of distractors
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
Relative vs. absolute judgements

- perceptual system mostly operates with relative judgements, not absolute
  - that’s why accuracy increases with common frame/scale and alignment
  - Weber’s Law: ratio of increment to background is constant
    - filled rectangles differ in length by 1:9, difficult judgement
    - white rectangles differ in length by 1:2, easy judgement

Further reading

  – Chap 5: Marks and Channels


Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – **Arrange Tables**
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmn/talks.html#vad16pitp
<table>
<thead>
<tr>
<th>Encode</th>
<th>Manipulate</th>
<th>Facet</th>
<th>Reduce</th>
</tr>
</thead>
</table>
| **Arrange**<br>→ Express<br>→ Separate<br>→ Order<br>→ Align<br>→ Use | **Map**<br>from *categorical* and *ordered* attributes<br>→ Color<br>→ Hue<br>→ Saturation<br>→ Luminance<br>→ Size, Angle, Curvature, ...<br>→ Shape<br>→ + ● ■ ▲ | **Change**<br>**Select**<br>**Partition**<br>**Superimpose**<br>**Juxtapose**<br>**Filter**<br>**Aggregate**<br>**Embed** | }

**How?**

---

**What?**

**Why?**

**How?**
Arrange tables

Express Values

Separate, Order, Align Regions

Separate  Order  Align

1 Key 2 Keys 3 Keys Many Keys

List Recursive Subdivision

Axis Orientation

Rectilinear  Parallel  Radial

Layout Density

Dense  Space-Filling

1 Key
List

2 Keys
Matrix

3 Keys
Volume

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**

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

- **Multidimensional Table**
  - Key 1
  - Key 2
  - Attributes
  - Value in cell

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

**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
Idiom: line chart

• one key, one value
  – data
    • 2 quant attrs
  – 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
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: **heatmap**

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

- Rectilinear
- Parallel
- Radial
Idioms: *scatterplot matrix, parallel coordinates*

• **scatterplot matrix (SPLOM)**
  – rectilinear axes, point mark
  – all possible pairs of axes
  – scalability
    • one dozen attribs
    • dozens to hundreds of items

• **parallel coordinates**
  – parallel axes, jagged line representing item
  – rectilinear axes, item as point
    • axis ordering is major challenge
  – scalability
    • dozens of attribs
    • hundreds of items

---

Task: Correlation

• scatterplot matrix
  – positive correlation
    • diagonal low-to-high
  – negative correlation
    • diagonal high-to-low
  – uncorrelated

• parallel coordinates
  – positive correlation
    • parallel line segments
  – negative correlation
    • all segments cross at halfway point
  – uncorrelated
    • scattered crossings


Figure 3. Parallel Coordinate Plot of Six-Dimensional Data Illustrating Correlations of $\rho = 1, .8, .2, 0, -.2, -.8, \text{and} -1.$
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/3887235](http://bl.ocks.org/mbostock/3887235)
[http://bl.ocks.org/mbostock/3886208](http://bl.ocks.org/mbostock/3886208)
[http://bl.ocks.org/mbostock/3886394](http://bl.ocks.org/mbostock/3886394)
Idiom: **glyphmaps**

- rectilinear good for linear vs nonlinear trends

- radial good for cyclic patterns

Orientation limitations

• rectilinear: scalability wrt #axes
  • 2 axes best
  • 3 problematic
    – more in afternoon
  • 4+ impossible
• parallel: unfamiliarity, training time
• radial: perceptual limits
  – angles lower precision than lengths
  – asymmetry between angle and length
    • can be exploited!

Further reading

  – Chap 7: Arrange Tables


• A Brief History of Data Visualization. Friendly. 2008.
  http://www.datavis.ca/milestones
Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp
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]*

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

[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

*Land Information New Zealand Data Service*
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
    • no derived geometry

Vector and tensor fields

- data
  - many attributes per cell
- idiom families
  - flow glyphs
    - purely local
  - geometric flow
    - derived data from tracing particle trajectories
    - sparse set of seed points
  - texture flow
    - derived data, dense seeds
  - feature flow
    - global computation to detect features
      - encoded with one of methods above


Vector fields

• empirical study tasks
  – finding critical points, identifying their types
  – identifying what type of critical point is at a specific location
  – predicting where a particle starting at a specified point will end up (advection)
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

Further reading

  - Chap 8: Arrange Spatial Data
Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – **Arrange Networks and Trees**

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp
@tamaramunzner
Arrange networks and trees

- **Node–Link Diagrams**
  - Connection Marks
    - ![Networks and Trees](Networks.png)

- **Adjacency Matrix**
  - Derived Table
    - ![Networks and Trees](Networks.png)

- **Enclosure**
  - Containment Marks
    - ![Networks and Trees](Networks.png)
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

---

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

---


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

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

Further reading

  – Chap 9: Arrange Networks and Trees


Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmn/talks.html#vad16pitp
@tamaramunzner
Idiom design choices: Encode

Encode

Why?

How?

What?

Arrange

Express

Separate

Order

Align

Use

Map

from categorical and ordered attributes

Color

Hue

Saturation

Luminance

Size, Angle, Curvature, ...

Shape

Motion

Direction, Rate, Frequency,

Map

Color

Size, Angle, Curvature,

Shape

Motion
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

Corners of the RGB color cube
L from HLS
All the same

Luminance values
Colormaps

Categorical

Ordered

Sequential

Diverging

Binary

Categorical

Diverging

Colormaps

→ Categorical
→ Ordered
  → Sequential
→ Diverging

Bivariate

Colormaps

- Categorical
- Ordered
  - Sequential
  - Diverging
- Bivariate

use with care!

Colormaps

- Categorical

- Ordered
  - Sequential
  - Diverging

- Bivariate

• color channel interactions
  – size heavily affects salience
    • small regions need high saturation
    • large need low saturation
  – saturation & luminance: 3-4 bins max
    • also not separable from transparency

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]
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


Map other channels

• size
  – length accurate, 2D area ok, 3D volume poor

• angle
  – nonlinear accuracy
    • horizontal, vertical, exact diagonal

• shape
  – complex combination of lower-level primitives
    – many bins

• motion
  – highly separable against static
    • binary: great for highlighting
  – use with care to avoid irritation

✦ Size, Angle, Curvature, ...
  ➔ Length
  ➔ Angle
  ➔ Area
  ➔ Curvature
  ➔ Volume

✦ Shape
  ➔ Motion
  ➔ Direction, Rate, Frequency, ...

Motion
Further reading

  – *Chap 10: Map Color and Other Channels*

• ColorBrewer, Brewer.
  – [http://www.colorbrewer2.org](http://www.colorbrewer2.org)

  – [http://www.stonesc.com/Vis06](http://www.stonesc.com/Vis06)


Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – **Manipulate:** Change, Select, Navigate
  – **Facet:** Juxtapose, Partition, Superimpose
  – **Reduce:** Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp
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**

**Reduce**
- **Filter**
- **Partition**
- **Aggregate**
- **Superimpose**
- **Embed**

**What?**

**Why?**

**How?**
How to handle complexity: 1 previous strategy + 3 more

- Derive
- Manipulate
  - Change
  - Select
- Facet
  - Juxtapose
  - Partition
  - Superimpose
- Reduce
  - Filter
  - Aggregate
  - Embed

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

- **Change over Time**
  - Select

- **Navigate**
  - Item Reduction
    - Zoom
    - Geometric or Semantic
  - Pan/Translate
  - Constrained

- **Attribute Reduction**
  - Slice
  - Cut
  - Project
Idiom: **Re-encode**

System: **Tableau**

made using Tableau, [http://tableausoftware.com](http://tableausoftware.com)
Idiom: **Reorder**

- data: tables with many attributes
- task: compare rankings

**System: LineUp**

Idiom: **Realign**

- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison

**System: LineUp**

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

Select and highlight

- selection: basic operation for most interaction
- design choices
  - how many selection types?
    - click vs hover: heavyweight, lightweight
    - primary vs secondary: semantics (eg source/target)
- highlight: change visual encoding for selection targets
  - color
    - limitation: existing color coding hidden
  - other channels (eg motion)
  - add explicit connection marks between items
Navigate: Changing item visibility

• change viewpoint
  – changes which items are visible within view
  – camera metaphor
    • zoom
      – geometric zoom: familiar semantics
      – semantic zoom: adapt object representation based on available pixels
        » dramatic change, or more subtle one
    • pan/translate
    • rotate
      – especially in 3D

– constrained navigation
  • often with animated transitions
  • often based on selection set

Navigate

Item Reduction

Zoom
  Geometric or Semantic

Pan/Translate

Constrained
Idiom: **Semantic zooming**

- visual encoding change
  - colored box
  - sparkline
  - simple line chart
  - full chart: axes and tickmarks

**System: LiveRAC**

Navigate: Reducing attributes

• continuation of camera metaphor
  – slice
    • show only items matching specific value for given attribute: slicing plane
    • axis aligned, or arbitrary alignment
  – cut
    • show only items on far slide of plane from camera
  – project
    • change mathematics of image creation
      – orthographic
      – perspective
      – many others: Mercator, cabinet, ...
Further reading

  – Chap 11: Manipulate View


Outline

• **Session 1 11:15am-12:45pm**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 2:15pm-3:45pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp
@tamaramunzner
Facet

- Juxtapose

- Partition

- Superimpose
Juxtapose and coordinate 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>
</tr>
</thead>
<tbody>
<tr>
<td>Same</td>
<td>Redundant</td>
</tr>
<tr>
<td></td>
<td>Overview/Detail</td>
</tr>
<tr>
<td></td>
<td>Small Multiples</td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
</tr>
<tr>
<td></td>
<td>Multiform, Overview/Detail</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**
Why not animation?

- disparate frames and regions: comparison difficult
  - vs contiguous frames
  - vs small region
  - vs coherent motion of group

- safe special case
  - animated transitions
System: **Improvise**

- investigate power of multiple views
  - pushing limits on view count, interaction complexity
  - how many is ok?
    - open research question
- reorderable lists
  - easy lookup
  - useful when linked to other encodings

Partition into views

• how to divide data between views
  – split into regions by attributes
  – encodes association between items using spatial proximity
  – order of splits has major implications for what patterns are visible

• no strict dividing line
  – **view**: big/detailed
    • contiguous region in which visually encoded data is shown on the display
  – **glyph**: small/iconic
    • object with internal structure that arises from multiple marks
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

![Diagram showing population by age and state]
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

System: HIVE

Partitioning: Recursive subdivision

- size regions by sale counts
  - not uniformly
- result: treemap

Superimpose layers

• **layer**: set of objects spread out over region
  – each set is visually distinguishable group
  – extent: whole view

• design choices
  – how many layers, how to distinguish?
    • encode with different, nonoverlapping channels
    • two layers achievable, three with careful design
  – small static set, or dynamic from many possible?
Static visual layering

• foreground layer: roads
  – hue, size distinguishing main from minor
  – high luminance contrast from background

• background layer: regions
  – desaturated colors for water, parks, land areas

• user can selectively focus attention

• “get it right in black and white”
  – check luminance contrast with greyscale view

Superimposing limits

• few layers, but many lines
  – up to a few dozen
  – but not hundreds

• superimpose vs juxtapose: empirical study
  – superimposed for local, multiple for global
  – tasks
    • local: maximum, global: slope, discrimination
  – same screen space for all multiples vs single superimposed

Dynamic visual layering

• interactive, from selection
  – lightweight: click
  – very lightweight: hover

• ex: 1-hop neighbors

System: Cerebral

Further reading

  – Chap 12: Facet Into Multiple Views


Outline

• **Session 1** 11:15am-12:45pm
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2** 2:15pm-3:45pm
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp
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, change, facet
Idiom: **dynamic filtering**

- item filtering
- browse through tightly coupled interaction
  - alternative to queries that might return far too many or too few

---

System: **FilmFinder**

Idiom: **DOSFA**

- attribute filtering
- encoding: star glyphs

Idiom: **histogram**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly
Idiom: scented widgets

• augment widgets for filtering to show information scent
  – cues to show whether value in drilling down further vs looking elsewhere
• concise, in part of screen normally considered control panel

Idiom: boxplot

- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attrs
    - 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: **Hierarchical parallel coordinates**

- dynamic item aggregation
- derived data: **hierarchical clustering**
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy

Dimensionality reduction

• attribute aggregation
  – derive low-dimensional target space from high-dimensional measured space
  – use when you can’t directly measure what you care about
    • true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    • latent factors, hidden variables

Tumor Measurement Data

data: 9D measured space

→ DR

derived data: 2D target space
Idiom: Dimensionality reduction for documents

Task 1
In HD data → Out 2D data

What?
In High-dimensional data
Out 2D data

Why?
Produce
Derive

Task 2
In 2D data → Out Scatterplot Clusters & points

What?
In 2D data
Out Scatterplot
Out Clusters & points

Why?
Discover
Explore
Identify

How?
Encode
Navigate
Select

Task 3
Out Scatterplot Clusters & points → Out Labels for clusters

What?
In Scatterplot Clusters & points
Out Labels for clusters

Why?
Produce
Annotate
Further reading

  – Chap 13: Reduce Items and Attributes


Outline

• **Session 1  2:00-3:40pm**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2  4:15pm-5:50pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Embed: Focus+Context

Outline

• **Session 1 11:15am-12:45pm**
  – Analysis: What, Why, How
  – Marks and Channels
  – Arrange Tables
  – Arrange Spatial Data
  – Arrange Networks and Trees

• **Session 2 2:15pm-3:45pm**
  – Map Color and Other Channels
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate
  – Q&A

http://www.cs.ubc.ca/~tmn/talks.html#vad16pitp
Sneak preview: Not covered today

- **Rules of Thumb**
  - **No unjustified 3D**
    - Power of the plane, dangers of depth
    - Occlusion hides information
    - Perspective distortion loses information
    - Tilted text isn’t legible
  - **No unjustified 2D**
  - **Resolution over immersion**
  - **Overview first, zoom and filter, details on demand**
  - **Function first, form next**
More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad16pitp

• 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