Data Visualization (CSCI 627/490)

Trees & Multiple Views

Dr. David Koop

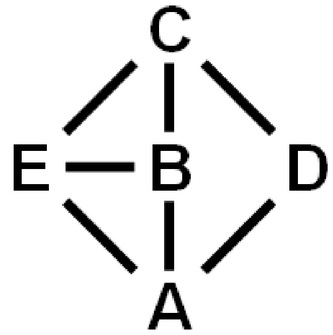
Node-Link Diagrams Need Layouts

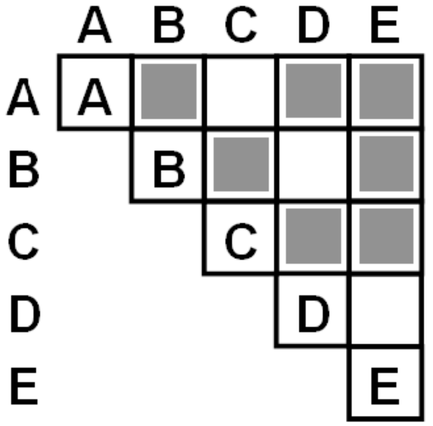
- Need to use spatial position when designing node-link visualizations
- Otherwise, nodes can **occlude** each other, links hard to distinguish
- How?
  - With bar charts, we could order using an attribute…
  - With networks, we want to be able to see connectivity and topology (not in the data usually)
- Possible metrics:
  - Edge crossings
  - Node overlaps
  - Total area

Adjacency Matrix

- Change network to tabular data and use a matrix representation
- Derived data: nodes are keys, edges are boolean values
- Task: lookup connections, find well-connected clusters
- Scalability: millions of edges

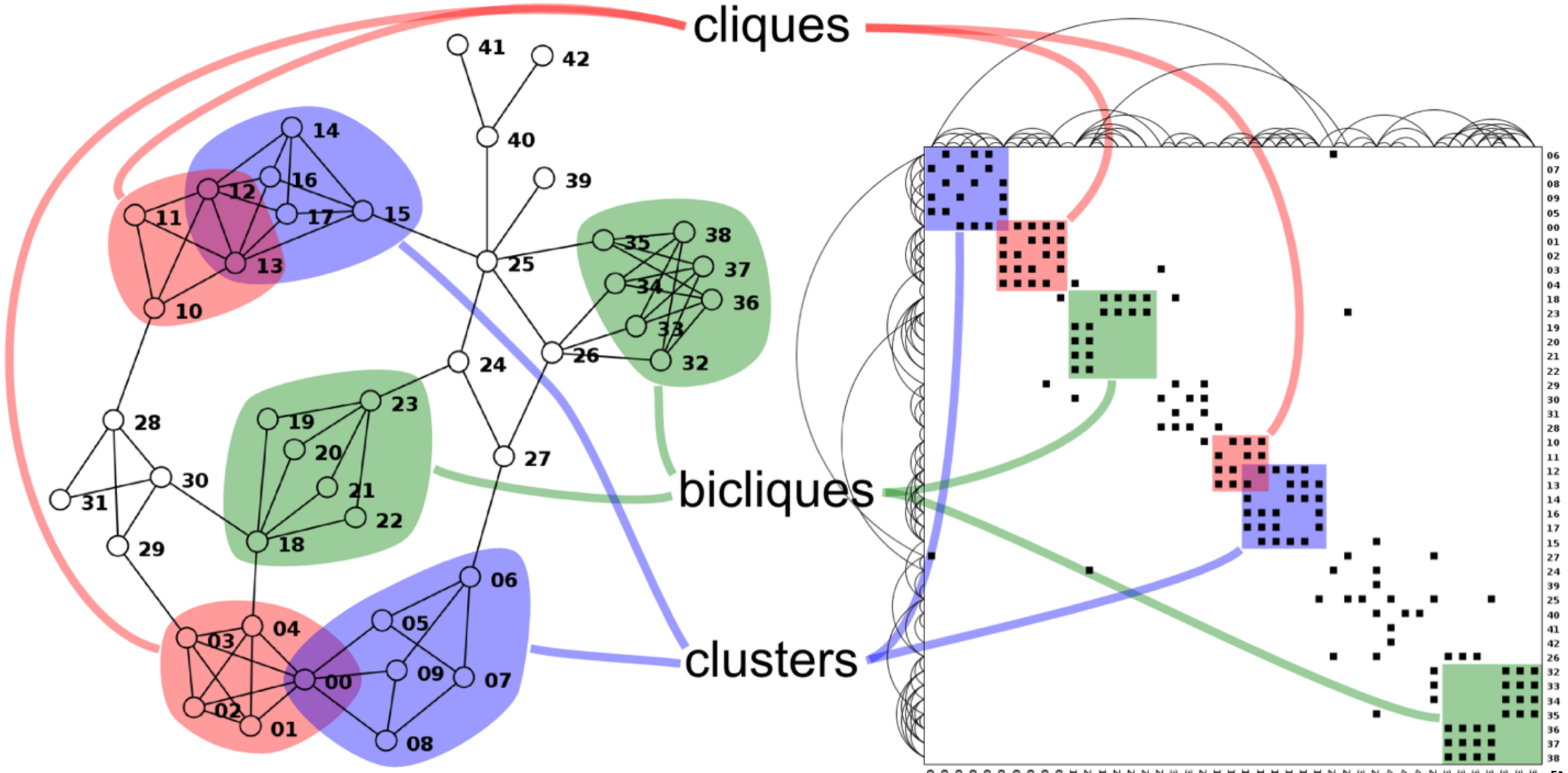
- Can encode **edge weight**, too





[Henry et al., 2007]

Structures from Adjacency Matrices

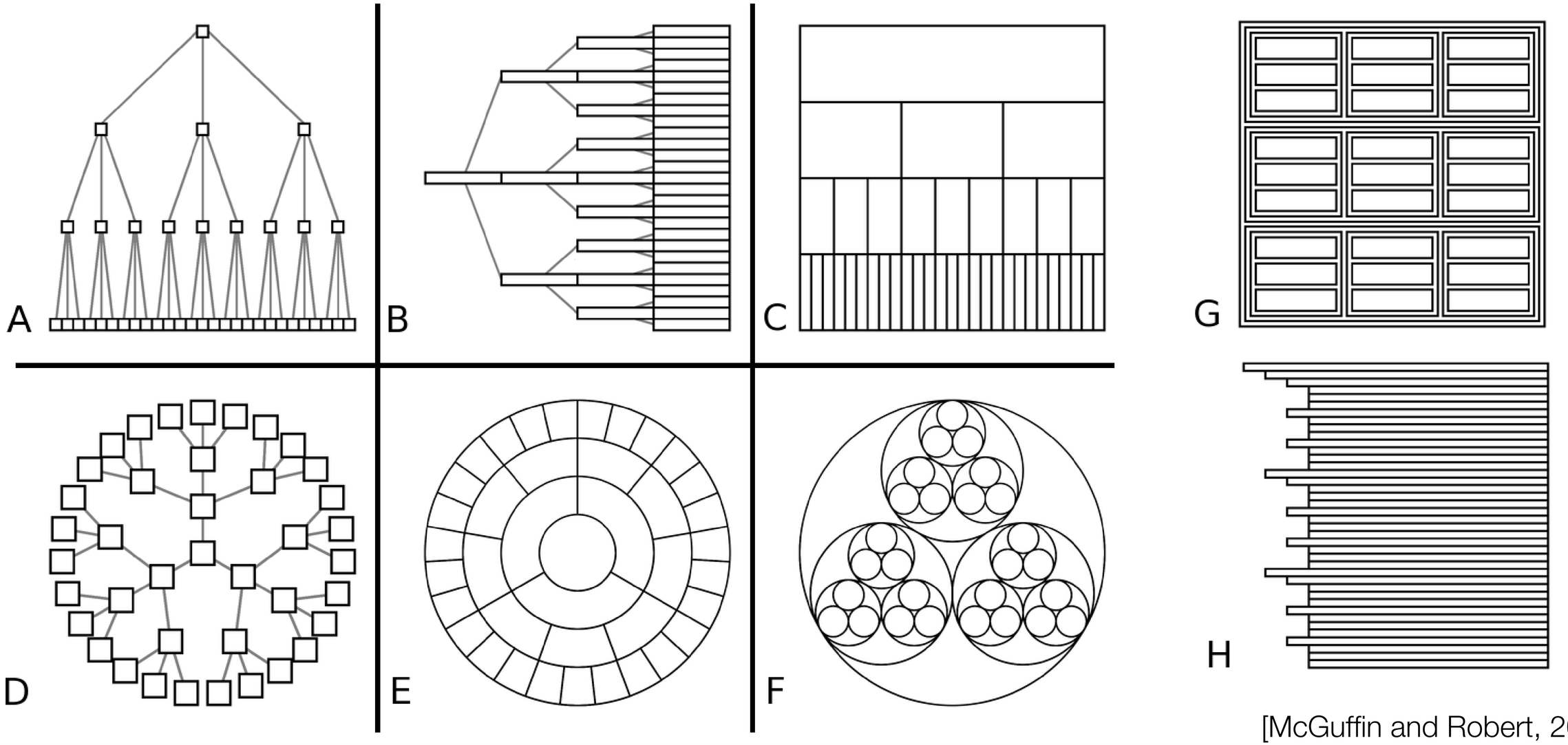


[McGuffin]

Node-Link or Adjacency Matrix?

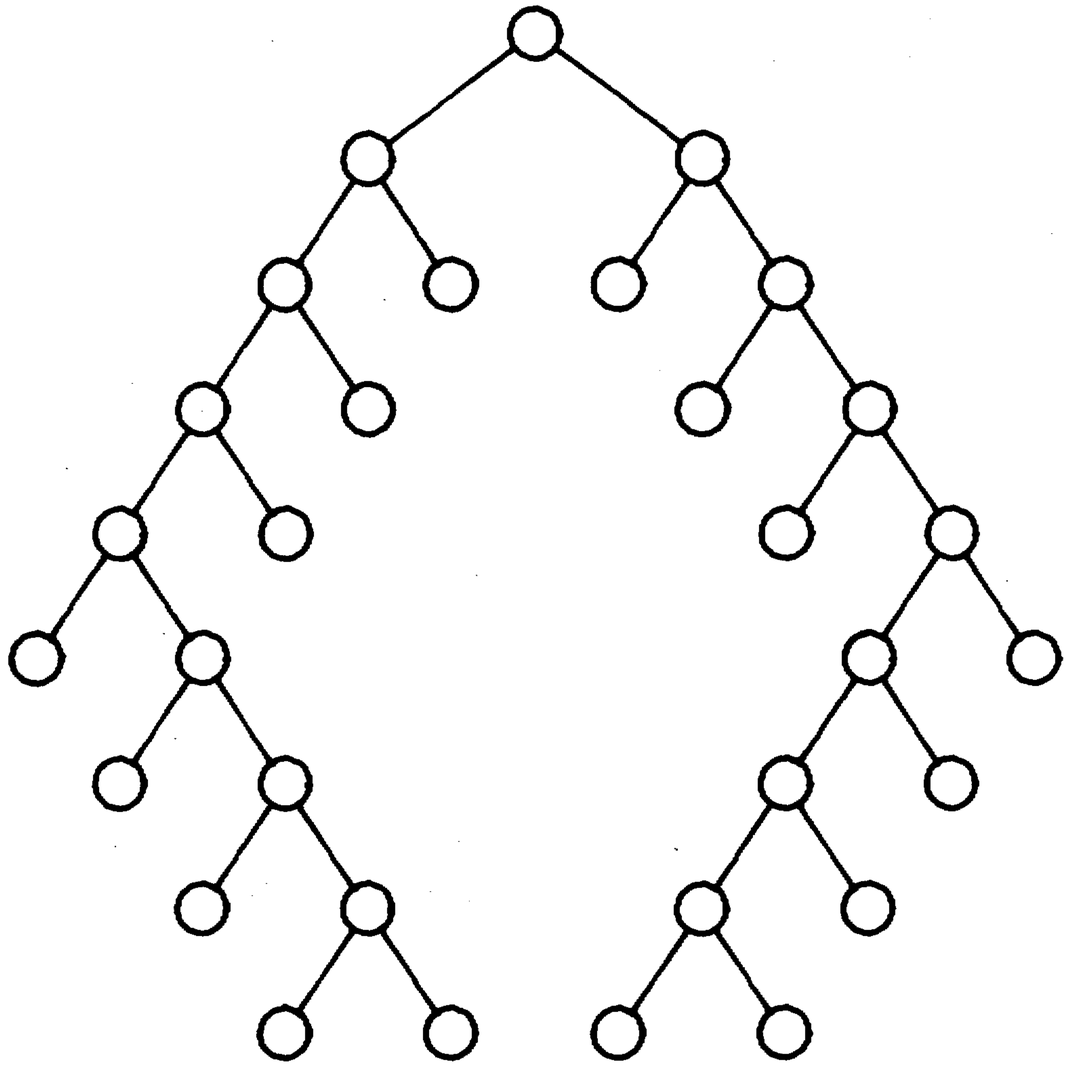
- Empirical study: For most tasks, node-link is better for small graphs and adjacency better for large graphs
- Multi-link paths are hard with adjacency matrices
- Immediate connectivity or neighbors are ok, estimating size (nodes & edges also ok)
- People tend to be more familiar with node-link diagrams
- Link density is a problem with node-link but not with adjacency matrices

Tree Visualizations



[McGuffin and Robert, 2010]

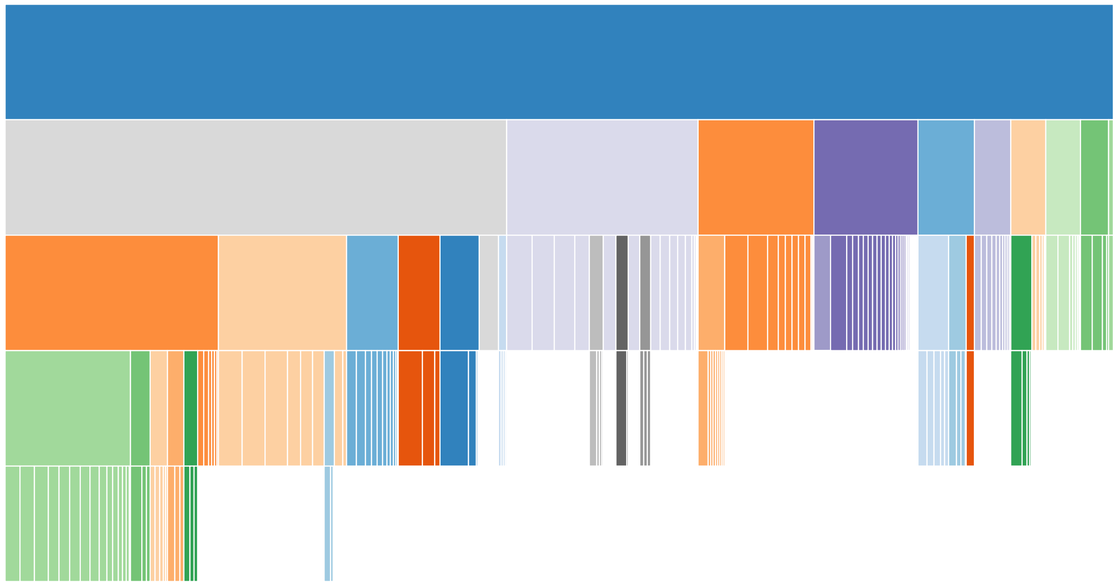
Reingold-Tilford Algorithm

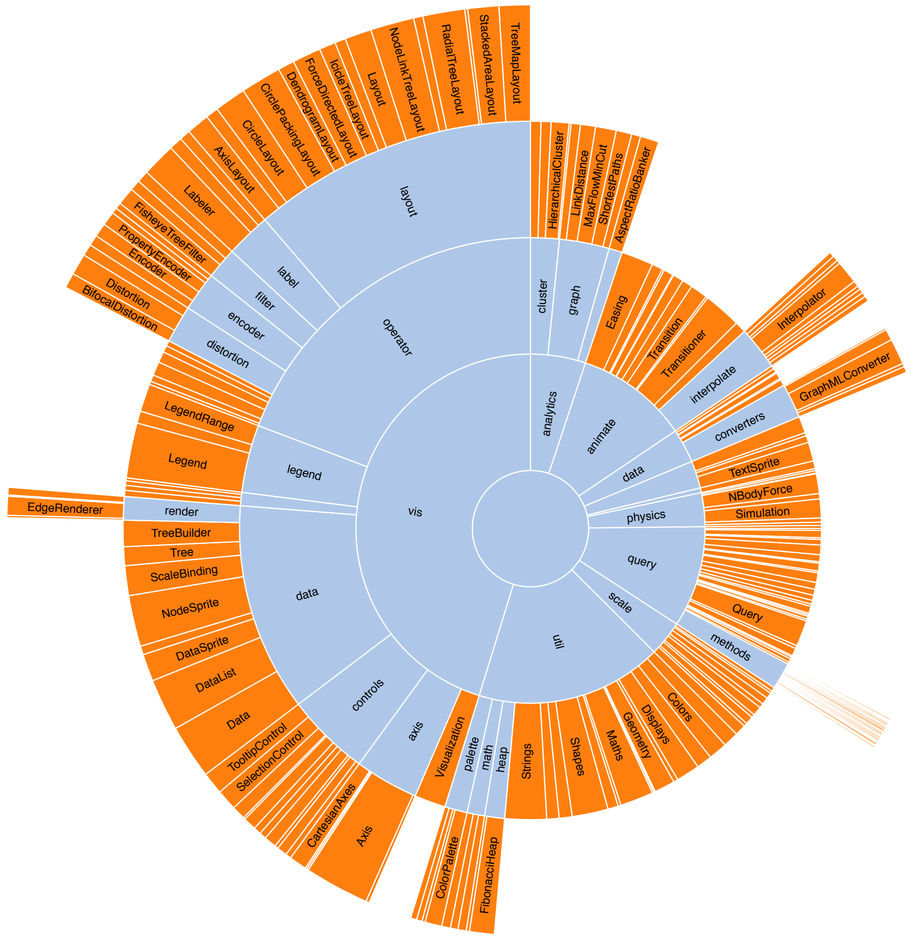


- Recurse on left and right subtrees
- Shift subtree over as long as it doesn't overlap
- Place parent centered above the subtrees
- Originally, only binary trees, extended by Walker

[Reingold and Tilford, 1981]

Icicle & Sunburst Visualizations



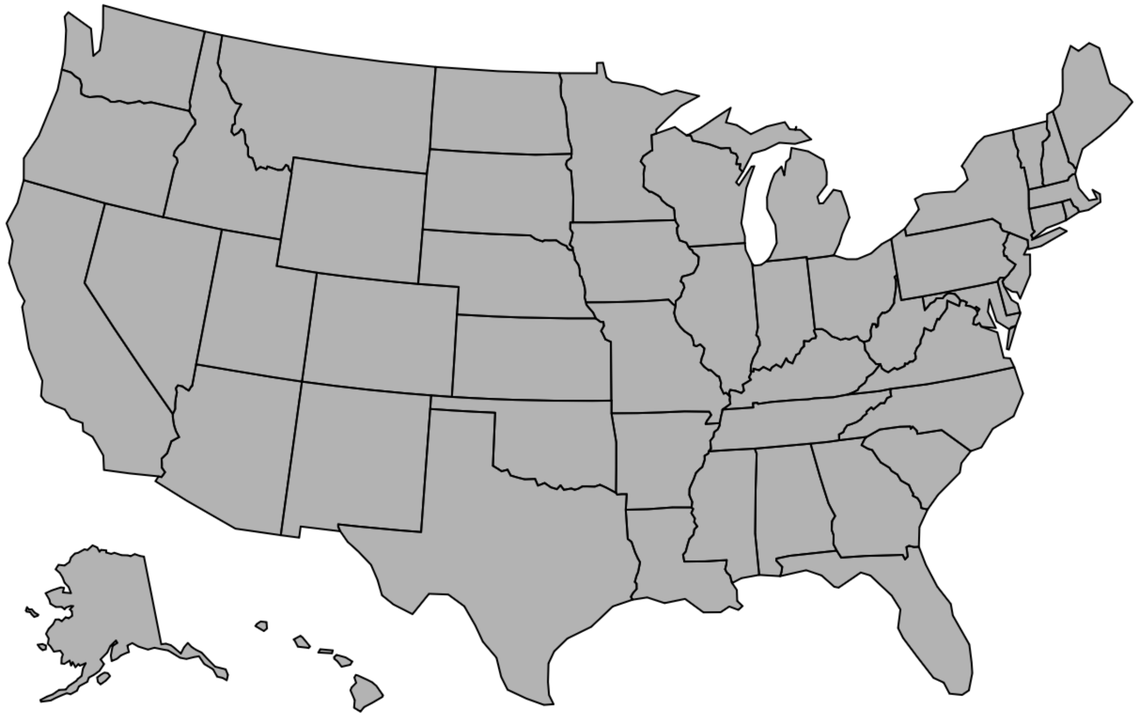


[Bostock (2011), Heer et al. (2012)]

Projects

- Proposal feedback up on Blackboard
  - Be creative: https://xeno.graphics/
- Designs:
  - Three different good designs
  - One bad design
  - Sketch these
  - Progress on implementation

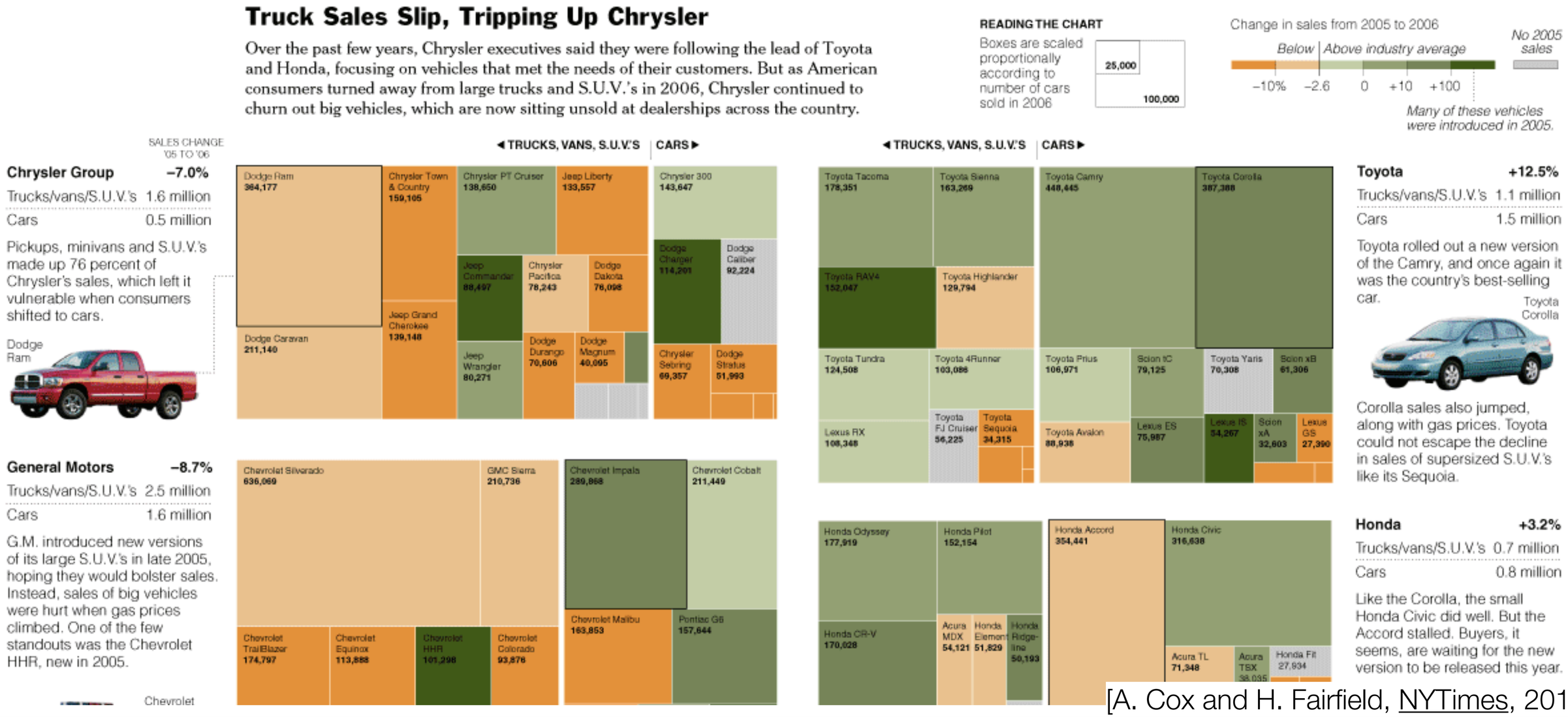
Assignment 4



- Choropleth Maps
  - Use D3 for Part 1
  - Can use either D3 or Plot for Part 2
  - Make sure the colormap is appropriate!
- Treemap [627]
- Two resources:
  - Courselet (Plot in python)
  - Observable Notebook on Maps

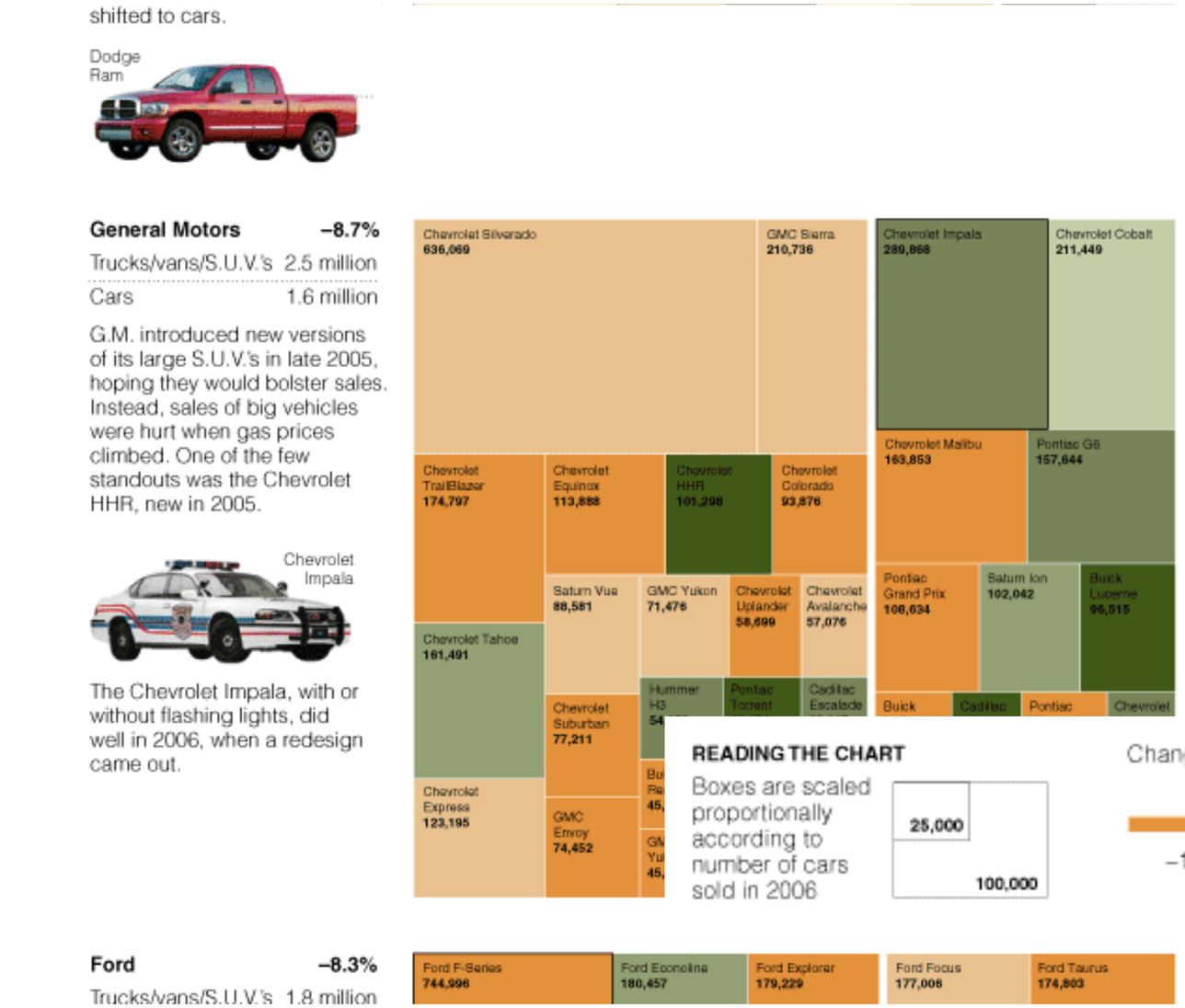
Treemaps

Car/Truck Treemap



[A. Cox and H. Fairfield, NYTimes, 2012]

Car/Truck Treemap



[A. Cox and H. Fairfield, NYTimes, 2012]

Treemap

- Containment marks instead of connection marks
- Encodes some attribute of the items as the **size** of the rectangles
- Not as easy to see the intermediate rectangles
- Scalability: millions of leaf nodes and links possible

- Need a layout algorithm!

Layout Algorithms

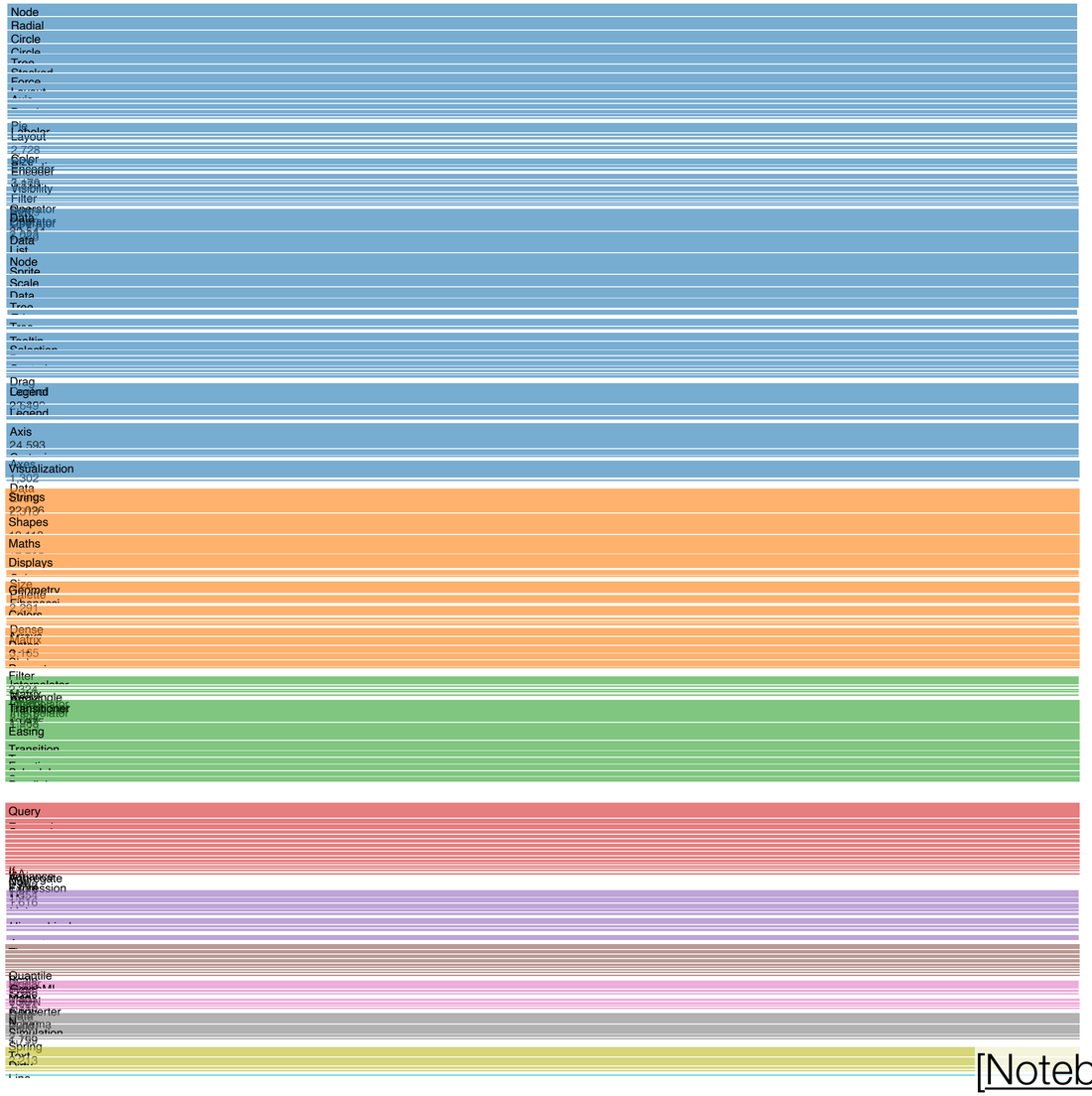
- How do we generate the area marks?
- What considerations should we try to keep in mind?

Layout Algorithms

- How do we generate the area marks?
- What considerations should we try to keep in mind?
  - area true to quantitative value
  - show hierarchy
  - aspect ratio
- Also…
  - ordering
  - stability

Treemap Layouts: Slice

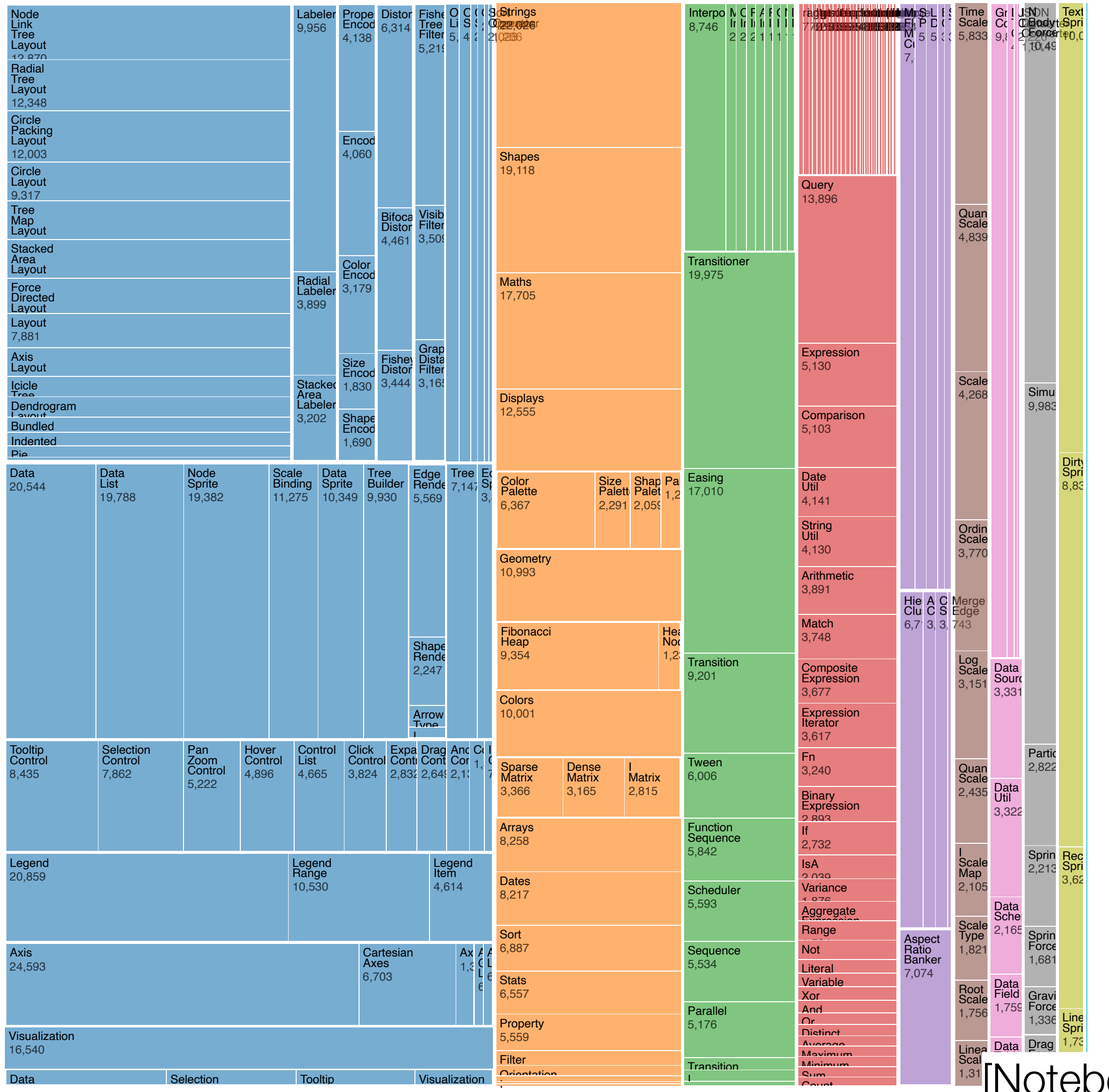
- Just divide horizontally
- Dice is similar, just vertical
- Problem: Bad aspect ratio!
  - Very skinny rectangles
  - Makes it harder to compare sizes, see labels, select rectangles
  - Want rectangles that are closer to squares
  - Aspect ratio = width/height



[Notebook]

Treemap Layouts: Slice & Dice

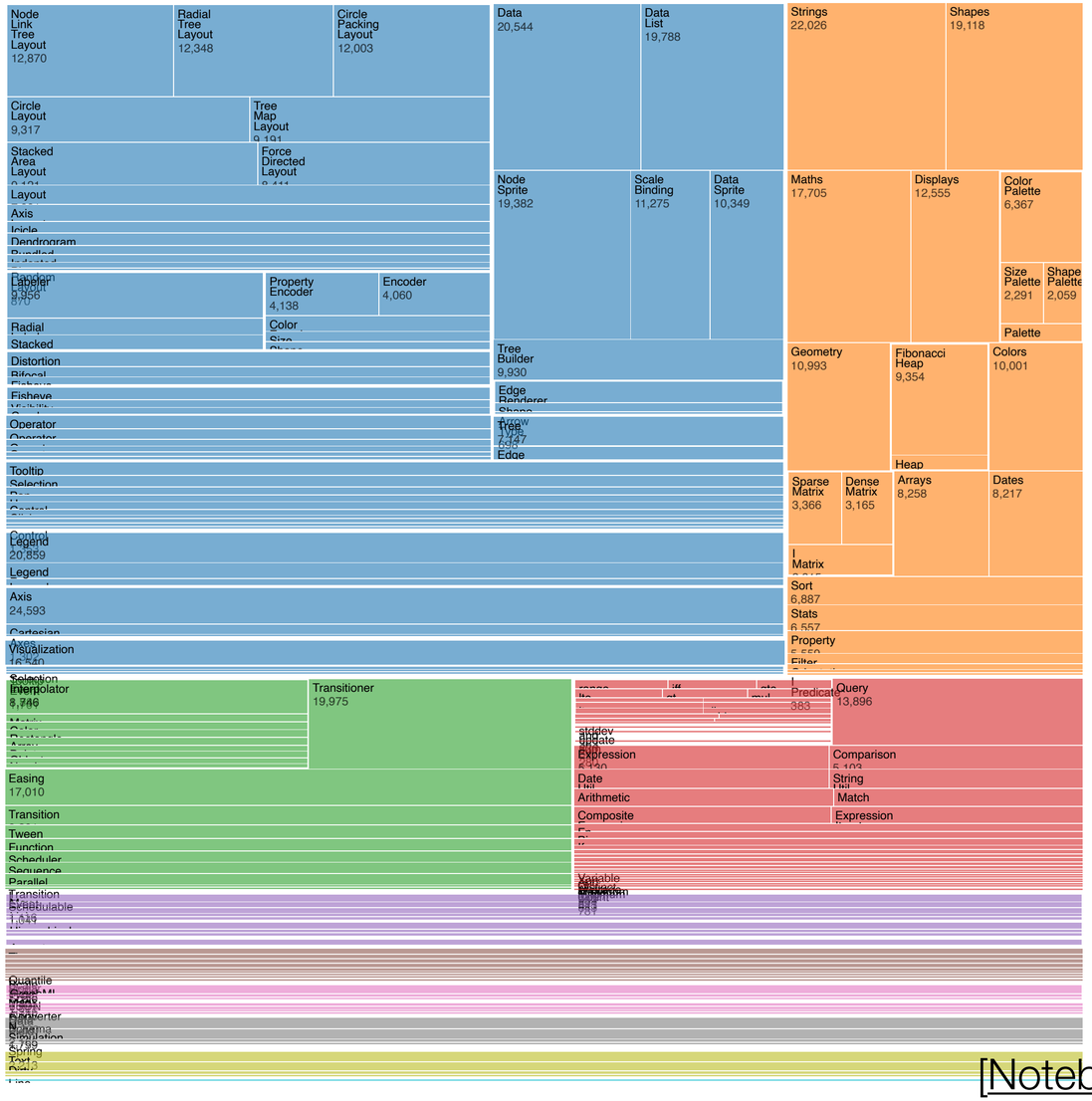
- Split at each level into strips
- At each step, orientation of division (horizontal/vertical) changes
- Better, but some rectangles still have bad aspect ratio



[Notebook]

Treemap Layouts: Strip

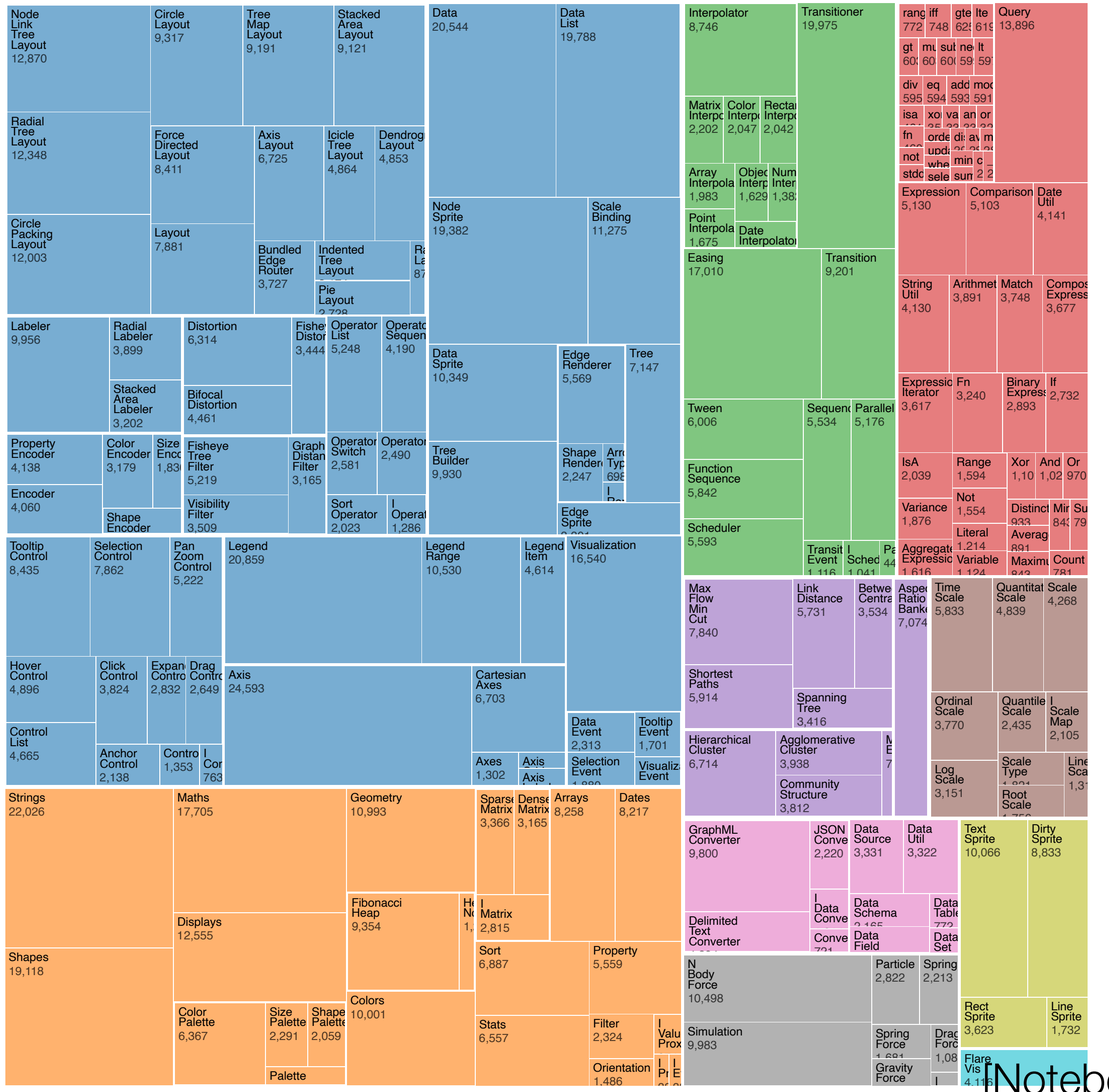
- Consider aspect ratio when adding rectangles
- Do one row at a time by processing rectangles in sorted order by size
  - Check if adding the next rectangle to the row improves aspect ratio
  - When it doesn't, go to next row
- Problem: Last rectangles have bad aspect ratios
- Solution: Look ahead to decide if would be better to add to previous row



[Notebook]

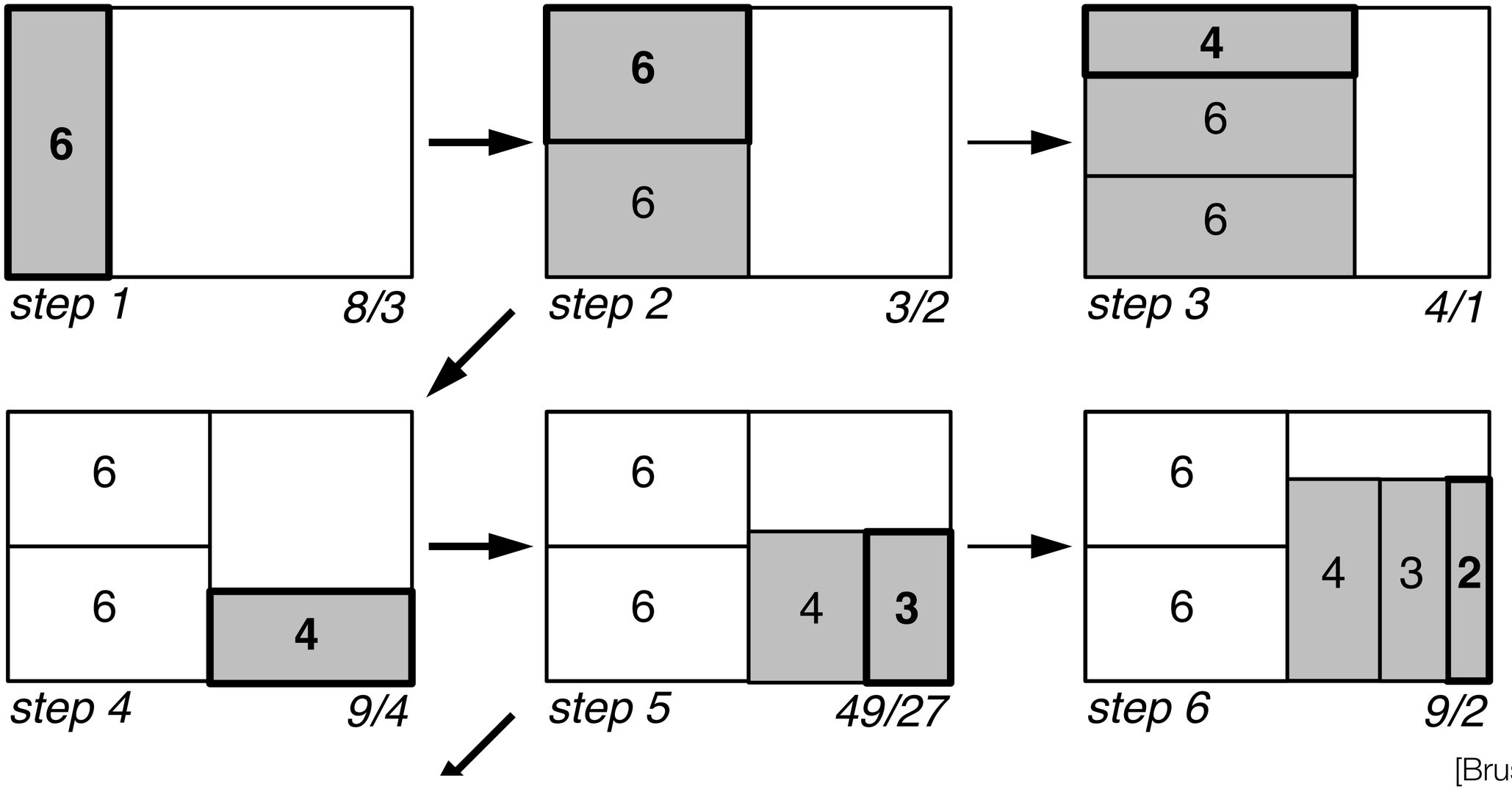
Treemap Layouts: Squarify

- Slice & Dice and Strip can lead to bad **aspect ratios**
- Solution: Strip only uses rows, allow columns to be used, too
- Choose divisions (x/y) based on the width/height of region in order to maintain good aspect ratios
  - Use left and right side
  - Process large rectangles first
- Ordering not preserved which may cause issues if the data is updated



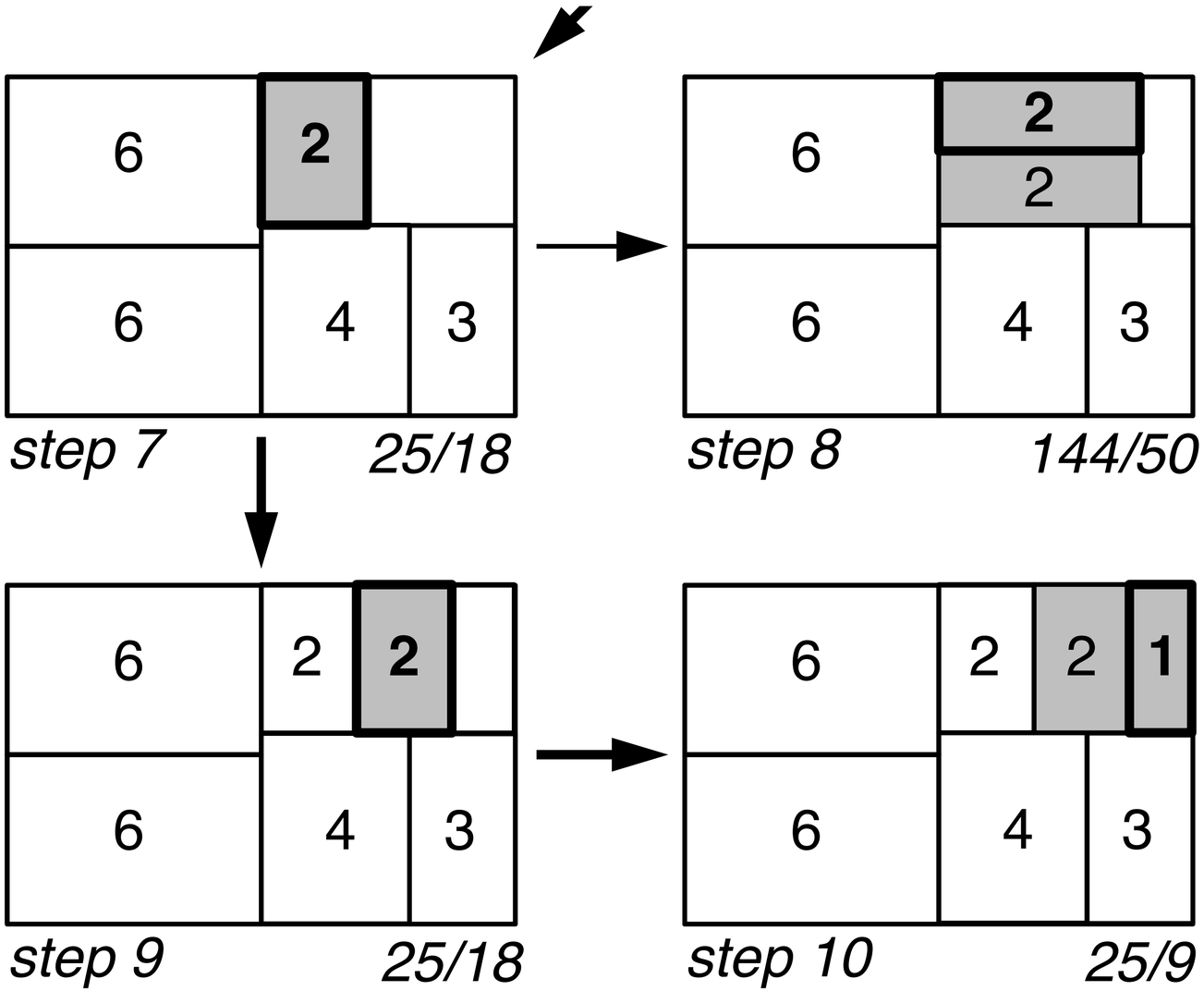
[Notebook]

Squarification Algorithm



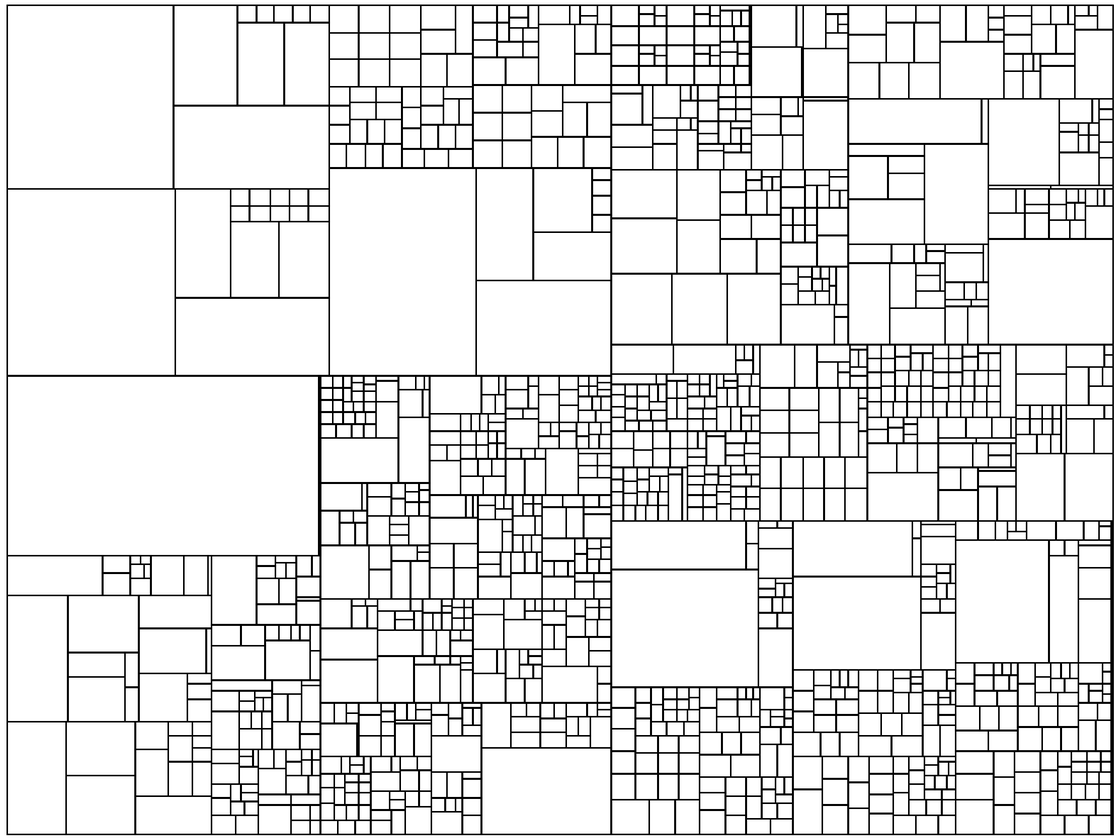
[Brus et al., 1999]

Squarification Algorithm

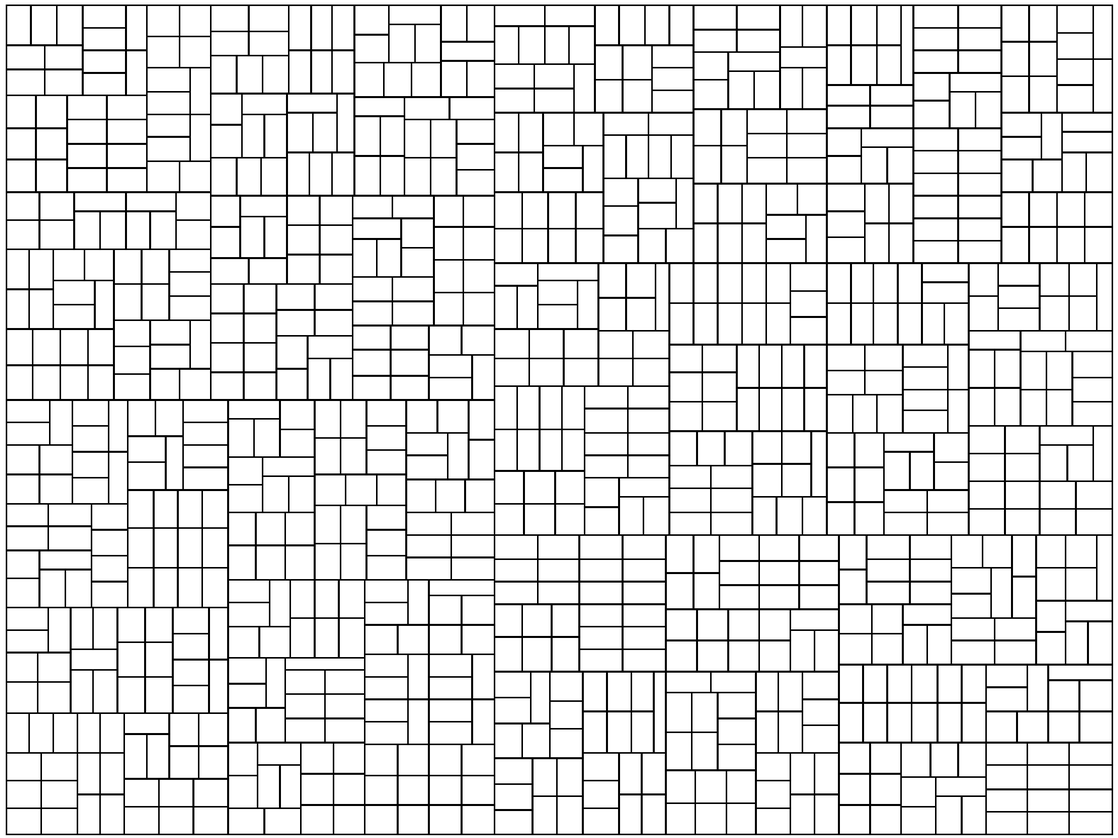


[Brus et al., 1999]

Squarified Treemaps



(a) File system



(b) Organization

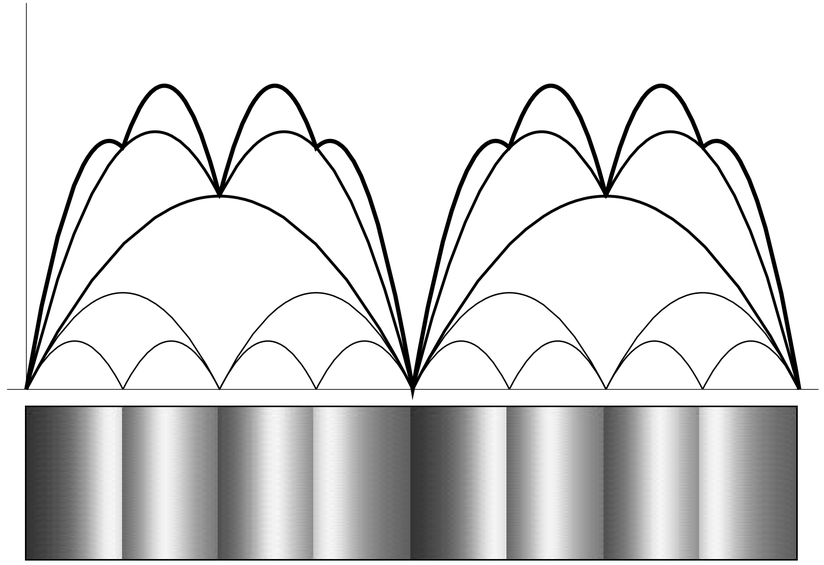
[Brus et al., 1999]

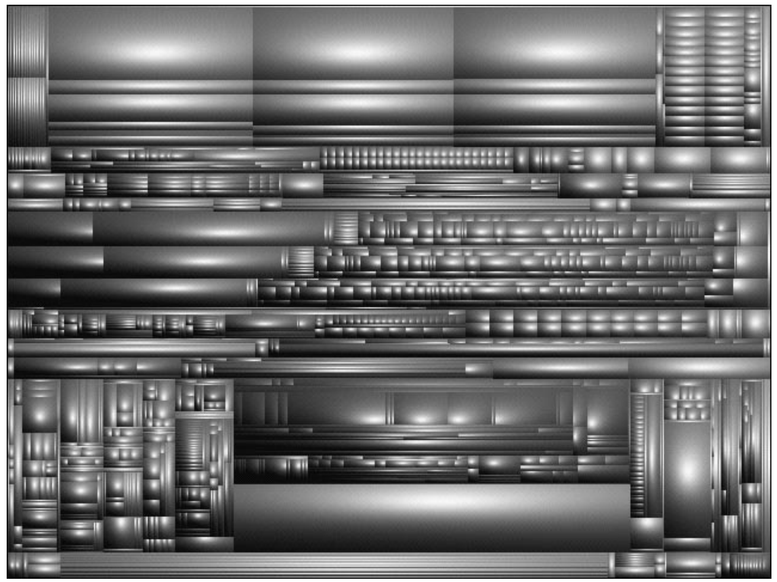
Squarified Layout

- Sort values
- Switch orientation whenever necessary to obtain best aspect ratios

Improving Treemaps (Cushion)

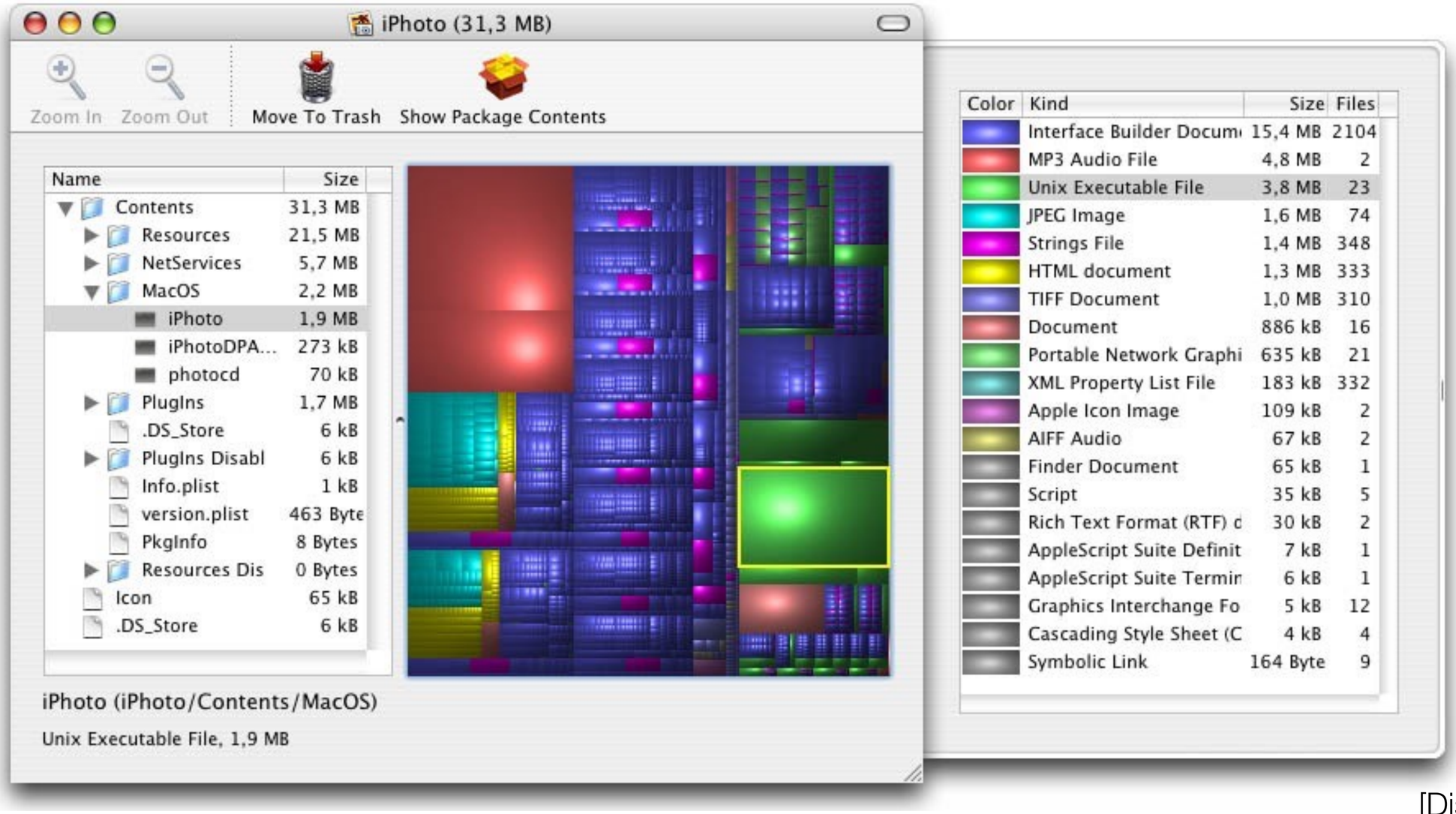
- Leaves are ok, but it can be difficult to find the hierarchy
- Encode this as shading information
- More effective to understand hierarchy





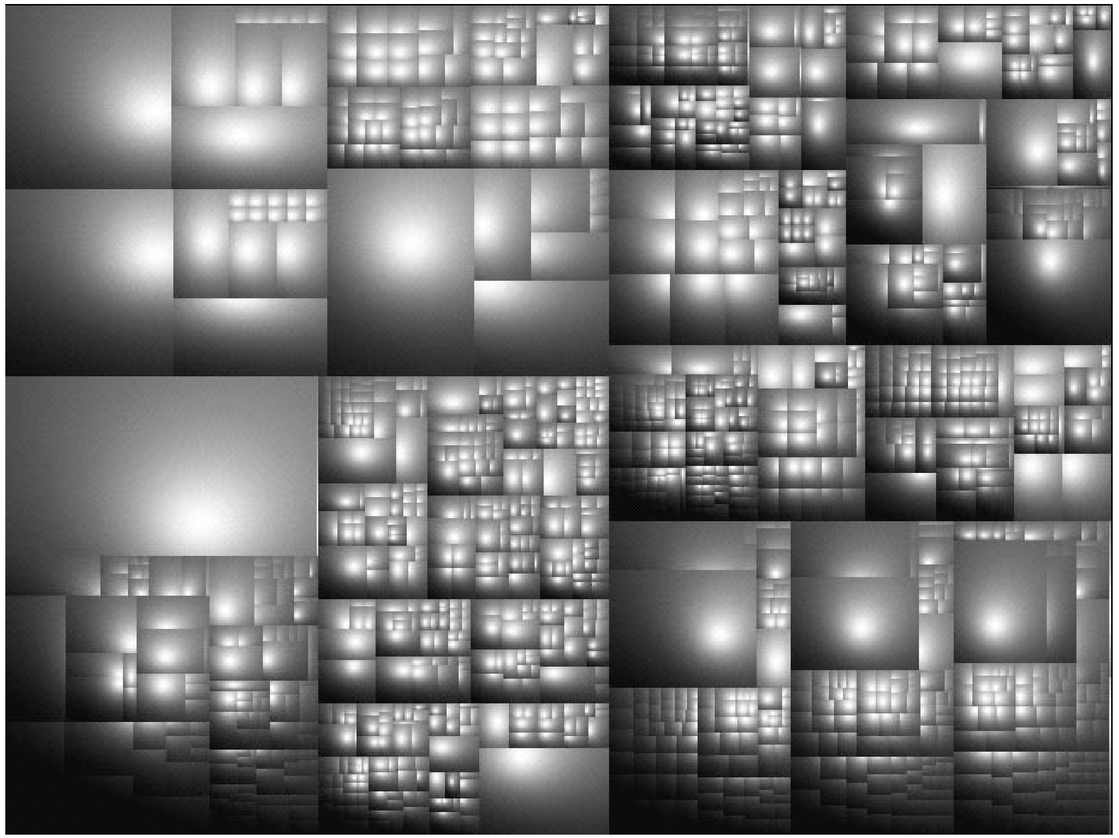
[van Wijk and van de Wetering, 1999]

Disk Inventory

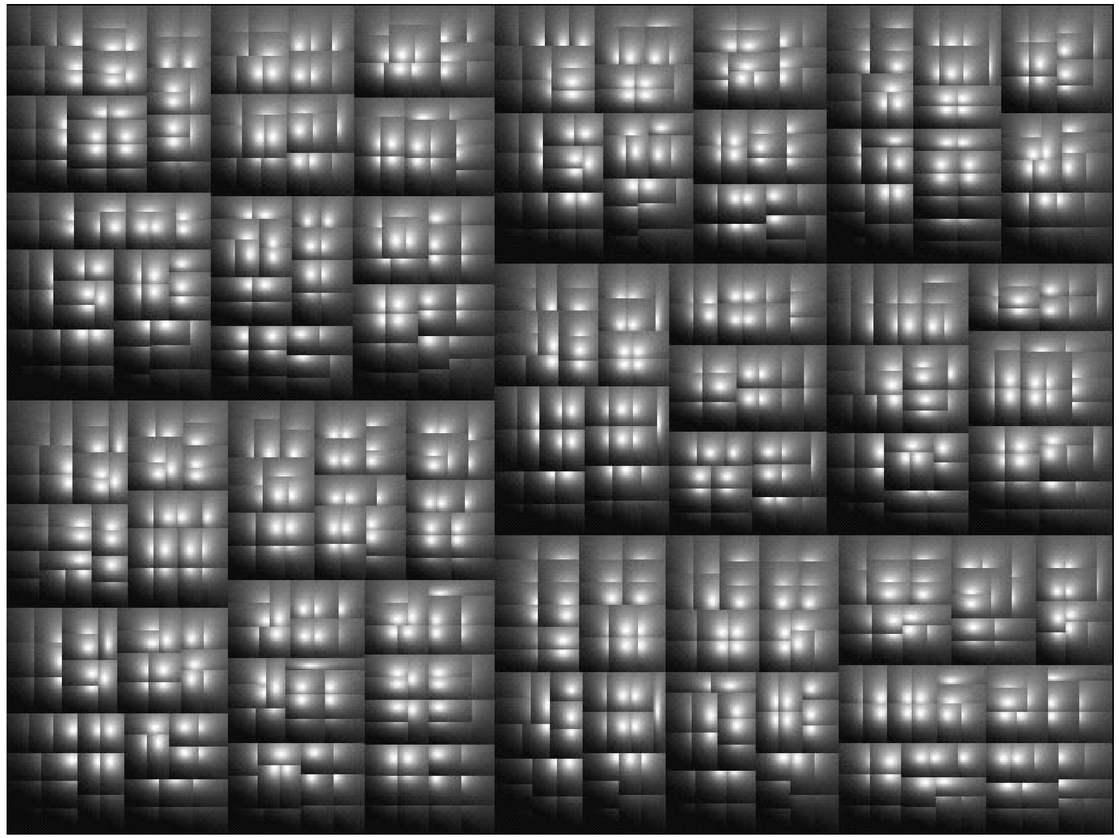


[Disk Inventory X]

Squarified + Cushioned Treemaps



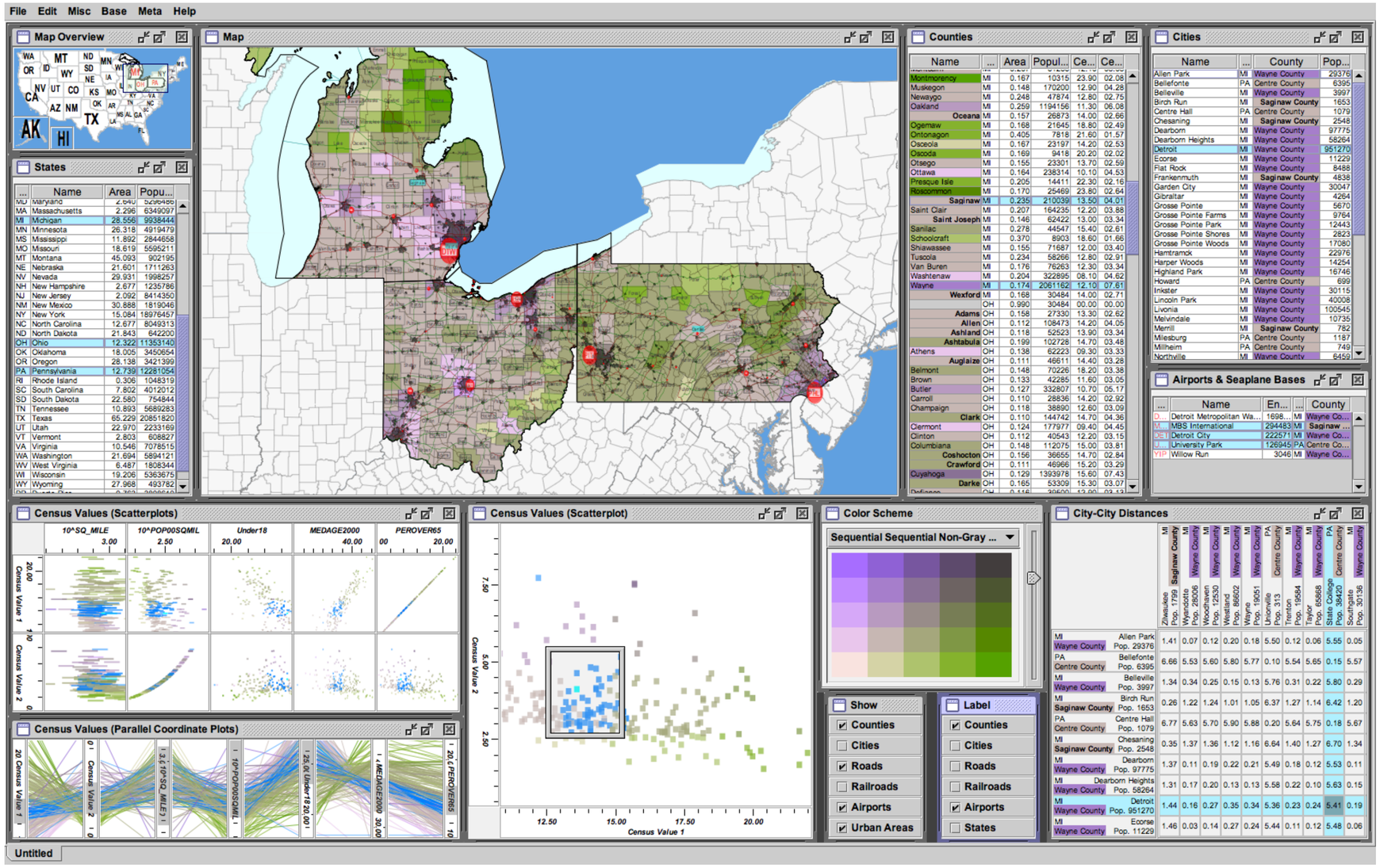
(a) File system



(b) Organization

[Brus et al., 1999]

Multiple Views



[Improvise, Weaver, 2004]

Multiple Views

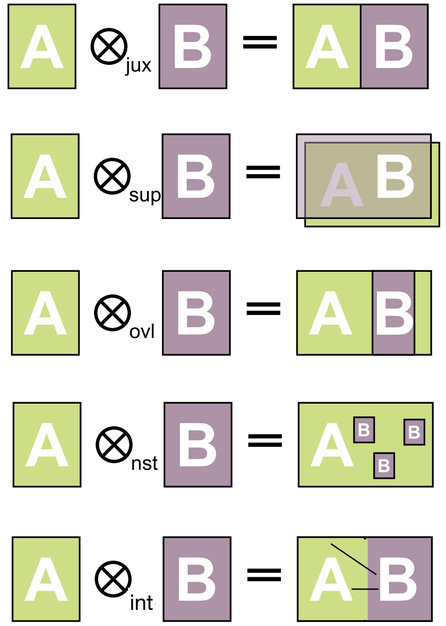
- Why have just one visualization?
- Sometimes data is best examined in more than one view
  - Clutter/visual overload
  - Different attributes (cannot show all attributes in one view)
  - Different scales (task requires overview or detail)
  - Different encodings (no single encoding is optimal for all tasks)
- Eyes Beat Memory (Ch. 6)
  - Aiding working memory:
    side-by-side/layers > animated > jump cuts
  - Showing all visual elements at once → don't need to remember

Multiple Views

- Big questions:
  - How to partition display or layer views?
  - How to coordinate views (e.g. navigation, selection)?
  - What data is shared?

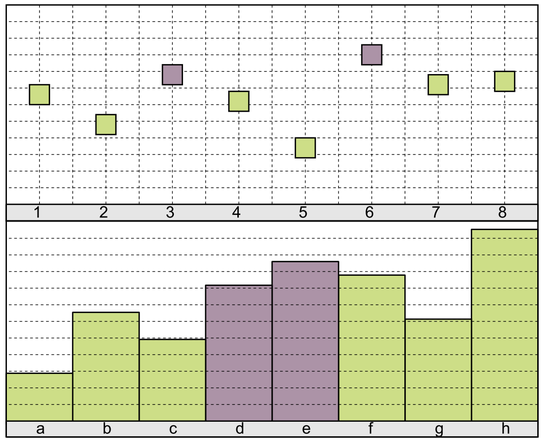
Design Space of Composite Visualization

- Composite visualization views (CVVs)
  - Includes Coordinated multiple views (CMV)
  - + More!
- Design Patterns:
  - Juxtaposition: side-by-side
  - Superimposition: layers
  - Overloading: vis meshed with another
  - Nesting: vis inside a vis (recursive vis)
  - Integration: "merge" views + links

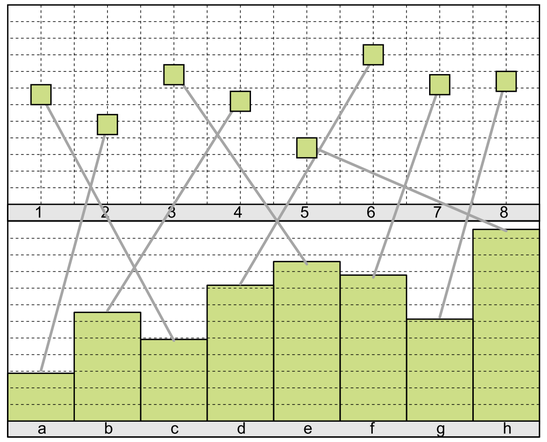


[W. Javed and N. Elmqvist, 2012]

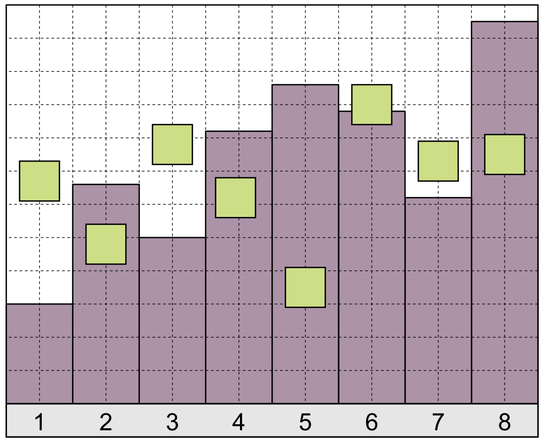
Composite Visualization Techniques



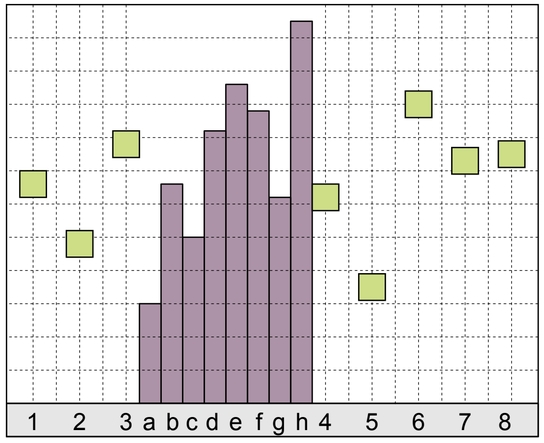
(a) Juxtaposed views.



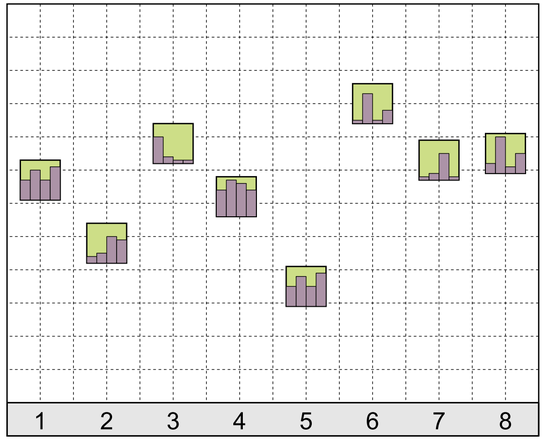
(b) Integrated views.



(c) Superimposed views.



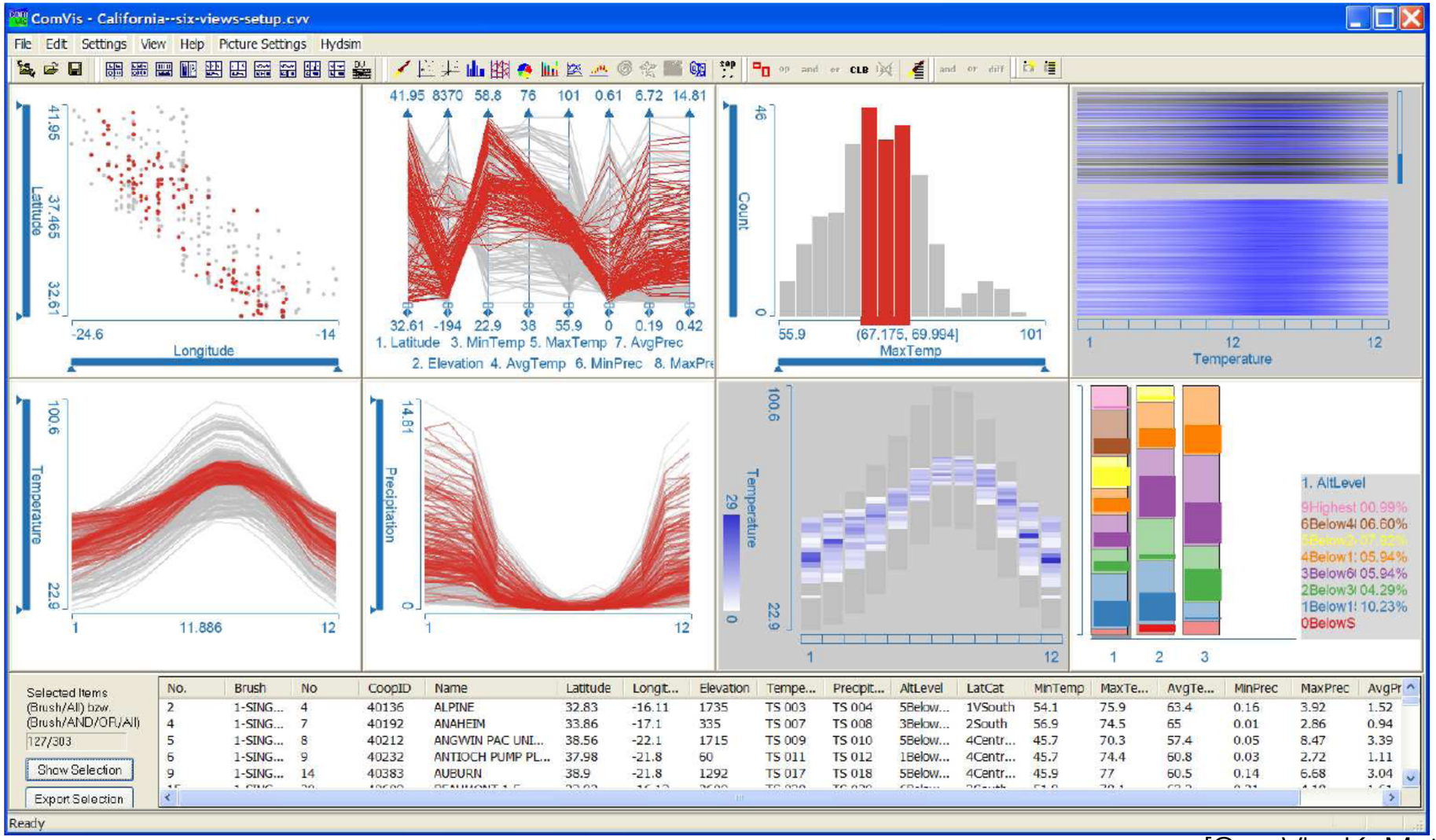
(d) Overloaded views.



(e) Nested views.

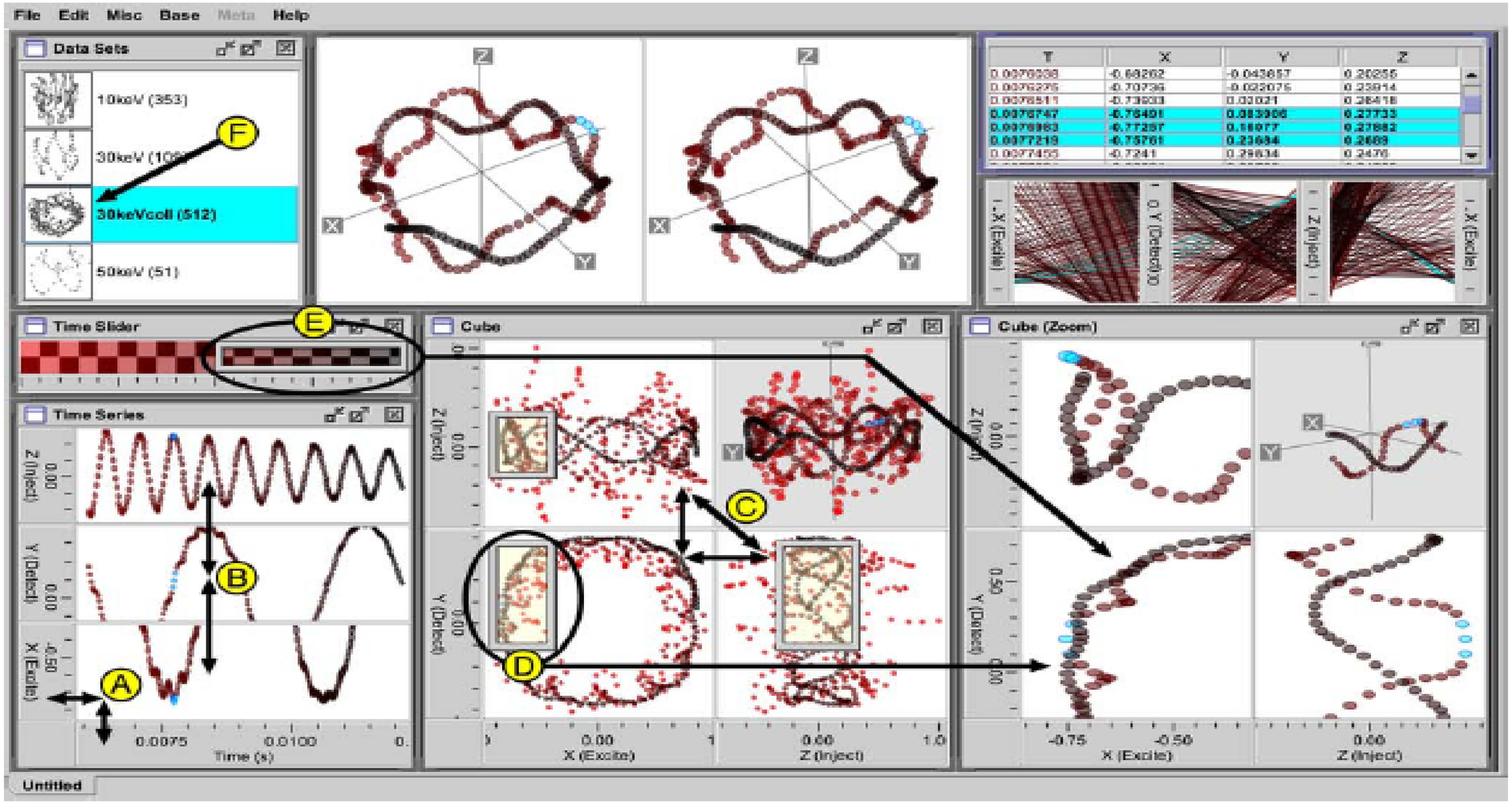
[W. Javed and N. Elmqvist, 2012]

Juxtaposition



[ComVis, K. Matkovic et al., 2008]

Juxtaposition



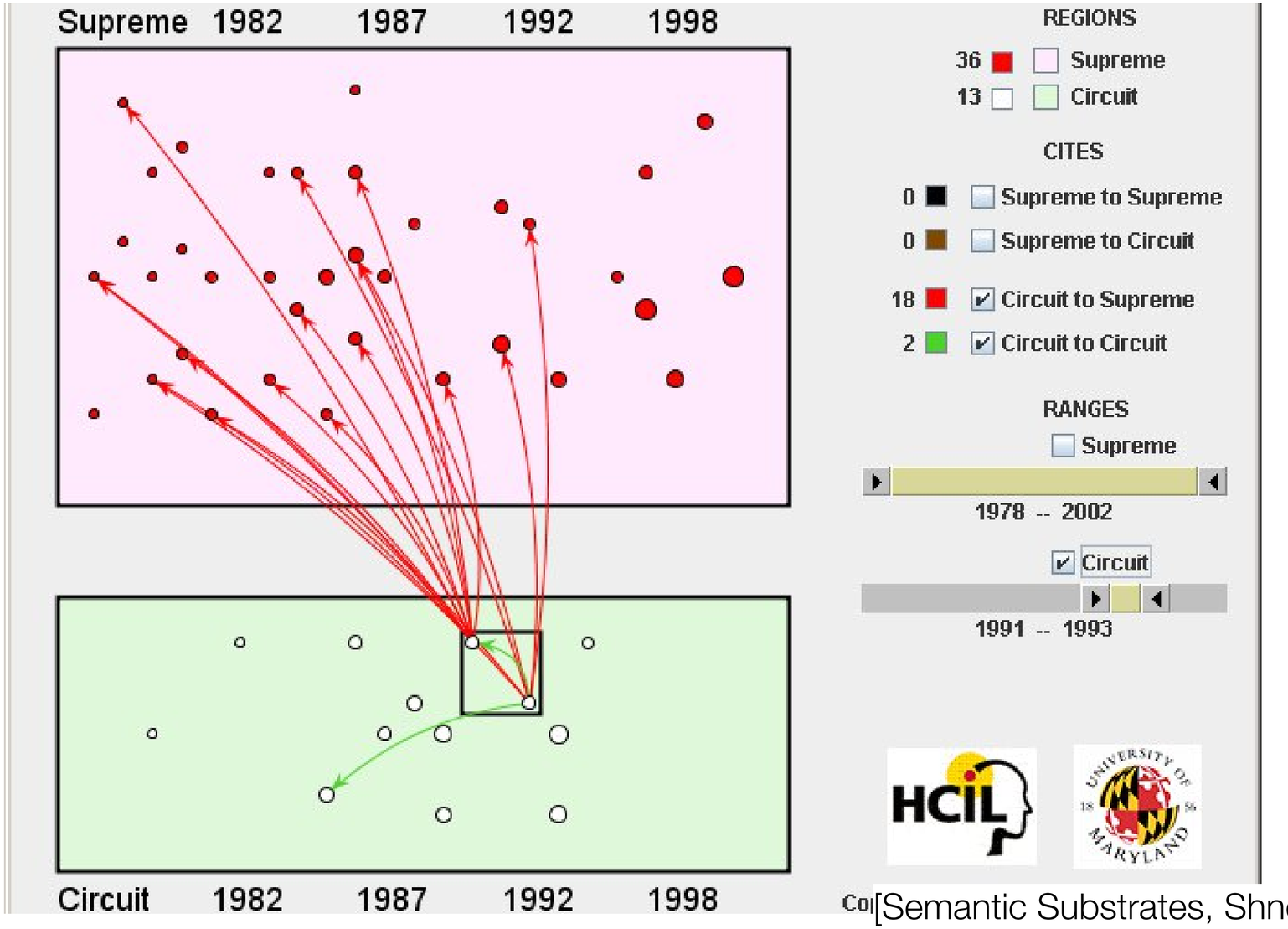
[Improvise, C. Weaver, 2004]

Juxtaposition Guidelines

- Benefits:
  - The component visualizations are independent and can be composed without interference
  - Easy to implement
- Drawbacks:
  - Implicit visual linking is not always easy to see, particularly when multiple objects are selected
  - Space is divided between the views, yielding less space for each view
- Applications: Use for heterogeneous datasets consisting of many different types of data, or for where different independent visualizations need to be combined.

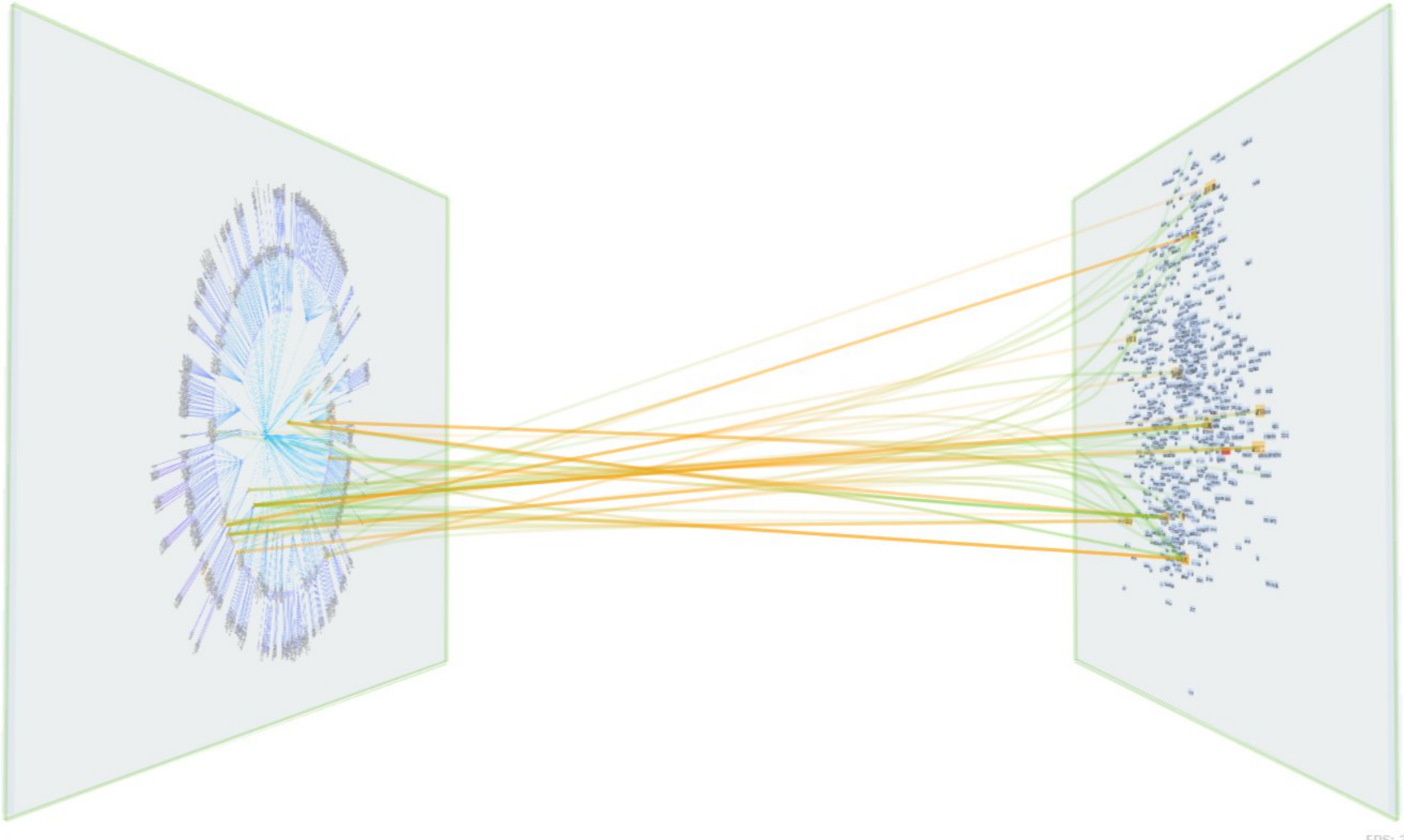
[W. Javed and N. Elmqvist, 2012]

Integration



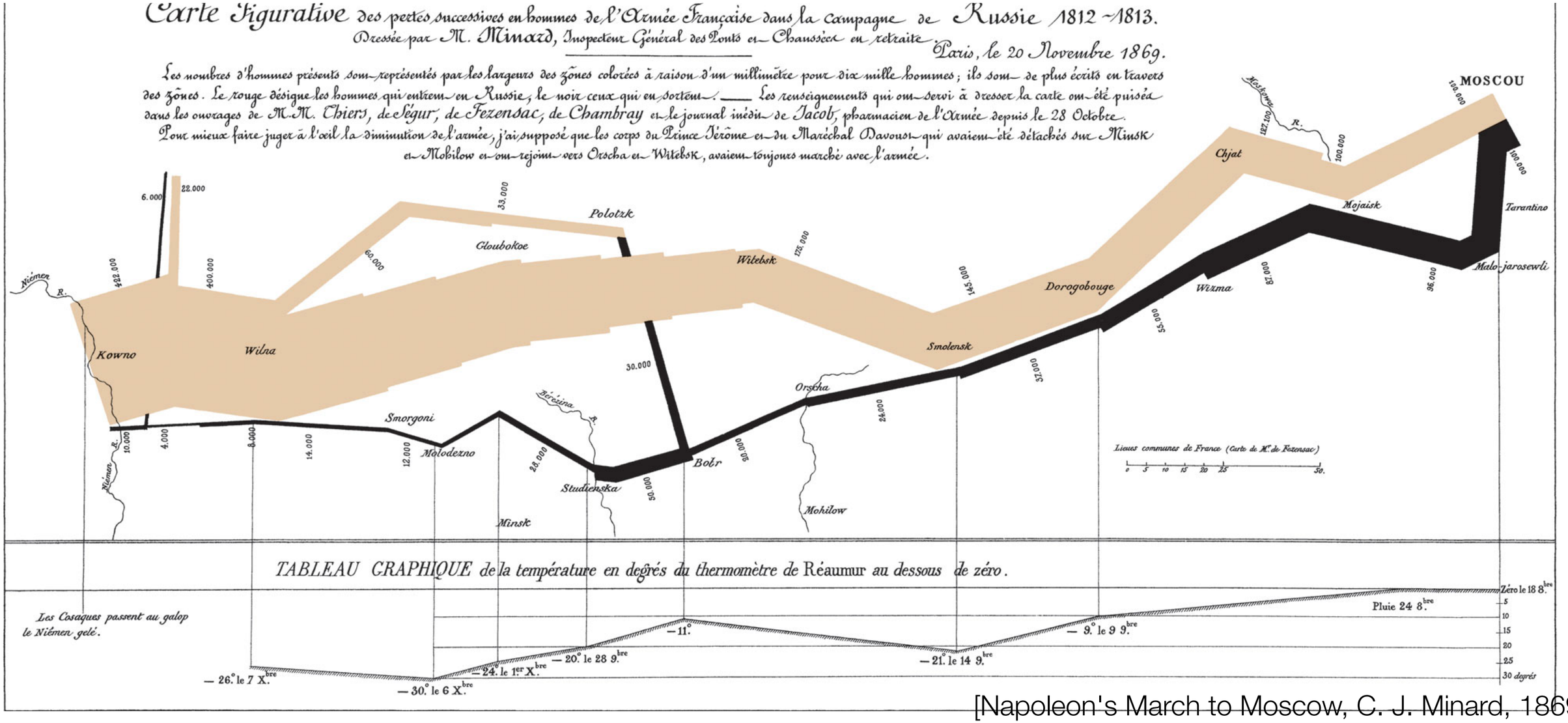
[Semantic Substrates, Shneiderman and Aris, 2006]

Integration



[VisLink, Collins and Carpendale, 2007]

Integration



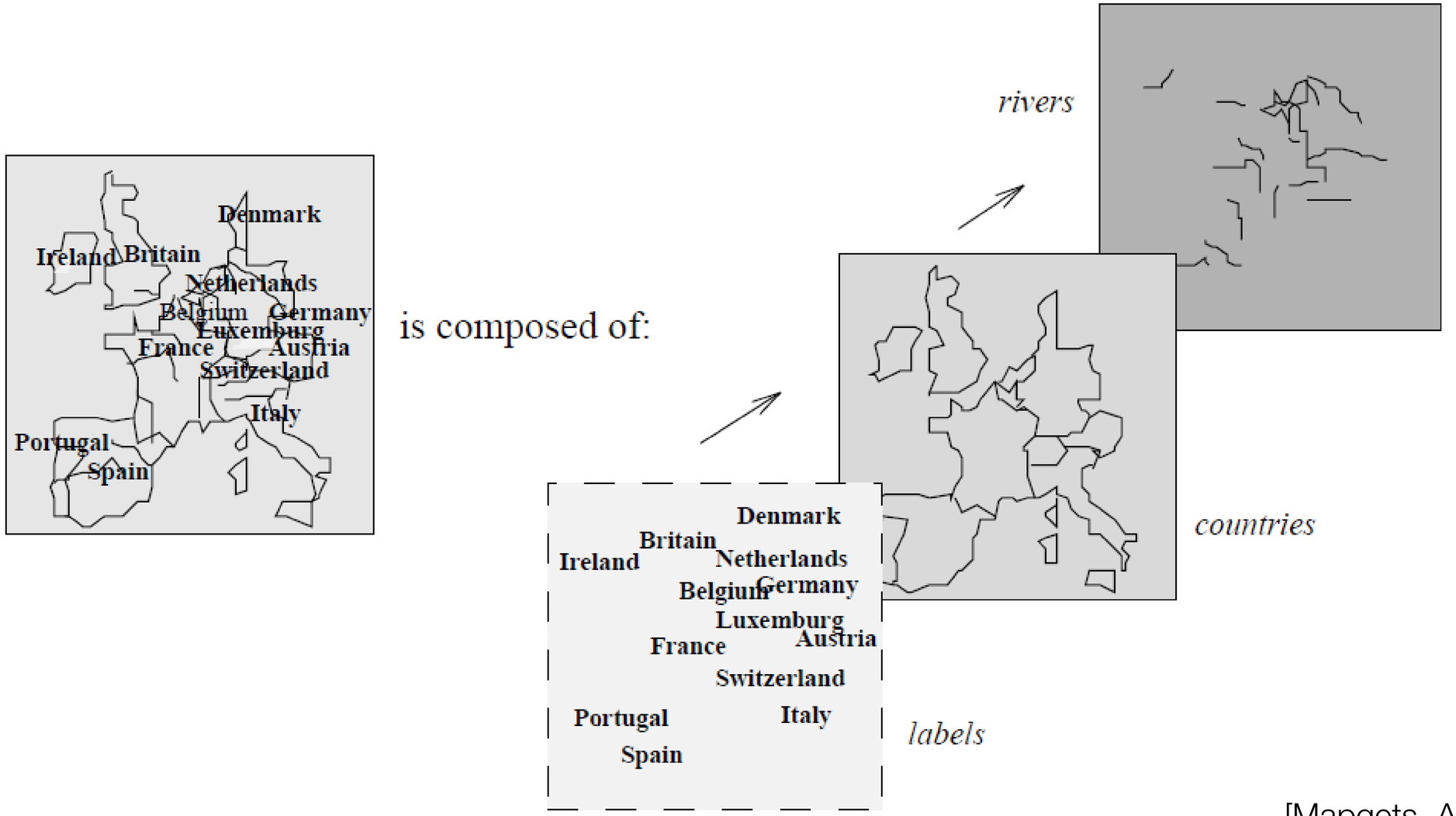
[Napoleon's March to Moscow, C. J. Minard, 1869]

Integration Guidelines

- Benefits:
  - Easy to perceive one-to-one and one-to-many relations between items in components
  - Visualizations are less independent compared to juxtaposed views, but still separate
- Drawbacks:
  - Extra visual clutter added to the overall view
  - Display space is split between the views
  - Some dependencies exist between views to allow for the visual linking
- Applications: Use for heterogeneous datasets where correlation and comparisons between views is particularly important.

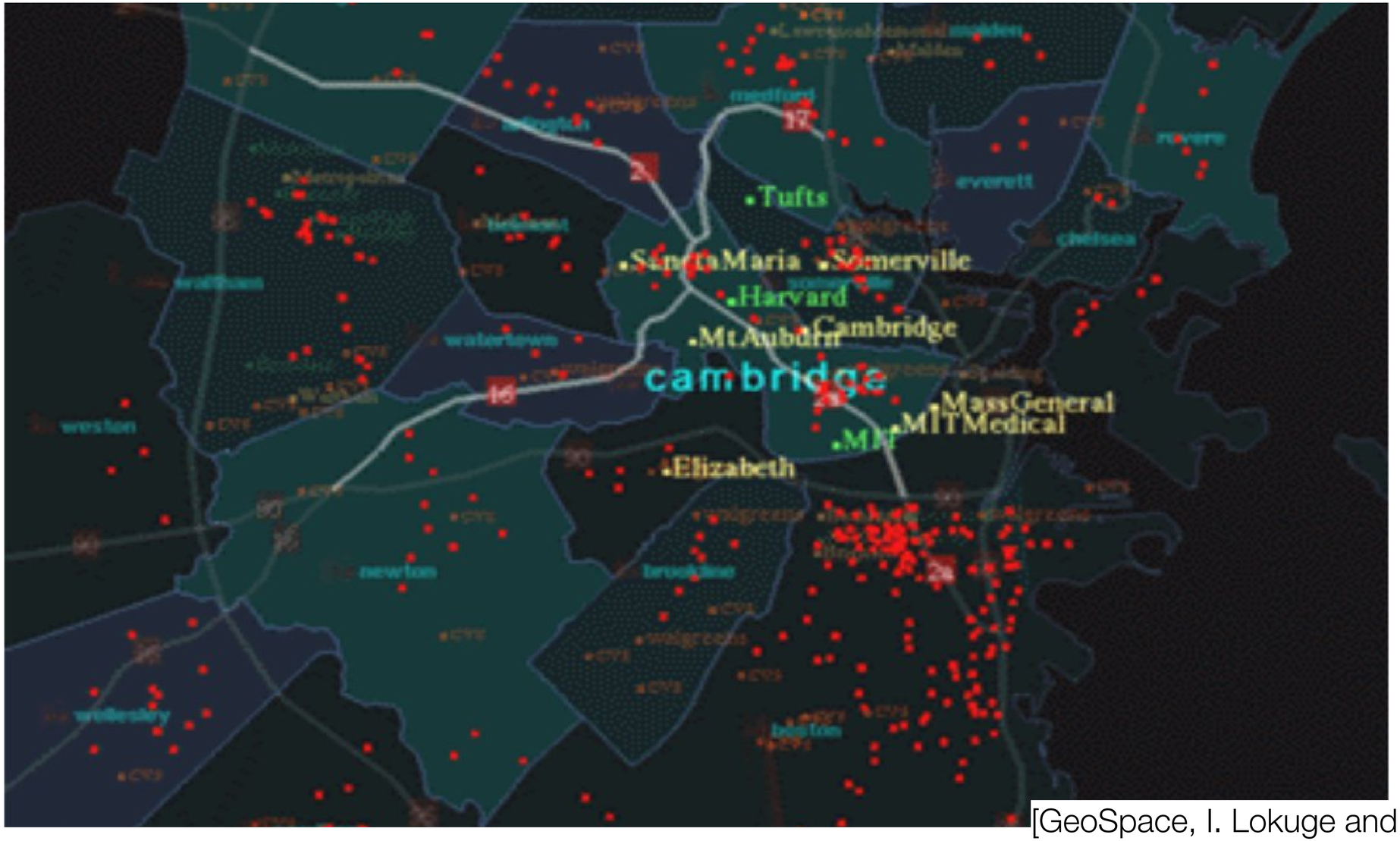
[W. Javed and N. Elmqvist, 2012]

Superimposition



[Mapgets, A. Voisard, 1995]

Superimposition



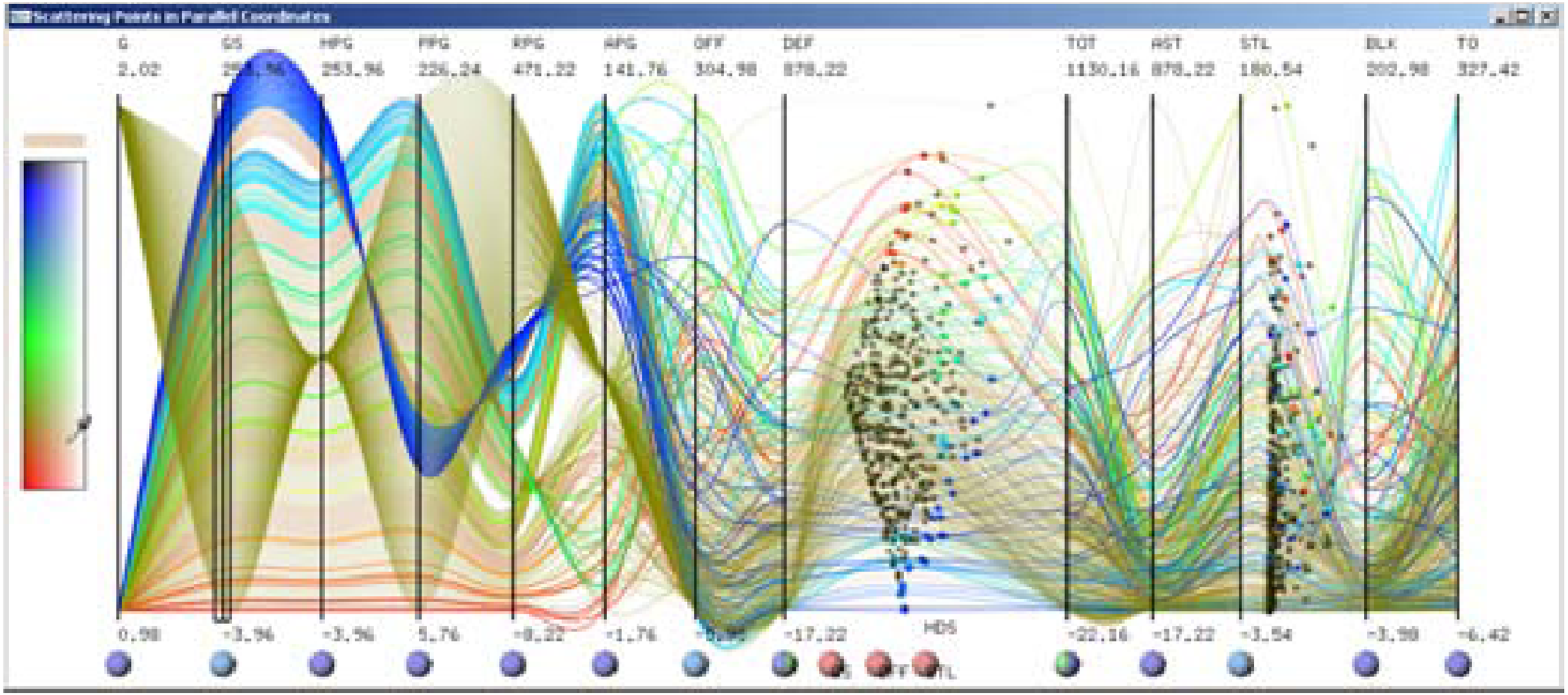
[GeoSpace, I. Lokuge and S. Ishizaki, 1995]

Superimposition Guidelines

- Benefits:
  - Allows direct comparison in the same visual space.
- Drawbacks:
  - May cause occlusion and high visual clutter.
  - The client visualization must share the same spatial mapping as the host visualization.
- Applications: In settings where comparison is common, or where the component visualization views need to be as large as possible (potentially the entire available space).

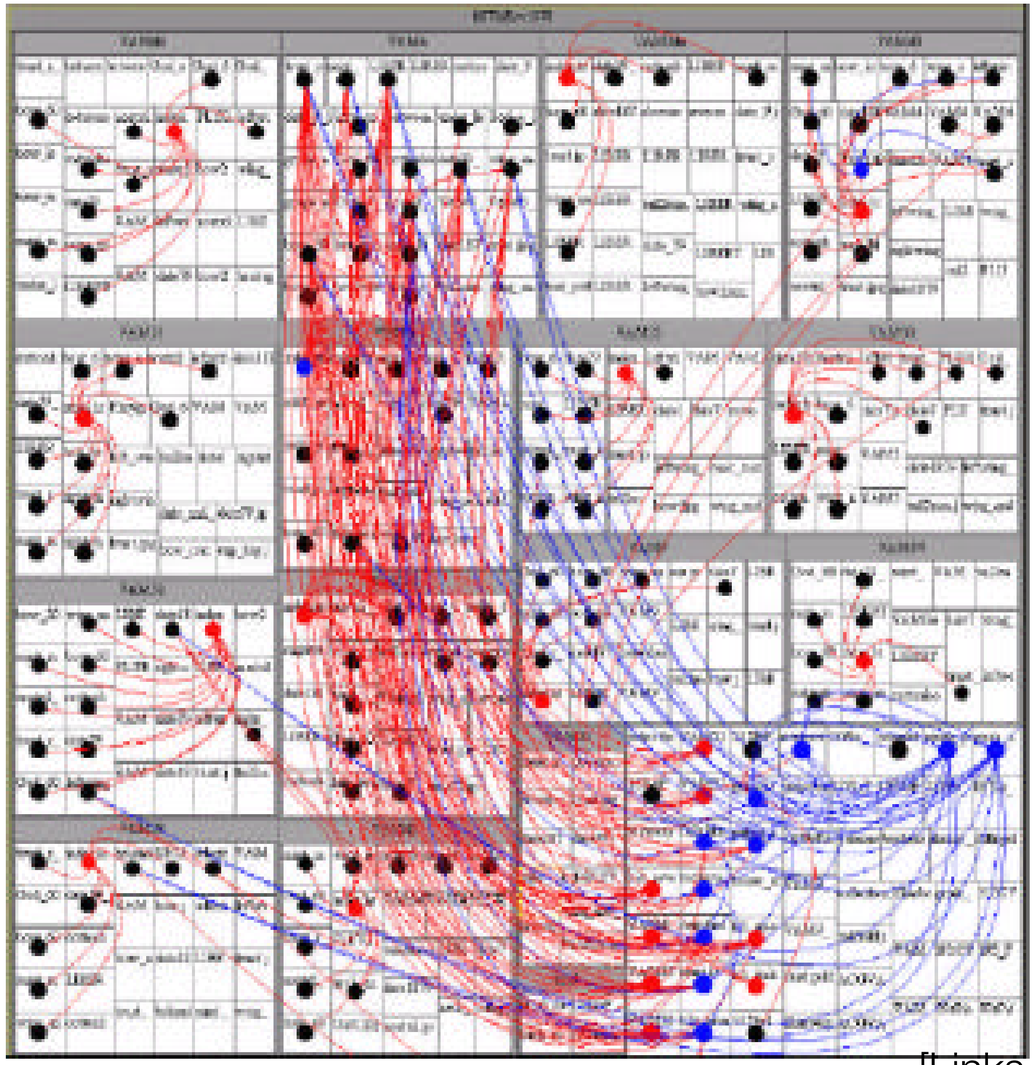
[W. Javed and N. Elmqvist, 2012]

Overloading



[SPCC, X. Yuan et al., 2009]

Overloading



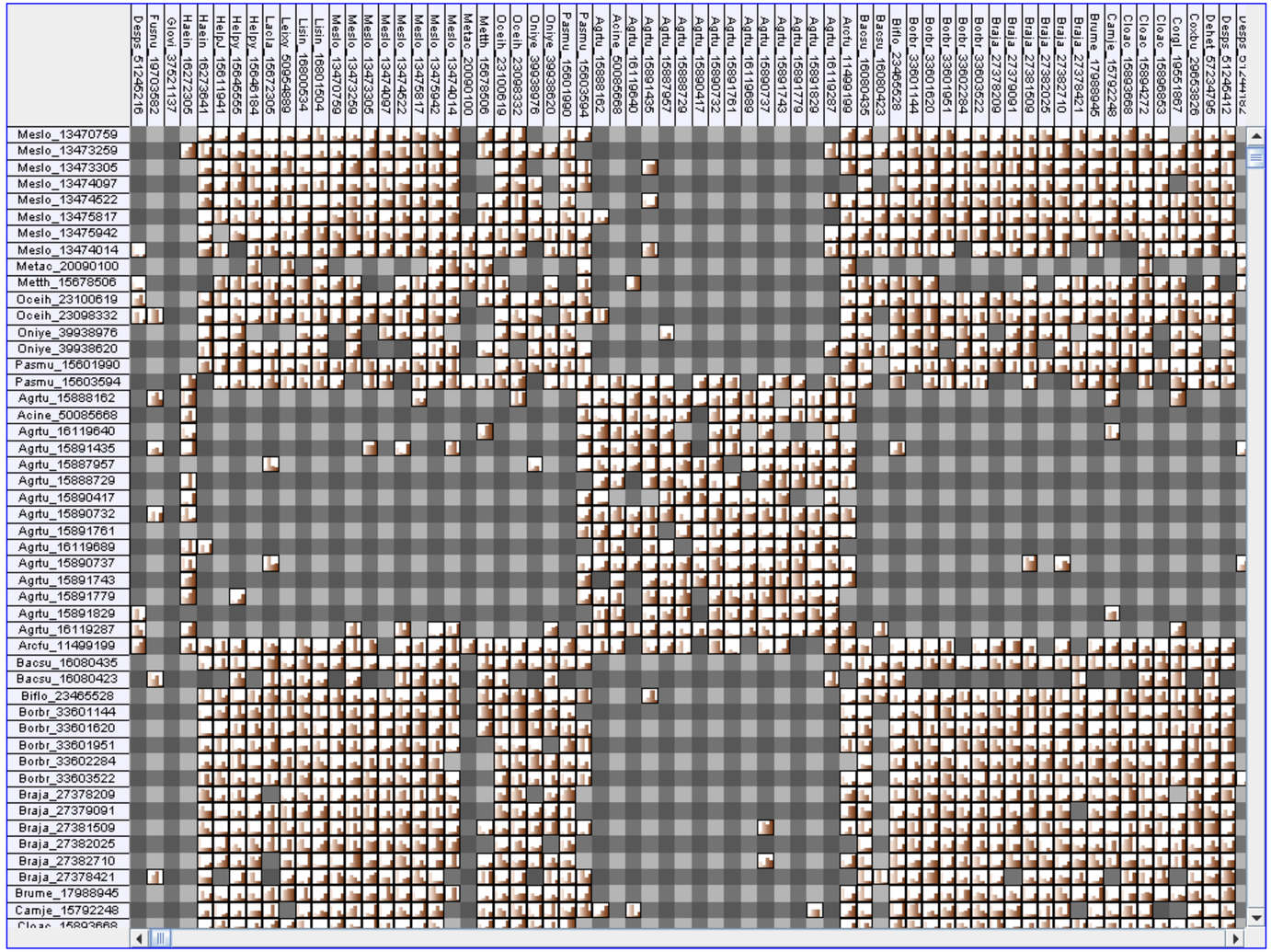
[Links on Treemaps, J.-D. Fekete et al., 2003]

Overloading Guidelines

- Benefits:
  - The client visualization does not have to share the same coordinate space as the host visualization
  - This also yield more flexibility and control over visual clutter
- Drawbacks:
  - Visual clutter is increased
  - Visual design dependencies between components are significant
- Applications: Situations where one visualization can be folded into another to yield a compact (and complex) result.

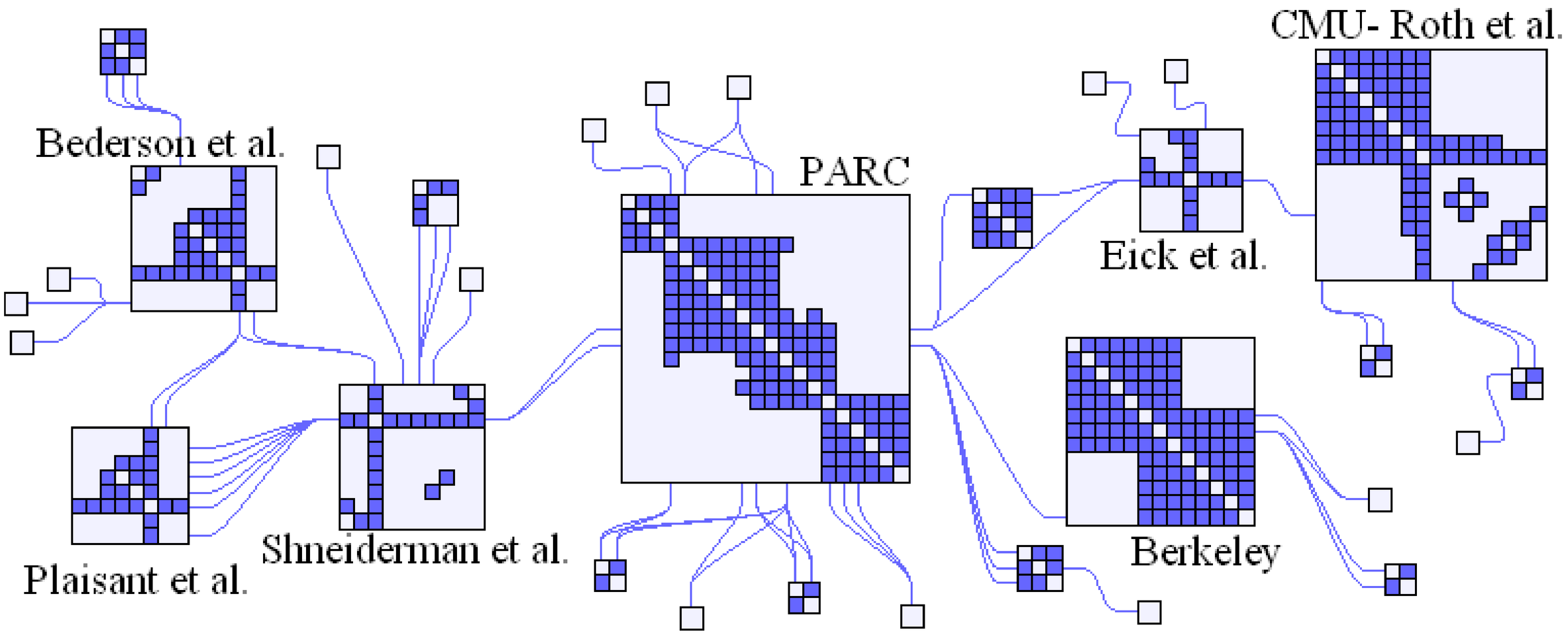
[W. Javed and N. Elmqvist, 2012]

Nesting



[ZAME, N. Elmqvist et al., 2008]

Nesting



[NodeTrix, N. Henry et al., 2007]

Nesting Guidelines

- Benefits:
  - Very compact representation
  - Easy correlation
- Drawbacks:
  - Limited space for the client visualizations
  - Clutter is high
  - Visual design dependencies are high
- Applications: Situations that call for augmenting a particular visual representation with additional mapping

[W. Javed and N. Elmqvist, 2012]

Design Space

- Visualizations: the techniques or idioms used
- Spatial relation: relationship between visual structures in display space
- Data relation: visual relationship between items in different views
  - None: No relation
  - Item-item: One-to-one
  - Item-group: One-to-many
  - Item-dimension: Item in one view is a **scale** in another

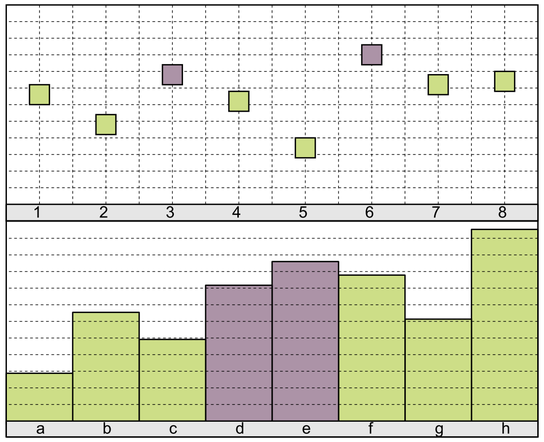
[W. Javed and N. Elmqvist, 2012]

Summary

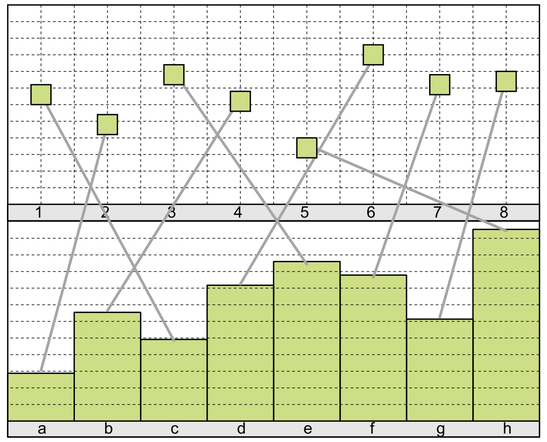
Technique	Visualization A	Visualization B	Spatial Relation	Data Relation
ComVis [24] (Figure 2)	any	any	juxtapose	none
Improvise [39] (Figure 3)	any	any	juxtapose	none
Jigsaw [36]	any	any	juxtapose	none
Snap-Together [30]	any	any	juxtapose	none
semantic substrates [34] (Figure 4)	node-link	node-link	juxtapose	item-item
VisLink [11] (Figure 5)	radial graph	node-link	juxtapose	item-item
Napoleon's March on Moscow [37]	time line view	area visualization	juxtapose	item-item
Mapgets [38] (Figure 6)	map	text	superimpose	item-item
GeoSpace [22] (Figure 7)	map	bar graph	superimpose	item-item
3D GIS [8]	map	glyphs	superimpose	item-item
Scatter Plots in Parallel Coordinates [45] (Figure 8)	parallel coordinate	scatterplot	overload	item-dimension
Graph links on treemaps [14] (Figure 9)	treemap	node-link	overload	item-item
SparkClouds [21]	tag cloud	line graph	overload	item-item
ZAME [13] (Figure 10)	matrix	glyphs	nested	item-group
NodeTrix [17] (Figure 11)	node-link	matrix	nested	item-group
TimeMatrix [44]	matrix	glyphs	nested	item-group
GPUVis [25]	Scatterplot	glyphs	nested	item-group

[W. Javed and N. Elmqvist, 2012]

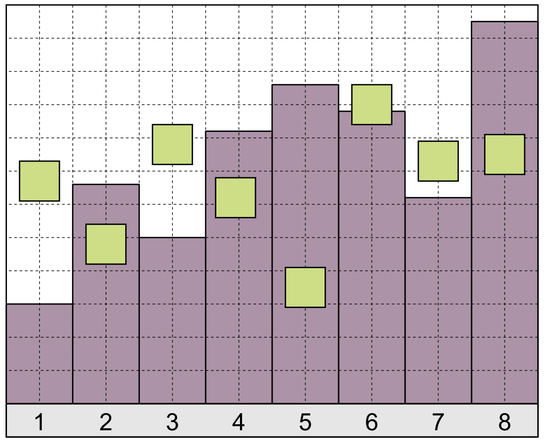
Summary (Scatterplot + Bar Chart)



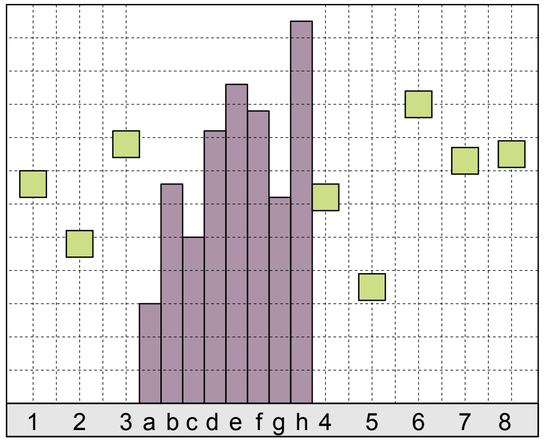
(a) Juxtaposed views.



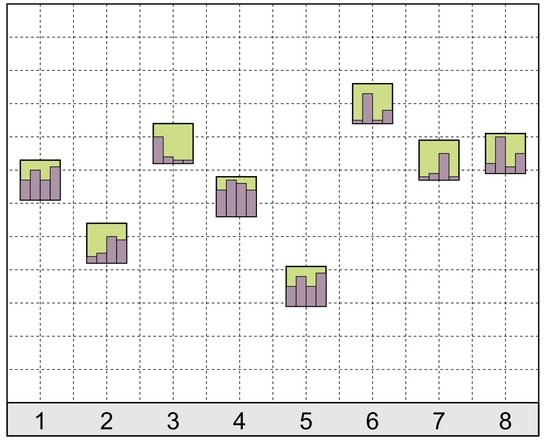
(b) Integrated views.



(c) Superimposed views.



(d) Overloaded views.



(e) Nested views.

[W. Javed and N. Elmqvist, 2012]