

# Graphs & Trees

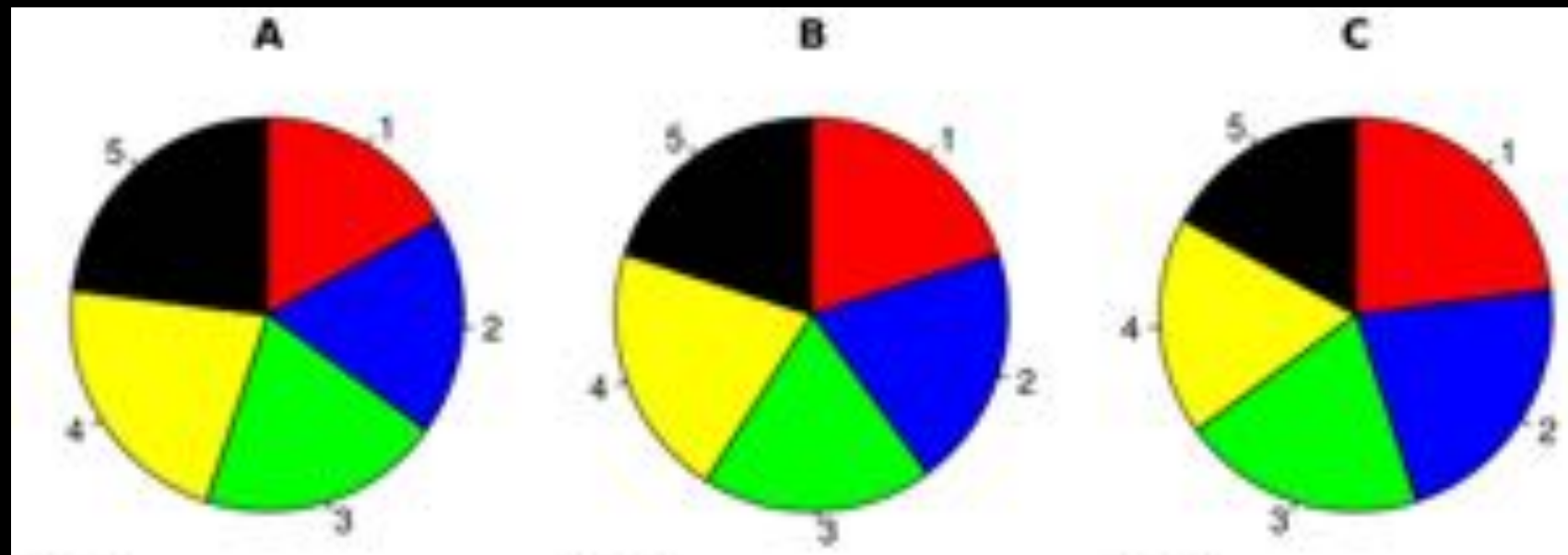
James EAGAN

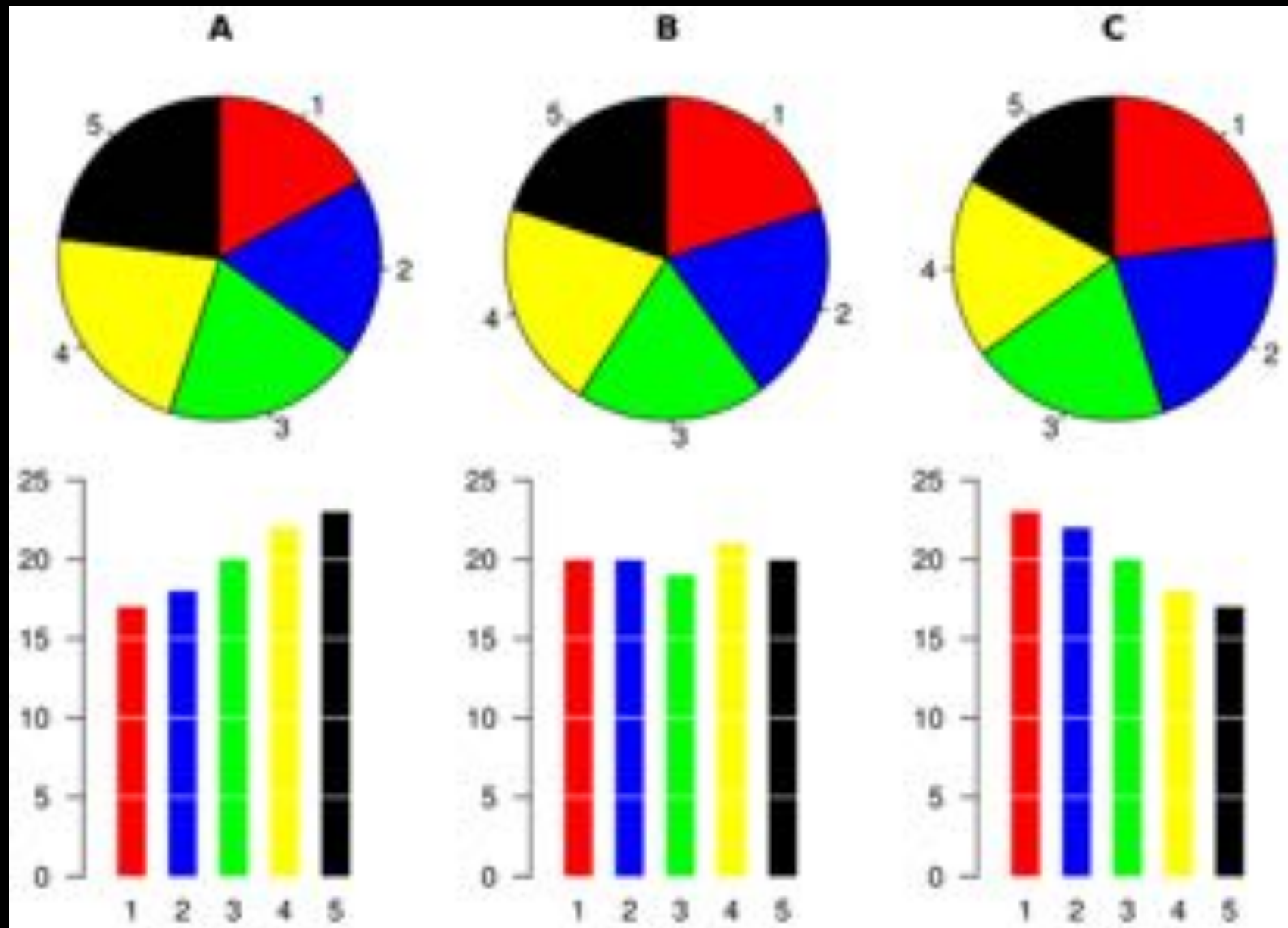


Includes content adapted  
from John Stasko (Georgia Tech)



Dernière mise à jour : mai 2017





# Connections are everywhere

- Circle of friends
- SNCF train network
- ...

# These connections form a graph

# What is a graph?

- *vertices and edges*

*a.k.a.*

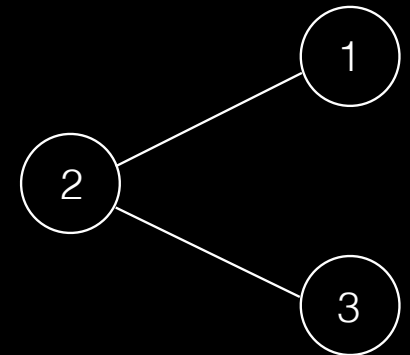
- *nodes and links*

	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

Adjacency matrix

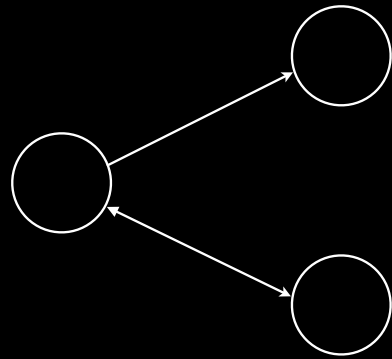
1: 2  
2: 1, 3  
3: 2

Adjacency list

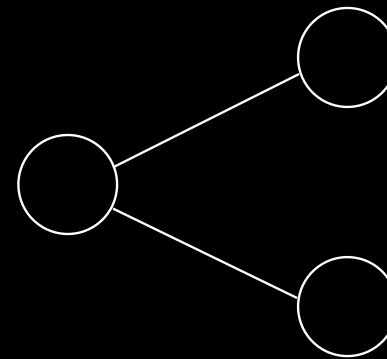


Drawing



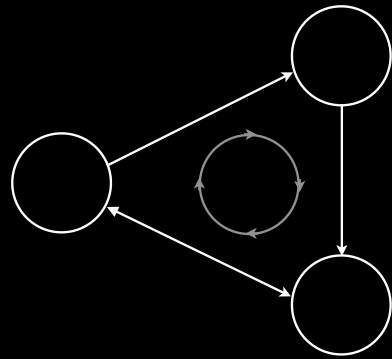


Directed

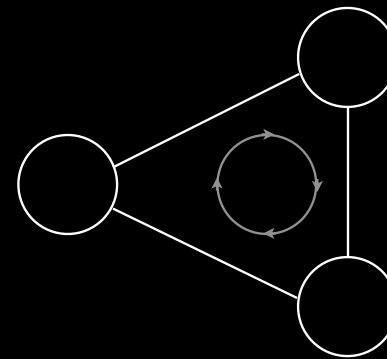


Undirected



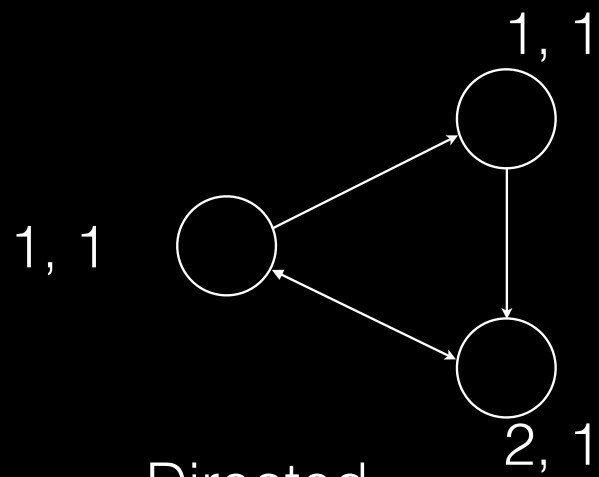


Directed

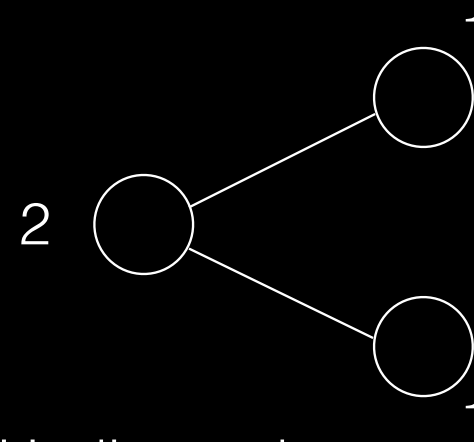


Undirected

Cycles

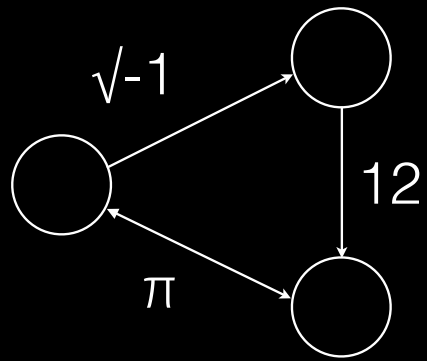


Directed

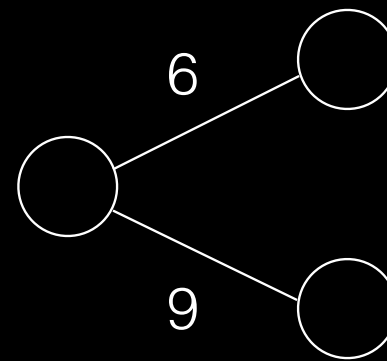


Undirected

Degree

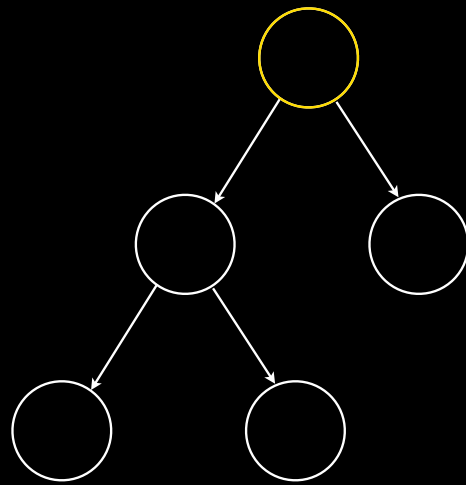


Directed



Undirected

Weights



**Coming soon...**

Trees

# What can we model as a graph?

- European/Global Telephone system
- World Wide Web
- Retail distribution networks
- Software call graph
- Semantic map in an AI algorithm
- Social connectivity
- ...

# Graphs are abstract mathematical objects

(But we want a concrete rendering.)

# Scaling graph layout algorithms is hard

(Not just hard, but NP-hard.)



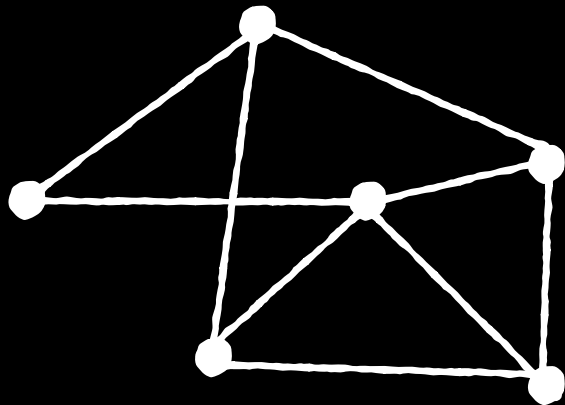
# **Users' focus frequently changes**

(But small changes can have big effects on layout.)

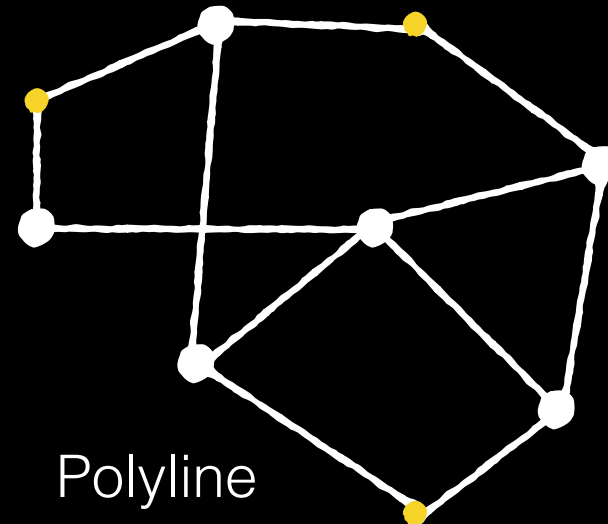
# Graph drawing community focuses just on layout

- [www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf](http://www.cs.brown.edu/people/rt/papers/gd-tutorial/gd-constraints.pdf)
- di Battista, Eades, Tamassia & Tollis, “Graph Drawing”, Prentice Hall, 1999

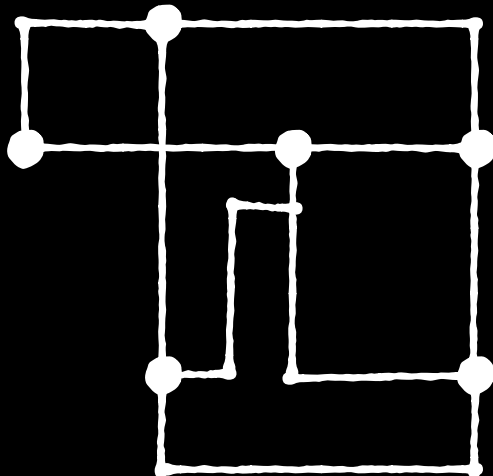




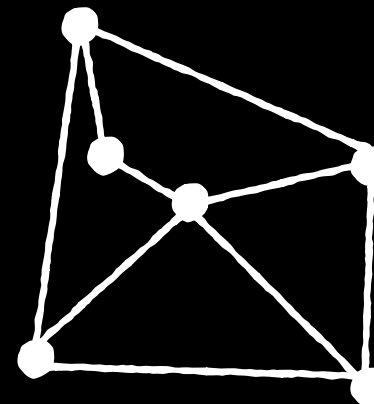
Node-link



Polyline



Orthogonal



Planar straight-line

# Vertices can have various properties

(Shape, color, size, location, label, ...)

# Edges, too

(Color, size, label, form, ...)

# What factors make a graph drawing look good?

(Edge crossings, edge length, area, bends, ...)

# Minimize number of edge crossings

(Ideally a planar layout.)



# Minimize total edge length

(Toward a proper scale.)

# Minimize area

(But still maintain inter-node spacing.)

# Avoid long edges

(Minimize maximum edge length.)

# Keep edge lengths uniform

(Minimize variance.)

# Avoid bends

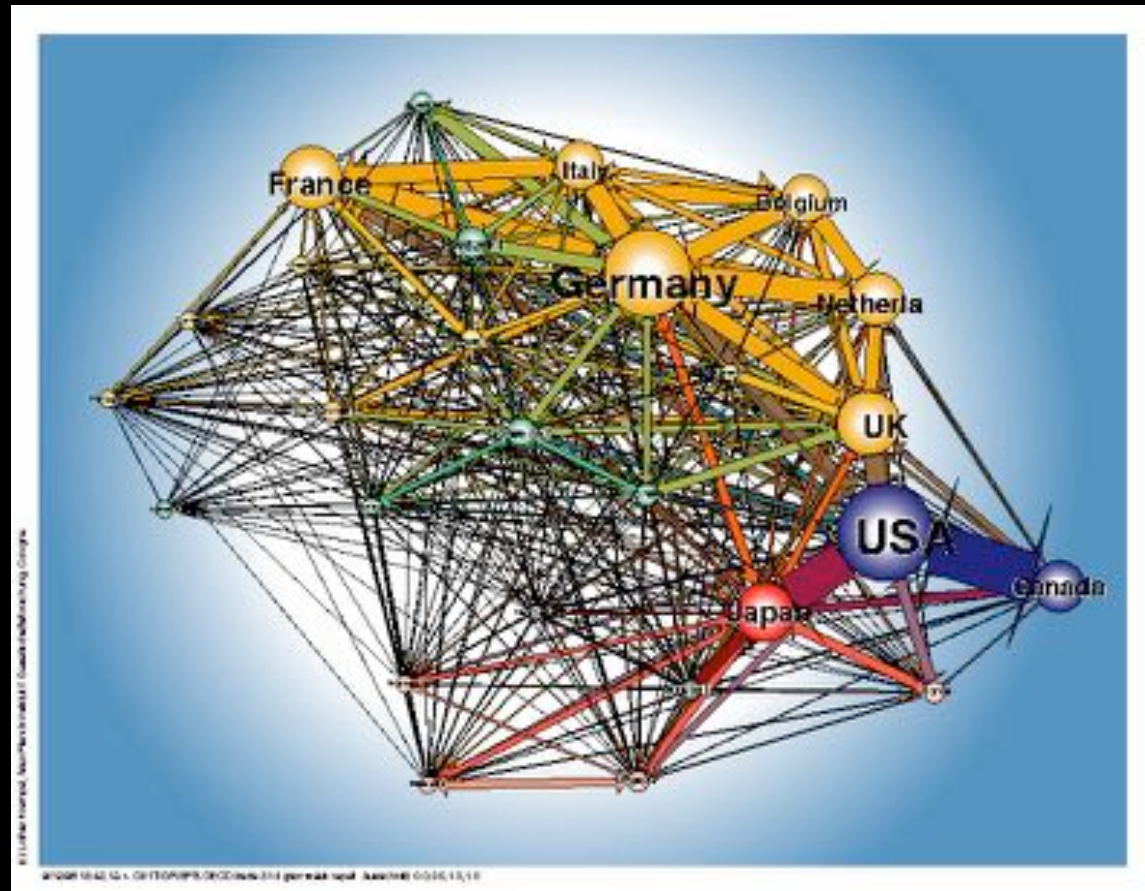
(Minimize orthogonal edges toward straight lines.)

# Which of these matters?

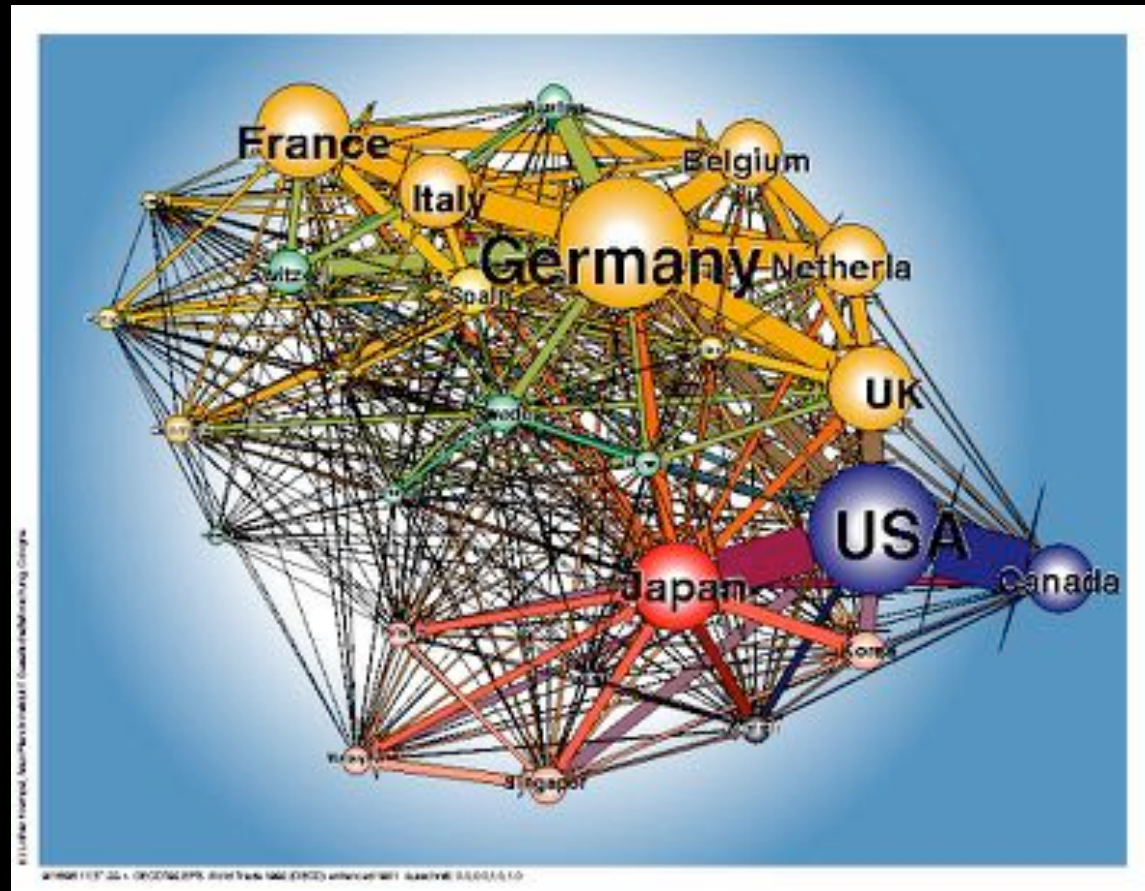
- Purchase, Graph Drawing 1997
- Ware et al., Info Vis 1(2), 2002
- Ghoniem et al., Info Vis 4(2), 2005

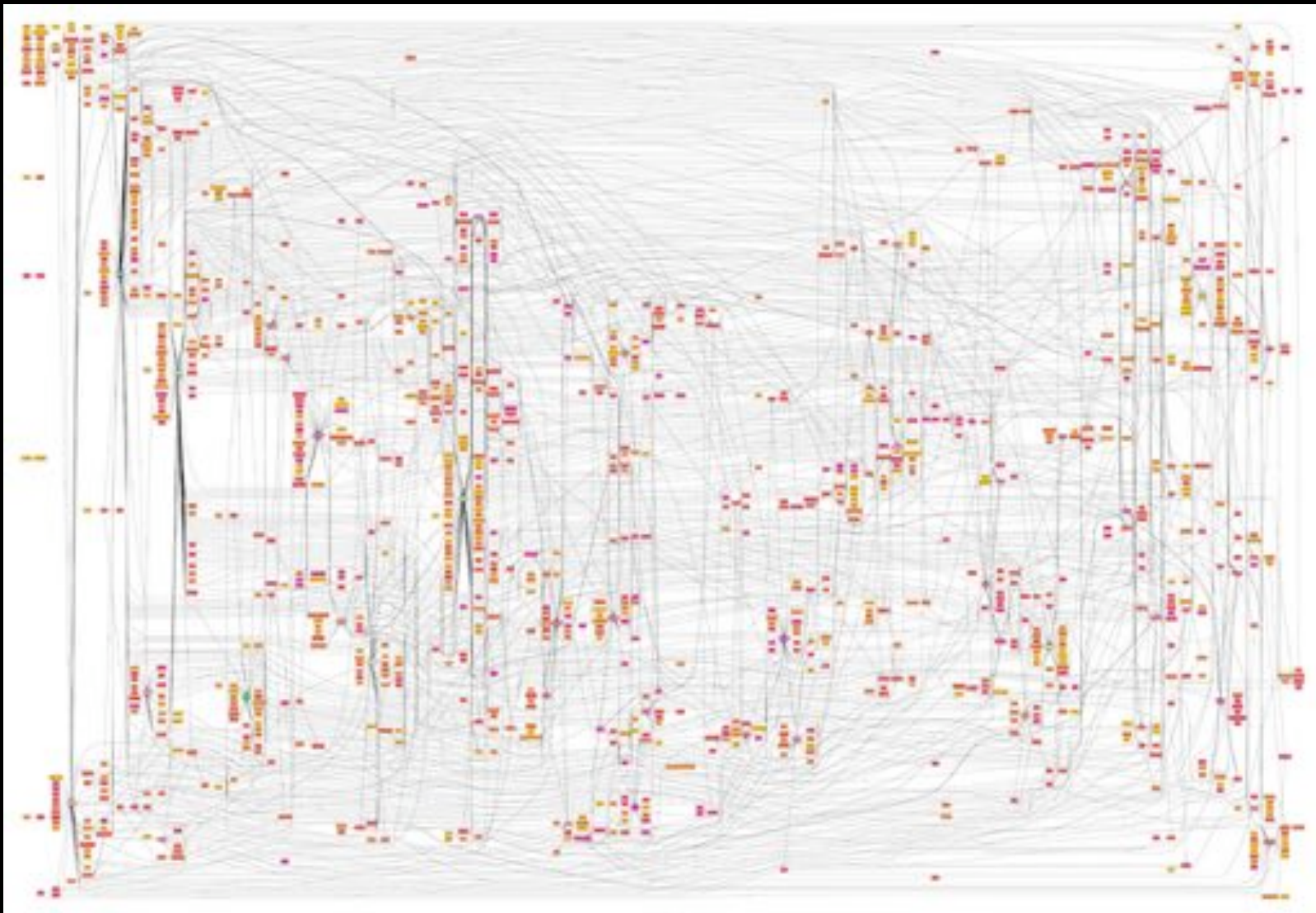
# But how does it scale?

- Drawing might turn into a “ball of string.”
- Algorithm complexity is usually  $\notin P$ .







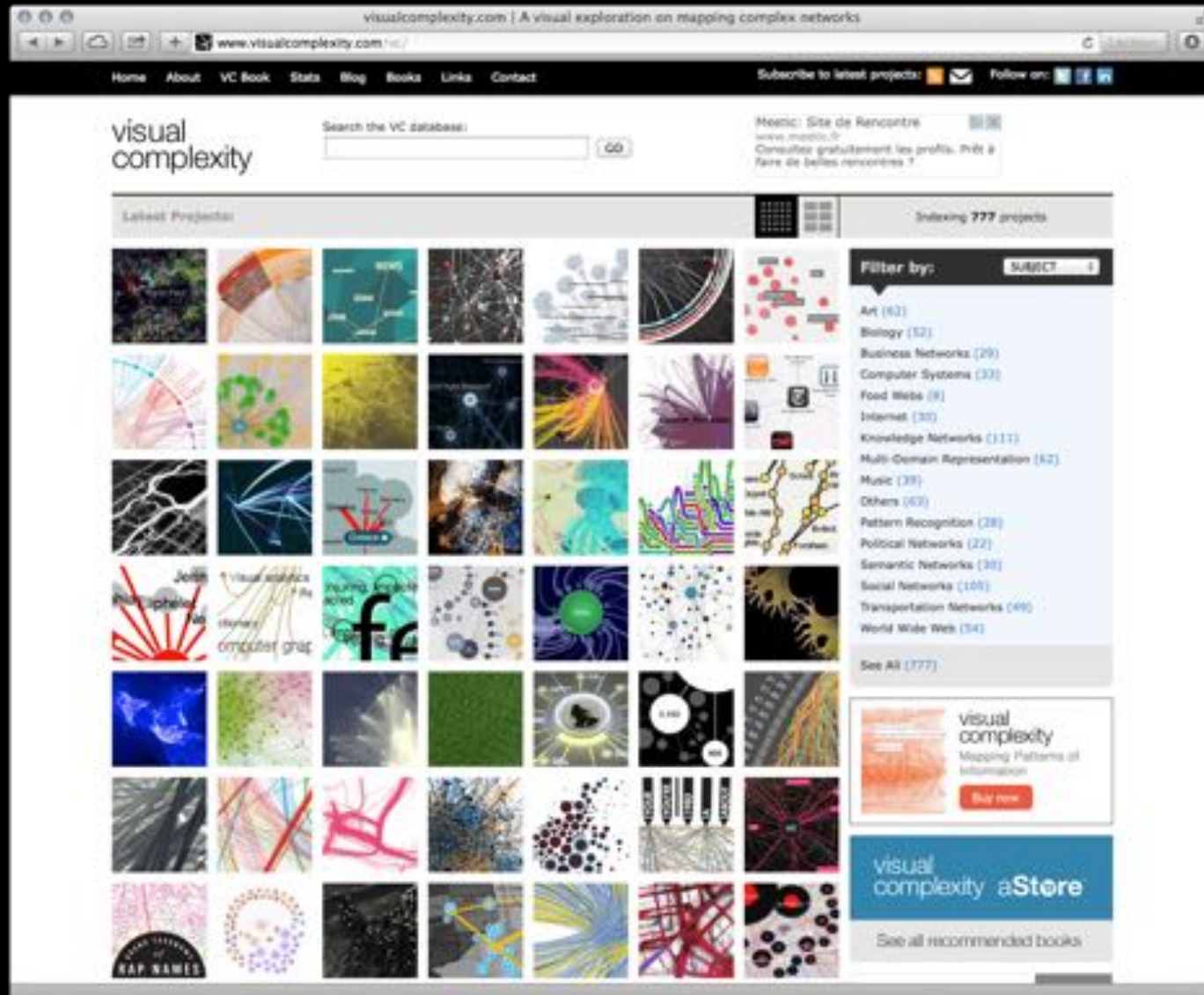


[ Charles Isbell, Cobot ]

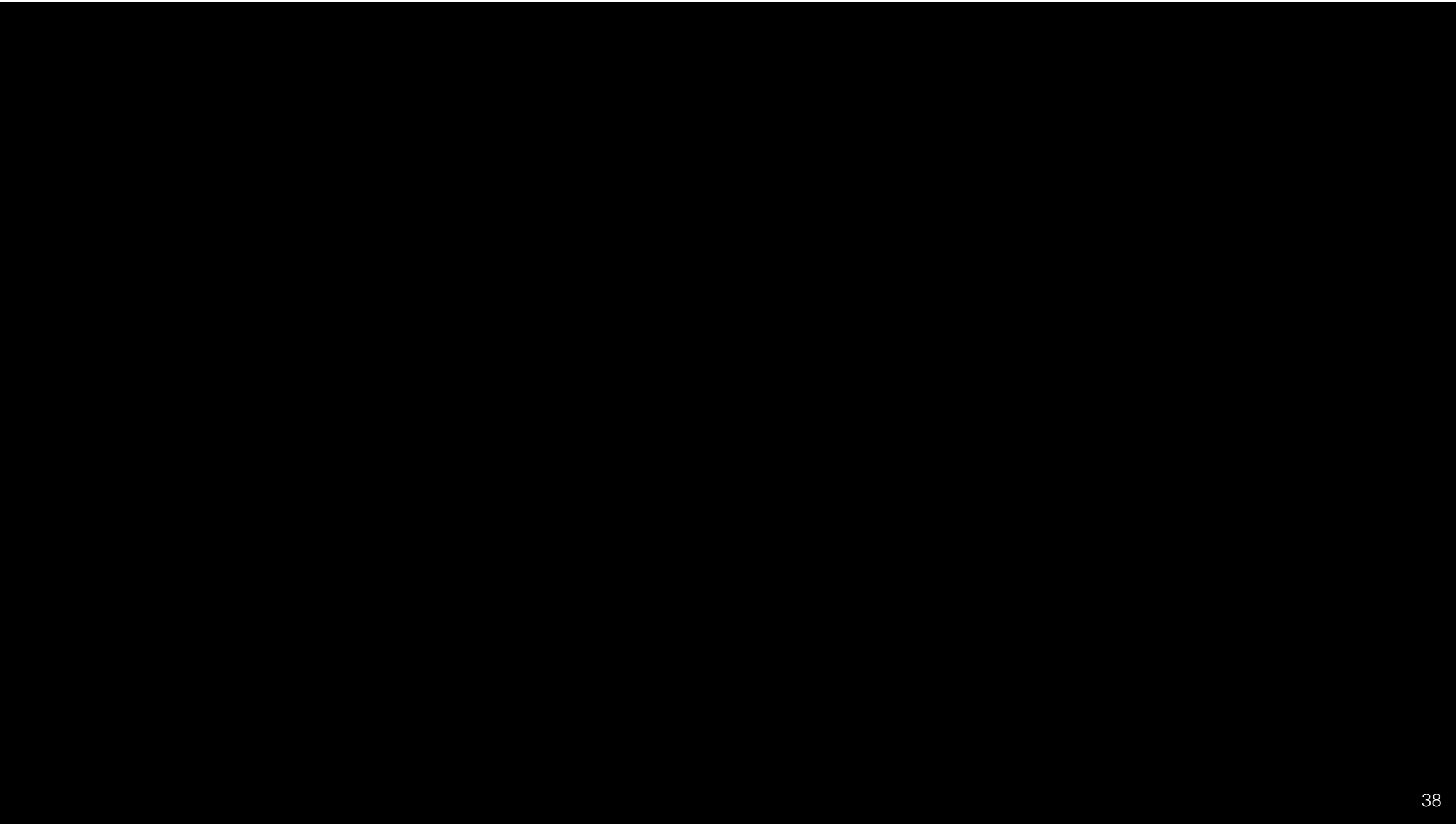
# Clustering can sometimes help

Aggregate highly connected subgraphs.

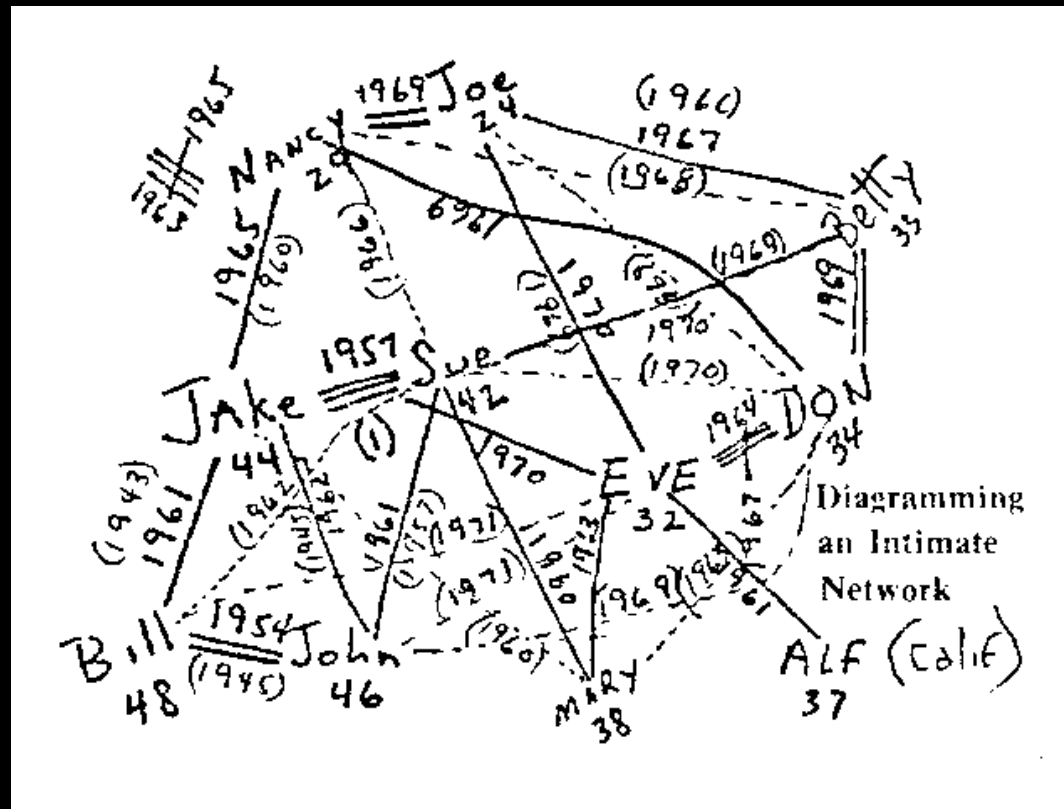




[ www.visualcomplexity.com ]



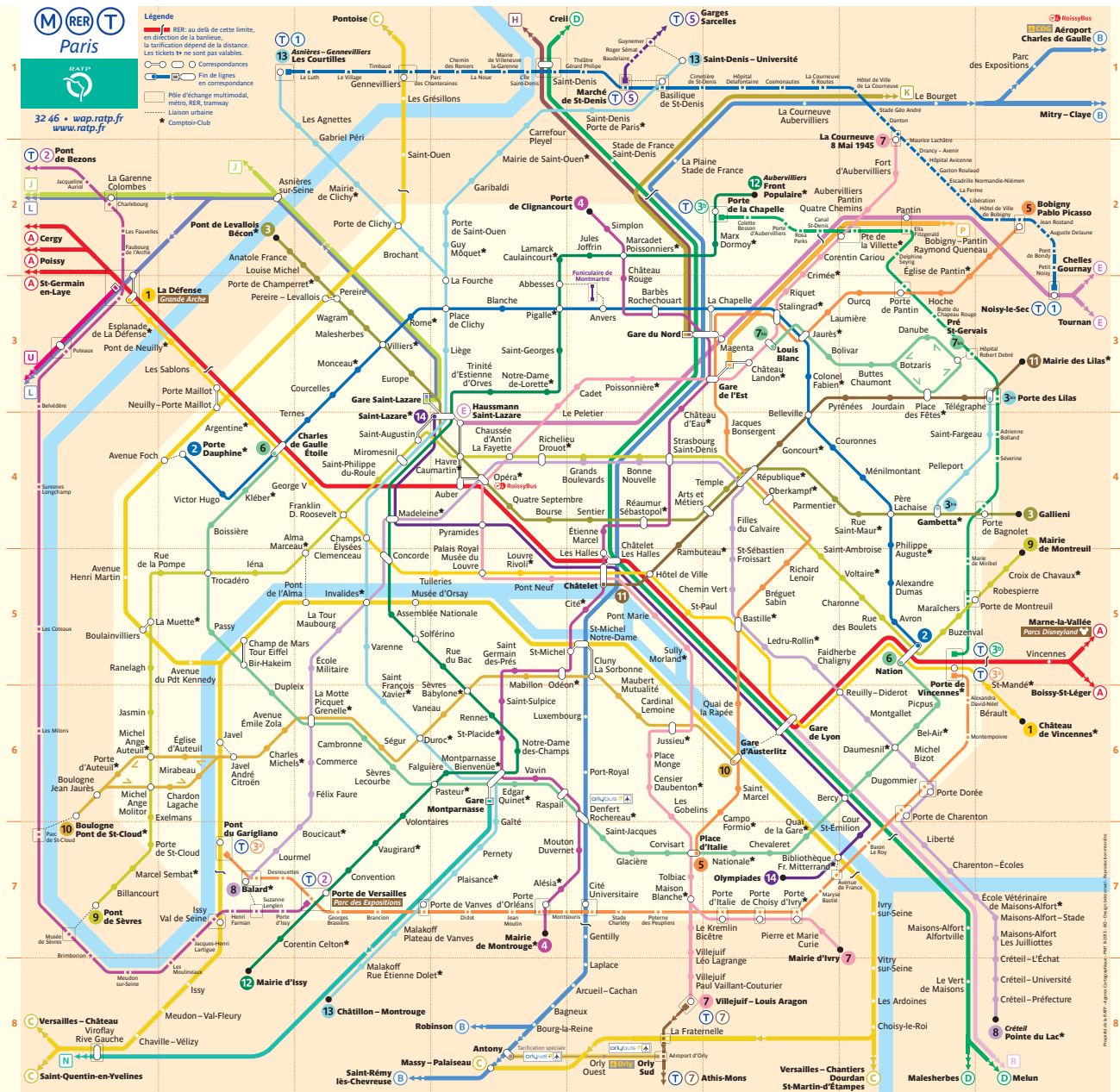










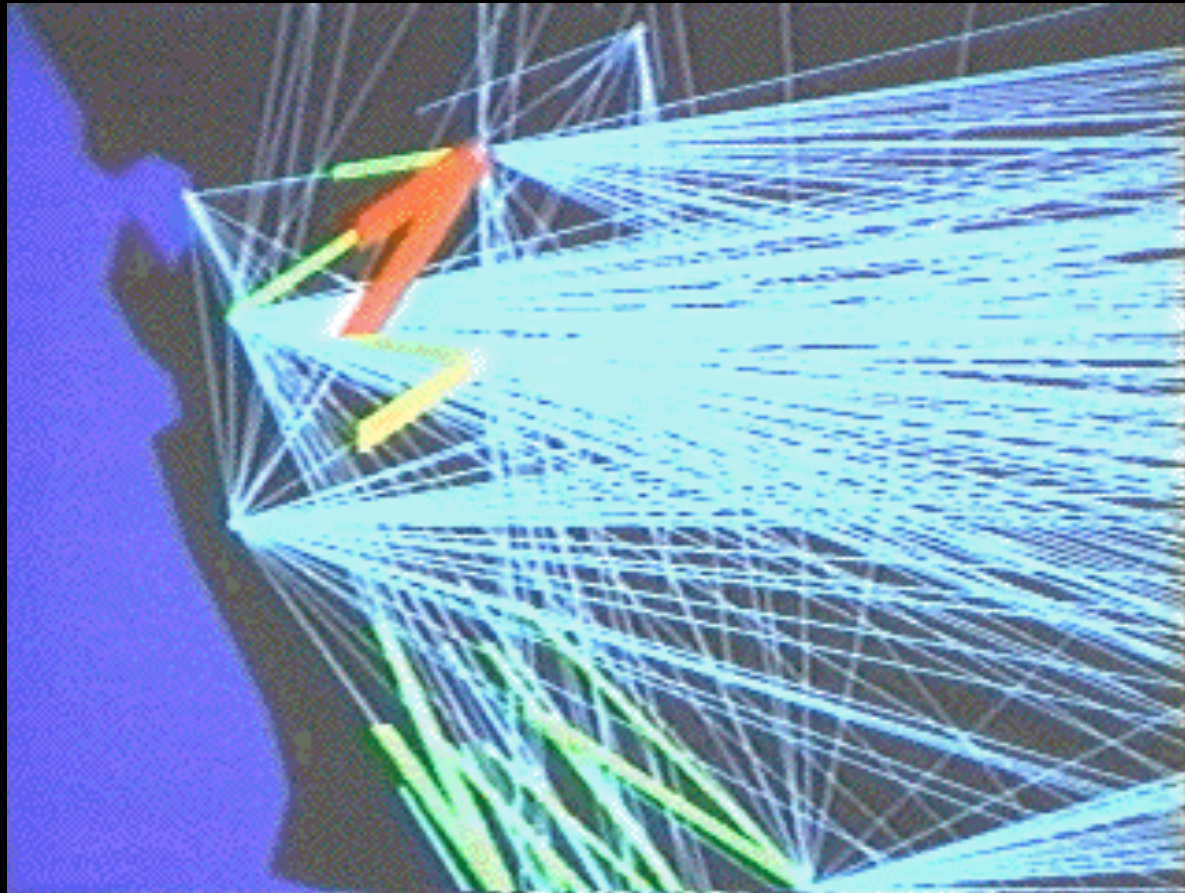




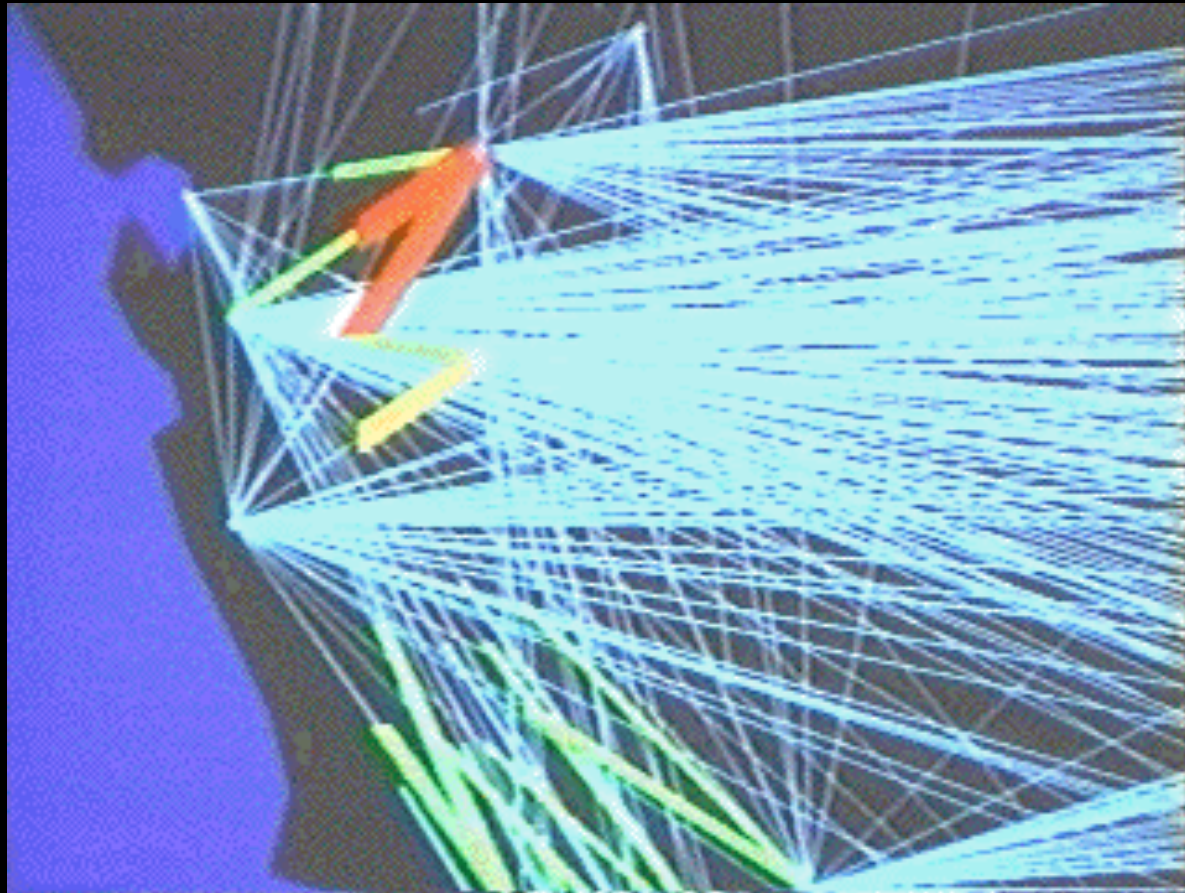


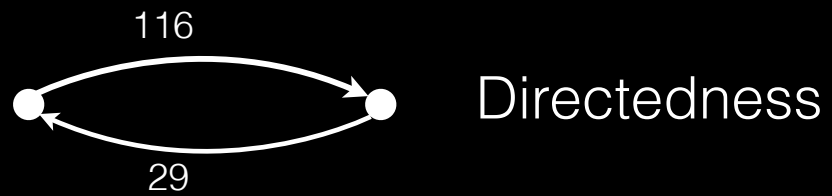




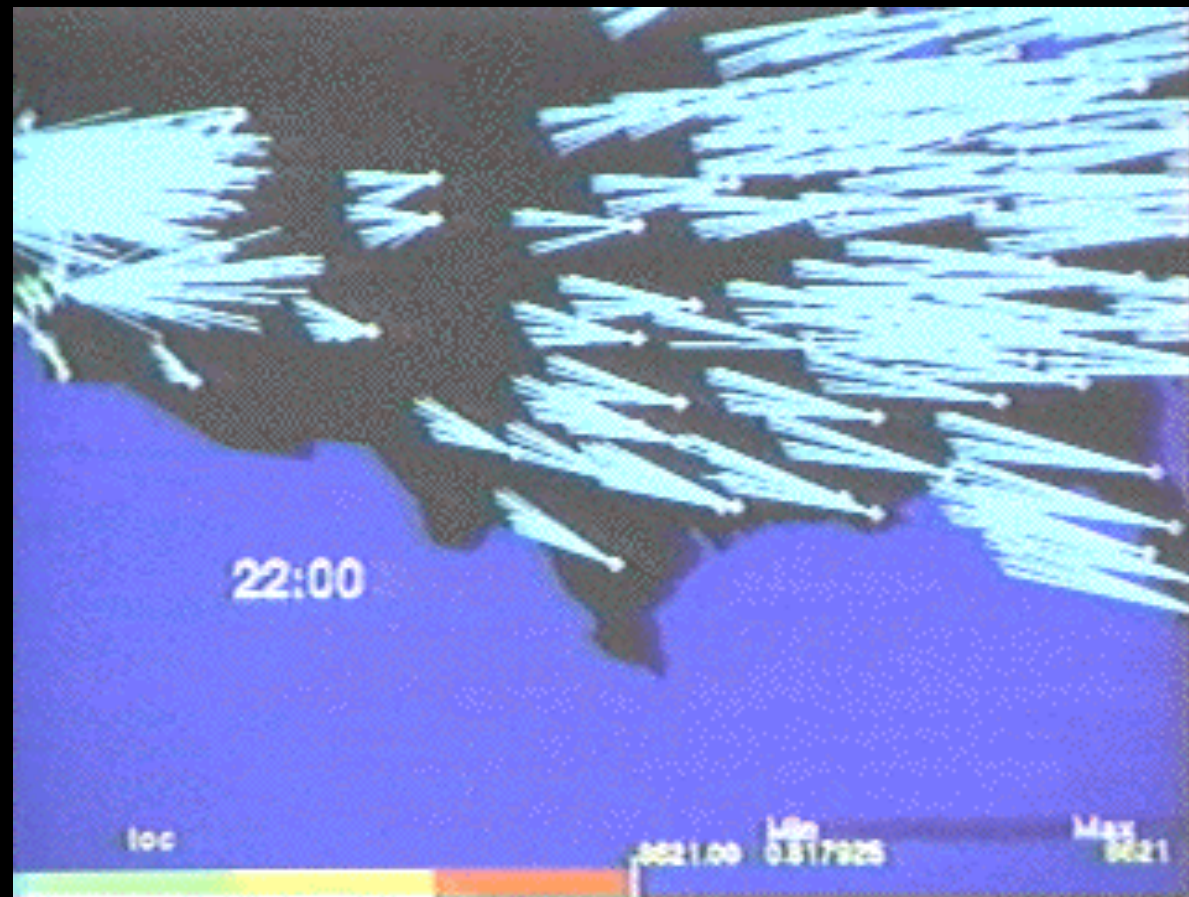






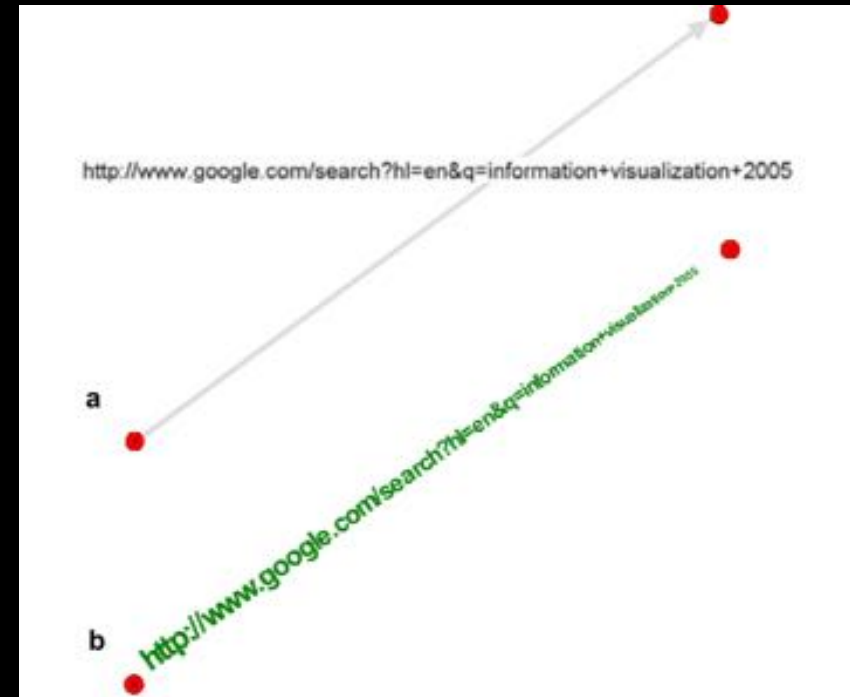






# Graphs with long edge labels

- Where to put the label?
- NP-hard
- Idea: draw label *as* edge



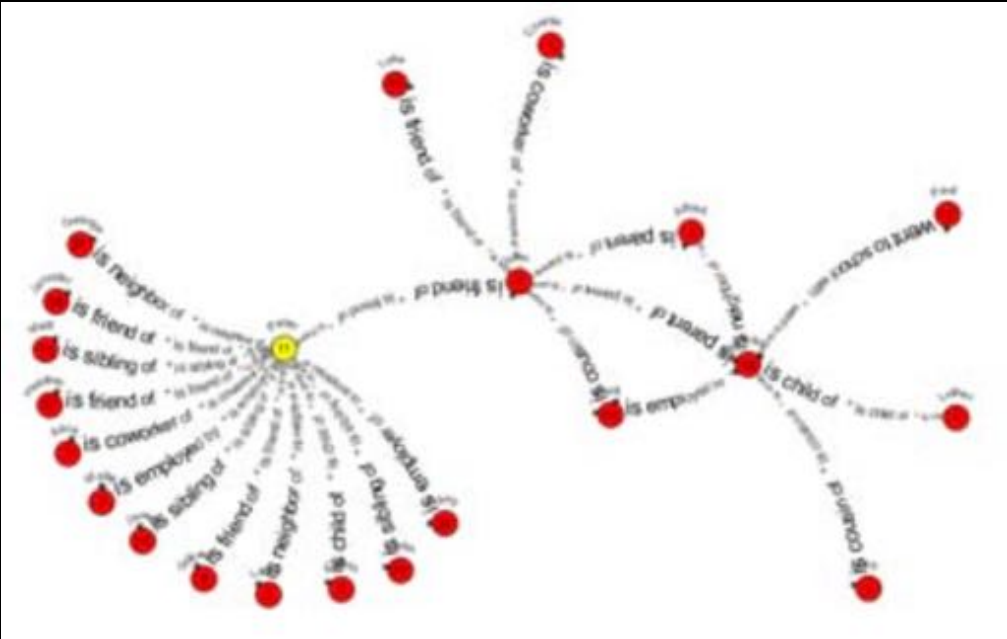
# Multiple scales



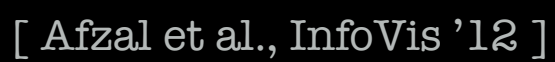
[ Wong, et al., InfoVis '05 ]

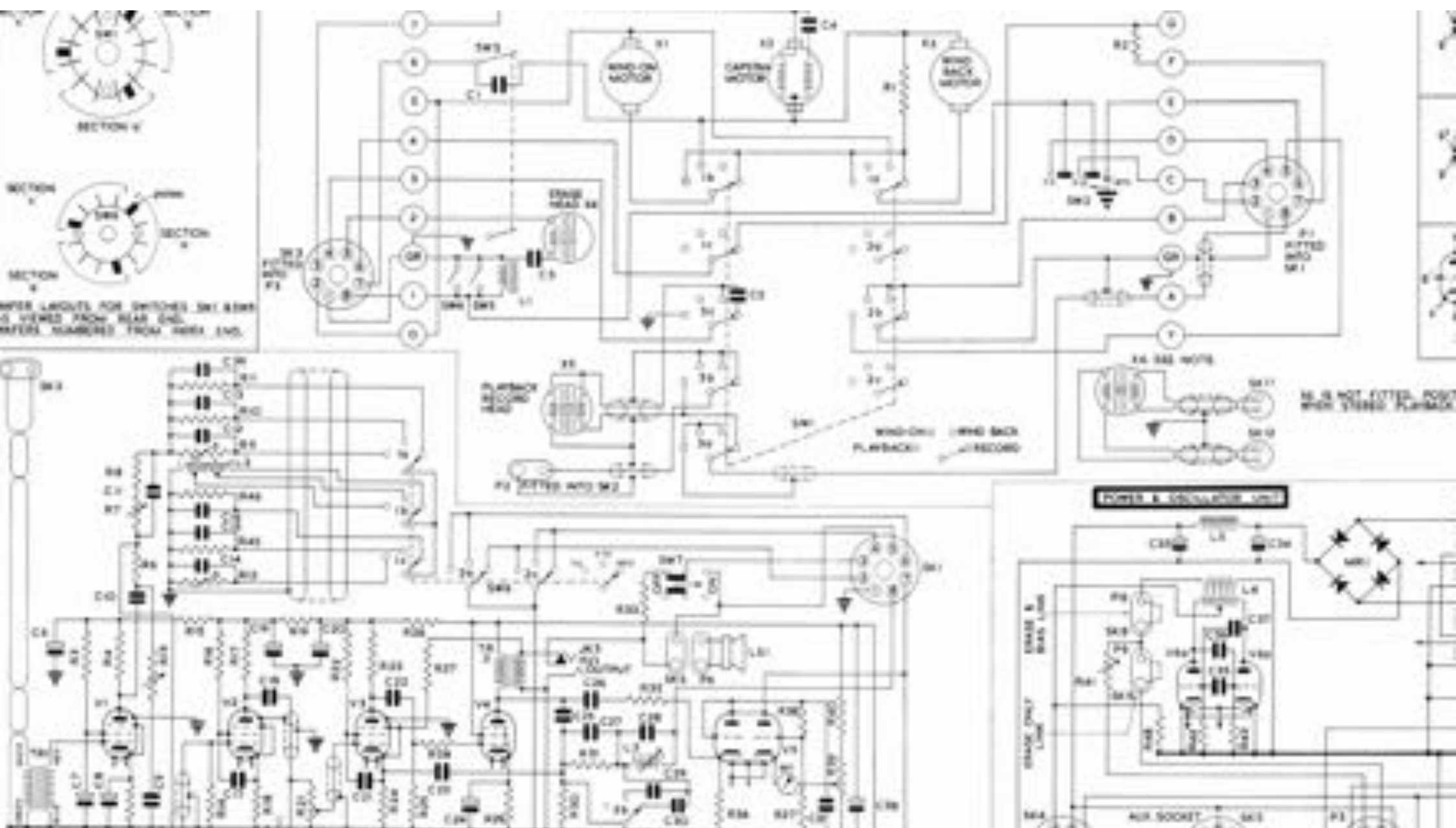
# Directed edges



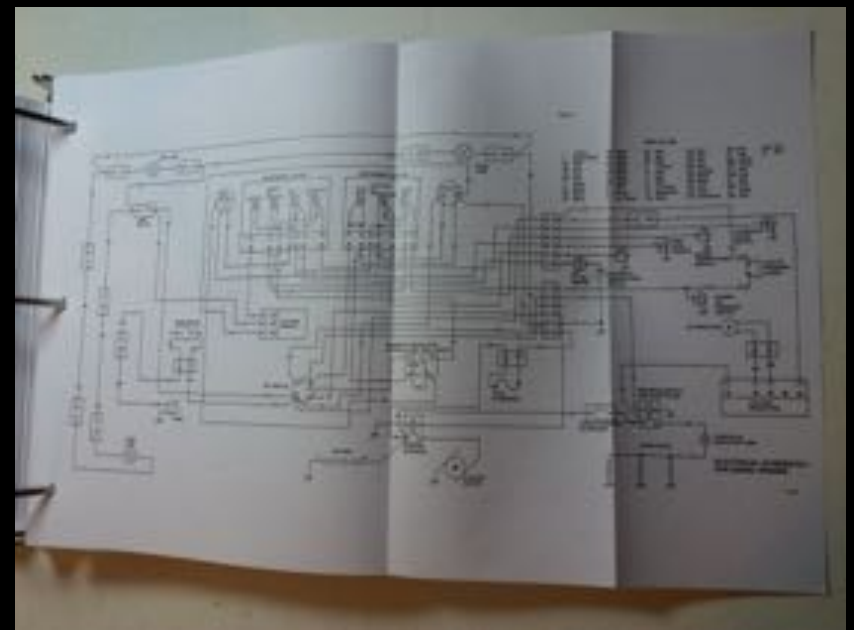
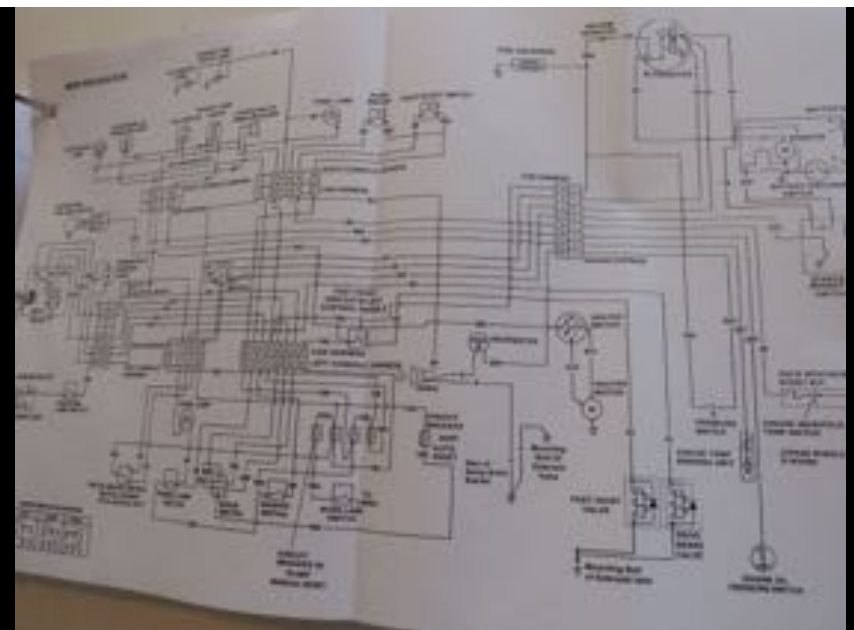






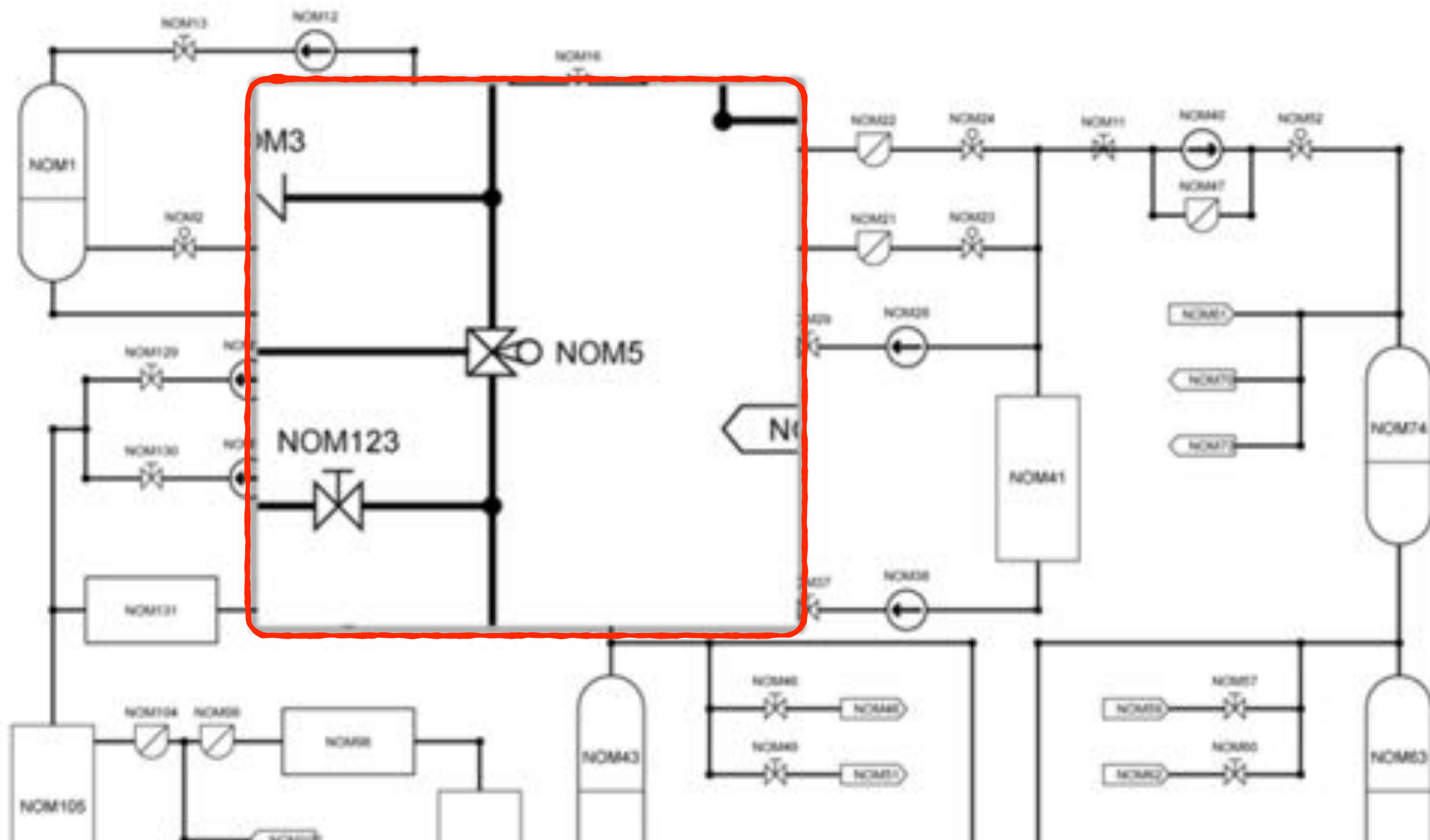




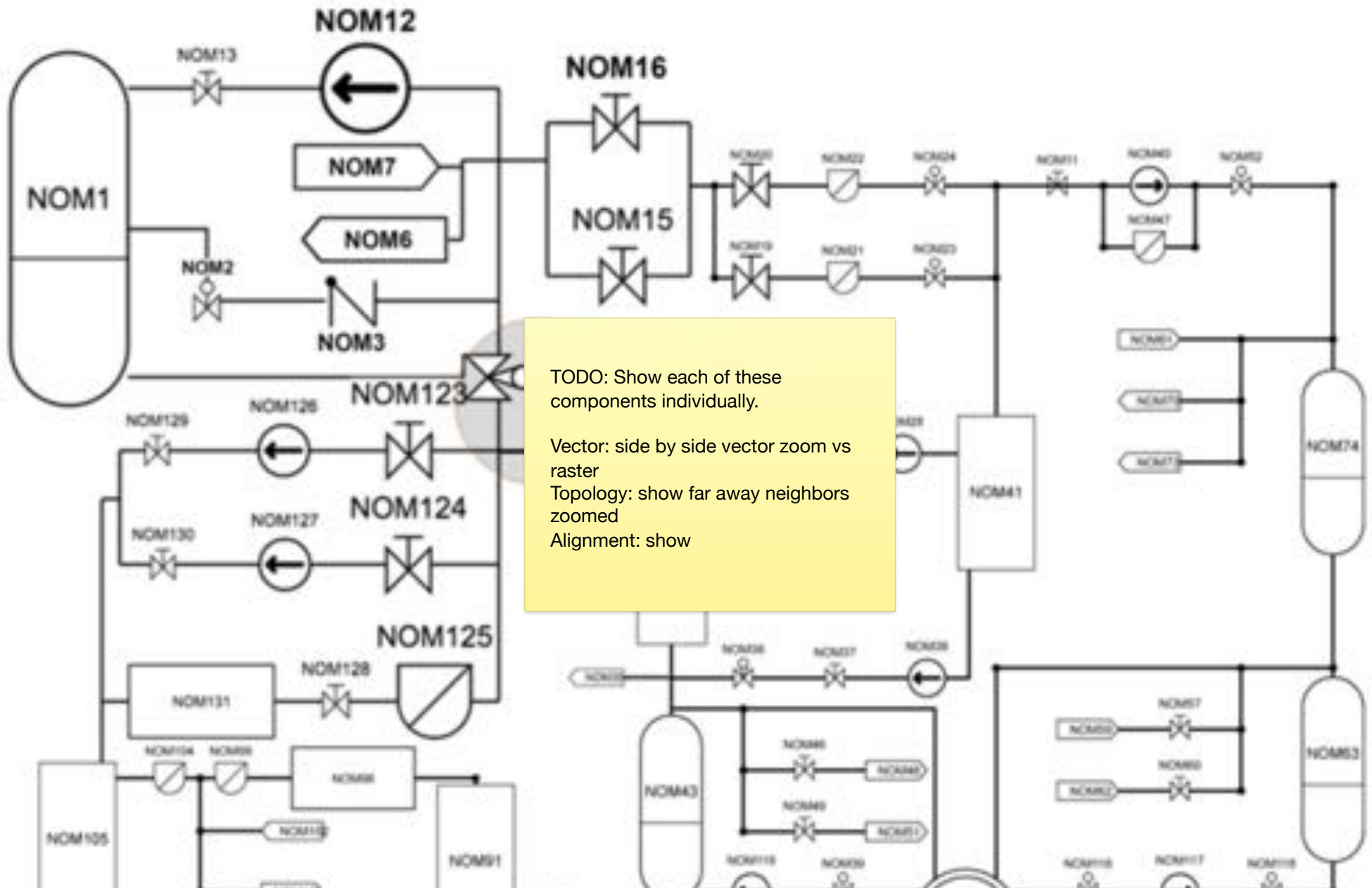




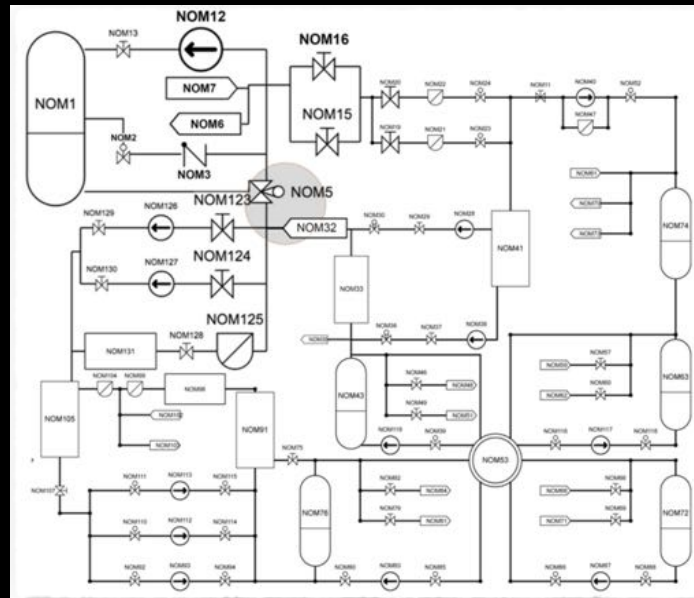
# Focus + Context

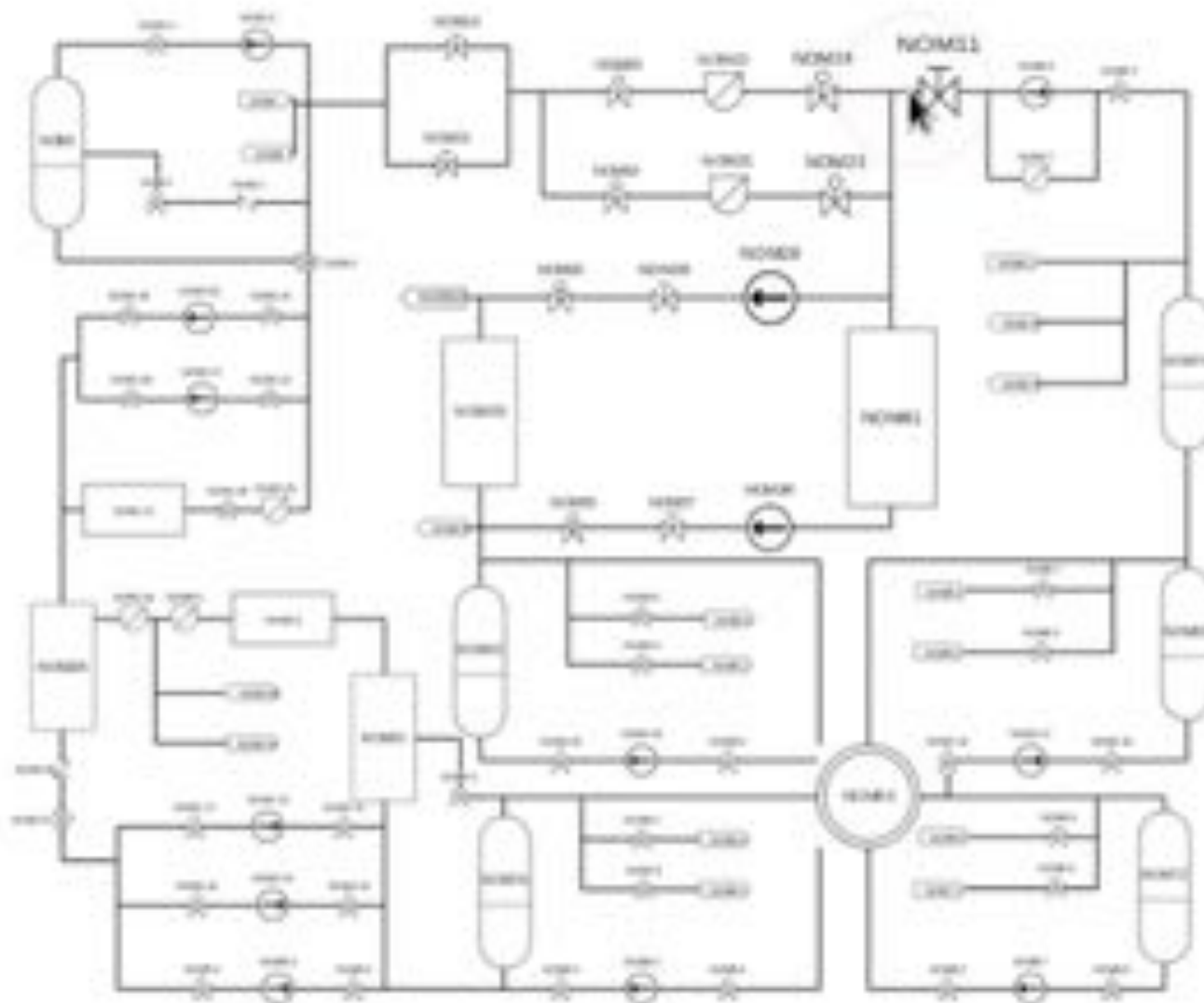






# SchemeLens





**Schemelens**

# What is a graph?

- *vertices and edges*

*a.k.a.*

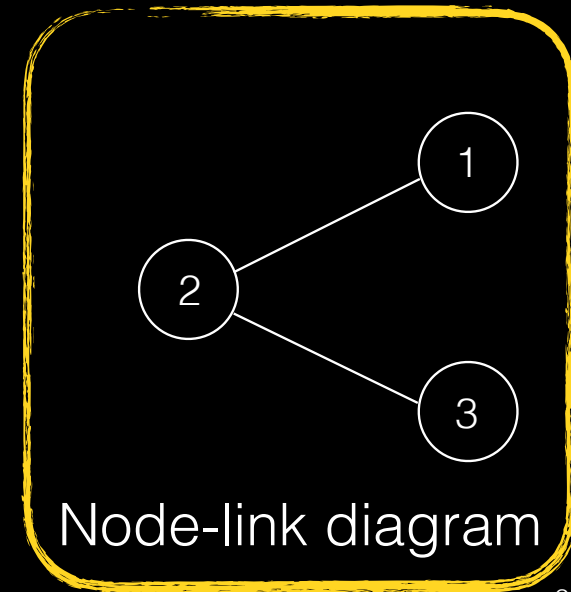
- *nodes and links*

	1	2	3
1	0	1	0
2	1	0	1
3	0	1	0

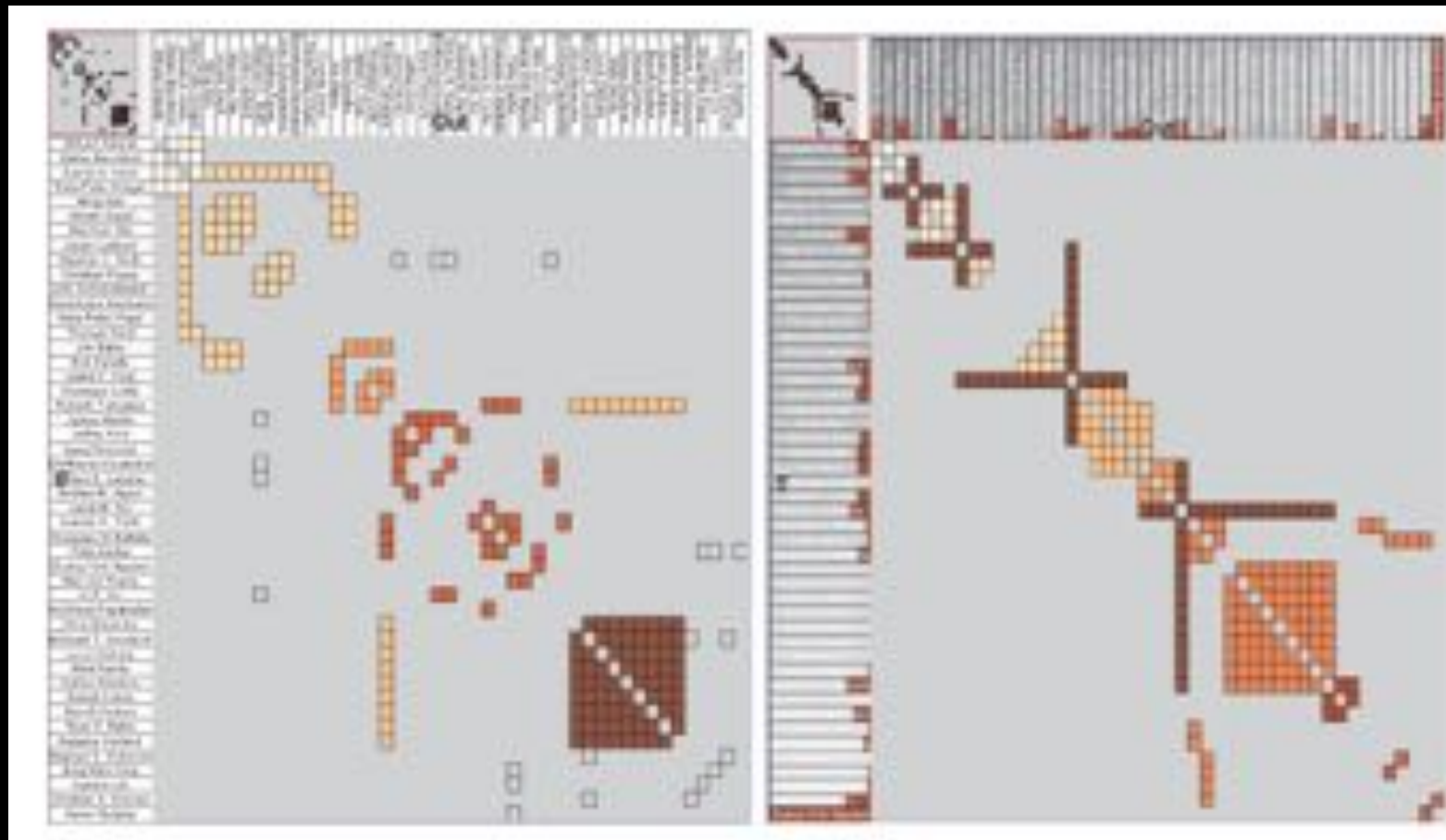
Adjacency matrix

1: 2  
2: 1, 3  
3: 2

Adjacency list

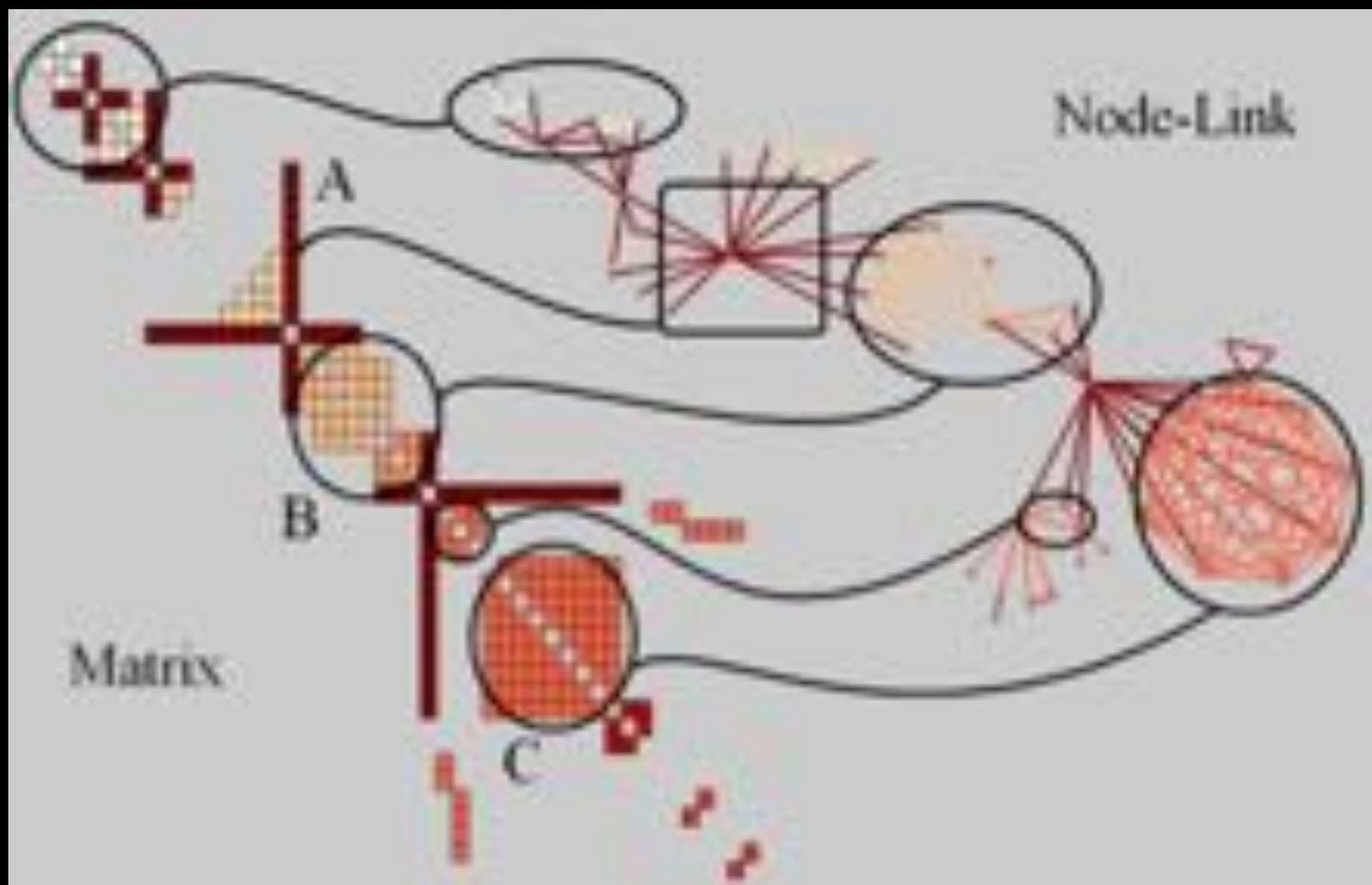


Node-link diagram



[ Henry & Fekete, InfoVis 2006 ]







# More useful resources

- Network visualization: [www.caida.org/projects/internetatlas/viz/](http://www.caida.org/projects/internetatlas/viz/)
- Graph layout: [www.csi.uottawa.ca/ordal/papers/sander/main.html](http://www.csi.uottawa.ca/ordal/papers/sander/main.html)

# Trees

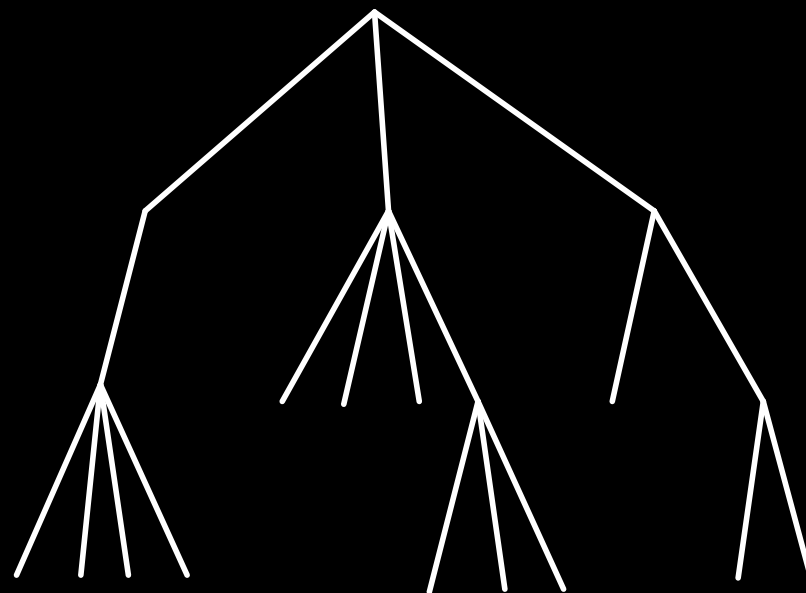
# What is a hierarchy?

- Data repository in which cases are related to subcases.

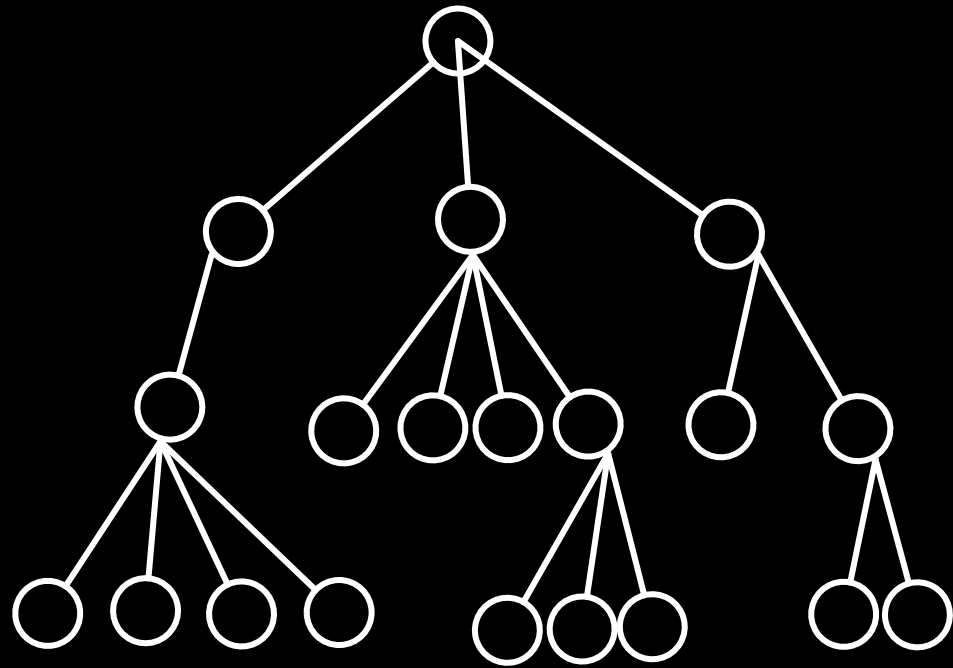
# Where do we see hierarchies in the world?

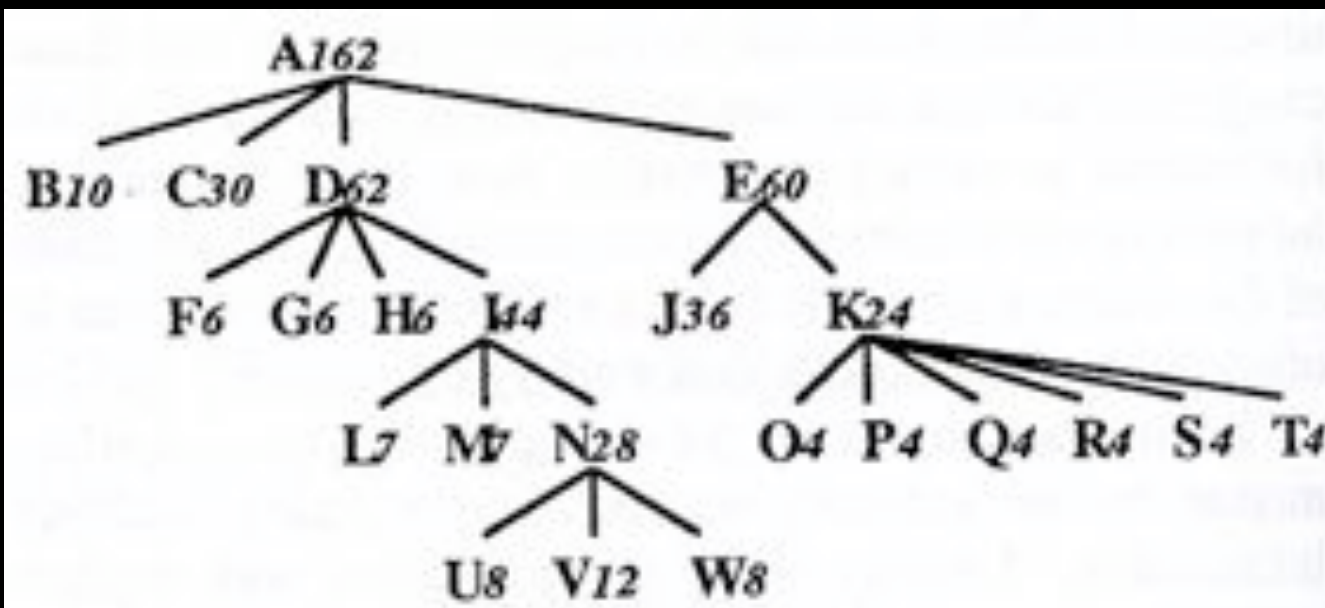
- Family histories
- File systems on computers
- Organization charts
- Animal kingdom (phylum, genus, etc.)
- OO software class structures
- ...



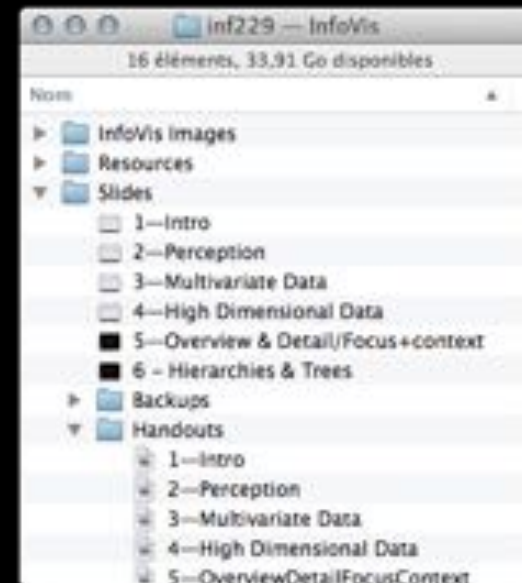
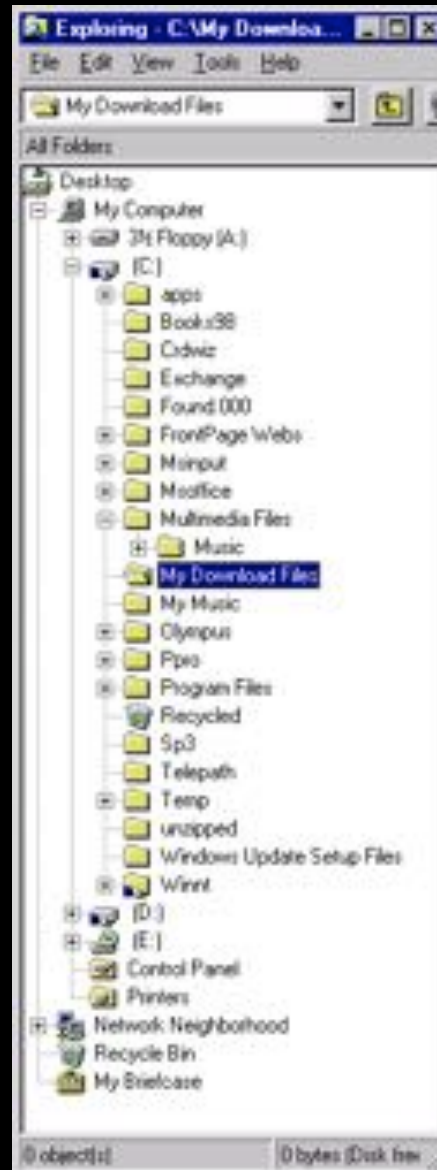


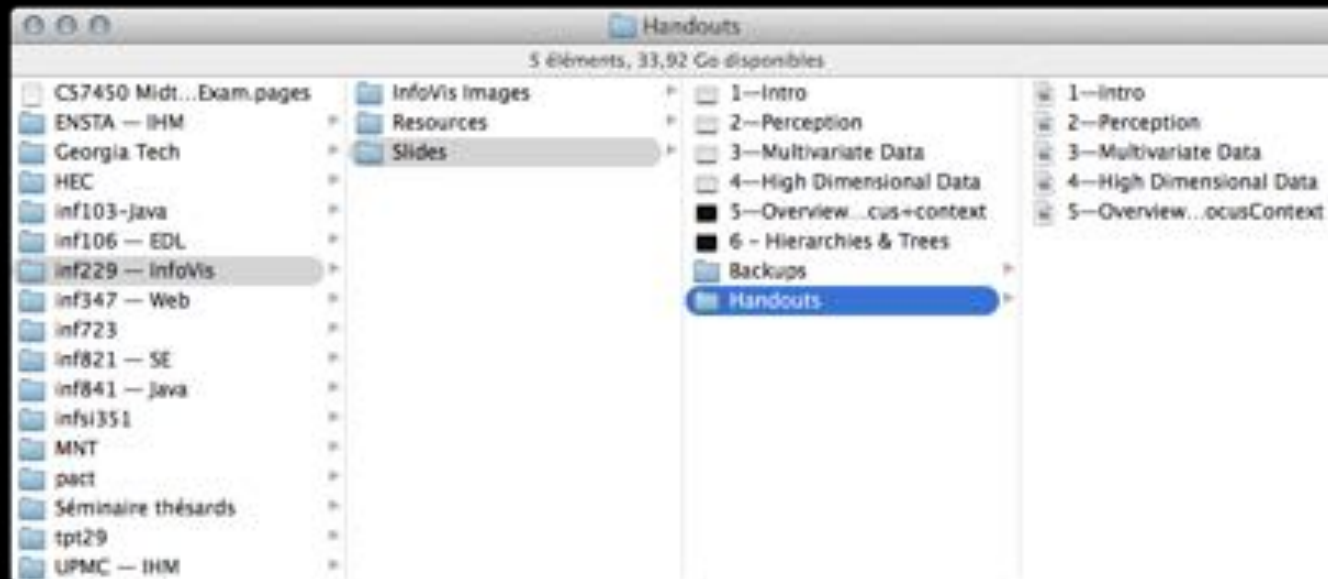


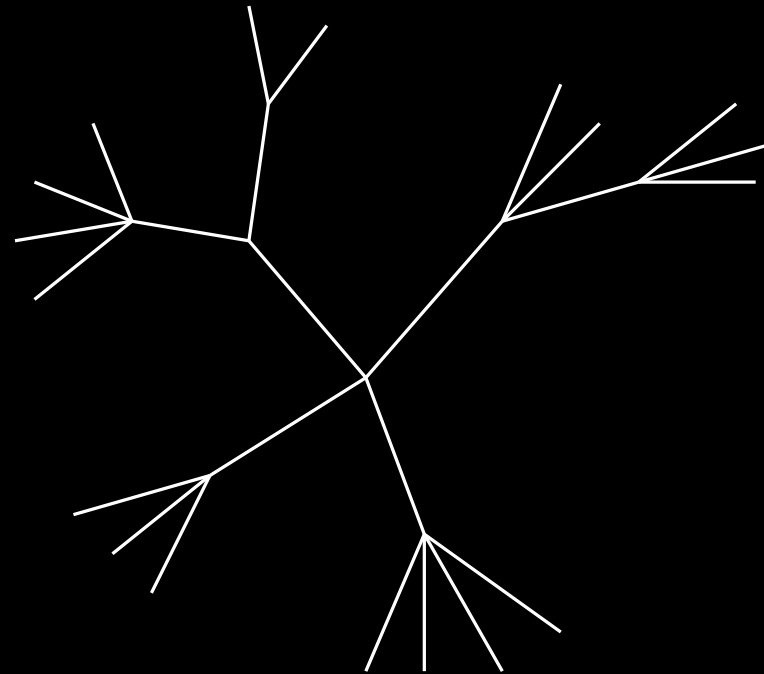


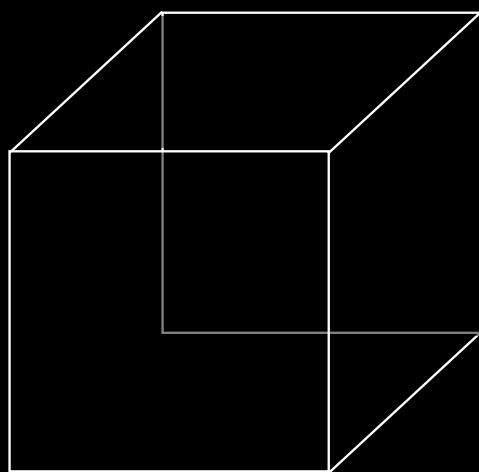


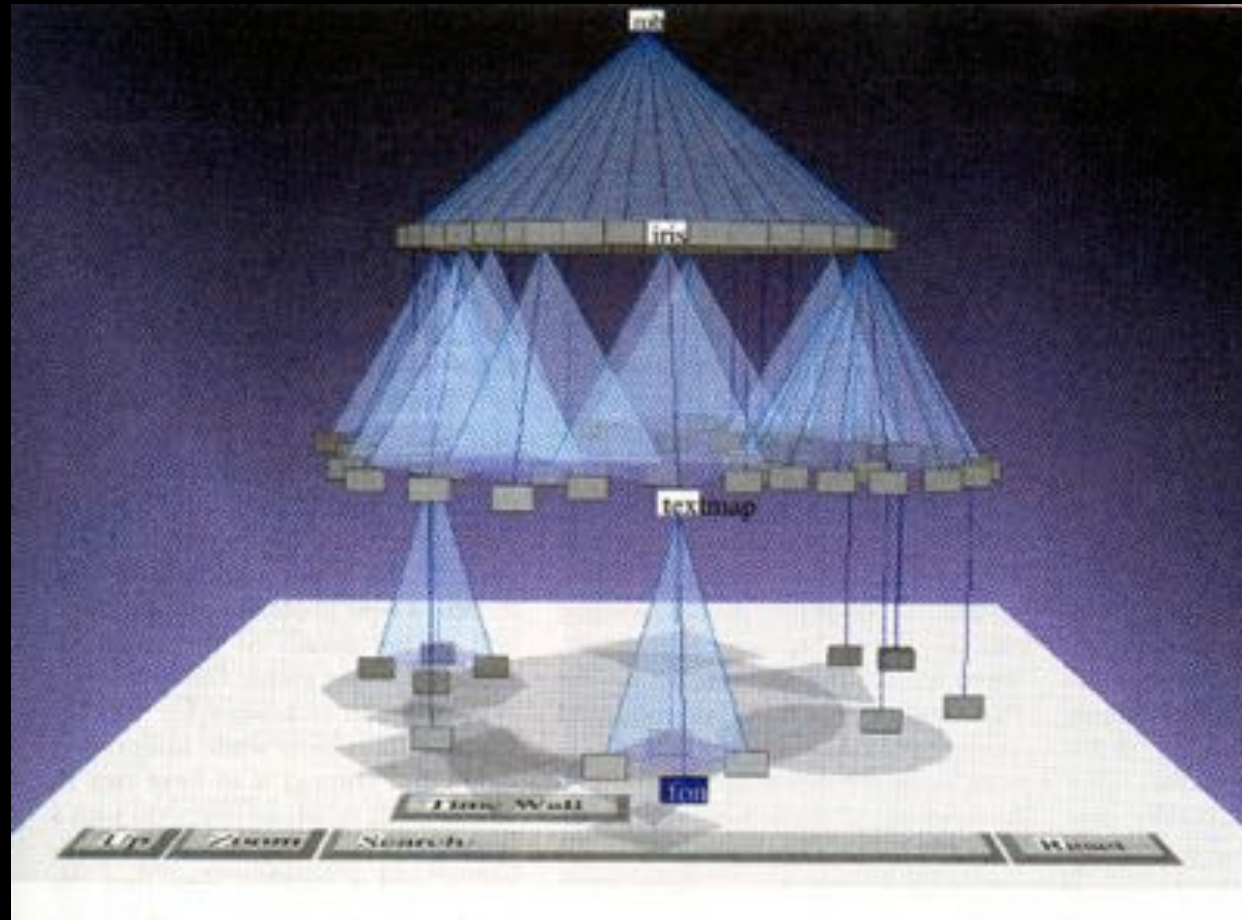
[ Johnson & Shneiderman, 1991 ]





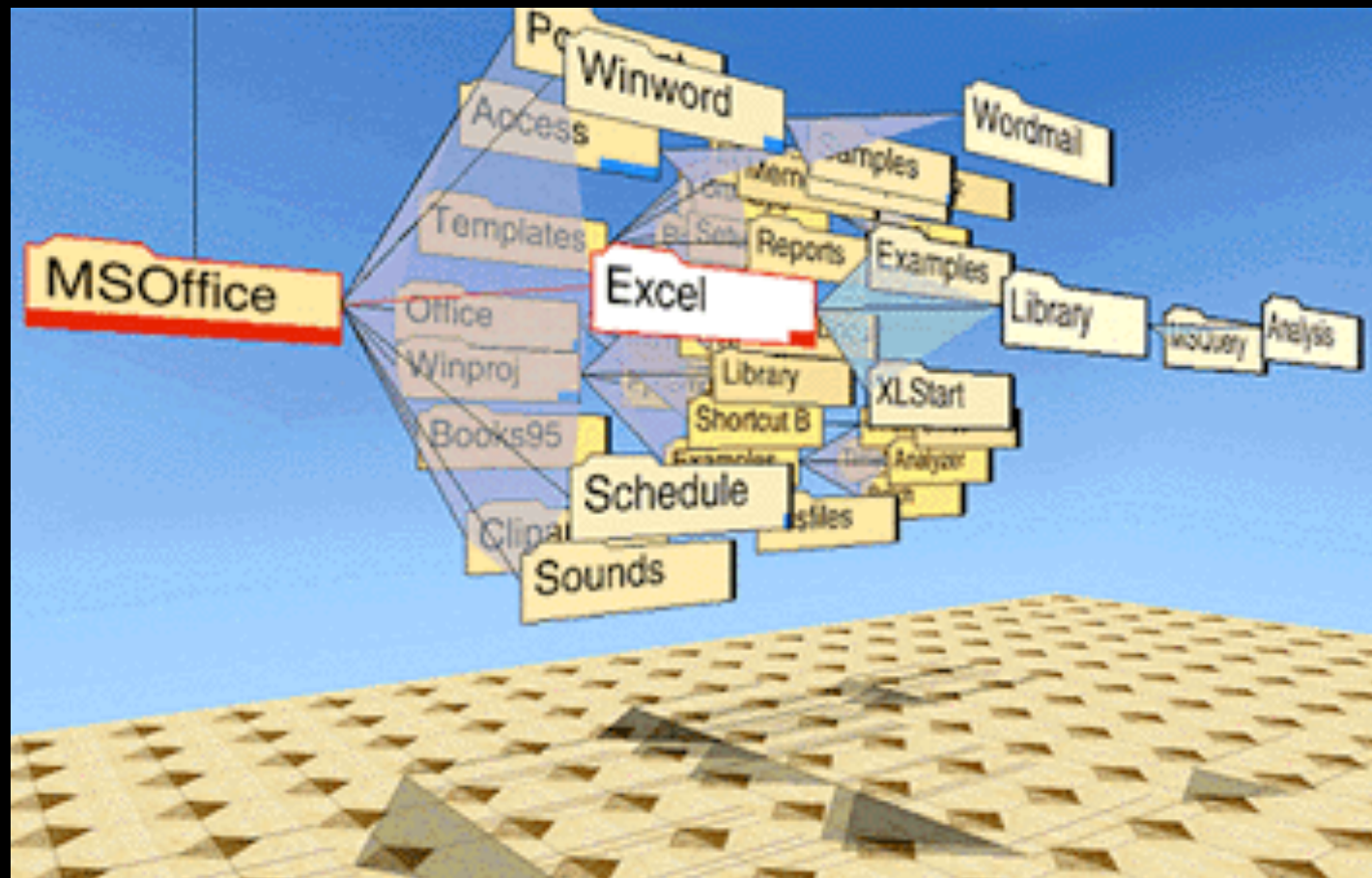






[ Robertson, Mackinlay & Card, CHI 1991 ]

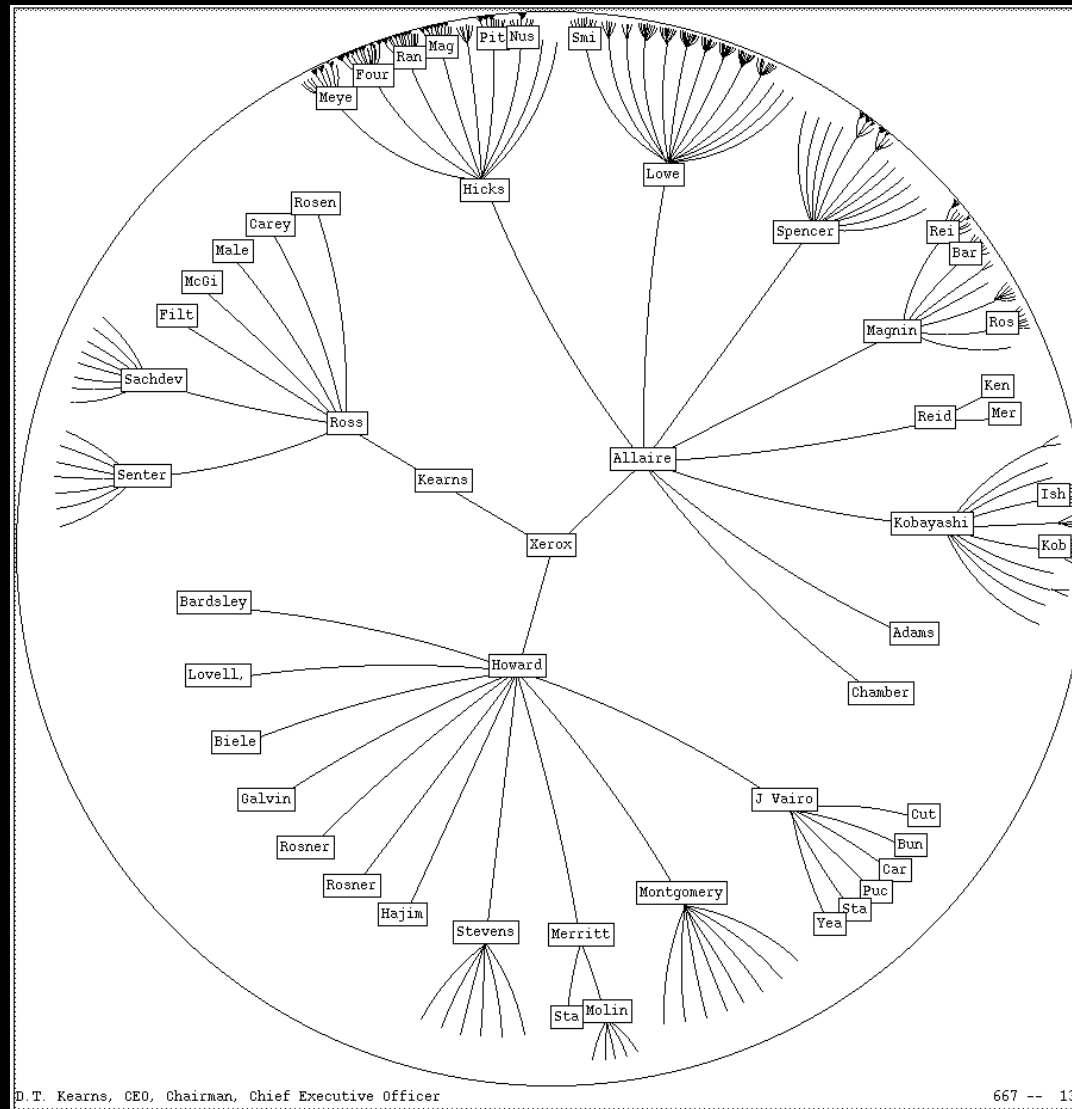




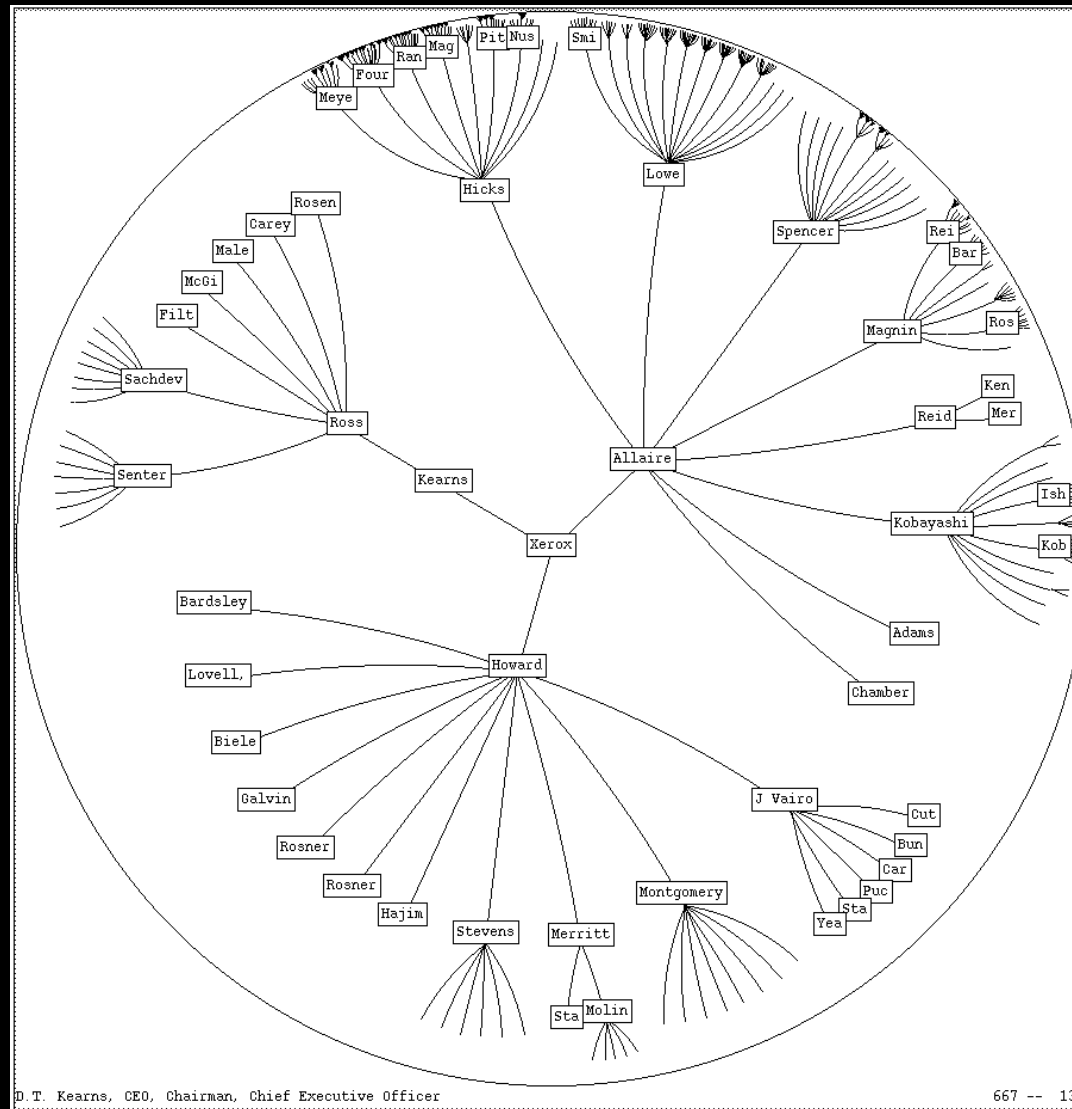




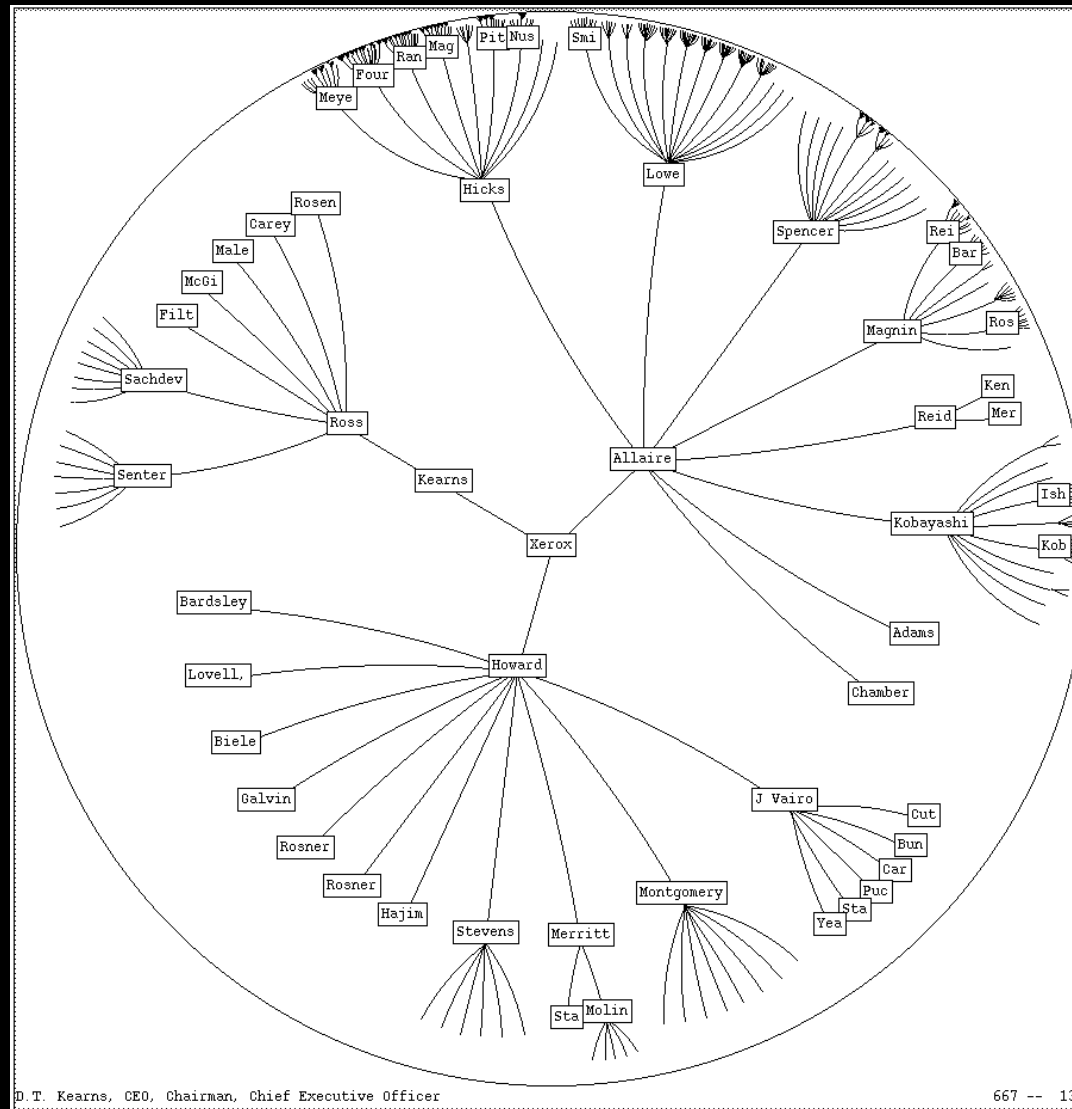
- Positive
  - More effective area to lay out tree
  - Use of smooth animation to help person track updates
  - Aesthetically pleasing
- Negative
  - As in all 3D, occlusion obscures some nodes
  - Non-trivial to implement and requires some graphics horsepower



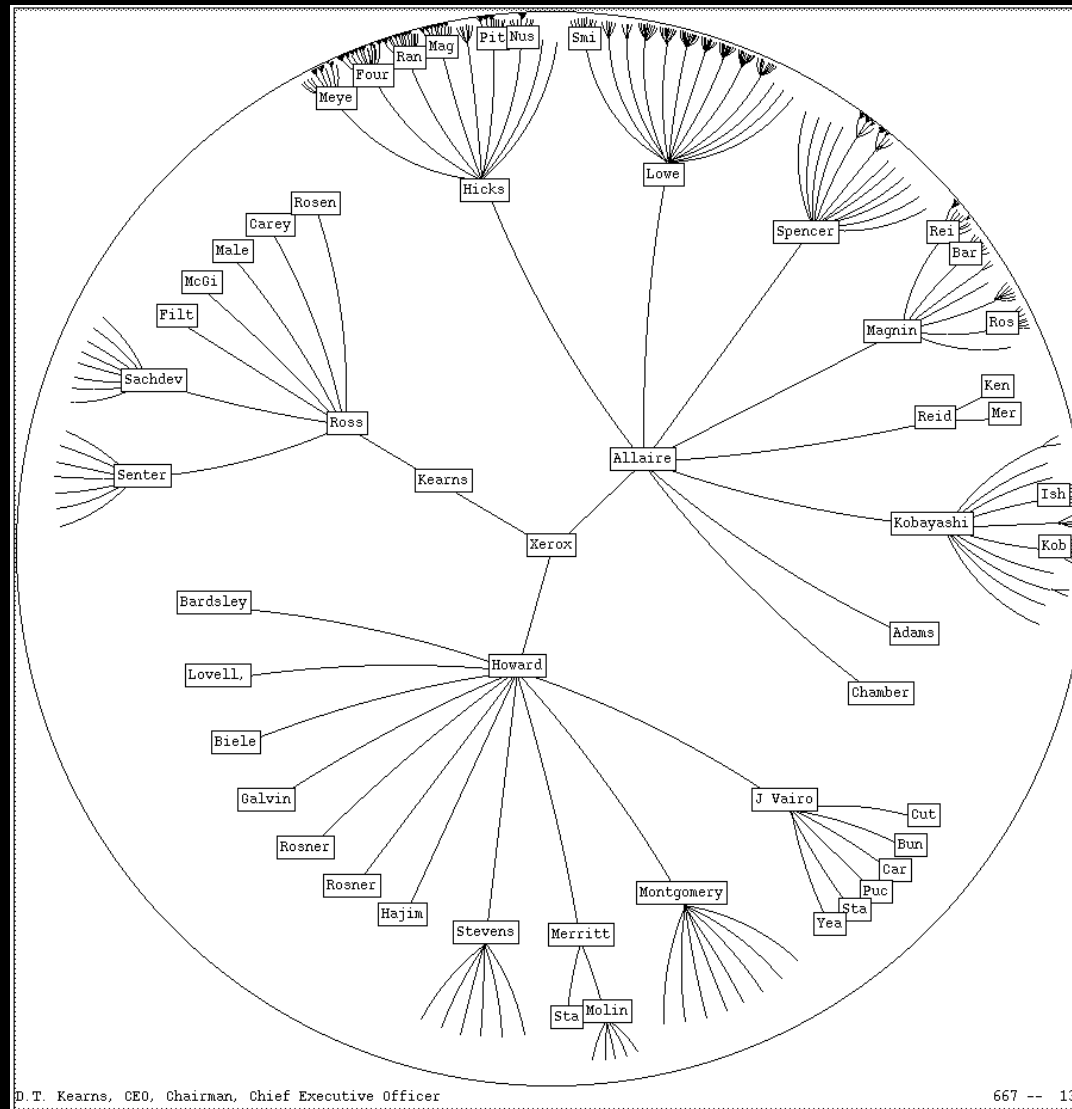
[ Lamping & Rao, JVLC 1996 ]



[ Lamping & Rao, JVLC 1996 ]



[ Lamping & Rao, JVLC 1996 ]



[ Lamping & Rao, JVLC 1996 ]



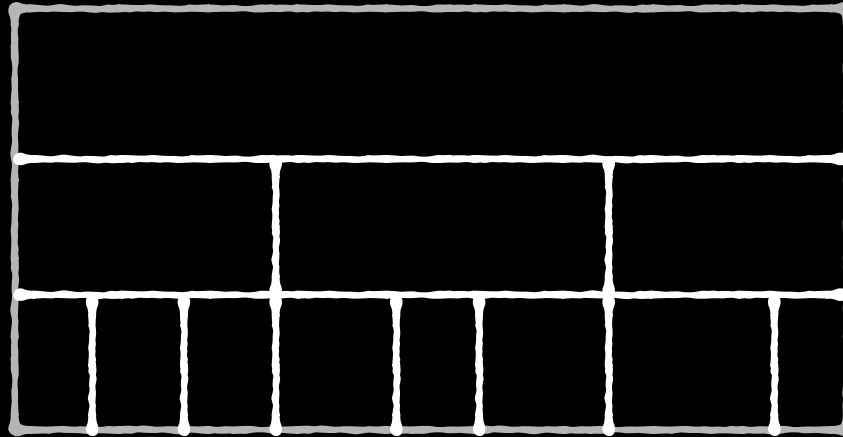
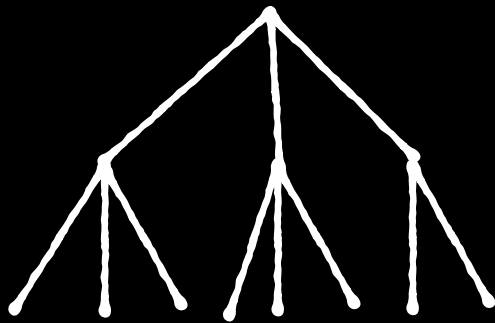
# Encoding more variables in node-link is difficult

- Shape
- Color
- Size



# Let's use all the space

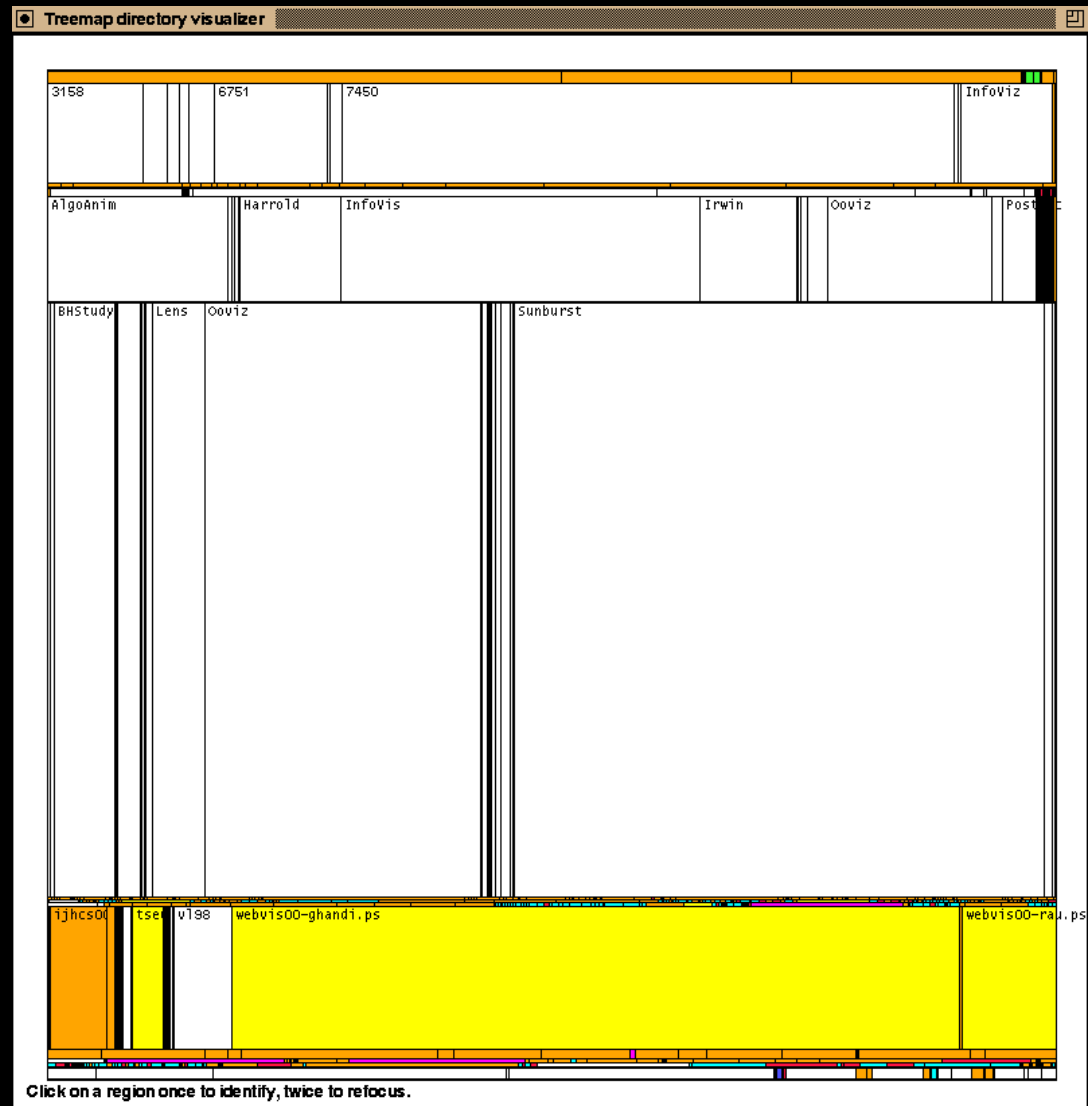


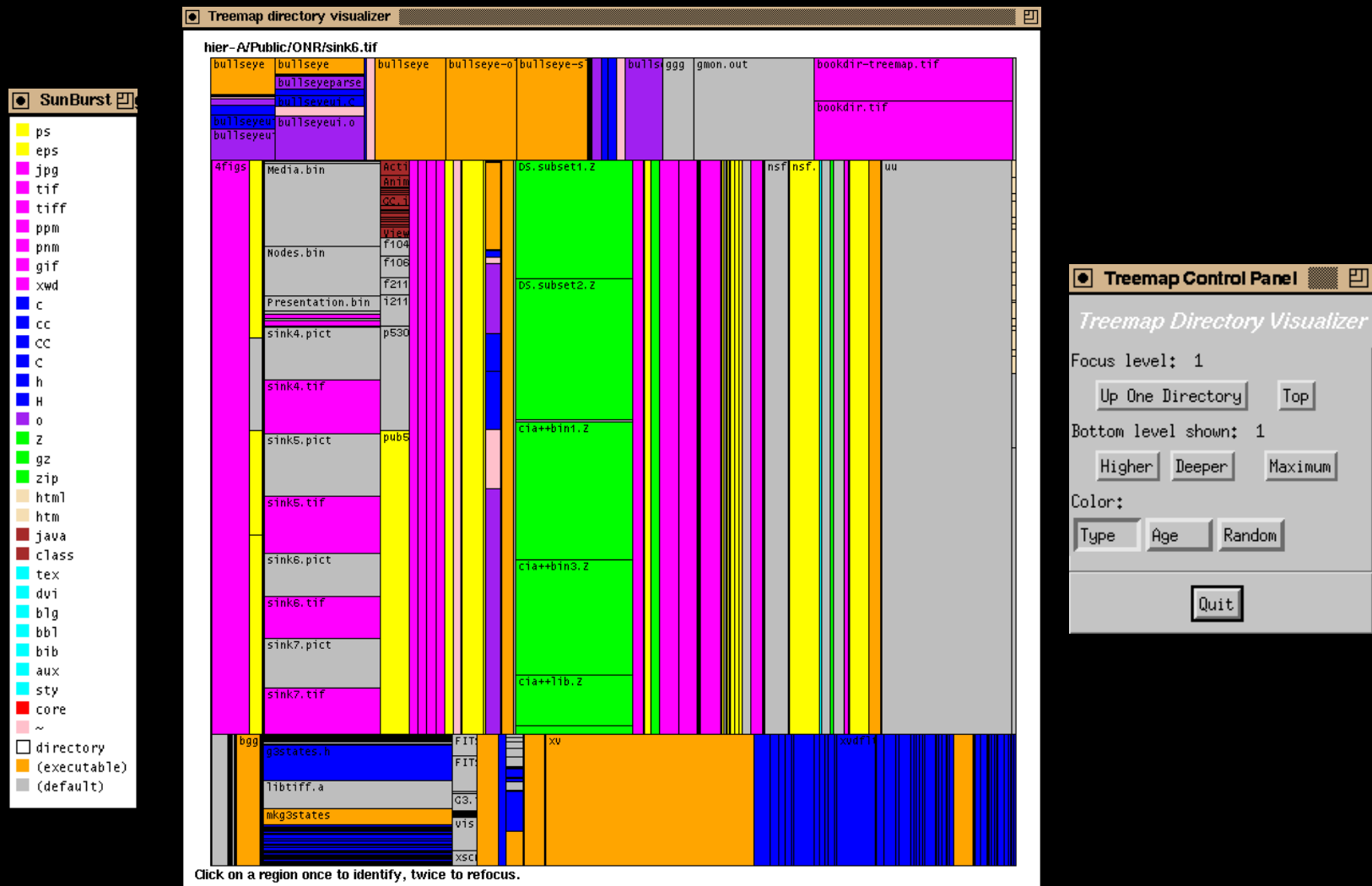


- Each item occupies an area
- Children contained under parent

# Treemap

- Space-filling approach
- Slice-and-dice
- Use area to encode a variable





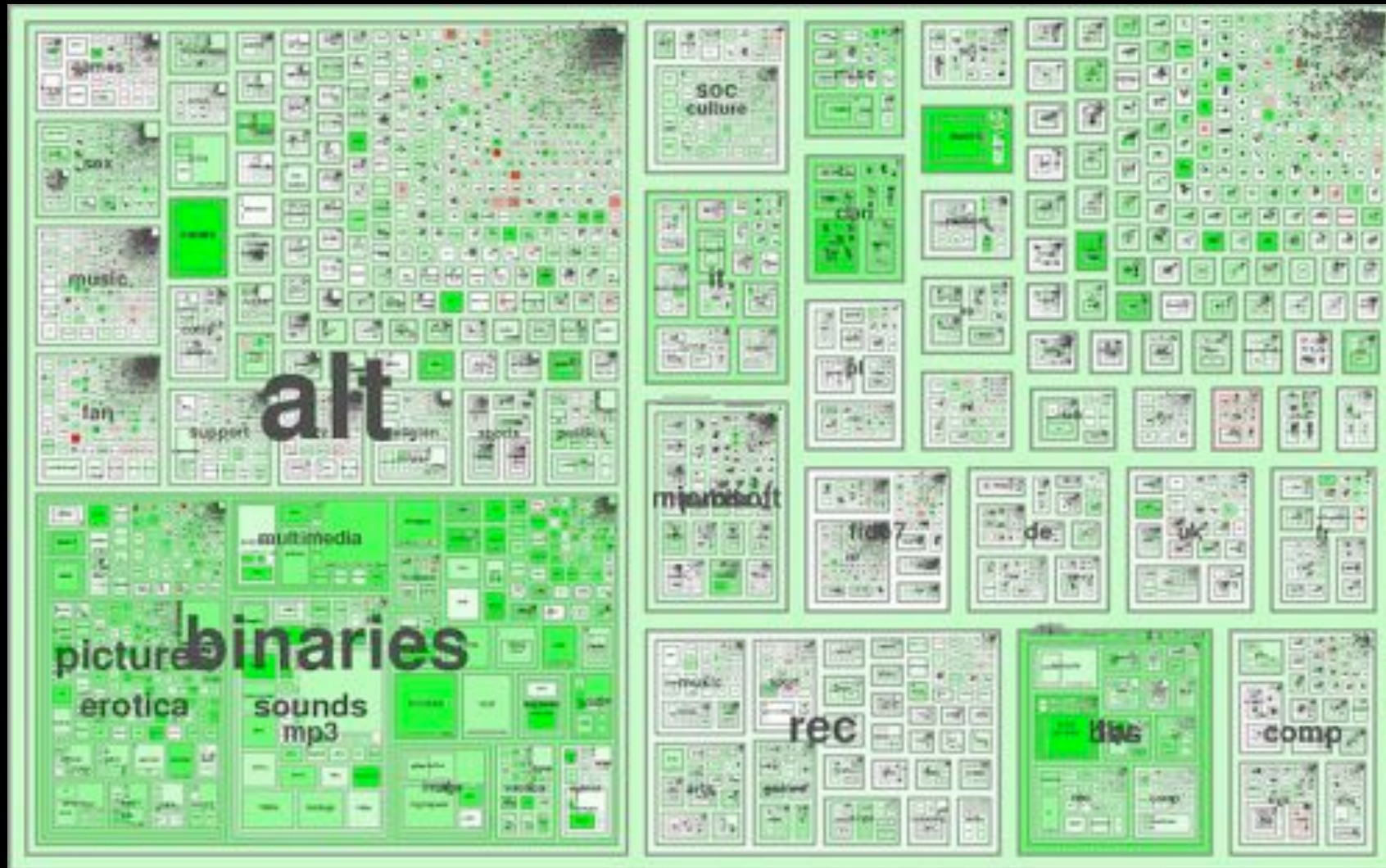
```
Draw()  
{  
    Change orientation from parent (horizontal/vertical)  
    Read all nodes at this level  
    Make rectangle for each, scaled to size  
    Draw rectangles using appropriate size and color  
    For each child  
        Make recursive call using its rectangle as focus  
}
```





96





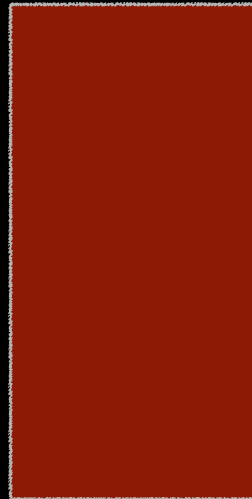
[ Fiore & Smith, Microsoft Tech Report '01 ]

# What's good about Treemaps ?

- Focuses on nodes and attributes
- Color
- Area

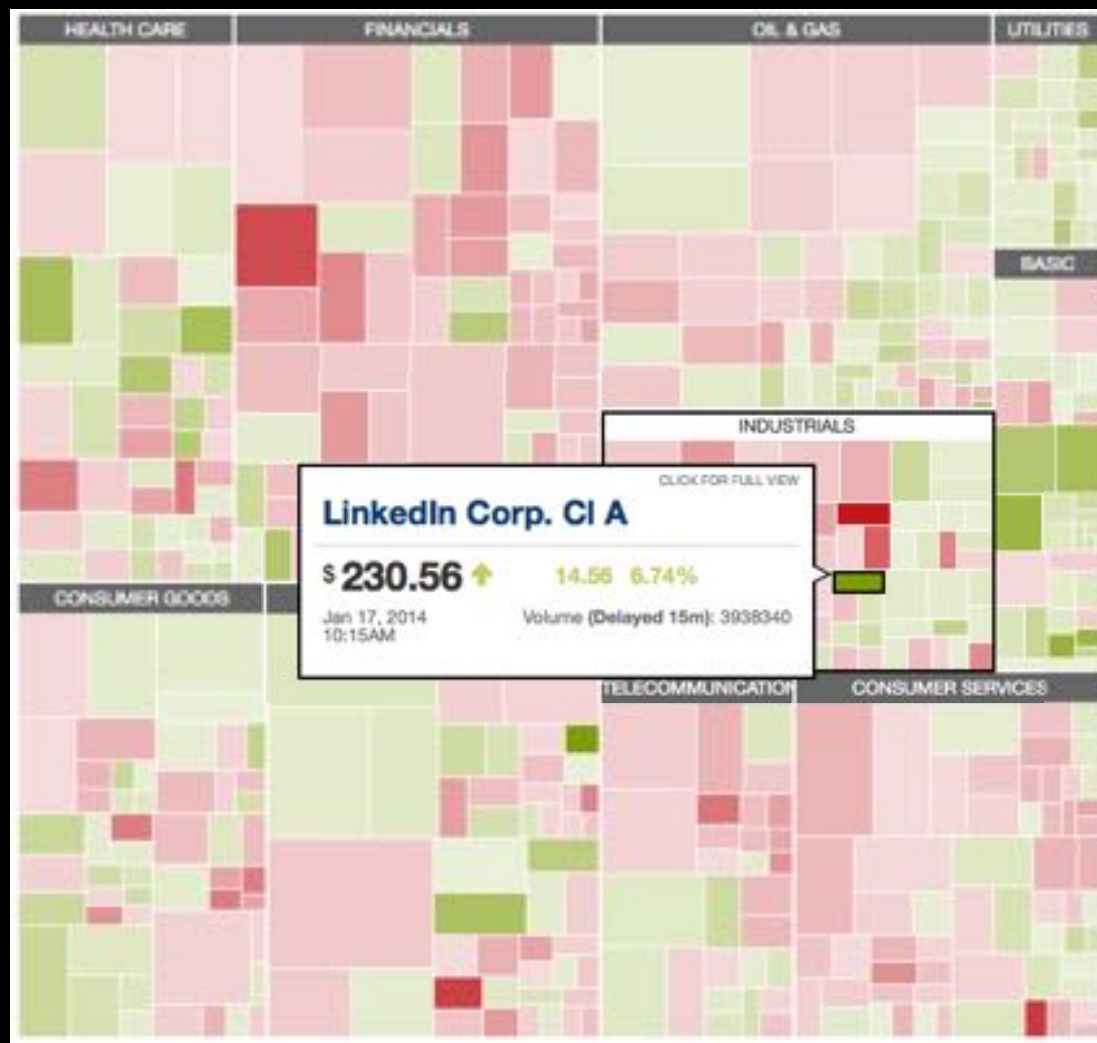
# What's not so good ?

These kinds of rectangles are visually unappealing



Which has the bigger area?

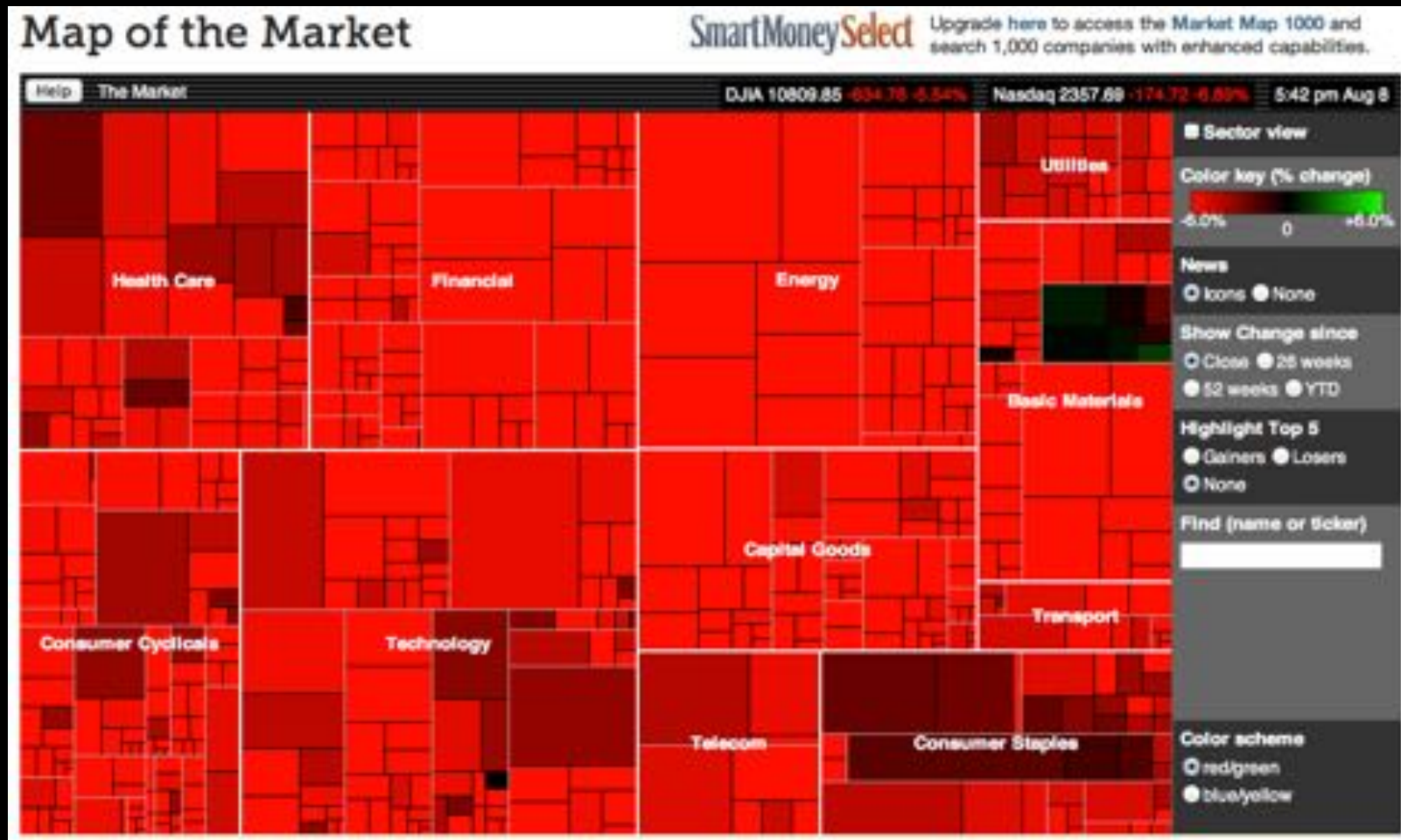
**Can we make rectangles more square?**



[ [www.marketwatch.com/tools/stockresearch/marketmap](http://www.marketwatch.com/tools/stockresearch/marketmap), January 17, 2014 ]



2011-08-08



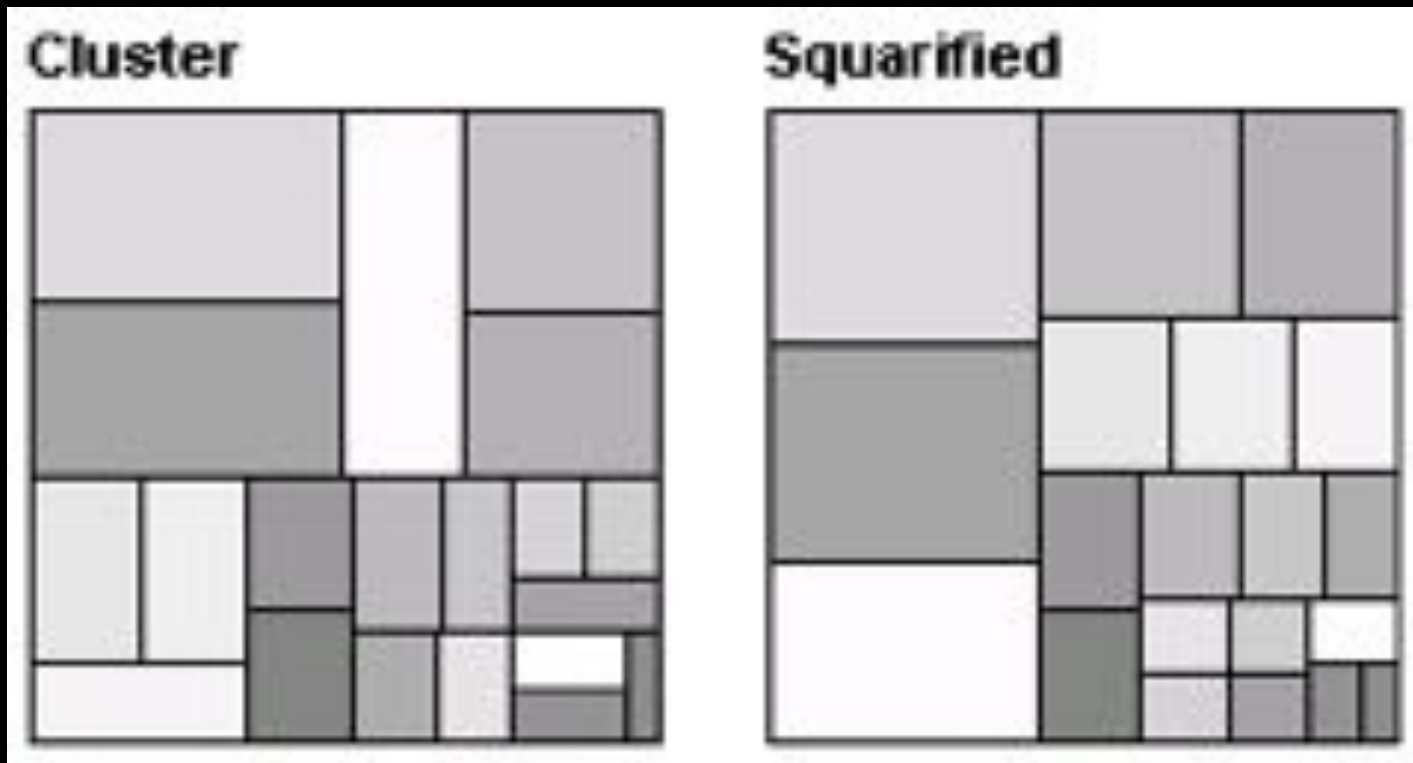
[ Wattenberg, CHI '99 ]

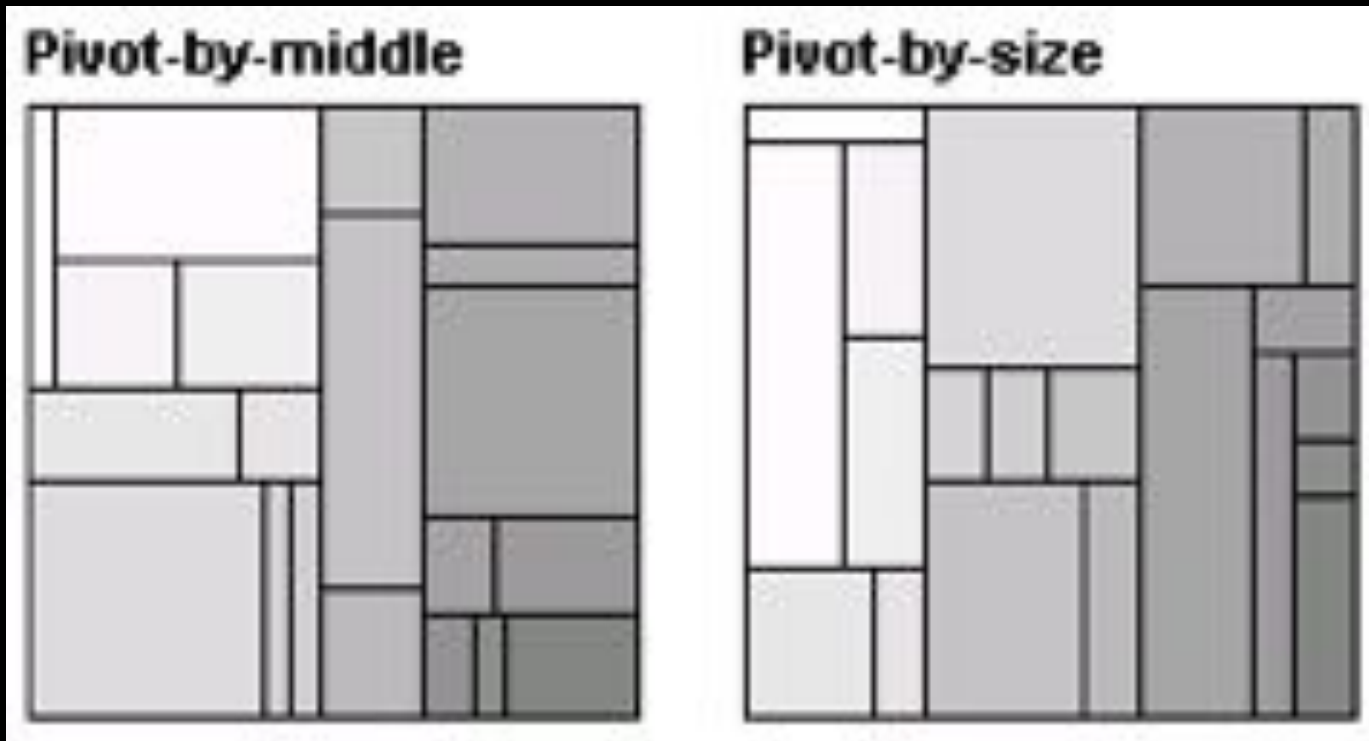
103

# What's good about the map of the market?

- Macro-micro readings
- Interaction
- Real-world application of InfoVis





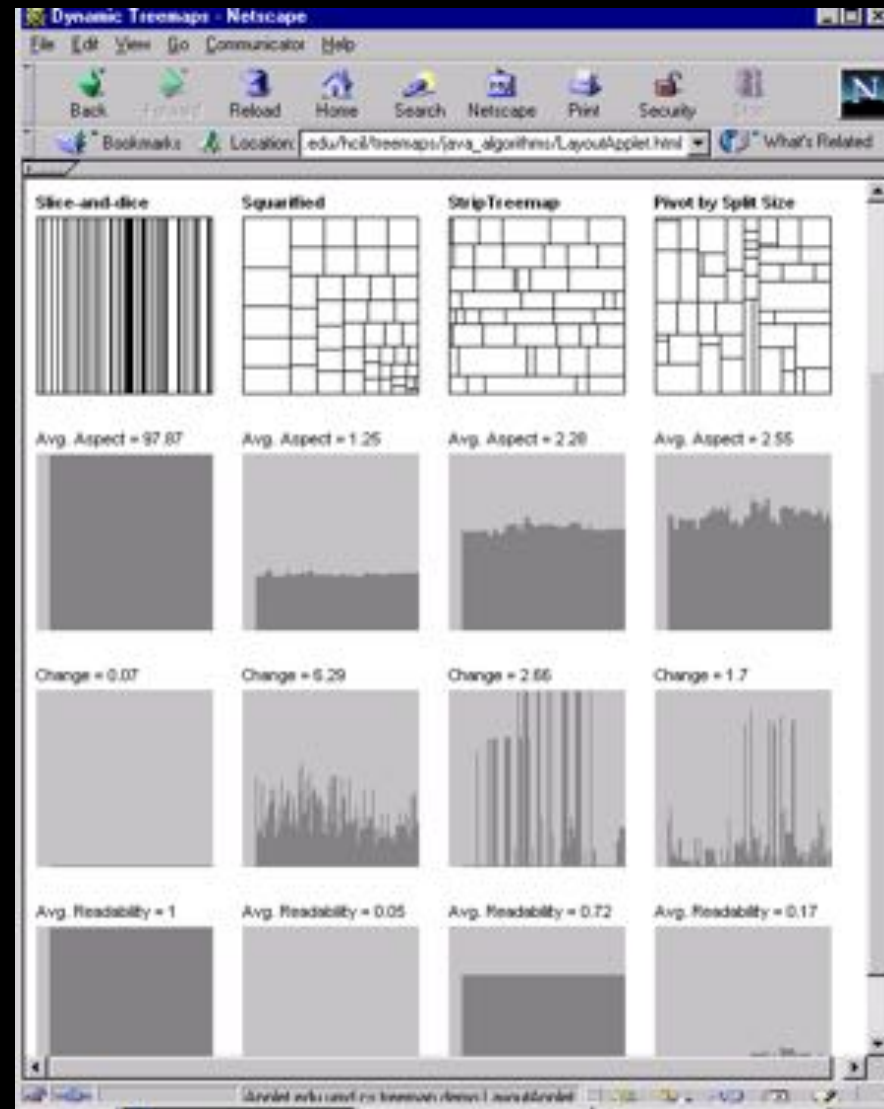


## StripTreemap

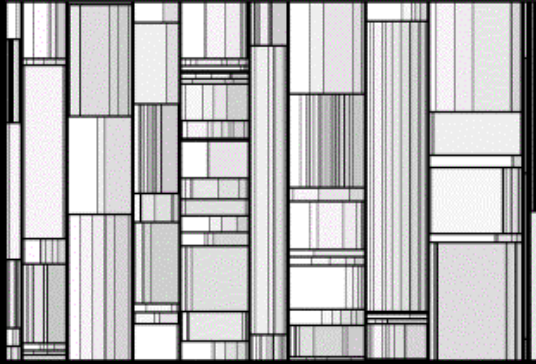
[ Bederson, Shneiderman &amp; Wattenberg, ToG '02 ]

Compare algorithms by

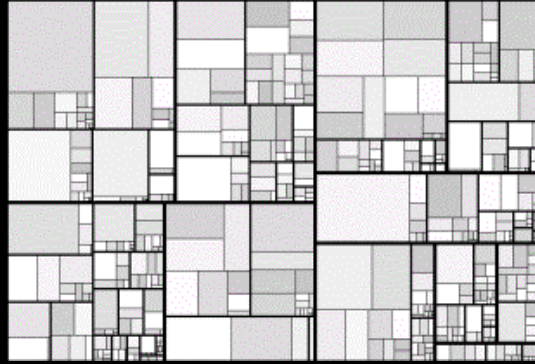
- \* aspect ratio
- \* structural stability
- \* readability



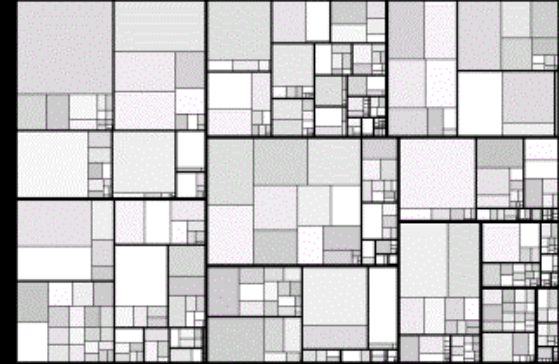
[ [www.cs.umd.edu/hcil/treemap-history/java\\_algorithms/LayoutApplet.html](http://www.cs.umd.edu/hcil/treemap-history/java_algorithms/LayoutApplet.html) ]



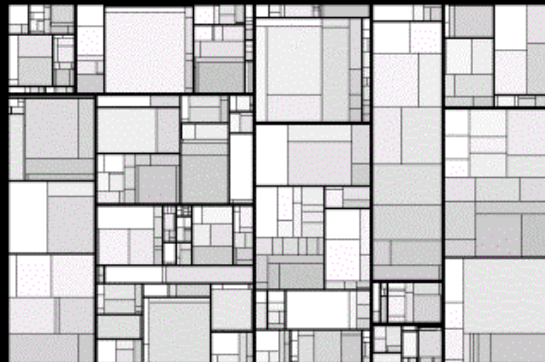
Slice and dice



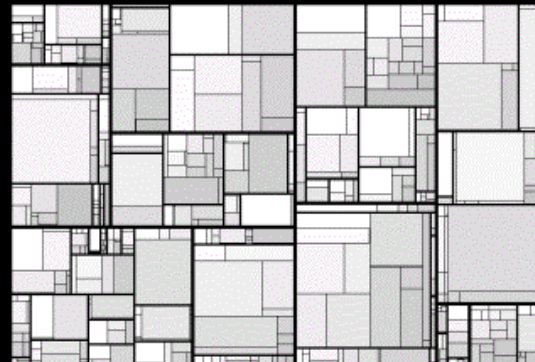
Cluster



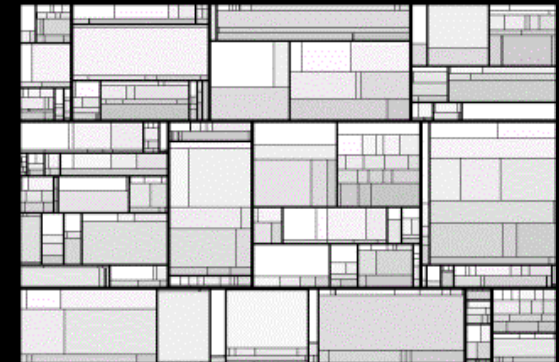
Squarified



Pivot-by-middle



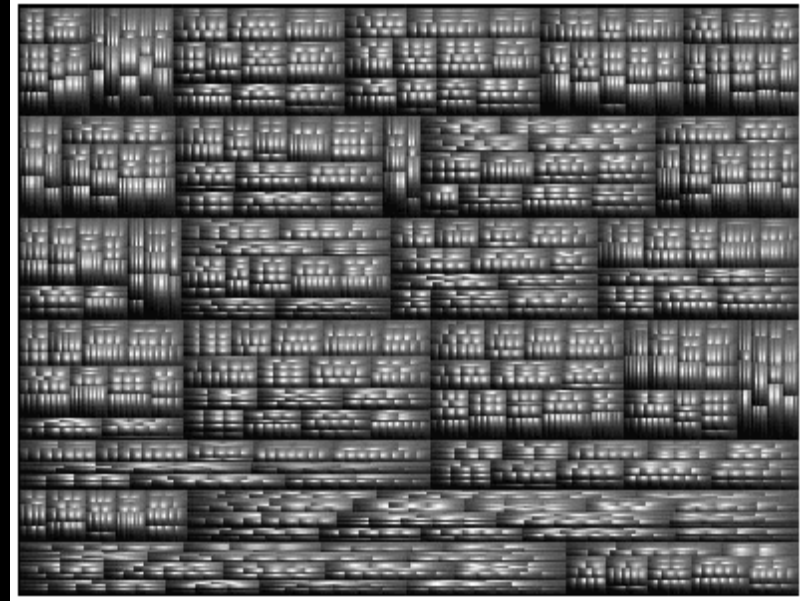
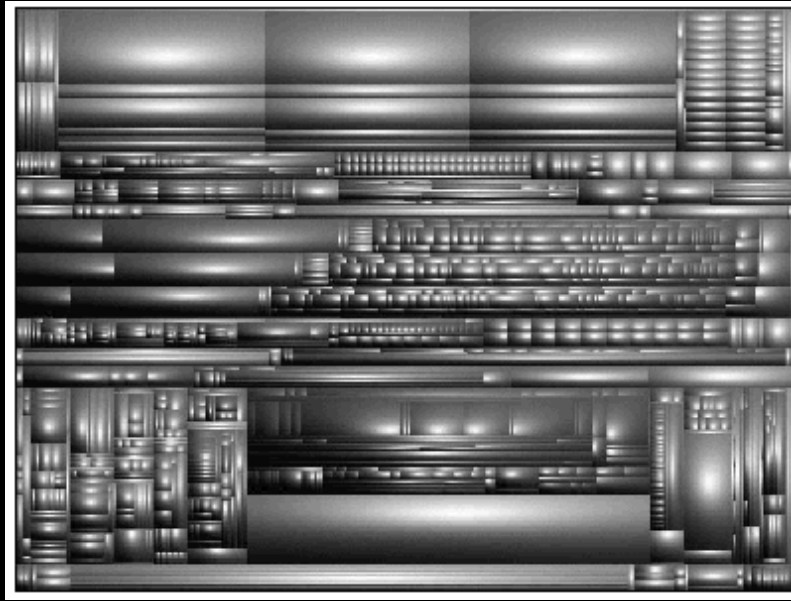
Pivot-by-size



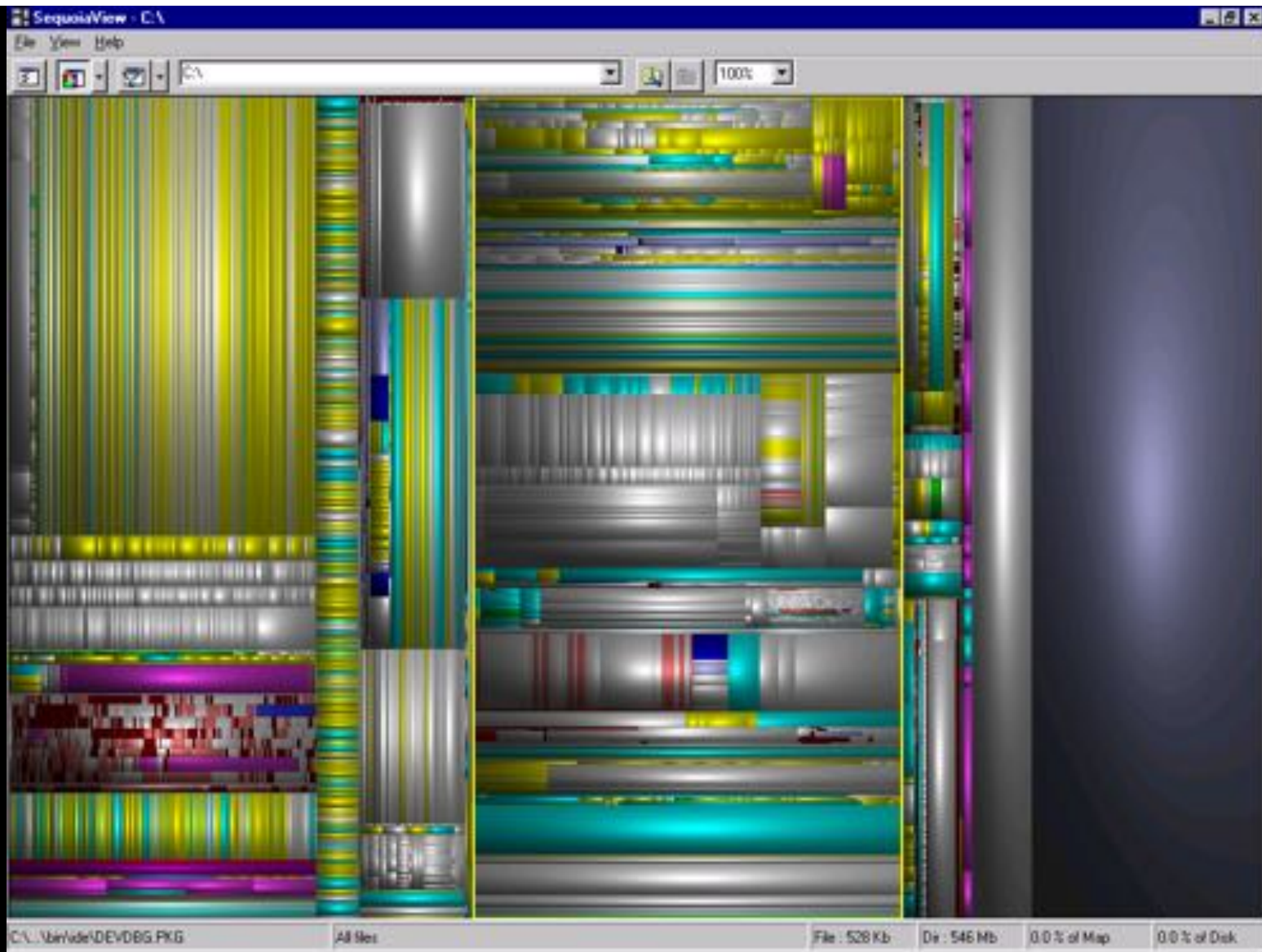
Strip







[ van Wijk & van de Wetering, InfoVis '99 ]



[ SequoiaView, '02; [www.win.tue.nl/sequoiaview](http://www.win.tue.nl/sequoiaview) ]

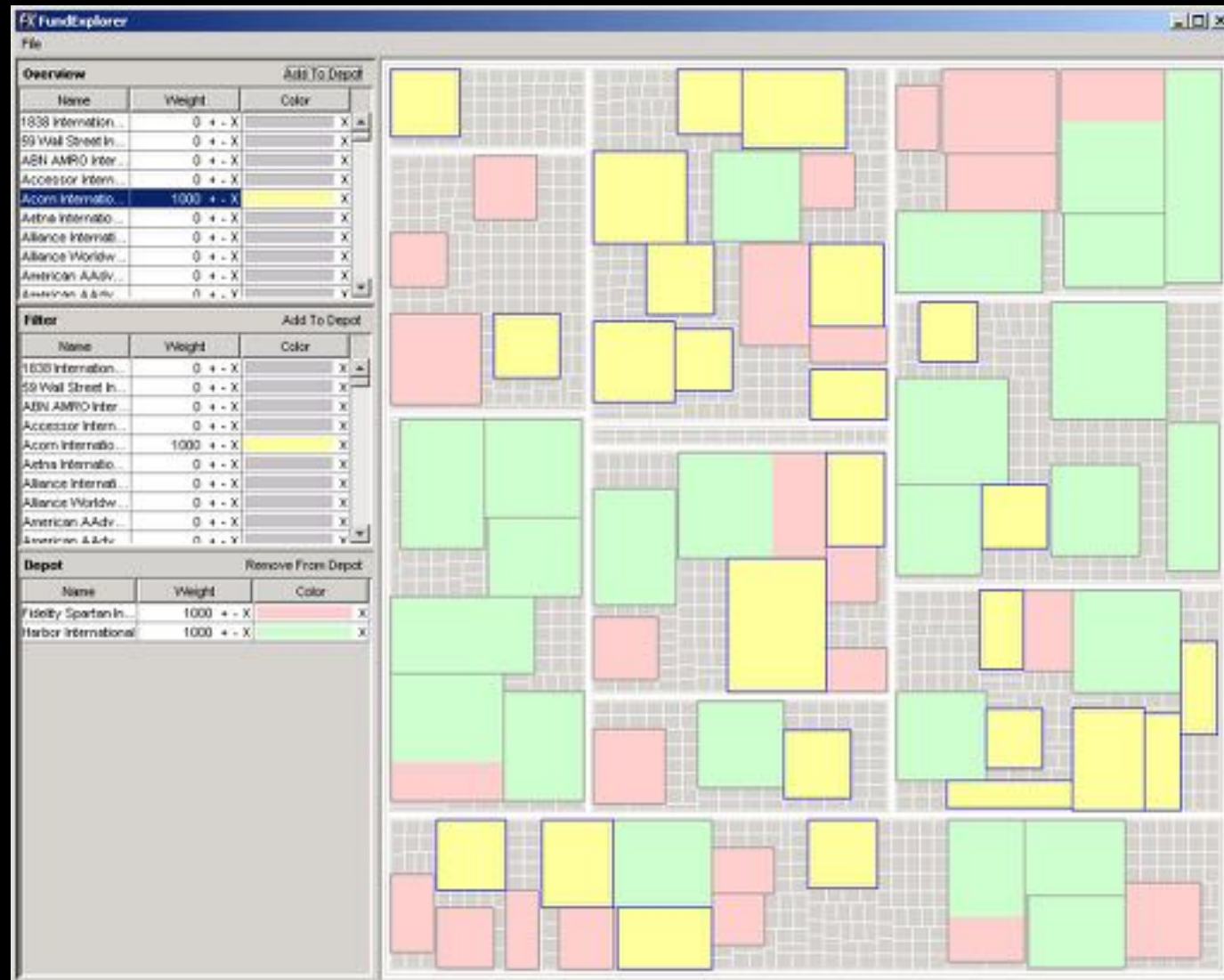


# What if zero is an important value?

- Stock or mutual-fund portfolios
- Zero size  $\rightarrow$  zero area

# FundExplorer

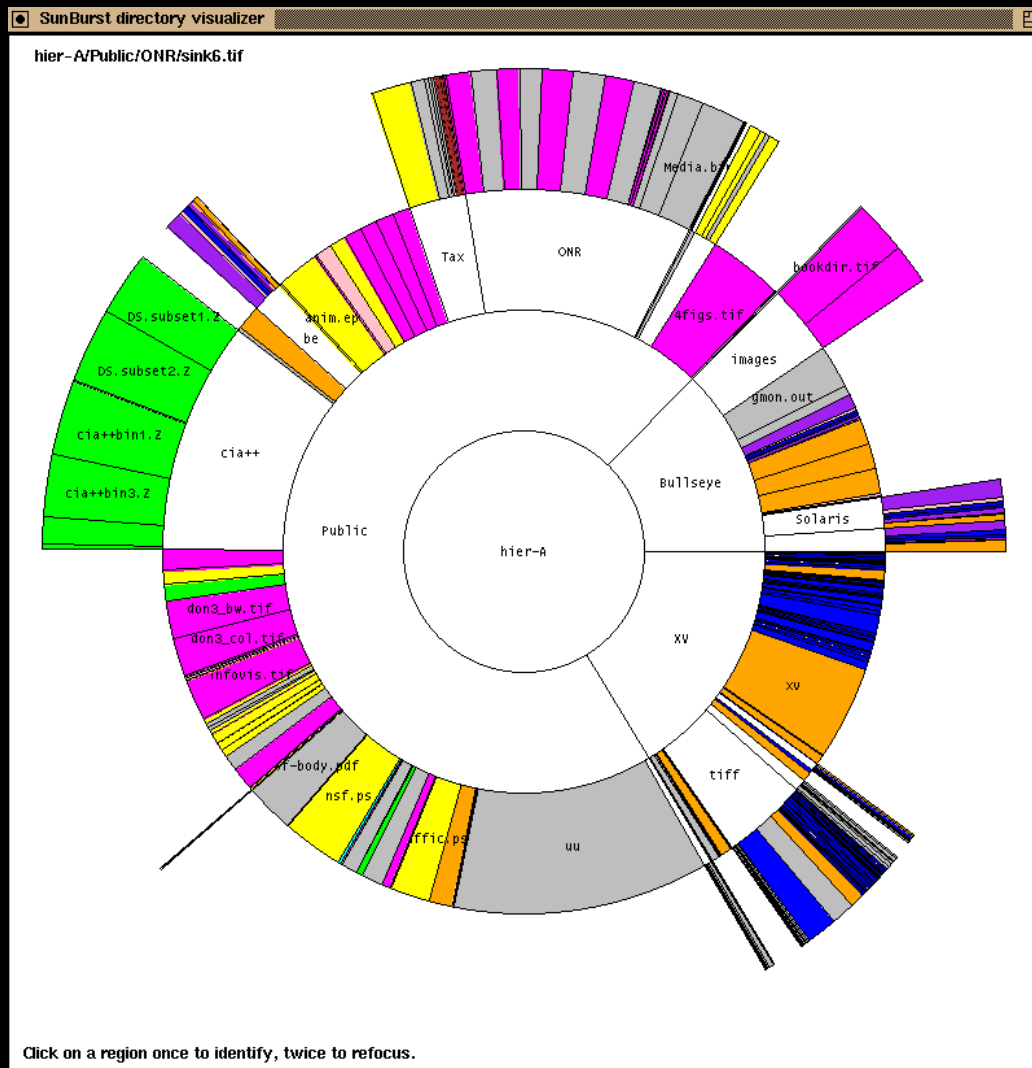
- Help the user manage diversification.



Student project!

**SunBurst**

- ps
- eps
- jpg
- tif
- tiff
- ppm
- pnm
- gif
- xwd
- c
- cc
- CC
- C
- h
- H
- o
- z
- gz
- zip
- html
- htm
- java
- class
- tex
- dvi
- blg
- bbl
- bib
- aux
- sty
- core
- ~
- directory
- (executable)
- (default)



**Treemap Control Panel**

*Treemap Directory Visualizer*

Focus level: 1

Up One Directory Top

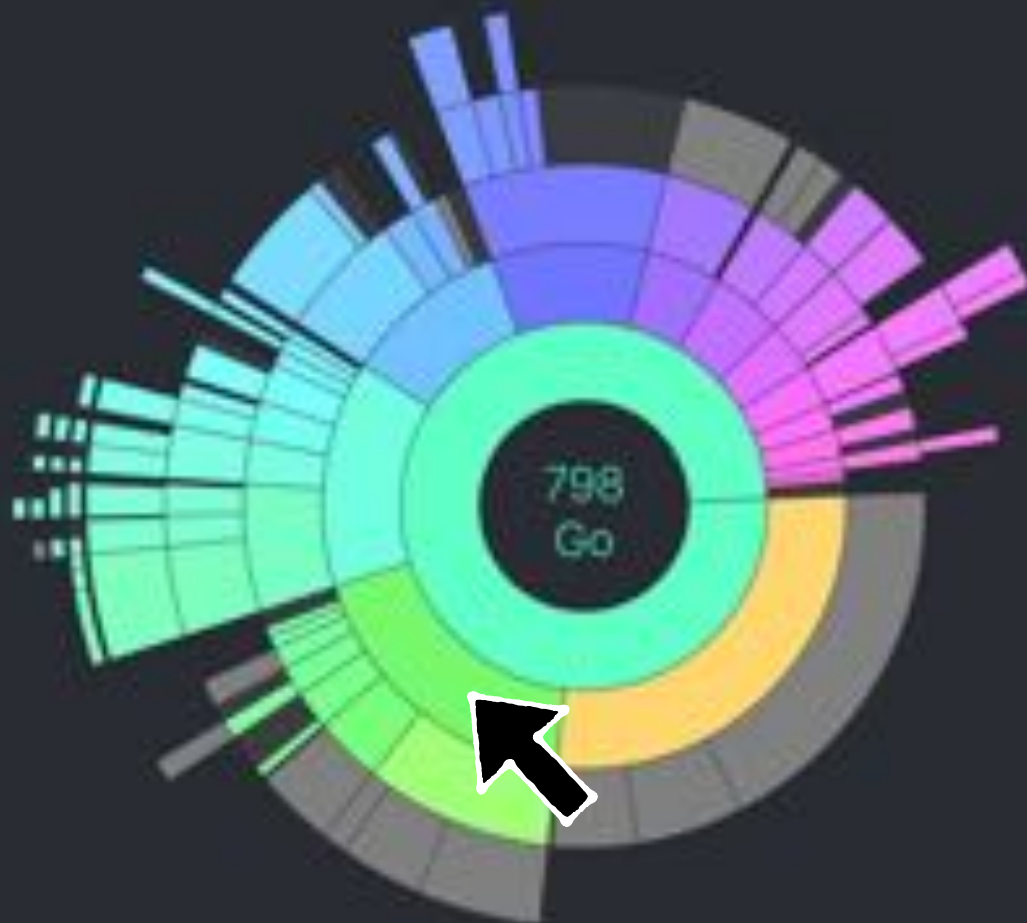
Bottom level shown: 1

Higher Deeper Maximum

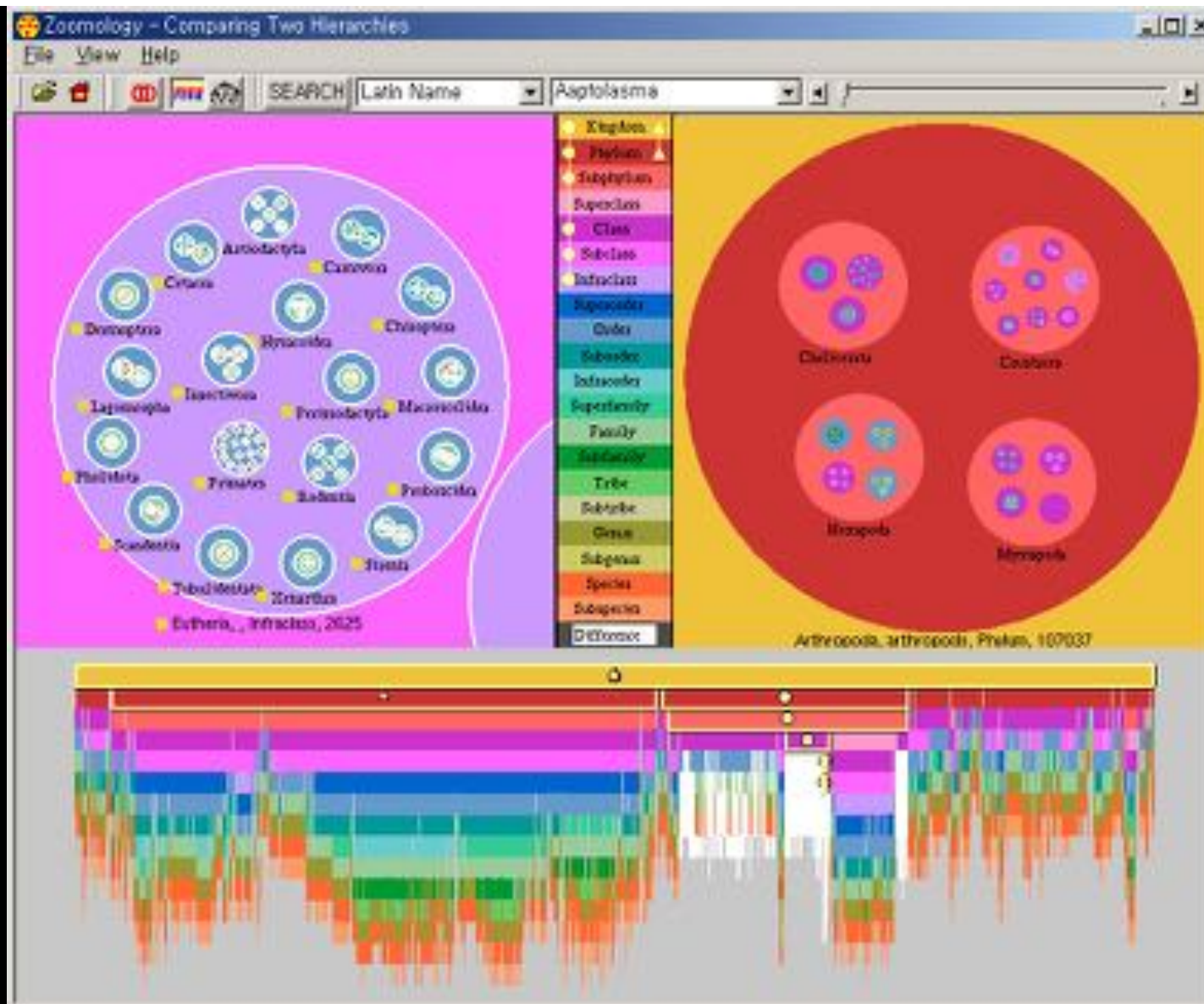
Color:

Type Age Random

Quit



● Documents	146,2 Go
● Parallels	71,6 Go
● Disc Images	25,7 Go
● Cours	14,6 Go
● Backups	11,1 Go
● Machines Virtuelles	8,3 Go
● Presentations	4,8 Go
● Bibliothèque calibre	3,7 Go
● ... objets plus petits	6,4 Go



[ Hong et al., InfoVis Contest Winner '03 ]

Student project!

118





# Multivariate Network Exploration

[ van den Elzen & van Wijk, InfoVis '14 ]

120



# Project update