

Trees aka Hierarchies



CS 4460 - Information Visualization

Jim Foley

- Some PPTs courtesy John Stasko
- Last Updated September 2016

Trees (aka as Hierarchies)



- Formal Definition
 - A directed acyclic graph AND
 - All nodes have in-degree one EXCEPT FOR
 - One distinguished node called the root
- Leaf node
 - Out degree = 0
- Binary tree
 - Out degree maximum = 2

Hierarchies/Trees in the World



- Pervasive
 - Family histories, ancestries
 - File/directory systems on computers
 - Organization charts
 - Animal kingdom: Phylum,..., genus,...
 - Object-oriented software classes
 - ...

Attributes

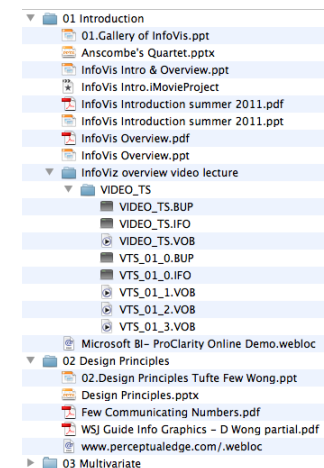
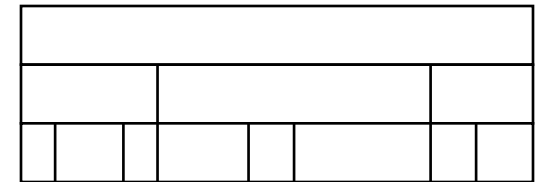
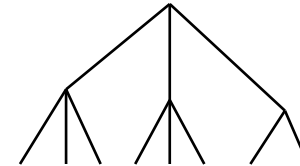


- In addition to structure, trees often have
- Attributes (information, variables, ...)
 - On links
 - On nodes
- Such as

Visualizing Trees – Four Methods



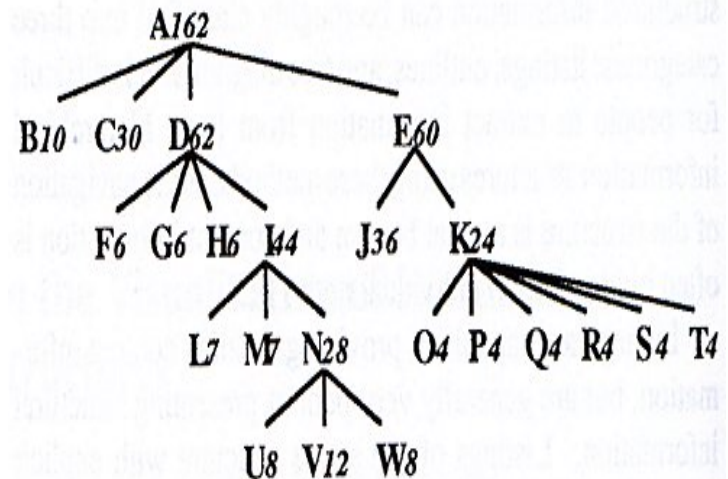
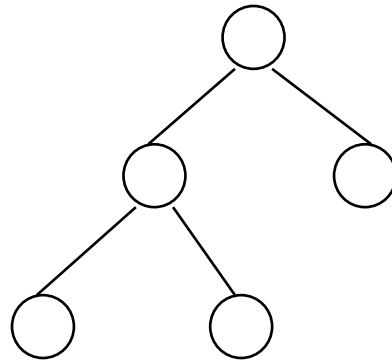
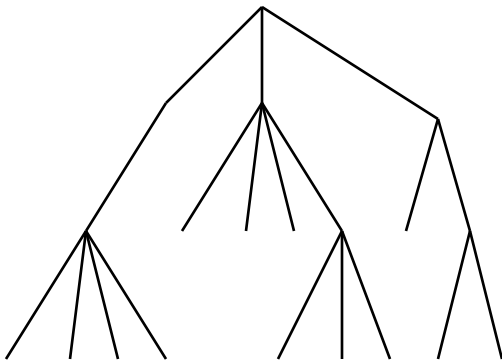
- Node-link
- Space-filling
- Connectivity matrix
 - For this case, a 12x12 binary matrix (12 nodes in tree)
- Indented list
- Many many more, see
 - <http://www.informatik.uni-rostock.de/~hs162/treeposter/poster.html>



Node-Link Diagrams



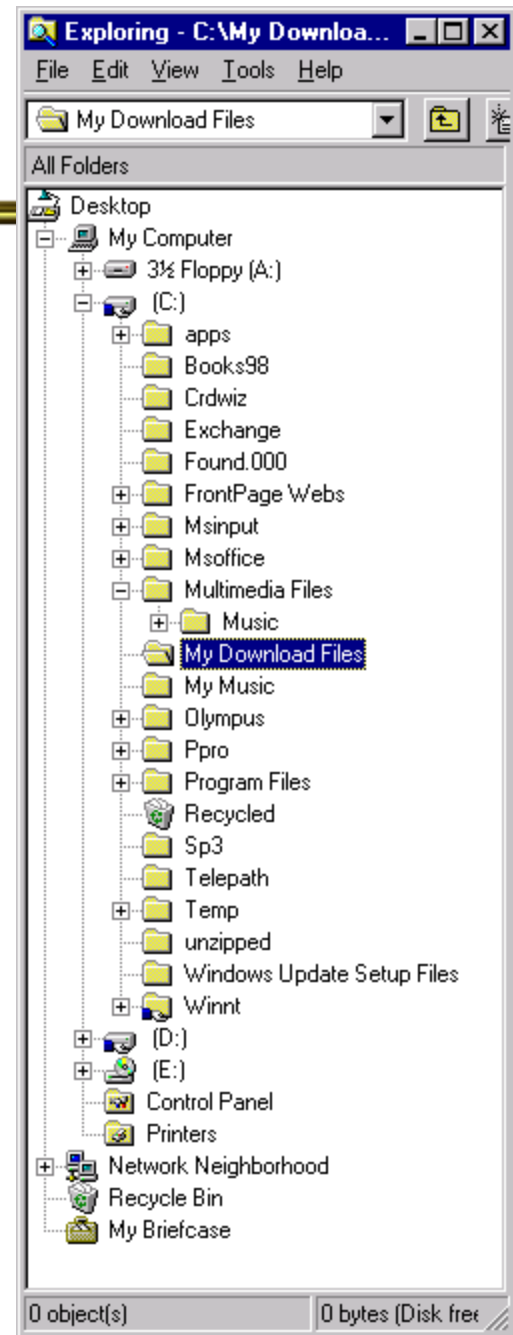
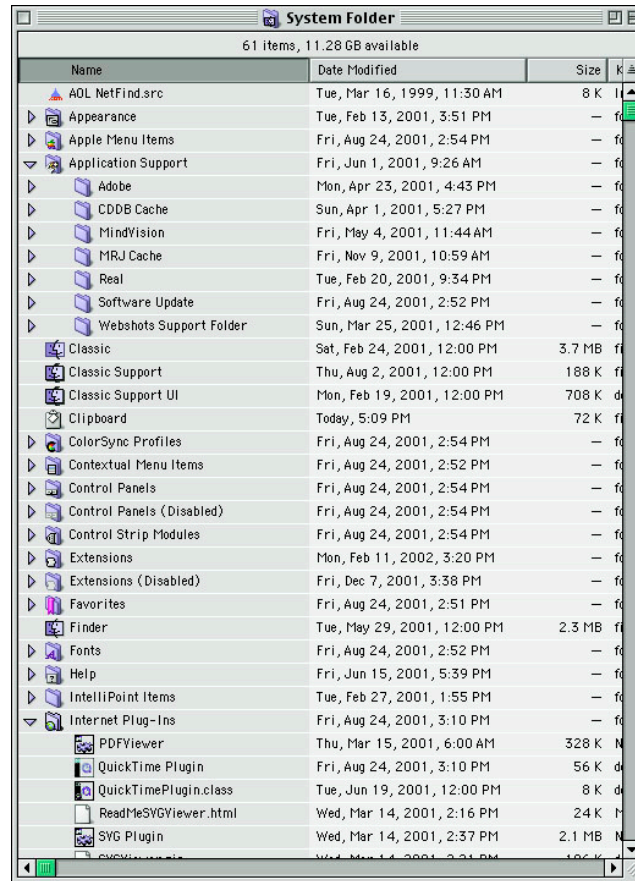
- Typically drawn with root at top, leaves at bottom



Indented List

Good for?
Maybe for
search

Bad for?
Structure
Why?



Given a Tree, what Questions?



- Useful to think about questions one would want to ask about a tree.
- Some tree visualizations better for some questions than others
- What questions might you ask?
- Confer in small groups; write down 4 questions

Typical Tree Questions (aka Tasks)



- Find a node whose name is X...
- Find a node that has properties x, y and z
- Find identical subtrees
- Find the deepest node (furthest away from root)
- Determine the parent of node x
- Determine grandparent of node x
- Determine the names of all nodes on path from node x to the root
- Determine depth of node x from root
- Find node with largest out-degree (for example, in org chart a large out-degree is problematical; parent with largest number of children)
- Is the tree balanced?

Note That



- Algorithms can accomplish some of these tasks
- Algorithms could be invoked by user in context, such as “find other subtrees identical to this one”
- Node-link diagrams show structure quite well, multiple attributes not as well
 - Need large nodes to encode multiple attributes, which limits number of nodes that can be seen

For Each Tree InfoVis, Consider



- How well Tree Tasks can be performed with each one
- How well they scale up to larger trees
 - Broader (higher out-degree)
 - Deeper
 - Broader and deeper
 - More attributes for each node
 - More attributes for each link
- How well they provide big picture and detail
- Recognize that displaying BIG trees is not so easy

Varieties of Node-Link InfoVis's

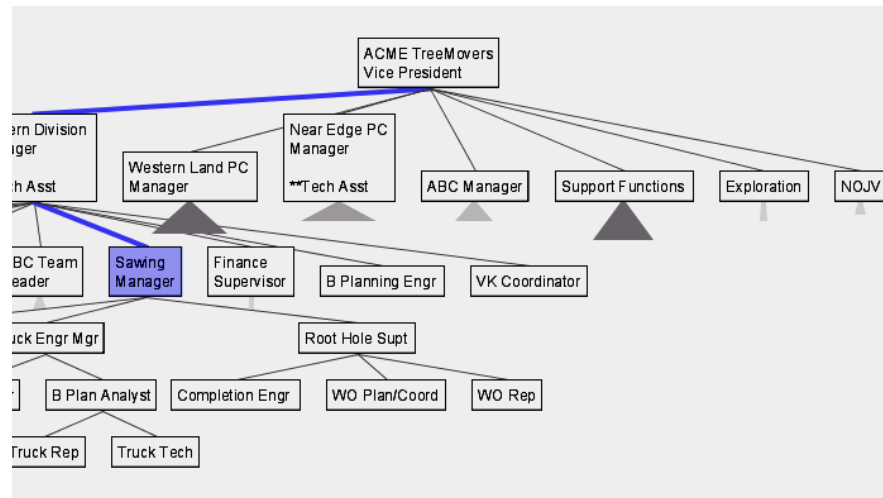


- Collapsing
 - Aka Degree of Interest Trees (DoIT)
- Hyperbolic

Collapsing (DoIT) Trees



- Use *focus + context* to limit how much of a tree is shown
- SpaceTree
 - Earlier lecture



- A similar d3 example

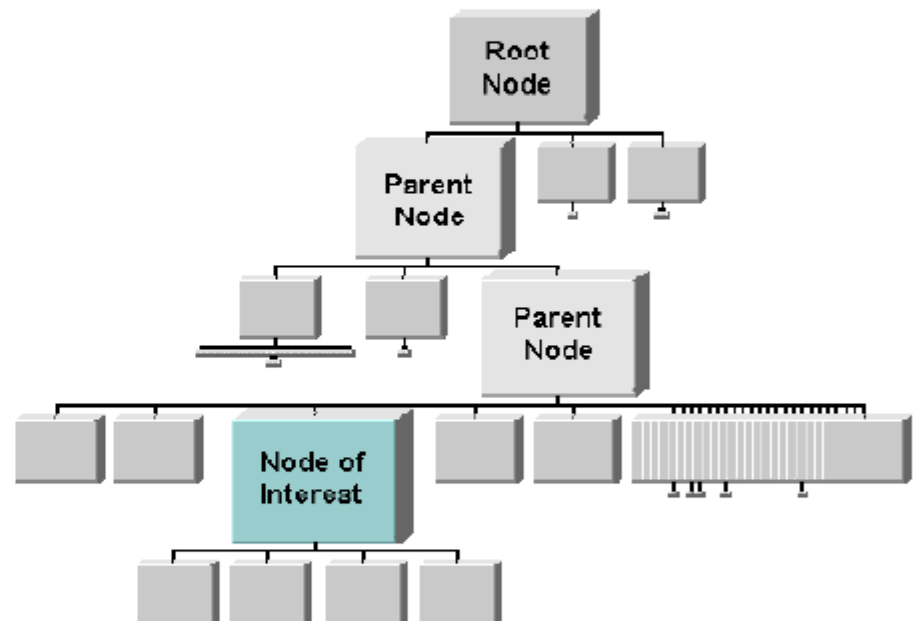
- <http://bl.ocks.org/robschmuecker/7880033>

Grosjean, Plaisant, Bederson - InfoVis '02

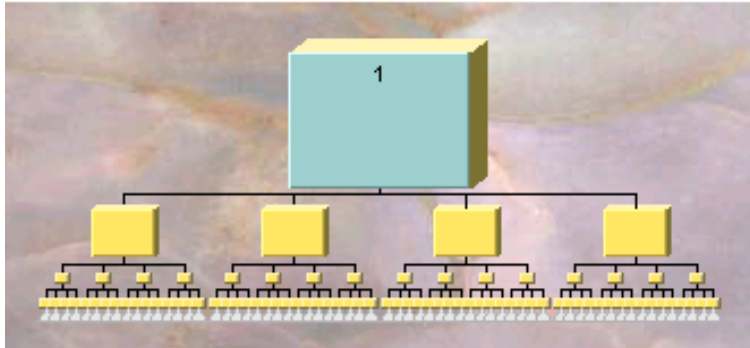
DoIT Approach



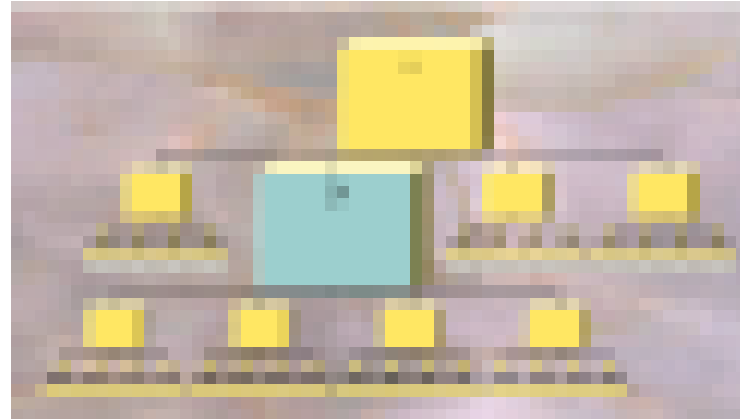
- Combine multiple ideas:
 - Expanded DOI computation
 - Logical filtering to elide nodes
 - Geometric scaling
 - Semantic scaling
 - Clustered representation of large unexpanded branches
 - Animated transition



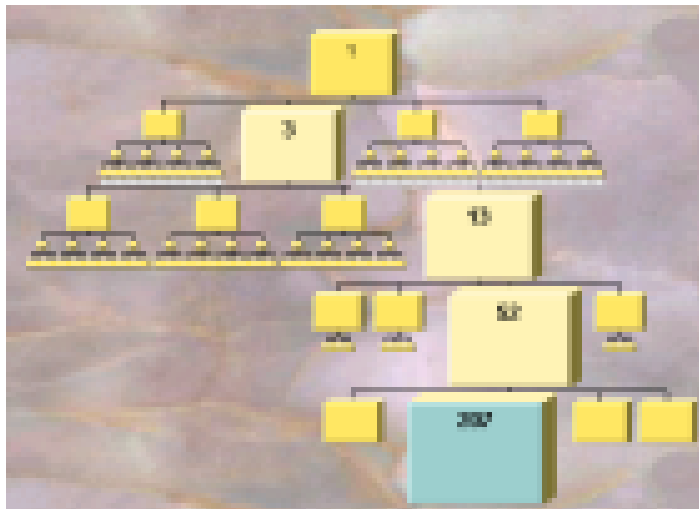
DoIT Example Operations



1. Display of a uniform tree of 4 levels

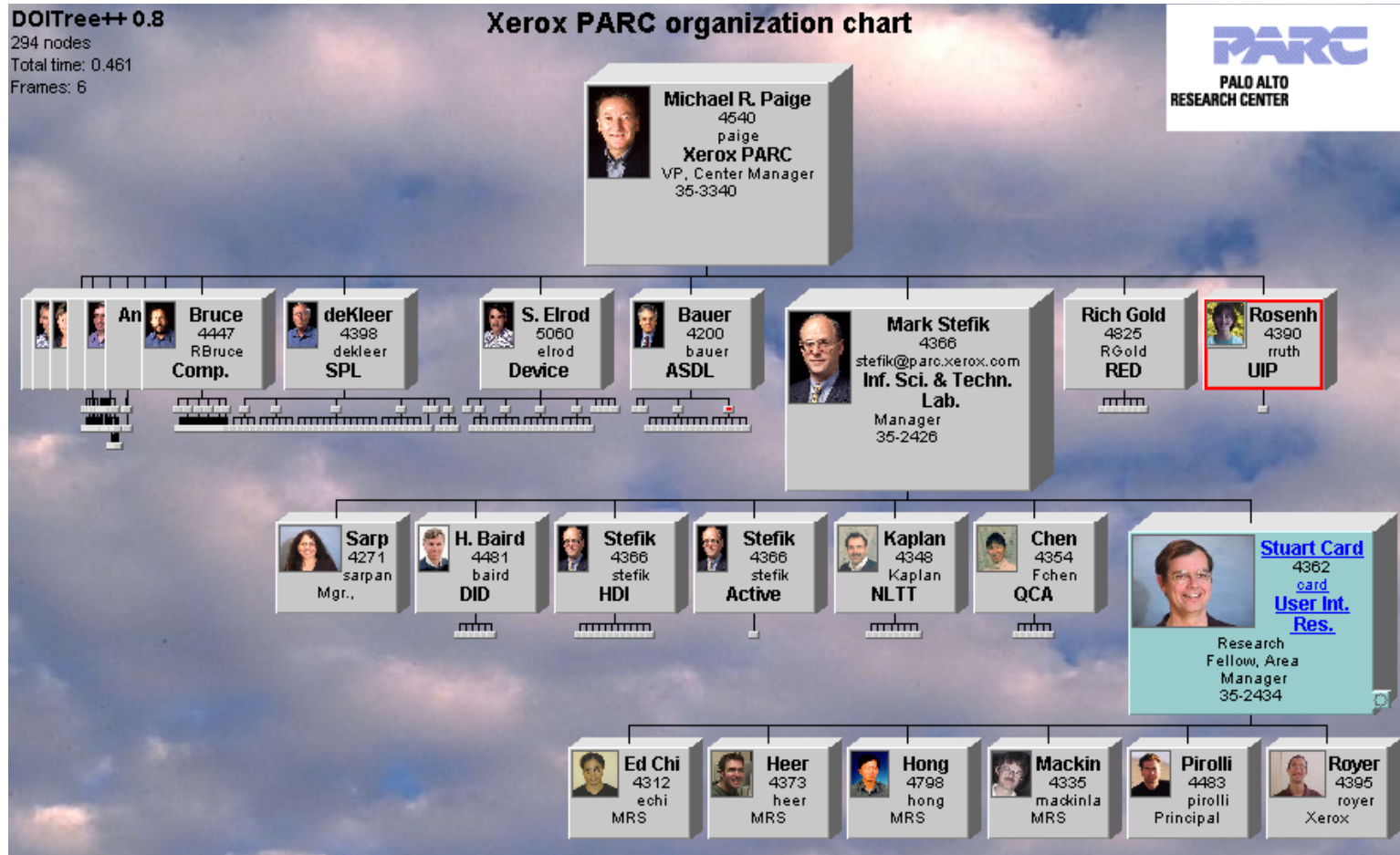


2. Same table with focus on Node 3



3. Same tree expanded down to a leaf node

DoIT Better View of Org Chart



Organization chart with over 400 nodes accessible over WWW through Web browser

Hyperbolic Tree Browser



- Focus + Context Technique
 - Detailed view blended with a global view
- First lay out the hierarchy on the hyperbolic plane
- Then map this plane to a disk
- Start with the tree's root at the center
- Demo
 - <https://philogb.github.io/jit/static/v20/Jit/Examples/Hypertree/example1.html>

Node-link Shortcoming



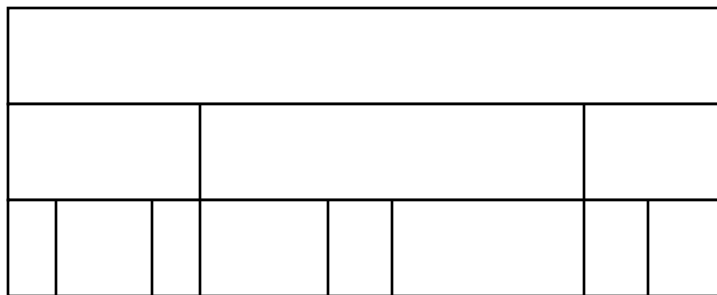
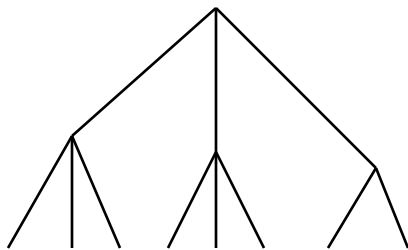
- Can encode multiple node attributes (DoIT, for instance) information for a few nodes, but not for many or all nodes
 - Run out of space
- At best, can visually encode a few node attributes using, for example
 - Color
 - Size
 - ...but all quickly clash with basic node-link structure

A Space-Filling Representation



Each item occupies an area

Children are “contained” under parent



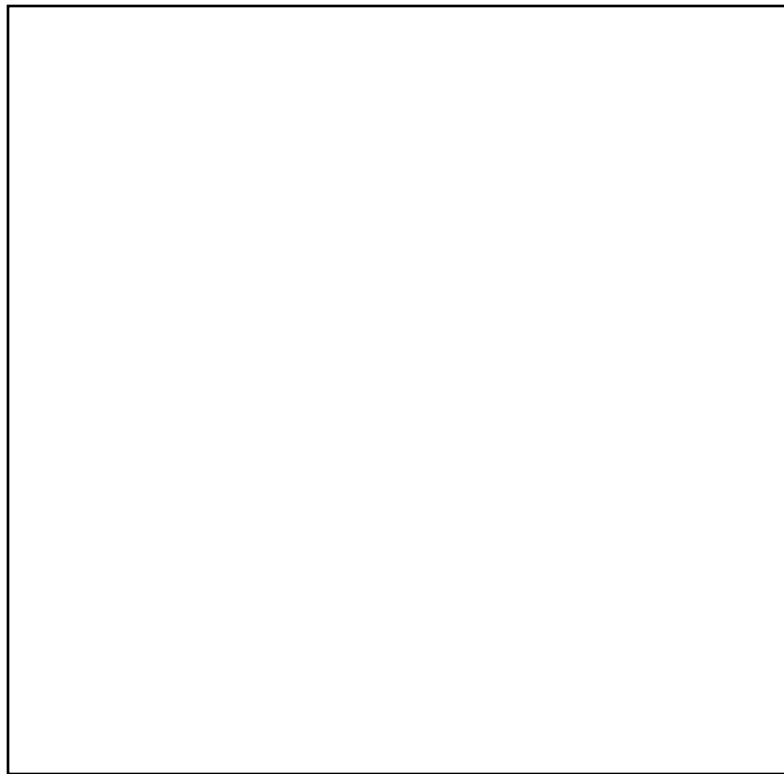
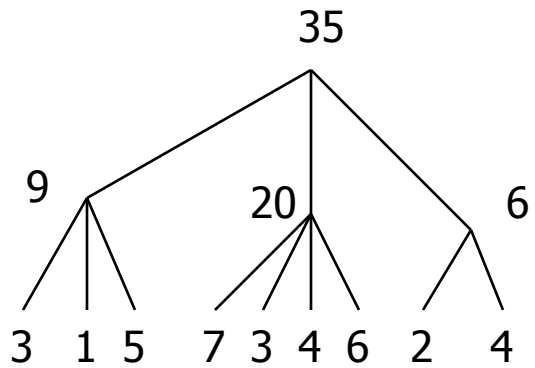
One example

Treemaps: Another Representation

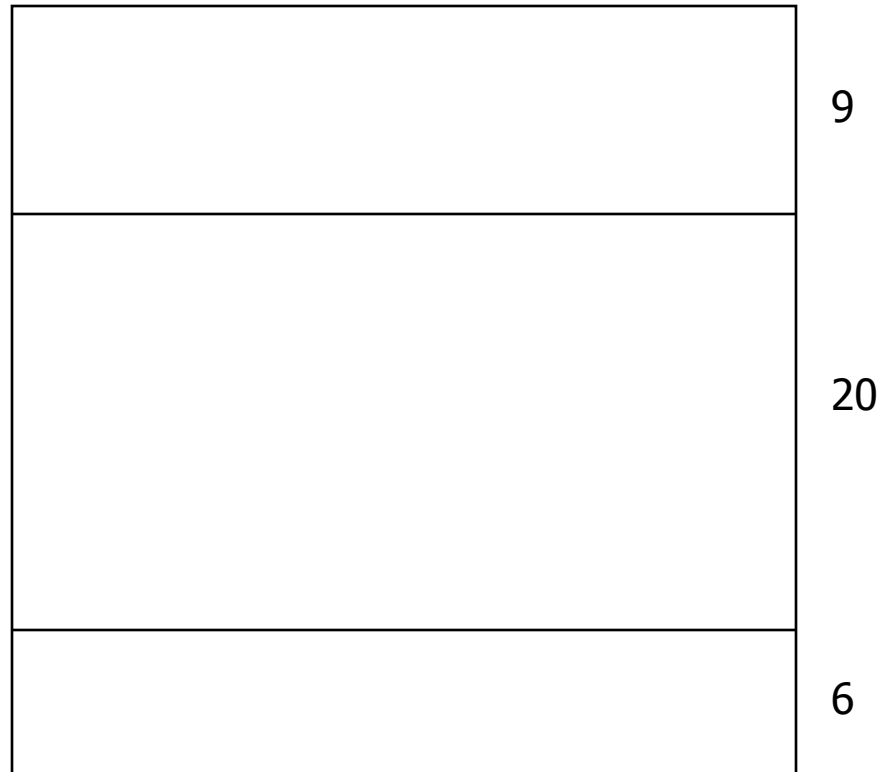
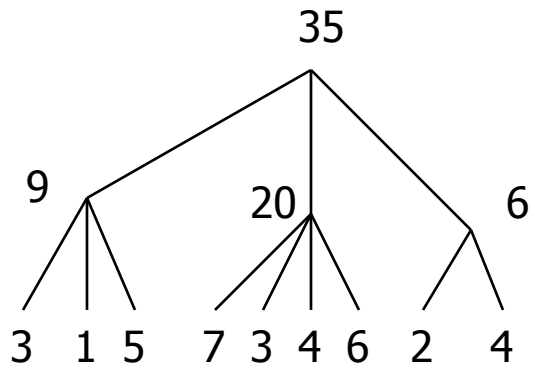


- Space-filling representation developed by Shneiderman and Johnson, Vis '91
- Children are drawn inside their parent
- Alternate horizontal and vertical slicing at each successive level
- Use area to encode one node attribute
 - Color can encode another attribute

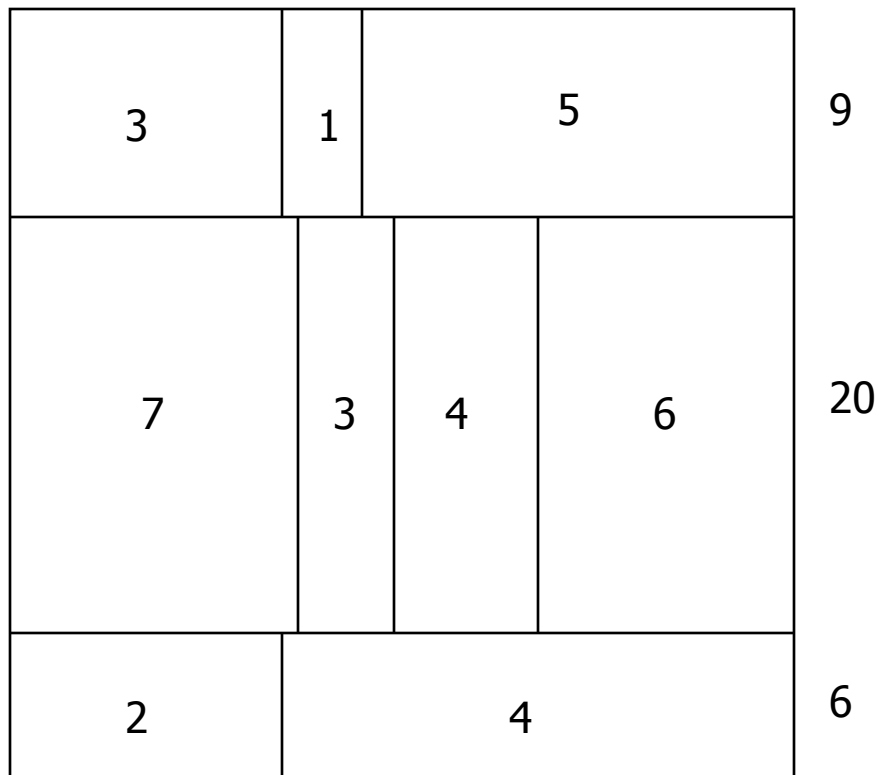
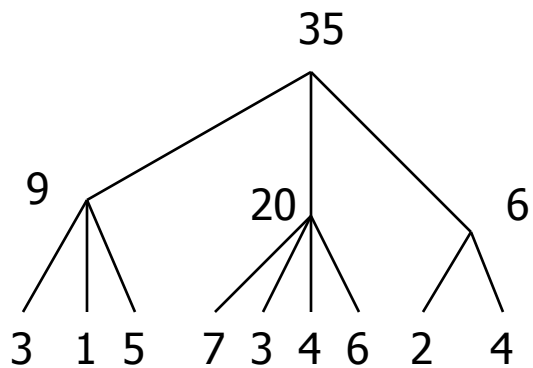
Example



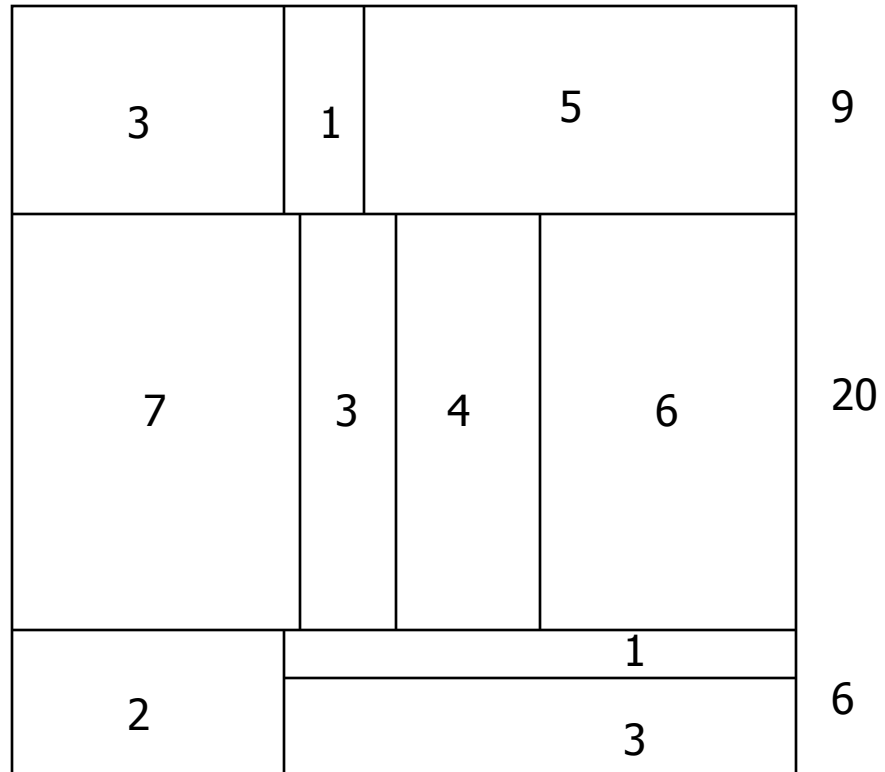
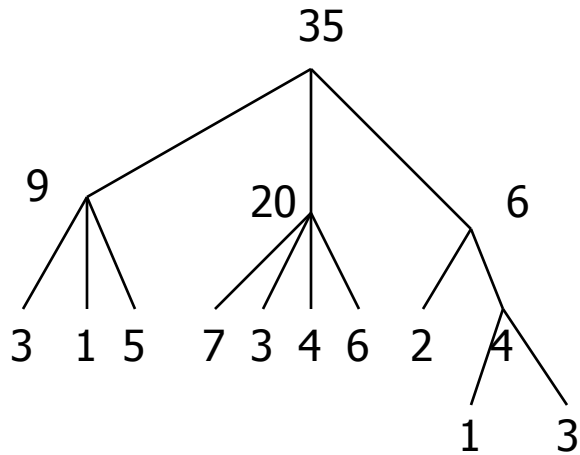
Example



Example



Example – add two more nodes



Applications



- Especially useful when sum of areas at each level has a meaning
 - File directory – disk space used
 - Stock market – market valuation or trading volume
 - Product sales – units sold, or \$\$ cost or profit
 - Software data – source code size, bug counts
 - What else?????

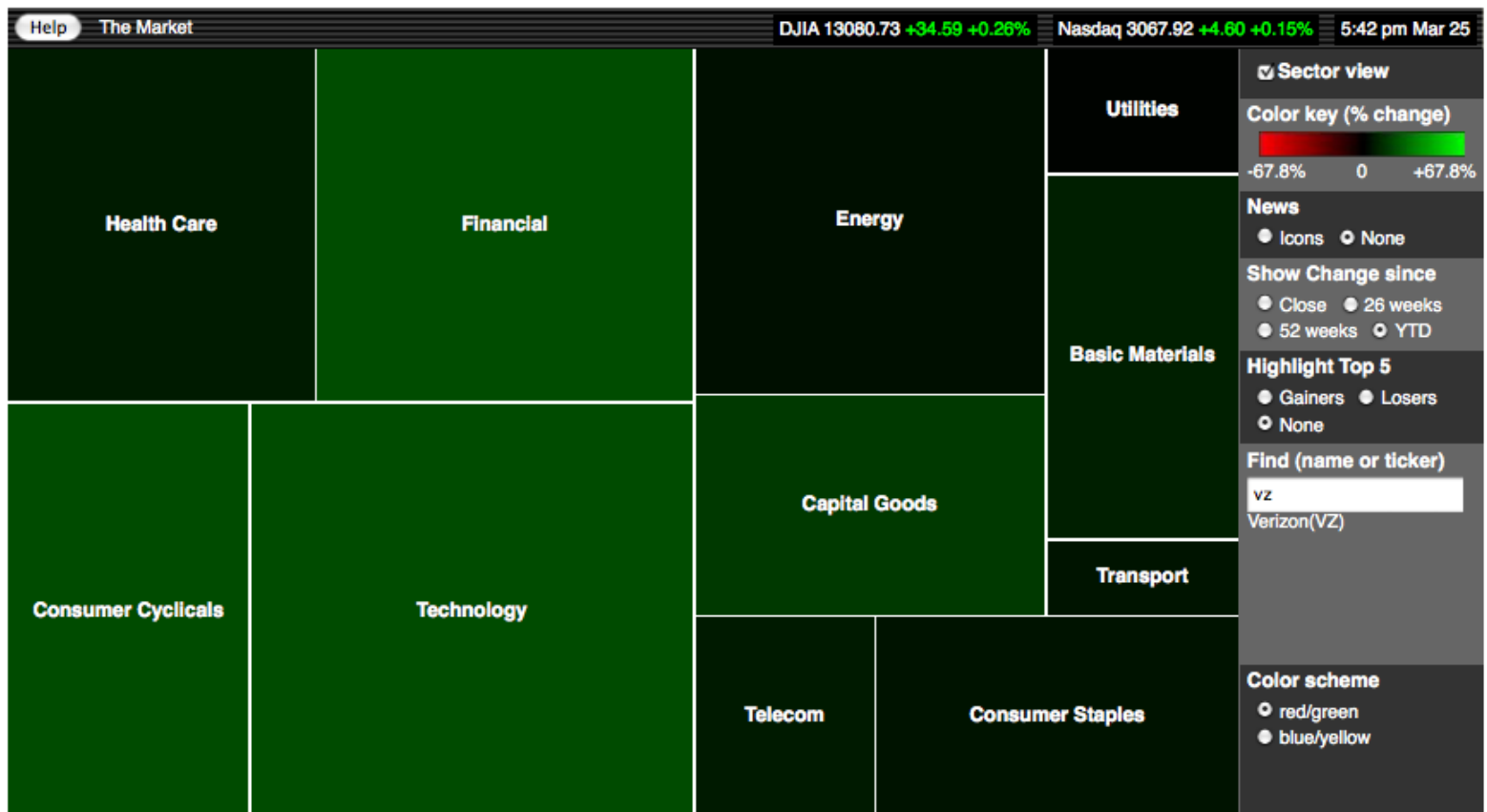
Map of the Market – Market Cap



<https://finviz.com/map.ashx>

Map of the Market

SmartMoneySelect Upgrade [here](#) to access the **Market Map 1000** and search 1,000 companies with enhanced capabilities.



Patent No.: US 6,583,794 B1

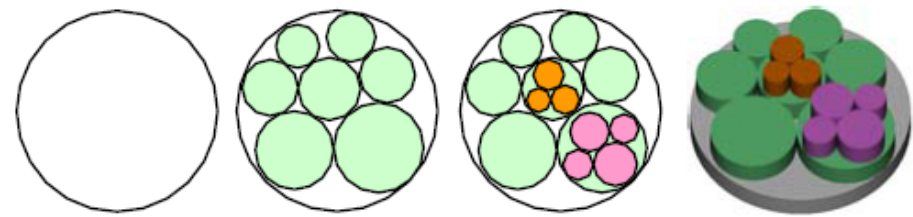
[Click Here to License the Map Applet](#)

Treemap Affordances



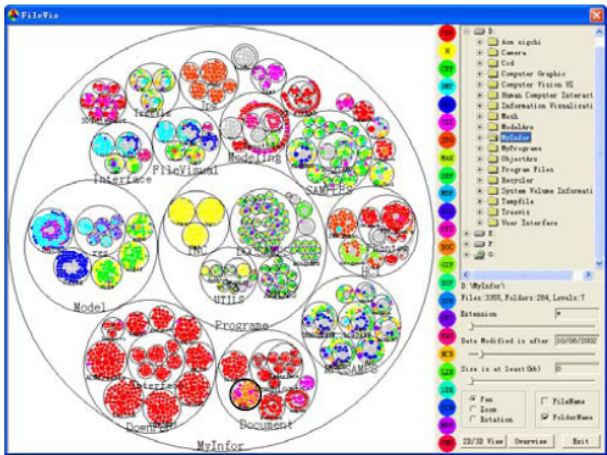
- Good representation of two attributes beyond node-link: color and area
- Not as good at representing structure
 - What happens if it's a perfectly balanced tree of items all the same size?
 - Also can get long-thin aspect ratios
 - Borders help on smaller trees, but take up too much area on large, deep ones
- What about link attributes?

Circle Packing

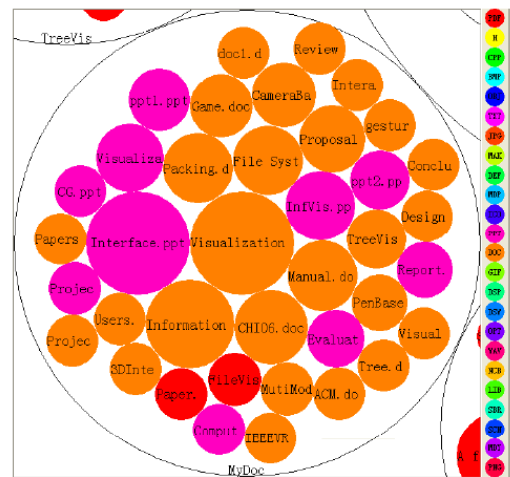


(a) Level 0 (b) Level 1 (c) Level 2 (d) 3D view

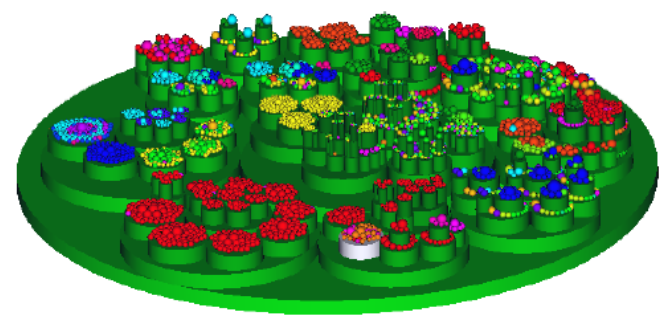
Figure 5. Pack circles into a circle recursively



(a) User interface and the overview of "D:\MyInfor"



(b) The details of the focus "MyInfor\Document\MyDoc"



(c) 3D nested cylinders and spheres

SunBurst



Treemap

- 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
- bb1
- bib
- aux
- sty
- core
- ~
- directory
- (executable)
- (default)

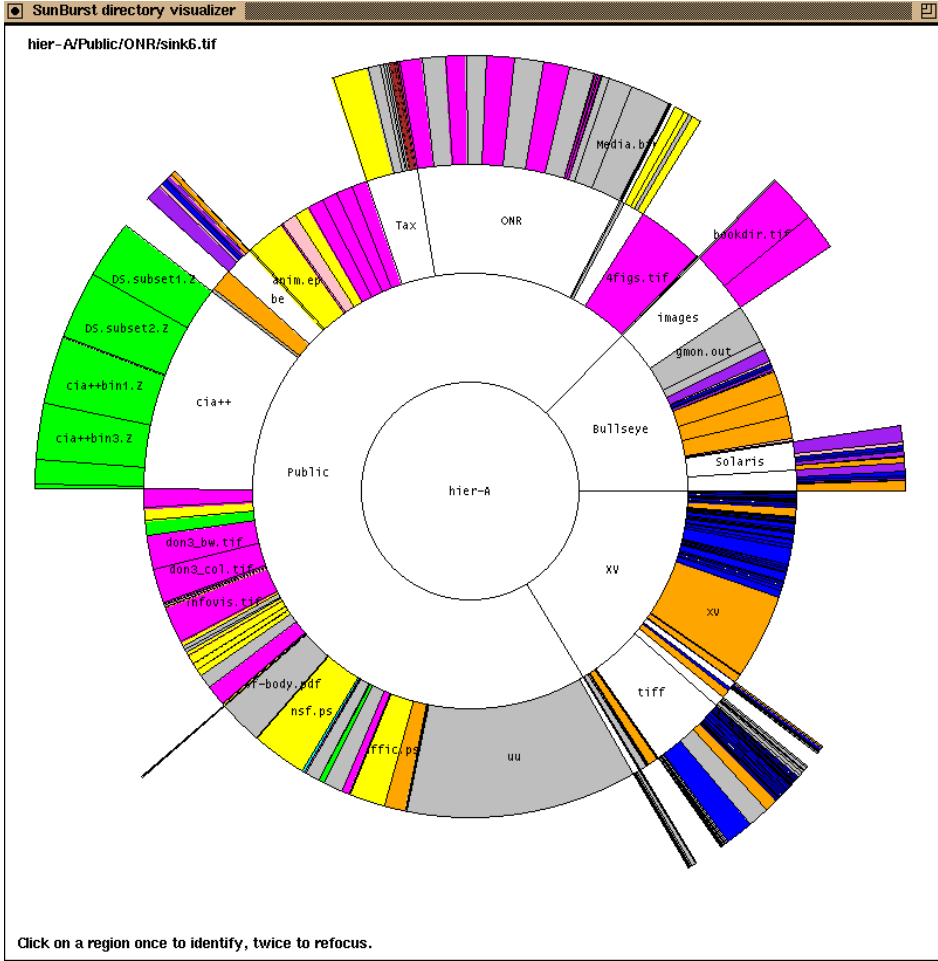
Treemap Control Panel

Treemap Directory Visualizer

Focus level: 1

Bottom level shown: 1

Color:



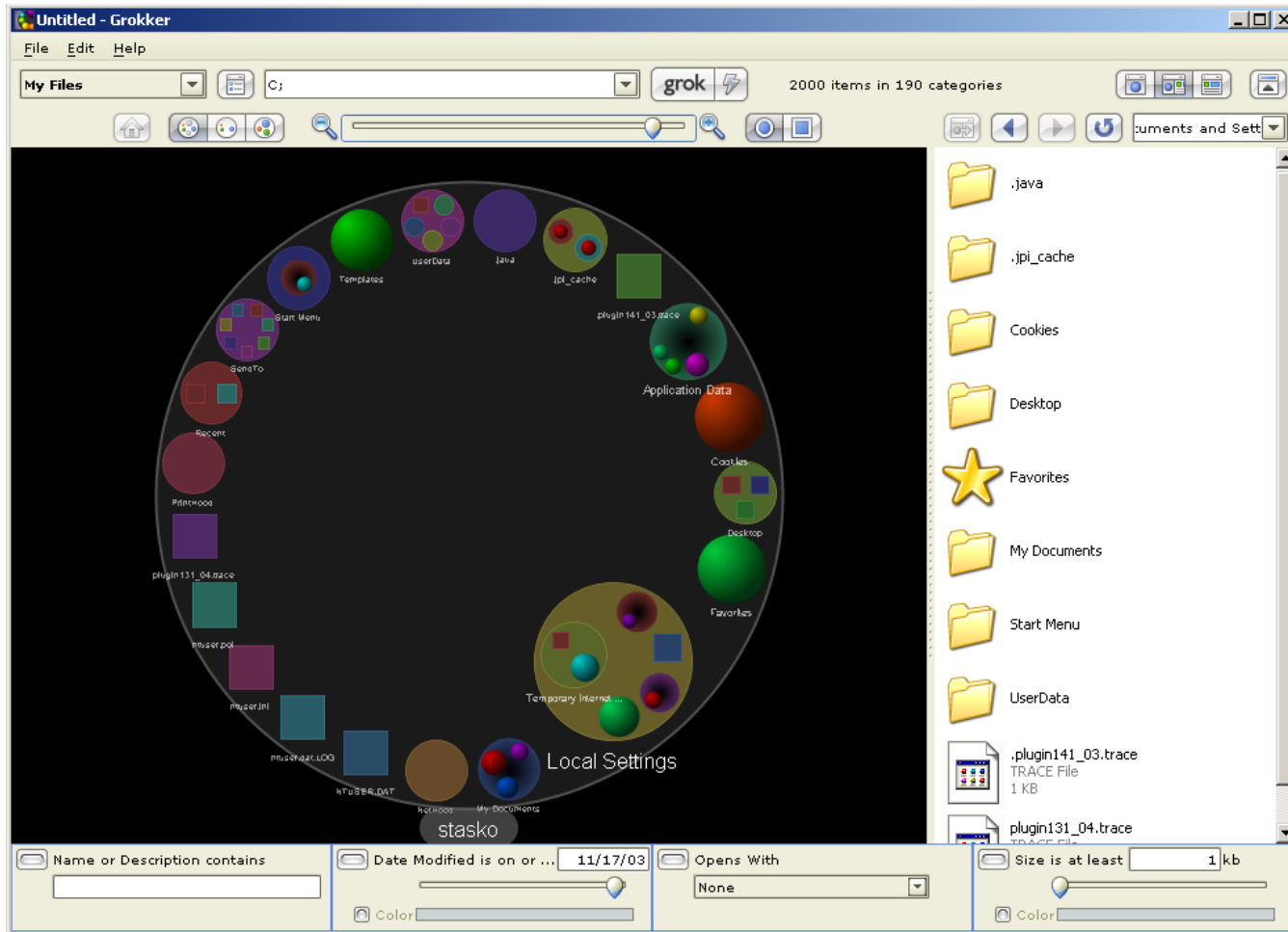
SunBurst



- Root directory at center, each successive level drawn farther out from center
- Sweep angle of item corresponds to size
- Color maps to file type or age
- Interactive controls for moving deeper in hierarchy, changing the root, etc.
- Double-click on directory makes it new root
 - d3 example: <http://bl.ocks.org/kerryrodden/477c1bfb081b783f80ad>
- Different idea <http://bl.ocks.org/kerryrodden/7090426>

Grokker

www.groxis.com - dead link

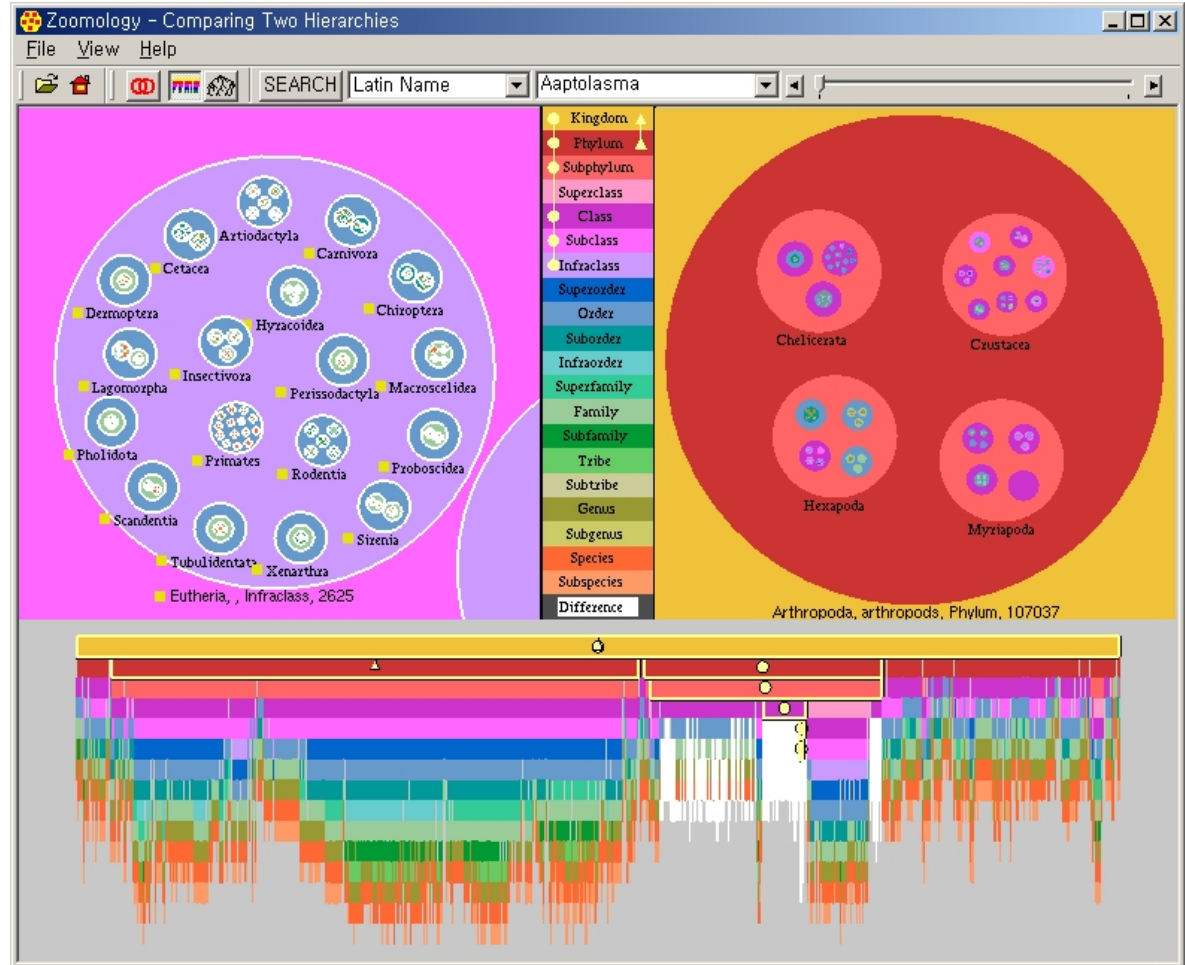


Zoomology

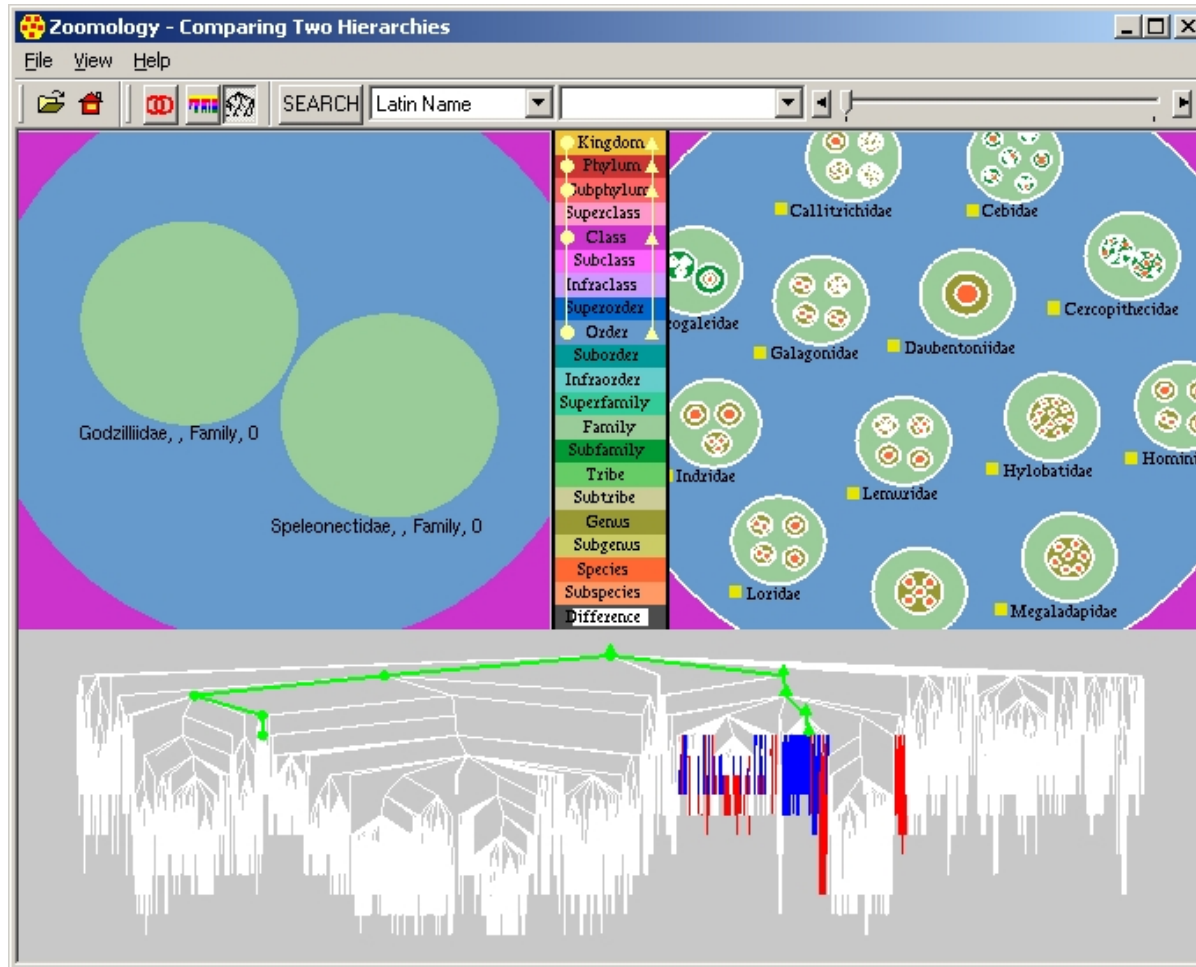


CS 7450
Spring '03
project

InfoVis '03
Contest Winner
Best Student
entry



Alternate View

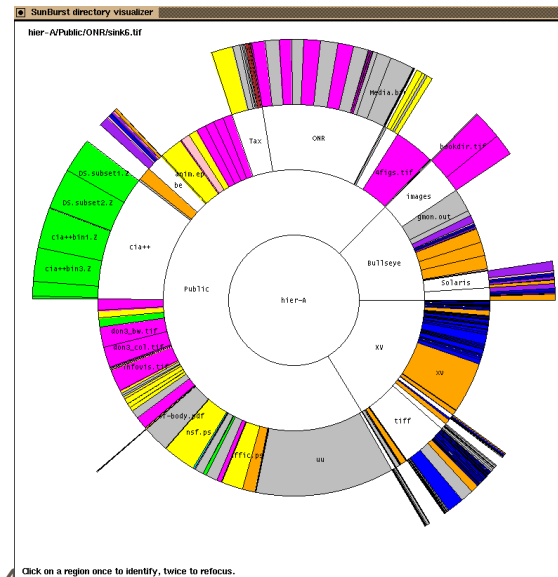
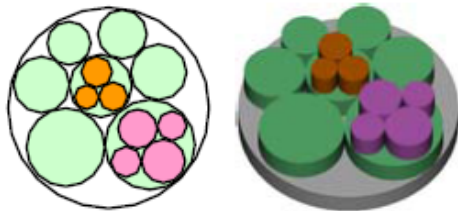


Four Varieties of Space-Filling



35					
9		20		6	

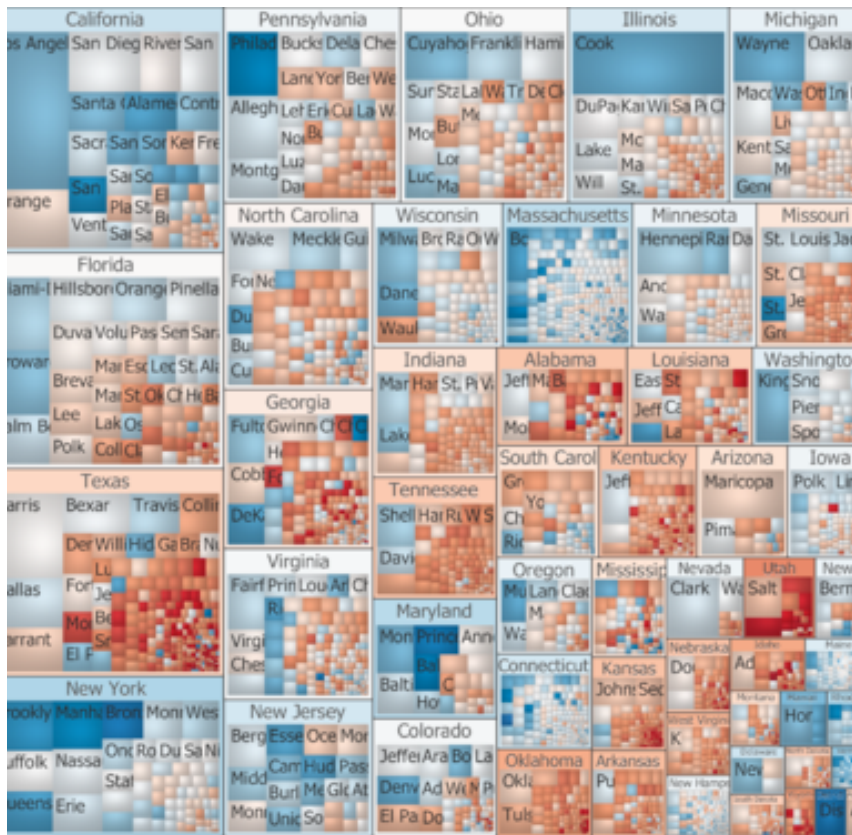
3	1	5		9	
7		3	4	6	20
2		4			6



Treemap and Sunburst



Treemap



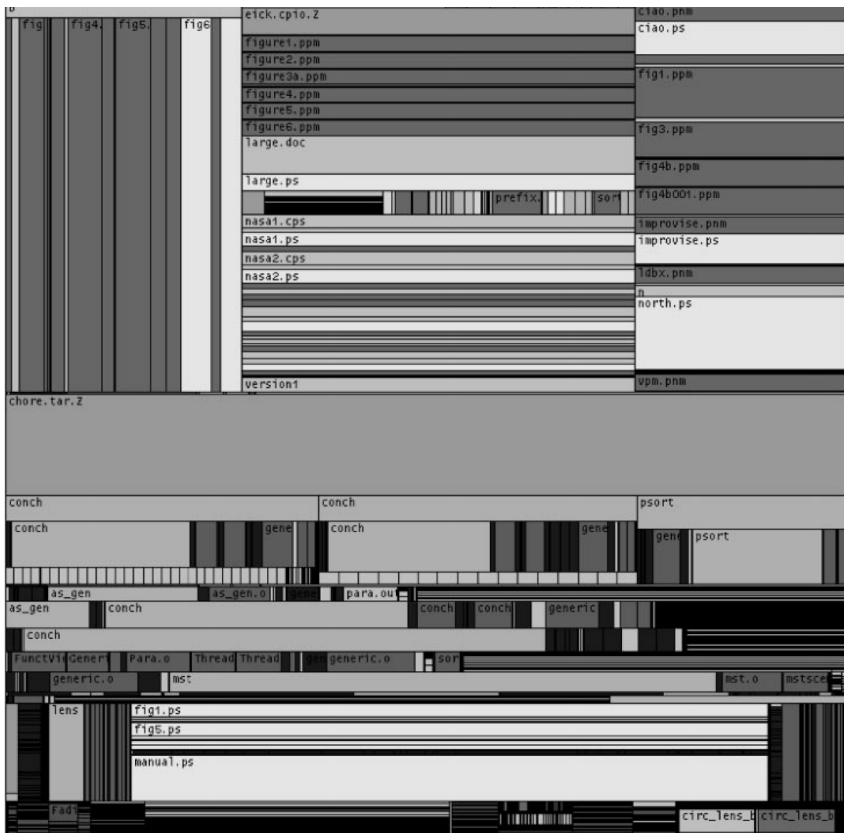
Sunburst



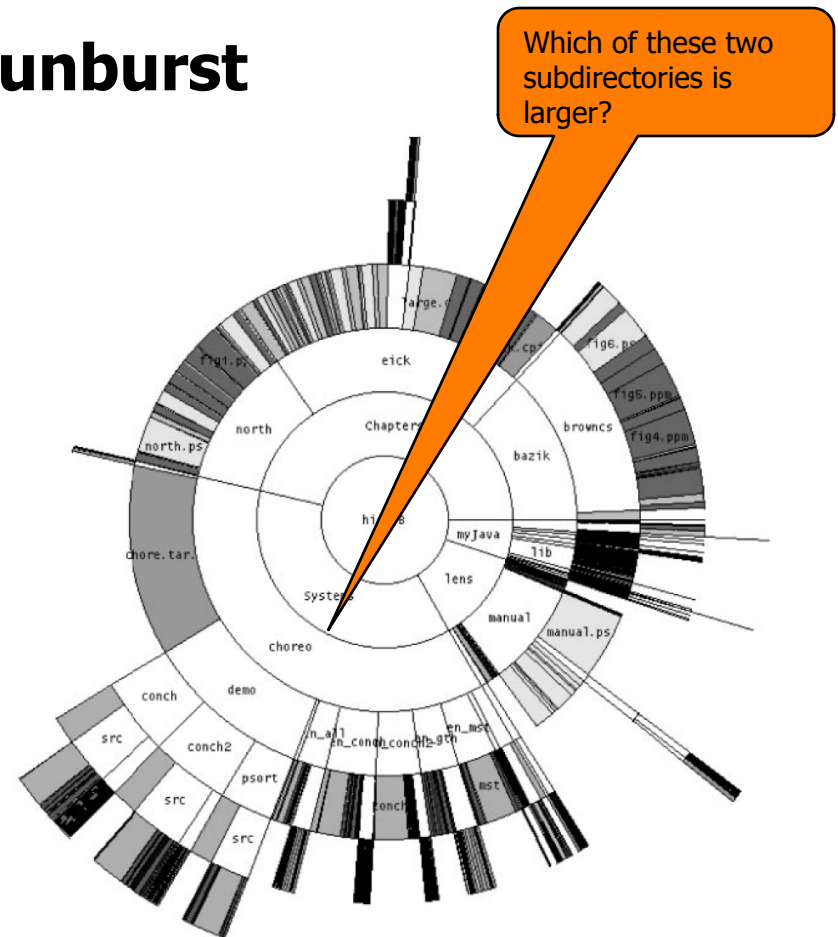
Treemap and Sunburst



Treemap



Sunburst



Treemap and Sunburst Pros/Cons



Treemap

- Unbalanced deep tree does NOT cause upper levels of tree to become small; does cause low-level nodes to become veryyy small
- Lower levels of deep tree hard to see
- Quantity encoded by area – easier to compare
- Uses full area no matter how wide or deep or unbalanced tree may be
- Small areas hard to judge

Sunburst

- Unbalanced deep tree causes upper levels (rings) to become narrow
- All levels clearly shown by concentric rings
- Small areas hard to judge
- Quantity encoded by angle subtended – harder to compare

Summary



- Node-link diagrams or space-filling techniques?
- It depends on user tasks!!
 - Node-link typically better at exposing structure
Essential if have link attributes
 - Space-filling good for focusing on one or two additional variables of cases
- Remember issue with link attributes
- Hybrids exist

Work With Your Neighbor



- Go back to your set of tree questions
- For each question, which basic approach better
 - Node-link?
 - Space filled?