

Information Visualization

Jing Yang
Fall 2007

1

Hierarchy and Tree Visualization

2

Hierarchies

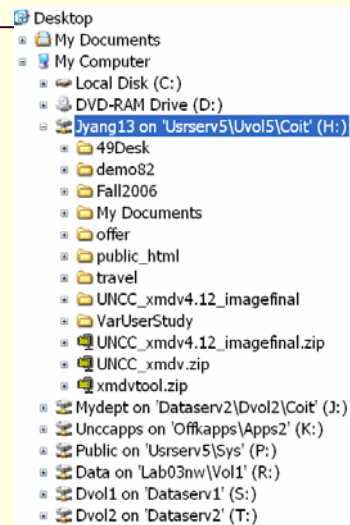
■ Definition

- An ordering of groups in which larger groups encompass sets of smaller groups.
- Data repository in which cases are related to subcases

3

Hierarchies in the World

- Family histories, ancestries
- File/directory systems on computers
- Organization charts
- Object-oriented software classes



4

Discussion

- What do you want to do with a hierarchy?

Assume that you have

- Family histories, ancestries
- File/directory systems on computers
- Organization charts
- Object-oriented software classes

5

Discussion

- How will you draw a hierarchy?

Assume that you have

- Family histories, ancestries
- File/directory systems on computers
- Organization charts
- Object-oriented software classes

6

Good Hierarchy Visualization

- Allow adequate space within nodes to display information
- Allow users to understand relationship between a node and its context
- Allow to find elements quickly
- Fit into a bounded region
- Much more

7

Trees

- Hierarchies are often represented as trees
 - Directed, acyclic graph
- Two major categories of tree visualization techniques:
 - Node-link diagram
 - Visible graphical edge from parents to their children
 - Space-filling

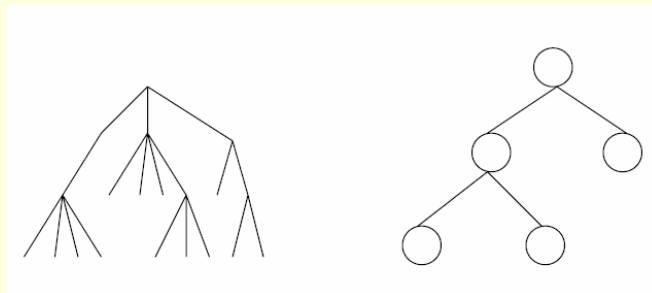
8

Node-Link Diagrams

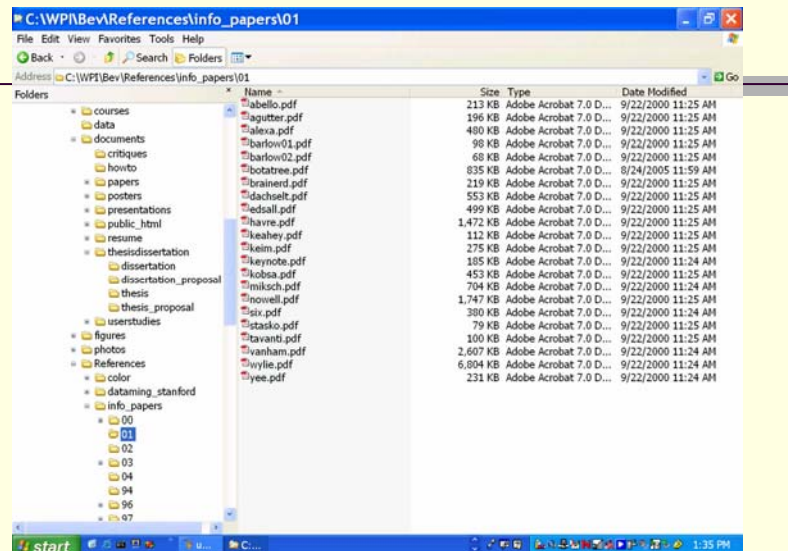
9

Node-Link Diagrams

- Root at top, leaves at bottom is very common



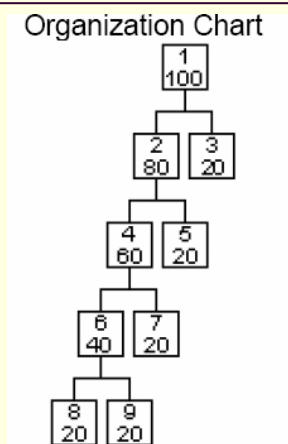
Microsoft Explorer



What do you like and dislike about it?

11

Organization Chart



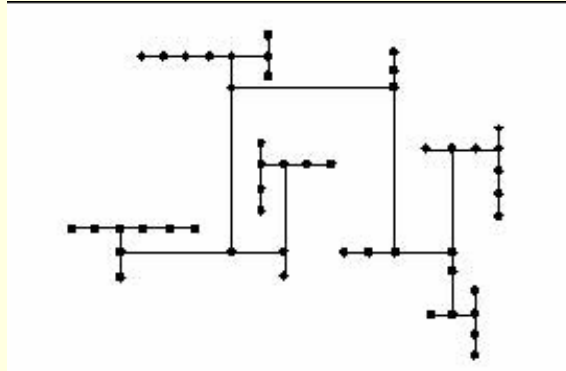
A decision tree

The figure is from Barlow and Neville InfoVis 2001

12

H-Tree Layout

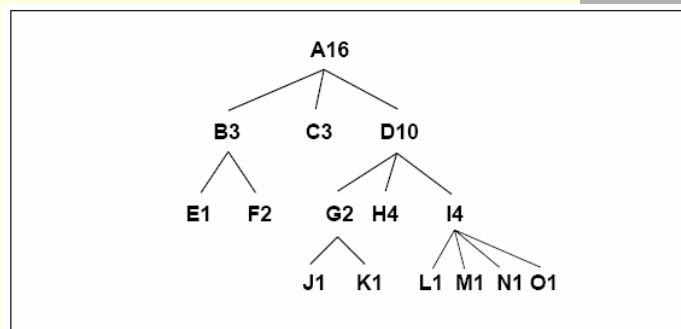
- Work well only for binary trees



Herman, G. Melançon, M.S. Marshall, "Graph Visualization in Information Visualization: a Survey" In: *IEEE Transactions on Visualization and Computer Graphics*, 2000, pp. 24-44.

13

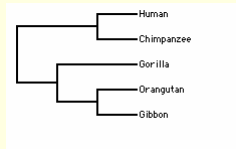
A Common Visualization



E. Kleiberg et. al. InfoVis 2001

14

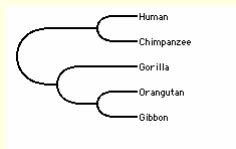
Different Styles



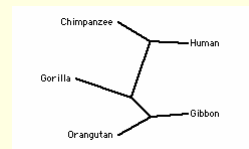
Rectangular: Well suited for displaying labeled/scaled trees.



Straight: Works well only on rooted binary trees.



Smooth Edges: Very similar to the rectangular mode

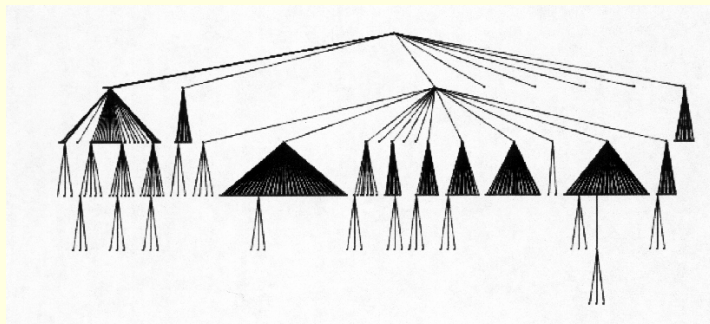


Radial: Works well for visualizing unrooted trees.

15

<http://www.hyphy.org/docs/GUIExamples/treepanel.html>

A Classical Hierarchical View



Position children "below" their common ancestors

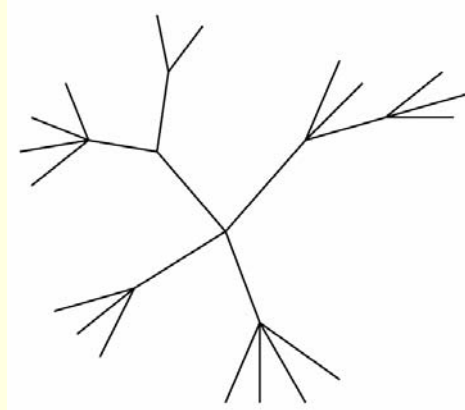
Layout can be top-down, left-to-right and grid like positioning

Fast: linear time

E. Reingold and J. Tilford. *Tidier drawing of trees*. IEEE Trans. Softw. Eng., SE-7(2):223-- 228, 1981

16

Why Put Root at Top (Left)



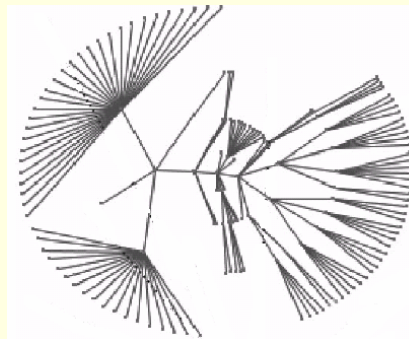
- Root can be at center with levels growing outward too
- Can any node be the root?

J. Stasko's InfoVis class slides

17

Radial View

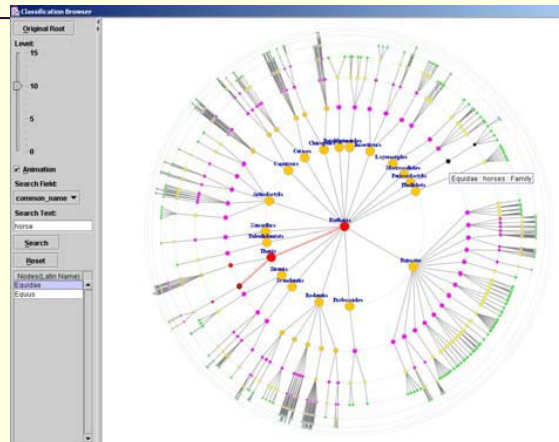
- Recursively position children of a subtree into circular wedges
- the central angle of these wedges are proportional to the number of leaves



P. Eades, "Drawing Free Trees", *Bulleting of the Institute fro Combinatorics and its Applications*, 1992, pp. 10-36.

18

Radial View



Infovis contest 03 Treemap, Radial Tree, and 3D Tree Visualizations
Nihar et. al. Indiana University 19

Balloon View

- Siblings of sub-trees are included in circles attached to the father node.

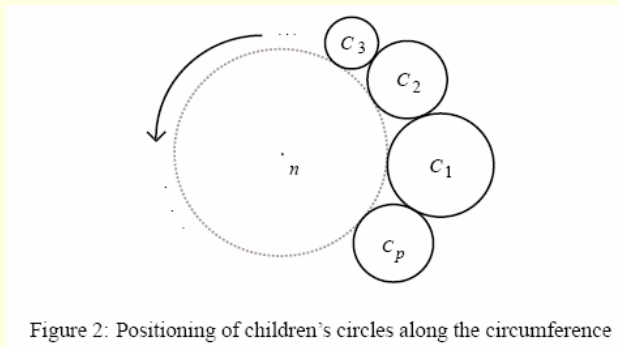
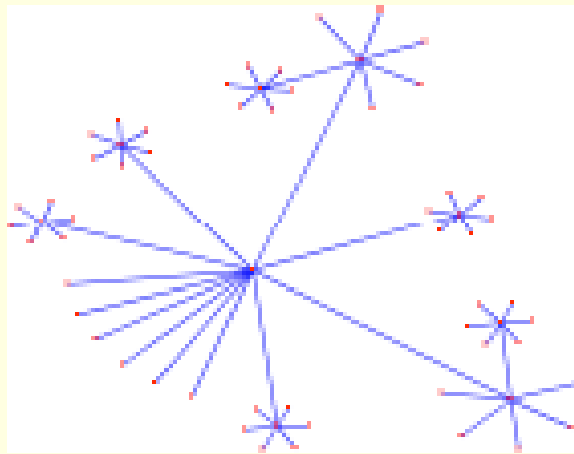


Figure 2: Positioning of children's circles along the circumference
Melancon, G., Herman, I.: Circular drawing of rooted trees. Reports of the Centre for Mathematics and Computer Sciences (CWI), INSR9817,

Balloon View



Melancon, G., Herman, I.: Circular drawing of rooted trees. Reports of the Centre for Mathematics and Computer Sciences (CWI), INSR9817,

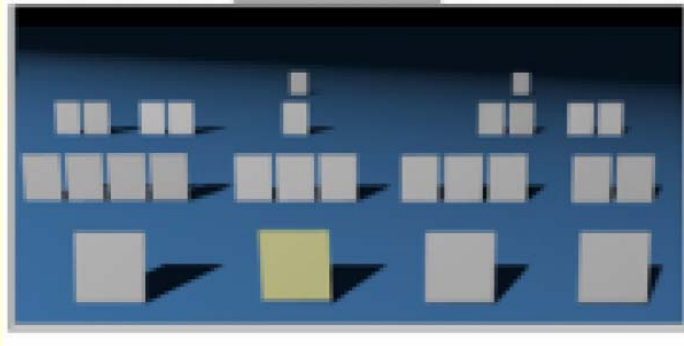
21

The Challenges

- Scalability
 - # of nodes increases exponentially
 - Available space increases polynomially (circular case)
- Showing more attributes of data cases in hierarchy or focusing on particular applications of trees
- Interactive exploration

22

3D Tree



Tavanti and Lind, InfoVis 01

23

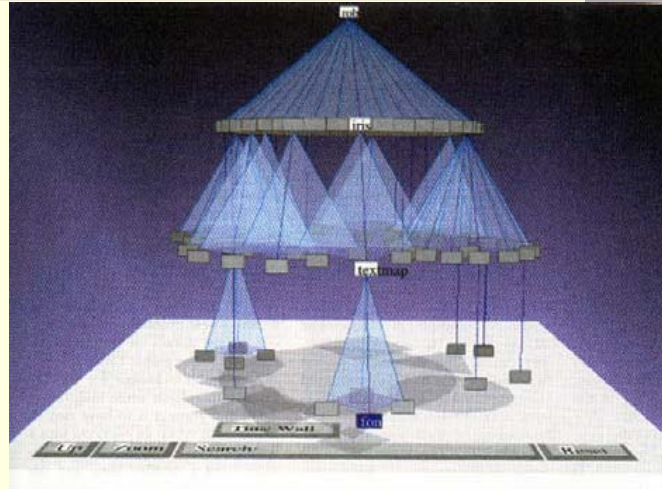
Cone Tree

- Key ideas:
 - Add a third dimension into which layout can go
 - Compromise of top-down and centered techniques mentioned earlier
 - Children of a node are laid out in a cylinder “below” the parent
 - Siblings live in one of the 2D planes

Robertson, Mackinlay, Card CHI '91

24

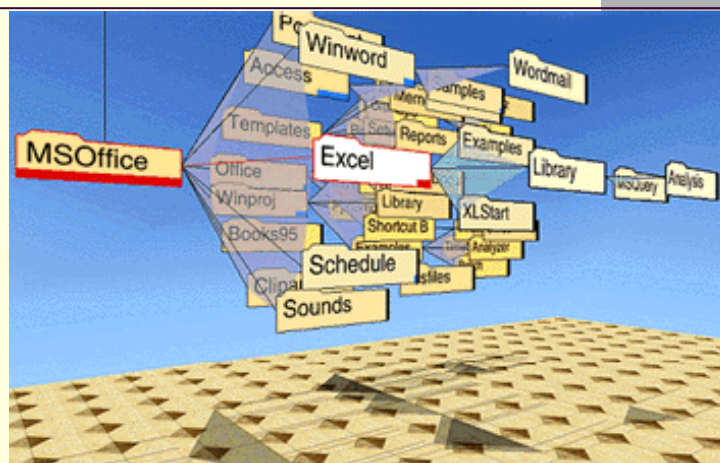
Cone Tree



Robertson, Mackinlay, Card CHI '91

25

Alternative Views



Robertson, Mackinlay, Card CHI '91

26

Advantages vs. Limitations

- 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

J. Stasko's InfoVis class slides

27

Hyperbolic Brower

- Key idea:
 - Find a space (hyperbolic space) that increases exponentially, lay out the tree on it
 - Transform from the hyperbolic space to 2D Euclidean space

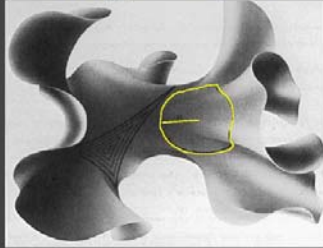
J. Lamping and R. Rao, "The Hyperbolic Brower: A Focus + Context Technique for Visualizing Large Hierarchies", *Journal of Visual Languages and Computing*, vol. 7, no. 1, 1995, pp. 33-55.

28

Hyperbolic space background

geometry with exponential "amount of room"
· good match for exponential node count of trees

2D hyperbolic plane



[Thurston and Weeks 84]

hemisphere area

hyperbolic: **exponential**
 $2\pi \sinh^2(r)$

euclidean: **polynomial**
 $2\pi r^2$

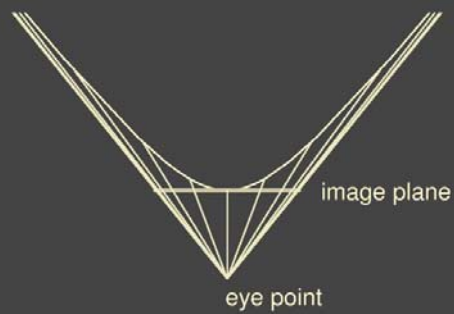
31

<http://graphics.stanford.edu/~munzner/talks/calgary02>

29

1D hyperbolic space

hyperbola projects to line

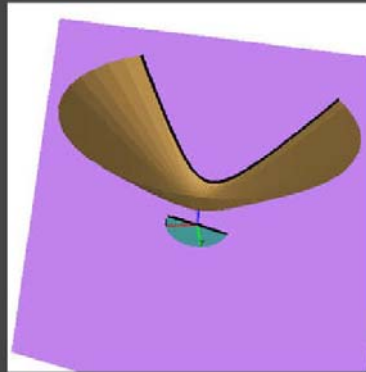
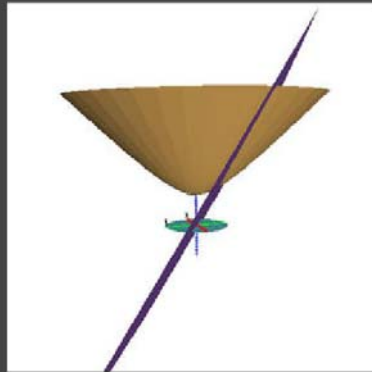


32

30

2D hyperbolic space

hyperboloid projects to disk

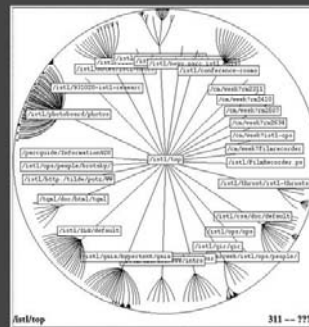
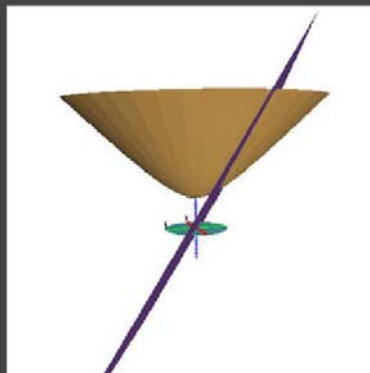


33

31

2D hyperbolic space

hyperboloid projects to disk

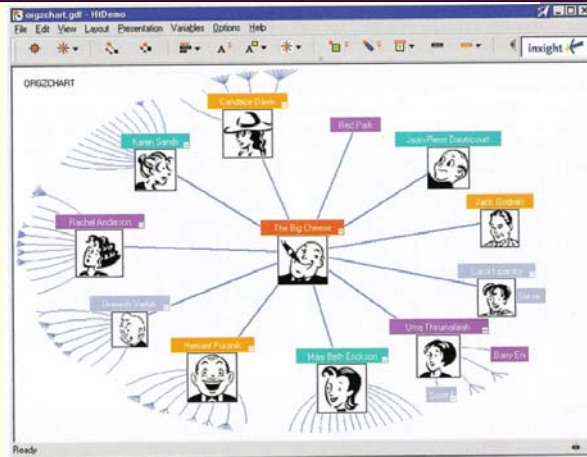


[Lamping et al 95]

34

32

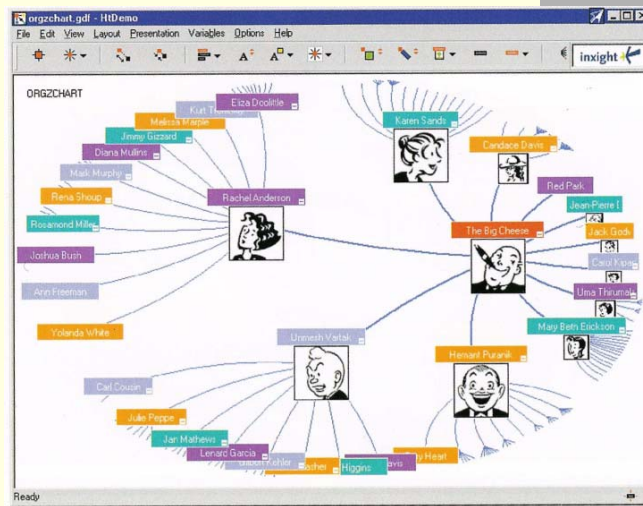
Hyperbolic Brower



R. Spence. Information Visualization

33

Change Focus



34

Key Attributes

- Natural magnification (fisheye) in center
- Layout depends only on 2-3 generations from current node
- Smooth animation for change in focus
- Don't draw objects when far enough from root (simplify rendering)

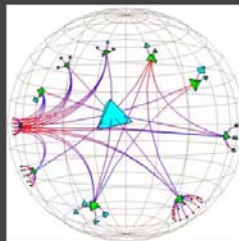
3D hyperbolic space

3-hyperboloid projects to solid ball



webviz [Munzner and Burchard 95]

- straightforward cone tree + 3D hyperbolic space
- poor information density



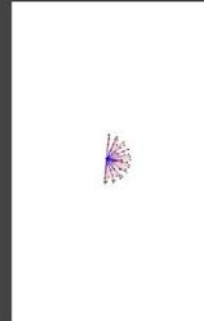
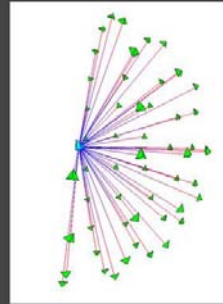
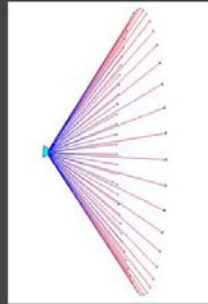
3D hyperbolic space

3-hyperboloid projects to solid ball



H3 layout

· circumference \rightarrow hemisphere

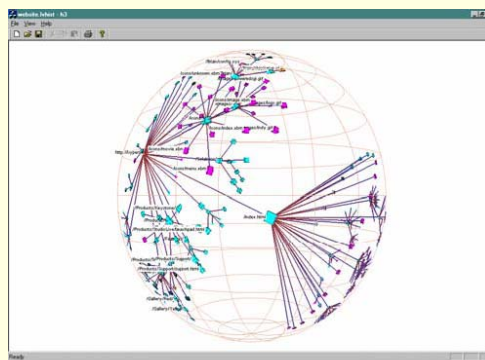


38

37

H3 Browser

- Use hyperbolic transformation in 3D space

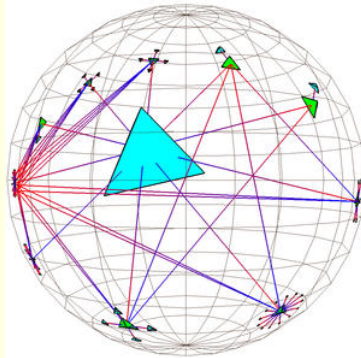


Demo

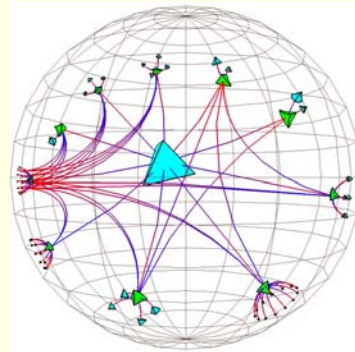
Tamara Munzner: H3: laying out large directed graphs in 3D hyperbolic space. [INFOVIS 1997](#): 2-10

38

Scalability - Model Selection



Projective model: keeps lines straight but distorts angles.



Conformal model: preserves angles but maps straight lines to circular arcs.

From Tamara Munzner's Ph.D. dissertation

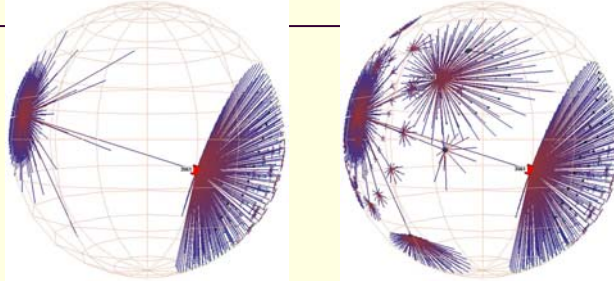
39

Projective vs. Conformal Model

- Projective model
 - Less aesthetically pleasing ☹
 - Transformation: 4X4 matrices ☺
 - Straight lines ☺
- Conformal model
 - More aesthetically pleasing ☺
 - Transformation: 2X2 complex matrices ☹
 - Curves ☹

40

Scalability - Adaptive Drawing



- Maintain a target frame rate
 - Draw only as much of the neighborhood around a center point as is possible in the allotted time
 - Fill in scene fringe using several bounded idle frames when the user is idle
- Question: Why use several idle frames?

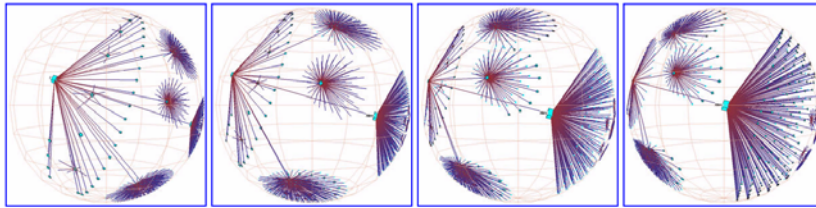
41
Movie 3

Scalability - Other Tricks

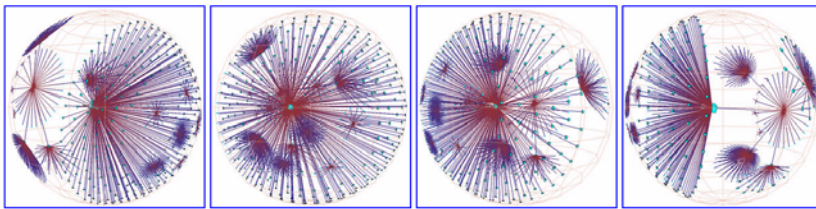
- Only draw a local neighborhood of nodes
 - Nodes sufficiently far from the center will project to less than a single pixel – terminate drawing when features project to subpixel areas
- Use front buffer for highlighting

42

Navigation



Translation of a node to the center



Movie0
Rotation around the same node
From Tamara Munzner's Ph.D. dissertation

43

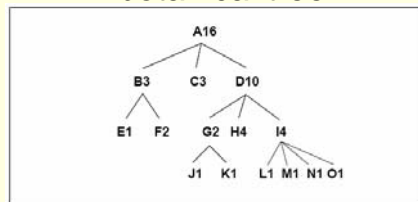
Problems

- Orientation
 - Watching the view can be disorienting
 - When a node is moved, its children don't keep their relative orientation to it as in Euclidean plane, they rotate
- Not as symmetric and regular as Euclidean techniques, two important attributes in aesthetics

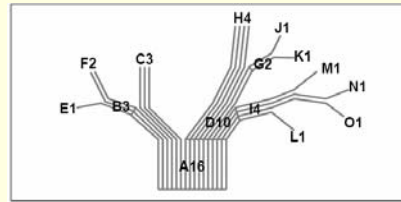
44

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Basic idea: we can easily see the branches, leaves, and their arrangement in a botanical tree
- Inspiration: Strand model of Holton
 - Strands: internal vascular structure of a botanical tree



Node and link diagram

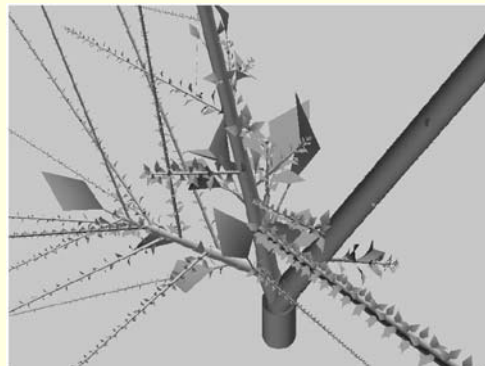


Corresponding strand Model

45

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Use strand model to create a 3-d directory tree:

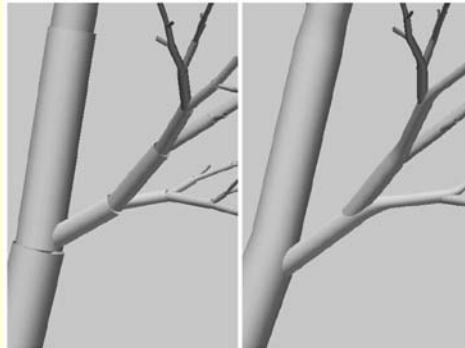


Unsatisfied features: 1. Branching points 2. long and thin branches 3. cluttered leaves

46

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Improve the first tree:

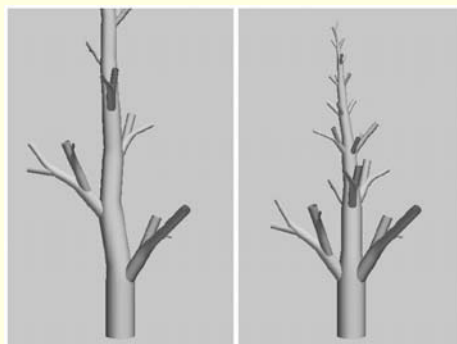


Adding smooth transition between two cylinders

47

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Improve the first tree:

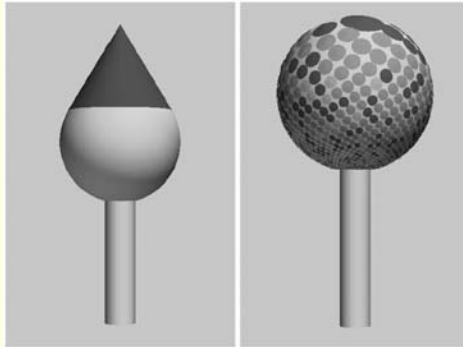


Use a general tree rather than a binary tree

48

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Improve the first tree:

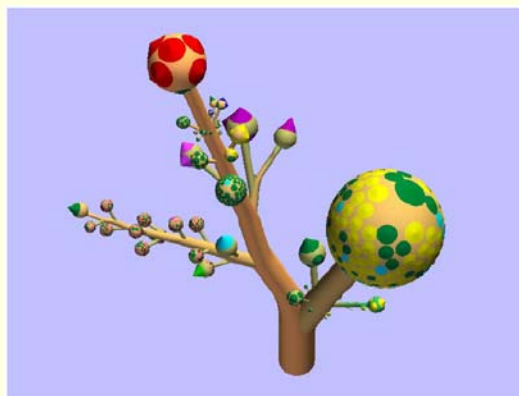


Phi-ball with one (left) and many (right) files

49

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Botanical tree:

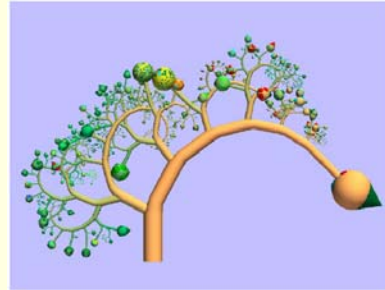
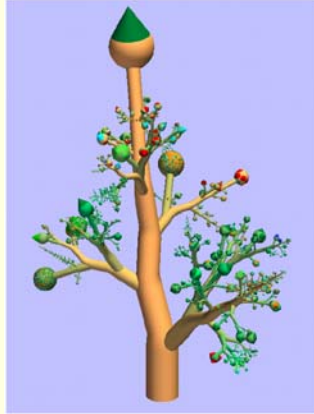


Final model with the improvements

50

Botanical Tree [E. Kleiberg et. al. InfoVis 2001]

- Botanical tree:



The same directory with different settings

51

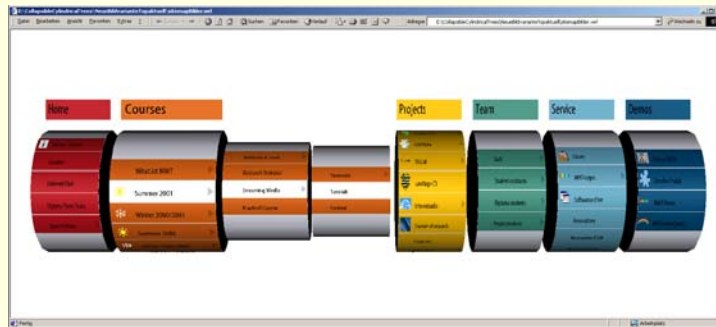
Dynamic Visualization of Graphs with Extended Labels [wong:infovis2006]

- Labeling issue in the node-link diagram
- Video

52

Collapsible Cylindrical Tree [Dachselt & Ebert Infovis 01]

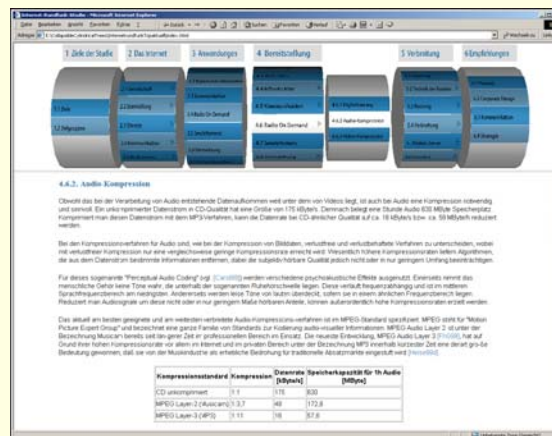
- Basic idea: use a set of nested cylinders according to the telescope metaphor
- Limitation: one path is visible in once
- Interactions: rotation, go down/up



53

Collapsible Cylindrical Tree [R. Dachselt, J. Ebert Infovis 01]

- Example application: web document browsing



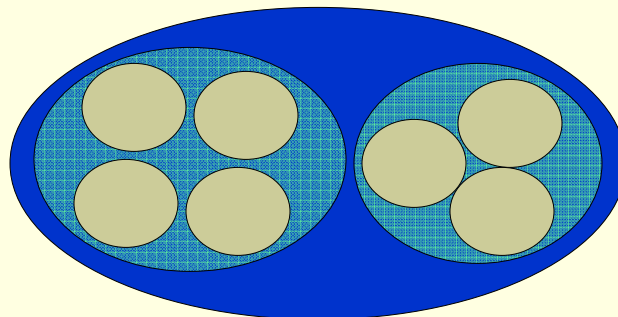
54

Space-Filling Techniques

55

Space-Filling Techniques

- Each item occupies an area
- Children are “contained” within parent



56

Treemap

- Children are drawn inside their parents
- Alternative horizontal and vertical slicing at each successive level
- Use area to encode other variables of data items

[B. Johnson](#), Ben Shneiderman: Tree maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. [IEEE Visualization 1991](#): 284-291

57

Treemap

- Example

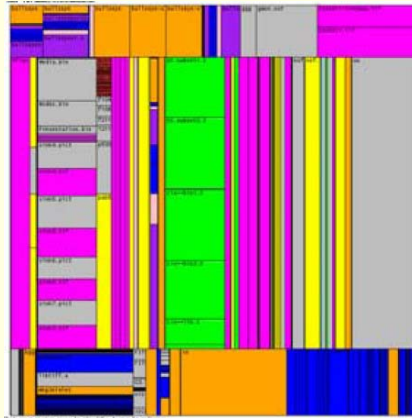


J. Stasko's InfoVis class slides

58

Treemap

■ Example



J. Stasko's InfoVis class slides

59

Treemap Algorithm

```
Draw()
{
    Change orientation from parent (horiz/vert)
    Read all files and directories at this level
    Make rectangle for each, scaled to size
    Draw rectangles using appropriate size and color
    For each directory
        Make recursive call using its rectangle as focus
}
```

J. Stasko's InfoVis class slides

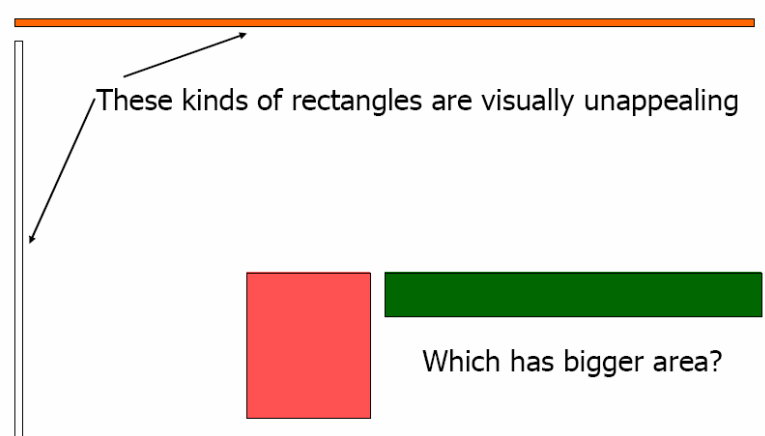
60

Treemap Affordances

- It is rectangular!
- Good representation of two attributes beyond node-link: color and area
- Not as good at representing structure
 - Can get long-thin aspect ratios
 - What happens if it's a perfectly balanced tree of items all the same size?

61

Aspect ratios



J. Stasko's InfoVis class slides

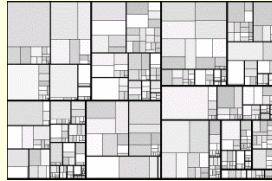
62

Treemap Variation

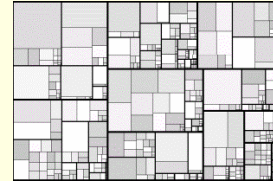
■ Make rectangles more square



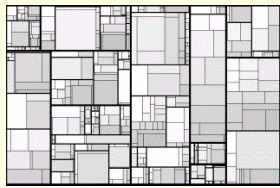
Slice-and-dice



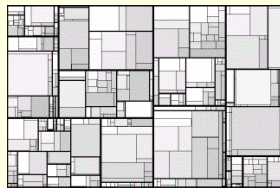
Cluster



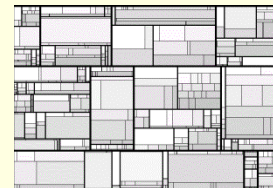
Squarified



Pivot-by-middle



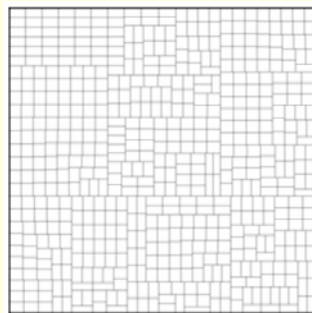
Pivot-by-size



Strip

63

Showing Structure



A tree with 698 node (from [Balzer:infovis2005])

How about a perfectly balanced binary tree?

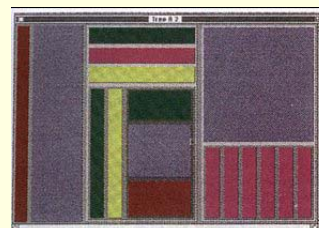
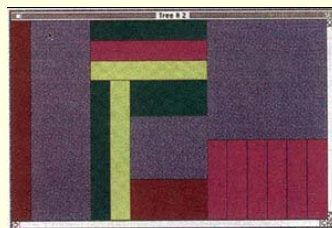
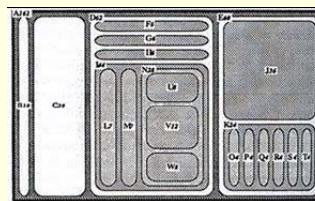
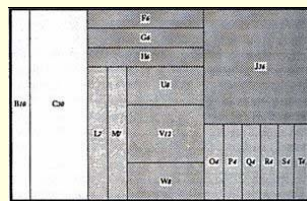
64

Showing Structure

- Borderless treemap: hard to discern structure of hierarchy
 - What happens if it's a perfectly balanced tree of items all the same size?
- Variations:
 - Use border
 - Change rectangles to other forms

65

Nested vs. Non-nested



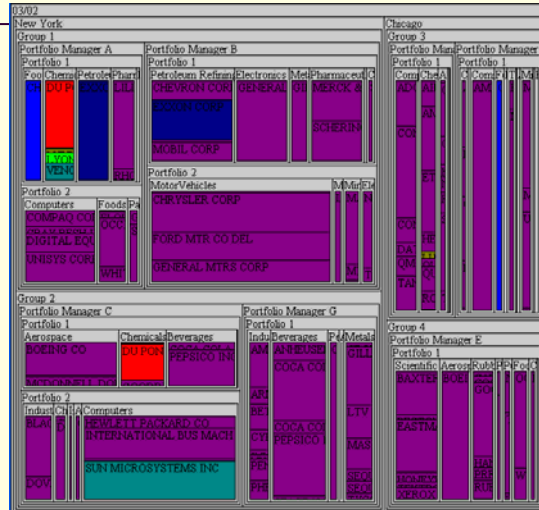
Non-nested Treemap

Nested Treemap

66

Nested Treemap

- Borders help on smaller trees, but take up too much area on large, deep ones

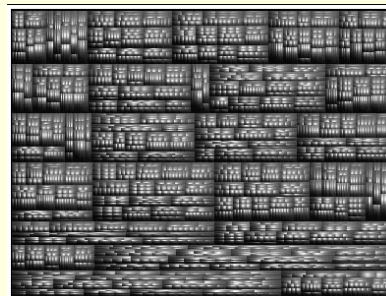
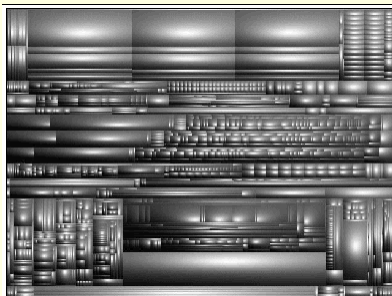


<http://www.cs.umd.edu/hcil/treemap-history/treemap97.shtml>

67

Cushion Treemap

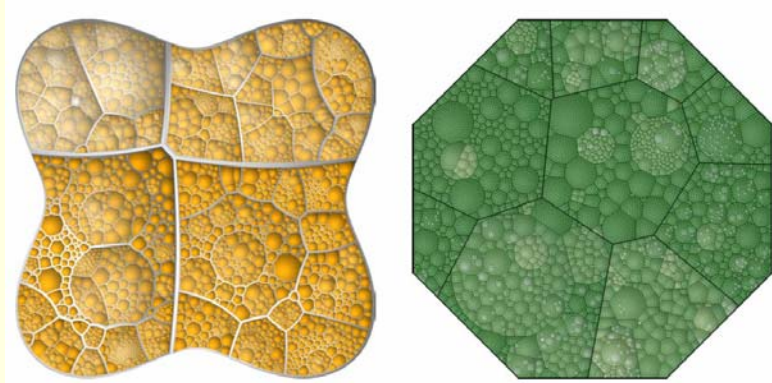
- Add shading and texture (Van Wijk and Van de Wetering InfoVis'99)



68

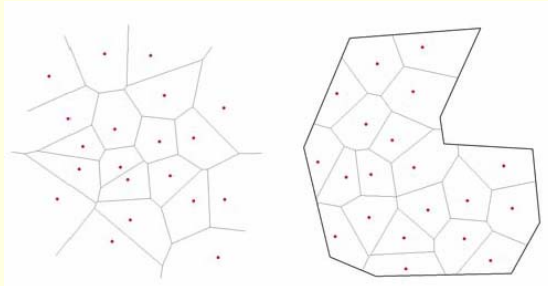
Voronoi Treemaps [balzer:infovis05]

- Enable subdivisions of and in polygons
- Fit into areas of arbitrary shape



Basic Voronoi Tessellations

- Enable partitioning of m-dimensional space without holes or overlappings
- Planar VT in 2D:
 - $P = \{p_1, \dots, p_n\}$ a set of n distinct points –generators
 - Divide 2D space into n Voronoi regions $V(P_i)$:
 - Any point q lies in the region $V(P_i)$ if and only if
 - $\text{distance}(p_i, q) < \text{distance}(p_j, q)$ for any $j \neq i$



70

Weighted Voronoi Tessellations

- Basic VT: $distance_{\varepsilon}(p_i, q) := \|p_i - q\| = \sqrt{(x_i - x)^2 + (y_i - y)^2}$

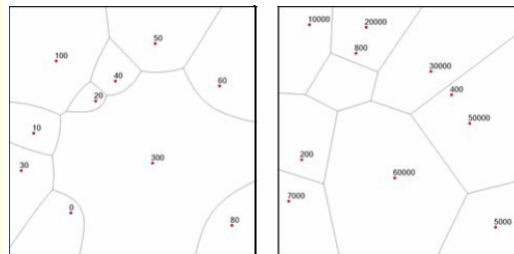
- Additively weighted Voronoi (AW VT):

$$distance_{aw}(p_i, w_i, q) := \|p_i - q\| - w_i$$

- Additively weighted power voronoi (PW VT):

$$distance_{pw}(p_i, w_i, q) := \|p_i - q\|^2 - w_i$$

Left: AW VT
Right: PW VT



71

Centroidal Voronoi Tessellations (CVT)

- Property of CVT: Each generator is itself center of mass(centroid) of corresponding voronoi region

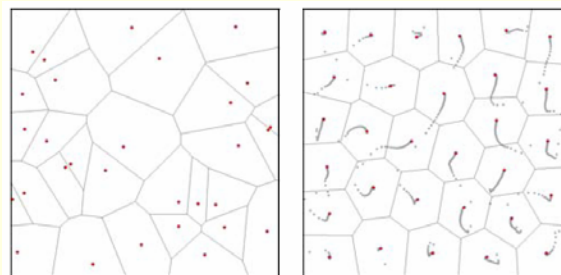


Figure 6: Voronoi tessellation of 20 random points and an associated CVT—traces illustrate the movements of the points during the computation of the CVT

72

Centroidal Voronoi Tessellations (CVT)

- CVT minimize the energy function:

$$\mathcal{K}(P, \mathcal{V}(P)) = \sum_i \int_{V(p_i)} \|x - p_i\|^2 dx$$

- The energy of the CVT is equivalent to the overall aspect ratio of the subareas of the treemap layout

73

Voronoi Treemap Algorithm

- Size of each Voronoi region should reflect size of the tree node
- Area size is not observed in CVT computation
- Extension:
 - Use iteration
 - In each iteration, adjust the area of regions by their weights
 - Weights are adjusted according to the size of the node
 - Iterate until the relative size error is under a threshold
- Video

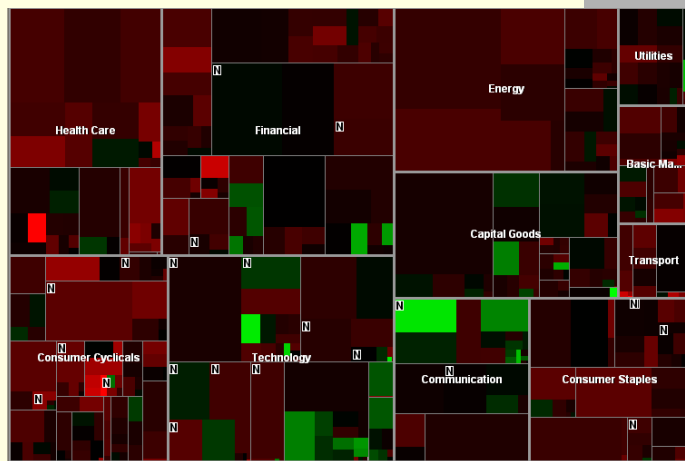
74

Treemap Applications

- Software visualization
- Multimedia visualization
- Tennis matches
- File/directory structures
- Basketball statistics
- Stocks and portfolios

75

Marketmap



<http://www.smartmoney.com/marketmap/>

76

Software Visualization

■ SeeSys (Baker & Eick, AT&T Bell Labs)

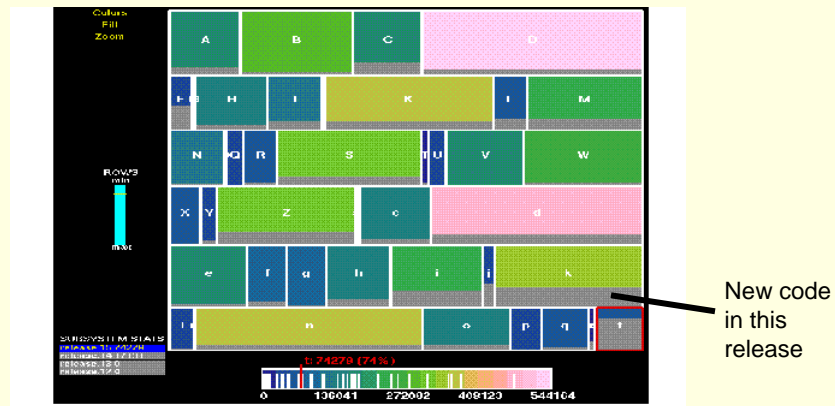
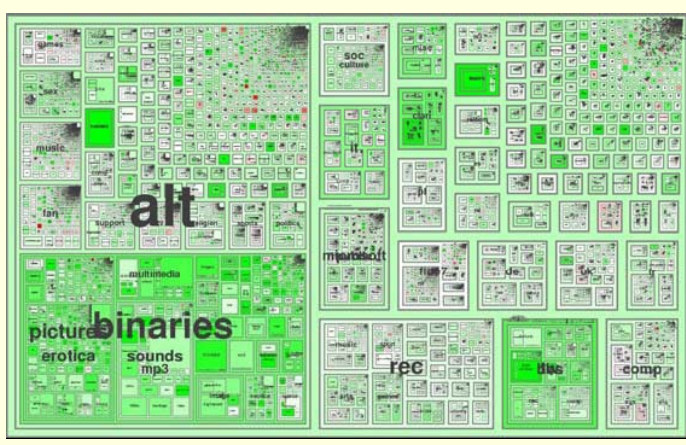


Figure 2: NCSL and new development by subsystem in a recent release.

77

Internet News Groups

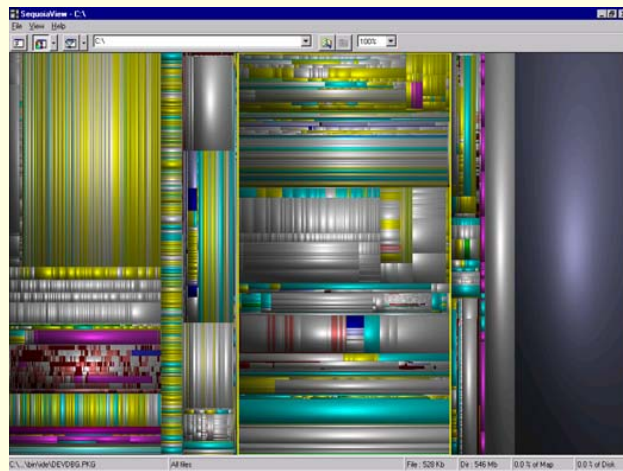
■ Netscan (Fiore & Smith Microsoft)



78

SequoiaView

- File visualizer www.win.tue.nl/sequoiaview/



79

Photemesa

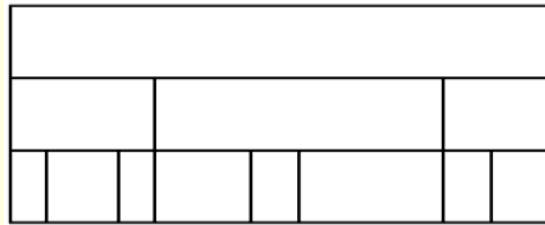
- Image browser (quantum and bubble treemap)
<http://www.cs.umd.edu/hcil/photomesa/>



80

Space-Filling Techniques

- Each item occupies an area
- Children are “contained” within (under) parent

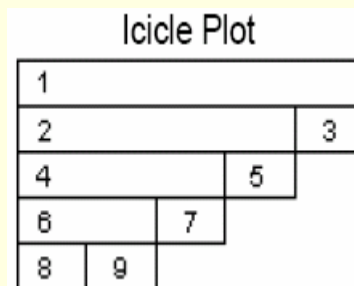
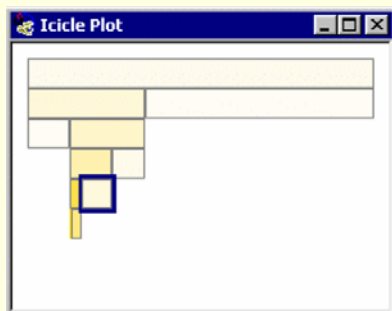


One Example

81

Icicle Plot

- Icicle plot (similar to Kleiner and Hartigan’s concept of castles)
 - Node size is proportional to node width

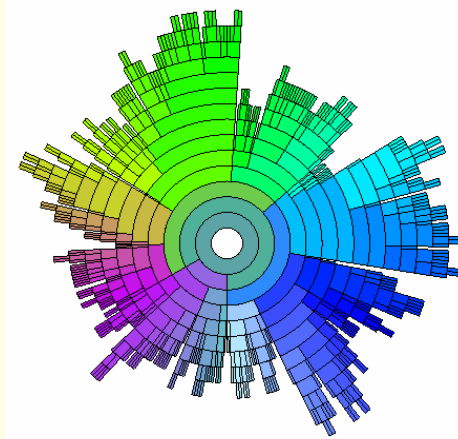


Barlow and Neville InfoVis 2001

82

Radial Space Filing Techniques

- InterRing [Yang02]



83

Node Link + Space Filling Techniques



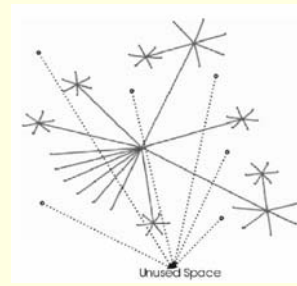
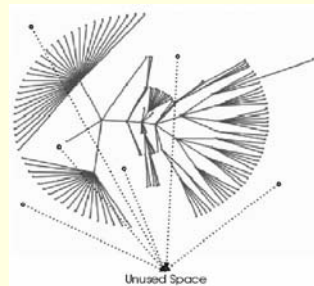
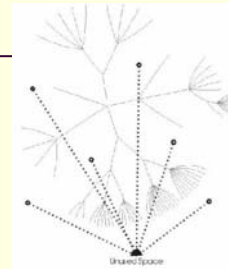
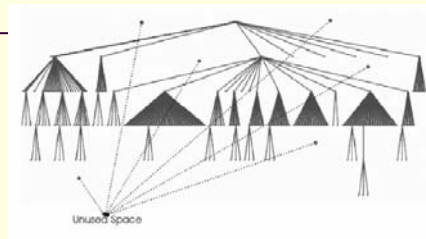
84

Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams [zhao:infovis 05]

- A hybrid approach
- Dynamic
- Video

85

Space-Optimized Tree - Motivation



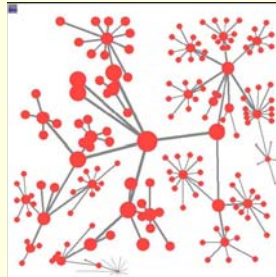
86

Q. Nguyen and M. Huang Infovis 02

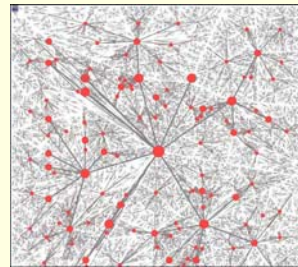
Space-Optimized Tree [Q. Nguyen and M. Huang Infovis 02]

Key idea:

- Partition display space into a collection of geometrical areas for all nodes
- Use node-link diagrams to show relational structure



Example: Tree with 150 nodes



Example: Tree with approximately 55000 nodes

Space-Optimized Tree [Q. Nguyen and M. Huang Infovis 02]

Algorithm for dividing a region:

1. weight calculation for each direct child
2. wedge calculation for each direct child
3. vertex position calculation for each direct child

Weight Calculation

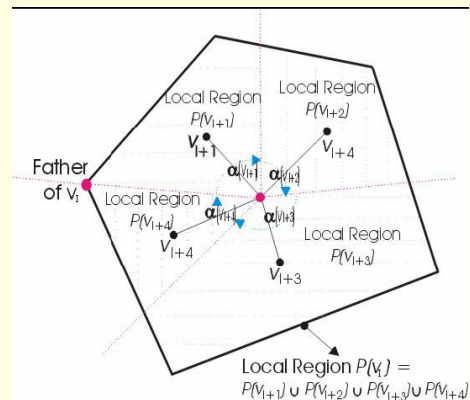
$$w(v_i) = 1 + C \sum_{j=0}^{k-1} w(v_{l+j})$$

- V_i : the direct child
- $V_l - V_{l+k}$: Direct children of V_i
- Constant C : decide difference between vertexes with more descendants and vertexes with fewer descendants.

89

Wedge Calculation

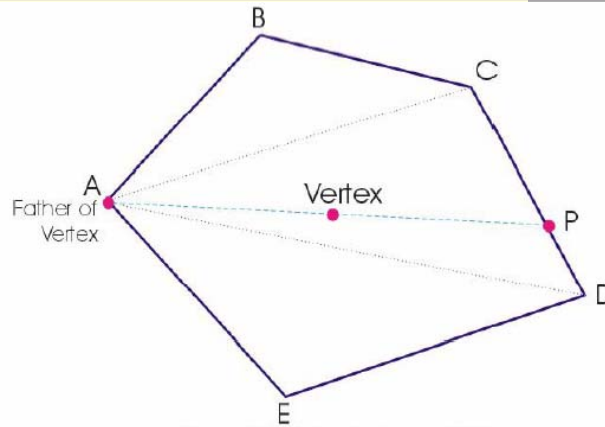
$$\alpha(v_{l+m}) = A \frac{w(v_{l+m})}{\sum_{j=0}^k w(v_{l+j})}$$



Example of dividing the local region of one node

90

Vertex Position Calculation



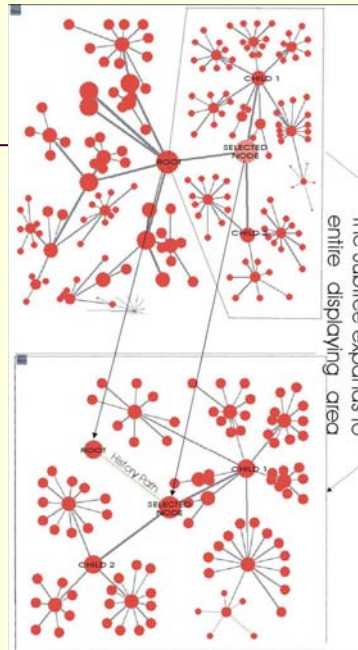
Area ABCP = Area AEDP

Vertex is the midpoint of line AP

91

Space-Optimized Tree

- Zooming by mouse clicking
- Animation during zooming



92

Reading Assignment

- Chapter 2 of the Spence book