Hierarchy and Tree Visualization

Fall 2009

Jing Yang

1

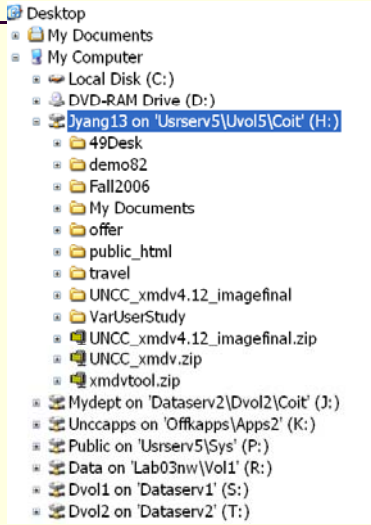
Hierarchies

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

2

Hierarchies in the World

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



3

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

4

Trees

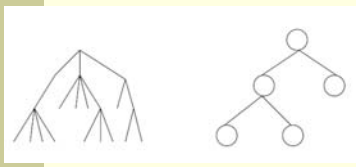
- Hierarchies are often represented as trees
  - Directed, acyclic graph
- Two major categories of tree visualization techniques:
  - Node-link diagram
    - Visible graphical edge from parents to their children
  - Space-filling

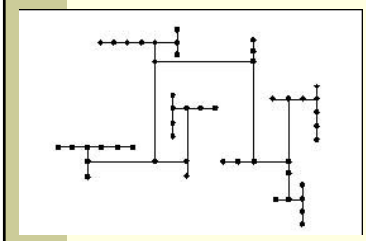
5

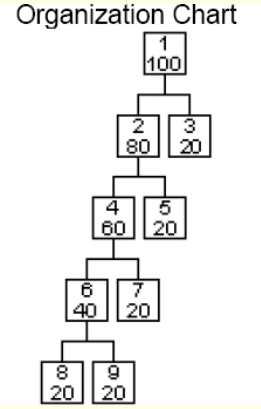
Node-Link Diagrams

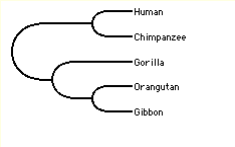
6

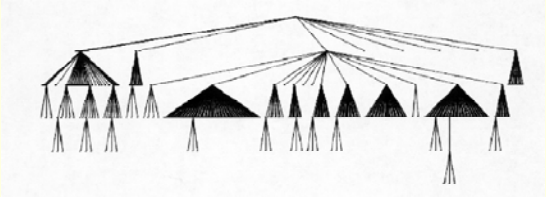
Put Root at Top or Left

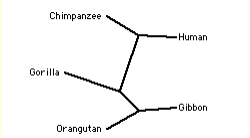




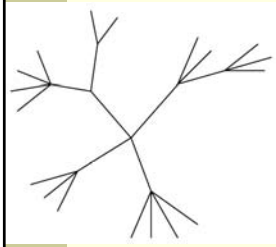


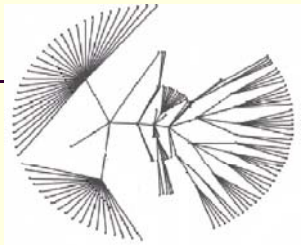


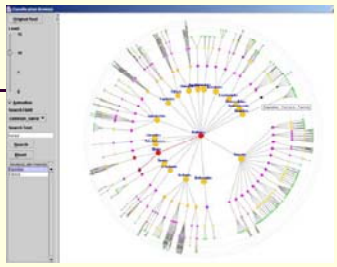




Put Root at Center







Radial View

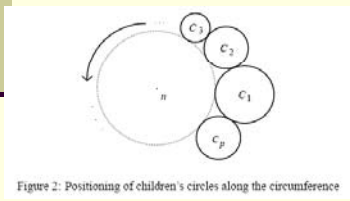
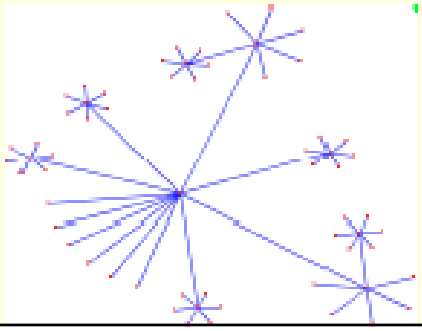


Figure 2: Positioning of children's circles along the circumference



Balloon View

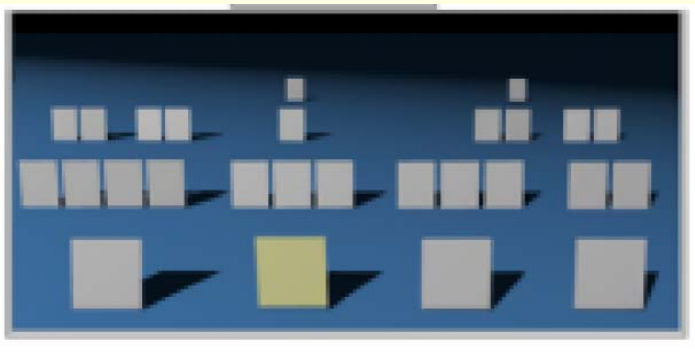
8

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

9

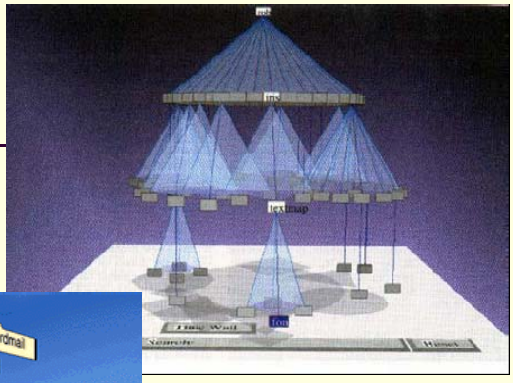
3D Approach 1 - 3D Tree

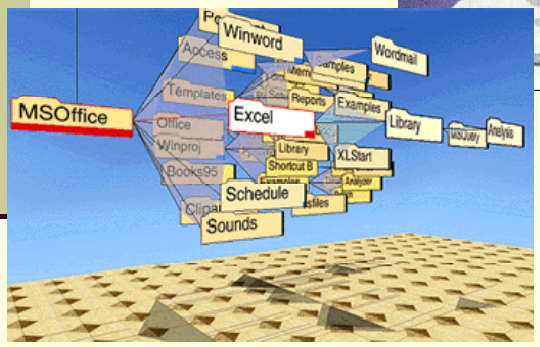


Tavanti and Lind, InfoVis 01

10

3D Approach 2 - Cone Tree





Robertson, Mackinlay, Card CHI '91

11

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

12

Hyperbolic Browser

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

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

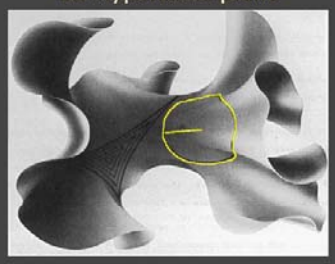
13

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

$$2\pi r^2$$

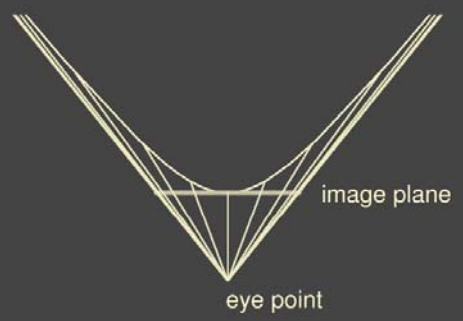
31

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

14

1D hyperbolic space

hyperbola projects to line

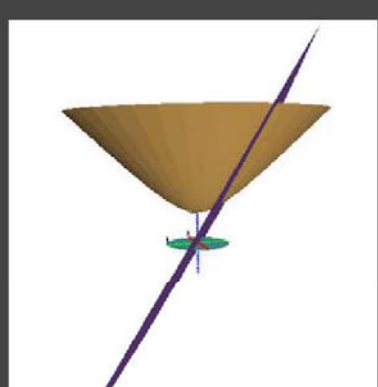


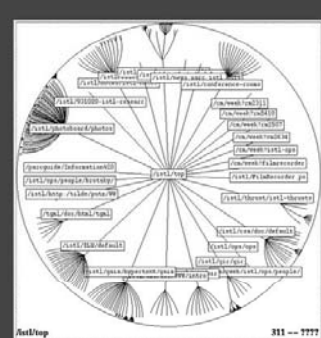
32

15

2D hyperbolic space

hyperboloid projects to disk



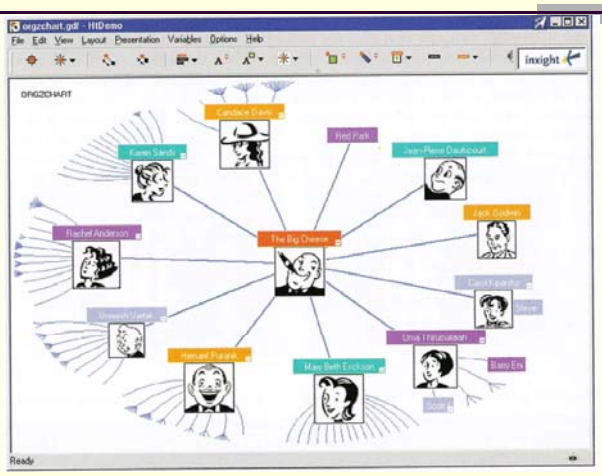


[Lamping et al 95]

34

16

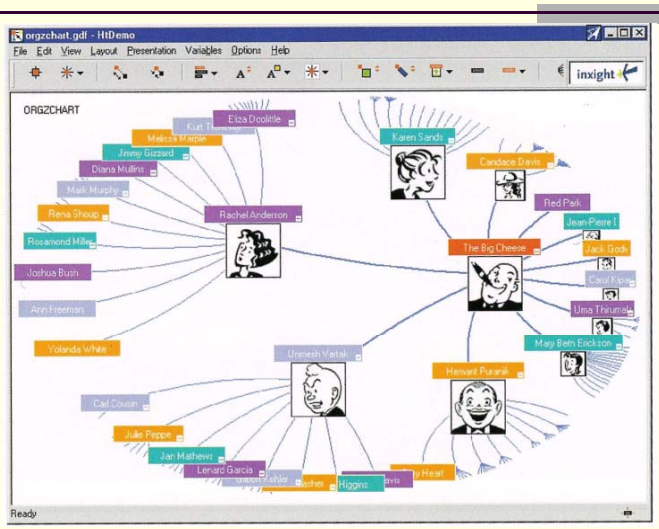
Hyperbolic Browser



R. Spence. Information Visualization

17

Change Focus



18

Key Attributes

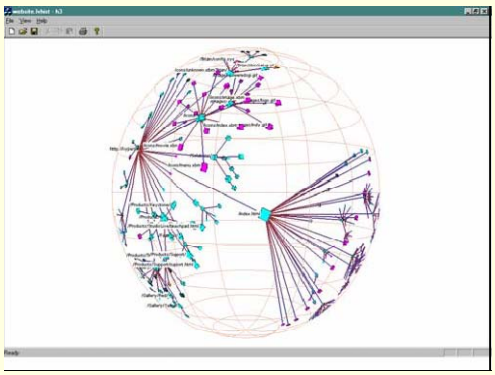
- Natural magnification (fisheye) in center
- Layout depends only on 2-3 generations from current node
- Smooth animation for change in focus
- Don't draw objects when far enough from root (simplify rendering)

J. Stasko's InfoVis class slides

19

H3 Browser

- Use hyperbolic transformation in 3D space

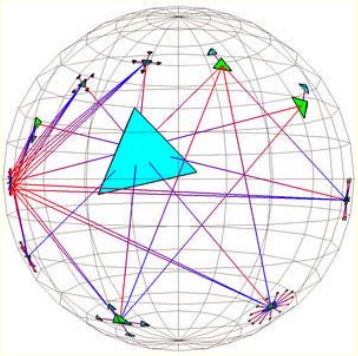


Demo: http://www-graphics.stanford.edu/videos/h3/

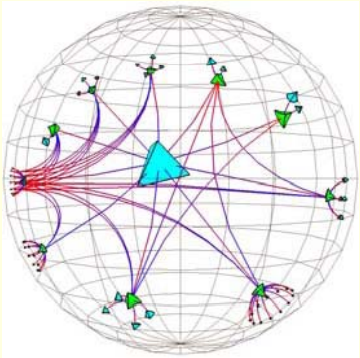
Tamara Munzner: H3: laying out large directed graphs in 3D hyperbolic space. INFOVIS 1997: 2-10

20

Scalability - Model Selection (1)



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

21

Scalability - Model Selection (1)

Projective vs. Conformal Model

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

22

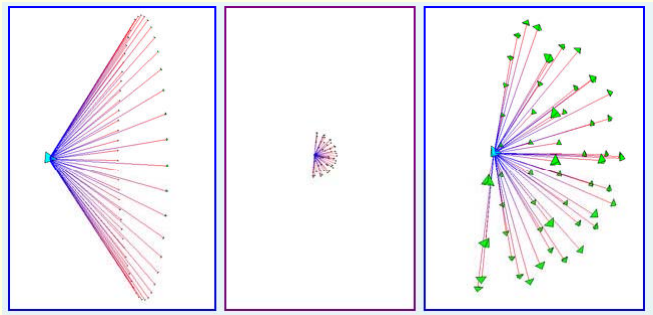
Scalability - Layout (1)

- Find a spanning tree from an input graph
  - Use domain-specific knowledge
- Layout algorithm
  - Nodes are laid out on the surface of hemispheres
  - A bottom-up pass to estimate the radius needed for each hemisphere
  - A top-down pass to place each child node on its parental hemisphere's surface

23

Scalability - Layout (2)

- Lays out child nodes on the surface of a hemisphere like sprinkles on an ice cream cone.

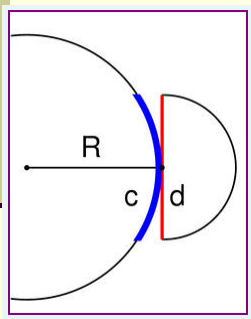


Left: cone tree approach Middle and right: H3 approach

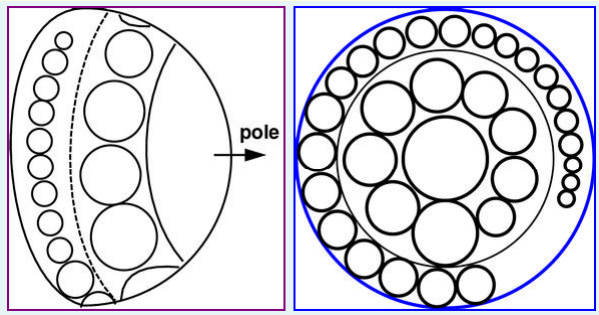
24

Scalability - Layout (3)

- An approximate layout is fine, whereas a perfect but slow iterative solution would be inappropriate



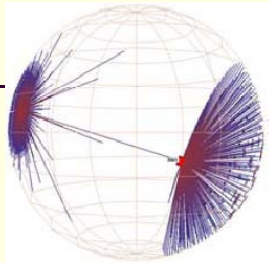
Bottom-up pass

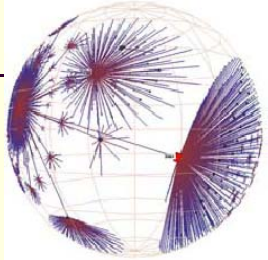


Top-down pass

25

Scalability - Adaptive Drawing (1)





- Maintain a target frame rate (movie 3)
  - Draw only as much of the neighborhood around a center point as is possible in the allotted time
    - Unterminated links
  - Fill in scene fringe using several bounded idle frames when the user is idle

26

Scalability - Adaptive Drawing (2)

Active mode - when the user is active dragging the mouse, or during animated transitions.

- **Initialization:** Initialize ActionQueue with a single node with the largest projected screen area in the previous frame (it is close to the ball's center)
- **Draw Loop**
  - Current node is popped off the ActionQueue.
  - Handle the current node.
  - Handle the nodes one hop away from the current one in the spanning tree: the parent node and the children nodes.
  - If the projected screen area of any of these neighboring nodes is at least one pixel, insert it into the ActionQueue, maintaining sorted order.
  - Terminate if: No more time left, or ActionQueue is empty.

27

Scalability - Adaptive Drawing (3)

Idle mode - when the user stops dragging the mouse, or an animated transition ends

- **Initialization:** ActionQueue from the previous frame is left untouched.
- **Draw Loop Termination:** Terminate if either: No more time left, or ActionQueue is empty.
- Several idle frames can be drawn back to back if no input is found
  - Why?

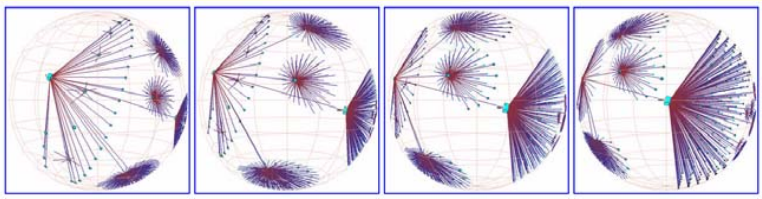
28

Scalability - Other Tricks

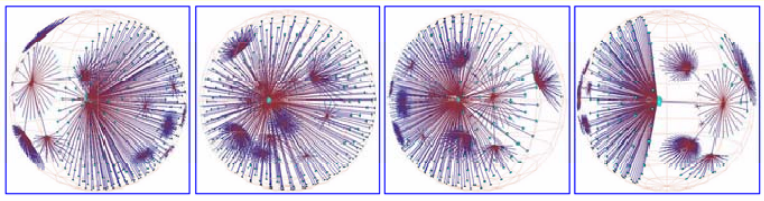
- Only draw a local neighborhood of nodes
  - Nodes sufficiently far from the center will project to less than a single pixel – terminate drawing when features project to subpixel areas
- Use front buffer for highlighting

29

Navigation



Translation of a node to the center

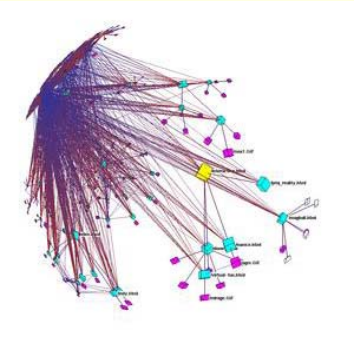


Rotation around the same node (Movie 0)

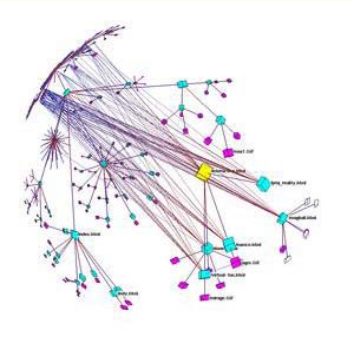
From Tamara Munzner's Ph.D. dissertation

30

Non-Tree Links



Drawing all the non-tree links



Drawing the outgoing non-tree links for the entire subtree beneath the highlighted yellow node

From Tamara Munzner's Ph.D. dissertation

Movie1

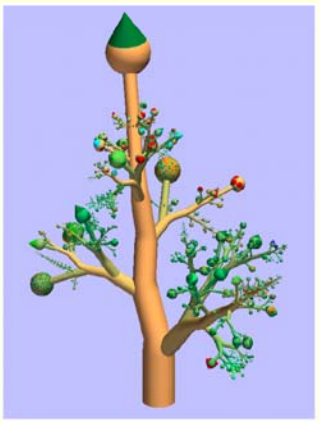
Problems

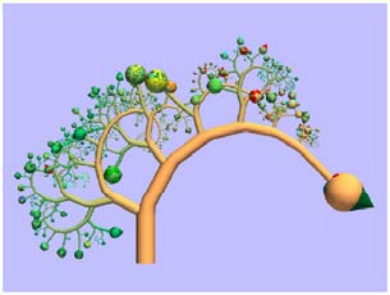
- Orientation
  - Watching the view can be disorienting
  - When a node is moved, its children don't keep their relative orientation to it as in Euclidean plane. They rotate
- Not as symmetric and regular as Euclidean techniques, two important attributes in aesthetics

J. Stasko's InfoVis class slides

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

- Botanical tree:



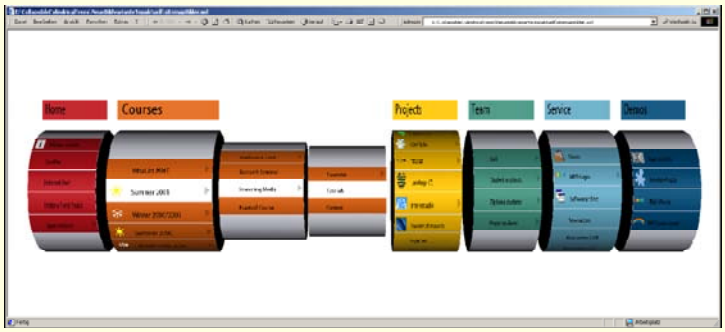


The same directory with different settings

33

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



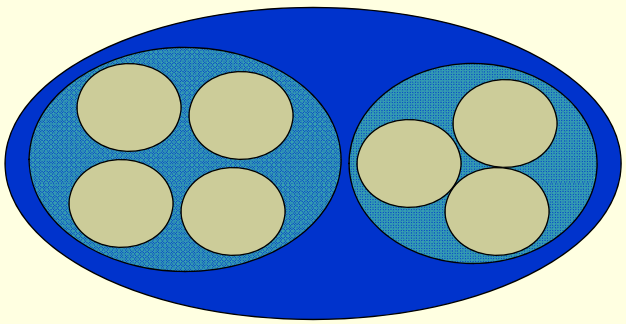
34

Space-Filling Techniques

35

Space-Filling Techniques

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



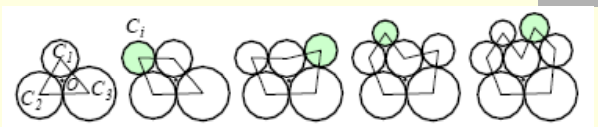
36

Visualization of Large Hierarchical Data by Circle Packing W.Wang et al. CHI 2006

- Key ideas:
  - tree visualization using nested circles
  - brother nodes represented by externally tangent circles
  - nodes at different levels displayed by using 2D nested circles or 3D nested cylinders

37

Visualization of Large Hierarchical Data by Circle Packing W.Wang et al. CHI 2006



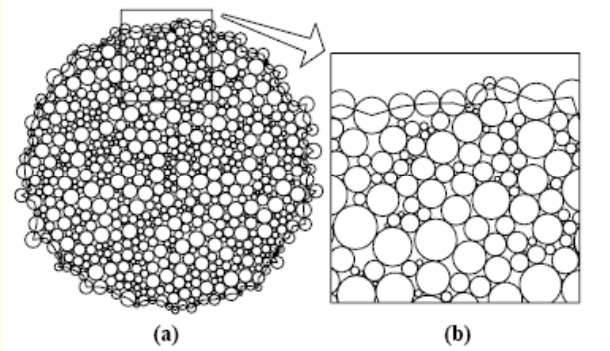
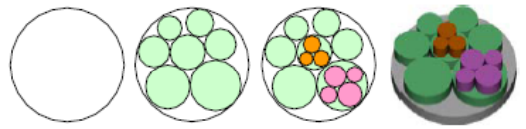


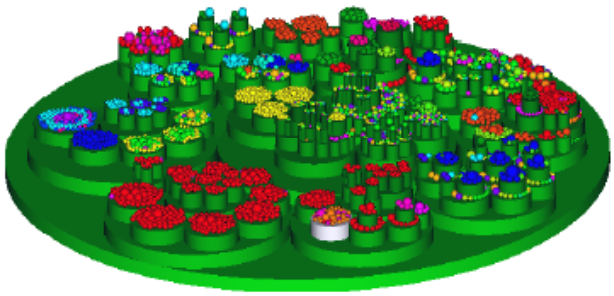
Figure 4. Packing 1000 circles with random radii

38

Visualization of Large Hierarchical Data by Circle Packing W.Wang et al. CHI 2006



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

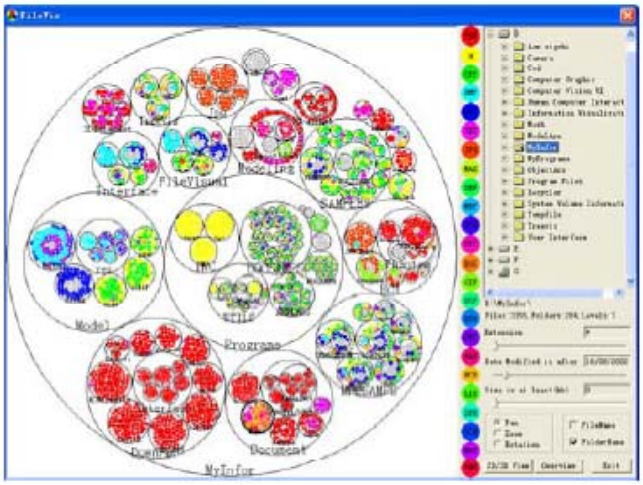


(c) 3D nested cylinders and spheres

Figure 6. The visualization of a file system

39

Visualization of Large Hierarchical Data by Circle Packing W.Wang et al. CHI 2006



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

40

Treemap

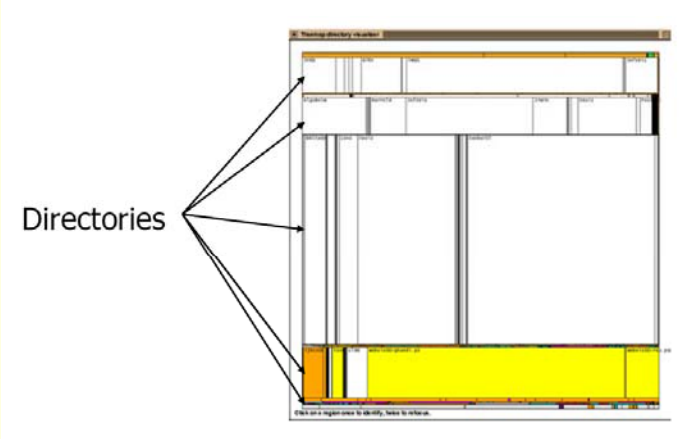
- Children are drawn inside their parents
- Alternative horizontal and vertical slicing at each successive level
- Use area to encode other variables of data items

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

41

Treemap

- Example

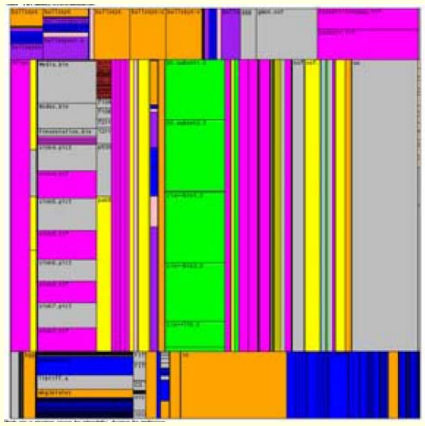


J. Stasko's InfoVis class slides

42

Treemap

- Example



J. Stasko's InfoVis class slides

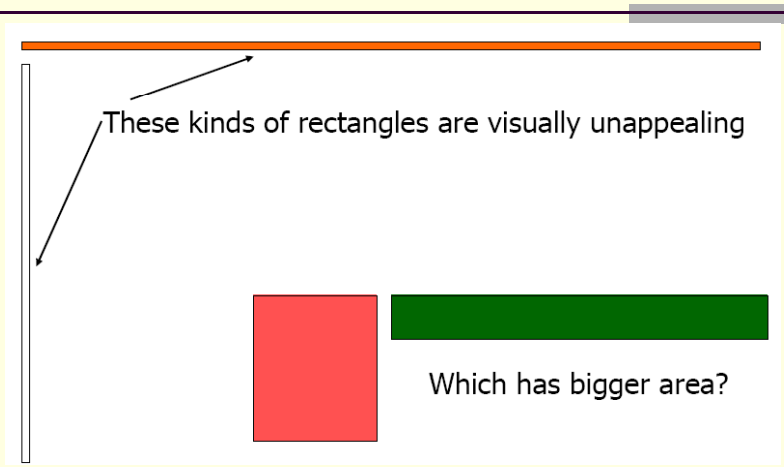
43

Treemap Affordances

- It is rectangular! It makes better use of space
- 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?

44

Aspect ratios



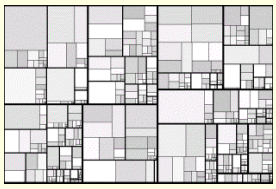
J. Stasko's InfoVis class slides

45

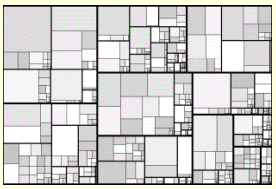
Treemap Variation

- Make rectangles more square

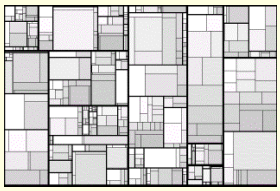
Slice-and-dice



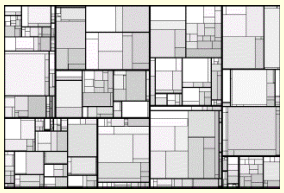
Cluster



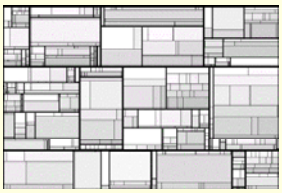
Squarified



Pivot-by-middle



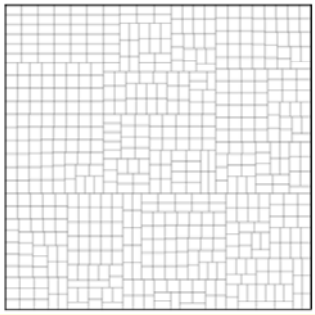
Pivot-by-size



Strip

46

Showing Structure



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

How about a perfectly balanced binary tree?

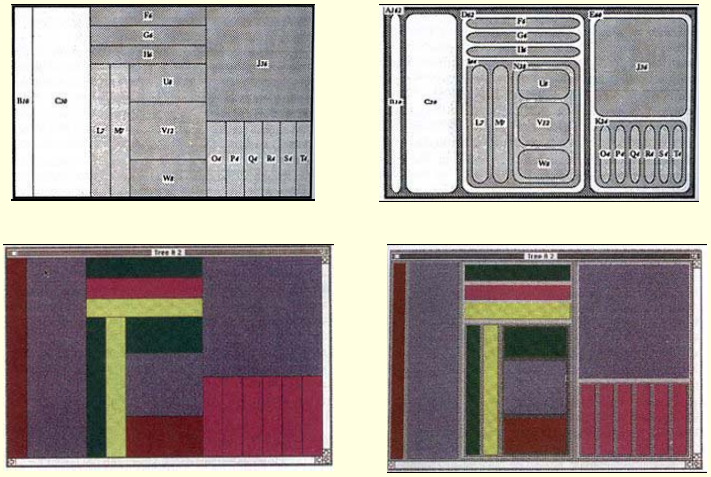
47

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

48

Nested vs. Non-nested

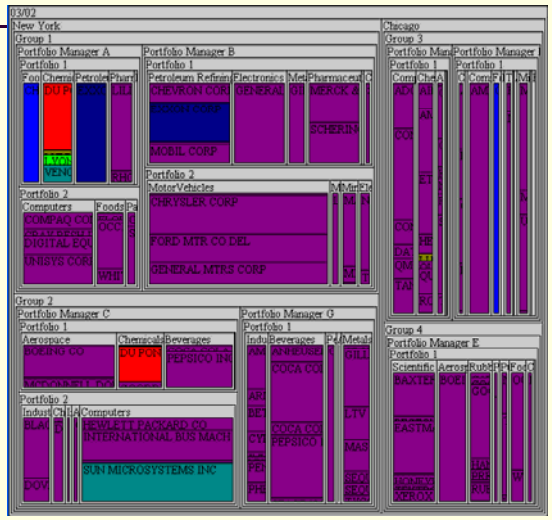


Non-nested Treemap Nested Treemap

49

Nested Treemap

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

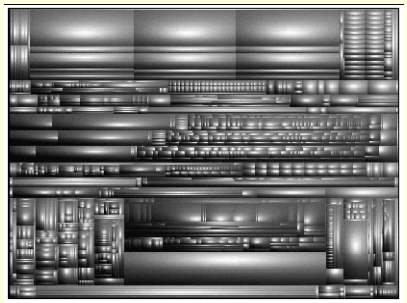


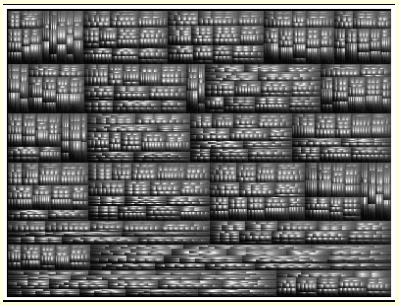
http://www.cs.umd.edu/hcil/treemap-history/treemap97.shtml

50

Cushion Treemap

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

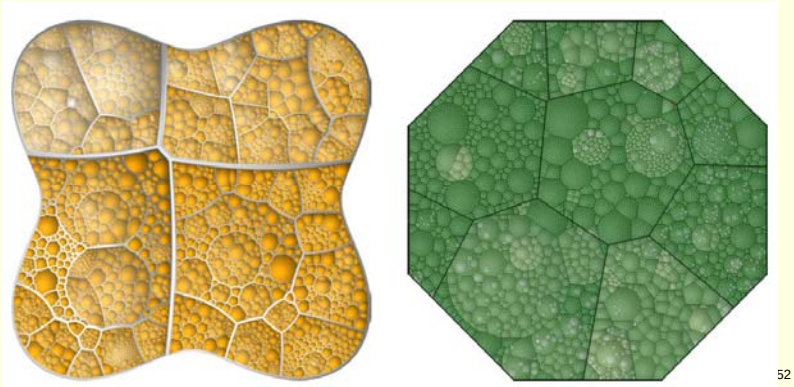




51

Voronoi Treemaps [balzer:infovis05]

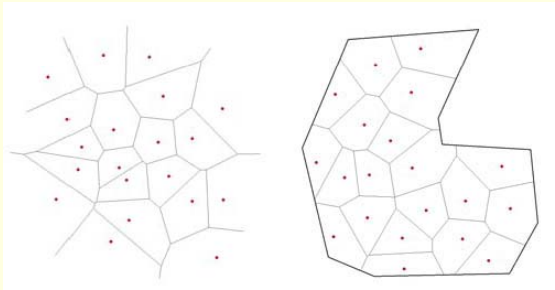
- Enable subdivisions of and in polygons
- Fit into areas of arbitrary shape



52

Basic Voronoi Tessellations

- Enable partitioning of m-dimensional space without holes or overlappings
- Planar VT in 2D:
  - P: = {p1, ..pn} a set of n distinct points –generators
  - Divide 2D space into n Voronoi regions V(Pi):
    - Any point q lies in the region V(Pi) if and only if
    - distance(pi, q) < distance(pj,q) for any j != i



53

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):

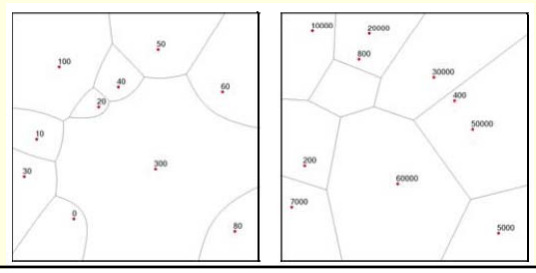
$$distance_{aw}(p_i, w_i, q) := \|p_i - q\| - w_i$$

- 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



54

Centroidal Voronoi Tessellations (CVT)

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

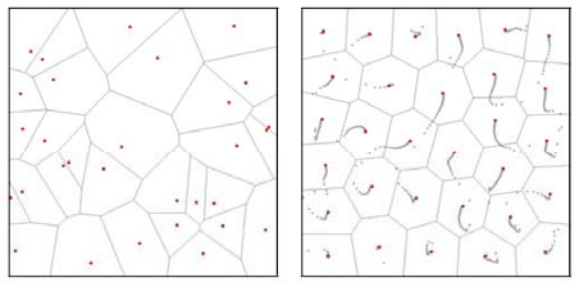


Figure 6: Voronoi tessellation of 20 random points and an associated CVT—traces illustrate the movements of the points during the computation of the CVT

55

Centroidal Voronoi Tessellations (CVT)

- CVT minimize the energy function:

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

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

56

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

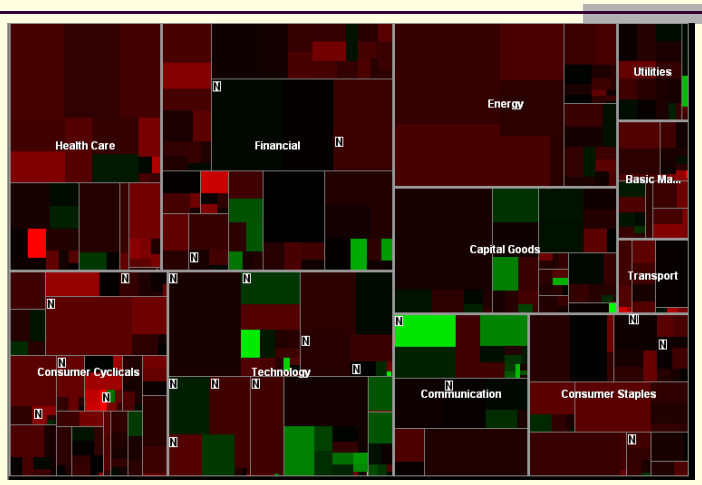
57

Treemap Applications

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

58

Marketmap

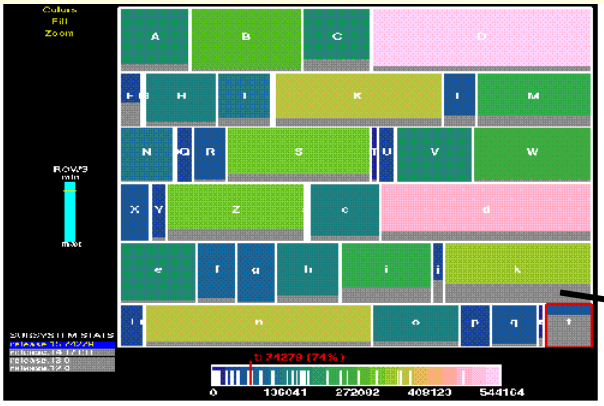


http://www.smartmoney.com/marketmap/

59

Software Visualization

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



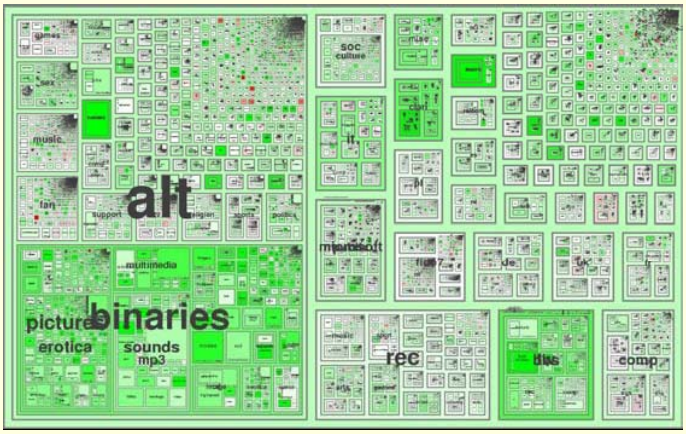
New code in this release

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

60

Internet News Groups

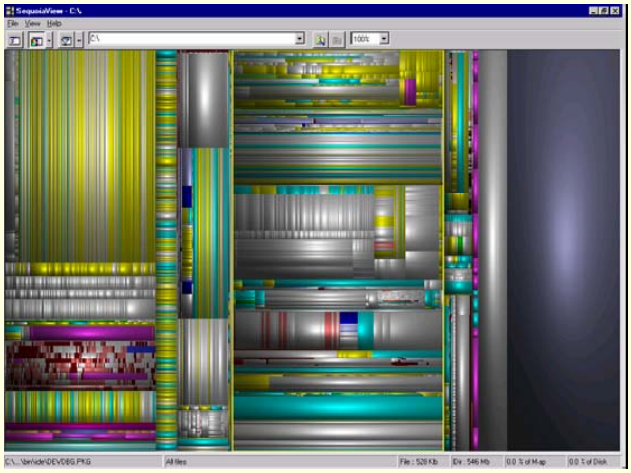
- Netscan (Fiore & Smith Microsoft)



61

SequoiaView

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



62

Photomesa

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

63

Space-Filling Techniques

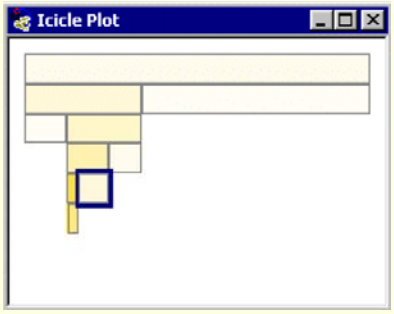
- Each item occupies an area
- Children are “contained” within (under) parent

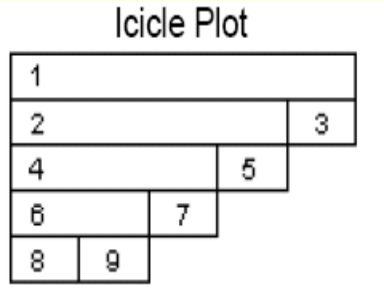
One Example

64

Icicle Plot

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



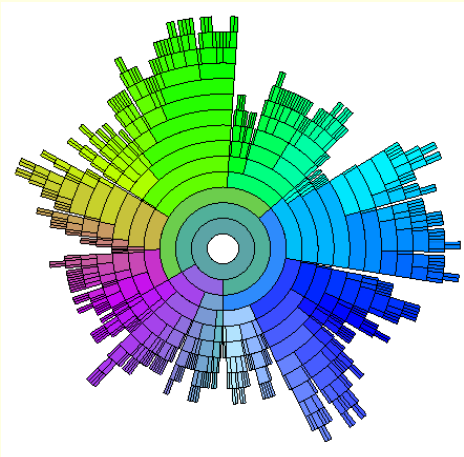


Barlow and Neville InfoVis 2001

65

Radial Space Filing Techniques

- InterRing [Yang02]



66

Node Link + Space Filling Techniques

67

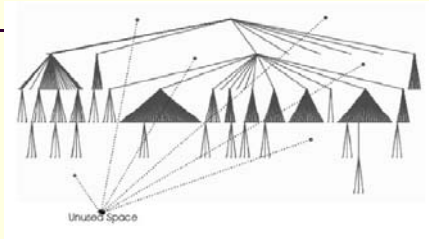
Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams [zhao:infovis 05]

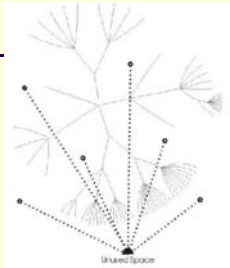
- A hybrid approach
- Dynamic

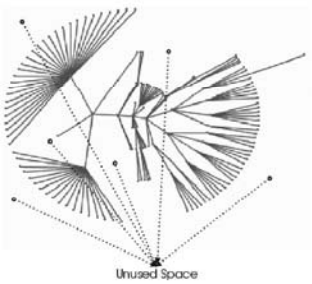
- Video

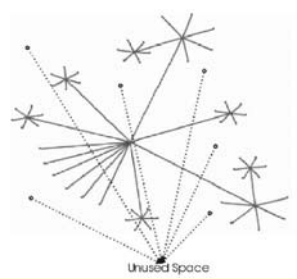
68

Space-Optimized Tree - Motivation









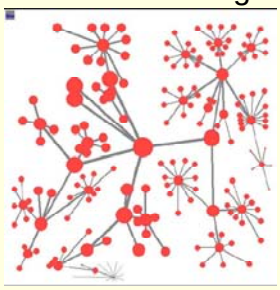
69

Q. Nguyen and M. Huang Infovis 02

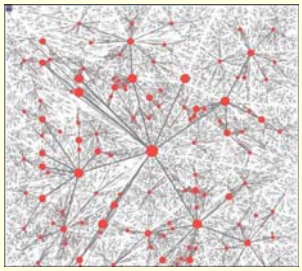
Space-Optimized Tree [Q. Nguyen and M. Huang Infovis 02]

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



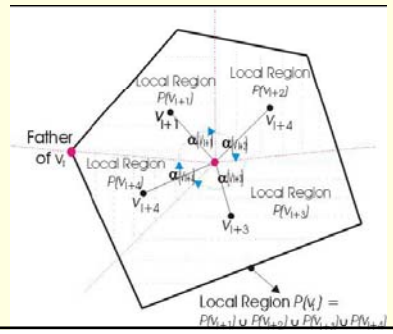
Example: Tree with approximately 55000 nodes

70

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



71

Weight Calculation

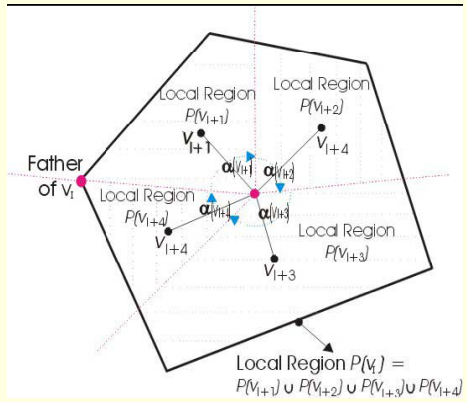
$$w(v_i) = 1 + C\sum_{j=0}^{k-1} w(v_{l+j})$$

- 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.

72

Wedge Calculation

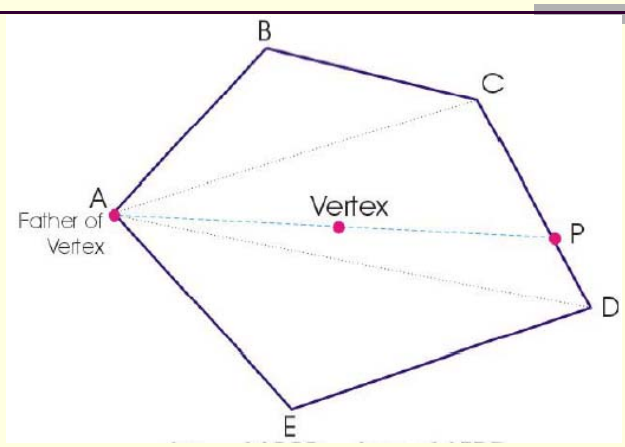
$$\alpha(v_{l+m}) = A \frac{w(v_{l+m})}{\sum_{j=0}^{k} w(v_{l+j})}$$



Example of dividing the local region of one node

73

Vertex Position Calculation

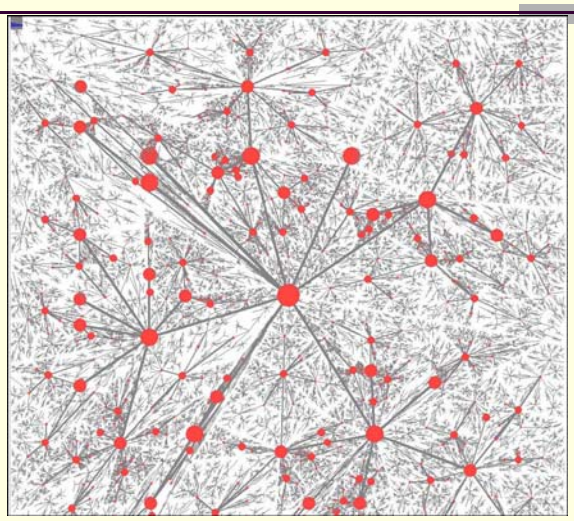


Area ABCP = Area AEDP

Vertex is the midpoint of line AP

74

Space-Optimized Tree



75

Example: Tree with approximately 55000 nodes