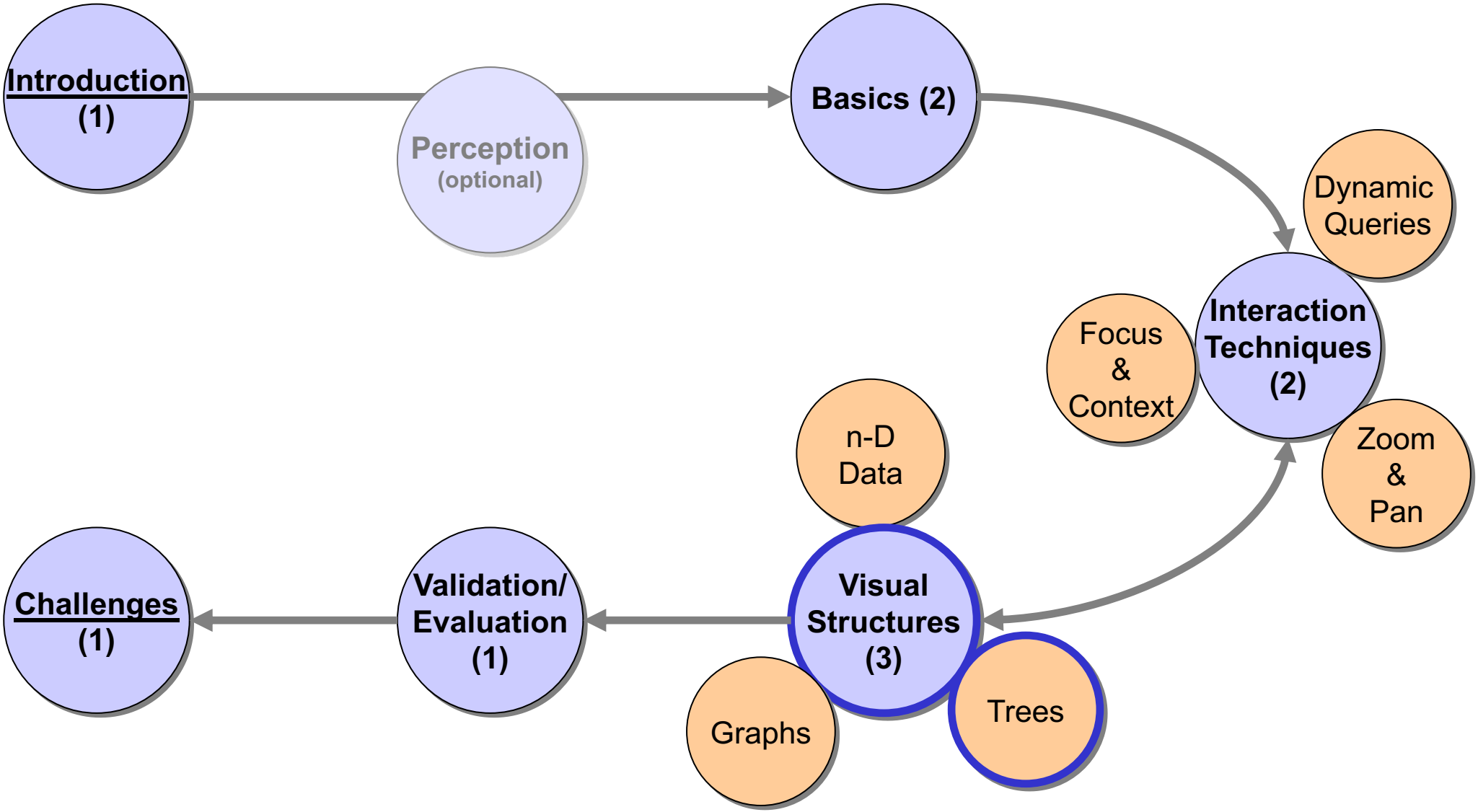
Information Visualization

5. Hierarchies (Trees)


Introduction (1)
Perception (optional)
Basics (2)
Dynamic Queries
Focus & Context
Interaction Techniques (2)
Zoom & Pan
n-D Data
Challenges (1)
Validation/ Evaluation (1)
Visual Structures (3)
Graphs
Trees

5.1 Motivation

- Often, data sets have inherent relations
- In this case, *graphs* are good candidates for the representation, whereas the nodes represent the data cases and the edges the relations
- This general type frequently occurs [→ next lecture/chapter]
- A special case are **hierarchies**
  - Cases are related to sub-cases
  - There is an order, in that cases describe parents or siblings of other cases

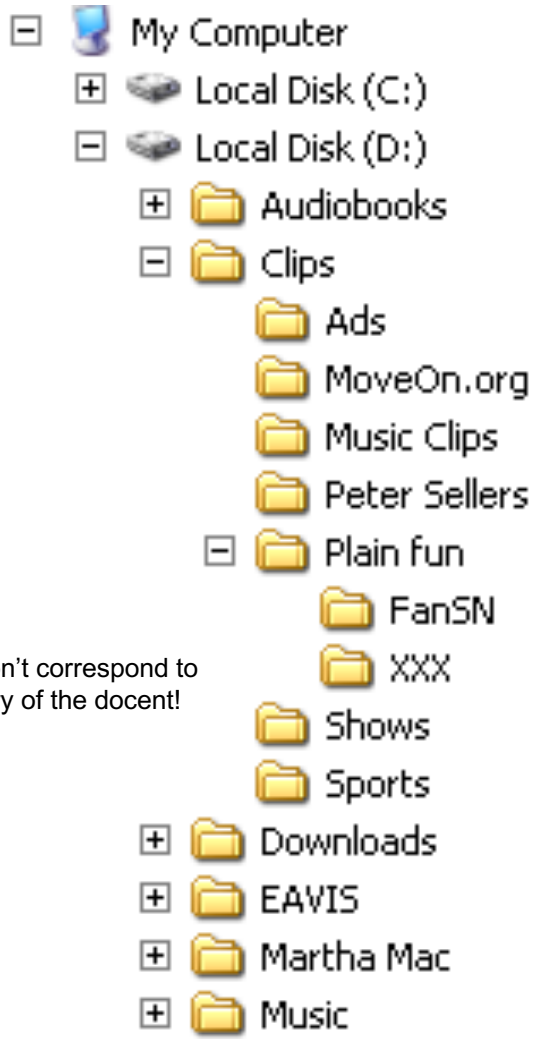
5.1 Motivation

- Examples of hierarchies
  - Organization map of a company
  - Genealogy
  - Taxonomy of biologic species (e.g. Linné)
  - Phylogenetic trees (evolutionary trees)
  - Web pages, including browsing history …
- Examples in Software Visualization
  - Class hierarchies
  - Syntax trees
  - File systems …

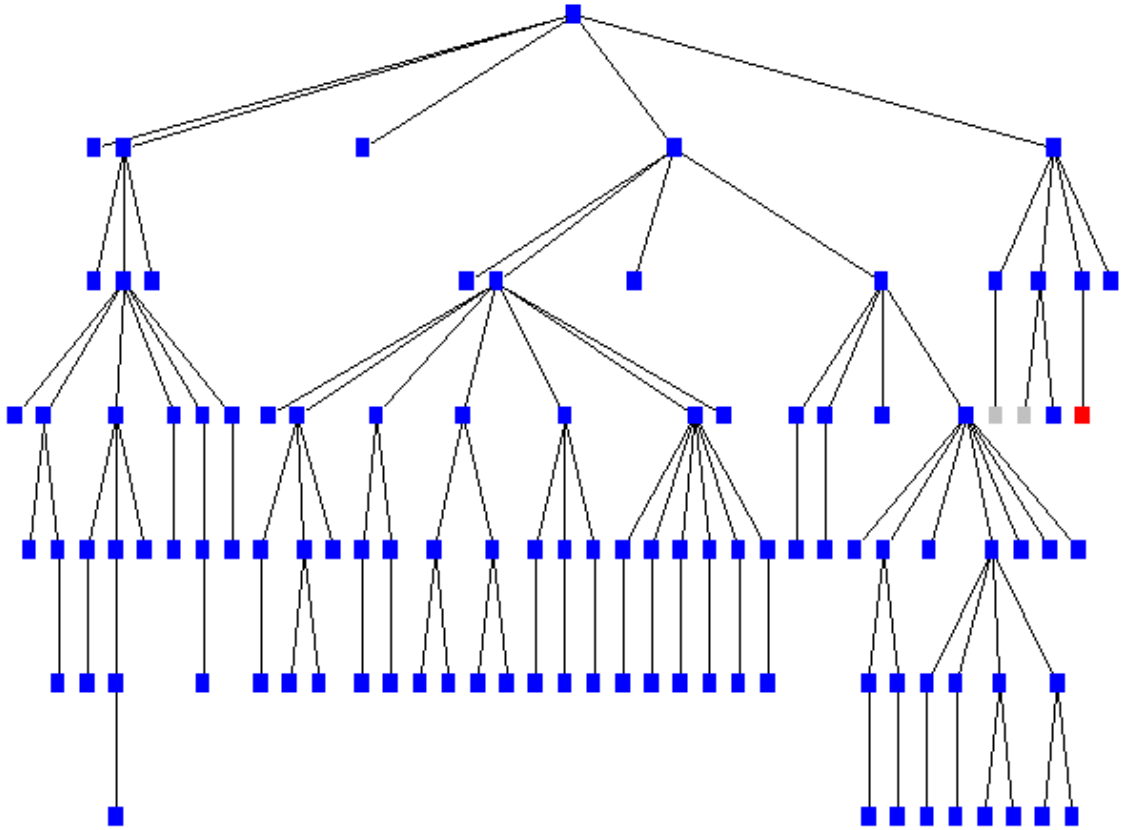
5.2 Representations

- Hierarchies are mostly drawn as *trees*
- Here, there are two basic concepts of representation:
  - *Node-Link Diagrams*
  - *Space-Filling Representations*
- Criteria of good visualizations
  - *Space efficiency* – Do they use the space well?
  - *Information abstraction* – Do they abstract from the data set and avoid information overload?
  - *Simplicity* – Do they base on a simple and elegant drawing method that is easy to implement?
  - *Navigation* – Is it possible to navigate efficiently, to change the focus, etc.?

5.3 Node-Link Diagrams



Node labels don't correspond to a file directory of the docent!



5.3.1 Tree Layout

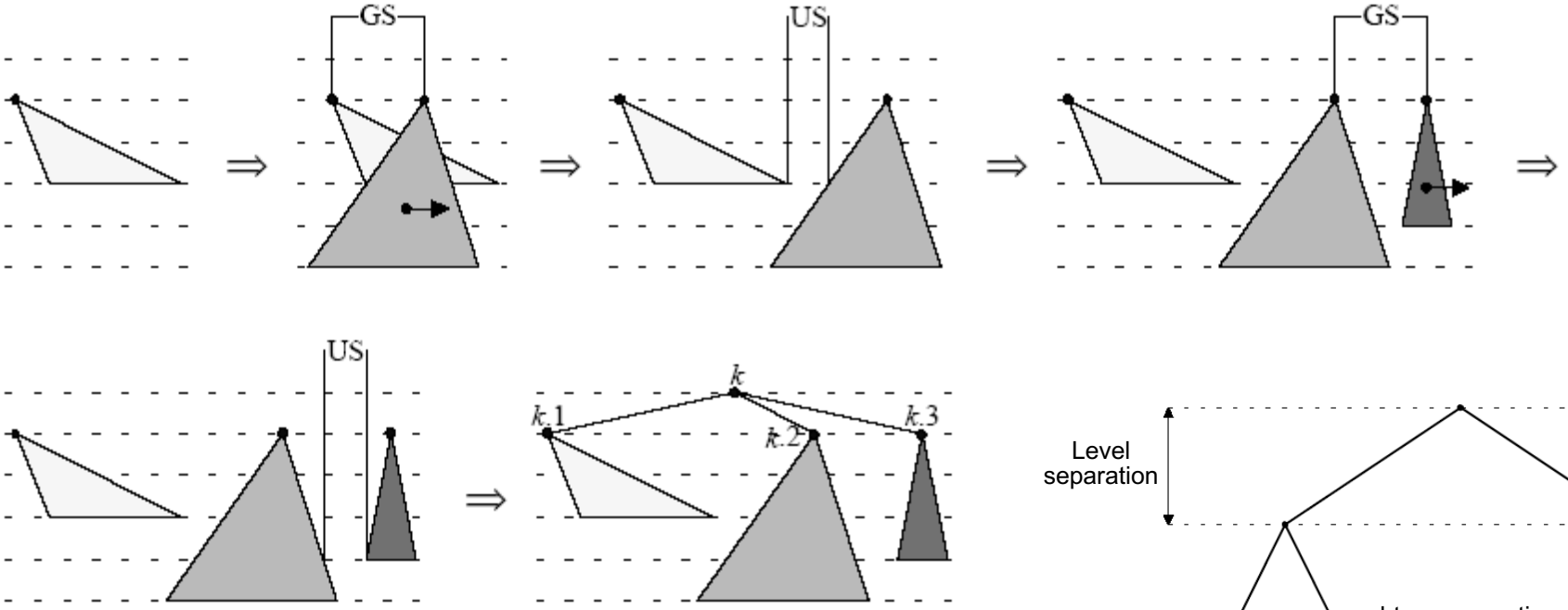
- There are many classic tree layout algorithms. We choose the following algorithm of Walker as an example:
  - [J. Walker. A Node-positioning Algorithm for General Trees. In Software-Practice and Experience, 20(7). pp. 685-705, 1990.]
- This 2-pass algorithm computes a tree layout in quadratic time (even if the author claims that it runs in linear time; Buchheim et al. presented a linear-time algorithm with identical results in 2002)
  - In a first postorder traversal, preliminary node positions are computed and complete subtrees are swung (like a pendulum) according to special layout criteria
  - In a second preorder traversal, the final position of the nodes are computed

5.3.1 Tree Layout

- Layouting by means of a node *k*

subtree of *k*.1
subtree of *k*.2
subtree of *k*.3

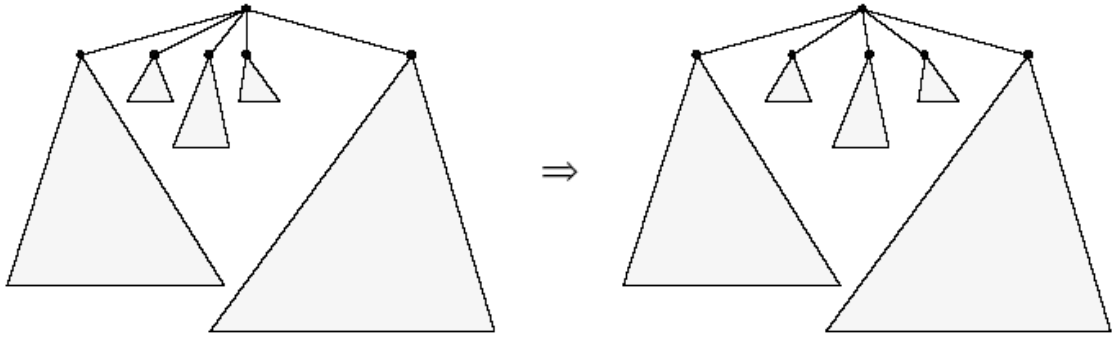
US = subtree separation
GS = sibling separation



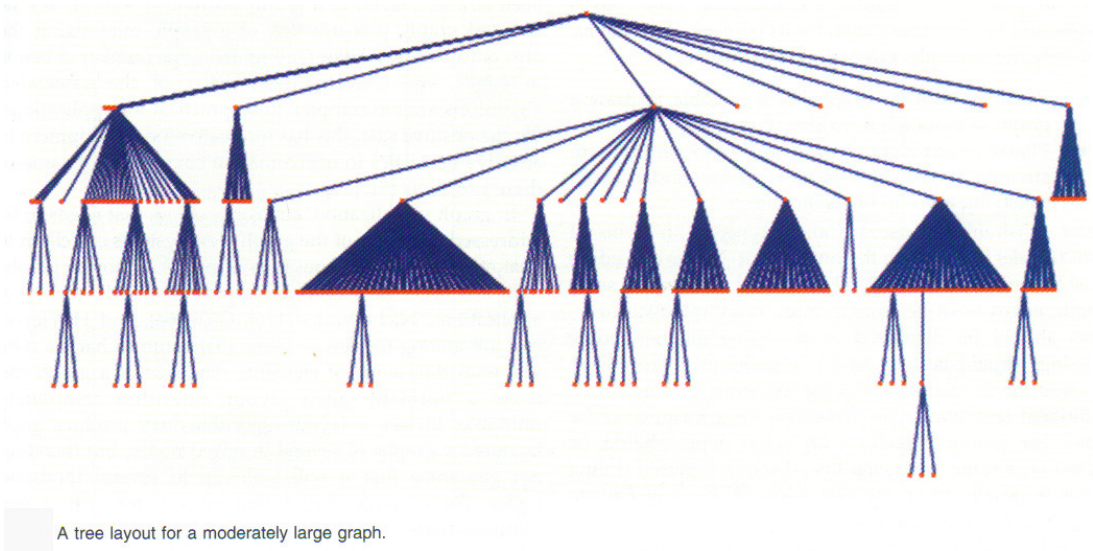
[A. Kerren. Animation der semantischen Analyse. Master's thesis, Universität des Saarlandes, Saarbrücken, 1997.]

5.3.1 Tree Layout

- Swinging of subtrees



- Result

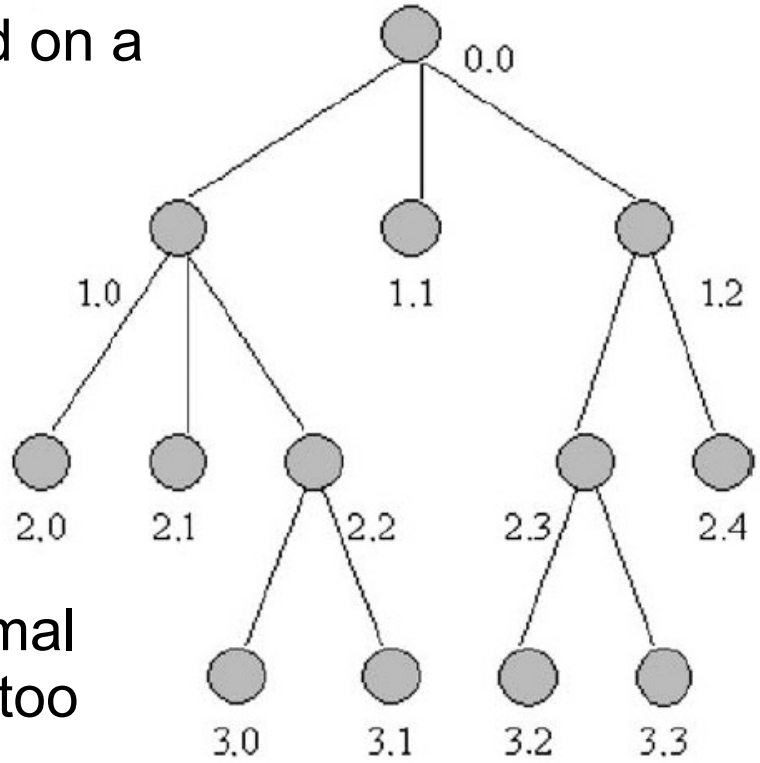


A tree layout for a moderately large graph.

[Demo]

5.3.1 Tree Layout

- Advantages of the method
  - Very familiar 
  - Is used in many applications
  - Nodes on the same level are positioned on a horizontal/vertical line
  - Simple and elegant
- Disadvantages (in general)
  - Space consuming
  - Eventually, no space for labels
  - Large trees cannot be drawn on a „normal screen“, because the representation is too broad

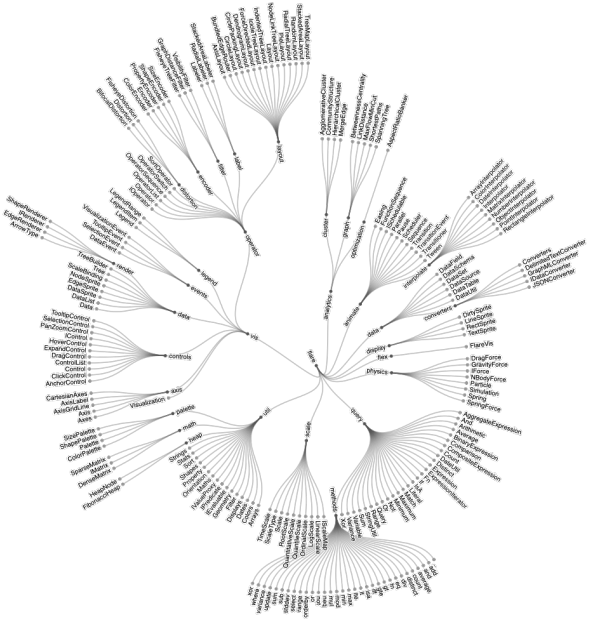


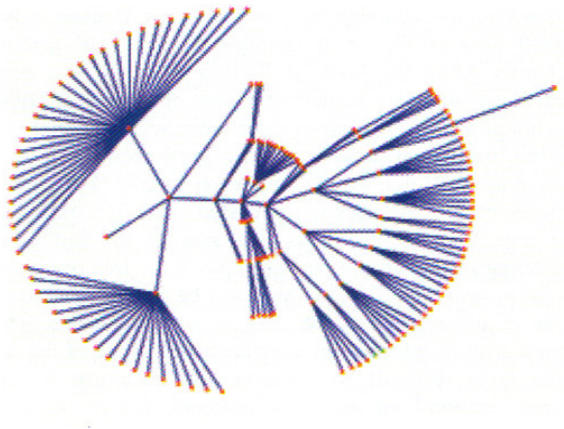
5.3.1 Tree Layout

- Some Layout Criteria
  - A parent should be centered over its offspring
  - Nodes at the same level should lie along a straight line, and the straight lines defining the levels should be parallel
  - A tree and its mirror image should produce drawings that are reflections of one another
    - The same should be true for subtrees
  - Small subtrees should not appear arbitrarily positioned
    - Small, interior subtrees should be spaced out evenly among larger subtrees
    - Small subtrees at the far left or far right should be adjacent to larger subtrees
  - The general layout should efficiently use the available space

5.3.1 Tree Layout

- Instead to place all the nodes of one level to a horizontal/vertical line, we can also arrange them radially
  - In the layout, the root is drawn in the center, and all other nodes of one level lie on concentric circles
  - Furthermore, algorithms avoid overlapping through determining of sectors for subtrees. It is also possible to soften the last condition to get good average results

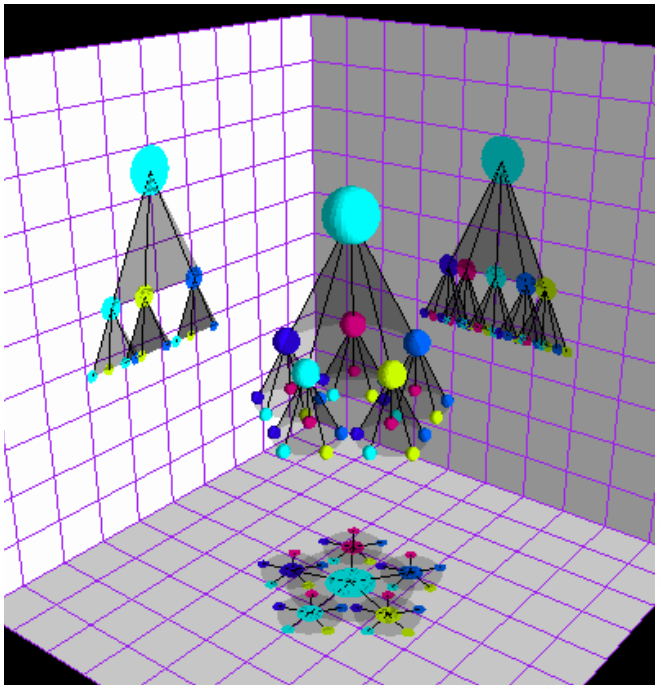


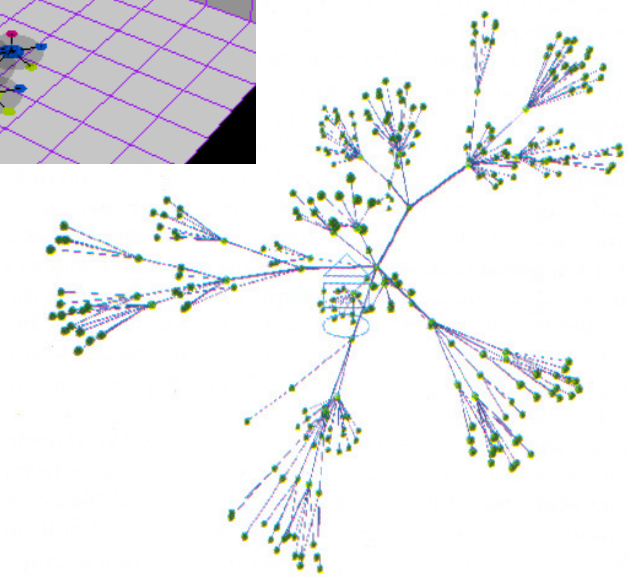


[I. Herman, G. Melancon, M. De Ruiter, and M. Delest. „Latour - A Tree Visualization System“. In Proc. of the Symp. on Graph Drawing, GD'99, pp. 392-399, 1999.]

5.3.1 Tree Layout

- Many tree layout algorithms were ported into 3D
- In this way, there is more space available, but on the other hand, we get other problems, such as
  - navigation problems
  - overlaps
  - design of the labels
  - …





3D version of a radial algorithm. (Courtesy of S. Benford, University of Nottingham, U.K.)

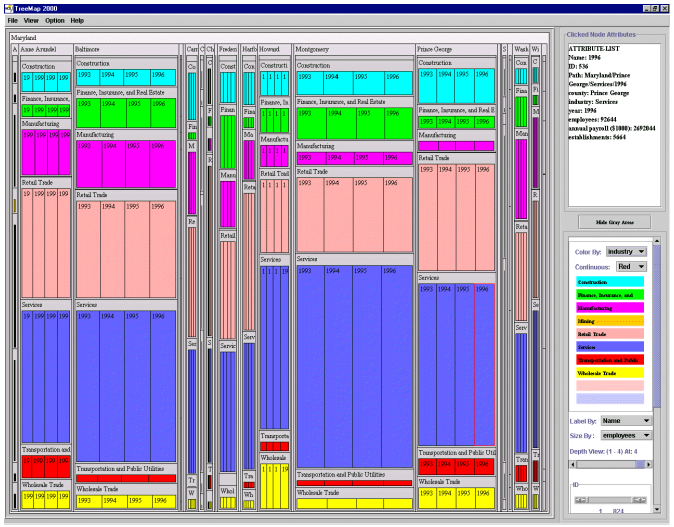
5.3.2 Discussion

- Node-link diagrams are intuitive, but the have disadvantages regarding scalability
- It is difficult to embed several variables for data objects (nodes), e.g., with the help of
  - shape
  - color
  - size
  - …

  ⇒ All possibilities collide more or less with the basic node-link structure

5.4 Space-Filling Representations

- *So-called Space-Filling Representations* try to solve the conceptual problems of node-link diagrams
  - space consumption
  - inclusion of many (and complex) attributes
- In general, space-filling visualizations are not very suitable for unbalanced trees
- Oldest and most prominent realization:
  - *Treemaps*



5.4.1 Treemaps

- Presented by Johnson and Shneiderman, Vis `91
- Hierarchy is recursively mapped to rectangles
  - Representations of the children (rectangles) are presented within the parents‘ representations
  - Alternate horizontal and vertical subdivision from level to level
- Attributes can be coded within the rectangles, e.g.:
  - Node type with color, size/value with area, text …
- Advantage: Very good overview

[Brian Johnson and Ben Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In Proc. IEEE Visualization '91, pages 284--291. IEEE CS, 1991.]

5.4.1 Treemaps

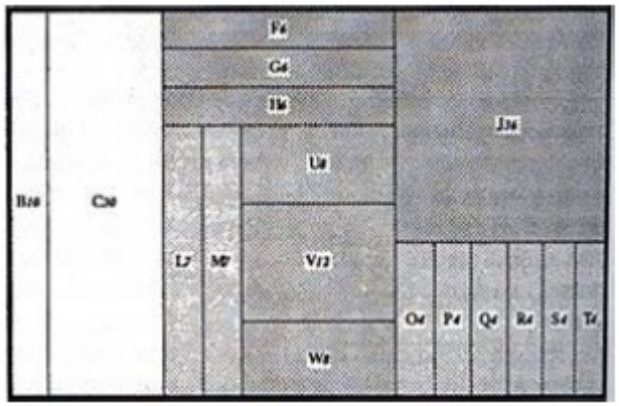
- Algorithm by means of file directories

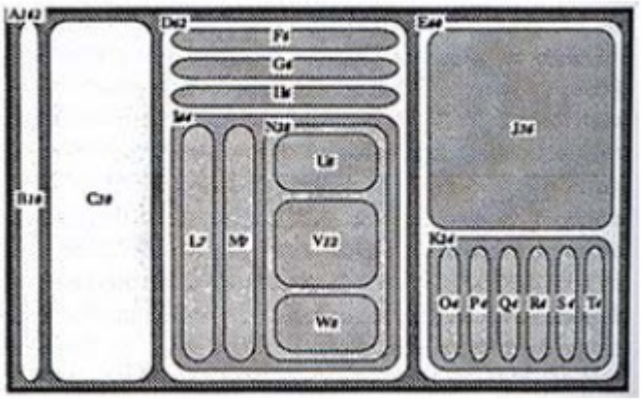
```
Draw()
{
   Change orientation from parent (horiz/vert)
   Read all files and directories at this level
   Make rectangle for each, scaled to size
   Draw rectangles using appropriate size and color
   For each directory
      Make recursive call using its rectangle as focus
}
```

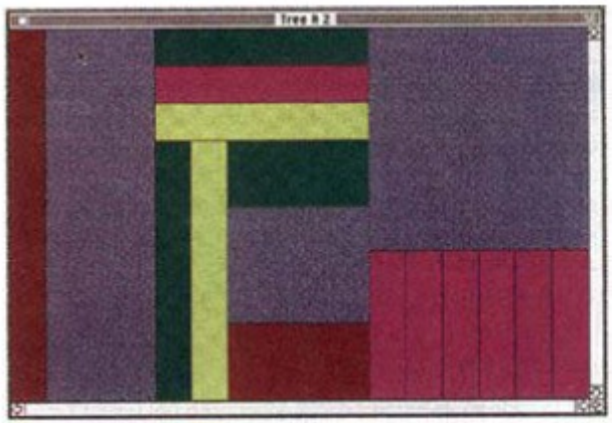
[taken from J. Stasko's course]

5.4.1 Treemaps

- Nested vs. non-nested Treemaps





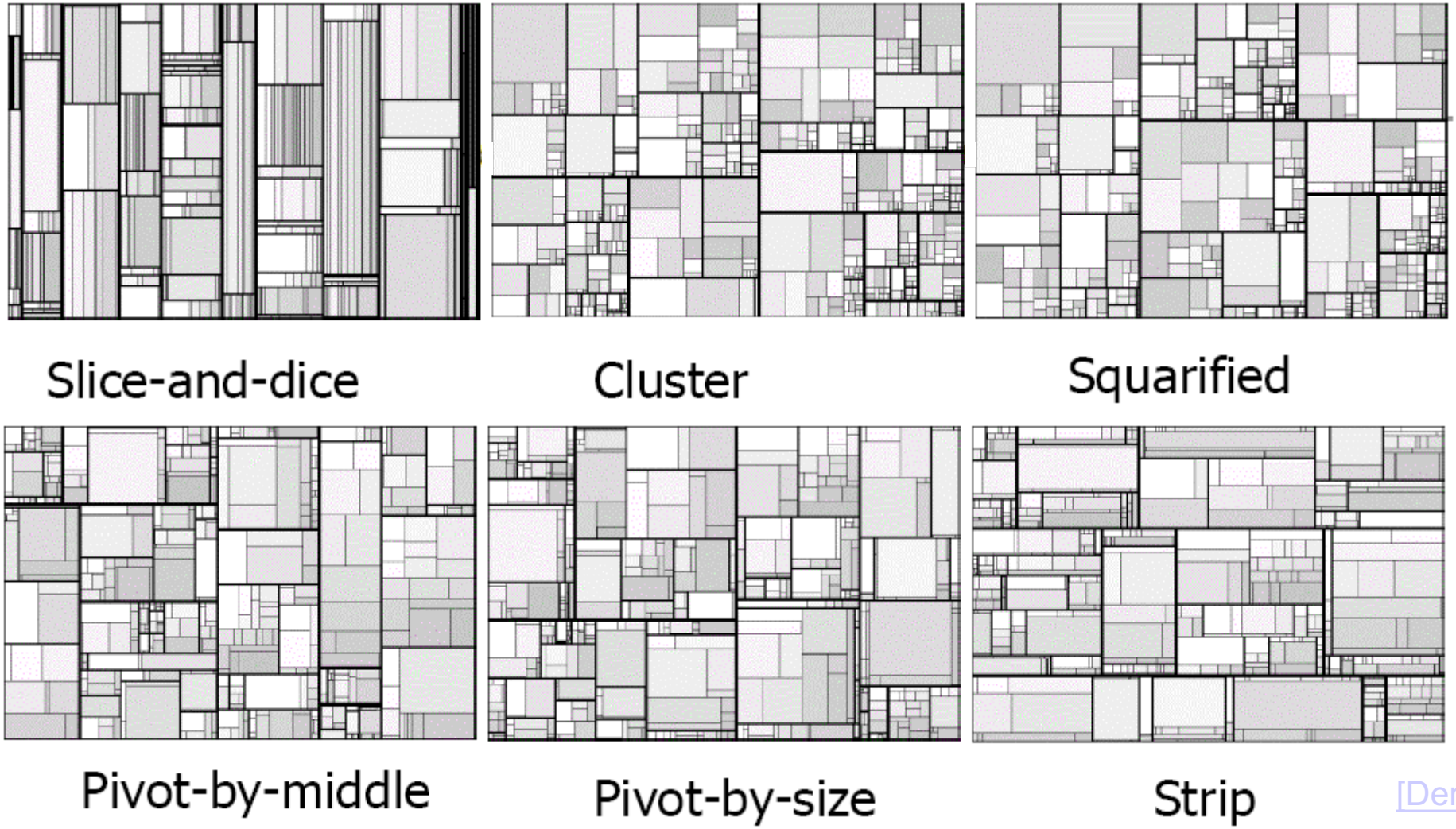


Non-nested Treemaps

Nested Treemaps

5.4.1 Treemaps

- Variations of Treemap algorithms

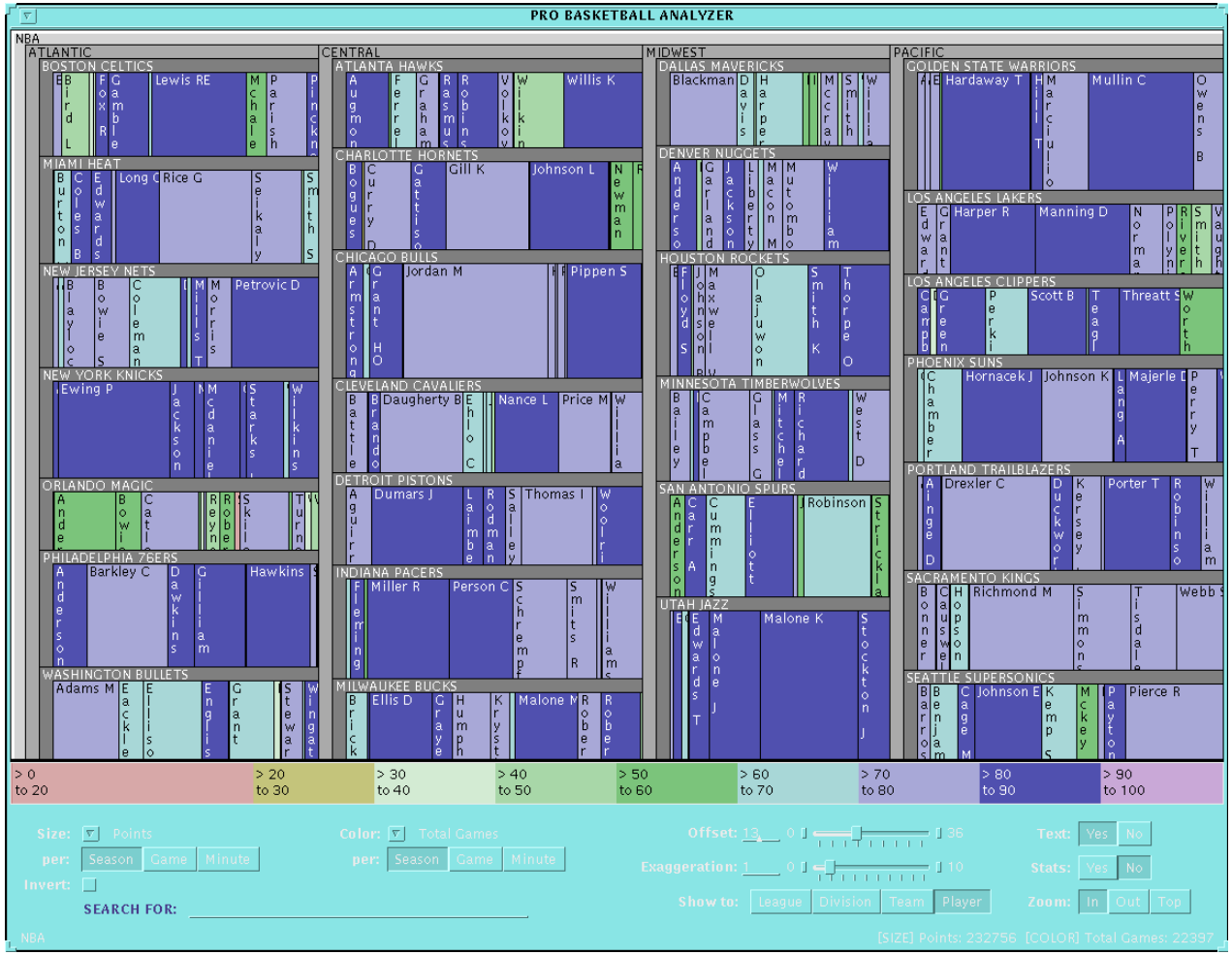


[taken from J. Stasko's course]

[Demo]

5.4.1 Treemaps

- Application: Sport Statistics

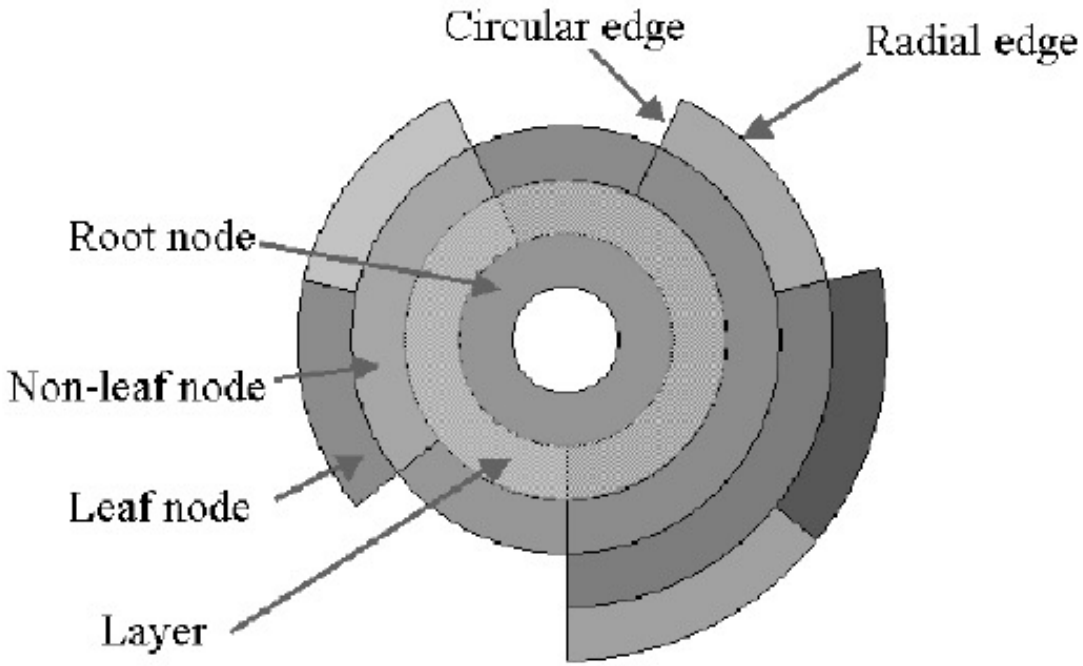


[Video]

https://www.cs.umd.edu/hcil/treemap-history/

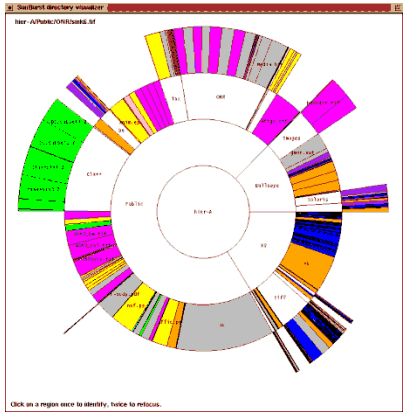
5.4.2 Radial Space-Filling

- Idea of Radial Space-Filling techniques
  - Use circle segments instead of rectangle areas for the representation of the tree nodes



5.4.2 Radial Space-Filling

Sunburst



- Presented by Stasko and Zhang, InfoVis `00
- Focus & Context display with 3 selection methods:
  - Angular Detail
  - Detail Outside
  - Detail Inside
- Empirical study shows advantages compared to Treemaps according to navigation time, learnability and preference

[John Stasko, Eugene Zhang. "Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations", p. 57, IEEE Symposium on Information Visualization 2000, 2000.]

http://www.cc.gatech.edu/gvu/ii/sunburst/

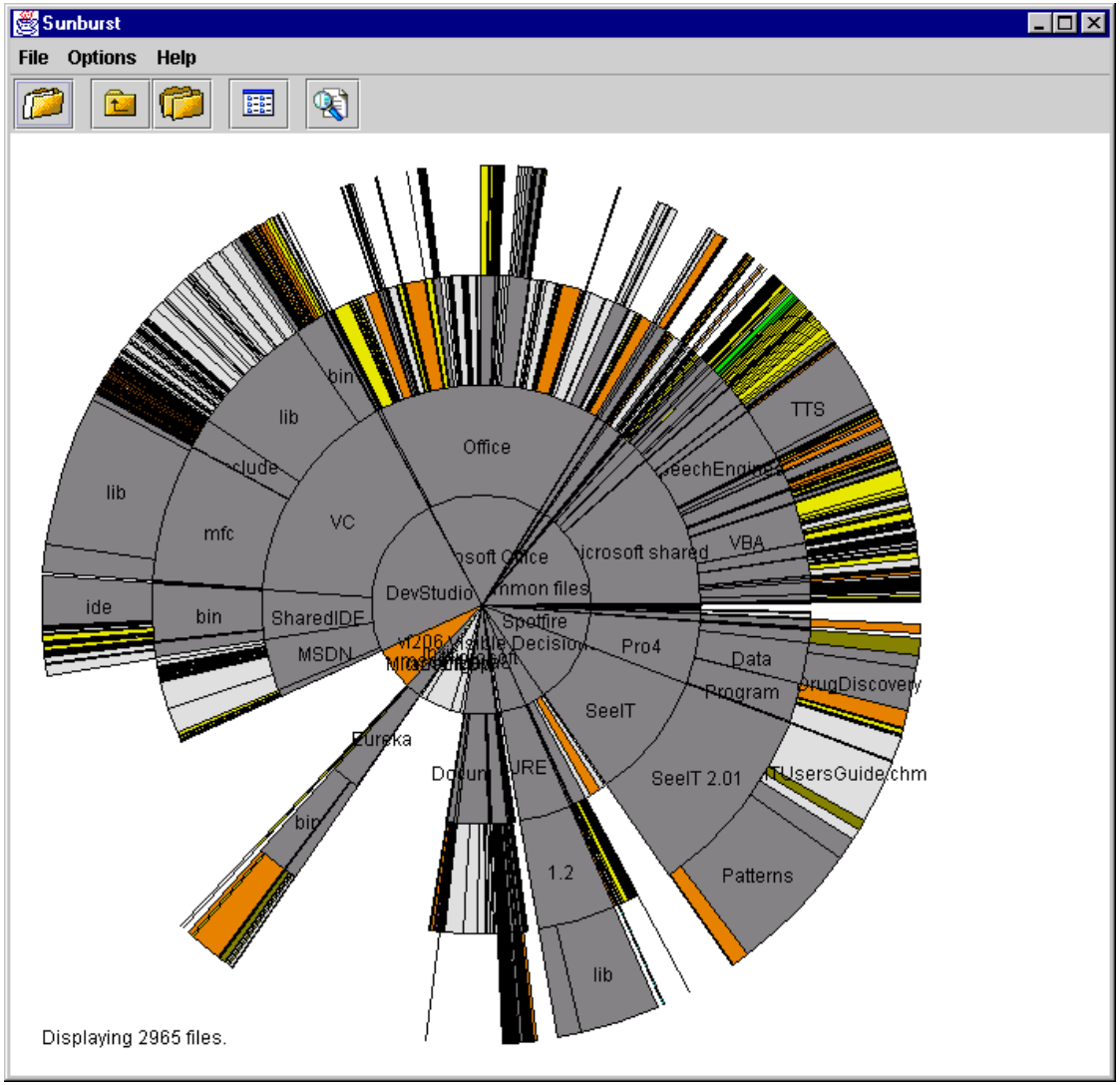
5.4.2 Radial Space-Filling

- Example: Sunburst

Visualization of a file hierarchy.

Here, we discover a disadvantage: Files, that are located in the periphery, are drawn very small.

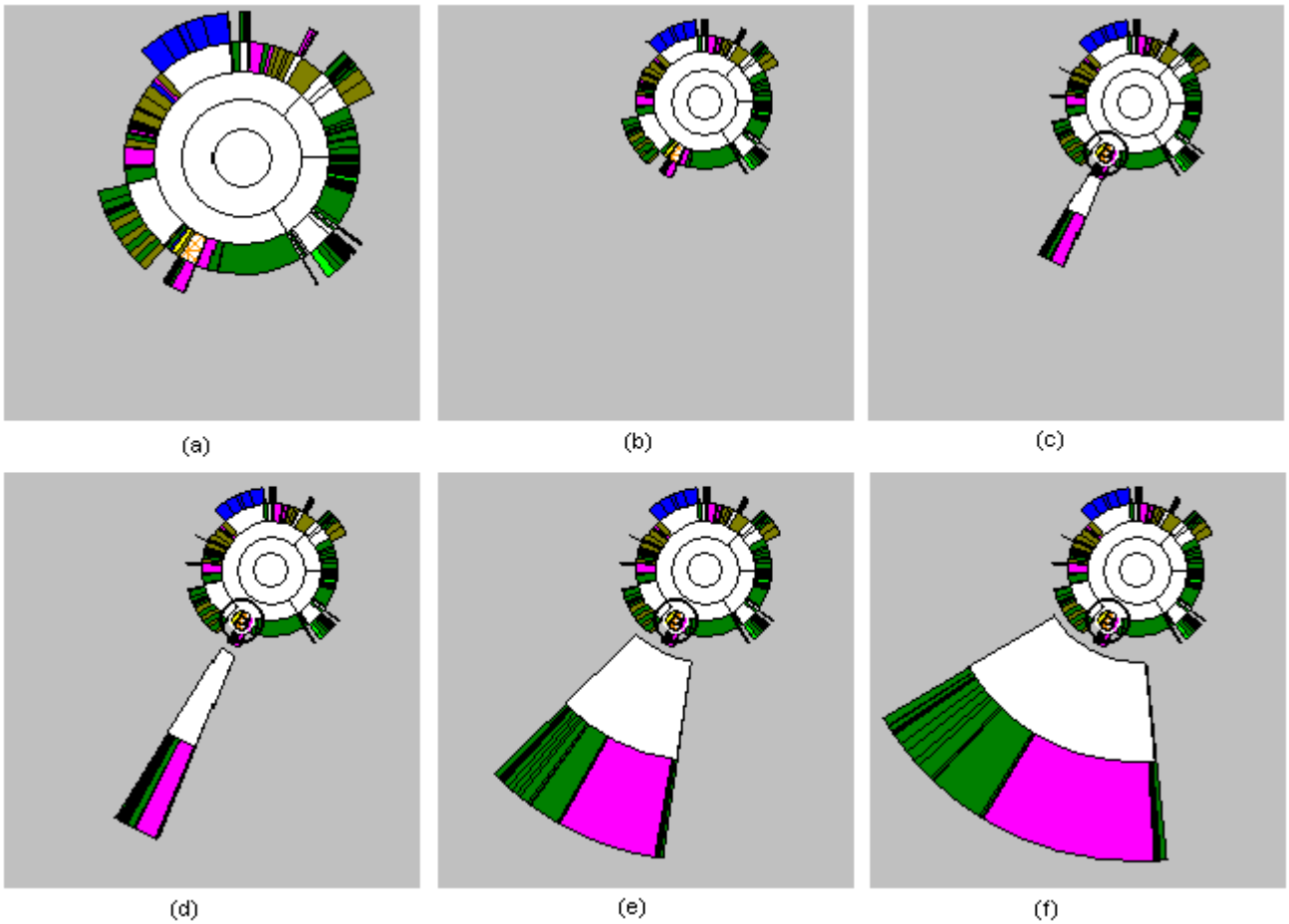
3 selection methods shall solve this problem!



5.4.2 Radial Space-Filling

Angular Detail

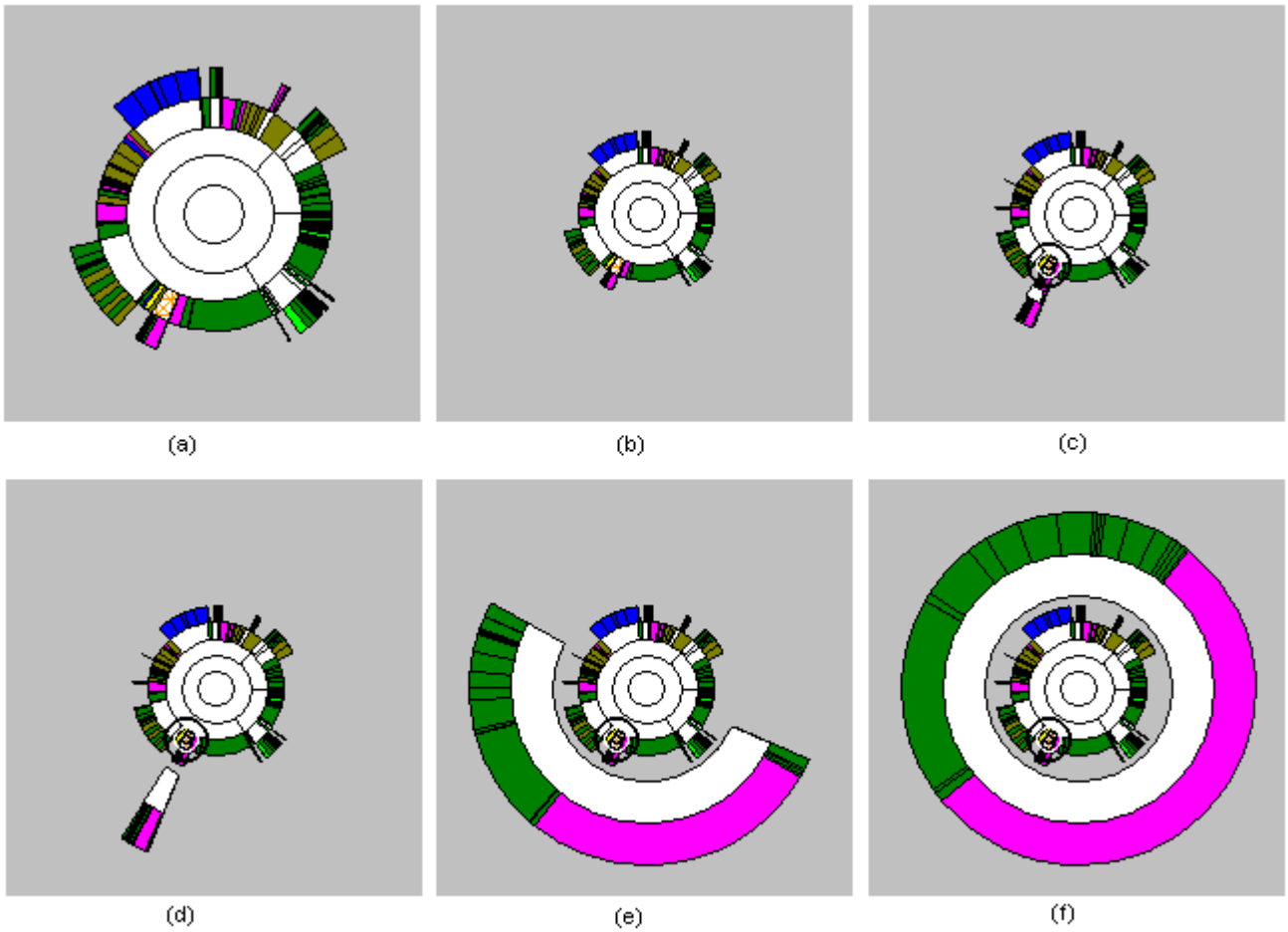
- Overview is scaled down
- Focus region expands out from the circle
- Less space saving!



5.4.2 Radial Space-Filling

Detail Outside

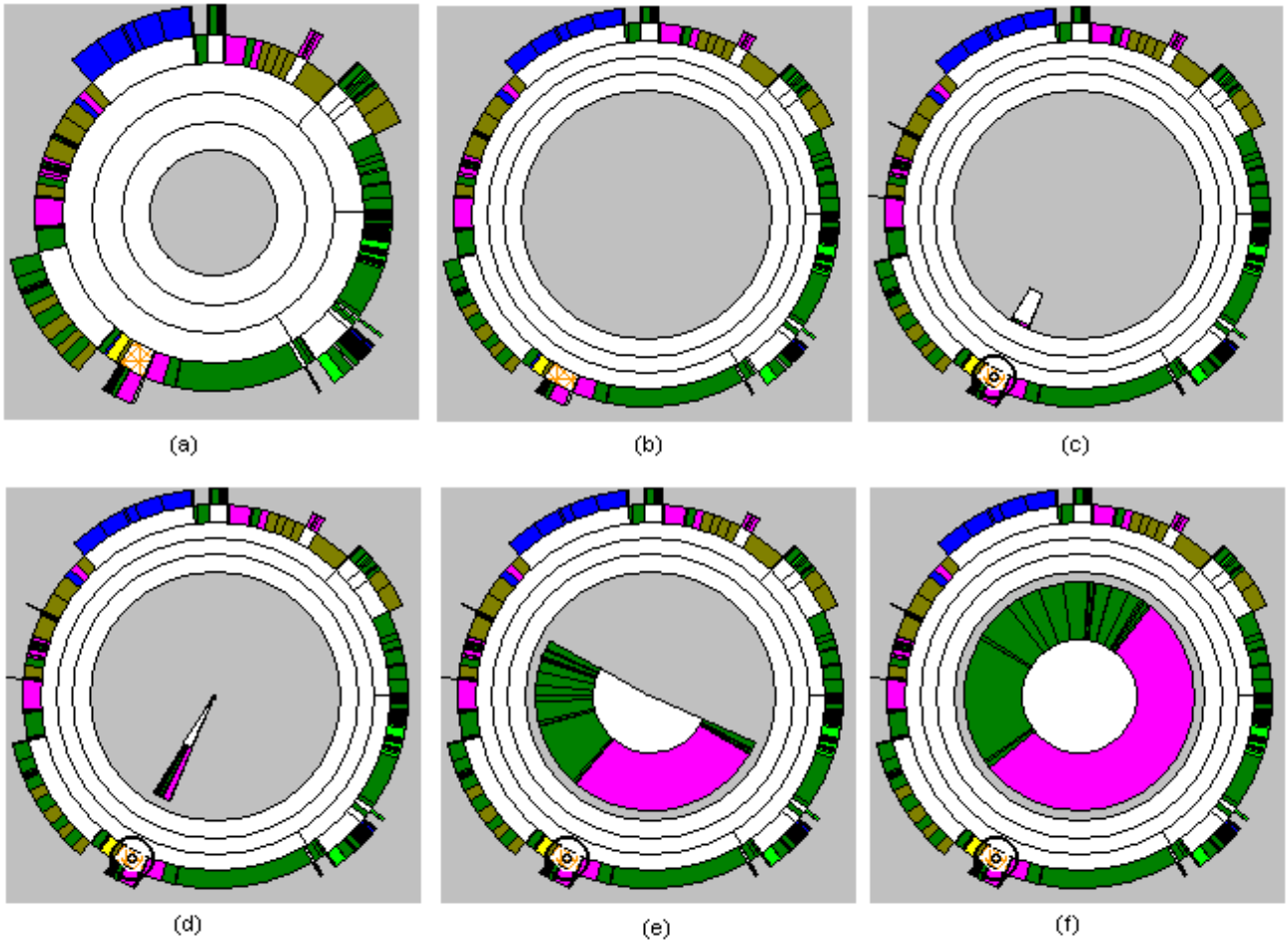
- Overview is scaled down
- Focus region builds a ring around this overview and is magnified
- Space saving!



5.4.2 Radial Space-Filling

Detail Inside

- Overview is magnified
- Focus region builds a new circular area within this overview
- Possibly, less intuitive, because of distortion
- Very space saving!



[Video] [Demo]

5.5 TreeVis Survey

- Treevis.net
  - A Tree Visualization Reference (currently 333 techniques)