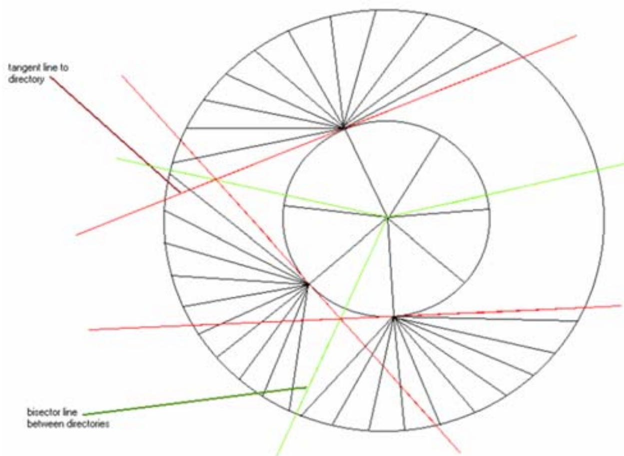
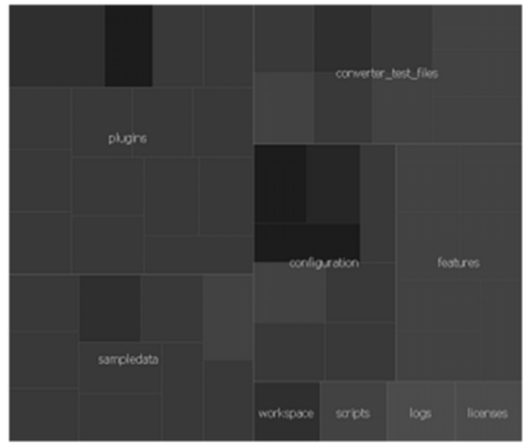




Information Visualization MOOC

Session 5 – "With Whom": Tree Data

Algorithm Comparison

- Radial Tree Layout
- Treemap Layout

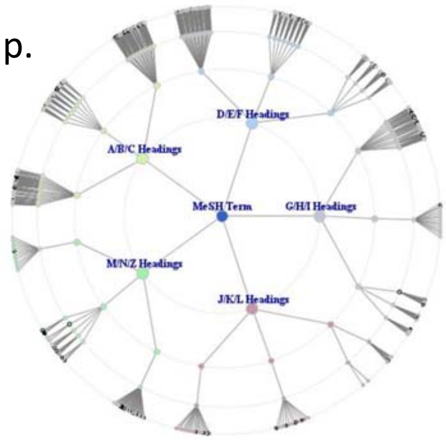
CNS Cyberinfrastructure for Network Science Center

http://ivmooc.cns.iu.edu

INDIANA UNIVERSITY

Radial Tree Layout

- All nodes lie in concentric circles that are focused in the center of the screen.
- Nodes are evenly distributed.
- Branches of the tree do not overlap.



Source: Greg Book & Neeta Keshary. 2001. "Radial Tree Graph Drawing Algorithm for Representing Large Hierarchies." University of Connecticut Class Project.

Circle Placement

Maximum size of the circle corresponds to minimum screen width or height.

Distance between levels d := radius of max. circle size / number of levels in the graph.

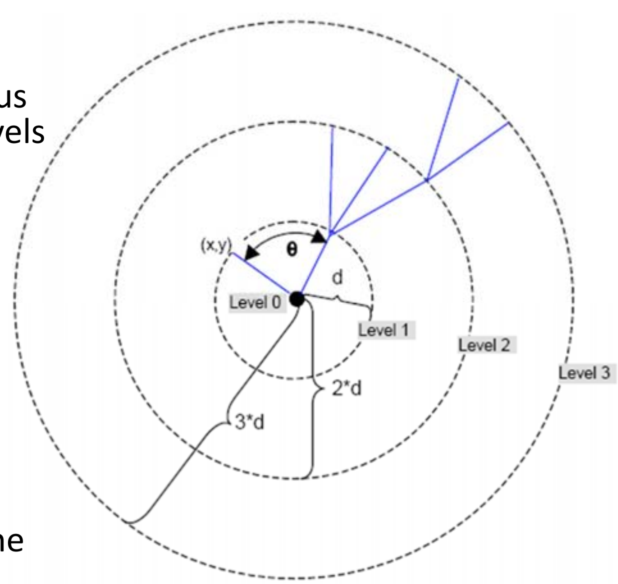
Node Placement

Level 0

The root node is placed at the center.

Level 1

All nodes are children of the root node and can be placed over all 360° of the circle. Divide 2pi by the number of nodes at level 1 to get angle space between the nodes on the circle.

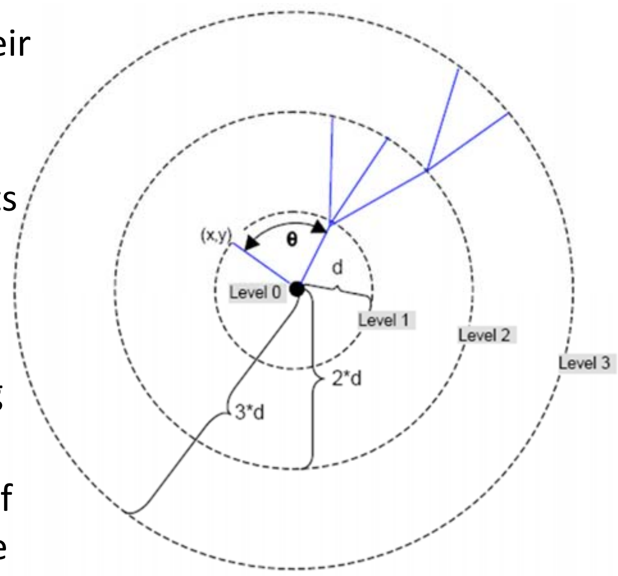


Levels 2 and greater

Use information on number of parents, their location, and their space for children to place all level x nodes.

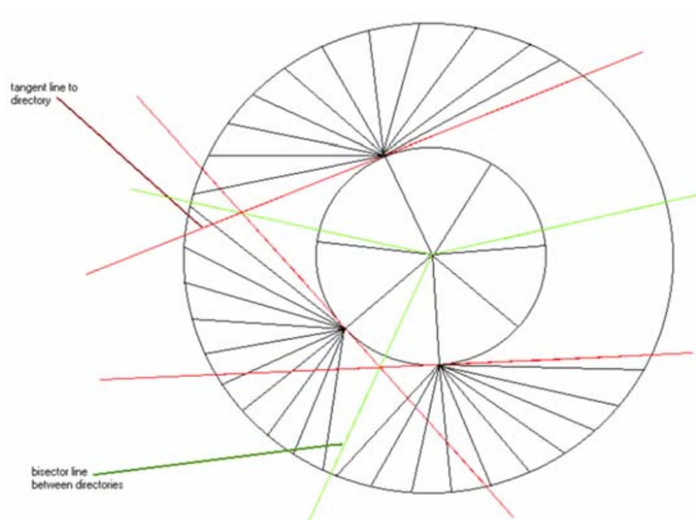
Loop through the list of parents and then loop through all the children for that parent and calculate the child's location relative to the parent's, adding in the offset of the limit angle.

After calculating the location, if there are any directories at the level, we must calculate the bisector and tangent limits for those directories.



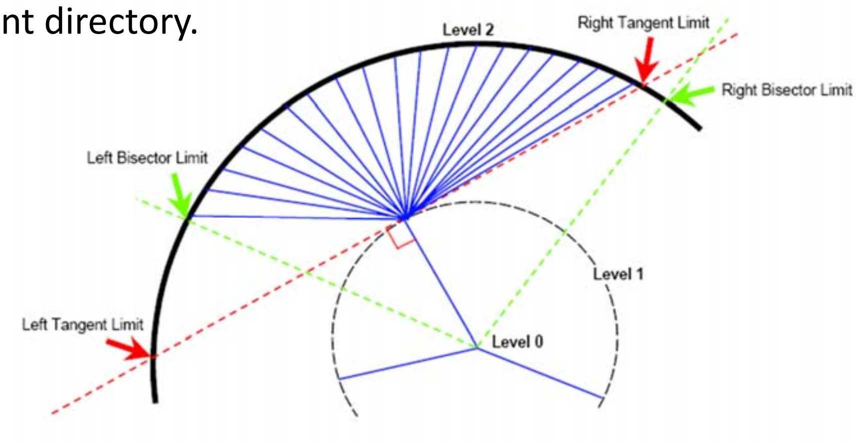
We then iterate through all the nodes at level 1 and calculate the position of the node.

Bisector Limits -->

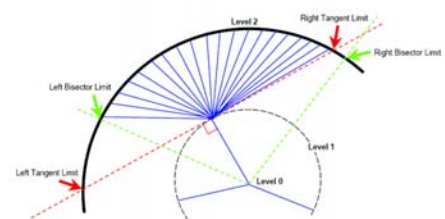


Tangent and bisector limits for directories

Between any two directories, a bisector limit is calculated to ensure that children do not overlap the children of an adjacent directory.

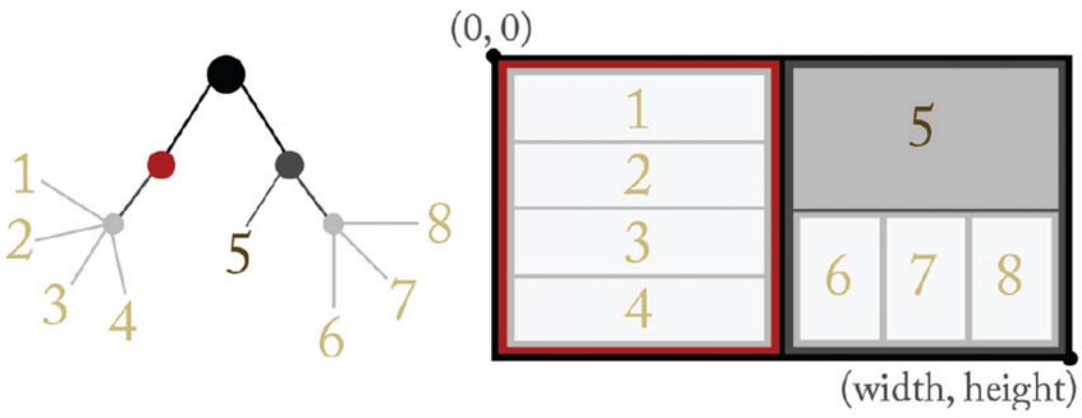


Pseudo Code




```
Loop through each level in the data structure

Switch level

    case 0:
        Find the center of the drawing area, to center the graph
        Set RootNode = CenterX, CenterY

    case 1:
        AngleSpace = (2Π radians/NumNodesAtThisLevel)
        Loop through all nodes at this level
            Calculate x,y positions:
            If (Node.type == Parent)
                Calculate bisector limits and tangent limits for the node
        End loop

    case 2:
        Nodes in levels two and higher must be grouped according to their parent.
        Loop through all nodes at this level and get a list of the parent nodes
        And get the number of children for each parent
        Calculate the AngleSpace for each parent:
            AngleSpace = (leftLimit – rightLimit)/NumNodesForThisParent
        Foreach parent
            Loop through all nodes for that parent
                Calculate x,y position for the child node
                If (childnode.type == Directory)
                    Calculate bisector and tangent limits
            End loop
        End foreach

End switch
```

Treemap Layout



Pseudo Code

Input

Tree root & a rectangular area defined by upper-left and lower-right coordinates P1(x1, y1), Q1(x2, y2).

Recursive Algorithm

```
active_node := root_node;
partitioning_direction := horizontal; // nodes are partitioned vertically at even
   levels and horizontally at odd levels

Tremap(active_node) {
   determine number n of outgoing edges from the active_node;
   if (n<1)
     end;
   if (n>1) {
     divide the region [x1, x2] in partitioning direction where the size of the n
          partitions correspond to their fraction (Size(child[i])/Size(active)) of
          the total number of bytes in the active_node;
     change partitioning_direction;
     for (1<=i<=n) do
          Treemap(child[i]);
   }
```

Strengths

- Utilizes 100% of display space.
- Shows nesting of hierarchical levels.
- Represents node attributes (e.g., size and age) by area size and color.
- Scalable to data sets of a million items.

Weaknesses

- Size comparison is difficult.
- Labeling is a problem.
- Cluttered display.
- Difficult to discern boundaries.
- Shows only leaf content information.

Acknowledgments

We would like to thank Miguel Lara and his colleagues at the Center for Innovative Teaching and Learning, University Information Technology Services at Indiana University, Bloomington.

The tool development work is supported in part by the Cyberinfrastructure for Network Science Center (http://cns.iu.edu) and Indiana University, the National Science Foundation under Grants No. SBE-0738111 and IIS-0513650, the US Department of Agriculture, the National Institutes of Health, and the James S. McDonnell Foundation.

Many visualizations used in the course come from the *Places & Spaces: Mapping Science* exhibit, online at http://scimaps.org, and from the *Atlas of Science: Visualizing What We Know*, MIT Press (2010).

INDIANA UNIVERSITY NSF USDA James S. McDonnell Foundation CNS Cyberinfrastructure for Network Science Center