Tree Visualization

Courtesy of Prof. Shixia Liu @Tsinghua University
Outline

• Introduction
• Node-Link diagrams
• Space-Filling representation
• Hybrid methods
Review: Trees

• Hierarchies often represented as trees
  – Directed, acyclic graph

• Two main representation schemes
  – Node-link
  – Space-filling
Review: Outline

- Introduction
- Node-Link diagrams
- Space-Filling representation
- Hybrid methods
Review: Node-Link Diagrams

• Root at top, leaves at bottom is very common
Review: Basic Algorithm

- Recursive algorithm
- Height on separate levels
- Divide columns with unique widths
- Make room for subtrees
Review: Other Tree Drawing Methods

- Radial drawing

- Ringed circular drawings

- Orthogonal drawings (alternating vertical and horizontal placement)
Review: Cone Trees

- 3D extension of the 2D ringed circular layout
- 3D views of hierarchies such as file systems
- Developed at Xerox PARC

[Robertson et al., 1993]
Review: Alternative Solutions

- Change the geometry
- Apply a hyperbolic transformation to the space
- Root is at center, subordinates around
- Apply idea recursively, distance decreases between parent and child as you move farther from center, children go in wedge rather than circle

Figure 2: Original inspiration for the hyperbolic tree browser. Circle Limit IV (Heaven and Hell), 1960, (c) 1994 M.C. Escher / Cordon Art – Baarn – Holland. All rights reserved. Printed with permission.
Review: Alternative Solutions

- Change the geometry
- Apply a hyperbolic transformation to the space
- Root is at center, subordinates around
- Apply idea recursively, distance decreases between parent and child as you move farther from center, children go in wedge rather than circle
Outline

• Introduction
• Node-Link diagrams
• Space-Filling representation
• Hybrid methods
Space-Filling Representation

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

One example: “Icicle plot”
Treemap

• 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 other variable of data items

[Johnson et al., 1991]
Example: A Simple One
Example: A Simple One

[Diagram of a tree and a grid with numbers]
Example: Directories
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
  }
Nested vs. Non-nested

Non-nested Tree-Map

Nested Tree-Map
Pros and Cons

• Good
  – Representation of two attributes beyond node-link: color and area

• Not as good
  – Also can get long-thin aspect ratios
  – Borders help on smaller trees, but take up too much area on large, deep ones
  – What if nodes with zero value (mapped to area) are very important?

How to solve these problems?
Aspect Ratios

These kinds of rectangles are visually unappealing

Which has bigger area?
“Cluster” Treemap

• “Compromises” treemap algorithm to avoid bad aspect ratios

• Basic algorithm (divide and conquer) with some hand tweaking
  – Creates a more fully 2-dimensional layout by employing both vertical and horizontal partitions at each level of hierarchy

• Takes advantage of shallow hierarchy
Typical Slice-and-Dice Layout
Cluster Treemap
Example: FinViz

- [http://finviz.com/map.ashx?t=sec](http://finviz.com/map.ashx?t=sec)
Squarified Treemap

- Alternate approach, similar results

[Bruls et al., 2000]
Squarified Treemap (cont’d)

- Aspect-ratios approach 1 as close as possible?
- NP-hard
- A greedy method

The order in which the rectangles are processed is important
**Pivot Treemap**

- Goal: create a layout in which items that are next to each other in the input to the algorithm are adjacent in the treemap.
- Inspired by Q-sort
  - Pivot
Pivot Treemap (cont’d)

- **Input**
  - Rectangle, $R$, to be subdivided
  - List of items with area, $L_1 \cdots L_n$

- **Output**
  - List of rectangles, $R_1 \cdots R_n$
Pivot Treemap (cont’d)

• Step 1: If the number of items is \( \leq 4 \), lay them out in either a pivot, quad, or snake layout.

• What if not?
Pivot Treemaps (cont’d)

• Step 2: Let P, the pivot, be the item with the largest area in the list of items.

• Step 3: If the width of total area R ≥ height, divide R into four rectangles, R1, RP, R2, and R3 (If height > width, flipped)
Pivot Treemap (cont’d)

- Step 4: Divide the items in the list
  - L1: index is less than P
  - L2: index less than those in L3
  - L3: others
  - Goal: the aspect ratio of RP is as close to 1 as possible

- Step 5: Put P in the rectangle RP

- Step 6: Recursively lay out L1, L2, and L3
Strip Treemap

- Processing input rectangles in order, and laying them out in horizontal (or vertical) strips of varying thicknesses
Strip Treemap (cont’d)

• Step 1
  – Create a new empty strip (the current strip)

• Step 2
  – Add the next rectangle to the current strip
  – Recomputing the height of the strip
  – Recomputing the width of each rectangle

• Step 3
  – If the average aspect ratio increased, remove the rectangle, push it back, and go to Step 1

• Step 4
  – If all rectangles have been processed, stop
  – Else, go to step 2
An Example

Input Rectangle: 10x10
Input Sequence: 10, 20, 30, 20, 20

The average aspect ratio of the current row after adding the new rectangle is shown.
Showing Structure

• Regular borderless treemap makes it challenging to discern structure of hierarchy, particularly large ones
  – Supplement treemap view
  – Change rectangles to other forms
Variation: Cushion Treemap

• Add shading and texture to help convey structure of hierarchy

[Wijk et al., 1999]
Example

• https://www.treemap.com/datasets/uselections/?goback=.gde_80552_member_184123140
Summary

• Node-link diagrams or space-filling techniques?
• It depends on the properties of the data
  – Node-link typically better at exposing structure of information structure
  – Space-filling good for focusing on one or two additional variables of cases
Further Reading

• Survey website
  – http://vcg.informatik.uni-rostock.de/~hs162/treeposter/poster.html
References


• G. Robertson, S. Card, and J. Mackinlay, "Information Visualization Using 3D Interactive Animation", *Communications of the ACM*, vol. 36, no. 4, Apr. 1993, pp. 57-71.


References


