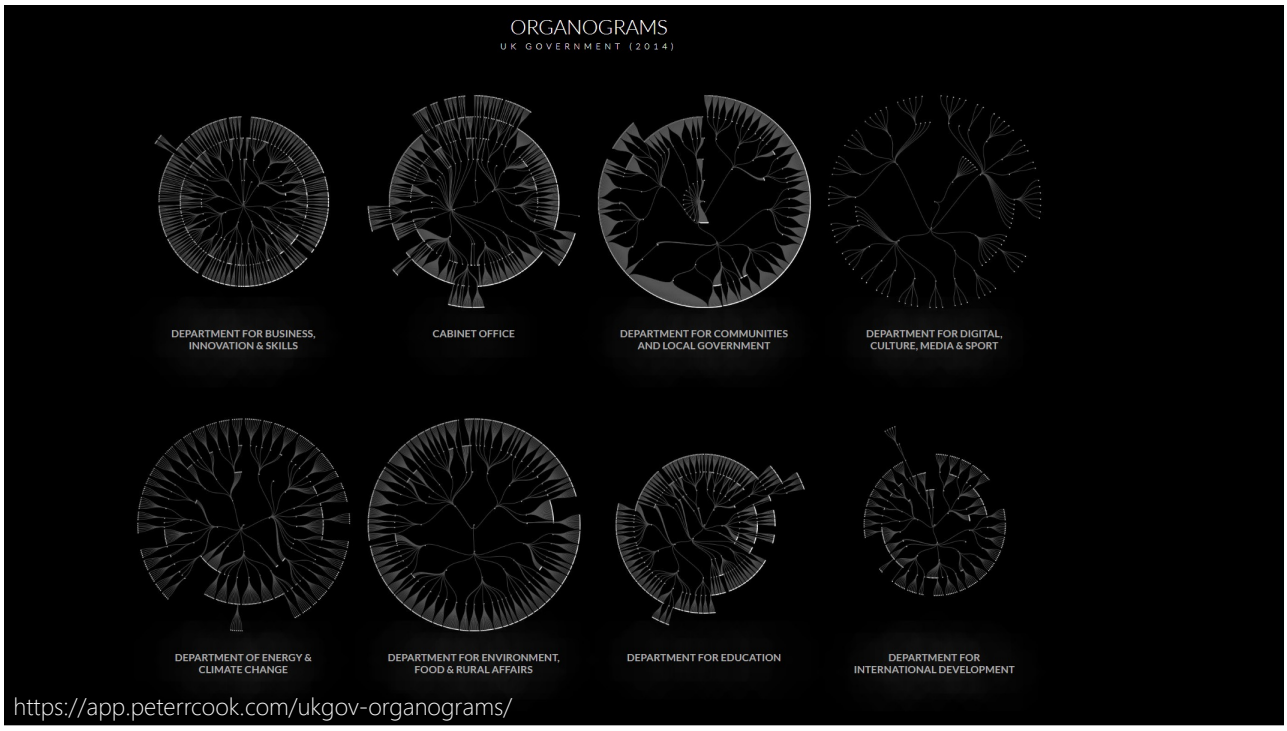
NETWORK VISUALIZATION

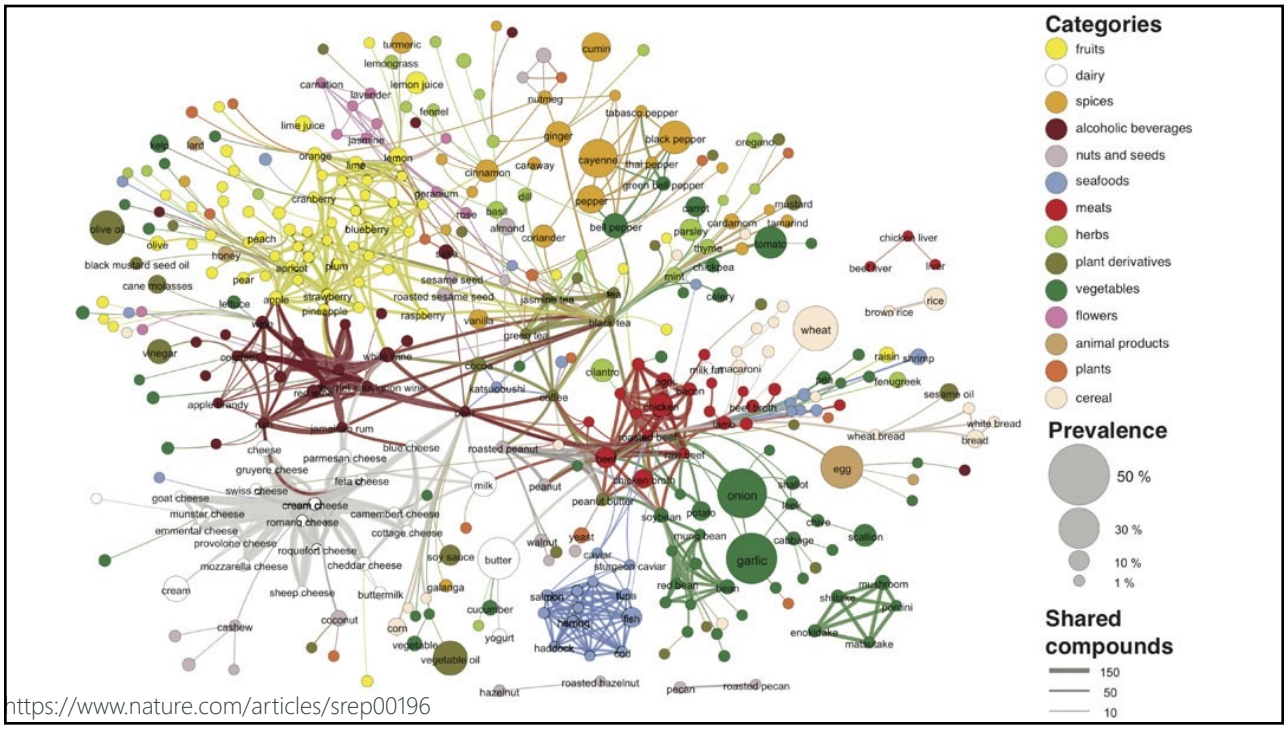
CS 448B | Fall 2023

MANEESH AGRAWALA

1



4



https://www.nature.com/articles/srep00196

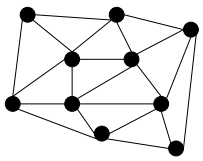
5

GRAPHS AND TREES

Graphs

Model relations among data

Nodes and *edges*

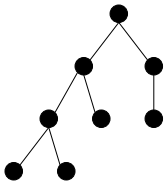


Trees

Graphs with hierarchical structure

Connected graph with N-1 edges

Nodes as *parents* and *children*



6

SPATIAL LAYOUT

Primary concern – positioning of nodes and edges

Often (but not always) goal is to depict structure

- Connectivity, path-following
- Topological distance
- Clustering/grouping
- Ordering (e.g., hierarchy level)

7

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

Attribute-based: properties associated with node or link

Browsing: follow paths in the data

Estimation: summarization and temporal changes

9

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link

Browsing: follow paths in the data

Estimation: summarization and temporal changes

10

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link

- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data

Estimation: summarization and temporal changes

11

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link

- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data

- Find Kermit's friend with first name Beaker and then find Beaker's mentor Bunsen

Estimation: summarization and temporal changes

12

NETWORK ANALYSIS TASKS [Pretorius 2013]

Structure-based: relationships and connectivity

- Find all the friends of friends of Kermit
- Find all the people who are friends of Animal and Gonzo
- Find shortest path between two people: Six degrees of separation

Attribute-based: properties associated with node or link

- Find all friends of Fozzie that are students at Stanford (node property)
- Find all friends of Fozzie that are their family (link property)

Browsing: follow paths in the data

- Find Kermit's friend with first name Beaker and then find Beaker's mentor Bunsen

Estimation: summarization and temporal changes

- How does Miss Piggy's friend group change over the course of a year

13

TODAY

Learning Objectives

1. Techniques for visualizing trees
2. Techniques of laying out graphs
3. Alternative techniques for visualizing node-link data

14

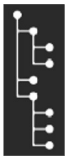
TREE VISUALIZATION

15

COMMON TYPES OF TREE VISUALIZATION

Indentation

Linear list, indentation encodes depth



Node-Link diagrams

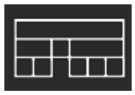
Nodes connected by lines/curves

Enclosure diagrams

Represent hierarchy by enclosure

Layering

Layering and alignment



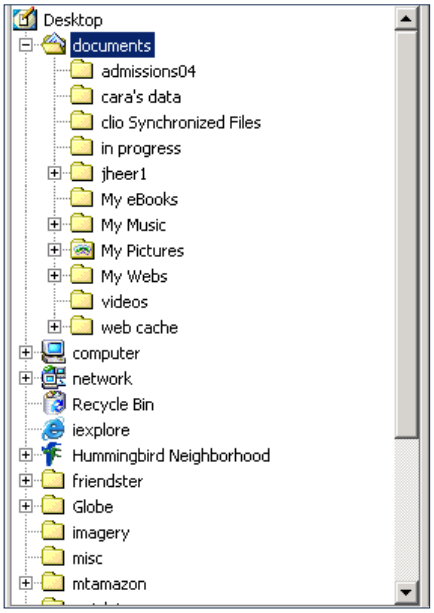
Tree layout is fast: O(n) or O(n log n), enabling real-time layout for interaction

16

INDENTATION

17

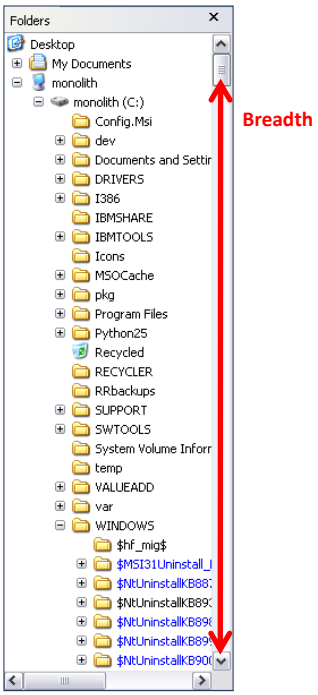
INDENTATION

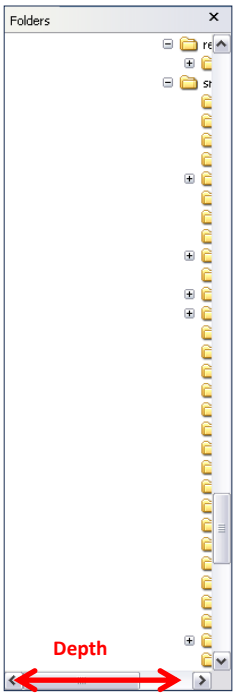


- Items along vertically spaced rows
- Indentation shows parent/child relationships
- Often used in interfaces
- Breadth/depth contend for space
- Often requires scrolling

18

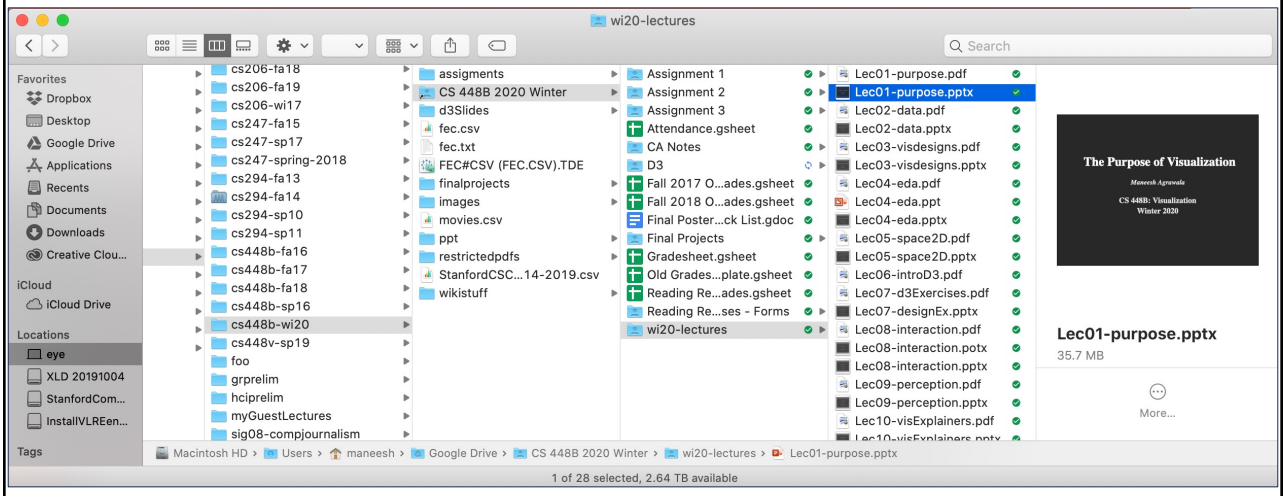
INDENTATION





19

SINGLE-FOCUS ACCORDION LIST

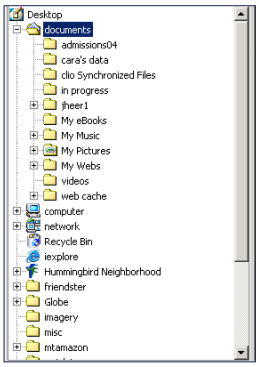


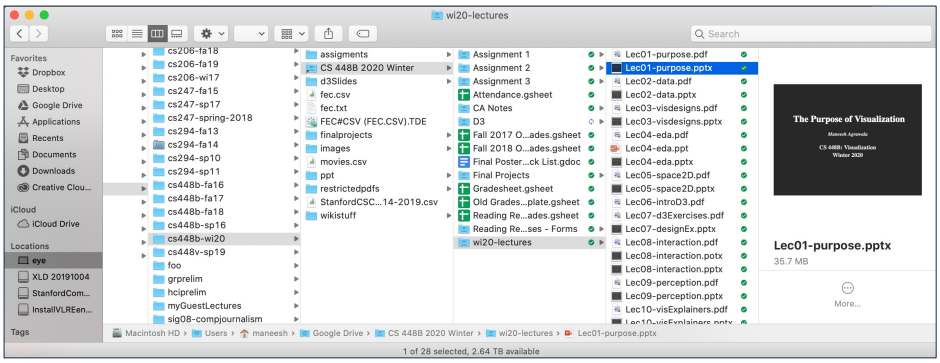
Separate breadth & depth in 2D

Focus on single path at a time

20

WHAT TASKS ARE THESE GOOD FOR?





Benefits

Navigation + Browsing, Parent-child relationships

Disadvantages

Network overview, Estimation, Comparison

21

NODE-LINK DIAGRAMS

22

NODE-LINK DIAGRAMS

Nodes distributed in space, connected by straight or curved lines

Use 2D space to break apart breadth and depth

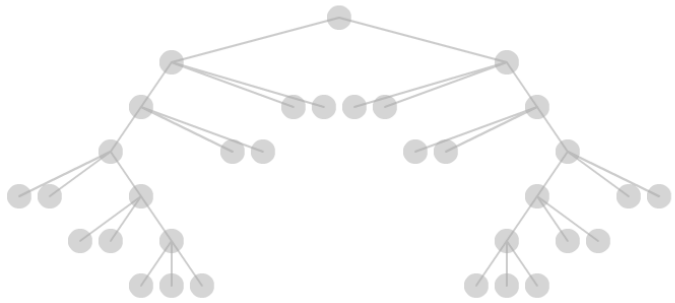
Space used to communicate hierarchical orientation

(e.g., towards *authority* or *generality*)

23

BASIC RECURSIVE LAYOUT

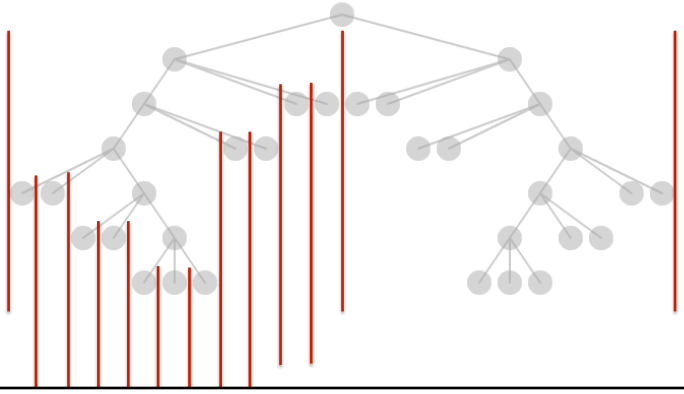
Repeatedly divide space for subtrees by leaf count



24

BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count



25

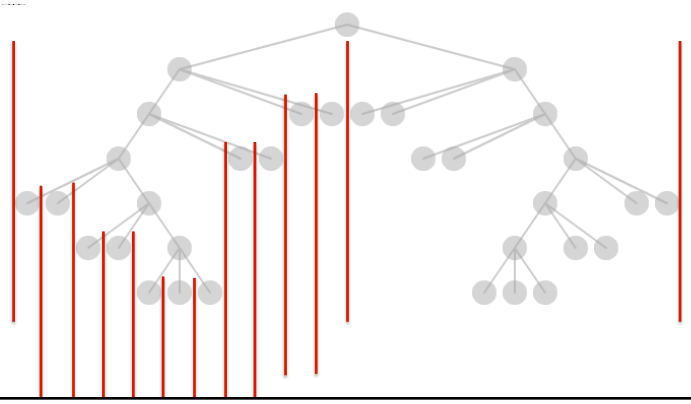
BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count

Breadth of tree along one dimension

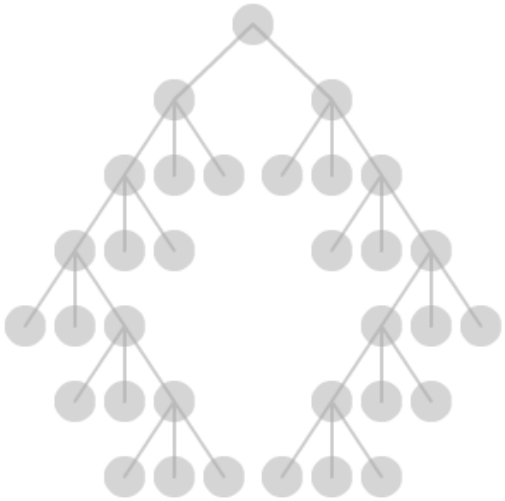
Depth along the other dimension

Problem: Exponential growth of breadth



26

REINGOLD & TILFORD'S "TIDY" LAYOUT



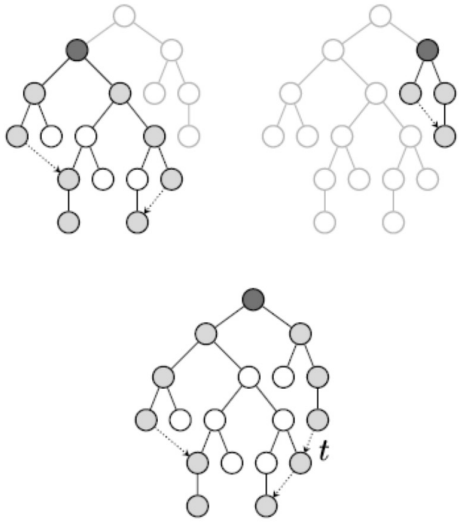
Goal: maximize density and symmetry

Originally for binary trees, extended by Walker to cover general case

Corrected by Buchheim et al. to achieve a linear time algorithm

27

REINGOLD & TILFORD LAYOUT



Design Considerations

Clearly encode depth

No edge crossings

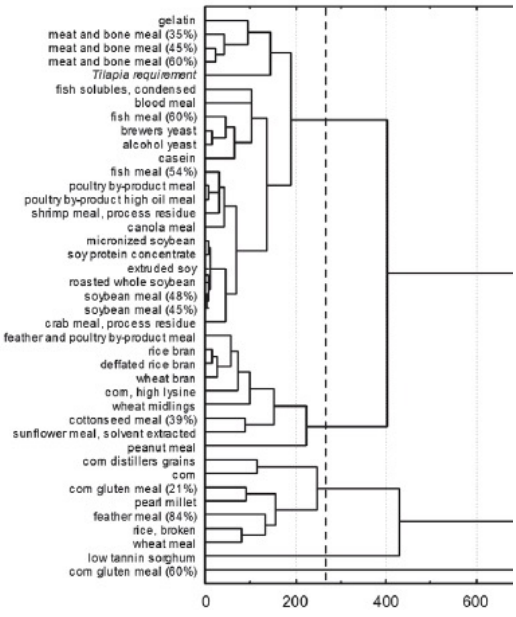
Draw isomorphic subtrees identically (same shape)

Preserve layout ordering and symmetry

Compact space saving layout (don't waste space)

28

CLUSTER DENDROGRAMS



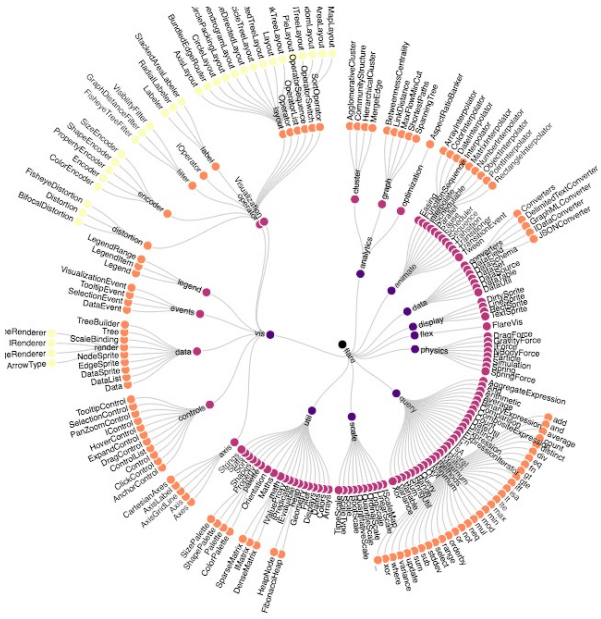
Depicts cluster trees produced by hierarchical clustering algorithms

Leaf nodes arranged in line, internal node depth indicates order/value at which clusters merge

Basic recursive layout with orthogonal two-segment edges

60

RADIAL LAYOUT



Node-link diagram in polar coordinates

Radius encodes depth, root at center

Angular sectors assigned to subtrees (basic recursive approach)

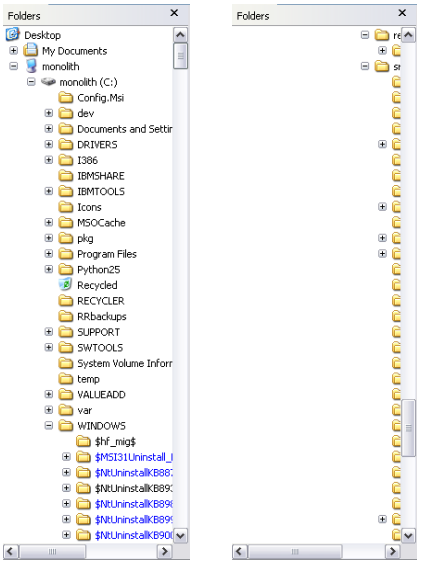
Reingold-Tilford approach can also be applied here

61

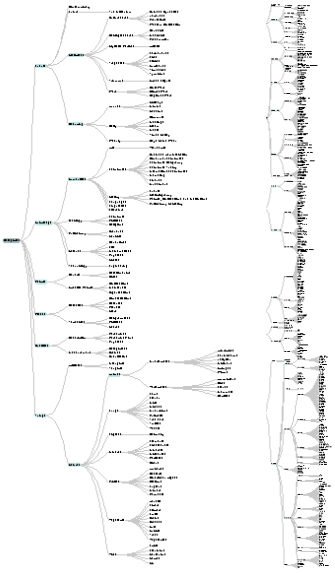
ANALYSIS TASKS: FOCUS + CONTEXT

62

VISUALIZING LARGE HIERARCHIES



Indented Layout



Reingold-Tilford Layout

63

MORE NODES, MORE PROBLEMS…

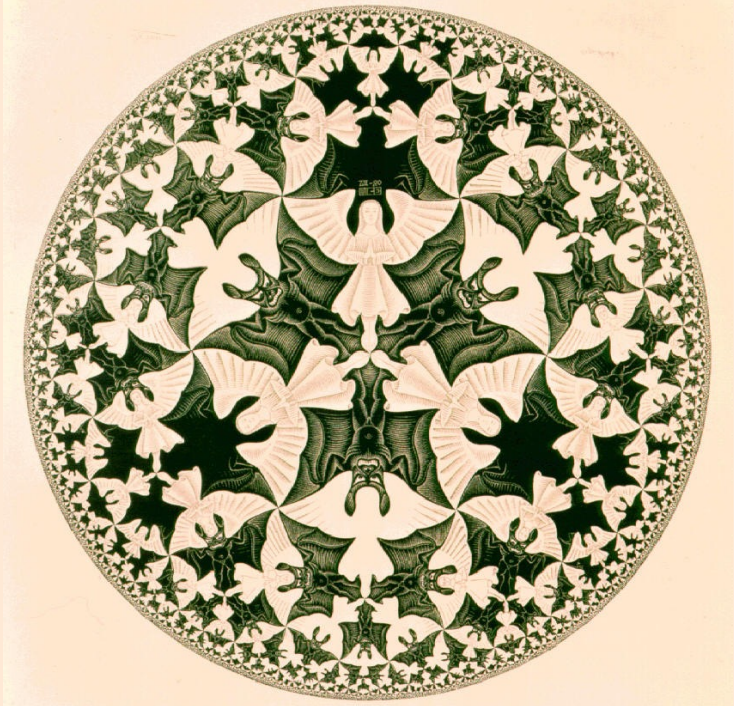
Scale

Tree breadth often grows exponentially
Even with tidier layout, quickly run out of space

Possible solutions

Filtering
Scrolling or Panning
Zooming
Aggregation
Focus+Context

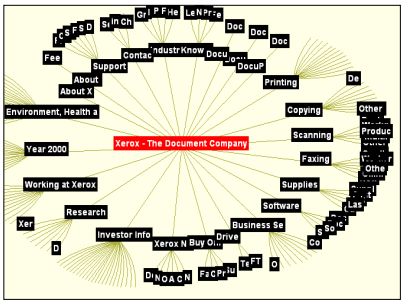
64

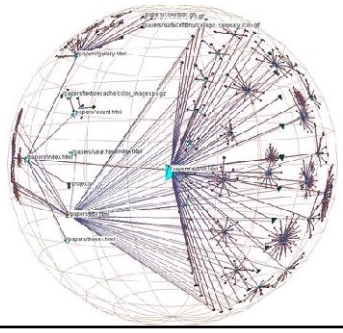


MC Escher, *Circle Limit IV*

65

HYPERBOLIC LAYOUT





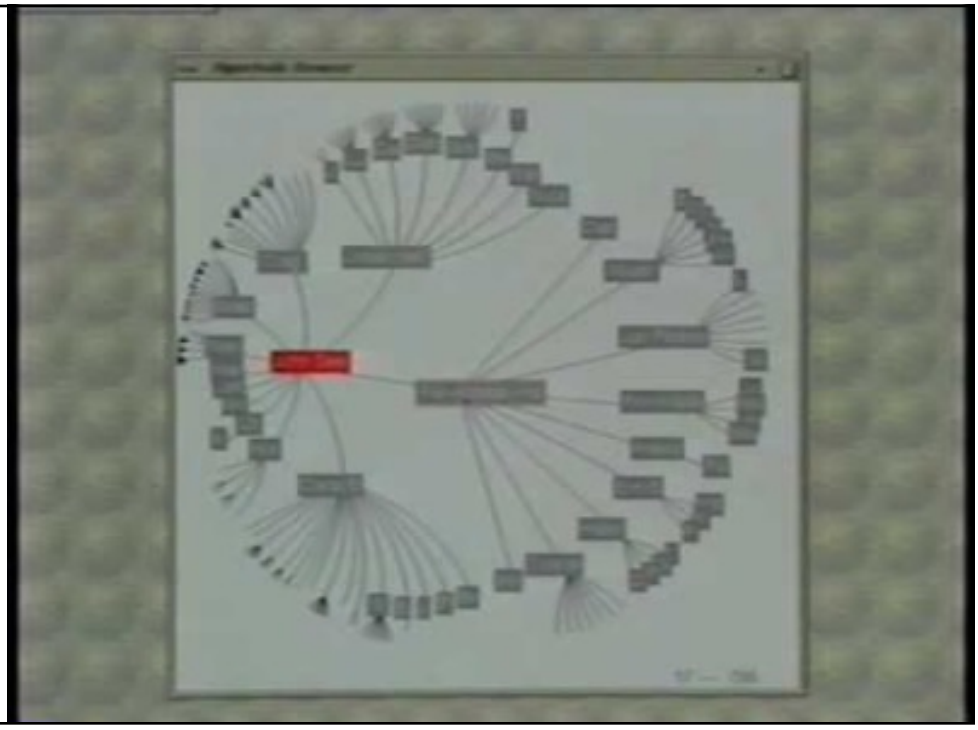
Perform tree layout in hyperbolic space, then project the result on to the Euclidean plane

Why? Like tree breadth, the hyperbolic plane expands exponentially!

Also computable in 3D, projected into a sphere

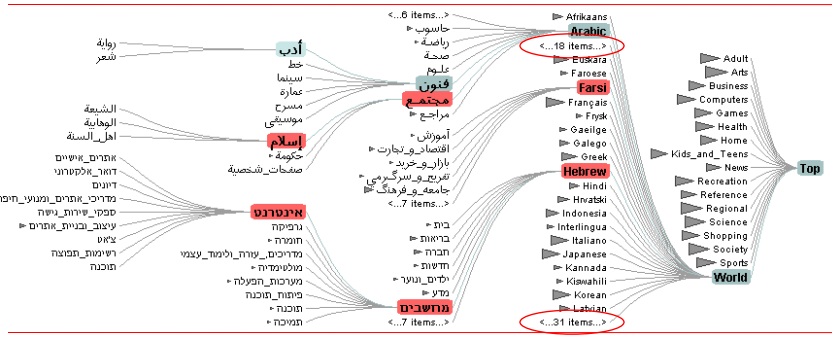
66

HYPERBOLIC LAYOUT [Rao 1995]



67

DEGREE OF INTEREST TREES

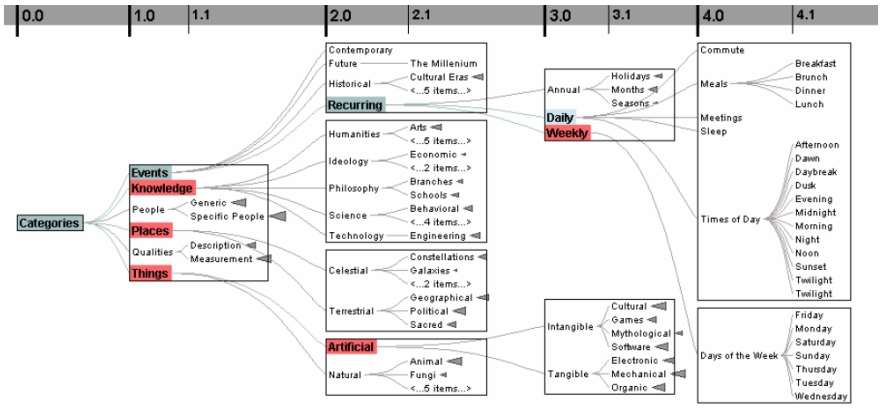


Space-constrained, multi-focal tree layout

https://www.youtube.com/watch?v=RTQ0N4QY0yc

68

DEGREE OF INTEREST TREES



Cull "low interest" nodes on a given depth level until all blocks in level fit within bounds
Attempt to center child blocks under parents

https://www.youtube.com/watch?v=RTQ0N4QY0yc

69

INDENTATION & NODE-LINK DIAGRAMS

Encode structure in **2D space** (breadth/depth)

Benefits

Clearly depicts node relationships / structure

Structure-based or browsing tasks

Problems

Even with tidy layout, quickly run out of space

Missing

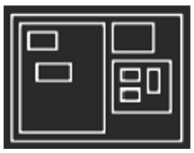
Attribute-based encodings

70

ENCLOSURE

71

ENCLOSURE DIAGRAMS



Encode structure using **spatial enclosure**
Popularly known as **treemaps**

Benefits

Provides a single view of an entire tree
Easier to spot large/small nodes

Problems

Difficult to accurately read structure/depth

72

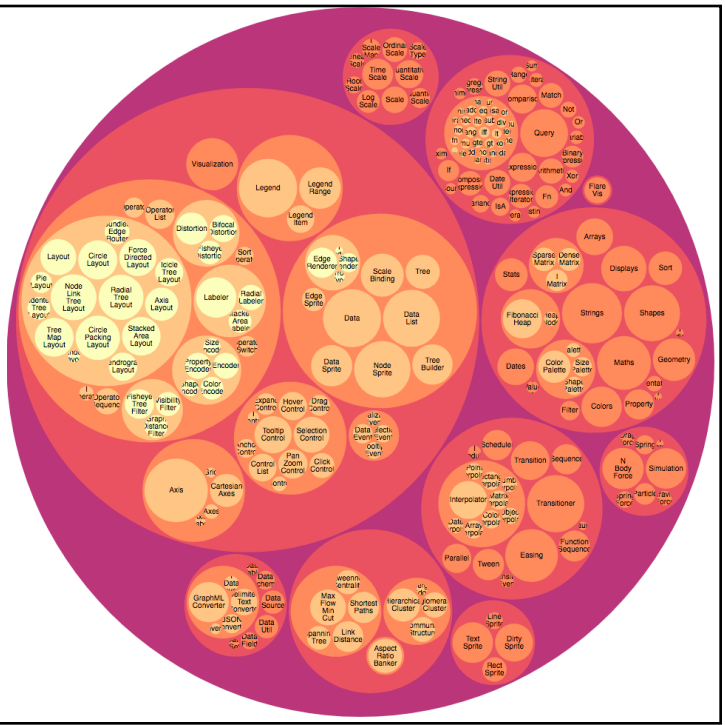
CIRCLE PACKING LAYOUT

Nodes represented as sized circles

Nesting to show parent-child relationships

Problems?

- Inefficient use of space
- Parent size misleading?



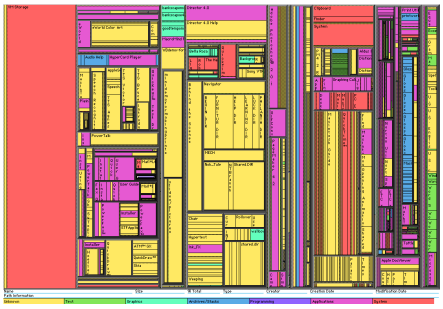
73

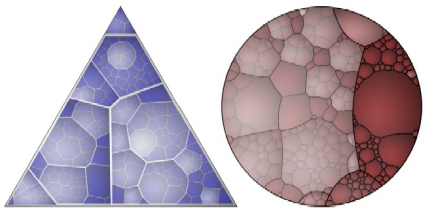
TREEMAPS

Hierarchy visualization that emphasizes values of nodes via area encoding

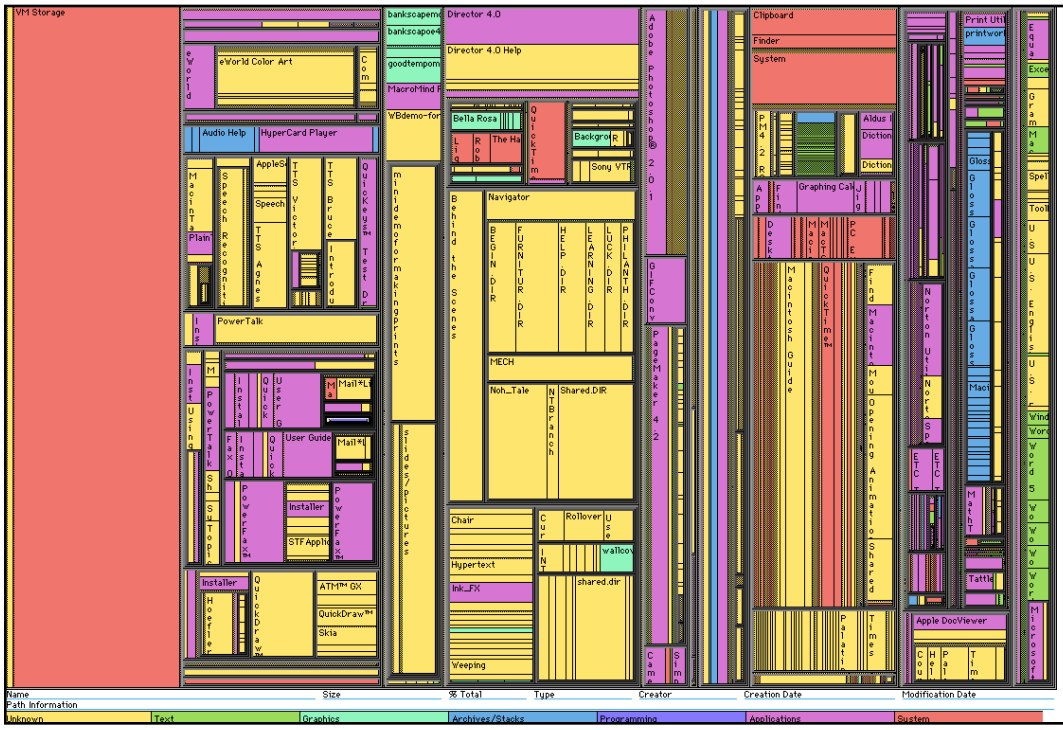
Partition 2D space such that leaf nodes have sizes proportional to data values

First algorithms designed to show file sizes on a hard drive [Shneiderman 1990]





75



Slice & Dice
layout: Alternate horizontal / vertical partitions

76

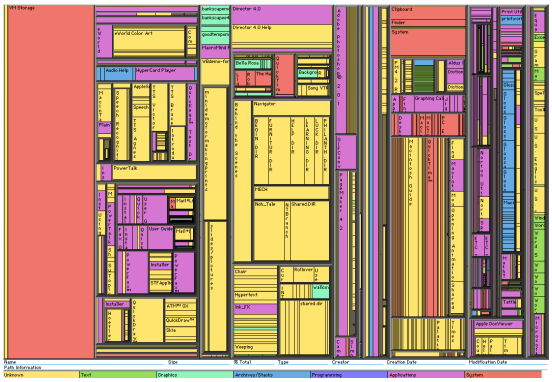
Squarified
layout: Try to produce square (1:1) aspect ratios

77

SQUARIFIED TREEMAPS [Bruls 2000]

Slice & Dice layout suffers from extreme aspect ratios. How might we do better?

Squarified layout: greedy optimization with objective of square rectangles. Slice/dice within siblings; alternate whenever ratio worsens



78

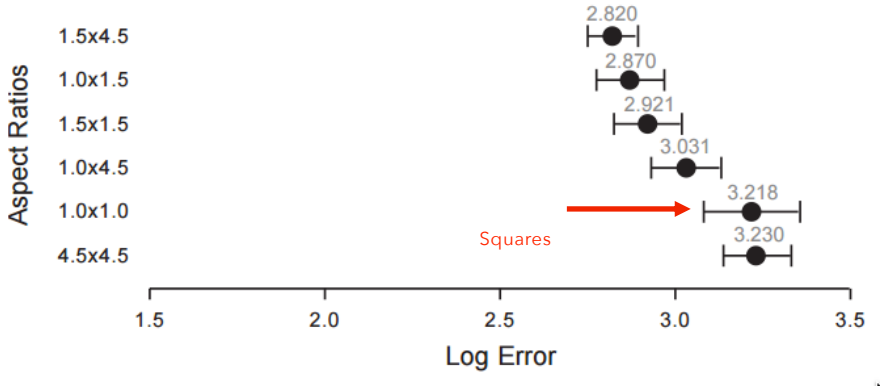
WHY SQUARES?

Posited Benefits of 1:1 Aspect Ratios

1. Minimize perimeter, reducing border ink
   Mathematically true!
2. Easier to select with a mouse cursor.
   Validated by empirical research & Fitt's Law!
3. Similar aspect ratios are easier to compare.
   Seems intuitive, but is this true?

79

COMPARISON: ERROR VS. ASPECT RATIO



Study by Kong et al. 2010

Comparison of squares has higher error! Comparison of extreme ratios even worse

Perhaps squarify works well because it fails to meet its objective?

80

WHY SQUARES?

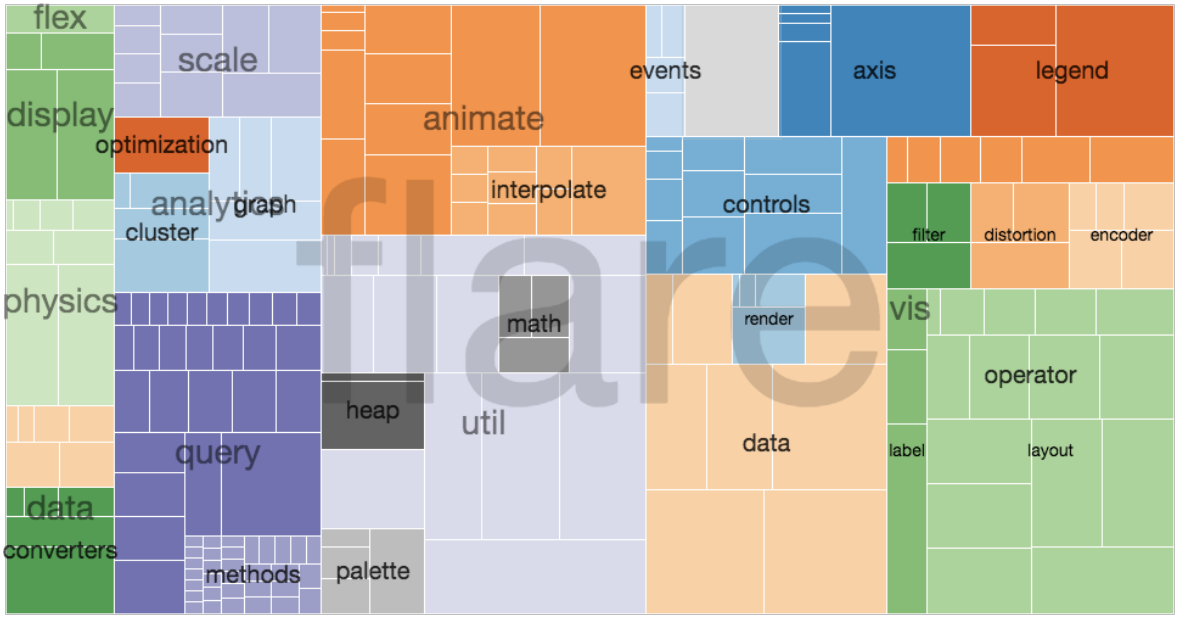
Posited Benefits of 1:1 Aspect Ratios

1. Minimize perimeter, reducing border ink
   Mathematically true!
2. Easier to select with a mouse cursor.
   Validated by empirical research & Fitt's Law!
3. Similar aspect ratios are easier to compare.
   ~~*Seems intuitive, but is this true?*~~
   Extreme ratios & squares-only more inaccurate.
   Balanced ratios better? Target golden ratio?

81

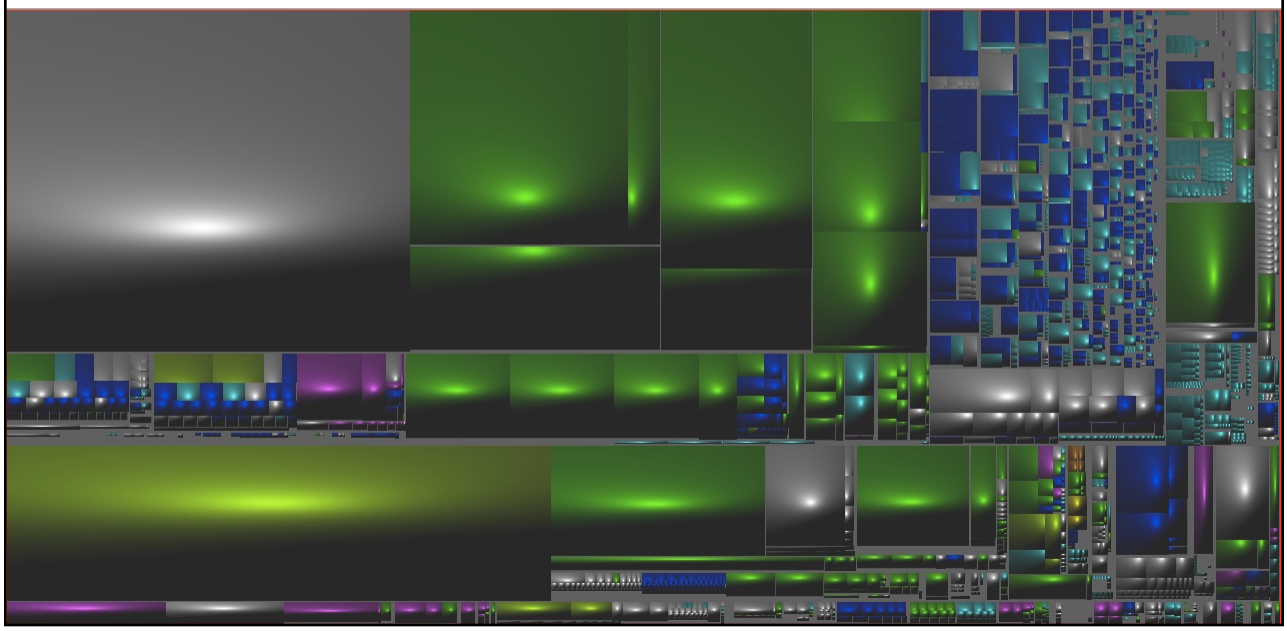
INTERACTIVE EXAMPLE

https://vega.github.io/vega/examples/treemap/



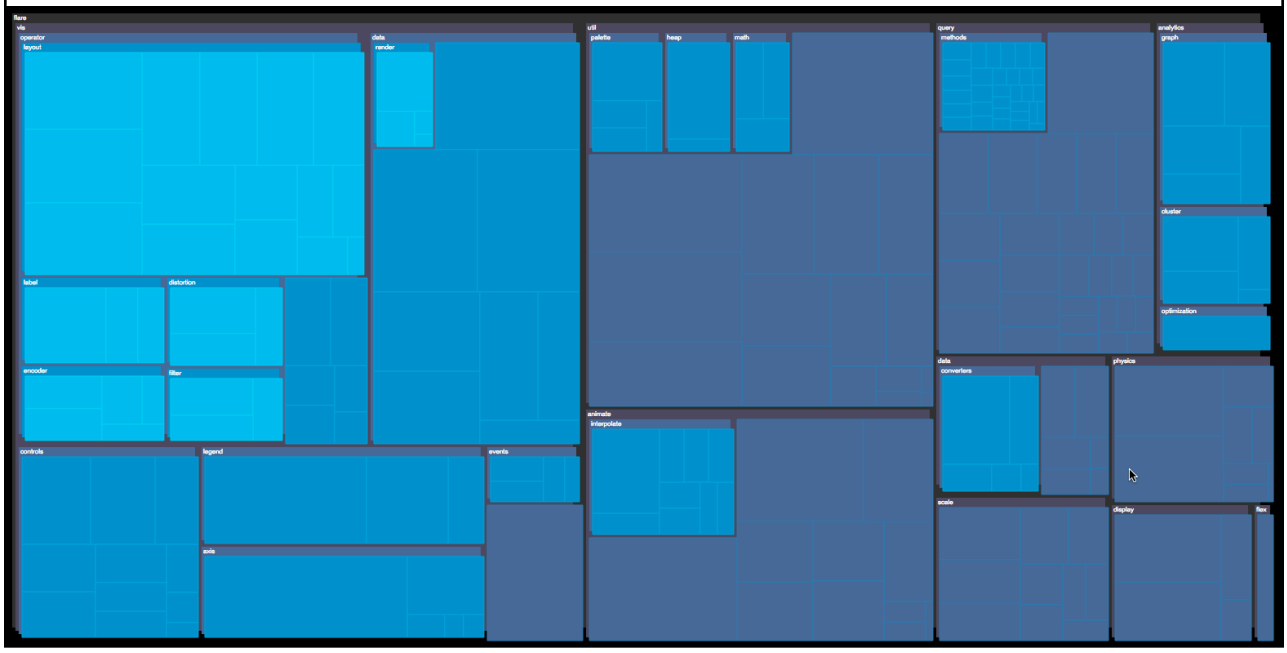
82

CUSHION TREEMAPS [van Wijk 1999]



85

CASCADED TREEMAPS [Lü 2008]

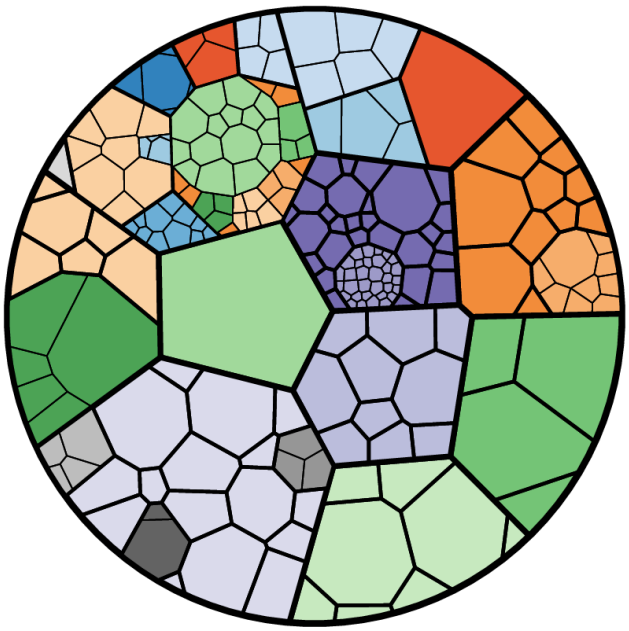


86

VORONOI TREEMAPS [Balzer 2005]

Treemaps with arbitrary polygonal shapes and boundary

Uses iterative, weighted Voronoi tessellations to generate cells with value-proportional areas

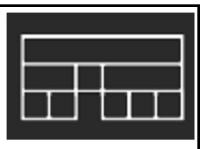


87

LAYERING

89

LAYERED DIAGRAMS



Signify tree structure using

- Layering
- Adjacency
- Alignment

Involves recursive sub-division of space

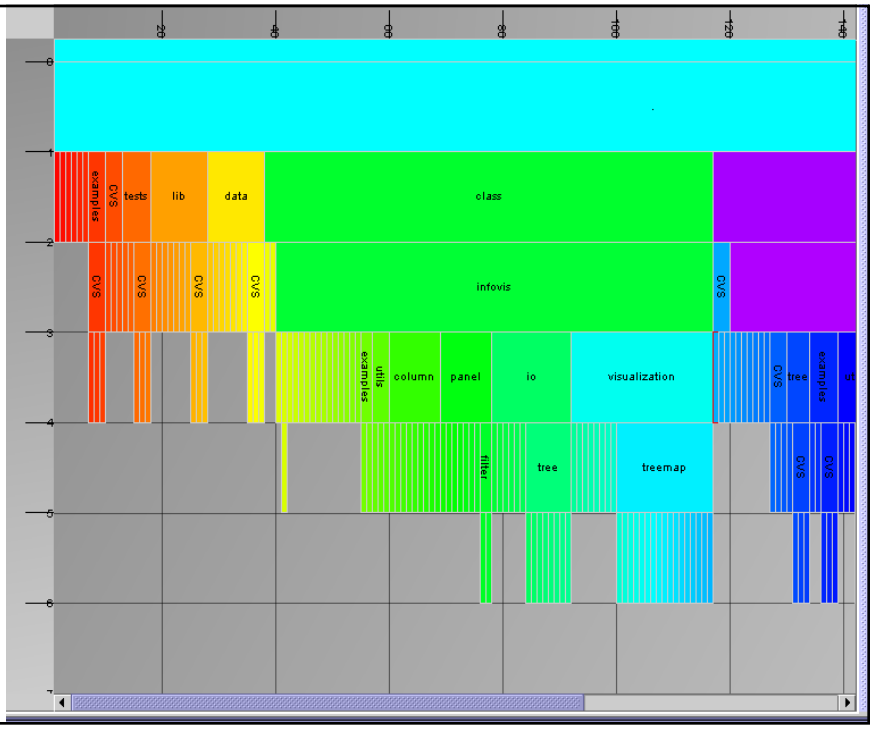
Leaf nodes may be sized by value, parent size visualizes sum of descendant leaf values

90

ICICLE TREES

Parent nodes get a larger layer area, (e.g. horizontal extent)

Child levels are layered, constrained to parent's extent

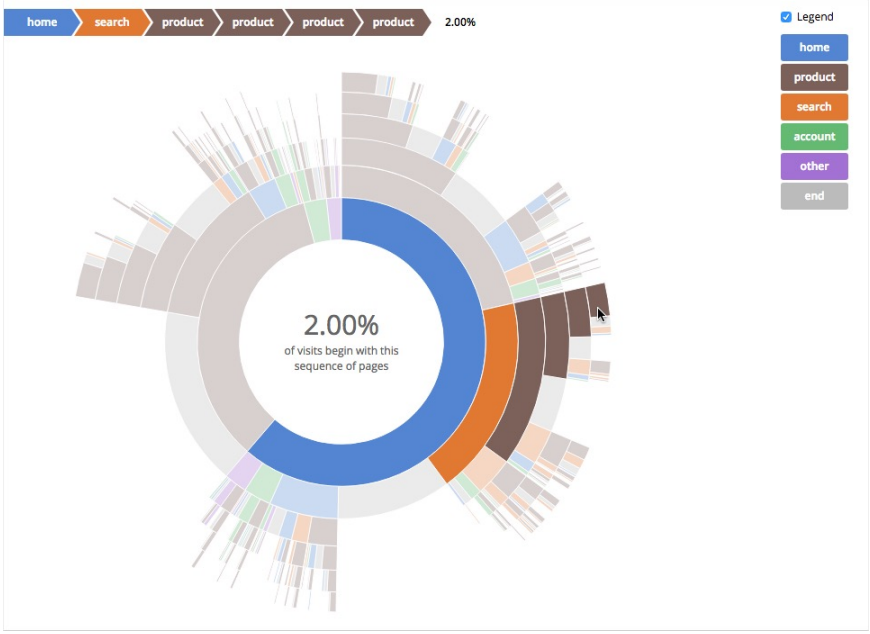


91

SUNBURST TREES

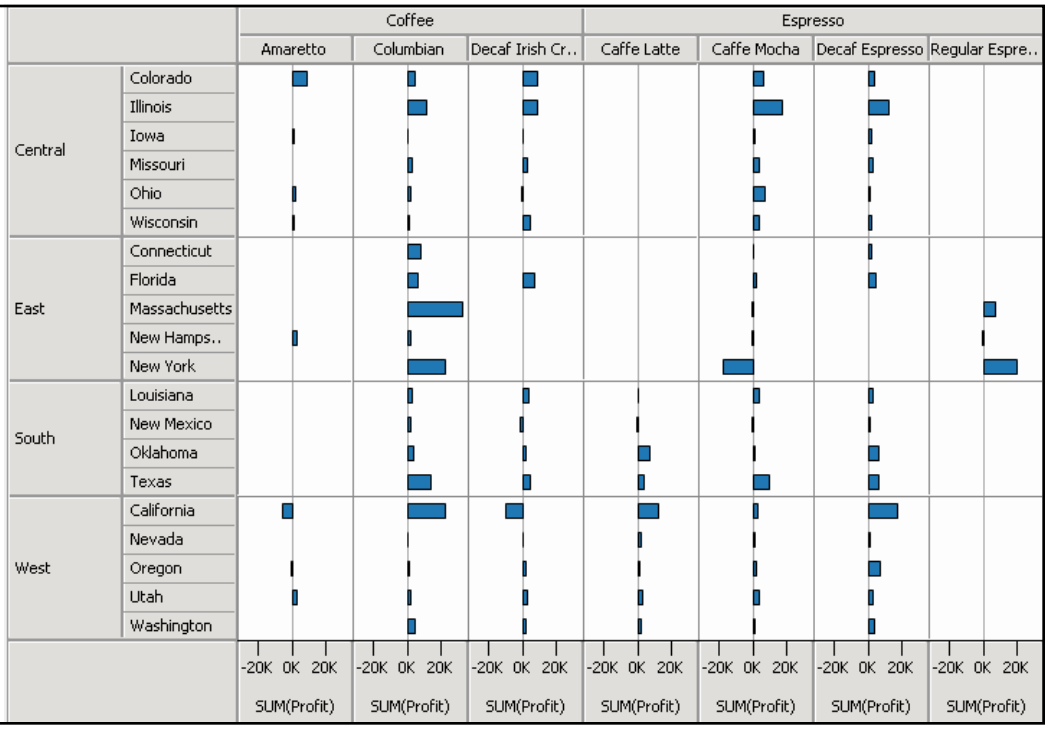
Parent nodes get a larger layer area, (e.g. angular extent)

Child levels are layered, constrained to parent's extent



92

ANOTHER USE OF LAYERED TREES



93

ANNOUNCEMENTS

95

FINAL PROJECT

Design Review Nov 27 and 29

Data analysis/explainer

Analyze dataset in depth & make a visual explainer

Deliverables

An article with multiple different interactive visualizations
Short video (2 min) demoing and explaining the project

Schedule

Project proposal: Mon 11/6
Design Review and Feedback: 9th week of quarter, 11/27 and 11/29
Final code and video: Sun 12/10 8pm

Grading

Groups of up to 3 people, graded individually
Clearly report responsibilities of each member

96

FINAL PROJECT GUIDELINES

Consider the audience

Your visual explainer should be of interest to a group of people beyond your immediate circle (an explainer about your own Spotify data unlikely be of interest to others you don't know)

Pick relatively less explored topics/datasets

Do some research on what has already been done for the topic/dataset(s)

Certain data like songs (e.g. Spotify) or movies (e.g. IMDB) are already well analyzed and should be avoided, unless you want to try to take a very different angle or use innovative analysis methods

Develop a narrative

In the early stages of the analysis process, try to uncover patterns to help you form and shape a narrative through-line for the explainer

97

FINAL PROJECT GUIDELINES

Design visualization interactions

Choose base visualizations that can support a high level of interactivity
- Bubble charts, tree maps, and word clouds typically aren't the most effective choices

Design interactive features that would enable viewers to interact with the data in a way that strengthens your narrative
- Tooltip is typically not enough interaction

Draw inspiration from sites like the New York Times and the Pudding

98

NODE-LINK GRAPH LAYOUT

99

NODE-LINK GRAPH VISUALIZATION

Nodes connected by lines/curves

Sugiyama-Style Layout - arranged by depth
Force-Directed Layout - physical simulation
Attribute-Driven Layout - arranged by value
Constraint-Based Layout – optimization
Arc Diagrams - aligned layout

100

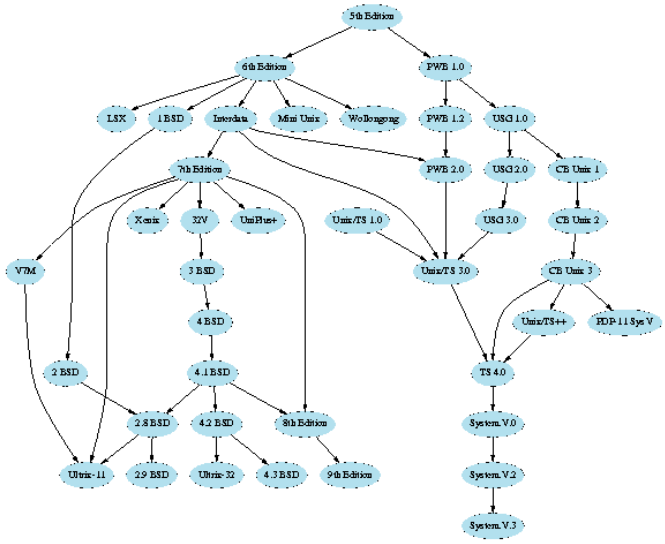
SUGIYAMA-STYLE LAYOUT

101

SUGIYAMA-STYLE GRAPH LAYOUT

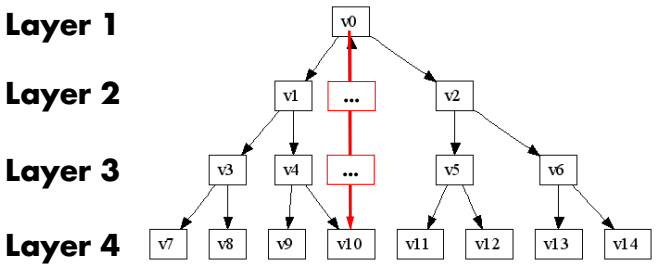
Evolution of the UNIX operating system

Hierarchical layering based on descent



102

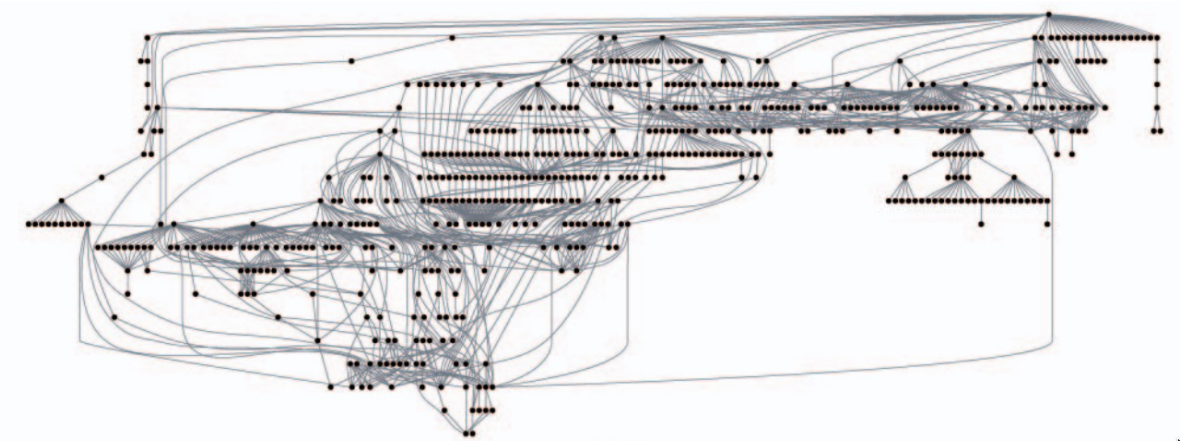
SUGIYAMA-STYLE GRAPH LAYOUT



Reverse some edges to remove cycles
Assign nodes to hierarchy layers → Longest path layering
  Create dummy nodes to "fill in" missing layers
Arrange nodes within layer, minimize edge crossings
  Route edges – layout splines if needed

103

PRODUCES HIERARCHICAL LAYOUT



Sugiyama-style layout emphasizes hierarchy

However, cycles in the graph may not be as apparent, and hierarchy may mislead

Long edges can impede perception of proximity

104