NETWORK VISUALIZATION

CS 448B | Fall 2025

MANEESH AGRAWALA

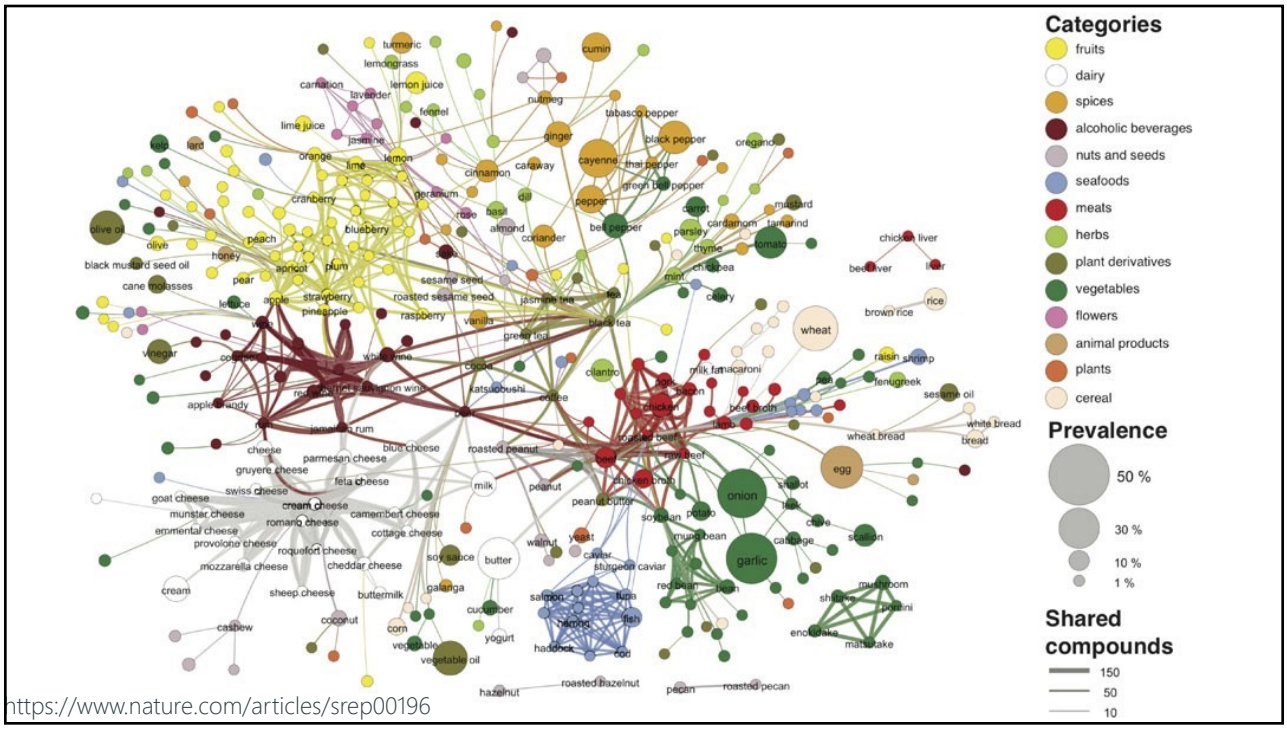
1

READING RESPONSE: QUESTIONS/THOUGHTS

One of the big questions that I had is, even if we add more control to our animations, more pauses, and better visual cues **how do we know if they are granular enough, stopping to highlight the "right" moments, and in the sweet spot to avoid cognitive overload,** especially **when we as subject matter experts might not catch these issues ourselves**?

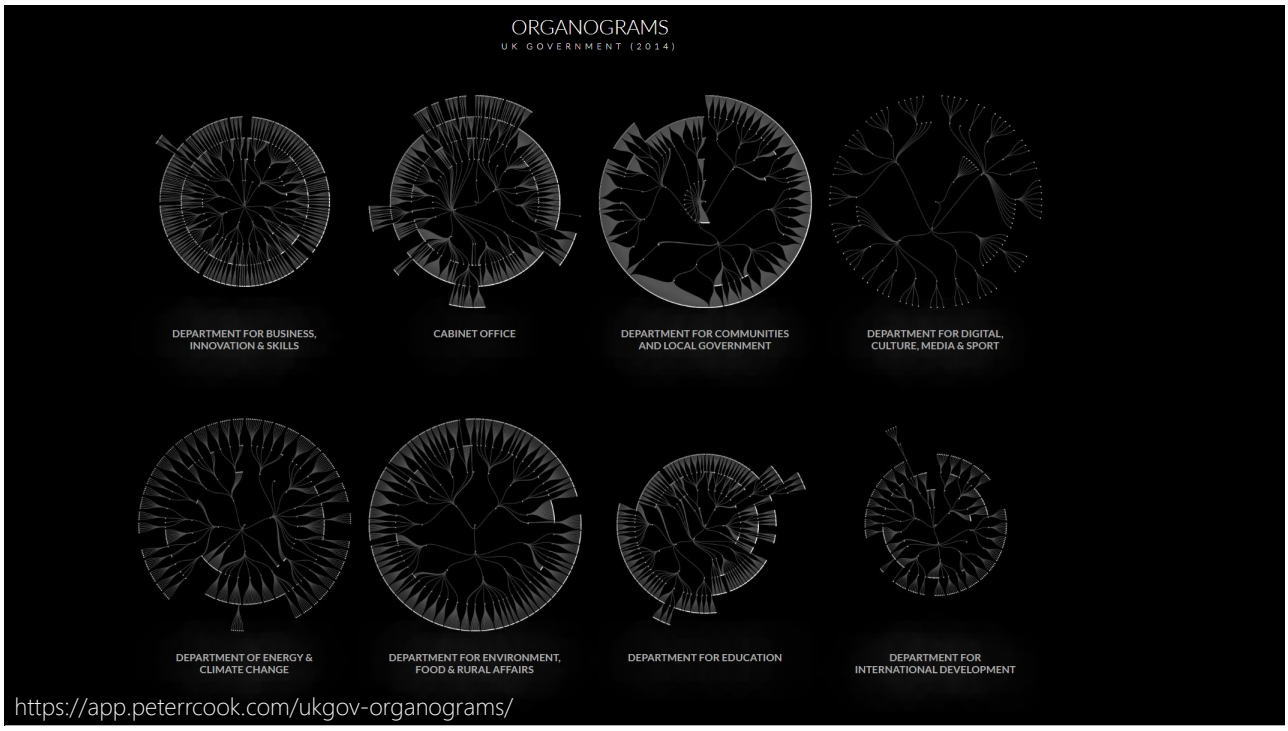
- Are animations validated through user testing just like apps, websites, or products are in other pockets of the HCI world?
- Is A/B or user testing a routine practice among people who design visualizations?

2



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

4



https://app.peterrcook.com/ukgov-organograms/

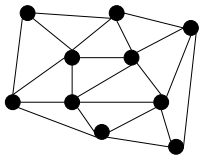
5

GRAPHS AND TREES

Graphs

Model relations among data

Nodes and *edges*

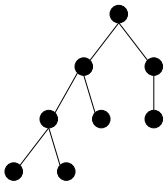


Trees

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

TODAY

Learning Objectives

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

9

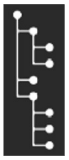
TREE VISUALIZATION

10

COMMON TYPES OF TREE VISUALIZATION

Indentation

Linear list, indentation encodes depth



Node-Link diagrams

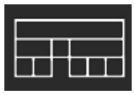
Nodes connected by lines/curves

Enclosure diagrams

Represent hierarchy by enclosure

Layering

Layering and alignment



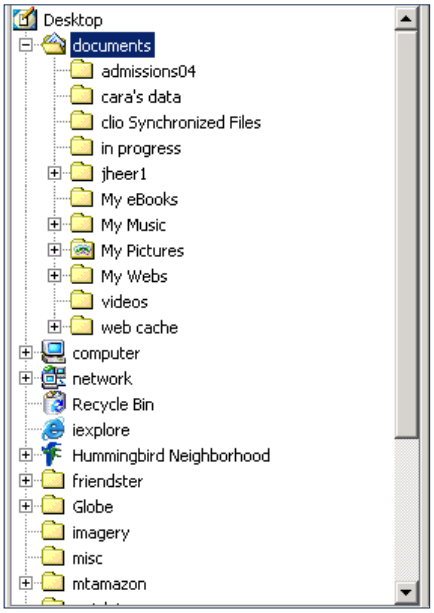
Tree layout is fast: O(n) or O(n log n), enabling real-time layout for interaction

11

INDENTATION

12

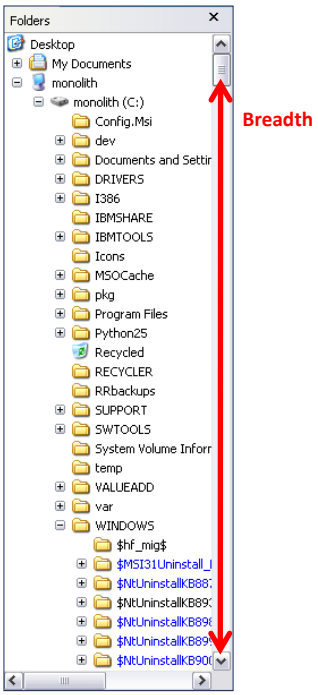
INDENTATION

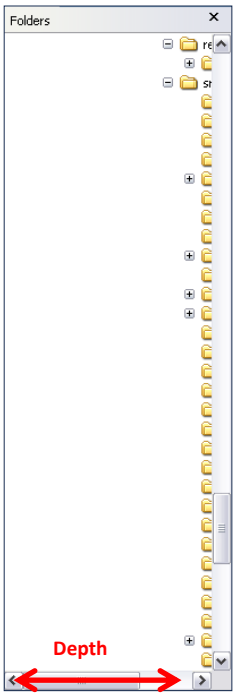


- Items along vertically spaced rows
- Indentation shows parent/child relationships
- Often used in interfaces
- Breadth/depth contend for space
- Often requires scrolling

13

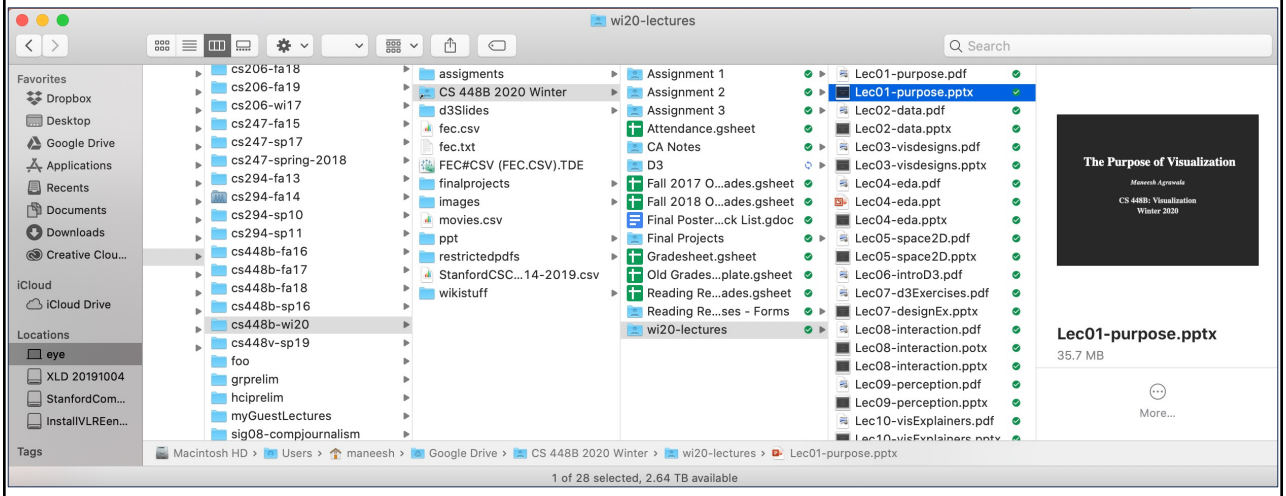
INDENTATION





14

SINGLE-FOCUS ACCORDION LIST

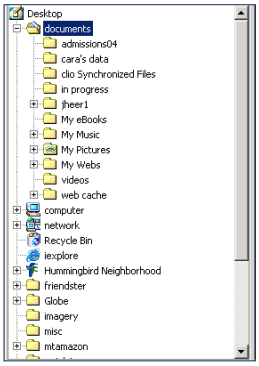


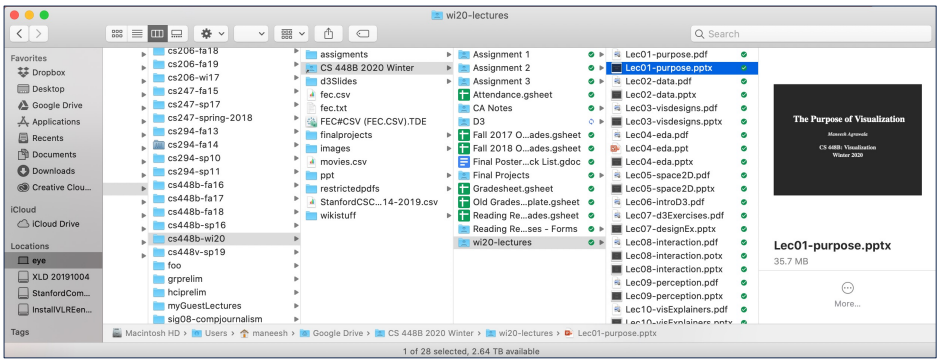
Separate breadth & depth in 2D

Focus on single path at a time

15

WHAT TASKS ARE THESE GOOD FOR?





Benefits

Navigation + Browsing, Parent-child relationships

Disadvantages

Network overview, Estimation, Comparison

16

NODE-LINK DIAGRAMS

17

NODE-LINK DIAGRAMS

Nodes distributed in space, connected by straight or curved lines

Use 2D space to break apart breadth and depth

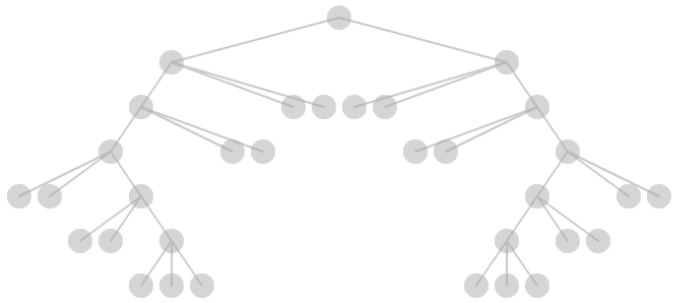
Space used to communicate hierarchical orientation

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

18

BASIC RECURSIVE LAYOUT

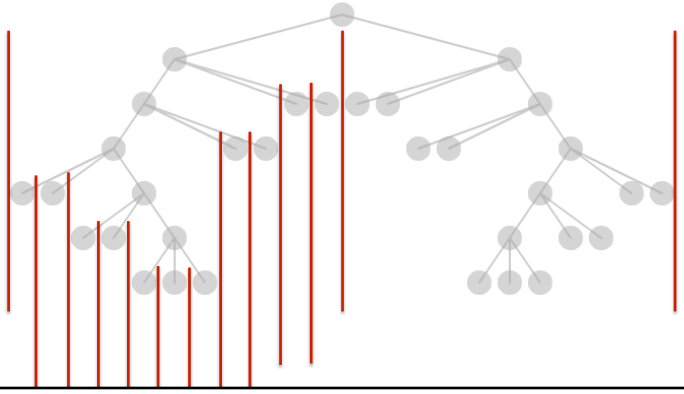
Repeatedly divide space for subtrees by leaf count



19

BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count



20

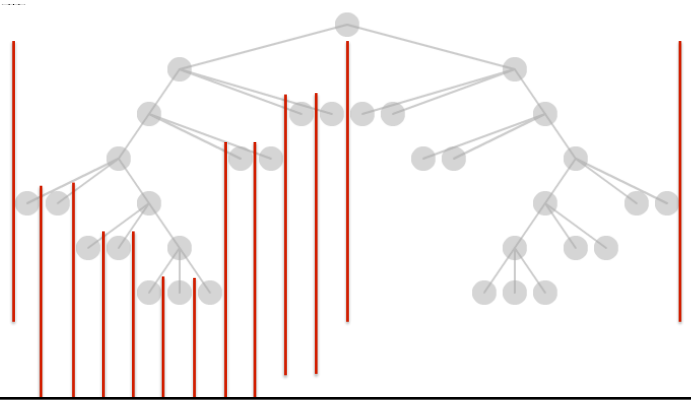
BASIC RECURSIVE LAYOUT

Repeatedly divide space for subtrees by leaf count

Breadth of tree along one dimension

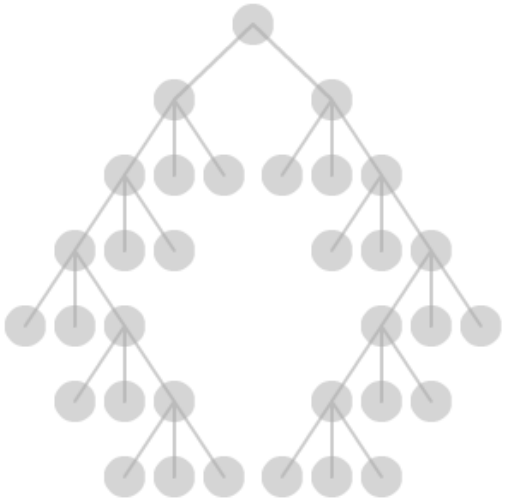
Depth along the other dimension

Problem: Exponential growth of breadth



21

REINGOLD & TILFORD'S "TIDY" LAYOUT



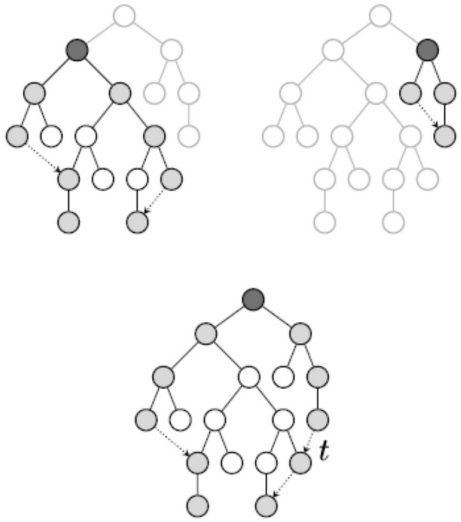
Goal: maximize density and symmetry

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

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

22

REINGOLD & TILFORD LAYOUT



Design Considerations

Clearly encode depth

No edge crossings

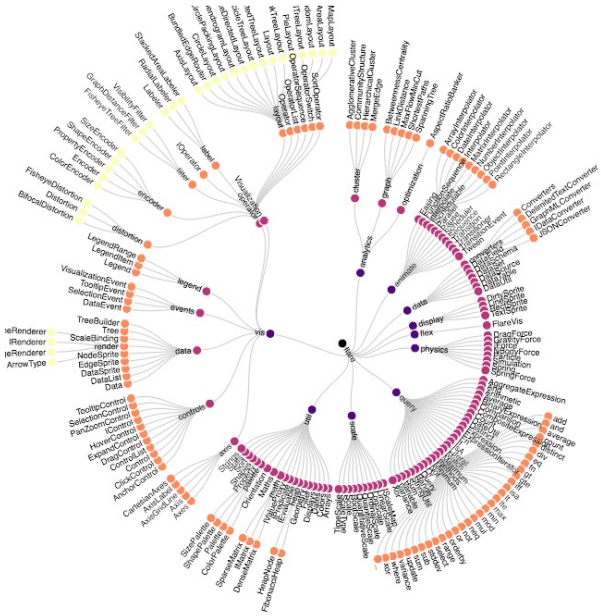
Draw isomorphic subtrees using same shape

Preserve layout ordering and symmetry

Compact layout (doesn't waste space)

23

RADIAL LAYOUT



Node-link diagram in polar coordinates

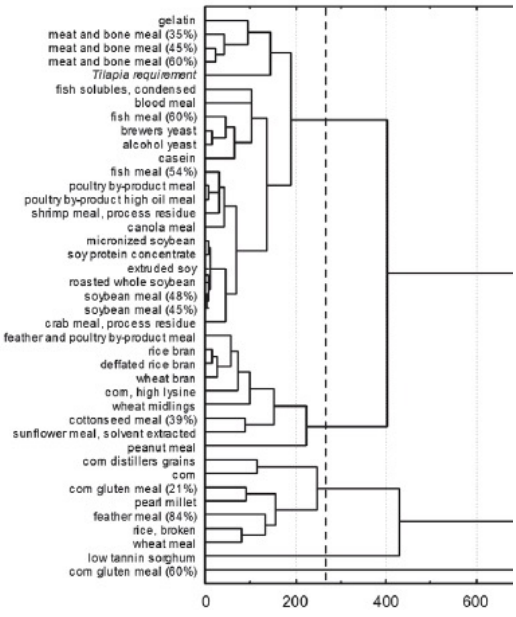
Radius encodes depth, root at center

Angular sectors assigned to subtrees (basic recursive approach)

Reingold-Tilford approach can also be applied here

55

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

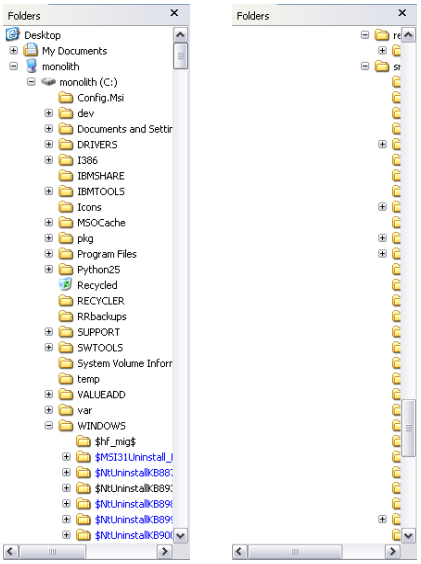
Apply basic recursive layout with orthogonal two-segment edges

56

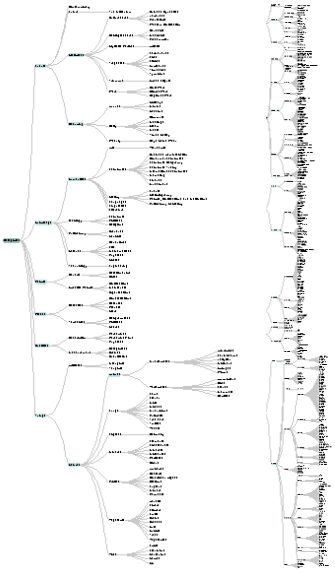
ANALYSIS TASKS: FOCUS + CONTEXT

57

VISUALIZING LARGE HIERARCHIES



Indented Layout



Reingold-Tilford Layout

58

MORE NODES, MORE PROBLEMS…

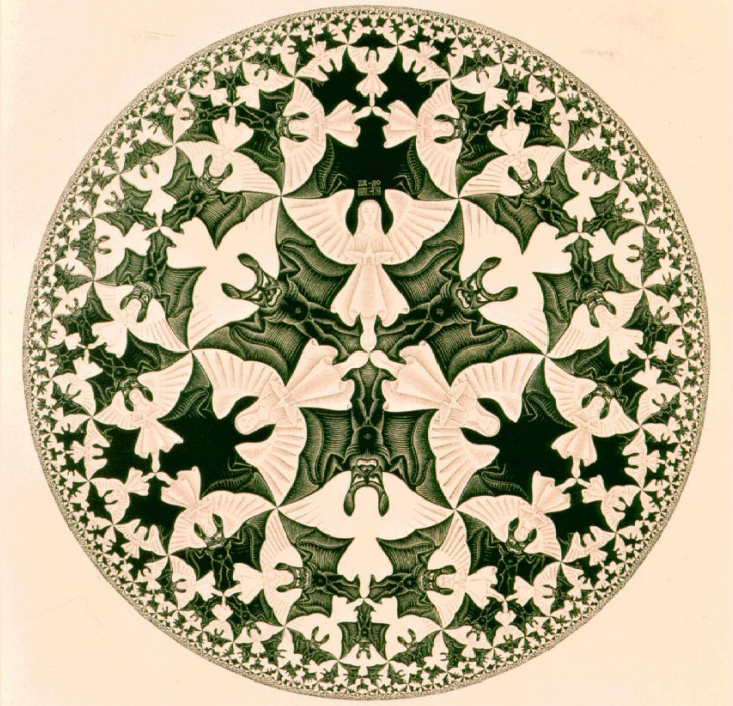
Scale

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

Possible solutions

Filtering
Scrolling or Panning
Zooming
Aggregation
Focus+Context

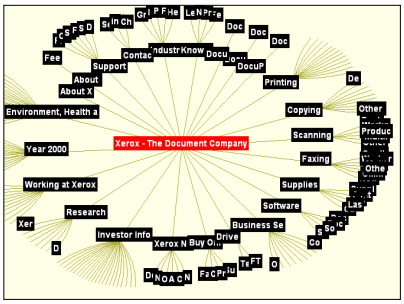
59

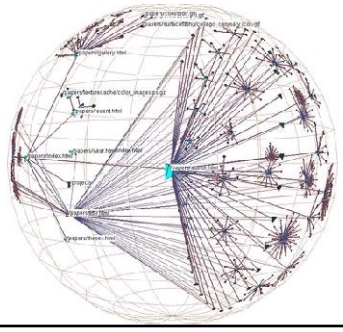


MC Escher, *Circle Limit IV*

60

HYPERBOLIC LAYOUT





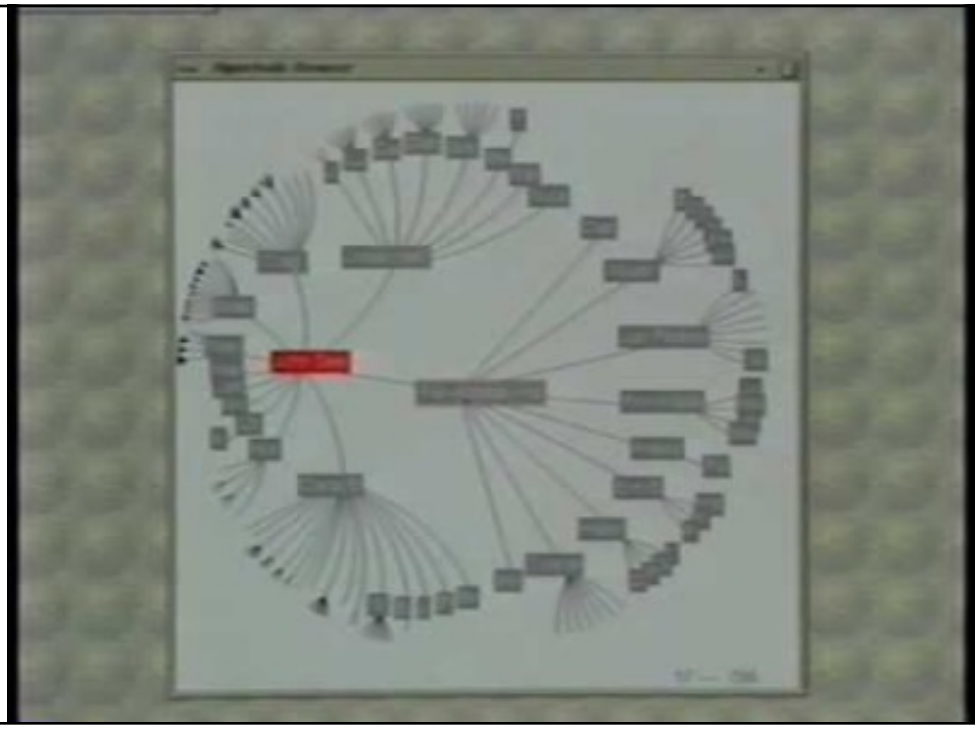
Perform tree layout in hyperbolic space, then project the result on to the Euclidean plane

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

Also computable in 3D, projected into a sphere

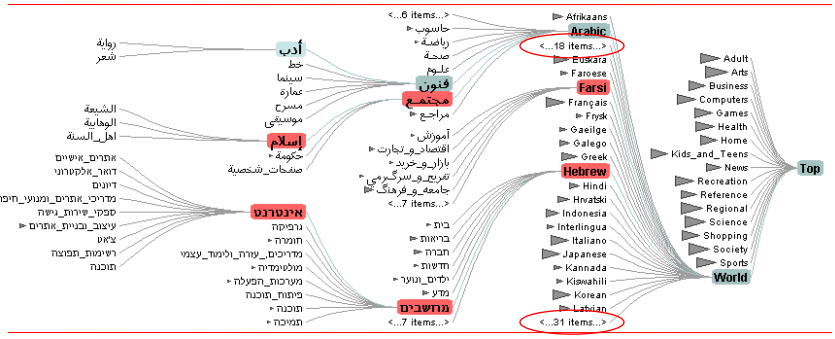
61

HYPERBOLIC LAYOUT [Rao 1995]



62

DEGREE OF INTEREST TREES

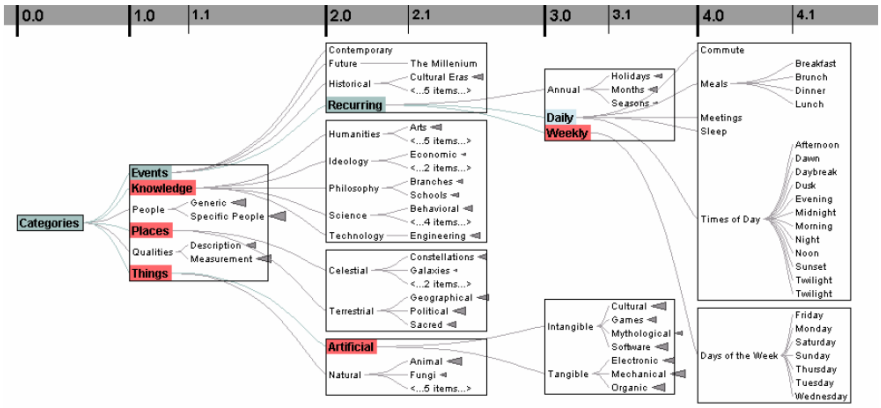


Space-constrained, multi-focal tree layout

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

63

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

64

65

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

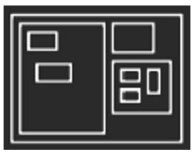
Attribute-based encodings

66

ENCLOSURE

67

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

68

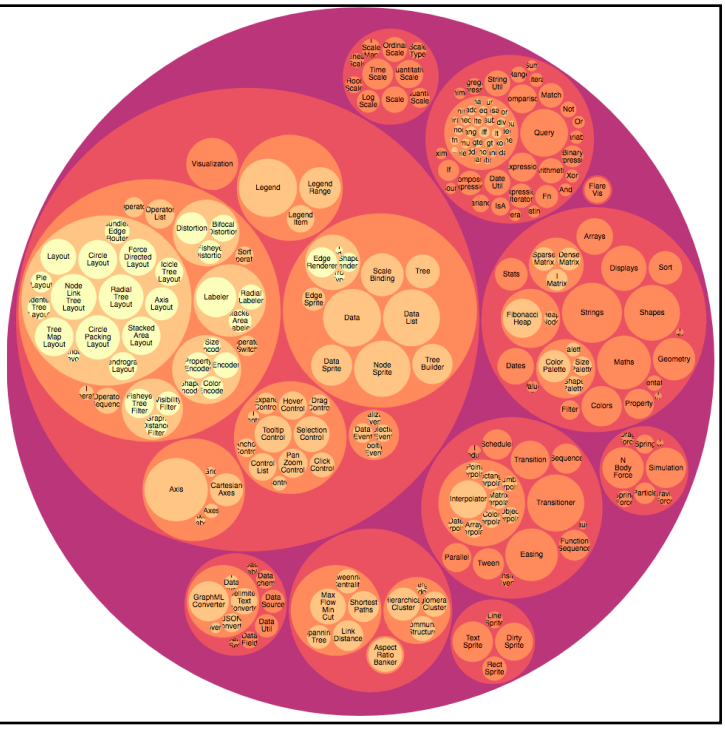
CIRCLE PACKING LAYOUT

Nodes represented as sized circles

Nesting to show parent-child relationships

Problems?

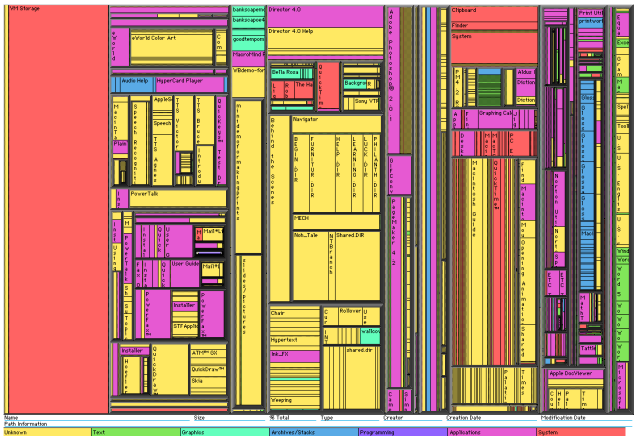
- Inefficient use of space
- Parent size misleading?



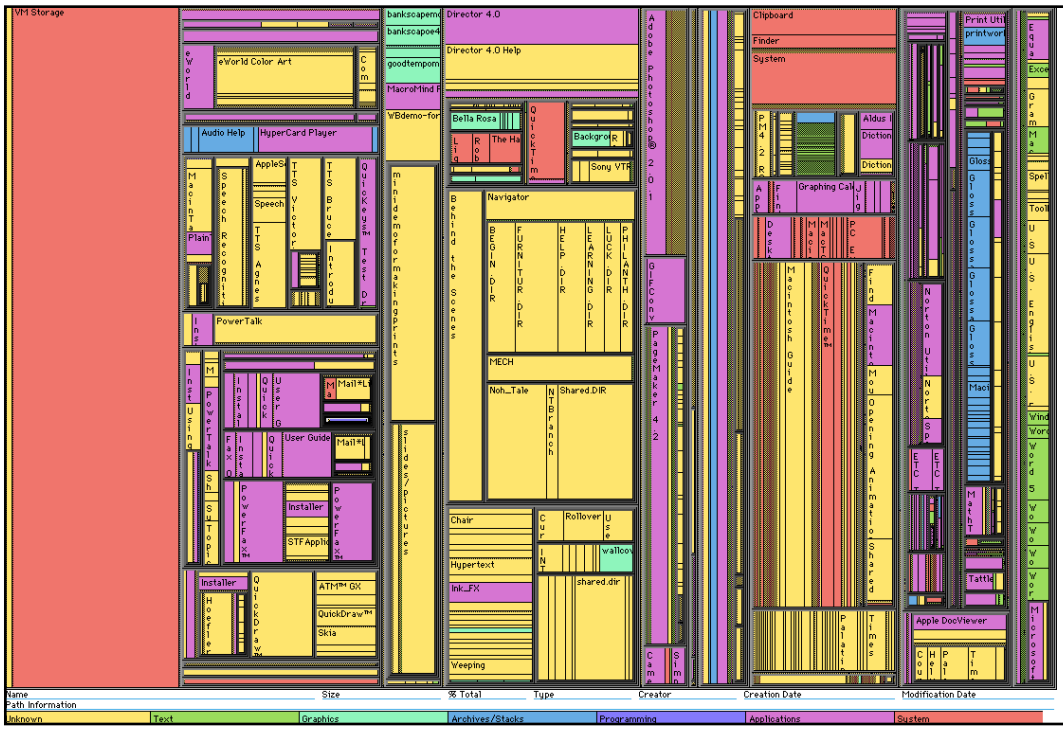
69

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]



71



Slice & Dice **layout:** Alternate horizontal / vertical partitions

72

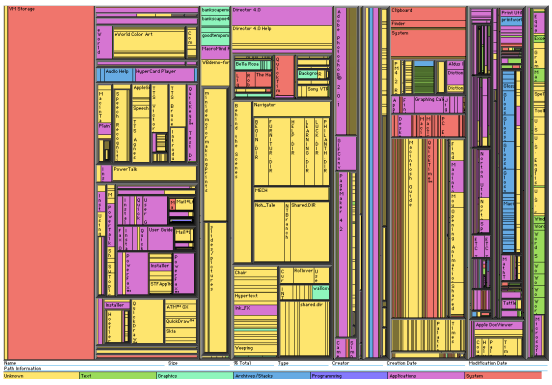
***Squarified* layout:** Try to produce square (1:1) aspect ratios

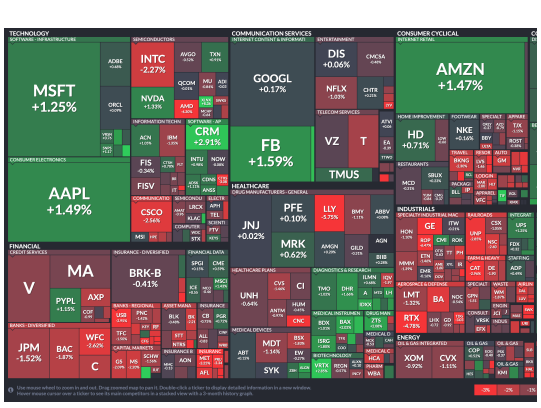
73

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





74

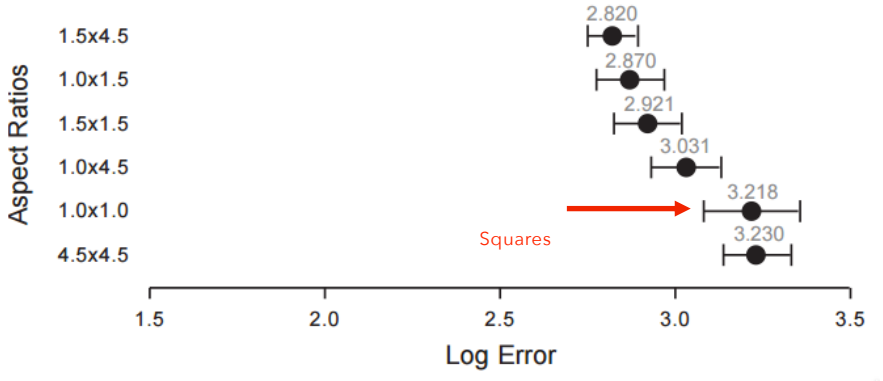
WHY SQUARES?

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?

75

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?

76

WHY SQUARES?

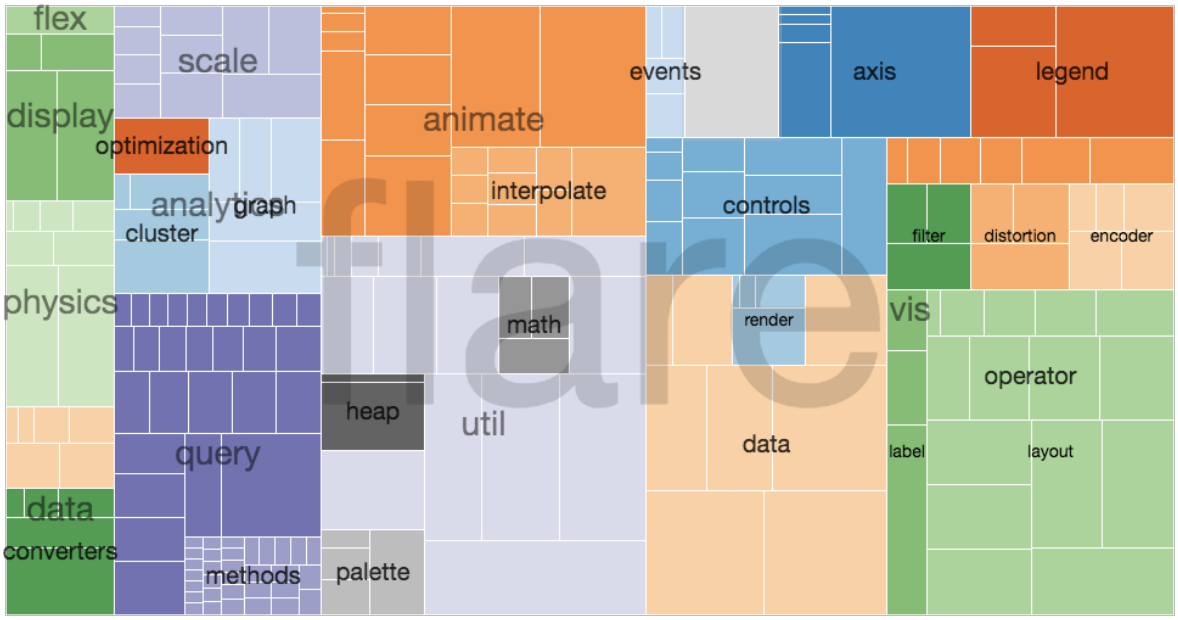
Posited Benefits of 1:1 Aspect Ratios

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, are more inaccurate.
   Balanced ratios better? Target golden ratio?

77

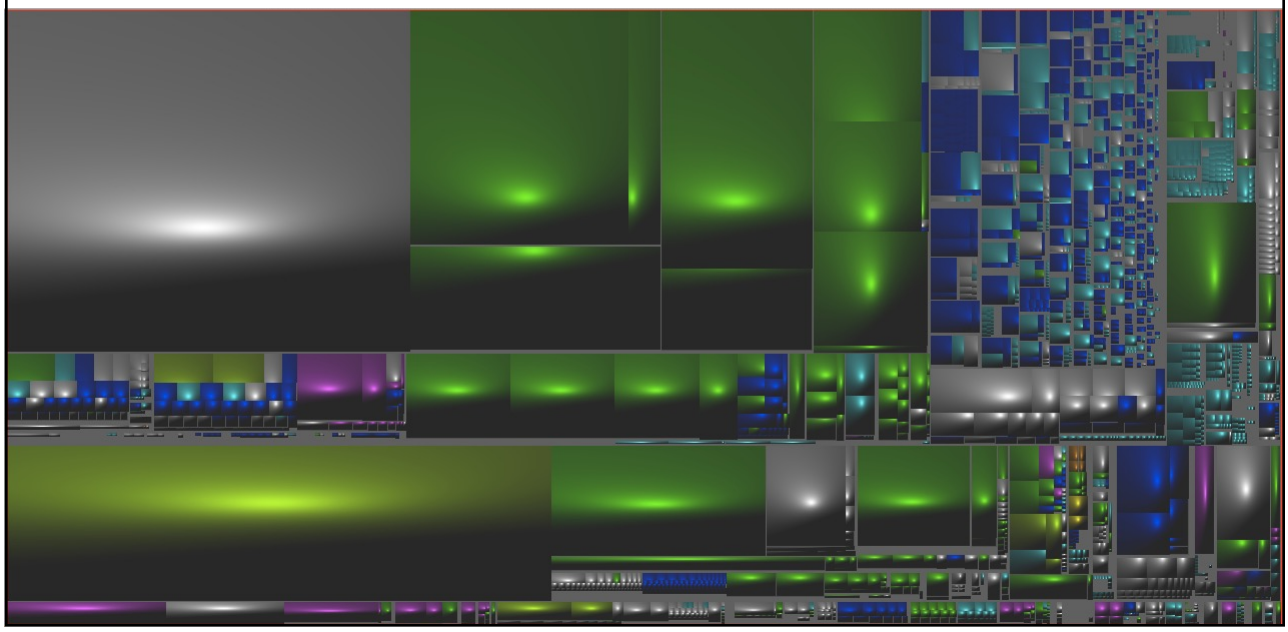
INTERACTIVE EXAMPLE

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



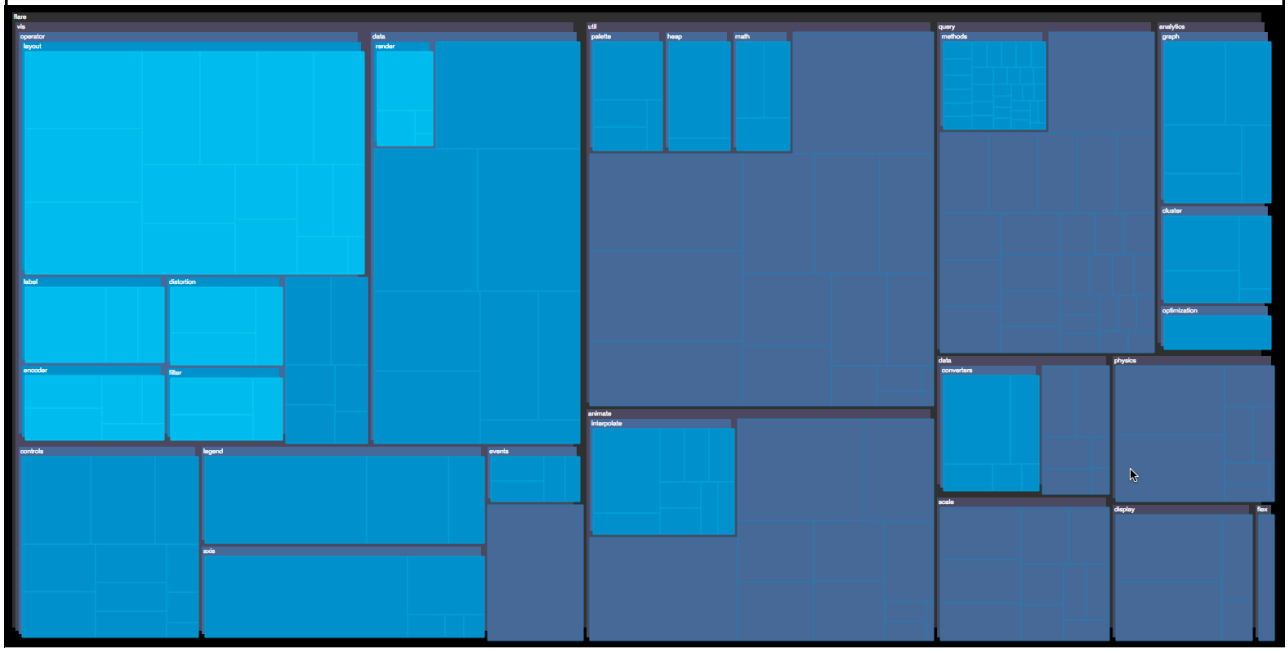
78

CUSHION TREEMAPS [van Wijk 1999]



81

CASCADED TREEMAPS [Lü 2008]

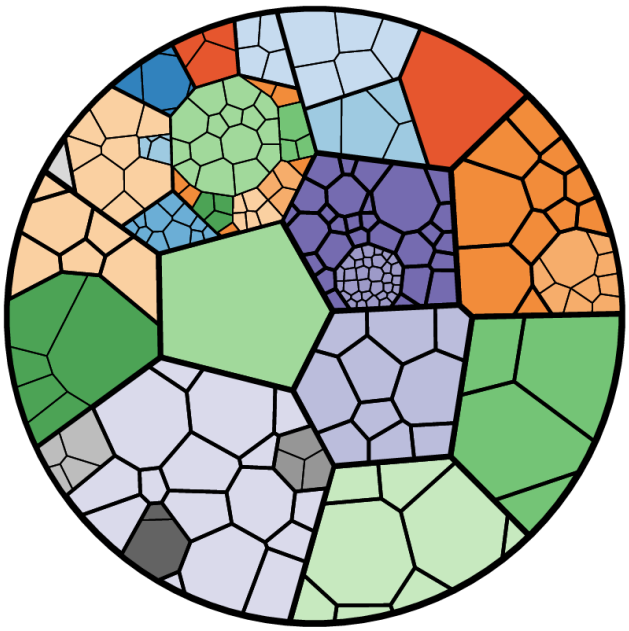


82

VORONOI TREEMAPS [Balzer 2005]

Treemaps with arbitrary polygonal shapes and boundary

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

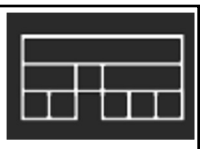


83

LAYERING

85

LAYERED DIAGRAMS



Signify tree structure using

- Layering
- Adjacency
- Alignment

Involves recursive sub-division of space

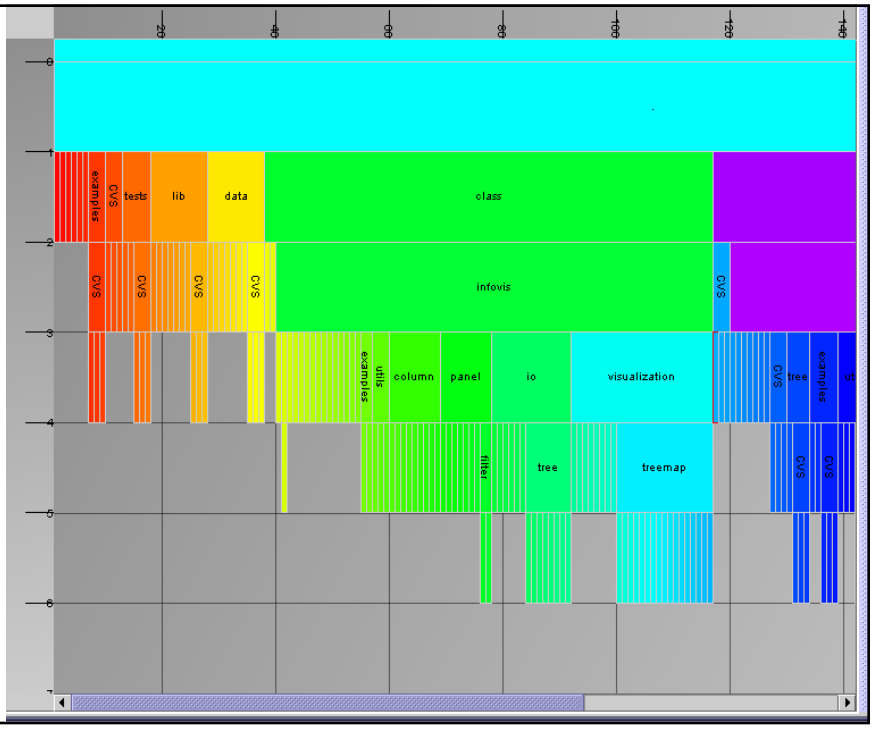
Leaf nodes may be sized by value, parent size visualizes sum of descendant leaf values

86

ICICLE TREES

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

Child levels are layered, constrained to parent's extent

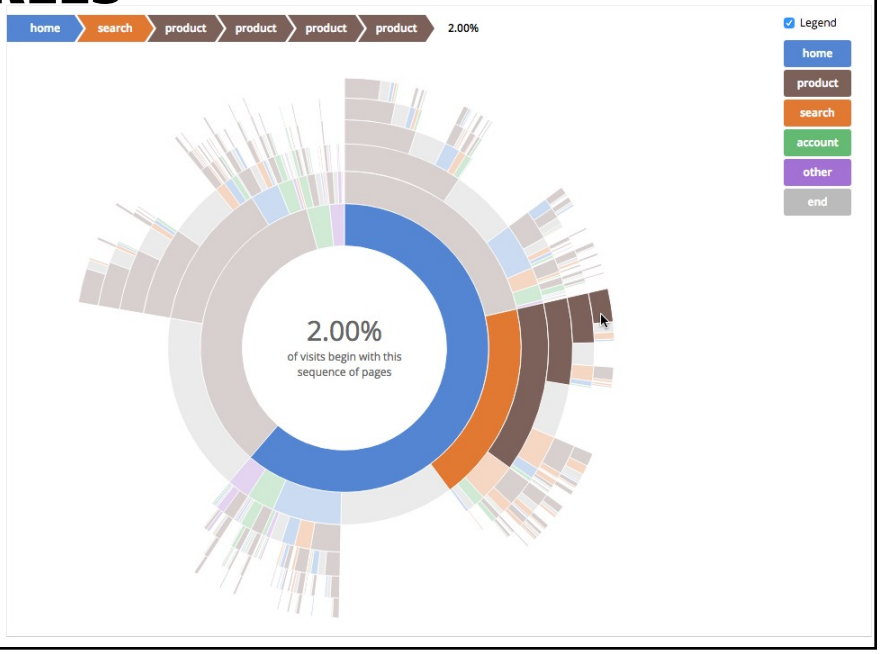


87

SUNBURST TREES

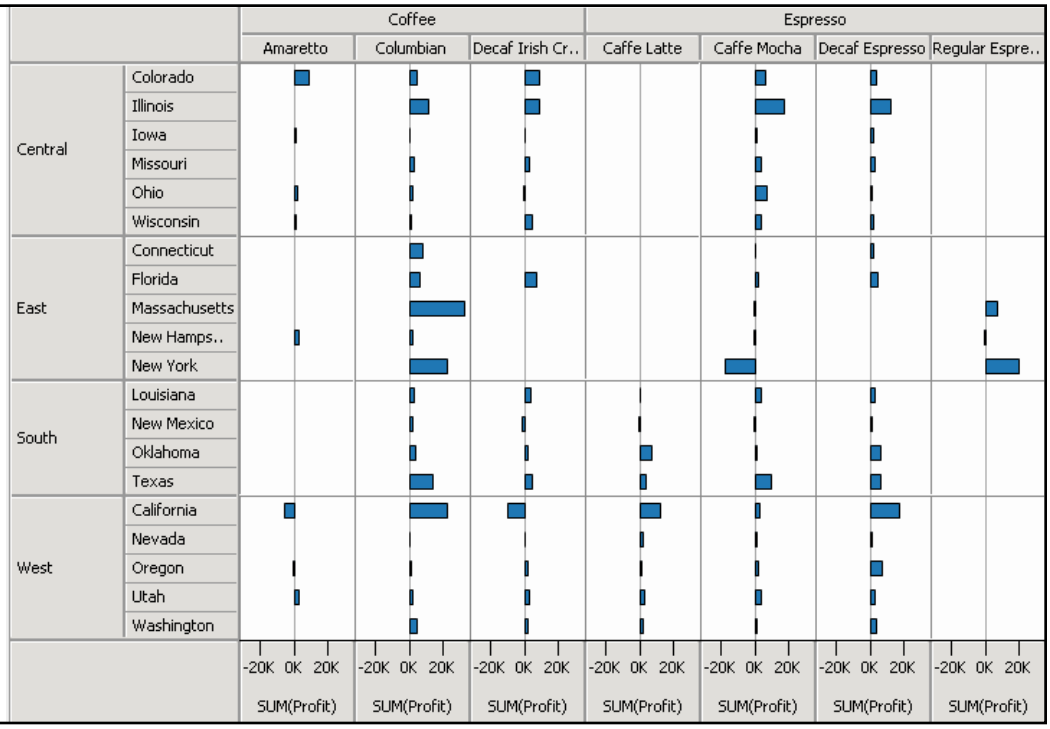
Parent nodes get a larger layer area, (e.g. angular extent)

Child levels are layered, constrained to parent's extent



88

ANOTHER USE OF LAYERED TREES



89

ANNOUNCEMENTS

91

FINAL PROJECT

Design Reviews Dec 1 and Dec 3

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: Today!
Design Review and Feedback: 10th week of quarter, 12/1 and 12/3
Final code and video: Sun 12/7 8pm

Grading

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

92

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

93

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

94

NODE-LINK GRAPH LAYOUT

95

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

96

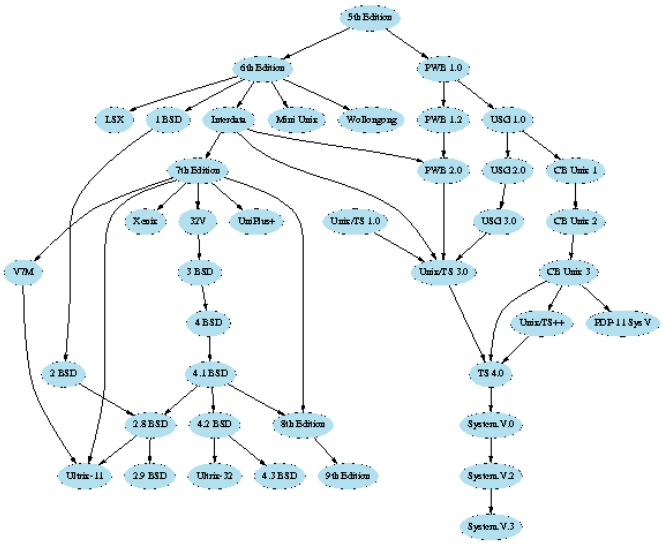
SUGIYAMA-STYLE LAYOUT

97

SUGIYAMA-STYLE GRAPH LAYOUT

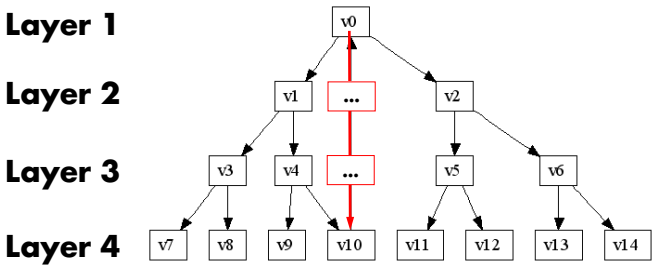
Evolution of the UNIX operating system

Hierarchical layering based on descent



98

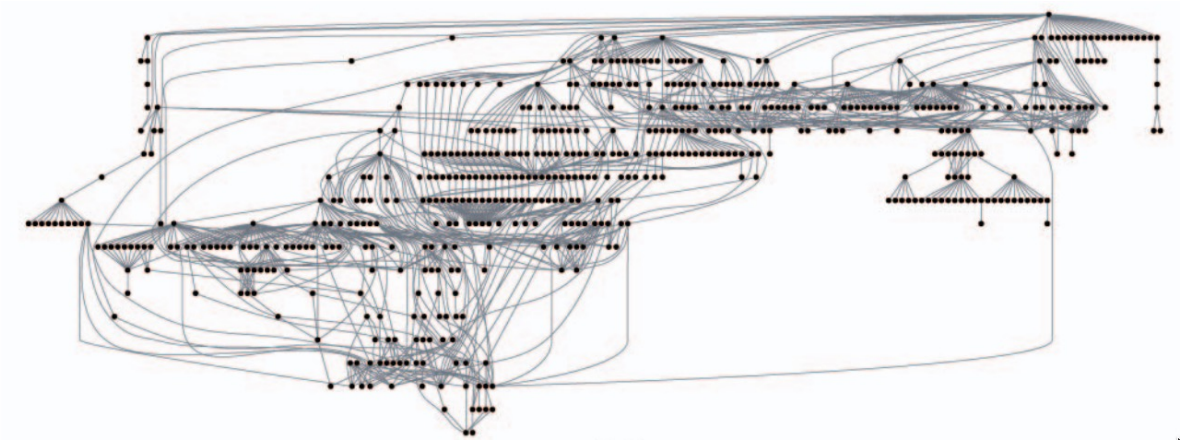
SUGIYAMA-STYLE GRAPH LAYOUT



Reverse some edges to remove cycles (if not already a DAG)
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

99

PRODUCES HIERARCHICAL LAYOUT

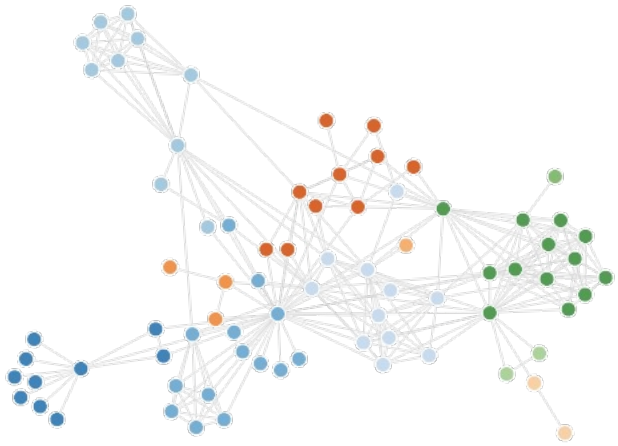


Sugiyama-style layout emphasizes hierarchy
  However, cycles in the graph may not be as apparent, and hierarchy may mislead
  Long edges can impede perception of proximity

100

FORCE-DIRECTED LAYOUT

101



Interactive Example: Configurable Force Layout

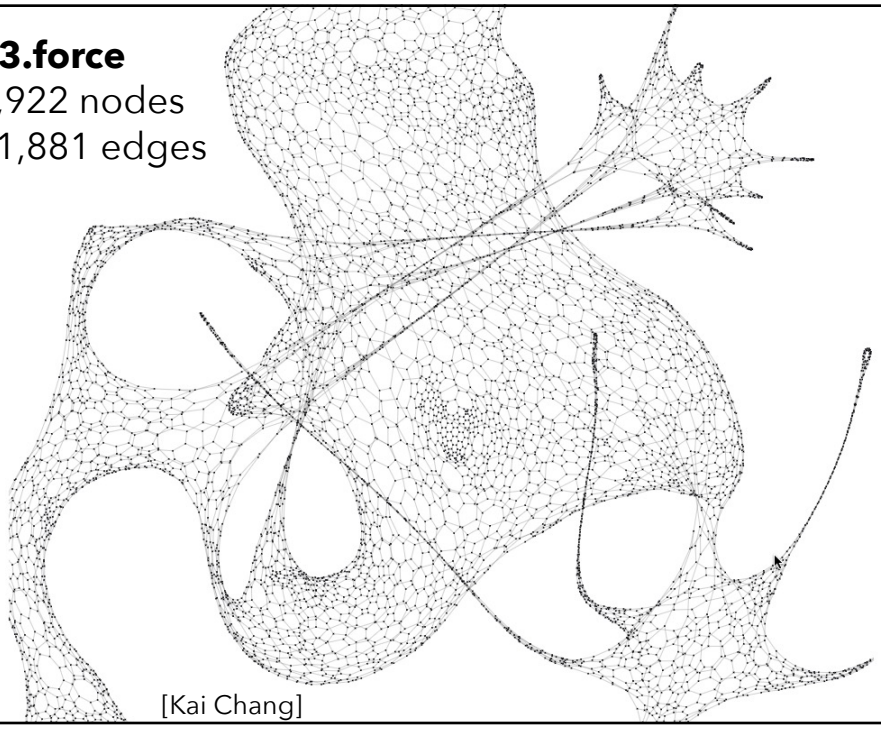
102

Use the Force!

http://mbostock.github.io/d3/talk/20110921/

104

d3.force
7,922 nodes
11,881 edges



[Kai Chang]

105

LAYOUT BY PHYSICS SIMULATION

Nodes = charged particles — $F = q_i * q_j / d_{ij}^2$
with air resistance — $F = -b * v_i$
Edges = springs — $F = k * (L - d_{ij})$

At each timestep, calculate forces acting on nodes.

Integrate for updated velocities and positions.

D3's force layout uses **velocity Verlet** integration

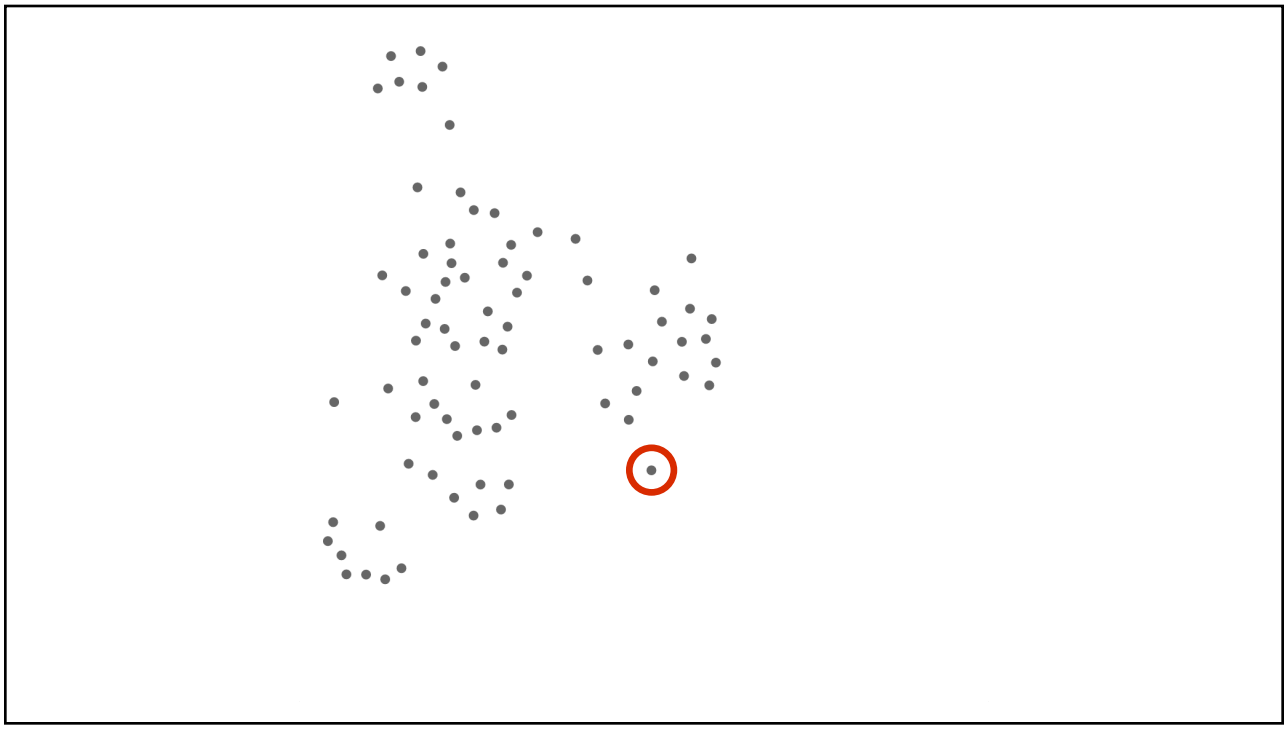
Assume uniform mass *m* and timestep *Δt*:

$F = ma \rightarrow F = a \rightarrow F = \Delta v / \Delta t \rightarrow F = \Delta v$

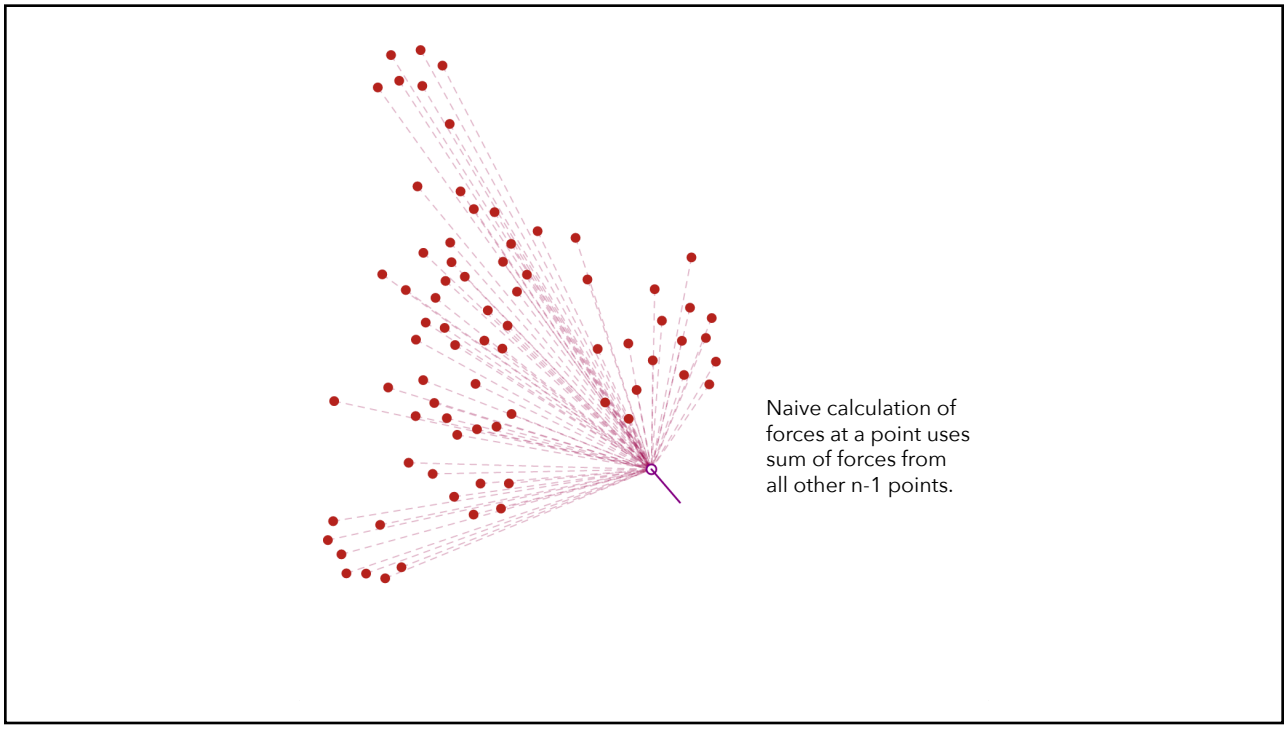
Forces simplify to velocity offsets!

106

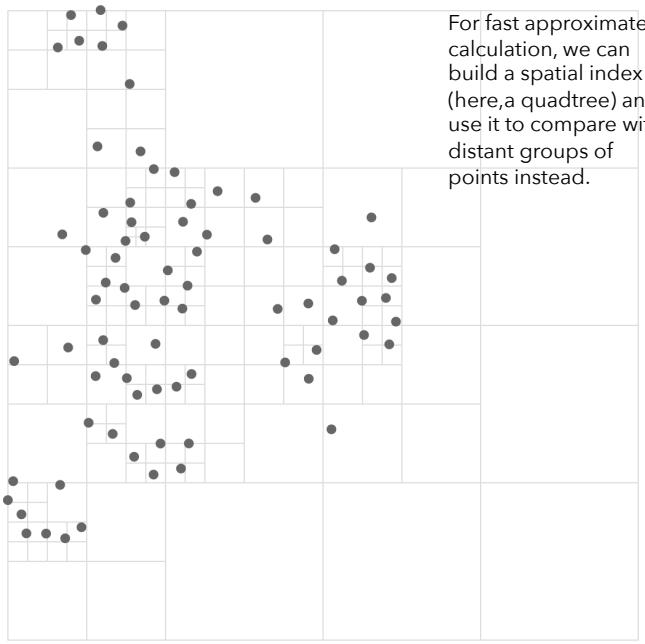
107



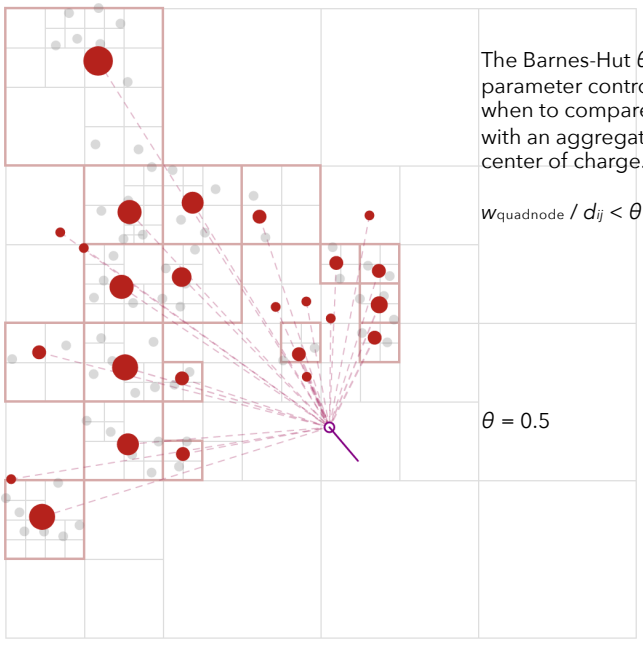
108


Naive calculation of
forces at a point uses
sum of forces from
all other n-1 points.

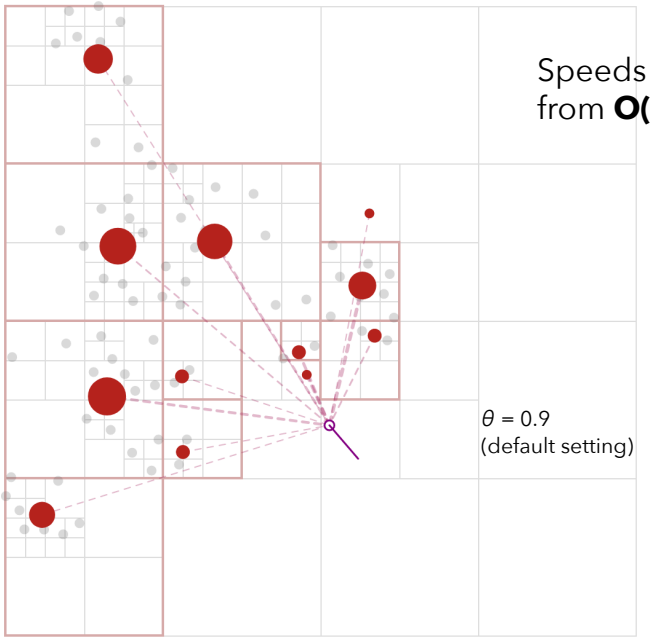
109


For fast approximate calculation, we can build a spatial index (here,a quadtree) and use it to compare with distant groups of points instead.

110


The Barnes-Hut θ parameter controls when to compare with an aggregate center of charge.
$w_{quadnode} / d_{ij} < \theta$?
θ = 0.5

111



112

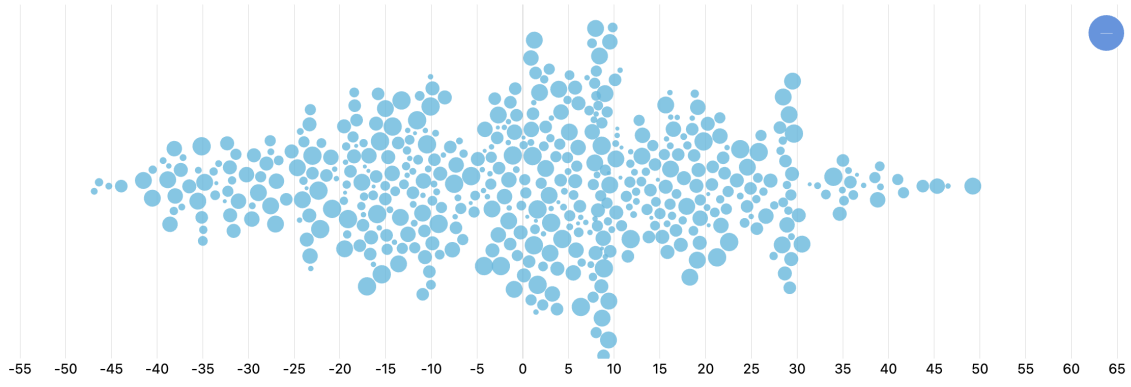
CUSTOMIZED FORCE LAYOUTS

https://www.amcharts.com/demos/beeswarm/

Different forces can be composed to create variety of custom layouts

A **beeswarm plot** can be made by combining:

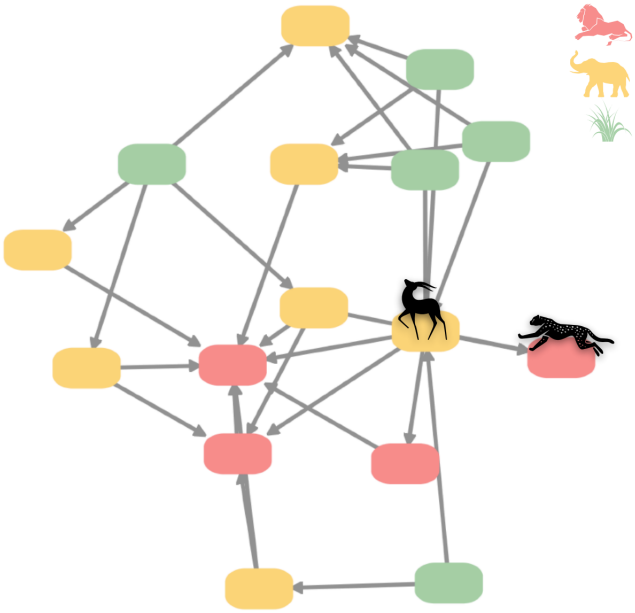
- Attractive **X** and **Y** forces to draw nodes of a certain category to a desired point
- **Collide** force to detect collision & remove overlap



116

ATTRIBUTE-DRIVEN LAYOUT

117



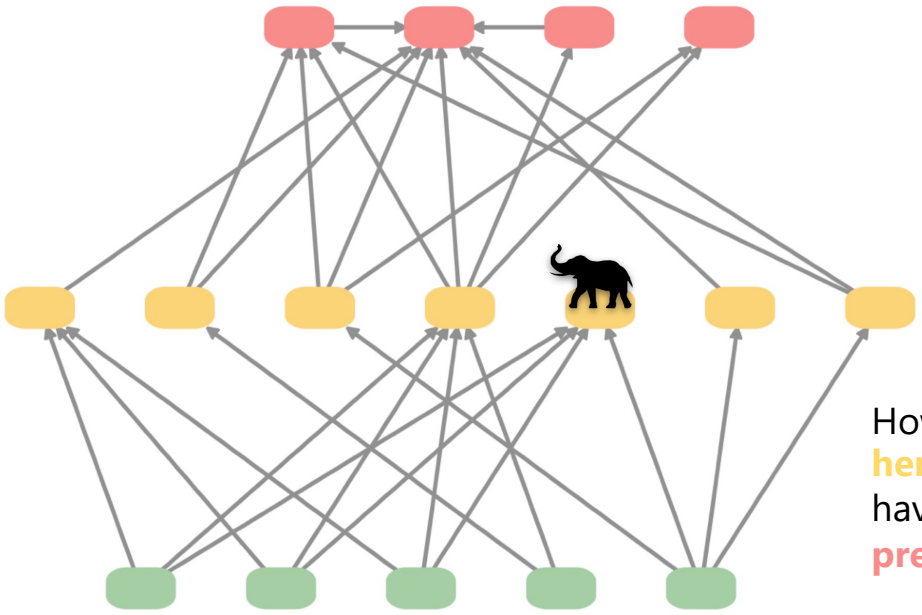
Carnivores (meat eaters)

Herbivores (plant eaters)

Plants

How many **herbivores** have no **predators**?

118



How many **herbivores** have no **predators**?

119

ATTRIBUTE-DRIVEN LAYOUT

Large node-link diagrams **get messy**!

Can we exploit additional structure?

Idea: Use **data fields/attributes** associated with nodes or edges to perform layout (e.g., scatter plot based on node values)

Attributes may also be statistical properties of the graph

Can apply dynamic queries & brushing on attributes/fields to explore…

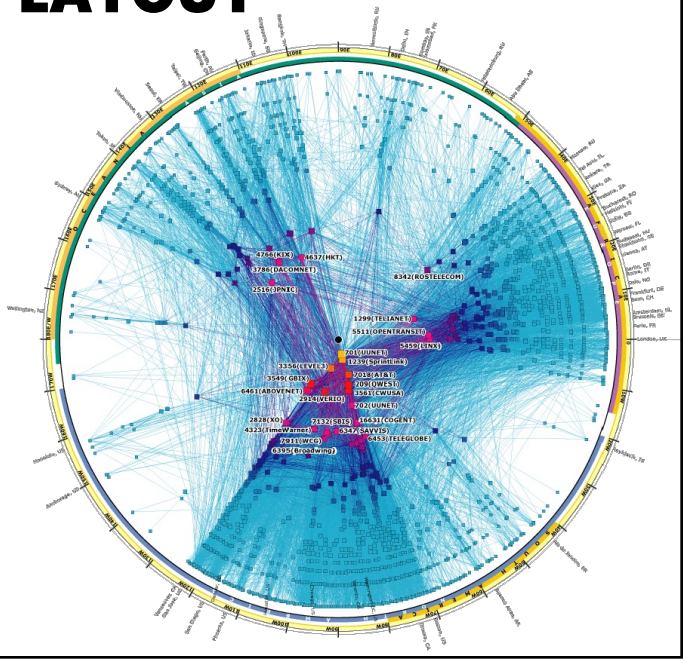
120

ATTRIBUTE-DRIVEN LAYOUT

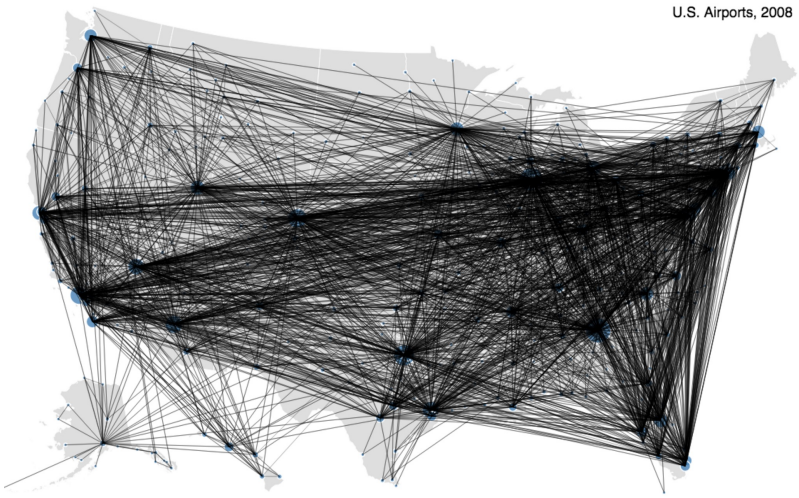
The "Skitter" Layout
Internet Connectivity
Radial Scatterplot

Angle = Longitude
Geography

Radius = Degree
# of connections
(a statistic of the nodes)

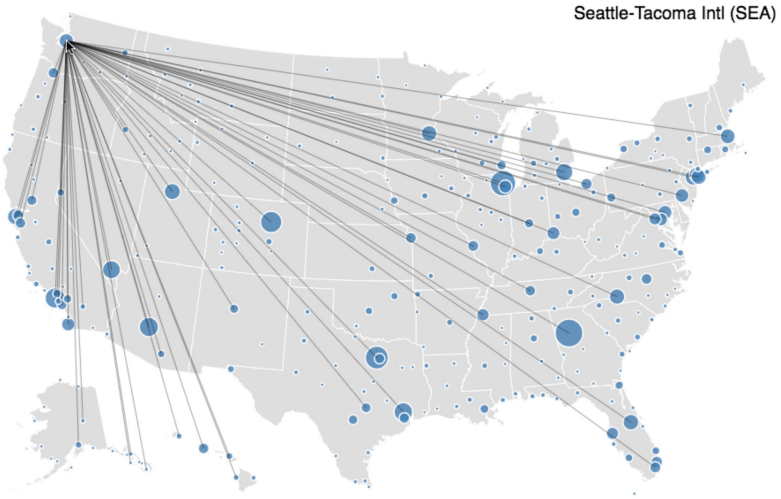


121



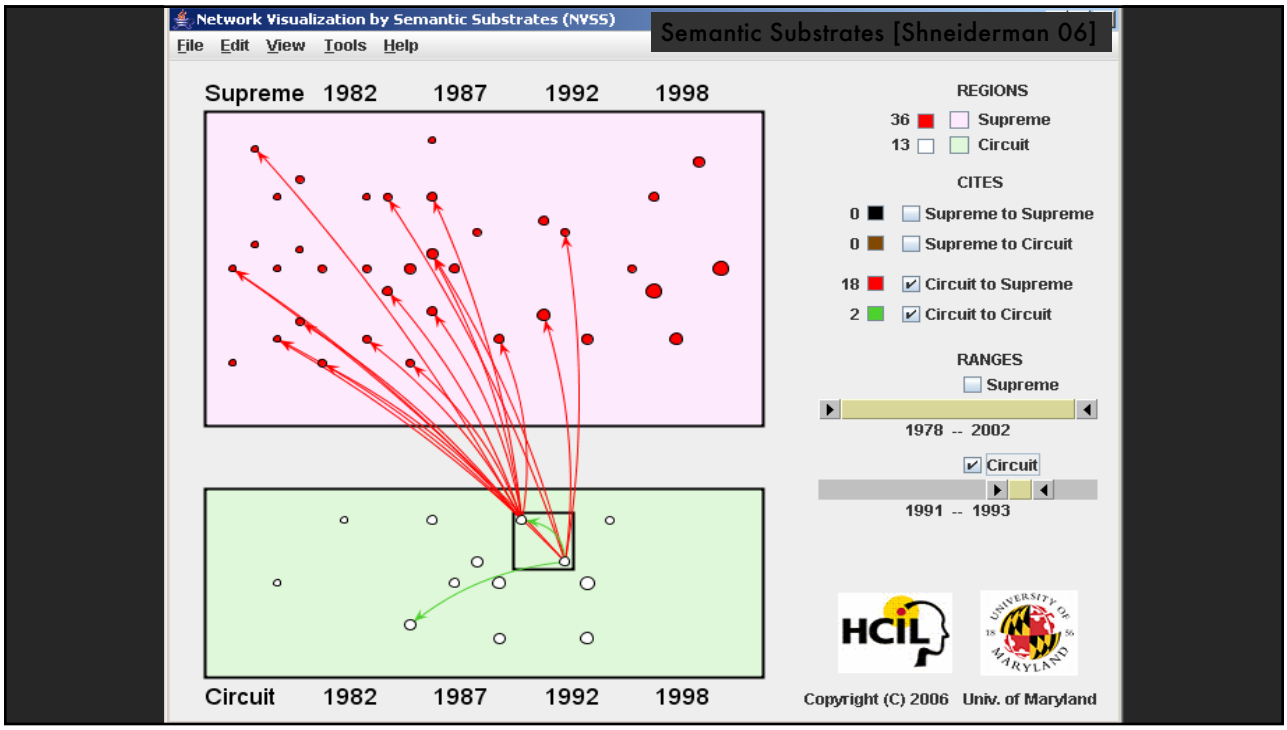
Drawing all edges is not particularly useful here…

122



Node layout determined by geographic location.
Adjacent edges shown on node selection.

123



124

CONSTRAINT-BASED LAYOUT

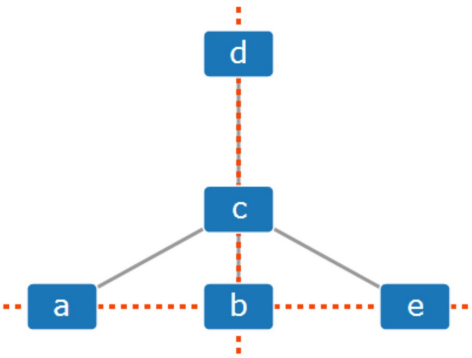
131

CONSTRAINT-BASED LAYOUT

Treat layout as an optimization problem

Define layout using an *energy model* along with *constraint equations* the layout should obey

Use optimization algorithms to solve:



Position Constraints

a must be to the ***left*** of b

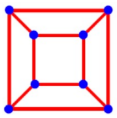
d, c, and b must have the same ***x position***

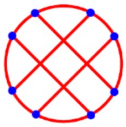
a, b, and e must have the same ***y position***

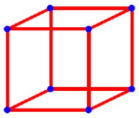
132

OPTIMIZING AESTHETICS

Minimize edge crossings
Minimize area
Minimize line bends
Minimize line slopes
Maximize smallest angle between edges
Maximize symmetry

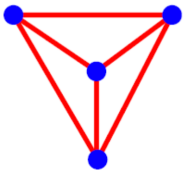




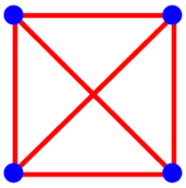


but, can't do it all

Optimizing these criteria is often NP-Hard, and requires approximations



min # crossings

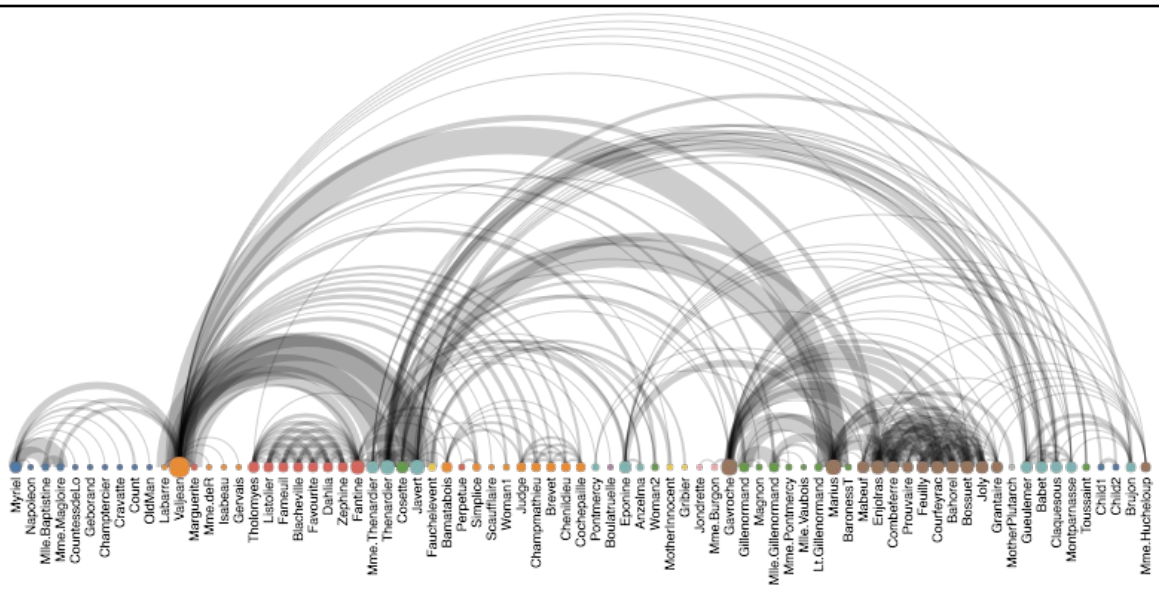


max symmetries

133

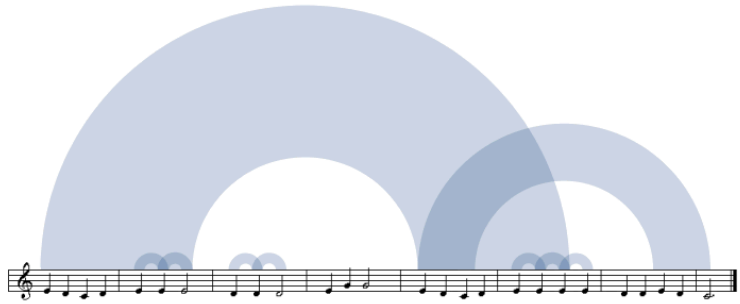
ARC DIAGRAMS

135



Linear node layout, circular arcs show connections.
Layout quality sensitive to node ordering!

136



For example, the picture above was built from the first line of a very simple piece: *Mary Had a Little Lamb*. Each arch connects two identical passages. To clarify the connection between the visualization and the song, in this diagram the score is displayed beneath the arches.

This diagram visualizes the refrain from the folk song *Clementine*. As you would expect, the refrain consists of multiple repetitions of the same passage--and that is exactly what the diagram shows. The score isn't shown in this diagram since the notes would be too small to read.

The Shape of Song
[Wattenberg 2001]

137

NODE-LINK GRAPH VISUALIZATION

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

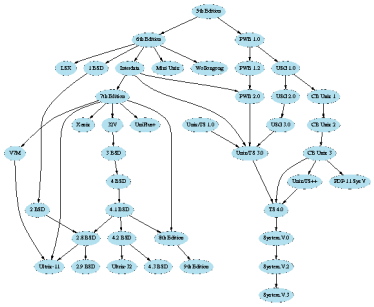
138

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges



Force-Directed Layout - physical simulation

Attribute-Driven Layout - arranged by value

Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

139

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

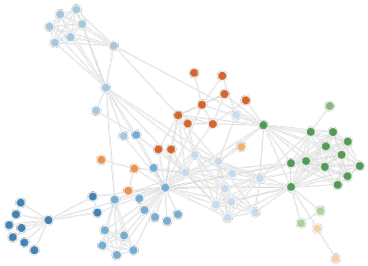
Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks



Attribute-Driven Layout - arranged by value

Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

140

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

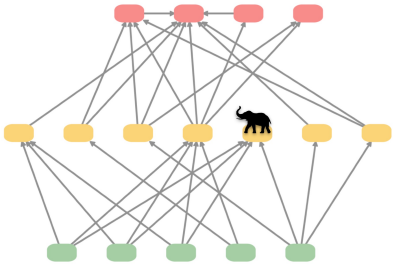
Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks



Attribute-Driven Layout - arranged by value

Good: Enables attribute-based analysis tasks

Bad: Difficult to design layouts appropriate to revealing attributes and network structure

Constraint-Based Layout – optimization

Arc Diagrams - aligned layout

141

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value

Good: Enables attribute-based analysis tasks

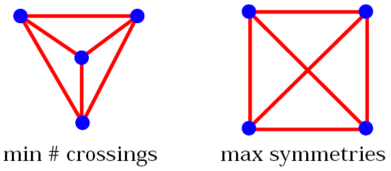
Bad: Difficult to design layouts appropriate to revealing attributes and network structure

Constraint-Based Layout – optimization

Good: Graph layout based on structural/aesthetic properties

Bad: Difficult to select appropriate constraints

Arc Diagrams - aligned layout



142

NODE-LINK GRAPH VISUALIZATION

Sugiyama-Style Layout - arranged by depth

Good: Structure-based analysis of hierarchical relationships

Bad: Browsing and path following due to long edges

Force-Directed Layout - physical simulation

Good: Structure-based analysis of closely related elements

Bad: Browsing and summarization of dense networks

Attribute-Driven Layout - arranged by value

Good: Enables attribute-based analysis tasks

Bad: Difficult to design layouts appropriate to revealing attributes and network structure

Constraint-Based Layout – optimization

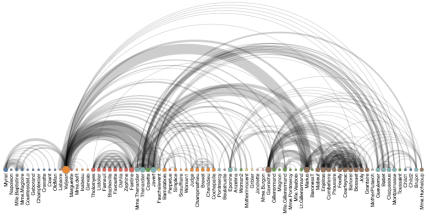
Good: Graph layout based on structural/aesthetic properties

Bad: Difficult to select appropriate constraints

Arc Diagrams - aligned layout

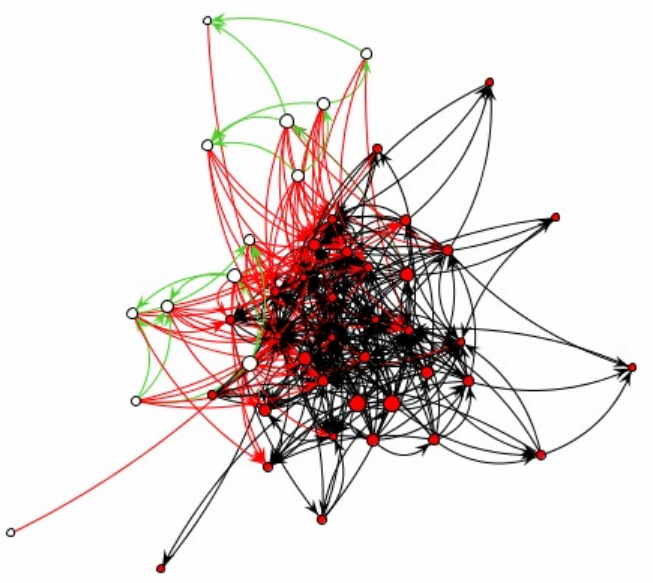
Good: Summarization and comparison of overall structure

Bad: Order matters for node layout; Structure-based and path following



143

LIMITATIONS OF NODE-LINK LAYOUTS



Edge crossings and occlusions!
Poor scalability…

144