Lecture 13: Graphs/Trees
Information Visualization
CPSC 533C, Fall 2009
Tamara Munzner
UBC Computer Science
Mon, 31 October 2011

Readings Covered
Online Dynamic Graph Drawing. Yair Frishman and Ayelit Tal. Proc EuroVis 2007, p 75-82.

Further Readings

Trees: Basic Node-Link Drawings
- task/data abstraction
- understanding detailed topological structure of tree
- visual encoding: layered node-link view
- vertical position: distance from root node in hops
- horizontal position: (as much symmetry as possible)

Trees: Radial Node-Link Drawings
- algorithm level:
  - Wetherell and Shannon 1978. Tidy Drawings of Trees
  - Reingold and Tidford 1981. Tidier Drawing of Trees
  - Buchheim et al 2002. Improving Walker’s Algorithm to Run in Linear Time

Trees: Treemaps
- data abstraction: tree nodes have attributes
- task abstraction: emphasize node attrs, not topological structure
- visual encoding: use containment not connection

Cushion Treemaps
- visual encoding: also show nesting/topo structure more clearly with shading cues
- interaction: scale parameter controls global vs. local

Scaling Up Treemaps: MillionVis
- visual encoding: treemaps, scatterplots
- darkness shows nesting level
- algorithm: many GPU tricks for speed
- dynamic queries through Z buffering

Graphs: Hierarchical Layout
- visual encoding
  - vertical position: distance from root
  - does not mean using containment
  - algorithms
    - Gansner et al 1993, A Technique For Drawing Directed Graphs
    - Eiglsperger et al 2005, An efficient implementation of Sugiyama’s algorithm for layered graph drawing

Graphs: Circular Layout
- visual encoding
  - nodes on circle
  - edge crossings minimized
  - algorithms
    - Spi and Tollis 1999, A Framework for Circular Drawings of Networks

Graphs: Force-Directed Placement
- visual encoding
  - non-deterministic placement
  - algorithm
    - spring forces pull together edges, repulsive forces pull apart nodes
    - optimization framework easy to extend, but tends to be brittle

Hermann Survey
- true survey, won’t try to summarize here!
- nice abstraction work by authors themselves
- derived data: skeletonization via Strahler numbers
- visual encoding techniques:
  - ghosting = layering
  - hiding = elision
  - grouping = aggregation
  - maintain neighbor order to stabilize (note prefuse bug!)
Online Dynamic Graph Drawing

- data abstraction: streaming data not static file
- task abstraction: dynamic stability (tradeoff)
- minimize visual changes
- stay true to current dataset structure

On-line Dynamic GD: Algorithm

- static graph layout algo unstable
- small changes in input can have large changes in output
- randomness, no constraints on maintaining geometric proximity
- dynamic online algorithm
- first step: initialize, layout
- later steps: merge, pin, layout, animate
- acceleration: partition before GPU force-directed layout

Online Dynamic GD: Validation

- algorithm level
- complexity analysis
- benchmarks: running time for CPU and GPU versions
- visual encoding level
- qualitative discussion of result images/videos
- quantitative metrics:
  - pairwise avg node displacement for stability
  - potential energy for quality
  - compare static, full dynamic, dynamic without pinning

Critique

- strengths
  - strong algorithmic contribution
  - previous work not scalable
  - very good validation, matches technique contribution
  - best paper award, EuroVis 2007
- weaknesses
  - using mesh datasets to test graph drawing claims
  - different topological characteristics than typical infovis case
  - does not scale past 2-3 levels of depth
  - no constraints on maintaining geometric proximity

Small World: Nested Quotient Graphs

- pro: very evocative of structure
- con: does not scale past 2-3 levels of depth

Small World: Coarsening

- remove low-strength edges
- maximal disconnected subgraphs
- quotient graph: subgraph = higher-level node

Small World: Nested Quotient Graphs

- visual encoding: containment: subgraph laid out within metanode

Critique

- strengths
  - strong algorithmic contribution
  - previous work not scalable
  - very good validation, matches technique contribution
  - best paper award, EuroVis 2007
- weaknesses
  - using mesh datasets to test graph drawing claims
  - different topological characteristics than typical infovis case
  - does not scale past 2-3 levels of depth
  - no constraints on maintaining geometric proximity

Topological Fisheye Views

- data abstraction: create cluster hierarchy on top of original graph (coarsening)

Topo Fisheye: Coarsening Strategy

- must preserve graph-theoretic properties
- topological distance (hops away), cycles
- cannot just use geometric proximity alone
- cannot just contract nodes/edges
- exploit geometric information with proximity graph

Topo Fisheye: Coarsening Requirements

- uniform cluster/metanode size
- match coarse and fine layout geometries
- scalable
Topo Fisheye: Hybrid Graph
- find active nodes


Topo Fisheye: Distort For Uniform Density
- visual encoding
  - geometric distortion for uniform density
  - color coded by depth in hierarchy to illustrate algorithm


Critique
- strengths
  - topologically sophisticated, not just geometric distortion
  - rigorous approach
- weaknesses (shared by many approaches)
  - what if mental model does not match coarsening strategy?
  - again, meshes for evaluating infovis claims

Presentation Topics
- see course page for your day/topic
- seed papers coming soon for Wed Nov 9 folks

PivotGraph
- task abstraction: show relationship between node attributes and connections in multiattribute graph
- data abstraction: rollup and selection transformations


PivotGraph
- visual encoding: line (1D) or grid (2D), area proportional to attribute
- grid nodes based on attribute count, not original graph node count!
- scalability through abstraction, not layout algorithms


PivotGraph
- visual encoding: line for 1D rollup, or grid for 2D case


PivotGraph
- interaction: changing rollup/selection choices, animated transitions between states


PivotGraph
- in general, more compact than matrix view


Fig 14, Fig 15, Fig 6, Fig 4, Fig 2,3, Fig 7,8.