Interactive, Tree-based Graph Visualization

Andy Pavlo
Rochester Institute of Technology
Department of Computer Science
Outline

- Introduction / Graph visualization
- Graph drawing
- Graph animation
- Evaluation
Our Contributions

- Graph drawing algorithm
- Graph animation algorithm
- Experimental analysis
Graph Visualization

- “Pictures” of graphs
- Helps to easily detect patterns
- Many applications

From Yee et al. [27]
From Munzner [18]
From GraphViz [14]
Graph Drawings

• What makes a drawing readable/useful?
• Prior research on drawing characteristics:
  – Usability studies [21,17,25]
  – Experiments using eye-tracking systems [12]
• **Minimizing edge crossings greatest impact**
• Not all graphs can be drawn without edge crossings.
Our Research

• How to address the problems of edge crossings when visualizing graphs?
• Trees as an inspiration:
  – Ubiquitous in nature.
  – Always planar

From Thierry Lombry / NASA
Context-free Radial Layout Graph Drawing Algorithm

- Spanning-tree-based drawings
- Each vertex is positioned relative to its parent
- Preserves connectivity between root and children
- Subtrees are drawn on a series of overlapping circles
Problems

- Single spanning tree view is a biased view
- Tree layouts are dependent on the root selected
- Removing edges removes information
- We need an interactive approach!
Interactive Graph Visualization

- View the graph through a sequential number of spanning-tree-based drawings
- Reintroduce information over multiple drawings
- Can return to the full graph drawing
Related Work

• *Graph Drawing* – Battista et al. 1999 [2]
• *Algorithms for Drawings Graphs* – Battista et al. 1994 [1]
• *Graph Visualization and Navigation in Information Visualization* – Melancon et al. 2000 [10]

• Incremental Exploration:
  – North 1996 [20]

• Focus+Context:
  – Sarkar et al. 1992 [22]
  – Schaffer et al. 1996 [23]
Related Work

• Graph animation:
  – Friedrich et al. 2002 [8]
  – Friedrich et al. 2002 [9]
  – Diehl et al. 2002 [3]
  – Nesbitt et al. 2002 [19]

• Radial layout animation:
  – Yee et al. 2001 [27]
Related Work: Radial Layout Drawings

• Methods for drawing rooted trees:
  - **Yee et al. 2001** [27]
  - Melancon et al. 1998 [16]
  - Wills 1999 [26]
  - Teoh et al. 2002 [24]
Related Work: Hyperbolic Layout Drawings

- Space-filling layout on the hyperbolic plane
- Projection into Euclidean space/plane
- Distant elements are displayed with less space

Examples:
- Lamping et al. 1995 [15]
- Munzner 2000 [18]
Drawing Algorithm (1)
Drawing Algorithm (1)

- Root is placed at the center
- Containment circle centered at root
- Root's children are positioned evenly on this circle
Drawing Algorithm (2)

- Root's children are allocated equal-sized circles
Drawing Algorithm (3)

- Calculate the area for the next subtree level
Drawing Algorithm (4)

- Process continues down each subtree
Drawing Algorithm (5)

- Final layout
- Runs in $O(n)$ time
Demo

- Randomly generated graph:
  - Erdos-Renyi model [7]
  - 30 vertices
  - 10% probability edge connectivity
- Initial force-directed graph drawing
- Spanning-tree-based drawings
Animation Algorithm

- Smoothly transition between graph drawings
- Vertices move on radial paths
- Help users relate one drawing to the next
Animation Goals

- Minimize the number of edge crossings
  - Sibling vertices
  - Parent-child with parent-grandparent
Animation Algorithm (1)

- Calculate polar coordinates to new parent in old drawing
- Calculate polar coordinates to parent after the animation in the new drawing
- Calculate deltas for each child
Animation Algorithm (2)

- The root moves in a straight-line path to the center
- The root's children move from an interpolation of their polar coordinates
- Children's coordinate system centered at parent
Animation Algorithm (3)

- Vertices move relative to their parent
- Minimal rotational movement
- No crossings between siblings
Demo

- Randomly generated graph:
  - Erdos-Renyi model [7]
  - 30 vertices
  - 10% probability of edge connectivity
- Initial force-directed graph drawing
- Spanning-tree-based drawings
- Animated transitions
- Fading of graph elements [8]
- Slow-in, slow-out timing [27]
Experimental Analysis

- Evaluate our visualization scheme
- Compare against Gnutellavision from Yee et al. [27]
Measurements

• Edge Crossings
  – Transient crossings
  – Final layout crossings
• Sibling Edge Lengths
Methodology

- Graph order - 30 to 100 vertices (inclusive)
- 10 trials per graph size
- Average measurements for all trials per graph size
- Randomly generated graphs
- Randomly selected root vertices
Experiment 1: Isomorphic Tree Transitions

- Measure transitions between drawings of two spanning trees $T_1$ rooted at $r_1$ and $T_2$ rooted at $r_2$, where $r_1 \neq r_2$ and $E(T_1) = E(T_2)$.

- Count the crossings between all edges
Experiment 1: Isomorphic Tree Transitions

- Our algorithms produce zero crossings.
- Yee et. al's algorithms:
  - Vertices move on the shortest radial path
  - Preserve the edge direction of the new root with its parent from previous drawing

![Graph showing Total Edge Crossings against Numbers of Vertices in Graph]
Experiment 2: Spanning-Tree-to-Spanning-Tree Transitions

- Measure transitions between drawings of two spanning trees $T_1$ rooted at $r_1$ and $T_2$ rooted at $r_2$, where $r_1 \neq r_2$ and $E(T_1) \neq E(T_2)$.

- Count the crossings between all edges
Experiment 2: Spanning-Tree-to-Spanning-Tree Transitions

- Results
Experiment 3: Full-Graph-to-Spanning-Tree Transitions

- Measure transitions between a force-directed layout drawing of the full graph to a spanning tree drawing rooted at randomly selected vertex.
- Count crossings between all edges
- Scenario is specific to our visualization system
Experiment 3: Full-Graph-to-Spanning-Tree Transitions

- Results

![Graphs showing Total Edge Crossings and Final Layout Edge Crossings for different numbers of vertices.](image)
Experiment 4: Spanning Tree Sibling Edge Lengths

- Measure the edge lengths of sibling vertices in a spanning tree drawing.
- Static spanning tree drawings
- Mean length and mean standard deviation for each graph size
Experiment 4: Spanning Tree Sibling Edge Lengths

- Our algorithm:
  - Siblings are equidistant to their parent.
  - Edge length conveys depth to the root.
Experiment 4: Spanning Tree Sibling Edge Lengths

- Yee et. al's algorithm:
  - Vastly different sibling edge lengths
  - Difficult to perceive parent-child relationships
Experiment 4: Spanning Tree Sibling Edge Lengths

• Results

Mean Length of Sibling Edges with Standard Deviation
Discussion

- Our drawings make structural properties apparent
- Our transitions produce fewer edge crossings
- Users can make better judgments about graphs
Future Work

- Behavioral experiments
- Graph animation aesthetics
- Guarantee planarity
Conclusion

- Graph drawing algorithm
- Graph animation algorithm
- Experimental analysis
- References attached in handout