Topological Fisheye Views for Visualizing Large Graphs

Emden R. Gansner
Yehuda Koren
Stephen C. North

Presented By
Menion Croll
Problem
Related Work

• Pan and Zoom
• 2D Geometric Fisheye
• Multiscale Display
• 3D Hyperbolic Projection
• Clustered Graph Abstraction
2D Geometric Fisheye
Multiscale Graph Drawing

- Coarsen the graph
- Draw the coarse graph
- Perform local optimizations on small neighborhoods
3D Hyperbolic Projection
Clustered Abstraction

- Cluster the nodes
- Allow for expansion and contraction of the clusters
The Approach

• Goals
  – Efficient use of display area
  – Informative in context displays
  – Variable abstraction preserving structure
The Approach

• Input:
  – Some graph layout
  – In this case a force directed layout

• Combine static multiscale display with fisheye view
Sidebar: Force Directed Graphs

• A technique for determining the layout of a graph automatically
• Assign an attraction force to the edges and repulsive forces to the vertices
  – Make the edges springs
  – Make the vertices charged particles
• Run until equilibrium is achieved
How to Combine multiscale and fisheye

• Create a hierarchy of graphs
• Given a focal node, merge the various levels into a single hybrid graph
• Use coarser graphs the farther out from the focal node you are
Coarsening the Graph

• Requirements
  – Preserve topological properties
  – Create uniformly sized clusters
  – The layout of the coarse graph should preserve the geometry of the fine graph
  – Be efficient
Coarsening the Graph

- Select a maximal set of disjoint pairs
- Contract these pairs
Coarsening by Contraction is a 2 Step Process

- Construct a candidate set of node pairs
- Select a maximal subset of disjoint pairs that will actually be contracted
Finding Candidates for Contraction

- Use the edges of the graph
- Use the edges of the proximity graph
Sidebar: Proximity Graphs

• A proximity graph is a graph derived from the geometry of a set of points

• The paper considers proximity graphs from 2 sources
  – Delaunay triangulation
  – Relative neighborhood graph

\[ \|p_i - p_j\| \leq \max\{\|p_i - p_k\|, \|p_j - p_k\|\}. \]
Proximity Graphs

Delaunay triangulation

relative neighborhood graph
Proximity Graphs

• The edges of the proximity graphs ignore the graph theoretic structure
• Contracting these points may cause unwanted cycles
Choosing the Best Candidates

1. Geometric proximity: \[ \frac{1}{\|p_i - p_j\|} \]
2. Cluster size: \[ \frac{1}{\text{size}_i + \text{size}_j} \]
3. Normalized connection strength: \[ \frac{w(i, j)}{\sqrt{\text{size}_i \cdot \text{size}_j}} \], where \( w(i, j) = 0 \) if \( \langle i, j \rangle \notin E \).
4. Similarity of neighborhood: \[ \frac{|N_i^* \cap N_j^*|}{|N_i^* \cup N_j^*|} \]
5. Degree: \[ \frac{1}{\deg_i \cdot \deg_j} \]
Creating the Hybrid Graph

- The hierarchy of the coarse graphs is represented as a binary tree
  - Tree nodes correspond to coarse graph nodes
  - Each Leaf corresponds to a unique graph node
The Horizontal Slice

• Each Tree level corresponds to a coarse graph
• A more elaborate horizontal slice corresponds to a hybrid graph
Determining Active Nodes

• Create a wish list associating each node with a desired coarseness level
• Make decisions about the actual coarseness level for each node
• Propagate the decisions
Distorting the Layout
Distorting the Layout

- Work with Polar coordinates
- Density can be classified on the radial component
- Segments become stretched based on their density
The result
Evaluation

• Other than performance metrics, there is none
Questions

• What extensions can be made to further enhance this tool?
Questions

• Are there any situations where this tool would be ill suited for visualizing a graph?