TugGraph: Path-Preserving Hierarchies for Browsing Proximity and Paths in Graphs

Daniel Archambault$^1,2$, Tamara Munzner$^2$, and David Auber$^{1,3}$

$^1$GRAVITE, INRIA Bordeaux Sud-Ouest, France
$^2$University of British Columbia, Canada
$^3$CNRS UMR 5800 LaBRI, Université de Bordeaux I, France
Drawing Large Graphs

- Drawing algorithms have high asymptotic complexity
- Layouts suffer from visual clutter
Hierarchy of Coarse Graphs

- Usual approach: decompose the graph recursively
  - a subgraph is replaced by a single metanode at its parent
  - process is repeated on graph forming a cluster tree
- The structure is a multi-level hierarchy in previous work
Hierarchy Cut

- A **cut** defines which nodes are visible or hidden
  - nodes on and above the cut are visible in the graph view
- Parts of graph can be drawn on demand
- foundation for steerable exploration of a graph
  - drawing algorithm only applied to selected regions
Limitations of Steerable Graph Drawing

- **Steerable** multi-level graph drawing
  - drawing computed on demand
- Problem: opening large metanodes (+190,000 nodes)
  - even automatic coarsening not adequate
- Solution: tug out structure nearby a small, interesting subgraph
Video
Steerable Graph Drawing Systems

(a) DA-TU, 2000
(b) ASK-GraphView, 2006
(c) GrouseFlocks, 2008

- Explore hierarchy by drawing metanodes on demand
- Hierarchies created by subgraph selection in input graph
- Limitation:
  - Interaction does not take into account subgraph of interest
Path-Preserving Hierarchy

- Defined in GrouseFlocks work
- A path in the hierarchy means at least one path in the graph
- **Path-preserving hierarchies** respect this property
Path-Preserving Hierarchy

- Metaedge if and only if a pair of descendants connected
- Metanodes contain connected subgraphs
- If preserved, paths in cuts are also in underlying graph
Hierarchies that are not path-preserving can be misleading

(a) Invalid Hierarchy
(b) Hierarchy Cut

- Cycles can appear when there are not cycles present.
Algorithm Step Selection

- User selects a node
- All nodes in input graph are selected
Algorithm Step Select Adjacent

- Adjacent nodes to the selection are chosen
Algorithm Step Decomposition

- Hierarchy reformed according to selection set
- Nodes distance one away are selected in red
Principal Internet backbone routers
Successive tugs reveal structure around UBC network
200,000 nodes and 400,000 edges
Bacon numbers 1 and 2 act together
Trend not seen for Bacon number 3
about 39,000 nodes and 2,000,000 edges
Demo
Future Work

- Extend to weighted graphs and other notions of proximity
- Speed up technique to make tugs interactive
  - preserving hierarchy is costly
- Interactive techniques with large disconnected graphs
Conclusion

- Presented a technique to tug out elements near a subgraph
- Executed in a path preserving way
- Provides fluid interaction with very large graphs in seconds
Software and Acknowledgements

- TugGraph available as Tulip perspective
  - released in Tulip shortly
  - www.tulipsoftware.org

- Thank you UBC and LaBRI for comments on the work
- Partially funded by the INRIA GRAVITÉ project

- Questions?