Smoother Transitions Between Breadth-First-Spanning-Tree-Based Drawings

Christopher Homan[1], Andrew Pavlo[2], and Jonathan Schull[1]

[1] Rochester Institute of Technology
[2] University of Wisconsin-Madison

We demonstrate a collection of techniques that seek to make the transition between drawings based on two topologically distinct spanning trees of the same graph as clear as possible.

As Herman, Melançon, and Marshall note [HMM00], one way to draw a large graph is to extract a spanning tree from it, use a tree layout algorithm [CK95, Ead92, RT81, II90, TM02, GADM04, LY05] to draw the spanning tree, and then add back the graph edges not included in the spanning tree. The problem with this approach is that the drawings tend to favor the edges that are part of the spanning tree, even though they may be no more important in the underlying structure than non-spanning tree edges. One way of dealing with this problem is to facilitate exploration of multiple spanning trees.

Yee et al. [YFDH01] describe a system that produces layouts based on Eades' radial layout algorithm [Ead92] and lets users interactively select a new node as root. When this happens, the system first calculates a breadth-first spanning tree rooted at the selected node, and then smoothly transitions to a topologically distinct spanning tree. Although Yee et al.'s static layouts are free of edge crossings, transitions between trees can be hard to follow because there edge crossings do occur.

A number of tree-based graph visualizations, such as RINGS [TM02], RDT [JP98], and others [LRP95, Mun97, Wil99] also allow users to reconfigure views of a given tree, and some even allow users to change the root node. They do not, however, let the user select and smoothly transition to a different spanning tree built from a different collection of edges. To our knowledge only Yee et al.'s system [YFDH01] and one mentioned by Melançon and Herman [MH98] support smooth transitions between different spanning trees of the same graph.

We have built a system to use as a test bed for improving the sort of transitions between topologically distinct graphs that Yee et al. and Melançon and Herman use. Here we present some preliminary results. Like Yee et al. we only use breadth-first search trees. These often share common subtrees, especially when the roots of the trees are closely related. A major thrust of our research concerns layouts that make it easier for users to perceive the migration of these common subtrees as they disconnect from their old parents, and then reconnect at their new parents' locations.

As illustrated in our poster, our static layouts, use a variant of what Lin and Yen call "a balloon drawing subtree with non-uniform size" [LY05] (flattened-out cone drawings [CK95, JP98]) rather than the radial layout of Eades [Ead92]

M. Kaufmann and D. Wagner (Eds.): GD 2006, LNCS 4372, pp. 442–445, 2007.

and Yee et al. Our variant balloon drawing method places children not around the entire balloon, but rather on the centrifugal semi-circle of the balloon that lies outside the parent's balloon. This allows us to guarantee that the distance from a nonroot node to the root monotonically increases with the depth of the node, even though balloon layouts forego the stronger invariant that all nodes of a given depth are equidistant from the parent. It also allows us to replace node and link diagrams with solid-looking structures with highly idiosyncratic silhouettes, and with sub-structures that can be gracefully detached and moved, to striking visual effect, from the parent in an old spanning tree to the new parent in the new spanning tree.

Thus, rather than adopt the classical balloon tree convention of drawing each node as a point lying on the perimeter of its balloon, we draw it as a hemisphere covering the node's balloon. Building on Biederman's [Bie87] and Irani and Ware's [IW03] research on the human visual system's pre-attentive capacity to construct shape representations from shading and silhouette, we also shade the hemispheres to emphasize the idiosyncratic form of each structure. (Empirical work will be required to determine whether a visualization this strange is useful, but the work of Irani and Ware suggests that such idiosyncratically shaped three-dimensional forms enhance memory and perception, especially by novices, of underlying graph relationships.)

Our animation algorithm is, to our knowledge, novel. Given a graph drawn according to a breadth-first spanning tree (hereafter known as the "old drawing") and a node chosen to be the new root, the algorithm:

1. Calculates a breadth-first spanning tree rooted at the chosen node.
2. Calculates a new drawing based on the new spanning tree. Stores the angle and distance from each nonroot node to its new parent in the new layout (in effect, each such node's parent becomes the origin of the coordinate system that holds the node's position) we call this distance and angle the *new relative coordinates* of the node.
3. Calculates for each nonroot node the angle and distance in the *old drawing* from its *new* parent. We call these the *old relative coordinates.*

Then the algorithm generates each frame of the animation by interpolating between the old and new relative coordinates and calculating the absolute position of each node by recursively calculating the absolute position of its parent. More details are in [PHS06].

References

[Bie87] Irving Bierderman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[CK95] J. Carrire and R. Kazman. Research report: Interacting with huge hierarchies: Beyond cone trees. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 74–81, 1995.

[Ead92] Peter Eades. Drawing free trees. *Bulletin of the Institute of Combinatorics and its Applications*, 5:10–36, 1992.

[GADM04] S. Grivet, D. Auber, J.P. Domenger, and G. Melançon. Bubble tree drawing algorithm. In *International Conference on Computer Vision and Graphics*, pages 633–641, 2004.

[HMM00] Herman, G. Melançon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[II90] J. Q. Walker II. A node-positioning algorithm for general trees. *Software: Practice and Experience*, 20(7):685–705, 1990.

[IW03] Pourang Irani and Colin Ware. Diagramming information structures using 3d perceptual primitives. *ACM Transactions on Computer-Human Interaction*, 10(1):1–19, 2003.

[JP98] Chang-Sung Jeong and Alex Pang. Reconfigurable disc trees for visualizing large hierarchical information space. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 19–25, 1998.

[LRP95] John Lamping, Ramana Rao, and Peter Pirolli. The hyperbolic browser: A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.

[LY05] Chun-Cheng Lin and Hsu-Chun Yen. On balloon drawings of rooted trees. In *GD '05: Revised Papers from the 13th International Symposium on Graph Drawing*, pages 285–296, London, UK, 2005. Springer-Verlag.

[MH98] Guy Melançon and Ivan Herman. Circular drawings of rooted trees. Technical Report INS-R9817, CWI (Centre for Mathematics and Computer Science), Amsterdam, The Netherlands, 1998.

[Mun97] Tamara Munzner. H3: Laying out large directed graphs in 3d hyperbolic space. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 2–10, 1997.

[PHS06] Andrew Pavlo, Christopher M. Homan, and Jonathan Schull. A parent-centered radial layout algorithm for interactive graph visualization and animation. Technical Report `http://arxiv.org/abs/cs/0606007`, arXiv.org, 2006.

[RT81] Edward M. Reingold and John S. Tilford. Tidier drawings of trees. *IEEE Transactions on Software Engineering*, 7(2):223–228, 1981.

[TM02] Soon Tee Teoh and Kwan-Liu Ma. Rings: A technique for visualizing large hierarchies. In *GD '02: Revised Papers from the 10th International Symposium on Graph Drawing*, pages 268–275, London, UK, 2002. Springer-Verlag.

[Wil99] Graham J. Wills. NicheWorks — interactive visualization of very large graphs. *Journal of Computational and Graphical Statistics*, 8(2):190–212, 1999.

[YFDH01] Ka-Ping Yee, Danyel Fisher, Rachna Dhamija, and Marti A. Hearst. Animated exploration of dynamic graphs with radial layout. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 43–50, 2001.

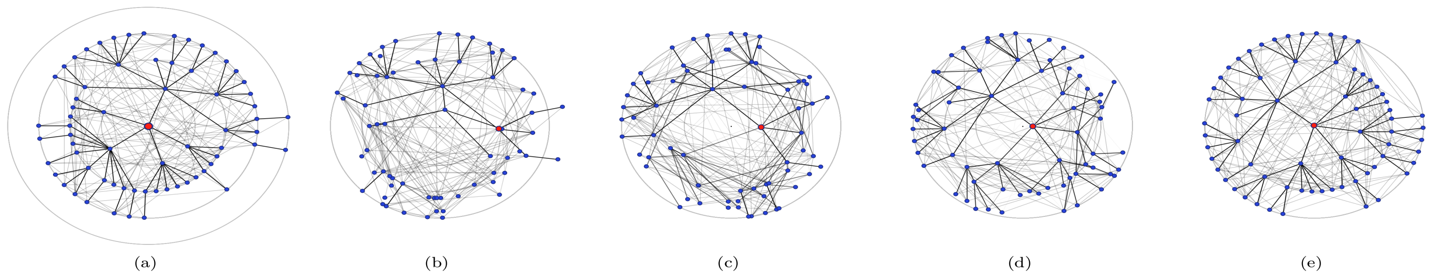

Fig. 1. An example transition between two spanning trees drawn using Eades' radial layout algorithm. Nodes are placed on rings concentric to the root node that correspond to their depth in the tree. Thick edges denote spanning tree edges; thin edges denote non-spanning tree edges. In Figure 1(b) a new root node is selected, the thick lines are redrawn to denote edges of the new spanning tree, and the graph begins to transition to the final layout for the new spanning tree, shown in Figure 1(e).

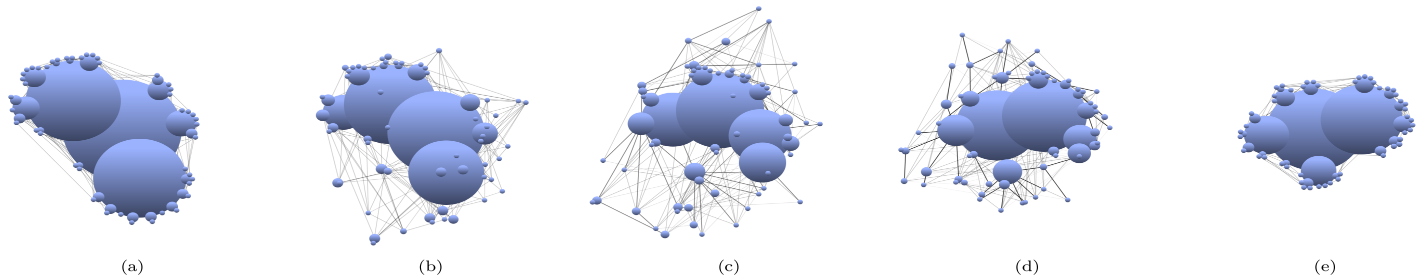

Fig. 2. In this alternative radial layout of the same tree as Figure 1, nodes are drawn as "balloons" where each child node is positioned on the centrifugal semi-circle of the balloon that lies outside its parent. No thick edges need to be drawn in Figure 2(a), because edges as well as nodes are denoted by the overlapping balloons. In Figure 2(b), a new node is selected, thick edges are used to denote the edges of the new spanning tree edges, and the graph begins to transition to the final layout for the new spanning tree in Figure 2(e), in which explictly drawn spanning tree edges are again unnecessary.