

Overlaying Graph Links on Treemaps

Jean-Daniel Fekete*
INRIA Futurs/LRI

David Wang, Niem Dang, Aleks Aris, Catherine Plaisant†
Human-Computer Interaction Laboratory
University of Maryland

Abstract

Every graph can be decomposed into a tree structure plus a set of remaining edges. We describe a visualization technique that displays the tree structure as a Treemap and the remaining edges as curved links overlaid on the Treemap. Link curves are designed to show where the link starts and where it ends without requiring an explicit arrow that would clutter the already dense visualization. This technique is effective for visualizing structures where the underlying tree has some meaning, such as Web sites or XML documents with cross-references. Graphic attributes of the links – such as color or thickness – can be used to represent attributes of the edges. Users can choose to see all links at once or only the links to and from the node or branch under the cursor.

CR Categories: H.5.2 [User Interfaces], E.1 [DATA STRUCTURES Graphs and Networks]

Keywords: Information Visualization, Treemaps, Bézier Curves.

1 Introduction

The general problem of graph drawing and network visualization is notoriously difficult. Instead of tackling it directly, we present a method that starts from a decomposition of the graph into a tree and a set of remaining edges. This decomposition can always be done but some data structures are easier and more meaningful when decomposed that way. For example, a Web site is almost always organized in a hierarchical file system corresponding to a meaningful hierarchical organization. An XML document with cross references (e.g. a table of contents, footnotes, and index entries) can also naturally be decomposed into an XML tree plus the cross reference

Our method uses Treemaps [1] for visualizing the tree structure and overlays links to represent the remaining edges (Figure 1). We initially used straight lines connecting the source and destination item centers but the results were very cluttered due to the superposition of lines [2]. There have been several attempts at simplifying the representation of links on node-link diagrams. Becker et al. proposed half-lines for that purpose [3], using a straight line from the source but stopping it halfway to the destination, avoiding drawing arrowheads. We have designed a novel method for drawing the links using a curved representation where the offset of curvature indicates the direction of the link.

* bat 490, Université Paris-Sud, F91405 ORSAY Cedex, FRANCE,

Jean-Daniel.Fekete@inria.fr

† UMIACS- HCIL, A.V. Williams Building, University of Maryland, College Park, MD 20742, U.S.A. plaisant@cs.umd.edu.

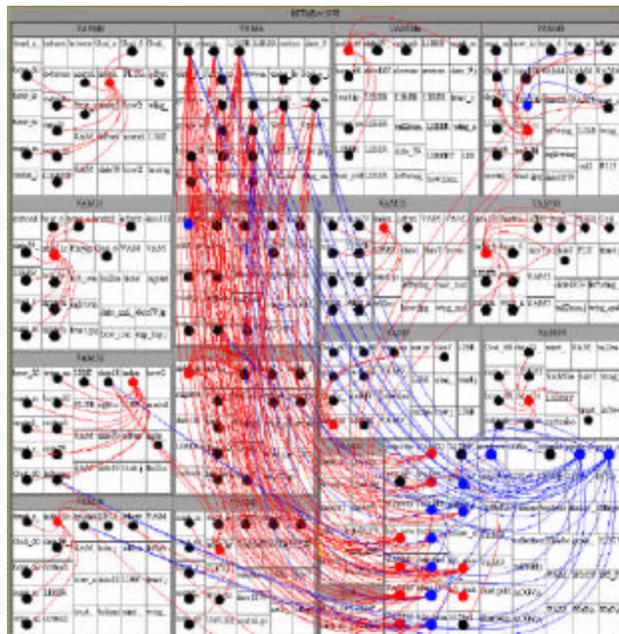


Figure 1: Directory structure of a Web site visualized as a Treemap with external links overlaid as curves. Blue curves are HTML links, red curves are image links.

The curved link is modeled using a quadrilateral Bézier curve (Figure 2a). The first and last points are placed at the middle of the source and target regions. The second point is placed at a distance halfway from the source position from the first point and is on a line forming an angle of 60 degrees from the source to the destination line (Figure 2b).

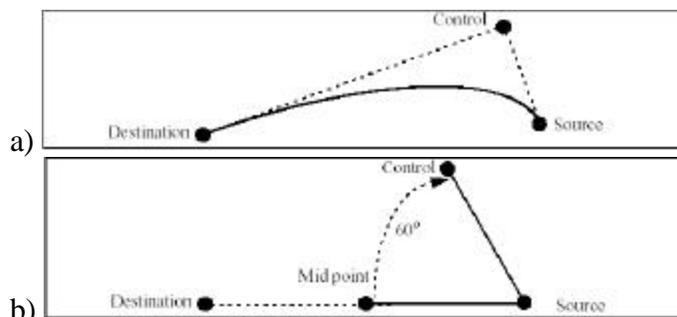


Figure 2: (a) The three control points of a quadrilateral Bézier curve and (b) the computation of the second point.

Using this method, the curve is not symmetrical but shifted towards the source and this shift is easy to recognize visually. When two items reference each other, the two curved links remain clearly distinguishable and are not occluded. Figure 3 shows several HTML pages pointing at each other's, where links are easy to follow. Links can also be colored depending on attributes associated with the edges. We have experimented with colors but line width could also be used within reasonable limits.

2 Interaction

The links visibility can be static or dynamic. By default, the visibility is static: all the links are shown. This setup is useful as an overview but can clutter the treemap representation, making item labels hard to read for example. When users want to focus on the tree structure or on a specific region of the visualization, they can select the dynamic visibility of links. This setup only shows links starting from or arriving at items that have been selected (nodes or branches), when a selection exists. Otherwise, it tracks the mouse and shows links starting from and arriving at the item under the pointer. This last setup is useful for dynamically exploring a visualization and watching connections between areas of interests.

The curved links visualization has been integrated into the latest version of the University of Maryland "Treemap 4.1" [5]. The data consists of a Treemap data file and a link file specifying the non-hierarchical links to be visualized. Treemap 4 can display data with a fixed variable depth hierarchy or - if the data does not include a fixed hierarchy - users can interactively create a hierarchy using the new "flexible hierarchy" feature of Treemap 4. When users load the link links the links are visualized over the Treemap visualization of this dataset (see Figure 4). Treemap implements several treemap layouts [2] and allows for dynamic queries based on attributes associated with the nodes as well as attributes computed from the data topology such as depth or degree or nodes. Treemap also allows users to select nodes or branches and hide them, which can be useful to hide nodes that have a large number of links (e.g. the company logo) and make the rest of the display more usable.

3 Conclusion and Future Work

Some graphs that can be meaningfully visualized as an underlying tree structure with overlaid links. For these graphs, we present the tree structure using a Treemap layout and overlay the edges as curved links. This graph visualization is therefore an enhancement of a tree visualization and is simpler to control and understand than general purpose network visualization systems.

We could generalize the idea and provide a tool that would transform any graph into a tree and a remaining set of edges. There are many algorithms to perform a tree extraction from a graph.

One limit of our current implementation is that the Treemap program is meant to visualize trees and doesn't currently perform dynamic queries or assign link visual attributes from edge attributes. This would belong to a more general graph visualization system. We are currently integrating this technique into the InfoVis Toolkit [6], which can visualize trees as well as graphs and will be able to integrate those features more easily.

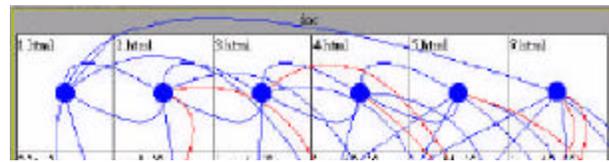


Figure 4: Details of six HTML files visualized in a Treemap with cross links without occlusions.

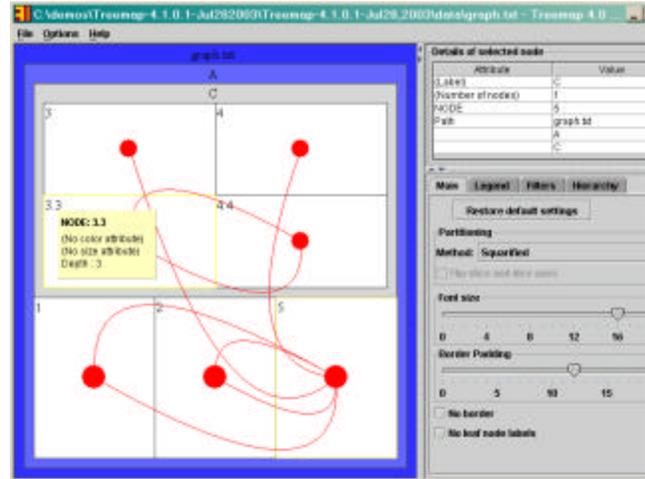


Figure 3: Integration of the curved links inside the Treemap system

Acknowledgments

This work has been supported by ChevronTexaco.

References

- [1] JOHNSON, B. AND SHNEIDERMAN, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures, Proc. IEEE Visualization' 91 (1991) 284 – 291, IEEE, Piscataway, NJ
- [2] BEDERSON, B.B., SHNEIDERMAN, B., AND WATTENBERG, M. *Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies*, ACM Transactions on Graphics (TOG), 21, (4), October 2002, 833-854..
- [3] R. BECKER, S. EICK AND A. WILKS. Visualizing Network data, In IEEE Transactions on Visualization and Computer Graphics, vol 1,no. 1, March 1995.
- [4] WANG, D, Graph Visualization: An Enhancement to the Treemap Data Visualization Tool, University of Maryland InfoVis class project report, <http://www.cs.umd.edu/class/spring2002/cmsc838f/Project/treemap.pdf>
- [5] Treemap 4.1, <http://www.cs.umd.edu/hcil/treemap>
- [6] The InfoVis Toolkit, <http://www.lri.fr/~fekete/InfovisToolkit>