Editor:
Miguel Encarnação

Treevis.net: A Tree Visualization Reference

Hans-Jörg Schulz
University of Rostock

Many people in the information visualization and graph-drawing communities consider tree visualization (see the sidebar) a solved problem. Although Kim Marriott and Peter Stuckey have shown that finding an optimal tree layout can be an NP-complete problem,[1] reasonably good tree layouts can nevertheless be computed efficiently in terms of runtime and screen space utilization. In the course of the search for heuristics to generate ever-tidier tree layouts, the comparatively simple problem of transforming parent-child relationships into graphical representations has been solved many times and is still the subject of information visualization research. Researchers have explored and published almost every way of arranging a tree's nodes in 2D and 3D; encoding them in different shapes or forms; and folding, unfolding, or otherwise interactively manipulating them.

The plethora of tree visualization techniques poses challenges to researchers and developers. Researchers, especially those new to the field, have no way of knowing every tree visualization that has been published, even over just the last two decades. So, they often reinvent existing techniques. Without pointing fingers—my colleagues and I have done our fair share of unwittingly reinventing visualizations—I've noticed that the published tree visualizations include a number of such reinventions. This is hardly surprising because it's almost impossible for peer reviewers as well to have a complete overview of prior research.

The same holds true for developers who implement tree visualizations for their customers, but with potentially direr consequences. Developing something that already exists could lead to ugly intellectual-property issues. And even though it seems like a good starting point to assume that something similar to your own idea has already been done, finding that similar technique can be extremely difficult.

However, opportunities also exist. The long history and remarkable coverage of the design space offer the opportunity to step back, take a look at the bigger picture, and learn from it. For example, we can identify recurring design patterns. Moreover, we can trace back the evolution of our modern visualization techniques to the visual archetypes that might have inspired them.

To address the challenges and exploit the opportunities, we must make a laborious but important first step: we must collect existing tree visualization techniques and form a reference for them that's as complete as possible. This is where the treevis.net project comes into the picture.

Hunting and Gathering Tree Visualizations

In early 2010, I set out to ramble through the available tree visualization literature and websites. Most tree visualizations could readily be excerpted from conference proceedings and journals. From these, I slowly built a "convex hull" by seeking those papers cited by the ones I found and those that cited the found ones. But this covered only the scholarly publications. Much harder to hunt down were the visualizations that appear on Flickr

Tree Visualization

Tree visualization (sometimes called *hierarchy visualization*) is a branch of information visualization dedicated to the graphical representation of connected, acyclic graphs—*trees*. Tree structures are common in many aspects of everyday life, such as ancestry (family trees) or file system organization (directory trees). Most tree visualizations are developed for *rooted trees*, which contain a selected top element, the *root node*; intermediate elements, the *internal nodes*; and bottom elements, the *leaves*. Drawing on the family tree metaphor, nodes standing in direct relation are called the *parent* node (the node closer to the root) and *child* node (the node further from the root).

Edge Representation

Edge representation can be explicit, with clearly drawn links connecting the nodes, or implicit, without drawn links but with positioning that represents the nodes' connection (see Figure A). In rare cases, hybrid variants use both types of representation for different parts of the tree—for example, to emphasize structural differences between subtrees or internal nodes and leaves.

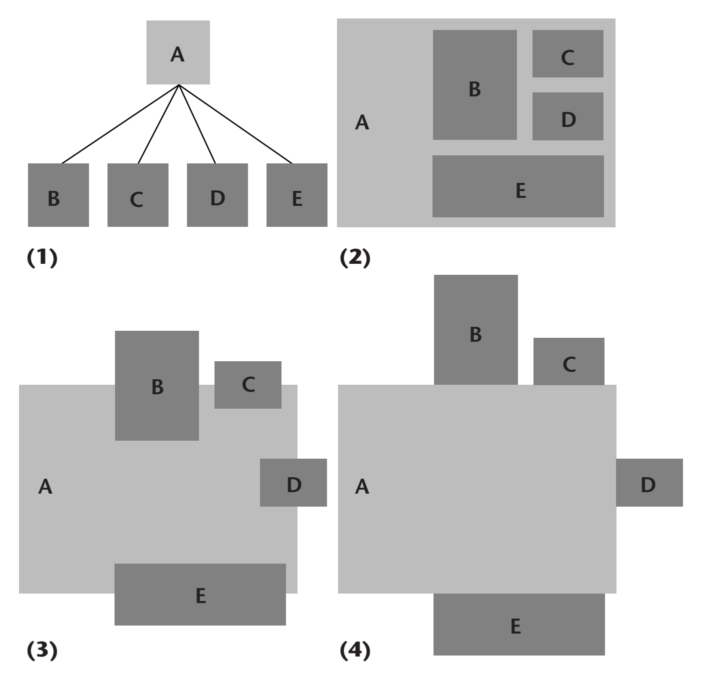

Figure A. The edge representation is a main characteristic that lets us break down the large number of tree visualizations and differentiate them. Essentially, four types of edge representation are known: (1) explicit, node-link; (2) implicit, inclusion; (3) implicit, overlap; and (4) implicit, adjacency.

or in blog posts. Those visualizations are often the most interesting; they tend to be the "wilder" ideas that have a hard time getting published, let alone successfully evaluated in comparative user studies.

After collecting the tree visualization techniques, I validated them because not all visualizations whose name includes "tree" are designed specifically for trees. Some span a wider range and display trees with additional cross edges or even full-blown networks. I omitted these because they're out of the tree visualization scope.

This first overview produced a collection of more than 100 techniques. To organize them, I chose the three design axes:

- dimensionality (2D, 3D, or hybrid),
- edge representation (explicit, implicit, or hybrid—see the related sidebar), and
- node alignment (radial, axis-parallel, or free).

These common properties can be determined for almost every tree visualization, no matter how strange it might look.

Susanne Jürgensmann and I presented the results at IEEE VisWeek 2010 as a visual survey and bibliography on a poster.[2] This poster's reception was out-and-out positive; it has been downloaded frequently and now decorates computer science buildings around the world.

However, updating and maintaining the poster's tight integration of content and design proved extremely cumbersome (to say the least). So, in May 2011, I launched a Web-based version of the survey at treevis.net; it now includes more than 180 visualization techniques. It separates content and display so that I can fix errors and add new techniques in minutes. The website displays the techniques in a compact mosaic-like form. Users can access details on demand and employ filters to reduce the set of shown techniques to a desired selection—for example, only 3D techniques. This makes exploring the techniques much more interactive than with a poster printout.

For a technique's primary resource, the website normally chooses that technique's first scholarly peer-reviewed publication. If no such publication exists, it chooses the first poster presentation, arxiv.org preprint, student paper, blog entry, or webpage the technique appeared on. This is a bit unusual and not quite in tune with established citing practices. Such practices give preference to the most recent publication on a technique because (hopefully) all the glitches in the earlier publications have been eliminated. Yet, the first publication indicates how long a technique has been around and whether it predates another similar technique. Treevis.net users can find later publications by looking at the related publications listed in the detail view. All references are linked via their DOI to the publisher's website, so the actual publications are only one mouse click away.

The treevis.net website has many more features, including a downloadable version of the website for offline and classroom use, the original poster as a high-resolution PDF file for plotting, and a BibTeX file with all the collected references. The Twitter account @treevisproject reports on webpage changes.

Using Treevis.net

Because the main challenge of the large number of visualization techniques lies in finding the ones of interest to the user, I conceived the website to do just that. It's shaped mainly as a gateway, which lists the techniques with only as much additional information and interaction functionality as the user needs to make an informed choice on which techniques to read further about in the original publications. Consequently, different ways of defining which techniques are of interest lead to different ways of using the site. From my experience, three types of searches are the most common.

The first is searching for a known technique to find out who the authors are or when it was first published. Researchers perform such searches, for example, to quickly double-check the spelling of names on lecture slides or to copy and paste the original paper's title. The easiest way to do this search is to type a known part of a technique's or an author's name into the full-text search box. The box also lets the programming-savvy use JavaScript regular expressions, such as `19\d{2}` to find all techniques from the 20th century, with `\d{2}` matching any two-digit sequence.

The second is searching for a technique with a look and feel similar to a given one. Both researchers and developers perform this kind of search to ensure that a new visualization technique is truly novel or that all close look-alikes have been correctly cited as related work. To perform such searches, users must scan all the thumbnails on the website. Using the buttons at the top of the interface, they can filter the thumbnails by choosing the kind of technique they're looking for—for example, only 3D node-link visualizations. This makes sense because any technique similar to another technique must share that technique's characteristics.

The third is searching for a suitable technique for a given dataset or application, as developers often do, possibly with their clients or customers. For this search, users employ the filter buttons to perform step-wise refinement. This refinement starts with the entire set of techniques. After reviewing the options, developers make the necessary design decisions by selecting a suitable dimensionality, edge representation, and alignment. In the end, a set of appropriate techniques remains, which the developer can use directly or as a basis for a new technique.

The benefits of a hand-curated visual index such as treevis.net become even more apparent if you try to perform these searches with a standard Web search engine. Even in the age of full-text book searches and reverse image searches, it's a long way from remembering a visualization you once saw to the publication it appeared in. And even if you know a technique's name, searching the Web to find all the related publications still requires much more digging than simply looking it up on treevis.net.

First Observations

The site also provides a good starting point for taking a look at the bigger picture. My colleagues and I at the University of Rostock have already used this opportunity to look for common design and use patterns for the subdomain of implicit tree visualizations. Our related paper discusses many such visualizations and evaluates their suitability for structure-centric and attribute-centric tree representations.[3] It also introduces ways to combine visualizations to form hybrids. This was made possible by having all techniques assembled on a single webpage that neither judges nor weighs the different visualizations. Any novel tree visualization is welcome, from the serious to the absurd, so that hopefully the website captures even the most remote corners of the tree visualization design space.

Even if you know a technique's name, searching the Web to find all the related publications still requires much more digging than simply looking it up on treevis.net.

Displaying all these techniques side-by-side also lets users seek visual archetypes. For example, Figure 1 illustrates the evolution of radially stacked tree visualizations. In this case, the fundamental radial design turned out to be much older than most people think. The increasing digitization of older works is making them readily available to a larger audience. So, more prior art will likely appear of which our young, fast-moving information visualization community is unaware. Of course, that doesn't belittle the contribution of the "reinventions" of our time. You can't compare hand-drawn diagrams for a few dozen items from the precomputer era with the intricate algorithms developed to automatically lay out trees with millions of nodes while permitting interactive navigation and manipulation of these vast datasets.

The abundance of tree-visualization solutions has another interesting effect. Because tree visualization is such a mighty hammer, people shape their problems into nails. The graph-drawing and graph visualization community has done this for ages—for example, by extracting a spanning tree from a network, using a tree layout for that spanning tree, and adding the remaining edges onto it. This might not yield the best possible layout, but the speedup is phenomenal.

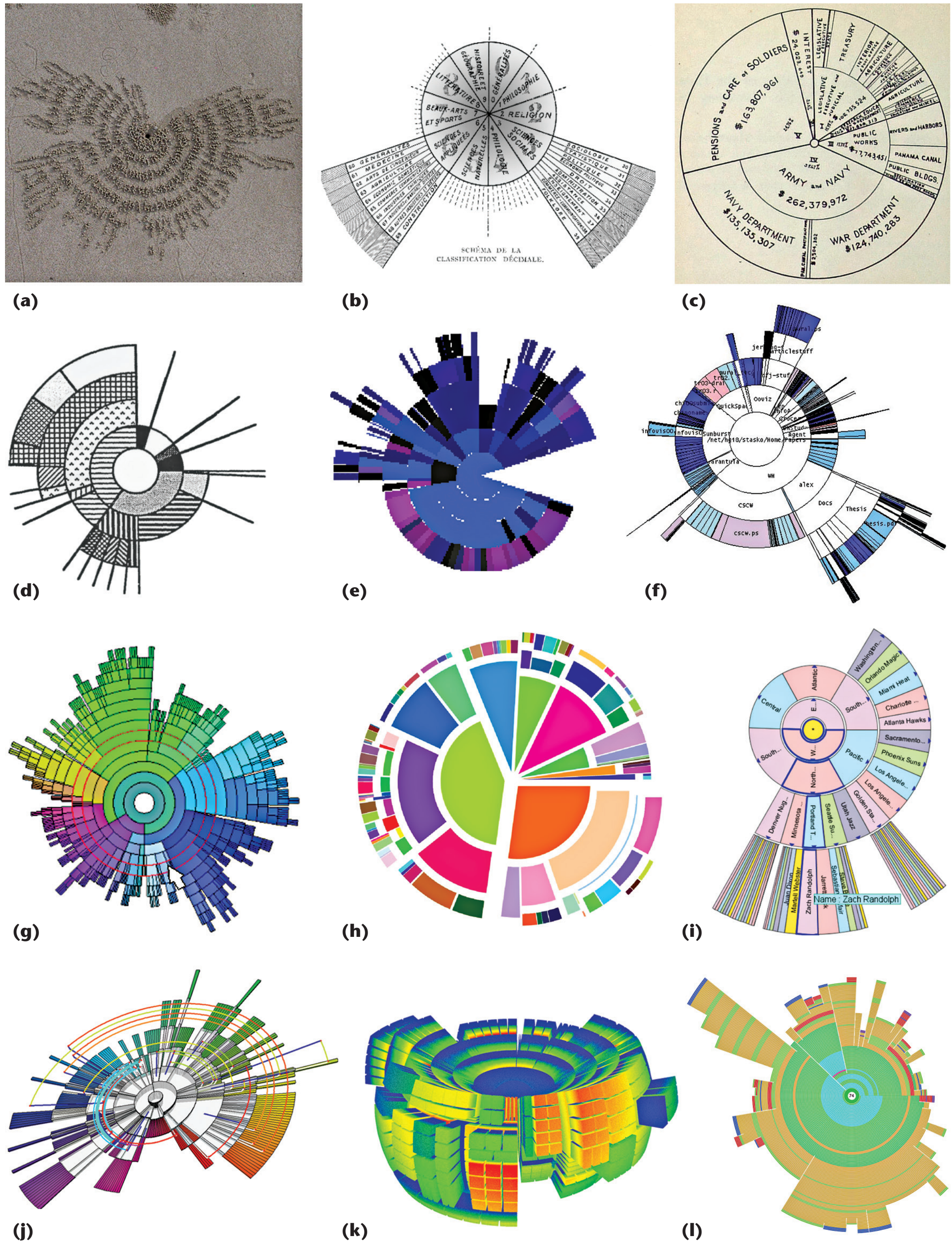

Figure 1. The evolution of radially stacked tree visualization. (a) Sand bubbler crab pattern (jkr1812 via Flickr). (b) Universal decimal classification (1905, P. Otlet). (c) Hierarchical sector chart (1921, Am. Soc. Mechanical Engineers). (d) Spoked polar tree map (1993, B. Johnson). (e) Aggregate tree map (1998, M. Chua). (f) Sunburst (2000, J. Stasko). (g) Interring (2002, J. Yang et al.). (h) PieTree (2006, R. O'Donnell et al.). (i) FanLens (2008, X. Lou et al.). (j) Enhanced radial space-filling layout (2009, M. Jia et al.). (k) 3D sunburst wheel (2010, H.-J. Schulz and S. Hadlak). (l) Trevis calling-context tree ring chart (2010, A. Adamoli and M. Hauswirth). The fundamental radial design turns out to be much older than most people think. (All images not in the public domain are used with permission.)

Recently, researchers have "hierarchized" other kinds of data into tree structures to exploit the wide range of tree visualizations. In 2011, Daniel Engel and his colleagues presented a fascinating way to transform multivariate data into a tree.[4] In 2009, Rimon Elias and his colleagues even transformed random data into hierarchical data just to visualize it as such.[5] So, the tree-visualization field is drawing more and more attention even from researchers in other fields, making treevis.net a valuable resource for them as well.

Outlook and Future Work

So that treevis.net can finally leave beta status, I plan to integrate several much-needed features. On top of the to-do list are mainly improvements to the search capabilities because searching for visualizations is how most people use the website. For example, I plan to add more intuitive full-text searching with wildcards instead of regular expressions. I also plan to add the ability to pass filters and search strings via the URL so that users can link to a found selection of interest and bookmark it.

Another important aspect of searching is a layout of visualization thumbnails that's engaging to explore yet still meaningful. The thumbnails' current grid-like arrangement is ordered simply by year. I'll soon replace it with a layout similar to bubble maps. The layout will try to mimic the original poster in that it will place techniques with common characteristics closer together, effectively forming a region for each combination of dimensionality, edge representation, and alignment. This will make it easier to find visualizations with given characteristics without having to filter. I also plan to extend the detail view for each visualization technique so that users can import bibliographical references directly to the Zotero and Mendeley reference management tools.

Although realizing these features is clearly my job, some issues require help from the community of visualization researchers and developers. The most pressing issue is that not all publications contain images the website can use as meaningful thumbnails. For example, this is the case with several graph-drawing publications, which contain mathematical proofs of how good the layouts are but never show an example. Another case is tree visualizations published as patent applications. These do contain figures, but they're usually meaningless, grainy black-and-white schematics showing how a monitor is hooked up to a PC to illustrate the "sophisticated apparatus" on which the visualization runs. The figures say nothing about the tree layout's actual appearance. So, I ask the researchers and developers who authored such publications or patents to contribute a screenshot or two of their visualizations so that the website can list their work.

I'm optimistic that this call will be heard. Treevis.net has already grown into a community effort. Many visualization authors are sending me preprints of their new tree visualization techniques. Others are pointing out additional resources and information. Such support has shaped the project into what I believe is one of the most up-to-date, complete, and accurate references on tree visualizations. I encourage everyone to join in—by recommending overlooked tree visualizations, downloading the data and creating mashups, or simply using the site.

References

1. K. Marriott and P. Stuckey, "NP-Completeness of Minimal Width Unordered Tree Layout," *J. Graph Algorithms and Applications*, vol. 8, no. 2, 2004, pp. 295–312.
2. S. Jürgensmann and H.-J. Schulz, "A Visual Survey of Tree Visualization," poster presentation at VisWeek 2011; www.informatik.uni-rostock.de/~hs162/pdf/treevis_abstract.pdf.
3. H.-J. Schulz, S. Hadlak, and H. Schumann, "The Design Space of Implicit Hierarchy Visualization: A Survey," *IEEE Trans. Visualization and Computer Graphics*, vol. 17, no. 4, 2011, pp. 393–411.
4. D. Engel et al., "Structural Decomposition Trees," *Computer Graphics Forum*, vol. 30, no. 3, 2011, pp. 921–930.
5. R. Elias, M. Al Ashraf, and O. Aly, "From Random to Hierarchical Data through an Irregular Pyramidal Structure," *Graph-Based Representations in Pattern Recognition*, LNCS 5534, Springer, 2009, pp. 324–333.

Hans-Jörg Schulz *is a postdoctoral researcher in the University of Rostock's Computer Graphics Department. Currently, he's on a research visit at Graz University of Technology's Institute for Computer Graphics and Vision. Contact him at contact@treevis.net.*

Contact department editor Miguel Encarnação at lme@computer.org.