Using TreeMaps and Hyperbolic Trees for Statistical Medical Data Visualization

Christophe Bouthier

Using Treemaps and Hyperbolic Trees for Statistical Medical Data Visualization

Christophe Bouthier

Abstract— **Medical studies often generate a large set of statistical data. Finding relationships and patterns in those data is a painful and error-prone task. Graphical representations has been recognized as a powerful tool for data analysis. In this paper, we will introduce two visualization techniques, TreeMaps and Hyperbolic Trees, that could be used for statistical medical data analysis.**

Keywords— **Statistical Medical Data, Information Visualization, Treemap, Hyperbolic Tree**

I. Introduction

Medical studies often generate a large set of statistical data. Those data are not useful by themselves. It is the relationships and the patterns inside that have a medical value [1]. Finding relationships and patterns inside a wide set of data could be done either automatically, with statistical mathematical formulas and algorithms, or visually, by looking directly at the data. Those two ways are complementing rather than opposing each other. To find an interesting result with statistical methods, one has to have an intuition of the result he want to find. Such an intuition is often found by looking directly at the data. So, the mathematical means is often used as a confirmation, a formal proof, that the result *seen* in the data is effectively a scientific result.

Graphical representation has been recognized as a powerful tool to help people find interesting relationships and patterns inside a wide set of data. Visual data analysis became a research domain, called *Information Visualization*. Some graphical representations are well known, like data table and plotted graph. Others are more mysterious in the eyes of everyday people, and rarely seen in any tools.

This paper is about two of those rarely-used data visualization techniques, namely the Treemap visualization and the Hyperbolic Tree visualization (also called *Hypertree*). Those two visualization techniques have in common the fact that they are made especially to represent large hierarchies of data. They could so be used for statistical medical data analysis. We will first describe these techniques and how they operate. Then, we will insist on the advantages that they offer versus others mode "classical" visualization techniques. We will speak also of their disadvantages, and why Treemaps and Hyperbolic Trees are really complementary. Finally, we will describe what those two techniques together can do for the statistical medical data analysis field.

Christophe Bouthier, LORIA, Nancy, France. E-mail: bouthier@loria.fr .

II. Background on Visualization Techniques

As Ben Shneiderman formulates it in [2], the ultimate goal of every visualization technique is: "Overview first, zoom and filter, then details on demand": the visualization should first give an overview of the whole set of data, then let the user restrict the set of data on which the visualization is applied, and finally give more details on the part the user is interested in.

A. TreeMap

The treemap visualization technique has been invented by Ben Shneiderman, at the university of Maryland, in 1991, to resolve a very common problem [3], [4]. As he was faced with a too common "no space left on device" problem, he wanted to know who (or what) was using most space of his hard drive. The treemap visualization represents a hierarchy tree of data as a set of nested rectangles (fig. 1). Each node of the tree is represented as one rectangle. The size of the rectangle and its filling depend directly from properties of the node. For example, for file system visualization, the size of the rectangle could depend on the size of the represented file, and the color filling on its last modification date.

The hierarchy information is conveyed in the way the rectangles are nested. In the original TreeMap approach, a rectangle is divided between its children's' rectangles in only one direction, horizontal or vertical, based on the deepness level of the node. But with this algorithm, small nodes are often very thin and elongated, almost a line, making them difficult to see or distinguish. Another slicing algorithm, "squarified

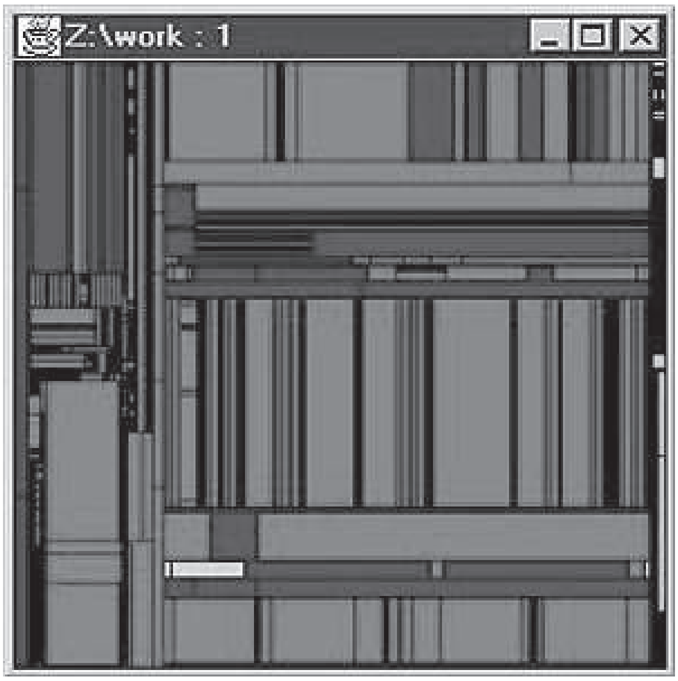

Fig. 1. An example of Treemap Visualization

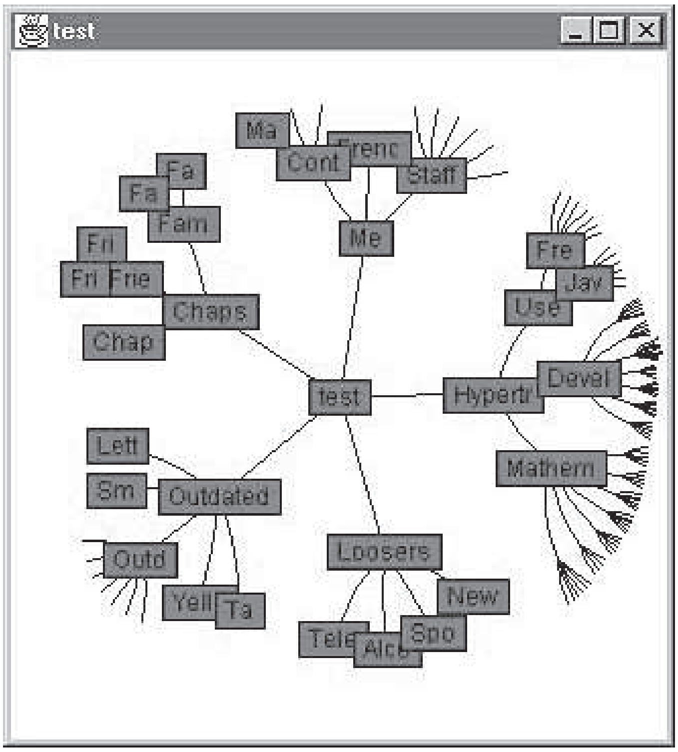

Fig. 2. An example of Hyperbolic Tree Visualization

treemap" [5], tries to keep the aspect ratio of rectangles as low as possible — thus "squarifying" rectangles — but at the expense of hierarchical information. Little nodes are easier to read, but it is difficult to see at which hierarchy level is a particular node. A way to enhance the hierarchy level information is by using "nested treemap", where a border is created around the rectangle of a node containing others nodes.

Since its first publishing, a lot of work has been done on the treemap visualization [6]: some works on new algorithms for a better presentation of the information, some on how to animate treemaps, based on well-known references [7], and others looks at new field were treemap could be used.

B. Hyperbolic Tree

The hyperbolic tree visualization technique has been invented by John Lamping and Ramana Rao [8], [9], at the Xerox Palo Alto Research Center (Xerox PARC), in 1995, by looking at a Esher woodcut, which use hyperbolic geometry properties to provide a "fish-eye" effect. The goal of the hyperbolic tree is to display and manipulate large hierarchical structure.

The hyperbolic tree is a focus+context visualization technique: it gives a detailed view on a small area (*focus*), while still providing an overview of the whole structure (*context*). This is obtained by first laying out in an uniform way the hierarchy as a tree, on a non-euclidean hyperbolic plan (hence the name). Then, by using the "Poincaré model" (from the name of the French mathematician who have discover this model in the last century), this layout is mapped back on the euclidean plane, into the unit disk (the disk of radius 1). As the whole hyperbolic plan is mapped into the unit disk, a space distortion appears: the mode a node is near the border of the disk, the less room it has (fig. 2). Thus, the center of the disk provides a detailed view on some nodes of the tree (*focus*), and the border of the disk provides a overall view of the structure of the rest of the tree (*context*).

Contrary to the treemap, the hyperbolic tree is an interactive technique. The tree could be dragged to put a part of it in the center of the disk, thus putting focus on this part. A node can be clicked to be put directly in the center, with smooth animation to provide users with feedback on the process.

Not too much work has been done on the hyperbolic tree visualization, by comparison to the treemap visualization. One reason to this is that Xerox has heavily patented the hyperbolic tree visualization. Fortunately, software patent are still illegal in Europe. Despite Xerox patents, some complementary research work has been done on the hyperbolic tree visualization, like using hyperbolic tree visualization to represent links between objects and methods in a Java profiler [10]. Another work, more recent, use the hyperbolic tree as as information browser in comparison with more conventional information or file browser to test the information scent theory of information foraging [11]. Finally, some work has been done to represent hyperbolic tree in 3D, by projecting not on the unit disk but on the unit sphere [12].

III. Pros and Cons of Visualization Techniques

A. TreeMap

A.1 Pros

The main advantage of the treemap visualization is its efficient use of display space. As it maps a whole hierarchy into a rectangle, it allows global view of even large hierarchies. This global view allows the user to compare nodes, based on two characteristics of their corresponding rectangles: their relative sizes and their fillings. Those two characteristics could be mapped to whatever characteristic of the corresponding node, with one restriction: the "size" characteristic should be cumulative, meaning that the "size" of a node that have children should be at least equal to the sum of its children's "sizes".

The global view of the whole set of data could be described as a "gestaltic view" [13] of the set of data, were the global view has properties that the sum of local views do not have. This gestaltic view of the characteristics of the whole hierarchy could be used either as a map (hence the name) to search for a specific node, based on some characteristic, or to get a global awareness of the data, like for example getting a peripheral awareness of the state of a system [14].

A.2 Cons

The main disadvantage of the treemap visualization technique is that it is not an intuitive visualization. Someone who has never seen a treemap before has to spend 10 to 15 minutes to learn how to read it. But once the learning has been done, the treemap is usually found useful and easy to understand. Another disadvantage is that the treemap visualization is limited to strict tree hierarchical data, without extra links between unrelated nodes.

Several mechanisms exist to bring navigation capacity to the treemap visualization. One could, for example, select which characteristic to represent in the treemap. A dynamic filtering on the characteristic is also possible, the treemap showing then only the relevant nodes. Finally, a zoom could be implemented so that a click on a node zooms the treemap on a deeper level in the hierarchy. But still the treemap strength is not in its navigation capacity, but in its global gestaltic overview of the whole hierarchy.

B. Hyperbolic Tree

B.1 Pros

The hyperbolic tree visualization technique is particularly adapted for navigation into large hierarchy. Several features help the user during this navigation task.

First, the hyperbolic tree maps the whole hierarchy into the unit disk. Even if, at some distance, some nodes are no more drawn, crunched by the hyperbolic distortion, a lot of nodes are still visible, thus giving a context to the local focus. This provides the user with a main focal point in the whole hierarchy. The hyperbolic geometry also makes the navigation more efficient than in euclidean geometry, as the distance covered in each click or drag is greater, and there are more nodes displayed in each of the tree's displacement.

Secondly, the user can choose its own navigation: either freely by dragging the tree, or in a more structured way by clicking on nodes he/she wants to navigate from one to another. The animation of the tree helps to visualize the navigation path and provides additional information with respect to the navigation direction.

The hyperbolic tree visualization technique has been mainly made for tree hierarchical data, but it can also support extra links between nodes unrelated. And contrary to the treemap, this visualization is readily usable by new users, even if the hyperbolic distortion could disorient the user the first time.

B.2 Cons

The main disadvantage of the hyperbolic tree visualization is that, because of the hyperbolic distortion, some nodes might be too close of the border and therefore may not be visible at all. The hyperbolic tree also does not provide a complete hierarchical view of the whole system, not even an uniform one on the visible nodes. So, the hyperbolic tree visualization does not allow for a comparison between any node and the rest of the tree, and allows comparison between nodes only for sibling ones. However, it is possible to assign color to nodes to help differentiates sub-structures in the hierarchy.

C. TreeMap + HyperTree

The main strength of the treemap visualization is the gestaltic view of the whole hierarchy, thus allowing comparison of nodes characteristics. Its weaknesses are to support only strict tree hierarchy, and to be non intuitive for first-time users. The treemap visualization could be used whenever one needs a global understanding of a set of data.

The main strength of the hyperbolic tree visualization is when used to navigate large hierarchy, to support extra links between unrelated nodes, and to be intuitive even for first-time users. Its weakness is to not allow comparison between nodes and to not give a global view on the whole hierarchy, even if it gives an idea on its repartition. The hyperbolic tree visualization could be used whenever one needs to navigate in a large hierarchy.

The two visualization techniques are really complementary. The treemap visualization brings its global "gestaltic" overview of the whole hierarchy, and the hyperbolic tree is used to navigate into this hierarchy. This is why we propose here to use the combinaison, the symbiose of those two visualization techniques for the analysis of statistical medical data.

IV. Applications to statistical medical data analysis

The combination of treemap and hyperbolic tree visualization is a great tool to analyse and navigate inside a large hierarchy of data. The statistical medical data resulting from medical studies often represent a large set of data, but not necessarely a hierarchical one. So, in order to use the visualization techniques, one has first to classify the data in a hierarchical form. This classification gives opportunity to differents point of views on the data.

Once the hierarchical classification has been done, one should choose which criterias, which characteristics of the data will be represented in the visualization. In the hyperbolic tree visualization, nodes can be color-coded with respect to one criteria. In the treemap visualization, the size of the node and its color filling should be associated with characteristics of the data. Those mapping could of course be changed dynamically, thus permitting comparison between several criterias.

The treemap visualization allows for a comparison of two criterias on the statistical medical data: the relative predominance of a data from the others, given by the size of the corresponding node, and the relative predominance of a criteria from the others, given by the color-filling. Seeing that a particular data is predominant is easy : the corresponding node will be the bigger one on the treemap. The predominance of a criteria will be given by the overall color of the treemap. The repartition of the color will also be a real pertinent information.

The hyperbolic tree visualization allows for a rapid and efficient navigation inside the whole data set. With a large hierarchy, it could be difficult to estimate how deep is buried a data. The hyperbolic tree visualization could be used for efficient browsing of the whole data set. One could also use it to get an insight on how the data are organized.

In order for those visualization techniques to be used like tools by end-users, it is really important that they are readily avalaible. To be useful, it is important that they are hightly customizable. That's why we developped two free and open-source java libraries implementing those two visualization techniques. The Treemap Java Library [15] and the Hypertree Java Library [16] are two free open-source java implementations of respectively the treemap visualization and the hyperbolic tree visualization.

Building them as Java libraries allows the user to be plateform-independant. Giving them with the source code allows knowledgeable user to customize them for specific use. Letting them free assures that everyone could use them (at least outside USA for the hyperbolic tree library).

V. Conclusion

We have seen in this article two visualization techniques that could be used for statistical medical data analysis. Each one has its strengths and its weaknesses, but together they are incredibly well complementing each others. Moreover, there are two free open-source java implementations of each visualization technique, so that everyone could use them.

Acknowledgments

This work has been made possible thanks to a grant from the CNRS/Région Lorraine.

References

[1] Antanas Zilinskas, Audrone Jakaitiene, Aurelija Podlipskyte, and Giedrius Varoneckas, "On intelligent optimization in bio-medical data analysis and visualization," in *Proceedings of the 10th World Congress on Health and Medical Informatics (medinfo 01)*, London, 2001.

[2] Ben Shneiderman, *Designing the User Interface, third edition*, Addison-Wesley, 1997.

[3] Brian Johnson and Ben Shneiderman, "Tree-maps: A space-filling approach to the visualization of hierarchical information structures," in *Proceedings of the International IEEE Visualization Conference*, San Diego, 1991, pp. 284–291.

[4] Ben Shneiderman, "Tree visualization with tree-maps: A 2-D space-filling approach," *ACM Transactions on Computer-Human Interaction*, vol. 11, no. 1, pp. 92–99, 1992.

[5] Mark Bruls, Kees Huizing, and Jarke J. van Wijk, "Squarified treemaps," in *Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization*, Vienna, 2000, pp. 33–42, Springer.

[6] Ben Shneiderman, "Workshop on treemap implementations and applications," `http://www.cs.umd.edu/hcil/soh/2001/abstracts.html`. 18th HCIL Symposium and Open House, University of Maryland, College Park, Maryland, USA, May 31, 2000.

[7] John Lasseter, "Principles of traditional animation applied to 3D computer animation," *Computer Graphics*, vol. 21, no. 4, pp. 35–44, 1987.

[8] John Lamping, Ramana Rao, and Peter Pirolli, "A focus+context technique based on hyperbolic geometry for visualizing large hierarchies," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI-95)*, Denver, 1995, pp. 401–408, ACM Press.

[9] John Lamping and Ramana Rao, "A focus+context technique for visualizing large hierarchies," *Journal of Visual Languages and Computing*, vol. 7, no. 1, pp. 33–55, 1996.

[10] Vladimir Bulatov, "HyperProf applet," `http://www.physics.orst.edu/~bulatov/HyperProf/`.

[11] Peter Pirolli, Stuart K. Card, and Mija M. Van Der Wege, "Visual information foraging in a focus+context visualization," in *Proceeding of the ACM Conference on Human Factor in Computing Systems (CHI-01)*, Seattle, 2001, pp. 506–513, ACM Press.

[12] Tamara Munzner, "H3: Laying out large directed graphs in 3D hyperbolic space," in *Proceeding of the IEEE Symposium on Information Visualization (InfoViz 97)*, 1997, pp. 2–10.

[13] Köhler, Wolfgang, *Gestalt Psychology*, Liveright, 1947.

[14] Pascal Molli, Hala Skaf-Molli, and Christophe Bouthier, "State treemap: an awareness widget for multi-synchronous groupware," in *Proceedings of the 7th International Workshop on Groupware (CRIWG'2001)*, Darmstadt, Germany, 2001.

[15] Christophe Bouthier, "Treemap Java Library," `http://treemap.sf.net/`.

[16] Christophe Bouthier, "Hypertree Java Library," `http://hypertree.sf.net/`.