

EVAT : Environment for Vizualisation and Analysis of Trees

David Auber *

Maylis Delest[†]

J.Philippe Domenger[‡]

Pascal Ferraro[§]

Robert Strandh⁺

LaBRI UMR 5800 - Université Bordeaux 1

Abstract

This paper presents a piece of software for the visualization or navigation in trees. It allows some operations as comparison between trees or finding common subtrees. It means a presentation of common things with same colors. Overall implemented tools are strongly based on intrinsic combinatorial parameters with as few references as possible to syntactical data .

CR Categories: D.2.2 [Software Engineering]: Tools and Techniques --User interfaces. G.2.1 [Discrete Mathematics]: Combinatorics -- Combinatorial algorithm

Keywords: Trees, analysis, combinatorics, visualization

1 Introduction

Exploration of trees is an important domain for information visualization. EVAT is designed for exploring one tree or comparing two or more trees, keeping a view on each tree analyzed within the session. Three sets of data were proposed and for each of them, EVAT helps answering user's questions. Help means that it shows, using colors, similarity or allows filtering on value in order to extract a selection view.

In this filtering process, the focus+context technique is applied in relation with the drawing. It means that the coordinate are recomputed taking in account the selected data.

EVAT allows importation of XML files and also direct importation of a file system. The whole session can be saved in EVAT format.

It runs under Linux Redhat 9.1 using QT library. It is useful but not mandatory to have 512MB memory and a graphic card Open GL 3D with acceleration.

EVAT manages trees up to 550 000 nodes using 140MB memory. Thus it is possible to deal during the same session with several huge trees.

In the following sections, this note presents the background tools, the menus and ergonomic feel, the main tools for filtering. Some examples are shown. Then in a last paragraph some strong and weak points of the software are discussed.

2 Background tools

In this work, we have used the Tulip framework [2] that can be downloaded at www.tulip-software.org. The main relevant features in this software are:

- a powerful kernel in terms of time and memory complexity,
- extensibility by plugins without recompiling,

*e-mail: aub@labri.fr

[†]e-mail: maylis@labri.fr

[‡]e-mail: domenger@labri.fr

[§]e-mail: ferraro@labri.fr

⁺e-mail: strandh@labri.fr

address : 351, Cours de la Libération, 33405 TALENCE CEDEX, FRANCE

- possibility to map texture and colors on edges and nodes without loosing "performance",
- easy management of cluster of clusters.

Searching analogue subtree is done for large trees more than 300 nodes) by our unpublished heuristic based on Strahler numbers [1], that we call fast algorithm. For smaller trees, the Zhang algorithm [7] is used. For two subtrees, similar nodes are displayed with same color.

Managing the colors of nodes in the interface is done by mapping attribute values on RGB or HSV values or size values. Two methods are proposed

- linear mapping,
- distribution mapping [4].

The color mapping on HSV is done on the Hue component and fits the rainbow scale. That means :

- Pink is associated to the highest value,
- Yellow is associated the lowest one.

Coloring edges is made by interpolating the colors of the two extremities. It is possible to map one attribute on the size of the nodes.

3 Ergonomy and menus

A view of the software is given in figure 1. During a session, the user can open several graphs or creates subgraphs. Each one is displayed in the left window (overview window). The active graphs are those displayed in the other upper right window (visualization window). In the lower right window (task window), four main tasks are reachable.

- Visualization contains all the functionalities for setting the display : drawing the tree [3,5], color, size and shape of the nodes.
- Search allows to select nodes according the value of the node attributes. In the advanced mode multiple selection using boolean operators can be done.
- Detailed Data allows the user to inspect all the attribute values of a node by clicking it or all the values of a given attribute in a tree.
- Comparison can be used if two visualization windows exist. It proposes the two algorithms quoted in the previous paragraph.

When several active windows are set up, EVAT automatically provides the tool associated with each one (see figure 2). Moreover, each task which is done on one window is done on the other.

4 Examples

Here, we shortly describe some tasks that can be issued using EVAT.

For system file data, it is possible to analyze an attribute for example the size associated to a node. The figure 1 shows the result after the following operation :

- map the attribute on the node size of the drawing using a linear mapping,
- map the attribute on the node color using HSV and uniform algorithm,

- select the jpg file,
- zoom in the tree.

For two trees, it is possible to identify possible common subtrees. On the figure 3, left and right subtrees with same color are probably similar. This is one using the fast algorithm. it also applies on one tree. On figure 4, the result of the algorithm applied on the logs_A file contest is displayed. The exploration with EVAT shows that some subtrees (blue or violet one) were duplication of directories.

5 Weak and Strong points

The strongest point of EVAT are in the fact that it can answer to most of the questions on trees (file system, phylogenic, classification, ...) in real time. All algorithms (except oneare less than $n \log(n)$ in complexity for memory and time. The algorithm that maps trees node by node has a complexity n^3 but the software EVAT does not run the process if the number of nodes is greater than 300. Nevertheless, the whole software is based on combinatorial properties and thus does not help in finding relation on the lexical meaning of the nodes. EVAT does not manage the history but successive views can be kept and saved in Tulip format in order to work later on the data.

References

- [1] D. AUBER, Using Strahler numbers for real time visual exploration of huge graphs, In *Proceedings of International Conference on Computer Vision and Graphics 2002*, Zakopane, 56-69.
- [2] D.AUBER, 2002., *Outils de visualisation de grandes structures de données*, PhD thesis, LaBRI, Université Bordeaux I.
- [3] S. GRIVET, D.AUBER , J.P. DOMENGER, G. MELANCON, Bubble Tree Drawing Algorithm, Preprint LaBRI, 2003.
- [4] I. HERMAN, M. MARSHALL, G. MÉLANÇON, Density Functions for Visual Attributes and Effective Partitioning in Graph Visualization, In *Proceedings of IEEE Symposium on Information Visualization 2000*, IEEE Computer Society, 49-56.
- [5] E.M. REINGOLD, J.S. TILFORD, Tidier Drawings of Trees, *IEEE Transactions on Software Engineering*, 1981 7(2), 223-228.
- [6] A. N. STRAHLER, Hypsomic analysis of erosional topography, *Bulletin Geological Society of America* 1952, 63, 1117-1142.
- [7] K.ZHANG, Aconstrained edit distance between unordered labeled trees, *Algorithmica*, 1996, 15, 205-222.

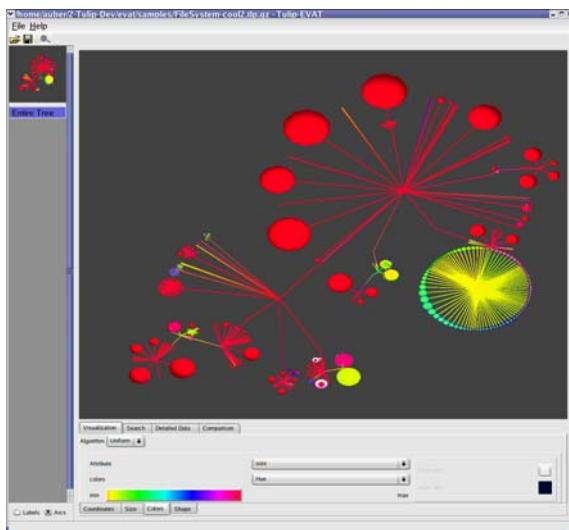


Figure 1. A view of EVAT

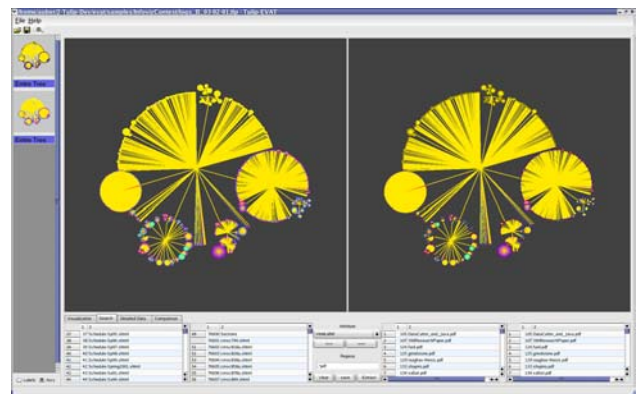


Figure 2. A view of multiple visualization windows.

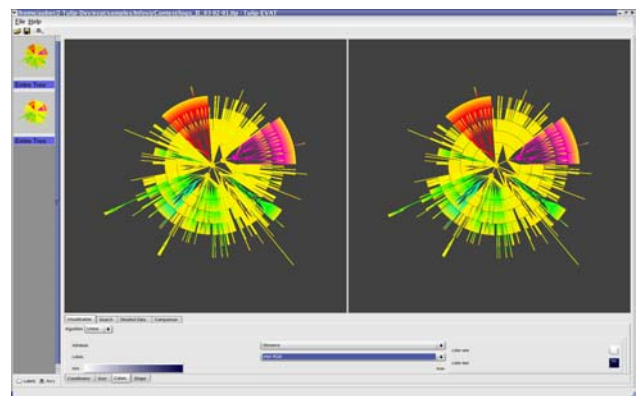


Figure 3. Common subtrees between logs_A and logs_B files.

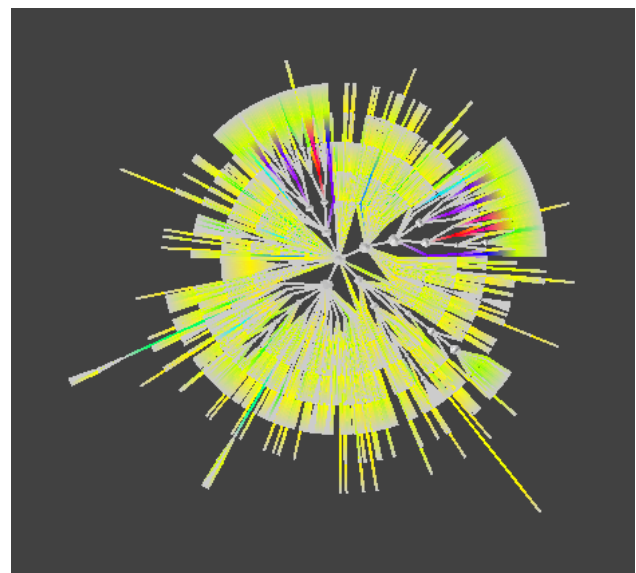


Figure 4. Similar subtrees in the logs_A file.