Flexible Trees: Sketching Tree Layouts
Javad Sadeghi, Charles Perin, Tamara Flemisch, Mark Hancock, Sheelagh Carpendale

To cite this version:
Javad Sadeghi, Charles Perin, Tamara Flemisch, Mark Hancock, Sheelagh Carpendale. Flexible Trees: Sketching Tree Layouts. Proceedings of the International Working Conference on Advanced Visual Interfaces, Jun 2016, Bari, Italy. pp.84-87, 10.1145/2909132.2909274. hal-01618626

HAL Id: hal-01618626
https://hal.archives-ouvertes.fr/hal-01618626
Submitted on 18 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Flexible Trees: Sketching Tree Layouts

Javad Sadeghi
University of Calgary
jsadeghi@ucalgary.ca

Charles Perin
University of Calgary
charles.perin@ucalgary.ca

Tamara Flemisch
Technische Universität
Dresden
tamara.flemisch@gmail.com

Mark Hancock
University of Waterloo
mark.hancock@uwaterloo.ca

Sheelagh Carpendale
University of Calgary
sheelagh@ucalgary.ca

ABSTRACT
We introduce Flexible Trees, a sketch-based layout adjustment technique. Although numerous tree layout algorithms exist, these algorithms are usually bound to fit within standard shapes such as rectangles, circles and triangles. In order to provide the possibility of interactively customizing a tree layout, we offer a free-form sketch-based interaction through which one can re-define the boundary constraints for the tree layouts by combining ray-line intersection and line segment intersection. Flexible Trees offer topology preserving adjustments; can be used with a variety of tree layouts; and offer a simple way of authoring tree layouts for infographic purposes.

CCS Concepts
•Human-centered computing → Information visualization; Gestural input; Dendrograms; Cladograms;

Keywords
Visualization; trees; sketching; interaction; authoring; infographics.

1. INTRODUCTION
We present Flexible Trees, a sketch-based technique for deforming tree layouts. Flexible Trees offer a new approach to creating emphasis and focus+context, and a simple way of creating aesthetically pleasing customized hierarchical data visualizations.

Hierarchical data are ubiquitous (e.g., family trees and file system directory trees), and hierarchical data visualizations have a long history. Tree layouts are certainly the most standard family of hierarchical data representations. They usually represent data entities as the nodes of a tree, and the relationships between these data entities as links, adjacency, or nesting [14] (see [13] for an overview). While these tree layouts provide a variety of representations, they are generally designed to fit into formal shapes such as rectangles, circles, and triangles; and they emphasize the root of the tree.

Constraining tree visualizations to pre-defined shapes makes these layouts ill-suited to narrative visualizations [15] and storytelling [9] – where flexibility, authoring capabilities, and aesthetics are important.

Figure 1: Tree layouts created using Flexible Trees, with the shapes of a city skyline, an eye, and Twitter's logo.

For example, one may want to illustrate a blog article with an aesthetically pleasing tree visualization that visually relates to the article; or may want to emphasize particular parts of the data, e.g., a sub-tree or a node. To empower non-experts in visualization, we designed a sketch-based interaction to deform tree layouts (see Figure 1) so that one can freeform layouts sketch directly instead of using buttons, menus and dragging operations.

We first discuss the literature related to Flexible Trees. Then we describe the Flexible Trees algorithm. Finally, we discuss the possibilities for authoring tree visualizations for infographic purposes.

2. RELATED WORK
Our work with Flexible Trees relates to research in the areas of tree layouts, sketch based interaction, and data story telling.

2.1 General background about trees
Tree layouts for visualizing hierarchical data can be implicit or explicit [14]. A treemap [7], where each child node is nested inside its parent, is an implicit layout which specifies each node size according to the underlying data, and the size of a node depends on the size of its children. Implicit layouts are not amenable to freeform deformation as it would interfere with the data representation.

In contrast, explicit layouts such as Reingold and Tilford’s trees [11] encode data relationship in the tree structure, which can be maintained when distorting the tree. However, such tree layouts are automatically computed and are not interactively adjustable. Flexible Trees offer layout adjustment and personalization through sketching.

2.2 Sketch-based Interaction
Hand-drawn sketching is known to be effective in promoting innovation, creativity, and thinking [3, 6]. Studies have investigated how people manipulate representations, e.g., observing how people spatially arrange nodes and links [18]. These studies showed that pre-defined layouts are not in accordance with layouts people draw.

Researchers have considered sketch-based interaction [17] for visualization [9, 20] as pre-defined visualizations can limit people in expressing their thinking about data [19]. Also, NapkinVis [4] and SketchVis [2] investigate sketching simple visualizations. However, we found no work about sketching tree layouts.
2.2 LayoutDeformation:RadialCladogram

We first illustrate FlexibleTrees with the RadialCladogram (see Figure2). This explicit tree layout arranges the nodes of a tree in acircle centered around the root node: 1) the leaf nodes are evenly spacedon the circumference of the circle; 2) the internal area of the cire is sliced into d concentric rings; 3) each parent node ispositioned on the inner ring corresponding to its depth suchthat it bisects the angular distance between its children; 4) finally, theedge originating at a child is drawn towards the root; when thisedge intersects the concentric ring of its parent it forms a joint andcompletes the edge line of the ring. Figure 2(a) shows a RadialCladogram and a sketched stroke S that sets the layout boundary.

We first find the closest intersection point between the raysfrom the tree’s center to the borders of the window. Rays are shotat uniform angular intervals, with one ray being shot for each leafnode. A ray R(t) = p + td is defined by a point p = (p_x, p_y) anda unit direction d = (cosα, sinα); t ≥ 0 is a scalar indicatingtime. Let w and h be the width and height of the window. For eachray, we compute the intersection times with each border, with t_{left} = \frac{p_y}{\cosα}, t_{right} = \frac{w-p_x}{\cosα}, t_{bottom} = \frac{p_y}{\sinα}, and t_{top} = \frac{h-p_y}{\sinα}. Given t^* = \min(t_{left}, t_{right}, t_{bottom}, t_{top}), we find the closestintersection point p^* = (p_x + t^* cosα, p_y + t^* sinα) for each ray.

Next, we find all intersection points between the line segmentsrunning from root to p^*’s and the line segments of S as follows.Let p_1^* = (a_1,b_1) and p_2^* = (a_2,b_2) be the two extremities of theline segment running from the root to p^*, and p_1 = (x_1,y_1), p_2 = (x_2,y_2) a line segment of S. Then we use the slopes m = \frac{y_2-y_1}{x_2-x_1}, and m = \frac{y-y_1}{x-x_1} to find the coordinates of the intersection point (x,y):

\begin{align*}
x &= \frac{m(x_2-a_2)+a_1}{m-a} \\
y &= \frac{m(x-y)+y_1}{m-a}
\end{align*}

If S intersects multiple times with the ray, we choose the intersectionclosest to the root. The last step is to adjust the radius of theRadialCladogram if a ray intersects the stroke. We calculate thedistance between the intersection point and the root node and use it as theleaf node’s new radius. Finally, we update the position of all ancestorsof this leaf node (except the root) using the average radius of theleaf nodes in the corresponding subtrees, as shown in Figure 2(b).

3.3 Other LayoutDeformations

We have detailed the FlexibleTrees algorithm for the RadialCladogram (a node-link radial layout). We now describe how toadapt the algorithm to a rectilinear node-link layout (Cladogram), aradlinear space-filling layout (Icicle Plot) and a radial space fillinglayout (Sunburst). Figure 3 shows these layouts being deformed.

The Cladogram, or dendogram, is the rectilinear version ofthe RadialCladogram. All leaves are drawn uniformly on the bottomof the tree, regardless of their distance from the root, or depth, d. Thenthe space between root and leaves is equally sliced according to d atthat point. Parents are aligned to be at the center of their children, andthe edge lines are drawn from parents to their children with a 90°angle turn at the same horizontal position of the children.

Implementation: As this layout only allows changes to the areabelow the root node, we need only find the intersections between S andthe vertical line segments V from the root to the bottom border. Onevertical line v ∈ V is shot for each leaf vertex. We find theclosest intersection point p^* for v and update the layout to be boundat p^* similarly to updating the radius of the RadialCladogram tree.

The Icicle Plot [8] is a rectilinear space-filling layout. It onlyrepresents nodes – as filled rectangles; edges are implied by adjacency.All nodes have the same height and are drawn at their exact depth.Parent nodes are sized according to their number of children.
4. DISCUSSION AND FUTURE WORK

We discuss applications of sketch-based layout and show the applicability of Flexible Trees to the authoring of infographics.

4.1 Layout Deformation via Sketching

Our sketch interpretation method can interpret freeform shapes. However, our layout deformation algorithm uses the closest stroke intersection after ray casting. We plan to investigate more complex shapes such as concave shapes by adapting the curvature of links and moving nodes, e.g., using physics.

Our algorithm produces fine results for node-link layouts. However, deformed space-filling layouts are more difficult to understand. We plan to explore new strategies for such layouts in future work.

Finally, we implemented the algorithm for four implicit layouts but adapting adjacency and nested layouts is a challenge. For example, drawing an arbitrary shape and fitting a treemap into it raises problems regarding maintaining the relative sizes of the nodes.

4.2 Flexible Trees for Focus+Context

Flexible Trees can create focus+context views. Using a sketched curve to expand part of the layout is similar to what can be achieved with a fisheye magnifying glass, but it acts differently. Figure 4 shows a sketched focus+context view of a Radial Cladogram: the leaf nodes of the condensed tree (a) are difficult to distinguish visually; but a simple hand-sketched stroke can open up the subtrees sitting next to that curve (b). It makes the nodes readable locally.

4.3 Flexible Trees for Infographics

Flexible Trees are extremely suitable to generating infographics. By drawing different shapes one can add more information about the tree itself (e.g., provide the context of the data) and about its data (e.g., provide information about the tree nodes and links). In this section, we provide examples of infographics that we created by modifying the original outputs of the algorithm in an SVG editor.

Figure 5 shows a tree in the shape of a cup that represents a subset of the SCAA Flavor Wheel dataset. The shape does not add data but it contextualizes the tree by conveying the topic. The coffee cup shape hints at a beverage and the brown colour is typical for coffee.

Figure 7 shows data from WorldLifeExpectancy. This pie chart shows the top causes of death for the United States, Germany and...
India. The shape adds quantitative data: slice angles represent birth rates for each country. In addition to conveying the information contained in the trees, this infographics puts this information in perspective by also comparing the areas of the trees in the pie chart.

Figure 6 shows data from WorldLifeExpectancy. Color distribution and slice angles represent the impact of each cause of death on total percentage of deaths. The shape adds information to the tree: it conveys the country that the data is related to. Filling parts of the countries also conveys quantities like a pie chart does.

Flexible Trees benefit infographics as they make it possible to include context (e.g., coffee cup), qualitative data (e.g., country shapes), and quantitative data (e.g., birth rates in a pie chart layout).

5. CONCLUSIONS

Flexible Trees is a topology-preserving distortion technique for adapting the layout of trees according to hand-sketched strokes. This initial exploration of sketch-based layout paves the way for further research. That includes exploring hand-sketching algorithms for implicit and nested tree layouts; and beyond trees, applying the method to e.g., graph drawing. Generalizing this approach has the potential to reach a large number of people, as it makes it possible to easily and rapidly create layouts of customized shapes. Our discussion of the applicability to infographics generalizes to other visualization types such as graph visualizations, and contributes to filling the gap between Infovis research and infographic design.

6. ACKNOWLEDGMENTS

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), SMART Technologies, and Alberta Innovates Technology Futures (AITF). We also thank Lindsay MacDonald for proofreading the paper.

7. REFERENCES


Figure 7: Infographics using small multiples of deformed trees in the shape of Germany, the United States, and India. The trees and the coloured areas represent the most common causes of death and their percentage.