# ShamsehTrees: Providing Hierarchical Context for Nodes of Interest

Katayoon Etemad and Sheelagh Carpendale
Department of Computer Science
University of Calgary

## Abstract

Visualizations of hierarchical data usually focus on conveying structure. However, with really large hierarchies, layouts tend to become overcrowded, making it difficult to see details about specific nodes. In contrast, ShamsehTrees focus on layouts centered on a node of interest, provide interactive nested layouts that were inspired by artistic and natural floral patterns, and make use of the natural symmetries in phyllotactic patterns. Instead of emphasizing overall tree structure, these layouts are created to make the most space available for the node of interest. The basic layout is comprised of nested circles that are centered on the node of interest. After selecting a new node of interest, the resizing and repositioning of nodes is animated as they transition to the new layout.

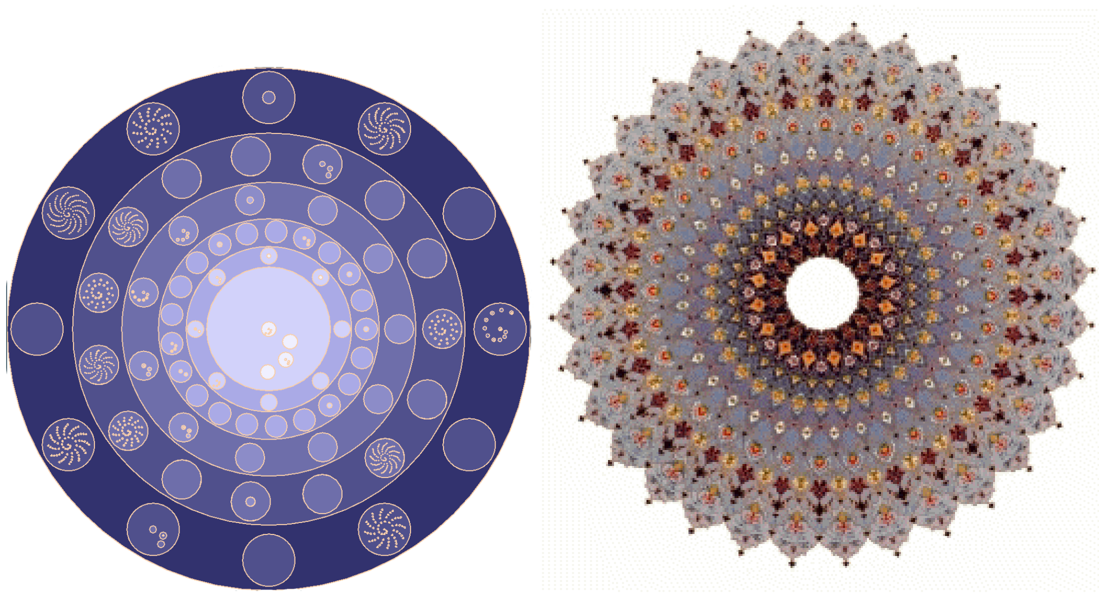

**Figure 1**: *Left: ShamsehTree, symmetric representation of the hierarchical context of the node of interest. Right: Traditional design of Shamseh, a circular shaped Persian Floral Pattern.*

## 1 Introduction

Many types of data in science, engineering and social sciences can be characterized as hierarchical. This hierarchical or tree structured data can be represented in a variety of layouts, which are usually designed to emphasize the structure and to show the relationships between parent, child, and sibling nodes. Hierarchically structured data occur sufficiently frequently that tree visualization continues to be an active research topic. However, as the data being represented becomes huge the resulting tree layouts can easily become too dense to be readable. While many tree visualizations exist (for surveys see [1, 5]) increasingly massive data sets, expanding computational power and still relatively limited display space makes this a topic of ongoing interest. The problem of visualizing tree layouts can be more challenging when it comes to representing huge trees in a clear and aesthetically pleasing layout.

In extremely large trees, representing overall structure can make access to and discovery of individual nodes difficult due to limitation of available space. To address this, the ShamsehTree layouts presented in this paper emphasize a node of interest (NOI) by making the most space available for the chosen node and by displaying the hierarchical structure of the tree around the NOI.

Our node focused layouts are based on inspiration from Persian floral patterns and spiral arrangement of phyllotactic patterns. In Persian floral patterns plant elements have been symmetrically arranged to form a beautiful and complex concentric circular pattern (Figure 1) [3]. We also make use of the natural symmetries and spiral arrangements of phyllotactic patterns [6, 11] to develop concentric, node-focused layouts (Figure 1).

Symmetry can be an important factor for recognizing elements within an image [13]. Symmetrically arranged elements can create a strong holistic impression and may be a useful feature for grouping individual elements with the same properties. In general, symmetry is a pervasive phenomenon in both natural and constructed environments. It has long been recognized as playing an integral role in geometry and architecture [13]. Symmetry is an important factor for harmony, balance, and proportion used in nature, art, mathematics and computational sciences [7]. A symmetric representation of information is usually more aesthetically appealing, and aesthetically appealing objects have been shown to better engage and motivate viewers [7]. Symmetry can be simple when the elements of an image are repeated by translation, rotation and reflection (translational symmetry) or can be more complex and dynamic when scaled repetition is used (dilational symmetry) [3]. In dilational symmetries, suitable scale factors and appropriate arrangements of the copies (e.g. spirals) can create an aesthetically appealing and proportionally pleasing final composition. Some use of symmetry has been explored in human perception [10] and graph layout [2, 8]. However, the full potential and power of symmetry, most notably dilational, in terms of layout and interactive exploration methods, has not yet been fully explored. The potential for a positive role of dilational symmetry in information visualization is partially evidenced by PhylloTrees [9] and Botanical trees [8].

Since ShamsehTrees are intended for large real world trees, the images used for illustration in this paper are ShamsehTree views of WordNet [4]. WordNet is a human-constructed lexical database in frequent use by many Natural Language Processing (NLP) techniques. WordNet has a hierarchical structure that has been created based on the IS-A relationship. For example, a cat IS-A mammal and a mammal IS-A animal. The WordNet is a collection of English noun synsets (groups of synonyms) rooted at the word *entity* using the IS-A relation (called hyponomy). The hierarchy consists of 73,736 synsets with a maximum depth of 14. The data identifies a synset by the first word in the set, creating a hierarchy of words related by IS-A. Figure 2 shows a ShamsehTree view with the WordNet with labels in place. For the other figures in the paper, we have not displayed the labels as the discussion is about the structure of visualization of ShamsehTrees not about WordNet.

## 2 Representation

Our nested ShamsehTrees use a circle as the basic shape for their nodes. Although, our method can work for any convex shape, the circle has a simple but rich geometry and is frequently found in nature and art. Also, it has been suggested that circles make it easier to see groupings and structural relationships [12]. We use a nested layout to visualize parent/child relationships without direct use of lines or curves. Wang et al. [12] also describe a method for a nested tree layout where tree nodes at different levels are displayed by using 2D nested circles with variable sizes. Unlike the available nested layouts such as those presented by Wang et al. [12], in our representation we use nodes with the same size in each level which creates more symmetrical patterns (Figure 1). The inspiration for the underlying layouts comes from the symmetric structure of Persian Floral Patterns [3] especially the circular shaped known as *Shamseh* which means sun in Arabic language

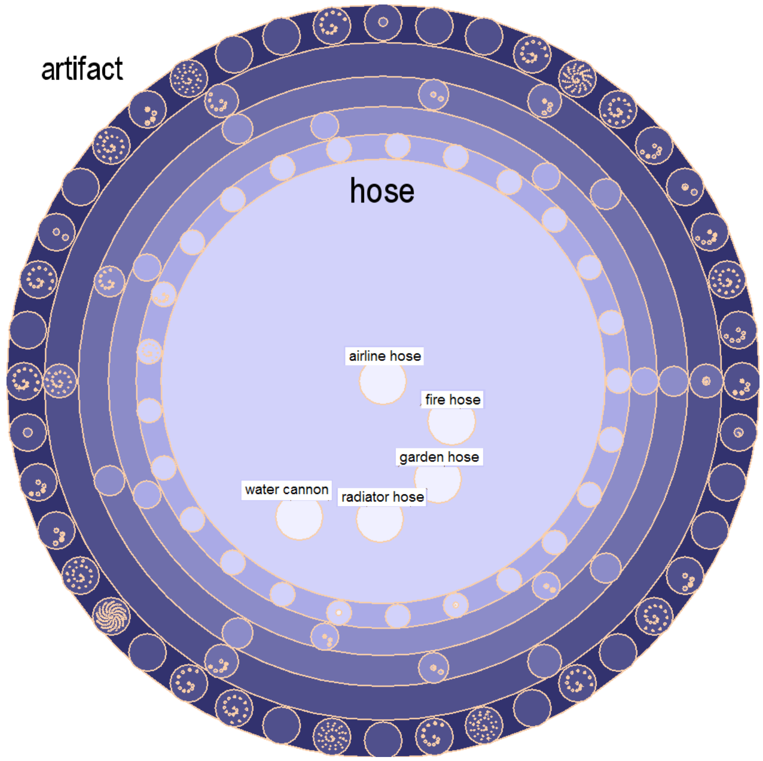


**Figure 2**: *The ShamsehTree view of WordNet with the NOI set at hose. The five children of the NOI are all a type of hose and a hose is a type of artifact.*

(Figure 1) and provides the name for our layout.

In our representation, if node $A$ is inside node $B$ then $A$ is a child of $B$. ShamsehTree layout shows a subset of the tree. Figure 3 illustrates this in comparison with a more traditional, node-link, tree layout. The red circle in the node-link tree layout on the left shows the NOI (node of interest). This same node is also outlined in red in the ShamsehTree (right). In the node-link tree, children of the NOI are shown in blue lines. In the ShamsehTree the children of the NOI are nested inside the NOI, and contain the NOI's children's children. The ancestors of the NOI are shown in green lines in the node-link tree, and are shown as progressively larger circles encasing the NOI in the ShamsehTree. In addition, two levels of children are shown for each ancestor of the NOI. In ShamsehTrees the NOI is central and nested inside all of its parents and contains two levels of its children. ShamsehTrees only display two levels of children in the NOI, however, if further levels of children exist, they can be reached interactively. Our approach is an abstraction that provides both ancestor and descendant context for an NOI. This dual context abstraction enables quick exploration around NOI and ready access to the rest of the tree via the root's children while keeping the entire tree in a traceable and interactive size.

Nested layouts may address edge congestion, however, new challenges arise. For instance, one issue is how to arrange sub-nodes. If $A_1, A_2, \ldots, A_n$ are all children of $B$, then all these nodes must be inside of $B$ but their arrangement needs to be organized, if possible, to enhance the clarity of visualization. Our layout is based on a node of interest (NOI), which is drawn as a circle and is nested within concentric circles that represent its ancestors. The outer most circle is the *root* of the tree. In the initial layout the node of interest would be the *root*. If the NOI has children, they are laid out inside the NOI using phyllotactic patterns. If the NOI has siblings, they are evenly spaced in the NOI's parent's ring. This is repeated until the outer ring is the tree's root. The NOI and any drawn ancestor node will display its children, and its children's children depending on how much space it has. For the NOI this creates both an ancestrally-

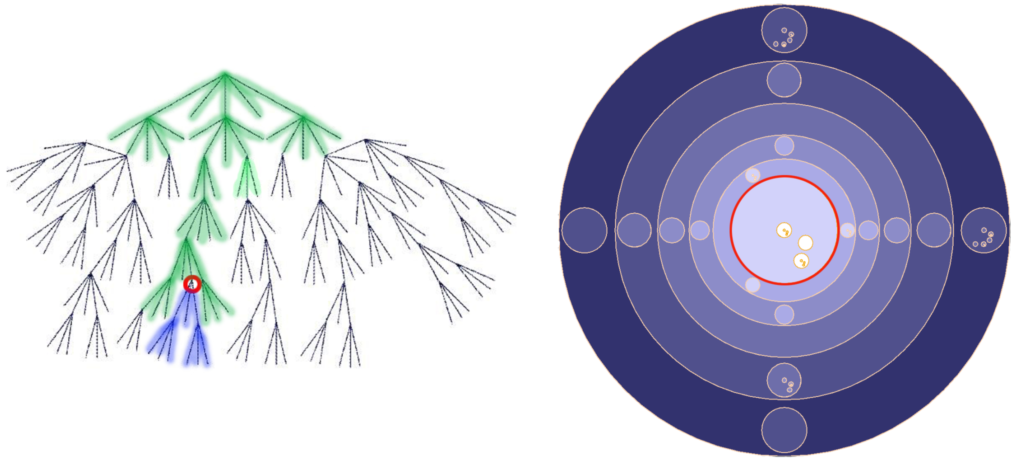

**Figure 3**: *Comparing the circular node focused tree layout with a traditional tree layout.*

based and a descendant-based context. For the layout of a node's children, we use spiral patterns based on basic phyllotactic arrangements. Suppose that $(\phi_n, r_n)$ is the polar coordinate of the center of child $n$ for $n = 1, 2, \ldots, n_{max}$ where $n_{max}$ is the number of children. We distribute the centers on spirals from inside out. Therefore, both $\phi$ and $r$ are increasing functions in term of $n$. In our implementation, we use:

$$\phi_n = n\,\alpha \quad , \quad r_n = c(n_{max})\,\sqrt{n}$$
$$n = 1, 2, \ldots, n_{max}$$

where the constant $\alpha$ controls the angular increment between successive nodes and $c(n)$ is used to control the radius of the spiral. Figure 4 shows how changing $\alpha$ affects the layout.

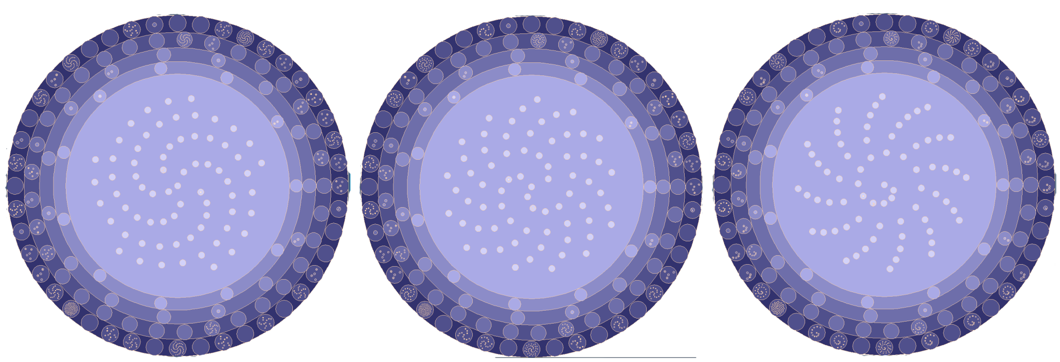

**Figure 4**: *Change of $\alpha$ creates different patterns. From left to right:* $\alpha = 10, \alpha = 130, \alpha = 100$.

The function $c(n_{max})$ controls the increment factor for the radius. To keep the symmetry of layout, maintaining all siblings as the same size, we reduce the effect of $c(n_{max})$ when there are more children. In our implementation we have used $c(n) = \frac{1}{\sqrt{n_{max}}}$ to normalize $r_n$ in the range between 0 and 1. Therefore, changing $n_{max}$ results in different layouts as demonstrated in Figure 5.

When a node $A$ is selected that node becomes the NOI and moves dynamically to the center of the layout and its size is appropriately increased. Following this action, the NOI's children are recursively resized and re-positioned. Since the NOI's siblings are to be arranged within the NOIs parent node $B$, the size of the NOI must be slightly smaller than its parent to leave enough space for the siblings. We uniformly space the

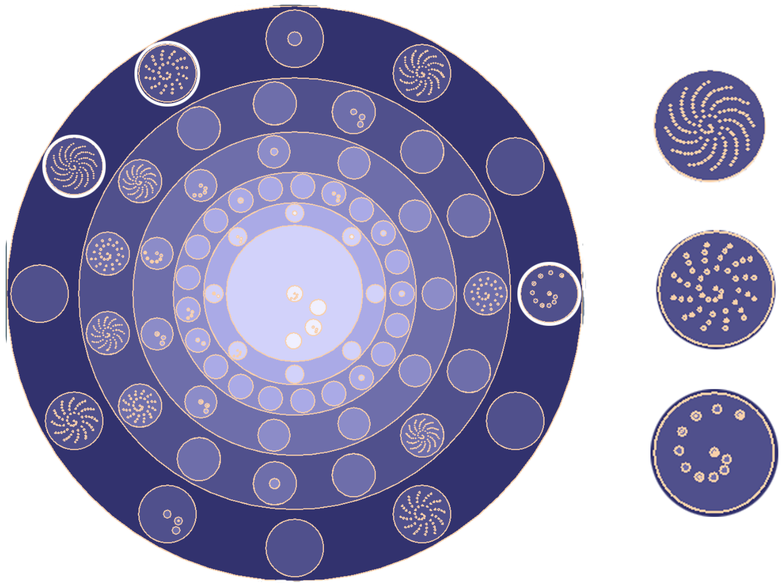

**Figure 5**: *The effect of n in the layout. The three nodes shown on the right have from top to bottom 156, 50, and 11 children, all of which have been laid out in the same size parent. For all three nodes $\alpha$ is the same. Note how in the node with 11 children only the inner part of the spiral is used.*

siblings of $A$ in the space around $A$ and inside $B$ as demonstrated in Figure 5 (left side). The NOI and its siblings are displayed with the same color. To describe these operations more precisely, let $(x_A, y_A)$ be the center of $A$ (NOI) and $r_A$ be its radius. The new position of the NOI can be obtained by:

$$(x_A, y_A) = (x_B, y_B).$$

And the size of the NOI is increased by:

$$r_A = d\, r_B, \;\; 0 \leq d \leq 1$$

where $d$ is a constant that controls how much smaller the NOI is than its parent. As illustrated in Figure 6, larger values for $d$ (left) provide more space for the NOI and smaller values for $d$ (right) provide more space for ancestors and their children. However, for really deep trees large values of $d$ do not provide enough space for the NOI. In addition, extremely small or large values for $d$ do not produce proportional and aesthetically appealing layouts. Setting $d$ to the values near to the golden ratio (the right layout in Figure 6) seems to make better layouts.

During the transition from one NOI to another, the NOI's siblings and their children are dynamically repositioned and resized as shown in the Figure 7. Assume $S_1, S_2, \ldots, S_k$ are siblings of NOI. As a simple solution, we distribute these nodes in a circle uniformly around the NOI and within their parent. More precisely, we use the new NOI's parent's center $(x_B, y_B)$ as the new NOI center and to obtain the position of the sibling's centers we calculate a sibling-base circle upon which all siblings will lie. The radius of the sibling-base circle is

$$\rho = \frac{1}{2}(r_A + r_B)$$

where $r_A$ is the NOI radius and $r_B$) is the NOI's parent's radius. For each sibling, we use a circle whose center is on the sibling-base circle and whose radius will keep the sibling inside the parent and outside the

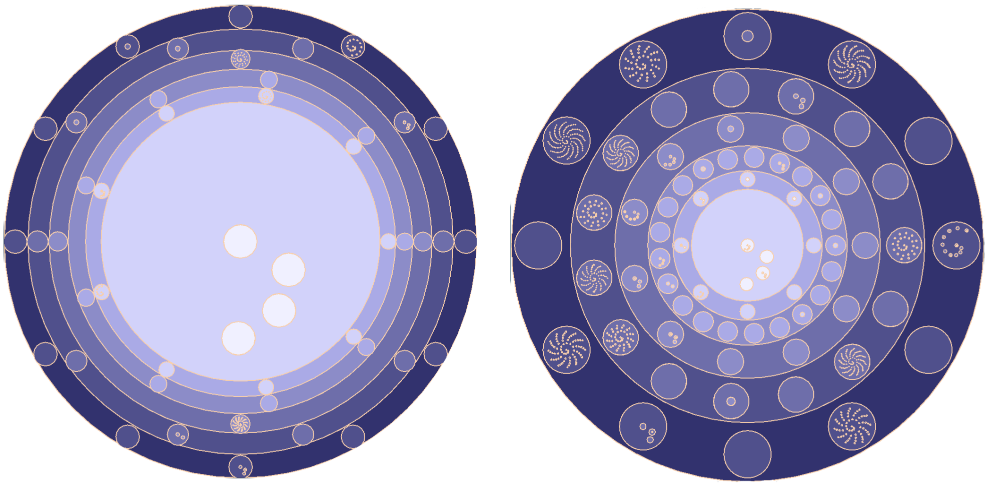

**Figure 6**: *Left: larger values for d gives more room for an NOI deeply nested in the tree; Right: smaller amounts for d provides more space for the ancestors and their children.*

NOI. To do this, we set the sibling-radius as

$$r_{s_k} = w(r_B - r_A)$$

where $w$ is a contraction factor sufficient to keep the edges of the sibling's circle just clear of both the parent and NOI circles (see Figure 7).

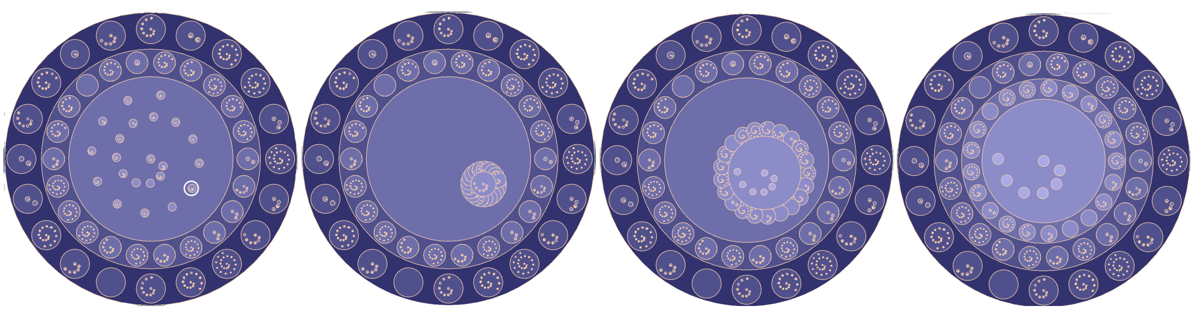

**Figure 7**: *Left: A node is selected node to be the next NOI; Left, middle: the new NOI's children encircle the new NOI; right, middle: the NOI starts to expand; Right: the new NOI and its siblings in their new positions. To make a smooth transition for animation position and radius are functions of time*

When an NOI is selected, the sibling nodes of the selected node are clustered around the new NOI. These new positions are in a flower-petal arrangement around the NOI and as the NOI expands, the sibling nodes spread apart similar to flower petals Figure 7. Animation plays an integral role in our interactive system. All of the repositioning and resizing is done dynamically as demonstrated in Figure 7. Starting from the left, the first image shows that the fourteenth of twenty-two siblings has been selected. The second and third images show the transition in progress. The newly selected NOI is expanding, moving towards the center, and its 21 siblings have arranged themselves around its edge. In the fourth (right-hand) image the new NOI is in place, centered in its ancestors and ringed by its siblings. To effect this transition, we change the center position $(x_o, y_o)$ and the radius $r_o$ of node $S$ to a new position $(x_f, y_f)$ and final size $r_f$ in a continuous manner. To make a smooth transition for animation, we make position and radius functions of time $t$

$$(x_t, y_t) = (1-t)\,(x_o, y_o) + t\,(x_f, y_f)$$
$$r_t = (1-t)\,r_o + t\,r_f$$
$$0 \leq t \leq 1$$

Color shades are used to show the hierarchy. The selected node can be seen as a circle with its children inside it and all of its parents with their hierarchy relationships are shown as circular layers with different shades of blue. Each circle in the hierarchy includes its children which are evenly and symmetrically positioned around the circle. The presentation shows the NOI with its children and its ancestors and their children through to the root of the tree.

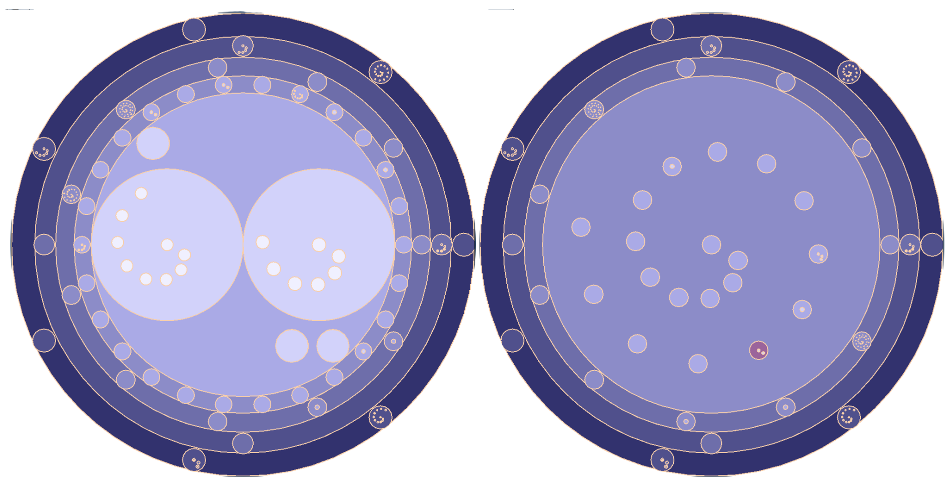

**Figure 8**: *Left: Comparing nodes. Right: Searching for a node with its label.*

## 3 Interactive Exploration

It is also possible to compare two different sibling nodes in our interactive system. The two nodes that are being compared are drawn the same size and are positioned inside their parent's node. All of their siblings are positioned around them inside their parent's node (Figure 8). While nodes can be dynamically selected node search is also supported. A search for a specific node can be conducted by entering the label of the node. The node related to the label will be highlighted using a contrasting color (Figure 8). The relation between the selected node and the root still stays visible. The tree can also be edited by adding and removing the nodes. During exploration, the tree can be modified by interactively adding and removing nodes in any desired level.

## 4 Conclusion and Future Work

In this paper, we introduced an interactive tree layout that makes some use of dilational symmetry and was inspired by natural and artistic floral patterns. This tree layout provides a node-focused approach that centers the node of interest and sets it in its symmetrically drawn context. Nodes of interest can be interactively changed and the transitions are animated. In a given layout the subset of the tree that contains the node of interest's ancestors and all of its immediate children. The whole tree can be explored interactively. Further investigations into the type of interactivity provided and the effectiveness of this type of contextual layout are underway.

**Acknowledgments:** Thanks to iLab members' useful advice and discussions. Further thanks to Dr. Samavati for his insightful input. The funding for this research is from NSERC, iCORE, and SMART Technologies.

## References


[1] G.D. Battista, P. Eades, R. Tamassia, and I.G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.

[2] P. Eades and X. Lin. Spring algorithms and symmetry. *Theoretical Computer Science*, 240(2):379–405, 2000.

[3] K. Etemad, F.F. Samavati, and P. Prusinkiewicz. Animating persian floral patterns. In *CAe'08: Proceedings of Computational Aesthetics in Graphics, Visualization, and Imaging*, pages 25–32, 2008.

[4] C. Fellbaum. *(Ed.): WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA, 1998.

[5] I. Herman, G. Melancon, and M.S. Malshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, 2000.

[6] R.V. Jean. *Phyllotaxis: A Systematic Study in Plant Morphogenesis*. Cambridge University, Cambridge, UK, 1994.

[7] Z. Kang. From abstract painting to information visualization. *Visualization Viewpoints: IEEE Computer Graphics and Applications*, 27(3):12–16, 2007.

[8] E. Kleigerg, H. Van De Wetering, and J. Van Wijk. Botanical visualization of huge hierarchies. In *INFOVIS'01: Proceedings of the IEEE Symposium on Information Visualization*, pages 87–94, Washington, DC, USA, 2001. IEEE Computer Society.

[9] P. Neumann, S. Carpendale, and A. Agarawala. Phyllotrees: Phyllotactic patterns for tree layout. In *EUROVIS'06: Proceedings of Eurographics / IEEE VGTC Symposium on Visualization*, pages 59–66. Eurographics, 2006.

[10] J. Thorpe and P.C. Oorschot. Graphical dictionaries and the memorable space of graphical passwords. In *Proceedings of the 13th USENIX Security Symposium*, pages 10–17, Berkeley, CA, USA, 2004. USENIX Association.

[11] H. Vogel. A better way to construct the sunflower head. *Mathematical Biosciences*, 44(3-4):179–189, 1979.

[12] W. Wang, H. Wang, and G. Dai. Visualization of large hierarchical data by circle packing. In *CHI'06: Proceedings of the Conference on Human Factors in Computing Systems*, pages 517–520, New York, NY, USA, 2006. ACM.

[13] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.