

**PEDVIS: A STRUCTURED, SPACE-EFFICIENT  
TECHNIQUE FOR PEDIGREE VISUALIZATION**

by

Claurissa Tuttle

A thesis submitted to the faculty of  
The University of Utah  
in partial fulfillment of the requirements for the degree of

Master of Science

in

Graphics and Visualization

School of Computing

The University of Utah

August 2011

Copyright © Claurissa Tuttle 2011

All Rights Reserved

**The University of Utah Graduate School**

**STATEMENT OF THESIS APPROVAL**

The thesis of \_\_\_\_\_ **Claurissa Tuttle** \_\_\_\_\_

has been approved by the following supervisory committee members:

\_\_\_\_\_ **Cláudio T. Silva** \_\_\_\_\_, Chair \_\_\_\_\_ **02/07/11**  
Date Approved

\_\_\_\_\_ **Juliana Freire** \_\_\_\_\_, Member \_\_\_\_\_ **02/07/11**  
Date Approved

\_\_\_\_\_ **Peter Shirley** \_\_\_\_\_, Member \_\_\_\_\_ **02/07/11**  
Date Approved

and by \_\_\_\_\_ **Al Davis** \_\_\_\_\_, Chair of  
the Department of \_\_\_\_\_ **School of Computing** \_\_\_\_\_

and by Charles A. Wight, Dean of The Graduate School.

## **ABSTRACT**

Public genealogical databases are becoming increasingly populated with historical data and records of the current population's ancestors. As this increasing amount of available information is used to link individuals to their ancestors, the resulting trees become deeper and denser, which justifies the need for using organized, space-efficient layouts to display the data. Existing layouts are often only able to show a small subset of the data at a time. As a result, it is easy to become lost when navigating through the data or to lose sight of the overall tree structure. On the contrary, leaving space for unknown ancestors allows one to better understand the tree's structure, but leaving this space becomes expensive and allows fewer generations to be displayed at a time. In this work, we propose that the H-tree based layout be used in genealogical software to display ancestral trees. We will show that this layout presents an increase in the number of displayable generations, provides a nicely arranged, symmetrical, intuitive and organized fractal structure, increases the user's ability to understand and navigate through the data, and accounts for the visualization requirements necessary for displaying such trees. Finally, user-study results indicate potential for user acceptance of the new layout.

## CONTENTS

<b>ABSTRACT .....</b>	<b>iii</b>
<b>LIST OF FIGURES .....</b>	<b>vi</b>
<b>LIST OF TABLES .....</b>	<b>viii</b>
<b>ACKNOWLEDGEMENTS.....</b>	<b>ix</b>
<b>CHAPTERS</b>	
<b>1. INTRODUCTION .....</b>	<b>1</b>
<b>2. BACKGROUND .....</b>	<b>4</b>
2.1 Overview of Tree Layouts .....	4
2.2 Graph Drawing Approaches.....	7
2.3 Methods of Displaying Genealogical Data .....	8
<b>3. THE H-TREE AND DISPLAY SPACE ANALYSIS.....</b>	<b>11</b>
3.1 The Layout.....	11
3.2 Display Space Analysis.....	14
<b>4. SOFTWARE PROTOTYPE .....</b>	<b>18</b>
4.1 Data Set Organization and H-tree Functionality .....	18
4.2 User Interaction .....	20
<b>5. CASE STUDIES.....</b>	<b>24</b>
5.1 Duplications, Patterns, and Anomalies .....	24
5.2 Layout Comparisons .....	27

<b>6. USER EVALUATION.....</b>	<b>31</b>
6.1 Comparative Tasks.....	31
6.2 Survey Analysis and Results .....	32
<b>7. DISCUSSION AND LIMITATIONS.....</b>	<b>36</b>
<b>8. CONCLUSION .....</b>	<b>38</b>
<b>APPENDIX .....</b>	<b>41</b>
<b>REFERENCES .....</b>	<b>42</b>

## LIST OF FIGURES

1. An illustration of the quadrant-based organization of the H-tree layout, which enables an efficient display space usage, making visualization of a considerable number of generations possible while still allowing precise identification of the family branches (bottom left frame shows 18 generations related to the maternal grandfather of the central individual). Unknown ancestor regions (void spaces in the top left frame) and duplicate nodes (blue nodes in the top right frame) can also be identified clearly. .... 2
2. The parents are alternately either horizontally or vertically aligned with the child. .... 12
3. The H-tree layout for 1 generation (left) to 5 generations (right). Generations 1 through 3 are labeled to show how the nodes are moved out as another generation is added. .... 13
4. Left: Each of the 4 grandparents are placed in the center of their respective areas and their ancestors are placed around them, but still within that area. Middle: An H-tree overlay shows how the individuals are organized within each section. Each of the 16 great-grandparents of the root is placed in the center of each of the 16 sections shown. This shows the organization of the data into natural family clusters. Right: A traditional binary tree. The 16 colored sections in the middle figure correspond to the 16 colored sections of this figure. .... 14
5. An 11-generation view. Empty space is an indication of unknown information. The nodes that are colored in cyan are duplicates. Blue and pink nodes represent unique individuals. .... 19
6. The root is changed from Daniel to Hannah, adding the individuals along the path between them to the roadmap. .... 22
7. The quadrant-based organization in connection with node highlighting allows users to easily identify common ancestors. A path between the two occurrences of the individual has been drawn in. .... 26

8. A typical tree visualization. Some areas of the graph are more populated than others, which areas reflect the importance placed on record keeping by royalty, religious leaders, political leaders, and others. .... 26
9. A representation of genealogy that is contained in the Bible. The vast amount of duplicate individuals is a result of the intermarriage that exists in Adam and Eve's time. .... 27
10. Comparing node-link diagram (left), fan chart (center), and H-tree (right) when displaying eight (top) and eleven generations (bottom). .... 29
11. This figure illustrates the H-tree's inherited self-similarity property, as the magnified subgraph demonstrates the same shape and structure as the graph. .... 29

## LIST OF TABLES

1.	Space Requirements (For $k$ generations of a complete binary tree and an assumed $1 \times 1$ unit square required for each individual).....	16
2.	The number of displayable generations $k$ of a complete tree given the length $l$ available for the most constrained dimension.....	17
3.	Data Set Statistics.....	25
4.	Preferences for comparison task 1. Most and least preferred are assigned when one or two layouts are preferred over the other(s) and neutral is assigned only when there is a first, second, and third preference.....	33
5.	Preferences for comparison task 2. ....	33

## ACKNOWLEDGMENTS

This material is reprinted with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Utah's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from IEEE by writing to [pubs-permissions@ieee.org](mailto:pubs-permissions@ieee.org). By choosing to view this material, you agree to all provisions of the copyright laws protecting it.

I would like to thank Luis Gustavo Nonato and Cláudio T. Sliva for the work their efforts in helping this work be successful. I would also like to thank those who participated in the user evaluation of the PedVis as well as the members of the Institutional Review Board for reviewing and permitting the evaluation. This work was funded by the National Science Foundation (grants 04-615, IIS-0905385, IIS-0844546, CNS-0855167, IIS-0746500, ATM-0835821, CNS-0751152, IIS-0713637, OCE-0424602, IIS-0534628, CNS-0514485, IIS-0513692, CNS-0524096), grants Fapesp-Brazil (2008/03349-6) and CNPq-NSF (491034/2008-3), the Department of Energy, and IBM Faculty Awards.

## CHAPTER 1

©2010 IEEE. Reprinted with permission from Claurissa Tuttle, Luis Gustavo Nonato, and Cláudio T. Silva. PedVis: A Structured, Space Efficient Technique for Pedigree Visualization. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1063-1072, Nov./Dec. 2010.

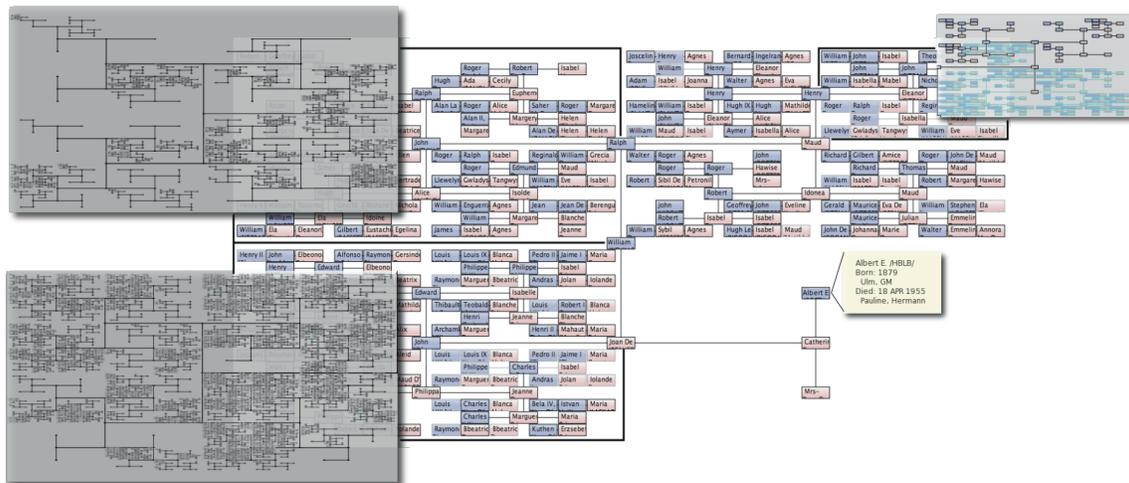
### INTRODUCTION

Whether for cultural, scientific, social, political, religious, or other reasons, genealogy has been a topic of great interest throughout the world. Currently, easy access to large genealogical databases has increased the public's interest in this kind of information, as valuable knowledge such as family traditions, historical trends, and medical information, can be obtained by tracking one's ancestors.

An important characteristic of ancestral data is the inherent binary hierarchy on which the historical trajectory is built. Such intuitive organization enables the use of a computationally well-established binary tree structure to represent and traverse genealogical data. In contrast with this representation structure, the problem of visualizing genealogical data motivates further research. As will be discussed in Chapter 2, most software designed to assist in the visualization of genealogical data relies on conventional graph drawing methods, whose ability to exploit the intrinsic characteristics

of the data is limited by space constraints. Moreover, the available ancestral visualization tools handle the natural exponential growth of the data by either displaying only a few generations at a time and/or by collapsing the space of unknown ancestors, thus necessarily giving up some structure to accommodate the demands of the space.

The present work describes a new approach to visualizing and interacting with ancestral data as it uses the H-tree layout [1]. Our methodology relies on this layout to develop an aesthetic and efficient pedigree chart visualization that both uses the display space effectively, and exploits the intrinsic characteristics of the data (see Figure 1). In fact, as will be detailed in Chapter 3, the H-tree layout devotes the same amount of screen space to each visible node of the tree while still allowing space for the unseen nodes (where the data is unknown or missing), thus providing regularity in the placement of relatives. Also in Chapter 3, we discuss the way families are grouped together and show



**Figure 1.** An illustration of the quadrant-based organization of the H-tree layout, which enables an efficient display space usage, making visualization of a considerable number of generations possible while still allowing precise identification of the family branches (bottom left frame shows 18 generations related to the maternal grandfather of the central individual). Unknown ancestor regions (void spaces in the top left frame) and duplicate nodes (blue nodes in the top right frame) can also be identified clearly.

the unique way that these groupings make effective use of the space while making the layout of the family more intuitive.

In addition, the present work shows that a user friendly and intuitive navigation mechanism can be built on the H-tree layout, allowing the traversal of the genealogical data set without losing context. Additionally, in Chapter 4, interactive mechanisms are presented to assist in the identification of loops (intersections between family branches) and navigation through the data set. In Chapter 5, the comparison of the H-tree layout with other layouts shows a visible improvement in space utilization. Fundamental questions such as “How much of my family tree is known?” and “Which parts of my family tree are known?” are answered. Finally, a task-based comparative user evaluation is provided.

The contributions of this paper follow from unique strengths of the H-tree layout, which will benefit users of genealogical software. These unique strengths include increasing the number of generations viewable at a time and using 100% of the space for the set of family groups for each generation. Other strengths of the H-tree layout, which may also be properties of other layouts, include a consistent and regular placement of ancestors, empty space for ancestors whose identities are not contained in the data set, a self-similarity property, and duplicate node and loop identification. Also, patterns in data as seen in the H-tree layout have potential to become familiar to the user, which will increase the user’s ability to navigate through the data. The user evaluation is the final contribution, as it gives an indication of user acceptance for the new layout.

## **CHAPTER 2**

### **BACKGROUND**

Effective techniques for displaying hierarchical data use the display space efficiently, ease the user's task of understanding the data, and have an aesthetic appeal. Furthermore, such techniques should rely on interactive mechanisms that encourage a natural traversal of the hierarchy, exploiting the intrinsic properties of this type of data. Nevertheless, forfeiting structure to increase space utilization, many plotting methodologies for hierarchical data are unable to fully represent these intrinsic properties. In the particular case of genealogical data set visualization, even fundamental questions such as "How much of my family tree is known?" and "Which parts of my family tree are known?" are not always answered by current software packages, particularly when the tree is larger than just a few generations.

The overview presented in the following section aims to support the aforementioned assertions, pointing out the strengths and weaknesses of the main techniques devoted to visualizing hierarchical data.

#### **2.1 Overview of Tree Layouts**

There are many methods for the layout and display of hierarchical trees. The traditional approach [14] uses a node-link representation organized in a top-down or left-

right manner, placing children below or to the right of the parent respectively. One main advantage of this layout is that individual generations are easily discernible. However, the visualization becomes crowded as the number of individuals in a particular generation becomes large.

SpaceTree [27] utilizes icons to represent unexpanded subtrees, scaling the displayed branches of the tree to best fit the available space. Although the SpaceTree has the same strengths and limitations of the traditional node-link layout, it allows a better understanding of the breadth and depth of the unexpanded subtrees through the use of triangular icons.

In the radial layout [6, 14, 25], children are placed in a circular wedge around the parent, forming a fan-like pattern. As in the top-down (or left-right) plot, the generations can easily be identified while still allowing a clear distinction between the paternal and maternal ancestral branches, making this plotting scheme popular for genealogical data visualization. However, the radial layout also faces directional exponential growth, often limiting the visualization to 5 or 6 generations. The balloon layout [14] is similar to the radial layout, except children are placed in a full circle around the parent node, producing a graph that is fundamentally the same as the radial layout when used with ancestral data.

Treemaps [14, 15, 31] have been designed to use display space effectively. In this plotting scheme, child nodes are recursively mapped to a rectangular region within the parent node's region. One of the treemap's strengths is that the entire data set is mapped into the display area, enabling a global view of the hierarchical information. Also, the sizes of rectangular regions as well as sound and color are used to represent the data more effectively. However, this method of visualization works best when using data sets with

few hierarchical levels to avoid congesting the visualization. Although viewing a genealogical data set with this method is possible, it is less intuitive because there are no links between nodes to signify the parent-child relationship.

Nguyen et al. [25] presented a different space-optimized tree visualization scheme. Their method uses weights and wedges to determine the position of the nodes in order to maximize space utilization. Although this technique can place thousands of nodes in a single display area, the lack of regularity in the placements both impairs the user's ability to comprehend the family relationships among elements and makes it nearly impossible to assess the completeness of the tree.

Hyperbolic Trees [14, 17] use a focus + context technique to display a hierarchy in two-dimensional hyperbolic space as well as in 3D [23]. This method gives the user the ability to navigate through the data by changing the focus. As the focus is changing, the context is maintained by smooth animation. Portions of the tree are expanded and collapsed as they are respectively brought closer to or moved further away from the focus. Although this method focuses on using the space effectively, the distinction between the paternal and maternal lines is less apparent due to less standard positioning of nodes, and is further diminished as the focus of the tree constantly changes through navigation. Also, although this method does well with visualizing binary trees, it is not specifically optimized for them, nor is it designed to show tree completeness.

Space Filling Curves [22] are particularly useful when working with dense networks that contain clusters, as the nodes within a cluster are placed in an aesthetically pleasing manner. However, due to the non-intuitive placement of nodes in regards to ancestral data, familial relationships and tree completeness cannot be easily identified.

The H-tree layout [1, 2, 14, 25], based on the H-fractal, has been previously used for representing binary trees (other fractal-based layouts are described in [16, 19]). One main advantage of this layout is its alternating growth mechanism, which effectively uses the space by spreading the nodes aesthetically onto the display area, not favoring one specific direction over another. Therefore, this plotting scheme is very effective, especially when dealing with fairly dense binary trees. Furthermore, by leaving space for all ancestors whether known or unknown, this layout clearly shows family relationships among individuals. In fact, along with having strengths of its own, the H-tree exhibits many favorable properties present in other methods, making this a very effective layout for visualizing ancestral trees. Despite these favorable properties, to our best knowledge, the H-tree layout has not been used for displaying pedigree data, making the present work the first to use such a plotting mechanism in that context.

Along with the mentioned favorable qualities of the H-tree layout, there are also weaknesses. First, we are dealing with a binary tree, so ancestors can be represented naturally, but descendants cannot. Second, although many individuals will belong to multiple generations, the H-tree layout puts individuals from each particular generation in a grid-like pattern, making it more difficult to distinguish generations. Also, bloodlines are represented in zigzag patterns rather than in straight lines.

## 2.2 Graph Drawing Approaches

The graph drawing community has developed many space-efficient techniques for drawing trees [8, 10, 30]. These techniques include HV-drawings [26] that achieve the optimal  $O(N)$  area for a complete binary tree and  $O(n \log n)$  for a general binary tree, and straight-line orthogonal drawings [3, 5, 33] that achieve  $O(n \log \log n)$  area.

Although these techniques prove to be very space-efficient, we are in need of a technique that includes a very regular node placement of the nodes. With some optimal techniques, “the relative positioning of sibling nodes varies greatly, making it difficult to perceive the overall structure of the tree” [20]. However, we are still interested in the H-tree layout because it is symmetric with regular placement of nodes.

### **2.3 Methods of Displaying Genealogical Data**

The literature has also presented a set of works specifically devoted to visualizing genealogical data. Furnas and Zacks [9], for example, interpreted family trees as multitrees, enabling the visualization of family tree data through a fisheye-based multitree browser. As an application, the authors presented a royal genealogy family visualization where the focus of the fisheye view was the royal line. The graph also contains the first-order ancestors as well as the first order descendants of that line. Although this scheme does not follow the traditional order by generation view, interesting information can be extracted by maintaining focus on the royal line. This method focuses on a selected subset of data, but does not attempt to organize individuals in a space-efficient way.

McGuffin et al. [19] presented an elegant method of navigating through a genealogical tree by selecting a subset of data to be displayed. The data is displayed using a dual tree, which generalizes the hourglass chart or centrifugal view. This technique provides a good visualization of a data subset as well as a very user-friendly interactive environment. However, this method presents the same limitations as the traditional node-link layout, especially when attempting to expand all ancestral branches of the tree, thus displaying a pedigree chart.

Wesson et al. [35] built an interactive mechanism into the traditional node-link structure for displaying family trees. The tree is arranged in an hourglass chart (ancestors are above the root and descendants are below the root) and the user is able to expand and collapse portions of the graph in order to see desired information. This method also allows subset visualization, but the overall view of the graph is not optimal.

GenoPro [12], Kim et al. [24], and Priestly [28] include additional ways to display family and/or historical relationships, two of which are timeline-based approaches. GenoPro uses symbols on a genogram to depict a variety of family relationships. Kim et al. also focus on representing different family relationships, but use a timeline approach to display the individuals. Additionally, Priestly arranged a large number of historical names on a timeline and organized them in the  $y$  dimension according to selected categories. Although these works focus on different aspects of genealogical data, these additional representations are complementary as they provide information that other views do not show. The H-tree layout, in particular, does not depict these types of relationships.

Current genealogical software packages include a variety of charts for data display. Ancestor, descendant, and hourglass charts are included in most of the top rated packages [7, 11, 18, 21, 29, 32] (as rated at [www.consumersearch.com/genealogy-software](http://www.consumersearch.com/genealogy-software) and [genealogy-software-review.toptenreviews.com](http://genealogy-software-review.toptenreviews.com)). Other charts included in many of these packages are the fan chart, maps, timelines, and the bow tie chart. Together, these charts give a variety of views to suit individual wants and needs. We feel that the H-tree would be a great addition to the existing variety.

As can be seen, the existing methods for pedigree visualization are useful and provide many interactive interfaces, but do not attempt to optimize display space usage, but instead focus on a portion of the data. These techniques tend to be limited by the number of generations that can be displayed and make it difficult to determine the completeness of the tree. We will show that the H-tree uses the display space effectively, enables easy identification of patterns in or density of regions by preserving the structure, and provides a very intuitive layout for navigating through the hierarchy.

## **CHAPTER 3**

### **THE H-TREE AND DISPLAY SPACE ANALYSIS**

As mentioned in previous work [14, 25], the H-tree layout has been previously used in graph drawing, and is known for performing well on balanced binary trees. It is also commonly used in VLSI design [2] to attain equal propagation delays, and in analyzing the traveling salesman tour [1]. The present work utilizes the H-tree for a completely different purpose, namely the visualization of genealogical data, particularly the pedigree chart.

The main reason we chose the H-tree visual arrangement is its efficient display space usage, as the inherent exponential growth of genealogical data makes the display space “a scarce commodity even in moderately-sized charts” [6]. Before comparing the effectiveness of the H-tree with other traditional genealogical layouts, we will describe how the H-tree layout is managed in our context.

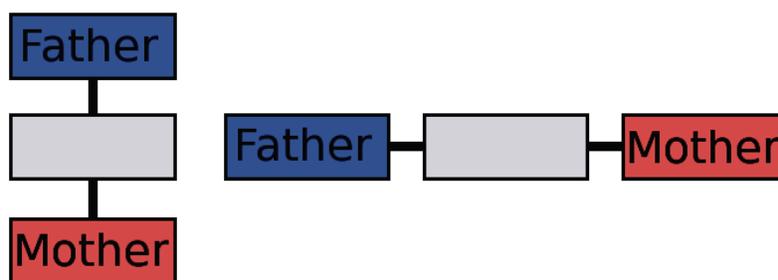
#### **3.1 The Layout**

To display a pedigree chart, we need to select a root individual and display all of his ancestors up to a selected number of generations. For example, in the traditional binary tree layout, the root node is placed at the left and the tree is built recursively in a

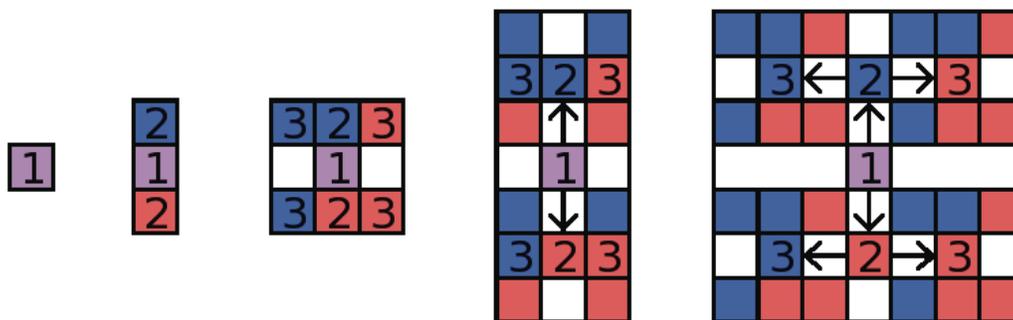
left to right manner by linking each node in the current generation to two nodes in the next generation.

In the H-tree layout, however, we place the root at the center of the display, positioning the root's parents either above and below or to the sides of the root individual. In fact, the positioning of parents is alternated between generations. For example, if an individual's parents are positioned vertically, then the grandparents are positioned horizontally. By convention, the father goes above or to the left and mother below or to the right of their child, making an organized and consistent position for each ancestor, which in turn makes their relationships more easily identifiable. Figure 2 illustrates the construction just described.

The formation of the H-tree over the first five generations is shown in Figure 3. Notice that an I shape rather than an H shape is seen in the odd generations. Rather than rotate the tree to force a consistent H shape, we keep the nodes in a consistent placement relative to each other. In other words, by allowing the I shape, if an individual's father is placed above him, then the father and all his ancestors remain above the individual (the same is valid for the mother), making a clear separation between both ancestral branches. In contrast to the consistency of the node placement, the distance between the nodes



**Figure 2.** The Parents are alternately either horizontally or vertically aligned with the child.

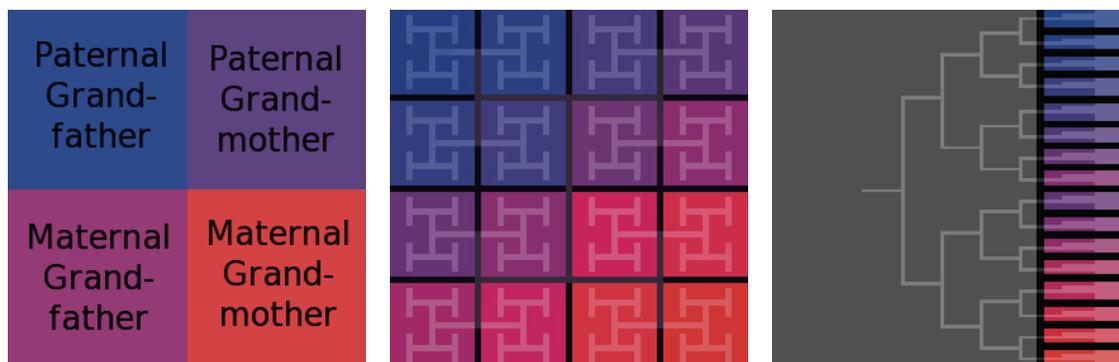


**Figure 3.** The H-tree layout for 1 generation (left) to 5 generations (right). Generations 1 through 3 are labeled to show how the nodes are moved out as another generation is added.

increases with the number of generations, as the links are stretched to make room for the new generation. As we can see in Figure 3, the individuals in generations 2 and 3 are moved further from their child to make room for generations 4 and 5.

Consider a root individual whose father is placed above and mother is placed below. A horizontal line can be drawn through the center of the display space, leaving the father's side of the family above it and the mother's side below it. Now, add the grandparents to the sides of the parents and draw a vertical line through the center of the display space so as to form four sections, quadrants, or family groups. Each section corresponds to a grandparent and his or her ancestors. This description can be repeated by recursively cutting each of the sections in half with each new generation, as illustrated in Figure 4. The traditional view of the binary tree can also be divided into sections, but the sections would be divided by horizontal lines only (see the right image of Figure 4), making it more difficult to visualize a specific branch of the family.

We use the term family grouping to describe the sections mentioned in the previous paragraph and define a family group of an individual as the area in which that individual and his/her ancestors are enclosed. Using this definition, we can now make



**Figure 4.** Left: Each of the 4 grandparents are placed in the center of their respective areas and their ancestors are placed around them, but still within that area. Middle: An H-tree overlay shows how the individuals are organized within each section. Each of the 16 great-grandparents of the root is placed in the center of each of the 16 sections shown. This shows the organization of the data into natural family clusters. Right: A traditional binary tree. The 16 colored sections in the middle figure correspond to the 16 colored sections of this figure.

the following observation about the H-tree layout: for each generation, the set of family groups of all of the individuals within that generation fill the entire space with non-overlapping rectangles of equal size. In other layouts, the set of family groupings for a particular generation does not fill the entire space. Thus the H-tree shows a unique asset, which provides a valuable feature when viewing pedigrees (observe from Figure 4).

### 3.2 Display Space Analysis

The purpose of this section is primarily to assist the reader in understanding the number of generations that can fit in a given space using each of the respective layouts. The space complexity of the H-tree is already known [30], as is that of the fan chart and the traditional tree. McGuffin et al. [20] do a good job of quantifying the space efficiency of the latter two layouts as well as of other 2D trees. In particular, they focus on using space well. For example, they place importance on the label area available for

even the smallest node. We place importance on this as well as being able to see the empty space. Therefore, we define  $N$  as the number of nodes in a complete binary tree and assume that each individual requires a  $1 \times 1$  unit square of space.

Observe the for a complete tree with  $k$  generations, the number of nodes  $N$  is:

$$N = 2^k - 1 \quad (1)$$

The traditional node-link layout requires an area of  $k \cdot 2^{(k-1)}$ . Let  $x_k$  and  $y_k$  be the dimensions of the H-tree layout. Then the area required is  $x_k \cdot y_k$ . Where  $x_k$  and  $y_k$  are found by induction, giving the following equations:

$$\begin{aligned} x_{k(odd)} &= 2^0 + 2^1 + \dots + 2^{(k-1)/2} = 2^{(k+1)/2} - 1 \\ x_{k(even)} &= 2^0 + 2^1 + \dots + 2^{(k-2)/2} = 2^{k/2} - 1 \end{aligned} \quad (2)$$

$$\begin{aligned} y_{k(odd)} &= 2^0 + 2^1 + \dots + 2^{(k-1)/2} = 2^{(k+1)/2} - 1 \\ y_{k(even)} &= 2^0 + 2^1 + \dots + 2^{k/2} = 2^{(k+2)/2} - 1 \end{aligned} \quad (3)$$

From equations (1), (2), and (3) we conclude that the display space required, in terms of the number of possible elements  $N$ , is  $O(2^k)$  or  $O(N)$  for the H-tree layout (see Table 1).

We would like to compare the display space required for the previous two layouts with that of the fan chart layout. However, due to the non-rectangular shape of the individual nodes, it is better to compare only the most crowded dimensions of the layouts. This is justified by the fact that the ability to distinguish between individual nodes is directly related to the space given to each node in the most limiting or crowded

**Table 1.** Space Requirements (For  $k$  generations of a complete binary tree and an assumed  $1 \times 1$  unit square required for each individual).

	Traditional	H-Tree
Number of Nodes	$2^k - 1$	$2^k - 1$
Area	$k \cdot 2^{k-1}$	Odd $k$ : $(2^{(k+1)/2} - 1) \cdot (2^{(k+1)/2} - 1)$ Even $k$ : $(2^{k/2} - 1) \cdot (2^{(k+2)/2} - 1)$

dimension. Observe that the most limiting dimension of the fan chart layout is around the circumference of the circle, which needs room for  $2^{k-1}$  nodes. Since the circumference is  $\pi \cdot \text{diameter}$  and the dimensions of the chart equal the diameter, the length required in both the  $x$  and  $y$  dimensions is  $2^{k-1}/\pi$ . Table 2 shows the number of generations that can be drawn in each layout given the length constraints of 10, 100, and 1000 units. For complete trees, in comparison with other layouts, the H-tree layout is able to organize a greater number of distinguishable individuals in the given space.

Although there are many complete or nearly complete family trees, the display space required is reduced for non-complete trees when branches representative of unknown individuals are collapsed. There are other reasons to collapse branches in a tree as well: the individuals have already been drawn elsewhere in the tree, some of the ancestors are unknown, or the particular portion of the tree is uninteresting to the user. Collapsing branches of the tree due to these or any other reasons reduces the space required. The caveat to this is that the user will have more difficulty in determining the completeness of the tree and the placement of relatives will change as the data changes, making it more difficult to identify how individuals are related to each other and the root. Thus, collapsing branches reduces space needed, but will reduce the effectiveness of the layout.

**Table 2.** The number of displayable generations  $k$  of a complete tree given the length  $l$  available for the most constrained dimension

Length ( $l$ )	Generations Possible ( $k$ )		
	Traditional	Fan Chart	H-Tree
10	4.32	5.97	5.92
1000	7.64	9.3	12.32
1000	10.97	12.62	18.93
Equations	$l = 2^{k-1}$	$l = 2^{k-1} / \pi$	$l = 2^{(k+1)/2} - 1$

Assuming a collapse in the traditional tree, but not in the H-tree, we will again evaluate the space efficiency while keeping the tradeoffs presented in mind. With a collapsed traditional tree, its most limiting dimension,  $ttmld$ , is the maximum of the number of generations,  $k$ , and the number of individuals in the most populated generation, which is at most  $2^{k-1}$ . Then, the H-tree is still less constrained in its limiting dimension if  $ttmld > 2^{k/2}$ , where  $k$  is the number of generations. Otherwise, the traditional tree is less constrained, and if space-efficiency is more important than seeing the empty space, either empty space in the H-tree can be collapsed, or it would be better not to use the H-tree view in this case.

Another consideration is the shape of the viewing space available. For example, if one viewing dimension is significantly larger than the other, the constraining dimension evaluation may also change.

We have shown strengths of the H-tree layout and its contributive qualities in terms of space along with a comparison with other layouts, presenting important tradeoffs. The H-tree is particularly useful and space effective when dealing with complete binary trees, with moderate to densely populated graphs, as well as with any trees in which the user desires a structured, consistent placement of respective relatives.

## CHAPTER 4

### SOFTWARE PROTOTYPE

This chapter provides an overview of the H-tree software prototype, PedVis, which has been developed to visualize ancestral data. The system, which was written in C++ and Qt in order to ensure portability, provides an interactive H-tree visualization with a roadmap that keeps track of the path between the original root, and the current root.

#### 4.1 Data Set Organization and H-tree Functionality

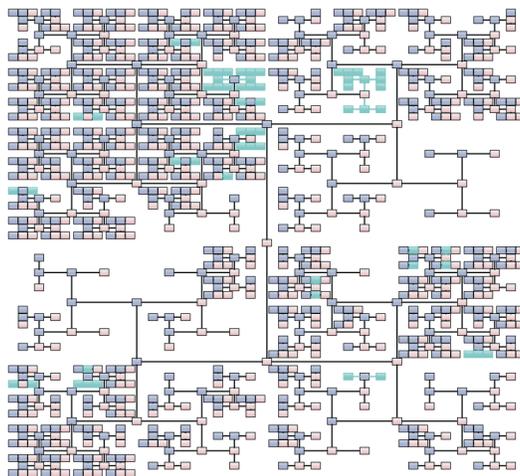
Most genealogical software programs export and import GEDCOM (Genealogical Data Communication) files. GEDCOM uses a data model that links individuals together through family records. The individual record contains reference to the family record where he/she is a child and where he/she is a spouse. Likewise, a family record contains reference to the individuals who are children or spouses in the family. The specification can be found online [4].

PedVis reads any number of GEDCOM files and stores the information in two arrays. The first array contains the family structure, which has pointers to the parents and children of the family. The other array stores the individual structure, that is, the individual's birth and death information as well as marriage and other genealogical

information. The individual's structure also contains pointers to the families to which she/he belongs.

The data that we have just described is not guaranteed to be cycle-free. Similar to the approach of McGuffin and Balakrishnan [19], cycles in the graph are eliminated through the allowance of duplicate individuals. In other words, when an individual is reached through more than one path, we create a duplicate of the individual, which eliminates the cycle. As a result, the individual will show up multiple times in the graph (see Figure 5). Henry et al. [13] provide studies on effective representations of duplicate nodes.

The position of a node in the graph is determined by its relation to the root node. This root is either preselected, or is the first individual in the first loaded file by default. The root is placed in the center and its ancestors are arranged in the manner described in Section 3.1. To increase readability, the software optionally draws arrows to help direct the eyes as they follow the links from one individual to another.



**Figure 5.** An 11-generation view. Empty space is an indication of unknown information. The nodes that are colored in cyan are duplicates. Blue and pink nodes represent unique individuals.

When displaying a pedigree chart, ordering by generation is usually very important. Although it is possible to determine which generation any individual belongs to based on his/her placement in the H-tree, the user may enable the levels feature where the visual distinction of generations is made more clear by varying the saturation of the pre-assigned color (blue for male and pink for female). This coloring choice is visually more pleasant and is more intuitive than hue variation.

## 4.2 User Interaction

When launched, the software comes up with a default five-generation view of a pedigree chart, which allows ample space for displaying individuals' information such as birth, death, and marriage (see Appendix for analysis of the space required for each individual). As the number of generations increases, it becomes more appropriate to display this information in a tool tip display, which can be launched for each individual by moving the mouse over his or her portion of the display. Also, clicking on an individual places his or her information in the side panel for easy reading.

The number of generations displayed is increased through a user parameter, and with each new generation, the available space per individual is reduced by a factor of two. In our implementation, the maximum number of viewable generations is 18, which is a natural limitation because the space occupied by each individual in an 18-generation display is so small that viewing more generations at a time would not be effective. However, additional generations can be viewed by selecting a new root for the tree. The new root is selected by pressing the shift key and clicking on the desired individual, who is automatically moved to the center of the display. In this way, the user is able to navigate through the data set.

To keep the context of how the new root is connected to the original root, a scrollable roadmap whose individuals are numbered by generation is provided at the bottom of the display. This roadmap shows all individuals on the path between the original root and the current root (see Figure 6). As each new root is selected, it is placed on a history stack to enable backtracking. When the back button is pressed, PedVis removes the current root from the top of the stack, redraws the H-tree with the previous root, and removes individuals from the previous root up to and including the current root from the roadmap.

Additionally, users are able to view the H-tree with any member of the roadmap as the root by pressing shift while clicking on the individual in the roadmap. When this happens, the selected (clicked) individual is highlighted in the roadmap and the H-tree is updated with the new root. The history stack and the individuals in the roadmap remain the same. For simplicity, let us call this state, where the root of the H-tree is an individual who is not at the end of the roadmap, an intermediate view. From this view, pressing the back button has the same behavior as before because the history stack has not changed. Selecting a member of the roadmap will take us to another intermediate view. However, when a member of an intermediate view's H-tree is selected, we proceed as follows: the individuals between the H-tree's root and the end are removed from the roadmap and the history stack if they are on it, the H-tree's root is placed on the stack if it is not already on it, the newly selected root is placed on the stack, the appropriate individuals are added to the roadmap, and the H-tree is updated with the new root. Thus, the roadmap remains updated and becomes an effective navigation tool.



The previously described tree navigation is enhanced by the ability to view portions of the tree in greater detail. To accomplish this, we use a panning and zooming scheme, which can be done manually or automatically. The view is scaled and translated manually with mouse and keyboard operations. Selection zoom also allows the user to select a region he/she wants to see in greater detail. The region is then brought to fill the display window through an animated transition [34]. A simple key-press and click initiates a return to the previous view.

Additionally, recall that cycles are eliminated through duplication. We distinguish these duplications by coloring the first occurrence of an individual in pink or blue as usual, but any additional occurrences in cyan, allowing the user to easily see the number of unique individuals as well as the number of duplicates. When the “highlight instances of selected” mode is turned on, the user is able to select an individual, causing its instances to be highlighted with magenta.

Finally, the fan chart and traditional views have been added to the tool for user evaluation purposes. In an effort to minimize bias toward any view, most of the interaction techniques described apply to all three views. The features that do not apply to all three views are most relevant to the view in which they are available. For example, although eliminating duplicates in the traditional view saves space, this is not relevant to the other two views. So, a checkbox has been added to allow for duplicate elimination in the traditional view only.

## CHAPTER 5

### CASE STUDIES

In this chapter, we show how the proposed system can be used for exploring ancestral charts as well as analyzing trends within the data. We also show the effectiveness of the H-tree layout in comparison with two other techniques that are widely used, namely the fan chart and the node-link diagram.

We perform our exploration and analysis on a set of three data sets, whose statistics are shown in Table 3. Notice that data set A is “shallow.” That is, information is known for a small number of generations. However, the ratio between individuals and generations reveals that it is more densely populated than the other data sets, which are deep trees that have some sparse regions, but also contain some very densely populated regions.

#### **5.1 Duplications, Patterns, and Anomalies**

Although the H-tree layout does not yet take advantage of all the potential statistics or trends that can be shown through color (i.e. color according to geographical location, genetic conditions, lifespan), the duplication feature is a taste of such potential features. When an individual is selected, all nodes on the screen that represent that individual are highlighted, making it obvious when the paternal and maternal ancestral lines of the root individual share a common ancestor. As a result, the question “Is she/he

**Table 3.** Data Set Statistics.

Data Set	Generations	Individuals	Individuals/Generation Ratio
A	14	1936	138.29
B	80	3513	43.91
C	122	8092	66.33

related to the root through both ancestral lines?” is answered. Those who study genetic disorders might ask this question when studying some genetic diseases that are inherited only when they are passed through both parents. Figure 7 uses data set A to illustrate this feature, revealing that the root individual has a common ancestor (highlighted) from paternal and maternal lines.

Making the logical assumption that older information is harder to find, it makes sense that ancestral trees begin with a root individual, grow exponentially for a number of generations, and then taper off. However, this is not always the case: the sparse tree representing data set B has a couple of full branches (Figure 8). The root individual’s generation has significantly fewer individuals than some of the previous generations. Thus, the tapering off has already occurred, and we are presented with two branches where the amount of information known suddenly increases. In this case, the sudden expansion reflects the royal-family tradition of keeping genealogical records. Although such trends will also be seen in other layouts such as the node-link layout, the structure-preserving property of the H-tree layout presents the user with an increased sense of where in the family this expansion occurs, and also organizes the data in such a way that the users are able to more easily remember and recognize patterns formed by specific subsets of the data.

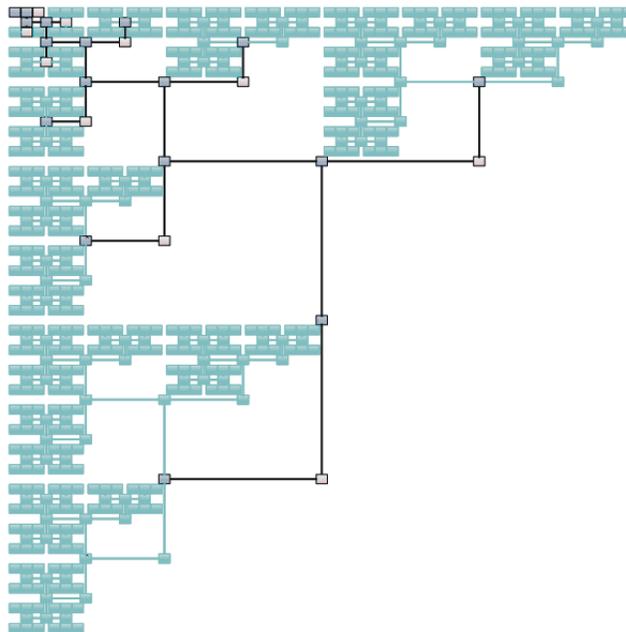


Another interesting pattern is seen in data set C. A branch of the tree connects to a person in the Bible whose genealogy links all the way back to Adam and Eve. Figure 9 shows an 11-generation ancestry chart with Adam and Eve in the 11th generation, which reveals a high rate of intermarriage.

The characteristics presented in this section are further investigated in the user study, as the user is able to explore and analyze the data with each of the three layouts. In addition, the figures we have provided show these characteristics in the H-tree layout, demonstrating its flexibility and potential in the respective visualizations.

## 5.2 Layout Comparisons

Rather than condensing a tree to fill empty space, leaving the space unfilled emphasizes unknown ancestry and makes it easier to determine certain kinds of



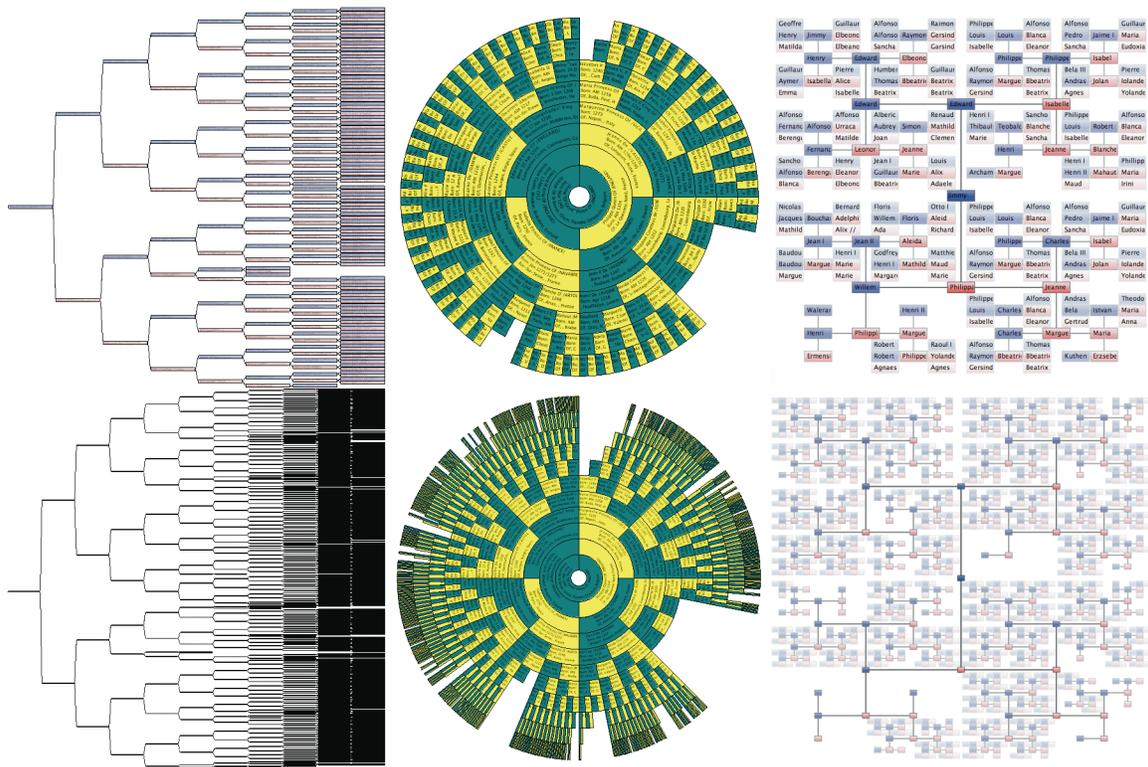
**Figure 9.** A representation of genealogy that is contained in the Bible. The vast amount of duplicate individuals is a result of the intermarriage that exists in Adam and Eve's time.

relationships. Therefore, although the empty rows and columns can be removed in the H-tree and branches can be interactively collapsed in the fan chart [6], we have chosen to depict empty space in both layouts to allow for visual identification of missing branches and to improve the visual representation of family relationships. For the traditional tree, we have chosen to fill the unoccupied space.

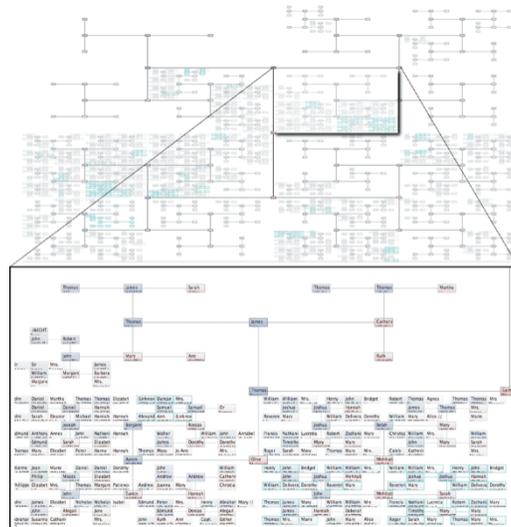
The H-tree alleviates exponential crowding. Figure 10 shows both 8 and 11 generation views of data set B using a node-link diagram (binary tree), fan chart, and H-tree respectively. It can be seen that the binary tree's 8th generation is already crowded, whereas in the other two representations, individuals are still distinguishable. With 11 generations, however, the binary tree and the fan chart both have generation(s) with indistinguishable individuals, whereas the 11th generation in the H-tree is still distinguishable enough that selecting the individuals with the mouse is still easy.

The H-tree preserves structure. Figure 10 demonstrates the increased ability to understand family structure through the H-tree's unique node organization scheme and non-collapsed branches. For example, consider the void region on the bottom left of H-tree layout (right column in Figure 10). Due to quadrant-based subdivision, association of the empty space with the root's maternal great grandmother is straightforward. However, notice that in the node-link diagram, it is difficult to tell the number and location of the unknown ancestors.

Finally, the H-tree provides an intuitive layout for navigation through a data set. In fact, the self-similarity property of the H-tree substantiates the preservation of the layout's organization under transformations (Figure 11). Readability is increased by this unvarying layout during interaction. Notice from Figure 11 that in the magnified region,



**Figure 10.** Comparing node-link diagram (left), fan chart (center), and H-tree (right) when displaying eight (top) and eleven generations (bottom).



**Figure 11.** This figure illustrates the H-tree’s inherited self-similarity property, as the magnified subgraph demonstrates the same shape and structure as the graph.

the father is still above and the mother is still below the individual. That is, the zoomed region looks like the original H-tree in shape and structure. This property is also apparent with the node-link diagram when branches are not collapsed, but is not a characteristic of the fan chart.

## **CHAPTER 6**

### **USER EVALUATION**

To evaluate the usefulness of displaying ancestral data in the proposed layout, we conducted a study with 21 users (1 advanced, 4 intermediate, 1 intermediate/beginner, and 15 beginner), whose experience level was self-determined. Whether or not users were domain experts was not confirmed. Participants were asked to perform some introductory tasks to familiarize themselves with the new layout (H-tree only) and two tasks aimed at comparing the new layout with the two others. The introductory tasks encouraged the user to become familiar with the system by identifying people within the same generation, locating respective relatives through the assistance of re-rooting the tree, and increasing the number of generations. Answers to associated task questions reveal accuracy. The comparative tasks are described below. An optional bonus task allowed interested participants to further explore the data with any layout while trying to find the biblical couple Adam and Eve. In addition, users completed a survey, evaluating the usefulness of the new layout relative to the others.

#### **6.1 Comparative Tasks**

Participants were asked to perform two comparative tasks and record their observations, the first of which involves following navigation instructions to come to a

view that is dense enough that crowding begins to become a problem. Although users were only asked to perform each of these two tasks and record their observations, many users provided rankings. Results are shown in Table 4 and an analysis is provided in the following section.

The second comparative task involves finding a royal line in the data set provided, which requires the user to re-root the tree and optionally increase the number of generations in order to navigate to the dense areas of the tree and find the royal line. As this task is a challenge to many users, they are given a walkthrough of this task with another data set. The royal line is to be found once with each layout, but since the order in which the layouts are used is likely to have an effect on the results due to an increased familiarity with the data set, the order is varied among the user studies. Results are shown in Table 5 and analyzed in the following section.

In the bonus task, users were asked navigate through some 100+ generations to find Adam and Eve. Of those who finished this task, two used the H-tree and one used the fan chart to find the couple.

## **6.2 Survey Analysis and Results**

In the comparative tasks, users responded to the request to record observations with the following about the H-tree, it: “addresses exponential growth very well,” “gives a much better indication of where ancestor groups are,” “has the best distribution about close ancestors,” and “conveys information about lineage more easily and [conveys] more information about the density of the graph.” In response to the navigation task of finding the royal line, one user wrote that it was “possible with all...but FAR easier with

**Table 4.** Preferences for comparison task 1. Most and least preferred are assigned when one or two layouts are preferred over the other(s) and neutral is assigned only when there is a first, second, and third preference.

All Users			
	Most Preferred	Neutral	Least Preferred
H-Tree	12	1	2
Fan Chart	4	6	5
Traditional	3	1	11
Intermediate/Advanced Users			
	Most Preferred	Neutral	Least Preferred
H-Tree	4	0	0
Fan Chart	1	2	1
Traditional	0	1	3

**Table 5.** Preferences for comparison task 2.

All Users			
	Most Preferred	Neutral	Least Preferred
H-Tree	6	2	1
Fan Chart	2	1	6
Traditional	4	2	3
Intermediate/Advanced Users			
	Most Preferred	Neutral	Least Preferred
H-Tree	3	1	0
Fan Chart	1	0	3
Traditional	2	1	1

the H-tree.” Similarly, in response to the task of finding Adam and Eve, one user wrote, “The H-tree was the only one I was able to use.”

In both comparison tasks, we assigned numerical values between 1 and 3 based on the rankings assigned by participants. ANOVA tests on this data reveal that for comparison task 1,  $p=0.000$  for all users and  $p=0.0034$  for intermediate/advanced users. For comparison task 2,  $p=0.071$  for all users and  $p=0.12$  for intermediate/advanced users. Further t-tests with comparison task 1 reveal that the preference for the H-Tree layout is statistically significant as  $p<0.01$  when the H-Tree layout is tested against each of the other two layouts (the largest  $p$  value being 0.0087).

The survey asks users to evaluate the proposed layout’s usefulness. The results are presented in four categories:

**Impression on user.** The direct answer to the question “Do you feel this layout would be useful if it were to be included in genealogical software?” is yes for 12 users and maybe for 9. Although only 12 users answered a direct yes to the previous question, 17 users either liked or loved the layout and only 4 were indifferent.

**Space Usage.** On a scale of 1-10, the average ranking for how well the space was used is 8.23 for the H-tree, 5 for the fan chart, and 4.81 for the traditional tree. The family clusters image (Figure 4) was provided in the background section of the user study. When asked whether or not the way these family clusters used the space made a difference in their ability to understand the data, the users responded with the following: 5 do not know, 1 no, 2 barely, 8 yes, and 5 definitely. All users felt that the empty space left by the fan chart and H-tree was useful (9 definitely, 10 yes, 2 barely).

**Organization.** We expected that the H-tree layout would make it easier to determine family relationships, but harder to determine an individual's generation. Our expectations were met as the median response for the difficulty of each was slightly easier and slightly harder respectively.

**Learning curve.** Most users did not use the arrows to assist them, but 4 users thought they were very helpful. The difficulty of getting used to the layout was ranked easy by 10 users and moderate by 11 users. The difficulty in using the software was ranked easy or normal by most users. However, the introductory task questions were only answered with 78% accuracy, which leaves room for improvement.

Overall, we found that the H-tree layout was easy to get used to, traded generational organization for family organization, used the space well, and made a good impression on the users. One goal of the study was to compare the different views at a point where crowding becomes a problem because that is where the H-tree is most useful. This goal was satisfied with favorable results.

Finally, the study is limited by the variety of users represented. We leave a more formal study with domain experts who are able to use the layout over time to future work. Also, it would be interesting to incorporate this layout into existing genealogical software and to receive user feedback.

## **CHAPTER 7**

### **DISCUSSION AND LIMITATIONS**

Chapters 3, 5 and 6 provide a theoretical as well as practical discussion about the contributions of the H-tree layout to genealogical display methods. In fact, in addition to retaining some qualities of other techniques, the H-tree layout holds particular characteristics that cannot be found in other display schemes, making it a useful addition to the currently available tools.

The fractal structure on which the H-tree relies provides a series of advantages that are absent in other genealogical data visualization tools, such as quadrant-based organization, zoom/translation invariance, and efficiency in display usage (reduction of exponential crowding). The self-similarity property of the fractal structure encourages a consistent navigation mechanism whose structure is preserved under transformations.

The ease of identifying the unknown ancestors' precise location in the data set is another important feature of the H-tree layout. In fact, the relation between the individuals from whom the ancestral information is unavailable and the known individuals becomes obvious through the unique distribution of the families (or family groups).

Among the functionalities implemented in the H-tree layout, we highlight the roadmap, which both provides a context of the location of the current root relative to the original one and eases the task of navigation through many generations.

This layout is sufficient when the user is in need of traversing ancestors, but when descendant traversal is desired, we propose that this layout be used in conjunction with an existing layout in genealogical software. For example, the user could be navigating through the data with a dual tree [19] and decide that he/she wants to explore the ancestry of a particular individual. The H-tree can then be drawn with that individual as root, and navigation can resume through the H-tree view. On the other hand, when navigating through ancestors in the H-tree view, users who desire to see the descendants of a particular individual can select to see that individual as root in the dual tree view (or any other view). Although the H-tree is sufficient when only ancestors are of interest, pairing the H-Tree with another view increases flexibility.

The limitations of this method are a direct result of its strengths. Due to its efficient use of space, the cost of displaying duplicate individuals as well as leaving space for unknown individuals is significantly less than it is in other representations. As a result, displaying duplicate individuals as well as empty space is affordable, but is limiting when viewing a sparsely populated portion of the tree, or when intermarriage rates are high and the user's priority is displaying only unique individuals. In addition, this layout displays only binary trees naturally and using it to display descendant charts does not seem logical, thus another layout should be used in conjunction with this one in order to display descendants.

## **CHAPTER 8**

### **CONCLUSION**

Inspired by the desire to display ancestral data with optimized space utilization, we have shown how the H-tree successfully combines space optimization ideas with the requirements imposed by the data's inherent attributes. Due to its structural organization and ability to both reserve space for unknown individuals, and display duplicate individuals at a low cost, this novel approach for the layout of ancestral data shows potential usefulness in visually analyzing genetic disorders, life span, physical location, and intermarriage.

We provided a space usage analysis, which shows the conditions in which the H-tree can be used effectively. We have also implemented an interactive software prototype that allows the user to navigate through and extract information from the ancestral tree. Interactive features give the user the ability to select the number of generations to view, select a new root to display a different portion of the tree, select a region to magnify, display all instances of a particular individual, pan, and zoom.

The case studies presented in this work demonstrate some characteristics of the data that are realized in its visualization. In particular, we point out some patterns that can be observed and questions that can be answered: "How much of my tree is complete?" "How am I related to this individual?" and "Is the ancestor related to the

individual through both parents?” Observed patterns include intermarriage, anomalies from biblical records and dense regions populated by royalty. A visual comparison of the proposed layout with existing layouts was given.

We have also conducted a user study in which users were able to observe the above-mentioned patterns. The study’s results, which give preliminary feedback on the usefulness of the proposed design, proved favorable as the H-tree turned out to be well-accepted by users. Although some users preferred existing layouts, most preferred the H-tree. Some felt the tasks were much easier when performed with the H-tree layout. Survey results show that the users liked the layout and felt it was useful. Associating an individual with his/her generation became slightly harder, but was exchanged for the ability to have a better understanding of the family structure. In particular, users liked how the dense parts of the graphs were easily visible. Most users felt that this would be a good addition to current genealogical software.

In the future, we are considering the use of color mapping to guide navigation. For example, using background colors to indicate tree depth helps the users to remember which branches of the data are deep. Another important functionality we are examining is how to improve the visualization of intermarriage and common ancestors, as other problems we are dealing with can greatly benefit from this improved scheme of visual analysis. It would also be nice to include this view in genealogical software, analyze its usage over time, and include expert users in the study. As far as the layout is concerned, it would be neat to see if a version of it where empty space is collapsed is useful.

Also, although a descendant tree is not binary, there are possible options to explore for representing descendants in this or a similar layout. One idea is to group the

descendants according to binary characteristics such as gender, or presence of a medical condition. This layout has potential for revealing some interesting patterns if they exist. For example, if descendants were grouped according to whether or not they had cancer, then interesting insights would be obtained if certain areas of the tree were denser than others.

Finally, we feel that the goals of this work have been fulfilled and that this layout has great potential for use within genealogical software. We also feel that using only this layout could be limiting, depending on the needs of the user, but that when combined with other layouts, it is useful and adds to the genealogical tools that we already have. Conclusively, we recommend this layout for use in genealogical software particularly when the data to be viewed consists of ancestral trees or subtrees that are densely populated and feel that it does a great job of presenting this data.

## **APPENDIX**

Here we analyze the amount of space needed to display an individual's name, birth, and death information using a 10-point font, assuming the smallest readable font to be 10-point. Based on our data, the respective information from an average record is representable using five 22-character lines. So, with the 10-point font, we use a 1.5-inch by 0.875-inch space. Assuming an 8 by 13 inch display with a resolution of 1,680 by 1,050, we are able to display approximately 129 pixels per inch. Thus, an individual needs 194 by 113 pixels to be displayed. Basically, by requiring the font to be readable, the physical space needed becomes fixed, and the number of pixels required to represent an individual depends on the resolution of the screen.

## REFERENCES

- [1] M. Bern and D. Eppstein. Worst-case bounds for subadditive geometric graphs. In *Proceedings of the 9th ACM Symposium on Computational Geometry*, pages 183–188, 1993.
- [2] S. A. Browning. *The Tree Machine: A Highly Concurrent Computing Environment*. PhD thesis, California Institute of Technology, 1980.
- [3] T. M. Chan. A near-linear area bound for drawing binary trees. *SODA*, pages 161–168, 1999.
- [4] The Church of Jesus Christ of Latter-day Saints. *Gedcom Standard Release 5.5.1*, October 1999.
- [5] C. S. Shin, S. K. Kim, and K. Y. Chwa. Area-efficient algorithms for straight-line tree drawings. *Computational Geometry*, 15(4):175–202, 2000.
- [6] G. M. Draper and R. F. Riesenfeld. Interactive fan charts: A space-saving technique for genealogical graph exploration. In *Proceedings of the 8th annual Family History Technology Workshop, Brigham Young University*, 2008.
- [7] FamilySearch. Personal ancestral file. <http://www.familysearch.org/paf/>.
- [8] F. Frati. Straight-line orthogonal drawings of binary and ternary trees. *Graph Drawing*, pages 76–87, 2007.
- [9] G. W. Furnas and J. Zacks. Multitrees: Enriching and reusing hierarchical structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '94)*, pages 330–336, 1994.
- [10] A. Garg. Straight-line drawings of binary trees with linear area and arbitrary aspect ratio. In *Journal of Graph Algorithms and Applications*, volume 8, pages 135–160, 2004.
- [11] W. Genes. The master genealogist. <http://www.whollygenes.com>.
- [12] GenoPro. Genopro. <http://www.genopro.com/>.

- [13] N. Henry. Improving readability of clustered social networks using node duplications. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1317–1324, 2008.
- [14] I. Herman, G. Melancon, and M. S. Marshall. Graph visualization and navigation in information visualization: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–43, January-March 2000.
- [15] B. Johnson and B. Shneiderman. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. *In Proceedings of the IEEE Visualization*, pages 284–291, 1991.
- [16] H. Koike and H. Yoshihara. Fractal approaches for visualizing huge hierarchies. *In Proceedings of the IEEE Symposium on Visual Languages*, pages 55–60, 1993.
- [17] J. Lamping, R. Rao, and P. Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '95)*, pages 401–408, 1995.
- [18] Leister. Reunion 9. <http://www.leisterpro.com>.
- [19] M. J. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. *In IEEE Symposium on Information Visualization*, pages 17–24, 2005.
- [20] M. J. McGuffin and J. M. Robert. Quantifying the space-efficiency of 2d graphical representations of trees. *In Information Visualization*, volume 9, pages 115–140, 2010.
- [21] Millennia. Legacy family tree. <http://www.legacyfamilytree.com>.
- [22] C. Muelder and K. L. Ma. Rapid graph layout using space filling curves. *In IEEE Transactions on Visualization and Computer Graphics*, volume 14:6, pages 1301–1308, 2008.
- [23] T. Munzner. H3: Laying out large directed graphs in hyperbolic space. *In Proceedings of the IEEE Symposium on Information Visualization*, pages 2–10, 1997.
- [24] N. W. Kim, S. K. Card, J. Heer. Tracing genealogical data with timenets. *In Advanced Visual Interface*, May 2010.
- [25] Q. V. Nguyen and M. L. Huang. A space-optimized tree visualization. *In Proceedings of the IEEE Symposium on Information Visualization*, pages 85–92, 2002.

- [26] P. Crescenzi, G. Di Battista, and A. Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Computational Geometry*, 2:187–200, 1992.
- [27] C. Plaisant, J. Grosjean, and B. B. Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 57–64, 2002.
- [28] J. Priestly. A chart of biography. *London: J. Johnson, St. Paul's Church Yard*, 1765.
- [29] RootsMagic. Rootsmagic. <http://www.rootsmagic.com>.
- [30] Y. Shiloach. *Arrangements of Planar Graphs on the Planar Lattice*. PhD thesis, Weizmann Institute for Science, 1976.
- [31] B. Shneiderman. Treemaps for space-constrained visualization of hierarchies. <http://www.cs.umd.edu/hcil/treemap-history>, 1998-2008.
- [32] Synium. Macfamilytree. <http://www.syniumsoftware.com/macfamilytree>.
- [33] T. Chan, M. T. Goodrich, S. R. Kosaraju, and R. Tamassia. Optimizing area and aspect ratio in straight-line orthogonal tree drawings. In *Computational Geometry*, volume 23, pages 153–162, 2002.
- [34] J. J. van Wijk and W. A. Nuij. Smooth and efficient zooming and panning. In *IEEE Symposium on Information Visualization*, pages 15–23, 2003.
- [35] J. Wesson, M. du Plessis, and C. Oosthuizen. A zoomtree interface for searching genealogical information. In *ACM AFRIGRAPH*, pages 131–136, 2004.