# *Exploring the PieTree for Representing Numerical Hierarchical Data*

**Richard O'Donnell, Alan Dix[1] and Linden J. Ball[2]**
*Computing Department[1] and Psychology Department[2]*
*Lancaster University, Lancaster, LA1 4WA, UK*
*http://www.hcibook.com/alan/papers/HCI2006-PieTree/*

**This paper describes the first full implementation and evaluation of an area-based tree visualization known as the PieTree. The PieTree was first proposed in papers in 1998 and 2000 but has never been fully implemented and evaluated. Informal evaluation was used to enhance the usability of the PieTree and compare it with the more well-known TreeMap. A controlled experiment considered parallel views' effect on task performance time. There were substantial differences between kinds of tasks and in participants' styles of use. Whilst suggesting that further development of PieTrees is worthwhile the experiments underline the importance of careful task fit.**

**Keywords:** PieTree, information visualisation, TreeMap, TreeView, hierarchies, hierarchy visualisation, individual difference.

## 1 Introduction

Hierarchical structures are common in computing systems, but in the Internet they are ubiquitous. Some are an intrinsic part of the technical infrastructure of the Internet and Web. Most obvious is the path hierarchy of HTML pages on a Website and at a lower level the Domain Name System (DNS), which stores the names of websites and maps them to specific IP addresses, is itself hierarchical. Usenet newsgroups feature a similar hierarchical structure, with groups either existing in isolation, or having related child groups, for example ‘24hoursupport.helpdesk’ exists by itself, whereas ‘alt.activism.death-penalty’ is a child newsgroup of ‘alt.activism’. Other hierarchies are created within specific websites: directory structures of information sites such as Yahoo! or Open Directory project, category structures of content management systems or product categories on eCommerce web sites such as Amazon. In browsers and email systems there are yet more hierarchies with bookmark folders and mail folders.

The primary aim of these hierarchies is often to organise textual or graphical information so that it can be accessed easily. For example an e-commerce site may have high-level categories, such as books, DVDs, toys etc. as well as specific pages for items in that category, such as a certain book. Some visitors may find items by browsing through the categories for the site and others may find a specific page by entering the site through an Internet search engine.

Often also there is some sort of numerical data associated with the hierarchy: number of pages within a folder, number of hits on a web page, volume or value of sales within a product category. Questions about these numbers are often critical for site owners: "what are the hot topics on my site?" or "what areas are being missed by visitors?"

Numerical hierarchical data exists in other areas: for example the size of files on a disk, the stock levels or sales of different kinds of goods in a conventional inventory system, and similar questions can be raised about such data: "where has all my hard disk space gone?", "what is selling well?" Not surprisingly there have been a number of visualisations aimed at this kind of data, perhaps most well known being the TreeMap [Johnson & Shneiderman 1991].

The importance of numerical hierarchical data is perhaps emphasised by the fact that the TreeMaps have been incorporated into several commercial applications. TreeMaps have also been included by Microsoft Research as part of their data visualisation components (see Figure 1), and they have been used to visualise data generated from Usenet newsgroups [Fiore & Smith 2001].

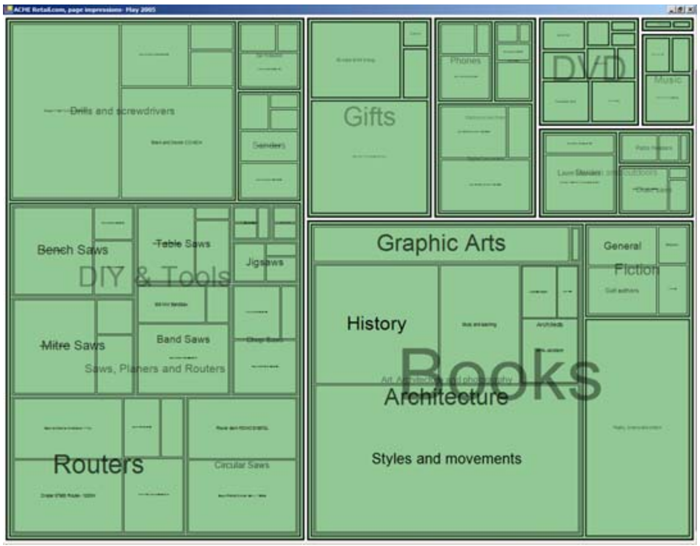


**Figure 1: Microsoft Research's TreeMap showing data for a fictional e-commerce site**

In this paper we discuss another novel visualisation technique known as a PieTree [Dix et al. 1998, 2000] (Figure 2), which like a TreeMap can be used to represent hierarchical numerical data. Like the TreeMap the PieTree maps count/size directly onto area, but instead adopts a circular layout exploiting users' familiarity with Pie diagrams. Earlier work discussed the concept of the PieTree and a partial implementation. In this paper we discuss both informal evaluation of a more fully featured prototype and compare it with Microsoft Research's implementation of the TreeMap. We also report a controlled experiment that investigates the benefits of using the PieTree in conjunction with a TreeView (the outliner style view of trees used in standard file browsers) to generate a parallel view of the hierarchy.

**Figure 2: A fully exploded PieTree representing a hierarchical data structure**

## 2 Related work

There are a large number of visualizations focused on simple hierarchies.

Cone Trees [Robertson et al. 1991] are one of the most well known. They present data using a top to bottom hierarchical approach that utilizes 3D techniques to display each level of nodes like a fairground Ferris wheel on its side. They achieve both focus+context by displaying nodes of interest in the foreground (focus) and the rest of the hierarchy (the context) in the background. Distorted nodes can be brought into focus by rotating the tree.

The hyperbolic browser [Lamping & Rao 1994] is another well-known technique for visualizing hierarchical trees. In this representation, nodes are positioned in a hyperbolic plane, a distortion technique that allows nodes in focus to be stretched, and nodes in context to be squeezed [Pirolli et al. 2000]. This allows the hierarchy to be drawn in a space efficient manner [Stasko & Zhang 2000].

Information Slices [Andrews & Heidegger 1998], uses concentric semi-circular discs to visualise large trees by using each disk to represent a number (usually 5–10) of levels of the hierarchy. They also include a way of focusing on parts of interest by spawning a fresh disk from a selected node.

Despite research into visualizing hierarchies, visualizations such as the hyperbolic browser and Cone Trees have still failed to make a commercial impact outside niche areas in industry and most graphical user interface toolkits only feature the ubiquitous TreeView for visualizing hierarchies. TreeViews, simple outliner-style lists of folders, are found in many applications such as Windows

Explorer. Due to their popularity they are familiar to most users, making them a viable option in applications because users already know how to use them [Jacobsson 2002].

A number of visualizations have also been developed that allow you to visualize hierarchies with values attached such as file sizes, web hits or sales.

The most well known of these is the TreeMap [Johnson & Shneiderman 1991], like the PieTree, it can display hierarchies where both leaf and child nodes contain numerical values. TreeMaps adopt a rectangle space filling approach. They work by slicing and dicing rectangles to create child items. This slicing and dicing is performed recursively (by slicing up child rectangles) until all child items are represented inside each parent rectangle on a TreeMap.

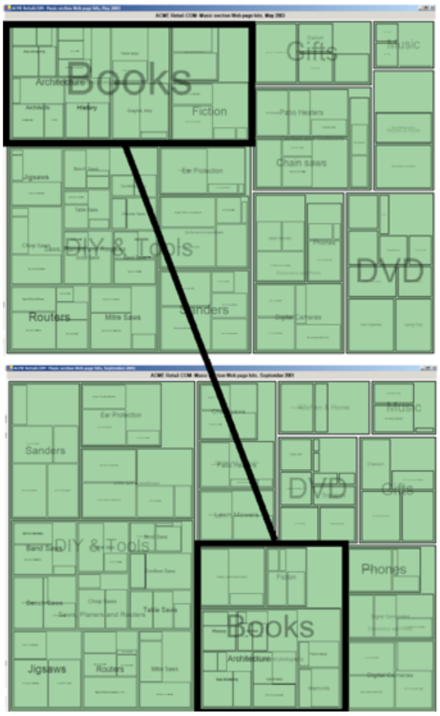


**Figure 3: TreeMap item 'Books' changing position when the values change**

Over the years various improvements have been made to the TreeMap visualization [Wattenberg 1999; van Wilk & van De Wetering 1999; Bruls etr al. 2000; Shneiderman & Wattenberg, 2001; Bederson et al. 2002] but problems still remain, the biggest two being that the positions of the rectangles in a given hierarchy are not guaranteed to stay in the same position when the items in the hierarchy change value (Figure 3) and the smallest node in the TreeMap may not be represented by the smallest rectangle displayed, due to spacing enhancements suggested by Fiore and Smith [2001] that allowed users to better understand the hierarchy that the TreeMap represents.

DiskTrees [Chi et al. 1998], like PieTrees, use a circular layout to visualise web site evolution. In DiskTrees angular space is allocated proportional to the number of leaf nodes, whilst page access counts are visualized by the thickness and brightness of lines.

Evaluation in visualisation has always been problematic and often lags behind new ideas. The CHI'97 browse-off was perhaps one of the most high-profile, if informal, comparisons where expert users of the Hyberbolic Browser were able to out perform the plain Tree View [Mullet et al. 1997]. In more formal evaluation, Cockburn and McKenzie [2000] found that locating items in a Cone Tree was slower than locating items in a TreeView, and Kobsa [2004] found that most hierarchical representations (including TreeMaps and the hyperbolic browser) did not perform as well as the TreeView in Windows Explorer for tasks related to file system management. Cone Trees suffer from considerable occlusion due to overlapping nodes [Spence 2001; Card et al. 1999], and empirical work has shown that Cone Trees become too difficult to comprehend when there are more than 1000 items in them [Carriere & Kazman, 1995]. Czerwinkski and Larson [1997] found that whilst the hyperbolic browser was better than the TreeView for keeping global/local focus, the TreeView was better for tracking where you had been and was more familiar to users and that they preferred using TreeViews to hyperbolic trees.

## 3 PieTree external behaviour

The basic concept of PieTrees was introduced in Dix and Ellis [1998] as an example of the principle of taking a standard visualisation (in this case a Pie chart) and adding interactivity. This was developed more fully (and the name PieTree used) together with a partial demonstrator-level implementation in Dix, Beale and Wood [2000]. PieTrees are seen as normal pie charts when collapsed. However nodes that contain child values can be exploded to reveal the child nodes of that segment. A collapsed PieTree segment represents the sum of the values for a category node + the sum of the values for all its children. An exploded PieTree category segment represents the true value of a category node, and has its child nodes shown outside it (Figure 4). The area of each PieSegment is proportional to the total value of the PieTree. Child values can also be categories and contain child nodes that can also be expanded. This allows the PieTree to display deep hierarchies. Note that in the case where the top level node has a value it is represented as a small circle of appropriate area at the centre (for example, hits on the home page of a web site).

The PieTree allows users to expand and collapse nodes in any order, either the whole tree can be expanded, or just specific nodes. In complex hierarchies users require mechanisms for hiding parts in order to focus on a particular sub hierarchy [Robertson et al. 1991], a requirement met by the PieTree.

Whilst the PieTree can be used in isolation, it was envisaged to be used alongside an outliner-style TreeView so that the two views compliment each other and the structure of the PieTree is made more apparent because of simultaneous folding/unfolding of both – a form of *temporal fusion*. The PieTree is isotropic in nature, so will always have the same width and height regardless of what area it is displayed in on a computer screen. This feature makes the PieTree fit well alongside a standard TreeView control as horizontal screen resolution is usually significantly greater than vertical resolution.

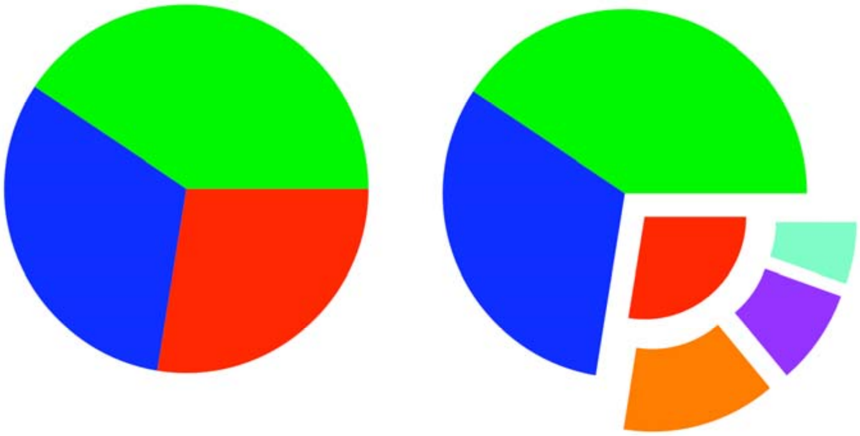


**Figure 4: Left- PieTree appearing as a pie chart (collapsed). Right- Exploded PieTree showing child nodes and value of category node.**

## 4 Informal evaluation

To enhance the design of the PieTree visualization, two rounds of informal evaluation were conducted in an iterative process which allowed important usability problems to be discovered. Usability problems discovered in the first round of evaluation were then re-evaluated in the second round. Problems discovered in the second round were then fed back into the PieTree design used for the controlled experiment.

The cooperative evaluation technique was used instead of the standard think-aloud observation technique. This technique, unlike think-aloud, allows participants to ask questions if they become stuck and additionally allows the evaluator to ask questions to the user if it is not immediately obvious what they are thinking [Dix et al. 2004].

### *4.1 Participants*

Eight postgraduate students at Lancaster University volunteered for the informal evaluation. None of the participants were paid for the study. All of the participants had normal or corrected to normal vision. The typical age range was 20-25, with an overall age range of 20-35. None of the participants used in the informal evaluation participated in the controlled experiment.

Participants reported that on average they had a slightly above average knowledge of mathematics and trigonometry. All of the participants reported that they used computers daily. Three quarters of the participants reported using spreadsheets on a monthly or weekly basis, one used them on a daily basis, and one had never used spreadsheets.

## *4.2 Design*

The evaluation followed a 2 x 2 design with each participant using both PieTrees and TreeMaps for both simple and more complex hierarchies.

The first and third conditions used the PieTree visualisation in conjunction with a TreeView to display the hierarchy and the second and fourth used the Microsoft Research TreeMap.

In the first two conditions a simple hierarchy was used containing 3 parent nodes, and 3 child nodes for each parent, each with 12 items. For the third and fourth conditions a more complex hierarchy was used which contained 125 nodes and a non-uniform hierarchical structure with a maximum depth of four child nodes deep. All data was fictional but given realistic names. To make the hierarchies in the third and fourth conditions even harder to understand, a much greater range of values was applied to the items than in tasks one and two.

The conditions were partially counterbalanced to allow for any learning effects. This spread the order in which the visualisations were displayed to the participants and also spread the order in which complex and simple hierarchies were presented.

The participants were not given explicit instruction in the interaction with either visualisation as we were interested in the extent to which the interfaces could be picked up by novice or infrequent users. Participants were asked to verbalise their thoughts throughout the experiment. In order to make them comfortable with using this technique they were asked to read the instructions aloud. There was no time limit given for any of the tasks, but the procedure typically took 45 minutes.

## *4.3 Tasks*

For the first two conditions using simple hierarchical data, participants were given the following tasks:

- Find the item with the largest value and specify:
    - What the item is.
    - What the item represents.
- Find the item with the smallest value and specify:
    - What the item is.
    - What the item represents.
- Specify what the data in general represents, and whether or not the items relate to each other.

The tasks deliberately did not define how to find the largest and smallest values; this was left to the participants own intuition. Note that for the first of these tasks both TreeMaps and PieTrees allow the option of efficient heuristic search whilst the second always requires every node to be visited. Both can be accomplished by simply looking at numbers using tool tips, but may, in principle, be aided by visually assessing the size of the targets.

These tasks were deliberately designed not to favour the use of PieTree visualisation, but covered a range of sub-tasks one might encounter when interacting in a more exploratory fashion with numerical hierarchical data.

These same three kinds of task were used in the later formal experiment.

For the third and fourth conditions users were also given the task of being allowed to explore the data and verbalise their actions and thoughts as they explored. Users were then given the tasks of finding specific items in the hierarchy that were expected to cause them difficultly (items that were deep down the hierarchy and had very small values). For both conditions users were given items to find which had the numerical value of 1 out of 5282 (less than 1 / 5000 of the visualisation). For the PieTree the task given was 'find and highlight the book: What You Wear Can Change Your Life'. In the case of the TreeMap, the task was to 'find and highlight the camera Canon Digital Ixus 40 4 megapixel'.

## 4.4 Apparatus

The informal evaluation was performed on an IBM Thinkpad R31 laptop computer running Windows XP Professional, with a 14 inch display at a resolution of 1280 x 1024 pixels. A Microsoft USB wheel mouse was attached to the laptop, as it offered more precision than the inbuilt TrackPoint. Participants' voices were recorded using an Olympus VN-3600 Digital Voice Recorder.

## 4.5 Results and Interpretation

Participants did not show any significant preference between the PieTree and the TreeMap. However participants suggested a number of changes to the visualisations, including highlighting the items selected using the TreeView in the PieTree representation.

The major problem with the PieTree was that users believed it was just a normal pie chart:

*"OK. Right. A pie chart."*

All participants incorrectly specified the smallest item in the PieTree as being the smallest item that appeared in its collapsed state. Because users thought the PieTree was a pie chart some of them failed to expand the PieTree at all on one of the tasks (they were not specifically instructed to do so):

*"OK. DIY and tools, how many page hits? Oh, that includes child items. Oh child items, never thought to do that (expand the items PieTree), that's really cool. Can I go back to other one (task one) and have a look for child items there?"*

It quickly became apparent during the first round of evaluation that users expected a greater degree of interactivity between the PieTree and the TreeView, highlighting fine interaction details that lead into re-design. Participants' actions suggested that the TreeView brought benefits to the PieTree, reinforcing the expected pattern of use in the early PieTree papers.

The different aspects ratios of the TreeMap (Figure 5) proved to be a burden when asked to find the smallest item, with participants finding it harder to figure out whether Beans were smaller than Bananas. One of the squarified versions of TreeMaps might have helped this.

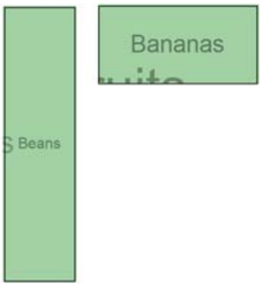


**Figure 5: TreeMap items with different aspect ratios**
**(N.B. partial letters from legends of higher level categories)**

Users generally displayed a poor understanding of the hierarchies displayed in the visualizations. This may have been due to the datasets not being familiar to the participants; however some problems seemed more fundamental. In the instructions they struggled on 'techie' terms like parent/child. In the PieTree as noted previously they often failed to realise that there was a hierarchy to expand and in the TreeView they found difficulty distinguishing labels of categories from those of contained sub-categories.

## 5 Formal evaluation

In order to investigate the benefits of using parallel views in the PieTree visualisation, an experiment was performed to see whether using a TreeView in conjunction with a PieTree brought any additional benefits over using just a PieTree or TreeView.

### *5.1 Participants*

A total of 16 students undertook the experiment, comprising of 15 postgraduates and 1 undergraduate student all at Lancaster University (6 females and 10 males). The average age for participants was 20-25 years old. All users had normal or corrected to normal vision. None of the participants were paid for the study nor made aware of the study hypotheses.

### *5.2 Design*

A 3 x 3 within-subjects design was used. The first factor was representation (PieTree and TreeView, PieTree only and TreeView only) and the second factor was task (find the smallest node, find the largest node, find a specific node).

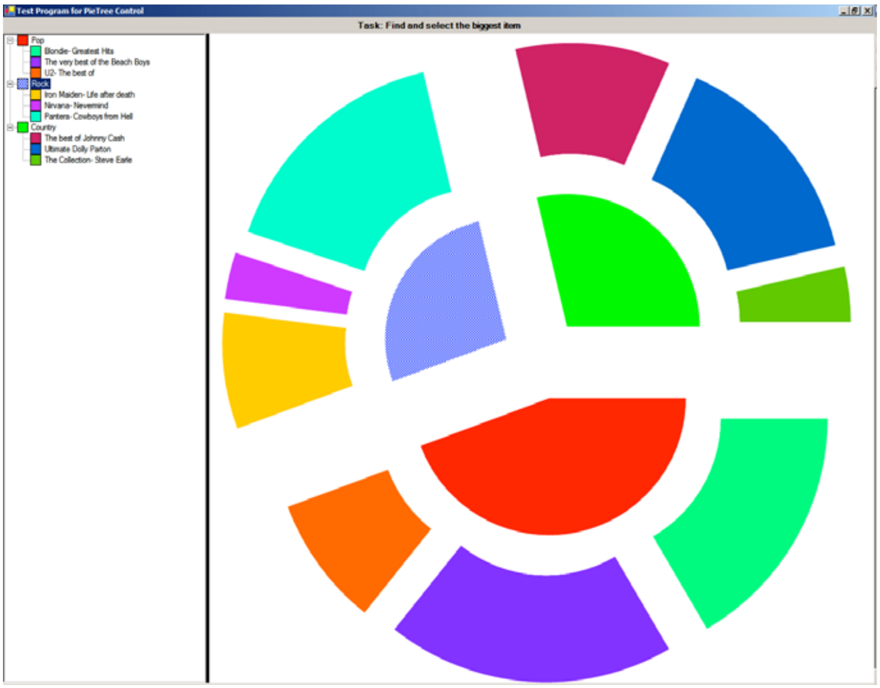
The dependent variables measured in the experiment included task completion time (measured in milliseconds) and the number of correct responses for each condition. There were three trials for each condition in the experiment, creating a total of 27 trials. The trial order was independently randomised for each participant using a pseudorandom number generator.

## *5.3 Apparatus and Materials*

The visualizations were displayed on a Pentium 4 desktop computer running Windows XP Professional with a 17" LG L1715S monitor at a display resolution of 1280 x 1024 pixels. A Microsoft USB optical mouse was used as the input device for the experiment; use of the keyboard was not permitted.

The experimental software was written by the authors to record the task completion time for the stimulus and the item chosen by the participant. The item chosen by the user was also recorded so as to enable analysis of the error totals.

The three visualizations shown to the participants included the PieTree in conjunction with a parallel TreeView (Figure 6), the PieTree in isolation, and the TreeView only.



Task: Find and select the biggest item

**Figure 6: The PieTree in conjunction with a parallel TreeView**

The hierarchy used in the experiment was a hierarchy for a fictional e-commerce store. For each of the 27 trials in the experiment, the value for each tree item was randomly generated between the values of 200 and 800.

## *5.4 Procedure*

The experiment was conducted using one participant at a time in a quiet room. Participants were given instructions on a piece of paper about the tasks they had to perform. After reading the instructions and signing the consent form, they were then asked to watch a five minute instruction video walking through the analysis of the data shown on a sample PieTree for a fictional company called StallMart.

Participants were instructed to complete the tasks as quickly and as accurately as possible. The participants were not told that the area of the segments of the PieTree corresponds to their value, nor were they informed of any strategies that will make the task completion time quicker. They were informed that the values of the data change for each trial.

A post experiment informal interview was conducted with participants in order to establish which visualization they preferred and why, and to discuss and clarify any strategies they appeared to develop over the duration of the experiment. The entire procedure took approximately 20 minutes per participant.

### 5.5 Results and Interpretation

Over all tasks, the average completion time for the PieTree was nearly 10% faster than the TreeView with the combined visualisation between the two. However, this difference is not statistically significant and moreover hides a richer story of individual differences and task interactions.

During the experiment it was observed that different participants adopted different strategies. There were three distinct types of participant: (i) those that relied solely on tooltips when finding the largest and smallest values, (ii) those that used both visualizations where the stimulus allowed and (iii) those that discovered using the areas on the PieTree aided in finding the largest and smallest items mid experiment. These three participant types were coded and used in the analysis.

It was noted that the participants who adopted a strategy of using the area of the Pie Segments (types (ii) and (iii)) appeared to perform better than those relying solely on tooltips (type (i)). Because of the wide variation between individuals we analysed the difference using non-parametric rank statistics as these are more robust to outliers. A Mann-Whitney U test confirmed that there was indeed a significant difference between the rankings of users that utilized the area of the Pie Segments in finding the largest and smallest items, than those that did not ($U = 6$, $N_1 = 9$, $N_2 = 7$, $p = 0.005$, two tailed).

It was observed that making comparisons between segments in the PieTree became difficult when there were several similarly sized segments in the visualization. This was also commented on by participants:

*“I found comparing data from different levels to be complex”*

*“Pie in the middle is hard to compare- it is like comparing a triangle to a rectangle”*

For angle segments, just as for rectangles with different aspect ratios in Figure 5, it is hard to compare areas. This is a problem for plain Pie charts and arguably these are not a good representation for this reason. However, perhaps just because of familiarity they are an accepted and largely understood visualization and it is on this existing understanding that PieTrees build.

**Table 1: Task completion times for task and representation in seconds (standard deviation displayed in parenthesis)**

| **Task** | **Representation** | | | |
|---|---|---|---|---|
| | Both | Pie only | Tree only | *Mean* |
| Biggest | 18.595<br>(5.62) | 16.210<br>(4.63) | 21.590<br>(8.92) | *18.798* |
| Smallest | 16.777<br>(5.23) | 16.236<br>(5.61) | 17.987<br>(5.44) | *17.000* |
| Specific | 9.436<br>(4.89) | 11.444<br>(5.41) | 8.154<br>(3.00) | *9.678* |
| *Mean* | *14.936* | *14.63* | *15.910* | |

As well as the substantial individual differences there were also significant differences between tasks. Table 1 shows task completion times for the three conditions and three tasks. The interaction effect is significant (ANOVA 3 x3 x 16, F(4,60) = 4.939, p = 0.02). This is not unexpected. The 'find the biggest' tasks permit rapid visual inspection that can lead to an efficient heuristic search and indeed the PieTree substantially reduces completion times for these tasks (with the presence of the additional TreeView appearing to act as a distraction). The 'find the smallest' tasks require looking at everything, but rapid visual scanning can cut down the possibilities needing to be considered; in these tasks the PieTree still enables faster performance, but more marginally so. However, in the tasks related to finding information about specific values, there is no advantage to size-based scanning and hence, reasonably, the simpler TreeView out performs the PieTree. Again for these last tasks the presence of the PieTree appears to distract from the more efficient use of the TreeView.

We do not have eye tracking or similar data to verify the search strategies used, so the explanations of these results are tentative, however, participants did note the differences between tasks:

*“The PieTree was useful for finding the largest and the smallest- it was easy to do based on the area”*

*“The more sizes tended to be similar the less useful was the Pie. But typically the Pie was more useful for finding the largest/smallest items”*

*“PieTree was useless for finding names”*

*“Tree is better for finding a specific item”*

## 6 Future Developments for PieTree

Like all novel visualisation techniques, the PieTree visualisation is not without its problems and extra research is necessary to further enhance the visualisation. TreeViews become problematic when displaying large hierarchies as users are required to perform a large amount of scrolling. Robertson et al. [2005] suggest using coalesced nodes in TreeViews to overcome this problem, allowing the hierarchy to be visualised in a much smaller space. This in turn allows nodes to be located faster. In future implementations it would be possible to combine the Robertson et al. coalesced TreeView with the PieTree to further enhance its performance when visualising large hierarchies.

Large hierarchies also cause more fine-grained interaction issues as the nodes towards the leaves become unclickable on larger trees. Perhaps some form of zooming into a part of the tree opening it as a fresh PieTree would be appropriate, similar to that used in Information Slices.

Colouring of segments is also an interesting issue. In principle a planar diagram like the PieTree can be coloured in four colours whilst never having two adjacent areas in the same colour (the famous Four Colour theorem). In practice the algorithms for doing this are quite expensive, but allowing a slightly larger palette makes this computationally tractable. However, nodes in the PieTree expand and collapse and if node colouring is to be preserved during interaction (it would be very confusing not to!), then it is neither clear how many colours are necessary, nor what an efficient algorithm for colouring would be.

The Gestalt law of connectedness, states connected objects are perceived as a single structure [Nesbitt & Friedrich 2002] and allows for automatic perceptual grouping [Palmer & Rock 1994]. This Gestalt law is exploited in many hierarchy visualisations including TreeViews, directed graphs [Knuth 1968; Gansner et al. 1988], Cone Trees [Robertson et al. 1991] and hyperbolic browsers [Lamping & Rao 1994]. This law could be exploited better in the PieTree visualisation to help show groupings between nodes, as it has been shown that connectedness is a more powerful grouping principle than proximity, size, colour or shape [Ware 2004]. The Gestalt law of common fate, states that objects moving in the same direction can be perceived as a group [Martinez-Trujillo & Treue 2004]. The PieTree could again exploit this law when it expands by animating exploding pie segments.

## 7 Discussion and Broader Issues

As noted the PieTree has previously only had partial implementation and no evaluation. This first implementation as a component within a standard toolkit and evaluation is thus important in assessing the value of the technique.

The informal evaluation compared this first PieTree implementation against the TreeMap, which has had extensive development over many years, and hence the lack of a clear 'winner' between the two suggests that further development and tuning of the PieTree is worthwhile. The specific problems highlighted in the Microsoft implementation of the TreeMap are largely ones that have been addressed in variants of the TreeMap, although problems identifying hierarchy are perhaps more fundamental as this is a bordered version of the TreeMap which should make this easier.

For the PieTree this was more a formative evaluation stage, however, beyond the comments that lead to incremental changes, the fact that users were slow to discover the interactive capabilities of the visualization is interesting. This is probably because it is rare to encounter interactive visualizations even in computer applications or the web. The familiarity of the PieTree was a two-edged sword whilst making the visualization easier to read, it led to problems where users saw it as 'just' a Pie chart and thus even less likely to explore than perhaps a more unusual visualization might have encouraged.

In the formal evaluation the PieTree did perform better on average than the more familiar TreeView. However, the main lesson we wish to draw (as in so much of interactive systems design) is 'it all depends' – upon user and upon task.

The substantial difference not just in users' 'baseline' speeds but, more critically, in the interaction between visual/cognitive style and visualisation strategy is of great importance. It is in many ways 'obvious', but does suggest that effective pre-tests for these styles should perhaps be standard procedure in order to interpret visualisation evaluation. This underlines the well known, but often overlooked, importance of providing alternative ways of visualising and interacting with data. In future experiments in this area it would be good to consider ways of detecting the users' strategies in solving tasks. Eye tracking would certainly help, although analysing the data from this on a dynamic visualisation would be very labour intensive or require bespoke analysis software. Also given the level of individual difference it would be good to pre-test participants using standard test of visual / spatial ability, cognitive style etc.

The task interaction is again not surprising, but also so important methodologically. It is not uncommon to see experiments where generalisations are based on a small set of kinds of task … and in the case of novel visualisations ones where you would expect the new visualisation to perform well! We deliberately chose a range of tasks where we would expect good and poor behaviour and this was evident in the outputs. One size does not fit all and visualisations need to fit the tasks for which they are being used.

In any particular task, the hybrid TreeView+PieTree solution was out performed by one of the other methods. It appears that more is not better; users presented with a combined visualisation, even with no competition for screen space, were not able to focus on the alternative best for the current task. Possibly this would be different for longitudinal or expert use, but within the timescale of a short controlled experiment participants were not able to learn effective decision strategies. Indeed a simple experiment using any one of the tasks might have led to a rejection of the hybrid approach, whereas in cases where there is a mix of tasks it is clearly the solution to give the most consistent performance.

In summary, the success of PieTrees on appropriate tasks suggests that further work in this area would be fruitful, especially given the relative novelty of the technique. The particular combination of TreeView and PieTree whilst non-optimal in any particular situation appears to be more robust over a range of tasks.

## 8 Acknowledgements

This work was supported by an EPSRC MTA studentship on the MRes in Design and Evaluation of Advanced Interactive Systems and Lancaster University.

## References

Andrews, K. & Heidegger, H. [1998]: Information slices: visualising and exploring large hierarchies using cascading, semicircular discs. *Proc. IEEE Information Visualization Symposium*, Carolina, USA, 9–12.

Bederson, B. B., Shneiderman, B. & Wattenberg, M. [2002]. Ordered and quantum treemaps: Making effective use of 2D space to display hierarchies. *ACM Transactions of graphics 21* (4). ACM Press, 833–854.

Bruls, M., Huizing, K. & van Wijk, J. J. [2000]. Squarified TreeMaps. *Joint Eurographics and IEEE TCVG Symposium on Visualization.* 33–42.

Card, S. K., Mackinlay, J., C. & Shneiderman, B. [1999]. *Readings in information visualisation. Using vision to think.* Morgan Kaufmann.

Carriere, J. & Kazman, R. [1995]. Interacting with huge hierarchies: Beyond Cone Trees. *IEEE Symposium on Information Visualization.* 74–78.

Chi, E. H., Pitkow, J., Mackinlay, J., Pirolli, P., Gossweiler, R. & Card, S. K. (1998). Visualizing the Evolution of Web Ecologies. *ACM Conference on Human Factors in Software (CHI '98)*, Los Angeles, 400-407.

Cockburn, A. & McKenzie, B. [2000]. An evaluation of cone trees. *People and computer XV.* Proceedings of HCI'2000. Berlin: Springer Verlag. 425–436

Czerwinkski, M. & Larson, K. [1997]. The new Web browsers: they’re cool but are they useful. Thimbeby, H., O’Conaill, B and Thomas, P (Eds.). *People and computer XII.* Proceedings of HCI ’97. Berlin: Springer Verlag.

Dix, A & Ellis, G. [1998]. Starting Simple - adding value to static visualisation through simple interaction. *Proceedings of Advanced Visual Interfaces* - AVI98, Eds. T. Catarci, M. F. Costabile, G. Santucci and L. Tarantino. L'Aquila, Italy, ACM Press. 124-134.

Dix, A. Beale, R. & Wood, A. [2000]. Architectures to make Simple Visualisations using Simple Systems. *The working conference on advanced visual interfaces.* ACM Press, 51-60.

Dix, A., Finlay, J., Abowd, G. D., & Beale, R. [2004]. *Human Computer Interaction (3rd Edn).* Prentice Hall.

Fiore, A. & Smith, M., A. [2001], TreeMap Visualizations of Newsgroups. *Microsoft Research Technical Report.* Redmond: Microsoft Research, Microsoft Corporation.

Gansner, E. R., North, S. C. & Vo, K. P. [1998]. Dag- a program that draws directed graphs. *Software Practise and experience 18* (11) : 1047–1062.

Jacobsson, P. [2002]. Using a TreeView metaphor to visualise hardware simulation for testing. *Human Factors in Computing Systems.*

Johnson, B. & Shneiderman, B. [1991]. Treemaps a space filling approach to the visualization of hierarchical information structures. *Proceedings of IEEE Visualization'91, IEEE.* 275-282.

Knuth, D. E. [1968]. *The Art of Computer Programming- Fundamental Algorithms (volume 1).* Addison-Wesley.

Kobsa, A. [2004]. User Experiments with Tree Visualization Systems. *The IEEE Symposium on Information Visualisation. INFOVIS 2004.* 9–16.

Lamping, J., & Rao, R. [1994]. A focus+context technique based on hyperbolic geometry for visualising large hierarchies. *The SIGCHI conference on human factors in computing systems. CHI94.* ACM Press. 401–408.

Martinez-Trujillo, J. C., & Treuse, S. [2004]. Feature-Based attention increases the selectivity of population responses in primate visual cortex. *Current Biology* 14, 744–751.

Mullet, K., Fry, C. & Schiano, D. [1997]. On your marks, get set, browse! (The Great CHI'97 Browse Off), Proceedings of the ACM CHI'97 Conference on Human Factors in Computing Systems, 1997 (March 22-27), Atlanta, Georgia. (see also http://www.baychi.org/calendar/19970812/ for a brief report).

Nesbitt, K. V., & Friedrich, C. [2002]. Applying Gestalt Principles to Animated Visualizations of Network Data. *IV 2002.* IEEE. pp. 737–143

Palmer, S. & Rock, I. [1994]. Rethinking perceptual organization. The role of uniform connectness. *Psychonomic bulletin and review* 1(1), 29–55.

Pirolli, P., Card, S. K., & Van Der Wege, M. M. [2000]. The effects of information scent on searching information: visualizations of large tree structures. *Proceedings of the working conference on Advanced Visual Interfaces, AVI2000.* 161–172

Robertson, G. G, Czerwinkski, M. P, & Churchill, J. E. [2005]. Visualisation of mappings between schemas. *The SIGCHI conference on human factors in computing systems. CHI2005*. ACM Press. 431–439

Robertson, G. G., Mackinlay, J.D., & Card S. K. [1991]. Cone Trees: animated 3D visualisations of hierarchical information. *The SIGCHI conference on human factors in computing systems. CHI91.* ACM Press. 189–194

Shneiderman, B. & Wattenberg, M. [2001]. Ordered TreeMap Layouts. *IEEE Information Visualisation. INFOVIS'01.* IEEE. 73–78

Spence, R. [2001]. *Information Visualisation*. ACM Press.

Stasko, J., & Zhang E. [2000]. Focus+Context Display and navigation techniques for enhancing radial, space-filling hierarchy visualisations. *Symposium on information visualisation.* IEEE. 9–12

van Wijk, J. J. & van de Wetering, H. [1999]. Cushion Treemaps: Visualization of Hierarchical Information. *IEEE Symposium on Information Visualization. INFOVIS99.* IEEE. 73–78

Ware, C. [2004]. *Information Visualization (2nd Edn). Perception for design.* Morgan Kaufmann.

Wattenberg, M. [1999]. Visualizing the stock market. *CHI ’99 extended abstracts on human factors in computing systems*. ACM Press. 188–189