

Elsevier required licence: © 2018. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
<http://creativecommons.org/licenses/by-nc-nd/4.0/>

Accepted Manuscript

Enabling Effective Tree Exploration Using Visual Cues

Quang Vinh Nguyen , David Arness , Carrissa J. Sanderson ,
Simeon Simoff , Mao Lin Huang

PII: S1045-926X(16)30009-X
DOI: [10.1016/j.jvlc.2018.06.001](https://doi.org/10.1016/j.jvlc.2018.06.001)
Reference: YJVLC 837



To appear in: *Journal of Visual Languages and Computing*

Received date: 20 January 2016
Revised date: 25 January 2018
Accepted date: 11 June 2018

Please cite this article as: Quang Vinh Nguyen , David Arness , Carrissa J. Sanderson , Simeon Simoff , Mao Lin Huang , Enabling Effective Tree Exploration Using Visual Cues, *Journal of Visual Languages and Computing* (2018), doi: [10.1016/j.jvlc.2018.06.001](https://doi.org/10.1016/j.jvlc.2018.06.001)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Enabling Effective Tree Exploration Using Visual Cues

Quang Vinh Nguyen¹, David Arness², Carrissa J. Sanderson², Simeon Simoff¹, and Mao Lin Huang³

¹*MARCS Institute and School of Computing, Engineering and Mathematics, Western Sydney University*

²*School of Social Sciences and Psychology, University of Western Sydney*

³*School of Software, Faculty of Engineering & IT, University of Technology, Sydney*

E-mail: Q.Nguyen@westernsydney.edu.au, D.Arness@westernsydney.edu.au,

Carrissasanderson@gmail.com, S.Simoff@westernsydney.edu.au, and Mao.Huang@uts.edu.au

Corresponding author: Quang Vinh Nguyen, E-mail: Q.Nguyen@westernsydney.edu.au

School of Computing, Engineering and Mathematics

University of Western Sydney

Locked Bag 1797, South Penrith DC, NSW 1797, Australia

Abstract

This article presents a new interactive visualization for exploring large hierarchical structures by providing visual cues on a node link tree visualization. Our technique provides topological previews of hidden substructures with three types of visual cues including *simple cues*, *tree cues* and *treemap cues*. We demonstrate the visual cues on Degree-of-Interest Tree (DOITree) due to its familiar mapping, its capability of providing multiple focused nodes, and its dynamic rescaling of substructures to fit the available space. We conducted a usability study with 28 participants that measured completion time and accuracy across five different topology search tasks. The *simple cues* had the fastest completion time across three of the node identification tasks. The *treemap cues* had the highest rate of correct answers on four of the five tasks, although only reaching statistical significance for two of these. As predicted, user ratings demonstrated a preference for the easy to understand *tree cues* followed by the *simple cue*, despite this not consistently reflected in performance results.

Keywords

Interactive visualization; tree visualization; visual cue; DOITree; topological preview.

1. Introduction

Hierarchical structures are observed in many forms, such as the organizational structure of a file system, the structure of a classification system, and the organization structure and taxonomy of objects, such as animals, plants, airplanes, etc. Such hierarchical structures not only play significant roles in their own right, but also provide means for representing the structure of a complex domain in a simplified form [1]. With the rapid growth of data, a hierarchical data set often contains thousands or even millions of items. Interactive visualization, with capability for deep exploration at different levels of granularity, is crucial for analysts in the knowledge discovery process. Such visualization can potentially provide benefits on viewing and exploring complex hierarchical information as well as visually analyzing key information hidden in the structures.

Existing visualization techniques for two-dimensional hierarchical structures (or trees) can be classified as connection, enclosure or hybrid [2]. Effectiveness of each approach is primarily evaluated in terms of the properties of the data in specific domains of application. The connection approach, such as *Classical Hierarchical View* [3], *Radial View* [4], *Balloon View* [5], *Rings* [6], *Space-Tree* [7], and *Hyperbolic Browser* [8], displays the relationships explicitly using node-link diagrams. Enclosed or space-filling approaches, such as *Treemaps* [9-11] and *Voronoi Treemaps* [12], are usually more effective when being applied to data sets to illustrate attributed properties. Hybrid techniques, such as *Space-Optimized Tree* [13] and *EncCon Tree* [14], combine both enclosure and connection in their visualization. Such hybrid methods are also applied to other visualization types such as social networks [33]. The traditional algorithms of tree layouts were summarized in Di Battista et al [15] and Herman et al [16].

Classical hierarchical views are universally adopted by various users and applications due to their simple and familiar layouts. Techniques were typically developed based on the original Reingold & Tilford's algorithm [3]. They utilize a modular approach to positioning nodes where child nodes are positioned below their ancestor for commonly top-down orientation, or on the right-side for left-to-right orientation. The Reingold & Tilford's algorithm were enhanced by Kennedy [16], Herman et al. [17], and Bruggenmann-Klein and Wood [18]. The original classical layouts, however, tend to expand dominantly in one dimension, which reduces practicality for visualizing large data sets due to the limitation of display space.

To overcome the above limitation, various interaction techniques have been applied to the classical hierarchical views, such as zooming [19], fisheye-view [20, 21] and 1D distortion [22]. The zooming approach reduces the amount of context in the display and enlarges the focused area(s). This reduction is done by filtering the information by selecting a subset of the data along a range of numerical values of one or more dimensions. Although zooming is a natural way to navigate through a tree hierarchy, this technique also suffers from loss of context during the navigation. Retaining some context information, the fisheye view and distortion techniques enlarge an area of interest using distortion, while other portions of the image are shown with successively less detail. However, these

approaches have their own difficulties, where the distorted views could prevent the user from perceiving the information easily in a natural way.

Other interactive visualization methods provide a focus+context view by showing the sub-structures of interest while shrinking others. Among those, *SpaceTree* [7] and *DOITree* [23, 24] are the popular techniques for large tree visualizations. *SpaceTree* combines a zooming environment with conventional tree layouts that can dynamically layout structures to best fit the available screen space. Although *SpaceTree* is an excellent method for exploring large hierarchical data sets, the expansion view is only applicable to a single focused branch (or substructure) at a particular time. The limitation of a single focus point is addressed by *DOITree*, which use multi-focal tree layout algorithms that optimize the display when one or more nodes are expanded. Additionally, visual cues provide ongoing information about the content of hidden branches, with smooth animations showing when nodes are focused or shrunk.

Use of visual cues for the hidden branches has demonstrated effectiveness in exploring large hierarchical structures [7, 24]. In the existing techniques, the visual cues are very limited, due to using unshaded triangles whose sizes are proportional to the weight or number of nodes in the branches. Although this view provides some hints of the hidden structures, the visualizations do not provide a clear representation of the hidden structures, such as the exact number of child nodes or their properties. This limitation could reduce its perceptual effectiveness in exploring large structures.

The present article extends the interactive visualization techniques proposed by Nguyen et al [29] that provide informative visual cues to enhance the readability of hidden structures in an interactive tree visualization. The topological previews of the shrunk substructures are presented with three visual cue types, including *simple cues*, *tree cues* and *treemap cues*. The use of visual cues is illustrated via various data sets and case studies on *DOITree*. We also present a controlled experiment to evaluate the effectiveness of the visual cues. Our study measures completion time, accuracy and user preference across five different topology search tasks.

2. Visual Cues in Tree Visualization

A consistent problem with visualizing tree structures is the issue of how to represent node characteristics. Large hierarchical node-link tree structures suffer considerably when the amount of child nodes increases, as the screen fast becomes overcrowded when nodes are expanded [25]. A simple and widely implemented solution is to hide unfocused substructures when the display space is limited to the current view. Unfortunately, missing visual cues on those shrunken parts could potentially lead to a loss of context information.

Figure 1 illustrates a visualization using *DOITree*, based on a data set of oncology safety events, with over 500 items and 4 hierarchical levels. Visual cues are not deployed to show the topology of the hidden structures. Highlighted

blue items indicate selected nodes during the interaction. The hierarchical structure is presented clearly on the visualization with multiple focus branches. Space is reasonably utilized where the algorithm maximizes the number of expanding levels at a time. Unfortunately, the lack of visibility of topology, or preview of hidden structures, prevents users from gaining understanding of the hidden branches. Without atopolological view of shrunk structures, user perception of information is less effective. For example, in Figure 1, it is impossible to gain any property information of the branches at the second level (highlighted by red-dash rectangles), unless the branches are expanded via the interaction.

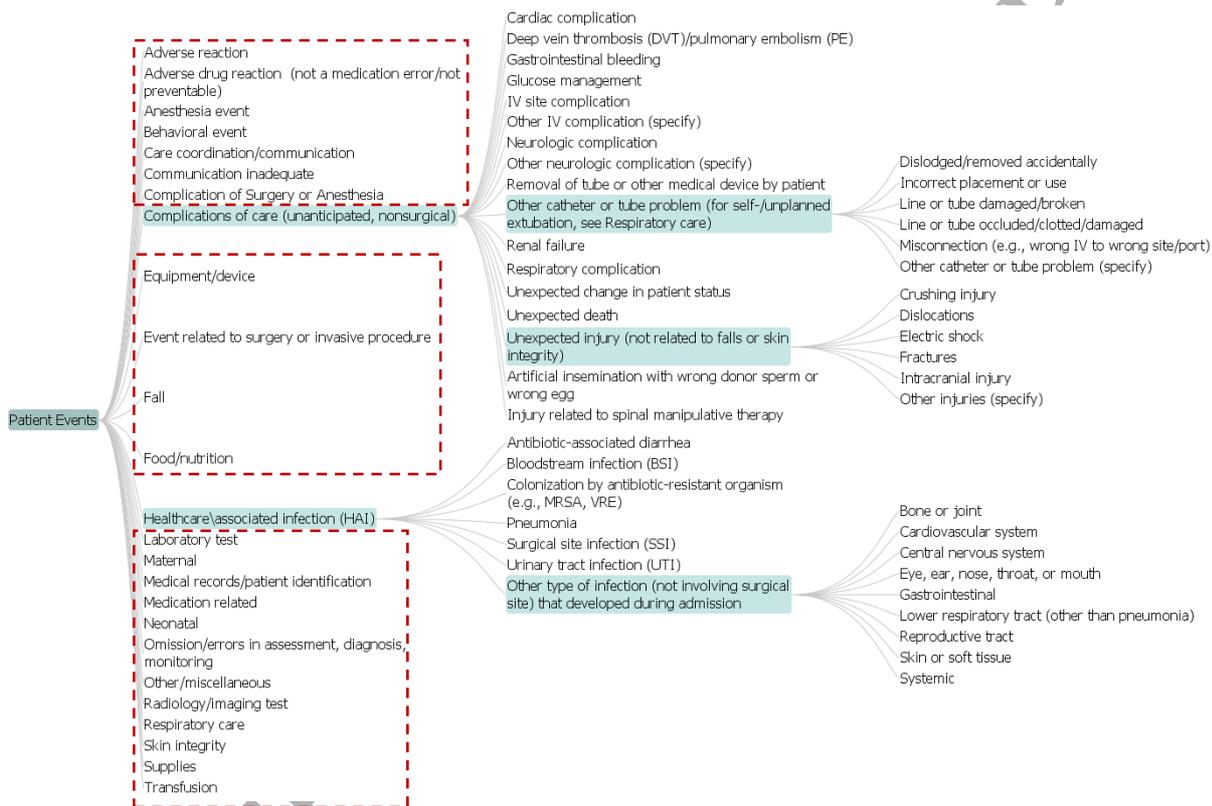


Figure 1. A visualization using *DOITree* that has no visual cues to preview the topological structure of the hidden branches. The figure does not show what nodes have additional children and could be expanded, even in the focus area (highlighted by red-dash rectangles)

Aiming to provide alternatives to the traditional interface of displaying all child nodes, Song et al [25] examined the additional options of a scrollable list interface and a multi-column interface compared to a traditional method of showing all nodes at once. The multicolumn interface showed significant promise as a better alternative, compared with performance in the traditional method of listing all child nodes, but has limited effectiveness when the number of nodes become too large as panning is still necessary.

Visual cues in node-link diagrams, such as search functions, shading, and arrow cues, were introduced to provide additional information on the direction of the branching system, highlight tree growth, and guide users to points of interest (such as [7, 25]). These innovations have used Horton-Strahler numbers as a successful algorithm to code for relevant branch hue and saturation [26, 27], structure based coloring to highlight hierarchical structures from related color palettes [28], and functions of zooming, fisheye-view, and one-dimensional distortion [7]. Unfortunately, display information is easily occluded when zooming or distortion takes place in those techniques. To overcome this limitation, focus+context techniques, such as *SpaceTree* [7] and *DOITree* [23], show sub-structures of interest whilst shrinking others into the background.

SpaceTree uses an isosceles triangle to preview branches of a hierarchical tree that are not expanded.. The height of the triangle is relative to the depth of the sub-tree, the shading proportional to the number of child nodes remaining in the branch, and width representing items divided by depth [7]. Some alternative cues were also presented at the experimental design. However, *SpaceTree* only allows users to view a single focus branch at a time in the expanded view. *DOITree* provides multi-focal display when one or more focus nodes are expanded, as well as a “bread crumb trail” effect of related nodes being highlighted [23]. The later variant of *DOITree* also included small, triangular cues next to unexpanded nodes to provide an indication of the topology of shrunk structures [24]. The absence of a triangle informs the user that it is a leaf node, meaning it cannot be expanded further. Although some indication of the topology is displayed via the visual cues, they provide limited information about hidden structure, including any specific properties of child nodes [24].

3. Visualization with Visual Cues

Our visualization maximizes the number of levels and nodes visible at any time when there is room to do so. However, it is not often possible to display an entire structure, especially for large trees. Following the lessons learned from [7] that nodes should be clearly readable when they are visible, our design does not progressively scale down the unfocused nodes by hiding them. Three visual cues are proposed in our implementation called *simple cues*, *tree cues* and *treemap cues*. The design rationale of the visual cues was based on the simplicity of *simple cues*, familiarization and interface consistence of *tree cues* to the main visualization, and effective presentation of attributed properties of *treemap cues*.

The cues are represented as layers on top of the existing visualization, with minimal obstruction of information.. Although the techniques can be applied to any tree visualizations, we implemented the visual cues on *DOITree* visualization [22] using the *Prefuse* platform [30]. *DOITree* represents tree structure in a classical way that also provides multiple-foci views with focus+context interaction, utilizes display space and provides smooth shrinking / growing and fade-out / fade-in animations among transitions. Ancestor paths of the current focused nodes are highlighted in blue to maintain landmarks during the navigation. Search is also provided in the visualization to

highlight those matching items. Following the design experiment that indicates user's preference for non-complicated preview icons [7], our visualization only shows nodes at the first level from the hidden branches.

3.1 Simple Cues

The simple cues provide a most simplistic view of the topology of the hidden branches. The technique aims to show the abstract preview of tree branches in a simplest-possible form where most users can quickly read the basic property of the hidden branches. The detail presentation of the structure and accuracy are not the main focus in this technique. In our technique, shaded triangular shapes are used to represent the properties of the structures.

Size of the triangles - size of the triangles is proportional to the weights of sub-root nodes or the number of their descendants. The weight W_v of a sub-root node v is calculated recursively from leaves to a vertex using formula:

$$W_v = 1 + C \sum_{i=0}^k W_{v_i} \quad (1)$$

where C is a constant ($0 < C < 1$), W_{v_i} is the weight assigned to the i^{th} child of node v , and k is the total number of child node of the sub-root node v . C is used to specify how the weights of child nodes would influence the parent node's weight. The higher value of constant C , the higher proportion of weight the child nodes will contribute to the parent node's weight. We use $C = 0.5$ in our experiment. The weights of all nodes are calculated in prior to the visualization.

The triangle's height S_v of a sub-root node v is calculated using equation??:

$$S_v = S_{\min} + \frac{(W_v - W_{\min}) \times (S_{\max} - S_{\min})}{(W_{\max} - W_{\min})} \quad (2)$$

where S_{\min} and S_{\max} are the smallest and the largest triangle's heights of the cues respectively, and W_{\min} and W_{\max} are the smallest and largest weights of vertices. To ensure that the visual cues do not oversight the main visualizations, we limit the maximum size S_{\max} of triangular shapes.

Shade of the triangles - when a triangle reaches its maximum size because of the constraint of maximum size of the cues, shade is used to represent weights of the larger hidden structures. Similarly, the shade darkness is proportional to a node's weight, e.g. the darker of the shade, the more nodes in the structure. If the size of a triangle is smaller than the maximum size, it is shaded by the lightest color.

The shade color SC_v of a sub-root node v is calculated using equation:

$$SC_v = SC_{\min} + \frac{(SC_{\max} - SC_{\min}) \times (W_{\max} - W_v)}{(W_{\max} - W_{\min})} \quad (3)$$

where SC_{min} and SC_{max} are the lightest and darkest shade colors of the cues, and W_{min} and W_{max} are the smallest and largest weights of vertices. The sizes and shading colors of the triangular shapes can be adjusted via an interactive menu.

Figure 2 illustrates the visualization at a navigational stage using the same data set as Figure 1. This view gives a much clearer preview of topological structure of the hierarchy compared to Figure 1. For example, the figure indicates a large number of descendants at branches rooted at “Complications of care (unanticipated, nonsurgical)”, “Event related to surgery or invasive procedure”, “Medication related” and “Transfusion”, indicated by the large black triangles. Although this visualization gives additional information for the hidden branches, it is not possible to gain further property information, such as how many children a sub-root may have, or what properties any hidden nodes may have.

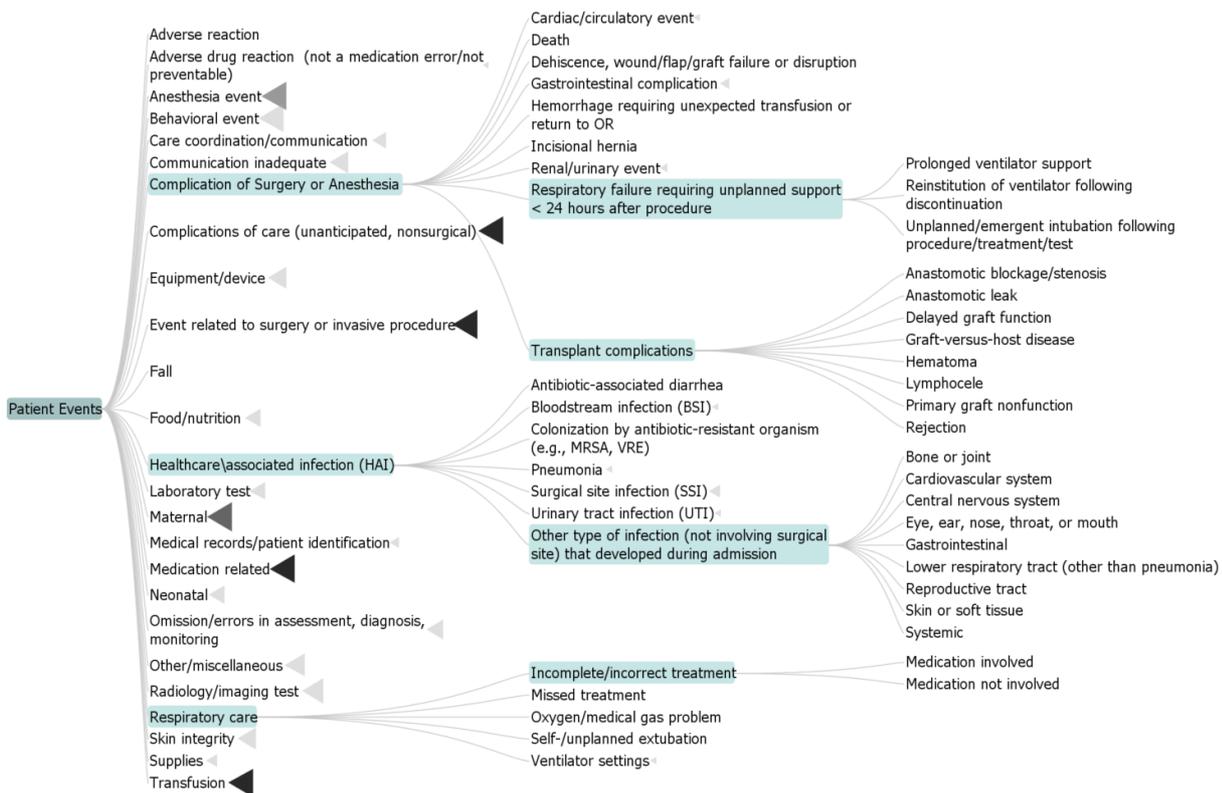


Figure 2. A visualization using DOITree on the same structure as Figure 1 where simple cues are used to preview the topology of the hidden branches. The view illustrates well the basic property of the shrunken structures.

3.2 Tree cues

Tree cues show the topology of the hidden branches by providing the classical tree view for the first level of the branches and the node's property. The design rationale was to provide a more detailed description of the direct children while maintaining low complexity in the view. Following the main node-link diagram visualization, *tree cues* employ the *classical hierarchical view* [3] to present the branches. The visualization does not present full structure of the branches as shown in the experimental design in *SpaceTree* [7] but only nodes at the first level. This method aims to simplify the visual cues when showing large branches.

Each child node is represented by a simple rectangle and an edge to its parent. The nodes are distributed evenly from their parent node. They are shaded to different degrees, representing the child node's weight (or the number of descendants of the child node). Similar to the shading method in simple cues, the darker the shade, the higher weight the child node has. The shade colors of tree cues are also calculated using formula 3.

Figure 3 illustrates the visualization with the tree view. It uses the same data set at the same navigational stage as Figure 2. In this figure, the properties of the first level of the hidden branches are presented quite clearly, such as the number of nodes at the next level, as well as whether the next level nodes have stronger weight. For example, in Figure 2, the branch rooted at "Complication of care (unanticipated, nonsurgical)" (highlighted by a red-dash rectangle) has approximately 20 children, and some of the child nodes also have a larger number of descendants than others.

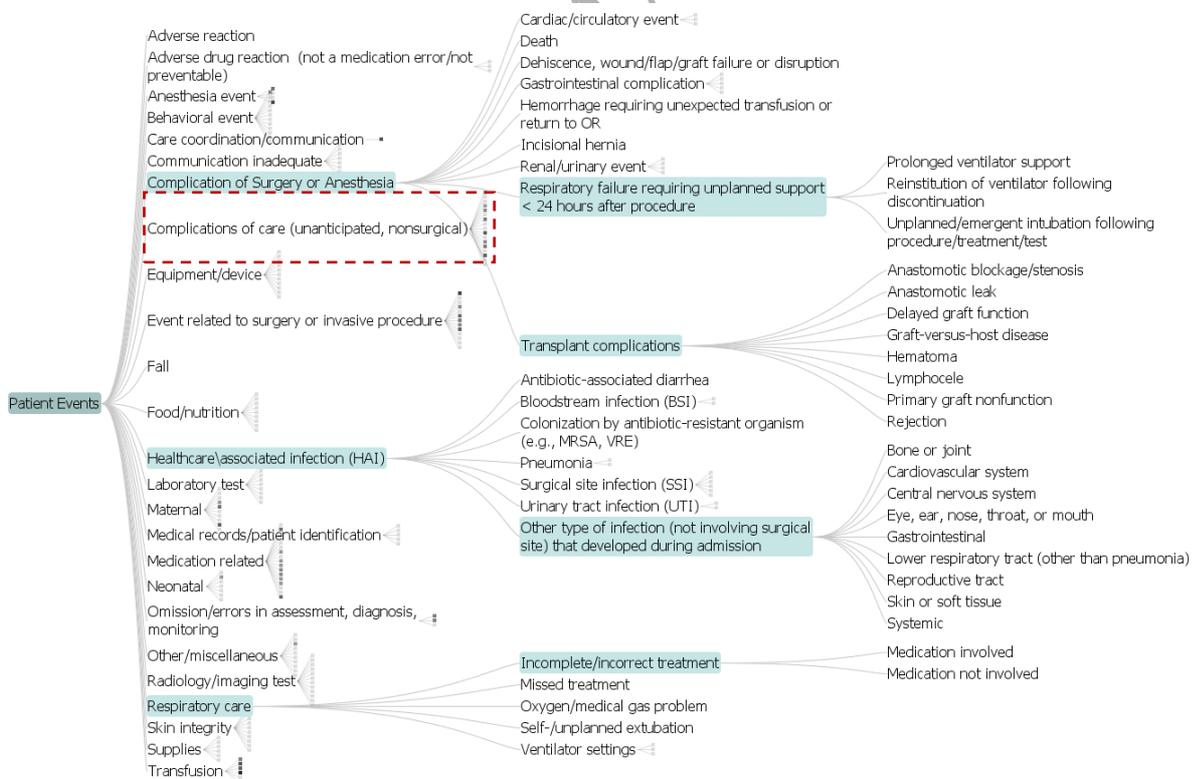


Figure 3. A visualization with the same navigational stage as Figure 2 using tree cues to preview the topology of the hidden branches. It illustrates the property of all immediate child nodes from shrunken structure clearly, such as at the node “Complications of care (unanticipated, nonsurgical)” (highlighted by a red-dash rectangle)

Although *tree cues* provide more details of the topological structure, the tree icons might expand significantly when having a large number of child nodes. This expansion might cause overlapping with the main view. Figure 4 illustrates the limitation of simple tree view when a node has a large number of children. Some tree cues overlap with the main view, as highlighted by the red-dash ellipses.

This limitation could be overcome by setting the constraint on the maximum number of the child nodes in vertical dimension (or horizontal dimension depending on the orientation) on the tree cues. The dimension is adjustable to ensure the lowest overlaps while maintaining largest possible size for better reading. The child nodes are displayed in orderly columns as a small icon on the cues. Figure 5 indicates the same visualization as Figure 4 when the dimension constraint is applied to the visualization. The overlaps present in Figure 4 are reduced significantly in Figure 5.

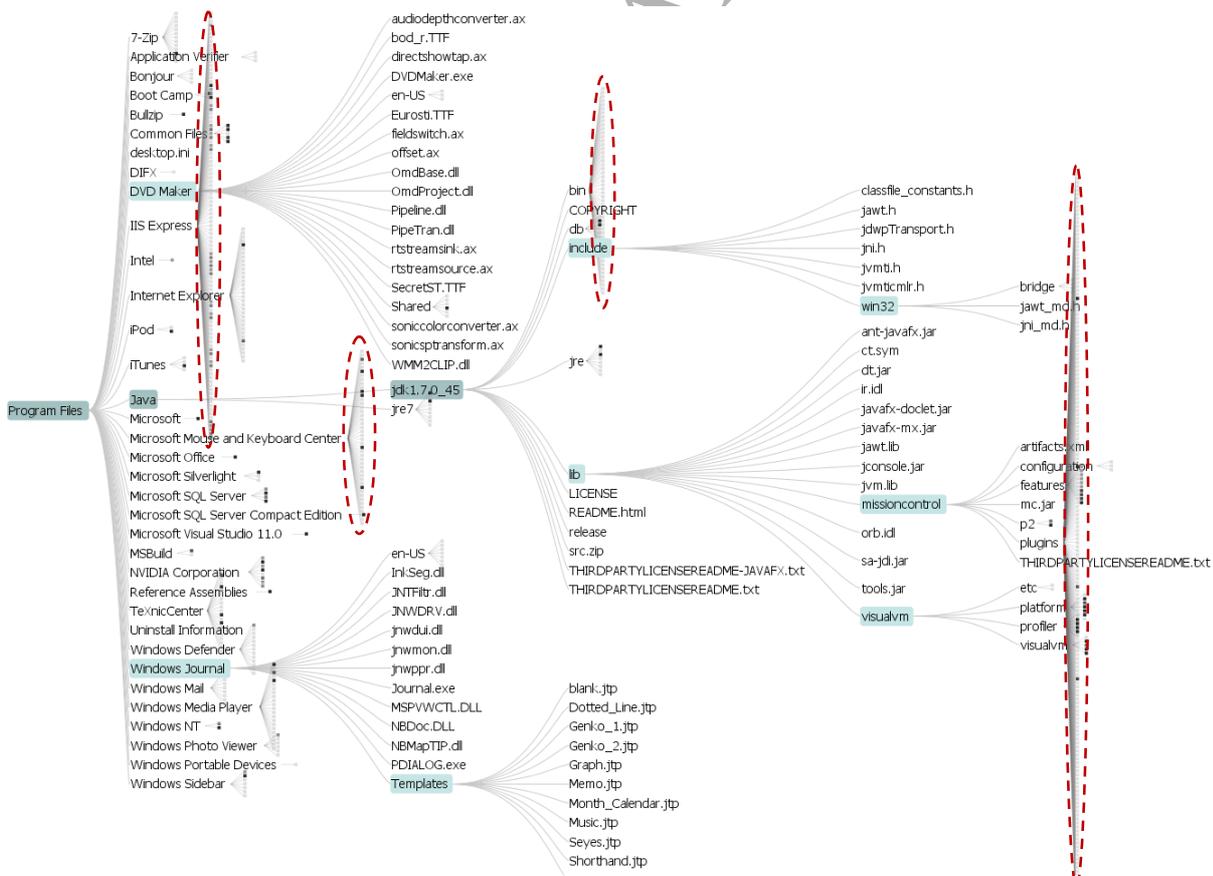


Figure 4. A visualization of a large file system at a navigational stage using *tree cues*. The figure shows overlapping might occur when a branch has many children (highlighted by red-dash ellipses).

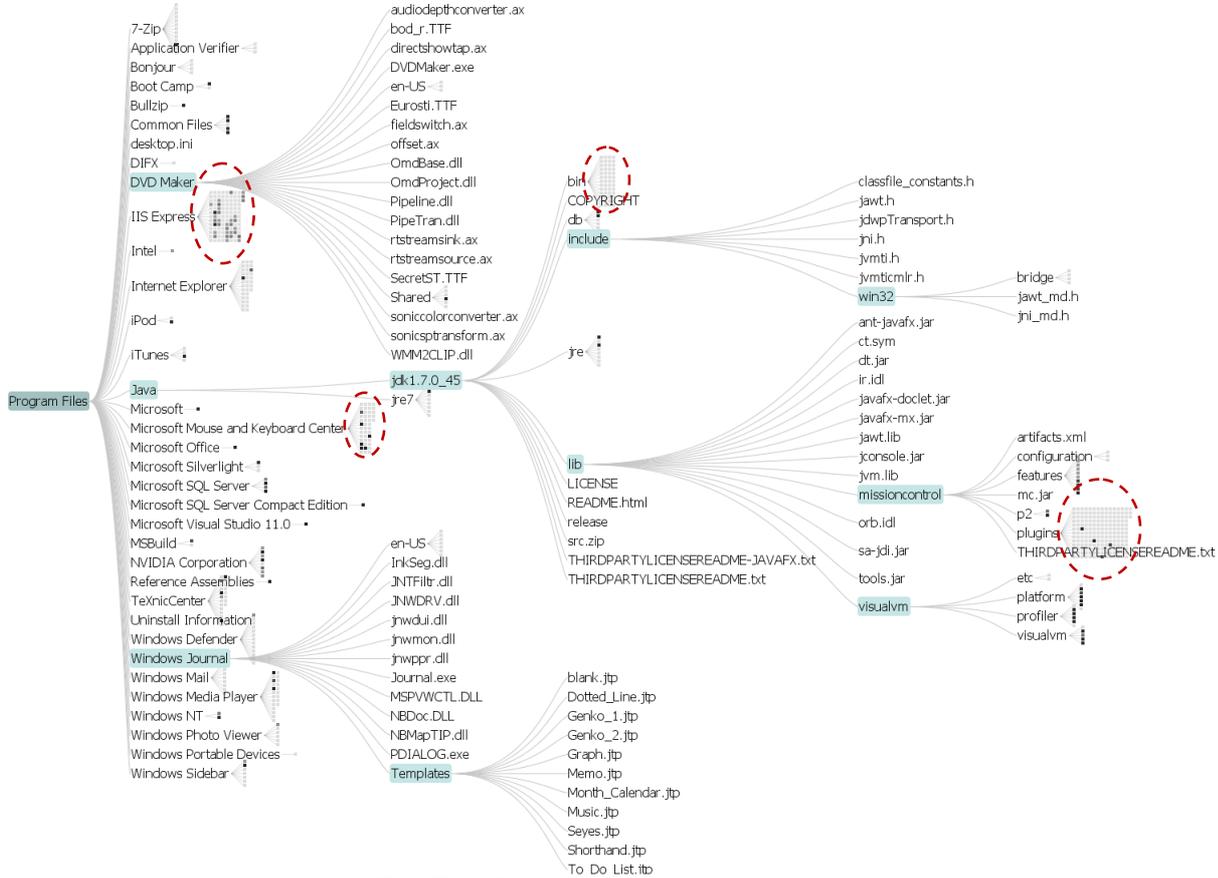


Figure 5. An improved visualization of the same data set and at the same navigational stage as Figure 4. The former overlapping is almost eliminated when using a constraint on the branches' vertical expansion (highlighted by red-dash ellipses).

3.3 Treemap cues

Treemap cues show the topology of the hidden branches using a treemap presentation of structures and node properties. This method aims to provide more effective presentation of attributed properties of the child nodes. We applied a space-filling algorithm called D&C Treemaps [31] on the first level of the hidden branches. This partitioning uses enclosure to represent the substructure, ensuring that all child nodes are located inside the sub-root's region. Although the algorithm is capable of generating different types of partition, we only apply the simplest and fastest vertical-horizontal rectangular partition to provide the treemap cues. Sizes and shades also represent the property of weights of child nodes in the treemap cues. If a node has more weight, its rectangular partition is larger. The outer height of a rectangular cue is calculated using equation 2. We also limit the maximum height of the outer

rectangles to reduce overlapping. The inner rectangles of the child nodes are calculated by the *D&C Treemaps* algorithm [31]. The areas of inner rectangles are also proportional to the weight of the child node in relation to its siblings. Shades are used to represent the relative weight of child nodes in comparison with other nodes. The shades are also calculated using equation 3.

Preliminary trials indicated that the treemap cues present better attributed properties of the hidden branches. For example, Figure 6 shows clearly the child nodes and their property (i.e. weight), from the branch rooted at “Complications of care (unanticipated, nonsurgical)” (highlighted by a red-dash rectangle). Although the treemap view is better for comparing node weight due to clear sizes and shades, it does have some problems. For example, large rectangular shapes might create overlapping among the treemap icons, or make it hard to distinguish nodes in small rectangles when the weights of one or more nodes are much larger than others (see Figure 7). Note: the size of the visual cues is easily adjusted via a menu.

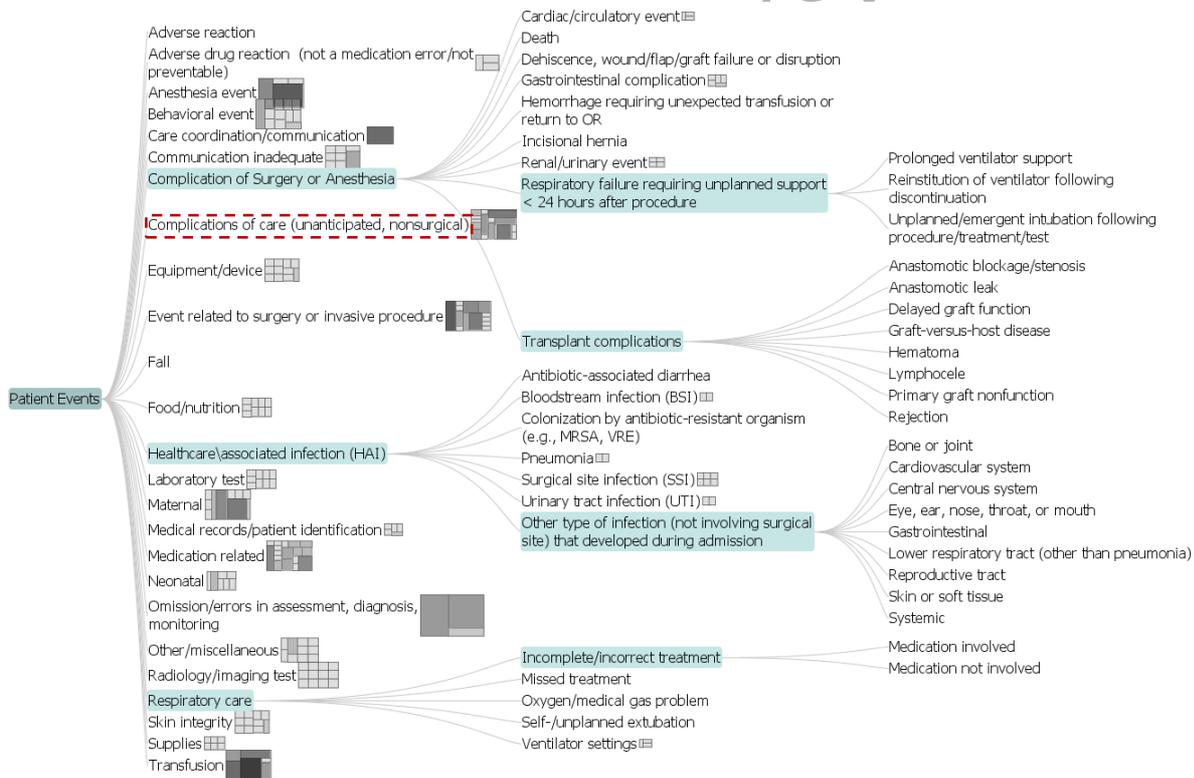


Figure 6. A visualization with the same dataset and navigational stage as Figure 3 using the treemap cues to preview the topology of the hidden branches. It illustrates the attributes of all immediate child nodes from shrunken structure where the nodes are easily compared to the others.

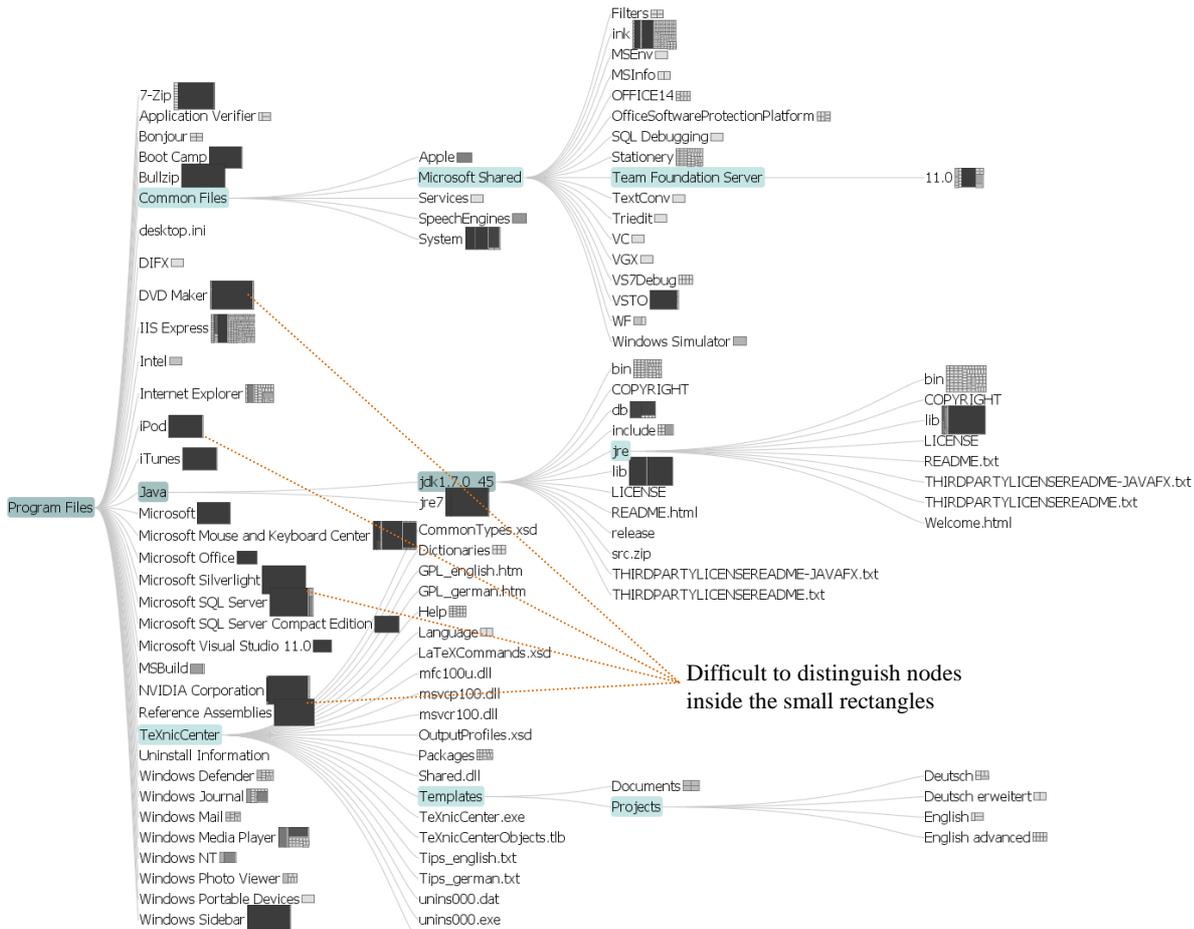


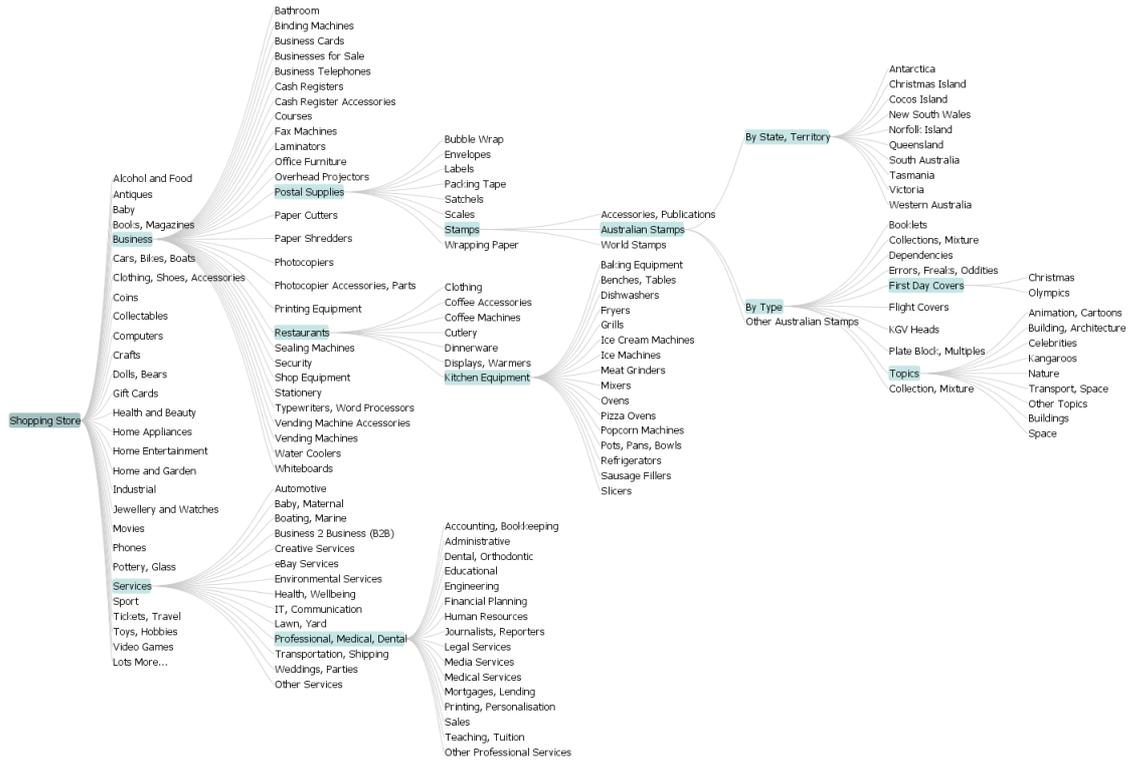
Figure 7. A visualization of a large file system at a navigational stage using treemap cues. The view shows the problem in identifying nodes inside the small rectangles when the node's weight variation is high.

4. Case studies

We have applied our visualization to various data sets in different applications, such as file systems, medical oncology and product categories. Preliminary evaluation indicates that users gain better understanding of the hierarchical structures and perform exploration tasks faster when using visual cues. However, it is still unclear which cue type (*simple cues*, *tree cues* or *treemap cues*) is most effective for previewing the topology of the hidden structures. However, it should be understood that this choice could be dependent on the nature of the data sets and inherent user preference.

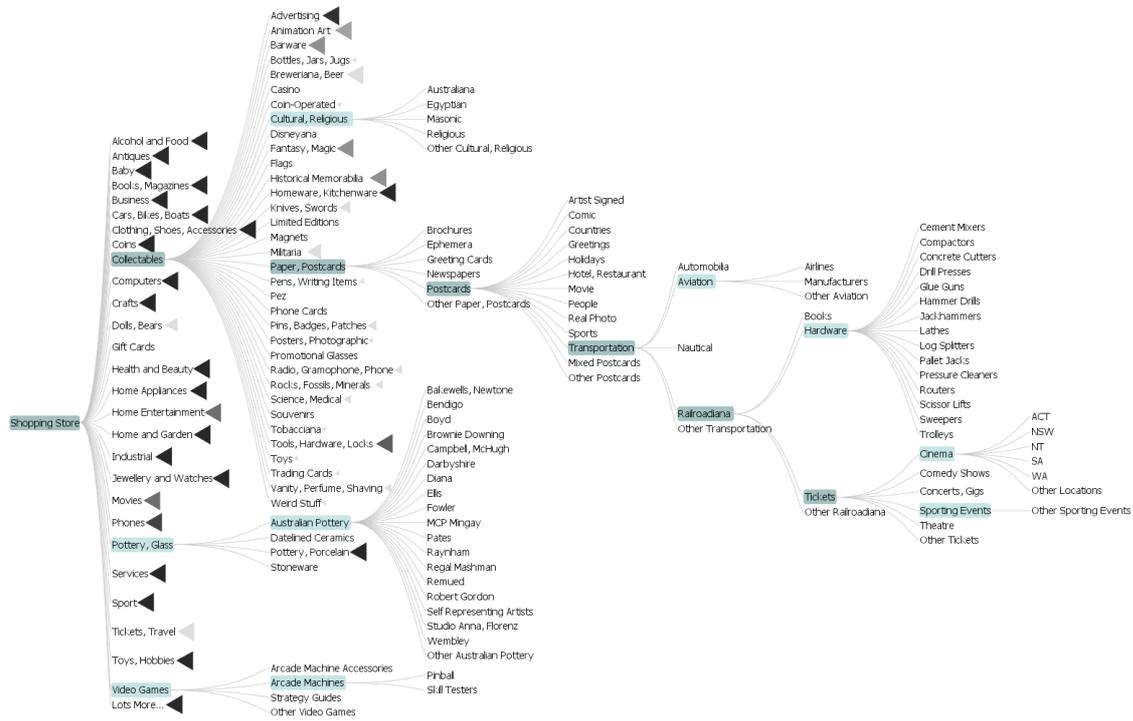
Figure 8 shows a visualization of a large set of product categories collected from ebay.com.au. The dataset has over 6,000 items and 7 levels. Figure 8a shows the visualization at a navigational stage without visual cues. Figures 8b, 8c and 8d present the visualizations at different navigational stages using *simple cues*, *tree cues* and *treemap cues* respectively. The figure 8a indicates quite clearly effectiveness of DOIs Tree visualization on showing the hierarchy

and focus nodes (blue nodes) when multiple nodes are selected concurrently. Unfortunately, it is not possible to see the topological property of the hidden branches in this figure. This limitation is overcome when using visual cues at figures 8b, 8c and 8d, ranging from simple triangular cues to more complex treemap cues. Figure 9 illustrates a visualization of a file system, collected from *Prefuse* package. It shows the same navigational stage when using *no visual cues*, *simple cues*, *tree cues* and *treemap cues* respectively for better comparison.

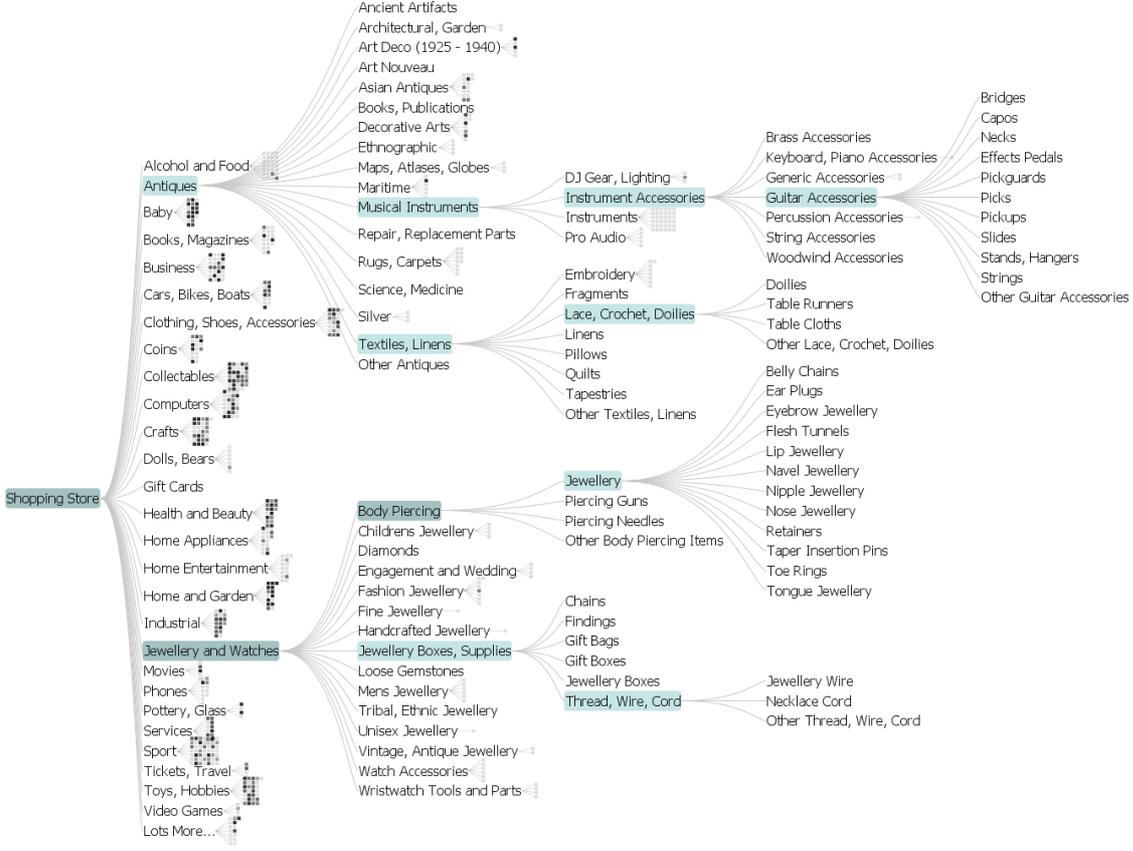


a

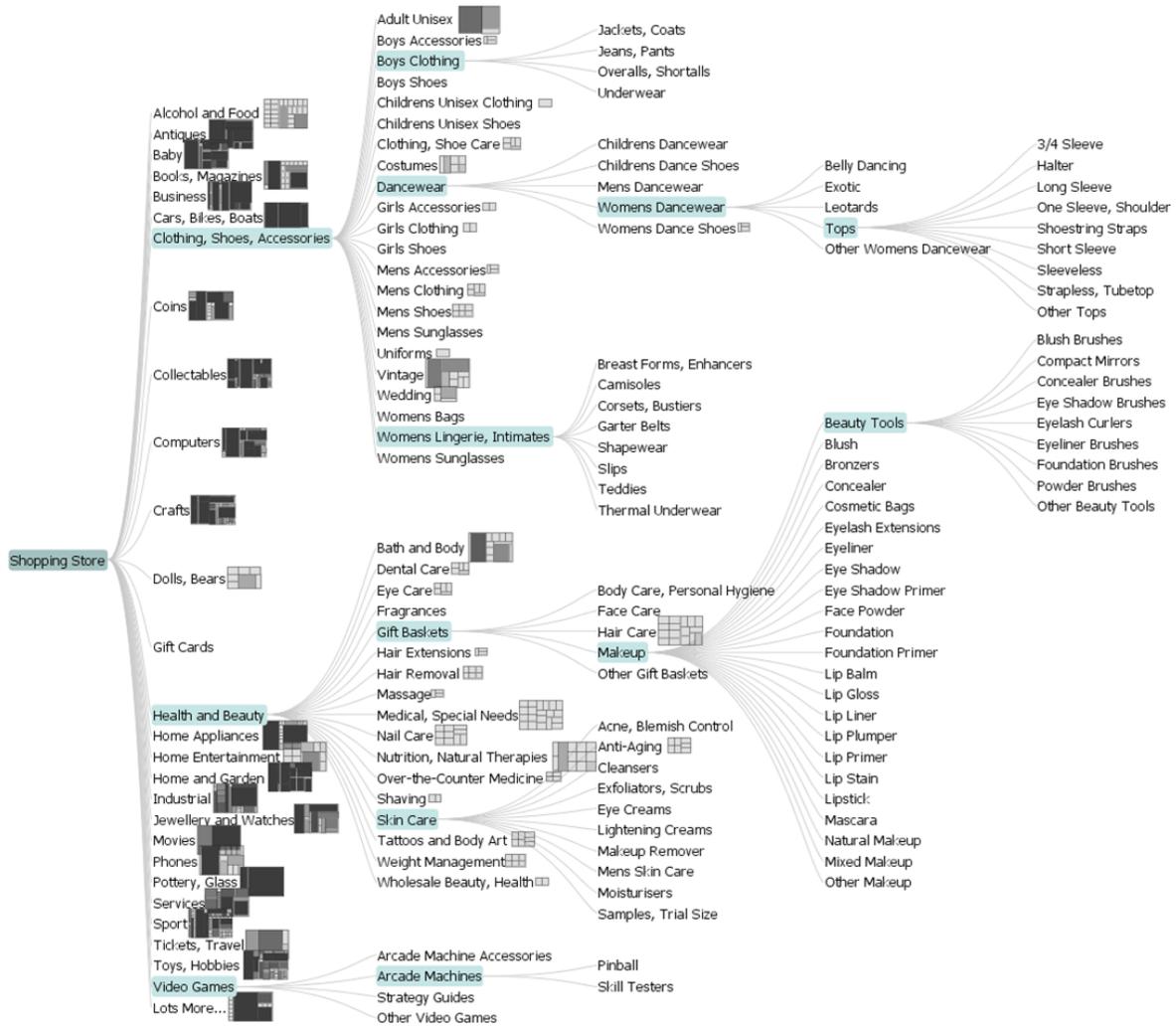
ACCEPTED



b

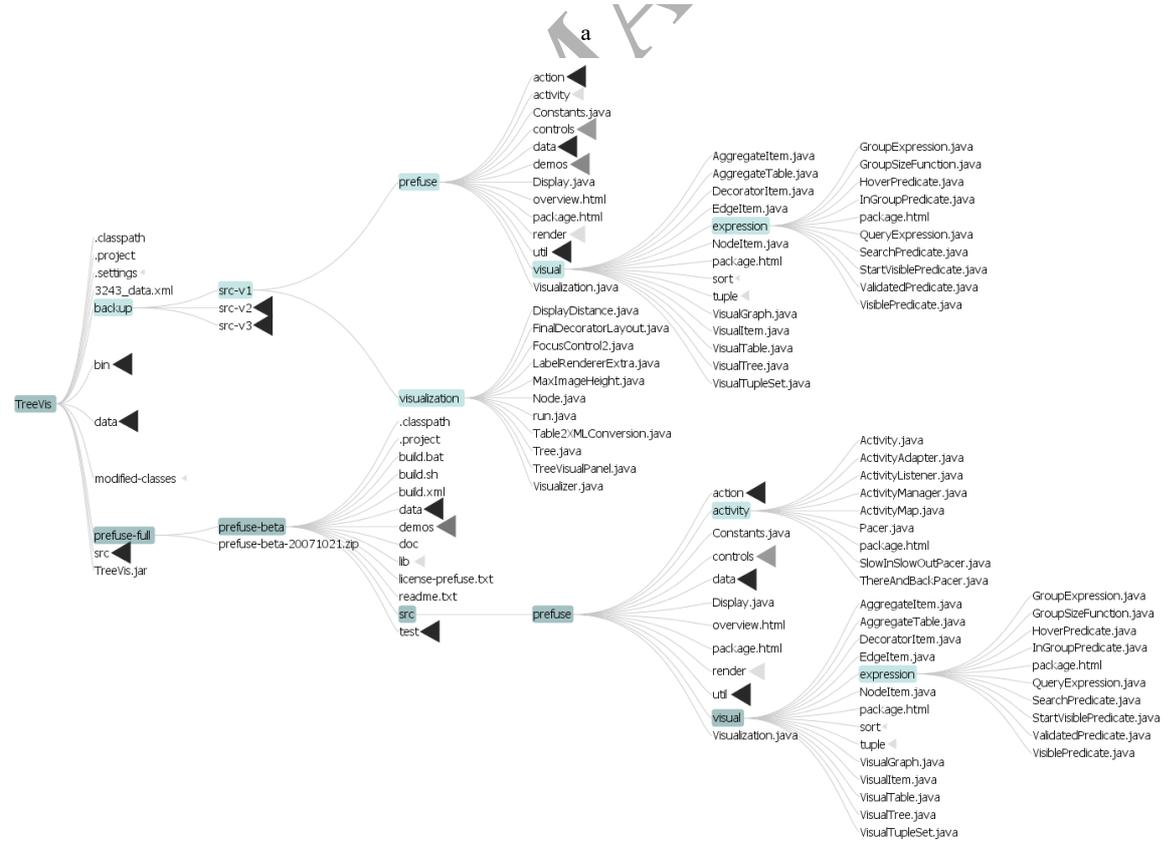
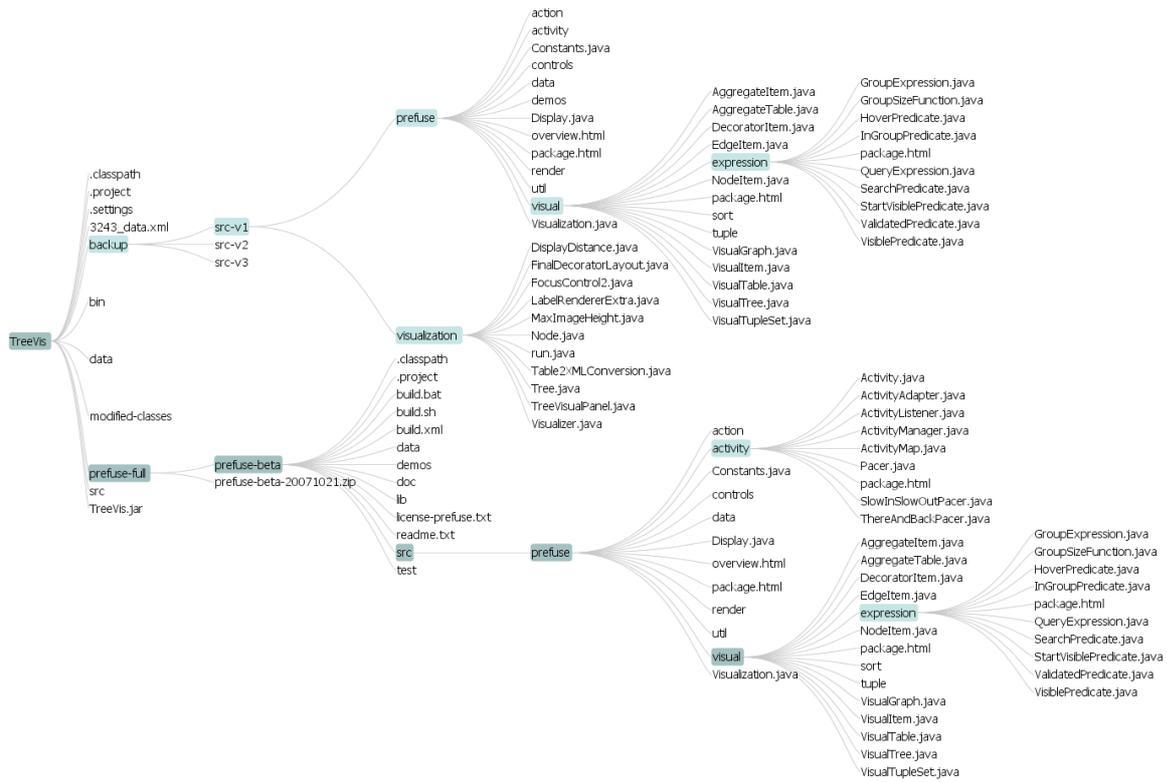


c

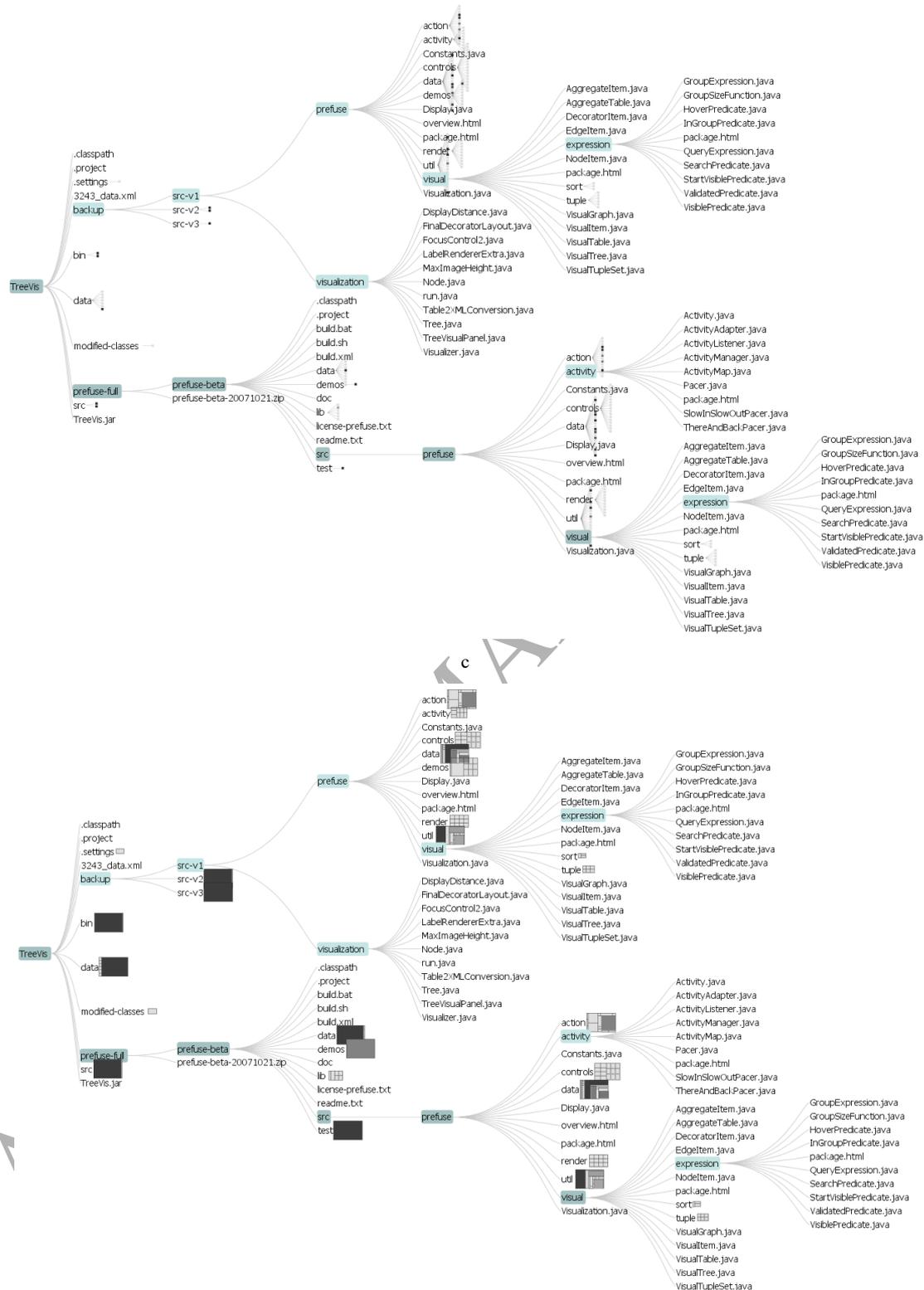


d

Figure 8. A visualization of a large product catalogue (over 6,000 items and 7 levels) at various stages, corresponding to a) no visual cues, b) simple cues, c) tree cues and d) treemap cues.



b



d

Figure 9. A visualization of a file system with more than 8 levels at the same stage corresponding to a) no visual cues, b) simple cues, c) tree cues and d) treemap cues.

5. Controlled Experiment

5.1 Aim

A controlled experiment was carried out to evaluate the effectiveness of visual cue types in terms of accuracy and time of completion, as well as overall user preference. Although early evidence indicates that a simpler cue was preferred in a former study [7], there is no body of evidence to drive predictions on whether a particular visual cue will prove superior. Instead, the nature of the tasks undertaken will attempt to highlight strengths and weaknesses of the different proposed visual cues when compared to the absence of a cue. There is no prior basis to assume there will be a difference between gender and age categories for preference. However, as this is the first time introducing the visual cues to users, an exploratory comparison is warranted.

5.2 Subjects

Recruited to the usability study were 28 participants (16 females and 12 males), with ages ranging from 21 to 55 ($M = 30.57$, $SD = 9.44$). Age was split into three categories for partial preference analysis. Four participants were 18 to 24, 16 participants were 25 to 34, and eight aged 35 to 55. One participant was a postgraduate student majoring in Psychology, whilst the other participants were recruited as a voluntary convenience sample from the general community with various backgrounds. Demographic information was screened, including gender, age, and perceived familiarity with the tree visualization interface. Only one participant reported familiarity with a node-link tree interface and software data visualization methods. Other requirements were met, including normal or corrected to normal vision, and general reading comprehension with no reported reading difficulties. All participants were fluent English speakers and accustomed with the left-to-right methodology of reading.

5.3 Experimental Design and Tasks

Prior to the experiments, we explained briefly the tree visualization and its interaction. We demonstrated three methods to the participants, including *simple cues*, *tree cues* and *treemap cues*. The participants were asked to familiarize themselves with the visualization and each visual cue method prior to the formal trial on a different sample data set. The training process took approximately 5 to 10 minutes to complete.

A within-subject, interactive usability study was conducted between the three visual cue conditions with a control condition of no visual cue. Efficiency, by means of completion time and accuracy, was recorded for five separate tasks. User preference on a 5-point Likert scale, choice ranking, and free-text comments were also requested.

For the interactive tasks, four similar data sets were created using a program extracted from the author's file systems. Each data set had the same number of nodes (2045 files and folders) and maximum deep level. The requirement for

separate data sets was necessary to avoid learning between each visual cue condition. File names were randomly altered as well as their locations to prevent participants relying on perceived familiarity between each data set. Branching systems were cropped and switched, whilst retaining semantic complexity. Each data set comprised a roughly equal number of child nodes from the root level, and a combined assortment of small, medium and large fan outs of nodes on randomized levels. Pilot testing was conducted to refine the stimuli and tasks, with all participants receiving brief standardized instruction prior to proceeding. The sample trees administered were randomly selected in each experience, with a randomization of visual cue conditions (e.g. simple cues, tree cues and treemap cues) to provide counterbalancing. The five tasks were as follows, and listed on a multiple choice questionnaire.

- Q.1. Identify the node on level two with the second least amount of direct child nodes on level three (root node is level one).*
- Q.2. Identify the node on level two that is the largest in total size, incorporating all descendants.*
- Q.3. How many leaf nodes exist on level three?*
- Q.4. What is the maximum number of levels the data set can expand to, going through the node [specified]?*
- Q.5. Identify a node on level four with more than 10 children on level five.*

Participants were then requested to rank each visual cue condition from 1 (strongly disagree) to 5 (strongly agree) across three questions then rank the visual cues in their overall preferred order. The questions cover the participants' understanding, effectiveness and user choice aspects, particularly "it was easy to understand the visual cues and their meaning", "the visual cues were effective in helping me complete the task" and "I would choose this visual cue interface to use" respectively (see Figure 12).

5.4 Results

5.4.1 Completion Time

Figure 10 depicts the completion time for each of the five tasks for each method. As each task was designed to explore various functions of the visual cues, a series of repeated measures (RM) ANOVAs were conducted for completion time. Alpha was set at .05.

For task one, no significant difference was present in the time to identify child nodes on level two, $F(3, 81) = 1.48$, $p = .23$, $\eta_p^2 = .38$.

For task two, there was a difference between time taken for participants to identify the largest node, $F(1.6, 44.81) = 24.83$, $p < .001$, $\eta_p^2 = .48$. Post hoc comparisons using a Sidak adjustment showed that the *no visual cues* condition was significantly slower than all other cue conditions: *simple cues* ($M_{\text{Diff}} = 61.71$, Sidak 95% CI [30.63, 92.80]), *tree cues* ($M_{\text{Diff}} = 73.71$, Sidak 95% CI [37.23, 110.20]) and *treemap cues*, $M_{\text{Diff}} = 68.93$, Sidak 95% CI [31.83, 106.03].

Task three saw a significant difference between visual cue conditions in time taken to identify the number of leaf nodes on level three, $F(1.5, 41.01) = 60.16$, $p < .001$, $\eta_p^2 = .69$. Post hoc tests showed the *no visual cue* condition performing considerably slower than *simple cues*, *tree cues*, and *treemap cues* ($M_{\text{Diff}} = 57.18$, $p < .001$ Sidak 95% CI [39.31, 75.05], $M_{\text{Diff}} = 48.75$, $p < .001$, Sidak 95% CI [29.72, 67.78], $M_{\text{Diff}} = 55.32$, $p < .001$ Sidak 95% CI [37.50, 73.15], respectively).

The time required to identify the maximum number of levels expandable was significant for task four, $F(2,1, 56.92) = 16.71$, $p < .001$, $\eta_p^2 = .38$. No *visual cues* were significantly slower than the *simple cues* ($M_{\text{Diff}} = 93.57$, $p < .001$, Sidak 95% CI [46.66, 140.49]), the *tree cues* ($M_{\text{Diff}} = 88.82$, $p < .001$, Sidak 95% CI [35.76, 141.89]), and the *treemap cues* ($M_{\text{Diff}} = 65.32$, $p = .01$, Sidak 95% CI [15.78, 114.86]).

Task five, $F(3,81) = 4.38$, $p = .01$, $\eta_p^2 = .14$, had post hoc comparisons reveal only *simple cues* are significantly faster than the *no visual cues* in identifying more than ten children on level five, ($M_{\text{Diff}} = 21.18$, $p = .01$, Sidak 95% CI [5.00, 37.38]).

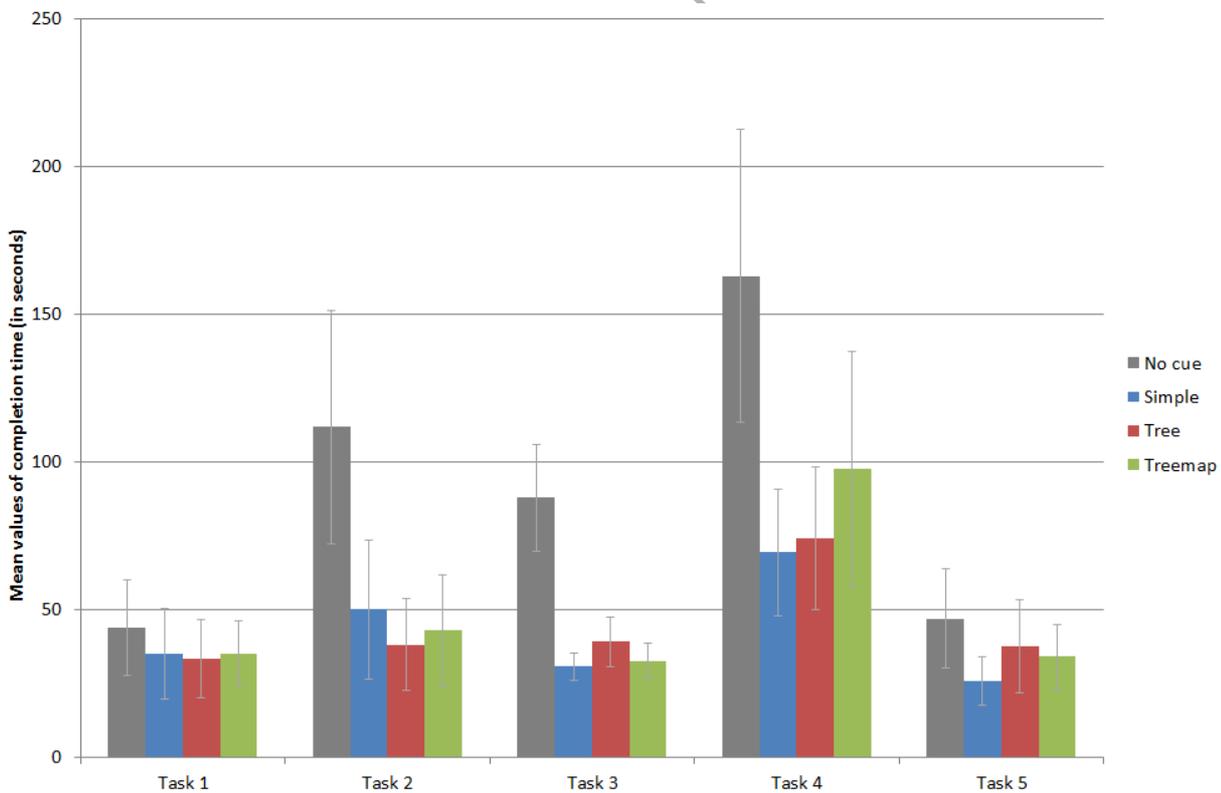


Figure 10. Completion time (with standard deviation) for all five tasks.

5.4.2 Accuracy

Figure 11 illustrates the accuracy graph for each of the five tasks for each method. For accuracy analyses, RM ANOVAs were once again conducted, with post hoc comparisons using Sidak adjustment. For task one, although not statistically significant, the *no visual cues* clearly had the highest rate of correct answers (89%) when compared to the visual cues, with *treemap cues* having the most incorrect answers (39%).

For task two, results indicated a statistical significance among the visual cue conditions when attempting to determine the node that was largest in total size ($F(3, 81) = 2.604, p < .001, \eta_p^2 = .336$). Post hoc tests revealed significant differences between *no visual cues* and *simple cues* ($M_{\text{Diff}} = .57, p < .001, \text{Sidak } 95\% \text{ CI } [.91, .23]$), and *no visual cues* and *treemap cues* ($M_{\text{Diff}} = .68, p < .001, \text{Sidak } 95\% \text{ CI } [.93, .42]$), with means suggesting *no visual cues* had a very high rate of 89% incorrect answers. *Treemap cues* had significantly more correct answers (79%) when compared to the relatively poor performing *tree cues* at 57% ($M_{\text{Diff}} = .39, p = .02, \text{Sidak } 95\% \text{ CI } [.73, .06]$).

There was a significant difference in error rates for task three, $F(3, 81) = 5.55, p = .002, \eta_p^2 = .17$. *Treemap cues* showed a higher rate of correct answers (82%) than all other cue conditions followed by *simple cues* and then *tree cues*. However, the only significant difference existed between *treemap cues* and *no visual cues*, ($M_{\text{Diff}} = .39, p = .01, \text{Sidak } 95\% \text{ CI } [.70, .09]$), with the *no visual cues* having only 43% of answers correct.

For task four, it was evident that participants performed rather poorly in this task, with roughly half of all answers incorrect. *Treemap cues* were the highest ranked for correct responses, and although not significantly different, had just over 60% answers correct. The remaining three conditions shared a range of 50-51% correct answers.

Task five did not show significant error variance, $F(3, 81) = .44, p = .73, \eta_p^2 = .02$. All visual cue conditions had a large proportion of correct responses, comprising between 80 to 90% for all conditions.

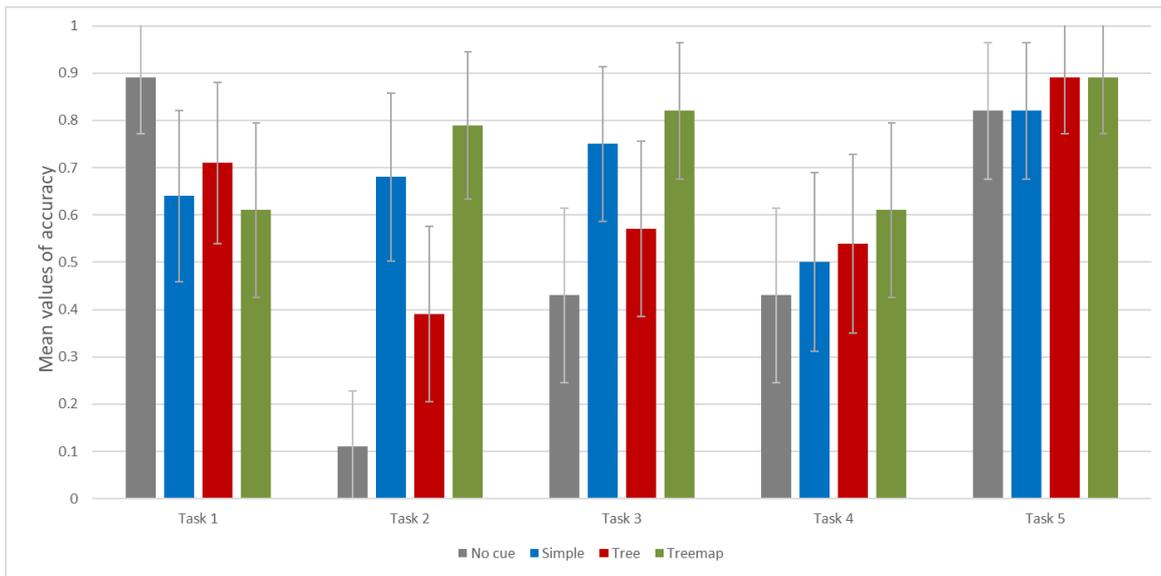


Figure 11. Accuracy (with standard deviation) for all five tasks.

5.4.3 Likert Scale

The Likert scale was analysed using a series of RM ANOVAs with post-hoc comparisons using Sidak adjustments, means are reported in Figure 12. The 5-point Likert scale is scored from 1 = strongly disagree to 5 = strongly agree. The higher the raw mean, the more favorably the visual cue was rated in relation to the question. There were no significant effects of gender or age.

Responses to question one differed significantly by visual cue type, $F(2, 44) = 6.62, p = .003, \eta_p^2 = .23$, with a pairwise effect between *tree cues* and *treemap cues* ($M_{Diff} = .34, p = .01$, Sidak 95% CI [-1.05, .36]), indicating participants rated the *tree cues* as easier to understand.

Question two also differed significantly by visual cue type, $F(2, 44) = 4.49, p = .02, \eta_p^2 = .17$. As with question one, there was a significant pairwise effect between *tree cues* and *treemap cues*, ($M_{Diff} = .67, p = .02$, Sidak 95% CI [-1.01, .60]), with *tree cues* considered more useful than both *simple* and *treemap* visual cues.

The third question also showed a significant main effect of visual cue type ($F(2, 44) = 4.28, p = .02, \eta_p^2 = .16$). *Tree cues* were significantly more likely to be selected for use than the *treemap cues*, $M_{Diff} = 1.07, p = .01$, Sidak 95% CI [.18, 1.95], with the *simple cues* falling in the middle.

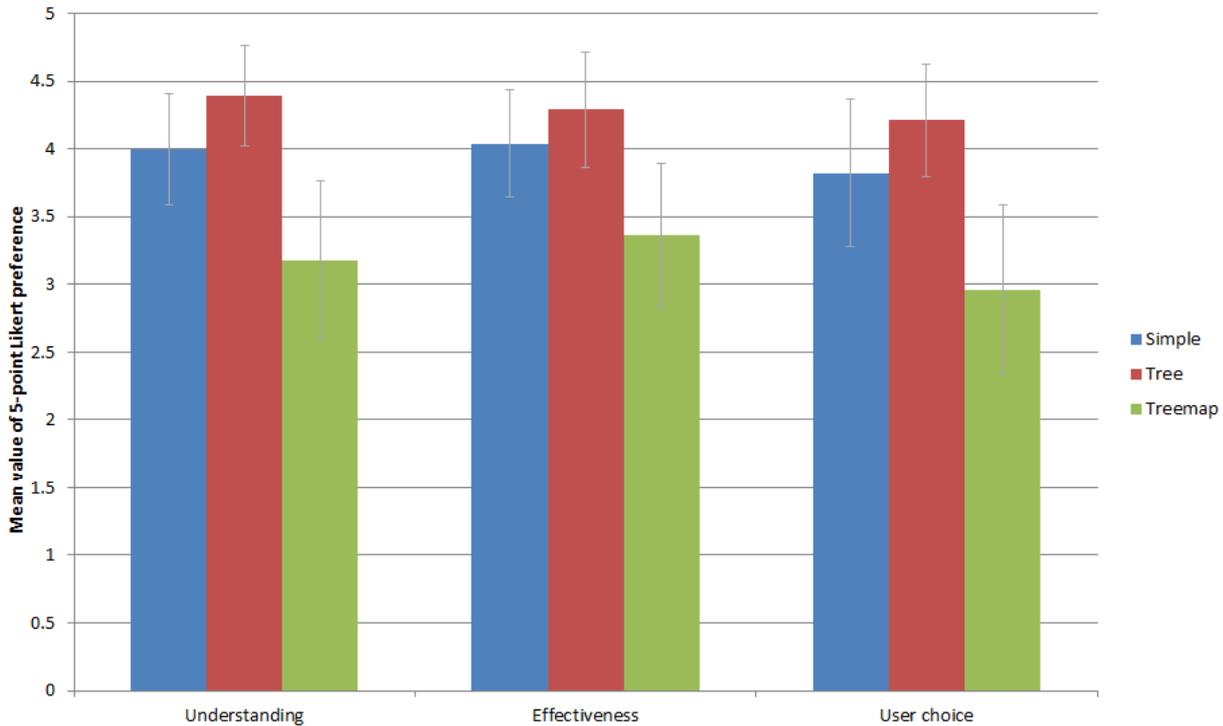


Figure 12. 5-point Likert preferences (with standard deviation) for 3 questions 1) Understanding (“it was easy to understand the visual cues and their meaning”), 2) Effectiveness (“the visual cues were effective in helping me complete the task”) and 3) User choice (“I would choose this visual cue interface to use”).

5.4.4 User Ranking

A nonparametric Wilcoxon signed-rank t -test was used to compare the ordinal rankings. No visual cue was ranked significantly lower than all three visual cues. *Treemap cues* showed significantly lower preference ratings when compared with *simple cues* ($z(N = 28) = 3.20$, $p = .001$, $r^2 = .10$) and *tree cues*, $z(N = 28) = 3.58$, $p < .001$, $r^2 = .11$. Combining these significant differences with the median scores, ranges, and mean signed ranks, the preferred visual cues were the *tree cues*, followed closely by *simple cues*, with *treemap cues* in distant third place.

6. Conclusion and Discussion

We have presented our approach using visual cues on DOITree for exploring Large Hierarchical Data. DOITree is an effective and elegant technique which provides multiple focused points using classical node-link views. By using visual cues to preview the topology of the branches, it provides the user with a better understanding of hidden structures and more insight during navigation. This paper presented three visual cue methods including *simple cues* as triangular shapes, a node-link *tree cues* and *treemap cues*. A usability study was also carried out to evaluate the effectiveness of the visual cue methods.

For the topology search tasks, the visual cues provided faster completion rates than when there were no cues. Despite not finding significant differences between the three conditions, the *simple cues* proved the fastest for four out of five tasks in relation to mean time. Against predictions, the *tree cues* underperformed in tasks relating to indicating specific child-node numbers. This may be due to increased time visually counting and comparing the small cue squares, as well as the possibility of visual cues overlapping.

Accuracy rates indicated that *treemap* cues were the most effective at allowing correct interpretation of branch topology, even exceeding the *simple cues* for total branch size estimation. This aligns with previous work that found *treemap* accurate at displaying tree topology more effectively [32]. A considerably even spread of mixed answers for total tree depth requires more investigation.

Despite the performance results, individuals still rate the *tree cues* overall most preferred option due to its perceived simplicity and clarity. *Simple cues* came in second, with *treemaps* third and considerably lower than the other two. Overall, the study provides a starting point for more specific context and experience related development for the visual cues. Further study could refine task selection, manipulate tree design and sizes, and incorporate more sophisticated testing technology to compare these novel options to tree visualization. Additionally, investigation in more ecologically valid contexts is warranted, as user characteristics (experience, familiarity, and motivation) may have played a role in performance.

References

- [1] Chen, C. *Information Visualization and Virtual Environments*. Springer-Verlag: London, 1999; 223pp.
- [2] Nguyen, Q.V., Huang, M.L. EncCon: An Approach to Constructing Interactive Visualization of Large Hierarchical Data. *Information Visualization*, 2005, 4(1), pp.1-21.
- [3] Reingold, E.M. and Tilford, J.S. Tidier Drawing of Trees. *IEEE Transactions on Software Engineering*, 1981, 7(2), pp. 223-228.
- [4] Eades, P. Drawing Free Trees. *Bulleting of the Institute of Combinatorics and its Applications*, 1992, pp. 10-36.
- [5] Melancon, G. and Herman, I. Circular Drawings of Rooted Trees. *Technical Report of the Centre for Mathematics and Computer Sciences*, 1998, INS-R9817, Amsterdam, The Netherlands.
- [6] Teoh, S.T. and Ma, K.L. RINGS: A Technique for Visualizing Large Hierarchies. *Graph Drawing*, 2002 (Irvine, California, USA), Springer; pp. 268-275.
- [7] Plaisant, C., Grosjean, J., and Bederson, B.B. SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation. *IEEE Symposium on Information Visualization (InfoVis 2002) 2002* (Boston, MA, USA), IEEE; pp. 57-64.
- [8] Lamping, J. and Rao, R. The Hyperbolic Browser: A Focus + Context Technique for Visualizing Large Hierarchies. *Journal of Visual Languages and Computing*, 1996, 7(1), pp. 33-55.
- [9] Johnson, B. and Shneiderman, B. Treemaps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures. *In Proceedings of the IEEE Visualization* (Piscataway, NJ, USA). IEEE Computer Society, 1991, pp. 284-291.
- [10] Bruls, M., Huizinga K., and van Wijk, J.J. Squarified Treemaps. *In Proceedings of Joint Eurographics and IEEE TCVG Symposium on Visualization*, (Vienna, Austria), 2000, Springer, pp.33-42.

- [11] Bederson, B.B., Shneiderman, B. and Wattenberg, M. Ordered and Quantum Treemaps: Making Effective Use of 2D Space to Display Hierarchies. *ACM Trans. Graphics*. 21(4), 2002, pp. 833-854.
- [12] Balzer, M. and Deussen, O. Voronoi Treemaps. In *Proceedings of IEEE Symposium on Information Visualization, USA*, IEEE Computer Society, 2005, pp. 165-172.
- [13] Nguyen, Q.V. and Huang, M. L. A Space-Optimized Tree Visualization. In *Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, IEEE, Boston, Massachusetts, October 2002, pp. 85-92.
- [14] Nguyen, Q.V. and Huang, M. L. EncCon: An Approach to Constructing Interactive Visualization of Large Hierarchical Data. *Information Visualization*, Palgrave, 4(1), 2005, pp. 1-21.
- [15] Di Battista, G., Eades, P., Tamassia, R. & Tollis, I.G. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, New Jersey, 1999.
- [16] Herman, I., Melançon, G. & Marshall, M.S. Graph Visualization in Information Visualization: a Survey. *IEEE Transactions on Visualization and Computer Graphics*, vol. 6, 2000, pp. 24-44.
- [17] Kennedy, A.J. Functional Pearls. *Journal of Functional Programming*. Cambridge University Press, 1996, 6(3), pp. 527-534.
- [18] Herman, I., Delest, M., and Melançon, G. Tree Visualization and Navigation Clues for Information Visualization. *Computer Graphics Forum*, 1998, 17(2), pp. 153-165.
- [19] Bruggemann-Klein, A. and Wood, D. Drawing Trees Nicely with TEX. *Electronic Publishing*, 1988, 2(2), pp. 101-115.
- [20] Furnas, G.W. Generalized Fisheye Views. In *Proceedings of ACM SIGCHI '86 Conference on Human Factors in Computing Systems*, Boston, Massachusetts, ACM Press, 1986, pp. 15-22.
- [21] Sarkar, M. and Brown, M.H. Graphical Fisheye Views. *Communications of the ACM (CACM)*, 1994, 37(12), pp. 73-84.
- [22] Nguyen, Q.V. and Huang, M.L. A Fast Focus+Context Viewing Technique for the Navigation of Classical Hierarchical Layout. In *Proceedings of the 7th International Conference on Information Visualisation (IV'03)*, IEEE, London, UK, July 2003, pp. 42-46.
- [23] Card, S.K. and Nation, D. Degree-of-Interest Trees: A Component of an Attention-Reactive User Interface. In *Proceedings of Advanced Visual Interfaces*. 2002, pp. 231-245.
- [24] Heer, J. and Card, S. K. DOITrees Revisited: Scalable, Space-Constrained Visualization of Hierarchical Data. In *Proceedings of Advanced Visual Interfaces*. 2004, pp. 421-424.
- [25] Song, H., Kim, B., Lee, B. and Seo, J. A Comparative Evaluation on Tree Visualization Methods for Hierarchical Structures with Large Fan-outs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 223-232. ACM, 2010.
- [26] Herman, I, Delest, M., and Melançon, G. Tree Visualisation and Navigation Clues for Information Visualisation. In *Computer Graphics Forum*, 1998, 17(2), pp. 153-165.
- [27] Herman, I, Marshall, M.S., Melançon, G., Duke, D.J., Delest, M. and Domenger, J.P. Skeletal Images as Visual Cues in Graph Visualization. In *Proceedings of Data Visualization '99*, Springer Vienna, 1999, pp. 13-22.
- [28] Yang, J., Ward, M.O. and Rundensteiner, E.A. Interring: An Interactive Tool for Visually Navigating and Manipulating Hierarchical Structures. In *Proceedings of IEEE Symposium on Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pp. 77-84.
- [29] Nguyen, Q. V., Simoff, S. and Huang, M. L. Using Visual Cues on DOITree for Visualizing Large Hierarchical Data." In *Information Visualisation (IV), 2014 18th International Conference on*, pp. 1-6. IEEE, 2014.
- [30] Heer, J., Card, S. K., and Landay, J. A. Prefuse: A Toolkit for Interactive Information Visualization. In *Proceedings of CHI 2005*, Portland, Oregon, USA, ACM, pp. 421-430.
- [31] Liang, J., Nguyen, Q. V., Simoff, S. and Huang, M. L. Visualizing Large Trees with Divide & Conquer Partition. In *Proc. VINCI 2013 – 6th International Symposium on Visual Information Communication and Interaction*, August 2013, Tianjin, China, ACM, 2013, pp. 79-87.

- [32] Kobsa, A. User experiments with tree visualization systems. In *Proceedings of IEEE Symposium on Information Visualization*, 2004, pp. 9-16.
- [33] Henry, N., Fekete, J. and McGuffin, M.J. NodeTrix: a hybrid visualization of social networks. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 2007, pp. 1302-1309.

ACCEPTED MANUSCRIPT