The Mobile Tree Browser

A Space Filling Information Visualisation for Browsing Labelled Heirarchies on Mobile Devices

Paul Craig
Dept. of Computer Science and Software Engineering
Xi’an Jiaotong-Liverpool University
Suzhou, China
p.craig@xjtlu.edu.cn

Xin Huang
Dept. of Computer Science and Software Engineering
Xi’an Jiaotong-Liverpool University
Suzhou, China
xin.huang@xjtlu.edu.cn

***Abstract*—This paper introduces a novel information visualization technique for browsing labelled hierarchies on mobile devices. The display is optimized for readability and navigation on devices with limited display space, using animation to smooth the transition between views when a user clicks on nodes to navigate the hierarchy. The technique is evaluated by comparing it to more traditional table and tree views with users expressing a clear preference for the space filling hierarchy view. This demonstrates the potential of information visualization to improve interface usability and functionality on mobile devices.**

***Keywords*—Pervasive Computing, Information Visualisation, Mobile Visualization**

I. Introduction

The beginning of this century has seen a marked improvement in the capabilities of mobile devices with increased processing power, graphics capabilities, memory capacity and touch-screen responsiveness. The mobile phone industry has been putting high end computing technology in the hands of an increasing number of users with an estimated figure of over 1.75 billion smartphone users worldwide. This means that a remarkable 25% of the global population now own a smartphone [1]. These numbers expected to rise in the near future as more services are available on our mobile devices, and the technology expands into new developing markets.

A parallel trend has been the rise of *big data* [2] where we are now able to record and collate massive amounts of information from personal devices and data acquisition technologies such as radio-frequency identification readers and wireless sensor networks. This has massive potential to enhance our lives by helping us improve things like our fitness, health, personal interactions, management of resources and business activities. We do, however, need to consider that the utility of the data only goes as far as our ability to use the data [3]. This depends on the availability of human-oriented techniques such as *information visualization* [4, 5] that work to reinforce our natural cognitive processes to make the data accessible and comprehensible. The *big data* is useless unless we actually have some way to work with it.

None-the-less, the sheer quantity of data available coupled with the amount of investment in the development of mobile apps aimed at profiting from the high smartphone adoption rates has raised user expectations of what mobile devices, and mobile applications, should be capable of. Twenty years ago at the end of the 20th century mobile devices could be used for little more than text messaging, e-mails and very occasional rudimentary web browsing for things like checking the weather or train times. These days mobile devices are being used for increasingly sophisticated operations such as social media communication [6], e-commerce [7] and online banking [8]. This is despite the feeling that most mobile apps are simply hacked-down versions of their desktop pc counterparts [9]. And mobile devices are set to play an even bigger role in our lives with the advent of the internet-of-things [10-12] which looks to connect a growing number of electronic devices via the internet.

The vision of the internet-of-things is one of pervasive and ubiquitous computing [13, 14] where computers are everywhere and applications run seamlessly between networked devices. Users should be able to move between devices and carry their data with them without restrictions on their mobility, access to data or the functionality of the software. Hence mobile devices should be able to interface with data and perform tasks that would previously only be accessible from a desktop pc [11].

A typical internet-of-things scenario might involve a user playing music by selecting files on their mobile phone to be piped through their computer to speakers in the different rooms of the house with sensors detecting which room the user is in to adjust the volume of the music accordingly. This would allow the user to access their entire music collection from their mobile device and mean that the mobile interface should have equivalent functionality to the software on their computer, including things such as allowing them to browse the file hierarchy or move files from different folders to create a custom playlist.

The problem is that it is not easy to adapt interfaces capable of more advanced functionality on a desktop pc to work well on a mobile devices [15-17]. This is due to the limitations of mobile input modalities and reduced display space. Critically, user selections are less accurate on a mobile device and mobile devices with smaller screens are incapable of displaying as much information on the screen at any one time.

Research has however shown that the application of information visualization techniques can extend the capability of mobile devices to handle more ambitions user requirements and more challenging data-sets [18]. The aim of this paper is to investigate how we can apply these guidelines to the design of an interface for browsing hierarchical data on a mobile device.

II. Related Work

Despite the recognized importance of interaction to information visualization and the opportunity of more natural interaction offered by mobile devices [16], relatively little work has been done to develop information visualization interfaces for mobile devices, or indeed understand how we can better design information visualization interfaces for mobile devices. This can perhaps be explained by a general attitude that displays for information visualization should be as large as possible and that there is essentially a trade-off between information communicated and available screen space [19]. It's also considered that a smaller display can only be used *effectively* for aggregates and overviews [20]. Other authors cite input peripherals and device limitations as a reason why mobile visualization is less feasible [21]. But the fact is that while mobile devices may not be *ideal* for information visualization, they *are* becoming increasingly popular and there is a clear need for the types of interface that allow us to interact with large-scale complex data.

General guidelines for the design of mobile visualizations tell us that the design should be simple and the user should be able to interact more directly with the data rather than have to operate menus and controls [16]. Other guidelines suggest that interaction should be fluid and seamless [15] while more recent research has taken a lower level approach considering how specific aspects of information visualization design should be implemented on mobile devices [18]. This has resulted in the more concrete guidelines for mobile information visualization interface design such as:

A. *Use techniques that make more efficient use of available screen space or do not require accurate selections.*
B. *Keep text and selection targets above a constant device-specific minimum size and scale other elements to fit these constraints.*
C. *Don't display too much information on the screen at the same time. Allowing the user to interact to view more data over time with animation used to smooth the transition between views.*
D. *Use virtual buttons to switch between different types of selection replacing the functionality normally accessable through keyboard-shortcuts and mouse buttons.*
E. *Don't allow important information to be hidden by the user's finger during interaction.*

These guidelines have been applied to the design of a mobile interfaces for bivariate data to be displayed in a scatter-plot (see figure 1) and an animated map views for event data (figure 2) [18].

The mobile scatterplot for bi-variate data (see figure 1) has buttons on the left-hand-side which can be used to switch between excentric labelling [22, 23] and box selection or move to the details view on another screen. Excentric labelling can be used to label all the elements in a given area, moving labels away from under the user's finger. Distortion, based on the distribution of data about the x and y axes, can be applied using the slider on the right-hand-side. This makes it easier to select smaller groups of items when outliers would normally push the body of the data into a small area of the plot. As the slider is moved, and the level of distortion changes, the points are animated to move gradually to their new positions so as not to disorientate the user.

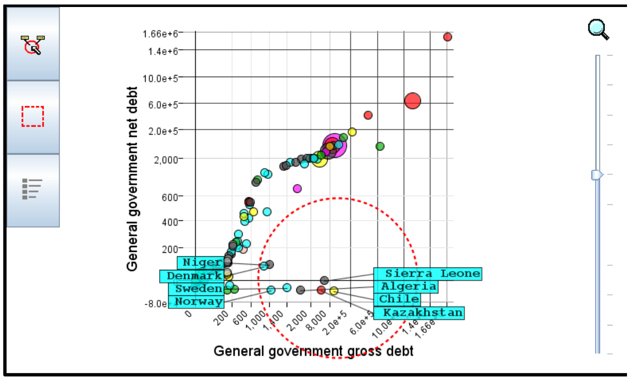
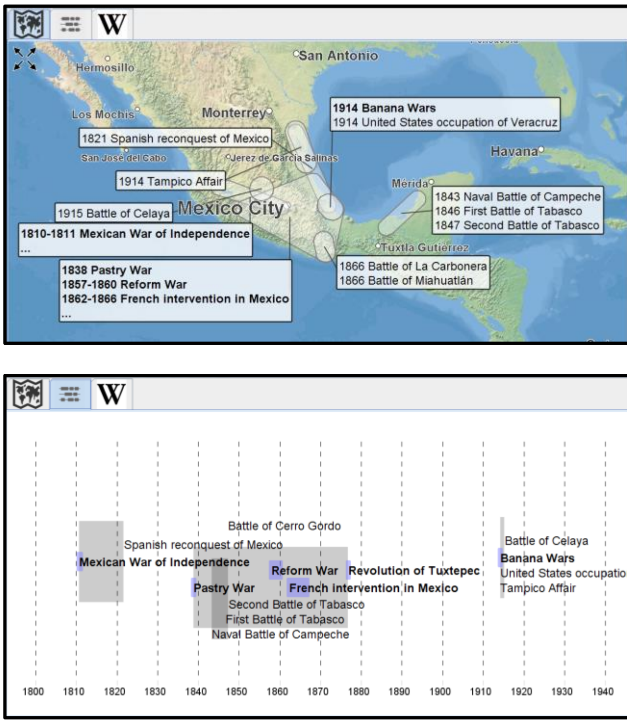


Fig. 1. Mobile device scatter-plot with variable distorion and excentric labelling



1914 Banana Wars
1914 United States occupation of Veracruz

1821 Spanish reconquest of Mexico

1914 Tampico Affair

1915 Battle of Celaya

1810-1811 Mexican War of Independence
...

1843 Naval Battle of Campeche
1846 First Battle of Tabasco
1847 Second Battle of Tabasco

1838 Pastry War
1857-1860 Reform War
1862-1866 French intervention in Mexico
...

1866 Battle of La Carbonera
1866 Battle of Miahuatlán

Battle of Cerro Gordo
Spanish reconquest of Mexico
Mexican War of Independence
Reform War
Revolution of Tuxtepec
Pastry War
French Intervention in Mexico
Second Battle of Tabasco
First Battle of Tabasco
Naval Battle of Campeche
Battle of Celaya
Banana Wars
United States occupatio
Tampico Affair

Fig. 2. Mobile device animated map and timeline interface for exploring geo-temporal event data. Map screen (top) and timeline (bottom).

The animated map and timeline for geo-temporal search results (figure 2) has been adapted from an original desktop application which clusters events across time for display on a normal desktop pc [24]. The mobile version increases the level of data aggregation using fewer clusters to display the same data. It also uses fewer labels and moves the animated map and timeline views onto different screens to save screen-space. The cluster outlines used in the map and timeline views provide

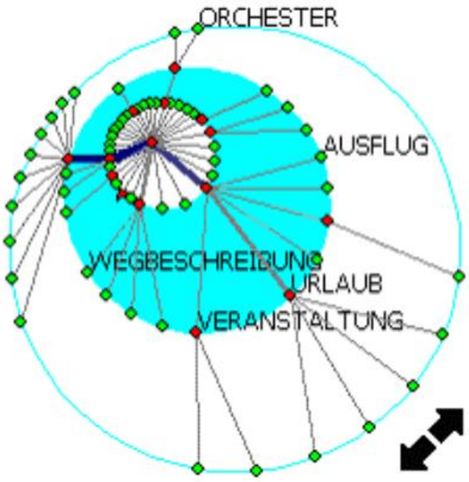


Fig. 3. *Magic eye view* radial heirarchy view for mobile devices.

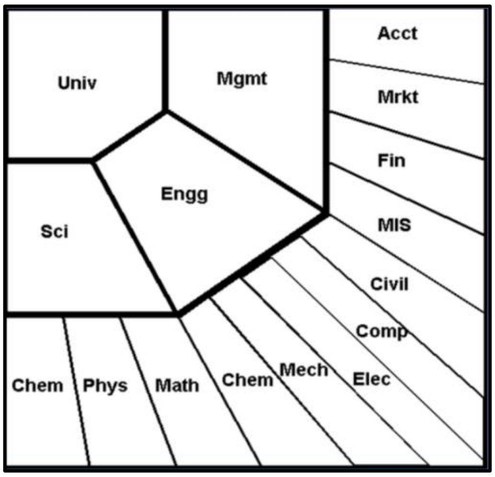


Fig. 4. Radial Edgeless Tree (RELT) space-filling radial layout for viewing hierarchies on mobile devices.

adequate space for inaccurate selections and selecting clusters to filter, and *drill-down* into the data, allows the user to view more of the data over time making up for the information lost in any single screen of the smaller display.

Other information visualization applications for mobile devices make efficient use of available screen space but fail to account for the inaccuracy of user selections. For example, the *magic eye view* visualization for hierarchical data [25] (see figure 3) uses a space-efficient radial layout for the nodes of a hierarchy. This is capable of displaying larger hierarchies on mobile devices but only a subset of nodes are labelled. Selection targets are small, and little attention is paid to how the user can interact with the display to select individual nodes in order to navigate the hierarchy.

Radial Edgeless Tree (RELT) is another technique for browsing hierarchical data on mobile devices [17, 26, 27]. This successively subdivides the display space using a radial space filling layout with labeled nodes occupying polygon cells (see figure 4). This claims to be more space efficient than *magic eye view* although it is difficult to accept any direct comparison since RELT displays labels and *magic eye view* doesn't. The chief benefit of RELT is that includes labels and the polygon cells containing the labels could be clicked to select nodes. The problem is that the radial layout could prove difficult for users since we overwhelmingly prefer to scan text that is left-aligned and stacked [28]. Moreover, more traditional stacked text is certainly a more space efficient layout for *labelled* text since an irregular polygon enclosing a horizontal label necessarily includes spare space that could be collapsed if the label could be simply placed above or below its neighbor.

The space filling layout for hierarchical data used by the Concept-Relationship Editor application [29] stacks adjacent nodes to optimize readability and uses animation to smooth the transition between views whenever the user clicks on a node to navigate the hierarchy. This, according to our aforementioned guidelines for mobile information visualization design, both A) *does not require accurate selection* and B) *displays more of the data over time according to user selection.* The layout of nodes

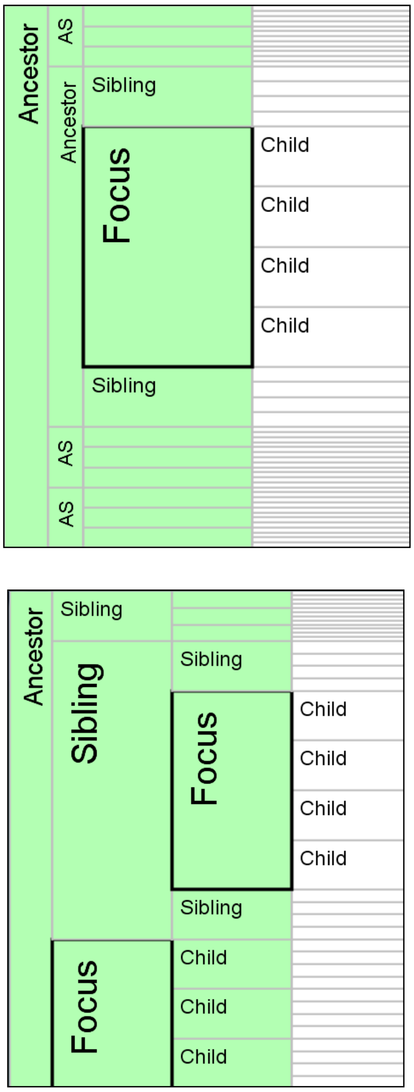


Fig. 5. Concept-relationship editor space-filling layout illustrated for one selected node (left) and two selected nodes (right). Priority vertical and horizontal spacing is given to children of the selected focus node, then siblings, then siblings of any ancestor (labelled AS).

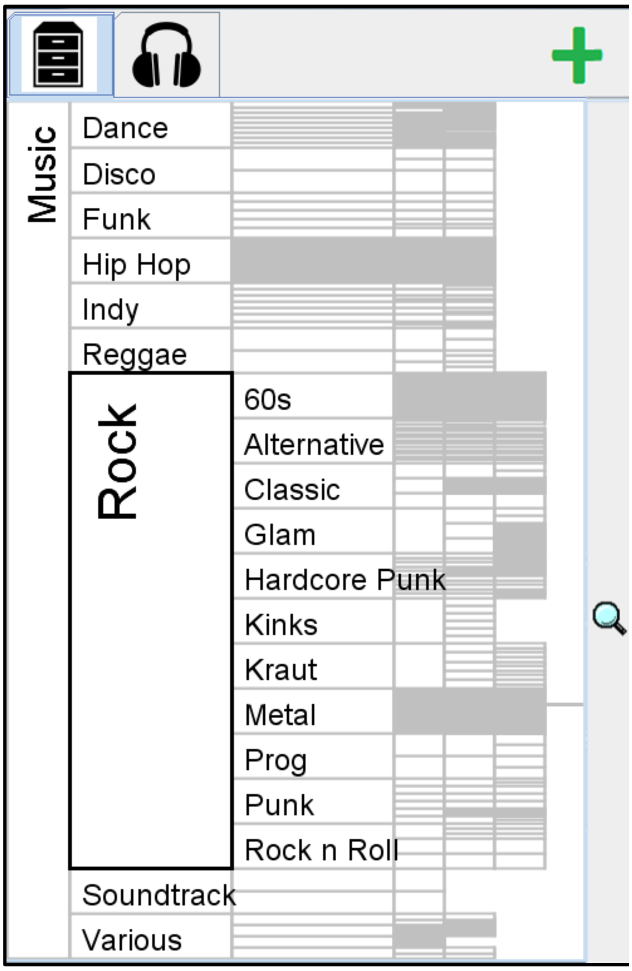


Fig. 6. The mobile tree browser interface. The selected focus node (60s) is outlined.

changes when different nodes are selected with priority vertical and horizontal spacing given to children of the selected focus node, then siblings, then siblings of any ancestor. This balances the user's need to immediately view interesting nodes while still being able to effectively navigate the hierarchy. The classification of different node types is illustrated in figure 5.

This paper looks at how we have adapted this layout to visualize hierarchies on a mobile device by making the visualization comply with our guidelines for mobile visualization design, as stated above. This includes the application of a fisheye distortion effect, allowing the user to view more nodes on-screen at the same time, and the provision of functionality that allows the user to switch between different modes to select multiple nodes.

III. Design

The design for our mobile tree browser (see figure 6) is based on the space filling layout initially developed for the Concept Relationship Editor application [29]. The modified design complies with each of the five guidelines for mobile information visualization design [18], set out at the beginning of the paper, as follows:

A. *Use techniques that make efficient use of screen space and do not require accurate selections.*

The Concept-relationship Editor layout makes efficient use of the available screen space by stacking labels horizontally or orientating them vertically when appropriate. Priority spacing is given to the nodes directly under the current selected node or nodes likely to be used for navigation up the hierarchy (i.e. ancestors of the selected node), across (siblings or uncles) or down the hierarchy (children).

In the mobile version of the layout, if there is not enough space to provide adequate height to all the nodes under the selected focus node(s), for either reading or selection, then fisheye distortion is used to magnify the labels closest to a movable magnifying glass icon. Here the height of each label is calculated using the original (undistorted) layout. For children of the selected focus node(s), siblings of the focus nodes and siblings of the focus node ancestors (see figure 7). The distortion is applied as follows.

To get SF_{lm}, the scaling factor for the label closest to the magnifying glass, we use the following equation.

$$SF_{lm} = \frac{h_i}{h_{lm}}$$

Where h_i is the ideal minimum height for a label (ensuring it can be clicked on and read) and h_{lm} is the original undistorted height of the label. If h_{lm} is already greater than h_i, then no distortion need be applied.

Next we calculate the scaled proximity of each label to the magnifying glass, YS_l. This is calculated using the following equation.

$$YS_l = \frac{y_l}{y_m}$$

If the label is above the magnifying glass, y_l and y_m are the distances from the label and the magnifying glass to the top of

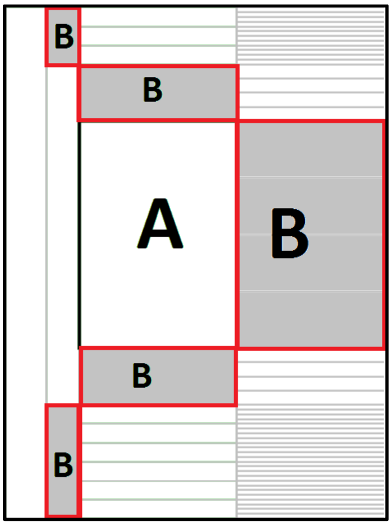


Fig 7. Layout with A) the selected focus node and B) nodes with distortion explicitly applied (i.e. children of the selected focus node(s), siblings of the focus node(s) and siblings of the focus node ancestors).

the panel. If the label is below the magnifying glass, y_l and y_m are the distances from the label and the magnifying glass to the bottom of the panel. This gives labels closest to the magnifying glass a value of 1 and labels furthest from the glass a value of 0.

Next we calculate the scaling factor for the height of each label, SF_l. This is given by the formula.

$$SF_l = SF_{lm} YS_l{}^{(SF_{lm}-1)}$$

This is applied to each label as follows so that the distorted height of the label h_{ld} is equal to its original height h_l multiplied by the scaling factor SF_l.

$$h_{ld} = h_l SF_l$$

This ensures that labels closest to magnifying glass are close to the ideal minimum height for labels and that all labels fit perfectly into the display space without the label closest to the magnifying glass moving (which would cause the focus of the glass to drift across labels). This is demonstrated in proof 1.

$$SF_l = SF_{lm} YS_l{}^{(SF_{lm}-1)}$$

$$\rightarrow \int_0^1 SF_l = {}_0^1\left[YS_l{}^{SF_{lm}}\right]$$

….. Approx. equivalent to the mean scaling factor above or below the magnifying glass.

$$= 1^{SF_{lm}} - 0^{SF_{lm}}$$

$$= 1$$

….. a mean scaling factor of 1 meaning that the changes in height due to distortion cancel each other out above and below the magnifying glass.

Proof 1. Showing that distorted labels fit into the same space as the original undistorted labels without the label closest to the magnifying glass moving.

In order to calculate the distorted height of the selected focus node and its ancestors (i.e. all nodes to the left of the gray area in figure 7) we simply sum the height of each of their children. To calculate the height of each remaining labels (i.e. labels to the right of the gray area) we evenly divide the height of parents between each of their children. Figure 8 shows the layout with distortion applied.

B. Keep text and selection targets above a constant device-specific minimum size and scale other elements to fit these constraints.

This requirement is also satisfied by our layout and distortion technique as an ideal minimum size is maintained for our text, specifically for the labels likely to be used for navigation, with all other elements scaled to fit this text. Text is also displayed with the optimal layout for scanning and we ensure that labels are big enough for inaccurate user selection on a small touch-screen.

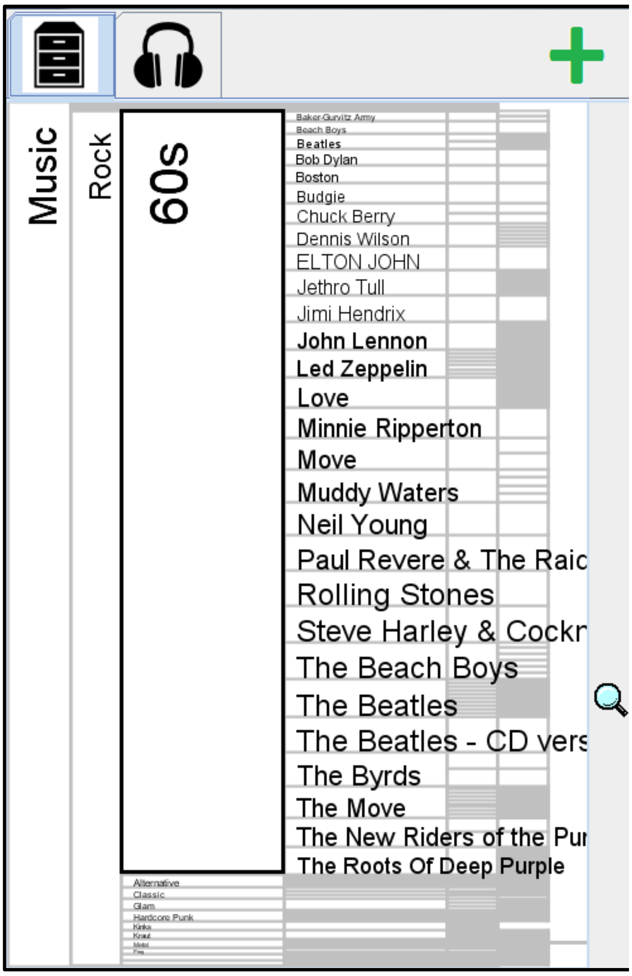


Fig. 8. The mobile tree browser interface with distortion applied over the label *The Beatles* in the selected node *60s*.

C. Don't display too much information on the screen at the same time. Allow the user to interact to view more data over time with animation used to smooth the transition between views.

Rather than attempting to display all the nodes at the same time, this visualization relies on user interaction that allows users to drill down into the hierarchy and view more of the data over time. Here the transition between views is animated with the positioning of labels interpolated between views. This helps the user to understand how the view changes according to their interaction and see the relationship between different views based on their selections.

For example, if the user is focused on the folder *Rock* then clicks on the folder *60s*, the *60s* folder will grow to take up vertical space previously occupied by other *rock* sub-genres such as *Alternative* and *Classic* with folders relating to various 60s rock artists gradually move into view. Throughout this process the different folders maintain their positioning relative to each other. This helps reinforce the user's conceptualization of the hierarchical data structure and gives them a better idea of how their interaction affects the state of the interface. If the effect of interaction was to be more sudden or abrupt this would make the interface less predictable and the user may not have such a good idea of how to reverse their actions or continue their interaction.

D. *Use virtual buttons to switch between different types of selection.*

Our interface includes a plus button that allows the user to select multiple folders at the same time. This action may be necessary if, for example, the user wishes to move songs between folders to create a custom playlist of their favorite music. This type of action is possible using multiple windows on a desktop pc, but having multiple windows open at the same time is impractical on a mobile device with limited screen space. Figure 9 shows our interface with two folders open at the same time.

E. *Don't allow important information to be hidden by the user's finger during interaction.*

The fish eye lens effect in our application is actioned by clicking and sliding a lens icon at the side of the screen. This allows us to differentiate between a press to select a label and a press and drag to move the lens. It also prevents the user's finger from obscuring important information while operating the lens.

IV. Evaluation

We evaluated the interface by comparing it with two alternative layouts, a traditional table view and a standard tree view. This evaluation involved twelve users who were asked to rate each interface from zero to six according to different metrics related to usability and effectiveness (see table 1). While the space filling layout achieved a relatively low score for learnability, since the users where already familiar with the other two representations, than the other two interfaces on all other counts. Our visualization was marked as considerably better for usability and legibility. Here users felt that the larger buttons for selecting nodes would be a clear advantage that differentiated our visualization from the other methods. The users were also impressed by the animation effect and became more positive as they used the interfaces and became more accustomed to the layout over time. They also felt that this interface was better at handling more items on a smaller display and that it gave them a better overview of the data. They felt that the interface was more usable and that this this type of interface would encourage them to interact with larger more challenging data-sets on mobile devices in the future.

TABLE I. User Ratings for Alternative Mobile Tree Browser

	Response Score (SD) [a]		
	Table View	Tree View	Mobile Tree Explorer
Functionality			
Efficiency			
Satisfaction			
Learnability			
Ease of Use			
Clarity			
Attractiveness			
Navigation			
Legibility			

[a] Responses are scored as follows: Awful=0, Very Poor=1, Poor=2, Moderate=3, Good=4, Very Good=5, Excellent=6. SD abbreviates standard deviation.

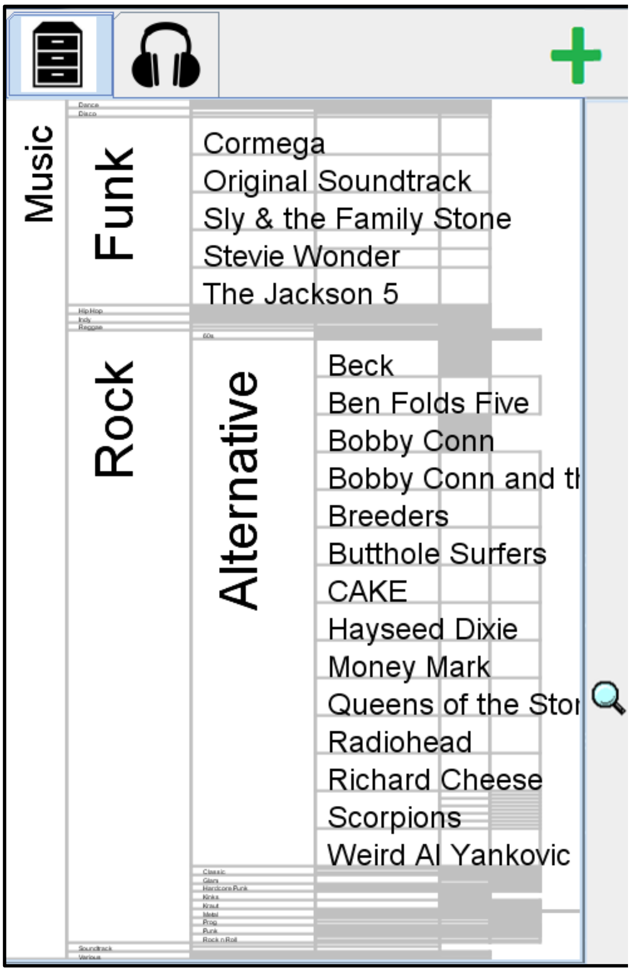


Fig. 9. The mobile tree browser interface with two different labels (*Funk* and *Alternative*) selected at the same tie.

V. Conclusion

We have developed and evaluated a new interface for visualizing hierarchies on a mobile device based on a set of draft guidelines for the development of mobile information visualization interfaces [18]. The technique uses an animated space-filling layout optimized for readability and navigation, and visualization techniques that can be applied with inaccurate touch-screen selection. This makes the display more interactive to allow the user to view more of the data over time without saturating the limited display space. Our evaluation showed us that the additional cognitive load of having to interact more with the data and having less of the data shown at any one time is reduced by using animation to smooth the transition between successive views. These results demonstrate the potential of information visualization to improve mobile device interface design. They also suggest that information visualization techniques can help facilitate the next generation of pervasive computing applications where increased functionality and unprecedented access to data and resources will be available through natural and usable mobile interfaces.

REFERENCES

[1] Lee, S. and S. Lee, Early diffusion of smartphones in OECD and BRICS countries: An examination of the effects of platform competition and indirect network effects. Telematics and Informatics, 2014. 31(3): p. 345-355.

[2] Chen, M., S. Mao, and Y. Liu, Big data: A survey. Mobile Networks and Applications, 2014. 19(2): p. 171-209.

[3] Chen, C.P. and C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. Information Sciences, 2014. 275: p. 314-347.

[4] Card, S.K., J.D. Mackinlay, and B. Shneiderman, eds. Readings in Information Visualization: Using Vision to Think. 1 ed. The Morgan Kaufmann Series in Interactive Technologies. 1999, Morgan Kaufmann: San Francisco. 686.

[5] Card, S.K., G.G. Robertson, and J.D. Mackinlay. The Information Visualizer, an Information Workspace. in ACM CHI. 1991. New Orleans, Louisiana, USA: ACM Press.

[6] Lenhart, A., et al., Social Media & Mobile Internet Use among Teens and Young Adults. Millennials. Pew Internet & American Life Project, 2010.

[7] Mahatanankoon, P., H.J. Wen, and B. Lim, Consumer-based m-commerce: exploring consumer perception of mobile applications. Computer standards & interfaces, 2005. 27(4): p. 347-357.

[8] Pikkarainen, T., et al., Consumer acceptance of online banking: an extension of the technology acceptance model. Internet research, 2004. 14(3): p. 224-235.

[9] Harrison, R., D. Flood, and D. Duce, Usability of mobile applications: literature review and rationale for a new usability model. Journal of Interaction Science, 2013. 1(1): p. 1-16.

[10] Huang, X., et al., SecIoT: A Security Framework for the Internet of Things. accepted to appear in Security and Communication Networks, 2015. 8(8).

[11] Sun, Y., et al., Advances on data, information, and knowledge in the internet of things. Personal and Ubiquitous Computing, 2014. 18(8): p. 1793-1795.

[12] Weber, R.H. and R. Weber, Internet of Things. 2010: Springer.

[13] Chung, K.-Y., Recent trends on convergence and ubiquitous computing. Personal and Ubiquitous Computing, 2014. 18(6): p. 1291-1293.

[14] Bashir, R.N., et al., Human Computer Interaction (HCI) in Ubiquitous Computing. International Journal of Innovation and Applied Studies, 2014. 9(2): p. 534-540.

[15] Roberts, J., et al., Visualization Beyond the Desktop-the next big thing. 2014.

[16] Bongshin, L., et al., Beyond Mouse and Keyboard: Expanding Design Considerations for Information Visualization Interactions. Visualization and Computer Graphics, IEEE Transactions on, 2012. 18(12): p. 2689-2698.

[17] Hao, J. and K. Zhang. A mobile interface for hierarchical information visualization and navigation. in Consumer Electronics, 2007. ISCE 2007. IEEE International Symposium on. 2007: IEEE.

[18] Craig, P., Interactive Animated Mobile Information Visualisation, in to appear at SIGRAPH ASIA. 2015: Kobe, Japan.

[19] Harrison, C., Appropriated interaction surfaces. Computer, 2010(6): p. 86-89.

[20] Chittaro, L., Visualizing information on mobile devices. Computer, 2006. 39(3): p. 40-45.

[21] Yusoff, N.a.M. and S.S. Salim, A systematic review of shared visualisation to achieve common ground. Journal of Visual Languages & Computing, 2015. 28(0): p. 83-99.

[22] Bertini, E., M. Rigamonti, and D. Lalanne, Extended Excentric Labeling. Computer Graphics Forum, 2009. 28(3): p. 927-934.

[23] Fekete, J. and C. Plaisant. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization. in Conference on Human Factors in Computing Systems. 1999. Pittsburgh, Pennsylvania, United States: ACM Press New York, NY, USA.

[24] Craig, P., N.R. Seiler, and A.D.O. Cervantes, Animated Geo-temporal Clusters for Exploratory Search in Event Data Document Collections, in Information Visualisation (IV), 2014 18th International Conference on. 2014, IEEE. p. 157-163.

[25] Karstens, B., M. Kreuseler, and H. Schumann. Visualization of complex structures on mobile handhelds. in Proc. International Workshop on Mobile Computing. 2003: Citeseer.

[26] Chhetri, A.P. and K. Zhang, Enhanced RELT for Visualizing Large Hierarchies on Handheld Devices.

[27] Hao, J., K. Zhang, and M.L. Huang, RELT–visualizing trees on mobile devices, in Advances in Visual Information Systems. 2007, Springer. p. 344-357.

[28] Craig, P. and N. Roa-Seïler, A Vertical Timeline Visualization for the Exploratory Analysis of Dialogue Data, in Information Visualisation. 2012: Montpellier, France. p. 68 - 73.

[29] Craig, P. and J. Kennedy. Concept Relationship Editor: A visual interface to support the assertion of synonymy relationships between taxonomic classifications. in Visualization and Data Analysis 2008. 2008. San Jose, CA: Society of Photo-Optical Instrumentation Engineers, Bellingham, WA.