Using Grid Visualization to organize visual data

Yan Xu[1], Florian Perteneder[1], Joanne Leong[2], Eva-Maria Schwaiger[1], Michael Haller[1]

[1]Media Interaction Lab
Upper Austria University of Applied Sciences
mi-lab@fh-hagenberg.at

[2]Systems Design Engineering
University of Waterloo
jslleong@uwaterloo.ca

ABSTRACT

This paper introduces an alternative novel visualization concept to organize visual data. Our design has been guided by the goal to make information visible with uniform snapshots. After a detailed description of design implications, our interaction model is fully demonstrated within a file management prototype that utilizes the new visualization. Results from an initial evaluation proved the Grid Visualization concept offers a high degree of flexibility and provides an interesting alternative to traditional hierarchical tree-structured file and folder systems.

Author Keywords

Creativity, Design, Overview + Detail, Interaction Techniques, Interactive Visualization

ACM Classification Keywords

H.1.2 Information Systems: User/Machine Systems –Human factors. H.5.2 Information Interfaces and Presentation – User Interfaces.

General Terms

Human Factors; Design; Measurement.

INTRODUCTION

The information age we are living in brings great changes to our daily lives. Paperless work results in more rapid, flexible, and reliable transmission of information. However, first and foremost people's work involves a growing number of versatile types of digital documents. The digital information is stored on various devices or cloud servers. For users, it is becoming less important where the files are actually located within the file system as long as they can be easily organized and accessed. This trend became most obvious with the introduction of the first smartphones where it was no longer necessary to deal directly with the file system by using a file browser. However, in traditional desktop computing, the frontend file system acts as a significant bridge between physical data storage and users; it presents where files are stored and retrieved. There are several kinds of possible structures but the most common directory structures in use today are hierarchical tree-structured directories. From a user's perspective the files are separated and identified mainly by file names and icons; folders are typically used to group relevant files into separate collections (cf. Figure 1).

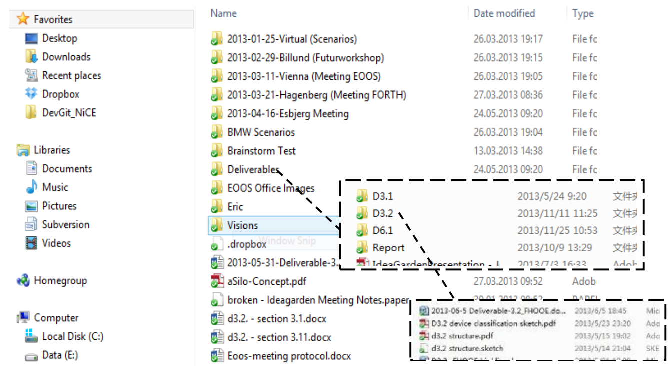

Figure 1: A tree-structured directory contains files and folders on multiple levels. This causes large amounts of content to remain hidden in subdirectories.

The strength of using a hierarchical tree structure is that they are clean, ordered and well-known. However, the structure relies heavily on file names which are usually specified by users. In the absence of strict naming conventions, an incorrect or ambiguous name would fail to remind users what the file is about. Furthermore, if the file system is not well maintained, users can easily become lost in huge tree structures and unaware of certain files or file locations. Rather, they may have an indistinct memory of such content. This issue may become more serious when collaborating with other people on a project, since some files are placed and named by others. As illustrated in Figure 1, only one level of subdirectories or contained files is visible at once. The content of subdirectories is invisible before entering it. Sophisticated search mechanisms, such as Spotlight, aim to improve this situation. However, while these approaches make the current system more useable, they rely heavily on textual information and undermine the need to upkeep a tidy organization of files or assets.

Summarizing, we conclude that the current hierarchical tree-based file system works effectively for digital documents which can be well described by text or includes searchable textual information. However, many professions that rely on the use of digital documents work with different kinds of data that are much more visual, and therefore harder to search through, organize and sort. Designers or photographers for instance heavily rely on images, sketches or other visual information snippets that are supposed to convey a fuzzy idea that is often hard to describe in words. Our goal is to find an adequate visualization method that supports work involving visual digital information.

Therefore, we introduce a concept called *Grid Visualization*. The content is displayed on a non-hierarchical layout as depicted in Figure 2. This file structure visualization provides similar interactions as traditional file browsers (e.g.: grouping, rearranging, duplicating,..).

Figure 2: The Grid Visualization with different file types (images, sketches, pdfs, etc.). The assets are grouped into distinct categories (islands) by placing them next to each other.

In this paper, we address a number of critical considerations for designing a tool that utilizes this visual method of working with files. Since the asset visualization and the general interaction are very important aspects of such a concept, we discuss the pros and cons of our design choices. Finally, we present our prototypical implementation and the results of a preliminary user study.

RELATED WORK

One of the major purposes of information visualization is gaining insight about data. It is about harnessing human's remarkable visual perception capabilities to help identify trends, patterns, and unusual occurrences in datasets [6].

Various methods have been developed to display information. The traditional node-link diagrams connect nodes with line segments in Euclidean space or hyperbolic space. They represent the branches and leaves of a tree [5]. However, compared with space-filling visualizations, node-link diagrams consume a lot of space and fail to give an effective overview of large datasets [2]. Treemaps make efficient use of the available display space by partitioning data into a collection of rectangular bounding boxes representing the tree structure [2]. The drawback of rectangular treemaps as well as radial space-filling visualizations [1] is that the hierarchical structures are hard to discern. Zhao et al. explored combinations of node-link and treemap forms to develop elastic hierarchies for representing trees [6]. However, this concept still does not maximize the use of space. Wang et al. [5] presents a space-filling approach for tree visualization that combines treemaps and non-intersecting Venn-diagrams (also known as Grokker visualization). Categories and subcategories are displays in nested circles. The drawback of this concept is that the varying sizes of closely spaced or nested circles make the interface look a little disorderly.

As humans have a strong spatial memory [3], the visual representation that frames the asset visualization is essential when aiming for support of file organization. Instead of relying on file names, the Grid Visualization concept leverages thumbnails and spatial location to help users manage and recall digital data assets.

CONCEPT

The spatial organization of information assets plays an important role, especially in visual and creative design processes. Vyas et al. [4] pointed out that designers use space to communicate, gather ideas, investigate design solutions, and manage design projects. **Error! Reference source not found.** represents an approach to lay out information in space to gain an overview.

Figure 3: Overview map of a design project created by a design student.

The *Grid Visualization* takes inspiration from this approach. It uses a tiled canvas that serves as an infinite but structured space to lay out and manage visual information. As depicted in Figure 2, each tile of the grid is used to represent *one* file or information asset. This can be an image, a sketch, a written note, a document etc. By placing them next to each other, assets can be categorized and structured. The spatial proximity of assets is a visual metaphor for their relationship or relevance to one another. The visual grid layout is used to view, search, organize, and manage information content (see Figure 2). It greatly reduces the dependence on file names, and presents the content to the users in a way that utilizes people's spatial memory.

DESIGN IMPLICATIONS

Based on the considerations above, we deducted a number of design implications that should be considered when designing an interface for visual representations of abstract data.

- **Visual Representation of Information Assets:** How assets are represented visually is critical to the design of the system; it greatly influences other aspects such as *navigation, structuring capabilities, the number of* *manageable assets*, *up-to-date file overview*, and *uniformity*.

- **Support of Different Granularities:** Digital information regarding a project or a certain context can be understood at different levels of granularity. It can be helpful to have an up-to-date overview; at the same time, it may be necessary to be able to work on a sub-topic level.
- **Lifecycle of Information Assets:** For long-term storage it is important to provide options to support the entire lifecycle of information assets; they can be *created*, *maintained* and *refined*, and *deleted* or *substituted*.
- **Unification of Perception**
In the context of collaborative work, it is beneficial to provide a way to exchange assets. A visual workflow can create opportunities to discuss and unify different perceptions of a problem.
- **Integration and Support of External Applications:** A large variety of specialized applications is used to create and refine assets. Similarly to most common file browsers, the interface should support a workflow that accommodates assets created with these specialized applications.

Visual Representation Concepts

The visual representation of information assets is the most crucial part of the entire system, as the goal is to support the users with a visual map of the up-to-date state. Therefore, in this paper, we focus mainly on this issue and present the Grid Visualization (see Figure 4) as a possible solution.

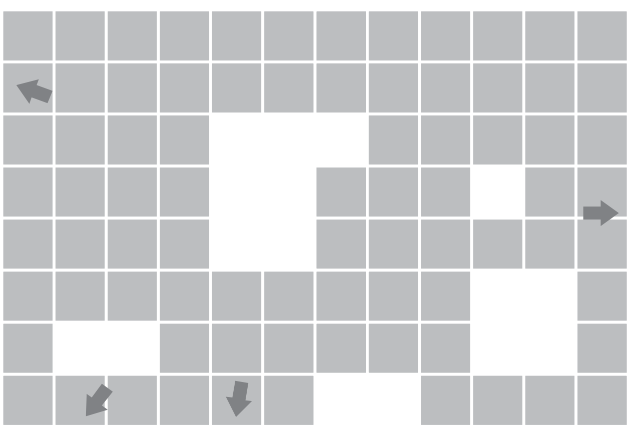

Figure 4: Concepts of Visual Representation for Information Assets: Grid Visualization

Grid Visualization

The Grid Visualization provides users with a single canvas on which they can collect, organize, and share files and visual information assets. The idea is that all information is placed on a canvas, which then represents the current big picture of files. Users can freely navigate on this map. To support the organization of information, the visualization allows users to structure assets within pre-defined slots. Assets adjacent to one other are considered to form a group. Groups that are off-screen are displayed as a label at the border of the screen. These labels can also be used to directly navigate to the corresponding group.

Visual Representation Discussion

In this section, we discuss the benefits and limitations of the Grid Visualization concept.

Navigation

The Grid Visualization provides a very simple navigation concept as it restricts free navigation to two dimensions. This restriction ensures that, while assets can be off-screen, there are no hidden views on different layers.

Structuring

The simple structure allows for easy rearrangement and automated grouping of assets, using spatial proximity as an indicator for relation. However, there is no support for hierarchical structures, which changes the way of how relationships between assets can be expressed.

Number of Manageable Assets

Theoretically, the Grid Visualization can handle an infinite number of assets. However, due to its limited structuring capabilities, maintaining order and establishing an effective overview may be difficult when the number of assets grows too large.

Overview

Due to its non-hierarchical structure, the Grid Visualization inherently provides an overview of the displayed files. The challenge is to keep large numbers of assets accessible and easily manageable. However, as managing multiple hierarchies increases the workload and puts the overall visualization at greater risk of becoming outdated, a flat hierarchy is beneficial for attaining and keeping an overview.

Uniformity

The size of assets often correlates with their perceived significance, as larger assets have the tendency to attract more attention. Especially with regards to visual assets, uniformity can mitigate the possibility of overlooking smaller sized representations. The Grid Visualization inherently supports asset uniformity by its design.

Support of different granularities

When aiming for providing universal support for files, the earlier mentioned issue of granularity becomes critical. Due to their deep hierarchical structure, tree-structured directories are capable of displaying multiple granularity levels. Making users responsible for both spatial management and the distribution of content on multiple levels, however, leads to a lot of overhead and could result in outdated structures. For the Grid Visualization, the flat hierarchy makes mapping different levels much more difficult, but reduces the maintenance effort.

PROTOTYPE

Due to the strict and limited organizational structure, the Grid Visualization allows the rearrangement of assets and the modification of the overall composition to be done with ease.

Hence, the effort needed to maintain existing structures is reduced. This can be critical for ensuring the adoption and integration of this concept in creative workflows, which are often very visual in nature.

We also face the situation that there is a high number of very specialized tools and applications that serve different purposes on multiple levels of granularity. Supporting all these levels competently within one visualization concept is nearly impossible to achieve. Furthermore, creating a visualization concept which attempts to do this would inhibit users from picking tools which have been optimized or specially designed for particular tasks or problems. Therefore, we think it is better to provide a thinner layer that covers the high-level support needed to provide an up-to-date overview of files, and to closely integrate other applications to afford users a variety of options for tackling versatile problems.

Technical Implementation

In order to verify the utility of the concept, identify usability issues, and gauge the potential for supporting collaboration, we decided to develop a file management application that utilizes the Grid Visualization concept. The prototype was developed using WPF and C#. It was designed to be used with a pen or stylus on a digital whiteboard or tablet, but it can also be used on a traditional desktop setup.

General Concept

Following the *Grid Visualization* idea, we provide users with a canvas on which they can collect and organize all the assets that belong to a certain project.

The aggregation of these assets provides a current snapshot of a project. Users can freely navigate on this map. There are two distinct zoom-levels - one that provides an overview, and another one that offers more detail. Files or assets are represented by uniform tiles that are placed on a grid. Users can freely arrange these tiles. To support and simplify the organization of information, the grid-based visualization allows users to structure assets within pre-defined slots. When moved to a different location, an asset will snap into place. If the destination slot is filled, the asset existing in this slot shifts aside to make space.

The grid visualization leverages the concept of proximity and spatial organization as a way to state the relationships between information assets. Therefore, tiles can be arranged to stick adjacent to one another to form *islands* (orderly spatial groupings).. In addition, displaying assets in a consistent spatial organization should facilitate easy access and referencing.

Islands (groups)

Tiles can be positioned on a grid to form clusters, or islands (see Figure 4). Their placement next to each other suggests a close relationship, while space in between suggests a semantic distance. This visual syntax can be used quite simply to describe the relationship between various assets and also groups of assets. As soon as two assets are placed next to each other, an island is automatically formed, and a preliminary island title appears on the top of the newly formed group. A border around the island confines it visually. These two characteristics indicate the topic of an island and distinguish it from other islands. In addition, the border and the title serve as a handle to easily drag and drop entire islands to different locations.

Figure 5: one island containing three PDF documents

The concept of islands provides a great degree of flexibility; there are several possibilities for the spatial arrangement of tiles and islands with respect to one another. Additionally, since humans have a strong spatial memory [3] and the concept of spatial organization is fundamental to design practice [4], arranging assets manually and deliberately should give users better content awareness than automatic placement (cf. file browser).

Asset Preview

Considering that the size of tiles may not be big enough for users to recognize content (especially for documents with a large number of tiny words and multiple pages) the system provides users with a preview (see Figure 5).

Figure 6: Preview of a PDF document. The user can navigate through the PDF without explicitly opening the document within a PDF reader.

Island Links

As mentioned before, users are operating on an infinitely large canvas. However, the size of the physical screen is limited. Hence, if there is a huge amount of assets and islands, some can end up off-screen. Islands that are off-screen are displayed as labels along the perimeter of the screen (see Figure 7). These labels point to the position of the off-screen islands, indicating their placement relative to the current viewing position. Additionally, the distance between the island and the current viewport is indicated by the label's level of transparency - the closer the island, the more opaque the label. These hints are supposed to promote off-screen objects, and ensure that important islands are not overlooked. Finally, the island links can also be used for navigation to jump to the corresponding group.

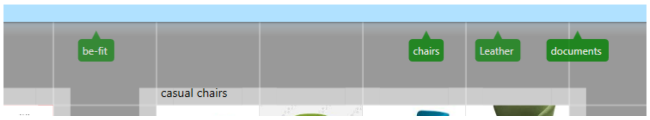

Figure 7: Four off-screen island labels positioned along the top border of the view

INTERACTION

The prototype uses two menus: the *Creation Menu* and the *Context Menu*. These are triggered directly on the grid and are associated with the tile they were invoked with.

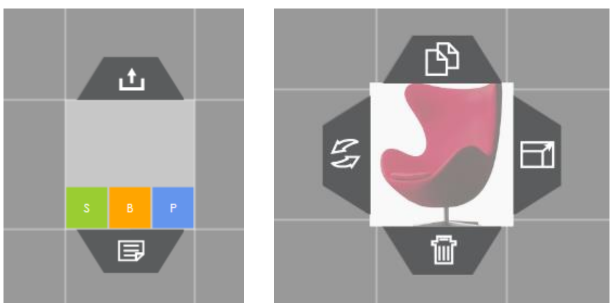

Figure 8: Creation Menu (left) and Context Menu (right)

Creation Menu

By tapping on an empty tile, users invoke the *Creation Menu* (see Figure 7, left). This menu provides users with three main options: to either *import a file*, *create a note asset*, or *start-up an external application*. After creating or importing a new asset, the menu appears at the tile location that was selected. This approach helps to streamline the creational process by allowing users to create assets directly on the project board in a continuous workflow.

Context Menu

If the user invokes a menu on top of an existing asset, the *Context Menu* appears which offers the user the option to *duplicate*, *enlarge*, *substitute* or *delete* the selected asset (see Figure 7, right). Substitution allows for replacing an asset with a more-up-to date version. Users can also open the asset with a corresponding external application. This mechanism ensures a fluid workflow whereby users can easily edit assets produced with specialized applications.

USER FEEDBACK

We conducted an informal user study, where we asked 8 students (6 male) aged from 24 to 29 (M=26.5, SD=1.60), to get a better understanding of and initial feedback on the developed system. While the efficiency of the interface was investigated, the primary aim of the study was to learn more about the general applicability of the Grid Visualization concept and whether this approach could support users in their work. All participants use computers on a daily basis. For the user study, the prototype was used on an interactive pen-based whiteboard (see Figure 9).

First, the participants were introduced to the general functionality of the prototype in comparison with a traditional file browser. They were given an explanation about the visualization approach and the integrated interaction techniques. Next, participants were invited to try the prototype in order to get a deeper impression of it. We asked them to perform small tasks, like gathering all blue chairs and grouping them into an island. After that, the participants answered interview questions and expressed their opinions about certain features.

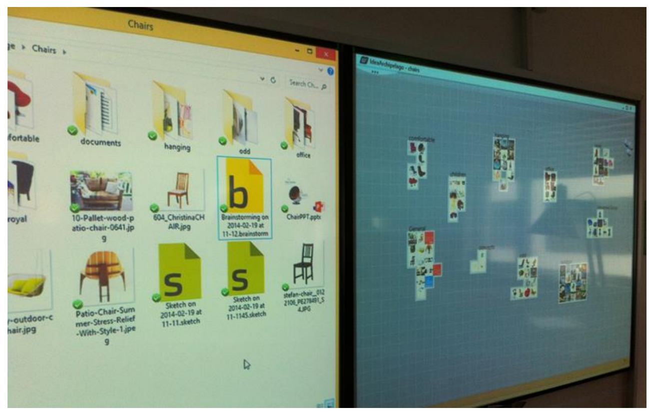

Figure 9: The prototype whiteboard setup (the left board displays a file browser, the right one the Grid Visualization).

Interface and Grid Visualization impressions

According to our observation and the verbal feedback, the Grid Visualization was rated as a good approach in displaying visual information assets. Six participants stated that the grid provides a good overview and therefore helps to easily find files, especially pictures. Two participants mentioned that they liked the fast file rearrangement possibilities due to the nonhierarchical structure, and that it provides fast editing options for specific files. Two other participants mentioned that they think that this visualization might be good for presentations, since relevant data can easily be grouped. All participants were interested in the prototype and contributed further ideas for development and improvement. Moreover, they asked for additional features, such as total zoom out and color encoding.

However, some participants were skeptical about the thumbnail preview for text documents, since it is difficult to recognize them without the document name.

Do users need to know about the type of a document?

The discussion about the thumbnail preview led to a debate about whether it is necessary to know the file type of an asset.

Five participants mentioned that in their opinion this depends heavily on the use case. They did not consider it as important for images, since in that case they care more about the content. But for other assets, it might be highly beneficial to know the file type. They claimed that this could also be helpful to identify the relations between assets. Ideally an asset is recognizable just by its visual representation. The participants agreed that textual files require better representation. Two participants suggested using a small logo or different colors to mark the types of files. One participant even raised the need to know the metadata of a file, such as size, creation, and last editing date.

Island Concept

During the testing phase our participants consciously gathered relevant data to form islands and leveraged the ability to spatially arrange things in order to indicate the relationships between assets. In addition, related islands were created next to each other. The island concept was considered as being very helpful, as they saw similarities to ordinary file folders. Nevertheless, two participants stated that the island concept is missing the options to formulate a deeper hierarchy, similar to what folders in a file browser can provide.

File Overview

All the participants stated that the overview is helpful to form a "*big picture*" of files and quickly acquaint users with the components. They liked that they could have all assets in their field of view, and that files are not hidden in a deeper file structure. One participant stated that the flat hierarchy saves time, because there is no step into a folder or a step back out. Another participant said that the visual search approach works better for him for searching visual assets than the file browser, if there is a clear structure maintained within the grid. However, with a growing number of assets it can easily happen that single files are overlooked and get lost.

Number of Assets

All participants raised concerns about a huge number of assets. They stated that it would be hard to find single assets, if the grid contains a large number of assets. A prepared canvas was shown to all participants, which included about 400 assets and 37 islands. None of the participants had the feeling that this was an overwhelming number of assets; they all still had the feeling that they had a good overview. Only one participant was concerned about losing track of the off-screen islands.

DISCUSSION

Although we performed only a small, informal study on the Grid Visualization, the feedback from the participants and the lessons learned during the implementation and testing of the prototype allows for a preliminary summery of the lessons learned.

The decision to implement a concept with a flat hierarchy had probably the most impact on the results. The feedback was very ambiguous. On the one hand, participants complimented the simplicity regarding searching and gaining an overview; on the other hand, some explicitly requested the integration of hierarchies. We believe that this is a design choice that has to be made and that it is hard to find a compromise in this case. However, there should be caution in regarding the use of the Grid Visualization as an overarching alternative to the hierarchical tree-structured file system. When we reduce the scope of application, i.e. to the organization of the assets affiliated with a concrete project, the lack of hierarchical structuring capabilities is less critical as the number of assets on the grid drops. This approach could also resolve some of the concerns regarding the number of assets. However, from our current observations we believe that this issue is less critical than the first impression suggests. The grid is capable of holding a vast number of assets, and as all of our participants confirmed, people tend to underestimate the number of assets when they see them on the grid.

Regarding the issue of providing information on the file types, there might be also ambivalent. We think that people do not necessarily need to know the type of a file if they are focusing on the content and the right application is launched as soon as they decide to work on it. However, it seems that users feel they have more control over what they are doing when they know the file type and that at least our participants, who have a strong computational background, are not willing to give up this perceived control.

CONCLUSION AND FUTURE WORK

Working with files and folders is a daily activity for people who do computational work. File handling on mobile devices, where it is hardly ever necessary to use a file browser, shows that there would be ways to overcome the need of interacting on a file and folder level.

Therefore, we proposed an alternative way of organizing and working with files or information assets, called *Grid Visualization*. This approach is especially suited for visual data as it enables users to construct a visual overview of files. Users are responsible for the creation, placement, and arrangement of file assets, ensuring that the overall visualization is kept up-to-date and well organized. This requires additional effort from the users. However, the application can be designed to mitigate this issue by encompassing clever interaction features that reduce these costs.

According to the participants' feedback, our system provides benefits in helping users organize and gain an overview of their files. Overall, the results from our study indicate that the Grid Visualization concept is promising, but needs further exploration in diverse real-world scenarios.

ACKNOWLEGEMENTS

The presented research has received funding from the European Union's Seventh Framework Programme FP7/2007-2001 under grant agreement n°318552. We would like to thank our colleagues for their helpful feedback.

REFERENCES

1. Andrews, K., Heidegger, H.: Information Slices: Visualising and Exploring Large Hierarchies using Cascading, Semi-Circular Discs (Late Breaking Hot Topic Paper). In: Proc. of the IEEE Symposium on Information Visualization (INFOVIS 1998), Research Triangle Park, NC, pp. 9 – 12 (1998)
2. Johnson, B., Shneiderman, B. Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information. In Proc. Visualization 1991, IEEE (1991), 284–291
3. Scarr, J., Cockburn, A., Gutwin, C., and Malacria, S. Testing the robustness and performance of spatially consistent interfaces. In *Proc. CHI 2013*, ACM Press (2013), 3139-3148.
4. Vyas D., Veer G.V., Heylen D., Nijholt A.: Space as a Resource in Creative Design Practices. In *Proc. INTERACT 2009*, Springer (2009), 169-172.
5. Wang, W., Wang, H., Dai, G., and Wang, H. Visualization of large hierarchical data by circle packing. In *Proc. CHI '06*, ACM Press (2006), 517-520.
6. Yi, J. S., Kang, Y., Stasko, J. T., Jacko, J. A. Understanding and characterizing insights: how do people gain insights using information visualization?. *In Proceedings of the 2008 Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV '08).* ACM, New York, NY, USA, Article 4, 6 pages.
7. Zhao, S., McGuffin, M. J., Chignell, M. H. Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams. In John T. Stasko & Matthew O. Ward, ed., 'INFOVIS' , IEEE Computer Society, pp. 8