EXPLORING THE EFFECTIVENESS OF 3D FILE BROWSING TECHNIQUES FOR FILE SEARCHING TASKS

by

ANAMARY LEAL

A thesis submitted in partial fulfillment of the requirements
for the Honors in the Major Program in Computer Science
in the College of Engineering and Computer Science
and in the Burnett Honors College
at the University of Central Florida
Orlando, Florida

Spring Term 2009

Thesis Chair: Dr. Joseph J. LaViola Jr.

Abstract

Based on an analysis of the existing literature, we extracted important features regarding 3D file organization and layout. In this way, three separate 3D file browsing techniques were evaluated in a comparison study. Block3D uses a priority weighting scheme to elevate and display files in a grid-based structure. Cluster3D uses sets of animated racks to display files. TreeCubePlus3D visualizes files and directories using groups of semi-transparent cubes within a larger cube-like structure. Across all techniques, each file was represented as a thumbnail image of the file's image, document or document with pictures data. The thumbnails were also augmented with meta-information such as filename and relevancy information to simulate a realistic search.

Our experiment explores the effectiveness of each 3D file browsing technique in a manual file searching task. Our evaluation is based on task completion time and a post-questionnaire used to gather subjective feedback on each technique in terms of user preference. The results indicate that users completed the manual file search task significantly faster using Block3D than both TreeCubePlus3D and Cluster3D. Subjective ranking also showed users preferred the Block3D technique.

Acknowledgments

There are many people that contributed so much for this work. First, my parents, for their encouragement, even if they didn't fully understand exactly what I was doing. The invaluable guidance and wisdom of my faculty advisor, Dr. Joseph LaViola, made this work possible. I also thank members of my committee for their ideas and contributions to my thesis.

I had so much support from the RAMP Scholars Program and the Honors in the Major Program from the Burnett Honors College, both at the University of Central Florida. This research was funded by both those organizations, and both provided a network of supporters of undergraduate research. I wouldn't have even been involved in undergraduate research if it weren't for these organizations.

Thanks to all my friends who were understanding and supportive enough to deal with me during this process, especially my big sister Diane, who always believed that I could finally finish this thesis, and whose encouragement helped me keep on working.

Finally, I absolutely could not do this work without the support and encouragement of my best friend and boyfriend, Christopher. He dealt with me staying up late, immensely stressed, listening to my research rants as boring as they may be, and encouraging me throughout the entire process. This could not be possible without you!

Contents

List of Tables

List of Figures

1. Introduction

With better and 3D-capable graphics processors becoming commonplace, the prospect of having futuristic 3D interfaces seems promising. Moving from the traditional windows, icons, menus, point and click (WIMP) [5] 2D style of interaction to the 3D desktop metaphor seems daunting, especially with managing hundreds of files within the same directory. Microsoft's shift from the 2D user interface (UI) found in Windows XP to a 3D UI in Windows Vista and the development of 3D UI desktop environments such as Sun Microsystem's Project Looking Glass [14] and Duan Hamar's SphereXP [10] indicate an interest in this move. However, the usefulness of 3D user interfaces for a general-purpose use is questioned in the research area of 3D User Interface. There is research affirming the benefits of 3D UI's [19] and other research that questions whether 3D user interfaces are useful in file organization and layout, noting that the 3rd dimension made subjects more confused in finding the appropriate files [7].

1.1. Motivation

Because of the debate over the benefits of 3D UI's, our goal is to make a contribution to the field by evaluating features specific to 3D UI's. There are many interesting 3D file browsing features, from audio cues [18], rotational movement in the environment [23], to placing files under or inside other files [10, 19, 21]. By finding optimal features, we can, in turn, compare an optimal 3D UI with an optimal, traditional WIMP-based 2D UI. Instead of comparing 3D versions of 2D ways of arranging files in a UI like previous 2D versus 3D studies[7, 22] or proposing a new 3D visualization technique alone, this research evaluates dissimilar visual arrangement techniques that are exclusively in 3D. In addition, through these experiments, we will reveal the utility and values for each visual arrangement approach.

A specific issue with respect to the trends in 3D UI development is a potential difficulty in using 2D files made from 2D interfaces. In a 3D spatial environment, concerns may arise about how such data like documents, images, files and folders, can best be imported and organized in a 3D environment. In particular, documents and images portray different kinds of data, and neither have the 3rd dimension to portray such data. Documents may visually vary by containing tables, paragraphs, lists, and other literary elements, but images range not only in the subject matter, but in color. Two images of an identical blue ball on grass can differ greatly due to camera angle, light exposure, emphasis on specific color channels, and other graphical elements. 2D files can be plastered on walls or stacked such as arrangements in the physical world, but we suspect that there are more novel, useful ways that arrange these 2D files and take advantage of the added degree of freedom.

1.2. Statement of Research Question

The goal of this research is to evaluate general features prevalent in 3D browsing techniques by evaluating 3D browsing techniques that span those features, specifically with 2D files like documents and images. By finding the usability benefits and drawbacks of each technique, we can evaluate the strengths and weaknesses of specific features. Ideal features would contribute to a better user understanding of file organization and layout using 2D files. This work will contribute to the discussion of 3D UI by finding characteristics of an optimal 3D user interface. This work will study the strengths and weaknesses of 3 existing 3D browsing algorithms with spanning features through a user study. A literature review of current 2D and 3D user interfaces was compiled, and the features between UI's were compared and highlighted. The most novel 3D browsing techniques were selected for evaluation, in the form of a user study with qualitative and quantitative measures.

1.3. Key Concepts

Throughout this document, we user several terminology with meanings specifically for this work. Other works may use different terminology, but for the purposes of this work and for clarity, the following important terms will be used.

1.3. 3D User Interfaces

A 3D user interface is an interface with a computer in which the interaction uses 3D data. This may be as simple as moving the forward button on a keyboard to move forward in a 3D game, or wearing gloves that process 3D movement[5].

1.3. 3D File Browsing Techniques

A variety of types of interaction exists depending on the task involved. There are navigation, wayfinding, selection, and more types of 3D interaction techniques [5]. 3D browsing techniques are interaction techniques revolved on arranging and selecting files in a 3D environments.

1.3. Logical Arrangement Techniques

3D browsing techniques require an algorithm that properly arranges a collection of files and directories within a given visual layout, defined as logical arrangement techniques. The visual organization techniques organize in 3D space, while the logical organization techniques order the files by ranking them according to some measurement of context, depending on the logical arrangement technique. A visual arrangement technique can be as simple as setting a stack of papers, but the logical arrangement technique can be sorting the papers by file name. The logical arrangement techniques serve as input for the visual organization techniques.

1.4. Thesis Overview

The thesis is structured in the following chapters:

Chapter 1 — Introduces the topic, the importance and promise of 3D UI's, key concepts, and research questions.

Chapter 2 — Assesses and surveys the literature review on the subject. This section will analyze relevant evaluations within 3D/2D UI's, survey presented 3D browsing techniques, and derive common characteristics between them.

Chapter 3 — Describes 3 common traits among 3D browsing techniques, and 3 3D browsing techniques that span those traits.

Chapter 4 — Details the experiment procedure to evaluate the features and the selected 3D browsing techniques.

Chapter 5 — Summarizes the qualitative and quantitative results from the evaluation.

Chapter 6 — Assesses the results, develop future conjectures and future work, and concludes on the work.

2. Related Work

2.1. 3D Browsing Techniques

There have been several techniques developed for 3D file browsing. Some 3D layout techniques have been used with respect to users interacting with applications. Robertson et. al 's Data Mountain placed web pages in a uphill mountain, with less visited pages layered behind more frequently viewed pages [18]. Microsoft's Task Gallery placed applications in a series of rooms, with multiple applications stacked on the side, ceiling, or the floor of a room, and the application of focus in the center [19]. Hamar developed SphereXP, a commercial desktop replacement software that manages applications through 3D file techniques. This software incorporates rotating individual objects to face the viewer in any direction [10].

Most 3D file browsing techniques use document files instead of applications. Tanaka's TreeCube3D is a 2D version of Shneiderman's TreeMap [20] that renders all files simultaneously with semi-transparent blocks representing nested folders [21]. Baumgartner 's file browsing technique combines a 2D interface with a 3D grid-like interface for using two different display approaches [3]. Tactile3D is a commercial 3D desktop environment that renders folders in nested sphere-like objects [12]. Zhua and Chen developed a geo-referenced 3D interface that uses topological data such as terrain to position documents [23]. Dengel et. al.[8] developed a variety of 3D file browsing techniques. For example, PlaneMode places all files in a plane orthogonal to the camera's view where each file's depth changes depending on a search query. ClusterMode is another example that groups similar files together using a variation of PlaneMode [8].

2.1. Zoom-In Interfaces (ZUI)

Zoomable user interfaces (ZUI) have used elements in 3D user interfaces to portray data. Pad++ renders the most hightly rated information as the largest and closest objects in the area, while rendering lesser rated information smaller. Relevant information, whether highly or lesser rated, would be closer together [4]. Focus Plus Context techniques describe a variety of ways to focus on an area of data, but still show nearby data, such as the fisheye lens that renders nearby objects with detail, and further objects with lesser detail[9], or non-linear magnification transformations [13]. Our work does not explicitly emphasize zooming to present further information, but techniques from ZUIs can be applied most presented 3D file techniques.

2.1. Logical Arrangement Techniques

3D spatial arrangement techniques require an algorithm that properly arranges a collection of files and directories within a given visual layout. The literature review has found several logical arrangement techniques relevant to the 3D browsing techniques. ClusterMode used a logical arrangement technique in which required the user to enter 4 items in 4 search queries. Each search query was represented by a unique color, and this aided its visual arrangement technique by generating its colored lines between groups [8]. ClusterMode requires the most interaction from the user, since no files would be arranged without the search queries, and uses file The logical arrangement technique used by the Geo-Referenced Knowledge Repository technique implements a Self-Organizing Map with respect to content of the files. The algorithm generates a list of phrases relevant to a file, and arranges the files according to how close the content is in relation to the file's neighbors [18]. The technique requires no input from the user, and also uses file content as a measure of context. FolderPredictor, the last logical arrangement technique, interprets context with respect to frequency of use and file access/saves/recency of use. Each folder and file is assigned weights with relation to other files, and the FolderPredictor algorithm would use the weights to find files the user is most likely to open. This algorithm, developed by Bao et. al., was originally tested under Windows Explorer, and their user studies found that most users follow the tree structure and not use the predicting algorithm [2]. However, with this technique, we consider contextual cues outside of file content alone compared to other techniques. Also, the FolderPredictor algorithm will be implemented to predict the next file opened within a folder only, not including a child folder's children. This study may also affirm or disprove the implications of Bao et. al.'s user studies by making this adjustment.

2.2. 3D File Browsing Evaluations

Two different types of comparisons are conducted with 3D file browsing techniques. The first is comparing the 3D technique with a 2D counterpart, and the results in this area have been mixed. Risden et. al. compared two 2D techniques against one 3D file browsing for accessing existing and added web content. The 2D file browsers were a collapsible tree browser and a category based browser, while the 3D visualization file browser was a hyperbolic tree graph structure, called 3DXML. 3DXML was faster than the 2D browsers when interacting with existing data, but lost its advantage when managing new or added categorical data. The 2D interfaces also had a slight user preference towards the 2D interfaces, but the results were not significant [17]. Robertson et. al's study on the Data Mountain allowed for manual placing of files to form a memory map [18]. The results found performance and user preference data favoring Data Mountain over the 2D Internet Explorer 4 [18]. Zhua and Chen's study on 3D file browsing interfaces for delivering spatial knowledge compared 2D and 3D versions of different visual elements such as aerial photos and semantic maps. They found that for most combinations, the 3D version is at least as effective, if not more effective, than its 2D counterpart [23]. Cockburn and McKenzie compared document file search for 2D, 2.5D, and 3D file environments, for both real and virtual environments. Each file environment was built by adding an additional dimension from the more basic environment, building from 2D to 2.5D and to 3D. The comparison resulted in decreased file search times as the dimensions increased, regardless of environment location [7]. Tavanti and Lind also performed a similar study where they developed 2 complementary environments; one for a 2D view of a folder tree, with and without the presence of a scroll wheel, and another being a 3D view of a tree, with elements sitting at an inclined plane. When testing for file search times, they concluded that the 3D form provided a spatial memory of the files that made file access faster than 2D [22]. Cockburn revisited and re-performed Tavanti's study with the exception of incorporating overlap in the file elements and incorporating perspective effects and found no significant difference between the 2D and 3D complement [6].

The second type of comparison is between different 3D file browsing techniques. Unfortunately, there have been only a few 3D file techniques that have a formal usability evaluation component. Some gave informal studies or user demos on their own technique without other comparisons [3]. Robertson et. al compared versions of the Data Mountain layout technique based on design implementations suggested from participants using a previous version of the technique [18]. The Task Gallery was developed from 3 prototypes, each with its own usability study, but the three versions were not directly compared [19]. While there have been some usability studies comparing 3D different design iterations of a given file browsing techniques, there has not been many studies that directly compare different techniques not based on design iterations from the literature.

2.3. 3D File Browsing Characteristics

Based on our analysis of existing 3D file browsing techniques, we have found several relevant characteristics. These are color coding, thumbnail preview usage, animations, a cameras focus exclusively on a folder's immediate children, and placing objects in a grid with respect to two dimensions. Some techniques apply a color coding scheme to individual files, whether user defined [3], category defined [23] or an indicator to file statistics [8]. Most techniques [3, 8, 18, 19, 21] use thumbnails to give snapshots of objects and elements that are more relevant to each other are placed by the user [3] or automatically [19, 8] and typically face the viewer [3, 8, 10, 18].

Some existing techniques use animation to portray an action done to a object like moving of task switching [16, 18, 19] or to show the contents of groups of files [8]. ClusterMode groups files within a folder, and the spinning variation uses animations to browse through the groups simultaneously [8]. We suspect that animation has not played a larger role not with existing techniques because the issue of iterating through a folder tree and switching between folders was often not addressed. For example, some techniques did not have folder structures [3, 8, 23] or to apply to files exclusively, and did not need to address this folder iterating issue. Those that did address the folder-iteration problem [12, 21] addressed them like current 2D WIMP interfaces do; the user can only see the immediate children of a file and when opening another folder, the objects in the last folder are pushed from view, and the new folder's children are viewed. One notable exception that utilizes animation is Tanaka's implementation when going down a folder tree. Previously focused children are shifted to the left, bringing into focus the current folder's children [21].

Most techniques also move the camera to focus on a folder's children exclusively. Active files may be defined as files that the user is generally interacting with, a feature similar to traditional interfaces. An example of this is within in 2D interfaces when looking through the contents of a folder, but not seeing a child folder's contents. However, because of the extra degree of freedom, the camera may focus on other elements aside from a folder's immediate children. Most 3D browsing techniques simply focus on a set of files [3, 8, 23], the consideration of how to visualize files that may not be in the active set, were not addressed. These techniques also do not address the folder-iteration interaction as well. The issue of not solving folder-iteration also results in most of these techniques fixing the camera to focus exclusively on an active group of files, without any additional files that the user may not be focusing on. Some techniques that display files not immediately active implement layering multiple elements

[3, 10, 18, 19, 23]. One technique, TreeCube3D [21], makes heavy use of layering by rendering all children of a folder tree in a condensed area, in contrast with other 3D file browsing methods [12] and current 2D WIMP interfaces that focus on a folder's immediate children alone.

Finally, another characteristic shared between current 2D WIMP interfaces and the several existing 3D file browsing techniques is to place files in a 2D grid with respect to a plane. File and folder representations are placed on this 2D plane and vary in height, usually related to the closeness between a file and the camera [3, 8, 12]. One technique that deviates from this by having additional groups within a folder is ClusterMode's Spinning racks variation that adds spacing to each group and places the files in a group such that the user can see two files per group at a time [8].

3. 3D Browsing Techniques

Based on the 3D file browsing characteristics described in the last section, we narrowed down the techniques we wanted to study to ones that spanned the most common characteristics, and found browsing techniques that spanned the characteristics. While we may be able to implement all the found 3D browsing techniques, we found it suitable to study the common characteristics for a more general holistic/comprehensive approach to the problem, and find the best techniques representing the characteristics. The most common characteristics are not necessarily the best or the most optimal for 3D browsing, but evaluating the most common characteristics also evaluates many 3d browsing techniques indirectly. These techniques were developed for evaluation, and some features were enhanced after initial pilot studies.

3.1. Descriptions of the Studied Characteristics

Thus, we decided to focus on techniques that make use of animations, 2D grids, and camera focus. Animations are used to assist in browsing data, in addition to expanding and rendering a folder by putting to focus its immediate children. Animations portray extra information when used dexterously and add to the aesthetics of the interface, but may simultaneously cause visual clutter, distraction, and impatience for the user. Using 2D grids place the files initially in a 2D plane where depth is varied based on a file ranking system. The plane hints of a traditional 2D interface, making it familiar and more user-friendly. However, shy of utilizing depth can be used to show the importance of a file, the 2D plane doesn't effectively take advantage of the 3D space. The characteristic of the camera focusing on the immediate children is also reminiscent of the 2D interface and prevalent in the 3D browsing literature, so its benefits are similar to that of the 2D grids. An additional benefit of this characteristic is that a user can understand the folder structure much better. There would be no additional elements to distract the user from searching through the active folder. Adding further information may also result in visual clutter and confusion in understanding the tree structure. Unfortunately like the 2D plane characteristic, with the added degree of freedom, more relevant information can be placed within the camera's view, such as the children of folders close to the active folder. After selecting the characteristics, we found 3 browsing techniques that spanned these characteristics, and differed significantly in interaction and appearance from each other. The techniques we implemented are Block3D, based on Dengel's PlaneMode [8], Cluster3D, based on Dengel's ClusterMode [8], and TreeCubePlus3D, based on Rekimoto and Green's TreeCube3D [16]. Table 3.1 shows how our chosen characteristics fit within these three techniques, showing the sharp differences between each technique.

	Block3D	Cluster3D	TreeCubePlus3D
Animation	No	Yes	Yes
Grid	Yes	No	Yes
Camera Focus	Yes	Yes	No

Table 3.1: Key characteristics for the selected 3D file browsing techniques. Animation refers to the implementation of animations in active file browsing. Grid implies placing the files in a grid layout. Camera focus refers to the focus of the virtual camera only on the current folder.

We could have also tested techniques that had the absence of all techniques, or ones that represented all qualities, but as a preliminary analysis we chose to focus our study on these techniques. We also found difficulty in finding techniques in the literature that already exemplified all techniques. Future studies can be expanded to consider the complete absence and presence of these characteristics.

3.2. Common Elements

When varying with whether a technique displayed a characteristics, we kept several visual and interactive elements constant.

3.2. Visual Elements

All our techniques render a file thumbnail, or a snapshot of what the file looks like, and the name of the file below it. Both these elements have no transparency effects applied for easier file viewing. The name of the folder is presented at the top of the screen, and the camera is dynamically positioned to view the entire folder with the closest distance from the folder's children as possible. All files initially face the camera. Each thumbnail is placed in front of a colored square, categorizing its important to a user.

3.2. Interactive Elements

The camera's pose is manipulated using the arrow keys to zoom in and out as well rotate the about the center of screen. Tanaka's folder-shifting animations, which push a folders siblings to the left and render the folder as the center of focus, are implemented throughout all techniques and browsing up and down the visible folder tree are also implemented through buttons [16], resolving the folder-iterator issue with all prototypes.

The arrangement of the file elements are influenced by ratings that serve as the result of a logical arrangement technique. One could arrange alphabetically, by file type, by size, by frequency of use, by date of recency, a combination of the

latter two [2], in a self-organized map [23], or in various other methods as well. To account for the different logical ways to arrange files, we have assigned each file in the evaluation a ranking from 1 - 10. This numbering system can be used to arrange relevant files together, or a quantity expressing frequency of use, or for various other methods of logically arranging the files. Because there are various methods of logically arranging, the ranking system generalizes this by assigning a number to each file so that a logical arrangement can utilize this ranking whenever appropriate.
This rank is represented visually as a color ranging from blue to red, on a square behind the thumbnail, similar to Dengel's work [8]. The visual arrangement of these files for all techniques is that elements with similar ranking are closer to each other than others. This can be synonymous in alphanumeric order as the 'A' files are closer to other 'A' files and 'B' files closer to 'A''s than 'Q' files, or that more frequently used files are closer to each other than others, or in whichever interpretation is appropriate for a logical arrangement technique. Folders have the rating of the weighted average of all its children and successors.

3.3. Block3D

The Block3D technique (shown in Figure 3.1) was originally for visualizing relations, based on Dengel's PlaneMode technique [8]. We chose to base Block3D on this technique because it has the characteristics of placing files in a 2D grid and having the virtual camera on the immediate folders. This technique is also the most similar to the traditional 2D WIMP interface, and through this technique if users would notice the similarity between the traditional and the 3D

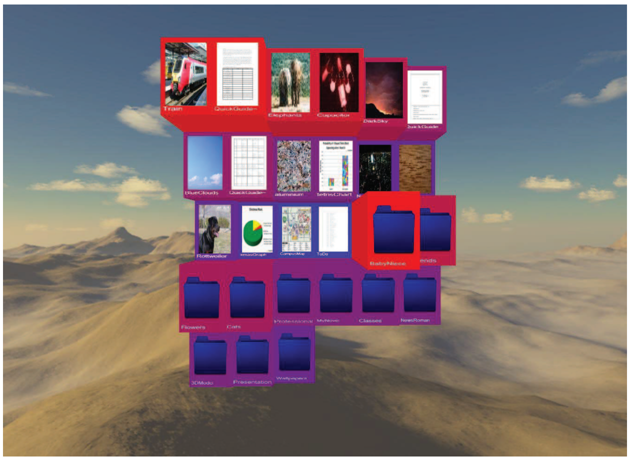


Figure 3.1: A screenshot of the Block3D file browsing technique.

interface.

3.3. Visual Elements

Depending on how many of the search queries were matched, those files were closer or farther to the camera. Most changes implemented to this technique were applying common visual elements to the technique. The original implementation used color as a category indicator, with files with closer categories grouped together and all files were arranged on the width-height dimension. The color implementation was replaced with the coloring scheme described in the common visual elements section. The only change that was specific to this technique was the rendering of the thumbnail on the box face farthest away from the front, in the case that the user decides to rotate the camera behind the visualization.

3.3. Interactive Elements

The PlaneMode implemented search queries that determined groups of document files that matched the different queries. We opted to exclude the search query because search queries were outside of the focus of the study. The implementation did not address the use of parent and children folders, so this technique also applied the folder-shifting element in the common interactive element.

3.4. Cluster3D

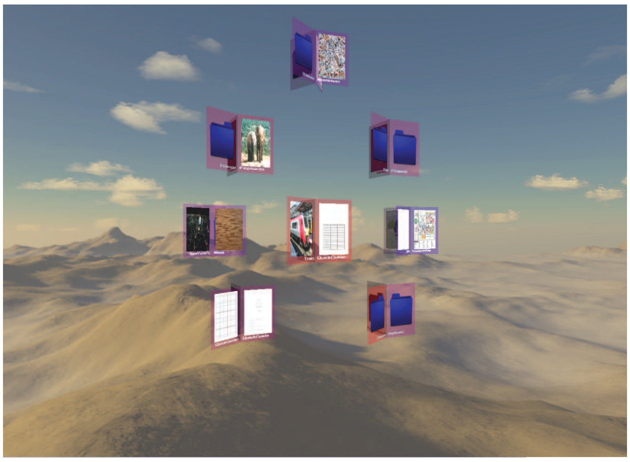


Figure 3.2: A screenshot of the Cluster3D file browsing technique.

Cluster3D, as shown in Figure 3.2, is a 3D file organization technique that is a variation of Dengel's ClusterMode technique[8]. The technique groups files into spinning clusters, not specific to the element count per group. The spinning variation on the technique arranges files in several small rotational racks on the screen, with respect to a file's information.

3.4. Visual Elements

The original implementation placed the most relevant groups towards the center of the camera's view. For this implementation, to accommodate for the rating system, similarly rated and colored files are grouped together to form groups. We altered the positioning of the files in that they are close relative to the average rating in the file group, similar to the original implementation. The highest-averaged group is positioned in the front, while the second, third and fourth highest rated were positioned left, right, above and below the centered group, respectively. Four more groups were placed on the corners of the camera's view, totalling a maximum number of groups to be 9. After each group is positioned relative to the others, the depth of each group is adjusted depending on the average rating. Each group is closer or farther to the camera depending on the magnitude of the average, similar to individual files in the Block3D.

The technique originally had colored lines showing relationships between each group. This was implemented when there were multiple colored search queries, and the lines were colored according to the search query that is common between each group. However, because of the removal of search queries and the maximum number groups only being 9, we found little use in the colored line, and could have made the screen appear more cluttered. As a result, we removed this feature.

3.4. Interactive Elements

The original implementation had the user view each folder through an automatic flipping animation, synonymous with a book flipping a page itself. Initial pilot studies showed users expressed a dislike for not being able to control the automatic flipping animation. As a result, we implemented a button that performs the flip animation for all visible rotating racks, and an animation speed control using the mouse's scroll wheel. In addition, users felt the colored lines often cluttered browsing, so we removed them in our final implementation. From Dengel's work, it was unclear what the minimum and maximum number of files should be for a particular cluster.

Visually, groups that were more highly rated were closer and more centered within the camera's view. For consistency of arranging based on a file's ranking, our initial implementation had all children files first binned according to ranking, so there are initially 10 empty bins, one per rating. Then if there is at least 1 file per unique rating, and each bin has at least 1 element, then the group with the fewest number of files would be distributed among 2 closer ranked groups, allowing for a maximum of 9 groups. The actual number if of groups may range from one group to 9, depending on the quantity of children files, and the distribution of ratings.

Our inital testing indicated how to best distribute the files. Initial pilot testing showed that users liked to 'eliminate' racks as they searched for a given file, and found that the number of files in a group was important in searching. If a group had an uneven number of files, a user would need to remember not only which clusters have been browsed, but also the number of files per group. For example, a folder may have a total of 13 children files, placed into 2 groups, group A with 3 files and group B with 10. After flipping 3 times, the user may notice repeating elements in A, and assume that both groups have already been browsed, eventhough there would be 7 elements left unseen in B. If the user forgot about the number of files per group, the user may stop searching pre-maturely. Our pilot studies found that an unequal number of files per group made it difficult to keep track of which groups have been fully browsed or not. If there were 2 groups with equal number of files, the user would focus on the two groups, and once repeating files are displayed, the user would know that this folder is thoroughly searched, and to search elsewhere. Thus, we chose to ensure that each group had the same number of files.

3.5. TreeCubePlus3D

Figure 3.3: A screenshot of the TreeCubePlus3D file browsing technique.

The final visual arrangement technique is a derivation of the TreeCube interface layout, developed initially by Rekimoto and Green [16] and evolved further by Okajima and Okada [15]. The technique's unique characteristic is that it gives an overall view of all the files contained in a folder tree. The technique renders all n files in a folder tree simultaneously, compared to the worst case of $\log N$ files as one branch of the folder tree with the Cluster3D and Block3D techniques.

3.5. Visual Elements

The TreeCubePlus3D layout, shown in Figure 3.3, uses the sizes of the files and folders to create cubes that fit within the parent folder. These cubes are arranged such that all items within a given folder are evenly distributed where all its children's files and its folder's children fit within a larger cube. We chose Okajima and Okada's quantum cube layout approach in our implementation. The quantum cube's flexibility on cube size for the larger, encompassing cube [15] indicates that this technique can best display folders, regardless of file quantity.

3.5. Interactive Elements

Our preliminary analysis of this technique found that seeing the nested files was difficult. Therefore, we developed a second way to browse through data in addition to clicking through nested folders. We devised an unwrapping animation that first removes a layer, or a group of files that is either a subfolder or the folder's immediate children, rotates the layer, and becomes the primary focus in view (see Figure 3.3). To view the next layer, the old current layer would move to the side of the current view, making room for the next layer, and so on. The user would be able to browse through all folders and sub-folders through this animation-based method.

4. Experimental Design

To begin to explore the effects of different 3D file browsing techniques, we conducted a formal user evaluation comparing the three techniques described above in a manual file searching task. We were interested in task completion time and user preference for each technique as well as for different types of files.

4.1. Technique Conjectures

Block3D is a technique most resembling the traditional 2D interfaces through its implementation of grids and lack of any 3D animations. While it somewhat uses the 3D space with the depth, we feel that it will only do fairly well in task completion times and user preferences. This technique was the most similar to the traditional 2D interface, so subjects should at least do fairly well due to familiarity. We believed that for more detailed searches like documents, files that generally do not visually differ from each other as much as images do, would better perform and be preferred for this technique.

The clustering of files in Cluster3D may prove to be useful at times, we believed that this feature will limit the user's view from seeing other files. The clustering does save 3D space, but does not use the remaining 3D space more effective than Block3D. The user must perform more actions to see all the immediate children of a folder compared to the other two techniques. Because of this cognitive effort on the user, we believed this technique would be the slowest in completion times, and consequently the least preferred technique.

TreeCubePlus3D takes full advantage of the 3D space by rendering all files, and this feature is one of the strongest reasons why we believed that this technique will excel and become the most favored technique. However, this technique does have the risk of having an overlap of files, and may appear overwhelmed to the users. We think that this technique will be best with one of the more easy files to search for, particularly images, and a mix of documents and images, where files are more likely to appear different from its neighboring files. These files are visually different from each other, the potential overlap disadvantage less of a problem.

4.2. Participants

Eighteen participants (11 male, 7 female) were recruited from the student population at the University of Central Florida with ages ranging from 18 to 38, from disciplines ranging from Journalism to Computer Science. Of the 18 participants, five have used 3D user interfaces, with the most common being simulations and Compiz-fusion, a 3D desktop window manager [1]. All but one participant played video games. The experiment took each subject approximately 60 minutes to complete and all participants were paid 10 dollars for their time.

4.3. Apparatus

The experiment took each subject approximately 60 minutes to complete and all participants were paid 10 dollars for their time. The experimental setup consisted of a 50 inch Samsung DLP monitor and a desktop PC with a 2.4 GHz quad-core processor and a Nvidia 8500 graphics card running Windows Vista (see Figure 4.1). The experimental software was designed on Windows C Sharp using Windows Presentation Foundation.

Figure 4.1: A user participating in the experiment with TreeCubePlus3D shown. On the left-hand side of the screen, the target image is shown.

4.4. Experimental Task

Users were asked to perform a manual file search task within a folder tree. Files that were higher rated were more likely to be searched for. The ranking simulates the idea that users frequently open specific files often, and consistently, would search for those same files often. In order to select a searched file, a random rank was generated, such that those rated higher would have more chances to be searched for. A targeted file was then randomly selected from a list of files in a given folder tree that have that same ranking. The target file was selected independent of its location in the folder tree, so files lower in the folder tree, and thus harder to find, were equally as likely to be picked as easier targets like files closer to the root of the tree.

There are 3 folder trees, each with different file types; documents, the second contained images, and a mix of both. The organization of these trees was organized in a binning process. In this way, we chose this approach to simulate what a user might typically have in their own file directory structure. For example, in the images folder tree, we have a cats folder and a Roman trip folder. To see the full listing of all 3 folder trees, please refer to Appendix A. Given information about a file to find, the participant had the following cues for searching:

- File Name: The name is associated contextually to the folder it belongs to. So files named "ClassAssignment" would be under the "Classes" folder.
- Image Thumbnail.
- Ranking and its associated color

These cues are independent of the file browsing technique used. Also, a file's original file size had no bearing on its visual representation, nor was the information displayed nor used during the experiment.

4.4. Generation of the File Trees

The folder trees, the number of children in a folder, and the identity of a child, whether a document or image file, or a folder, were randomly generated with each folder containing 3 - 40 children. The maximum number of children in any folder tree was set to 40. With the 40 children cap, any one folder tree had approximately 180 to 200 total number of files. This number was sufficient for the trial generator to select a reasonable number of files, given a rank. Please refer to Appendix A to see the folder trees used in the experiment.

4.4. Generation of the Files

A file's identity and rank are determined by a random number generator. A file's identity, either document or image, in a mixed folder tree was determined by a random number generator with 2 numbers where one is half as likely to occur as another. In the mixed folder tree, images are approximately half as prevalent as documents. A file's rank is also determined by a random number generator, from 1 - 10. An individual file's importance is dependent on the user, and if particular topics were more preferred than others, users may remember it more strongly than others; A subject who visited Rome may memorize the contents of the Roman Trip folder much better than someone who hasn't, and this bias could skew the results. To make the folder tree general and to simulate importance, the ranking was generated randomly. A file's rating was made to be a static constant to assure that higher ranked files would be more likely to be searched for than lower ranking ones.

4.5. Experimental Design and Procedure

We used a 3x3 within subjects factorial design, where the independent variables are file browsing technique and file type. The independent variable varied between the Block3D, Cluster3D, TreeCubePlus3D file browsing technique. The file types varied between documents, images, and a combination of both. The dependent variable is task completion time, measured by the time between displaying the file to search for and when the participant correctly clicks on its representation in a technique.

The experiment began with a consent form and a pre-questionnaire that asks for a participant's gender, age, experiences with file browsing, 3D user interfaces, and video games experience, specifically the types of games played and how many hours a week played. Participants were then introduced to all 3 folder trees using a printed sheet with a tree diagram, similar to Appendix A, and the types of files found in each folder. Then participants went through a practice session where they searched for files using each file browsing technique for approximately 5 minutes. The participant was then familiar with the interactive techniques associated with each technique, and proceeded to the experiment. The experiment consisted of 3 x 3 blocks with 4 trials each, resulting in 36 total trials. Target files were randomly selected with a bias for more highly ranked files, and no file is searched for twice during the experiment. The experiment concluded with a post-questionnaire.

4.6. Usability Criteria

In addition to task completion times, we gave users a post-questionnaire to gather preference data. For each file type (images, documents, and mixed), participants were asked to choose which 3D file browsing technique was preferred the most and the least, and why. Participants were also given the opportunity to provide any comments that had on each technique.

5. Results

5.1. Task Completion Time

A repeated measures two-way analysis of variance (ANOVA) was performed on the dependent variable, task completion time. The independent variables were 3D file browsing technique (BT) and file type (FT). Table 5.1 summarizes the main effects for both independent variables and their interaction. Significant effects were found for both independent variables, but not their interaction.

Effect	Task Completion Time
BT	$F_{2,16} = 15.42$ $p < 0.05$
FT	$F_{2,16} = 7.84$ $p < 0.05$
BT × FT	$F_{4,14} = 1.51$ $p = 0.21$

Table 5.1: The main and interaction effects for browsing type (BT) and file type (FT) for task completion time.

To gain a better understanding of how the different conditions affected task completion time, we conducted a post-hoc analysis, performing pairwise comparisons on the three BT (three comparisons) and on the three FT (three comparisons). To control for the chance of Type I errors, we used Holm's sequential Bonferroni adjustment [11] with 3 comparisons at $\alpha = 0.05$ for each test.

	Block3D	Cluster3D	TreeCubePlus3D
Mean:	15.34s	19.28s	21.05s
SD:	4.41s	5.43s	4.66s

Table 5.2: Mean completion times (in seconds) for the three file browsing techniques when file type is collapsed.

For BT, (see Table 5.2), there were significant differences between Block3D and

TreeCubePlus3D ($t_{17} = -5.16, p < 0.0083$) and Block3D and Cluster3D ($t_{17} = -3.78, p < 0.01$). There was no significance between Cluster3D and TreeCubePlus3D ($t_{17} = -1.59, p = 0.131$). These results indicate that participants performed significantly faster with Block3D was significantly faster at the file searching task than either Cluster3D or TreeCubePlus3D.

	Images	Documents	Mixed
Mean:	15.34s	20.19s	19.95s
SD:	3.91s	4.94s	6.52s

Table 5.3: Mean completion times (in seconds) for the three file types when file browsing technique is collapsed.

For FT (see Table 5.3, there were significant differences between Images and Documents ($t_{17} = 3.91, p < .0.083$) and Images and Mixed ($t_{17} = -3.09, p < 0.01$). There was no significant difference between Documents and Mixed ($t_{17} = 0.18, p = 0.857$). The results imply that participants were able to search for images significantly faster than documents or the combination of images and documents.

5.2. Post-Questionnaire Results

For each file type (documents, images, and mixed) we asked participants to select which technique they preferred the most and the least. The results are shown in Figures 5.1 and 5.2. Chi-squared analysis of the data revealed that the Block3D technique was significantly favored ($\chi^2_2 = 6.33, p < 0.05$) as the most preferred technique when searching for documents.

We also analyzed general trends in the reasoning behind why chose their most and least preferred browsing technique. We found several themes in the

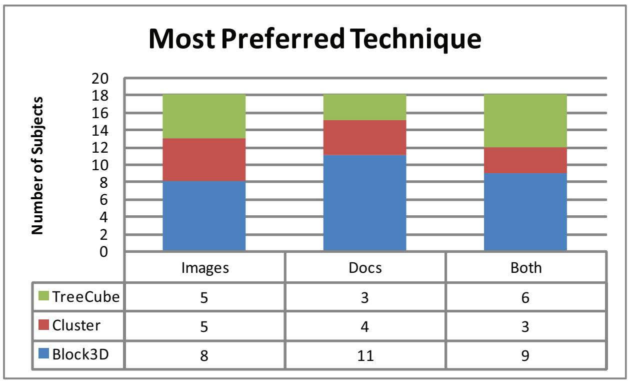


Figure 5.1: The number of participants who most preferred each 3D file browsing technique for each file type.

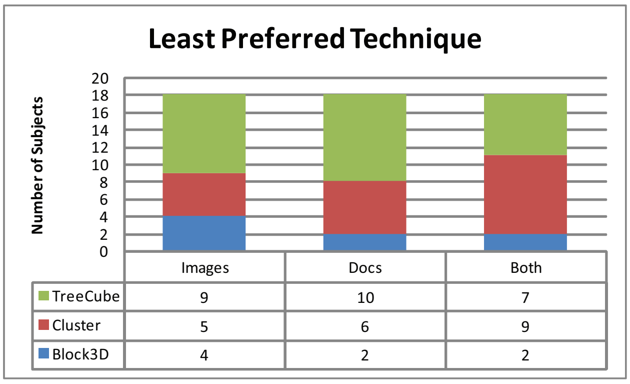


Figure 5.2: The number of participants who least preferred each 3D file browsing technique for each file type.

comments for each respective technique. For Block3D, the most frequent comments made, from those who preferred it, was its ease of navigation and file organization for all three file types. One participant said that "It was easier to control and the files were neatly organized". A few participants noticed the similarities between this technique and traditional 2D interfaces. For those participants who least favored favored this technique, their predominant comment was that it was tiresome and overwhelming.

Those who most preferred Cluster3D claimed it easy to use, whether navigating of finding files, had a fast pace, and enjoyed the flip animation and the spacing. A comment noted that "Flipping through files provided encouraging visual feedback. The way all the files were spread out made the task of searching for a specific document easier". Those participants who favored this technique the least did not like the animation and felt the technique obscures objects.

The most predominantly positive comments from participants preferring TreeCubePlus3D were an ease of searching, a preference for the visuals and fast speed. A participant wrote that "I was able to see all the files at the same time. It helped to have everything all at once". Some participants found the ability to have all of the files in one place to be useful, a quality unique to this technique. The most negative comments in relation to this technique was that it had a poor organization, a slow speed, and gave an overwhelming feeling.

5.3. Results with respect to Technical And Non-Technical Disciplines

To gain further insight on the qualitative and quantitative data and to attempt to isolate technical from non-technical users, we repeated the analysis on subjects in technical fields only (8 subjects) and subjects in non-technical fields. (10 subjects) We defined technical fields as disciplines such as Engineering and Computer Science, while others were considered non-technical. This distinction of technical and non-technical fields allowed for an almost-even division of the subjects. By analyzing these two groups, we gain insight on some further reasons on the technique preferences.

5.3. Task Completion Time

For task completion times across technical and non-technical fields, we found the averages for each group across the techniques (See table 5.4) and for each file type (See table 5.5).

Trends continue to indicate that subjects are faster at finding Images than

	Block3D	Cluster3D	TreeCubePlus3D
Technical:	14.35s	17.46s	20.33s
Non-Technical:	16.13s	20.73s	21.63s

Table 5.4: Mean completion times (in seconds) for the three techniques, varying on whether the subject was in a technical or non-technical field.

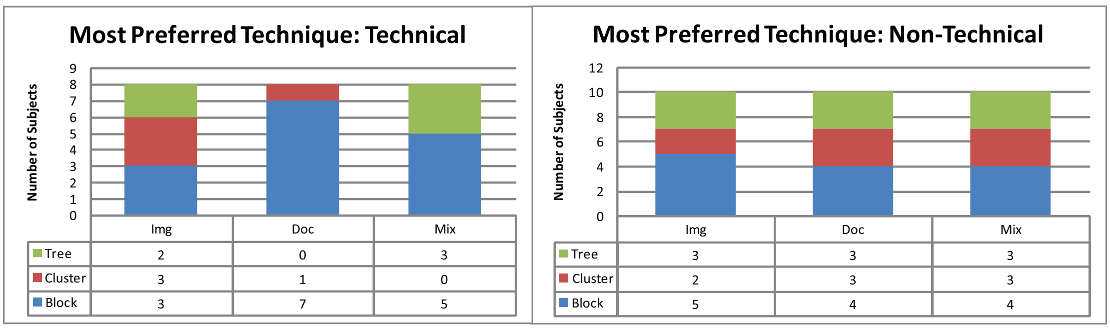
	Images	Documents	Mixed
Technical:	14.59s	19.91s	17.65s
Non-Technical:	16.27s	20.43s	21.79s

Table 5.5: Mean completion times (in seconds) for the three file types when file browsing technique is collapsed.

Documents and Mixed, and that subjects found files using Block3D faster than using Cluster3D or TreeCubePlus3D.

5.3. *Post-Questionnaire Results*

The results for the most favored technique and the least favored for both groups are shown in Figures 5.3 and 5.4. Chi-squared analysis of technical group data revealed that the Block3D technique was significantly favored ($\chi^2_2 = 10.75, p < 0.05$) as the most preferred technique when searching for documents and overall($\chi^2_2 = 9.25, p < 0.05$). The Technical group data also significantly disliked the TreeCubePlus technique overall($\chi^2_2 = 10.75, p < 0.05$). The non-technical group had no significant data points.



	Img	Doc	Mix
Tree	2	0	3
Cluster	3	1	0
Block	3	7	5

	Img	Doc	Mix
Tree	3	3	3
Cluster	2	3	3
Block	5	4	4

Figure 5.3: The number of participants who least preferred each 3D file browsing technique for each file type.

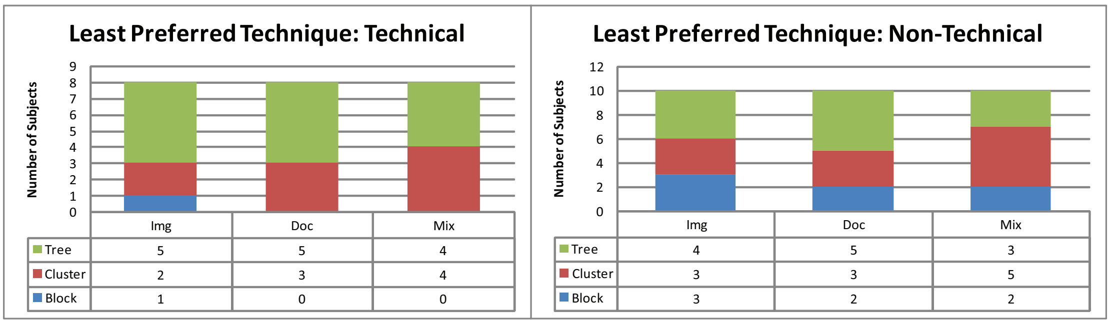


Figure 5.4: The number of participants who least preferred each 3D file browsing technique for each file type.

6. Discussion and Conclusion

6.1. Discussion

The Block3D technique was a familiar interface to users. This familiarity may explain why this technique significantly performed the best in the task completion times. The most often mentioned comment, the ease of navigation, can be explained by its familiarity, not necessarily the approach. Block3D also was the most familiar to the traditional 2D interface, so the possibility exists that subjects performed better with that technique due to either positive qualities of the traditional 2D interface, or out of familiarity. Subjects that perform familiar tasks like file search, in a very familiar environment are more likely to perform better in that environment than in a more novel one. the results indicate that elements in a traditional 2D interface can be implemented in a 3D environment and perform well with files. Block3D was significantly the most preferred technique for documents, a confirmed conjecture.

The Cluster3D was not the worst or slowest technique as conjectured. The technique did not perform as well as Block3D, as indicated by task completion times, but did not seem to suffer as badly as TreeCubePlus3D when compared to document preference data. Cluster3D's completion times, though not significant, indicated that it was not the slowest technique. A more surprising result was that the participants commented that they liked the spacing between the groups, not necessarily the grouping characteristic that generates the spacing.

Similarly, like Cluster3D, TreeCubePlus3D's task completion times were the slowest, though not significantly different than Cluster3D. Even though this technique shows all files in a folder tree simultaneously, it wasn't significantly the slowest technique compared to the other two, implying that the number of files rendered did not affect completion times. TreeCubePlus3D was the least preferred in the significant data, and the most disliked technique, though not significant. The qualitative data of feeling overwhelmed and poor organization can be a result to the technique's unique trait, showing all the files at once in a cube. The technique's rendering of all files may also cause the technique to be slower, as commented in the qualitative data. Informal comments to the moderator also support the participants perceiving TreeCubePlus3D as the slowest technique. Because of the slowness and user's sentiments, along with the poor task completion times and usability data, the characteristic of showing all files may not be a positive one. Techniques that had additional animations, TreeCubePlus3D and Cluster3D, did not seem to provide additional benefits in task completion times and usability preferences, compared to the Block3D. This result implies that additional animations may hinder file searching.

The more technically inclined group would be the group to represent power-users to technologies. Because the technical group was more likely to have power-users, the significant data associated with the technical group can help anticipate some 3D browsing preferences. The task completion times for all subjects already indicated that Block3D was the fastest techniques, so the technical group's preference for the Block3D may be due to speed. Compared to the non-technical group, however the technical group's preferences and task completion times were also more likely to be influenced by the familiarity quality of Block3D because such disciplines encourage frequent use of a computer.

The non-technical group was more varied in their technique preferences, with no significant data. Compared to the technical group with 3 significant data points, the non-technical group appears to be more varied in preferences with no significant data points, and such a lack implied that the group may be more welcome to experiment with these new 3D browsing techniques, whether this may be out of novelty, utility, or aesthetic. While the data analysis for both groups was was smaller than desired, the data still provided implications already supported with the overall data.

In relation to the characteristics, because Block3D significantly performed the fastest and was the most preferred for searching for documents, the removal of animations appears to be a positive effect to improve search. The only feature that was not portrayed in Block3D and portrayed in others were animations. Block3D succeeding in task completion times do indicate that animations slow down the task. When speed is a strong consideration in designing these browsing techniques especially with documents only, the strongest implication would be a decrease in animations due to Block3D's strong performance and preference success.

The camera focus and the gridding placement were harder characteristics to evaluate because there was less significant data associated with those spanned techniques. However, those features are promising due to the possible positive qualities from the Cluster3D and TreeCubePlus3D, so further studies can aid in evaluating these features.

6.2. Future Work

From the results of our evaluation, there is a significant amount of work that can be done to better determine the efficacy of 3D file browsing techniques before a comparison between traditional 2D browsing techniques is conducted. For example, one comment exclusive to the TreeCubePlus3D technique was a sense of enjoyment. More work is needed to understand the aspects that make this enjoyable. Spacing, one of the noted benefits of Cluster3D, can also be explored for other techniques. The slowdowns perceived by participants for both Cluster3D and TreeCubePlus3D is an implementation issue rather than inherent to the technique. Performing the experiment under a computer with a better graphics card, or switching to a different 3D environment for the implementation, may improve the usability of both TreeCubePlus3D and Cluster3D.

In terms of scalability, having 3-40 children per folder is a practical cap, but future studies can expand this range. We suspect that whenever the children count is too large for the Block3D browsing technique in which users become overwhelmed, the Cluster3D technique, only showing a defined number of files simultaneously, will be more favored in usability criteria. For the purposes of this study, we proposed using two differing file types, documents and images. We can expand this list for web pages, audio, videos and animated GIF's, the latter two being a series of images.

The rating system was meant to generalize the output of logical arrangements of files, but the experiment did not allow for any opportunity for the ratings to change dynamically. Introducing new ratings and new data may result in different technique preferences, and researching different logical arrangement techniques to dictate how the ratings change, would aid in testing this.

One of the factors that was found to speed up file search in a 3D environment is having the user manually position files. By manually positioning, the user has the opportunity to form a memory map of the file's individual location[18]. We were unable to accomplish this for this work in order to maintain a constant folder tree for all subjects during the experiment. Ideally the optimal execution of this experiment would be using a subject's own personal files, but the difficulty of attempting constant variable such as a constant number of files and constant folder structure, may be unmanageable. The validity of experiment may dictate certain constrains, such as a random distribution of images and documents, or a maximum or minimum number of files per folder, or a certain maximum number of levels in the folder tree, but this may be unreasonable to maintain when dealing with a subject's personal files, and more research must be made to find alternatives to accomplish using personal files.

Future studies can be expanded to consider the complete absence and presence of these characteristics, and add more characteristics for evaluation. By adding more characteristics, more techniques can be tested as well. More novel techniques can be derived from the absence/presence of specific techniques.

6.3. Conclusions

We have presented an experiment that tested three distinct 3D file browsing techniques, Block3D, TreeCubePlus3D and Cluster3D. These techniques were chosen based on an analysis of the literature which highlighted important characteristics that shaped this class of browsing techniques. The results indicate that participants were significantly faster in file searching task using Block3D compared to TreeCubePlus3D and CLuster3D. Participants seemed to find Block3D a more usable technique based in part on its similarity to traditional 2D file browsing interfaces. However, participants did find positive aspects in TreeCubePlus3D and Cluster3D, namely enjoyment and file spacing respectively. We believe this experiment provides an initial foundation for determining which characteristics are most important for 3D file browsing techniques so true comparisons can be made with their 2D counterparts.

Bibliography

[1] Erkin Bahceci, Ronald Bar, Danny Baumann, Robert Carr, Roi Cohen, Dennis Kasprzyk, Kristian Lyngstl, Patrick Niklaus, Andrew Riedi, and Guillaume Seguin. Compiz fusion.

[2] Xinlong Bao, Jonathan L. Herlocker, and Thomas G. Dietterich. Fewer clicks and less frustration: reducing the cost of reaching the right folder. In *IUI '06: Proceedings of the 11th international conference on Intelligent user interfaces*, pages 178–185, New York, NY, USA, 2006. ACM Press.

[3] Sebastian Baumgartner, Achim Ebert, Matthias Deller, and Stefan Agne. 2d meets 3d: a human-centered interface for visual data exploration. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, pages 2273–2278, New York, NY, USA, 2007. ACM Press.

[4] Benjamin B. Bederson and James D. Hollan. Pad++: a zooming graphical interface for exploring alternate interface physics. In *UIST '94: Proceedings of the 7th annual ACM symposium on User interface software and technology*, pages 17–26, New York, NY, USA, 1994. ACM.

[5] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice.* Addison Wesley Longman Publishing

Co., Inc., Redwood City, CA, USA, 2004.

[6] Andy Cockburn. Revisiting 2d vs 3d implications on spatial memory. In *AUIC '04: Proceedings of the fifth conference on Australasian user interface*, pages 25–31, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[7] Andy Cockburn and Bruce McKenzie. Evaluating the effectiveness of spatial memory in 2d and 3d physical and virtual environments. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 203–210, New York, NY, USA, 2002. ACM Press.

[8] Andreas Dengel, Stefan Agne, Bertin Klein, Achim Ebert, and Matthias Deller. Human-centered interaction with documents. In *HCM '06: Proceedings of the 1st ACM international workshop on Human-centered multimedia*, pages 35–44, New York, NY, USA, 2006. ACM Press.

[9] G. W. Furnas. Generalized fisheye views. *SIGCHI Bull.*, 17(4):16–23, 1986.

[10] Dusan Hamar. Spheresite, 7/26 2006.

[11] Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

[12] Upper Bounds Interactive Inc. Tactile 3d interface - software to browse, explore, and organize your file-system in 3d.:, Wednesday, December 17, 2008 2008.

[13] T. Alan Keahey and Edward L. Robertson. Techniques for non-linear magnification transformations. In *In Proc. IEEE Symposium on Information Visualization, IEEE Visualization*, pages 38–45, 1996.

[14] Sun Microsystems. lg3dcore: Lg3d core project, 30 Oct 2007 2007.

[15] Seiji Okajima and Yoshihiro Okada. Treecube+3d-visom: Combinational visualization tool for browsing 3d multimedia data. In *IV '07: Proceedings of the 11th International Conference Information Visualization*, pages 40–45, Washington, DC, USA, 2007. IEEE Computer Society.

[16] J. Rekimoto and M. Green. The information cube: Using transparency in 3d information visualization. In *In Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, pages 125–132, 1993.

[17] Kirsten Risden, Mary P. Czerwinski, Tamara Munzner, and Daniel B. Cook. An initial examination of ease of use for 2d and 3d information visualizations of web content. *Int. J. Hum.-Comput. Stud.*, 53(5):695–714, 2000.

[18] George Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data mountain: using spatial memory for document management. In *UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 153–162, New York, NY, USA, 1998. ACM Press.

[19] George Robertson, Maarten van Dantzich, Daniel Robbins, Mary Czerwinski, Ken Hinckley, Kirsten Risden, David Thiel, and Vadim Gorokhovsky. The task gallery: a 3d window manager. In *CHI '00:*

Proceedings of the SIGCHI conference on Human factors in computing systems, pages 494–501, New York, NY, USA, 2000. ACM Press.

[20] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, 1992.

[21] Yoichi Tanaka, Yoshihiro Okada, and Koichi Niijima. Interactive interfaces of treecube for browsing 3d multimedia data. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 298–302, New York, NY, USA, 2004. ACM.

[22] Monica Tavini and Mats Lind. 2d vs 3d, implications on spatial memory. In *Proceedings of the IEEE Symposium on Information Visualization 2001 (INFOVIS'01)*, San Diego, 22–23 October 2001 2001. IEEE.

[23] Bin Zhu and Hsinchun Chen. Using 3d interfaces to facilitate the spatial knowledge retrieval: a geo-referenced knowledge repository system. *Decis.Support Syst.*, 40(2):167–182, 2005.

A. Experiment Folder Trees

Below shows the folder trees that subjects interacted with. Each folder tree holds a specific data type. (Documents, Images, or Mixed). The participants were given a sheet with this exact information, and the moderator reviewed the folders and the finds of files found in these folders.

A.1. Documents Folder Tree

Documents
— +— DocsWithShapes
— +— flyers
— +— homework
— — +— Chemistry
— — +— ForeignEssays
— — +— ForeignSpeech
— — +— ForeignVocab
— — +— MathEquations
— — +— PhysicsNotes
— — +— SolvedMathProblems
— +— PersonalFiles
— — +— Cartoons
— — +— FreeWriting
— — +— GameCheats
— — +— Recipes
— — +— WritingContests
— +— ScholarshipApplications
— +— WorkRelated
— +— CV
— +— Memos
— +— Pre-Resume
— +— Resume
— +— ResumeOld
— +— WorkFinances
— +— WorkPromoMaterials

A.2. Images Folder Tree

Images
— +— AnimalsNotCats
— +— Avatars
— +— Cats
— — +— 2Cats

— — +— 3MoreCats
— — +— CatsAndOtherAnimals
— — +— OrangeCat
— +— EgyptTrip
— — +— KingTutBelongings
— — +— KingTutStatues
— — +— Statuettes
— +— People
— +— RomeTrip
— — +— Colosseum
— +— Sky
— — +— Rays
— — +— Stars
— +— Transportation

A.3. Mixed Folder Tree

Mixed
— +— 3DModel
— +— BabyNiece
— +— Cats
— +— Classes
— — +— ArchSurvey
— — +— EgyptHistory
— — +— GeneticsLab
— — +— InternetEssay
— — +— PandaEssay
— — +— PsychoSurvey
— +— CompSpeech
— +— Flowers
— +— Friends
— +— NewsRoman
— +— Novel
— +— Professional
— — +— 2ndLifeOffice
— — +— CEOReport
— — +— OfficeTutorial
— +— Wallpapers

B. Pre and Post-Questionnaire Documents

Below are the two forms used to extract usability-relevant information. This information was placed in an electronic form within the experimental software for fast data collection.

B.1. Pre-Questionnaire Form

Subject Number:_____

Pre-Questionnaire

Age: _____

Gender:

M F I do not wish to answer question

Major(s):____

Have you ever used the Windows or Mac operating systems?

Yes No

Have you ever had to search for a file by clicking on various directories till you found what you were looking for?

Yes No

Have you ever used a 3D Interface?

Yes No

If so, what was it?

____

____

Have you ever played video games?

Yes No

If Yes, how many hours per week do you spend playing video games? _____

If Yes, what types of video games do you play?

____

____

B.2. Post-Questionnaire Form

Subject Number:_____

Post-Questionnaire

The three visualization techniques you used today were Cluster3D, TreeCube, and 3DBlockVis.

Searching for Images

1. Of the three techniques you used today, which one made it the easiest to find the files you were looking for?

    Cluster3D TreeCube 3DBlockVis

2. Of the three techniques you used today, which one helped you find the files you were looking for the fastest?

    Cluster3D TreeCube 3DBlockVis

3. Overall, which technique did you like best?

    Cluster3D TreeCube 3DBlockVis

4. Why did you like this technique the best?

____

____

____

____

____

____

____

5. Overall, which technique did you like least?

    Cluster3D TreeCube 3DBlockVis

6. Why did you like this design the least?

____

____

____

____

____

____

____

Searching for Documents

7. Of the three techniques you used today, which one made it the easiest to find the files you were looking for?

   Cluster3D TreeCube 3DBlockVis

8. Of the three techniques you used today, which one helped you find the files you were looking for the fastest?

   Cluster3D TreeCube 3DBlockVis

9. Overall, which technique did you like best?

   Cluster3D TreeCube 3DBlockVis

10. Why did you like this technique the best?

____

____

____

____

____

____

____

11. Overall, which technique did you like least?

   Cluster3D TreeCube 3DBlockVis

12. Why did you like this design the least?

____

____

____

____

____

Searching for Both Documents and Images (Mixed)

13. Of the three techniques you used today, which one made it the easiest to find the files you were looking for?

    Cluster3D TreeCube 3DBlockVis

14. Of the three techniques you used today, which one helped you find the files you were looking for the fastest?

    Cluster3D TreeCube 3DBlockVis

15. Overall, which technique did you like best?

    Cluster3D TreeCube 3DBlockVis

16. Why did you like this technique the best?

----

17. Overall, which technique did you like least?

    Cluster3D TreeCube 3DBlockVis

18. Why did you like this design the least?

----

19. Please provide any general comments on Cluster3D

----

20. Please provide any general comments on TreeCube

----

21. Please provide any general comments on 3DBlockVis
