

# The Effect of Animated Transitions on User Navigation in 3D Tree-Maps

Thomas Bladh  
*Dept. of Computer Science  
and Electrical Engineering  
Luleå Univ. of Technology  
SE-971 87 Luleå, Sweden  
tbladh@sm.luth.se*

David A. Carr  
*Dept. of Computer Science  
and Electrical Engineering  
Luleå Univ. of Technology  
SE-971 87 Luleå, Sweden  
david@sm.luth.se*

Matjaž Kljun  
*Faculty of Education Koper  
University of Primorska  
Cankarjeva 5  
6000 Koper, Slovenia  
mkljun@pef.upr.si*

## Abstract

*This paper describes a user study conducted to evaluate the use of smooth animated transitions between directories in a three-dimensional, tree-map visualization. We looked specifically at the task of returning to a previously visited directory after either an animated or instantaneous return to the root location. The results of the study show that animation is a double-edged sword. Even though users take more shortcuts, they also make more severe navigational errors. It seems as though the promise of a more direct route to the target directory, which animation provides, somehow precludes users who navigate incorrectly from applying a successful recovery strategy.*

## 1. Introduction

Animation is a curiously neglected area of academic study, even though its history is quite long and its use widespread. When the desktop metaphor started to emerge as the primary user interface concept, animation evolved along with it. Animation is now used for expanding icons into windows, moving windows, scrolling their contents, and in displaying menus. Many of these animations are so commonplace that we hardly notice them. Even more sophisticated animation behaviors such as fading and morphing are becoming increasingly common as memory and processing power becomes less of a limiting factor.

StepTree is a 3D file-system visualization tool based on the tree-map concept. When run on the current generation of workstations, it is capable of rendering tens of thousands of files and directories simultaneously. In an earlier user study [3], we compared StepTree to its 2D counterpart, Treemap 4.05 from the Human Computer Interaction Lab (HCIL) at the University of Maryland [9]. StepTree performed comparably to Treemap, and it even excelled in a task where participants were asked to find the deepest directory.

However, while working with both tools, we observed that something was causing confusion.

Navigating with either tool consisted of a number of abrupt transitions as users browsed among directories. When navigating tree-maps there is usually some overlap between views. Either some part of the original view becomes the new view (expand), or the original view becomes a smaller part of the new (collapse). During these operations participants sometimes had difficulty reconciling the two views. We believe that this is in part due to the properties of the “squarified” tree-map layout algorithm [5]. That algorithm tries to lay out all files in an arrangement as close to a square as possible. Expanding and collapsing a directory may cause that directory’s aspect ratio to change. This, in turn may result in changes in the locations and aspect ratios of its descendants. Therefore, after selection, the new view of a directory often differs in more than size.

Neither Treemap nor StepTree animates transitions on selection. It became apparent that participants often became confused after entering a particular directory, especially if the new directory was several levels from the current one or had an aspect ratio different from the current directory. This form of transitional confusion is not a new problem and has been the subject of study in cognitive psychology [10, 24], visualization [6], and virtual reality [4, 14].

After adding smooth (at 25 frames per second) animated transitions to StepTree, we felt that the users did gain a deeper understanding of the structures explored during navigation. Lacking anything but anecdotal evidence, we decided that a small user study might serve to bring clarity. The purpose behind the study presented in this paper is to clarify and if possible, quantify exactly, the effect of animated transitions.

The rest of the paper is organized as follows: Section 2 describes previous work in the area. Section 3 provides an overview of StepTree with an emphasis on our implementation of animated transitions. Section 4 summarizes the user study that we conducted to meas-

ure the effect animation has on navigational performance and behavior. Finally, section 5 wraps up with conclusions and our plans for the future.

## 2. Previous Work

This section gives a general overview of previous work on space-filling visualizations as well as a more comprehensive overview of research on animation. The different types of space-filling visualizations discussed include tree-maps, radial space-filling visualizations, and information cubes. The section on animation includes work on animated transitions in virtual reality, information visualization, and cognitive psychology, as well as on animation in traditional user interfaces.

### 2.1. Space-Filling Visualization

Tree-maps belong to the group of visualizations called “space filling”. Traditionally, node-link diagrams are used to represent trees and hierarchies. As an alternative, space-filling visualizations use enclosure or adjacency to express both set and hierarchical relationships. In a tree-map each node is a rectangle with an area proportional to one of its attributes such as size or value. Nodes can be nested to form parent and child relationships. Tree-maps are, therefore, a form of nested Venn diagram where all internal space is used.

Shneiderman, in his seminal paper [17], outlines the theoretical foundation for tree-maps. Elaborating on the original concept, Johnson and Shneiderman [12] provide a more user-centered view of tree-maps and introduce them as an alternative method for viewing large file systems. They also introduce some basic usability issues that require additional research. Among these are: aesthetics, interactivity, comprehension, and efficient space utilization. Space utilization includes topics such as: layout, filtering, zooming (including traversing the hierarchy), coloring, and labeling of files. Turo and Johnson [21] present an empirical study, consisting of two experiments that demonstrate the advantages of tree-maps when compared to traditional command line interfaces and financial reports on paper. Since their introduction, tree-maps have been used to visualize a wide range of hierarchical structures such as stock portfolios [13], tennis matches [11], and photo collections [1].

StepTree is a variation of the tree-map concept that extends the rectangular, space-filling methodology to three dimensions by stacking levels of the hierarchy one on top of the other. Similar tools such as FSV [16] and VisFS [20] have existed for some time, but they have not had their usability tested. Our comparison [3] of StepTree to its 2D counterpart Treemap 4.05 [9] showed the 3D layout to have the advantage when de-

termining depth in the hierarchy without sacrificing performance on other tasks.

The problem of clearly delimiting the structure of the hierarchy in a tree-map has been the subject of several other investigations. In their study [22], van Wijk and van de Wetering propose a shading technique called “cushioning” as a solution. Wattenberg and Fisher [23] describe an algorithmic method for automatically analyzing the visual organization of arbitrary grayscale images. Using this method to analyze tree-maps, they find that the use of thick outlines might help to convey the underlying structure of the graph better than the current faint outlines.

Another member of the class of space-filling visualizations is the Information Cube [15], which is a true 3D analog of the tree-map. However, Wiss and Carr [25] found that the Information Cube was difficult to use even with relatively small data sets.

The radial SunBurst [18] is also a space-filling visualization and can be likened to a hierarchical pie chart. It is based on an angular subdivision of concentric circles rather than, as is the case with both tree-maps and information cubes, a hierarchical subdivision of space.

### 2.2. Animation

Since the birth of the graphical user interface, animation has been used to illustrate the changes between interface states. For example, the Macintosh used expanding rectangles to illustrate the opening of an icon into a window. The general idea behind animation is to help users relate two states of a system. But does animation really help? If so, how? Does it improve navigational performance, and if so, how?

Stasko and Zhang [19] describe three interesting focus+context navigational techniques applied to the radial, space-filling visualization, SunBurst. What the three techniques have in common is that, through an animated expansion, they give users details regarding a selected branch or node of the tree while still keeping the original data in view. Our expansion animation is conceptually similar even though our animation technique does not use focus+context.

Gonzales [8] performed one of the first user studies looking at how animation helps users to make decisions. Her empirical study showed that the effect of animation is closely related to its properties. This includes image realism, transition smoothness, and interaction style. The task domain and the user's experience also affect performance. Smooth animation was shown to have a greater positive effect on task accuracy than more abrupt animations. The use of realistic images was also shown to have a greater positive effect on task performance than more abstract imagery.

A study by Donskoy and Kaptelinin [7] compared three different navigational techniques (scrollbars, zooming, and fish eye), with and without animation. Animation was accomplished by inserting a single additional frame between the initial and final display states. The results did not show any significant improvement in favor of animated transitions. The authors concede that only one intermediary frame might have been inadequate. To achieve smoothness of movement, 10 frames per second are generally considered the minimum required frame rate [6].

Bederson and Boltman [2] conducted a user study where they evaluated the effect of viewpoint animation in a 2D panning interface on the ability of users to build mental maps of spatial information. They hypothesized that animation would help users navigate and later reconstruct the information space by facilitating recall of previously seen information. Their results indicate that animated transitions do help users learn relationships such as the spatial location of a picture in a family tree. But, animation was found not to help users to learn more complex logical relationships such as the actual family relationships.

The use of animation for moving the viewpoint in 3D virtual-reality environments is similar to the corresponding use of movement in 2D environments. When this movement has infinite speed, the effect produced is often called “teleportation”. Problems related to teleportation and their solutions are explored in a study conducted by Bowman et al. [4]. It showed that instantly changing location without giving users a clue as to how they moved increases disorientation – most likely because there is no analog for this type of motion in the real world.

When describing their “Information Visualizer” Card et al. [6] discuss issues such as the loss of object constancy due to transitions becoming too abrupt. Conversely, transitions that take too long run the risk of becoming tedious, and in the end, annoying to users. One solution to the problem of abrupt transitions is to make the transition gradual by either introducing a brief but smooth animated transition of the viewpoint, or if the visualization is object centric, an animated transforma-

tion of the object or objects.

Animations, some argue, have their own problems. Pausch, et al. [14], for example, mention the problems of compressing long distances into short animations and of avoiding obstacles during viewpoint motion.

“Visual momentum” is a term originally coined by Hochberg and Brooks [10] and later expanded by Woods [24]. In their seminal study on visual momentum, Hochberg and Brooks [10] show a relationship between the complexity and length of successive film shots to “glance rate”. They described the term visual momentum as, “an impetus to gather visual information”. After a scene change, the glance rate increases and then starts to drop off as we familiarize ourselves with the scene. Thus, skillful editing of a motion picture serves to continually pique the viewers’ interests while not confusing them too much. The definition of visual momentum given by Woods [24] is more abstract. He defines visual momentum as being inversely proportional to the mental effort required to integrate a succession of views. Thus, the use of smooth animated transitions between views would be indicative of a user interface supporting higher visual momentum.

Animation obviously has great aesthetic appeal; otherwise, it would not have been widely adopted. But even though the case for its positive impact on usability may seem clear cut, it arguably lacks strong empirical backing.

### 3. StepTree and View Transitions

StepTree’s user interface design is similar to that of Treemap from HCIL at the University of Maryland. It is a 3D (more accurately 2½D), space-filling visualization of file-system hierarchies where nodes are visualized as boxes, and relationships between them are

determined by properties of enclosure and adjacency. Boxes are assigned an area proportional to the size of the file or directory that they represent. Colors are determined by file type. Nodes can also be filtered and subsequently hidden based on modification date.

Figure 1 shows a screen shot of StepTree. The tabbed panel on the left provides controls for dynamic-query filters

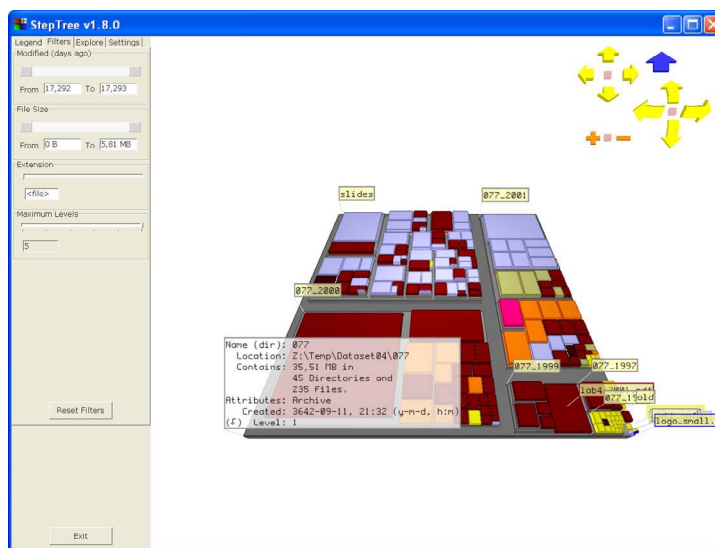
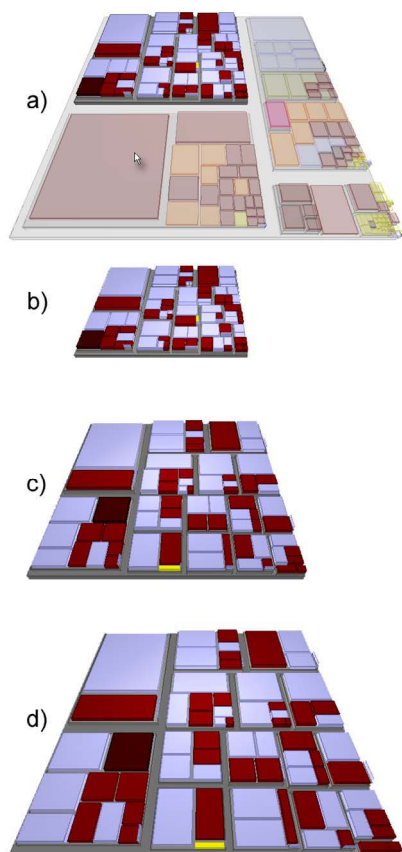


Figure 1. The file system visualization tool StepTree

and configuration. The icons at the top-right are for navigation. Navigational controls are also assigned to mouse motion with depressed buttons. The tree-map itself is displayed in the center. Translucent labels appear on flags for the current directory and its immediate children. Details are provided for items on mouse-over. StepTree is intended to act as a test bed for design ideas and has thus been redesigned several times. Incremental improvements such as: dynamic queries, animated transitions, adaptive on-screen controls, and navigational restrictions have all been incorporated. Our description of StepTree will focus on navigation among various views of the file hierarchy. We can divide these roughly into structural-view navigation (navigating the hierarchy), and spatial-view navigation (navigating the 3D space). It is on the structural-view navigation that we focused in our study.

Navigating into the hierarchy structure using StepTree entails clicking on a directory (or its label). This causes a view of the selected directory to completely replace the view of the current directory. Depending on whether or not animation is enabled, this transition is



**Figure 2.** The animation for entering a directory begins with a fade-out (a) followed by a short pause (b) after which the new current directory is expanded to fill the space formerly occupied by its ancestor (c, d).

either gradual or instantaneous.

Animation during transitions is used to demonstrate the relationship between directories, particularly ancestors and subdirectories. We believe that this helps users learn the relationships between different levels of the hierarchy. The exact animation behavior depends on whether you are traversing down the hierarchy or up. When traversing down you select a subdirectory of the current root by clicking it. Next, everything but that directory starts to fade away (Figure 2:a). The selected subdirectory is now left alone on-screen (Figure 2:b), in the same position and size as it had before its surroundings started to fade. Next (Figure 2:c), the directory is gradually enlarged and moved until its base is in the same location and of the same size as the previous directory (Figure 2:d). When traversing up in the hierarchy the above procedure is reversed. The current directory is shrunk and moved into the position it will occupy on the new tree-map image. Then, the remainder of the new current directory fades into view around the previous directory. We will for the remainder of this paper refer to these methods as “transformations”.

In our first design, the animation was timed so that the whole sequence always took the same amount of time regardless of the magnitude of the transformation required. In accordance with what seems to be an accepted upper limit on animation time [6], the animation was set to last exactly 1 second. After some use, this method seemed unnatural. Entering a small (in terms of the surface area assigned to it), subdirectory of the current directory should take longer than entering a large subdirectory. We therefore decided to divide the animation into a fixed fade in/out step of 400 ms, an intermediary 50 ms pause, and a variable transformation sequence of at most, 550 ms. The length of the transformation sequence depends on the magnitude of the transformation. What we did, in short, is to make transformations constant speed rather than constant time.

Animating transitions between directories may have two possible benefits. First, users can see the directory enlarging and moving from its previous location into the new location. If the user were to click on and enter the wrong directory, it should be easier to realize the error with animation. If, on the other hand, the transition had been abrupt the user might easily continue working in the wrong directory. Second, the animation emphasizes the relation between the subdirectory and its parent. This may simplify learning the directory structure.

Both of these advantages should lead to shorter interaction times, especially on tasks where the user returns to a previous location. These “return” tasks are frequent. Users tend to begin navigational activities at some root location (e. g., “C:”, “/”, “My Documents”, etc.), and it is likely that they will return there from

time to time to begin searching for something else. Examples of this type of task include: finding a file (documents, multimedia, etc.), moving or copying a file (managing your files), and relating files in different locations to each other (comparing two directories).

## 4. User Study

As we describe in Section 3, animations were introduced to make the transitions between directories less abrupt and more informative. Naturally, we felt that the possible presence and magnitude of such effects needed study. We therefore decided to conduct a small between subjects experiment.

### 4.1. Tasks

The task set used consists of two types of tasks: Tasks in the first set (tasks 1, 2, 4, 5, and 7), all involve returning to a previously visited directory either immediately after pressing the home button or after completing an unrelated intermediate task. We will refer to this type of task as a “return” task. The second set is a small collection of three tasks (tasks 3, 6, and 8), taken from our previous study [3]. These tasks were chosen to diversify the task set and because we felt that there was no reason that they should benefit from animation. A point sometimes made is that “animation takes time” [2]. Thus, if there is no benefit, the net effect should be longer task times. We will refer to tasks from this second set as “general” tasks. The tasks were:

#### 1 Tutorial Task

- a) Locate a directory given a path and select the largest file in it.
- b) Click on the home button (returning to the root).
- c) Return to and select the directory in a).

#### 2 Practice Task

- a) Locate a directory given a path and select a file with a given size (theme: “fruit”).
  - b) Click on the home button (returning to the root).
  - c) Return to the “fruit” directory in a) and select a file with a given name.
- 3 Locate and select the smallest file of the type “PDF”.
  - 4 Same as 2 but with a different target directory. (theme: “architecture”)
  - 5 Same as 2 but with a different target directory. (theme: “animals”)
  - 6 Locate and select a file given a path.
  - 7 Return to the directory from task 5 with the animal theme.
  - 8 Select the directory containing the only file of a certain type.

All return tasks except for task 1 are based on the themes “fruit” (task 2), “architecture” (task 4), and “animals” (tasks 5 and 7). The use of themes means that directories along the path to the target directory, and the files in the target directory had names based on a particular concept. This is a way of providing a non-visual memory cue for task 7. When the participant is located at the root, these theme-based file and directory names are not visible. To complete tasks 4 and 5, participants had to bring up detailed information (by way of a mouse over), on files in the target directory to determine which of them was of a certain size. While doing this they are also exposed to the theme related filenames (“chicken.html”, “cow.html”, etc.), which should reinforce their memory of the themed directory. This pattern gave participants a way to solve task 7 without providing a visual description that would have been directly observable on the display. (For example, “Return to the directory with the checkerboard pattern.”) The experiment relied on navigational decisions being based on participants remembering either the path to or the spatial location of the target directory.

### 4.2. Procedure

Sixteen students (all save one recruited from a Human Computer Interaction class at Luleå University of Technology), volunteered to participate in our study. Participants ranged in age from 21 to 24 years, with an average age of 22.4 years. Participants used computers on an average of 35.6 hours a week, and all but two participants had experience with 3D games. Those with 3D game experience spent an average of 8 hours per week on this activity. Five participants were female, and 11 were male. The two participants without 3D game experience were both female. All tests were conducted on a 1.7 GHz Pentium 4 workstation at a resolution of 1024x768 pixels on a 21-inch monitor. The primary input device was a mouse. A keyboard was present, but participants were not instructed to use it.

Participants were given movie tickets as an incentive to participate, and these tickets were handed out at the start of the test sessions. Participants were then asked to read and sign a consent form followed by a pre-test questionnaire with questions about gender, age, and experience with computers and 3D games, as well as time spent per week on computers and 3D games. The test began with a tutorial session centered on a tutorial task (task 1). A detailed checklist was used to make sure that the participants received the same training. After an introduction to the tool, the participant was encouraged to explore it freely until he or she felt ready to finish the task. During pilot testing we had observed that it took a while for participants to notice and base decisions on the animation effect. For

this reason, we decided to consider task 2 a practice task and not analyze it. The tutorial and practice tasks were followed by the actual test (tasks 3 through 8).

Treatments (non-animated or animated) were randomly assigned to time slots and participants in pairs. (A pair was two successive slots in the experiments schedule.) The test was automated, and all action by the test conductor ceased once the tutorial and practice tasks were over. Participants could start the next task at their own leisure by clicking on a button. Tasks were shown at the bottom of the screen. Participants ended the tasks themselves by clicking on a “Done” button when they felt that they had completed the task. There was no time limit, but participants were told that they could elect to end the task without completion if they had seriously tried but failed to solve it (this was classified as an error). Timing was performed by logging all relevant events (task start, mouse clicks, clicking done, etc.), to a file. A post-test questionnaire was used to capture subjective impressions through a series of 12 statements to which participants could indicate either agreement or disagreement.

### 4.3. Hypotheses

Our primary hypothesis was that animation would result in more shortcuts on the return tasks (4, 5, and 7). There should thus be a statistically significant increase in the number of shortcuts taken, with the animated treatment. If shortcuts are taken there should also be gains in time spent on the task, and consequently, we believed that there should be a statistically significant difference in task times for tasks 4, 5, and 7 favoring animation. There should be no statistically significant difference in task times for tasks 3, 6, and 8. These more general tasks should, in terms of completion time, neither benefit nor suffer from the introduction of animated transitions.

### 4.4. Results

The primary hypothesis was upheld. Participants using animation were more likely to take shortcuts. Navigational behavior was categorized as to whether participants completed a task in fewer steps than the length of the path (Shortcut), as many steps as the length of the path (Path), or either took more steps than the path or made an error (Miss). We calculated frequencies for these types of navigational behaviors for all return tasks combined. (See Table 1.) These show a statistically significant effect of animation on navigational behavior ( $\chi^2 = 9.14$ ,  $df=2$ ,  $p < 0.025$ ) for return tasks. Participants were nearly four times as likely to take a shortcut when using animation. If we investigate this further and look at the raw step data for return

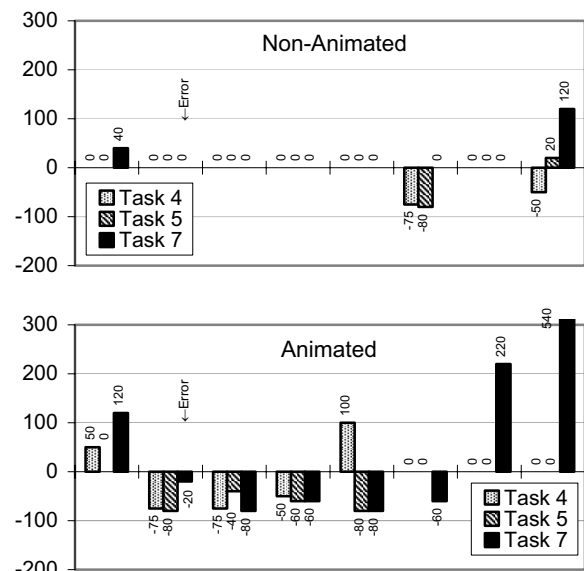
**Table 1. Frequencies of navigation behaviors for tasks 4, 5 and 7 combined.**

Categories	Non-Animated	Animated
Shortcut	3	11
Path	17	7
Miss	4	6

tasks (Figure 3), an interesting picture emerges. In the non-animated group, performance tends strongly towards the path length. In contrast, the number of steps taken by participants in the animated group is more varied. When animation is used, a greater number of scores fall both above and below the path length.

This variation evident in the animated group is most pronounced on task 7 where participants were required to remember either the path or location of a directory. Here, four participants managed a shortcut, three got lost, and one made an error. One participant went on a journey exceeding the path length by 540% (the truncated bar in the bottom chart of Figure 3).

ANOVA’s on the effect of treatment on task times for tasks 4, 5, and 7 both separately and combined however, show no significant effects for treatment. For task 7, this result can be readily explained by the great variation in the number of steps taken by participants in the animation group. Any gains from shortcuts are swamped by time spent recovering from navigational mistakes. Overall, for return tasks, variation within the treatments seems to have swamped any variation between them. In short, the experiment was probably not sensitive enough to detect any differences. The return tasks may not have been comprehensive enough for any significant differences to be detected, and paths to



**Figure 3. Deviations from the direct path length in percent for “return” tasks.**

the target directories (at four and five steps respectively), may have been too short. It might also have been useful to repeat navigating into and later returning to, different target directories several times.

There was no significant effect of treatment on task times for the general tasks (3, 6, and 8), either separately or combined. This would suggest that time spent on transitional animations does not cause a significant negative effect on tasks in general.

The overall error rate for tasks 3 through 8 was seven on a total of 96 completed tasks, which can be considered low. Of those, four were committed by participants in the non-animated group, and three by participants in the animated group.

Looking at the post-test questionnaire data (Figure 4), two statements resulted in responses different enough between participants who used the two versions of StepTree to be worthy of note. The first is the statement: "I never felt lost." Participants in the animated group seemed to more strongly agree with this statement. The second is the statement: "Transitions between directories were smooth." It was included in the questionnaire to determine whether participants really noticed the animated transitions. Participants in the animated group seem to agree more strongly with this statement than those in the non-animated group.

#### 4.5. Discussion

That we failed to find an effect of animation on task times for the return tasks was a bit unexpected but seems to jibe with the findings of others [2, 7]. Our analysis of participants' navigational behaviors showed significantly more shortcuts being taken by participants in the animated treatment. Thus, we suspect that an effect on task time does exist even though we failed to

find one in this study. The effect, if it exists, is likely to be weak and might require more comprehensive navigational tasks and more training for participants for it to be noticeable.

That participants in the animation group who got lost on task 7 seemed to get so thoroughly lost is very intriguing. A majority of participants in the non-animated group did remember and successfully followed the five-step path on task 7 (6 out of 8). In the animated group no one followed the path. Instead, all tried their luck at a taking a shortcut. Those who remembered correctly completed the task in fewer steps than the stepwise path, whereas those who failed paid a high price in the number of steps it took for them to recover from their mistake.

What we seem to see in the results on task 7 (and less strongly on tasks 4 and 5), is a contrast between orderly navigational behavior in the non-animated group and a more chaotic behavior in the animated group. In the animated group about half seem to succeed in taking a shortcut, putting them in a distinct group, performance wise, from those who do not. From this it would appear that animation is a double-edged sword. Participants using animation appear to have difficulty applying an effective recovery strategy when their shortcut goes wrong. Within the limits of our study, participants do not always have adequate spatial memory of previously visited directories. However, they are apparently slow to realize this, and they delay before returning to the root and tracing the path. In contrast, participants without animation almost uniformly retrace the path on revisits. This strategy is quite successful. It would appear that there are sufficient cues to trigger participants' long-term memory if they retrace the path.

#### 5. Conclusions and Future Work

Our study shows how navigational behavior in an object-centered, 3D environment is affected by the introduction of transitional animations. Users learn spatial relationships through animation, but the supposed benefits seem to come at a high price. Participants who try to return by memory alone and who misjudge the spatial location of the target directory seem unable or unwilling to revert to a more reasonable stepwise navigational strategy. Instead they continue ineffectually trying to locate the target directory through a series of failed shortcut attempts.

Some alternate designs have been suggested that might improve animated transitions. One suggestion was to fade the surrounding visualization while keeping it in view, continuously changing its size and layout in step with the expansion of the target directory. Another was to give collapsed directories a gradually

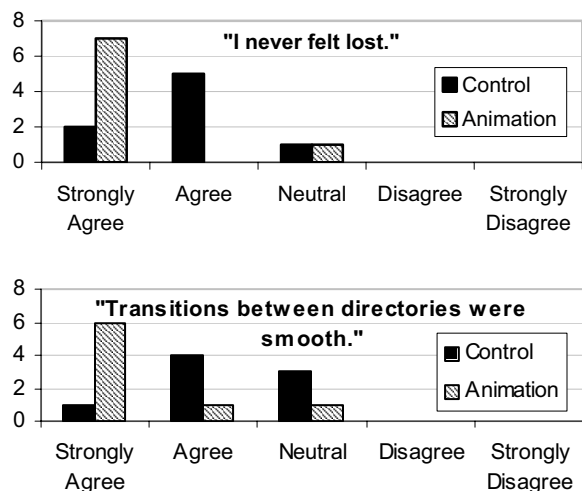


Figure 4. Responses to selected questionnaire statements.

diminishing glow, further reinforcing the connection established through animation.

Humans are well adapted to recognizing landmarks in the environment. We learn how to get from place to place not by memorizing the length of every stretch of road and the degrees of every turn. Instead we move from landmark to landmark. To minimize the cognitive effort, we divide the problem of navigation into manageable chunks. When we introduce animated transitions (showing us “artificially” how one place relates to another), it might be that this short-circuits our landmark navigational mechanism. We become predisposed to apply a different navigational strategy. Instead of relying on waypoints for navigation, we put our faith in our fading memory of the transition and the information that we can derive from it.

We would very much like to see a more comprehensive study on the impact of animation. Our study has been exploratory in nature but has given us a great deal to think about. Even though we have been able to show that certain differences exist in navigational behavior, we would like to more thoroughly explore how animation affects navigation and how to best exploit it in design.

## Acknowledgements

We would like to thank those who participated in both the study and pilot testing. We would also like to thank Carl Rollo for proofreading the drafts of this paper.

## References

- [1] Bederson, B. B., PhotoMesa: A Zoomable Image Browser using Quantum Treemaps and Bubblemaps, *Proceedings of the 2001 ACM Symposium on User Interface Software and Technology, CHI Letters*, 3(2), Orlando, FL, 11-14 Nov. 2001, 71-80.
- [2] Bederson, B. B. and Boltman, A., Does Animation Help Users Build Mental Maps of Spatial Information?, *Proceedings of INFOVIS'99*, San Francisco, CA, 24-29 Oct. 1999, 28-35.
- [3] Bladh, T., Carr, D., and Scholl, J., Extending Tree-Maps to Three Dimensions: a Comparative Study, *Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI 2004)*, Rotorua, New Zealand, 29 June - 2 July 2004, 50-59.
- [4] Bowman, D. A., Koller, D., and Hodges, L. F., Travel in Immersive Virtual Environments: an Evaluation of Viewpoint Motion Control Techniques, *Virtual Reality Annual International Symposium '97*, Albuquerque, NM, 1-5 March 1997, 45-52.

- [5] Bruls, M. Huizing, K. and van Wijk, J. J., Squarified Treemaps, *Proceeding of Joint Eurographics and IEEE TCVG Symposium on Visualization*, Amsterdam, the Netherlands, 29-30 May 2000, 33-42.
- [6] Card, K., Robertson, G. G., and Mackinlay, J. D., The Information Visualizer, an Information Workspace, *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, New Orleans, LA, May 1991, 181-188.
- [7] Donskoy, M. and Kaptelinin, V., Window Navigation With and Without Animation: a Comparison of Scroll Bars, Zoom, and Fisheye View, *Extended Abstracts of CHI'97*, Atlanta, GA, 22-27 March 1997, 279-280.
- [8] Gonzales, C., Does Animation in User Interfaces Improve Decision Making?, *CHI '96: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Vancouver, BC, Canada, 13-18 April 1996, 27-34.
- [9] HCIL, Treemap, <http://www.cs.umd.edu/hcil/treemap/>, January 2005.
- [10] Hochberg, J. and Brooks V., Film Cutting and Visual Momentum, in Senders, J.W., Fisher, D.F., and Monty, R.A. (Eds.), *Eye movements and the higher psychological functions*, Erlbaum, Hillsdale, NJ, 1978.
- [11] Jin, L. and Banks, D. C., TennisViewer: a Browser for Competition Trees, *IEEE Computer Graphics and Applications*, 17(4), July/Aug. 1997, 63-65.
- [12] Johnson, B. and Shneiderman, B., Tree-Maps: A Space-Filling Approach to the Visualization of Hierarchical Information Structures, *Proceedings of IEEE Visualization '91*, San Diego, CA, Oct. 1991, 284-291.
- [13] Jungmeister, W. A. and Turo, D., Adapting Treemaps to Stock Portfolio Visualization, Univ. of Maryland Center for Automation Research, tech. report CAR-TR-648, 1992.
- [14] Pausch, R., Burnette, T., Brockway, D., and Weiblen, M., Navigation and Locomotion in Virtual Worlds via Flight into Hand-Held Miniatures, *Proceedings of SIGGRAPH'95*, 1995, 399-400.
- [15] Rekimoto, J. and Green, M., The Information Cube: Using Transparency in 3D Information Visualization, *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, Orlando, FL, December 1993, 125-132.
- [16] Richard, G. D., File System Visualizer (FSV), <http://fsv.sourceforge.net>, January 2004.
- [17] Shneiderman, B., Tree Visualization with Tree-Maps: a 2-D Space-Filling Approach, *ACM Transactions on Graphics*, 11(1), Jan. 1992, 92-99.
- [18] Stasko, J., Catrambone, R., Guzdial M., and McDonald, K., An Evaluation of Space-Filling Information Visualizations for Depicting Hierarchical Structures, *Inter. Journal of Human-Computer Studies*, 53(5), Nov 2000, 663-694.

- [19] Stasko, J. and Zhang, E., Focus+Context Display and Navigation Techniques for Enhancing Radial, Space-Filling Hierarchy Visualizations, *Proceedings of IEEE Information Visualization 2000*, Salt Lake City, UT, Oct. 2000, 57-65.
- [20] Schmidt, H., Visual File System (VisFS), [http://www.heiko-schmidt.info/project/visfs/visfs\\_de.html](http://www.heiko-schmidt.info/project/visfs/visfs_de.html), Jan. 2004.
- [21] Turo, D. and Johnson, B., Improving the Visualization of Hierarchies with Treemaps: Design Issues and Experimentation, *Proceedings of IEEE Visualization '92*, Boston, MA, 19-23 Oct. 1992, 124-131.
- [22] van Wijk, J. and van de Wetering, H., Cushion Tree-maps: Visualization of Hierarchical Information, *Proceedings of INFOVIS'99*, San Francisco, CA, 25-26 Oct. 1999, 73-78.
- [23] Wattenberg, M. and Fisher, D., Analyzing Perceptual Organization in Information Graphics, *Information Visualization*, 3(2), March 2004, 123-133.
- [24] Woods, D. D., Visual Momentum: a Concept to Improve the Cognitive Coupling of Person and Computer, *International Journal of Man-Machine Studies*, 21(3), Sept. 1984, 229 - 244.
- [25] Wiss, U. and Carr, D., An Empirical Study of Task Support in 3D Information Visualizations, *Proceedings of the IEEE Conference on Information Visualization*, London, UK, 14-16 July 1999, 392-399.