Does Animation Help Users Build Mental Maps of Spatial Information?

Benjamin B. Bederson
Computer Science Department
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
+1 301 405 2764
bederson@cs.umd.edu

Angela Boltman
College of Education
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
aboltman@umiacs.umd.edu

ABSTRACT

We examine how animating a viewpoint change in a spatial information system affects a user’s ability to build a mental map of the information in the space. We found that animation improves users' ability to reconstruct the information space, with no penalty on task performance time. We believe that this study provides strong evidence for adding animated transitions in many applications with fixed spatial data where the user navigates around the data space.

Keywords

Evaluation, animation, real-time computer graphics, Zoomable User Interfaces (ZUIs), multiscale interfaces, Pad++.

INTRODUCTION

During the past decade, researchers have explored the use of animation in many aspects of user interfaces. In 1984, the Apple Macintosh used rudimentary animation when opening and closing icons. This kind of animation was used to provide a continuous transition from one state of the interface to another, and has become increasingly common, both in research and commercial user interfaces. Users commonly report that they prefer animation, and yet there has been very little research that attempts to understand how animation affects users' performance.

A commonly held belief is that animation helps users maintain object constancy and thus helps users to relate the two states of the system. This notion was described well by Robertson and his colleagues in their paper on "cone trees", a 3D visualization technique that they developed.

> "Interactive animation is used to shift some of the user's cognitive load to the human perceptual system. ... The perceptual phenomenon of object constancy enables the user to track substructure relationships without thinking about it. When the animation is completed, no time is needed for reassimilation." [19, p. 191].

Researchers including Robertson have demonstrated through informal usability studies that animation can improve subjective user satisfaction. However, there have been few controlled studies looking specifically at how animation affects user performance. These studies are summarized below.

Animation takes time

One potential drawback of adding animation to an interface or visualization is that animation, by definition, takes time. This brings up a fundamental trade-off between the time spent animating and the time spent using the interface. At one extreme with no animation, system response can be instantaneous. Users spend all of their time using the system. However, the user may then spend some time after an abrupt transition adjusting to the new representation of information and relating it to the previous representation.

At the other extreme, each visual change in the interface is accompanied by a smooth transition that relates the old representation to the new one. While developers of animated systems hope that this animation makes it easier for users to relate the different states of the system, there is clearly a trade-off in how much time is actually spent on the transition. If the transition is too fast, users may not be able to make the connection, and if the transition is too long, the users' time will be wasted. The ideal animation time is likely to be dependent on a number of factors, including task type, and the user's experience with the interface and the data. In pilot studies and our experience building animated systems, we have found that animations of one second seem to strike a balance. Others have also found one second animations to be appropriate [9, p. 185].

In the worse case, animations can be thought of as an increase in total system response time. Typically, system response time is defined to mean the time between when the user initiates an action, and when the computer *starts* to display the result. This definition comes from the days of slow displays on computer terminals. This metric was chosen because users could start planning their response as soon as the first data were displayed. With many animations, however, the user does not see the relevant data until the animation is nearly finished, and thus the animation time is an important part of system response. We thus define the *total system response time* to include the animation time Figure 1).

In many application domains, the system may need some time to gather data (such as with the World Wide Web), or to process it. In these cases, inserting an animation where a delay is necessitated is not likely to harm productivity because users would have to wait anyway. However, since the delay associated with the Web is often hard to predict, matching animations to the Web retrieval time could be difficult. The bigger problem is when the computer could have responded instantly, and the animation slows down computer's response time.

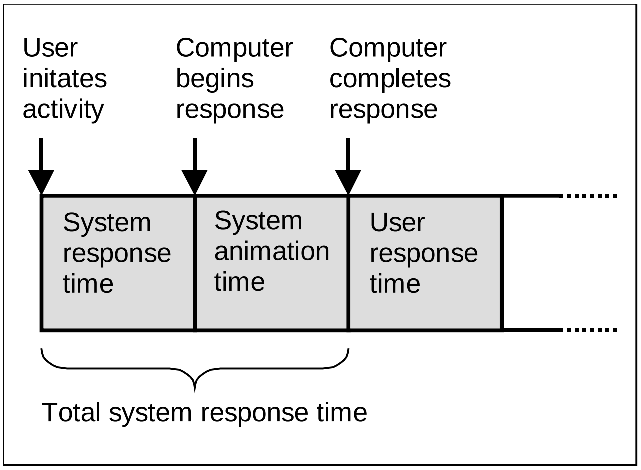

Figure 1: Model of user interface timing with animation (adapted from [21, p. 353]).

Researchers have been studying system response time since the 1960s, and as it happens, users' responses to system delays are more complex than it may at first appear. There is much research showing that user satisfaction decreases as delays increase (see the recent report on the World-Wide Web for a typical example [20]). However, this satisfaction does not necessarily correlate with performance. One paper showed that users pick different interaction strategies depending on system response times [24]. This paper showed that users actual performance depends on a complex mix of the task, the delay, and the variability of the delay among other things. One typical study shows productivity increasing as delays decreased for data entry tasks [12]. However, another study found an increase in data entry productivity when delays *increased* to a point [3].

Thus, the fact that animations take time does not necessarily imply that they will hurt productivity. Since they may, in fact, reduce cognitive effort, as suggested by Robertson and others, we believe that animation may improve some kinds of task performance.

Types of computer animation

Animation in computer interfaces can actually mean many different things. Baecker and Small summarized many of the ways that objects can be animated on computer screens [1]. Animation can consist of moving a static object within a scene, or the object may change its appearance as it is moved. A scene may be larger than can fit on the screen, and the viewpoint can be changed with animated movement by rendering "in-between frames" part way between the starting and ending state. There are numerous other types of animations as well.

In general, animation is often used to help users relate different states in the interface. These changes can be in the data within the interface or in the interface itself. Some systems that use animation of the data include the Information Visualizer [9], Cone Trees [19], WebBook and WebForager [10]. Chang and Ungar discussed the application of animation principles from the arts and cartoons to user interfaces, showing that more than just simple movement of interface objects is possible [11].

Some researchers have investigated the use of animated icons [2]. Others have used animation to try to improve teaching how algorithms work [8, 23]. There are also several user interface systems that include explicit support for creating animated interfaces. A good example of this is the Morphic system [18].

Zoomable User Interfaces

We are interested in understanding animation because for the past several years, we have been exploring Zoomable User Interfaces [4, 5, 6, 7, 15, 17]. Zoomable User Interfaces (ZUIs) are a visualization technique that provides access to spatially organized information. A ZUI lets users zoom in and out, or pan around to view much more information than can normally fit on a single screen. We have developed a system called Pad++ to explore ZUIs.

The ZUIs we have built typically provide three types of animated movement, as well as other kinds of animated transitions such as dissolves. The three types of animated movement are motion of objects within a scene, manual change of viewpoint (through various steering mechanisms), and automatic change of viewpoint (during hyperlinks).

We believe that animating changes of viewpoint during hyperlinks are the most important kind of animation in Pad++, since these animations appear to help users understand where they are in the information space. They are also easy to understand and use. As one child using KidPad (an authoring tool for children within Pad++ said, “With [traditional hypertext] it is like closing your eyes and when you open them you’re in a new place. Zooming lets you keep your eyes open” [14].

We and others have also used Pad++ to make animated zooming presentations, and regularly receive very positive feedback from users. However, as HCI researchers, we want to develop an understanding of exactly where, if at all, animated ZUIs perform better than traditional approaches.

As we started to design a study that would help us to understand the benefits of ZUIs, we realized that the animation we employ is orthogonal to the use of scale to organize data. It is possible to have an interface with or without animation, and with or without a multi-scale structure. In order to understand ZUIs better, we decided to attempt to understand the effects of animated movement, and multi-scale structure separately.

Thus, in this paper, we start by examining the most basic and fundamental kind of animation used in ZUIs. We examine how animated changes of viewpoint during hyperlink transitions affects users' ability to build a mental map of a flat information space. We specifically chose not to investigate zooming or multiscale structures in this work because we felt zooming would be a confounding variable to the animation effects we are investigating.

Previous Studies

There have been few studies that looked specifically at how animation affects user's abilities to use interfaces. One study looked at transition effects (such as a dissolve) and animation of an object within the view [16]. This study found that both a dissolve transition affect and animated motion of objects within a scene independently helped users to solve problems. This study is important because it motivates the common belief and intuition that animation can help user's maintain object constancy and thus improve task performance. However, this study did not address animation of viewpoint which is the primary focus of the current study.

Another study that is perhaps more relevant looked at animating the viewpoint of a spatial information visualization [13]. That study compared users' ability to find items where more items than could fit in a single display were used. Different navigation techniques were used to move through the items (scrollbars, zoom, and fisheye view), and each navigation technique was tested with and without animation. In this experiment, the use of animation did not have a significant affect on any of the navigation techniques. However, animation was implemented with just a single in-between frame. For animation to be perceived as smooth apparent movement, there must be several in-between frames, and they must be shown quickly (typically greater than 10 frames per second, and preferably 20 or 30 frames per second). Thus it does not appear that the results accurately describe the effects of animation. One interesting aspect of the study that does appear significant, but not relevant to animation, is that the zooming visualization technique performed significantly better than either the scrollbar or fisheye view visualization technique.

EXPERIMENT

We performed a study to test the effectiveness of animation of viewpoint on subjects' ability to build mental maps of spatial information spaces. Our hypothesis was that animation would improve subjects' ability to navigate through the information space, recall information, and to reconstruct the information space. In addition, we hypothesized that subjects would prefer animation to the non-animated condition.

Equipment

The computer system used was a 166 MHz Pentium PC with 32 megabytes of RAM running Linux. The PC had 17” monitor and was running at a resolution of 1280x1024. Navigation was performed using only the left button of a standard 2-button mouse. The questions and tasks that were presented to the subjects were automated and recorded by a program that we wrote.

Stimuli

Subjects were asked to navigate family trees created in the Pad++ program. Subjects navigated these trees clicking on hyperlinks that were represented by yellow arrows. For each family member, subjects saw a single photograph. Above the photograph, subjects saw the family member's first name and yellow hyperlink arrows with the words "parent", "child", "sibling", and "spouse", where appropriate (Figure 2). In the non-animated condition, clicking on a link would result in the destination of the link immediately appearing on the screen. The animated condition would smoothly move the viewpoint to the destination, showing all of the places in-between the source and destination along the way. All animations took exactly one second, and they were animated at 20 frames per second with an average speed of 750 pixels per second. Animations used a "slow-in, slow-out" technique which means that the animation speed increases smoothly at the beginning, continues at a constant rate in the middle, and decreases smoothly at the end.

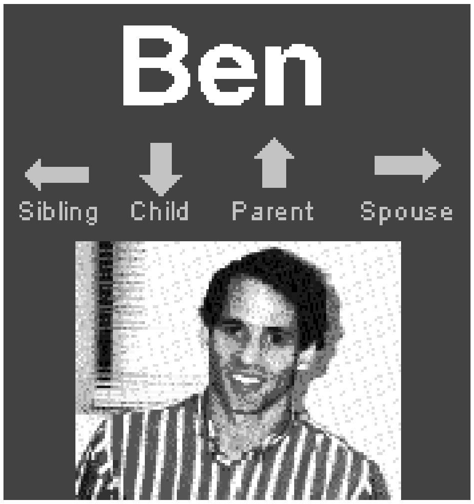

Figure 2: Presentation of person in a family tree. Arrows are hyperlinks for navigation. Note that the actual faces that were used in the experiment were completely unfamiliar to the subjects.

Two different fictitious family trees were used in the test – the Goodman family and the Flemming family. We matched the two treatments by creating two family trees that were exactly alike in both the structure and the number of family members the tree contained, although the names and photographs of the family members varied (Figure 3). This design has the risk that users would remember the structure of the tree. However, we decided that this risk was better than the alternative of having different structures that may not be comparable. Each subject was presented with one family tree with animation, and the other family tree without animation.

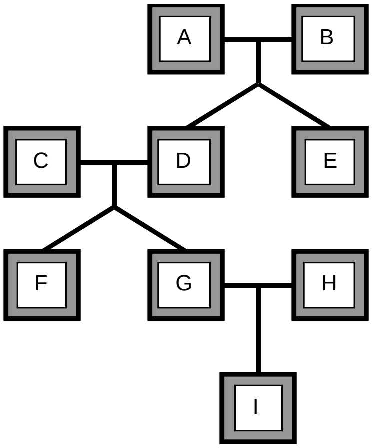

Figure 3: The structure of the family tree. This view was never available to subjects. Lines represent marriage and child relationships (i.e., A & B are married, and D & E are their children).

We created family trees with nine individuals in each tree, and displayed only one or two nodes at a time. We used this size because the goal of our experiment was to understand how people navigated information spaces where each view showed a subset of the complete information. In addition, we wanted to test memory, and if we had hundreds of family members, it would be more difficult to test how well subjects could learn the whole tree. In a pilot test, subjects specifically stated that the nine-node tree was a reasonable amount of information to try and learn – a challenge, but not impossible.

Training

Prior to beginning the experiment, subjects were trained in the use of Pad++, and in the nature of a family tree and its relationships. The goals of the training exercise were to verify that our subjects understood family relationships, and to orient them to the interface used in the experiment. In the training section of the experiment, they were given a family tree with and without animation and asked to answer four questions about the family relationships and the photographs. We monitored our subjects during their training and only accepted subjects into the experiment who completed their training tasks with 100% accuracy. No subjects were turned away due to errors in training.

Method

We used a 2x2x2 incomplete block design for this study. Each subject was given matched tasks for the two family trees, one with animation, and one without. Half of the subjects were given an animated family tree first; and half of the subjects were given an animated family tree second. The independent variables were transition type (with or without animation), order (animation first or non-animated first), and family tree (Goodmans or Flemmings).

Within both of the family trees, subjects were given three kinds of tasks. First, the subject was presented with a series of nine statements and asked to navigate the family tree until the subject felt that he/she had the appropriate information to determine if the statement was true or false. The questions were presented one at a time in randomized order. These questions were solely about the family relationships. (e.g., "Victor's sister is Margaret".) This task was given to evaluate speed of navigation.

The second set of tasks evaluated recall memory. The subject was given three minutes of exploratory time, where he/she could navigate the family tree at leisure. After this three minutes of exploratory time, the family tree was hidden, and the subject was presented with 10 multiple choice questions in random order. These questions were about both the family relationships and the photographs of the family members. An example of a relationship question was "Who was Billy's mother?". An example of a photograph question was "Who was holding a puppy?".

The third set of tasks evaluated reconstruction ability. The subject was given the off-computer task of assembling the photos of all the family members into the organization of the family tree. The subjects were given a stack of the nine photographs of the family tree members, and were asked to duplicate the family tree that they had seen in the computer interface. They were given as much time as they needed. The position of each photo was recorded for future analysis. The subject was not given the opportunity to specify specific relationships. Subjects could only specify position. Thus, in this part of the experiment, we could not distinguish between a marriage and a sibling relationship.

The subject then repeated all three activities for a second family tree. Lastly, the subject was asked to complete two user satisfaction surveys – one about their experience with animation, and the other about their experience without animation.

Seven dependent variables were analyzed in the experiment. For each of the three task types, we measured accuracy and speed. The time to complete a question was measured from the time the subject initiated a request for a new question until they completed the task by providing an answer. Only questions a subject completed correctly were included in the time averages.

The seventh dependent variable was subjective satisfaction. We measured this using a subset of the Questionnaire for User Interaction Satisfaction (QUIS) developed at the University of Maryland [22]. The satisfaction level of animation and non-animation was based on a series of questions: overall reaction, locating information, and remembering relationships. Subjects were also asked to give their opinions of the advantages and disadvantages of an interface with animation.

The task of reconstructing the family tree was performed away from a computer using printed photographs of each family member without the name of the person. For each subject, we recorded the time to complete the task and the number of errors. Because there were different kinds of errors, we measured them in two ways. Our goal was to try to understand if the kinds of errors that were made were different in the two conditions. First, we counted the number of faces placed in an incorrect position. Second, we weighted the errors by how far away from the correct position the faces were placed. For each incorrect placement, we assigned a weight of one if the face was simply swapped with another face, or if the face was away from its correct placement on the same row. We assigned a weight of two if the face was one row too high or low, and a weight of three if the face was two rows too high or low.

Subjects

Twenty subjects participated in the experiment. 55% of the subjects were male and 45% of the subjects were female. Subjects ranged in age from 20 to 44. Mean age was 27. All but one of the subjects were students at the University of Maryland at College Park.

Of the student subjects, eight were computer science majors, five were library and information science majors, two were engineering majors, one was an animal science major, one was a math major, one was a journalism major, and 1 was a clinical psychology major. 15% of subjects reported spending under 10 hours per week using a computer, 20% reported averaging 11-20 hours, 35% reported averaging 21-30 hours, 30% reported over 31 hours. Subjects were paid $10 for their participation.

Results

We observed a statistically significant improvement in accuracy of the reconstruction task by navigation type with both the unweighted count ($F_{1,19}$ = 16.165, p=.001) and the weighted count ($F_{1,19}$ = 16.816, p=.001) as reported in Table 1. The animation treatment had fewer errors than the non-animated treatment. There was no statistically significant difference in any of the other tasks or subjective satisfaction by animation. For an effect to be considered significant, p had to be less than or equal to 0.01.

Measure	Animated	Non-Animated
Navigation Task Mean time/question	18.6 secs	18.9 secs
Navigation Task Mean errors	0.3 errs	0.1 errs
Exploratory Task Mean time/question	7.2 secs	7.3 secs
Exploratory Task Mean errors	2.4 errs	2.0 errs
Reconstruction Task Mean time	37.8 secs	50.9 secs
Reconstruction Task Mean errors, unweighted	**0.4 errs**	**1.8 errs**
Reconstruction Task Mean errors, weighted	**0.4 errs**	**3.35 errs**
Subjective Satisfaction	6.0 (max 9)	5.9

Table 1: Summary results for experiment. Significant effects are shaded

There was also a significant ordering effect in the reconstruction task accuracy by order ($F_{1,19}$ = 8.780, p=.009). This ordering effect showed that if the animation was first, then there was little difference between animation and no animation. If the animation treatment was second, however, then performance was significantly better with animation (Figure 4).

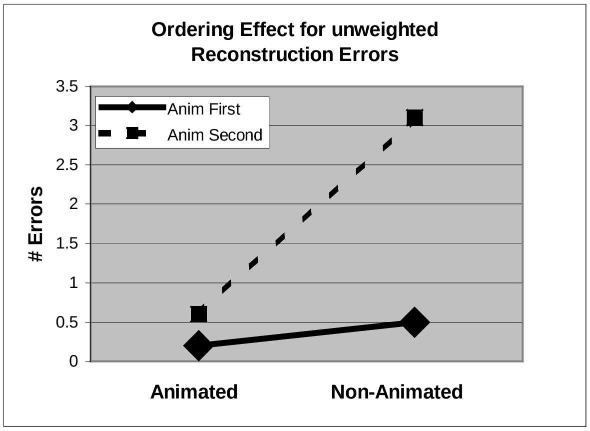

Figure 4: Ordering effect for unweighted reconstruction errors.

Finally, there were several significant secondary effects that are listed in Table 2. These five effects were not consistent in that they did not each show that one treatment performed better than another based on order.

After the entire experiment, subjects were asked to write down what they thought were the advantages and disadvantages of animation, and they were asked to give suggestions to improve the system. Each comment was categorized, and we report the totals of all of the comments in Table 3.

	Measure	**Effect**
Animation by Order	Navigation Time	$F_{1,19}$=20.924, p<.001
	Explore Errors	$F_{1,19}$=11.845, p=.003
	Explore Time	$F_{1,19}$=13.411, p=.002
	Reconstruction Errors	$F_{1,19}$=9.979, p=.006
	Satisfaction	$F_{1,19}$=8.450, p=.010

Table 2: Secondary significant effects.

Question	**Response**	**# of Responses**
Advantages of animation	Improved understanding of relationships	12
	Easier to use	2
	Less strict	1
	More fun	1
	Prettier	1
Disadvantages of animation	Slower	5
	Distracting	2
Suggestions	Add overall view	12

Table 3: Number of subjects per comment.

Analysis

This experiment shows that, for the spatial map of the family tree that we used, animation improved subjects' ability to learn the spatial position of family members within the tree without a speed penalty. Subjects did not perform navigation and recall tasks better with animation, although the animation did not change their response time either. Further, while subjects did not prefer the animated transitions to the non-animated transitions, 75% of them specifically stated that they thought the animations helped them learn the relationships between the data, and only 25% of them said that the animations slowed them down.

We find these data to support our hypothesis that animated transitions help users built mental maps of spatial information. While our dataset is small, the visual space that was navigated relative to the screen size was fairly large, and so while it would be useful to run another study with more data, we think this experiment's results can be generalized to other information systems where the space is larger than the display.

Subjects clearly performed better reconstructing the family tree in this experiment. It is interesting, however, that subjects did not perform better navigating the space, or recalling relationships about family members. Let us examine these three task types more closely.

Animation had no significant effect on the first task of navigating the family tree in order to learn the relationships between family members. This task involved clicking on hyperlinks to explore the tree. Since this task involved many animated transitions, each of which took one second, it is not surprising that performance time was not improved. The fact that the animated transitions did not slow down performance implies that either subjects were able to find what they were looking for while following fewer links, or by following each link quicker. Unfortunately, we did not record the number of links followed, and so we can not report on this. There was also no significant difference in correctness on this first task. This is also not surprising because nearly all of the questions were answered correctly in both cases.

The questions for the second task were answered from memory after the subjects had explored the family tree. This eliminated the time spent animating the viewpoint as a factor. It might be expected that if users really had built better mental maps of the information space, then performance would have been improved on this task. However, no performance improvement was seen for the animated condition.

Our explanation is that although animation directly aids in learning spatial relationships, it does not directly help learning more complex relationships among data. It could be that such a cross over takes more time. Or it could be that spatial memory and symbolic reasoning are independent enough that learning one does not necessarily improve the other – even if they are related, as they were in this study.

For the third task of reconstruction, animation did have a significant effect, reducing the errors that subjects made reconstructing the tree given pictures of the family members' faces. This effect was particularly strong because not only were the number of errors smaller, but the kinds of errors were less serious. All of the eight errors in the animated treatment involved swapping of two faces, while errors in the non-animated treatment involved switching of three or four faces. Errors in the non-animated treatment included 14 swaps, eight faces that were misplaced by sliding an entire row out of place, and 14 more errors that were the result of confusing three or four faces together. This effect appears to be strong because this task was heavily dependent on remembering the spatial position of family members.

This reconstruction task was the one that had an ordering effect (Figure 4). Subjects did significantly worse with the non-animated treatment if it was first. However, they performed only slightly worse with the non-animated treatment if it was preceded by the animated treatment. One explanation for this is that the subjects learned the structure better with the animated treatment, and because the family trees for each treatment had the identical structure, they could remember that structure from one treatment to the next. Since those subjects that had the non-animated treatment first did substantially better on the second animated treatment, this implies that the non-animated treatment did not help subjects learn the family tree structure as well as the animated treatment, because if it had, we would expect only a small improvement in the second animated treatment.

Finally, we were surprised that despite the positive comments about animation, subjects did not report a preference either way. This was especially puzzling because we have received positive informal feedback on previous occasions regarding animation [15, 17].

Our explanation here is twofold. The first issue is that subjects reported being quite frustrated doing this experiment. Most subjects wanted some way to see an overview of the entire family (see Table 3). The point of our experiment was to study how people navigated spaces where they could not see an overview. However, there were few enough people in the trees that subjects realized this was an unrealistic situation, and thus apparently felt that with or without animation, there were better alternatives for exploring family trees. Thus, subjects reported their dissatisfaction equally between treatments. A second issue is that we administered the satisfaction questionnaire separately for the animated and non-animated conditions, and we never specifically asked people to compare them. Since subjects did report positive effects of animation when asked specifically about animation (see Table 3), per aps if we had asked subjects to directly compare the two conditions, we would have seen different results.

So, why does animation appear to improve people's ability to learn spatial position and relationships of data? We do not know the answer to this question, but think our initial intuition is part of the answer. We think that animation helps users to maintain object constancy, and that without animation, users must spend time rebuilding an understanding of which object is which. When an object does not appear in all the views, then animation may be even more important because users can see which direction it is going, and so they can build a mental map of where things are.

CONCLUSION

While the current study does provide interesting results, it does not answer all of our questions. For instance, we would like to run an animation experiment on a larger data set, and we would like to know what the factors are in determining an optimal time to be spent for the animated transitions. We also would like to study the issue of animation time more closely, making sure that the animation is responsible for the performance improvement, and not just the increased time of the transition. Animation has a strong visual impact, and not all users like it. So, future studies should be careful to consider user satisfaction as well as performance. Finally, we also would like to understand animation well enough so that we can pursue our original question of how well multiscale presentations of data work.

Our positive results for animation lead us to some preliminary design guidelines. If a task requires subjects to know something about objects' spatial position, and the viewpoint is changed, then animating that change in viewpoint appears to help users. While our experiment was for 2D data, our experience leads us to think that our results will generalize to other spatial formats such as 1D and 3D data (although we do not have any data to support this belief). This has direct implications for many applications that present linear data – including word processors, spreadsheets, and Web browsers.

Based on this study, we feel that we can make certain recommendations. We suggest that when a movement is made within a document, that movement should be animated. This applies to using the page-up and page-down keys, clicking in the trough of a scrollbar, and jumping to another page. There is no reason to think that a full one-second animation is necessary in all circumstances, but some animation does seem to help, and the time spent animating does not seem to hurt. Indeed, some current applications already do support some animation in this fashion. Microsoft Internet Explorer 4.0, for instance, animates page-up and page-down actions. However, this animation is not controllable by the user, and is not consistent across other interactions (such as Back). Most applications today still do not perform such animation.

ACKNOWLEDGEMENTS

We especially appreciate the efforts of Jim Beisaw at the University of Maryland and Susanne Jul at the University of Michigan who helped with the early design of this study. We thank Maya Venkatraman who helped us analyze the data, and Britt McAlister who helped write the automated data recording program. In addition, we thank all of the members of the HCIL at UMD who gave advice and help throughout this work. Finally, we wouldn't be doing this work in the first place if it weren't for the other members of the Pad++ team, especially Jim Hollan and Jon Meyer.

This work, and Pad++ in general has been largely funded by DARPA to whom we are grateful.

REFERENCES

1. Baecker, R., & Small, I. (1990). Animation at the Interface. B. Laurel *The Art of Human-Computer Interface Design* (pp. 251-267). Addison-Wesley.
2. Baecker, R., Small, I., & Mander, R. (1991). Bringing Icons to Life. *In Proceedings of Human Factors in Computing Systems (CHI 91)* New York: ACM, pp. 1-6.
3. Barber, R. E., & Lucas, H. C. (983). System Response Time, Operator Productivity and Job Satisfaction. *Communications of the ACM, 26*(11), 972-986.
4. Bederson, B. B., & Hollan, J. D. (1994). Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics. *In Proceedings of User Interface and Software Technology (UIST 4)* New York: ACM, pp. 17-26.
5. Bederson, B. B., Hollan, J. D., Perlin, K., Meyer, J., Bacon, D., & Furnas, G. 1996). Pad++: A Zoomable Graphical Sketchpad for Exploring Alternate Interface Physics. *Journal of Visual Languages and Computing, 7,* 3-31.
6. Bederson, B. B., Hollan, J. D., Stewart, J., Rogers, D., Druin, A., Vick, D., Ring, L., Grose, E., & Forsythe, C. (1998). A Zooming Web Browser. C. Forsythe, J. Ratner, & E. Grose (eds.), *Human Factors and Web Development* (Chap. 19, pp. 255-266). New Jersey: Lawrence Erlbaum.
7. Bederson, B. B., & Meyer, J. (1 98). Implementing a Zooming User Interface: Experience Building Pad++. *Software: Practice and Experience,* (in press).
8. Brown, M. H. (1988). Perspectives on Algorithm Animation Visualization. *In Proceedings of Human Factors in Computing Systems (CHI 88)* ACM Press, pp. 33-38.
9. Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The Information Visualizer, an Information Workspace. *In Proceedings of Human Factors in Computing Systems (CHI 91)* ACM Press, pp. 181-188.
10. Card, S. K., Robertson, G. G., & York, W. (1996). The Webbook and the Web Forager: An Information Workspace for the World Wide Web. *In Proceedings of Human Factors in Computing Systems (CHI 96)* ACM Press, pp. 111-117.
11. Chang, B.-W., & Ungar, D. (1993). Animation: From Cartoons to the User Interface. *In Proceedings of User Interface and Software Technology (UIST 9)* ACM Press, pp. 45-55.
12. Dannenbring, G. L. (1984). System Response Time and User Performance. *IEEE Transactions on Systems, Man, and Cybernetics, 14*(3), 473-478.
13. Donskoy, M., & Kaptelinin, V. (997). Window Navigation With and Without Animation: A Comparison of Scroll Bars, Zoom, and Fisheye View. *In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 279-280.
14. Druin, A. (Ed.), (In Press). *The Design of Children’s Technology*. San Francisco, CA: Morgan Kaufmann.
15. Druin, A., Stewart, J., Proft, D., Bederson, B. B., & Hollan, J. D. (1997). KidPad: A Design Collaboration Between Children, Technologists, and Educators. *In Proceedings of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 463-470.
16. Gonzalez, C. (1996). Does Animation in User Interfaces Improve Decision Making? *In Proceedings of Human Factors in Computing Systems (CHI 96)* ACM Press, pp. 27-34.

17. Hightower, R. R., Ring, L., Helfman, J., Bederson, B. B., & Hollan, J. D. (1998). Graphical Multiscale Web Histories: A Study of PadPrints. *In Proceedings of ACM Conference on Hypertext (Hypertext 98)* ACM Press, pp. 58-65.
18. Maloney, J. H., & Smith, R. B. (1995). Directness and Liveness in the Morphic User Interface Construction Environment. *In Proceedings of User Interface and Software Technology (UIST 95)* ACM Press, pp. 21-28.
19. Robertson, G. G., Mackinlay, J. D., & Card, S. K. (1991). Cone Trees: Animated 3D Visualizations of Hierarchical Information. *In Proceedings of Human Factors in Computing Systems (CHI 91)* ACM Press, pp. 189-194.
20. Sears, A., Jacko, J. A., & Borella, M. S. (1997). Internet Delay Effects: How Users Perceive Quality, Organization, and Ease of Use of Information. *In Proceedings of Extended Abstracts of Human Factors in Computing Systems (CHI 97)* ACM Press, pp. 353-354.
21. Shneiderman, B. (1997). *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Massachusetts: Addison-Wesley.
22. Slaughter, L. A., Harper, B. D., & Norman, K. L. (1994). Assessing the Equivalence of Paper and On-Line Versions of the QUIS 5.5. *In Proceedings of 2nd Annual Mid-Atlantic Human Factors Conference* pp. 87-91.
23. Stasko, J. T., Badre, A., & Lewis, C. (1993). Do Algorithm Animations Assist Learning? An Empirical Study and Analysis. *In Proceedings of Human Factors in Computing Systems (InterCHI 3)* ACM Press, pp. 61-66.
24. Teal, S. L., & Rudnicky, A. I. 1992). A Performance Model of System Delay and User Strategy Selection. *In Proceedings of Human Factors in Computing Systems (CHI 92)* ACM Press, pp. 295-305.