The Alphaslider: A Compact and Rapid Selector

Christopher Ahlberg and Ben Shneiderman*
Department of Computer Science,
Human-Computer Interaction Laboratory & Institute for Systems Research
University of Maryland, College Park, MD 20742
E-mail: ahlberg@cs.chalmers.se, ben@cs.umd.edu
Tel: +1-301-405-2680

ABSTRACT
Research has suggested that rapid, serial, visual presentation of text (RSVP) may be an effective way to scan and search through lists of text strings in search of words, names, etc. The Alphaslider widget employs RSVP as a method for rapidly scanning and searching lists or menus in a graphical user interface environment. The Alphaslider only uses an area less than 7 cm x 2.5 cm. The tiny size of the Alphaslider allows it to be placed on a credit card, on a control panel for a VCR, or as a widget in a direct manipulation based database interface. An experiment was conducted with four Alphaslider designs which showed that novice Alphaslider users could locate one item in a list of 10,000 film titles in 24 seconds on average, an expert user in about 13 seconds.

KEYWORDS: Alphaslider, widget, selection technology, menus, dynamic queries

INTRODUCTION

Selecting items from lists is a common task in today's society. New and exciting applications for selection technology are credit card sized phone directories; personal digital assistants such as the Apple Newton with complete telephone, address, and business registers; handheld computers for maintenance workers with selections of prepared reports, objects, maps, and drawings; selection mechanisms for Laser Disc players where frame numbers between 1 and 54,000 need to be selected rapidly and electronic calendars where hours, days, months and years must be selected rapidly and accurately.

Obviously there is a need for methods for selecting items quickly and accurately, without a keyboard and in a small space. Traditional computers with large screens have used methods such as scrolling lists, menus, and keyboard entry to select items. For new emerging handheld technologies space is limited which makes scrolling lists and menus hard to implement effectively.

* Current address: Dept. of Computer Science, Chalmers University of Technology, S-412 96 Göteborg, Sweden

Much of the research done on selection mechanisms has focused on menus [5,16]. To make menu selections effective various techniques have been explored, such as menus with different ratios of breath and width, and menus where items are sorted by how frequently they are selected [16,20]. The RIDE interface explored in [19] allows users to incrementally construct strings from legal alternatives presented on the screen and thereby elminate user errors.

Scrolling lists [2,17] share many of the attributes of menus and are often used for selecting items from lists [Figure 1]. Research has shown that items in scrolling lists should be presented in a vertical format [3], items should be sorted [10], and that 7 lines of information is more than adequate for retrieving alphanumeric information [8]. Introducing an index to the scrolling list can shorten the search time [4].

Figure 1: Motif scrolling list.

The Alphaslider

The Alphaslider [Figure 2] was first proposed by [18]. It is used to rapidly scan through and select from lists of alphanumeric data. The essential components of an Alphaslider are a slide area, a slider thumb, a text output and an index to the elements that the slider operates over. Whereas a traditional slide area lets users page through the content of a scrolling list, the Alphaslider slide area lets users move directly to a certain part of the slider range by clicking in it. The index below the slide area guides that operation. The index, as shown in [18] and earlier proposed in [4], is proportionally spaced to the number of items that start with each character. The most infrequent starting characters of the items in the searched list does not show up in the index. The value of the Alphaslider is reflected in a single line text item, which should update immediately upon user movement of the slider thumb.

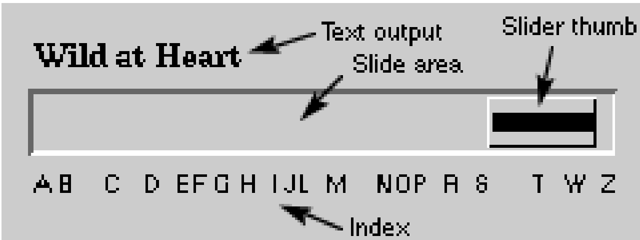

Figure 2: Alphaslider for selecting movie titles .

The Alphaslider can be used in direct manipulation database querying systems, such as Dynamic Queries [1]. Applications of Dynamic Queries have so far been limited to domains where the attributes of the database are numerical, such as real estate databases [22] and the chemical table of elements [1], but the Alphaslider makes it possible to query alphanumeric attributes, such as names, titles, and objects.

Although selecting words or names with an Alphaslider might in some cases be slower than typing on a keyboard, the use of an Alphaslider has several advantages compared to a keyboard. Using a keyboard, inexperienced users must search the keyboard for the appropriate key and the keyboard does not prevent misspellings. Users may type a value for a field that is inappropriate such as a number when a person's name is required [21]. An Alphaslider by definition contains all valid input choices and can continuously have its query range updated, which effectively eliminates queries that will result in an invalid or empty query result.

Design issues

Some major design constraints for the Alphaslider are the small size, one line of text output, and the mapping of a large number of items to a small number of pixels, i.e. each movement of the slider thumb corresponds to a large number of items. Alphasliders, just as many other controls, should be operatable without looking at them continuously. This is important if the Alphaslider is used in a direct manipulation interface such as a public information system or a control panel for a medical image retrieval system, where users want to concentrate on the output rather than the input – because they are visually separated.

The issue with the richest set of design possibilities is how the slider thumb should be controlled. The Alphaslider described in [18] was implemented with up to 320 entries, which mapped one item to each pixel. In some applications this is sufficient, but as has been argued above, in many emerging technologies there is a need for a much larger range. This causes a problem when there are more items than pixels. Traditional scroll bars solve this by allowing users to click on the arrow buttons to change the view without scrolling the slider thumb which is a good solution in some cases.

Another solution is to separate the user's movement of the mouse (trackball, finger on a touchscreen) from the display of the slider thumb - so that when the mouse is moved the position in the list is changed, but not necessarily the position of the slider thumb. This technique makes it possible to map tens of thousands of items, if not hundreds of thousands, to a slider. The items are easily selectable and with proper feedback the task can be accomplished rapidly. This class of techniques has the advantage of users being able to operate the control without looking at it.

The small size of the Alphaslider calls for a compact text display, i.e. one line of text output. Displaying text in a one–line display can be done in either of the following ways: (i) by rapidly displaying text at a fixed location, referred to as RSVP, rapid, serial, visual presentation. RSVP has been used by psychologists to study reading behavior and it has been shown that people can read text presented in RSVP format at approximately the same speed as they can read text presented in page format [11,15], (ii) by scrolling text horizontally from the right to the left, referred to as Times Square. Reading comprehension using Times Square with a smooth scrolling can be at least as high as for RSVP, and with a higher user preference [12], and (iii) by scrolling text vertically – a technique that is rarely used for one–line displays [12].

For situations where the viewing window is narrow and presentation rate is high, we conjecture that RSVP may be a more suitable display method and also an efficient way to search through lists [12]. Accordingly, the Alphasliders described in this paper all use RSVP as the display method.

EXPERIMENT

Introduction

An experiment was conducted to compare different designs making it possible to map 10,000 items to a small number of pixels in an Alphaslider.

Apparatus

The interfaces used in the experiment were built using the Galaxy user interface development environment with the Motif look and feel. A Sun Microsystems SparcStation with a 17–inch color monitor and optical three button mouse was used. The resolution of the screen was 1180 x 876 pixels. A 14 point Times Roman Medium font was used to display the text. The experimental setup used 10.5 cm x 3.5 cm of the screen, while the Alphaslider used 7.5 cm x 2.5 cm within the larger area.

Interfaces

Four different designs of the Alphaslider were included in the experiment [Figures 3 to 6]. Their look and feels were similar in several aspects. The text output was one line RSVP in all cases and was displayed over the slider. Under each slide area, an index provided cues about the distribution of the elements alphabetically. A timing mechanism for the experiment included two buttons for each interface. The target title was display directly above the Alphaslider value to minimize vertical eye movement. The non-scrollbar Alphasliders would move the slider thumb directly to where the mouse was clicked in the slide area. All the interfaces were based on the Motif look and feel [17].

Position interface. The first interface [Figure 3] allowed subjects to select the granularity of their mouse movements by initiating dragging in different parts of the slider thumb. The top part of the thumb corresponded to the coarse granularity of 100 elements per mouse movement, the middle part to the medium granularity of 20 elements per mouse movement, and the lower part to the fine granularity of one element per mouse movement. While dragging, the active part was turned black.

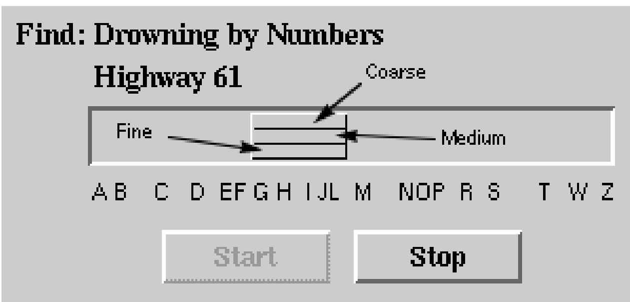

Figure 3: Position interface. Users select granularity by clicking in different parts of the slider thumb.

Scrollbar interface. The second interface [Figure 4] was based on the standard Motif scroll bar [17, page 4–5]. To select and move by the coarse granularity, subjects would drag the slider thumb. To move by the medium granularity, subjects clicked or held down the mouse button on the slide area, on either side of the thumb, and finally to move by the fine granularity subjects would click on the arrow buttons at the ends of the slide area. With this interface subjects were not able to move directly to a particular part.

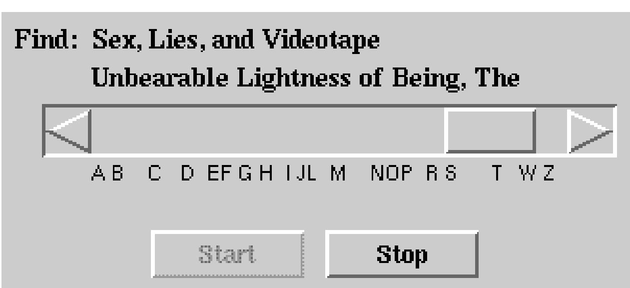

Figure 4: Scrollbar interface. Users select granularity in a fashion similar to traditional scrollbars.

Acceleration interface. The third interface [Figure 5] let subjects select granularity by moving the mouse at different speeds. If subjects moved the mouse more than a certain trigger level of pixels in one mouse event, the granularity would be changed to the medium granularity, and if the speed reached a second trigger level, the granularity would be changed to coarse.

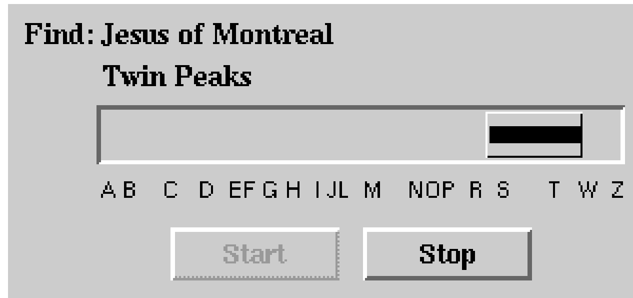

Figure 5: Acceleration interface. Granularity is proportional to the velocity of the mouse movements.

Micrometer interface. The fourth and last interface [Figure 6] allowed subjects to change the granularity of their movements by moving the mouse vertically - moving up or down switched to coarse and fine granularity respectively. Upon release of the mouse button the granularity switched back to medium. A simple stabilization algorithm allowed users to move the mouse vertically without effecting the setting of the Alphaslider.

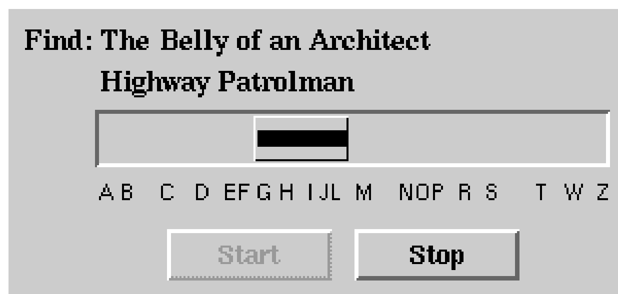

Figure 6: Micrometer interface. Users select granularity by moving the mouse vertically.

Hypotheses

A very basic model for comparison of the time to locate an item with different Alphasliders, T_{locate}, estimates it to be the time spent dragging and moving the slider thumb to the correct position. Dragging can be estimated with Fitt’s Law [9,14], but dragging done with the Alphaslider differs substantially from tasks described in those papers. A simple estimate of T_{locate} for comparison purposes is the time users spend moving the thumb to approximately the right spot, $T_{rough-aim}$, plus the time spent adjusting the thumb to find the correct item, T_{adjust}.

Based on these assumptions, the following hypotheses were stated for *expert* mouse users:

- The Acceleration and Micrometer interfaces would perform best as the change of granularity could be done without releasing the mouse button, which would make transition from the coarse and medium granularity to fine granularity short.
- The Position interface where the transition to fine granularity only asked for a very small cursor movement would follow in performance.
- The Scrollbar interface would perform worst for expert users because of the requirement to move the cursor between the ends of the Alphaslider. Also this interface did not allow users to move directly to a particular part of the slider – though this was not expected to account for a large part of differences in time between the Alphasliders.

Subjects were required to have previous mouse experience, and having worked with mouse most probably implies that it was done in a graphical user interface environment. Consequently, subjects were expected to have used scrollbars before, which could lead to better performance for the Scrollbar interface.

For subjective evaluations it was expected that the Scrollbar interface would be preferred due to its similarity to many commercially available scroll bars – especially the Windows 3.0 scroll bar which many subjects were assumed to have used previously.

Experiment variables

The independent variable was the type of interface:
(i) Position interface, (ii) Acceleration interface,
(iii) Micrometer interface, and (iv) Scrollbar interface.

The dependent variables were:
(i) time to locate an item in the list
(ii) subjective satisfaction.

Tasks

For each interface 25 tasks were generated by presenting random items from a list of 10,000 film titles averaging 19 characters in length. The tasks were generated at run–time when subjects pushed the start button. For each interface subjects were presented with 5 practice tasks. The slider thumb was returned to the middle of the slider before each task.

Participants

Twenty–four subjects participated and were paid $10 each. Experience in using a computer mouse was required. Subjects were recruited from the University of Maryland campus and were mainly non-computer science undergraduate students in the range of 18 to 35 years old. Nine females and fifteen males participated.

Procedures

A counterbalanced within–subjects experimental design was used. A pilot study with four subjects was conducted. Each session lasted 1.5 hours. Subjects read a general instruction sheet, were presented with interface–specific instructions for each interface and were then given five practice tasks to complete. While reading instructions and completing practice tasks subjects were free to ask questions. During the timed tasks for each interface, subjects were not allowed to ask questions and were asked to work as quickly as possible. The experimenter sat next to the subject and observed the interaction. When finished, subjects filled out a shortened QUIS–form [6]. After using all interfaces, subjects filled out a forced-choice preference rating for each possible pairing of interfaces.

RESULTS

Analysis of the timed tasks was done using an ANOVA with repeated measures for interface type. Observing the mean time for each subject to complete 25 tasks for each interface shows a significant main effect, $F(3,69) = 17.2$, ($p<0.001$)

Tukey's post-hoc HSD analysis was used to determine which interface was significantly faster. The Position and Scrollbar interfaces were found to be significantly faster than the Micrometer and Acceleration interfaces ($p<0.001$). Subjects used approximately 24 seconds to complete all tasks for the Position interface and 25 seconds for the Scrollbar interface. For the Micrometer and Acceleration interfaces subjects used approximately 32 seconds. An expert Alphaslider user - the first author - used approximately 13, 16, 14 and 19 seconds respectively for the Position, Micrometer, Acceleration and Scrollbar interfaces.

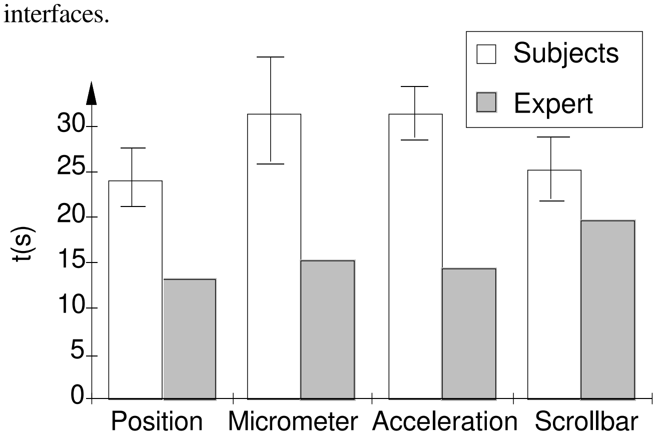

Figure 7: Graph showing mean time to complete all tasks for each interface. Standard deviation indicators on top of bars.

The pairwise forced-choice preference ratings were converted to ranks and the Freidman test was used to determine the extent to which subjects ranked the interfaces in the same order. The results indicate that subjects consistently rated the Scrollbar interface highest, the Position interface second highest, and the Micrometer and Acceleration interfaces worst ($\chi^2 = 30.6$, $p < 0.001$). The mean preference rankings were 1.3 (0.7), 2.45 (1.0), 3.1 (0.7), and 3.1 (0.9) for the Scrollbar, Position, Micrometer, and Acceleration interfaces respectively (standard deviations in parentheses).

DISCUSSION

Timed tasks

Subjects completed their tasks on the average one second faster for the Position interface compared to the Scrollbar interface, although the difference was not statistically significant. The success of both interfaces was probably due to the fact that they both were found to be stable and predictable by the subjects. Observing subjects revealed different behavior for the two interfaces. The Position interface was appreciated by some subjects for the possibility to fine-tune without releasing the mouse button, while the scrollbar interface was appreciated by others for the arrow buttons which made it possible to fine-tune the setting by repeated mouse clicks instead of dragging.

The hypothesis predicted the Acceleration and Micrometer interfaces to perform better than the Position and Scrollbar interfaces, but this was not the case. An explanation for the Micrometer interface's bad performance may be found in that subjects found it somewhat complicated. The time for changing granularity, i.e. the time for moving the mouse vertically (Micrometer), was expected to be less than the time to release the mouse button, locate a new target, and hold down the mouse button again (Position & Scrollbar). But the functionality of moving the mouse vertically probably interfered with subjects' notion of mouse movements and slowed them down. A similar explanation can be found for the Acceleration interface. By unintentionally triggering the acceleration mechanism, subjects overshot their targets and were discouraged by fast mouse movements.

The expert Alphaslider user performed nearly twice as fast as the experimental subjects. Observing the expert user's mean times revealed a different ordering of the interfaces' performance. The order followed the predictions of the hypotheses, except for the Position interface which was the fastest for the expert user as well.

Comparisons to other selection mechanisms

Landauer & Nachbar let subjects select words and numbers from menus with 4,096 items, using the whole screen [13]. When subjects selected words of length 4-14 characters, average selection times varied from 12.5 to 23.4 seconds for different menu structures. Doughty & Kelso had subjects select numbers from 1 to 4,096 and selection times varied from 9 to 17 seconds for different menu structures [7]. Alphaslider subjects had to select from film titles, probably a more difficult task, from a list which was 2.5 times as big, only using a fraction of the screen size, and their selection times varied from 24 to 32 seconds - a performance that compares favorably.

Interface characteristics

General observations. Several subjects were frustrated by doing fine tuning work with the mouse while holding down the mouse button. Holding down the mouse button while moving the mouse is a fairly complicated motor action, and subjects were found to repeatedly release the mouse button by mistake, which has been observed in other studies too [14]. Releasing the mouse button while dragging caused the cursor to leave the slider thumb and forced subjects to initiate dragging again. Subjects' ability to do the necessary fine tuning was also affected by holding the mouse button down. Subjects were observed pressing the button too hard and thereby generating friction between the mouse and the mouse pad. For the Scrollbar interface this behavior was not observed, as subjects clicked the arrow buttons to fine tune the value of the Alphaslider. It is reasonable to conjecture that a good design of an Alphaslider should include arrow buttons for fine adjustments.

Feedback about subjects change of granularity was provided for the Position, Micrometer, and Acceleration interfaces through a speed indicator in the slider thumb. Although the thumb was very close to the displayed film title, it is obvious from the results of the subjective ratings of the interfaces that this feedback is not enough. Feedback is an important design issue for the Alphaslider and will be discussed further below.

For the Position, Micrometer and Acceleration interfaces subjects were observed to mainly use the middle and fine granularity and for the Scrollbar interface mainly the thumb and the arrow buttons. The functionality of moving directly to a certain part of the slider was used extensively by subjects.

Position interface. The Position interface allowed subjects to select one of three parts of the thumb to set granularity, which was greatly appreciated. Subjects stated "With this interface I can exactly determine by what speed I'm going to move". It also caused some problems because the selection areas were small. As subjects were found to nearly always use the middle and fine granularities, this could be addressed by just allowing subjects to select from two granularities on the thumb - with accordingly large areas to select.

Scrollbar interface. Subjects found it easy to do fine tuning with the Scrollbar interface, they just had to click the arrow button and the elements would flash by rapidly. Some subjects experienced problems having to move the mouse between the end points of the slide area to change directions – this was particularly the case for expert mouse users who were more comfortable with the position interface where they could change directions by just moving the mouse.

Acceleration interface. The acceleration interface was expected to do well in performance, but the reverse occurred; it both performed badly and was rated low. Subjects overshot the goal by mistake, by moving too fast and thereby triggering the acceleration. Feedback was provided in the slider thumb but, as subjects concentrated on the text value of the slider, this feedback was overlooked in many cases. Most subjects found changing granularity with the Acceleration interface too abstract. Some subjects did adjust very well to the acceleration interface and found it easy because they did not have to do anything else than move the mouse horizontally to set the value.

Micrometer interface. Whereas it was expected that the Micrometer interface would perform well, some subjects found it surprisingly difficult to operate. An experienced user operating the Alphaslider can concentrate on the output without looking at the Alphaslider itself. Subjects were confused by the different semantics of moving the mouse vertically and horizontally.

When releasing the mouse button the Alphaslider returned to the middle granularity, to avoid modes that the Alphaslider could be left in. While this was not detected as a design flaw in the design process and in the pilot experiment, during the experiment it became obvious that this design caused frustration for subjects – especially those who frequently released the mouse button by mistake.

Subjective evaluation

From both the forced choice ratings and the QUIS analysis it is obvious that subjects preferred the Scrollbar interface. One explanation for this is that the slide area and thumb part of the Alphaslider was similar to other scrollbars subjects had previously used. The particular feature of the Scrollbar interface that subjects liked was the arrow buttons. One subject stated about the Scrollbar interface: “This is the interface type I am most familiar with, and thus I was able to apply many of my personal strategies to it. It was neither as fast nor as intuitive as #3 (Position interface), however”.

Subjects’ reactions to the Acceleration interface were interesting. One subject stated: “Why accelerate at all, as you can just click and go to a particular place directly?” - he avoided the acceleration by clicking on the bar and then moving the slider thumb slowly. Reflecting the opposite opinion, one subject stated: “It’s much easier than the other interfaces, you just need to move the mouse [as opposed to other more complicated schemes]”. Subjects appreciated the stability of the Position interface; “With this interface I can exactly determine by what speed I’m going to move”.

GUIDELINES FOR DESIGNERS

In the light of the experiment described above, a redesigned Alphaslider can be proposed. It was obvious from the experiment results that the arrow buttons of the Scrollbar interface were helpful to users.

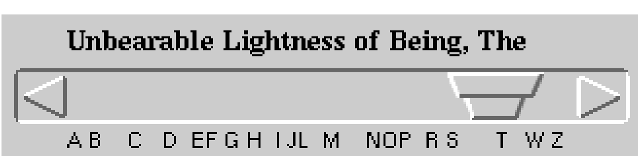

Figure 8: Redesigned Alphaslider

The Position interface performed well as it allowed subjects to move directly to a particular part of the Alphaslider, the value could be set by just moving the mouse, and it still allowed coarse movement with the thumb.

The Alphaslider in [Figure 8] would allow subjects to select either coarse or fine movement by selecting different parts of the thumb. Fine tuning can also be done by clicking on the arrow buttons.

FURTHER RESEARCH

- The use of the Alphaslider together with other input devices, such as touch screens, pens, trackballs, and joysticks, should be studied. The results in this paper should generalize to trackballs and joysticks, but for touchscreens and penbased systems several interesting design alternatives emerge.
- Although [12] suggest the use of RSVP for searching lists, they also show that the Times Square method of displaying text is highly effective. This should be explored as a design option.
- Providing feedback is important when browsing large information spaces. Sound could indicate granularity and granularity changes. Possible visual cues include indicators in the index below the slide area, display of the text field in different colors, a speed bar displayed just under the text field, zooming in the index, etc.
- The Alphaslider in this paper has one line of text output. The use of two or more lines of text output is certainly possible and should be explored.

CONCLUSIONS

The Alphaslider is a widget that makes it possible to rapidly select items from long lists without a keyboard using minimal screen space. Four different designs of an Alphaslider were evaluated in a controlled experiment.

Lessons learned from the study tell implementors and designers that Alphasliders are ready to be included in interactive systems and user interface management systems. With good use of feedback techniques, the Alphaslider is a powerful, compact, and rapid way of selecting items from lists. The University of Maryland is seeking to patent the Alphaslider.

ACKNOWLEDGMENTS

We appreciate support from Chalmerska Forskningsfonden, Adlerbertska Forskningsfonden, and Kungliga Hvitfeldska Stipendieinrättningen which made this research possible.
This research was done in the “Widget Carvers of College Park” group - Rich Chimera, Catherine Plaisant, Ninad Jog, Harsha Kumar, and Marko Teittinen, who all contributed at lively design meetings, late night hacking sessions, and early morning email reports.

REFERENCES

1. Ahlberg, C., Williamson, C., Shneiderman, B., Dynamic Queries for Information Exploration: An Implementation and Evaluation, *Proceedings ACM CHI'92 Conference*, 619-626, 1992

2. Apple Computer Inc., *Human Interface Guidelines: The Apple Desktop Interface*, Reading, MA: Addison–Wesley, 1987.

3. Backs, R. W., Walrath, L. C., and Hancock, G. A., Comparison of horizontal and vertical menu formats, *Proceedings of the Human Factors Society Conference 31st Annual Meeting*, 715-717, 1987

4. Beck, D., Elkerton, J., Development and evaluation of direct manipulation lists, *ACM SIGCHI Bulletin*, 20 (3), 72-78, 1989

5. Card, S., User perceptual mechanisms in the search of computer command menus, *Proceedings ACM CHI'82 Conference*, 190-196

6. Chin, J., Diehl, V., Norman, K., Development of an instrument measuring user satisfaction of the human-computer interface, *Proceedings ACM CHI'92 Conference*, 213-218, 1988

7. Doughty, R., Kelso, J., An evaluation of menu width and depth on user performance, unpublished student paper, described in Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction, 1st ed.*, Addison-Wesley, 1986

8. Elkerton, J., Williges, R., Information retrieval strategies in a file-search environment, *Human-Factors*, 26, 171-184, 1984

9. Gillan, D., Holden, K., Adam, S., Rudisill, M., Magee, L., How does Fitts' law fit pointing and dragging?, *Proceedings ACM CHI'90 Conference*, 227-234, 1990

10. Harpster, J., Random versus Ordered Menus in Self-Terminating Menu Searches, *Proceedings of the Human Factors Society 31st Annual Meeting*, 718-721, 1987

11. Juola, J., Ward, N., McNamara, T., Visual search and reading of rapid, serial presentation of letter strings, words, and text, J*ournal of Experimental Psychology*: General, 111, 208-227, 1982

12. Kang, J., Muter, P., Reading dynamically displayed text, *Behaviour and Information Technology,* 8(1), 33-42, 1989

13. Landauer, T., Nachbar, D., Selection from alphabetic and numeric menu trees using a touch screen: breadth, depth, and width, *Proceedings ACM CHI'85 Conference*, 73-78, 1985

14. MacKenzie, S., Abigail, S., Buxton, W., A comparison of input devices in elemental pointing and dragging tasks, *Proceedings ACM CHI'92 Conference*, 161-166, 1992

15. Masson, M., Conceptual processing of text during skimming and rapid sequential reading, *Memory & Cognition*, 11, 262-274, 1983

16. Norman, K, *The Psychology of Menu Selection: Designing Cognitive Control at the Human/Computer Interface*, Ablex, 1991

17. Open Software Foundation, *OSF/Motif Style Guide*, Englewood Cliffs, NJ: Prentice Hall, 1990

18. Osada, M., Liao, H., Shneiderman, B., AlphaSlider: searching textual lists with sliders, *Proceedings of the Ninth Annual Japanese Conference on Human Interface*, October 1993

19. Sears, A., Kochavy, Y., Shneiderman, B., Touchscreen field specification for public access database queries: let your fingers do the walking, *Proceedings of ACM CSC'90 Conference*, 1990

20. Shneiderman, B., *Designing the User Interface: Strategies for Effective Human-Computer Interaction, 2nd ed.*, Addison-Wesley, 1992

21. Welty, C., Correcting user errors in SQL, *International Journal of Man-Machine Studie*, 22(4), 463-477, 1985

22. Williamson, C., Shneiderman, B., The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system, *Proceedings of ACM SIGIR Conference*, 339-346, 1992