CAR-TR-673
CS-TR-3078
ISR-93-52

April 1993

AlphaSlider: Searching Textual Lists with Sliders

Masakazu Osada
Holmes Liao
Ben Shneiderman

Department of Computer Science,
Human-Computer Interaction Laboratory,
and Institute for Systems Research
University of Maryland
College Park, MD 20742

Abstract: AlphaSlider is a query interface that uses a direct manipulation slider to select words, phrases, or names from an existing list. This paper introduces a prototype of AlphaSlider, describes the design issues, reports on an experimental evaluation, and offers directions for further research. The experiment tested 24 subjects selecting items from lists of 40, 80, 160, and 320 entries. Mean selection times only doubled with the 8-fold increase in list length. Users quickly accommodated to this selection method.

Keywords: keyboard, data entry, touchscreen, direct manipulation, sliders

Contact: Ben Shneiderman, ben@cs.umd.edu

1. Introduction

As portable, palm-top, and pocket computers become more popular, the pressure to create and easy-to-use interfaces with non-keyboard input devices has grown. For example, a pocket-sized electronic phone book would be more effective if users did not have to type names, but could select them easily. Similarly, medical image workstations in hospital settings need direct and instant responses for physicians to make prompt and accurate diagnoses. Physicians should not have to take their eyes from the images to operate workstations, and many of them do not want to operate keyboards to retrieve patient data.

Voice recognition and handwriting recognition are often suggested as good possibilities for non-keyboard input. However, both of these options are limited in their practical applications at the present time because recognition and response rates are not fast enough, and end-users also have to adjust their speech or handwriting patterns to make recognition reliable. In contrast, direct manipulation interfaces with graphic objects have been introduced to many places such as bank teller machines, information kiosks of museums and libraries, etc.

We believe that AlphaSlider is practical for small portable devices the size of a credit card, and also as a component for larger and more complex applications. Many refinements and extensions seem possible, but inclusion of similar widgets in graphic user interface management software and toolkits seems appropriate.

2. Previous Research

Dynamic queries (Ahlberg, Williamson & Shneiderman, 1992; Williamson & Shneiderman, 1992) is a database searching technique using direct manipulation. It provides direct manipulation widgets such as sliders, and gives immediate feedback which inform users how close they are to finding the data they are searching for. Users can adjust the sliders, thereby performing dozens of queries, and see results immediately with no syntax errors. The data used in this application was numerical information, but dynamic queriea are also applicable to text data entry and retrieval.

A touchscreen keyboard, which is one of the solutions for alphanumeric data entry without physical keyboards, has been explored by Weisner (1988), and Sears Revis, Swatski, Crittenden & Shneiderman (1993). The former study showed that a QWERTY touchscreen keyboard was acceptable for entering limited quantities of alphanumeric data. The latter study also showed that it was usable, that the keyboard on a screen could be reduced in size without increasing errors, and that the performance was 25 words per minute(wpm) by computer science students, which was better than mouse-activated keyboard (17wpm), and less than the standard keyboard (58wpm). Plaisant & Sears (1992) demonstrated that repeated use of a touchscreen keyboard over a two hour period did not lead to fatigue and that users could cope with complex typing tasks involving capitals, numerics, and special characters.

3. 800-Number Yellow Pages: An example of AlphaSlider

We chose a potential consumer electronics application that would involve a small credit card sized pocket computer with travel information such as toll-free telephone numbers for car rental, airlines, hotels, etc. A screen from the prototype of AlphaSlider, to retrieve 800-Number Yellow Page entries is shown in Figure 1. The horizontal bar is 10 pixels high and 320 pixels long. Just underneath this bar the letters 'A' through 'Z' are displayed with spacing between each letter proportional to the number of entries in the database. The slider button is 15 pixels high and 5 pixels wide. Two left and right triangular fine tune buttons are attached to the main slider button. Users can drag the slider button to go from one letter to another. They can also point to any place in the bar so that the slider button will move automatically to the selected location. The two fine tune buttons reappear beneath the slider button when the slider button is released.

Four versions of the 800-Number Yellow pages were created on an Apple Macintosh II fx using Aldus SuperCard, a prototyping package. A mouse was used to manipulate the interface. Response time for clicking and dragging was 17msec, giving the users the perception of smooth movement. All of the four slider bars and slide buttons looked the same, and were manipulated in the same manner. The only difference was the width of each list entry in pixels on the horizontal bar (1, 2, 4, and 8 pixels). Since the total width was fixed at 320 pixels (approximately 12 cm), this corresponds to 320, 160, 80, and 40 entries.

The question field (seen as "Find: xxxxx) and Pause/Resume button were put on the top for experimental purposes. Subjects were required to find entries appearing in the question field, then push the 'Dial' button. If the selected entry matched the question, the system emitted a bell-like sound and the next question appeared. Otherwise, the system emitted a short beep and the question field remained unchanged. Users pushed the 'Pause' button when they wanted to stop working, and then pushed 'Resume' button, to continue.

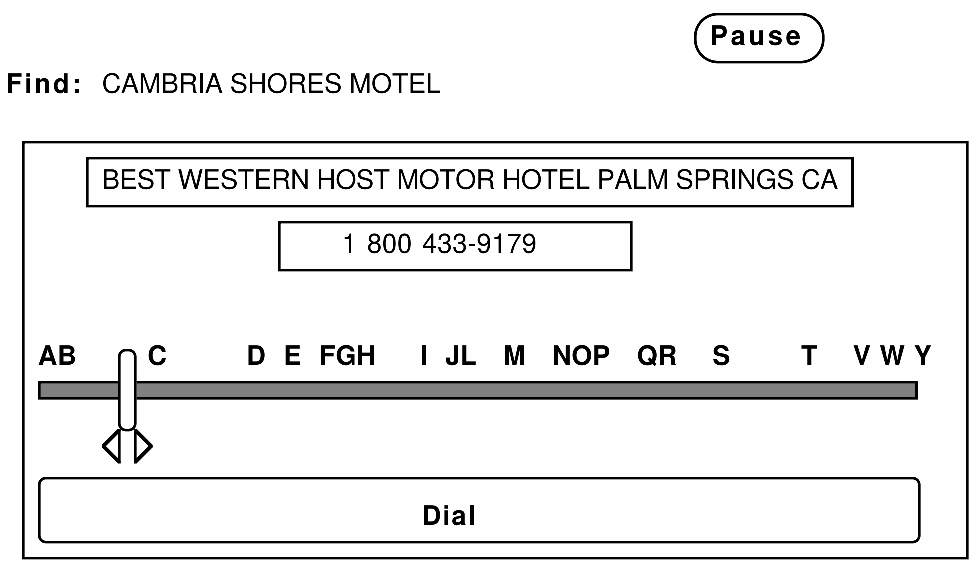

Figure 1. 800-Number Yellow Pages

4. Theoretical Estimates of End-User Performance

Figure 2 shows how AlphaSlider would be used. Users start searching by looking at a question field to find the target name. Then they would click on the slider bar, aiming for the letter of the target name. If the result box did not contain the target name, they would either click fine-tune buttons or drag the slider button to scroll through the list. When they found the targeted name, they would click the Dial button and complete the question.

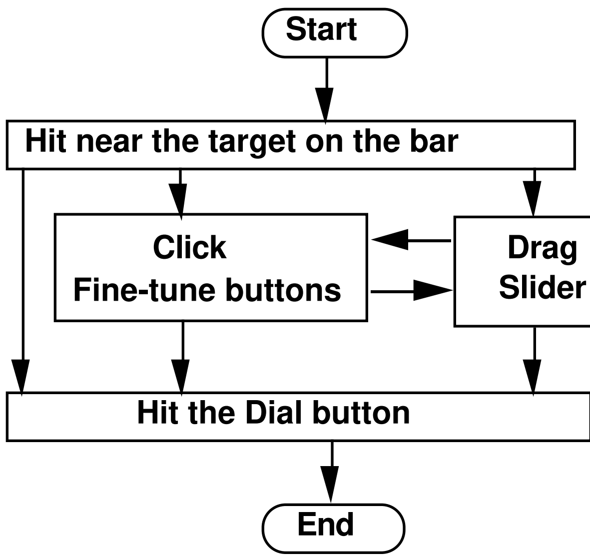

Figure 2. Transition diagram of 800-Number Yellow page usage

Estimated time for each action can be calculated using the Model Human Information Processor (Card, Moran, and Newell, 1983). For example, the action of clicking to a position nearest the targeted word consists of :

- eye movement to look at the question τ_{eye} =230msec
- storing the question in visual working memory τ_p = 100msec
- recognition of the alphabetical order $\tau_{alphabet}$
- eye movement to search mouse cursor τ_{search}
- eye movement to look at the bar τ_{eye}=230msec
- locate the target $\tau_p+\tau_c$ = 170msec
- move and click the mouse from Dial button to the bar

τ_m+T_{pos1} = 70msec+602msec =672msec

- see the result $\tau_{eye}+\tau_p+\tau_c$=400msec

T_{pos1} is a time estimation applying Fitts's law (Fitts, 1954; Card, Moran, and Newell, 1983), where I_M=100msec/bit, and D/S=33. D represents a distance between the starting position (on the Dial button) and a target position (on the bar). S represents a width of a target. Since the questions appear in a random manner, the average distance between the starting position and the target position is calculated as $D_{max}+D_{min}/2$. The overall estimated time for clicking on the slider bar was computed as

$$\tau_{eye}+\tau_p+\tau_{alphabet}+\tau_{search}+\tau_{eye}+\tau_p+\tau_c+\tau_m+T_{pos1}+\tau_{eye}+\tau_p+\tau_c$$

$$= 1802 + \tau_{alphabet} + \tau_{search} \text{ msec.}$$

In a similar fashion, the estimated time for fine-tuning the slide button movement on the slider bar was calculated as

$$(\tau_{eye}+\tau_p+\tau_c+\tau_m+T_{pos2}+\tau_{eye}+\tau_p+\tau_c)+(\tau_m+\tau_{alphabet}+\tau_c)*(m-1)$$

$$=1070+(170+\tau_{alphabet})*(m-1) \text{ msec.}$$

where m represents the number of clicking on the fine tune buttons. The estimated time for dragging the button across the slider bar is

$$(\tau_{eye}+\tau_p+\tau_c+\tau_m+T_{pos3}+\tau_{eye}+\tau_p+\tau_c)+(\tau_m+T_{pos3}+\tau_{alphabet}+\tau_c)*(n-1)$$

$$=970+(270+\tau_{alphabet})*(n-1) \text{ msec.}$$

where n represents the number of dragging movement made. The estimated time for clicking on the "Dial" button is

$$\tau_{eye}+\tau_p+\tau_c+\tau_m+T_{pos4} = 570 \text{ msec.}$$

Overall the estimated time to complete one question depends upon how many times users click on the fine-tune, or drag on the bar. The shortest time to complete one question is,

$$\tau_{search}+\tau_{alphabet}+1802+570 \text{ msec.}$$

if users click the exact target on the bar so that there is no clicking or dragging. If users click on the bar a point α pixels far from the target, they have to click α/k times on the fine-tune buttons or to drag α/k times, where k is the number of pixels between each data entry on the bar. The estimated time to complete one question using the dragging strategy is

$$t = \tau_{search}+\tau_{alphabet}+3102 + (270+\tau_{alphabet})\alpha/k \qquad \text{msec} \quad (\alpha/k > 1)$$
$$t = \tau_{search}+\tau_{alphabet}+2372 \qquad \text{msec} \quad (\alpha/k = 1)$$

The estimated time to complete one question using the clicking strategy is

$$t = \tau_{search}+\tau_{alphabet}+3442 + (170+\tau_{alphabet})\alpha/k \qquad \text{msec} \quad (\alpha/k > 1)$$
$$t = \tau_{search}+\tau_{alphabet}+2372 \qquad \text{msec} \quad (\alpha/k = 1)$$

5. Experiment

The goals of this experiment were to (1) measure end-users' speed in locating entries in the four versions of the 800-Number Yellow Pages and (2) investigate whether they achieved a significant improvement in performance across trial blocks.

5.1. Prototype Design

Four versions of 800-Number Yellow Pages were used with 40, 80, 160, and 320 entries, and 8 pixels, 4 pixels, 2 pixels, and 1 pixel between each entry on the bar, respectively.

During the experiments, questions were presented in random order, and the slider buttons were reset automatically to the middle of the bar at the beginning of each question, to eliminate sequencing bias. Times and the usage of the fine tune buttons were automatically recorded .

5.2. Tasks

Subjects were trained and had a practice session before the experiment. Subjects performed 24 tasks using each of the four versions (a total of 96 tasks), in a counterbalanced ordering.

5.3. Subjects

24 subjects were used for the experiment. 12 were female and 12 were male. Their ages

ranged from 25 to 52. 2 of them had little computer experience, 16 had intermediate levels of experience (having used 2-3 systems), and 6 of them were advanced computer users. 21 of them did not speak English as their native language. All subjects had at least 6-years English education and had lived in the United States for at least 1 year. One subject, in addition to the 24 subjects, was trained as an expert AlphaSlider user so that experienced performance could be estimated.

5.4. Independent Variables and Dependent Variables

Independent variables of this experiment were

(1) Number of entries (number of pixels for each entry)

- 40 entries (8 pixels)
- 80 entries (4 pixels)
- 160 entries (2 pixels)
- 320 entries (1 pixel)

and

(2) Trial blocks

- first 8 tasks
- second 8 tasks
- third 8 tasks

Dependent variables of this experiment were

(1) Time to accomplish each task, and

(2) Number of times the fine-tune buttons were used in each task.

5.5. Procedure

Subjects were given a brief explanation of the 800-Number Yellow Pages and procedures for the experiment. Training included clicking and dragging the slider button, dragging left and right, and clicking on the slider bar, 'Dial' button, and Pause/Resume buttons. A practice session was given to help users understand how the actual tasks would be done and a standard consent agreement was signed. After the experiment, participants filled out a questionnaire on their computer skills and educational backgrounds.

5.6. Results

The mean times and standard deviations for subjects to select one entry appear in Table 1 and Figure 3. An ANOVA for the number of pixels between entries showed a significant main effect for version ($F(3,92)=24.4$, $p<0.01$). The mean times for the experienced AlphaSlider user to select one entry are also shown in Table 1.

The mean number of times required by subjects to press the fine-tune buttons per question are 4.1, 2.1, 1.1, and 0.23 where the number of pixels between entries is 1, 2, 4, and 8, respectively.

	1 pixel	2 pixels	4 pixels	8 pixels
Mean: 24 subjects	12.8	9.8	7.7	6.9
(Standard deviation)	(3.8)	(2.3)	(2.3)	(1.9)
Mean: expert	10.8	7.1	5.5	4.3

Table 1. Mean and standard deviation for time to complete one task (sec)

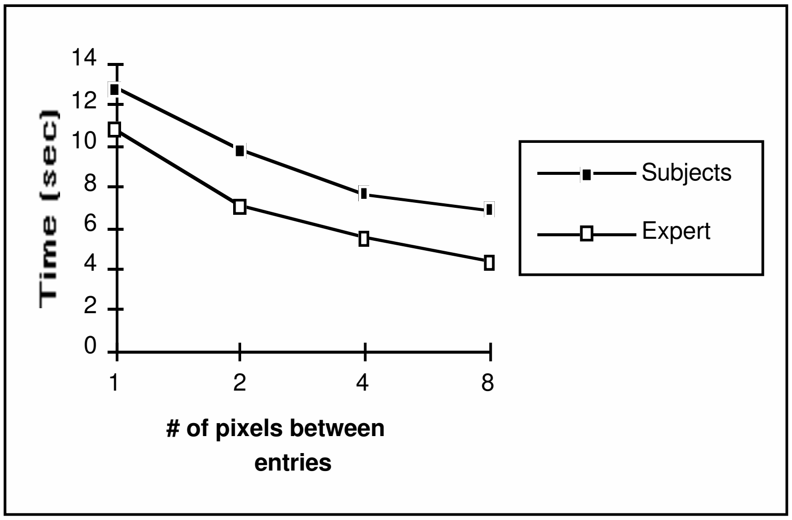

Figure 3. Graph of mean time to complete one task

The mean times required for subjects to select one entry, grouped by the time line in each session are shown in Table 2 and Figure 4. Improvements over the three trial blocks were statistically significant ($p < 0.05$) for the 2 and 8 pixel cases.

	1 pixel	2 pixels	4 pixels	8 pixels
First 8 questions	13.55	10.73	8.46	7.48
Second 8 questions	12.95	9.54	7.83	6.83
Third 8 question	12.35	9.71	7.78	6.41

Table 2. Mean times to complete one task grouped by time line in each trial block (sec)

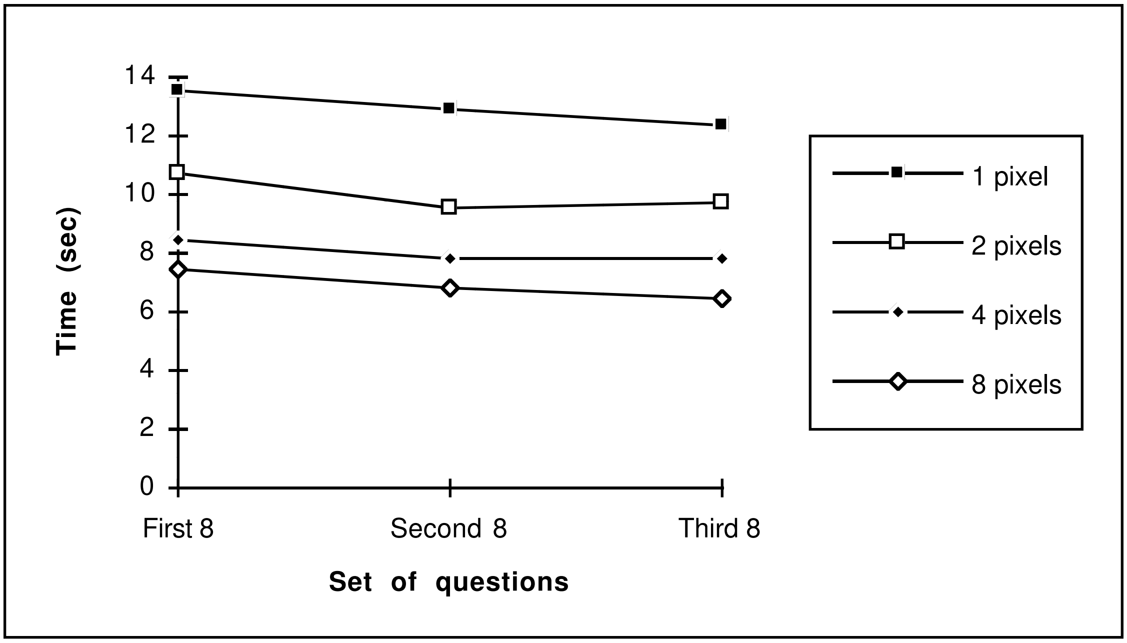

Figure 4. Graph of mean times (sec) to complete one question grouped by time line in each trial block

6. Discussion

The results of the mean times to select one entry (Table 1 and Figure 3), the mean number of times to push fine-tune buttons in one question, follow the formulae of performance estimation for overall time. Approximate values derived from the results are, $\alpha = 4$, $\tau_{search} = 2141$, and $\tau_{alphabet} = 1301$. It takes as much as 1.3 seconds for subjects and the expert to recognize the alphabetical order of the entry because each entry has several words. The average number of words per entry is 3.7.

This experiment was run using a computer display and mouse input, but users of the proposed credit card size device might perform more rapidly. Using a hardware slider button would eliminate the search time for mouse cursor on the display and provide better tactile feedback.

Since the average entry had 3.7 words, users' performance might be computed as 17wpm (1 pixel), 23wpm (2 pixels), 29wpm (4 pixels), and 32wpm (8 pixels). This hypothetical performance would increase as the average number of words per entry increase.

The expert user performed 15-37% better than the mean for the 24 subjects suggesting that the AlphaSlider does require some skill to master. The experiment monitors observed that the 24 subjects often used fine tune buttons while the expert user tried to drag the slider and avoid using the fine tune buttons. This suggests that a longer training period is needed for subjects to master the slider widgets.

7. Conclusion and Future Direction

Results of this experiment suggest that AlphaSlider can give users a new input method to retrieve data, and its performance is at the same level as touchscreen keyboards. They also suggest that further research about AlphaSlider is needed. Important items to examine are:

- slider design when the number of entries exceeds the number of horizontal pixels
- performance when sliders and buttons are made of hardware devices
- combination of AlphaSliders and other information visualization techniquges.

The advantages of this AlphaSlider approach are that there are no typing errors, that performance increases as the number of words per entry increases, and that keyboards are not necessary. Therefore, once data entries are entered to the computer system, this approach is suitable for retrieving data by personal and corporate names, technical terms, etc., which have more than 2 words and are often misspelled when typed out on a keyboard. The disadvantages of the current implementation are that the number of entries is limited by the number of pixels, and that it also does not allow users to input words freely. Considering these advantages and disadvantages, potential applications of AlphaSlider are personal database retrieval on palmtop computers, corporate database retrieval on electronic white boards, and patient database retrieval on medical workstations, where data is already entered into the systems and users are not allowed to use keyboards to retrieve data.

Acknowledgements: We thank the members of the Human-Computer Interaction Laboratory, who made many useful suggestions, and the subjects for their cooperation. We also appreciate Toshiba Corporation for its financial support of this research.

References

Ahlberg, C., Williamson, C., and Shneiderman, B. (1992), "Dynamic Queries for Information Exploration: An Implementation and Evaluation," *Human Factors In Computing Systems - CHI '92 Conference Proceedings*, 619-626.

Card, S., Moran, T., and Newell, A.(1983), *The Psychology of Human-Computer Interaction*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Fitts, P. M. (1954), "The information capacity of the human motor system in controlling the amplitude of movement," *Journal of Experimental Psychology*, **70**, 193-242.

Plaisant, C. and Sears, A. (1992), "Touchscreen interfaces for flexible alphanumeric data entry," *Proc. of the Human Factors Society - 36th Annual Meeting.*

Sears, A., Revis, D., Swatski, J., Crittenden, R., and Shneiderman, B. (1993), "Investigating Touchscreen Typing: The effect of keyboard size on typing speed," *Behaviour & Information Technology*, **12**, 1 (Jan-Feb 1993), 17-22.

Weisner, S. (1988), "A Touch-Only User Interface for a Medical Monitor," *Proceedings of the Human Factors Society - 32nd Annual Meeting*, 435-439.

Williamson, C. and Shneiderman, B. (1992), "The Dynamic HomeFinder: Evaluating dynamic queries in a real-estate information exploration system", *Proc. ACM SIGIR Conference*, 339-346.