*First publ. in: Proceedings / IEEE Conference on Information Visualisation, 2007, pp. 145-149*

# Dynamic Text Filtering for Improving the Usability of Alphasliders on Small Screens

Thorsten Büring, Jens Gerken & Harald Reiterer
University of Konstanz
HCI Group
Universitätsstrasse 10, 78457 Konstanz, Germany
{buering, gerken, reiterer}@inf.uni-konstanz.de

## Abstract

*Previous research has shown that Alphasliders are an effective tool for searching an alphabetically sorted list when only limited screen space is available for the graphical user interface. To improve user satisfaction, we propose equipping the widget with a novel text filter to dynamically limit the slider range. In this way, users are supported in locating target items and in identifying records that are missing. The results of a comparative user evaluation run on a Personal Digital Assistant showed that 8 out of 12 participants preferred the filter widget to the classic interface. We further suggest an enhanced Alphaslider design to speed up user interaction.*

## 1. Introduction

The Alphaslider is an interface widget that allows users to rapidly search and explore lists with thousands of alphanumeric items. Since it requires only a small amount of screen space, it thus provides a valuable solution for integration into mobile devices. The slider may, for instance, serve as a powerful interface to a music list on a mp3 player, to a contact directory on a smartphone or as a dynamic query module for Starfield-like applications on Personal Digital Assistants (PDAs) (e.g. [3]). PDAs and state-of-the-art mobile phones like the recently announced Apple iPhone are operated using a touch screen. While touch-input can be more straightforward than using a mouse, it is also less precise. The Alphaslider is a highly interactive widget and each screen contact causes a system response. Thus operating it with touch-input requires a very high level of accuracy and good fine-motor skills.
Our contribution is to propose a novel filter mechanism for improving Alphaslider interaction. In this paper we review previous work and discuss our motivation for developing a filter feature. We also present the results of a comparative PDA usability study that investigated the effect of the filter on user preference and performance. The paper concludes with some suggestions for further Alphaslider improvement.

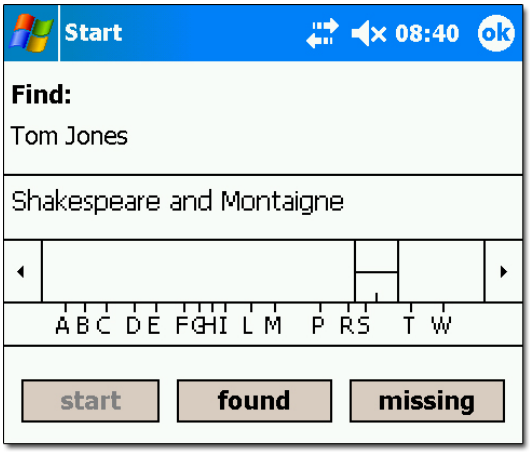


**Figure 1. The classic Alphaslider interface that was implemented for the user test. The second text line displays the target item ('Tom Jones'), the line below is the viewport to the list.**

## 2 The Alphaslider

The Alphaslider was first proposed in 1994 by [1]. Based on various prototypes and a user study, the authors suggested a slider design as shown in Figure 1. It consists of a one-line text output, a slide area, a two-tiled slider thumb, arrow keys and a letter index that visualizes the distribution of initial letters inside the alphabetically sorted

list. The most infrequent letters are removed to avoid overlappings. For searching a given record in the list, users may first, and guided by the index, jump to the list area containing the initial letter of the target item. For further list navigation, users can then drag the slider thumbs with different granularities. Dragging the upper tile of the thumb causes the system to skip 20 items per mouse movement, the lower tile corresponds to a granularity of one item per mouse movement. The direction of the list navigation is determined by the direction of the drag operation. While dragging, the active tile is highlighted and the text output is rapidly and continuously updated. The arrow keys provide an alternative to the fine-tuning of the lower thumb. Tapping one of them causes the next item to be displayed, tapping the other, the previous item.
When searching for a particular item, more conventional techniques like keyboard-filtering may in some cases provide a better performance than using an Alphaslider. On the other hand, typing text does not prevent misspellings or inserting inappropriate values. On devices such as PDAs, text input widgets also hamper interface interaction. A virtual keyboard, for instance, overlaps a considerable amount of the limited screen real estate. However, the main drawback of keyboard filtering is that it does not allow for a non-target-oriented list exploration. Using the Alphaslider, users can conveniently browse through a list without a particular target item in mind.

## 3 Related Work

There have been a number of papers with the objective of improving the Alphaslider design, but all of them focus solely on providing a more flexible control of the granularity of slider movement. For instance, [2] propose a popup vernier that is invoked on demand by pressing an auxiliary button. The control shows a zoomed grid and supports the user in dragging objects with sub-pixel-pitch resolution. The same caliper metaphor, but enhanced by a pressure modality for fluently adjusting the vernier granularity, was implemented in [6]. The FineSlider as described in [5] uses a virtual rubber-band to control the slider movement. When users tap on the slider bar and drag, a rubber-band appears between the cursor position and the slider thumb. The more the band is stretched, the faster the thumb is moving towards the cursor. Another approach to searching a sorted list on devices such as cell phones is BinScroll [4]. The technique is based on a manually operated binary search algorithm but, just like keyboard filtering, it does not allow for an intuitive list exploration.

## 4 Filter Design

Alphasliders have proved very successful for searching lists on a limited screen real estate. On the other hand, using such a slider can be exhaustive and even error-prone. The obvious way of using the widget is to apply an oscillating search strategy. Users drag the slider thumb, say from left to right, until the output value shows a title that alphabetically follows the target title. The constant need for determining the alphabetical order is cognitively very demanding and causes a delay in reaction. Having eventually decided to switch the dragging direction, the same procedure is then repeated for the right to left movement. With growing proximity to the target title, users become increasingly stressed as each pen movement too coarse sets them back in the search process. Another factor causing stress is uncertainty. All Alphaslider evaluations have been performed under the assumption that target items are always present in the list. This assumption is fairly restrictive since in most real-world scenarios users do not know the content of a given data set in advance. Thus the primary user objective when using an Alphaslider is often not to find an item but rather to determine whether an item is present in the list. In the case of a missing item, users frequently need several navigational attempts to assure themselves that the item really does not exist. This is due to the highly sensitive slider interaction, which causes users to worry whether they have unintentionally skipped the target item.

To improve the Alphaslider's usability we propose a novel text filter that enables users to restrict the range of list items over which the slider thumb can operate. The idea of the filter is based on the following observation: depending on the size of the underlying list it may take users quite some time to locate a particular target item, but at the same time, and due to the alphabetic ordering, they are very likely to come across items that start with the same substring as the target item. A substring may range from a single letter to several words. Users can set a filter by dragging the pen from left to right over the output text line or tapping the last character of the matching substring. As shown in Figure 2, the selected text is highlighted by a gray bar. The filter limits the slider range to only those items that start with the given substring. Dragging the thumb or tapping the arrow buttons only works within the subset of valid items. To remove the filter, users simply unselect the highlighted text.

While searching the list, users can extend the filter iteratively and thus level off to the target item. Overall, they gain more control of the slider interaction, and the consequences of any unintentionally coarse pen movements are

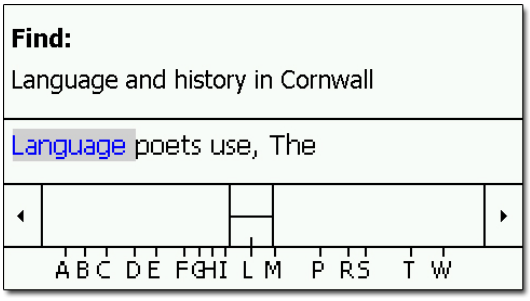


**Figure 2. The Alphaslider equipped with the novel text filter. Users can limit the slider range to items that contain a previously highlighted substring (in this case 'Language ').**

very much less time-consuming and thus less frustrating. Since implementing the filter does not require a change in the general slider layout, the text filter could also be applied as an add-on feature to the Alphaslider derivatives discussed in the previous section.

## 5 Experiment

To investigate the effect of text filtering on the usability of Alphasliders, we conducted a comparative usability study in which users had to search for target items in a list of 10,000 book titles. Most of the titles were in English, some in German. The list was ordered alphabetically with blanks preceding characters, i.e. 'Language of the Arts' was sorted before 'Languages'. Common leading articles such as 'A' and 'The' were moved to the end of the title, separated by a comma. For simplicity and to avoid minor but nevertheless irritating difficulties about ordering, no title contained any special characters such as umlauts, accents or punctuation marks. Only half of the target items were present in the list.

For the test we developed two Pocket PC applications using the Microsoft .NET compact framework 1.1. Both interfaces featured a classic Alphaslider widget as discussed in Section 2 and shown in Figure 1. One interface was additionally equipped with the novel text filter mechanism (Figure 2). The experiment was run on an Ipaq hx4700 Pocket PC with Windows Mobile 2003. The device featured a 624 MHz processor, 64 MB SDRAM and a 480x640/64K color VGA touchscreen.

Task completion time and accuracy were automatically logged on the device. A pre-test questionnaire was used to collect demographic data. In a post-test questionnaire we asked our participants for their system preference and which of the two interfaces they thought was more efficient to use, and which more effective. To encourage a more realistic time-sensitive interaction, we offered a reward of EUR 60 to the participant who completed the given tasks within the shortest time and with the highest accuracy.

### 5.1 Hypotheses

1. *Users would prefer the text-filter interface to the classic Alphaslider.*
   Due to the better control of the slider interaction, we expected that the filter mechanism would increase user satisfaction. We also anticipated that the feature would have a positive effect on user confidence in respect of the task results for missing items.
2. *There would be no improvement in terms of task completion time.*
   While the text filter may enable users to limit the oscillation of the slider movement, it was assumed that this benefit would not compensate for the additional time span that would be needed to first interrupt the thumb-dragging, then set the filter and subsequently re-initiate the drag operation.

### 5.2 Participants

For the study we selected 12 participants, 6 male and 6 female. 11 of them were students at the University of (*removed for review*). Their ages ranged from 20 to 29 years. The oldest participant was 36 years of age and worked as a supervisor. Their fields of study varied to a great extent. Two participants were students in the field of computer science. The pre-test questionnaire also revealed that three of our participants actually owned a PDA and one other had at least tried one, and that these four were therefore familiar with the general pen-interaction concept. All of our users were regular PC and internet users.

### 5.3 Experimental Design

We used a counter-balanced within-subjects design, balancing the two interface types and the two sets of task-items. This resulted in four different groups mirroring all possible variations of interface and task set order. We randomly assigned three participants to each group. For analysis, we mainly used repeated measures ANOVAs (RM-ANOVAs). Our independent variable was interface type (classic Alphaslider vs. text-filter interface). The dependent variables were task-completion time (in seconds), system preference, and accuracy (number of incorrectly answered tasks).

### 5.4 Procedure

The session started with a short written introduction and the pre-test questionnaire. Next, users were introduced to the pen handling on the PDA. During this process, the pens were recalibrated by the participants themselves. Next, users were handed a written description of the first interface, and they had time to try the application on their own and ask questions. During this training period, each participant searched for at least five different items. When participants showed that they had understood the interface, they started the search for the first of 20 target items by tapping a button labeled 'start'. The task-items were presented within the interface shown in Figure 1. Having found an item, users tapped a button labeled 'found' and then continued with the next item. If they could not find the target item, they tapped a button labeled 'missing' and were then asked how confident, on a 5-point-scale, they were that the item really did not exist in the list. participants were aware of the fact that some items were missing, but they did not know how many items were missing, or how many items in total they had to search for. They were just told that the application would stop automatically at a certain point. After completing the search with the first interface, the second interface was introduced in the same manner i.e. with a written introduction and a training session. At the end, the users had to complete a post-test questionnaire that asked for their system preference and their opinion about the efficiency and effectiveness of the two interfaces. Each participant was given a movie theatre voucher worth EUR 5.-. Experimental sessions lasted about 30-45 minutes.

### 5.5 Results

In our first hypothesis we assumed that participants would prefer the text-filter interface to the classic Alphaslider. Eight of our participants did indeed prefer the text-filter interface and only three the classic Alphaslider; one was unsure. A statistical analysis revealed that this difference is not significant ($X^2(1, N = 11) = 2.273, p = 0.132$), which might be surprising at first but is a common statistical problem of rather small sample sizes. So although we cannot confirm our hypothesis the results support it quite noticeably. participants gave as their reasons that the text-filter interface was more convenient and faster to use, and some also stated that, as predicted in Section 4 (Filter Design), it helped to reduce the effect of errors such as unintentional tapping beside the slider thumb. However, the three participants who preferred the classic Alphaslider mentioned that it caused them less thought and was less confusing and therefore easier to use. One of them actually stopped using the text filter mechanism during the experiment and used this interface in the same way as the classic Alphaslider.
When we asked participants about their confidence that the current item really was not in the list after they had chosen that option, the answers were not as expected. Our participants were always very confident in their choice, regardless of which interface had been used and of whether their choice had, in fact, been correct. On average, participants made 0.45 errors with the classic Alphaslider but 1.18 with the text-filter interface. It seems that the possibility of limiting the slider range resulted in higher inattentiveness. Nine participants completed the tasks with the classic Alphaslider without an error but only four did so while working with the text-filter interface. Another interesting observation is that all participants selected their preferred interface as being the more efficient one, although this perception was correct for only five of our participants. With regard to the second hypothesis concerning the task completion time, we could find no significant or notable difference between the two interface types. It took our participants on average 26.67 seconds (SD: 11.74s) to find an item with the classic Alphaslider and 28.4 seconds (SD: 12.55s) with the text-filter interface ($F(1,11) = 1.195, p = 0.298$, not significant). Furthermore, there was no difference between the interface types for tasks in which items were not actually in the list (30.27 seconds for the classic Alphaslider compared to 30.73 seconds for the text-filter interface, $F(1,11) = 0.45, p = 0.837$, not significant). In consideration of individual results, showing that it is indeed possible to be more efficient by using the text-filter interface and the strong preference voting, we were encouraged to further optimize that approach.

## 6 Design Recommendations

To isolate effects in the user evaluation, the test interfaces differed only in the availability of the filter mechanism. A separate, additional objective is to improve the general Alphaslider interaction design. Its performance depends heavily on the extent to which users can focus solely on the output text line. Using the classic Alphaslider this is hardly feasible since, with each initiation of a drag operation, users have to pay attention to the slider thumb. If they are not able to hit the thumb precisely with the first pen contact on the screen, they are penalized by a rather arbitrary jump in the list. This problem becomes even more acute when dealing with small displays. While in the desktop world users can drag the thumb over long distances, PDA users quickly reach the end of the physical screen, in particular when the thumb is close to one of the slider extremes. Hence users are frequently forced to switch their attention to reinitiating the drag operation. In the redesign implemen-

tation as shown in Figure 3, the slider thumb was replaced by a continuous two-tiled slider area. A vertical red line indicates the current position in the list. Users can initiate a drag operation anywhere on the slider area. The two tiles still correspond to the two different drag granularities, and with each pen movement the position of the red line is updated. Jumping to an initial letter is achieved by directly tapping on the character index below the slider. Unlike the classic Alphaslider, the redesign does not require users to hit a small size thumb. Instead the system allows for longer drag distances, demands a lesser degree of fine-motor skills and causes less user distraction.

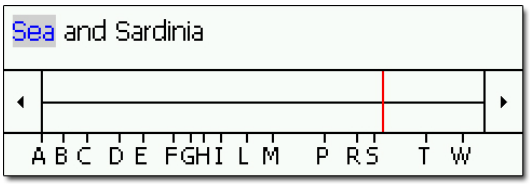


**Figure 3. The Alphaslider redesign to speed up user interaction. Dragging operations can be initiated anywhere on the slider tiles.**

## 7 Conclusion

In this paper we presented a novel text filter for supporting users when searching an ordered list using an Alphaslider. The results of a user test with 12 participants showed that users preferred the suggested mechanism to the classic slider interface. A downside is that the greater complexity of the filter application seems to have a negative effect on accuracy. More research is needed to clarify this aspect. To improve system performance we suggested a redesign to speed up the slider interaction. For future work, we also plan to extend the Alphaslider to support searches for any word within any item that the list contains. Based on an inverted index, users could for instance search for all book titles that feature a certain keyword without it being necessarily the first term of the title.

## 8 Acknowledgments

This work was supported by the DFG Research Training Group GK-1042 ”Explorative Analysis and Visualization of Large Information Spaces”.

## References

[1] C. Ahlberg and B. Shneiderman. The alphaslider: a compact and rapid selector. In *CHI ’94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 365–371, New York, NY, USA, 1994. ACM Press.

[2] Y. Ayatsuka, J. Rekimoto, and S. Matsuoka. Popup vernier: a tool for sub-pixel-pitch dragging with smooth mode transition. In *UIST ’98: Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 39–48, New York, NY, USA, 1998. ACM Press.

[3] T. Büring and H. Reiterer. Zuiscat: querying and visualizing information spaces on personal digital assistants. In *MobileHCI ’05: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*, pages 129–136, New York, NY, USA, 2005. ACM Press.

[4] J. Lehikoinen and I. Salminen. An empirical and theoretical evaluation of binscroll: A rapid selection technique for alphanumeric lists. *Personal Ubiquitous Comput.*, 6(2):141–150, 2002.

[5] T. Masui, K. Kashiwagi, and I. George R. Borden. Elastic graphical interfaces to precise data manipulation. In *CHI ’95: Conference companion on Human factors in computing systems*, pages 143–144, New York, NY, USA, 1995. ACM Press.

[6] G. Ramos and R. Balakrishnan. Zliding: fluid zooming and sliding for high precision parameter manipulation. In *UIST ’05: Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 143–152, New York, NY, USA, 2005. ACM Press.