Data Aware Selection Tools

|  |  |  |
| --- | --- | --- |
| Paymahn Moghadasian  University of Manitoba  66 Chancellors Cir  umpaymah@cs.umanitoba.ca  ? | Pourang Irani  University of Manitoba  66 Chancellors Cir  irani@cs.umanitoba.ca  ? | Niklas Elmqvist  Purdue University  ?  elm@purdue.edu  ? |

# ABSTRACT

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies

bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*CHI’12,* May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...$10.00.

In this paper we present selection tools that utilize pre-existing knowledge of the distribution of the underlying data to ease browsing and selection.  These new data-aware selection tools include a slider, a range slider, and an improved lasso selection method.  Standard models of these tools suffer from two common problems: uniform visual representation of non-uniform data, and poor subpixel data querying.  We use embedded visualisations to better represent the data encoded by the tools and provide data-aware interactions to give users more efficient subpixel data querying.  Through several controlled user studies, we find that our proposed data-aware slider outperforms standard alphasliders and default trackbars.  We also find that a data-aware range slider and lasso selection outperform current state-of-the-art methods.

## Author Keywords

Dynamic query, information visualization, slider, data selection, Alphaslider, lasso,

## ACM Classification Keywords

Need help with this

## General Terms

Need help with this

# INTRODUCTION

In 2008 the average American consumed, on average, 34 gigabytes of information daily which corresponds to a national daily consumption of 3.6 zettabytes (1021 bytes or a million million gigabytes) [[1](#1)]. On a daily basis users need to be able to turn the vast amounts of information they come across in more manageable pieces. Dynamic queries provide easy to use, powerful and efficient tools and interfaces which allow users to rapidly and reversibly query data [[2](#BSh96)]. Currently, there are many tools available to users which benefit from the advantages of dynamic queries such as the slider, checkbox or button. The Alphaslider is a slider designed to query large lists of alphabetically sorted alphanumeric items [[3](#Ahl94)]. The range slider is a slider which allows users to query a range of data items. This is a natural extension of the Alphaslider which only allows for single item querying. The lasso, a technique used to select multidimensional data is an extension of the range slider where users lasso graphed or visualized data to highlight or zoom in on it.

The first major problem with these dynamic querying tools is that they either do not possess the ability for subpixel querying or it is very limited. This means that when the encoded information exceeds the total number of pixels allotted to the interface the user either cannot access data found within each pixel or the user has great difficulty doing so. A second problem is the visualisation of data encoded by the tools. For example, in the Alphaslider, low frequency indices are not shown which forces the user to guess the existence of the index itself and any items that may fall within it.

We proposed several solutions for each tool to address the issue of subpixel querying. In the case of the slider we compared our prosposed solutions against the Alphaslider and the default Windows trackbar. Our proposed slider solutions are variations of a common idea: allow the user to query a pixel and then to query all items within that pixel and a few items in the neighbouring pixels. For the range slider we did have not reached this part of the research yet. For the lasso we did…To solve the issue of data visualisations we used embedded visualisations such as those proposed by Willett et al [[4](#Wil07)].

Our results show that users can query significantly faster with data aware widgets when compared against the standard tools. Our proposed interface X allowed subjects to query y% faster than the Alphaslider and z% faster than the default trackbar. Our proposed range slider outperforms standard range sliders by t% while our lasso technique gave users no improvement for low and medium density information but showed a% improvement over the current lasso technique.

This paper provides novel solutions to an aging problem. Although the proposed solutions perform more efficiently than previous standards there is still work to do in fine-tuning and possibly developing hybrid designs that take advantage of aged and novel solutions. In this paper we discuss will present the problem in greater detail, discuss related work and then go on to describe the experimental design and results of our controlled user study. We will wrap up the paper by discussing the implications of the paper.

# RELATED WORK

Need to write this

# Experimental design

## Introduction

An experiment was conducted to compare different designs of density aware sliders against the Alphaslider and the standard Trackbar found in WinForms.

## Apparatus

The interfaces used in the experiment were built using Visual Studio 2010Professional. A 23.5 inch Dell monitor with a resolution of 1920x1080 pixels with a standard 3 button mouse was used. Query results were displayed in Microsoft Sans Serif with a font size of 8.25 while query targets were displayed in Microsoft Sans Serif with a font size of twelve. An Intel i5-2400 CPU with a clock speed of 3.1 GHz was used along with 4 GB of RAM.

## Interfaces

A total of 7 different sliders were used in the experiment (Figures 1 through 6). In all cases the text output was RSVP and was displayed under the slider. Between searches subjects had to press a “Start” button indicating they were ready and upon completion of the search the subject had to press a “Confirm” button indicating they have completed the search. This is going to be changed. Instead of clicking on buttons the user will toggle states by pressing the space bar.

### Windows Trackbar

The Trackbar (Figure 2) is what software designers using Visual Studio find as the default slider. With this slider users can navigate through the list by dragging the slider thumb to various pixels. Clicking to the left or right of the thumb will navigate five items behind or ahead of the current one while using the left and right arrows on the keyboard will navigate one item at a time.

### Alphaslider

The Alphaslider (Figure 1) has four navigation techniques. Users can jump directly to an item in the list by clicking anywhere in the bounded area above the letters. Users can navigate through the list at a rate of ten items per mouse movement by clicking in the top tile of the slider thumb and dragging. Users can also navigate one item at a time either by clicking on the arrows at the ends of the slider or by clicking in the bottom tile of the slider thumb and dragging.

### ActiveArea Slider

The ActiveArea Slider (Figure 3) allows users to navigate by clicking on and dragging the slider thumb, by clicking on and dragging the secondary red slider or by pressing the left or right arrow keys on the keyboard. The main slider has a variable size which changes based on the density of information of the pixel it queries. Dragging the secondary red slider allows users to query other items found in the pixel being queried by main slider. The red slider is bound by the edges of the main slider.

### Histogram Slider

The Histogram Slider (Figure 4) functions very similarly to the ActiveArea Slider. The most significant difference is that the secondary red slider moves up and down along the histogram as opposed to left and right along the slider. The size of the slider stays constant. Users can still navigate one item at a time by pressing the left and right arrow keys.

### MouseWheel Slider

The MouseWheel Slider (Figure 5) is also quite functionally similar to the ActiveArea slider. However, this slider allows subpixel querying through the use of the mouse wheel. Rolling the mouse wheel up will cause the next item in the list to be queried while rolling the mouse wheel down has the opposite effect.

### MultiValue Slider

Need to do pilots and find out which (if either) multivalue slider one is better

## Hypotheses

This paper is primarily concerned with designing the most efficient slider. While accuracy is an important factor in designing a slider that attribute falls largely upon the user. Because of this speed is the most significant factor to measure. The speed with which a user locates an item is largely dependent on total mouse movement. For each of the sliders there is a period of querying where the user approximates the area of the target and a period of querying where the user does fine adjustments to acquire the target. Based on this assumption the following hypotheses can be made:

1. The MultiValue slider will outperform all other sliders in low local density areas because additional mouse movement will be minimized.
2. The Histogram Slider will be faster than the ActiveArea Slider because reacquiring the ActiveArea Slider’s slider in low local density areas is difficult.
3. The MouseWheel Slider will perform better than the Alphaslider because rolling the mouse wheel is far more precise than moving the mouse.
4. The Trackbar will perform the worst because it depends on keyboard input for single item navigation.

## Experiment Variables

### Independent Variables

1. Type of interface
2. Local target density
3. Slider density

### Dependent Variables

1. Speed of acquisition
2. Error rate
3. Subjective satisfaction

## Tasks

Subjects completed three trials for each permutation of the independent variables. Each target was randomly generated at runtime. To mitigate learning effects new data was randomly selected upon interface change. With each new task the thumb was brought to the beginning of the slider.

## Participants

## Procedures

A pilot study was first done to weed out poorly designed and inefficient interfaces. Prior to the timed trials the subjects were given five minutes to familiarize themselves with each interface while reading interface specific instructions and ask questions. During the timed trial the subjects were not allowed to ask questions. Upon completion of the experiment the subjects were asked to fill out a questionnaire.

# Results

Need to write this

# Conclusion

Need to write this

# ACKNOWLEDGMENTS

Need to write this

References

x

|  |  |
| --- | --- |
| [1] | Roger E. Bohn and James E. Short, "How Much Information?," San Diego, December 2009. |
| [2] | Shneiderman B., "Dynamic Queries for Visual Information Seeking," Maryland, 1994. |
| [3] | Christopher Ahlberg and Ben Shneiderman, "The Alphaslider: A Compact and Rapid Selector," in *Human Factors in Computing Systems*, Boston, 1994, pp. 365-371. |

x

**The columns on the last page should be of approximately equal length.   
Remove these two lines from your final version.**