Value Bars: An Information Visualization and Navigation Tool for Multi-attribute Listings

Richard Chimera

Human-Computer Interaction Laboratory
A.V. Williams Building
University of Maryland
College Park, MD 20742-3255
(301) 405-2757
carm@cs.umd.edu

INTRODUCTION

The need for better information visualization and navigation tools is widely recognized [1], [5]. It is difficult to sort and continuously resort tables or listings by more than one attribute and still maintain an understanding of the origin and natural position of items. The concept of "value bars" was created to help users visualize and navigate large information spaces that have characteristics of a line-oriented listing with multiple, quantifiable attributes. In general, value bars are useful for analyzing multi-attribute listings and tables where a particular sort order should be maintained and analysis of the top percentage of items within each attribute is beneficial. The main features are:

- the ability to see in one view an attribute distribution overview for the "important" items (as defined by attribute values) in a fisheye view [3] variant,
- very small screenspace footprint,
- the ability to see at once many attribute overviews,
- the ability to locate outliers and exceptions, and
- extremely low cognitive load navigation.

This work was spawned from the study of a novel way to visualize large tree data structures, called Tree-maps [4]. A more detailed description, discussion, future value bars research, and results of a usability study can be found in [2].

DESCRIPTION

Value bars are thin, vertical strips added to a text window (figure 1), placed next to the scrollbar if one exists; any number of value bars may be added as screen space allows. Each value bar maps one specific quantifiable attribute shared by items. A partitioning algorithm equates the sum of the items' values, or *weights*, to the value bar height. Each item's weight is converted to a height in the value bar proportional to its part of the total weight of all items. If the height exceeds the minimum threshold height, the region is stacked vertically in the value bar, placed from top to bottom in the same order as appearing in the listing sequence. The end result is a graphic image that looks like a ladder with varyingly spaced rungs. Once partitioned the value bar regions are not changed, moved, or scrolled, all regions are always in view. Items represented in the value bar provide a global view of the distribution of an attribute's values in the listings, providing another variant of the fisheye view [3]. An item is represented by differing height regions in different value bars; an item may not be represented in every value bar.

Item selection, either by a mouse click in the text window or in any value bar, highlights all representations of that item. Clicks in value bars scroll the text window to center the selected item's text representation. This immediate navigation allows users to jump around and examine individual items as the need arises. There is no guess work as to how much to scroll, users continually concentrate on the task with minimal effort spent at the interface level.

Another value bar component is its "visibility marker." Many scrollbars provide an indicator of how much of the entire listing is currently in view; this concept is a requirement of a good value bar implementation. When the text window is scrolled, the visibility marker in each value bar is moved appropriately up or down within the value bar. Visibility markers change size and move independently of each other (though synchronized with the scroll), further showing attributes' differing distribution characteristics.

APPLICATION TO UNIX DIRECTORY LISTINGS

Consider a Unix directory listing with value bars for file size and file age (figure 1). Along the right side of the window are two value bars. The S value bar maps file size largeness (Size) and the Y value bar maps file modification recency (Youth); both value bars use a linear weighting assignment. The rest of the window is a normal scrolling textpane of the directory listing. In the Youth value bar, more recently modified files have a larger weight. The Size value bar has bigger variances in region heights because its values vary more widely than values in the Youth value bar.

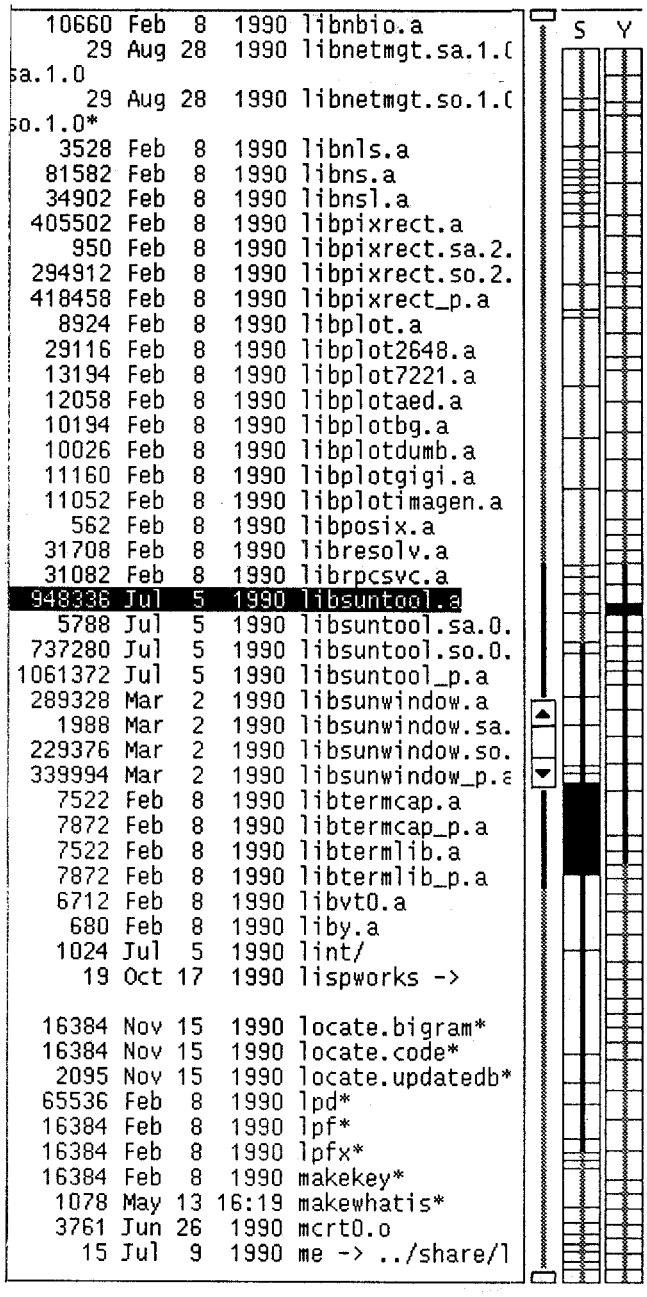

Figure 1. The (familiar parts of a) Unix directory listing using the command "ls -l /usr/lib". The two value bars to the right of the scrollbar represent file size (S) and file modification recency, or youth (Y). The taller the region in a value bar the greater is that listing item's attribute value weight. The currently selected item is the same among the listing and the value bars and is highlighted in inverse video for all representations. Notice the use of value bar specific visibility shading, equivalent in function to the scrollbar's visibility shading, to show which items of the whole are currently visible in the text window. Only the topmost weighted items, different for each attribute, are represented in a value bar.

DISCUSSION

One of the most important value bar advantages is its ability to provide a global distribution overview of attribute values in a single view. Users are able to compare an item's attribute value to other items in the same value bar without the need to scroll or rearrange the sort order many times. Noticing clusters of large regions together may be an important insight to users. The fact that an attribute has values that are relatively equal throughout the items (e.g. all files are similar in size) can be recognized easily. Noticing that one item has large regions in many value bars may be enlightening. There are many ways the single view of multiple attribute distributions can help users distill trends or notice interesting clusters. This also may allow new discoveries that couldn't have been made due to the high cognitive load of resorting a list many times while retaining acquired knowledge about items.

CONCLUSION

The value bar represents a unique combination of information visualization and navigation for multi-attribute listings. The powerful information visualization provides items' local detail in a global context of attribute values and their distributions. Navigation and interaction are simple, clean, and build on generic GUI concepts. Value bars can be applied in many diverse domains whose data have characteristics similar to multi-attribute listings or tables. Value bars are a useful tool which can be integrated unobtrusively into existing application environments.

REFERENCES

1. Beard, D., and Walker, J. Navigational Techniques to Improve the Display of Large Two-dimensional Spaces. Behaviour & Information Technology 9, 6, 451-466.

2. Chimera, R. Value Bars: An Information Visualization and Navigation Tool for Multi-attribute Listings and Tables. University of Maryland Department of Computer Science technical report CS-TR-2773.

3. Furnas, G. Generalized Fisheye Views. In Proceedings ACM CHI'86 Human Factors in Computing Systems Conference (Boston, MA, April 13 - 17). ACM, New York, 1986, 16-23.

4. Johnson, B., and Shneiderman, B. Tree-Maps: A Space-filling Approach to the Visualization of Hierarchical Information Structures. In Proceedings of ACM Visualization '91 Conference (San Diego, CA, October 22 - 25). ACM, New York, 1991, 284-291.

5. Mackinlay, J., Robertson, G., and Card, S. The Perspective Wall: Detail and Context Smoothly Integrated. Proceedings ACM CHI'91 Human Factors in Computing Systems Conference (New Orleans, LA, April 27 - May 2). ACM, New York, 1991, 173-179.