Beyond the Scrollbar: An Evolution and Evaluation of Alternative Navigation Techniques

D. Scott McCrickard
College of Computing
and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0280
mccricks@cc.gatech.edu

Richard Catrambone
School of Psychology
and GVU Center
Georgia Institute of Technology
Atlanta, GA 30332-0170
rc7@prism.gatech.edu

Abstract

Scrollbars provide a simple way to traverse an information space, but they provide little data about the actual contents of the space. Of the many visualization techniques that have been proposed, few have maintained the simple functionality of the scrollbar while showing improved performance on typical scrollbar tasks. This paper presents two enhancements to the scrollbar, a mural bar and a pile bar, which encode data about the information space contents into the trough of the bar. Results from an experiment suggest that these new devices lead to improved user performance on several common scrollbar tasks.

1 Introduction

Visual programming languages suffer from the *scaling up problem*: as a program's complexity increases, the information space it occupies can become large and unwieldy [5]. Indeed, many computer applications and environments require a user to deal with information spaces such as source code, textual lists, or graphical images that can become too large to be displayed in the desired level of detail using the available physical space. One solution to this problem is to use a navigation device to help the user traverse and understand the information space.

One common navigation device is the scrollbar, a small but powerful widget that informs the user of the relative size and position of the visible portion of the space and provides control over the information that is seen. The limited screen space requirement allows the remainder of the screen to focus on more detailed views, while the size and positional cues provide a sense of context. However, considerable information about the content of the space is not communicated in the scrollbar.

Many information visualization systems have been developed to address this detail-context problem (Section 6 describes some of the most relevant). However, the familiarity and ease of use of the scrollbar is often sacrificed. This paper explores whether the scrollbar can be augmented with information visualization techniques to create a navigation device that leverages the widely used scrollbar paradigm but provides additional information. This allows users familiar with scrollbars to build on an existing mental model and potentially shorten the time required to learn the interface.

In developing alternative navigation devices, we leverage the scrollbar navigation paradigm while increasing the amount of information that can be provided. Our navigation devices use the space inside the scrollbar to represent the information space with graphical lines, where properties of the space are reflected in properties of the lines. In our work, informational properties such as orderings, relative importance, and categorizations are shown with graphical properties such as location, overlapping, and color. Section 2 introduces the mural bar and the pile bar, two navigation devices that use information murals and the pile metaphor to communicate information.

The danger is that changes to the appearance of the navigation device will cause ease of use to suffer: tasks that could be easily accomplished with a scrollbar would become more difficult with the additional information in mural bars and pile bars. Since part of the appeal of a scrollbar is its simplicity, other navigation devices should not complicate the basic interface style yet must provide additional information in a sufficiently obvious and straightforward manner. Section 3 describes an experiment that tests whether users can perform a number of scrollbar-based tasks more quickly and easily with a mural bar or pile bar than with a scrollbar. Section 4 analyzes the results of this experiment, which suggest that participants perform equally well or better on the tasks using enhanced navigation bars. The paper concludes by examining the impact of this work on current and future applications and environments.

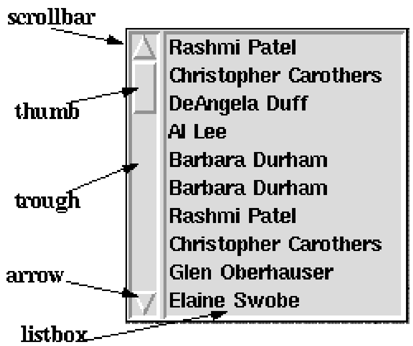

Figure 1: A typical scrollbar (left) and accompanying listbox. The thumb indicates the position and size of the portion of the list that is visible in relation to the entire list. Arrows move the thumb up and down within the trough and change the visible portion of the list. Clicking in the trough jumps the display toward the clicked location.

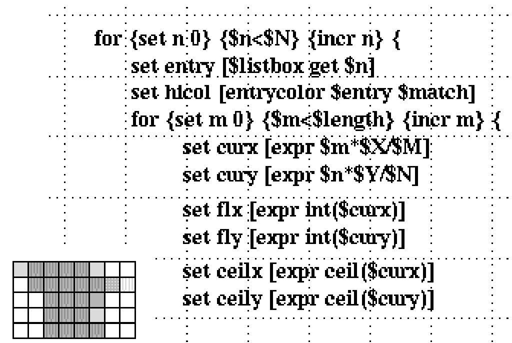

Figure 2: An illustration showing how an information mural is created. The inset grid of pixels represents the source code shown in the background. The density of the shading for a given pixel corresponds to the number of characters that touch it. While the code itself cannot be ascertained from the graphical representation, the general structure of the code, including indentations and line lengths, is evident.

2 Augmenting the Scrollbar

Scrollbars provide a familiar and widely accepted method for navigating information spaces. *Arrows* at the top and bottom of the scrollbar can be clicked with the mouse button to change the visible portion of the list. A rectangular *thumb* provides information about the relative size and position of the listbox entries as well as another means to navigate within it. The thumb slides up and down within a *trough* as the list is scrolled up and down. The size and position of the thumb in the trough are proportional to the size and position of the visible portion of the list. Clicking in the trough above or below the thumb jumps the display toward the clicked location. See Figure 1 for an illustration of the components.

While scrollbars adequately portray the size and position of the visible information relative to the entire space, data about the nature of the contents is not communicated. This section introduces two new scrolling devices, the mural bar and the pile bar, that contain many of the same features and functionality as a scrollbar but contain encoded information in the trough. The thumb, trough, and arrows have the same positions and functionality in mural and pile bars as in scrollbars (though in our implementation one cannot grab and move the thumb). To expand the visible information, the mural and pile bars contain graphical encodings in the trough area that display data about the space.

Although the mural and pile techniques are similar in their representation of individual entries, they differ in their allocation of screen real estate to these representations. We will discuss the particulars of each encoding technique, focusing on example scenarios where each technique may prove useful.

2.1 Mural Bars

Information murals, compressed graphical displays of an information space, have been used in a wide range of applications, including visualizations of software execution, numerical data, and general information [11]. Information murals represent a large information space with a smaller graphical space by mapping the elements in the large space into the smaller one. In the case of a text document, the characters in the document map to the pixels in the information mural to create a scaled down picture of the entire document. Highlighting certain words or phrases (such as function names, document headings, or user-specified search terms) with color can facilitate the identification of features of the document.

To understand how the information mural is created, think of a rectangle that is M pixels wide and N pixels high as an MxN grid that overlaps the entire list. A pixel in the grid is colored with intensity corresponding to the number of characters that touch it: the more characters that map to a pixel, the more intense the coloring becomes. The coloring effect that results from this mapping shows the size and number of elements in the list. See Figure 2 for a visual explanation of this process or read [11] for details on the algorithms used.

Information murals provide an overview of the entire information space using whatever physical space is available. Since an equal amount of space is given to each item, any item can be seen with equal clarity no matter where they are in the list. Murals are particularly effective at highlighting the general structure of an information space.

The *mural bar* combines the functionality of a scrollbar with the visual information of an information mural by encoding the contents of the scrolled information space in the trough of a scrollbar using the mural technique. The thumb of the scrollbar encompasses the portion of the mural that is visible in the larger view. The visible portion of the information space can be altered by clicking on the arrows, trough, or thumb as with typical scrollbars.

Mural bars can be an important part of a visual environment for textual programming languages. As

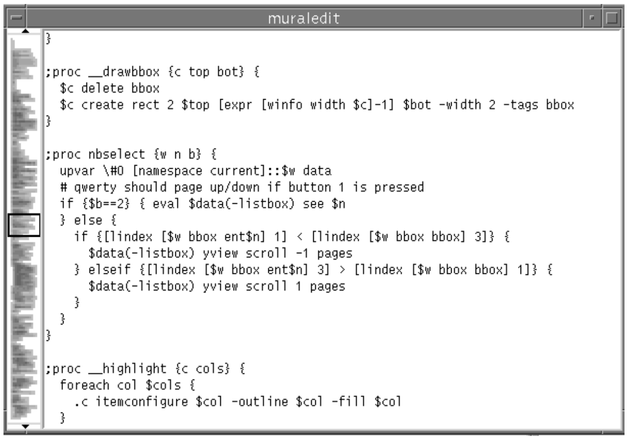

Figure 3: A mural bar shown with a large code file. Note that many of the structures found in a scrollbar (arrows, thumb, trough) are present in the mural bar. In addition, the general structure of the file (indentations, line lengths) is visible in the trough.

we see in Figure 2, indentations and line lengths are evident when displaying code. A programmer using a mural bar would be able to identify short and long functions, portions of the code that contain long print statements, and blocks of unindented comments. Incorporating colorings for certain command blocks could increase the number of structures a programmer could identify. Figure 3 shows an entire C++ file with an overview provided by a mural bar.

Visual programming languages (VPLs) seem particularly well-suited for use with the mural bar. Just as characters from words map to a single pixel in the code example, pixels from an original image could map to a smaller number of pixels in the mural bar. Our expectation is that programmers would recognize certain elements of the program by their general appearance, even with the reduced size.

2.2 Pile Bars

The pile metaphor introduces onto the computer desktop the real-world concept of piling items on top of each other. Just as documents can be piled on a desk and identified by their appearance, items on the computer desktop can overlap and be recognized by their color, size, or shape. The pile metaphor was introduced in a desktop document layout system [18]. Piles of documents created by the user had a disheveled appearance with parts of the document icons sticking out. The position and appearance of an icon could help the user find a document even if it were in the middle of a pile.

Applying the pile metaphor to a *pile bar* navigation device requires that the information to be displayed be delineable as distinct objects. Possible object delineations for information spaces include functions or objects from a code file, graphical entities from a visual programming language, or individual entries from a text list.

Given a division of an information space into objects, it is necessary to create reduced size object representations that will fit in the available graphical space. Rather than reduce the size of all of the objects equally as with murals, the objects are shrunk by some fixed amount. The final size should be large enough to maintain the visible appearance of the object, yet small enough to fit in the available space. For example, in the mural bar we developed for textual spaces, each line of text is reduced until it has a height of five pixels. The width remains proportional to the length of the line of text; thus, a user could see where longer and shorter lines are located within the text space.

Figure 4: The pile bar in use with a rolodex program. The piling effect adds an organizational dimension: in this example, the list ordering is alphabetical, the coloring is based on domain (gatech.edu is white), and the stacking is based on access time. Thus, the representation for the currently selected item (Greg Gee, at the gatech.edu domain) is highlighted in white and placed on top of the pile.

Since the reduced-size objects may no longer fit in the available space without overlapping, some bars may be partially or completely obscured. A heuristic is needed to describe the stacking order of the objects. Just as a good coloring scheme can highlight the locations of classes of items, a good stacking heuristic can enhance the visibility of items of interest to a user. Access time is one possible heuristic: much like for a pile of papers, the item that was accessed most recently would appear on top of those accessed previously. After a viewer accesses something in another portion of the pile, it would be easy to find the previously accessed item. The choice of heuristics is an interesting problem that is a topic of further research; for more information see [12].

As an example, consider a computerized rolodex containing names, phone numbers, and email addresses. Just as a rolodex on someone's desk might have recently accessed cards sticking out a bit, the computerized rolodex has the most recently accessed cards on top of the pile, and just as a rolodex might have colored labels or other highlights, the computerized rolodex can color entries based on company or location. Figure 4 shows what this application looks like.

3 Experiment: Evaluating the Navigation Bars

One might think that adding graphical information to an interface would always improve its ability to communicate with the user. However, additional information can sometimes clutter the interface and obscure that which was previously obvious. This experiment attempts to determine whether users can improve speed and maintain accuracy on a variety of tasks when using the mural bar and the pile bar.

One of the first decisions to be made is the choice of an information space. In choosing an information space, we wanted to select a space that is used in a wide variety of situations where scrollbars, mural bars, and pile bars each could have potential benefits. *Textual lists* are used in a wide variety of applications from directory listings to email lists to shopping lists. Unlike computer programs, lists can be interpreted quickly and require little domain knowledge. Unlike a text document, each line in a list is complete entity unto itself, yet there can be relationships between items in a list that are candidates for highlighting techniques. In particular, we chose to focus on lists of names such as those found in email programs or Usenix news readers. Repetition of names in these lists is to be expected as one often receives multiple emails or news articles from a person, and these repeated names can be leveraged in the highlighting schemes for mural and pile bars.

There were four tasks for each of the eight lists. The tasks to be performed on each list were, in order: 1) Estimate the size of the list, 2) Estimate the number of items from the list with a particular name, 3) Find three items from the list with a particular name, and 4) Find a cluster of three items (i.e., three consecutive items) from the list with a particular name. The order of the tasks and lists was the same for all groups, the only difference was the type of interface used to do the tasks (scrollbar, pile bar, or mural bar).

3.1 Implementation of the Navigation Bar

Our *navigation bar* is implemented as a single widget with three modes: scrollbar, mural bar, and pile bar. By using a single widget, we are able to maintain uniform appearance and functionality between the different modes. For practical purposes, we chose to implement our navigation bar in Tcl/Tk, a graphical scripting language with rapid prototyping capabilities. Not only did the Tcl/Tk language make it easy to incorporate the navigation bar into new and existing interfaces, but built-in timing and mouse action detection methods simplified the data collection process. More information about the availability and use of the navigation bar can be found in [15].

In the mural and pile bar modes of our navigation bar, list items are represented by horizontal lines. The lines are in the same order as the corresponding list entries. The length of each line is proportional to the length of its list entry, so a longer entry would be represented by a longer line. Colors correspond to groupings of emails from frequent senders. Repeated color patterns in the encodings can then reveal related items. It is our hope that participants can quickly identify related items by their similar encodings. Entries that do not belong to a specific category are shown with grey lines.

3.2 Method

Seventy-six undergraduate students participated in this experiment for class credit. All participants had 20/20 vision (possibly corrected) and none were color-blind. The experiment was conducted on Sun Sparcstation 2 workstations, each connected to a 15-inch monitor and an optical mouse. The time and position of all mouse clicks on the interface were recorded. The result of the click was noted as well; that is, whether the click caused an entry to be selected or caused the interface to scroll. The time at which each task was successfully completed was also noted.

The independent variable in the experiment was type of navigation bar used, thus there were three groups: mural bar ($n = 25$), pile bar ($n = 31$), and scrollbar ($n = 20$). The unequal group sizes were due to unanticipated scheduling problems. Dependent variables were time to do each task, the accuracy of each estimation task, and the technique used for each task.

Participants were run in small groups, one participant per computer. Participants first completed a questionnaire in which they provided background information relating to their major, GPA, and computer usage. The on-line portion of the experiment began with a series of introductory screens that described the experiment and gave participants warm-up tasks to ensure that they could use the system as designed.

Following the instructions and warm-up, participants began the experimental tasks on a series of lists. Each list contained between 34 and 195 names, which seems to be a reasonable estimate for the daily traffic of many email lists, Usenet news groups, and Web news wires. For each task, the participant first read a description of the task and then would click a "Begin" button when the participant was ready to begin the task. The time was measured until the participant finished the task, at which time a "Next" button was activated that allowed continuation to the next task.

The order of the four tasks was the same for all groups. For the first two tasks (estimating the list size and estimating the number with some label) it was stressed that a good time with some error was better than a poor time with little or no error. Participants were not given feedback on the accuracy of their estimates during the experiment. For the next two tasks (finding individual items and finding clusters of items) the entries to be selected were spaced so that the participant had to use the navigation bar to find them (they were not immediately visible in the listbox). The entries to be found were highlighted with some color in the pile bar and mural bar.

3.3 Expected Results

Consider the task of estimating the size of a list. Using a scrollbar, one can roughly calculate the total size by estimating the number of visible entries and the percentage of the trough occupied by the thumb and then multiplying to find the number of entries in the entire list. While the mural and pile bars provide more information that could be used in calculating the size, simple thumb-based calculation method may not be as obvious with the bars in the trough.

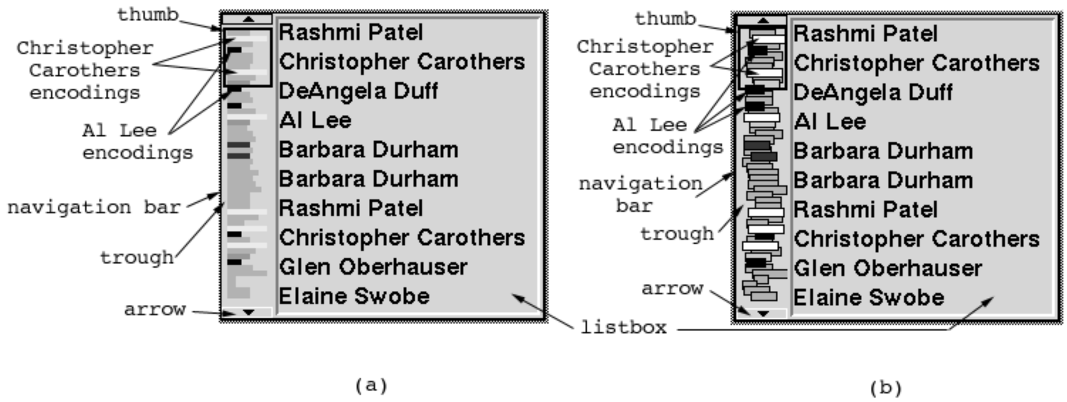

Figure 5: The mural bar (a) and pile bar (b) navigation bars with their accompanying listboxes. Each line of text is represented by a graphical bar. The bars work much like a standard scrollbar: the thumb slides up and down in the trough when a user clicks on the arrows or in the trough.

For the task of estimating the number of entries of some type, the scrollbar presents problems because it does not contain representations of the individual items. Thus, the user must scroll through the list and observe the contents of the listbox. The information mural and pile view should provide faster estimates since the color codings mean that no scrolling is required. We predicted that the loss of accuracy will not be significant.

For the two search tasks, the pile view and information mural should show better performance than the scrollbar as the encodings are always visible (though smaller or partially obscured) in these bars. We expect the pile view will produce better performance than the information mural when a user has to find individual items (Task 3) because the graphical representations will be larger, making them easier to see and select. However, when finding clusters of items, the information mural should be better because the multiple items of the same type will create a larger representation, while in the pile view the representations will overlap, thus not producing as large a visible target.

4 Results

The average number of correct Task 1 (estimating size of list) and Task 2 (estimating number of list items with a certain label) estimates for each group across the eight lists are presented in Table 1. For data analysis purposes, we considered an estimate to be correct if it was within 15 percent of the actual value. Table 1 also shows the average times for doing the tasks.

There was no significant difference among the three conditions on the accuracy of estimating list lengths (Task 1), $F(2,73) = 1.40$, $MSE = 4.54$, $p = .25$, or in estimating the number of list items with a certain label (Task 2), $F(2,73) = 0.78$, $MSE = 3.92$, $p = .46$. There was also no difference among the groups in the time to do Task 1, $F(2,70) = 1.62$, $MSE = 47.61$, $p = .21$ (this average does not include the time for estimating the size of list 7 since that list was considerably longer than the others). However, there was a difference among the groups in the time for estimating the number of list items with a certain label, $F(2,73) = 5.78$, $MSE = 79.42$, $p = .005$. Pairwise comparisons indicate that both the mural bar and pile bar conditions tended to be faster than the scrollbar condition (ps= .08 and .001 respectively; required $p = .05$ using Shaffer [20] sequential Bonferroni pairwise comparisons for providing a familywise α of 05 for multiple comparisons; see also [19]) while the first two groups did not differ ($p = .11$).

The average times for finding three items in the list with a certain label (Task 3) and finding a cluster of three items with a certain label (Task 4) are also shown in Table 1. There was a significant difference among the groups in the time for finding three list items with a certain label, $F(2,73) = 16.59$, $MSE = 59.17$, $p < .0001$. Pairwise comparisons indicate that both the mural bar and pile bar conditions were faster than the scrollbar condition ($p < .0001$ in both cases) while the first two groups did not differ. Similarly, there was a significant difference among the groups in the time for finding a cluster of three list items with a certain label, $F(2,73) = 4.11$, $MSE = 10.17$, $p = .02$. Pairwise comparisons indicate that both the mural bar and pile bar conditions were faster than the scrollbar condition (ps $= .04$ and $.006$ respectively) while the first two groups did not differ.

The data also revealed three techniques with which participants determined the solutions to the tasks. With the *look* technique, the participant did not move through the list but instead only visually scanned it. The *scroll* technique requires that the user clicked on the arrows above or below the trough to navigate the list. The *page* technique means the user clicked in the trough to move through the list a screenful of items at a time.

The results showed that the mural bar and pile bar participants were more likely to gain information by simply looking at the display, and for tasks where scrolling was necessary mural bar and pile bar participants moved through the lists by taking larger steps (paging rather than simply scrolling). The summary finding of a regression analysis showed that techniques

	Task 1 Time	Task 1 Correct	Task 2 Time	Task 2 Correct	Task 3 Time	Task 4 Time
mural bar	17.72	5.36	18.98	5.40	18.58	7.14
pile bar	15.95	5.03	15.09	5.90	17.60	6.61
scrollbar	19.51	6.05	23.76	6.10	29.54	9.18

Table 1: Average number of correct estimates (maximum 8) and task completion times (in seconds) for the tasks.

such as looking and paging were typically a strong predictor of task time, with clear advantages for longer lists. Table 2 summarizes the results.

The first task in Table 2 shows that the groups did not significantly differ in the frequencies of use of the various navigation techniques. That is, participants used any given technique with about the same frequency regardless of the their navigation bar type. We had worried that the additional information in the trough of the mural and pile bars would obfuscate the simplicity of determining the list size using the thumb-to-trough ratio; it appears that this was not the case.

The second task in Table 2 shows that the mural bar and pile bar groups used the look technique more often than the scrollbar group, $F(2,73) = 9.63$, $MSE = 7.33$, $p = .0002$. This result was to be expected since it is impossible to make a reasonable estimate of the list contents in the scrollbar condition simply by looking at the visible contents and the appearance of the scrollbar. This result shows that participants do indeed use the encoded information effectively, that is, they do not simply scroll through and count the number of entries.

The last two tasks required participants to find items with the same label. Since these tasks required participants to use the navigation bars, by default no one could use the look technique. Interestingly, we found that mural bar and pile bar participants had a greater likelihood of using the page technique for both Tasks 3 and 4 compared to scrollbar participants, $F(2,73) = 13.20$, $MSE = 6.00$, $p < .0001$; $F(2,73) = 3.83$, $MSE = 5.32$, $p = .03$. In other words, mural bar and pile bar participants were more likely than scrollbar participants to move through the lists by taking larger steps, and scrollbar participants were more likely to move through the lists by taking smaller steps.

The tendency of the mural bar and pile bar participants to use more efficient techniques for finding information in a list is consistent with the finding that they did most of the tasks more quickly than scrollbar participants. A regression analysis was carried out on the time to do the various tasks as a function of various predictors such as navigation bar used, technique used for that task, and other factors (details available from the authors). The summary finding was that technique was typically a strong predictor of tasks time. In addition, considering the technique interacted with list length suggested that the advantage of an efficient technique grew stronger for longer lists.

5 Discussion

The *estimation* accuracy and times showed equal accuracy among the different interfaces but a tendency towards faster performance for the mural and pile bar groups relative to the scrollbar group (at least for one of the estimation tasks). The *search* results clearly indicate faster performance by the mural and pile bar groups compared to the scrollbar group. Performance by the mural and pile bar groups did not reliably differ from each other. Taken as a whole these results suggest that graphical interfaces that attempt to encode certain features of list entries can improve users' accuracy and speed as they do typical tasks such as trying to get a sense of the number of related items in a list or the location of certain related items.

The search technique results suggest that the mural and pile bar interfaces encouraged users to move through the list more efficiently. Obviously on some tasks (i.e., estimating list length and number of entries with a particular label) there was no need to physically move through the list since all entries were graphically represented in the trough. However, for tasks that did require moving through the list, mural and pile bar users were more likely to move in larger units using a paging approach while scrollbar users were more likely to move item by item. This may be because mural and pile bar users could see the graphical representations of the items they wished to find and thus, could take larger leaps towards those items. Scrollbar users, on the other hand, had to move more blindly through the lists searching for the desired items. Thus, they may have been inclined to be more cautious. This difference in efficiency grew larger as the list length increased. This suggests that the benefits of the graphical encodings used in the information mural and pile view were not eliminated by increased clutter as the lists grew longer.

6 Related Work

Numerous visualization techniques have used the shrinking of an information space to assist in communication and navigation. The Pad [16] and Pad++ [4] interfaces address the usefulness of zooming for several information spaces, including text documents and hypertext spaces. The fisheye technique [9] smoothly incorporates focus and context by magnifying a portion of a display. It has been used in a variety of systems, including the Perspective Wall [13] and the Information Visualizer [6]. Bell Laboratories' SeeSoft tools use a reduced-size pixel representation of program source code to show data such as code version history, program slices, execution hot spots, and static properties

Task 1 (estimate size)

	Look	Page	Scroll	Scroll+Page
mural bar	1.56	3.24	0.36	2.84
pile bar	2.61	2.39	0.97	2.03
scrollbar	2.60	2.25	0.65	2.50

Task 2 (estimate count)

	Look	Page	Scroll	Scroll+Page
mural bar	4.16	1.92	1.16	0.76
pile bar	5.36	1.71	0.71	0.23
scrollbar	1.95	2.25	2.05	1.75

Task 3 (find three items)

	Look	Page	Scroll	Scroll+Page
mural bar	0	6.40	0.76	0.84
pile bar	0	6.45	0.84	0.71
scrollbar	0	3.15	2.30	2.55

Task 4 (find cluster of items)

	Look	Page	Scroll	Scroll+Page
mural bar	0	7.04	0.60	0.36
pile bar	0	7.16	0.61	0.23
scrollbar	0	5.45	2.20	0.35

Table 2: Average number of times (out of a possible 8) that participants used various techniques to determine solutions for each task.

of code [3]. Projects at the University of Maryland's HCI Lab such as LifeLines [17] and FilmFinder [2] use a variety of zoomable overview techniques to help the user identify and filter information.

The techniques used in these interfaces integrate information about the information space with the information currently of interest throughout the entire display area. While this can be advantageous in understanding the contents of the space, often not enough of the display area is dedicated to showing details at a workable level. For example, when authoring a piece of code, while it can be helpful to understand the relative position of the current lines of interest to the entire program, it is often more useful to be able to see the surrounding lines. Since the scrollbar occupies only a small percentage of the screen, we chose to focus on techniques that can be integrated with it.

Several systems have leveraged the scrollbar metaphor to communicate various types of information. Alphasliders support selection from long lists of information using multigranular selection [1]. An alphabetic index is provided next to the slider trough to show where specific ranges of a text space can be found, and the thumb provides different granularities for scrolling. Data visualization sliders provide a scrollbar-like interface for specifying numeric values [8]. The space inside the sliders contains a color-coded barplot or density plot representing the range of numeric values. Value bars represent a list entry using thin colored stripes next to the scrollbar [7]. The color and size of a stripe represents the goodness of the attribute for a particular entry. The attribute-mapped scrollbars of read wear and edit wear map data about editing changes to marks on a scrollbar [10]. Masui's LensBar uses a highlighted line technique to show filtering and zooming results for dictionaries, email lists, and program files [14].

The previously described systems provided interesting insight into the understanding and navigation of information spaces, but they tended to suffer from one or more drawbacks. One drawback stems from the complexity of many of the communication and navigation methods. If it is not immediately obvious how an interface works, it can be difficult to establish a significant user population. Second, the systems typically targeted a specific information space, and extensions to other spaces are often not obvious. More general navigation techniques can be applied to novel information spaces with little programmer effort. Finally, few rigorous empirical studies have been conducted to study whether these techniques are helpful. Empirical testing is important in the comparison of two or more design alternatives for typical tasks.

The work described in this paper addresses each of these drawbacks. By leveraging the familiar scrollbar device, the mural and pile bars allow users familiar with them to build upon an existing mental model and hopefully shorten the time required to learn the interface. Unlike many of the applications described previously, our navigation bars provide a balance of flexibility and power that shows potential utility for a wide variety of numerical, textual, and graphical browsing and searching situations. Finally, our usability study shows that the techniques we describe have potential utility for several typical scrollbar tasks.

7 Conclusions

This paper introduced the mural and pile bars, devices with which different properties of information spaces can be communicated using graphical representations. Programming environments and visual programming languages could benefit from the overview capability provided by these augmented scrollbars. In addition, the new landscape of online information provides many information resources where the mural and pile bars could prove useful: email programs, Web pages, and online news articles just to name a few. The mural and pile bars provide a low-effort way to identify features of an information space that are of interest, and they provide a familiar method for navigating to that portion of the space.

The results of our experiment are promising for the use of navigation bars in new environments. The participants in our experiment took advantage of the increased amount of information while still using the familiar navigation methods that are present in a standard scrollbar. Furthermore, the evidence suggests that the benefits of mural and pile bars grow even larger as the size of the information space increases, though almost certainly some upper limit exists.

Future work will examine whether and when the pile bar and mural bar show better performance. We are considering different types of information spaces and spaces with larger numbers of items in them. We expect that finding items in larger lists with fewer highlighted items will show better performance with the pile view. On the other hand, the information mural should show better performance with denser clusters of items since they appear larger in the mural while they merely overlap in the pile. As it is important to study the impact of these navigation devices on real world applications, we have integrated them into several applications and hope to include them in various visual programming languages and environments. Based on both empirical and anecdotal evidence that we are collecting, we plan to develop recommendations for when certain highlighting and stacking heuristics are most useful and the best uses for each type of navigation bar.

Acknowledgments

The authors would like to thank John Stasko, Don Allison, and Lisa Wenner for their helpful comments on various drafts of this paper. We would also like to thank the many participants for their suggestions and feedback. This work has been partially supported by a grant from the USENIX Association.

References

[1] C. Ahlberg and B. Shneiderman. Alphaslider: A compact and rapid selector. In *Proceedings of CHI '94*, pages 365–71, Boston, MA, April 1994.

[2] C. Ahlberg and B. Shneiderman. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In *Proceedings of CHI '94*, pages 313–17, Boston, MA, April 1994.

[3] T. A. Ball and S. G. Eick. Software visualization in the large. *IEEE Computer*, 29(4):33–43, 1996.

[4] B. B. Bederson and J. D. Hollan. Pad++: A zooming graphical interface for exploring alternative interface physics. In *Proceedings of UIST '94*, pages 17–26, Marina del Rey, CA, November 1994.

[5] M. M. Burnett, M. J. Baker, C. Bohus, P. Carlson, S. Yang, and P. van Zee. Scaling up visual programming languages. *IEEE Computer*, 28(3), March 1995.

[6] S. K. Card, G. G. Robertson, and J. D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of CHI '91*, pages 181–8, New Orleans, LA, 1991.

[7] R. Chimera. Value bars: An information visualization and navigation tool for multimedia listings. In *Proceedings of CHI '92*, pages 293–294, Monterey, CA, 1992.

[8] S. G. Eick. Data visualization sliders. In *Proceedings of UIST '94*, pages 119–120, Marina del Rey, CA, November 1994.

[9] G. W. Furnas. Generalized fisheye views. In *Proceedings of CHI '86*, pages 16–23, Boston, MA, April 1986.

[10] W. C. Hill and J. D. Hollan. History-enriched digital objects: Prototypes and policy issues. *The Information Society*, 10(2), April 1994.

[11] D. Jerding and J. Stasko. The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3), 1998.

[12] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2), April 1986.

[13] J. D. Mackinlay, G. G. Robertson, and S. K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of CHI '91*, pages 173–9, New Orleans, LA, 1991.

[14] T. Masui. Lensbar – visualization for browsing and filtering large lists of data. In *Proceedings of InfoVis '98*, Raleigh, NC, October 1998.

[15] D. S. McCrickard. Enhancing interfaces with animated widgets. In *Proceedings of Tcl/Tk '98*, page 201, San Diego, CA, September 1998.

[16] K. Perlin and D. Fox. Pad: An alternative approach to the computer interface. In *Proceedings of SIGGRAPH '93*, pages 57–64, August 1993.

[17] C. Plaisant, B. Milash, A. Rose, S. Widoff, and B. Shneiderman. Lifelines: Visualizing personal histories. In *Proceedings of CHI '96*, pages 221–27, Vancouver, BC, April 1996.

[18] D. E. Rose, R. Mander, T. Oren, D. B. Ponceleon, G. Salomon, and Y. Y. Wong. Content awareness in a file system interface: Implementing the 'pile' metaphor for organizing information. In *Proceedings of SIGIR '93*, pages 260–269, Pittsburgh, PA, 1993.

[19] M. A. Seaman and R. C. Serlin. New developments in pairwise multiple comparisons: Some powerful and practical procedures. *Psychological Bulletin*, 110(3):577–586, 1991.

[20] J. P. Shaffer. Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, 81:826–831, 1986.