The Usability of Stacked-Block Overviews for Retrieval

Carl Gutwin, Department of Computer Science, University of Saskatchewan, gutwin@cs.usask.ca

HCI-TR-05-02

Abstract

Stacked-block overviews represent a large data set in a compact space by placing items in a closely-packed grid. As awareness displays, they provide a bird's-eye view of state, activity, and change in a data set. However, stacked-block overviews have a distinct disadvantage when it comes to retrieval, because they provide few clues to the identity of objects other than spatial position. To investigate the usability of stacked-block overviews, we carried out two studies in which participants retrieved items from either an overview or a standard scrolling list. Retrieval performance with the stacked-block overview was much better than we initially expected (and at its best was significantly better than the scrolling list), and several people strongly preferred the overview. However, the stacked-block representation was unusable for nearly a quarter of the participants, and even those who succeeded with it required extensive training to reach a high level of performance. Despite these problems, our results show that stacked-block overviews can in many cases be used as more than just awareness displays.

Key words: Overviews, stacked-block overviews, scrolling lists, retrieval performance, spatial memory.

1 Introduction

Graphical overviews are a space-efficient way to show information about a large number of data items. They can show how many items there are, the items' states and attributes, and changes that occur over time. A *stacked-block overview* is a particular type of overview that represents items as a compact grid (see Figures 1,2, and 5). Although the stacked approach limits what can be shown about relationships between data elements, it is the most space-efficient way to display a large number of individual items. The small size of the stacked overview is valuable when the overview is part of a larger workspace such as an integrated development environment (IDE).

We initially developed stacked-block overviews to help people maintain awareness of group activity in a software project. In Figure 1, the stacked representation shows the packages and files in the project, organized by creation date. The overview indicates who is working where (by colour), and indicates which files have changed recently (by highlight) [6]. Software projects are a good match for this representation because new files can be added to the end of the stack without rearranging existing items.

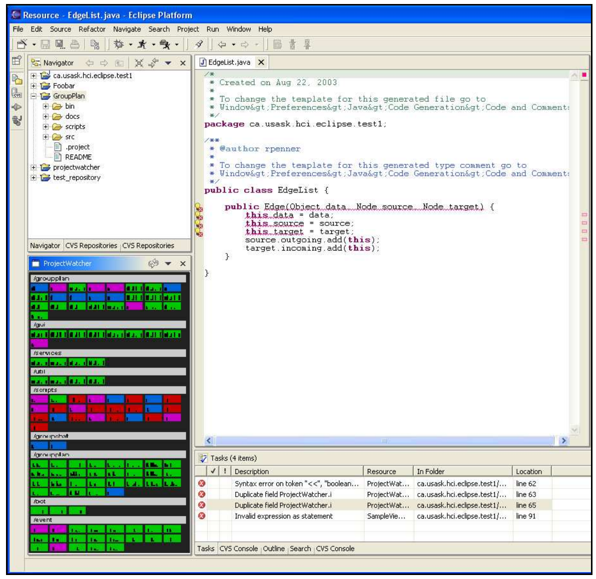

Figure 1. A stacked overview in the Eclipse IDE [6]; the stacked overview is shown at lower left, and a traditional list-based navigation window shown upper left.

In an IDE, stacked-block overview are useful as awareness information displays. However, it seemed inefficient to use the overview to observe that a file had changed, but then move to a list tool in another window to open or manipulate the file. We thus began to consider the possibility of retrieving items from the stacked-block overviews. We initially assumed that it would be very difficult for users to learn the locations of particular items, since the stacked representation provides little identifying information about individual items; however, we observed several instances of people doing exactly this for common files in the display.

To further investigate how the performance of stacked-block overviews compares to traditional sorted lists, we carried out two studies that tested people's ability to find items in both displays. We were interested in how long it would take people to learn the locations of a reasonable number of items in a stacked-block overview, and how performance in retrieving those items would compare with retrieval from a list.

The studies asked participants to learn the locations of 25 targets, and for most of the participants, it took about twelve trials before they were able to remember the locations consistently. Once items were learned, retrieval times quickly dropped below that of retrieval from a list (at best, retrieval times were half that of the list). In addition, most participants preferred the overview to the list. However, we also found that for several participants in the second study, the stacked-block representation was not good for retrieval even after extensive training.

Despite these limitations, our results still suggest that designers should consider retrieval capabilities for stacked-block overviews (and possibly from other awareness displays). In situations where people have long-term exposure to a set of items (such as in software projects), adding retrieval capabilities to these displays may be beneficial for many users.

2 Background

2.1 Overviews and Stacked-Block Overviews

Overviews allow a user to see all the items in a data space at once, although the data is shown in reduced detail. At a high level, they allow the user to see overall patterns in the data and how the space is organized. At a lower level, they show information about each of the items in the data set, and although there is limited representational space for each item, it is still possible to show the values of several attributes for each one using colour, border, shape, or other visual variables [2].

One of the main organizational factors in a two-dimensional overview is what the X and Y dimensions represent. A stacked overview, however, does not use X and Y to represent values of the data. Instead, items are stacked sequentially into a grid; items may be grouped (see Figure 5), but the sequential organization means that only one ordinal variable can be encoded. For example, the stacked overview in Figure 1 shows the packages and files in a software project; the items are organized sequentially by creation date.

There are limits to the number of items a stacked overview can display, particularly if it is to be an adjunct display as in Figure 1. Depending on the size of the individual blocks, several hundred items can be shown fairly easily. This allows representation of many medium-sized data spaces – for example, of the top 18 projects at the Sourceforge directory (sourceforge.com), two-thirds had fewer than 500 files, and therefore would fit comfortably in an overview.

Stacked-block overviews are a type of space-filling representation. The most well-known example of this category is the treemap [12]; the main difference between treemaps and stacked-block representations is that treemaps change the size and position of items to fill the available space, whereas stacked overviews keep items fixed in place, in order to assist spatial memory.

2.2 Spatial Memory in Interactive Systems

Spatial object location memory is knowledge of where things are located in a space [1,5]. Spatial knowledge in two-dimensional spaces is built up primarily through interaction; that is, people remember locations after having had experience with that location [4]. People may remember particular items based on landmarks in the space, or with more experience, may be able to maintain a more complete 'mental map' in which they can remember and find many different objects very quickly [13,16].

Studies have shown that although spatial abilities can vary widely [14], people are capable of using object location memory to remember large numbers of items [3]. For example, Robertson and colleagues tested a spatial memory technique (the Data Mountain) in which people placed thumbnails of web pages on a simulated inclined plane [11]. Once 100 pages were placed, participants carried out a number of find-and-select retrieval tasks. The study found that retrieval was significantly faster with the spatial technique than with a standard bookmarking system. In addition, the memory of where items were placed persisted over a long time: participants who returned six months later were able to retrieve items at the same level of performance, with only brief retraining [3]. (The Data Mountain involved self-placement of objects, whereas a stacked-block overview uses a fixed placement system; still, this research still shows that a large number of locations can be remembered.)

However, the items in these studies contained symbolic information (thumbnails and names) as well as spatial position, and when only location is used as the retrieval strategy, spatial memory fares less well. An early study by Jones and Dumais [8] showed that retrieval of items using location only was slower and less accurate than when items were represented by name.

In the studies described below, we test a display where the primary retrieval strategy involves memory of locations and landmarks (the stacked-block overview), against one where the primary strategy is symbolic (the sorted list).

2.3 Revisitation

Learning spatial locations is a function of experience with the items in the data space [4]. Therefore, the degree to which a user will be able to build a mental map is related to the amount of revisitation in the task. Different situations have different revisitation patterns, but in most information tasks, users will go back to particular items repeatedly.

One well-known example of revisitation is navigation on the WWW: Tauscher and Greenberg [15] found that more than half of pages seen were revisits, and that revisitation occurs mainly to the last few pages visited – the last ten pages seen cover about 85% of revisits.

Software development projects also see strong revisitation (e.g. [10]). For example, the Apache httpd project (www.apache.org) shows frequent revisitation. One-third of the files were committed eight times or more by a particular author (and the true revisitation level is higher than this amount since developers likely open a file several times for each commit). Second, if each author remembers the spatial locations of the 25 most frequently-used files, the likelihood that the next file accessed would be in that set is more than 62%. That is, a working set of the size used in our experiments covers almost two-thirds of the file accesses for any author. The revisitation that we saw in software projects convinced us that there was an opportunity for stacked overviews to be used for retrieval.

3 Study One: Remembering 25 Items

Our first study looked at whether people would be able to remember the locations of twenty-five items in a stacked-block overview, and how much training would be required before retrieval performance equaled that of a sorted list.

3.1 Methods – study 1

Eight participants (6 men and 2 women) were recruited from a local university. Participants ranged in age from 21 to 32 years (mean of 25 years), and were very familiar with mouse-and-windows applications (more than 12 hrs/wk).

Custom software was built to display and test the stacked-block overview and the traditional list (see Figure 2). The items represented in the display were documents titled with city names (these were considered more understandable than filenames for those participants not in computer science). The study was run on a P4 Windows system with a 1600x1200 screen.

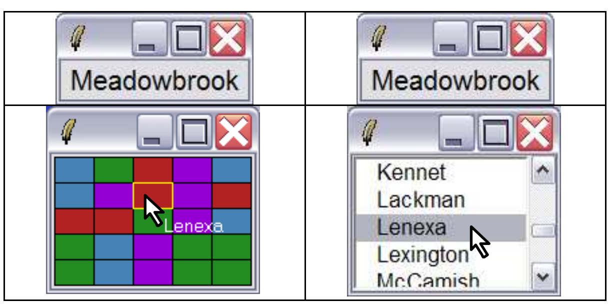

Figure 2. Stacked overview (left) and list (right); note that participants saw only one representation at a time. Prompt window with item to find is above each display.

The overview. The stacked-block overview represented the 25 items as rectangular blocks in a 5x5 grid. The items were randomly assigned to positions in the grid. Whenever the mouse cursor was over one of the blocks, that block would be highlighted with a yellow border, and the item's name would be displayed below the cursor (see Figure 2). The blocks were filled arbitrarily with one of four colours, to simulate the representation of some other variable, such as authorship, that would be present in a real overview (this encoding was not discussed with participants). These colours, however, were the only additional visual variable that was used: even though we could have made the items more memorable with icons, patterns, or even a few characters of the item's name, we wanted to test the representation without these additional memory aids.

The sorted list. The list representation was a standard listbox with items sorted alphabetically, and with a scrollbar at the right of the widget. The list was configured such that each list item was approximately the same height as one of the blocks in the stacked overview, and such that the two windows were exactly the same size. The list could be scrolled either by pressing the arrow buttons at the ends of the scrollbar, or by dragging the scroll thumb (which allowed scrolling at a variable speed controlled by the mouse).

Task

Participants carried out a series of retrieval tasks using the two different representations. In each trial, participants had to find and click on the particular item whose name appeared in a prompt window (e.g. 'Meadowbrook' in Figure 2). Participants had to find the correct item before the system would proceed. Participants were not given any introduction to the locations of the items; in the first trials, therefore, they had to find the item using a scanning search with the mouse.

The target items were chosen without replacement from within the set of all 25. After all items in the set had been used, the set was refilled (participants were allowed to rest during these transitions). Participants saw only one representation at a time, either the list or the stacked overview.

Procedure and Study Design

Participants were randomly assigned to one of two order groups (list first or stacked overview first), and then carried out training and test trials with each representation. Three hundred training trials were used with the stacked overview (each item retrieved 12 times), and one hundred with the list (each item four times). These numbers were chosen during pilots as the average amount of training needed for performance to level off.

We recognize that different amounts of training introduce a confound into the study. However, from our observations, we do not believe that performance with the list would have improved much with added training: all of our participants were already experts with both the list widget itself and the idea of searching through an alphabetized list, and all participants appeared to be using a symbolic search strategy that did not improve much with practice. In addition, one hundred trials was the maximum number that our participants would stand for: repeated retrievals from the list were seen as far more tedious and annoying than retrievals from the overview.

After training was complete, participants carried out 100 test trials with each representation (each item was retrieved four times). After all trials had been completed, participants filled out a questionnaire asking them whether they felt that they were faster or more accurate with either representation, and which display they preferred overall.

The study used a 1x2 within-participants factorial design. The main factor was representation type (list or stacked overview). The system recorded completion time and errors for each selection trial.

3.2 Results – Study 1

Learning Curve

One of our main questions was how quickly people learned the items in the stacked-block representation. Figure 3 shows mean completion times for all trials (including training). List performance is relatively constant from the outset; retrieval performance in the overview starts at approximately twice that of the list but equals it by about 200 trials (i.e., after having seen each item eight times).

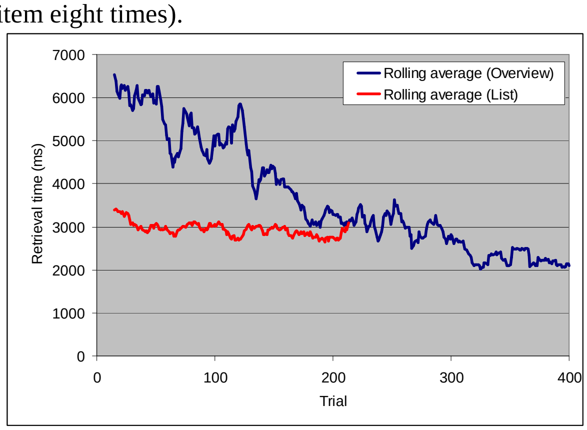

Figure 3. Training curve (rolling average of 15 trials) of mean retrieval times for stacked overview and list representations.

Retrieval time

Figure 3 shows that any performance comparison between the stacked-block overview and the sorted list is highly depending on amount of training. It is clear that for the first 200 trials, the list is significantly faster than the overview. However, it was less clear whether the final performance of the stacked overview would be better than the list, so we carried out an analysis of variance for the final 100 trials. This ANOVA showed that there was an effect ($F_{1,7}$=6.25, $p<0.05$); for this data, retrieval times from the stacked overview were approximately half a second less than from the list.

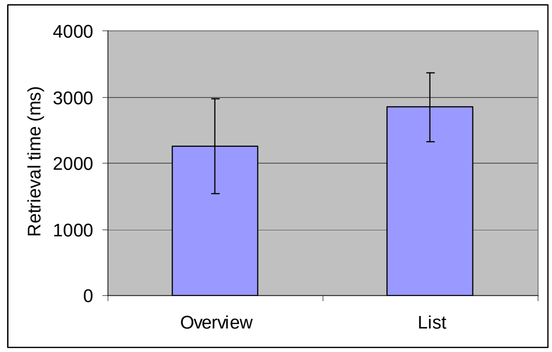

Figure 4. Mean retrieval times (final 100 trials only).

Errors

Since participants received feedback about which item they were selecting (the label in the list, and the tooltip in the overview), mistakes were relatively rare and were generally simple targeting errors. Participants made 0.017 errors per trial with the stacked overview (one in about sixty), and 0.014 errors per trial with the list (one in seventy). No effect of representation type was found on error rate ($F_{1,7}$=0.465, p=0.52).

Preferences

Preference data was gathered from the post-test questionnaires. Six of the eight participants stated that they preferred the stacked overview representation. In particular, participants who started with the overview and then moved to the list made several negative comments about having to use the list representation. One participant preferred the list, however, stating that it was too difficult to remember the items in the overview.

3.3 Discussion – Study 1

The first study shows that people can successfully remember a fairly large set of item locations, even without symbolic labels or landmarks (other than the colours of the items themselves). However, although the performance of the stacked-block representation eventually matched the list, it is clear that the amount of training required to become proficient with the overview is much larger. People needed to see each item about eight times before they could remember it reliably; in contrast, no training was required to use the list.

Since the amount of data in this study space was less than would be seen in many real-world applications, we decided to look at an expanded version of the task, described below.

4 Study Two: Remembering 25 Items From 250

The goal of the second study was to determine whether people could learn a set of items within a much larger set of possible distracters. In this study, we used a data set of 250 items, from which the participants had to learn and remember 25. Items were grouped into arbitrary groups of 25 to simulate a two-level hierarchy, and groups were given arbitrary names.

4.1 Methods – Study 2

Twelve participants (six men and six women) were recruited from a local university. Seven were majors in Computer Science, and five in Psychology. Ages ranged from 20 to 38 years (mean of 27). As discussed below, only eight participants (four men, four women) were eventually used in the analysis.

Apparatus was similar to that used in the first study, except that the list and overview representations were expanded to 250 items. Figure 5 shows the two representations: in the stacked overview, items were organized into ten sets of 25 blocks, all visible on the screen. The list was organized similarly, although only about 60 items were visible at once. As in the first study, the name of the item appeared when the mouse cursor was over the item's block. Items in each block of 25 were randomly assigned to a grid location; however, the named groups were organized alphabetically. In the list representation, both the groups and the items within the group were organized alphabetically.

Training

Training was modified from the first study since participants now had to learn 25 items from within the entire set, and it would not be possible for them to search for each item during training. Therefore, the 25 target items were always highlighted during training (Figure 5). Highlighting was turned off during testing.

Participants carried out 300 training trials with the overview (each of the 25 items was retrieved 12 times), and 100 with the list as described above. After this training, participants were asked to find each of the items once, to see whether they had learned the location. All subjects were able to find the items in the list; however, for four of the twelve participants, 300 training trials were not sufficient for the overview. Data from these participants was not used in the analysis. This large number of exclusions presents a problem for this study. However, we present the analysis on the assumption that even though the overview display may not be useful for all users, it could be effective for a substantial number. The reason for the differences in our subject population is not clear. Individual differences could account for the variation, although none of academic major, programming experience, gaming experience, or gender seemed to show any patterns. We did not, however, give participants a spatial memory or spatial abilities test.

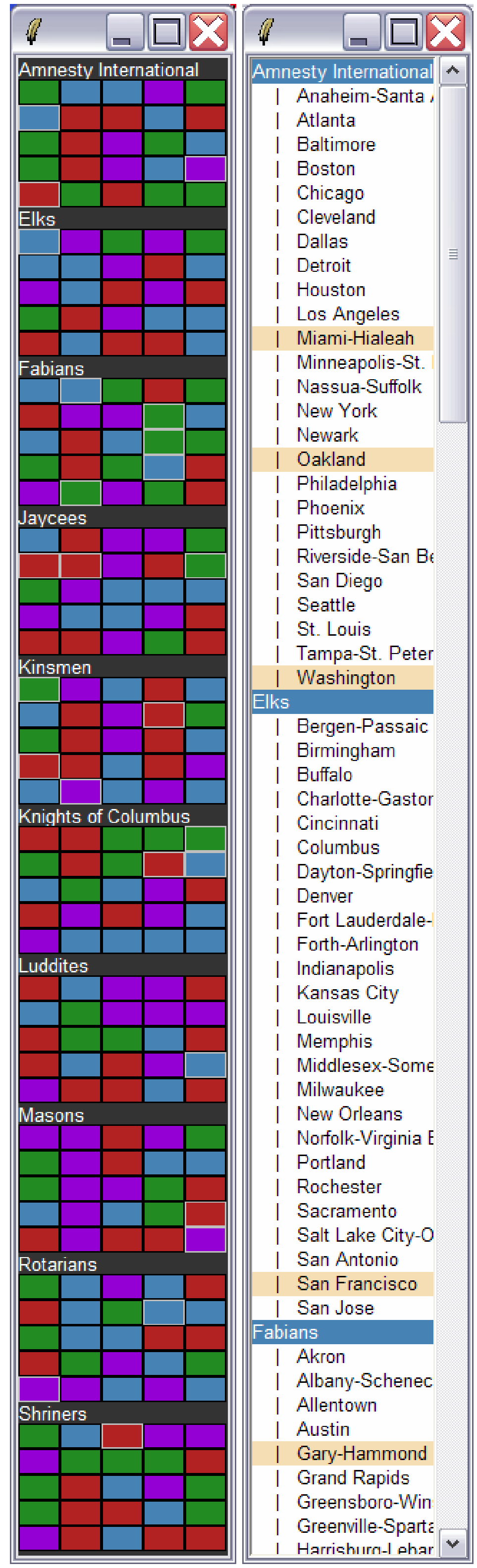

Figure 5. 250-item versions of the stacked overview and sorted list, showing highlighting used during training.

Procedure and Design

The study procedure was also similar to that described above. Participants were placed into an order group and then carried out training and test trials with each representation. After all trials were complete, participants completed a questionnaire asking them about their experiences and preferences.

The second study also used a 1x2 within-participants design, and again the factor was representation type (list or stacked overview). The system recorded completion time for each selection trial. With 100 test trials, two representations, and eight participants who completed the session, there was a total of 1600 data points collected.

4.2 Results – Study 2

Learning Curve

Figure 7 shows mean completion times for all trials including training. List performance is more variable than in the first experiment, but once again does not improve dramatically with increased experience (although there is a downward trend that we plan to investigate further in future work). In contrast, retrieval performance in the overview is quite poor when participants are first seeing each item, but improves rapidly, and passes the list by the time items have been seen four times. Note that the removal of the highlight on the target items does not appear hinder performance – performance after trial 300 is very similar to performance before that point.

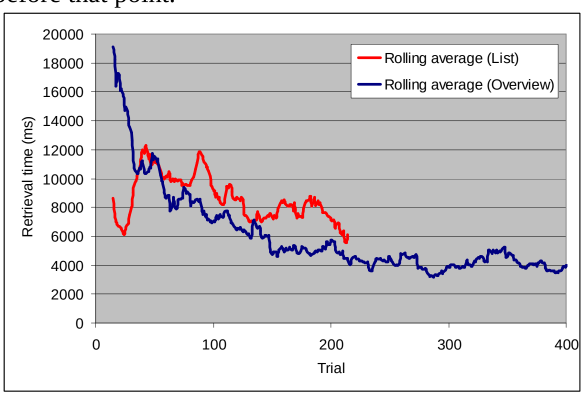

Figure 7. Learning curve (rolling average of 15 trials) of mean retrieval times for the stacked overview and list representations. Note that only the last 100 trials were the ones used for the comparison.

Retrieval time

Again, differences between the two representations depends on how much training is allowed for the stacked-block overview. In the first few trials, the overview is almost ten seconds slower than the list, but after training the overview is faster. For the eight participants who were used in the analysis, there was a significant difference between the two representations (once training was complete). Analysis of variance showed a significant main effect of representation type on completion time ($F_{1,7}$=14.10, p<0.01). As shown in Figure 6, retrieval from the stacked-block overview over the final 100 trials was four seconds less than from the list.

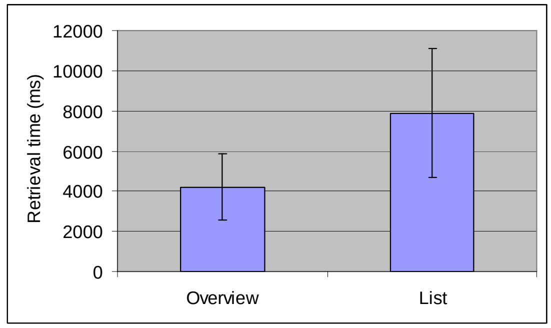

Figure 6. Mean completion time for overview and list, 250-item version of display.

Errors

Errors were also infrequent in the second study, although there were more erroneous clicks with the list than with the overview. Participants made 0.02 errors per trial with the stacked overview (one in fifty), and 0.14 errors per trial with the list (one in seven). No effect of representation type was found on error rate ($F_{1,7}$=4.13, p=0.09).

Preferences

There were marked differences in preference, and people's opinions were felt strongly. Not surprisingly, those participants for whom the stacked-block representation did not work preferred the list representation. Several people found the overview frustrating; others simply felt that it did not work for them and so was time-consuming and error-prone.

In contrast, all of the participants who were used in the analysis preferred the overview to the list. This was particularly evident in those who worked with the overview first, and then the list – during the list trials, we noted disparaging comments about the list and several joking requests to go back to the stacked overview.

5 Discussion

Our overall goal in these studies was to investigate whether stacked-block overviews could be used for retrieval as well as for awareness, and if so, how much training would be required for people to remember a reasonable set of items. Our results are mixed: on the one hand, people liked the overviews and performed better with them after training; on the other, extensive training was required, and even then, the stacked-block representation was not usable for all participants.

Below we consider what factors in the stacked-block overview led to its (eventual) strong performance, look at the issue of training and ways that training might be accomplished, and outline avenues for future work in the area.

5.1 Advantages of the stacked representation

There are several reasons why retrieval from the stacked-block overview could be fast, once training was completed. The main reason is that once the item locations are known, fewer actions are required in the interface to get to the target. Even if the user knows where to look in the list, they must still grab the scroll thumb, move to the correct location, and then target the item. Thus, the overview has one constant-time advantage over the list, and one variable advantage depending on how well the locations are memorized. The list, in contrast, has a guaranteed backup strategy (alphabetic search) that limits overall search time even when locations are not known.

Overall, it appeared that the stacked-block overview succeeded because it was more amenable to practice and learning than was the list. Although this is perhaps not a surprising result (with enough practice, people can become good at anything), it is interesting to note that people did not seem to use location nearly as much with the list representation, and continued to use an alphabetic search strategy throughout. In addition, it is useful to determine that a reasonable number of items can be recalled in the overview, and that performance can surpass the standard solution under certain conditions.

5.2 The issue of training

The stacked-block representation only works well if users have extensive practice. Since people are not likely to engage in explicit practice as they did in our studies, this appears to be a clear limitation to the usefulness of the technique. However, there are certain real-world task situations where people work with a set of data objects over a long term, and where it is possible that training will occur as a consequence of other activities. Software projects, the original motivation for stacked-block overviews, are a situation where this could occur: in many projects, files and packages are relatively stable and revisitation is common.

The key to implicit training is that the stacked overview is still useful a an awareness display, even if it is not used for retrieval. This means that it will be on the user's screen, even if they have not yet learned any object locations. Once the display has a foothold on the screen, several possibilities exist for helping users to remember where things are.

First, users will naturally inspect various items in the display (e.g., after a change to a file has been indicated), and these experiences will provide an initial exposure to the item locations. Second, the display could mirror the activities undertaken through other parts of the interface. For example, when a user opens a file using a traditional file browser, the stacked overview can highlight the item in question to introduce or reinforce its location. Third, read-wear techniques [7] could be used to help users remember which items in the overview they have accessed recently or frequently.

Although these strategies do not guarantee that users will learn the item locations, they are likely to be effective in situations where revisitation is common. There is a natural fit in these cases, since frequently-used items are more easily remembered, and remembered locations are more likely to be used.

5.3 Future work

Although our results suggest that retrieval from overviews can be useful for some people, there are a number of issues that should be considered further. The first task is to look more closely at why the representation did not work well for such a large minority of the participants. A second issue concerns the realism of the retrieval task. Our next project will be to look at more realistic usage patterns and the effects of these on retrieval from the overview. We are particularly interested in determining whether it is more common for people to maintain a single working set over a long period (e.g., return regularly to the same set of files over an entire project), or whether the working set of items slowly changes over time. In addition, real-world data can also involve gradual changes to the other displayed attributes of the items (e.g. most recent author). We need to determine if and how these changes affect performance with the stacked-block overview.

Since people will want to remember different numbers of items in different tasks, we also want to explore the tradeoff between the number of items that the user remembers, and the amount of training. For example, can a person remember a few things with only a few visits? Similarly, if a user will be working with a data set over a longer term, will they be able to continually increase the number of item locations that they remember? Related to this is the issue of whether users can maintain more than one spatial mapping; that is, whether they could be able to use multiple stacked overviews simultaneously in different contexts.

Finally, there are several ways that the stacked overview display could be improved to provide better retrieval performance. A simple addition would be to add more information to each block; as mentioned above, icons, patterns, or the first few letters of the item's name could make items memorable and help to differentiate between nearby items. A larger change would be to allow users to arrange the items themselves. For example, users could arrange files within a package area to match their own conception of that module. The work of Czerwinski and colleagues [3] suggests that this change could increase the number of items that users are able to remember. The only drawback is that user effort is required to create the new arrangement.

6 Conclusions

Stacked-block overviews provide a compact representation of a data space, but require that people remember the location of each item. To investigate whether overviews should include retrieval capabilities, we carried out two studies to compare retrieval performance in a stacked-block overview and a sorted list. We found that once the items were learned, the overview was significantly faster than the list; however, extensive training was required, and the overview did not work at all for several participants. These problems mean that retrieval from stacked-block overviews will only be effective in situations where users spend a lot of time with a particular set of items, and where they revisit items often. Nevertheless, in settings such as software development, retrieval from awareness overviews could provide a useful shortcut for experts.

References

[1] Baddeley, A. D., *Working Memory*. New York: Oxford Univ. Press, 1986.

[2] Card, S., Mackinley, J. and Shneiderman, B., eds., *Readings in Information Visualization*, San Francisco: Morgan Kaufmann, 1999.

[3] Czerwinski, M., van Dantzich, M., Robertson, G., and Hoffman, H., The Contribution of Thumbnail Image, Mouse-over Text and Spatial Location Memory to Web Page Retrieval in 3D Viewing, *Proc. IFIP INTERACT 1999*, 163-170.

[4] Darken, R., and Sibert, J., Wayfinding in Large-scale Virtual Environments, *Proc. ACM CHI 1996*, 142-150.

[5] Golledge, R., and Stimson, R., *Spatial Behaviour*, New York: Guilford Press, 1997.

[6] Gutwin, C., Schneider, K., Paquette, D., and Penner, R., Supporting Group Awareness in Distributed Software Development, Proc. IFIP EHCI 2004, 109-117.

[7] Hill, W., Hollan, J. Wroblewski, D. and McCandless, T., Edit Wear and Read Wear. *Proc. ACM CHI 1992*, 3-9.

[8] Jones, W., and Dumais, S., The Spatial Metaphor for User Interfaces: Experimental Tests of References by Location versus Name. *ACM TOIS*, 4, 1986, 42-63.

[9] Lynch, K., *The Image of the City*, Cambridge: MIT Press, 1960.

[10] Mockus, A., Fielding, R., and Herbsleb, J. Two Case Studies of Open Source Software Development: Apache and Mozilla, *ACM ToSEM*, 11, 3, 2002, 309-346.

[11] Robertson, G., Czerwinski, M., Larson, K., Robbins, D., Thiel, D., and van Dantzich, M., Data Mountain: Using Spatial Memory for Document Management, *Proc. ACM UIST 1998*, 153-162.

[12] Shneiderman, B. Tree Visualization with Treemaps: A 2-D space-filling approach. *ACM Transactions on Graphics* 11, 1, 1992, 92-99.

[13] Siegel, A., and White, S., The Development of Spatial Representation of Large-Scale Environments. In H.W. Reese ed., *Advances in Child Development and Behavior*, Academic Press, 1975.

[14] Silverman, I., and Eals, M., Sex differences in spatial abilities: Evolutionary theory and data. In J. Barkow, L. Cosmides, & J. Tooby, eds., *The Adapted Mind*, New York: Oxford Univ. Press, 1992, 533-549.

[15] Tauscher, L. and Greenberg, S., Revisitation Patterns in World Wide Web Navigation. *Proc. ACM CHI 1997*, 398-406.

[16] Thorndyke, P., and Goldin, S., Spatial Learning and Reasoning Skill, in *Spatial Organization: Theory, Research, and Application*, H. Pick and L. Acredolo, eds., New York: Plenum Press, 1983, 195-217.