Icon Abacus: Positional Display of Document Attributes

Eric A. Bier
Palo Alto Research Center, Inc.
3333 Coyote Hill Road
Palo Alto, CA 94304
+1 650-812-4439
bier@parc.com

Adam Perer
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20742
+1 301-405-7445
adamp@cs.umd.edu

ABSTRACT
This paper presents *icon abacus*, a space-efficient technique for displaying document attributes by automatic positioning of document icons. It displays the value of an attribute by using position on a single axis, allowing the other axis to display different metadata simultaneously The layout is stable enough to support navigation using spatial memory.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *graphical user interfaces.* H.3.7 [**Information Storage and Retrieval**]: Digital Libraries – *user issues*.

General Terms
Algorithms, Design, Human Factors.

Keywords
Visualization, metadata, spatial memory, computer-aided reading.

1. INTRODUCTION
We are interested in tools that help knowledge workers, such as researchers, intelligence analysts, patent attorneys, teachers, and financial analysts, perform in-depth reading. By in-depth reading, we mean reading many documents on a topic over an extended time. During in-depth reading, the reader builds up a collection of documents, decides what order to read them in, and keeps track of information about each. Visualization can support in-depth reading by displaying the current document collection in a way that reveals important attributes of its documents. If a visualization is designed carefully, the user builds up spatial memories of its layout over time, allowing rapid re-finding of documents and document groups, as was explored in the Data Mountain system for manually-positioned documents [6].

We want the user to see the same layout repeatedly so as to form strong spatial memories of it [3]. By using layouts that express more document attributes, we allow the user to switch layouts less often and thus to form better spatial memories of these layouts.

Some of the document attributes that are important during in-depth reading have a small range of values. For example, users want to distinguish documents that have been read from those that have not; this requires an attribute with two values. Users may wish to distinguish between documents that are unrated or have a low, medium, or high rating, requiring an attribute with four values. Thus, a good visualization of attributes with small value sets can be immediately applied to aid in-depth reading.

JCDL '05, June 7–11, 2005, Denver, Colorado, USA.

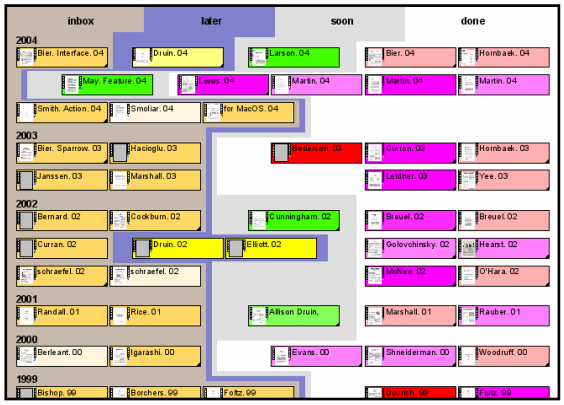

Figure 1. Icon abacus with four columns

As described by Mackinlay [4], there are many graphical presentation techniques available to encode an attribute. Of these, *positional* techniques are the most expressive because they are the most accurately perceived by users. They can also be used by color-blind users. Finally, they can be constructed to place similar objects close together, where they are easy to visually compare or select as a unit.

There has been previous work that uses icon-positioning as a way to show document information. For example, systems by Shneiderman [7] and by Nowell [5] use two-dimensional grids of document icons to display two metadata attributes of each document simultaneously. However, these grid representations waste screen space in those cases where some grid cells contain many icons while others are sparsely filled. Another approach, quantum strip treemaps [1], lays out groups of icons in strips of variable-width rectangles. This approach is compact but encodes only a single document attribute in the position of each icon group. To solve these problems, we have developed *icon abacus*, a positional layout technique that uses screen space efficiently and displays multiple attributes. It is part of the Instant Bookplex system [3].

2. ICON ABACUS
Like a two-dimensional grid, the icon abacus technique presents a collection of document icons in a way that shows two attributes of each document at once. In one dimension (e.g., vertically), the information display is much like a grid; the icons are displayed in disjoint horizontal strips that are stacked vertically. However, in the horizontal direction, icon abacus breaks with the grid model. Instead of placing the icons in rectangular cells where each cell spans the same range of x values as the cell above and below, we allow the left and right edge of each cell to have a jagged shape,

and we allow cells in different rows to start and end at different *x* values than the ones above and below. Using the extra degrees of freedom in this design, we are able to display the same amount of information in less space or more information in the same space.

The original inspiration for icon abacus came from physical abacuses. They have enough empty space on each wire so that the beads can move left and right along the wire, allowing them to be separated into a left group of beads and a right group of beads, for example. If we imagine that each bead is an icon and that each group represents one value of a document attribute, we almost have a way to display metadata. What is missing in our physical abacus is that the beads cannot be reordered along the wire. Using software, we remove this limitation and the result is icon abacus.

Figure 1 shows an icon abacus that has four columns to represent documents that are unexamined, to be read later, to be read soon, and already read, respectively. The icons are sorted vertically by the date of publication; each new date is introduced with a section header, such as "2004". Within each vertical section or "icon paragraph" the icons are sorted alphabetically by first author; in particular all of the authors in the first line of the paragraph come before all authors in the second line, etc. Also, within a group on a single line, the icons are sorted left to right by author. In this way, this abacus provides positional clues to three document attributes at once: date, reading status, and first author.

In addition to horizontal displacement, the icon groups in the abacus are distinguished by spacing and background color. Adjacent groups along a given line are separated by a gap equal to $1/n$ where n is either 3 or the number of columns in the abacus, whichever is larger. Not only do the gaps provide some empty horizontal space between groups on a single line, but they also add a displacement that helps the user distinguish different groups on adjacent lines, even if those groups have overlapping *x* extents (e.g., in Figure 1, the "done" group from the second line of icons sits over the "inbox" group from the third line, but it is clear from displacement alone that the two groups are distinct).

A colored background polygon behind each group of icons further aids the user in seeing group boundaries. In Figure 1, the polygons are (left to right) tan, blue, gray, and white. To reduce visual clutter, we apply a smoothing algorithm to the polygons. In particular, each vertical segment of the polygon tries to make itself collinear with the segment immediately above it, if this can be done without touching any icons from a different group or crossing other polygons.

As users work with documents, their attributes may change. For example, after a user reads a document, it should no longer be grouped with the unread documents. When this occurs, an animation shows the icon's movement from one abacus group to the other (and of other icons moving out of the way), so the user can expend fewer cognitive resources to see what has changed, and maintain stronger spatial memories [2]. The moving icon stays at the same vertical coordinates, moving only horizontally. This simple motion is easy for the user to follow.

While icon abacus can display an attribute with any finite number of discrete values, it works best when the number of values is relatively small. With a larger number of values, the jagged nature of the abacus layout may make the columns difficult to distinguish. However, icon abacus can be combined with quantization (mapping many values onto a few) or in a hybrid approach with grids, to handle attributes with more values.

Icon abacus has an interesting advantage over other graphical presentations of attributes, such as color changes, that show which documents are or are not of interest to read next. Because the two groups of documents are separated in space, the reader's eye can scan over the group of interesting documents, without encountering many of the irrelevant documents.

Relative to a grid, the icon abacus provides greater layout flexibility, predictability, and average compactness. The designer can choose any number of icons per row, and any rule for sorting icons within groups and sections. As icons are added to an icon paragraph in the abacus, the vertical size of the paragraph grows only when a line is full of icons. By contrast, the vertical size of a grid row grows whenever any cell in that row requires more vertical space. As a result the user will often see a space savings. We estimate a savings of roughly 50% over a four-column grid in typical cases where most rows have sparse data.

We are in the process of testing these techniques on several different categories of knowledge workers to discover the extent to which they are valuable in practice.

This research was funded in part by the Advanced Research and Development Activity NIMD program (MDA904-03-C-0404).

3. REFERENCES

[1] Benjamin B. Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: making effective use of 2d space to display hierarchies. *ACM Transactions on Graphics*, Vol. 21, No. 4, October 2002, pages 833-854.

[2] B.B. Bederson and A. Boltman. Does Animation Help Users Build Mental Maps of Spatial Information? *Proceedings of IEEE InfoVis '99*, 1999, pages 28-35.

[3] Eric Bier, Lance Good, Kris Popat. A document corpus browser for in-depth reading. *Proceedings of the Joint Conference on Digital Libraries 2004*, pages 87-96.

[4] Jock Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, Volume 5, Issue 2 (April 1986), pages 110-141.

[5] Lucy Terry Nowell, Robert K. France, Deborah Hix, Lenwood S. Heath, and Edward A. Fox. Visualizing Search Results: Some Alternatives to Query-Document Similarity. *Proceedings of SIGIR '96*, 1996, pages 67-75.

[6] George G. Robertson, Mary Czerwinski, Kevin Larson, Daniel C. Robbins, David Thiel, and Maarten van Dantzich. Data Mountain: Using Spatial Memory for Document Management. *Proceedings of ACM UIST '98*, 1998, pages 153-162.

[7] Ben Shneiderman, David Feldman, Anne Rose, and Xavier Ferré Grau. Visualizing Digital Library Search Results with Categorical and Hierarchical Axes. *Proceedings of the 5th ACM Conference on Digital Libraries*, 2000, pages 57-66.