A Scrollbar-based Visualization for Document Navigation

Donald Byrd

Center for Intelligent Information Retrieval
Computer Science Department
University of Massachusetts
Amherst, MA 01003

dbyrd@cs.umass.edu

ABSTRACT

We are interested in questions of improving user control in best-match text-retrieval systems, specifically questions as to whether simple visualizations that nonetheless go beyond the minimal ones generally available can significantly help users. Recently, we have been investigating ways to help users decide—given a set of documents retrieved by a query—which documents and passages are worth closer examination.

We built a document viewer incorporating a visualization centered around a novel content-displaying scrollbar and color term highlighting, and studied whether the visualization is helpful to non-expert searchers. Participants' reaction to the visualization was very positive, while the objective results were inconclusive.

Keywords

Digital libraries, systems, information retrieval, user interface, GUI, user study, visualization, scrollbar.

1. INTRODUCTION

The advent of the World Wide Web has resulted in an explosion of text searching by end users as opposed to expert intermediaries. Most of the searching on the Web is via best-match systems, especially those of the so-called "search engines". However, it is clear that, for a great many users, current best-match text-retrieval systems leave much to be desired. If anything, experts (primarily librarians and intelligence analysts) are even more dissatisfied with best-match systems than "ordinary" users are. As user-interface designers and researchers, we have long felt that much of the problem is a question of control.

We have recently been investigating the "review of results" aspect of the task. Once the user has run a search and a number—often a very large number—of documents have been retrieved, how can they decide where to focus their attention? Which documents and passages are worth closer examination? We believe that, with appropriate visualizations, result lists could make it much easier to decide which documents are really likely to be relevant, and document viewers could make it much easier to decide whether

the document being shown is in fact relevant. This is hardly a new idea, but we believe that the issues involved are subtle and that the optimal visualizations have not yet been seen. We devised a visualization centered around a novel content-displaying scrollbar and color term highlighting, built a document viewer incorporating the visualization, and studied whether the visualization is helpful to non-expert searchers.

In this paper, we discuss the state of the art of visualizations of text-document content; describe our new visualization, document viewer, and study; and show how it could work with a previous visualization of our own. We then report on a preliminary user study with our new visualization. Participants' reaction to the visualization was very positive, while the objective results were inconclusive. Finally, we attempt to draw conclusions from our experience and we make suggestions for future research.

2. VISUALIZATIONS FOR TEXT RETRIEVAL

Several aspects of the information involved in a text-retrieval program can be visualized. A minimal list of sensible visualizations for document retrieval of any kind, with the "phases" of the task they apply to, might look like Table 1 (phases are named in the terms of Shneiderman et al [21, 22]).

Each of these visualizations can be done in many ways. First, even for a given visualization, different pieces of information might be visualized. For instance, VQ might show query structure, term weights, etc. VQR might show the numbers of occurrences of each query term (as in the commercial system CALVIN), the contributions of each term to the document's score (as in our earlier work: see [21] and Fig. 1 below), or the progression of appearances of terms in the document (as in the current research or Hearst's TileBars [13]). At its most basic, it might give term-occurrence information in Boolean form simply by listing terms that appear in the document (as in PRISE [11]).

Second, there are various graphical ways to realize a visualization of given information, varying in complexity, clarity, etc. For example, VRR might be realized in either 2-D or 3-D. In VQ, relative weights of terms might be shown in a pie chart or a histogram. But the possibilities go far beyond these simple questions: see any of several books by Tufte ([23], e.g.) for extensive discussion.

Third, while the term "visualization" suggests a passive display, visualizations can also be interactive, with affordances to let the user control the system. It is certainly possible (and it may well be desirable) to offer control of the full query-expressing power of a modern IR system in the framework of VQ and VQDB.

Fourth, for performance reasons, one might prefer to visualize an approximation to the desired information. For VQDB, for

	Information	Phase
VQ	the query alone	formulation
VQDB	the query in relation to the database(s)	formulation
VQR	the query in relation to individual retrieved documents	review of results
VRR	the retrieved documents in relation to each other	review of results

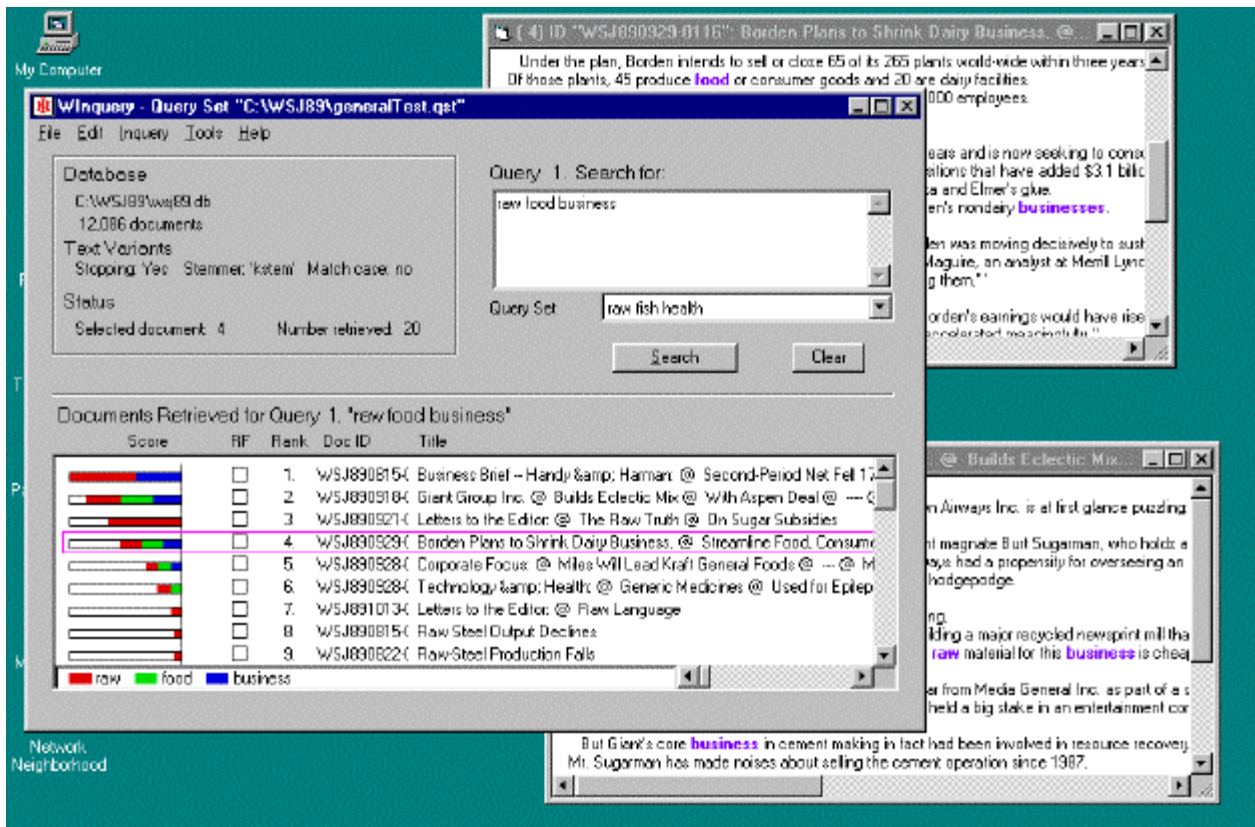**Table 1. Visualizations for Document Retrieval**

Figure 1. Visualization of query-term contributions to document score

example, one can show the query against the actual databases to be used, or against a "proxy" query-formulation database. The former is obviously preferable, but the latter is often much more practical, especially in a client/server situation (and most especially on the World Wide Web). This is basically the "query previews" idea of Doan et al [6].

Finally, note that some of these visualizations might be more or less tightly integrated: for example, VQR and VRR could be shown on a single display, as in LyberWorld [14].

A number of visualizations in text-retrieval systems are shown in a special digital-libraries issue of *Communications of the ACM* [8].

3. VISUALIZATIONS OF TEXT-DOCUMENT CONTENT

IR researchers have proposed many VQR's, i.e., ways to visualize the content of text documents as it relates to a query, for example:

- the document lens [19]

- TileBars [13]

- multiple bargraphs for term contributions to document score [24]

- our own single bars for term contributions to document score [21]

- VOIR [10]

- dynamic document viewers [4]

- thumbnails [16]

- multiple fisheye views [16]

These VQR-type visualizations can be cleanly divided into those which show where features occur within the document and those which do not. Our own earlier visualization mentioned above is of the latter type, but the present research is concerned with the former.

3.1 TileBars, Scrollbars, and Other Visualizations That Show Feature Locations

Among the best-known visualizations of text-document content in IR is "TileBars". In addition to descriptions such as [13] and [18], an online demo of TileBars is available [3]. Rao makes the thought-provoking observation [18]: "The TileBars interface allows the user to make informed decisions about which documents and passages to view, based on the distributional behavior of the query terms in the documents. The goal is to simultaneously and compactly indicate (i) the relative length of the document, (ii) the frequency of the terms sets in the document, and (iii) the distribution of the term sets with respect to the document and to one another." TileBars are displayed in a result list, one for each document retrieved.

Helping users make informed decisions about which *documents* to view is indeed important; so is helping them make informed decisions about which *passages* to view. But these are essentially independent questions. If you are going to show where terms are in a document and your visualization is as compact as TileBars are, you can certainly do it in a result list, and that way, the user gets help with both types of decisions at the same time. But we feel that seeing term locations in an overview is not that helpful. We will return to this point later. If, on the other hand, you are going to show where terms are with each individual document, there's already a place to do it: in the scrollbar.

Scrollbars are of course implemented in the standard user-interface toolkits for virtually all modern operating systems: see for example the user-interface guidelines for the Mac OS [1] or for Microsoft Windows [17]. Scrollbars are nearly always used to visualize and control the portion of a document that is displayed in an adjacent and much larger area. When they are used in this way, they are without exception filled with a neutral pattern that conveys no information about the document's content. However, we know of several systems that display an overview of a document's content in a small greatly-elongated window that functions somewhat like a scrollbar in terms of both what it shows and how it is used.

First, in [2], see the smaller window in Fig. 3, and comments on it in the text (p. 35). Second, consider Microsoft's WinDiff text-file-comparison utility for MS Windows. Besides displaying the exact text in each file in a large window, WinDiff (version 4.0) shows overviews of both files in narrow vertical strips to the left of the window, with colored bars marking differences. Clicking in either strip jumps the text display to that point. But no documentation we know of even mentions the strips.

The navigation aid these two "widgets" provide may be very useful, but overall, they are far less powerful than standard scrollbars. Nor do the non-standard appearances of these widgets facilitate learning to use them. But a third project actually shows document content inside a standard scrollbar much as we do. This work is described in two U.S. patents by Wroblewski et al [25, 26]; [15], by most of the same authors, describes a related idea, and Shneiderman's well-known text [20], pp. 451–452, briefly describes ideas that are somewhat related. Wroblewski and his colleagues do not fill their scrollbars with a neutral pattern: instead, they display what they call an "enhanced scrollbar", where the enhancements include "maps of significant tasks-specific attributes of the data file....displayed in the scroll bar field of the display along with the scroll bar."

In contrast, TileBars are even more remote from standard scrollbars. The view of actual document content does not appear until the user clicks on the TileBar; even then, the view replaces the entire contents of the window, including the TileBar, and it has a conventional scrollbar, which however allows only scrolling within the current segment of the document. So the TileBar widget bears only casual resemblance to a scrollbar.

Many visualizations that show where features occur within the document are examples of generalized fisheye views [9]. Kaugars' multiple fisheye view is one, of course, but the document lens is also a clear-cut case. It is less obvious that TileBars or scrollbars that show feature locations have anything to do with fisheye views, but, if one considers space occupied as just one way to display salience, the basic idea is the same. The scrollbar or the entire TileBar is an independent view of the document, with a degree-of-interest function whose value is zero for non-features, and with color or intensity replacing area as the way of displaying salience.
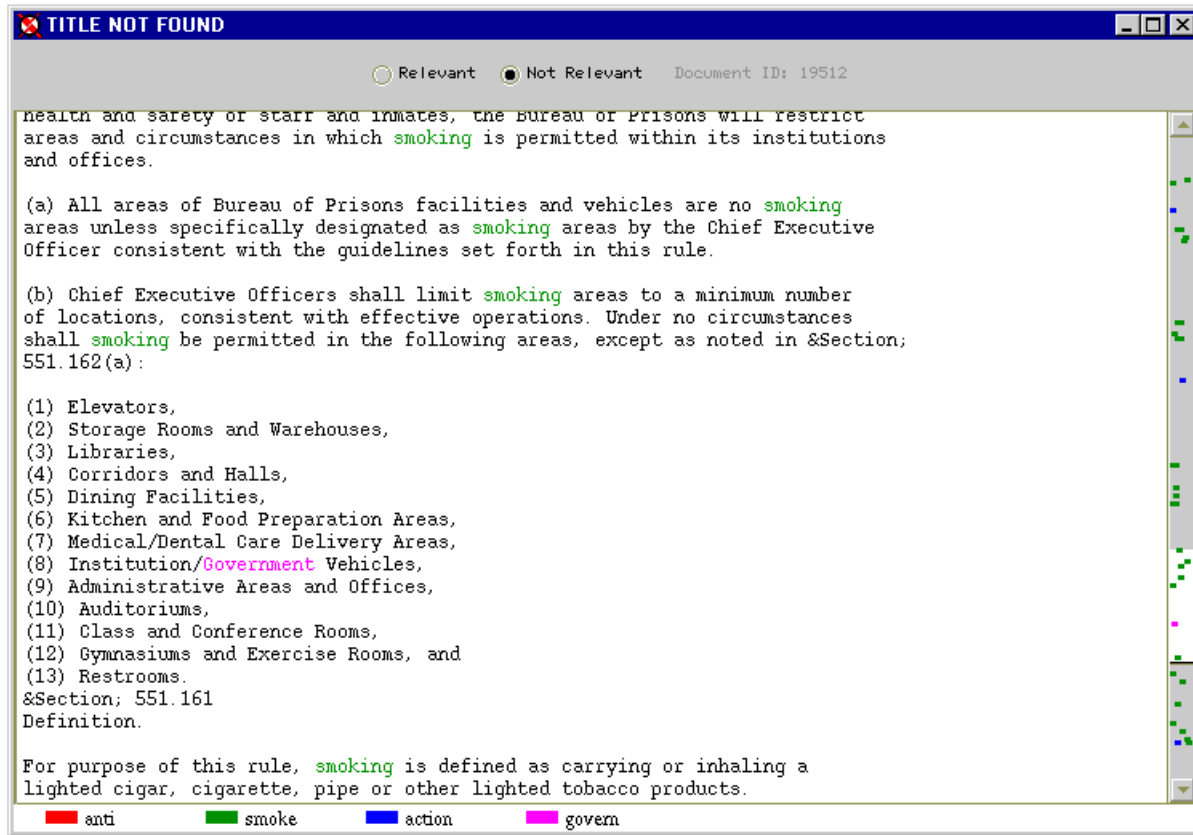
4. OUR VISUALIZATION

A typical screen display of our document viewer is shown in Fig. 2. The visualization has the following elements:

- Occurrences of each different word in the query and its variants are highlighted in a different color.

- The vertical scrollbar contains small icons in the same colors. This is the central feature; it has been characterized as the "scrollbar with confetti" or (particularly meaningful to parents of younger children) "scrollbar with rainbow sprinkles".

- An area at the bottom of the window contains a "legend" relating the words and colors.

(Unfortunately, the black-and-white rendering in printing the figure loses much of the clarity of the original. On a standard color monitor, it is obvious that the word "smoking" appears six times in the window and the word "government" appears once. Also, from the scrollbar, it is obvious that the latter is the only recognized variant of "govern" in the entire document.)

The scrollbar icons show where in the document occurrences of the corresponding query words, or variants of them, are. The idea is to help the user find as quickly as possible the parts of the documents that are most likely to be relevant. The icons could be of any size and shape, but we use 3-by-3 pixel squares. The horizontal positions of the icons as well as their vertical positions correspond to the positions of the words in the text area. In effect, the scrollbar contains a miniature view of highlighted words in the entire document.

Note that, despite its unusual appearance, the vertical scrollbar works just like any vertical scrollbar: the top of the scrollbar

Figure 2. Scrollbar-based visualization

corresponds to the beginning of the document, and the bottom of the scrollbar corresponds to the end of the document. The icons are simply superimposed on the neutral pattern that normally fills scrollbars. To make the colors as easy to see as possible in at least part of the scrollbar, our "thumb" or "car" is plain white instead of the usual (platform-dependent) color and/or pattern.

This visualization is of course yet another instance of VQR, showing the query in relation to individual retrieved documents. We have previously implemented the term-score-contribution bars form of VQR mentioned above [21]. Now, calling that visualization "VQRa" and the present one "VQRb", it is particularly interesting to compare our work to Hearst's TileBars. VQRa consists of stacked colored bar segments; the size of each segment represents a term's contribution to the total belief score. Such a set of bar segments requires very little space, and—as with TileBars—a set is displayed with each document in a result list.

For allowing users to make informed decisions about which *documents* to view, we believe our VQRa is better than TileBars because it considers term weights, not raw term occurrences, and thereby shows why the documents were retrieved and ranked as they were. For allowing users to make informed decisions about which *passages* to view, we believe our VQRb is better than TileBars because it shows where terms occur in the text in the best possible way, via the scrollbar, so users can examine documents as efficiently as possible. In fact, VQRb should help the user determine whether the document discusses the desired concepts with far more confidence than either VQRa or TileBars

do. If the document really does discuss those concepts, VQRb should also help determine whether it discusses the concepts *in relation to each other* with at least as much speed and confidence as TileBars, and with much more confidence than VQRa.

We designed the experiment described later to begin shedding light on whether VQRb is actually useful.

5. IMPLEMENTATION

CIIR's InQuery retrieval engine is written in C; more recently, CIIR has developed JITRS (for Java InQuery Text Retrieval System), a Java class library that uses the JNI (Java Native Interface) package to allow Java programs to communicate with InQuery on a client/server basis. We implemented a document viewer incorporating the content-displaying scrollbar in Java, using JITRS for retrieval, and using the "Swing" package (part of Sun's Java Foundation Classes) for the user interface. Swing contains an object-oriented GUI toolkit, and the capability it offered of overriding scrollbar methods greatly eased implementation.

6. THE EXPERIMENT

We compared an experimental system incorporating our full visualization, to a control system with no visualization except for highlighting words in the text in a single color. We made two types of measurements: objective, including comparisons of participants' relevance judgements to the "official" ones, and how quickly they could judge documents; and subjective, i.e., how

much they liked using the visualization. To minimize irrelevant differences between the experimental and control systems, the code for the control system's scrollbar was in fact identical to that for the experimental system except that the control system skipped drawing the icons.

6.1 Participants

There were six participants, four male and two female, all college students. All were adult native speakers of English, at most 30 years old, with at least some experience with computers, and with normal color perception. All had experience with online searching (averaging over three years), but none had professional training or professional experience as a searcher. Characteristics of the searchers are summarized in the Appendix.

6.2 Tasks

The study was modeled to a considerable extent after the TREC 6 Interactive track experiment [5, 12]. Each participant did the same 10 tasks in the same order; the tasks involved identifying relevant documents in a given database.

Specifically, for each task, we gave the participant a description of an information need, plus—since we were interested only in the document viewer—a fixed query and a fixed number of documents to retrieve. The combination of fixed database, fixed query, and fixed number of documents to retrieve means that, effectively, a result list was predefined for each query. We asked participants to consider each result list and to judge relevance of as many as documents as possible in five minutes.

The number of documents in each result list was 30. Why that number? First, because it is generally agreed that 30 at the most is an upper limit on the number of documents users of best-match interactive IR systems will bother with, at least on the Web. Second, because this is a large enough number to make the chances of a ceiling effect minimal with only five minutes per search.

Database and Topics. For the usual reasons (so we could use TREC relevance judgments, etc.), we chose to use part of the TREC document collection with information needs from the TREC topic collection. Note that the content-displaying scrollbar is not likely to be of much use with short documents, since a user can browse through such documents very quickly with no more aid than conventional single-color highlighting of query terms. But we wanted to encourage users to rely on our scrollbar icons as much as possible, so we needed long documents. The Federal Register consists of official U.S. government documents. In general, these documents are long; the longest are well over a megabyte. Also, they tend to contain large amounts of "bureaucratese" and/or trivial details, and they have no titles that a program can recognize as such and display, even though most contain something a human being can recognize as a title. All these factors make Federal Register a very difficult place to find information and a potentially fruitful test collection for a document viewer. For this study, we chose the 1989 Federal Register (FR89), which is one of the TREC Volume 1 document collections. FR89 contains about 26,000 documents whose raw text totals over 260 megabytes.

Queries. Wanting short and unstructured queries, we started with the TREC topic titles, and made minor changes in two cases.

Although FR89 contains many long documents, not all queries will find them. We selected queries whose top 30 documents had an average length against FR89 of over 1000 words.

Additional criteria for the queries we chose were:

• Maximum length of any retrieved document not too high. This is mostly because our document viewer takes quite a while to display a long document. We set a limit of 50,000 words.

• Neither too few nor too many non-stopped terms. If there's only one term, our multiple-color feature wouldn' t be used; if there are too many, distinguishing the colors would be very hard. We deemed 2 through 5 terms to be acceptable.

• Top-30 precision neither too high nor too low, to avoid ceiling and floor effects. Our queries had a minimum of 0.10 and a maximum of almost 0.65.

The queries we ended up with, together with the original TREC topic numbers, are listed in Table 2. Note that two of the queries differ slightly from the corresponding topic titles: we omitted a word from the title of topic 182 to reduce the number of terms to five, and we replaced "U.S." with "American" in the title of topic 106 to sidestep a problem with InQuery.

	TREC number	Query (TREC title, if different)
1.	95	computer-aided crime detection
2.	106	American control of insider trading (U.S. control of insider trading)
3.	108	Japanese protectionist measures
4.	115	impact of the 1986 immigration law
5.	119	actions against international terrorists
6.	123	research into & control of carcinogens
7.	125	anti-smoking actions by government
8.	174	hazardous waste cleanup
9.	182	commercial overfishing food fish deficit (commercial overfishing creates food fish deficit)
10.	188	beachfront erosion

Table 2. Queries

6.3 Procedure

We ran the experiment in our usability laboratory on campus. A "facilitator" was in the room with the participant all of the time except while the participant was doing the tutorials. The same person acted as facilitator for all participants.

First, each participant filled out a questionnaire to give us basic demographic information. Then they took a standard psychometric test from ETS [7], a test of structural visualization (VZ-2, the Paper Folding test): the mean score was 14.8 of a possible 20. More information is given in the Appendix.

Next, the participant was given a tutorial to learn one system, then they worked on the first five topics. After a short break they were given a tutorial on the other system, then they worked on the other five topics. Each search had a 5-minute time limit, and the participant was instructed to stop working if they had not finished in 5 minutes. A countdown timer displayed on-screen ran continuously, even while the user was waiting for the system to show a document: we will discuss the implications of this later.

We gave the participant a short questionnaire after each search. After all the searches were finished we gave them a final questionnaire, then "debriefed" them. The study was conducted single blind: the participants were not told until the debriefing which system was the control and which was the experimental system. However, it would have been obvious to many people which was the experimental system.

We ran each participant through the entire study in a single essentially continuous period of about two and a half hours. Half did the first five searches with the experimental system, and the other half did the first five with the control system: thus, there were two conditions. (We considered randomizing the order in which participants were given the searches, to minimize order effects. However, this would introduce significant complications, especially since we did not want participants to switch systems repeatedly, and we decided—as the TREC 6 designers did—that the benefit did not justify the added complexity.) With six participants, this design gives 6 x 5 = 30 data points per cell, enough for a meaningful analysis of variance.

6.4 Results

For objective measurements, we analyzed the participant's relevance judgments by comparing them to the official TREC judgments. We then performed an ANOVA (ANalysis Of VAriance) using query, participant, and system as factors. For dependent variables, we used

- Number of documents judged

- Number of documents correctly judged

- Accuracy

Query- and participant-dependent results were significant. However, we found no system-dependent results. The differences between the experimental system and the control system were what would be expected by chance. We did observe a slight increase in accuracy with the experimental system, but it was not enough to be statistically significant.

We also made subjective measurements by asking participants whether they preferred FancyV (the full visualization) or SimpleV (the very limited one), and how strong their preference was on a five-point scale ("not at all" to "extremely"). Combining these questions gives nine values. Using -4 = extremely strong

preference for SimpleV, 0 = no preference, and +4 = extremely strong preference for FancyV, we got one -2, two +3, and three +4, for a mean of 2.67: a fairly strong preference for FancyV.

Participants made a number of illuminating comments. Two who preferred FancyV commented that—while the visualization wasn't always useful—when it was not useful, it didn't get in the way. One went on to say that he couldn't understand why anyone would not prefer the scrollbar icons: "if you want, you can just ignore them."

One participant who started with FancyV said while using SimpleV "I'm pretty much flying by the seat of my pants; it's much more hit-and-miss...I felt like, with the colors and dots [of FancyV], I had much more chance of forming a mental model of each document."

The one person who liked SimpleV better said she preferred it because of its simplicity.

6.5 Discussion

It is not surprising that we found no system effects of statistical significance: six is a very small number of participants. In addition, there were some problems with our implementation.

- Once started, the countdown timer ran continuously, even while the participant was waiting for the system to show a document they had requested. This was a serious problem because the system took a long time to open long documents, so that participants spent a significant part of the time—often over a minute of the five available—doing nothing.

- For at least one of the tasks, the query did not describe relevant documents very accurately. Query 8 was "hazardous waste cleanup", but the description of the information need made it clear that only documents pertaining to hazardous waste cleanup under the Superfund program were relevant. Several participants complained about this discrepancy. Of course the word "Superfund" was not highlighted with either SimpleV or FancyV. But with FancyV, participants had to scroll through the text just to look for occurrences of the word "Superfund"; with SimpleV, they were already scrolling through the text. Therefore it is plausible that this omission had more effect on participants' performance with the visualization.

It is also quite possible that the visualization simply has too long a learning curve to see any effect in at most 25 minutes of real use after a short training period.

On the other hand, the strong preference participants had for the visualization is very encouraging: user satisfaction with an interface is important independent of any objective criteria. Though all of the participants in the study were end users, we also have some evidence that the visualization will make expert searchers happy. We had two experts (university librarians) try two or three queries with SimpleV, then two or three queries with FancyV. Both felt the visualization was very useful; one commented that it was easy to pick out what she was looking for by color alone, and that FancyV was "200 times better" than SimpleV.

6.6 Preliminary Report on a Followup Study

As of this writing, we have just finished running a followup experiment identical to the one just described, except with 20 participants instead of six, and with the countdown-timer problem corrected. Unfortunately, analysis of the objective data has not yet

been completed, though the initial analysis again shows no system-dependent results.

We made the same subjective measurement with the same nine-point scale as before. This time, we got one -2, one +1, five +2, six +3, and seven +4, for a mean of 2.75. This again represents a fairly strong preference for FancyV. But this time, with the much larger number of subjects, this result is highly statistically significant: by the weakest applicable test, the sign test, it is significant at $p < .0001$.

7. CONCLUSIONS

There is reason to believe that appropriate visualizations for the content of retrieved text documents will make life easier for expert searchers as well as end users. In this initial study and first followup, we tested only ordinary users; in a later study, we expect to test both types, as we did for TREC 6 [5].

The overwhelming approval our visualization got from the users we tested, both in the initial study and in the followup, presumably means that they felt it would help them find information more quickly and/or accurately. Yet the objective data (at least according to analysis so far) shows no such effect. We believe that the visualization is really capable of helping, but that the problems we have identified with the implementation of the study nullified the effect. It would be extremely interesting to see another study with these factors changed.

Like many visualizations, ours does not scale well in all respects. In particular, as we have mentioned, it is difficult to distinguish more than about five colors. This could be alleviated by using larger icons, though of course there are drawbacks to that. Another solution, and one commonly used in situations like this, is to cluster the query terms, either manually (as with TileBars) or automatically.

Finally, note that displaying in scrollbars indications of the locations of interesting features is in no way limited to text. Nor is it limited to showing the results of searches: an outline or HTML editor could display icons at the positions of important hierarchic levels. All that is required is that the system be able to identify interesting features of documents. Non-icon-based displays could be useful in such applications as signal-processing programs. We believe that displaying indications of document content in scrollbars in whatever form has great potential to make programs of many types easier to use.

8. ACKNOWLEDGEMENTS

9. APPENDIX

9.1 Detailed Characteristics of Participants

The following is a summary of the participants' responses to the Entry questionnaire.

A. General information. For Education, we show only the current level of each participant.

	Total
Education:	
Undergraduate	3
Master's student	2
Doctoral student	1
Age:	
Under 21	1
21-30	5
Male/Female	4 Male / 2 Female

B. Computer and searching experience. For each item, the mean is given, followed by the median. Except for "Years searching", all are on a scale of 1 to 5, with 1 = none, 5 = a lot.

	Mean, median
Computer usage	4.3, 4.5
Years searching	3.4, 2.5
Search library catalogs	3.5, 2.5
Search CDROMs	2.3, 2
Search commercial services	1.2, 1
Search the WWW	3.7, 3.5
Search other	1, 1
Full-text databases	1.3, 1
Ranked output	1.8, 1
Mouse-based interface	4.7, 5
3-D interfaces	2.7, 2.5

9.2 Test Scores

Participants' scores on the VZ-2 psychometric test ranged from 5 to a perfect score of 20. Here is a summary of their scores. The mean is given, followed by the median.

	Mean, median
Paper Folding (VZ-2)	14.8, 17.25

10. REFERENCES

[1] Apple Computer, Inc. (1992). *Macintosh Human Interface Guidelines*. Reading, Massachusetts: Addison-Wesley.

[2] Ball, Thomas A., and Eick, Stephen G. (1996). Software visualization in the large. *IEEE Computer* 29, 4 (April 1996), pp. 33-43.

[3] Berkeley Digital Library Project. TileBar demo. http://galaxy.cs.berkeley.edu/tilebars/ .

[4] Boguraev, B., Kennedy, C., Bellamy, R., Brawer, S., Wong, Y., and Swartz, J. (1998). Dynamic Presentation of Document Content for Rapid On-Line Skimming. *Spring Symposium*, AAAI.

[5] Byrd, D., Swan, R., and Allan, J. (1997). TREC-6 Interactive Track Report, Part I: Experimental Procedure and Initial Results. Tech Report IR-117, University of Massachusetts Computer Science Dept.

[6] Doan, K., Plaisant, C., and Shneiderman, B. (1996). Query Previews in Networked Information Systems. *Proc. Third Forum on Research and Technology Advances in Digital Libraries, ADL '96*. IEEE CS Press, 120–129. Also available as TR 95-16 at http://www.cs.umd.edu/projects/hcil/Research/tech-report-list.html#1996 .

[7] Ekstrom, R.B., French, J.W., Harman, H.H., and Dermen, D. (1976). *Manual for Kit of Factor-Referenced Cognitive Tests 1976*. Princeton, New Jersey: Educational Testing Service.

[8] Fox, E., and Marchionini, G. (eds.) (1995). Digital Libraries. *CACM* 38,4 (entire issue).

[9] Furnas, George W. (1986). Generalized Fisheye Views. *Proceedings of SIGCHI '86*, pp. 16–23.

[10] Golovchinsky, Gene (1997). What the Query told the Link: The Integration of Hypertext and Information Retrieval. *Proc. ACM Hypertext 97*, pp. 67–74.

[11] Harman, Donna (1992). User-Friendly Systems Instead of User-Friendly Front Ends. *JASIS* 43, pp. 164–174.

[12] Harman, Donna (1997). *The Fifth Text REtrieval Conference*. Gaithersburg, MD: National Institute of Standards and Technology.

[13] Hearst, M.A. (1995). Tilebars: Visualization of term distribution information in full text information access. *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*.

[14] Hemmje, M., Kunkel, C. and Willett, A. (1994). LyberWorld - A Visualization User Interface Supporting Fulltext Retrieval. *Proceedings of the 17th Annual Int. Conference on Research and Development in Information Retrieval* (SIGIR '94, Dublin, 3-6 July 1994), pp. 249–257.

[15] Hill, W.C., Hollan, J., Wroblewski, D., and McCandless, T. (1992). Edit Wear and Read Wear. *Proc. ACM SIGCHI Conference on Human Factors in Computing Systems*.

[16] Kaugars, Karlis J. (1998). *A Hierarchical Approach to Detail and Context Views*. Unpublished doctoral dissertation, New Mexico State University.

[17] Microsoft Corporation (1995). *The Windows Interface Guidelines for Software Design*. Microsoft Press.

[18] Rao, R., Pedersen, J., Hearst, M., Mackinlay, J., Card, S., Masinter, L., Halvorsen, P.-K., and Robertson, G. (1995). Rich Interaction in the Digital Library. *CACM* 38, 4, pp. 29–39.

[19] Robertson, G., and Mackinlay, J. (1993). The document lens. *Proceedings of the ACM Symposium on User Interface Software and Technology*. ACM Press.

[20] Shneiderman, Ben (1998). *Designing the User Interface*, 3rd ed. Addison-Wesley.

[21] Shneiderman, B., Byrd, D., and Croft, W.B. (1997). Clarifying Search: A User-Interface Framework for Text Searches. *D-Lib Magazine*, January 1997. Available at http://www.dlib.org . Also available in a slightly revised version as Tech Report IR-107, University of Massachusetts Computer Science Dept.

[22] Shneiderman, B., Byrd, D., and Croft, W.B. (1998). Sorting out Searching: A User-Interface Framework for Text Searches. *CACM* 41,4 (April 1998), pp. 95–98.

[23] Tufte, Edward (1983). *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press.

[24] Veerasamy, A., and Belkin, N.J. (1996). Evaluation of a Tool for Visualization of Information Retrieval Results. *Proceedings of SIGIR '96*, pp. 85–92.

[25] Wroblewski, D., Hill, W., and McCandless, T. (1994). Computer Display Unit with atttribute enhanced scroll bar, U.S. patent number 5339391.

[26] Wroblewski, D., Hill, W., and McCandless, T. (1995). Attribute-enhanced scroll bar system and method, U.S. patent number 5479600.