Multi-perspective ranking and visualization of Web data

Mei Kobayashi, Georges Dupret *, Oliver King †, Hikaru Samukawa and Kohichi Takeda
IBM Japan, Ltd., Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242-8502 Japan
mei,samukawa@trl.ibm.co.jp, Kohichi_Takeda@jp.ibm.com

Abstract: We present a computationally fast and portable software tool to present information in a database from different perspectives and draw correlations between the perspectives. Although the tool may be used to understand the contents of any large database, our studies focus on retrieval and ranking of Web documents.

keywords: information retrieval, information outlining, Lanczos algorithm, site outlining, search engine, World Wide Web.

1. Background

The exponential growth of information available on the World Wide Web has been documented in numerous studies, e.g., [6]. The studies also find that Internet users are turning to search services in increasing numbers to find the information they are seeking, but they are not necessarily satisfied with their performance. The speed of transmission and retrieval of information and the format for presenting results from searches have been cited as major factors contributing to user dissatisfaction. In this paper, we present a ranking, retrieval and visualization system which aims to resolve some of these issues.

The idea for our work can be traced back to early information retrieval systems which were not necessarily Web-targeted. These prototypes have a visualization interface for examining contents of large data bases and digital libraries [7],[10],[15]. Recently, development of visualization tools specifically designed for examining Web document databases is increasing; examples include tools for "*mapping*" Web sites, i.e., visualizing hot links between sites and the organizational hierarchy within an individual user or group's home pages [8] and tools to help users search and classify relevant Web documents [1],[16],[18],[20]. Our system extends concepts and functions found in *Information Outlining* and *Web Site Outlining* systems [11],[17],[19] which organize and facilitate understanding of information in very large data bases and on the Web through the use of non-textual tools, such as graphs, charts and tables.

2. Multi-perspective Information Outlining

Our system (shown in Figure 1) is a hybrid of information and site outlining and a computationally fast document retrieval and ranking package [5],[9]. It enables users to simultaneously view information in a very large data base from several perspectives and to understand relationships between the perspectives on a PC in real time using pre-computed data of relationships between documents and query terms plus non-textual information about a Web site, e.g., most recent update, number of hits per day.

In our system, each perspective is represented initially by a set of query terms. After the first set of retrievals (example shown in Figure 2), the perspective(s) can be refined or modified using an interactive editing interface. For example, retrieved documents which are not wanted can be deleted manually on the screen (Figure 1, bottom LHS), and rankings for two documents can be swapped (Figure 1, bottom center). To facilitate comparisons between different perspectives, changes in the ranking of individual documents can be marked according to the perspective (Figure 1, bottom – arrows show changes in rankings of documents A, B, and C), and relationships between clusters according to different perspectives can be easily seen.

Our system can be used for a variety of purposes. For example, a prospective investor for a movie may want to know data associated with recent films with respect to different perspectives, such as: director, actors, screenplay writer, movie genre. To determine the casting of a proposed film, an investor may want to examine the prospective actors from several perspectives. Information on box office and video rental shop receipts for movies which were made with prospective actors might be useful for identifying potentially outstanding as well as disastrous working

*Student Intern, Inst. of Policy and Planning Sciences, Univ. of Tsukuba, 1-1-1 Tennoudai, Tsukuba-shi, Ibaraki 305 Japan, e-mail: gedupret@hotmail.com

†Student Intern, Dept. of Mathematics, Univ. of California at Berkeley, Berkeley, California 94720 U.S.A., e-mail: king@math.berkeley.edu

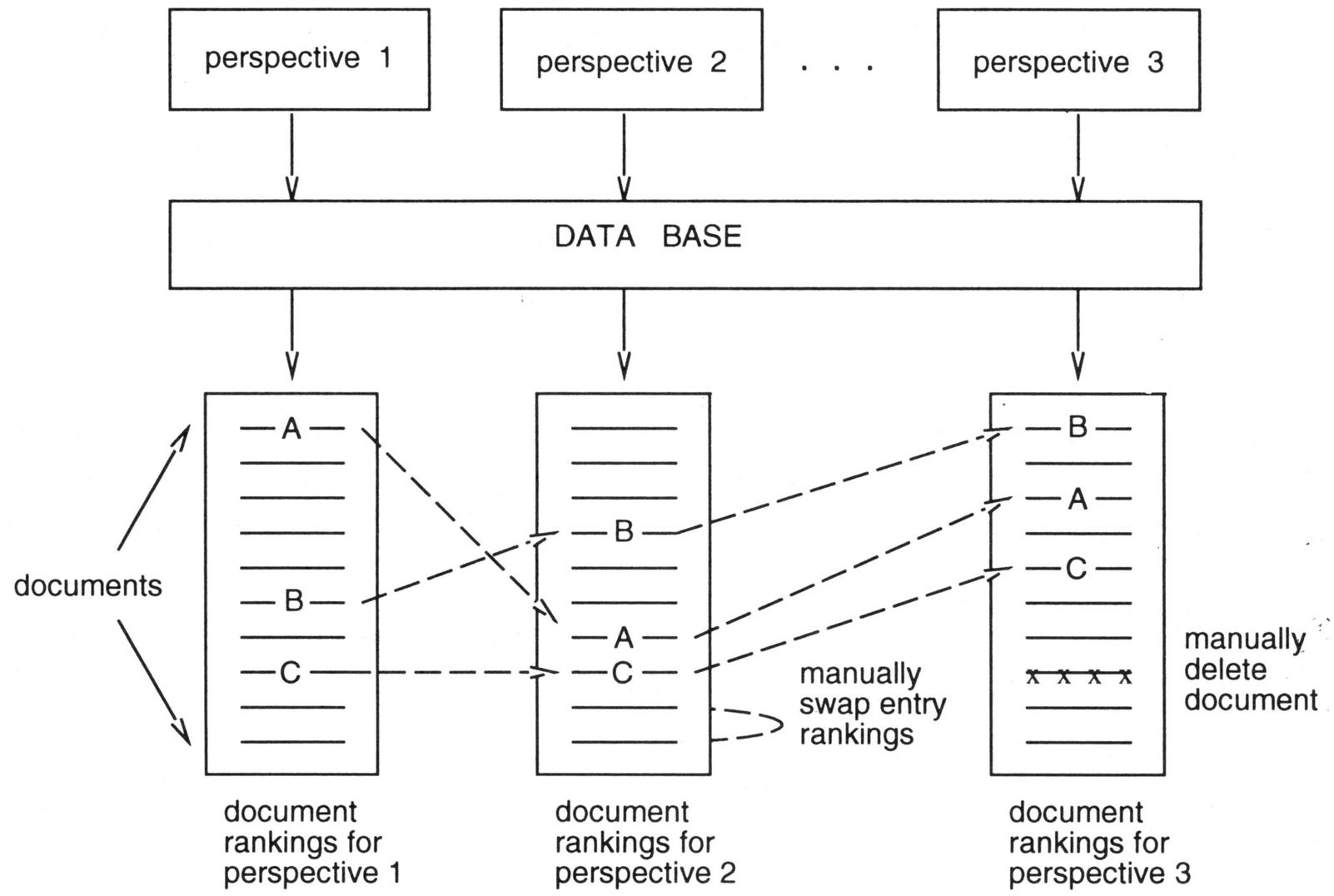

Figure 1: Multi-perspective retrieval and ranking

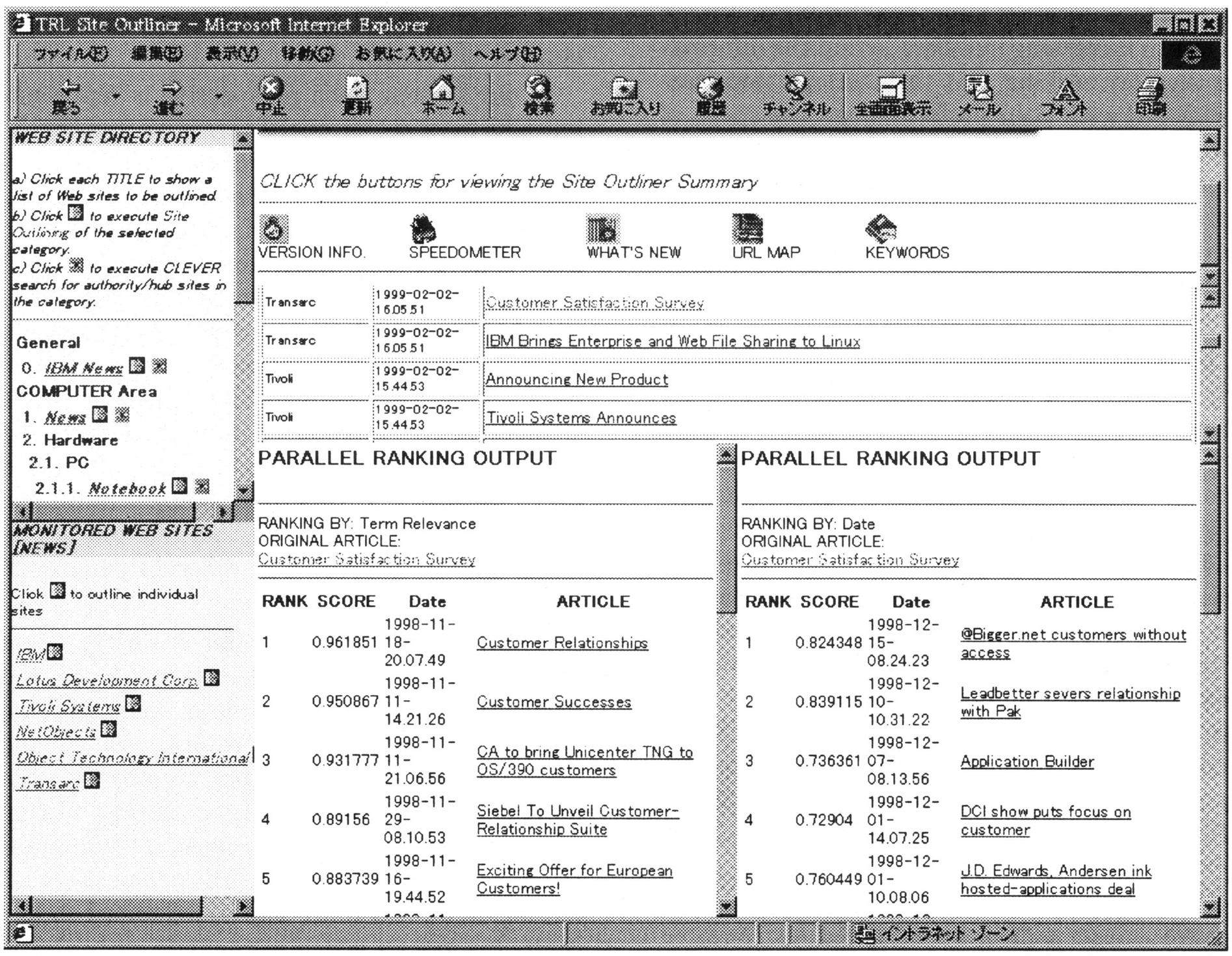

Figure 2: Screen view of results from multi-perspective retrieval

relationships. Acting experience within the same genre of movie and its success or failure is another important consideration.

Another application is to analyze Web pages from a reliable news organization to predict future corporate profits. For example, if an investor is considering the purchase of stocks of company A, searches using the keyword A plus additional terms will likely yield valuable information. However, an investor may also want to weight the search with information about major competitors, suppliers of parts, or the projected demand for the industry as a whole. An investor may want to examine how rankings of retrieved documents change when perspectives are changed before deciding whether and how much to invest.

Our system can also be used to expedite query formulation and refinement by processing several queries in parallel, then displaying the retrieved results in a manner which facilitates comparisons between the different queries.

3. Ranking

Our system relies on a pre-processed matrix model of document-query space to rank documents. Relationships between possible query terms and Web page documents are represented by an m-by-n matrix A

$$A = [a_{ij}] \ .$$

The entries a_{ij} consist of information on whether term i occurs in document j and weighting information to take into account: the length of the document; the importance (or relevance) of the query term in the document; and the frequency of appearance of the query term in the document. $A = [a_{ij}]$ is usually a very large, sparse matrix because the number of keywords in a single document is usually a very small fraction of the union of keywords in all of the documents.

This type of matrix model of document-query space first appeared in *latent semantic indexing* (LSI) [4], but the similarity of our technique and LSI ends there. Our algorithm also takes into account the ephemeral and free-style nature of the Web, i.e., the likelihood of being a keyword spam site; average time spent browsing a site; the most recent update; the probability of being an important authority on the subject being queried or merely a list of pointers; and the possibility of being mirrored or being a mirror site. In our system, the first factor is handled in the preprocessing step of ranking, i.e., before the computation of the *singular value decomposition* (SVD), described below. The remaining factors can be handled in pre- or post-processing, i.e., before or after computing the SVD. We have chosen post-processing.

To speed up our computations, we eliminate a column (or row) of the matrix if all of the entries of the column (or row) are zero or very small, since the document (or query term) is likely to have very little correlation with query terms (or documents). We do likewise if all of the entries of one column (or row) of a matrix are unity or close to unity, since the query term is a word found in a typical stoplist or the document may be a Web page used for keyword spamming.

The next step is the computation of the SVD of A [12], which must be fast enough to enable frequent updating. According to one study, text on a Web page remains unchanged, on the average, 75 days [2]. Many news, portal and stock/currency quote sites change several times a day or even every few minutes, e.g., *Bloomberg, CNN* [1]. It is impossible to update the document-query space matrix and its SVD as frequently as news sites, but a few times per week is a reasonable requirement.

We used the Lanczos algorithm to compute the SVD of the modified document-query matrix following an algorithm in a text by Parlett [12], pp. 288–289, for (partial) tridiagonalization of a symmetric matrix. The computed Lanczos vectors cease to be orthogonal to one another after some steps, and duplicate copies of eigenvalues are recovered. Parlett points to several techniques to maintain orthogonality of the vectors, such as: full reorthogonalization; selective orthogonalization without modifications (first proposed in [14], summary given on pp. 382–383 in [3]); selective orthogonalization with modifications (p. 371 in [3]); and Scott's orthogonalization (p. 321 in [12]). In the algorithms described above, we must compute (a user-specified [2] number of) eigenvalues and eigenvectors of the tridiagonal matrices T_j. Using Sturm sequences one may compute the number of eigenvalues of T_j which are larger than any given σ. And by using bisection we may locate all the eigenvalues to the desired accuracy. Once the eigenvalues are known, the eigenvectors may be found using inverse iteration (shifted by the approximate eigenvalue). Although a random starting vector for this iteration usually works well and converges within a couple of iterations, we also implemented *the method of Fernando* (pp. 252–255 in [13]), for finding the approximate eigenvector, which can be used as is, or further refined by inverse iteration. Graphs of our experimental results to determine the singu-

[1] http://www.bloomberg.com, http://www.cnn.com

[2] When the user specifies the maximum number of documents to be retrieved, the system will compute several more eigenvalues and eigenvectors than the number.

lar triplets (i.e., singular values and their corresponding singular vectors) of a document-query matrix of Japanese Web data from *Nikkei* newspaper (1994) appear in [5],[9]. Our computations of the singular triplets of matrices on the order of tens of thousands-by-tens of thousands are fast enough to allow updating every day.

4. Future Work

Demmel [3], p. 383 and Parlett [12], p. 319 discuss the Lanczos algorithm with partial reorthogonalization which can be used in together with selective orthogonalization. Their method exploits a recurrence relation for estimating the loss of orthogonality among the Lanczos vectors. Orthogonalization with respect to converged vectors is performed only when the loss in orthogonality threatens to become unacceptable. We have not yet implemented this method; it may lead to further reduction in computation time.

Extending our program to handle matrices of even larger order (e.g., up to 10 million-by-10 million) requires more than just straightforward enlargement of memory space and data structures. Fast matrix and vector access and computation will require new algorithms and possibly different data structures. Implementation of solutions which recognize compromises, such as finding *most* of the relevant documents, rather than *all*, may lead to significant speed-up.

Development of computationally inexpensive and more friendly user interfaces is an important area for investigation. Specialized tools developed by computer graphics experts for visualization are often too computationally intense to be useful and are sometimes awkward for non-technical users to manipulate. Investigation of speech-based user interfaces (both isolated or combined with visual interfaces) may also lead to fruitful solutions.

Finally, our system has been tested and benchmarked using Japanese Web page data, which consist of English and Japanese text. Extension of searches to multilingual data bases is an interesting topic for further study.

Acknowledgements: The authors would like to thank Professor Beresford Parlett for providing us with manuscripts of his work on the Lanczos algorithm prior to submission for publication.

[1] M. Baldonado, T. Winograd, "Sensemaker: an information-exploration interface supporting the contextual evolution of a user's interests", *Proc. ACM SIGCHI Conf. Human Factors in Computing Systems*, Atlanta, GA, March 1997, 11–18.

[2] D. Brake, "Lost in cyberspace", *New Scientist Magazine*, June 28, 1997: http://www.newscientist.com/keysites/networld/lost.html

[3] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, PA, 1997.

[4] S. Deerwester et al., "Indexing by latent semantic analysis", *Journal of the American Soc. Info. Science*, 41, 1990, 391–407.

[5] G. Dupret, M. Kobayashi, "Information retrieval and ranking on the Web: part I", *IBM Research Report*, RT0300, 1999.

[6] Graphics, Visualization, and Usability Center, Georgia Inst. of Tech., *User Surveys*: http://www.gvu.gatech.edu/user_surveys/

[7] M. Hearst, "Interfaces for searching the Web", *Scientific American*, March 1997, 68–72.

[8] P. Kahn, "Mapping Web sites", *Seminar Notes*: http://www.dynamicdiagrams/seminars/mapping/maptoc.htm

[9] O. King, M. Kobayashi, "Information retrieval and ranking on the Web: part II", *IBM Research Report*, RT0298, 1999.

[10] M. Maybury, W. Wahlster (eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufmann, San Francisco, CA, 1998.

[11] M. Morohashi et al., "Information outlining", *Proc. Int'l. Symp. on Digital Libraries*, Tsukuba, Aug. 22–25, 1995, 151–158.

[12] B. Parlett, *The Symmetric Eigenvalue Problem*, SIAM, Philadelphia, PA, 1998.

[13] B. Parlett, I. Dhillon, "Fernando's solution to Wilkinson's problem: an application of double factorization", *Linear Algebra and Applications*, 267, 1997, 247–279.

[14] B. Parlett, D. Scott, "The Lanczos algorithm with selective orthogonalization", *Mathematics of Computation*, 33, 1979, 217–238.

[15] R. Rao et al., "Rich interaction in the digital library", *Comm. ACM*, 36, Apr. 1993, 29–39.

[16] M. Sahami et al., "SONIA: a service for organizing networked information autonomously", *Proc. Digital Libraries '98*, Pittsburgh, PA, June 23–26, ACM Press, NY, 200–209.

[17] T. Sakairi, "A site map for visualizing both a Web site's structure and keywords", *Proc. 1999 IEEE Systems, Man and Cybernetics Conf.*.

[18] D. Shin, H. Jang, H. Jin, "BUS: an effective indexing an retrieval system in structured documents", *Proc. Digital Libraries '98*, Pittsburgh, PA, June 23–26, ACM Press, NY, 235–243.

[19] K. Takeda, H. Nomiyama, "Site outlining", *Proc. Digital Libraries '98*, Pittsburgh, PA, June 23–26, 1998, ACM Press, NY, 309–310.

[20] J. Tatemura, Y. Ogawa, "DocSpace: an interactive document space visualizer that combines querying and navigation", *Proc. Int'l. Symp. Research, Dev't and Practice in Digital Libraries*, Tsukuba, Nov. 18–21, 1997, 107–114.