Fluid Interface for Personal Digital Libraries

L.E. Good, A.C. Popat, W.C. Janssen, and E.A. Bier

Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, California 94304
{good,popat,janssen,bier}@parc.com

Abstract. An advanced interface is presented for fluid interaction in a personal digital library system. The system employs a zoomable planar representation of a collection using hybrid continuous/quantum treemap visualizations to facilitate navigation while minimizing cognitive load. The system is particularly well suited to user tasks which, in the physical world, are normally carried out by laying out a set of related documents on a physical desk — namely, those tasks that require frequent and rapid transfer of attention from one document in the collection to another. Discussed are the design and implementation of the system as well as its relationship to previous work.

1 Introduction

The persistence of paper as a preferred medium for document interaction, in the face of distinct and growing advantages of electronic representations, has in recent years been well-studied [1]. Perhaps nowhere are the advantages of paper over extant electronic alternatives so strong as in the case where the reader wishes to compare a set of passages appearing in several documents, or to cycle attention among passages from several documents in rapid succession. Such patterns of interaction arise naturally during a variety of knowledge tasks, including the preparation of lectures, the reviewing of papers, the analysis of intelligence briefings, and the writing of a report, among many others. We shall refer to such patterns as "reading from multiple sources," with "reading" to be taken in its general sense to include such activities as annotation, extraction, summarization, and the like.

We consider reading from multiple sources in the context of a personal digital library. By the qualifier "personal" here we mean that the user (1) already has the right to use all of the data objects in the library, and (2) already has local possession of those objects. An example of such a personal digital library is the result of scanning one's office filing cabinet onto a local hard drive.

In recent years, the importance of effective visualization tools and visual interfaces to digital library collections has been recognized [2–4]. In the case of a

personal digital library, an opportunity exists to blur the traditional separation between searching for materials and using them. The system presented in this paper pursues this goal.

In the physical world, reading from multiple sources often involves spreading out a set of documents on a large desk or table. During the course of the activity the reader is at various times absorbed in the close study of one document or another, while at other times, be it in transition or in respite, the reader broadens his or her focus momentarily to regard the workspace as a whole. Fluid transitioning among these modes of attention is achieved, and the continuity of orientation maintained, by virtue of the persistence of the layout of the physical documents in a single visual field, and through the ability of the reader to adjust his or her center and field of attention rapidly and at will.

The system described in this paper is an attempt to simulate and, at least with respect to navigation, surpass in the electronic realm the affordances of reading from multiple paper sources. A uniform and consistent interaction interface is provided to every document in a zoomable virtual workspace, irrespective of its native electronic "format" or "type." This universality is enabled by a fast, transparent, and automatic means of converting arbitrary documents of interest into a common intermediate working representation. To support the navigational cues that get built up in spatial memory during the course of the activity; advanced visualizations known as *treemaps* are used in the system's interface. Provision is made for navigation by interactive incremental textual search.

1.1 Universality of Representation and Consistency of Interface

Phelps and Wilensky [5] note that "picking a format is often tantamount to choosing a browser/editor/viewer with its packaged bundle of features and limitations." The diversity of applications required to interact with different document types leads to a proliferation of user interfaces, placing the additional burden on the user of having to learn and remember their proper operation.

Personal digital library systems can mitigate these problems by allowing documents to be converted into a common intermediate format for purposes of indexing and viewing. Reading applications need then support only this single common format. Because the addition of new information sources is a frequent and integral part of the reading activity, conversion into this common format — including the extraction and processing of any requisite metadata, must be as fast, transparent, and automatic as possible.

1.2 *UC*: a System for Fluid, Seamless Interaction

This paper describes a system for reading from multiple documents which addresses the above mentioned shortcomings of current digital reading systems. The system, called *UC* (a name deriving from its initial but no longer used internal name "UpLib Client") is built on the *UpLib* personal digital library platform [6], which provides an extensible, format-agnostic, and secure document

repository layer. *UpLib* achieves a degree of universality in accommodating multiple document formats through heavy reliance on two principal transductions of the original document, one into the *image* domain and the other into the *text* domain. From the perspective of *UC*, the image projection facilitates simulation of the visual experience of reading from multiple paper documents, while the text projection enables search-based navigation. By virtue of its leverage of *UpLib*, incorporating a new document into the *UC* system becomes largely a matter of producing its two fundamental projections. To accomplish this in a manner that imposes the least amount of latency and cognitive load on the user involved in a reading task, an area of the user's file system is designated for scanning and processing, so that any document placed there is automatically incorporated into the workspace.

For each document in the workspace, the *UC* system uses an instance of a single consistent reader application, *ReadUp*, described in detail in a parallel submission to the present conference. Despite the compositional relationship between *UC* and *ReadUp*, the user experiences fluid interaction within what appears to be a single application. This is accomplished by making every document behave as if it were always open in a separate instance of *ReadUp*, independent of whether it is an object of current focus. Thus, from the user's point of view, there is no need to launch a separate viewer application to interact with a document; one merely zooms, pans, and focuses on a document to interact with it.

The remainder of this paper is arranged as follows. Section 2 summarizes relevant prior work. Section 3 motivates key design aspects of *UC* by considering those affordances most valuable when reading from multiple documents. A detailed description of the *UC* is provided in Section 4. Section 5 lists future work and Section 6 concludes.

2 Prior Work

The importance of effective visual interfaces to collections has been recognized by a number of researchers [2–4], but in the context of remote digital libraries where finding and using materials are constrained to be separate activities.

Several previous systems have used treemaps and zoomable user interfaces (ZUIs) similar to those described in this paper. The original implementation of this idea was in the PhotoMesa image browser [7]. Many of the principles driving our system design are derived from this work. The International Children's Digital Library (ICDL) [8] builds on PhotoMesa by laying out children's book cover thumbnails in zoomable treemaps. Our work differs from the ICDL in several respects, including the use of hybrid treemaps, the specific modes of navigation supported, the ability to generate the collection automatically, and the accommodation of large, heterogeneous collections.

Scatter/Gather [9] and Visual Relevance Analysis [10] both provide visualization of document collections based on interactive clustering techniques, and the latter presents an interactive treemap visualization of topics. Unlike the present

work, neither attempts to support document use (i.e., reading, annotation, etc.) as an integral part of the activity.

The *Web Book* and *Web Forager* [11] uses a 3D book metaphor to organize collections of web pages into a three level hierarchy for navigation and reading. Our work differs in that it provides visualizations of multiple collections simultaneously. It is also not limited to web pages or to a fixed hierarchy. *Data Mountain* [12], another technique for interacting with collections of web pages, provides a planar 3D workspace in which the user can manually organize web page thumbnails. In contrast, our techniques provide automatic visualization layouts, overviews of multiple collections, and support for a variety of document types.

Much of the prior work on electronic reading has focused on reading individual documents. *Document Lens* [13] presents a fisheye view of a grid of thumbnails of a document's pages. *XLibris* [14] focuses on providing for freeform annotations on an electronic document. The *Multivalent Browser* [5] attempts to create a standardized reading and annotation platform independent of a document's originating format; our work extends this idea in dealing with document collections. In contrast to the above cited work, we focus on multi-document reading tasks where the ability to transition from collection-browsing and overview mode to detailed comparison of the contents of the documents are typical and frequent operations.

3 Effective Interaction

3.1 Fluid Reading

Current document browsers and reading tools have overhead that can be cognitively disruptive for reading. One limitation in these systems is that not all document types are supported by the same reading application. As a result, not all documents support the same functionality, such as thumbnail overviews, highlighting, freehand annotations, text notes, or keyword search. Moreover, even when similar functionality is available in a document's native applications, the user interface controls to that functionality are not standardized.

A related limitation is that document finding and browsing is divorced from document reading. As a result, reading a document typically involves opening a new application external to the one used to find it. This act of opening an application often introduces small delays both for the system to respond and for the user to orient to the change. Moreover, if the new application is displayed in its own window, it can also introduce window management problems. These issues substantially increase the cognitive load of reading.

A more subtle effect of the separation between browsing and reading is that the reader is often forced to prematurely commit to spending time with a document that turns out not to be most relevant. Applications for finding and browsing typically display a fixed amount of information about the document so the user must make a decision whether to incur the costs described above based on limited information.

3.2 Multi-Document Reading

Because document finding is typically separated from document reading, multiple document reading tasks are particularly difficult in the electronic realm. Many applications provide facilities for navigating between multiple documents of the same type, such as tabs in a web browser. However, reading tasks may involve moving between several types of documents such as web pages, word processing documents, presentation slides, etc. This type of task suggests the need for techniques to transfer attention rapidly among documents, regardless of type, in small working sets.

A related problem is being able to view the pages of multiple documents at once. With paper, this is often accomplished by spreading out documents on a table. Reading applications often provide thumbnail facilities for viewing multiple pages from a single document but not from multiple documents. The ability to do so can be important for making comparisons [15] and for getting an overview of a collection of documents [16].

3.3 Finding Documents

Perhaps the most widely accepted method of finding relevant documents is keyword search. This method has a number of limitations in the case of personal digital libraries. Keyword search is fundamentally a recall-based user interface. A classic principle in user interface design is to minimize the memory load on the user [17]. This is often referred to informally as "supporting recognition rather than recall."

A general problem with keyword search is that a user's vocabulary often does not match the desired document's vocabulary. This means that users may not be able to find certain documents or they may have to experiment with several queries before they find what they want. A number of techniques have been introduced to deal with this problem such as term aliasing or personalizations like those in Haystack [18]. Nevertheless, these heuristics are not likely to work in all cases. It is also often the case that the user is not able to formulate a query at all. Some examples include searching for a picture, a specific page layout, or some other visual property of a document. In these cases, the user needs techniques for interacting with documents visually rather than textually.

4 The *UC* System

The *UC* system developed in our research group integrates a number of recent user interface, information visualization, and digital library techniques with the goal of addressing the problems described in the foregoing. The system is also designed to work well with pen-based tablet computers where traditional user interface controls, such as scrollbars and text boxes, are of limited effectiveness. While the foregoing sections have described the motivation and design rationale for this system, the present section and the next describe the system in detail.

UC is in part a user interface for interacting with documents in an *UpLib* [6] repository. As such, it is able to connect to existing *UpLib* repositories, such as those that have been manually constructed. However, it also provides a powerful facility for finding documents in the user's file system and automatically adding them to the repository.

4.1 Continuous and Quantum Treemaps

UC uses continuous and quantum treemap [19] layouts to present collections of documents. Continuous treemaps are space filling visualizations of trees that assign area to tree nodes based on the weighting of the nodes. In continuous treemaps, the aspect ratio of the cells is not constrained even though square cells are often preferred. Quantum treemaps extend this idea by guaranteeing that cell dimensions are an even multiple of a unit size. These layouts are described in more detail by Bederson et al. [19].

The page thumbnails used in *UC* are based on document icons created by the *UpLib system.* In addition to portraying the often unique appearances of the documents themselves, these icons can also be augmented with overlays such as important text or pictures to further assist users in differentiating one document's icon from another. These unique document icons serve to enhance spatial memory of document locations within *UC*'s spatial treemap layouts.

Users often collect large numbers of documents that are more than can be meaningfully displayed using these thumbnail based visualizations. As a result, the system has a threshold number of documents above which thumbnails are no longer displayed and standard continuous treemaps are shown instead. The continuous treemap attributes of cell size and color can then be used to display quantitative information about the underlying collections such as number of documents, number of pages, file sizes, last modified or last viewed dates, alphabetic ordering, values for manually assigned metadata, or other semantic properties extracted from the documents. Table 1 lists some common operations in *UC* and their allowable starting and ending treemap types. When an operation supports both treemap types, the system chooses between them based on the number of documents in the current view.

A number of algorithms can be used to arrange continuous treemaps. Our system supports both squarified and strip treemap layouts. Because our data is typically ordered by date, file name, etc., the default layout is a strip treemap. Bederson et al. present evidence that the strip layout is preferable over other available treemap algorithms for presenting ordered data [19]. This layout also has reasonable behavior as documents are added to collections or the window aspect ratio changes.

4.2 Navigation

An important aspect of the interface is the fluidity of navigation. This allows the user to focus on the documents rather than on interacting with the tool. In *UC*, the navigation controls are similar to those in other ZUIs. Left click on an

	Zoom Out	Zoom In	In Context Search	Limit View to Search Results	Explode to Pages	Read Document
C → C	×	×	×	×		
C → Q		×		×		
Q → C	×					
Q → Q	×	×	×	×	×	×

Table 1. The available transitions between the two types of treemaps. In the row header,"C" represents continuous treemaps and "Q" represents quantum treemaps. An "X" in a cell indicates that the column's operation can start in the first treemap type and end in the second.

object or group of objects zooms in and either button clicked on the background zooms out.

One problem that arises from combining a zoomable user interface with continuous treemaps involves conflicts with aspect ratios. The cells in continuous treemaps have a range of aspect ratios, as demonstrated in Fig. 1. Each of these different aspect ratios may differ from that of the view window. As a result, zooming in on these cells, as you might zoom into a country on a map, may not increase the amount of screen space devoted to a cell. For example, if a cell consumes the full width of the window but only half the window's height, then there is no way to increase the size of the cell while keeping it entirely on screen.

There are two solutions to this problem. The first is to zoom in so that some of the cell is off the screen. In the previous example, this would mean zooming in to the point where the cell's height is equal to the window's height and the cell's width is twice the window's width. The advantage of this approach is that it preserves the kind of standard geometric zooming used in many other systems. The disadvantage is that the change in scale needed to zoom a cell is essentially unlimited. In our previous experience with ZUIs, we have found that this type of large change in scale can be disorienting even when the transition is smoothly animated.

The second solution, and the one used in *UC*, is to zoom and morph the cell to the window size and aspect ratio while leaving the rest of the layout in place. The primary advantage of this approach is that it minimizes the visual disturbance of the display since only a single cell moves. Animating the transition can further help orient the user during the change.

The continuous treemap views also provide previews of the layout at the next level down when the user moves the mouse into a cell. One bit of information this preview provides is whether the layout at the next level down is a continuous treemap or a quantum treemap. The preview can also give a rough idea of the number and structure of groups at the next level.

Importantly, the quantum treemaps in *UC* are navigated with standard viewpoint animations while the document layouts remain static. This allows the user

to build awareness and memories for spatial relationships within smaller working sets of documents.

4.3 Refining and Searching Collections

UC provides controls to refine which documents are displayed. First, the interface provides mechanism to search for specific content within the documents. The system incrementally highlights matching documents as letters are added to the search query to immediately indicate matching documents. The user can also choose to update the view to re-layout with only documents that match the current query. An example of this type of search, ending in reading a document, is shown in Fig. 1.

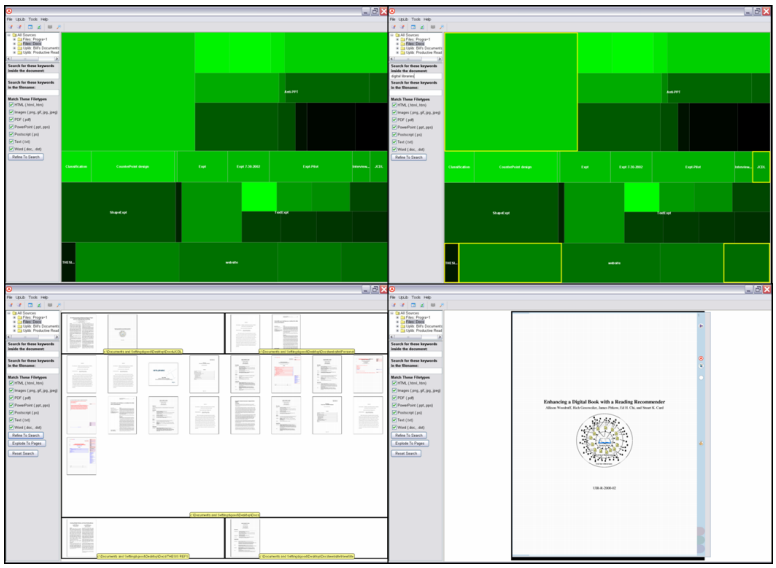

Fig. 1. An example keyword search scenario in *UC*. The scenario begins with a continuous treemap visualization of a document collection (top left). As the user types search terms (top right), interactive highlights appear for groups with matching documents. The user presses a button to limit the view to only matching documents (bottom left). Finally, the user zooms in on a document and begins reading with the *ReadUp* reader (bottom right).

For *UpLib* repositories generated from a file system, the interface also provides controls for finding patterns in the file path name and for limiting which file types are displayed. We also plan to generalize this to include controls for limiting file size, number of pages, last access or modification times, and other types of document metadata.

4.4 Explode to Pages

For multiple document reading, users need a mechanism to easily make comparisons between multiple documents. This is accomplished in the *UC* by allowing the user to explode a set of documents into their pages. As with collections of document icons in *UC*, these page thumbnails are laid out in quantum tree maps. Using an identical layout at both the document and individual page level allows the user interface to provide a consistent set of functionality and interaction techniques between the two types of views. These exploded document layouts not only support comparisons between pages of multiple documents, they can also provide overviews of collections and a mechanism for quickly jumping between pages in multiple documents. Fig. 2 shows the layouts in use.

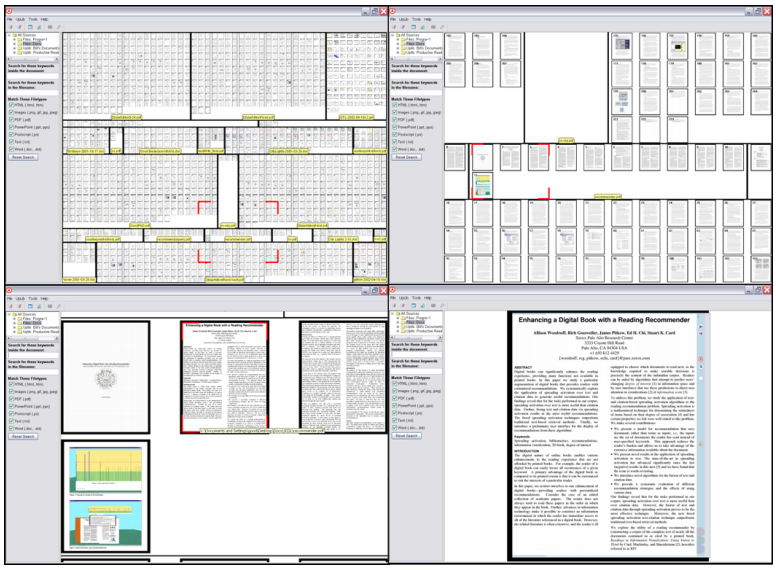

Fig. 2. An example of exploding a document collection to pages. A working set of documents is obtained through a search (top left). Pressing a button explodes the documents to pages. The user zooms in on the pages of several interesting documents (top right). The user continues zooming in to a portion of a single document (bottom left). The user then selects a document page and begins reading at that page in *ReadUp* (bottom right).

4.5 Integrated *ReadUp* Reader

An important principle in the system design is that reading be fluid and not require opening a separate application. This is achieved in *UC* by integrated the *ReadUp* document reader provided by the *UpLib* system. This reader provides

thumbnail overviews, freehand pen annotations, highlighting, text sticky notes, bookmarks, and full text keyword search.

Annotations made in the reader are automatically stored in the same *UpLib* repository that stores the image and text projections. This allows the user to fluidly read and annotate documents without having to manage annotated files or explicitly save changes. The user's activity, such as how long the user viewed a particular page, is also stored in the *UpLib* repository which can be used to inform the visualizations provided by *UC*.

4.6 Implementation Issues

UC is a Java application that connects to one or more *UpLib* repositories as a network client. The *UpLib* server also runs an extensible set of document analysis and processing operations on each document when it is acquired. These operations extract metadata, create various thumbnail versions of each page image, and perform full-text indexing. The *UC* system requests information from the server, such as thumbnails, word bounding boxes, or document metadata, to enable interaction with documents in the repository. *UC* also includes a utility to scan a portion of the file system specified by the user. This scanner then adds supported document types that it finds to a specified instance of an *UpLib* repository. *UC* extends the set of document types handled by the *UpLib* system to include those produced by popular word-processing, spreadsheet, and presentation applications.

5 Future Work

In this paper we have described the design rationale, feature set, and implementation of *UC*, but not its evaluation in controlled user trials. User evaluation is essential to fully understand the benefits of the approach and to identify weaknesses and opportunities for improvement. A complicating factor in designing such a user study is that the time scale of the sort of multi-document reading tasks for which this system is intended can span days and weeks, involves the interaction of many factors, and is subject to a great deal of variability in both the methods currently employed and their effectiveness. The common expedient of devising a small, artificial task on which to compare and report statistical p-values has become *de rigueur* in the user-interface research community, but likely would not lead to much insight. Instead, working with our colleagues we intend to undertake a descriptive ethnographic study in an authentic multi-document reading setting.

UC's user interface currently supports refineable automatic treemap layouts. We would also like to look at combining these automated layouts with user defined layouts such as those in Data Mountain [12] and VKB [20]. The automatic layouts could then bootstrap the users manually constructed working sets for particular tasks.

6 Conclusion

This paper has described an advanced visual interface system for navigating within a personal digital library and interacting with its contents in a fluid and consistent manner. The system is based on a zoomable planar representation of the collection in which sub-collections and individual items become accessible as the scale is varied. In contrast to most existing collection visualization and management systems, provision is made for interacting with elements of the collection seamlessly and *in situ,* without having to open a separate application. In contrast to most document reading interfaces, the system facilitates rapid and fluid movement from one document to another document, again without leaving the overarching spatial representation of the collection, thus preserving a sense of orientation and reinforcing spatial memory throughout the course of interaction. In addition, the system streamlines the incorporation of new documents within the personal digital library by automating the conversion of those documents into a common compatible format.

If accepted for publication, an extensive demonstration of the *UC* system will be provided at the conference.

7 ACKNOWLEDGMENTS

This research has been funded in part by contract #MDA904-03-C-0404 to Stuart K. Card and Peter Pirolli from the Advanced Research and Development Activity, Novel Intelligence from Massive Data program.

References

1. Sellen, A.J., Harper, R.H.: The Myth of the Paperless Office. MIT Press (2003)
2. Chang, M., Leggett, J.J., Furuta, R., Kerne, A., Williams, J.P., Burns, S.A., Bias, R.G.: Collection understanding. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 334–342
3. Perugini, S., McDevitt, K., Richardson, R., Perez-Quiñones, M., Shen, R., Ramakrishnan, N., Williams, C., Edward A. Fox, .: Enhancing usability in citidel: multimodal, multilingual, and interactive visualization interfaces. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 315–324
4. Sumner, T., Bhushan, S., Ahmad, F., Gu, Q.: Designing a language for creating conceptual browsing interfaces for digital libraries. In: JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries, IEEE Computer Society (2003) 258–260
5. Phelps, T.A., Wilensky, R.: The multivalent browser: a platform for new ideas. In: DocEng '01: Proceedings of the 2001 ACM Symposium on Document engineering, ACM Press (2001) 58–67
6. Janssen, W.C., Popat, K.: Uplib: a universal personal digital library system. In: ACM Symposium on Document Engineering. (2003) 234–242

7. Bederson, B.B.: Photomesa: a zoomable image browser using quantum treemaps and bubblemaps. In: UIST '01: Proceedings of the 14th annual ACM symposium on User interface software and technology, ACM Press (2001) 71–80
8. Druin, A., Bederson, B.B., Weeks, A., Farber, A., Grosjean, J., Guha, M.L., Hourcade, J.P., Lee, J., Liao, S., Reuter, K., Rose, A., Takayama, Y., Zhang, L.: The international children's digital library: Description and analysis of first use. First Monday **8** (2003)
9. Cutting, D.R., Pedersen, J.O., Karger, D., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. (1992) 318–329
10. Pediotakis, N., Hascoët-Zizi, M.: Visual relevance analysis. In: DL '96: Proceedings of the first ACM international conference on Digital libraries, ACM Press (1996) 54–62
11. Card, S.K., Robertson, G.G., York, W.: The webbook and the web forager: an information workspace for the world-wide web. In: CHI '96: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press (1996) 111–ff.
12. Robertson, G., Czerwinski, M., Larson, K., Robbins, D.C., Thiel, D., van Dantzich, M.: Data mountain: using spatial memory for document management. In: UIST '98: Proceedings of the 11th annual ACM symposium on User interface software and technology, ACM Press (1998) 153–162
13. Robertson, G.G., Mackinlay, J.D.: The document lens. In: UIST '93: Proceedings of the 6th annual ACM symposium on User interface software and technology, ACM Press (1993) 101–108
14. Price, M.N., Schilit, B.N., Golovchinsky, G.: Xlibris: the active reading machine. In: CHI '98: CHI 98 conference summary on Human factors in computing systems, ACM Press (1998) 22–23
15. Adler, A., Gujar, A., Harrison, B.L., O'Hara, K., Sellen, A.: A diary study of work-related reading: design implications for digital reading devices. In: CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM Press/Addison-Wesley Publishing Co. (1998) 241–248
16. O'Hara, K.P., Taylor, A.S., Newman, W.M., Sellen, A.: Understanding the materiality of writing from multiple sources. Int. J. Hum.-Comput. Stud. **56** (2002) 269–305
17. Shneiderman, B.: Designing the user interface: strategies for effective human-computer interaction. third edn. Addison-Wesley Longman Publishing Co., Inc. (1998)
18. Adar, E., Kargar, D., Stein, L.A.: Haystack: per-user information environments. In: Proceedings of the eighth international conference on Information and knowledge management, ACM Press (1999) 413–422
19. Bederson, B.B., Shneiderman, B., Wattenberg, M.: Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. ACM Trans. Graph. **21** (2002) 833–854
20. Shipman, F.M., Hsieh, H., Moore, J.M., Zacchi, A.: Supporting personal collections across digital libraries in spatial hypertext. In: JCDL '04: Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries, ACM Press (2004) 358–367