Exploratory Navigation in Large Multimedia Documents using Context Lenses

Andreas Dieberger, Daniel M. Russell
IBM Almaden Research Center, USER Group
650 Harry Road
San Jose, CA 95120
{andreasd, daniel2}@us.ibm.com

Abstract

The Context Lens (CL) is a focus+context visualization and navigation tool particularly suited for navigating large documents, or collections of documents. Context Lenses have been applied successfully to navigating Web pages, video collections and slide presentations. In this paper we discuss our experiences both with linear as well as with hierarchical Context Lenses. We focus on the use of information scent in the hierarchical Context Lens, and on supporting an exploratory navigation style in documents.

Exploratory navigation is supported by the fact that a CL delays commitment in the user interface through an interaction style called brushing. Brushing lowers the cost of exploring a section of the document and therefore does not only support, but actively encourage an exploratory navigation style. This aspect of Context Lenses is especially valuable for navigation of large amounts of media data on low bandwidth devices, such as wireless PDAs.

1. Overview

A Context Lens (CL) is a navigation tool for data, such as Web pages, videos or Web server log files. It provides a global overview of the document using an abstract visualization of the relevant content. This visualization, in combination with an interaction style called *brushing* allows users to glance at detailed information at a very low overall cost [11]. As the cost -- in terms of user time and number of user actions -- is low, the CL promotes exploring the data set in various places and thus promoting an exploratory style of navigation.

The Context Lens is designed to keep the relationship between part and whole visible while exploring sections. The visualization of the document as a whole provides global context while a section can be displayed in a focus view. Therefore, a Context Lens is a focus+context visualization technique.

Navigation using a Context Lens is typically top-down: from a larger-scale context or segment to a smaller one. Top-down navigation approaches often have a serious disadvantage; it is difficult to decide where to drill deeper into the data. This problem is most often caused by having condensed information available at the top-level views. While this is a traditional way to organize large documents and collections, it does not afford easy viewing of the desired information in context. Information scent, introduced in Information Foraging Theory [7, 8] is residual information that can guide users towards information they are interested in. Thus, mechanisms that improve the availability of scent to be available quickly and rapidly are very useful.

As we describe in more detail below, a Context Lens shows an abstraction of a base document based on what the CL assumes is of interest to the user. The hierarchical Context Lens displays either a very long document or a collection of documents by condensing information of several Context Lenses into a higher-level overview CL. This overview contains sufficient residual information of the lower level CLs so that users can decide which part of the document to explore further.

2. The Context Lens and its variants

In this section we describe the development of several variants of Context Lenses. We focus on what we learnt in each step and outline the design rationale for each modification.

2.1. The Horizontal Context Lens

The Context Lens project started with an idea to check Web pages for keywords a user is interested in and to provide information about these keywords at the top of the page. In our first attempts this information consisted of a list of keywords with links to occurrences in the document. These lists were either inserted at the top of the Web page or displayed in a separate window. In the latter case, the keywords were shown with a few words of immediate context (concordances). Naturally, this solution proved inadequate. We decided to provide a more compact, interactive visualization of the occurrence of keywords using a Java applet embedded in the page.

Our first visualization was inspired in part by Marti Hearst's Tilebar interface [5]. A Tilebar visualizes the results of a database query and shows where key terms occur in the result set of documents. Like the Context Lens, the Tilebar uses a segmentation of the document, displayed as a grid of segments and keywords. Grid cells indicate the presence of specific keywords in a section using gray levels. Tilebars are arranged horizontally. The beginning of each document shown is visualized at the left end of the Tilebar. Several Tilebars, one below the other, allow users to compare a number of documents found in the search.

Superficially, the horizontal Context Lens appeared very similar to this Tilebar interface, as it used a horizontal grid of cells as well. However, the Context Lens visualizes only a single document. Furthermore, the Context Lens introduced the concept of *brushing*: when moving the mouse pointer over the Context Lens, additional information about grid cells is displayed, whereas a Tilebar is a static visualization.

Brushing over a grid cell indicating that a keyword was found in that section triggers display of a *focus window*, showing the keyword in the immediate context within the document. In Figure 1, the dark cells indicate that a certain keyword was found. For example, the term "Squeak" was found in a middle section of the document. The focus view, displayed on top of the CL shows this keyword in its immediate context within the document.

Additionally, during brushing, the CL shows the start of each section, as the beginning of a paragraph often contains key information. In the original design, this information is shown below the CL. At no point does the Context Lens display the base document as a whole. This way, brushing allows users to study relevant parts of the document without actually opening the document.

This aspect of the CL makes it very interesting for wireless devices with limited bandwidth. Information contained in a CL, including the content of all possible focus windows is much smaller than the amount of information contained in the base document. This is especially true for documents including multi-media data.

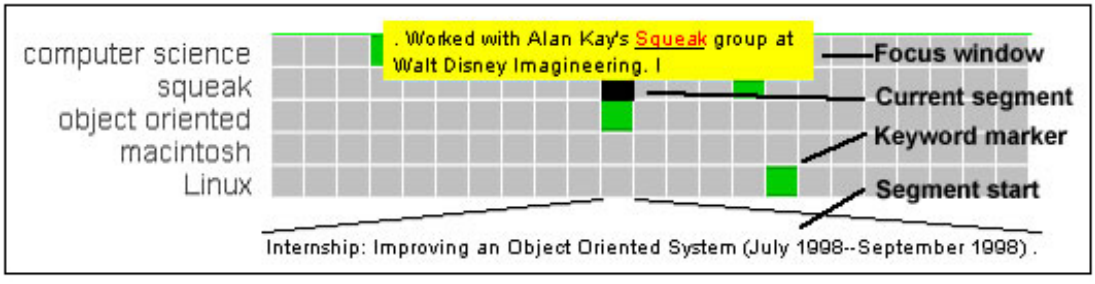

Figure 1 – The Horizontal Context Lens: Columns in the grid represent sections or the document and rows represent keywords. A grid cell is marked if that section contains a particular keyword. Brushing over a cell displays the keyword with its immediate context in the box above the cell. Below the grid, the start of that brushed section is shown for additional context.

Users can explore the document using the small amount of information contained in the Context Lens and then download only those sections of the document that are indeed relevant for them.

This scenario requires an intermediary to handle access to the base document. An intermediary [6] is a piece of software, which provides services for a client. In this case, it provides the data in the CL and access to sections of the base document when needed. As discussed in [4] we found that the information in a CL often is sufficient to meet users' information needs, which may eliminate the need to access the document as a whole.

2.2. The Vertical Context Lens

The horizontal layout of the original Context Lens does not directly relate to the vertical layout in most document types. Therefore, we soon switched to the current vertical organization of Context Lenses.

In the vertical CL keywords are represented in columns and document sections in rows. In order to make the CL scale to a larger number of sections, the grid cells of the vertical CL are drawn smaller, which makes it difficult to brush over single grid cells. We addressed this problem by brushing only entire sections, instead of single cells. The focus window now shows all relevant keywords of a section together. This design offers the additional advantage of allowing users to examine several concordances side by side, as shown in Figure 2. Like in the horizontal CL, the focus view shows the beginning of a section. Due to the much narrower arrangement of the CL, we designed the focus view as a floating window. This way we do not obstruct the CL during brushing and keep the size of the CL very small when the focus view is closed.

We used the vertical CL not only for navigating Web pages, but also in a Slide presentation tool and for navigating digital video. When we started working with video we realized that the concept of keywords was impractical. Instead, we refer to the columns of a CL as *features*. A feature can represent any kind of meta-data about a document segment. In the case of the Slide presentation tool, a feature shows whether that slide has been presented already or not [3]. In a video CL features can indicate the presence of a sound track or subtitles or show that a certain keywords was found in a textual transcript of the video sound track [9].

Figure 2 – The Vertical Context Lens: Rows represent segments and columns represent keywords. As the rows are smaller, brushing provides information for an entire segment. In this example, we are brushing a section containing 4 keywords. On top of the context window the CL shows the start of the segment for additional context. Note the keyword labeling in the CL. Also note that the CL is shown side by side with the base document in this case. The focus window floats on top of the Web page, which allows us to keep the CL very small.

2.3. The Hierarchical Context Lens

Although the vertical Context Lens provides immediate access to a large number of segments, it does not scale well enough for very large documents. We address this issue with the hierarchical Context Lens. It segments potentially very large documents into large-scale sections, to be brushed using a Context Lens. The focus view of this high-level CL contains a lower-level CL representing that segment split into much smaller segments. The focus view of the secondary CL provides access to elements in the base document, or shows yet another Context Lens.

We use the Hierarchical Context Lens to navigate collections of digital video. A screenshot of our system is shown in figure 3. It allows users to explore a video documentary consisting of 5 episodes of one hour each. The first level CL is segmented into 5 sections. On the second level, each of these 60-minute long videos was segmented into sections of 2 minutes. The third level of this Context Lens shows a thumbnail of the video segment with a snippet from the textual transcript. For more details of this system see [9].

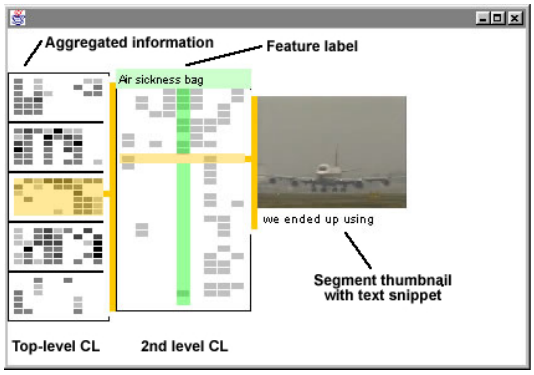

Figure 3 – Hierarchical Video Context Lens containing 5 hours of video: The user here brushes over the 3rd segment in the top level document, which opens the CL for that section. It contains 30 2-minute segments of video (totaling 1 hour). When brushing over one of these segments, the corresponding thumbnail and a snippet of the textual transcript is shown. Clicking that segment starts playing the video from this point. Note the visual similarity between the 2nd level CL and the top-level representation of the third segment.

A hierarchical Context Lens's utility depends largely on the representation of the higher level segments. This representation needs to contain sufficient information to allow users to decide which of the lower-level segments to study in more detail. Even with the low-cost operation of brushing, users do not want to look at every single segment to determine where interesting information is hidden. By making *information scent* of lower levels available in the top level, we provide guidance for users exploring the Hierarchical CL. Information scent was introduced by Pirolli in information foraging theory, see [7, 8]. It is residual information about items that are not directly visible from a current viewpoint in an information space. Information scent can guide a user towards this distant information. Information Scent becomes even more essential in hierarchical CLs containing more than just 2 levels.

Information scent for a Context Lens can be determined in a variety of ways. Currently we use a simple, but surprisingly powerful scheme to aggregate features of several segments into one scent marker. The scent markers in the higher levels use gray levels to indicate how many lower level segments contain a particular feature. Individual features stay separated in this process.

For example, the top-level segments in Figure 3 represent one hour of video each. For each of these segments, the second levels contain 30 segments of 2 minutes each. We treat every feature as a value of 0 or 1. By adding up 5 such values, we obtain a number in the range [0..5], representing one feature over 10 minutes of video. We visualize these values as gray levels. 6 of these representations, taken together, represent one hour of video in the top level CL. Even using this highly aggregated representation it is easily visible, that a few features, which occur frequently in the 2nd and 4th video, are absent in the other videos (see figure 3).

The simple aggregation technique works surprisingly well in a two level video context lens. For Context Lenses with more levels more sophisticated aggregation strategies might need to be developed (future work.)

3. How does the CL differ from other interaction styles?

The Context Lens combines two kinds of data manipulation in one interaction: the CL visualization provides a visual filter for search results, based either on pre-selected keywords or

on features of general interest in a certain task domain. The user interacts with this visualization using brushing to access detailed information about specific segments. During brushing only one section of the document is revealed in full detail. Because of this focused access to a section it is possible to maintain global context.

The best method to segment documents into sections depends largely on the application domain and the type of document. In the case of the Web CL, segmentation into paragraphs on the Web page proved most useful. In the video CL we used a time-based segmentation. An alternative would be to use a shot-boundary detection mechanism to determine topically coherent variable-size segments as described in [9].

We use the term *exploratory navigation* to describe that the Context Lens allows users to probe a specific section without losing global context. As it is easy to pull back from a section to probe elsewhere, users can rapidly explore a multi-level hierarchical information structure.

Brushing offers a number of obvious advantages over point-and-click selection navigation. Traditionally, a user would have to select a section (possibly by clicking on it), and wait for the data to be shown. In a deeper hierarchical structure, several selections are required to reach the level of highest detail. Should the user decide to look at a different section, she would have to actively close the current view, or cancel the selection and then select another section. Alternatively, the user might be able to directly jump a few levels up and then probe deeper into the structure again.

A good example for this kind of interaction is the navigation model employed by many popular Web sites, such as www.yahoo.com. Such "browser style" navigation requires several selection actions (mouse clicks) and possibly a specific command to revert a selection (undo). This style of navigation can be found in many standard user interfaces where users have to click through a series of selection dialogs or open a series of folders to finally find the information they are looking for.

Brushing in a (hierarchical) Context Lens, instead, is a fluid activity, similar to interacting with a hierarchical popup-menu. The user does not actually *select* a section of the data. Instead she brushes a section, which causes the display of additional information about the section. Once a user decides she found the desired section, she can perform a selection (by clicking). In the Video Context Lens selecting a section starts playback of the video from that point. Alternatively, the user might never need to make any selection at all! As we pointed out, it is sometimes possible to retrieve the information desired from the focus information in the Context Lens alone. In such a case, a selection is never necessary.

We believe, that brushing is a qualitatively different interaction style than a series of selections in a hierarchical interface. The key difference lies in "delayed commitment in the user interface". Indeed, the Context Lens permits users to postpone commitment to a specific document section until they are certain they are interested in that section.

Note, that it is possible to perform computation on a partially committed selection during brushing. For example it is possible to build a CL, which permits users to compare two segments in the document, for example in the log data of a Web server. In this scenario, a user would first select one segment of interest, such as one particular hour. Then, when brushing the document again the focus view would show a comparison of the log contents between the previously selected hour and the currently brushed one.

Another, similar example would involve selecting a range of segments in the Context Lens to obtain statistics for these segments. The focus view for the selected or brushed range would show summary information describing that set of segments. One could argue that a lot of computing power is necessary to achieve this in a real-time. Instead, the Context Lens itself contains preprocessed information about its segments, typically summary information like the information we would compare in such a scenario. This information is the reason why it is often possible to retrieve information from a Context Lens without actually accessing the base document, as was described above. As long as the comparison of two segments uses only the information contained in the Context Lens itself, such a range brushing can be performed in real-time even on average computers.

Considering these arguments, we think that brushing in a Context Lens provides a number of

advantages over other selection and navigation techniques: due to delayed commitment in the user interface, brushing is a fluid activity. As the Context Lens does visualize information about the data segments, brushing goes far beyond the interaction in a hierarchical popup menu: every step in brushing does provide possibly useful information to the user. This fact makes it possible to sometimes find required information without ever making a selection in the CL. Due to the low interaction cost of brushing users are encouraged to explore documents to a larger extent than they might using other interaction styles.

4. Future work

Recently we began work on a new Context Lens which will contain the selection and analysis features we outlined in the previous section. In the case of web server logs, a statistical summary of segments is actually more interesting than the raw data itself. Rarely will the user of a server log context lens want to access the actual log file. Instead she will be interested in seeing if there were any anomalies in the average number of – say – bad requests to a server, as these might indicate a hacker attack or server malfunction. Additionally, the user of such a Context Lens will want to select time ranges and access statistical information about these ranges and compare them to other ranges etc.

Also in this scenario it is likely that users might want to define their own, custom made features to be shown in the context lens. We envision extending the Context Lens with the ability to generate new features based on combinations of existing features.

When several users study the same data set it might also be useful to incorporate collaborative features into a CL. For example when several system administrators use a CL to monitor a Web server and find an anomaly in the data, it would be sufficient if one of the administrators checks out this anomaly instead of all administrators studying the same data. Using an "already checked" feature it would be immediately obvious that certain sections of the log have been examined already. Such a feature would be similar to a "breadcrumbs" feature in hypertext systems, which show that certain links were used recently.

In the context of a collection of video data, such a feature might turn into a social navigation tool visualizing the popularity of sections in the video material. For more information on social navigation tools see [1, 2].

As we mentioned in the previous section, the CL contains a lot of aggregated data about the base document. This makes it possible to generate overviews of a range of segments while brushing. However, the data for such operations has to be available to the CL in some form already. An obvious extension for the CL would be to allow the dynamic definition of new features. In case a new feature can be computed out of features already contained in the CL, such features will be available immediately. However should such a feature require a complex search of the base document, such features will not be immediately available. Dynamically defined features of this kind might allow users to interact with a CL as a kind of graphical spreadsheet on document features. We believe such a CL, while quite a complex visualization tool, would be very valuable for analyzing process data, such as web server logs. It would allow users to manipulate the data in the CL in a variety of ways while maintaining the focus+context aspects of the tool. Another focus+context tool pursuing this goal is Inxight's Table Lens system, see also [10].

Our experiences with the Context Lenses we built demonstrated, that Context Lenses do have a certain learning curve associated with them. A hierarchical CL with several features in it provides a very dense representation of information, which is not intuitively understandable for a novice user. CLs with few features, such as the slide presentation CL, which contains only one feature, are much easier to understand than CLs with many features.

We do not see this as a serious problem as the Context Lens was designed as a tool for trained users. Even so, we are aware that a more detailed usability evaluation, especially of the hierarchical Context Lens, is still necessary. The focus of this evaluation will be on the aggregation feature we described above, which we think is critical for the utility of a hierarchical CL.

5. Summary

We described Context Lenses as a class of visualization and navigation tools for large documents or document collections. Context Lenses are focus+context visualizations as they provide a global overview of documents while at the same time providing access to detailed information about small-scale sections of the document under consideration.

Users interact with Context Lenses using an brushing interaction style: when rolling a mouse cursor over a section of the context lens, additional information on this section is temporarily overlaid on the base display. Brushing allows users to explore a CL in an exploratory navigation style: users can delay committing deeper investigation in a document section until they are certain they want to retrieve that section. As the Context lens displays snippets of information deemed most relevant for the user during brushing, it is often possible to retrieve the information required without actually accessing the base document itself. This is particularly valuable when accessing very large (multi-media) documents over a low bandwidth connection.

This scenario requires an intermediary pre-processing the document under consideration. The intermediary sends only the information required to create the Context lens to the client. Typically this amount of information is negligible compared to the amount of information contained in the entire base document.

We describe several different Context Lenses we built and discuss how the design of these CLs evolved out of the issues we encountered with earlier CLs. We also describe current work on a new Context Lens.

Acknowledgements

We wish to thank Cameron Miner, Dulce Ponceleon, and Shumin Zhai for valuable discussions and design input for various Context Lenses. We also would like to thank Eser Kandogan for comments on a draft of this paper.

6. References

[1] Dieberger, A. and K. Höök, Applying Social Navigation Principles to the Design of Shared Virtual Spaces, in Proc. of WebNet'99. 1999: Honolulu, HI.

[2] Dieberger, A. and P. Lönnqvist. Visualizing Interaction History on a Collaborative Web Server. Proc. of ACM Hypertext'2000. 2000. SanAntonio, TX: ACM Press.

[3] Dieberger, A., C. Miner, and D. Ponceleon. Supporting narrative flow in presentation software. CHI 2001 Extended Abstracts. 2001. Seattle, WA: ACM Press.

[4] Dieberger, A. and D.M. Russell. Context Lenses -- Document Visualization and Navigation Tools for Rapid Access to Detail. in Proc. of Interact 2001. 2001. Tokyo, Japan: IOS Press.

[5] Hearst, M.A., TileBars: Visualization of Term Distribution Information in Full Text Information Access, CHI'95. 1995, ACM Press: Dever, CO. p. 59-66.

[6] Maglio, P. and R. Barrett, Intermediaries Personalize Information Streams. Communications of the ACM, 2000. **43**(8): p. 96-101.

[7] Pirolli, P., Computational Models of Information Scent-Following in a Very Large Browseable Text Collection, in CHI'97. 1997, ACM Press: Atlanta, GA. p. 3-10.

[8] Pirolli, P. and S. Card, Information Foraging in Information Access Environments, in CHI'95. 1995, ACM Press: Denver, CO. p. 51-58.

[9] Ponceleon, D. and A. Dieberger. Hierarchical Brushing in a Collection of Video Data. Proc of HICSS'34 (Hawaii International Conference on Systems Science). 2001. Maui, HI.

[10] Rao, R. and S.K. Card, The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus+Context Visualization for Tabular Information, in CHI'94. 1994, ACM Press: Boston. p. 318-322.

[11] Russell, D.M., et al., The Cost Structure of Sensemaking, in InterCHI'93. 1993: Amsterdam. p. 269-276.