THE DOCUMENT LENS

George G. Robertson and Jock D. Mackinlay

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304
415-812-4755, robertson@parc.xerox.com

ABSTRACT

This paper describes a general visualization technique based on a common strategy for understanding paper documents when their structure is not known, which is to lay the pages of a document in a rectangular array on a large table where the overall structure and distinguishing features can be seen. Given such a presentation, the user wants to quickly view parts of the presentation in detail while remaining in context. A fisheye view or a magnifying lens might be used for this, but they fail to adequately show the global context. The *Document Lens* is a 3D visualization for large rectangular presentations that allows the user to quickly focus on a part of a presentation while continuously remaining in context. The user grabs a rectangular lens and pulls it around to focus on the desired area at the desired magnification. The presentation outside the lens is stretched to provide a continuous display of the global context. This stretching is efficiently implemented with affine transformations, allowing text documents to be viewed as a whole with an interactive visualization.

KEYWORDS: User interface design issues, interface metaphors, graphic presentations, screen layout, 3D interaction techniques.

INTRODUCTION

In recent years, several efforts have been made to take advantage of the advances in 3D graphics hardware to visualize abstract information [3, 4, 7]. Our work on the Information Visualizer [7] has described a range of interaction techniques for understanding information and its structure. In particular, we have developed visualizations for hierarchical and linear structures, called the Cone Tree and the Perspective Wall. However, users often start with information with unknown structure. For example, a user may not know that a document is hierarchical such as a book that contains chapters and sections, or linear such as a visitor log. Therefore, we are also developing general visualization techniques that can be used for unfamiliar information.

Our basic goals remain the same as with our other work on the Information Visualizer. We want to use 3D to make more effective use of available screen space. We want to use interactive animation to shift cognitive load to the human perceptual system. We want a display that provides both a detailed working area and its global context (as in both the Cone Tree and Perspective Wall). We want to aid the user in perceiving patterns or texture in the information. The *Document Lens* is an experimental interaction technique implemented in the Information Visualizer to address this set of goals when information is placed in a rectangular presentation.

THE PROBLEM

If you lay the entire contents of a multi-page document out in two dimensions so it is all visible, the text will typically be much too small to read. Figure 1 shows a document laid out in this way. Yet, we would like to be able to do this so that patterns in the document can be easily perceived (especially when a search is done and the results are highlighted in a different color). Futhermore, we want the user to be able to quickly zoom into a desired part of the document so it can be read, *without* losing the global context.

We are particularly interested in revealing the texture or pattern of relationships between parts of a document. In Figure 1, a search has been done for the term "fish-

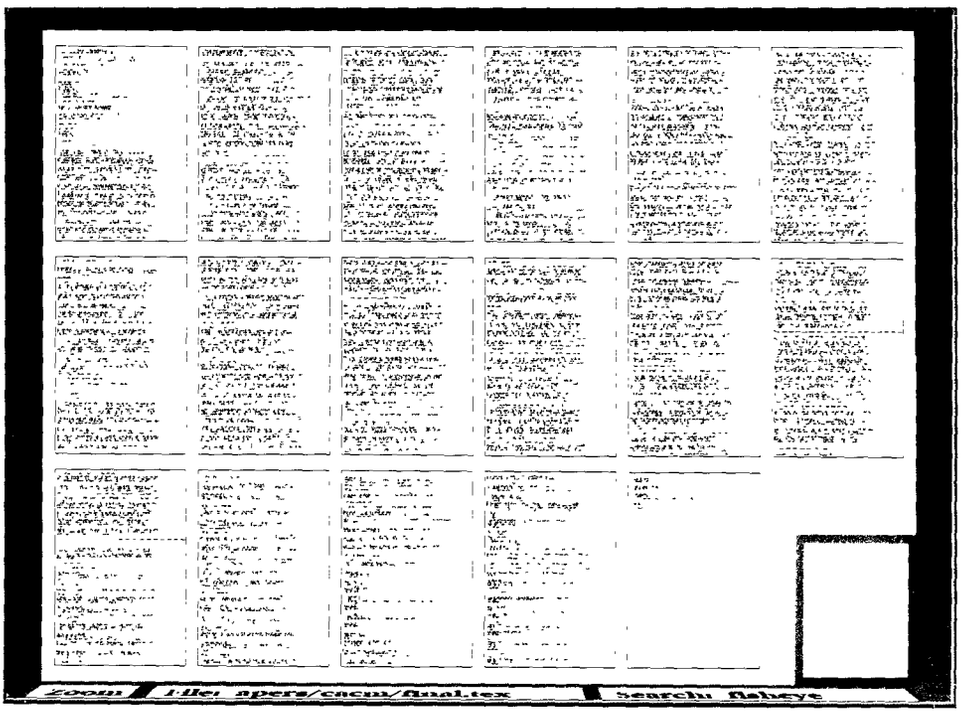

Figure 1: Document laid out on a 2D surface. Red highlights are the result of a search.

eye" in the document, which is the text from our recent CACM article [7]. If you look closely, you will see five places highlighted in red in the document that refer to the term, and most of these occurrences are close together. You can imagine that the sections of a structured document could be highlighted so that the pattern of references to a term can show how the sections relate to one another.

In order to focus on one part of a document while retaining the global context (so you can continue to see the interesting patterns), you need what we call a *Focus + Context Display.*

If you try to do this with a traditional magnifying lens (either a physical one or one implemented in software), you will necessarily obscure the parts of the document immediately next to the lens, thus losing the global context. Figure 2 illustrates this problem. Thus, a simple magnifying lens does not provide a focus + context display.

One possible solution is to use an optical fisheye lens (like looking at something through a glass sphere). Silicon Graphics has a demonstration that uses this technique on images. The problem is that the distortions that result from such a lens make reading text difficult even for the text in the middle of the lens.

Another strategy is to distort the view so that details and context are integrated. Furnas developed a general framework called *fisheye views* for generating distorted views [5]. Fisheye views are generated by Degree of Interest functions that are thresholded to determine the contents of the display. However, thresholding causes the visualization to have gaps that might be confusing or difficult to repair. Furthermore, gaps can make it difficult to change the view. The desired destination might be in one of the gaps, or the transition from one view to another might be confusing as familiar parts of the visualization suddenly disappear into gaps.

Sarkar and Brown developed a generalization of Furnas' fisheye views specifically for viewing graphs [8]. Their technique works in real time for relatively small graphs (on the order of 100 vertices and 100 horizontal or vertical edges). They acknowledge that the technique does not scale up to significantly larger graphs. The text in Figure 1 is from a relatively small document (16 pages). Even so, it requires approximately 400,000 vectors, or about 4000 times larger than the Sarkar and Brown technique can handle in real time.

Spence and Apperley developed an early system called the Bifocal Display that integrates detail and context through another distorted view [9]. The Bifocal Display was a combination of a detailed view and two distorted

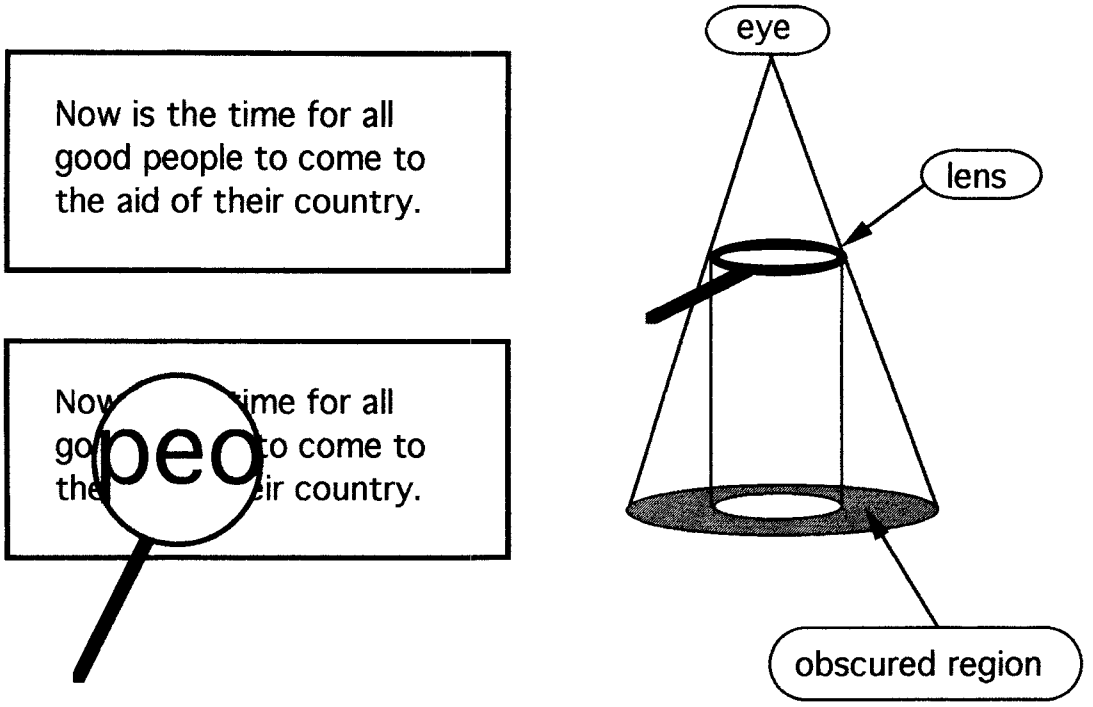

Figure 2: Illustration of the problem with a magnifier lens: parts of the image near the edges of the lens are obscured by the lens.

views, where items on either side of the detailed view are distorted horizontally into narrow vertical strips. For example, the detailed view might contain a page from a journal and the distorted view might contain the years for various issues of the journal. Because Bifocal Displays are two dimensional, they do not integrate detail and context completely smoothly. Two versions of an item are required, one for the detailed view and one for the distorted view. The relationship between these versions may not be obvious. As the focus moves, items suddenly expand or shrink, which may be confusing. Furthermore, the distorted view treats all contextual items identically, even those near the detailed view.

The Perspective Wall [7] is a technique for visualizing linear information by smoothly integrating detailed and contextual views. It folds wide 2D layouts into intuitive 3D visualizations that have a center panel for detail and two perspective panels for context. The Perspective Wall provides a fisheye effect without distortions by using the natural fisheye effects of 3D perspective. However, the Perspective Wall does not handle well 2D layouts that are large in both dimensions, such as a document laid out as pages in a rectangular array. Furthermore, it is unclear how to distort efficiently the corners of a 2D sheet when it is folded both horizontally and vertically. Hence a different approach is required.

THE DOCUMENT LENS

Assume that the document pages are laid out onto a large rectangular region. In general, what we need is a way of folding or stretching that region in 3D so that part of it is near you, but the rest is still visible (giving you the desired focus + context display). The 3D deformation should be continuous to avoid the discontinuities of fisheye views and the Bifocal Display and it should be possible to implement it efficiently on a wide class of graphics machines.

We propose a new kind of lens, called the *Document Lens*, which gives us the desired properties. The lens itself is rectangular, because we are mostly interested in text, which tends to come in rectangular groupings. The Document Lens is like a rectangular magnifying lens, except that the sides are elastic and pull the surrounding parts of the region toward the lens, producing a truncated pyramid. The sides of the pyramid contain all of the document not directly visible in the lens, stretched appropriately. This gives us the desired focus + context display; that is, the whole document is always visible, but the area we are focusing on is magnified. Figure 3 shows the lens moved near the center of the document and pulled toward the user. The resulting truncated pyramid makes the text in and near the lens readable. Notice that the highlighted regions

are still visible, helping retain the global context. Also notice that the Document Lens makes effective use of most of the screen space.

In the current implementation, the lens size is fixed. It could obviously be made changeable by adding resize regions on the corners of the lens, similar to the familiar way of reshaping a window in many window systems.

The lens is moved using a technique similar to the general technique for moving objects in 3D described in [6], using only a mouse and keyboard. The mouse controls motion in the X-Y plane, and the Space and Alt keys move the lens forward and backward in the Z plane. Obviously, a 3D input device could be used as well, but we have found that the mouse and keyboard are sufficient. When the lens is moved, the movement is done with interactive animation, so that user always understands what is being displayed. This helps reduce cognitive load by exploiting the human perceptual system.

As we move the lens towards us, we are faced with two problems that must be solved to make this technique practical. First, we have a problem of fine control as the lens moves toward the eye. If you use a constant velocity, you will not have sufficient control near the eye. So, we use a logarithm approach function as we did in the general object movement technique [6].

Second, and more subtle, as the lens moves in the Z direction toward you, it moves out of view. In fact, up close, you can no longer even see the lens, and therefore cannot use it to examine anything except the center of the overall region is minute detail. Figure 4 illustrates this problem. Our solution is to couple movement of the lens with viewpoint movement, proportional to the distance the lens is from the eye. In other words, when the lens is far away, there is very little viewpoint movement; but, when the lens is near you, the viewpoint tracks the lens movement. Done properly, this can keep the lens in view and allow close examination of all parts of the whole document.

This method of display makes it quite easy to show search results. If you use the traditional technique of color highlighting the search results, then patterns in the whole document become evident, even when viewing part of the document up close. The simple search result shown in the figures is based on a simple string match, and is the only search currently implemented. More complicated searches could easily be added. In the Information Visualizer, we use relevance feedback search [1] and semantic clustering algorithms [2] to show relationships between documents. In a similar way, we could apply these to the elements of a document to show relationships between parts of a document. We could use relevance feedback search to select paragraphs and search for other paragraphs with similar content. Using the clustering algorithms, we could group paragraphs into semantically similar clusters. These search techniques enhance the richness of texture that we could make visible in documents.

Although we have focused on documents and text, the Document Lens can also be used to view anything laid on a 2D plane (e.g., images).

IMPLEMENTATION ISSUES

We have implemented a version of the Document Lens in the Information Visualizer. There are at least two ways to implement the truncated pyramid that results from moving the lens toward you, and get real time response. If you could produce a high resolution image of the 2D layout, you could use either software or hardware texture mapping to map the image onto the truncated pyramid. Currently, we know of no way to produce the required high resolution texture to make either of these approaches practical.

Conceptually, our approach involves rendering the text five times. Each of the five regions (the lens, top side, left side, bottom side, and right side) is translated, rotated, and scaled (in X or Y) to give the proper view of that side. For example, if the lower left corner of the lens is $(x1, y1, z1)$, then the left side is rotated $-\frac{180 \arctan(z1,x1)}{\pi}$ degrees about its left edge, and is stretched along the X axis by a factor of $\frac{\sqrt{x1^2+z1^2}}{x1}$. The top side is rotated about its top edge and stretched along the Y axis, and so on. Most graphics machines provide efficient implementations of these affine transformations. The next step is to clip the trapezoid parts to their neighbors' edges. This step can be implemented in software, but is relatively expensive. We do this step efficiently using the SGI graphics library user specified clipping planes. Finally, culling is done so that only the necessary pages of text need be rendered for each region. The result is that each page of text is rendered about two times on the average.

Another performance enhancement technique, shown in Figure 5, replaces text outside of the lens with thick lines. This is known as *greeking* the text. Greeking is used during all user interaction (e.g., during lens movement), so that interactive animation rates are maintained. Also, the user can choose to keep the text greeked at other times.

The limiting factor in this technique is the time it takes

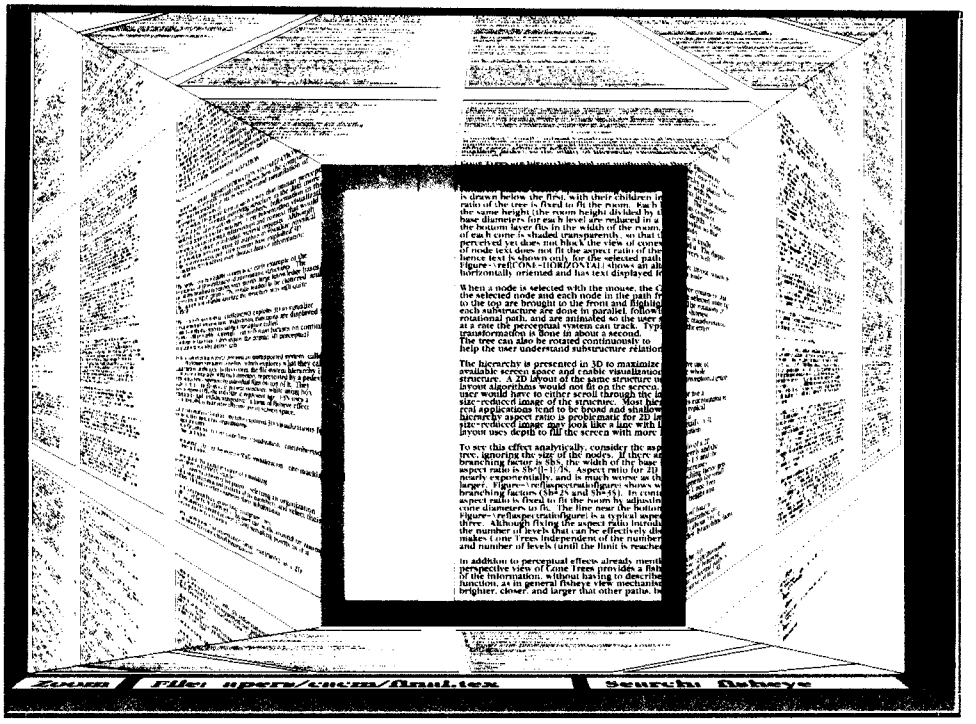

Figure 3: Document Lens with lens pulled toward the user. The resulting truncated pyramid makes text near the lens' edges readable.

to render text in 3D perspective. We use two methods, both shown in Figure 6. First, we have a simple vector font that has adequate performance, but whose appearance is less than ideal. The second method, due to Paul Haberli of Silicon Graphics, is the use of texture mapped fonts. With this method, a high quality bitmap font (actually any Adobe Type 1 outline font) is converted into an anti-aliased texture (i.e., every character appears somewhere in the texture map, as seen on the right side of Figure 6). When a character of text is laid down, the proper part of the texture map is mapped to the desired location in 3D. The texture mapped fonts have the desired appearance, but the performance is inadequate for large amounts of text, even on a high-end Silicon Graphics workstation. This application, and others like it that need large amounts of text displayed in 3D perspective, desperately need high performance, low cost texture mapping hardware. Fortunately, it appears that the 3D graphics vendors are all working on such hardware, although for other reasons.

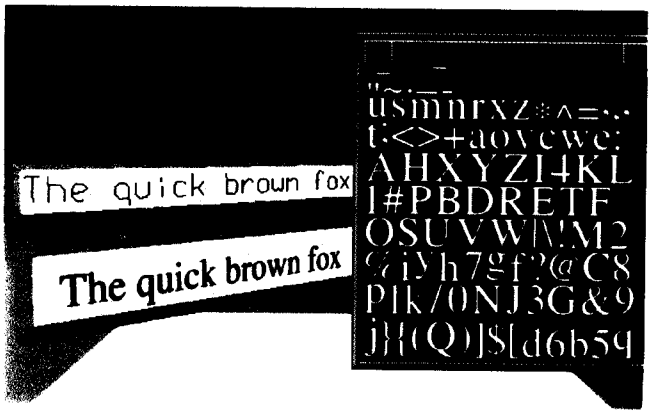

Figure 6: Vector font, texture-mapped font, and font texture map.

SUMMARY

The Document Lens is a promising solution to the problem of providing a focus + context display for visualizing an entire document. But, it is not without its problems. It does allow the user to see patterns and relationships in the information and stay in context most

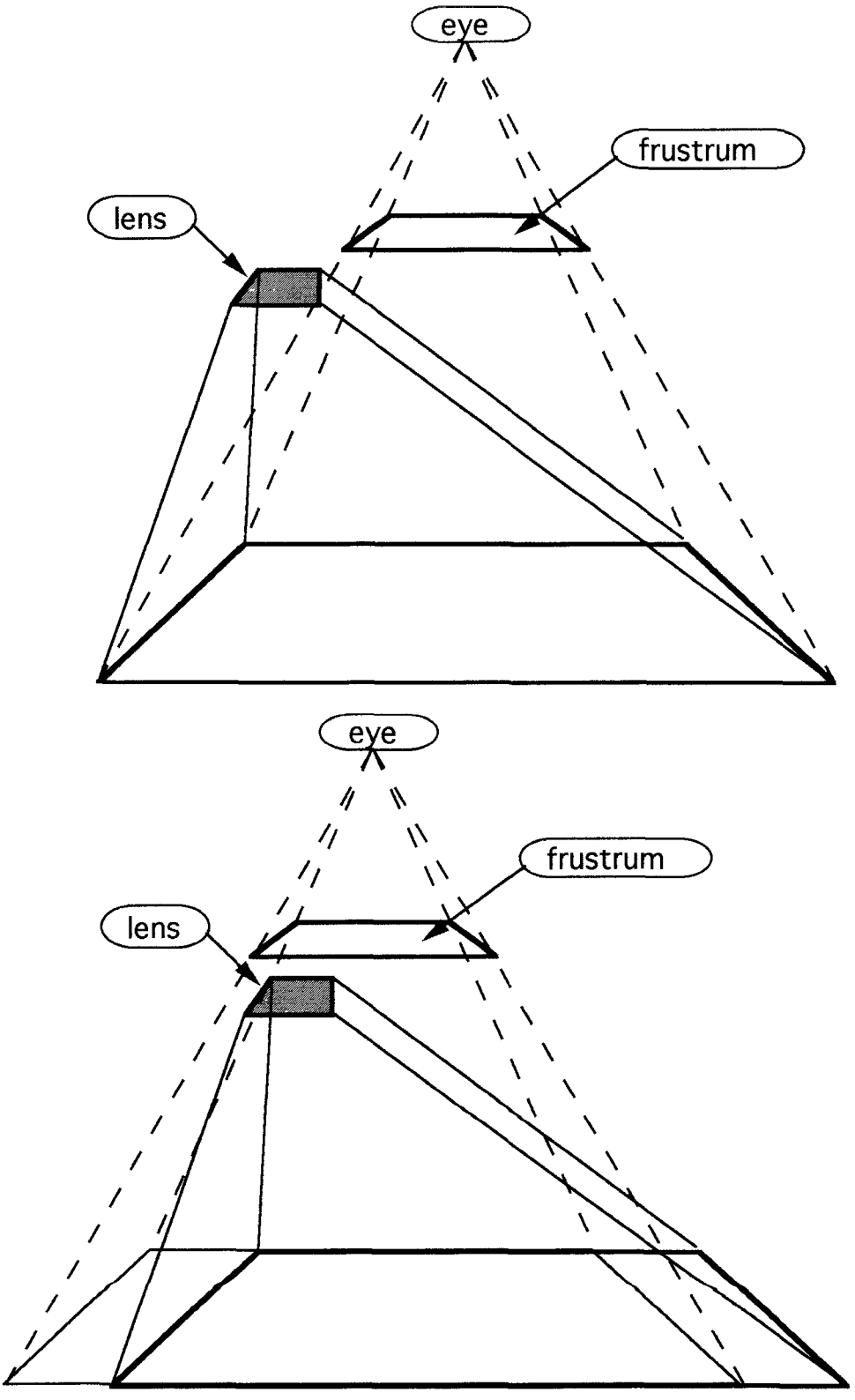

Figure 4: (a) Illustration of how the truncated pyramid may leave the viewing frustrum if lens movement is not coupled to viewpoint movement. (b) The frustrum after viewpoint movement.

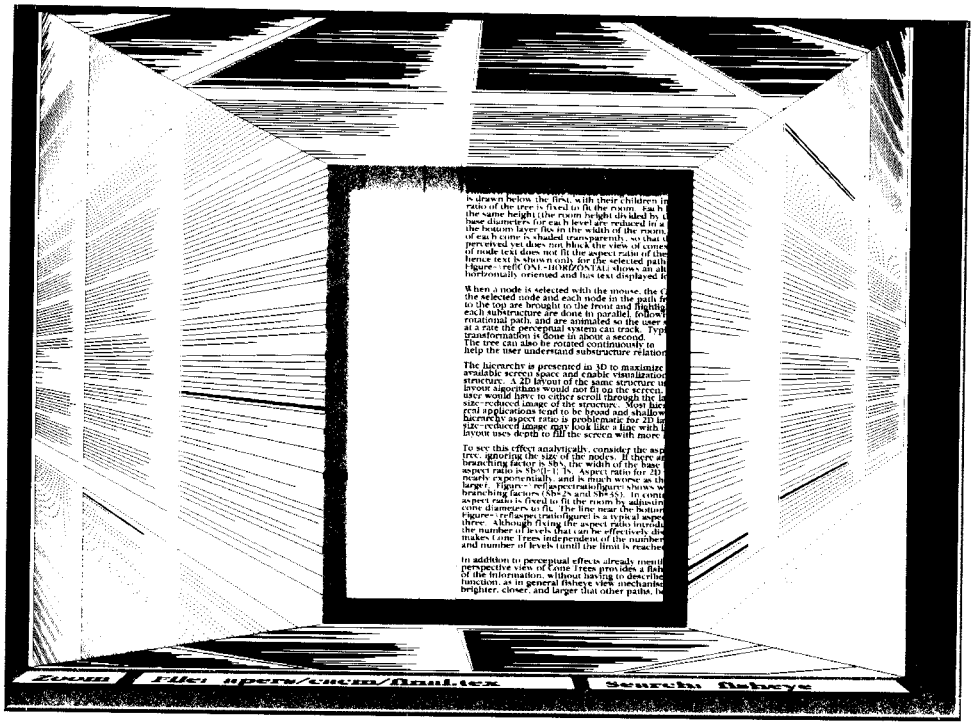

Figure 5: Document Lens with text on the sides greeked.

of the time. But, as the lens moves towards you, beyond a certain point the sides of the lens become unreadable or obscured, and you lose the context. This happens when the lens is close enough that it occupies most of the viewing frustrum. Sarkar and Brown observed the same problem for their distortion technique [8].

The coupling of lens movement with viewpoint movement is a critical part of this interaction technique. Without it, the Document Lens is useless. It may be that the obscuring problem of a close lens could be solved by coupling the size of the lens to its movement as well (making the close lens smaller).

The Document Lens has broader applicability that just viewing text documents. It could also be used to view any 2D graph (e.g., a map or diagram), providing a 3D perspective fisheye view. In that sense, it has some similarity to the Sarkar and Brown fisheye graph viewing technique [8]. However, generalized distortion is expensive. In contrast, the Document Lens works in real time for much larger graphs, efficiently doing a particular distortion using common affine transforms (3D perspective view, scaling, rotation), clipping, culling, and greeking.

There are some obvious additions that could be made to our current implementation, including adjustment to lens size and shape, and more elaborate search methods. But, these additions and usability testing have not been done because we need better hardware support for rendering large amounts of high quality text in 3D perspective. Fortunately, hardware trends (both in processor speed and 3D graphics hardware, particularly in texture mapping hardware) should make this a viable approach in the near future.

References

[1] Cutting, D. R., Pedersen, J. O. and Halvorsen, P.K. An Object-Oriented Architecture for Text Retrieval. In *Proceedings of RIAO'91, Intelligent Text and Image Handling,* 1991, pp. 285-298.

[2] Cutting, D. R., Karger, D. R., Pedersen, J. O. and Tukey, J. W. Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections. In *Proceedings of SIGIR'92,* 1992, pp. 318-329.

[3] Fairchild, K. M., Poltrock, S. E. and Furnas, G. W. Semnet: three-dimensional graphic representations of large knowledge bases. In *Cognitive science and its applications for human-computer interaction,* Guindon, R. (ed), Lawrence Erlbaum, 1988.

[4] Feiner, S. and Beshers, C. Worlds within worlds: metaphors for exploring n-dimensional virtual

worlds. In *Proceedings of the UIST'90,* 1990, pp. 76-83.

[5] Furnas, G. W. Generalized fisheye views. In *Proceedings of SIGCHI'86,* 1986, pp. 16-23.

[6] Mackinlay, J. D., Card, S. K., and Robertson, G. G. Rapid controlled movement through a virtual 3d workspace. In *Proceedings of SIGGRAPH '90,* 1990, pp. 171-176.

[7] Robertson, G., Card, S., & Mackinlay, J. (1993) Information visualization using 3D interactive animation. *Communications ACM, 36, 4,* April 1993, pp. 57-71.

[8] Sarkar, M. & Brown, M.H. (1992) Graphical fisheye views of graphs. In *Proceedings of SIGCHI'92,* 1992, pp. 83-91.

[9] Spence, R. and Apperley, M. Data base navigation: An office environment for the professional. *Behavior and Information Technology 1* (1), 1982, pp. 43-54.