

# Viewing Text With Non-Linear Magnification: An Experimental Study

T. Alan Keahey\*  
*tkeahey@cs.indiana.edu*

Julianne Marley  
*jmarley@cs.indiana.edu*

Department of Computer Science  
215 Lindley Hall  
Indiana University  
Bloomington, IN 47405  
Phone: (812) 855-3609  
Fax: (812) 855-4829

April 24, 1996

## Abstract

This paper presents the results of an experimental study in which users were given the task of reading and searching text files with both a paging viewer and a “magnifying” viewer which allows the entire file to fit on a single screen, with a bar of magnification which can be used to make portions readable. A comparison is made between the two different viewers on two different types of text layout, one highly structured, and the other with little structure. Results indicate a suitability for using the magnifying browser on structured text.

## Keywords

Visualization, magnification, fisheye, text viewing, warping, structured text.

---

\*This work was partially supported by US Dept. of Education award number P200A502367.

# 1 Introduction

Normal text viewers (whether paging or scrolling) present text to the user screen by screen, with the result that contextual cues for location within a file are hidden from the user's view. For paging browsers, typically zero or one lines of text are provided as context for each new page (i.e. the last line on one page becomes the first line on the next). For scrolling browsers the context (group of surrounding lines, typically 10+ lines up and down in the file) is updated continuously during scrolling. However, neither of these methods provides the user with a *global* view of the file along with a sense of location within that view.

This paper will investigate the possibility that providing such a global view could facilitate more efficient user interaction. The tool for creating this global view is non-linear magnification, which we will use to create a “magnifying text viewer” which allows the user to view an entire text file on a single screen.

## 2 Viewer Design

A fundamental principle in our design of the magnifying text viewer was to keep the user interface simple enough that subjects could be easily trained in its use in just a few minutes. For this reason, the viewer discussed here represents only a subset of the functionality that can be implemented with the magnifying viewer (and has been implemented in other versions).

Figure 1 shows a snapshot of the magnifying text viewer in action. Notice that the entire file is displayed in a single screen, although only the section underneath the window of magnification is clearly readable at any one time. A small box in the left hand margin indicates the vertical posi-

tion on the page which is the center of magnification. The non-linear magnification used for this system involves a transformation on the  $y$  coordinate only, using a narrow rectangle the width of the screen to constrain the domain of magnification (details for how to implement this transformation are described in [6]).



Figure 1: Text Viewing Program

### 2.1 Magnification Transformation

The magnification transformation produces vertical coordinates which are used to position each line of text, however no magnification of the actual text occurs in this system, only the spatial separation between lines is changed. The effect of this transformation is a “magnification bar” which the user can move over the text.

Because all of the text is rendered at the same font size, adjacent lines of text in non-magnified areas will tend to overlap. Techniques for dealing with this overlap are discussed in subsection 2.2, however we note here that: 1) this overlap allows us to guarantee that the entire file can fit in a single window, and 2) this overlap is not likely to skew study results in favour of the magnifying interface.

One drawback of the bounded magnification

used here is that neighboring text tends to overlap even more than the areas further outside the region of magnification. An alternative to this is to spread the magnification over the entire text file as in figure 2, which offers the benefit of pushing the areas of minimal magnification as far away as possible from the center of magnification. This approach introduces unwanted side-effects however, as the global context of the view is no longer static, and any movement of the magnification bar will cause the entire text to shift in the window. For this reason we decided to use the constrained domain for this study, which despite the problem mentioned above, does provide a static global context.



Figure 2: Unbounded Magnification

## 2.2 Text Representation

For this system every line of text was rendered at the same size, regardless of the degree of magnification. Variable font sizing systems such as the one discussed in [5] present some potential problems:

- For fast rendering, bit-mapped fonts are generally preferred. However bit-mapped fonts only come in discrete sizes, and so it is not possible to display smoothly increasing

or decreasing character size, instead the size of the text will tend to jump up and down.

- Different font sizes implies different font spacing. A line of text at 24pt size is going to be wider than the same line of text at 12pt size. Exactly where the two sizes should be lined up is unclear. If all lines are aligned by their leftmost character, then the right-hand edges will no longer line up, and in the case of textual tables the entries may also not line up.

An alternative is to use a method such as the one described by Ball and Eick [1], where lines of text in “demagnified” areas are rendered as simple lines having appropriate indentation.

## 2.3 User Interaction

In the magnifying interface, subjects could use the mouse or the keyboard for navigation. Use of the keyboard was limited to the up and down arrow keys. The down arrow scrolled the magnification bar down several lines and the up arrow scrolled it up. In each case, the text which appeared in the magnification bar after striking an arrow key included at least one line which was readable in the previous magnification bar location.

Using the mouse in the magnifying interface allowed the user to: 1) Click the left mouse button on any part of the text page and the magnification bar would smoothly scroll to that point (with the clicked location appearing in the vertical center of the magnification bar). 2) Click and hold the left mouse button to drag the magnification bar up or down the screen.

## 3 Experiment Design

### 3.1 Overview

An evaluation of a magnifying interface was based on trials in which subjects read two texts (one using the magnifying interface and one using a typical paging interface), answering questions after each text about the interface and text. Sixteen subjects each participated in a single trial. Two texts were used in each trial.

### 3.2 Interfaces

#### The Magnifying Interface

The magnifying interface allows the full text of a document to fit in the screen window – although the lines of text appear so close together they are unreadable. The magnification bar is used to separate lines of the text making them readable. The magnification bar makes approximately 8 lines of text readable at a single time. With this interface, subjects were allowed to use the mouse or the keyboard for navigation.

#### The Paging Interface

The unix program "less" was used as the paging interface. While "less" has considerable navigation capabilities, subjects were limited to using only the space bar and b key. The space bar scrolled down one screen and the b key up one screen. Scrolling up or down using "less" replaces all lines on the screen. (This is unlike "more" which repeats a single line of text from the previous screen on the new one.) This interface allowed 40 lines of text to be viewed in the text window at a single time.

In both interfaces, the top and bottom of the texts were delimited by double horizontal lines.

In the case of the magnifying interface, the top and bottom double lines were visible in the text window at all times.

### 3.3 Texts

Two different texts were used in each trial. One text (see figure 3) had few formatting cues (headings, indentation, white space) to indicate its structure while the other was more heavily formatted (see figure 4). The subject of the structured text was the 1993 federal budget. The subject of the less structured text was state and local taxes in 1993. The less structured text was 150 lines long and was much more dense (there were fewer blank lines) than the structured text, which was 120 lines long.



Figure 3: Unstructured Text

### 3.4 Trials

The sixteen trials were all conducted on silicon graphics machines using a text window 80 characters wide and 40 lines long. The subjects were divided (randomly) into four groups based on which text and which interface were to be used together and which interface was to be used first (refer to table 1).



Figure 4: Structured Text

Group	First Article	First Interface
1	unstructured	magnifying
2	unstructured	paging
3	structured	magnifying
4	structured	paging

Table 1: Distribution of Subjects

During each trial, the subject viewed a sample text in the interface to be used first. (The sample text was the text version of the Indiana University Computer Science Department home web page, see figure 1). Use of the first interface was explained by the experimenter reading a short set of instructions, which were left with the subject. The subjects were given as much time as they wanted to familiarize themselves with the interface before reading the first article (we did not record the time the user used for familiarization, but it was typically under 1 minute).

When the subject was ready, the first article was brought to the text window, and the subject was asked to read the text. The subject was instructed to inform the experimenter when he/she was finished reading.

When the subject was finished reading, he/she was given a piece of paper with two general questions about the text and asked to answer the

questions without referring to the text. These general questions asked for a summary of the article and what the perspective of the author was, in order to verify that the subject had actually read and understood the article..

Following the first two written questions were four specific questions the subjects had to answer verbally. Most specific questions required numerical answers. For these questions subjects were allowed to refer back to the text (using the same interface in which the text had just been read). Response times for these questions was measured. Each question was first read to the subject by the experimenter. The moment the experimenter finished stating the question, timing began. The typed question was left with the subject so that he/she could refer back to it. Timing stopped the moment the subject began a verbal reply. The response was recorded.

Subjects were then given two more general questions requiring written replies. For these questions subjects were told they could refer back to the text if they so chose. These questions regarded whether the subject liked or disliked the article and whether he/she found it interesting.

Upon completing the written questions, subjects were given a 1 - 5 minute break before beginning the second text. After one minute, the subject was told to inform the experimenter when he/she was ready to continue.

The second trial proceeded the same as the first. Upon completion of the questions on the second text, the subject was given a final series of written questions about his/her experience with related interfaces and his/her opinions of the two interfaces for the tasks of reading and answering questions.

### 3.5 Subjects

Subjects were chosen having a variety of experience with computers. 10 of the subjects had a lot of experience with using computers (6 of these were Computer Science graduate students), 5 subjects had a fair bit of experience with using computers, and 1 subject had almost no experience using computers.

Question	Magnifying	Paging
unstruct 1	73.14 (51.61)	28.78 (21.24)
unstruct 2	53.25 (52.05)	42.31 (26.49)
unstruct 3	27.78 (26.48)	13.00 (7.23)
unstruct 4	89.50 (87.09)	25.85 (13.94)
<b>Total</b>	243.67 (154.62)	109.94 (45.40)
struct 1	11.08 (4.73)	49.49 (86.60)
struct 2	8.64 (7.00)	7.22 (4.43)
struct 3	38.27 (14.05)	51.35 (54.83)
struct 4	12.95 (7.42)	12.65 (7.86)
<b>Total</b>	70.94 (24.97)	120.71 (144.49)

## 4 Experiment Results

Table 2 shows the mean average times (in seconds) required by the subjects to locate items in the text. For the less structured file, it took subjects significantly longer (120% longer on average) to find the items with the magnifying browser than with the paging browser. For the hypothesis that the users would be able to find facts faster in the unstructured text, the data produce a t-statistic of  $-2.347$  ( $\text{prob} < t = 0.0341$ ).

When the more structured file was being searched however, users were able to find the facts more quickly (41% faster on average) using the magnifying browser than with the paging browser. To test the hypothesis that users were able to find facts faster with the magnifying browser, the data produce a t-statistic of  $0.959$  ( $\text{prob} > t = 0.3534$ ).

Table 3 shows the effect of changing the order in which the browsers are used for the subjects (mean time in seconds). Based on these data it appears that subjects did take longer (36% on average) to answer questions when they used the magnifying browser in the second trial instead of in the first trial. This could possibly be due to user fatigue, however our methodology of switching order of browsers *and* files in all possible combinations should help eliminate

Table 2: Mean Time (Standard Dev.) to Find Facts With Different Browsers

this error from the averages shown in table 2. To test the hypothesis that users would take longer to answer a question if they used the magnifying viewer in the second trial instead of in the first, we obtained a t-statistic of  $0.6810$  ( $\text{prob} > t = 0.5070$ ) from the data.

Question	First	Last
unstruct 1	60.60 (57.85)	41.32 (26.06)
unstruct 2	49.68 (52.19)	45.88 (27.33)
unstruct 3	12.80 (8.58)	27.98 (25.94)
unstruct 4	45.03 (22.25)	70.33 (96.09)
<b>Total</b>	168.11 (121.80)	185.51 (145.52)
struct 1	11.27 (5.08)	49.30 (86.68)
struct 2	7.49 (4.22)	8.37 (7.17)
struct 3	36.64 (15.22)	52.98 (54.01)
struct 4	11.48 (8.37)	14.13 (6.53)
<b>Total</b>	66.88 (20.16)	124.78 (143.51)

Table 3: Mean Time (Standard Dev.) When Using Magnifying Browser First/Last

We asked all subjects which browser they preferred for reading the text file, and which they preferred for locating items in the text. None of the subjects preferred the magnifying browser for reading the files. For the less structured file, only 1 out of the 8 subjects who used the magnifying browser on that file stated a preference for using the magnifying browser for locating items in the text. However, when the more structured

text was presented with the magnifying browser, *all* of the subjects stated a preference for using that browser for locating items in the text (see table 4). The different responses for reading and searching suggests that usability of this interface is highly task dependent.

	Reading	Searching
<b>Unstructured</b>	0%	12%
<b>Structured</b>	0%	100%

Table 4: Percentage of Users Who Preferred the Magnifying Browser

50% of the subjects reported having previous experience with paging browsers similar to the one used in the study. Three of those subjects also cited this previous experience as a reason for preferring the paging browser.

In 3 cases (all using the paging viewer, 2 on the unstructured file and 1 on the structured file) subjects had memorized the answer to one of the questions, and did not have to refer back to the text to answer the question. Also, in a few specific instances (3 times with the magnifying interface on the less structured file, and 3 times with the paging interface on the structured file) subjects took an extremely long time to find the answer, or in one case (with the magnifying viewer on the unstructured text) gave up before finding it. These outliers were not removed from the data.

## 4.1 Analysis

The machine performance issue was definitely significant in this study. In particular, the text rendering routines proved to be prohibitively slow for large text files. On the development platform near-interactive frame rates (8.5 FPS) were recorded, however on the testing platforms under normal loads response was much slower (5 FPS). One common complaint about the mag-

nifying browser was that the response was too sluggish (37.5% of subjects). A significant reason for the slow performance is that in the current prototype implementation the entire file is redrawn in each frame in order to provide the global view. Future versions could incorporate more sophisticated algorithms for updating only those portions of the screen which change from frame to frame.

Another common remark from subjects was that the window of magnification bar should have been taller for the magnifying interface, so that more lines of text can be read without having to slide the bar so often (50% of subjects). It seems possible that changing this (or making the width user-settable), would make the magnifying program easier for normal reading of text.

In addition, some subjects noticed that there was a slight inconsistency of the magnifying browser in response to their mouse clicks. Any mouse click on a line of text *outside* the magnification bar caused that spot on the page (and that line of text) to be placed at the center of the magnification bar. However, when subjects clicked on a line of text *inside* the magnification bar, the line of text associated with the *untransformed page* (as opposed to magnification bar) coordinate would become the center of magnification, rather than the line of text that was actually clicked on. Clearly this inconsistency would need to be fixed for the program to be useful, this can be done via an inverse transformation of the mouse click coordinates before the new center of magnification is set.

We are encouraged that despite the small amount of subject training time and the shortcomings of the interface described above, users still were able to find facts faster in the structured file, and also stated a preference for using the magnifying interface in this case (although the significance level for the hypothesis that subjects found facts faster with the magni-

fyng interface on the structured text is not very strong). Future experiments addressing the issues raised here might obtain stronger results. Also, the results indicate that this type of interface is better suited for structured text than relatively unstructured text. Further applications of this technology should be developed with this in mind.

It should be noted that this study does not exhaustively test all types of text viewing paradigms or types of texts. In particular, a comparison between the magnifying browser and a scrolling text browser might produce different results. Here the similarities between browsers would be greater, perhaps better isolating the distinct “global view” property which the magnifying browser allows.

## 5 Conclusions

Our study provides evidence for the suitability of this type of magnifying interface for visual searching of more structured text files. There are many applications where such structuring of text is commonplace, such as in text views of HTML (see figure 1) and program source code.

Subjects in this study did not perform better with the magnifying interface on unstructured files however. Although it is still an open question whether further improvements to the interface would change this finding, this study suggests that the advantages of the magnifying interface are intrinsically tied to the “landmarks” which are present in the global view of more structured texts.

## 6 Related Work

Greenberg and Gutwin [5] have developed a groupware text editor using fisheye views (with multiple focus points) to provide global context of text files. Their system does not cause the text to overlap, however the discrete font sizes used for the text representation also do not allow a guarantee that an entire text file will fit in a single window.

Ball and Eick [1] have worked on using smaller line and pixel representations of text to enable more of a source code file to be “seen” on a single screen. With discrete resolution levels, detail is not smoothly integrated into the context however, and is instead displayed in separate windows or neighboring areas.

Mackinlay, Robertson and Cards’ “Perspective Wall” [8] describes a technique for providing limited context for text viewing, but does not provide the global file view described here.

## 7 Further Work

Multiple bars of magnification can allow for more than one area of the text to be viewed at once. Such functionality has been implemented for our browser, but has yet to be subjected to user studies. In addition, piecewise linear transformations [6] can be used to produce more arbitrary magnification transformations, similar to the customizable “fisheye lens” described in Greenberg and Gutwin [5].

Piecewise linear functions also offer the intriguing possibility of constructing magnification transformations based on the *content* of text files. For example, an HTML viewer might magnify header lines, and suppress details in listed data until they are magnified by the user. Sim-



ilarly for source code viewers, the viewer might be able to collapse entire functions into a single line represented by the function declaration, and expand the function only when magnified.

## 8 Acknowledgments

This experiment was conducted as a project for an HCI course offered by Dr. Andrew Dillon at the Indiana University School of Library and Information Science in Fall of 1995. The authors would like to thank Dr. Dillon for his input and feedback.

## References

- [1] Thomas Ball and Stephen G. Eick. Software visualization in the large. *IEEE Computer*, April 1996.
- [2] Benjamin B. Bederson and James D. Hollan. Pad++: A zooming graphical interface for exploring alternate interface physics. In *UIST '94 Proceedings*, 1994.
- [3] G. W. Furnas. Generalized fisheye views. *Human Factors in Computing Systems, CHI '86*, pages 16–23, April 1986.
- [4] George W. Furnas and Benjamin B. Bederson. Space-scale diagrams: Understanding multiscale interfaces. In *Proceedings of CHI '95 Human Factors in Computing Systems*, 1995.
- [5] S. Greenberg and C. Gutwin. Sharing fisheye views in relaxed WYSIWIS groupware applications. Technical Report 95/577/29, Department of Computer Science, University of Calgary, November 1995.
- [6] T. Alan Keahey and Edward L. Robertson. Techniques for non-linear magnification transformations. Technical Report 455, Department of Computer Science, Indiana University, March 1996. Submitted to IEEE Visualization '96, Information Visualization Symposium.
- [7] Y.K. Leung and M.D. Apperly. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, 1994.
- [8] Mackinlay Roberston and Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems 1991*, pages 173–179, 1991.
- [9] Manojit Sarkar and Marc H. Brown. Graphical fisheye views. *Communications of the ACM*, 37(12):73–84, 1994.