Automatic Text Reduction For Changing Size Constraints

Lance Good[1], Benjamin B. Bederson[2], Mark Stefik[1], Patrick Baudisch[1]

[1]Xerox Palo Alto Research Center
3333 Coyote Hill Rd
Palo Alto, CA 94304 USA
+1 650 812 4661
{good,stefik,baudisch}@parc.xerox.com

[2]Human Computer Interaction Lab
University of Maryland
College Park, MD 20742 USA
+1 301 405 2764
bederson@cs.umd.edu

ABSTRACT
This paper introduces a technique for viewing text objects under changing size constraints in 2D environments. Our approach automatically combines font size reduction and content reduction to preserve legibility of key words. Unlike traditional semantic zooming, our approach creates intermediate representations and transitions automatically. The main benefit is that it provides more meaningful views for different object sizes without additional authoring effort.

Keywords
Text reduction, semantic zooming, spatial hypertext.

INTRODUCTION
When working in 2D environments, such as VIKI [4], or in Zoomable User Interfaces, such as those based on Jazz [1], authors often organize short notes using spatial arrangements, for example to make sense of material that resulted from a brainstorming session. Most computer displays are only large enough to show the equivalent of one or two pages of text at a readable size. This limitation can have a significant effect on external cognition in comparison to physical notes on tabletops or whiteboards (for example, see [2]).

When running out of space in computer systems, users often want to make objects smaller in order to fit more material into the limited screen space. Making text objects smaller can be done using two main approaches:

1. *Reducing font size,* equivalent to scaling or zooming
2. *Reducing content*, achieved by replacing the current text with a shorter text while preserving certain semantics (e.g. [3])

We built a prototype system that combines these two approaches in generating views for reduced size text objects. In our prototype, users are able to zoom out to reduce the size requirements of all text objects in the 2D workspace. Then, the system automatically displays the appropriate reduced representation for each text object based on the size constraints at the current zoom level.

RELATED WORK
Standard semantic zooming requires authors to create multiple representations for an object and switches between them based on the zoom level [1]. Our approach differs from standard semantic zooming in that it does not require representations to be manually created by the author. Instead, it automatically generates representations to support a text object's changing size requirements.

Shipman et al also proposed reducing text object sizes to address limited display space in a 2D environment [4]. They concentrated on allocating space for a collection of text objects based on multiple foci visualization techniques. In contrast, our text reduction concentrates on automatically generating textual representations for changing space requirements.

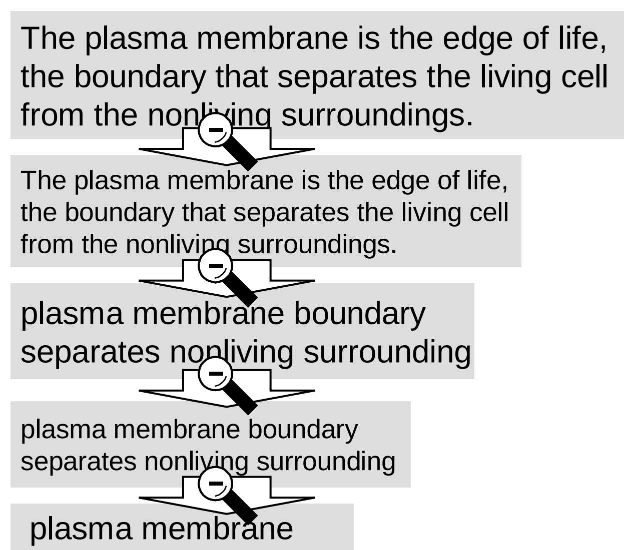

Figure 1: The text reduction technique used in our prototype. This technique automatically shortens text and reduces font size in order to meet the user's space reduction request.

SCALABLE TEXT
In the example in Figure 1, the text object initially displays three lines of text (top) and is reduced in size through zooming in each successive step. When the font reaches a

minimum size, the system automatically replaces the current representation with a shortened version of the text at the original font size. Our implementation uses cross-dissolve transparency effects to create smooth transitions between reduction levels.

For our authoring application, we also intend to provide awareness of content reduction. As a result, we plan to implement visualizations on each text objects, such as thermometers, to indicate the current level of reduction. These indicators are intended to provide feedback during operations such as editing and scanning of condensed text.

The automation offered by our technique is achieved through a text reducing function. This function generates multiple levels of text reduction based on the given size requirements. In our application, users author, collect, and organize text as part of a sensemaking task. Since users are familiar with the textual content in this application scenario, the purpose of text reduction is primarily to allow users to recognize text elements they created or selected earlier. Therefore the reduced text produced by our reducing function need not carry full meaning or be comprehensible and strictly correct.

Because our application focuses on condensing small collections of text snippets or paragraphs rather than larger texts, it violates the assumptions of many traditional linguistic techniques [3]. As a result, we have informally experimented with combinations of several simple methods including rankings based on universal word frequencies, word length, word position, syntactic role, and TFIDF [3]. One common trend we found in several of these rankings is that they eliminate short words. This seems to be a result of these words' frequent occurrence, their appearance in stop word lists, and their use in limited syntactic roles. Because of these similar results, we found that many of these reduction techniques work sufficiently well for our purposes. We expect that user testing will determine whether our application requires more complex language models for optimal text reduction.

HOW IT WORKS

The basic idea behind text reduction is that a given space requirement can typically be met by several different combinations of the available space-relevant attributes. The diagram in Figure 2 illustrates the space of possible font size/text length states that a text object can assume. The arrows show the resizing transition and the black dots denote the states that the text object takes on during that transition. The curved lines in the diagram connect all states that have an equal object size. The object size is roughly proportional to the product of font size and text length, so the curves have the shape of $f(x) = c/x$, where c is the area (a constant).

Figure 2 illustrates the oscillating behavior displayed by the text sample from Figure 1, in which the system reduces font size, and then shortens the text while increasing font size. This behavior results from the fact that text reduction decreases text length not continuously, but in larger steps, such as words or lines. In order not to waste space, the system then picks the biggest font size that fits in the available space, resulting in the described fluctuation in font size. Although this is the particular strategy used in our current prototype, other applications can choose different transitions through this graph based on the requirements of the task.

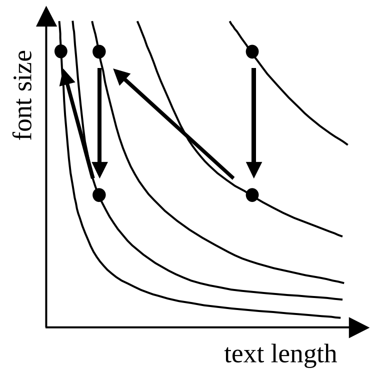

Figure 2: A graph of text reduction in terms of font size and text length. The black dots indicate the states represented in Figure 1.

CONCLUSIONS

As future work, we plan to evaluate the impact of automatic text reduction on user performance in a controlled user study involving various authoring tasks. We believe that scalable text, in addition to increasing practical screen size, has the potential to assist users in abstraction. Using reduction techniques such as eliminating common words may help users to more easily identify patterns such as rare, recurring key words or related concept terms.

ACKNOWLEDGMENTS

We would like to thank Sascha Brawer, Lauri Karttunen, and Paula Newman for advising us about summarization techniques.

REFERENCES

1. Bederson, B. B., Meyer, J., & Good, L. (2000). Jazz: An Extensible Zoomable User Interface Graphics Toolkit in Java. *UIST 2000, ACM Symposium on User Interface Software and Technology, CHI Letters*, *2*(2), pp. 171-180.
2. Henderson, D. A., and Card, S. (1986). Rooms: the use of multiple virtual workspaces to reduce space contention in a window-based graphical user interface. ACM Transactions on Graphics (TOG), v5, n.3, p.211-243.
3. Marcu, Daniel. (2000) *The Theory and Practice of Discourse parsing and Summarization*. MIT Press.
4. Shipman, F.M., Marshall, C. C., LeMere, M. (1999). Beyond Location: Hypertext Workspaces and Non-Linear Views. In *Proceedings of Hypertext '99*. p. 121-130.