Pad++: A Zoomable Graphical Interface System

Benjamin B. Bederson and James D. Hollan

Computer Science Department
University of New Mexico
Albuquerque, NM 87131
(505) 277-3112
{bederson,hollan}@cs.unm.edu

Abstract

Large information spaces are often difficult to access efficiently and intuitively. We are exploring Pad++, a graphical interface system based on zooming, as an alternative to traditional window and icon-based approaches. Objects can be placed in the graphical workspace at any size, and zooming is the fundamental navigational technique. The goal is to provide simple methods for visually navigating complex information spaces that ease the burden of locating information while maintaining an intuitive sense of location and of relationship between information objects.

Keywords:

Navigation, interactive interfaces, multiscale interfaces, zooming, authoring, information navigation, hypertext, information visualization, multimedia, world wide web.

Introduction

Over the past several years, a variety of techniques have been introduced for viewing large information spaces, including: SDMS [4], fisheye views [5], information visualizer [3], graphical fisheye views of graphs [10], Pad [9], and Pad++ [1][2][7]. Space-scale diagrams have been used as an analytical tool for some of them [6].

This paper accompanies a CHI'95 demonstration of Pad++, an interface system based on zooming. Pad++ workspaces are large high resolution areas, allowing the viewing of complex collections of information at multiple scales. Zooming and panning are the primary methods of navigation in Pad++. Several efficiency mechanisms [2] are employed to maintain interactive frame rates for animation, even when scenes get complicated.

Pad++ is a general-purpose substrate for exploring user interfaces. It directly supports creation and manipulation of zoomable graphical objects and navigation within a zoomable workspace. Pad++ is built as a new widget for Tk using Tcl, an interpreted scripting language. Increasingly popular, Tcl and Tk [8] combine a scripting language and Motif-like library for creating graphical user interfaces and applications without the need to write C code. The Tcl interface to Pad++ is similar to the interface to the Tk Canvas widget, a surface for drawing structured graphics.

Objects in Pad++ can be implemented so that they change the way they look depending on, among other things, their size, complexity of the current view, characteristics of users' task, lenses positioned over them, or the type of interface physics [2] currently operational. Such changes in view we call *semantic zooming*. It provides, for example, a simple method for representing abstraction. When you zoom out, you see a simplified rendering of the object, and when you zoom in, you see more details. Perlin [9] described a prototype zooming calendar based on this notion.

NAVIGATION

Finding information effectively with a Pad++ interface is important because intuitive navigation through large information spaces is a primary motivation. To accomplish this, Pad++ supports visual searching with zooming in addition to traditional mechanisms, such as content-based search.

Searching in Pad++ produces smooth animations to the desired objects. Animations interpolate in pan and zoom to bring the view to the specified location. If the end point, however, is more than one screen width away from the starting point, the animation zooms out to a point midway between the starting and ending points, far enough out so that both points are visible. The animation then smoothly zooms in to the destination. This gives a sense of context to the viewer and helps maintain object constancy. In addition it speeds up the animation since most of the panning is performed when zoomed out and thus covers more distance than panning while zoomed in. We use space-scale diagrams [6] to help analyze and construct these trajectories.

SAMPLE APPLICATION - HYPERTEXT

Hypertext systems confront the problem of how to give users an intuitive sense of location as they navigate through large information spaces. An example is NCSA's Mosaic system. It allows traversal of a vast information space across the internet via hyperlinks. In Mosaic, as with many other window-based hypertext systems, following a link replaces the contents of the window with the linked data, or sometimes brings up a new window. However, there is no graphical depiction of the relationship among windows - even when there is a strong semantic relationship. Thus, it is quite common to hear users complain of losing a sense of relationship between where they are and where they've been.

Pad++ attempts to address this problem by using a very high resolution zoomable surface to graphically

layout the links representing traversals. When a hyperlink is selected, the linked data is loaded to the side and made smaller while the user's view is animated to center the new information. The nodes are layed out in such a way that no traversal of links can cause overlapping. Pad++ can read in hypertext files written in Hypertext Markup Language (HTML). Figure 1 shows a snapshot with a number of home pages loaded and several links followed.

FIGURE 1: A screen dump showing a series of World Wide Web home pages loaded into Pad++.

CONCLUSION

Pad++ is a tool for exploring interfaces and visualizations based on zooming. We believe that multiscale interfaces provide effective alternative mechanisms for addressing problems associate with navigation of very large information collections. Our goal is to provide simple methods for visually navigating that ease the burden of locating information while maintaining an intuitive sense of location and of relationships between information objects.

We are currently, in collaboration with NYU and Bellcore, continuing development of the Pad++ substrate as well as starting work in several application domains.

Acknowledgments

We acknowledge Ken Perlin and his students and staff, David Fox, Matthew Fuchs, Jon Meyer and David Bacon, at NYU for many enjoyable discussions and for initiating our interest in zooming interfaces. We thank Mike Lesk, George Furnas, and Al Aho for releasing an early version of Pad++ we developed at Bellcore. We especially appreciate Craig Wier's support as part of ARPA's new HCI Initiative.

This work was supported in part by ARPA contract #N66001-94-C-6039 to the University of New Mexico. For more information about this project see http://www.cs.unm.edu/pad++.

References

1. [1] Bederson, B. B., Stead, L., and Hollan, J. D. Pad++: Advances in Multiscale Interfaces, *Proceedings of ACM Human Factors in Computing Systems Conference Companion (CHI`94)*, 315-316.
2. [2] Bederson, B. B. and Hollan, J. D. Pad++: A Zooming Graphical Interface for Exploring Alternate Interface Physics, *Proceedings of ACM User Interface Software and Technology Conference (UIST'94)*, 17-26.
3. [3] Card, S. K., Robertson, G. G., and Mackinlay, J. D. The Information Visualizer, an Information Workspace,*Proceedings of ACM Human Factors in Computing Systems Conference (CHI `91)*, 181-188.
4. [4] Donelson, W. C. Spatial Management of Information, Proceedings of 1978 ACM SIGGRAPH Conference, 203-209.
5. [5] Furnas, G. W. Generalized Fisheye Views, *Proceedings of 1986 ACM SIGCHI Conference*,

16-23.

6. [6] Furnas, G. W. and Bederson, B. B. Space-Scale Diagrams: Understanding Multiscale Interfaces, *Proceedings of ACM Human Factors in Computing Systems Conference (CHI`95)*, In Press.
7. [7] Meyer, J., Perlin, K., Bederson, B. B., and Hollan, J. D. Two Document Visualization Techniques for Zoomable Interfaces, *Proceedings of ACM Human Factors in Computing Systems Conference Companion (CHI`95)*, submitted.
8. [8] Ousterhout, J. K. Tcl and the Tk Toolkit, Addison Wesley, 1994.
9. [9] Perlin, K. and Fox, D. Pad: An Alternative Approach to the Computer Interface, *Proceedings of 1993 ACM SIGGRAPH Conference*, 57-64.
10. [10] Sarkar, M., Snibbe, S. S., Tversky, O. J., and Reiss, S. P. Stretching the Rubber Sheet: A Technique for Viewing Large Layouts on Small Screens, Proceedings of ACM User Interface Software and Technology Conference (UIST`93), 81-91.