

Pad++: A Zoomable Graphical Interface

Benjamin B. Bederson

Bell Communications Research
445 South Street
Morristown, NJ 07960
(bederson@bellcore.com)

James D. Hollan

Computer Science Department
University of New Mexico
Albuquerque, NM 87131
(hollan@cs.unm.edu)

KEYWORDS

Interactive user interfaces, multiscale interfaces, multimedia authoring, information navigation, hypertext, information visualization.

ABSTRACT

We have implemented an infinite resolution multimedia sketchpad as a base for exploring a *stream-of-consciousness model of computation* where information creating, sharing and retrieval becomes so intuitive that the interface becomes invisible. Motivation to pursue this came from work on Pad [4], which can be thought of as a kind of traditional sketchpad or windows environment in the sense that it is a general-purpose substrate for visualizing two dimensional graphics and text. But Pad also supports the radical notion of being infinite in extent and resolution.

We implemented Pad++ to explore smooth zooming for navigation and to serve as a platform for multimedia authoring and information visualization. The ability to work with very large datasets has been a primary design consideration in the development of Pad++.

INTRODUCTION

The motivation for our work on multiscale interfaces is to move beyond windows, icons, menus, and pointers to explore interfaces that more effectively support navigation through complex sets of multimedia information [1, 2, 5]. One premise of our work is the belief that navigation in information spaces is best supported by tapping into our natural spatial and geographic ways of thinking [3].

Information can be entered interactively by the user or dynamically by a program on Pad++. It can be placed at any location and at arbitrary scale. Navigation is performed by zooming in and out and moving around the surface. Our metaphor for searching for information is that all the information is there and to see more detail, you just have to *take a closer look*. Using Pad is meant to feel like you have a gigantic piece of paper that you can look at with a microscope with arbitrary resolving power and smooth zooming.

RECENT ADVANCES

Published in CHI '94: short paper category

Our focus in this implementation has been to provide smooth zooming in a system that works with very large datasets. The nature of the Pad++ interface requires consistent high frame-rate interactions, even as the dataspace becomes large and the scene gets complicated. In many applications, speed is important, but not critical to functionality. In Pad++, however, the interface paradigm is inherently based on interaction. The searching strategy is to visually explore the dataspace, so it is essential that interactive frame rates be maintained.

We implemented Pad++ in C++ on Silicon Graphics computers using SGI's graphics language facilities (the GL) to provide smooth zooming. We wrote a standard set of drawing widgets to enter text, lines, squiggles, etc., to load files and images, and a standard set of operators such as grouping, copying, deleting, etc. These give a multimedia author the basic tools for creating a dataspace.

In order to keep the animation frame-rate up as the dataspace size and complexity increases, we implemented several standard efficiency methods, which taken together create a powerful system. We are able to load in over 600,000 objects and maintain interactive rates. Briefly, these methods are:

- **Spatial Indexing:** Create a hierarchy of objects based on bounding boxes to quickly index to visible objects.
- **Spatial Level-Of-Detail:** Render only the detail needed, do not spend time rendering what can not be seen.
- **Clipping:** Only render the portions of objects that are actually visible.
- **Refinement:** Render fast with low resolution while navigating and refine the image when still.
- **Adaptive Render Scheduling:** Keep the zooming rate constant even as the frame rate changes.

One challenge in navigating through any large dataspace is maintaining a sense of relationship between what you are looking at and where it is with respect to the rest of the data. The rough animation or jumpy zooming as implemented in the original Pad [4] can be disorienting and thus not provide the most effective support for the cognitive and perceptual processing required for interactive information visualization and navigation. We implemented smooth zooming where

